

BAB I Pendahuluan

A. Latar Belakang

Komunitas alumni merupakan aset berharga bagi almamater, berfungsi sebagai jaringan pendukung, sumber inspirasi, dan bukti nyata dari keberhasilan pendidikan. Bagi SMAN 1 Pernalang, ikatan yang kuat di antara ribuan alumni yang tersebar di berbagai wilayah dan profesi adalah potensi besar yang perlu dikelola secara optimal. Namun, seiring berjalannya waktu, muncul tantangan signifikan dalam menjaga konektivitas dan mengelola data komunitas ini secara efektif. Permasalahan utama yang melatarbelakangi kebutuhan akan sebuah sistem terintegrasi adalah data alumni yang tersebar, tidak akurat, dan sulit diperbarui. Pendataan yang masih mengandalkan metode konvensional menyebabkan informasi penting mengenai kontak, domisili, dan karier alumni menjadi usang. Akibatnya, terjalin jurang komunikasi baik antar angkatan alumni maupun antara alumni dengan pihak almamater. Informasi krusial seperti agenda reuni, berita duka, atau pencapaian alumni seringkali tidak tersampaikan secara merata, sehingga menghambat partisipasi dan mengurangi rasa keterlibatan.

Kondisi ini secara langsung berdampak pada melemahnya jaringan dan solidaritas alumni. Potensi untuk kolaborasi profesional, mentoring, maupun kegiatan sosial menjadi hilang karena tidak adanya wadah digital yang terpusat dan terpercaya. Komunikasi yang terbatas dan data yang tidak terkelola dengan baik membuat komunitas alumni SMAN 1 Pernalang belum dapat berfungsi sebagai sebuah ekosistem yang solid dan berkelanjutan. Oleh karena itu, pengembangan Sistem Informasi Alumni SMAN 1 Pernalang

(Ganissa) diinisiasi sebagai solusi strategis untuk mengatasi tantangan tersebut. Ganissa dirancang untuk mentransformasi metode komunikasi dan manajemen data yang konvensional menjadi sebuah platform digital yang terpusat, interaktif, dan mudah diakses. Proyek ini bertujuan untuk membangun kembali dan memperkuat koneksi dengan menyediakan "rumah digital" bagi seluruh alumni, di mana mereka dapat terhubung, berbagi informasi, dan berpartisipasi aktif dalam kegiatan komunitas, sehingga pada akhirnya dapat mempererat rasa kekeluargaan serta melestarikan warisan dan ikatan emosional terhadap almamater.

Dengan hadirnya Ganissa, diharapkan seluruh alumni dapat merasakan kembali semangat kebersamaan yang dulu tumbuh di bangku sekolah, namun kini hadir dalam bentuk yang lebih modern dan relevan. Lebih dari sekadar database, Ganissa adalah jembatan lintas generasi yang memungkinkan setiap alumni untuk tetap terhubung, saling mendukung, dan berkontribusi bagi kemajuan bersama. Inilah langkah awal menuju komunitas alumni yang inklusif, aktif, dan berdampak nyata bagi almamater tercinta.

B. Project Charter

Tabel 1. 1 Project Charter

1. Informasi Umum Project	
Nama Project	Sistem Informasi Alumni Ganissa Peralang (SMA N 1 Peralang)
Nama Mitra	Alumni Ganissa Peralang
Manfaat Project	Mentransformasi metode komunikasi dan manajemen data komunitas alumni dari konvensional menjadi sebuah platform digital yang terpusat, interaktif, dan mudah diakses. Hal ini bertujuan untuk membangun kembali dan memperkuat

koneksi dengan menyediakan "rumah digital" bagi seluruh alumni, di mana mereka dapat terhubung, berbagi informasi, dan berpartisipasi aktif dalam kegiatan komunitas. Pada akhirnya, proyek ini akan mempererat rasa kekeluargaan serta melestarikan warisan dan ikatan emosional terhadap almamater.

2. Tim Project

	Name	Jobdesk	NO Hp	E-mail
Project Manager	Noval Lias Ramadani	Project Manager, Front End 1, Back End 2	0812681819 97	2200018083@webmail.uad.ac.id
Keanggotaan				
1	Hasan Nur Rasyid	Back End 1, Sekretaris		
2	Andriansyah	System Analist		
3	Farqad Assanji	UI/UX, Bendahara		
4	Destyawan Satria Nugraha	Tester/QA		

3. Stakeholders

Admin : Pihak yang memiliki wewenang tertinggi di dalam sistem. Stakeholder ini bertanggung jawab atas pengelolaan keseluruhan konten (seperti blog, acara, dan pengumuman) serta manajemen data pengguna alumni yang terdaftar. Peran ini kemungkinan besar dipegang oleh pengurus inti organisasi alumni

Alumni : pengguna utama dan target utama dari sistem ini. Sebagai pengguna terdaftar, mereka memiliki akses ke fitur-fitur interaktif seperti mengelola profil pribadi, mendaftar pada suatu kegiatan, serta memberikan komentar pada konten yang ada untuk membangun jaringan

Pengunjung : Pengguna tidak terdaftar atau masyarakat umum yang memiliki akses terbatas. Peran mereka adalah sebagai audiens pasif yang dapat melihat informasi publik yang dibagikan di platform, seperti berita, pengumuman, atau informasi acara

4. Project Scope

Project Purpose

Pengembangan Sistem Informasi Alumni SMAN 1 Pemalang, yang disebut 'Ganissa', dilatarbelakangi oleh kebutuhan untuk menciptakan ekosistem digital yang terstruktur dan berkelanjutan bagi komunitas alumni. Data alumni yang tersebar, tidak ter-update, dan komunikasi yang terbatas menghambat terbentuknya jaringan yang solid dan hilangnya potensi kolaborasi. Ganissa dirancang sebagai platform yang hidup dan dinamis untuk mentransformasi interaksi dan pengelolaan informasi komunitas alumni SMAN 1 Pemalang.

Scope Definition

Ruang lingkup proyek ini mendefinisikan batasan fungsional dari sistem Ganissa, yang secara spesifik ditentukan oleh interaksi dan kapabilitas masing-masing aktor yang terlibat.

Ruang Lingkup Fungsional :

1. Aktor Admin : Sebagai pengelola utama, Admin memiliki wewenang penuh atas operasional sistem. Kapabilitasnya meliputi :
 - a. Melakukan autentikasi untuk mengakses panel administrasi sistem.
 - b. Mengelola data dan akun seluruh pengguna yang terdaftar sebagai Alumni.
 - c. Memiliki hak akses penuh untuk membuat, membaca, memperbarui, dan menghapus (CRUD) seluruh konten dinamis, yang mencakup artikel blog, informasi acara (events), dan pengumuman resmi.
2. Aktor Alumni : Sebagai anggota inti komunitas, Alumni mendapatkan akses ke fitur-fitur interaktif setelah terautentikasi. Kemampuan mereka mencakup :
 - a. Melakukan pendaftaran akun baru secara mandiri dan melakukan login untuk masuk ke dalam sistem.
 - b. Mengelola informasi pada profil pribadi mereka, seperti data kontak, pekerjaan, dan informasi relevan lainnya.
 - c. Melihat daftar acara yang akan datang dan melakukan registrasi partisipasi secara langsung melalui sistem.
 - d. Membaca artikel yang dipublikasikan di blog dan berpartisipasi dalam diskusi dengan cara memberikan komentar.
 - e. Melihat pengumuman penting serta memiliki opsi untuk mengunduh atau membagikan informasi tersebut.
3. Aktor Pengunjung (Guest) : Aktor ini merepresentasikan publik atau alumni yang belum login. Akses mereka bersifat pasif dan terbatas untuk:
 - a. Melihat konten-konten publik yang telah disetujui Admin, seperti informasi umum tentang web alumni, artikel blog, daftar acara, dan pengumuman.

- b. Mengakses halaman yang berisi informasi kontak pengelola alumni.

Objectives

Tujuan fundamental dari pengembangan sistem ini adalah :

1. Membangun Basis Data Terpusat dan Akurat : Menciptakan satu sumber data utama (single source of truth) untuk seluruh alumni SMAN 1 Pematang Jaya melalui mekanisme registrasi mandiri dan pengelolaan profil oleh setiap alumni, mengatasi masalah data usang dan tidak valid.
2. Menciptakan Saluran Komunikasi Resmi: Menjadi medium komunikasi yang efisien dan terpercaya dari pengelola alumni kepada seluruh anggota melalui fitur pengumuman dan blog, memastikan informasi penting tersampaikan secara merata.
3. Mendorong Keterlibatan dan Partisipasi Aktif : Memfasilitasi dan meningkatkan keterlibatan alumni dalam berbagai kegiatan komunitas dengan menyediakan informasi acara yang terstruktur dan kemudahan untuk mendaftar secara online, serta mendorong diskusi melalui fitur komentar pada blog.
4. Memperkuat Jaringan dan Solidaritas Alumni: Menjadi wadah digital yang memungkinkan alumni untuk tetap terhubung, berbagi informasi, dan membangun jaringan profesional maupun sosial yang kuat, sehingga mempererat rasa solidaritas dan kekeluargaan di antara alumni SMAN 1 Pematang Jaya.

Deliverables

1. **Platform Web Ganissa** : Sebuah aplikasi web yang dapat diakses melalui peramban web (browser) dari berbagai perangkat (desktop, tablet, mobile) dengan desain responsif.
2. **Basis Data Alumni Terpusat** : Sistem database yang terstruktur dan aman untuk menyimpan seluruh data alumni (profil, informasi kontak, riwayat pendidikan, karier), data artikel blog, detail acara, serta pengumuman.
3. **Modul Registrasi dan Autentikasi Pengguna** :
 - a. Fungsionalitas registrasi akun baru secara mandiri bagi alumni.
 - b. Sistem *login* dan *logout* yang aman untuk setiap aktor (Admin dan Alumni).
4. **Modul Pengelolaan Profil Alumni**:
 - a. Alumni dapat melihat dan memperbarui informasi pribadi mereka (kontak, pekerjaan, domisili, angkatan, dll.).
 - b. Admin memiliki kemampuan untuk melihat, menambah, memperbarui, dan menghapus (CRUD) data profil alumni.
5. **Modul Artikel Blog**:
 - a. Admin dapat membuat, membaca, memperbarui, dan menghapus (CRUD) artikel blog.

- b. Alumni dan Pengunjung dapat membaca artikel blog.
- c. Alumni dapat memberikan komentar pada artikel blog.

6. Modul Informasi Acara (Events):

- a. Admin dapat membuat, membaca, memperbarui, dan menghapus (CRUD) informasi acara (nama, deskripsi, tanggal, waktu, lokasi, kuota).
- b. Alumni dan Pengunjung dapat melihat daftar acara yang akan datang.
- c. Alumni dapat melakukan registrasi partisipasi pada acara secara online.

7. Modul Pengumuman Resmi:

- a. Admin dapat membuat, membaca, memperbarui, dan menghapus (CRUD) pengumuman resmi.
- b. Alumni dan Pengunjung dapat melihat pengumuman penting.
- c. Pengumuman dapat memiliki opsi untuk diunduh atau dibagikan.

8. Antarmuka Admin Panel: Dashboard khusus untuk Admin dengan fitur manajemen pengguna, konten, dan laporan sederhana.

9. Laporan Dasar : Laporan mengenai jumlah alumni terdaftar dan jumlah partisipan acara.

10. Dokumentasi Proyek : Meliputi dokumen desain sistem, panduan penggunaan untuk Admin, dan panduan dasar untuk Alumni.

Project Milestones

No	Milestone	Deskripsi	Tanggal Target
1	Inisiasi Proyek	Penetapan tujuan proyek, pembentukan tim, dan pengumpulan kebutuhan awal dari stakeholder.	27 Mei 2025
2	Analisis Kebutuhan	Pengumpulan dan analisis kebutuhan fungsional dan non-fungsional sistem dari pengguna (alumni dan pengurus).	28 Mei 2025
3	Perancangan Sistem (Desain UI/UX & Database)	Pembuatan desain antarmuka pengguna (UI), perancangan alur sistem, dan desain database.	30 Mei 2025
4	Pembangunan Sistem (Development)	Proses pengembangan sistem informasi berdasarkan desain yang telah dibuat (frontend dan backend).	06 Juni 2025

5	Pengujian Sistem (Testing)	Uji coba sistem secara fungsional dan non-fungsional oleh tim pengembang dan beberapa alumni sebagai tester.	28 Juni 2025
6	Revisi dan Perbaikan	Perbaiki bug dan penyesuaian sistem berdasarkan hasil pengujian.	06 Juli 2025
7	Pelatihan dan Sosialisasi	Pelatihan kepada admin dan sosialisasi penggunaan sistem kepada alumni.	23 Juli 2025
8	Implementasi dan Peluncuran Sistem	Sistem mulai digunakan secara resmi oleh komunitas alumni Ganissa Pemalang.	30 Juli 2025
9	Evaluasi dan Pemeliharaan Awal	Evaluasi awal penggunaan sistem dan perbaikan kecil jika diperlukan.	30 Agustus 2025

Major Known Risks

No	Risiko Utama	Penjelasan	Dampak	Strategi Mitigasi
1	Kurangnya Partisipasi Alumni	Alumni tidak aktif memberikan data atau berpartisipasi dalam pengisian profil dan penggunaan sistem.	Tinggi	<ul style="list-style-type: none"> - Adakan sosialisidan kampanye digital aktif. - Libatkan tokoh alumni sebagai duta promosi sistem.
2	Data Tidak Akurat atau Tidak Terverifikasi	Data alumni yang diinput bisa tidak lengkap, tidak valid, atau palsu.	Sedang–Tinggi	<ul style="list-style-type: none"> - Terapkan sistem verifikasi oleh admin - Validasi data dengan dokumen pendukung seperti ijazah.
3	Keterbatasan Sumber Daya Pengembang	Tim pengembang terdiri dari relawan atau mahasiswa yang mungkin memiliki keterbatasan waktu dan pengalaman.	Sedang	<ul style="list-style-type: none"> - Buat timeline realistis dan prioritaskan fitur inti .- Menerapkan agile development dengan iterasi bertahap.
4	Masalah Keamanan	Data pribadi alumni bisa disalahgunakan jika sistem tidak aman.	Tinggi	<ul style="list-style-type: none"> - Gunakan enkripsi password dan proteksi akses.

	dan Privasi Data			- Terapkan kebijakan privasi dan akses berbasis peran (role-based access).
5	Keterbatasan Anggaran dan Infrastruktur	Keterbatasan dana untuk hosting, domain, atau server berdampak pada performa sistem.	Sedang	- Gunakan hosting gratis atau murah seperti GitHub Pages + Firebase. - Ajukan sponsor atau donasi dari alumni.
6	Kesulitan Penggunaan oleh Pengguna Awam	Tidak semua alumni familiar dengan sistem digital.	Sedang	- Rancang UI yang sederhana dan intuitif. - Buat panduan dan video tutorial penggunaan.
7	Downtime Server atau Gangguan Teknis	Sistem tidak bisa diakses karena gangguan teknis atau server overload.	Sedang	- Gunakan layanan hosting terpercaya. - Lakukan monitoring dan backup rutin.

Constraints

No	Keterbatasan (Constraint)	Penjelasan
1	Waktu Pengembangan Terbatas	Proyek harus diselesaikan dalam jangka waktu tertentu (misalnya 3 bulan), sehingga pengembangan harus efisien dan tepat waktu.
2	Anggaran Terbatas	Tidak ada atau sangat sedikit dana untuk pengembangan, sehingga perlu memanfaatkan sumber daya gratis atau open-source.
3	Sumber Daya Manusia Terbatas	Tim pengembang berasal dari internal (mahasiswa/pengurus) dengan keterbatasan waktu, pengalaman, dan jumlah personel.
4	Akses Data Awal Terbatas	Basis data alumni mungkin belum lengkap atau belum terorganisir, sehingga input awal tergantung pada partisipasi manual.
5	Keterbatasan Infrastruktur Teknis	Hosting dan server yang digunakan bersifat terbatas (misalnya hanya menggunakan shared hosting atau platform gratis).

6	Kompetensi Teknologi Pengguna	Tidak semua alumni memiliki kemampuan digital yang baik, sehingga sistem harus sangat mudah digunakan (user-friendly).
7	Komitmen Stakeholder	Keterlibatan stakeholder (pengurus alumni, sekolah, alumni senior) bisa terbatas karena kesibukan masing-masing.
8	Keterbatasan Regulasi dan Legalitas	Sistem perlu mematuhi aturan privasi data dan penggunaan informasi pribadi, meskipun belum ada pedoman formal dari lembaga alumni.

BAB II Perencanaan Proyek

A. Analisis Kelayakan

Analisis kelayakan berupa SWOT (*Strenghts, Weakness, Opportunities, Threats*) dalam proyek sistem informasi website alumni ganissa pada Tabel 2.1.

Tabel 2. 1 Analisis SWOT

Strenghts	<ol style="list-style-type: none">1. Menyediakan layanan manajemen data alumni yang cepat, mudah, dan aman.2. Menyajikan informasi alumni secara terstruktur dan real-time, termasuk data pendidikan, pekerjaan, dan kontak.3. Kompatibilitas sistem Ganissa yang berbasis web untuk berbagai perangkat.4. Fitur notifikasi pengingat (misalnya event reuni, update profil) memudahkan keterlibatan pengguna.5. Sistem berbasis web dengan autentikasi pengguna sehingga fleksibel diakses dari mana saja.
Weakness	<ol style="list-style-type: none">1. Ketergantungan pada SDM teknis dapat menjadi kendala jika terjadi gangguan sistem.2. Masih terbatasnya data alumni yang lengkap dan tervalidasi.3. Kemungkinan kendala teknis seperti bug atau crash dapat mengganggu kenyamanan pengguna.4. Kurangnya pelatihan rutin bagi operator sekolah atau admin sistem dalam pengelolaan platform.5. Fitur analisis data alumni belum dioptimalkan untuk kebutuhan sekolah dan organisasi alumni.
Opportunities	<ol style="list-style-type: none">1. Berpeluang menjalin kemitraan dengan instansi pendidikan, perguruan tinggi, dan dunia kerja untuk kolaborasi tracer study dan lowongan kerja alumni.2. Dukungan pemerintah dalam program digitalisasi data pendidikan dan alumni.

	<ol style="list-style-type: none"> 3. Potensi integrasi dengan e-sertifikat, atau event alumni nasional. 4. Meningkatnya tren digitalisasi sekolah di seluruh Indonesia. 5. Peluang menarik minat sekolah lain di wilayah Pematang untuk mengadopsi Ganissa sebagai sistem alumni standar.
Threats	<ol style="list-style-type: none"> 1. Ancaman terhadap keamanan data pribadi alumni (seperti nama, alamat, pekerjaan, dll). 2. Adanya sistem alumni lain yang lebih lengkap dan kompetitif. 3. Regulasi baru tentang privasi dan pengelolaan data pendidikan dari pemerintah. 4. Ketergantungan pada koneksi internet bisa menjadi kendala di wilayah dengan jaringan terbatas. 5. Potensi kesalahan input data yang menyebabkan ketidaktepatan informasi alumni.

Strategi berdasarkan analisis SWOT bisa dilihat pada Tabel 2.2.

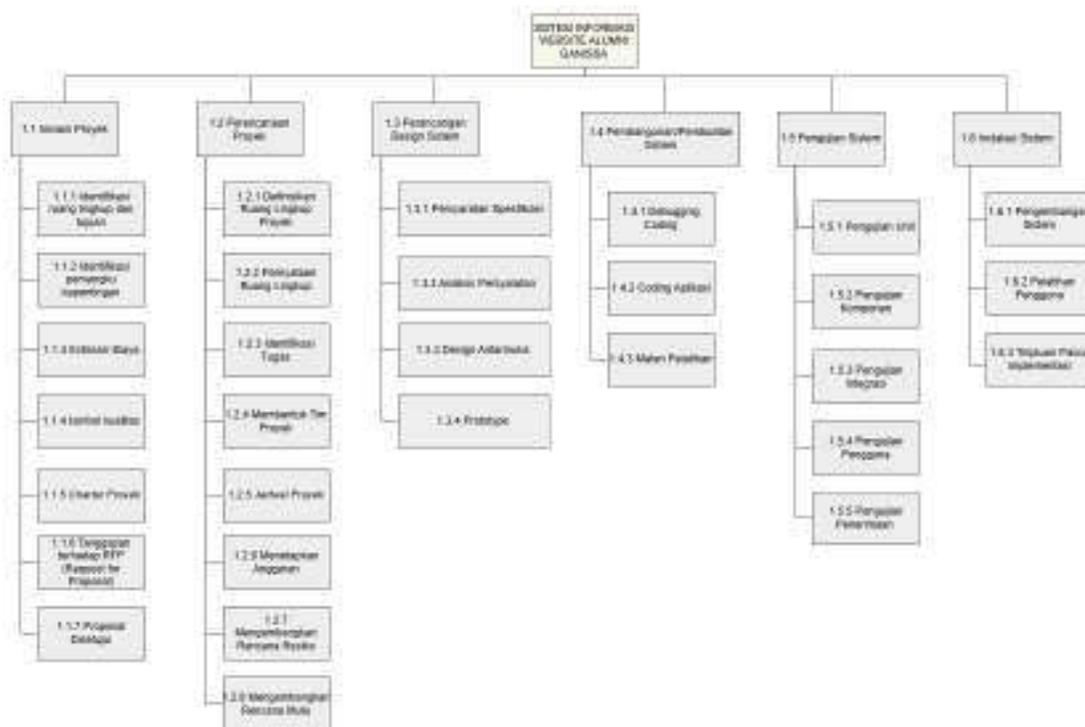
Tabel 2. 2 Analisis Strategi Berdasarkan SWOT

Strategi S-O (Strength – Opportunites)	<ol style="list-style-type: none"> 1. Menambahkan modul tracer study alumni berdasarkan data pekerjaan mereka untuk mendukung kebutuhan sekolah dan dinas pendidikan. 2. Menyediakan fitur direktori alumni dengan filter berdasarkan angkatan, profesi, atau lokasi. 3. Menyediakan dashboard statistik alumni secara visual (grafik profesi, domisili, pendidikan). 4. Menawarkan katalog profil alumni inspiratif yang bisa diakses publik untuk memperkuat citra sekolah. 5. Membuka akses integrasi dengan layanan pihak ketiga (misalnya job portal, universitas, e-learning).
--	---

Strategi W-O (Weakness - Opportunities)	<ol style="list-style-type: none"> 1. Mengembangkan fitur pengumpulan data offline (formulir input manual) yang bisa diunggah saat koneksi tersedia. 2. Menyediakan pelatihan digital literasi dan manajemen sistem bagi admin/operator sekolah. 3. Membangun fitur validasi ganda untuk data alumni (verifikasi email/WA). 4. Menyediakan dukungan multi-sekolah jika ingin direplikasi ke SMA lain di Pematang Jaya. 5. Menambahkan integrasi form Google/Excel API untuk percepatan input data awal.
Strategi S-T (Strength - Threats)	<ol style="list-style-type: none"> 1. Menerapkan teknologi enkripsi dan two-factor authentication (2FA) untuk menjaga keamanan data alumni. 2. Menyediakan log aktivitas admin dan pengguna untuk pelacakan jika terjadi pelanggaran. 3. Menyediakan fitur unggulan seperti sertifikat digital keaktifan alumni, sistem voting internal (misal pemilihan ketua alumni). 4. Berkoordinasi dengan dinas pendidikan untuk menjaga agar sistem sesuai standar operasional dan kebijakan pemerintah.
Strategi W-T (Weakness - Threats)	<ol style="list-style-type: none"> 1. Menyusun SOP (Standard Operating Procedure) pengelolaan risiko seperti kesalahan input, gangguan server, dan backup data otomatis. 2. Membuat modul disaster recovery untuk pemulihan sistem cepat saat terjadi kendala. 3. Menyediakan dokumentasi lengkap dan panduan penggunaan sistem untuk sekolah dan alumni.

	<p>4. Melakukan pengujian berkala terhadap fitur utama dan menerima masukan dari pengguna melalui feedback form.</p> <p>5. Melakukan sosialisasi intensif kepada alumni agar mereka aktif memperbarui data mandiri.</p>
--	---

B. Work Breakdown Structure



Gambar 2. 1 Work Breakdown Structure

1. Inisiasi Proyek

Tahap ini adalah fondasi awal proyek, di mana ide proyek diresmikan dan kelayakannya dieksplorasi.

1.1 Identifikasi ruang lingkup dan tujuan: Pada sub-bagian ini, tim proyek akan secara jelas mendefinisikan batas-batas proyek. Untuk "Sistem Informasi Alumni SMAN 1

Pemalang," ruang lingkungannya mungkin mencakup fitur-fitur utama seperti pendaftaran alumni, profil alumni (data pribadi, pendidikan, karier), direktori alumni, modul berita/blog, modul acara, dan modul pengumuman. Tujuannya bisa jadi untuk mempermudah komunikasi antar alumni, menyediakan platform informasi, memfasilitasi *networking*, dan mendukung kegiatan sekolah/alumni.

1.2 Identifikasi pemangku kepentingan: Ini melibatkan pengenalan semua individu atau kelompok yang akan terpengaruh oleh proyek atau memiliki kepentingan dalam keberhasilannya. Untuk proyek ini, pemangku kepentingan utama meliputi: manajemen SMAN 1 Pemalang, staf IT sekolah, alumni SMAN 1 Pemalang (sebagai pengguna akhir), calon alumni (siswa kelas akhir), komite alumni, dan pengembang/tim proyek itu sendiri.

1.3 Estimasi Biaya: Menentukan perkiraan sumber daya finansial yang dibutuhkan untuk menyelesaikan proyek. Ini mencakup perkiraan biaya pengembangan (upah tim, lisensi *software* jika ada), infrastruktur (server, domain), pemeliharaan, pelatihan, dan biaya tak terduga.

1.4 Kontrol kualitas: Mendefinisikan standar dan prosedur untuk memastikan bahwa hasil proyek memenuhi persyaratan yang ditetapkan. Ini bisa berupa metrik kualitas untuk *software* (misalnya, jumlah *bug* yang diizinkan), kecepatan respons sistem, atau kepuasan pengguna.

1.5 Charter Proyek: Dokumen formal yang secara resmi mengotorisasi keberadaan proyek dan memberikan manajer proyek wewenang untuk menerapkan sumber daya

organisasi untuk kegiatan proyek. Ini akan merangkum ruang lingkup, tujuan, pemangku kepentingan, anggaran estimasi, dan daftar tim inti.

1.6 Tanggapan terhadap RFP (Request for Proposal): Jika proyek ini merupakan hasil dari permintaan pihak eksternal, tahap ini melibatkan penyusunan proposal rinci yang menjelaskan bagaimana tim proyek akan memenuhi kebutuhan yang digariskan dalam RFP, termasuk metodologi, jadwal, dan biaya.

1.7 Proposal Disetujui: Ini adalah titik balik penting di mana semua pemangku kepentingan kunci menyetujui rencana proyek yang diajukan, memberikan lampu hijau untuk melanjutkan ke tahap berikutnya.

2. Perencanaan Proyek

Tahap ini berfokus pada pengembangan peta jalan yang rinci untuk mencapai tujuan proyek.

2.1 Definisikan Ruang Lingkup Proyek: Memperjelas dan mendokumentasikan tujuan proyek, hasil yang dapat disampaikan, dan batasan-batasannya, seringkali dalam dokumen *Statement of Work (SOW)* atau dokumen ruang lingkup proyek. Ini lebih detail dari pada inisiasi.

2.2 Pernyataan Ruang Lingkup: Dokumen tertulis yang sangat detail mengenai apa yang akan dan tidak akan dilakukan dalam proyek. Ini berfungsi sebagai landasan untuk menghindari *scope creep*.

2.3 Identifikasi Tugas: Memecah ruang lingkup proyek menjadi tugas-tugas yang lebih kecil dan terkelola. Ini bisa menggunakan *Work Breakdown Structure (WBS)* untuk

memetakan semua pekerjaan yang perlu dilakukan. Contoh: "Desain database", "Buat modul profil", "Uji coba login".

2.4 Membentuk Tim Proyek: Mengidentifikasi dan mengalokasikan peran serta tanggung jawab untuk setiap anggota tim. Untuk sistem informasi, ini bisa termasuk *Project Manager, Business Analyst, UI/UX Designer, Backend Developer, Frontend Developer, Quality Assurance Tester*, dll.

2.5 Jadwal Proyek: Membuat *timeline* proyek yang merinci kapan setiap tugas akan dimulai dan selesai. Ini bisa menggunakan Gantt chart atau metode *Critical Path Method (CPM)*.

2.6 Menetapkan Anggaran: Mengembangkan anggaran yang terperinci berdasarkan estimasi biaya dan jadwal, mengalokasikan dana untuk setiap tugas atau fase proyek.

2.7 Mengembangkan Rencana Risiko: Mengidentifikasi potensi risiko (teknis, operasional, finansial, keamanan data, dll.) yang dapat memengaruhi proyek dan mengembangkan strategi mitigasi untuk setiap risiko.

2.8 Mengembangkan Rencana Mutu: Mendetailkan bagaimana kualitas akan dikelola di sepanjang proyek, termasuk standar yang akan dipatuhi, proses pengujian, dan bagaimana *feedback* akan diintegrasikan.

3. Perancangan (Desain Sistem)

Fase ini mengubah persyaratan menjadi rencana arsitektur dan desain yang lebih konkret.

3.1 Persyaratan Spesifikasi: Menguraikan semua fungsionalitas dan non-fungsionalitas yang diharapkan dari sistem secara terperinci. Ini adalah dokumen yang sangat detail tentang apa yang harus dilakukan sistem. Contoh: "Sistem harus dapat memverifikasi alumni menggunakan NIS," "Sistem harus dapat menampilkan profil alumni secara publik dengan persetujuan alumni."

3.2 Analisis Persyaratan: Menganalisis persyaratan yang dikumpulkan untuk memastikan konsistensi, kelengkapan, dan kelayakan. Ini mungkin melibatkan pembuatan diagram alur, diagram kasus penggunaan (UML), atau cerita pengguna (*user stories*) untuk memvisualisasikan bagaimana pengguna akan berinteraksi dengan sistem.

3.3 Desain Antarmuka: Membuat sketsa, *wireframes*, dan *mockups* visual tentang bagaimana antarmuka pengguna (UI) sistem akan terlihat dan berfungsi. Ini mencakup tata letak halaman, elemen navigasi, skema warna, dan tipografi.

3.4 Prototype: Membuat versi awal atau model fungsional dari sistem (bisa berupa *low-fidelity* atau *high-fidelity*) untuk mendapatkan *feedback* awal dari pemangku kepentingan dan menguji konsep desain sebelum pengembangan penuh.

4. Pembangunan / Pembuatan Sistem

Ini adalah fase di mana kode dan komponen sistem sebenarnya dibuat.

4.1 Debugging Coding: Proses mengidentifikasi, menganalisis, dan memperbaiki kesalahan (bug) dalam kode program. Ini dilakukan secara iteratif sepanjang proses *coding*.

4.2 Coding Aplikasi: Penulisan kode sumber untuk semua modul dan fitur sistem berdasarkan desain yang telah disetujui. Ini mencakup pengembangan *backend* (API, logika bisnis, database) dan *frontend* (antarmuka pengguna interaktif).

4.3 Materi Pelatihan: Menyiapkan panduan, manual, atau modul pelatihan untuk pengguna akhir (alumni dan admin sekolah) tentang cara menggunakan sistem setelah diimplementasikan.

5. Pengujian Sistem

Fase ini bertujuan untuk menemukan *bug* dan memastikan sistem memenuhi semua persyaratan fungsional dan non-fungsional.

5.1 Pengujian Unit: Menguji komponen individual atau unit kode secara terpisah untuk memastikan mereka berfungsi dengan benar. Dilakukan oleh pengembang.

5.2 Pengujian Komponen: Menguji sekelompok unit yang terintegrasi (misalnya, satu modul penuh) untuk memastikan interaksi antar unit berfungsi.

5.3 Pengujian Integrasi: Menguji interaksi antara berbagai modul atau sub-sistem yang berbeda untuk memastikan mereka bekerja sama dengan mulus. Contoh: Apakah registrasi event berhasil dicatat di profil alumni?

5.4 Pengujian Pengguna (UAT - User Acceptance Testing): Pengguna akhir (perwakilan alumni atau staf sekolah) menguji sistem dalam lingkungan yang realistis untuk memvalidasi bahwa sistem memenuhi kebutuhan bisnis dan persyaratan pengguna.

5.5 Pengujian Penerimaan: Tahap akhir pengujian di mana pemangku kepentingan utama secara formal menerima sistem setelah semua *bug* kritis diperbaiki dan persyaratan terpenuhi.

6. Instalasi Sistem

Fase terakhir proyek, di mana sistem disebarkan dan mulai digunakan.

6.1 Pengembangan Sistem (Deployment): Proses menginstal dan mengonfigurasi sistem di lingkungan produksi (server live), membuatnya tersedia untuk pengguna akhir. Ini bisa melibatkan migrasi data lama jika ada.

6.2 Pelatihan Pengguna: Melakukan sesi pelatihan untuk alumni dan admin sekolah agar mereka terbiasa dan mahir menggunakan sistem informasi alumni yang baru.

6.3 Tinjauan Pasca Implementasi: Mengevaluasi keberhasilan proyek setelah sistem berjalan selama beberapa waktu, mengidentifikasi pelajaran yang dipetik, dan mengukur apakah tujuan proyek awal telah tercapai. Ini juga bisa menjadi awal dari fase pemeliharaan dan peningkatan berkelanjutan.

C. Kebutuhan Sumber Daya

1. Sumber Daya Manusia

Tabel 2. 3 Analisis Sumber Daya Manusia

Jobdesk	Nama	Keterangan
Project Manager	Noval Lias Ramadani	<ol style="list-style-type: none">1. Bertanggung jawab atas keseluruhan perencanaan, pelaksanaan, pengawasan, dan penyelesaian proyek.2. Menentukan ruang lingkup proyek, jadwal, dan anggaran.3. Menjadi penghubung utama antara tim proyek dan pihak stakeholder.4. Mengelola risiko proyek dan membuat keputusan strategis untuk kelancaran proyek.5. Memastikan seluruh tim bekerja sesuai timeline dan tujuan proyek tercapai tepat waktu.
System Analyst	Andriansyah	<ol style="list-style-type: none">1. Menganalisis kebutuhan sistem dan merancang solusi teknis yang sesuai.2. Mengumpulkan dan mendokumentasikan kebutuhan pengguna (requirement gathering).3. Membuat dokumentasi seperti Use Case, DFD, ERD, dan dokumen spesifikasi sistem lainnya.4. Berkolaborasi dengan UI/UX dan developer untuk memastikan desain sistem dapat diimplementasikan dengan baik.

UI/UX	Farqad Assanji	<ol style="list-style-type: none"> 1. Mendesain tampilan antarmuka pengguna (UI) yang menarik, intuitif, dan sesuai dengan kebutuhan pengguna. 2. Melakukan riset pengalaman pengguna (UX) untuk memastikan sistem mudah digunakan. 3. Membuat prototipe (mockup dan wireframe) serta desain visual yang siap dikembangkan. 4. Berkoordinasi erat dengan frontend developer untuk memastikan implementasi sesuai desain.
Frontend 1	Noval Lias Ramadani	<ol style="list-style-type: none"> 1. Mengembangkan bagian antarmuka pengguna aplikasi sesuai desain dari UI/UX. 2. Mengimplementasikan logika tampilan menggunakan bahasa pemrograman menggunakan bahasa pemrograman seperti HTML, CSS, dan JavaScript (atau framework terkait). 3. Menjamin responsivitas dan performa tampilan di berbagai perangkat. 4. Berkolaborasi dengan backend untuk menampilkan data secara dinamis.
Frontend 2	Farqad Assanji	<ol style="list-style-type: none"> 1. Mendukung tugas frontend 1 dalam pengembangan antarmuka pengguna. 2. Menyempurnakan elemen visual, animasi, dan interaksi pengguna. 3. Membantu pengujian tampilan lintas perangkat dan browser.

		<ol style="list-style-type: none"> 4. Berkoordinasi dengan UI/UX untuk memastikan konsistensi tampilan dan pengalaman pengguna.
Backend 1	Hasan Nur Rasyid	<ol style="list-style-type: none"> 1. Mengembangkan dan mengelola server, database, dan API sistem. 2. Menjamin integrasi data berjalan aman dan efisien. 3. Bekerja sama dengan frontend untuk penyediaan dan pengolahan data. 4. Mengimplementasikan logika bisnis sistem dan keamanan data.
Backend 2	Noval Lias Ramadani	<ol style="list-style-type: none"> 1. Mendukung tugas backend 1 dalam pengembangan server dan integrasi sistem. 2. Mengoptimalkan performa server dan pengelolaan database. 3. Membantu proses deployment backend dan manajemen infrastruktur teknis.
Quality Assurance	Destyawan Satria Nugraha	<ol style="list-style-type: none"> 1. Menguji aplikasi untuk memastikan tidak ada bug dan sesuai dengan kebutuhan pengguna. 2. Menyusun test case dan melakukan testing (manual maupun otomatis). 3. Mencatat dan melaporkan error atau ketidaksesuaian fitur kepada tim developer. 4. Menjamin kualitas sistem sebelum rilis ke pengguna akhir.

Sekretaris	Hasan Nur Rasyid	<ol style="list-style-type: none">1. Mencatat hasil rapat tim dan dokumentasi penting lainnya.2. Mengelola jadwal rapat, pembagian tugas, dan pengingat deadline.3. Menyimpan dokumen proyek secara terstruktur dan dapat diakses oleh seluruh anggota tim.4. Menjadi penghubung administratif antara tim dan pihak eksternal jika diperlukan.
------------	------------------	---

2. Sumber Daya Fisik

- a. Editor : Visual Studio Code
- b. Bahasa Pemrograman : PHP, HTML, CSS, JavaScript
- c. Framework : Laravel
- d. Libraries :
 - i. Php 8.1
 - ii. barryvdh/laravel-dompdf 3.1
 - iii. Doctrine/dbal 3.9
 - iv. Guzzlehttp/guzzle 7.9
 - v. Laravel/framework 10.0
 - vi. Laravel/tinker 2.8
 - vii. Laravel/sanctum 3.2
 - viii. Milon/barcode 12.0
 - ix. Nyholm/psr7 1.8
 - x. php-http/guzzle7-adapter 1.1

- xi. Railsware/mailtrap-php 3.1
 - xii. Simplesoftwareio/simple-qrcode 4.2
 - xiii. Symfony/http-client 7.3
 - xiv. Symfony/mailer 6.4
- e. Database : MySQL
 - f. Server : Laragon
 - g. Domain : Site
 - h. Tool Design : Figma, IbisPaint X
 - i. Browser : Microsoft Edge, Chrome

D. Rencana Jadwal Pelaksanaan Proyek

Tabel 2. 4 Rencana Jadwal Pelaksanaan Proyek

NO	Kegiatan	Juni				Juli
		1	2	3	4	1
A	Pengumpulan Data					
B	Analisis					
C	Desain Website					
D	Programming					
E	Testing					
F	Hosting					
G	Launching					

E. Rencana Nilai Proyek

Tabel 2. 5 Rencana Nilai Proyek

No.	Uraian	Volume	Jumlah
1	Jasa Pembuatan	1 Paket	Rp 500.000
2	Hosting	1 Tahun	Rp 405.000
Total (Rp)			Rp 950.000

BAB III PELAKSANAAN PROYEK

A. Realisasi Tugas dan Jadwal Pelaksanaan

1. Detail Pelaksanaan Proyek

Tabel 3. 1 Detail Pelaksaan dan Penjadwalan Proyek

Fase Proyek	Tugas Spesifik	Penanggung Jawab	Estimasi Mulai	Estimasi Selesai	Durasi (Hari)	Kategori
Fase 1: Inisiasi & Perencanaan Proyek	Identifikasi ruang lingkup dan tujuan proyek	Noval Lias R. (PM), Andriansyah (System Analyst)	2025-05-27	2025-05-27	1	Perencanaan
Fase 1: Inisiasi & Perencanaan Proyek	Identifikasi pemangku kepentingan (stakeholders)	Noval Lias R. (PM), Hasan Nur R. (Sekretaris)	2025-05-27	2025-05-27	1	Perencanaan
Fase 1: Inisiasi & Perencanaan Proyek	Estimasi Biaya & 2.6 Menetapkan Anggaran	Noval Lias R. (PM), Farqad A. (Bendahara)	2025-05-28	2025-05-28	1	Perencanaan
Fase 1: Inisiasi & Perencanaan Proyek	Membuat Project Charter	Noval Lias R. (PM), Hasan Nur R. (Sekretaris)	2025-05-28	2025-05-28	1	Perencanaan
Fase 1: Inisiasi & Perencanaan Proyek	Mengembangkan Rencana Risiko	Seluruh Tim	2025-05-29	2025-05-29	1	Perencanaan
Fase 1: Inisiasi & Perencanaan Proyek	Finalisasi & Persetujuan Proposal Proyek	Noval Lias R. (PM)	2025-05-29	2025-05-29	1	Perencanaan

Fase 2: Analisis & Perancangan Sistem	Analisis Kebutuhan Fungsional & Non- Fungsional	Andriansyah (System Analyst)	2025- 05-30	2025- 06-01	3	Desain
Fase 2: Analisis & Perancangan Sistem	Desain Antarmuka (Wireframe & Mockup)	Farqad Assanji (UI/UX)	2025- 06-01	2025- 06-04	4	Desain
Fase 2: Analisis & Perancangan Sistem	Pembuatan Prototype Fungsional	Farqad Assanji (UI/UX)	2025- 06-04	2025- 06-05	2	Desain
Fase 2: Analisis & Perancangan Sistem	Perancangan Arsitektur & Database Sistem	Andriansyah (System Analyst), Hasan Nur R. (BE 1)	2025- 05-30	2025- 06-02	4	Desain
Fase 3: Pembangunan Sistem (Development)	Pengembangan Backend (API, Logika Bisnis, Database)	Hasan Nur Rasyid (BE 1), Noval Lias R. (BE 2)	2025- 06-06	2025- 07-08	33	Pengembangan
Fase 3: Pembangunan Sistem (Development)	Pengembangan Frontend (Implementasi UI/UX & Interaktivitas)	Noval Lias R. (FE 1), Farqad Assanji (FE 2)	2025- 06-10	2025- 07-15	36	Pengembangan
Fase 3: Pembangunan Sistem (Development)	Coding Aplikasi & Debugging Berkelanjutan	Hasan Nur R., Noval Lias R., Farqad Assanji	2025- 06-06	2025- 07-15	40	Pengembangan
Fase 3: Pembangunan Sistem (Development)	Integrasi Frontend & Backend	Noval Lias R., Hasan Nur R., Farqad Assanji	2025- 07-01	2025- 07-15	15	Pengembangan
Fase 4: Pengujian & Revisi	Pengujian Unit & Komponen	Destyawan Satria N. (QA)	2025- 07-16	2025- 07-17	2	Pengujian
Fase 4: Pengujian & Revisi	Pengujian Integrasi Sistem	Destyawan Satria N. (QA)	2025- 07-18	2025- 07-19	2	Pengujian
Fase 4: Pengujian & Revisi	Pengujian Pengguna (User)	Destyawan Satria N. (QA),	2025- 07-20	2025- 07-21	2	Pengujian

	Acceptance Testing - UAT)	Perwakilan Mitra				
Fase 4: Pengujian & Revisi	Revisi dan Perbaiki Bug berdasarkan hasil pengujian	Noval Lias R., Hasan Nur R., Farqad Assanji	2025-07-16	2025-07-22	7	Pengujian
Fase 5: Implementasi & Peluncuran	Menyiapkan Materi & Dokumentasi Pelatihan	Hasan Nur R. (Sekretaris), Andriansyah (SA)	2025-07-23	2025-07-24	2	Peluncuran
Fase 5: Implementasi & Peluncuran	Pelatihan kepada Admin & Sosialisasi kepada Alumni	Noval Lias R. (PM)	2025-07-25	2025-07-27	3	Peluncuran
Fase 5: Implementasi & Peluncuran	Deployment Sistem ke Server Produksi (Hosting)	Noval Lias R. (BE 2), Hasan Nur R. (BE 1), Andriansyah (SA)	2025-07-28	2025-07-29	2	Peluncuran
Fase 5: Implementasi & Peluncuran	Peluncuran Resmi Sistem GANISSA	Seluruh Tim & Mitra	2025-07-30	2025-07-30	1	Peluncuran
Fase 6: Pasca-Peluncuran	Evaluasi & Pemeliharaan Awal	Noval Lias R. (PM), Destyawan Satria N. (QA)	2025-07-31	2025-08-30	31	Pemeliharaan
Fase 6: Pasca-Peluncuran	Tinjauan Pasca Implementasi & Laporan Akhir	Noval Lias R. (PM), Hasan Nur R. (Sekretaris)	2025-08-15	2025-08-30	16	Pemeliharaan

No. Task	Task Name	Start Date	End Date	Status	Progress	Responsible	Progress %
1	Analisis Kelayakan & Kebutuhan	27 Mei 2025	27 Mei 2025	Selesai	100%	Team Lead & PM, Team Lead	100%
2	Identifikasi Ruang Lingkup & Fitur	27 Mei 2025	27 Mei 2025	Selesai	100%	Team Lead & PM, Manajer	100%
3	Merencanakan Struktur	27 Mei 2025	27 Mei 2025	Selesai	100%	Team Lead & PM, Team Lead	100%
4	Identifikasi & Analisis Risiko	27 Mei 2025	27 Mei 2025	Selesai	100%	Team Lead & PM, Manajer	100%
5	Penyusunan & Penetapan Jadwal	27 Mei 2025	27 Mei 2025	Selesai	100%	Team Lead & PM	100%
6	Penyusunan Struktur & Definisi UI	27 Mei 2025	27 Mei 2025	Selesai	100%	Manajer	100%
7	Analisis & Penetapan UI/UX	27 Mei 2025	27 Mei 2025	Selesai	100%	Manajer	100%
8	Analisis & Penetapan UI/UX	27 Mei 2025	27 Mei 2025	Selesai	100%	Manajer	100%
9	Penyusunan & Penetapan Jadwal	27 Mei 2025	27 Mei 2025	Selesai	100%	Manajer	100%
10	Penyusunan & Penetapan Jadwal	27 Mei 2025	27 Mei 2025	Selesai	100%	Manajer	100%
11	Penyusunan & Penetapan Jadwal	27 Mei 2025	27 Mei 2025	Selesai	100%	Manajer	100%
12	Penyusunan & Penetapan Jadwal	27 Mei 2025	27 Mei 2025	Selesai	100%	Manajer	100%
13	Penyusunan & Penetapan Jadwal	27 Mei 2025	27 Mei 2025	Selesai	100%	Manajer	100%
14	Penyusunan & Penetapan Jadwal	27 Mei 2025	27 Mei 2025	Selesai	100%	Manajer	100%
15	Penyusunan & Penetapan Jadwal	27 Mei 2025	27 Mei 2025	Selesai	100%	Manajer	100%
16	Penyusunan & Penetapan Jadwal	27 Mei 2025	27 Mei 2025	Selesai	100%	Manajer	100%
17	Penyusunan & Penetapan Jadwal	27 Mei 2025	27 Mei 2025	Selesai	100%	Manajer	100%
18	Penyusunan & Penetapan Jadwal	27 Mei 2025	27 Mei 2025	Selesai	100%	Manajer	100%
19	Penyusunan & Penetapan Jadwal	27 Mei 2025	27 Mei 2025	Selesai	100%	Manajer	100%
20	Penyusunan & Penetapan Jadwal	27 Mei 2025	27 Mei 2025	Selesai	100%	Manajer	100%
21	Penyusunan & Penetapan Jadwal	27 Mei 2025	27 Mei 2025	Selesai	100%	Manajer	100%
22	Penyusunan & Penetapan Jadwal	27 Mei 2025	27 Mei 2025	Selesai	100%	Manajer	100%

Gambar 3. 1 Detail Pelaksanaan Proyek

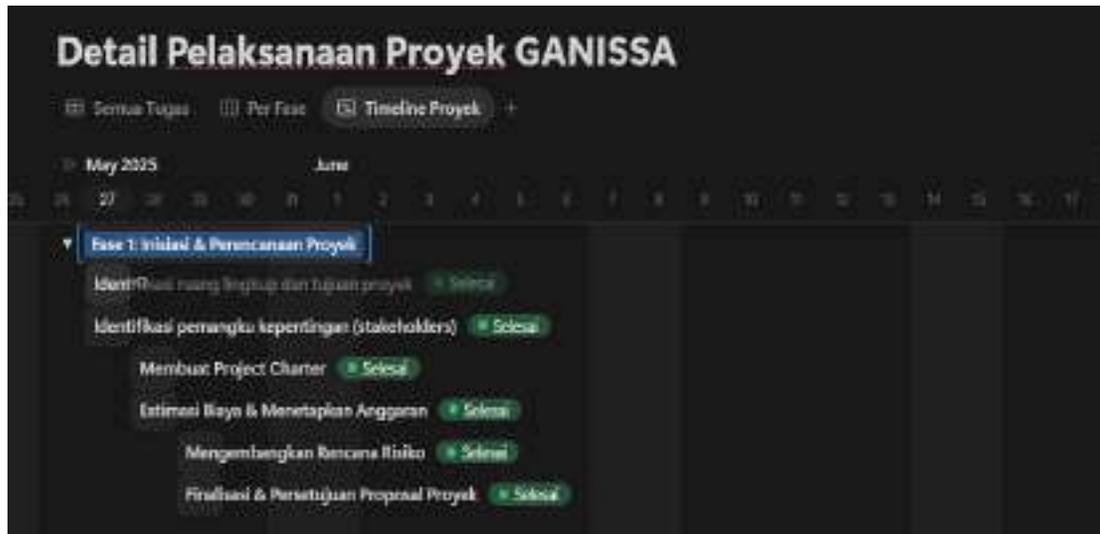
Tabel dan gambar tersebut merupakan detail rencana dan realisasi pelaksanaan untuk sebuah proyek yang dinamakan "Proyek GANISSA". Dokumen ini berfungsi sebagai alat pemantauan yang merinci setiap tugas dari awal hingga akhir proyek. Informasi yang disajikan sangat komprehensif, mencakup durasi setiap tugas dalam hari, estimasi tanggal mulai dan selesai, fase proyek, kategori pekerjaan (seperti Perencanaan, Desain, Pengembangan), serta penanggung jawab untuk setiap tugas. Berdasarkan kolom status "Selesai" dan progress "100%" untuk semua item. Tabel ini menampilkan rekam jejak sebuah proyek yang telah berhasil diselesaikan seluruhnya, yang berlangsung dari 27 Mei 2025 hingga 30 Agustus 2025.

Proyek ini dijalankan secara sistematis melalui enam fase yang jelas, mengikuti alur pengembangan sistem pada umumnya. Fase-fase tersebut adalah Fase 1: Inisiasi & Perencanaan, Fase 2: Analisis & Perancangan, Fase 3: Pembangunan Sistem, Fase 4: Pengujian & Revisi, Fase 5: Implementasi & Peluncuran, dan diakhiri dengan Fase 6: Pasca-Peluncuran. Tugas-tugas di dalamnya sangat terperinci, mulai dari estimasi biaya, perancangan arsitektur, pengembangan backend dan frontend yang memakan waktu paling lama (40 dan 38 hari), hingga pengujian pengguna (UAT), dan evaluasi setelah implementasi. Struktur ini menunjukkan pendekatan manajemen proyek yang matang dan terorganisir.

Proyek ini dikerjakan oleh sebuah tim dengan peran yang beragam. Terdapat seorang Project Manager (PM) yang dipegang oleh Noval Lias, serta anggota tim lainnya seperti Hasan Nur R., Andriansyah, Faragad Assanji, dan Destyawan Satria yang memiliki tanggung jawab spesifik pada tugas-tugas teknis seperti analisis sistem, pengembangan UI/UX, dan Quality Assurance (QA). Keterlibatan "Seluruh Tim" pada beberapa tugas kunci juga mengindikasikan adanya kolaborasi yang erat dalam penyelesaian proyek GANISSA ini hingga tuntas.

2. Fase-Fase Proyek

Fase 1 : Inisiasi & Perencanaan Proyek



Gambar 3. 2 Inisiasi & Perencanaan Proyek

Fase awal ini merupakan fondasi dari keseluruhan proyek, yang berlangsung pada akhir Mei 2025. Di tahap ini, tim fokus pada pendefinisian kerangka kerja proyek secara menyeluruh. Kegiatan utamanya meliputi identifikasi ruang lingkup, tujuan, dan para pemangku kepentingan (stakeholders). Tim kemudian menyusun dokumen krusial seperti Project Charter, melakukan estimasi biaya untuk menetapkan anggaran, mengembangkan rencana mitigasi risiko serta finalisasi proposal proyek untuk mendapatkan persetujuan resmi sebelum melangkah ke tahap selanjutnya.

Fase 2 : Analisis & Perancangan Sistem



Gambar 3. 3 Analisis & Perancangan Sistem

Berlangsung dari akhir Mei hingga awal Juni 2025, fase ini bertujuan untuk menerjemahkan kebutuhan bisnis menjadi rancangan teknis yang siap untuk dikembangkan. Tim melakukan analisis mendalam terhadap kebutuhan fungsional dan non-fungsional dari sistem yang akan dibangun. Hasil analisis tersebut kemudian digunakan untuk merancang arsitektur sistem dan struktur database. Selanjutnya, tim membuat desain antarmuka pengguna melalui wireframe dan mockup, yang diakhiri dengan pembuatan prototipe fungsional untuk memvalidasi konsep desain sebelum proses coding dimulai.

Fase 3 : Pembangunan Sistem (Development)



Gambar 3. 4 Fase 3 : Pembangunan Sistem (Development)

Ini adalah fase terpanjang dalam proyek, membentang dari awal Juni hingga pertengahan Juli 2025, di mana rancangan sistem diwujudkan menjadi aplikasi yang berfungsi. Proses pengembangan dibagi menjadi dua jalur utama: pengembangan backend yang mencakup pembuatan API, logika bisnis, dan pengelolaan database, serta pengembangan frontend yang berfokus pada implementasi desain UI/UX agar interaktif. Selama fase ini, proses coding dan debugging dilakukan secara berkelanjutan, dan puncaknya adalah integrasi antara frontend dan backend untuk memastikan keduanya dapat berkomunikasi dan bekerja sebagai satu kesatuan sistem.

Fase 4 : Pengujian & Revisi



Gambar 3. 5 Fase 4 : Pengujian & Revisi

Setelah sistem selesai dibangun, fase ini berfokus pada penjaminan kualitas yang dilaksanakan pada pertengahan Juli 2025. Pengujian dimulai dari level teknis dengan unit & component testing, dilanjutkan dengan integration testing untuk memastikan semua modul bekerja sama dengan baik. Setiap bug atau kesalahan yang ditemukan selama pengujian akan direvisi dan diperbaiki. Fase ini ditutup dengan User Acceptance Testing (UAT), di mana pengguna akhir menguji sistem untuk memastikan fungsionalitasnya telah sesuai dengan kebutuhan mereka dan siap untuk diluncurkan.

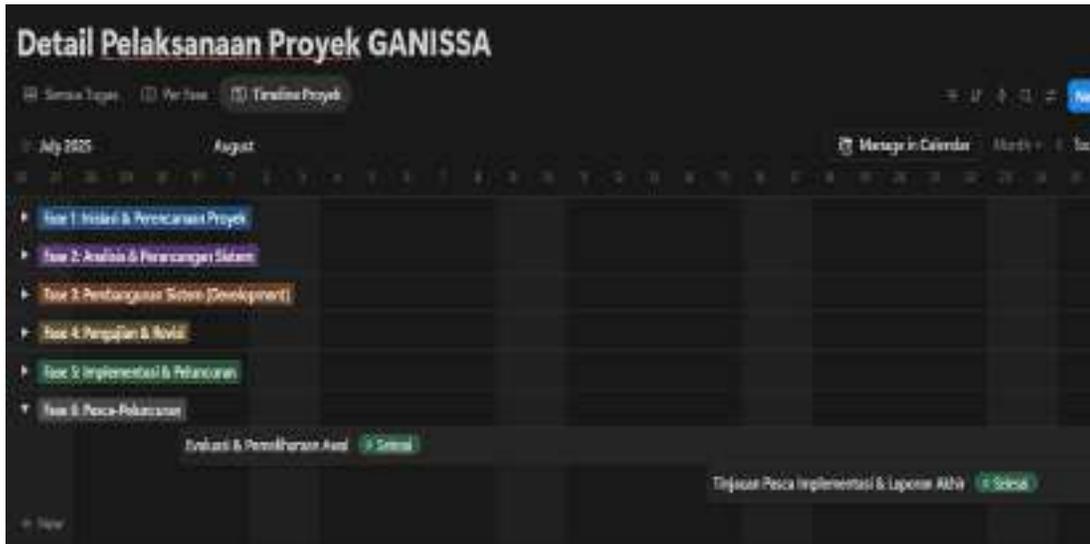
Fase 5 : Implementasi & Peluncuran



Gambar 3. 6 Fase 5 : Implementasi & Peluncuran

Pada akhir Juli 2025, sistem yang telah lulus pengujian siap untuk diimplementasikan dan diluncurkan ke pengguna. Persiapan dalam fase ini meliputi pembuatan materi dan dokumentasi pelatihan. Selanjutnya, diadakan sesi pelatihan untuk admin sistem dan sosialisasi kepada pengguna (alumni). Puncak dari fase ini adalah proses deployment sistem ke server produksi (hosting) sehingga dapat diakses secara publik, yang kemudian diresmikan melalui peluncuran resmi Sistem GANISSA

Fase 6 : Pasca-Peluncuran



Gambar 3. 7 Fase 6 : Pasca-Peluncuran

Fase terakhir ini berlangsung sepanjang bulan Agustus 2025 dan bertujuan untuk memastikan sistem berjalan dengan baik setelah diluncurkan serta mendokumentasikan hasil proyek. Kegiatan utama adalah evaluasi dan pemeliharaan awal untuk memonitor kinerja sistem dan menangani masalah yang mungkin muncul setelah digunakan secara massal. Proyek secara resmi ditutup dengan adanya tinjauan pasca-implementasi dan penyusunan laporan akhir yang merangkum seluruh proses, pencapaian, dan pelajaran yang didapat dari Proyek GANISSA.

B. Realisasi Hasil Pekerjaan

a. Realisasi Project Manager : Noval Lias Ramadani

A. Project Website Sekolah SMA N 2 Singkep

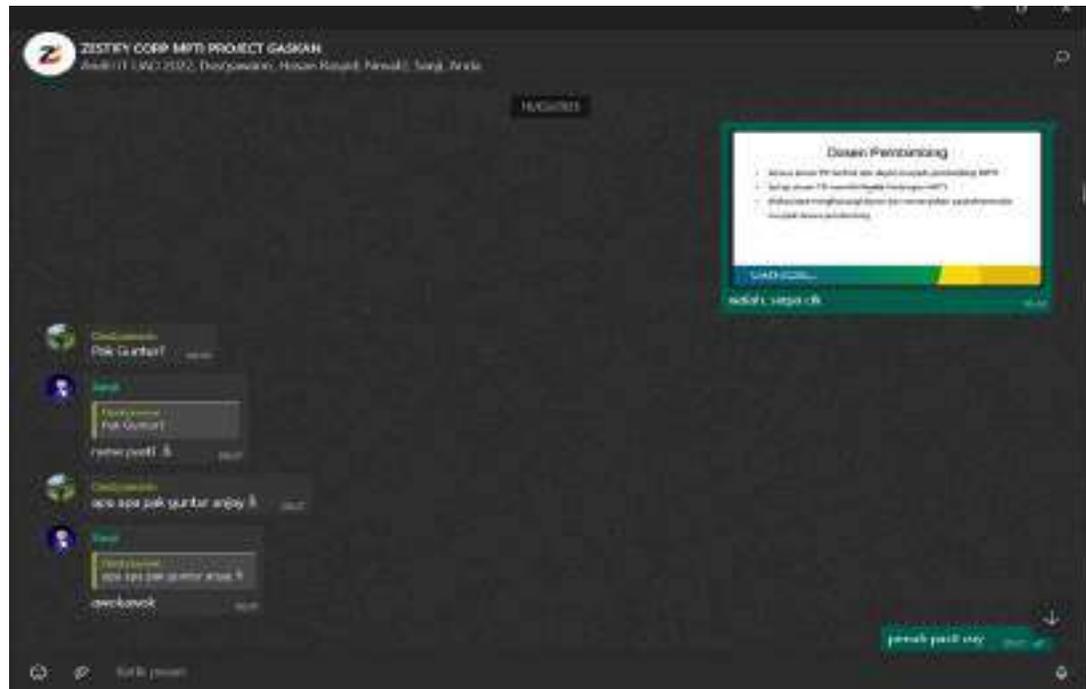
1) Koordinasi dengan tim terkait pencarian dosen pembimbing MPTI

Koordinasi dengan tim terkait pencarian dosen pembimbing untuk mata kuliah MPTI (Manajemen Proyek Teknologi Informasi) dilakukan secara intensif guna memastikan kesesuaian antara topik proyek yang diusulkan dengan kompetensi dosen. Tim kami terlebih dahulu menyusun daftar calon dosen yang memiliki latar belakang keahlian di bidang manajemen proyek, sistem informasi, dan pengembangan teknologi informasi. Selanjutnya, kami melakukan diskusi internal untuk menilai preferensi masing-masing anggota tim terhadap karakteristik dosen, seperti gaya bimbingan, pengalaman, dan kesediaan dalam membimbing.

Setelah daftar calon dosen disusun, tim melakukan pendekatan informal kepada beberapa dosen yang masuk dalam pertimbangan utama. Kami menjelaskan secara singkat ruang lingkup proyek, tujuan yang ingin dicapai, serta ekspektasi terhadap bimbingan yang akan kami terima. Dari hasil komunikasi tersebut, kami mendapatkan gambaran lebih jelas mengenai kesiapan dan ketertarikan masing-masing dosen terhadap proyek yang kami ajukan. Selain itu, masukan dari dosen juga kami jadikan pertimbangan dalam proses seleksi.

Berdasarkan hasil evaluasi dan pertimbangan bersama, kami akhirnya sepakat untuk memilih Bapak Murinto sebagai dosen pembimbing kami. Keputusan ini

diambil karena beliau memiliki latar belakang yang kuat di bidang manajemen proyek teknologi informasi serta pengalaman yang luas dalam membimbing mahasiswa. Selain itu, kesediaan dan respons positif dari beliau terhadap proyek kami menjadi faktor penentu dalam pengambilan keputusan. Kami yakin, dengan bimbingan dari Bapak Murinto, proyek ini dapat berjalan dengan baik dan memberikan hasil yang optimal. Di bawah ini ada proses pemilihan berdasarkan koordinasi dengan tim yang dijadwalkan pada 10 Maret 2025

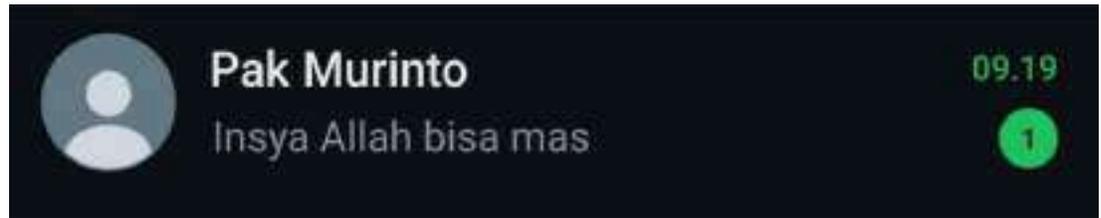


Gambar 3. 8 Diskusi Pemilihan Dosen Pembimbing MPTI part1



Gambar 3. 9 Diskusi Pemilihan Dosen Pembimbing MPTI part2

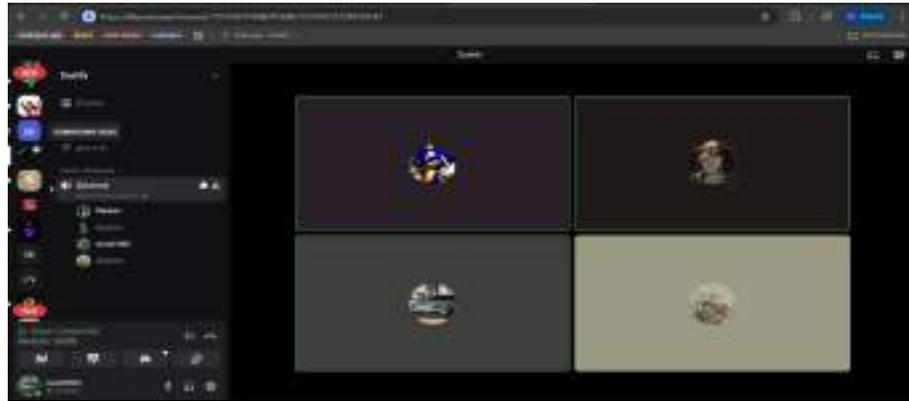
Proses Persetujuan oleh Pak Murinto



Gambar 3. 10 Proses persetujuan oleh dosen pembimbing

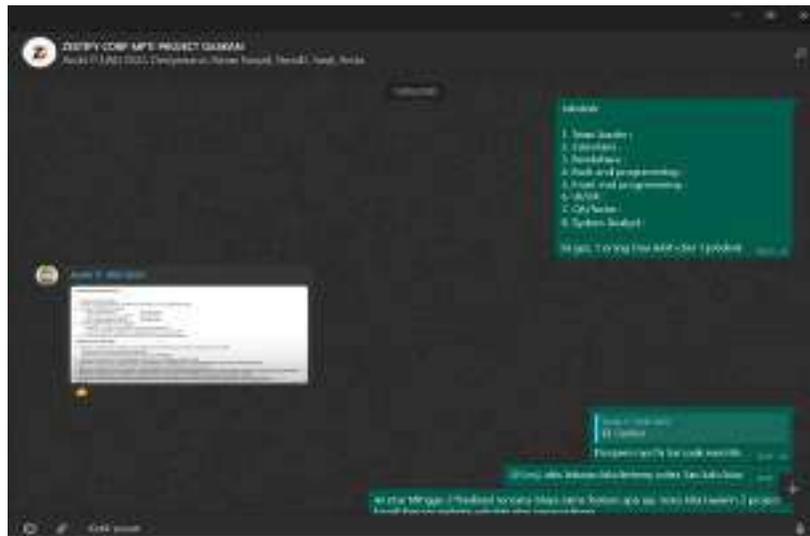
- 2) Selanjutnya pada minggu ke 2 melakukan koordinasi dengan tim, terkait project yang ingin dibangun serta mitra

Pada minggu kedua, tim melanjutkan koordinasi internal untuk membahas secara mendalam terkait ide proyek yang akan dibangun serta pemilihan mitra yang akan dilibatkan. Dalam diskusi ini, masing-masing anggota menyampaikan gagasan dan pertimbangan mengenai permasalahan nyata yang bisa diangkat sebagai dasar proyek, kemudian dilakukan pemilahan berdasarkan kelayakan, ketersediaan data, dan relevansi dengan mata kuliah MPTI. Setelah melalui beberapa pertimbangan, tim menyepakati satu ide proyek utama dan mulai mengidentifikasi mitra eksternal yang sesuai, dengan harapan dapat mendukung kelancaran pengumpulan kebutuhan serta validasi sistem yang akan dikembangkan.

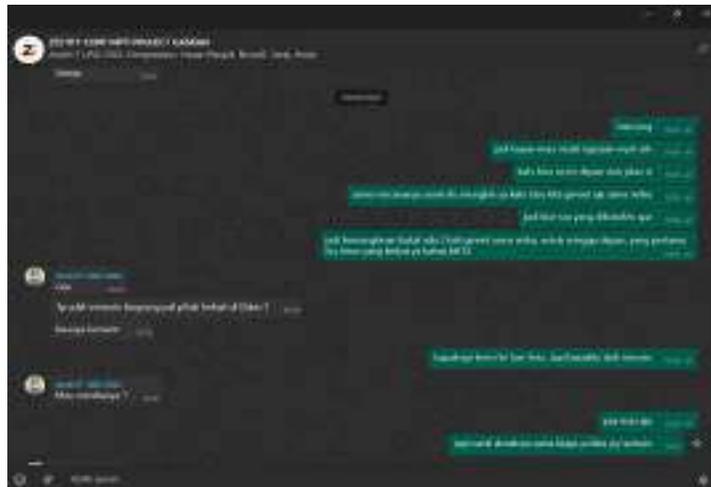


Gambar 3. 11 Diskusi Proyek via Discord

- 3) Setelah mendapatkan rencana kemitraan, melakukan koordinasi dengan tim terkait pembagian jobdesk untuk rencana lanjut kedepannya

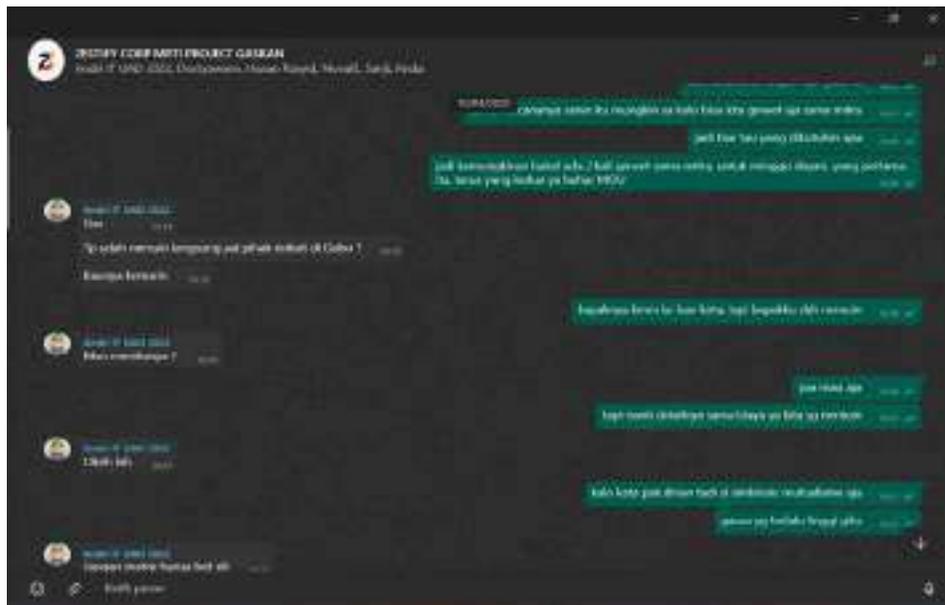


Gambar 3. 12 Pembagian Tugas Setiap Anggota



Gambar 3. 14 Kendala Pertemuan Mitra

- 6) Perencanaan gmeet dengan mitra untuk membahas keberlanjutan project lebih lanjut terkait dengan fitur apa saja yang akan dibuat

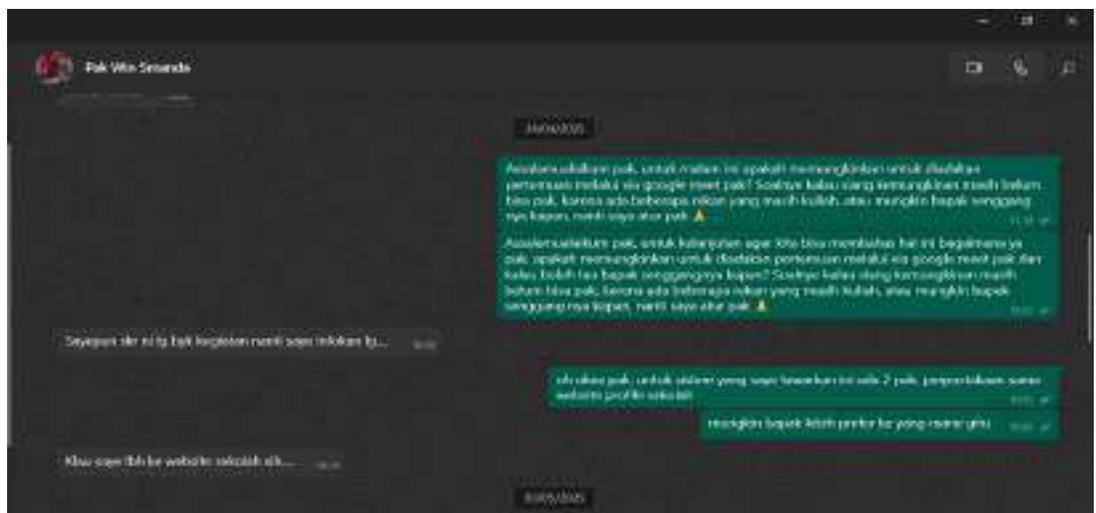


Gambar 3. 15 Perencanaan Perubahan Pertemuan Menjadi via Gmeet



Gambar 3. 16 Perubahan Pertemuan Menjadi via Gmeet

- 7) Karena ada sedikit kendala waktu antara tim dan mitra yang membuat komunikasi menjadi sedikit terkendala, maka untuk pertemuan online di jadwalkan secara ulang.



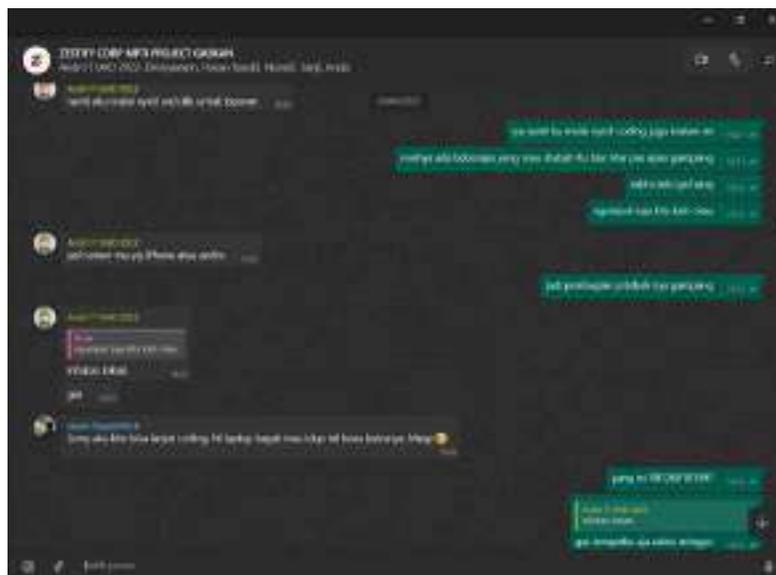
Gambar 3. 17 Penjadwalan Ulang Gmeet Dengan Mitra

- 8) Koordinasi dengan tim terkait pengembangan project yang dibuat melalui meet discord.



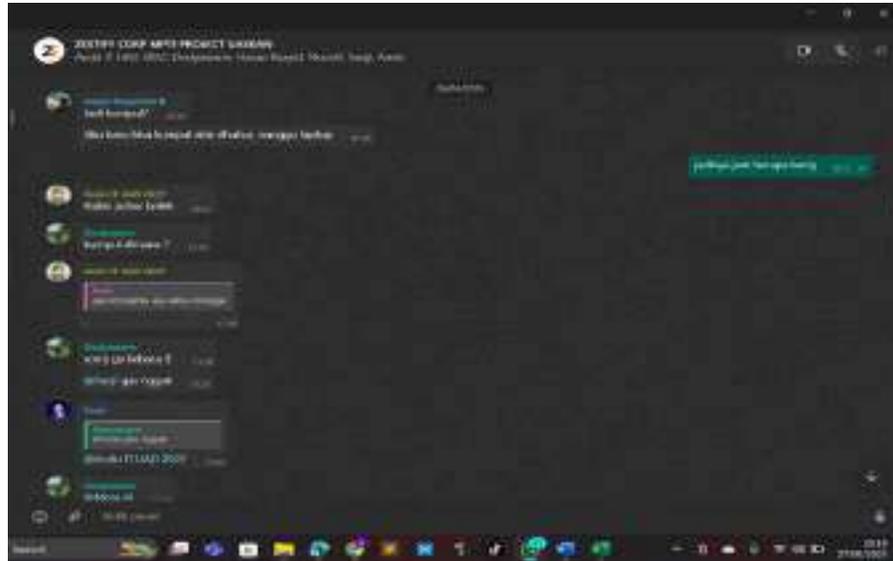
Gambar 3. 18 Koordinasi Tim Terkait Pengembangan Project

- 9) Koordinasi dengan tim dan kumpul pertama secara offline untuk keberlanjutan project mpt ini

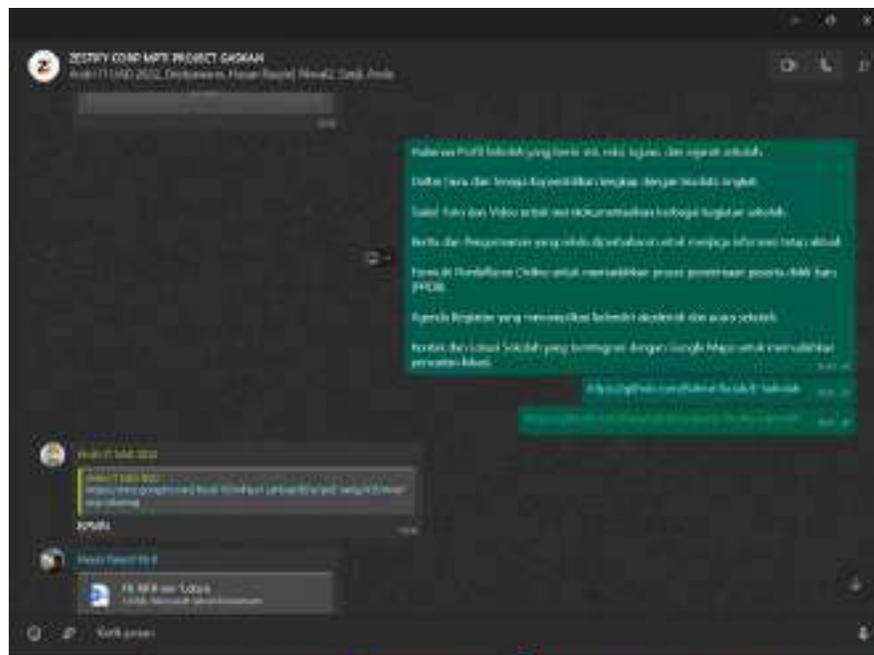


Gambar 3. 19 Koordinasi Tim Untuk Pertemuan Offline

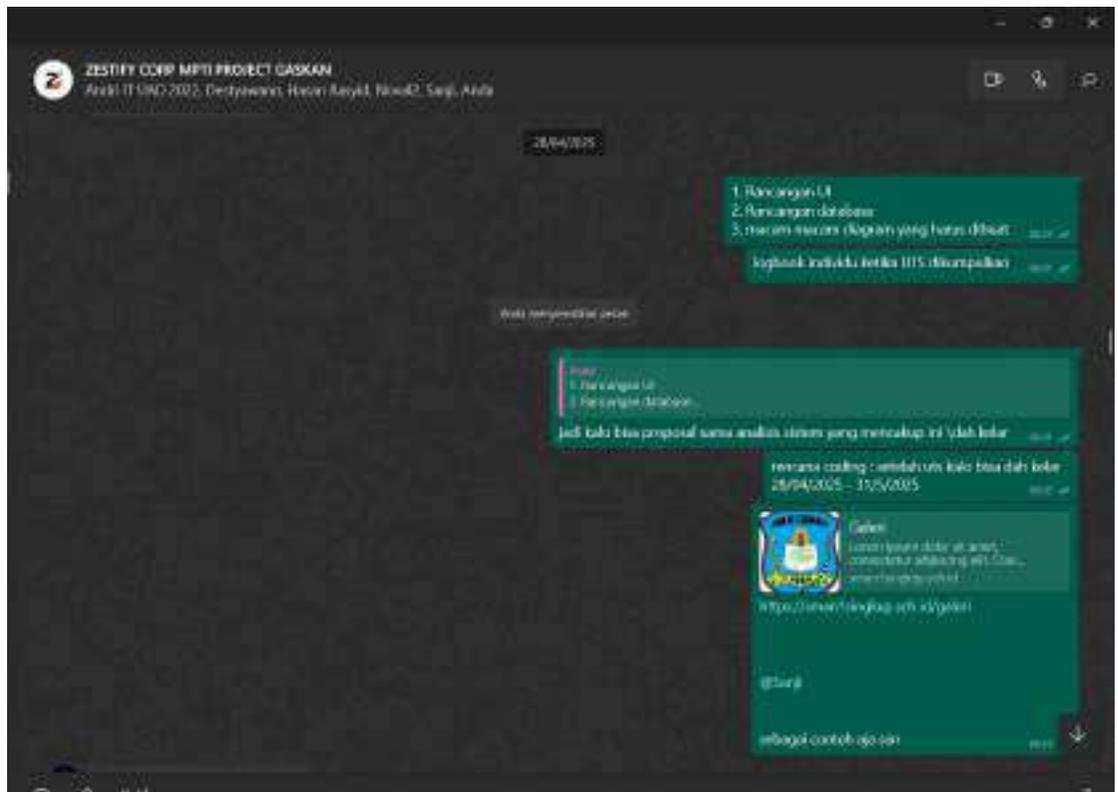
10) Pembahasan terkait langkah pengembangan coding yang telah dilakukan dan dilakukan secara offline



Gambar 3. 20 Diskusi Terkait Jam Pertemuan Tim

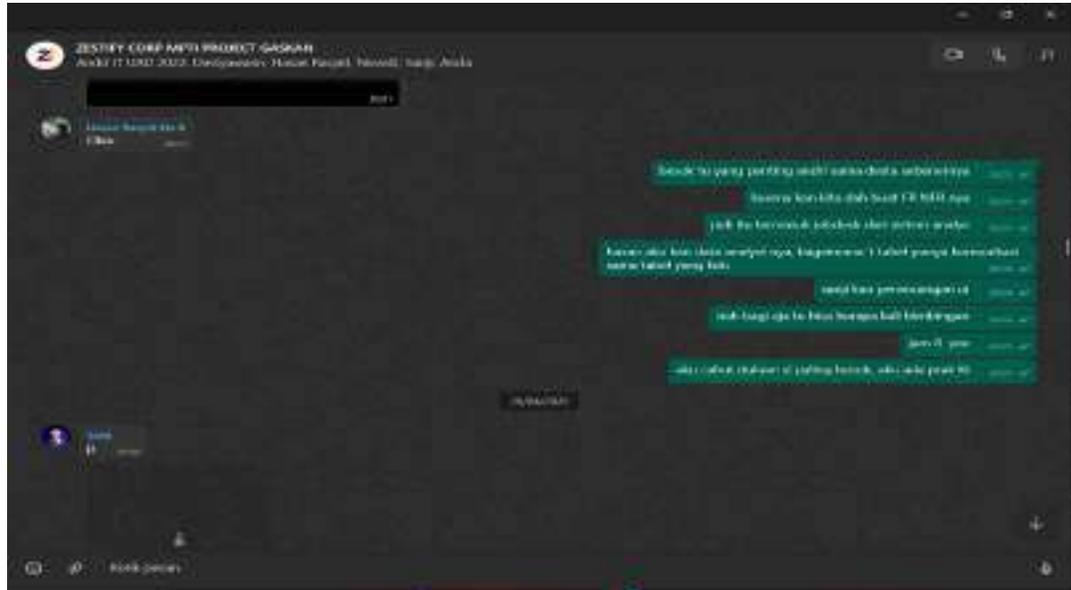


Gambar 3. 21 Diskusi Tim Terkait Fitur Sistem



Gambar 3. 22 Diskusi Tim Terkait Referensi Proyek

11) Perencanaan pertemuan dengan dosen pembimbing terkait project mpti yang dibuat



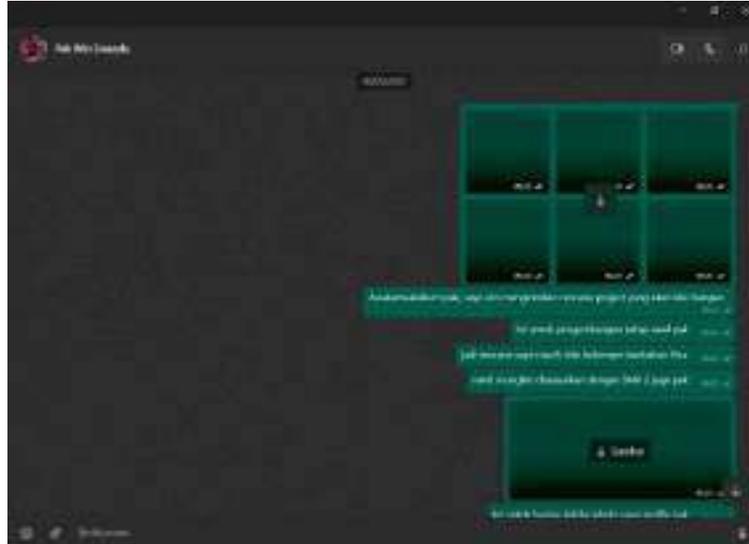
Gambar 3. 23 Perencanaan Bimbingan MPTI

12) Perencanaan kembali pertemuan via google meet dengan mitra terkait proses pengerjaan dan fitur yang akan dibuat



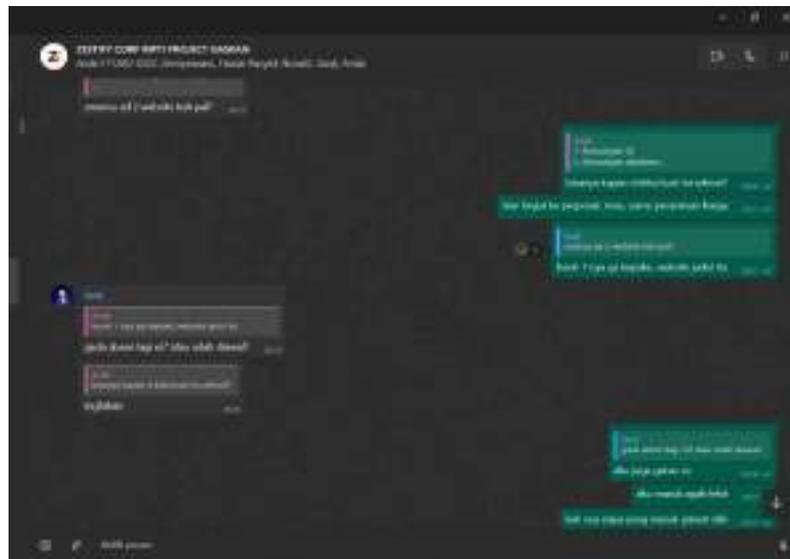
Gambar 3. 24 Perencanaan Penjadwalan Pertemuan Mitra via GMeets

- 13) Pengiriman hasil pengerjaan sementara project yang dibangun dan mitra memberikan persetujuan



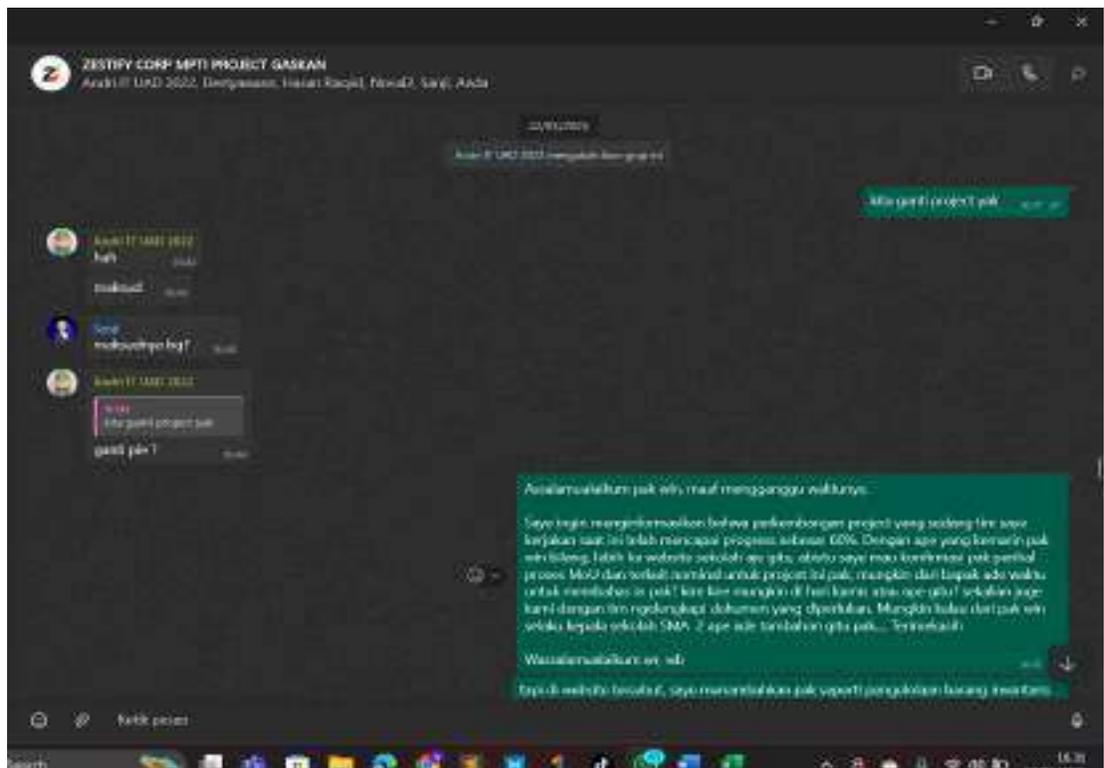
Gambar 3. 25 Pengiriman Progress Pengerjaan Kepada Mitra

- 14) Koordinasi dengan tim terkait perkembangan jobdesk dari masing-masing tim dan menanyakan progress mereka

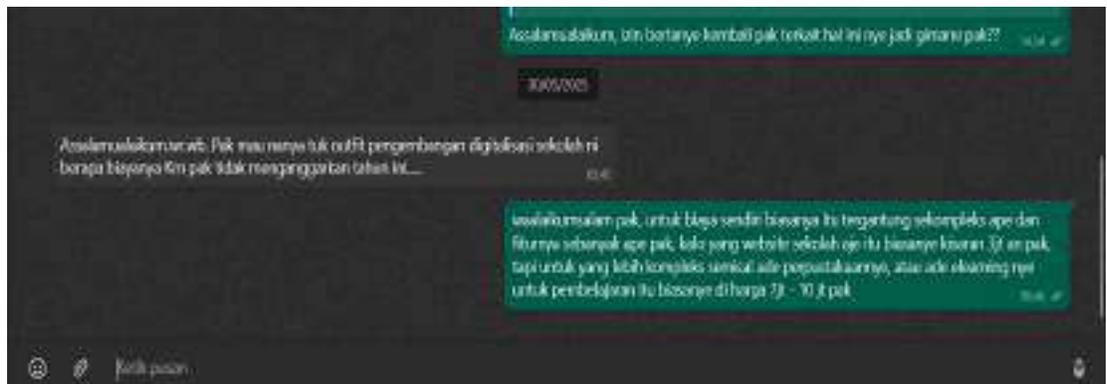


Gambar 3. 26 Memantau dan Melakukan Koordinasi Progress Setiap Anggota

15) Tetapi disini kami mengalami kendala, karena dari mitra setelah komunikasi yang lebih intensif, mitra tidak menganggarkan untuk pengembangan website tersebut yang alhasil kami tidak jadi melanjutkan untuk kesepakatan dari mitra tersebut dan belum sampai pada tahap mou, tetapi project sudah berjalan sekitar 60%, dan rencananya memang akan kami lanjutkan setelah penandatanganan mou dan pembayaran dp yang pasti



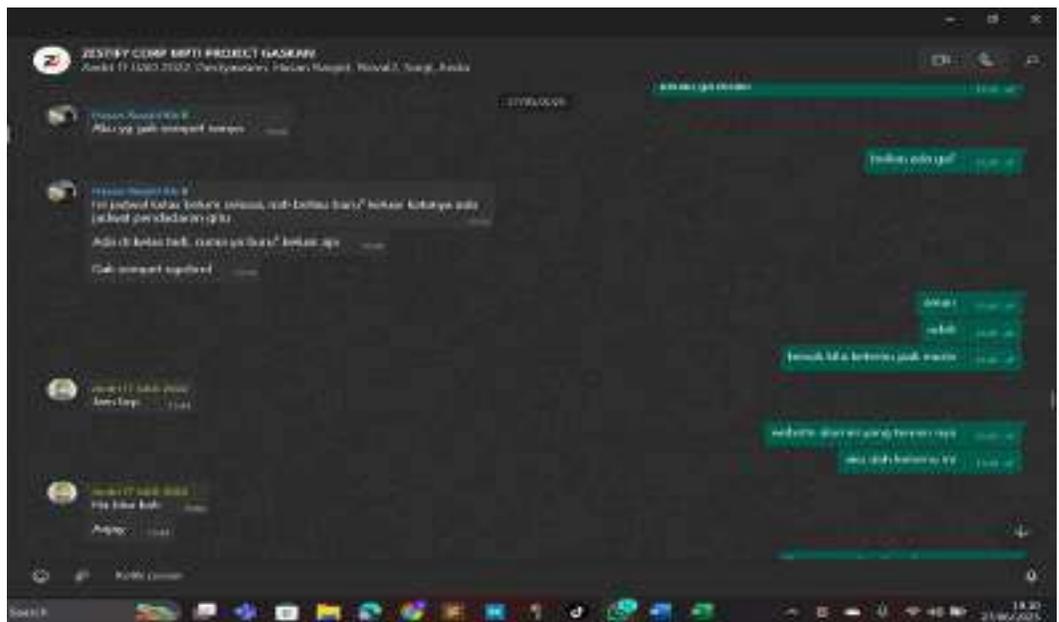
Gambar 3. 29 Pemberitahuan Kepada Anggota Tim Untuk Penggantian Proyek dan Mitra



Gambar 3. 30 Konfirmasi dari pihak sekolah jika mereka tidak mengganggu anggaran tahun ini

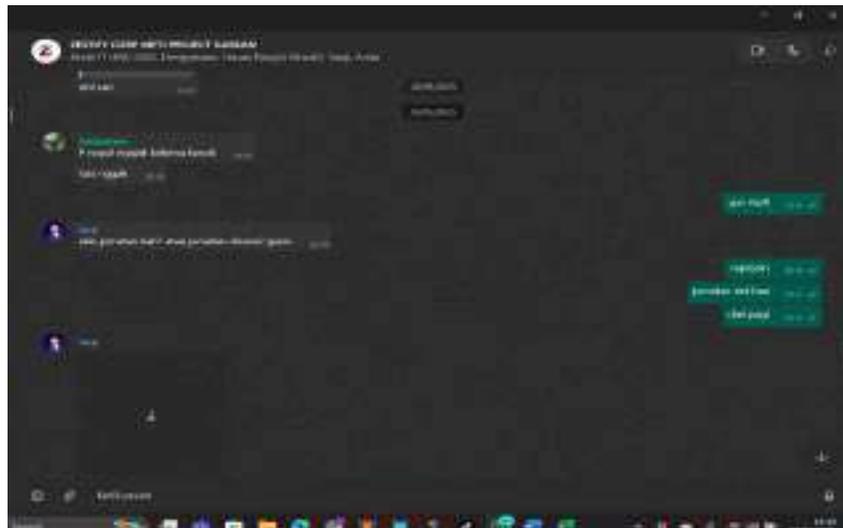
B. Project Sistem Informasi Alumni Ganissa Pernalang (SMA N 1 Pernalang)

- 1) Melakukan koordinasi dengan pembimbing terkait perubahan mitra karena kendala komunikasi. Dalam pertemuan langsung di kampus, Pak Murinto selaku pembimbing menawarkan rekannya sebagai mitra dengan proyek berupa sistem informasi alumni.



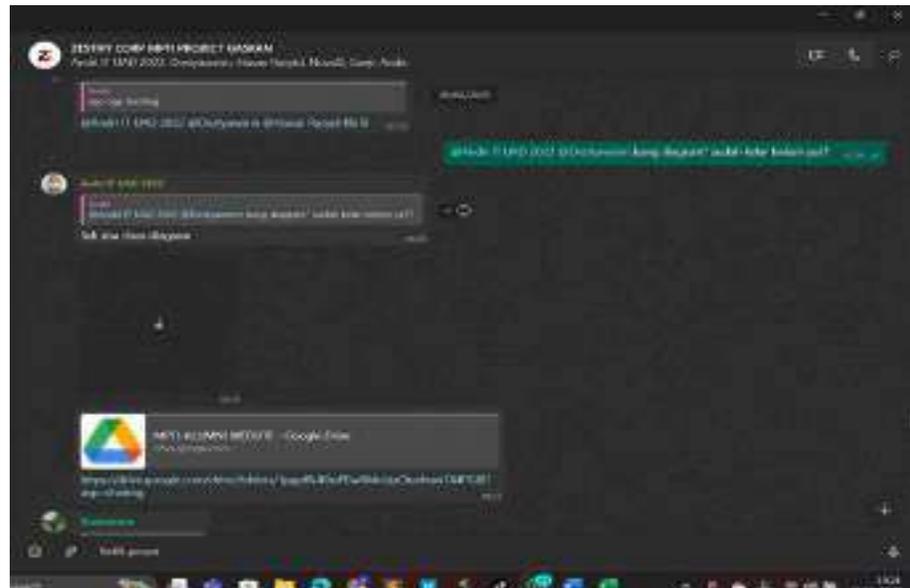
Gambar 3. 31 Hasil Pertemuan Cepat dengan Pembimbing

3) Melakukan koordinasi dengan tim untuk pengerjaan tahap awal



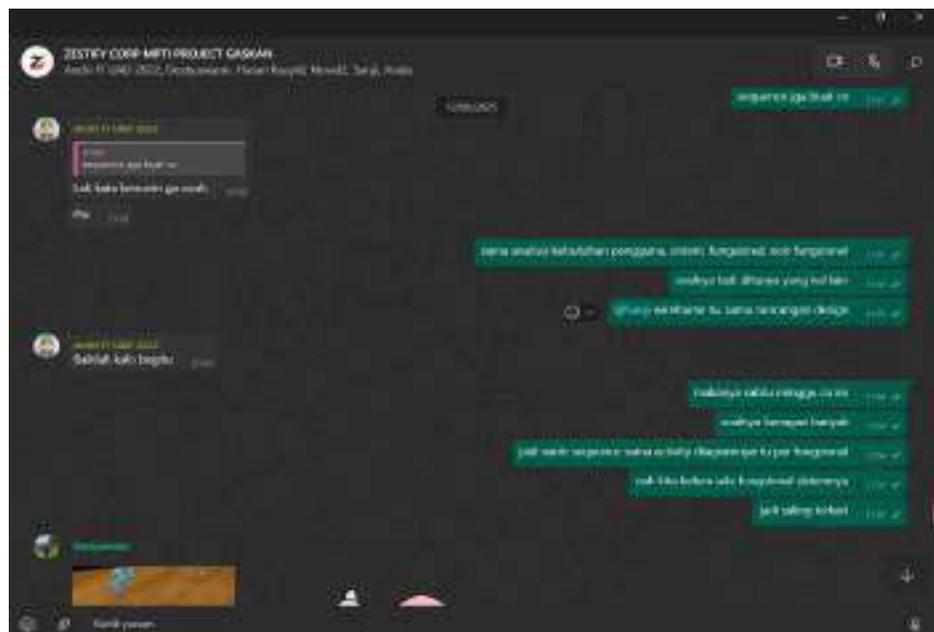
Gambar 3. 33 Koordinasi tim pengerjaan tahap awal

4) Memantau Perkembangan dari masing-masing anggota tim



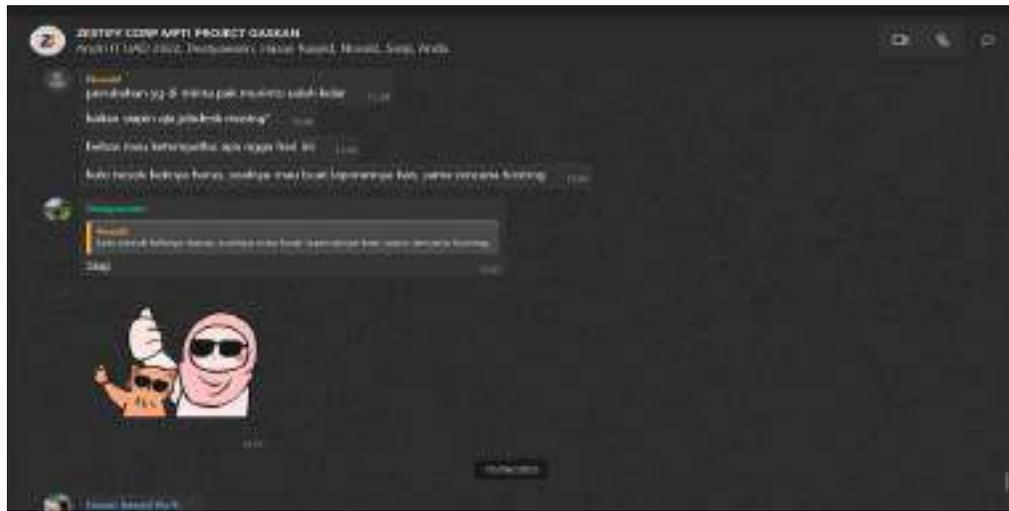
Gambar 3. 34 Pemantauan progress dari masing-masing anggota tim

5) Melakukan penilaian dan revisi terhadap pekerjaan anggota tim

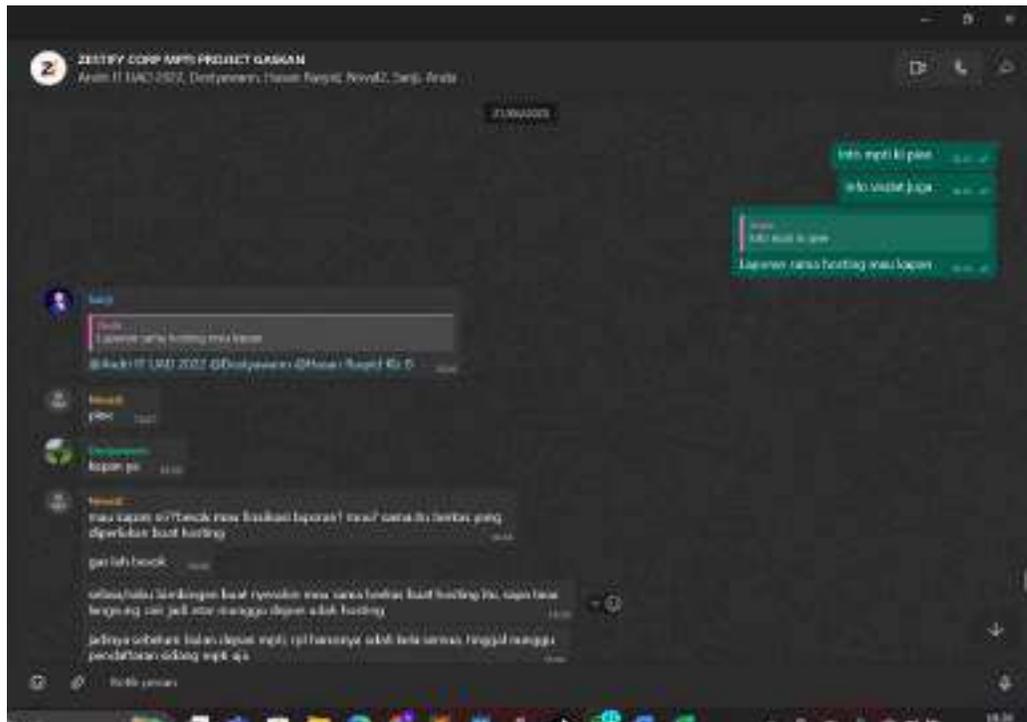


Gambar 3. 35 Penilaian dan revisi terhadap hasil kerja anggota tim

6) Melakukan koordinasi terkait pelaporan dan finalisasi project

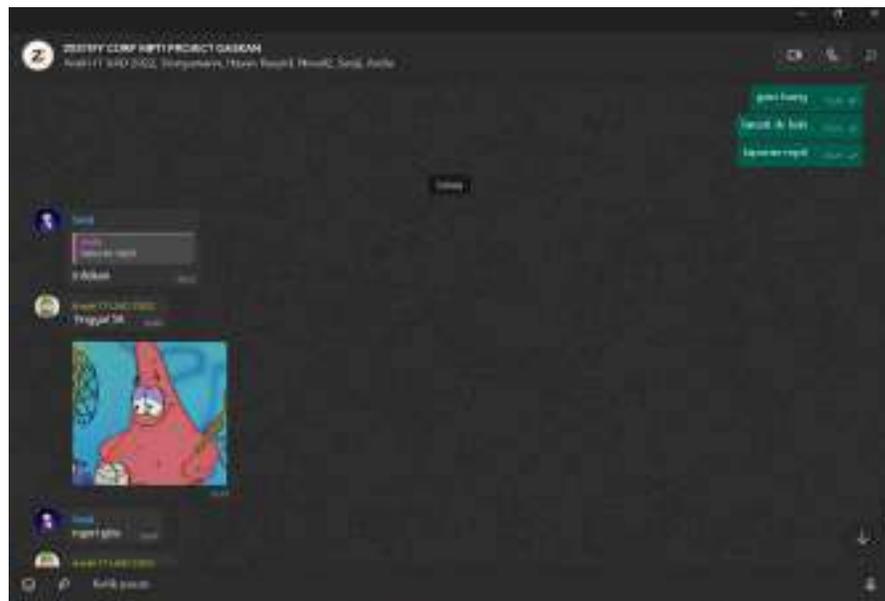


Gambar 3. 36 Koordinasi dan penjadwalan terkait pelaporan dan finalisasi tahap pengkodean



Gambar 3. 37 Melakukan koordinasi dengan tim terkait deploy dan laporan akhir

7) Melakukan koordinasi tim terkait laporan mpti via discord



Gambar 3. 39 Koordinasi keberlanjutan pengerjaan laporan

8) Bimbingan tahap akhir denan dosen pembimbing sekaligus finalisasi



Gambar 3. 40 Bimbingan akhir dan finalisasi

9) Melakukan Penyerahan dan Presentasi ke mitra yang bersangkutan



Gambar 3. 41 Penyerahan Produk kepada mitra

b. Realisasi System Analyst

a. Perancangan Sistem

Perancangan sistem ini bertujuan untuk menyajikan gambaran menyeluruh kepada pengguna mengenai sistem yang akan dikembangkan. Proses perancangannya dilakukan melalui tahapan-tahapan menggunakan pendekatan Unified Modeling Language (UML), yang mencakup analisis sistem, pembuatan use case diagram, activity diagram, serta class diagram. Berikut perancangan sistem pada sistem informasi alumni sman 1 pemalang (ganissa) :

a) Kebutuhan Fungsional

Kebutuhan fungsional merupakan sekumpulan fungsi yang didokumentasikan untuk menggambarkan kemampuan sistem dalam menjalankan berbagai tugas

guna memenuhi kebutuhan pengguna. Dalam konteks aplikasi rental mobil berbasis web, kebutuhan fungsional ini dijabarkan secara rinci dalam bentuk tabel

Tabel 3. 2 Analisis Kebutuhan Fungsional

Kode Fungsional	Fungsionalitas Sistem	Deskripsi Fungsionalitas Sistem
F-ADM-001	Manajemen Akun Admin	Sistem harus memungkinkan Admin mengelola akun admin lain, termasuk membuat, mengedit, menghapus, dan mengatur hak akses.
F-ADM-002	Manajemen Data Profil Alumni	Sistem harus memungkinkan Admin melakukan CRUD data profil alumni secara lengkap.
F-ADM-003	Dashboard & Pelaporan Lanjutan	Sistem harus menyajikan dashboard dengan angka-angka pengguna, konten, dan aktivitas.
F-ADM-004	Manajemen Konten Umum (Berita, Artikel, Pengumuman)	Sistem harus menyediakan CMS (Content Mangement System) untuk mengelola berita, artikel, dan pengumuman dengan editor, media, jadwal publikasi, dan riwayat revisi.
F-ADM-005	Pengelolaan Kategori & Tag Dinamis	Sistem harus memungkinkan Admin mengelola kategori dan tag konten secara dinamis dan bertingkat.
F-ADM-006	Manajemen Akun Pengguna & Otorisasi	Sistem harus mendukung manajemen akun pengguna (Admin, Alumni, Guest) dengan pengaturan peran dan hak akses berbasis RBAC.
F-ADM-007	Sistem Moderasi Konten & Komunikasi	Sistem harus menyediakan modul moderasi untuk menyetujui, menolak, atau menghapus konten dan komentar.
F-ADM-008	Manajemen Event Komprehensif	Sistem harus memungkinkan Admin mengelola event secara komprehensif, termasuk data peserta.
F-ALUM-001	Registrasi & Aktivasi Akun Mandiri	Sistem harus memungkinkan calon alumni mendaftar akun baru secara mandiri melalui formulir.
F-ALUM-002	Manajemen & Pembaruan Profil Mandiri	Sistem harus memungkinkan Alumni memperbarui dan mengelola profil pribadi secara mandiri.
F-ALUM-003	Pencarian & Filter Alumni Lanjutan	Sistem harus menyediakan fitur pencarian alumni dengan filter dan urutan hasil berdasarkan kriteria tertentu.

F-ALUM-004	Akses & Interaksi Konten (Berita, Artikel)	Sistem harus memungkinkan Alumni mengakses, berinteraksi, dan membagikan konten seperti artikel atau berita.
F-ALUM-005	Registrasi & Partisipasi Event Online/Offline	Sistem harus memungkinkan Alumni melihat, mendaftar, dan mengelola partisipasi event secara online/offline.
F-ALUM-006	Kontribusi Konten (Artikel, Pengalaman, Portofolio)	Sistem harus memungkinkan Alumni mengirim kontribusi konten seperti artikel, pengalaman kerja, atau portofolio.
F-GUEST-001	Akses Informasi Publik Terbatas	Sistem harus memungkinkan Guest mengakses informasi publik tanpa login.
F-GUEST-002	Penampil Jadwal & Detail Event Publik	Sistem harus memungkinkan Guest melihat jadwal dan detail event publik.
F-GUEST-003	Proses Registrasi Awal Alumni	Sistem harus memungkinkan Guest mengisi formulir pendaftaran awal alumni sebelum aktivasi akun.

b) Kebutuhan Non Fungsional

Kebutuhan non fungsional merupakan jenis kebutuhan yang tidak berhubungan langsung dengan fungsi utama sistem, melainkan berfokus pada aspek-aspek penting lainnya seperti kinerja, keamanan, dan kemudahan penggunaan. Pada aplikasi pemesanan rental mobil berbasis web, kebutuhan non fungsional ini dijelaskan lebih lanjut dalam bentuk tabel.

Tabel 3. 3 Analisis Kebutuhan Non-Fungsional

Kode Non-Fungsional	Kategori Kebutuhan	Deskripsi Kebutuhan
NF-001	Kinerja	Sistem harus memiliki waktu respons cepat (misal, < 3 detik untuk 90% permintaan umum) dan mampu menangani peningkatan beban pengguna secara efisien tanpa penurunan performa yang signifikan.

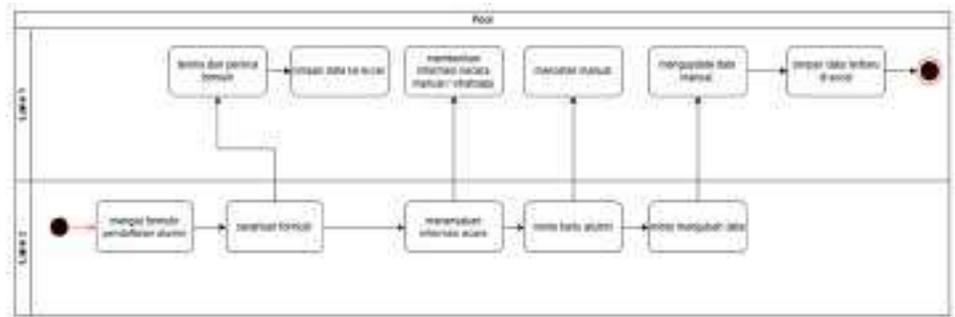
NF-002	Kinerja	Sistem harus dirancang untuk dapat ditingkatkan kapasitasnya (skalabilitas) untuk menampung pertumbuhan data dan jumlah pengguna di masa mendatang (misal, hingga 10.000 pengguna terdaftar).
NF-003	Kinerja	Penggunaan sumber daya server (CPU, memori) harus dioptimalkan untuk memastikan stabilitas dan kinerja yang konsisten selama operasional.
NF-004	Keamanan	Sistem harus menerapkan mekanisme otentikasi dan otorisasi yang kuat (misal, password hashing, RBAC) untuk melindungi akun pengguna dan data sensitif.
NF-005	Keamanan	Seluruh transmisi data (data in transit) harus dienkripsi (HTTPS/TLS), dan data sensitif yang disimpan (data at rest) juga harus dienkripsi untuk mencegah akses tidak sah.
NF-006	Keamanan	Sistem harus memiliki pertahanan terhadap kerentanan keamanan umum (misal, XSS, SQL Injection) dan menyediakan audit trail untuk melacak aktivitas pengguna penting.
NF-007	Keandalan	Sistem harus memiliki tingkat ketersediaan yang tinggi (misal, 99.5% uptime) untuk memastikan pengguna dapat mengakses layanan kapan pun.
NF-008	Keandalan	Sistem harus memiliki strategi pencadangan data otomatis yang teratur dan mekanisme pemulihan data yang jelas untuk mencegah kehilangan data akibat kegagalan.
NF-009	Keandalan	Sistem harus mampu menangani kesalahan dengan baik, memberikan pesan yang informatif kepada pengguna, dan meminimalkan dampak error pada fungsionalitas keseluruhan.
NF-010	Kegunaan	Antarmuka pengguna (UI) harus intuitif, mudah dinavigasi, dan konsisten di seluruh aplikasi, dengan desain yang responsif untuk berbagai perangkat (desktop, mobile).
NF-011	Kegunaan	Pengguna baru harus dapat dengan mudah mempelajari dan menggunakan fitur-fitur inti sistem tanpa perlu pelatihan ekstensif.

NF-012	Kegunaan	Sistem harus menyediakan pesan kesalahan yang jelas dan informatif, serta panduan bantuan kontekstual untuk mendukung pengguna.
NF-013	Pemeliharaan	Arsitektur sistem harus modular dan didokumentasikan dengan baik untuk memudahkan perbaikan, penambahan fitur, dan pembaruan di masa mendatang.
NF-014	Pemeliharaan	Sistem harus mendukung logging aktivitas dan monitoring kinerja untuk memudahkan identifikasi masalah dan pemecahan masalah.
NF-015	Kompatibilitas	Sistem harus berfungsi dengan baik di browser web modern (misal, Chrome, Firefox, Edge, Safari) dan berbagai sistem operasi perangkat.
NF-016	Kompatibilitas	Sistem harus memiliki kemampuan untuk berintegrasi dengan sistem eksternal penting melalui API yang terdefinisi dengan baik jika diperlukan.

b. Perancangan Proses Bisnis

a) Proses Bisnis Sebelum Ada Sistem

Proses bisnis yang lebih sederhana ini memaparkan proses bisnis manual dalam pengelolaan data alumni yang terjadi sebelum adanya sistem digital. Aktivitas utama bergantung pada interaksi langsung antara alumni dan admin, di mana alumni menyerahkan formulir fisik dan admin secara manual memasukkan data ke dalam spreadsheet Excel. Seluruh komunikasi, seperti pemberian informasi acara, juga dilakukan secara manual melalui Whatsapp atau komunikasi langsung, sementara pembaruan data dan pencetakan kartu alumni juga memerlukan langkah-langkah manual, membuat proses ini rentan terhadap kesalahan manusia, tidak efisien, dan sulit untuk diakses secara real-time.

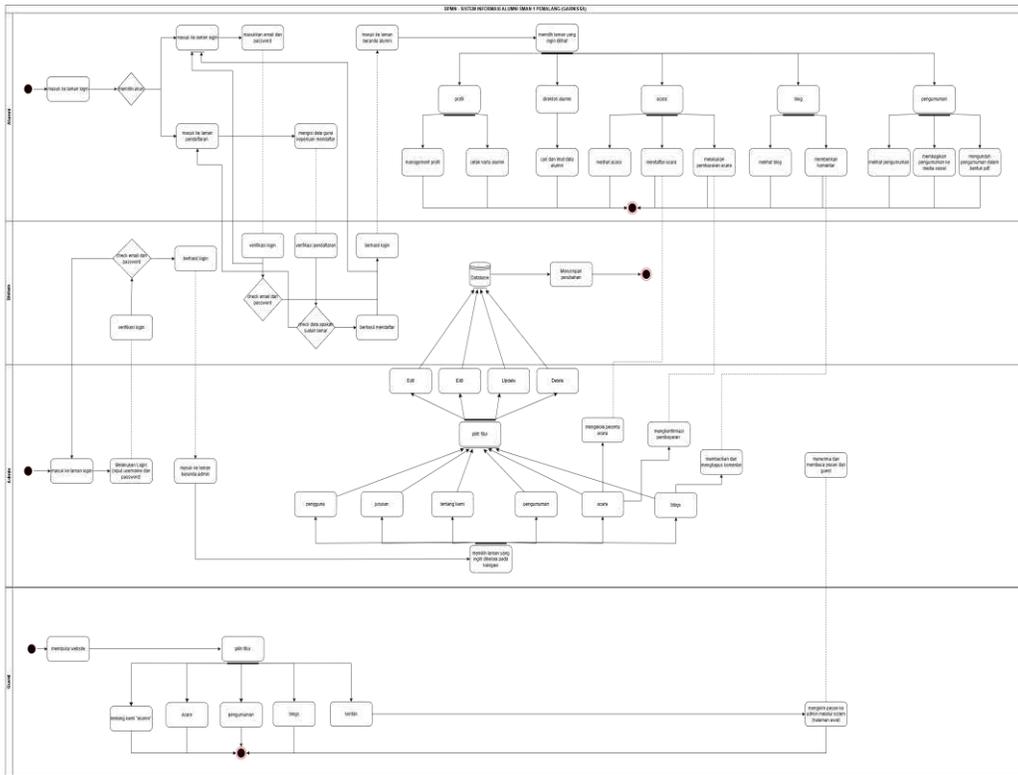


Gambar 3. 42 BPMN Sebelum ada Sistem

b) Proses Bisnis Setelah Ada Sistem

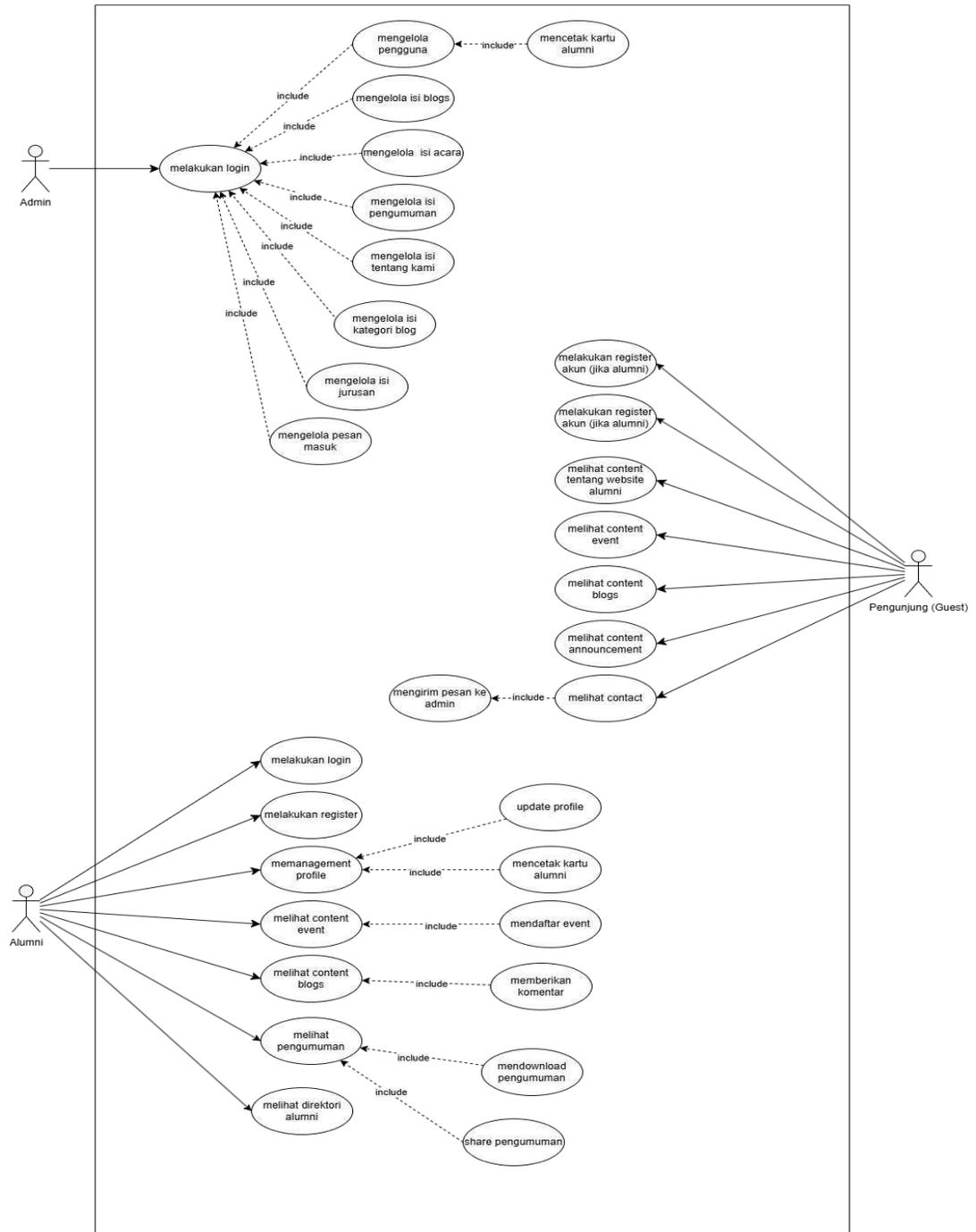
Melihat proses yang digambarkan, dapat diidentifikasi bahwa sebelum adanya 80system terpusat, proses pendataan dan komunikasi dengan alumni kemungkinan besar dilakukan secara manual dan terfragmentasi. Panitia, Humas, dan Alumni tidak memiliki satu platform tunggal untuk berinteraksi dan berbagi informasi. Data alumni mungkin tersebar, informasi lowongan kerja dan acara sulit disebarakan secara efisien, dan tidak ada wadah khusus untuk komunikasi antar alumni.

Namun, setelah sistem informasi ini dibangun, semua proses menjadi terintegrasi. Alumni dapat secara mandiri memperbarui data mereka, mengakses berbagai informasi penting seperti lowongan kerja dan acara, serta berinteraksi melalui forum. Di sisi lain, Panitia dan Humas dapat mengelola dan menyebarkan informasi dengan lebih mudah dan terstruktur melalui satu portal website.



Gambar 3. 43 BPMN Setelah ada Sistem

c. Perancangan Use Case Diagram



Gambar 3. 44 Use Case Diagram

Berikut penjelasan terkait use case dari masing-masing aktor :

- i. Admin adalah pengguna dengan hak akses tertinggi dalam sistem. Untuk dapat menggunakan fungsinya, Admin harus melakukan login terlebih dahulu. Setelah berhasil login, Admin memiliki wewenang untuk mengelola berbagai aspek sistem, yang mencakup:
 1. Mengelola Pengguna: Admin dapat mengatur akun-akun pengguna yang terdaftar di sistem.
 2. Mengelola Isi Blogs: Admin bertanggung jawab atas konten blog, termasuk membuat, mengubah, dan menghapus postingan.
 3. Mengelola Isi Acara: Admin dapat membuat dan mengelola informasi mengenai acara atau kegiatan.
 4. Mengelola Isi Pengumuman: Admin dapat mempublikasikan pengumuman penting untuk pengguna lain.
 5. Mengelola Isi Tentang Kami: Admin bisa mengubah informasi pada halaman "Tentang Kami".
 6. Mengelola Isi Kategori Blog: Admin dapat mengatur kategori-kategori yang digunakan untuk mengelompokkan artikel blog.
 7. Mengelola Isi Jurusan: Admin dapat mengelola data jurusan yang ada.
 8. Mengelola Pesan Masuk: Admin dapat melihat dan menindaklanjuti pesan yang dikirimkan oleh pengunjung melalui halaman kontak.
 9. Mencetak Kartu Alumni: Fungsi ini merupakan bagian dari "Mengelola Pengguna", di mana Admin dapat mencetak kartu untuk alumni

- ii. Alumni adalah pengguna yang telah terdaftar dalam sistem dan merupakan lulusan dari institusi. Alumni memiliki akses ke fitur-fitur yang lebih personal dan interaktif.
 - 1. Melakukan Login: Sama seperti Admin, Alumni harus login untuk mengakses sebagian besar fitur.
 - 2. Melakukan Register: Jika belum memiliki akun, mereka dapat mendaftar terlebih dahulu.
 - 3. Manajemen Profile: Setelah login, Alumni dapat mengelola profil mereka. Ini mencakup:
 - a. Update Profile: Memperbarui informasi pribadi mereka.
 - b. Mencetak Kartu Alumni: Alumni dapat mencetak kartu identitas alumni mereka sendiri.
 - 4. Melihat Content Event: Alumni dapat melihat informasi acara. Dari sini, mereka bisa:
 - a. Mendaftar Event: Mengikuti atau mendaftar pada acara yang tersedia.
 - 5. Melihat Content Blogs: Alumni dapat membaca artikel blog yang dipublikasikan. Dari sini, mereka dapat:
 - a. Memberikan Komentar: Berinteraksi dengan meninggalkan komentar pada postingan blog.
 - 6. Melihat Pengumuman: Alumni dapat melihat pengumuman yang ada. Dari sini, mereka bisa:
 - a. Mendownload Pengumuman: Mengunduh lampiran atau file pengumuman.
 - b. Share Pengumuman: Membagikan pengumuman tersebut.

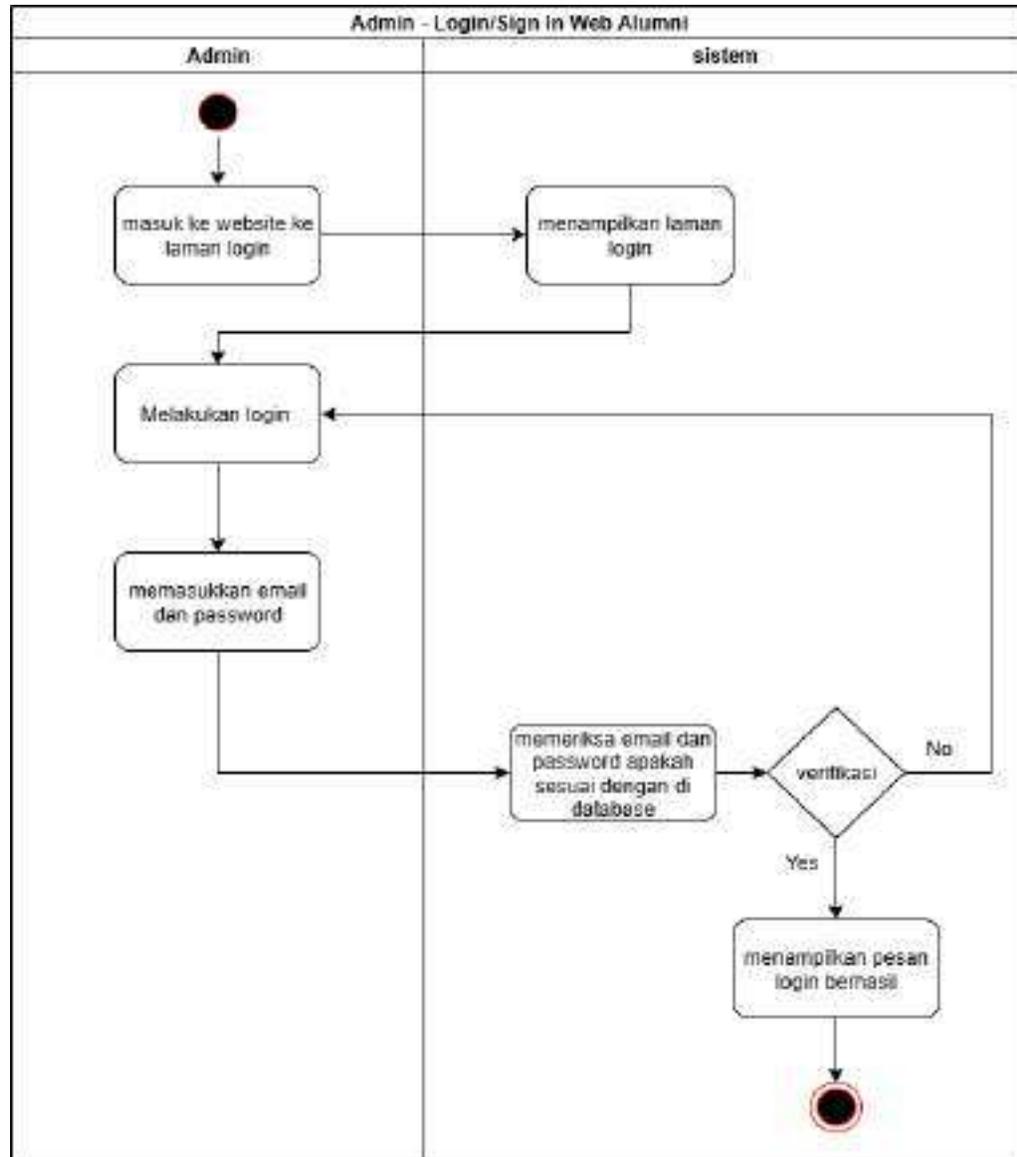
7. Melihat Direktori Alumni: Alumni dapat melihat daftar dan informasi sesama alumni yang terdaftar.
- iii. Pengunjung atau Tamu adalah pengguna umum yang mengakses sistem tanpa perlu login. Akses mereka terbatas pada melihat informasi publik.
1. Melakukan Registrasi Akun (Jika Alumni): Pengunjung yang merupakan alumni dapat melakukan pendaftaran untuk membuat akun baru.
 2. Melihat Profil Alumni: Dapat melihat profil publik dari para alumni.
 3. Melihat Content Event: Dapat melihat informasi acara yang dipublikasikan.
 4. Melihat Content Blogs: Dapat membaca artikel-artikel yang ada di blog.
 5. Melihat Content Announcement: Dapat melihat pengumuman yang dibagikan untuk umum.
 6. Melihat Contact: Dapat mengakses halaman kontak, yang memungkinkan mereka untuk:
 7. Mengirim Pesan ke Admin: Mengirimkan pertanyaan atau pesan kepada Admin

d. Perancangan *Activity Diagram*

c. Admin Login

Diagram aktivitas ini menggambarkan proses login pengguna (Admin) ke dalam sistem web. Alur dimulai saat pengguna mengakses halaman login. Pengguna kemudian memasukkan email dan password. Sistem akan memverifikasi kredensial tersebut dengan data di database. Jika data sesuai (valid), pengguna berhasil login. Jika tidak sesuai, sistem akan mengarahkan pengguna untuk mencoba login kembali.

Diagram ini menunjukkan interaksi antara pengguna dan sistem untuk proses autentikasi.



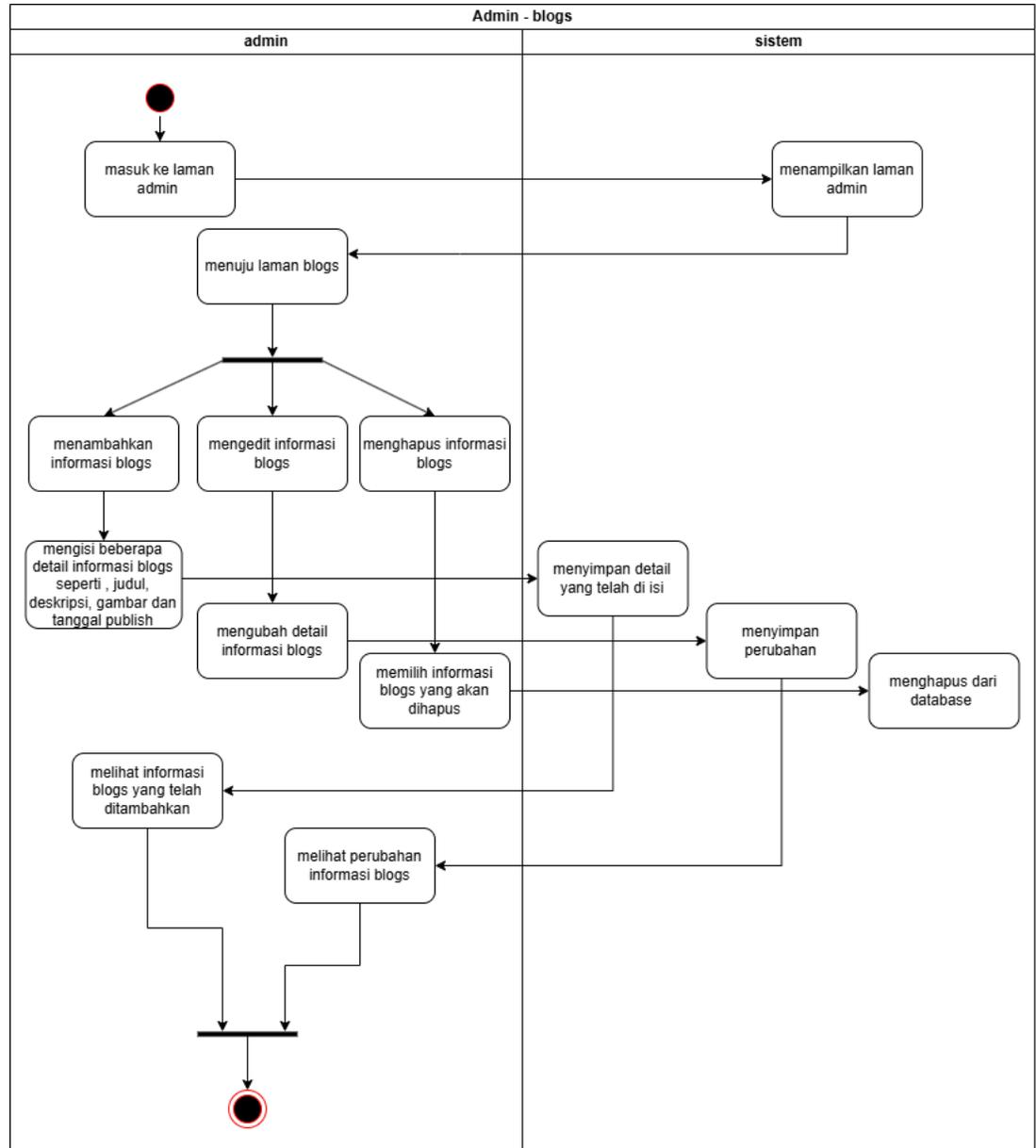
Gambar 3. 45 Diagram Activity Login-Admin

d. Admin – Manajemen Blog

Diagram aktivitas ini menggambarkan alur kerja admin saat mengelola blog.

Prosesnya dimulai saat admin masuk ke laman blog, di mana ia dapat memilih untuk

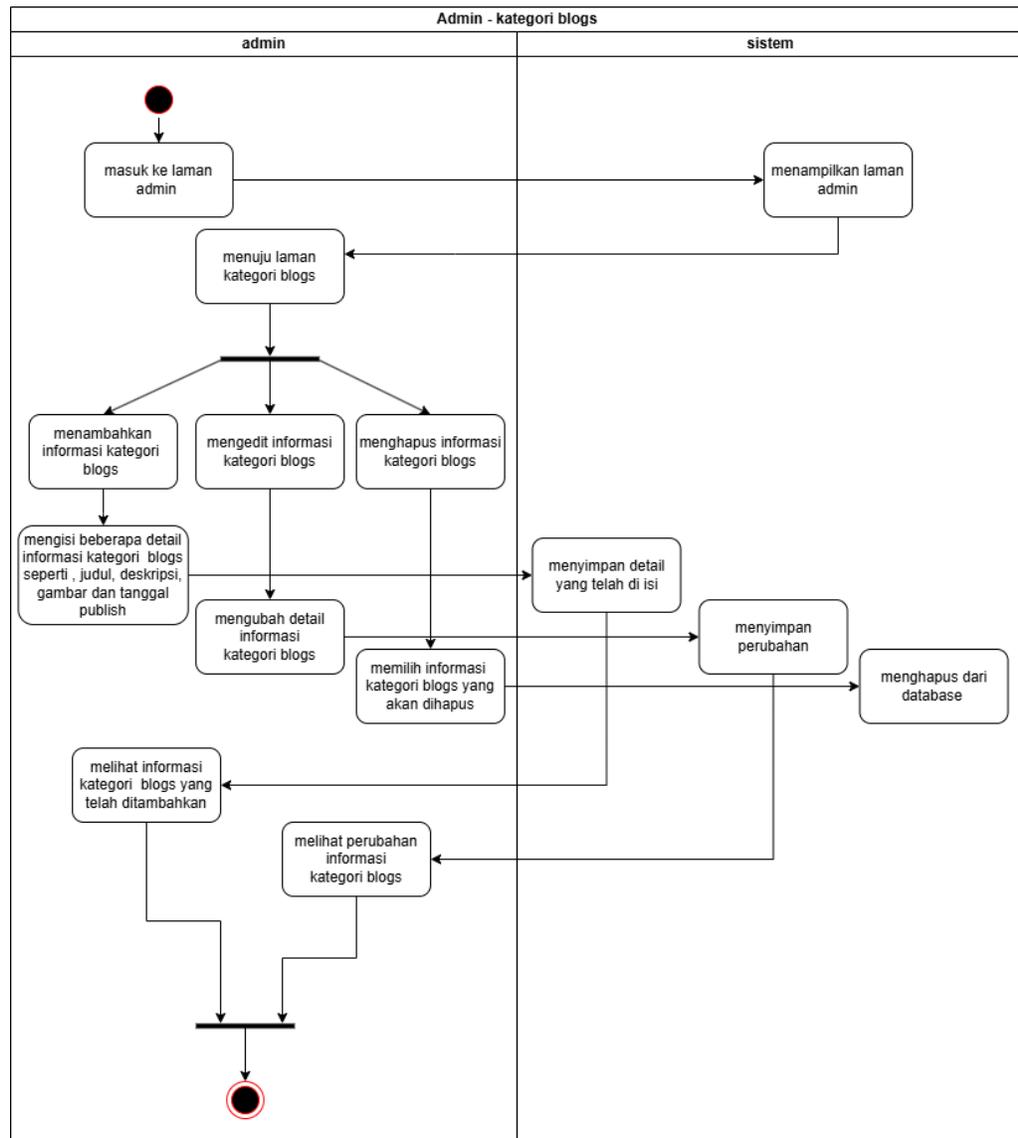
menambah artikel baru, mengedit artikel yang sudah ada, ataupun menghapus sebuah artikel. Setiap tindakan yang dipilih oleh admin akan ditindaklanjuti oleh sistem dengan menyimpan data baru, memperbarui perubahan, atau menghapus data tersebut dari database.



Gambar 3. 46 Diagram Activity Manajemen Blog - Admin

e. Admin – Manajemen Kategori Blog

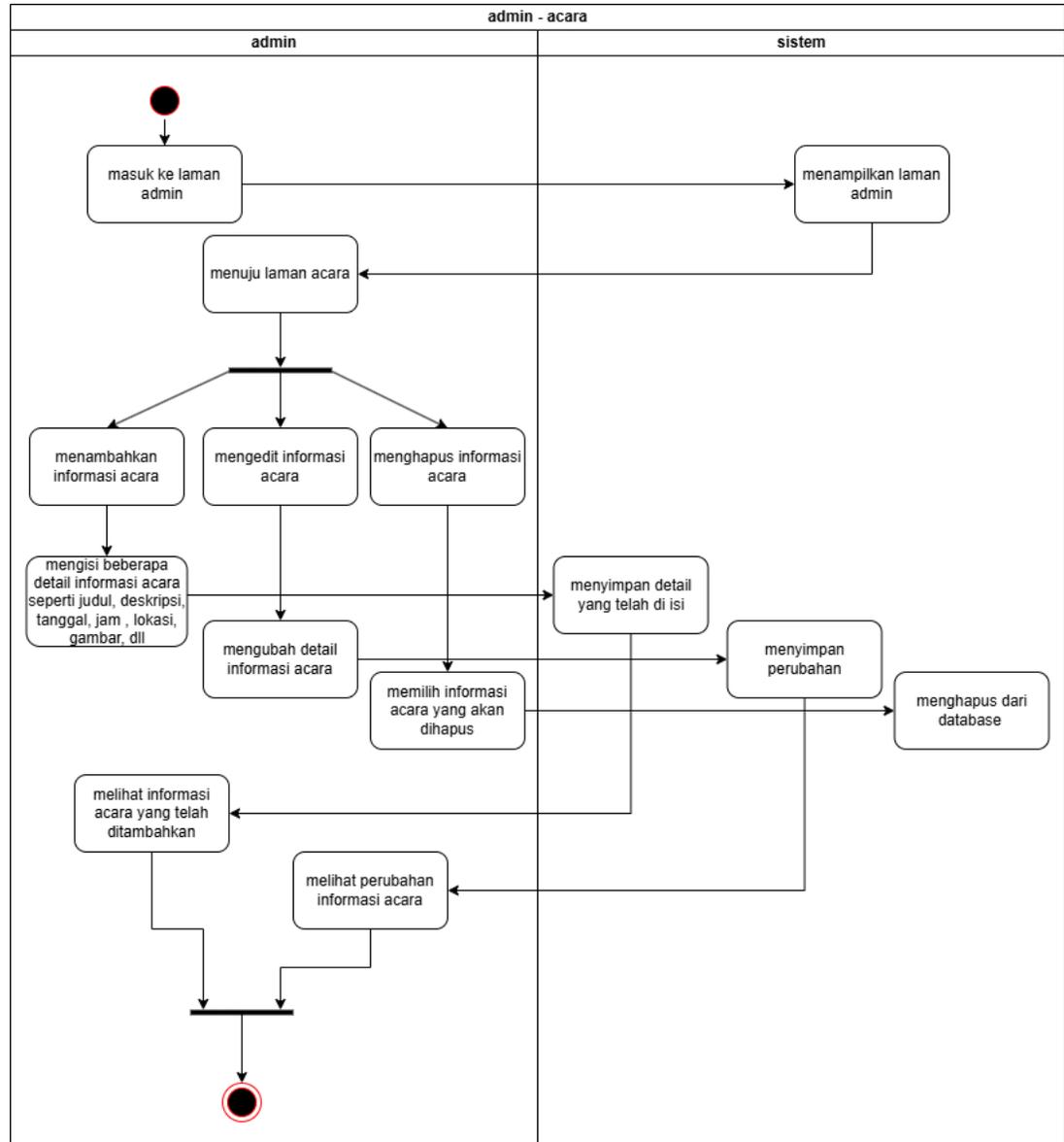
Diagram aktivitas ini menunjukkan alur kerja admin dalam mengelola kategori blog, mulai dari menambah, mengedit, hingga menghapus kategori. Setiap aksi akan diproses oleh sistem untuk menyimpan, memperbarui, atau menghapus data di database.



Gambar 3. 47 Manajemen Kategori Blog - Admin

f. Admin – Management Acara

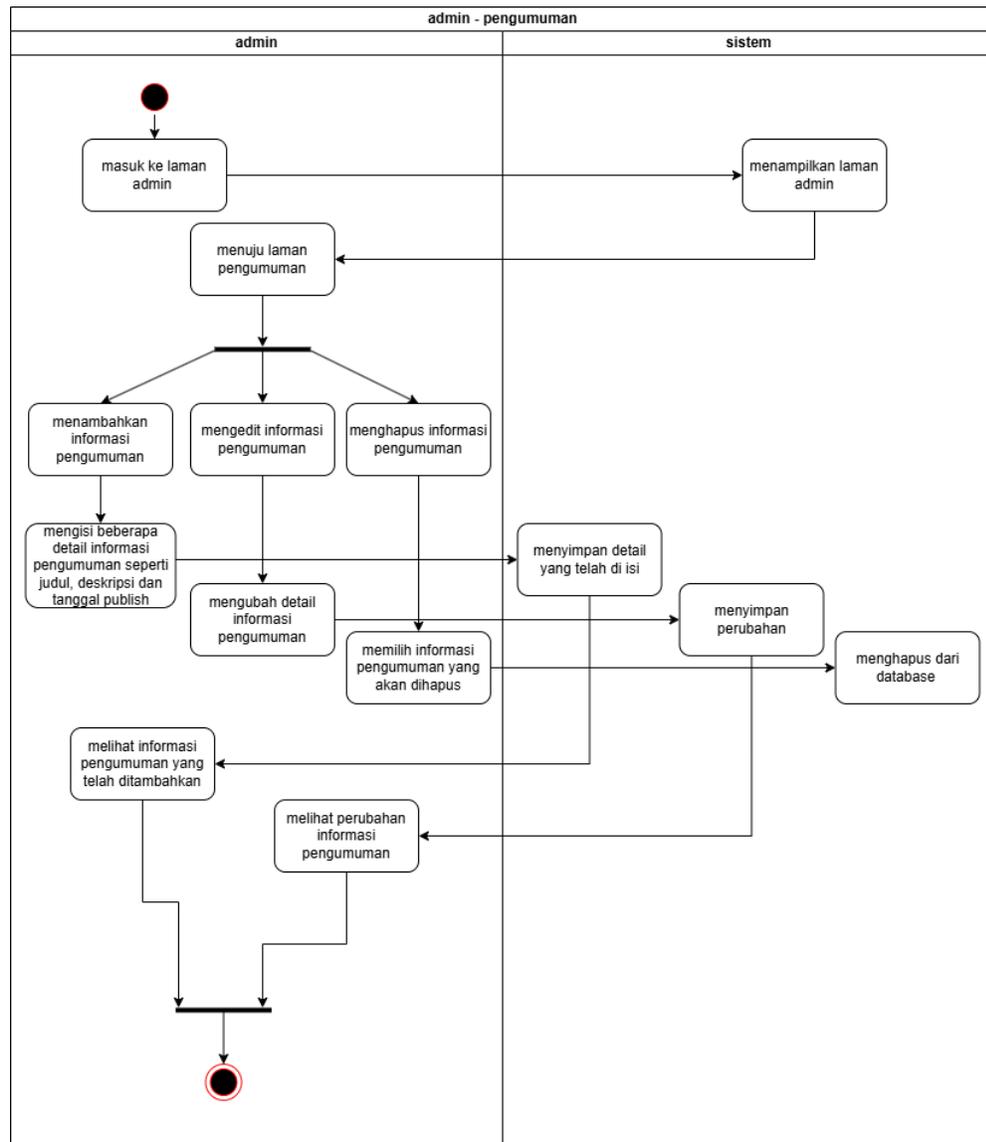
Diagram aktivitas ini menunjukkan alur kerja admin dalam mengelola data acara, mulai dari menambah, mengedit, hingga menghapus acara. Setiap aksi diproses sistem untuk menyimpan, memperbarui, atau menghapus data di database.



Gambar 3. 48 Diagram Activity Manajemen Acara - Admin

g. Admin – Manajemen Pengumuman

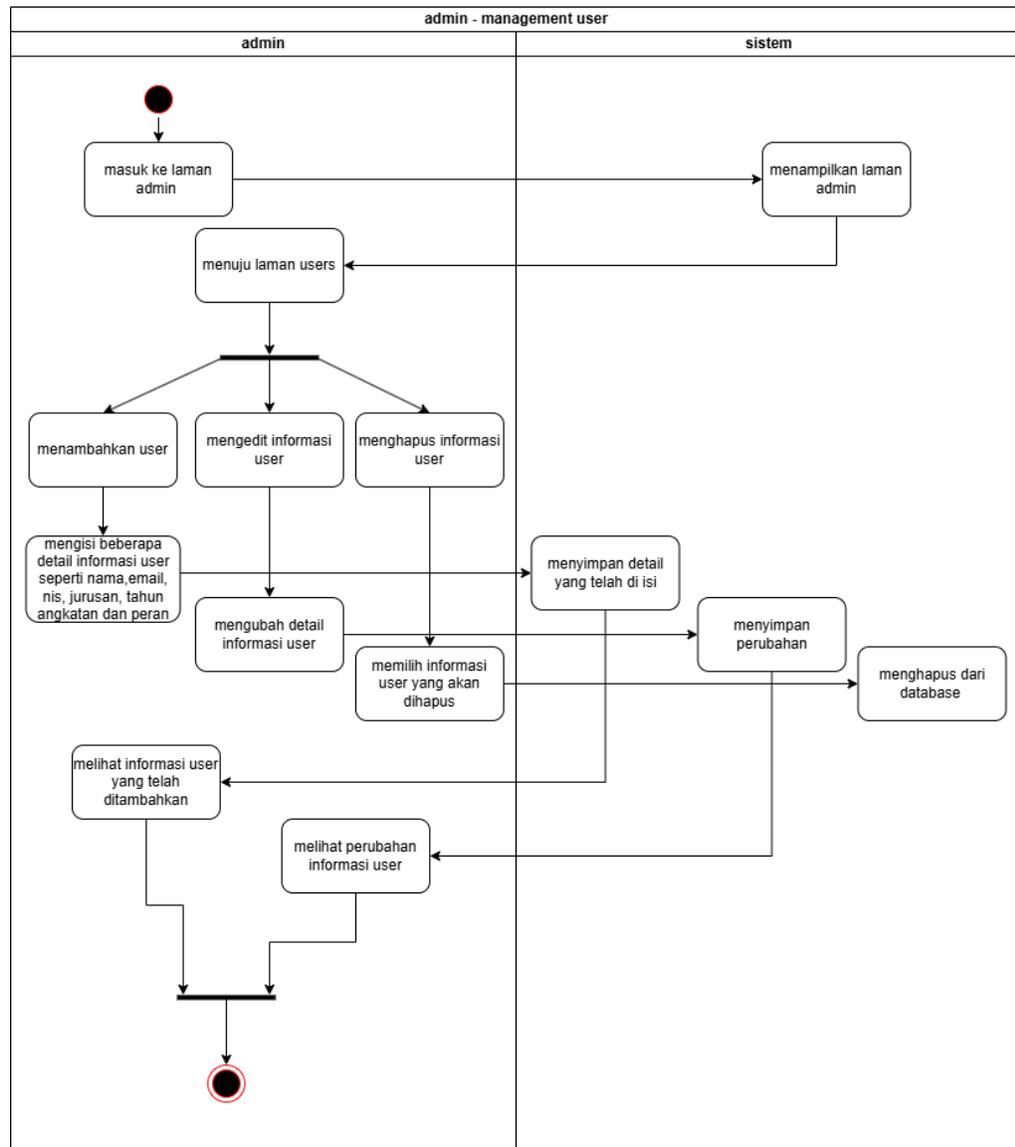
Diagram aktivitas ini menunjukkan alur kerja admin dalam mengelola pengumuman, mulai dari menambah, mengedit, hingga menghapus pengumuman. Setiap Tindakan diproses system untuk menyimpan, memperbarui, atau menghapus data dari database.



Gambar 3. 49 Diagram Activity Manajemen Pengumuman - Admin

h. Admin – Manajemen User

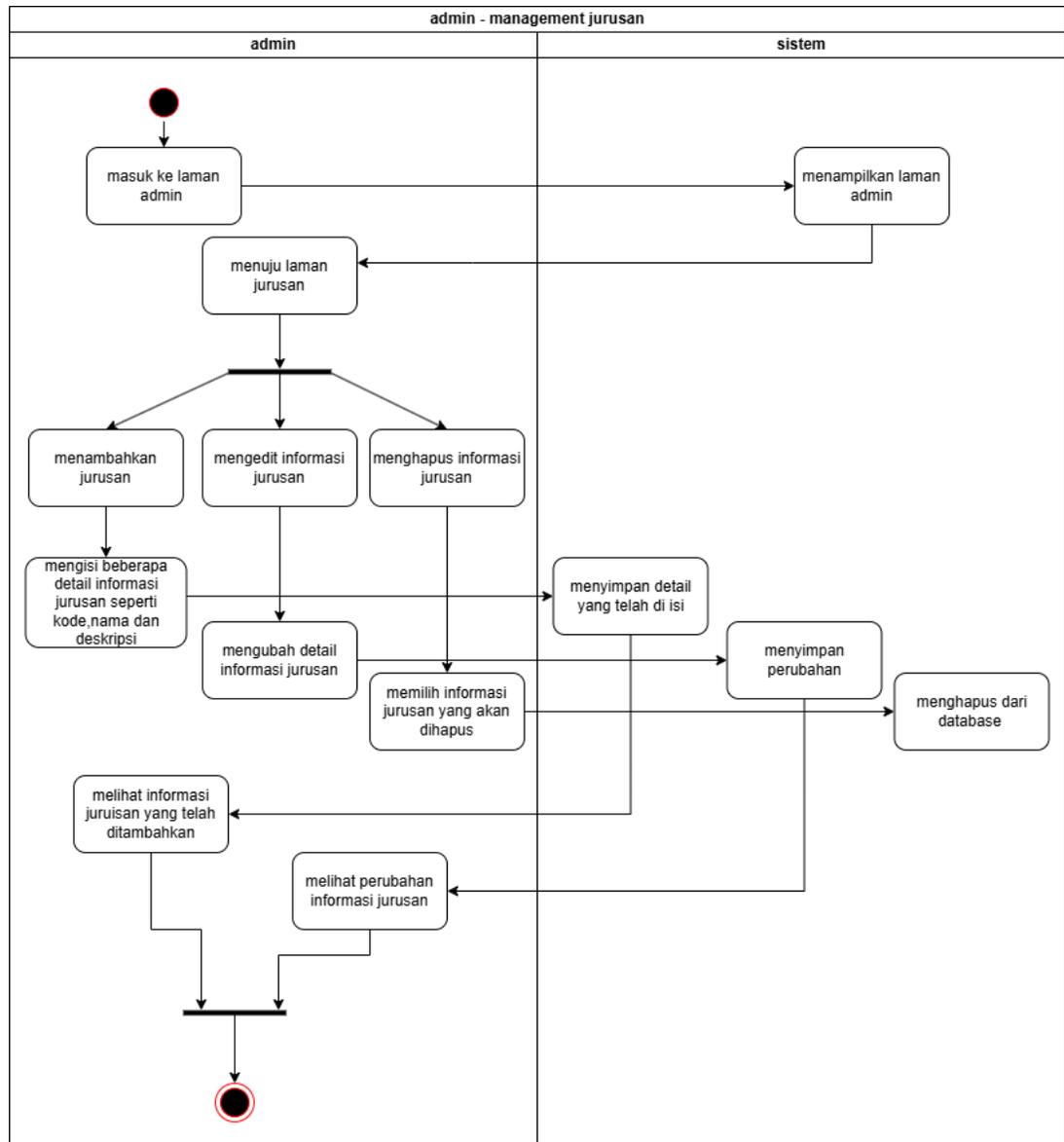
Diagram aktivitas ini menunjukkan alur kerja admin dalam mengelola data pengguna, mulai dari menambah, mengedit, hingga menghapus pengguna. Setiap Tindakan diproses sistem untuk menyimpan, memperbarui, atau menghapus data dari database.



Gambar 3. 50 Diagram Activity Manejemen User - Admin

i. Admin – Manajemen Jurusan

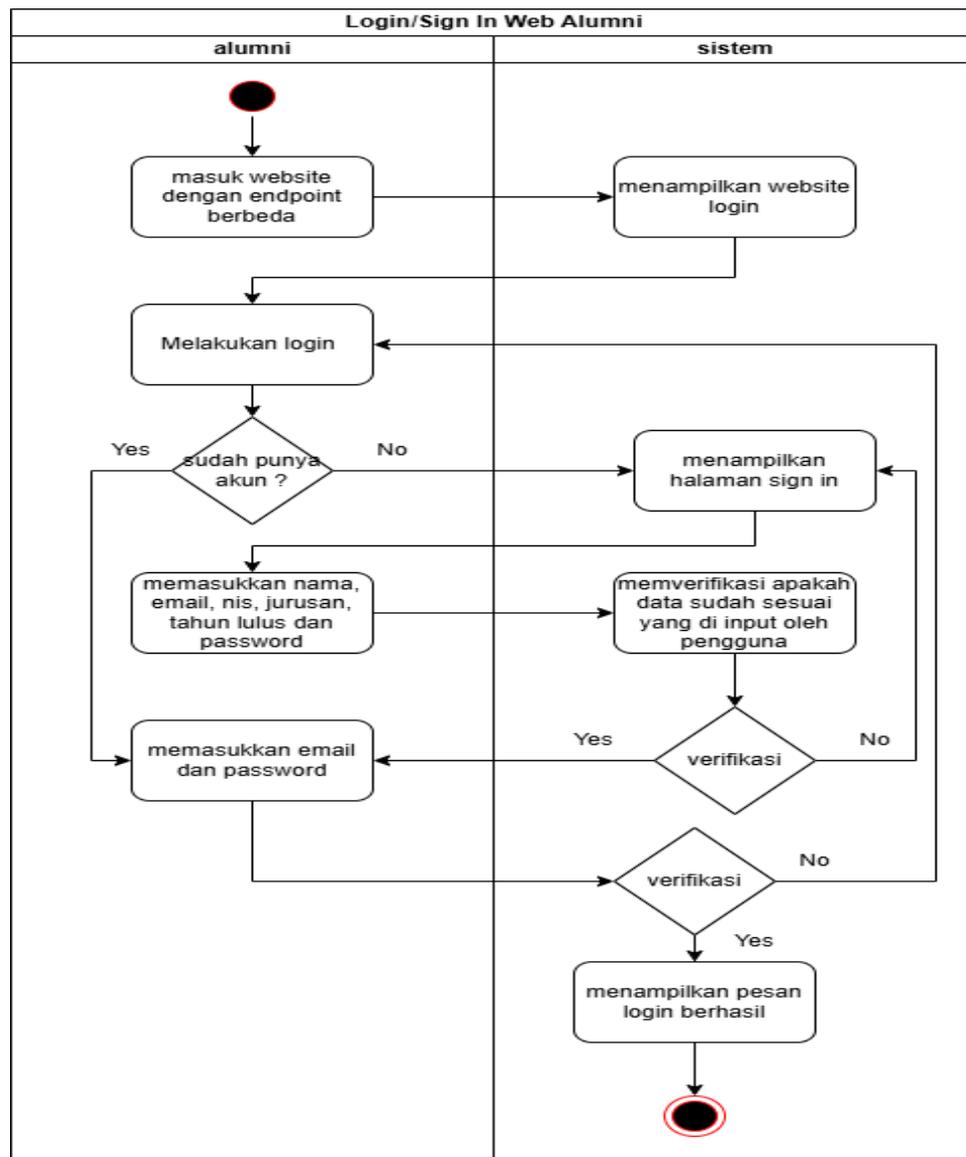
Diagram aktivitas ini menunjukkan alur kerja admin dalam mengelola data jurusan, mulai dari menambah, mengedit, hingga menghapus jurusan. Setiap tindakan diproses sistem untuk menyimpan, memperbarui, atau menghapus data dari database.



Gambar 3. 51 Diagram Activity Manajemen Jurusan - Admin

j. Alumni – Login/Register

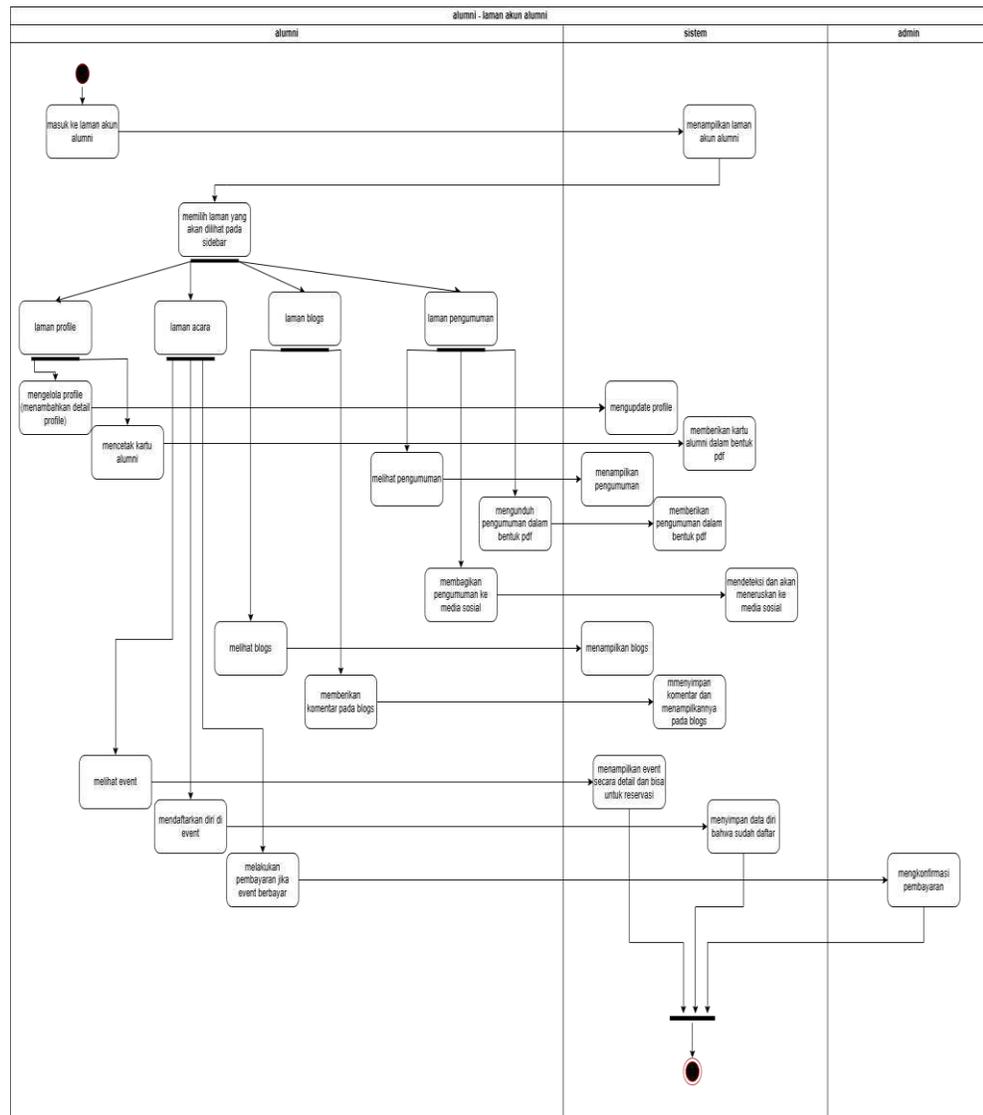
Diagram aktivitas ini menggambarkan alur login dan registrasi alumni. Alumni memilih login jika sudah punya akun, atau registrasi jika belum. Data yang dimasukkan akan diverifikasi oleh sistem. Jika gagal, pengguna diminta mengulang; jika berhasil, sistem menampilkan pesan login berhasil.



Gambar 3. 52 Diagram Activity Login/Register - Alumni

k. Alumni – Laman Akun Alumni

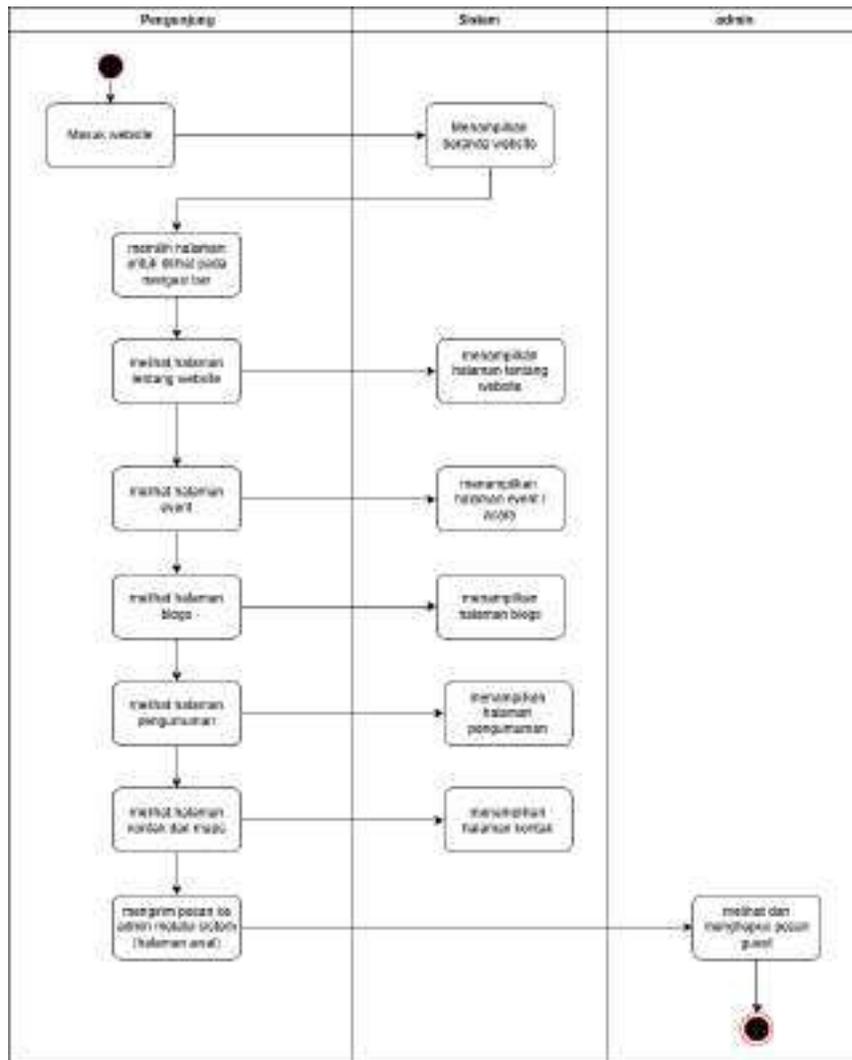
Diagram aktivitas ini menggambarkan fitur-fitur yang dapat diakses alumni setelah login, seperti mengelola profil, mencetak kartu anggota, mendaftar acara (dengan konfirmasi admin untuk pembayaran), berinteraksi di blog melalui komentar, serta melihat, mengunduh, atau membagikan pengumuman dari sistem.



Gambar 3. 53 Diagram Activity Laman Akun Alumni

I. Guest – Akses Halaman Utama

Diagram aktivitas ini menggambarkan alur interaksi pengunjung yang belum login. Pengunjung dapat mengakses halaman publik seperti tentang kami, event, blog, pengumuman, dan kontak, serta mengirim pesan kepada admin yang kemudian akan ditangani oleh pihak admin.



Gambar 3. 54 Diagram Activity Halaman Utama - Guest

Class diagram ini menggambarkan struktur sistem informasi yang digunakan untuk mengelola data alumni, admin, dan konten institusi seperti blog, event, dan pengumuman.

1. Admin adalah pusat kendali yang memiliki akses untuk menambahkan, melihat, mengedit, dan menghapus data pada hampir seluruh entitas dalam sistem.
2. Alumni merupakan pengguna utama yang dapat mendaftar, login, memperbarui profil, dan mengikuti event.
3. Profile berisi data pribadi lengkap dari alumni seperti alamat, pekerjaan, dan kontak.
4. Jurusan berfungsi untuk mengelompokkan alumni berdasarkan program studi.
5. Blogs, Events, dan Pengumuman adalah fitur konten yang dapat dikelola oleh admin untuk menyampaikan informasi kepada alumni.
6. Tentang Kami menyajikan informasi umum mengenai institusi.
7. Management Users digunakan untuk mengelola akun pengguna selain alumni, seperti staf atau pihak jurusan.

Relasi antar class menunjukkan keterkaitan data, seperti satu alumni memiliki satu profil, satu jurusan memiliki banyak alumni, dan satu event bisa diikuti oleh banyak alumni.

3. Realisasi UI/UX

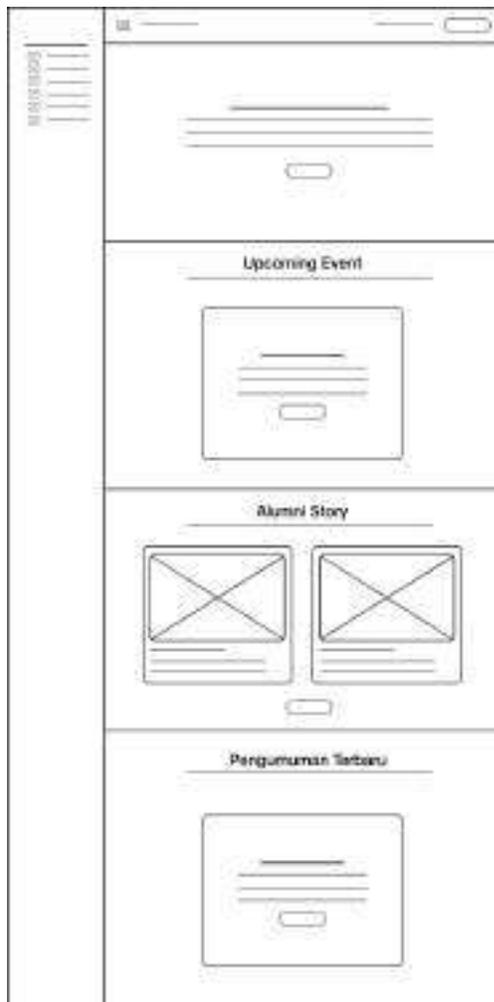
Dalam pengerjaan proyek ini, saya bersama tim saya sudah menentukan tema, pembuatan menu navigasi, dan tim kami sepakat dengan tema berwarna kuning untuk halaman utama, warna biru dan merah untuk halaman dashboard, tema kuning diambil dari logo ganissa yang cenderung ke warna kuning.



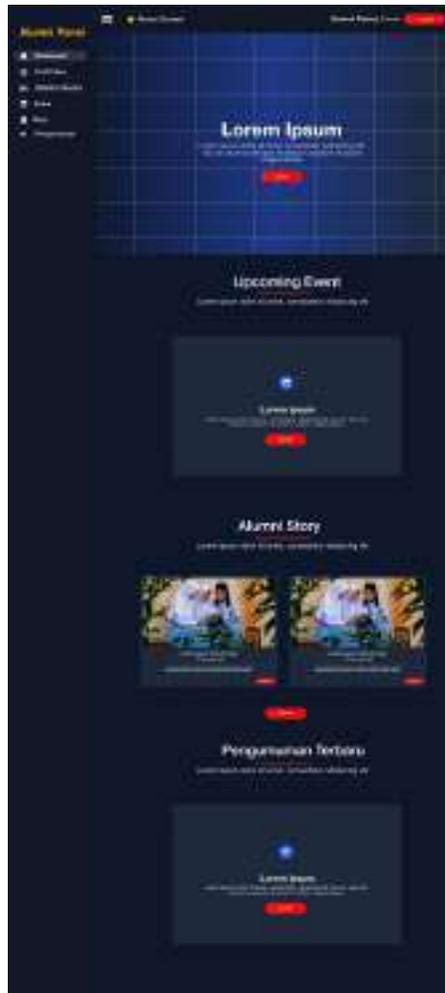
Gambar 3. 56 Desain awal website alumni bertema kuning

a. Halaman Alumni

Halaman Dashboard berfungsi sebagai pusat informasi utama bagi alumni, menampilkan ringkasan berbagai bagian penting. Fitur utamanya mencakup bagian untuk "Upcoming Event" yang menampilkan acara-acara mendatang, "Alumni Story" untuk berbagi kisah sukses alumni, dan "Pengumuman Terbaru" untuk informasi terkini. Ini memungkinkan alumni untuk mendapatkan gambaran cepat tentang aktivitas dan berita terbaru dalam komunitas mereka.

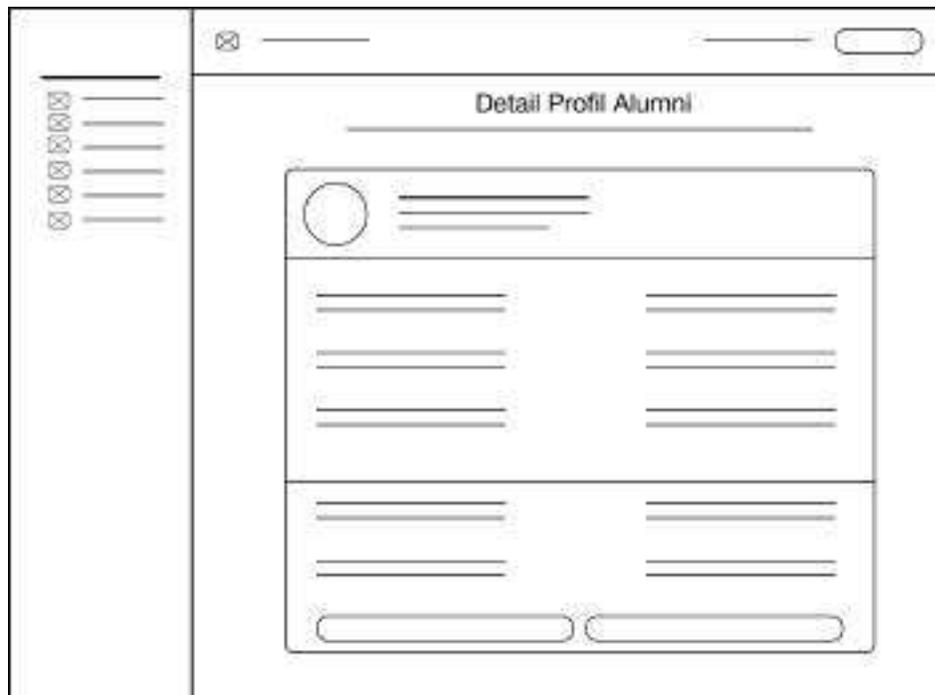


Gambar 3. 57 Wireframe Laman Alumni

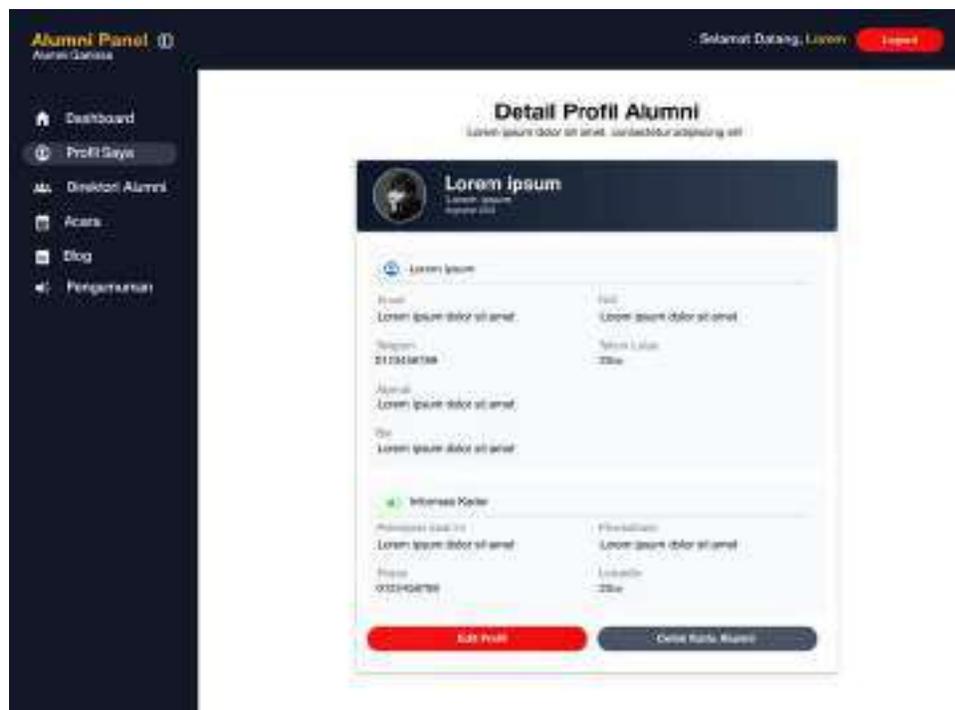


Gambar 3. 58 Mockup Laman Alumni

Halaman Profil Saya (Frame 4.png) Halaman "Profil Saya" adalah tempat bagi setiap alumni untuk mengelola dan melihat informasi pribadi mereka. Fitur-fitur yang tersedia meliputi tampilan informasi dasar (seperti email, NIS, telepon, alamat, tahun lulus), serta detail karier (pekerjaan saat ini, perusahaan, posisi, dan tautan LinkedIn). Pengguna juga memiliki kemampuan untuk "Edit Profil" untuk memperbarui data mereka, dan "Cetak Kartu Alumni" untuk keperluan identifikasi.

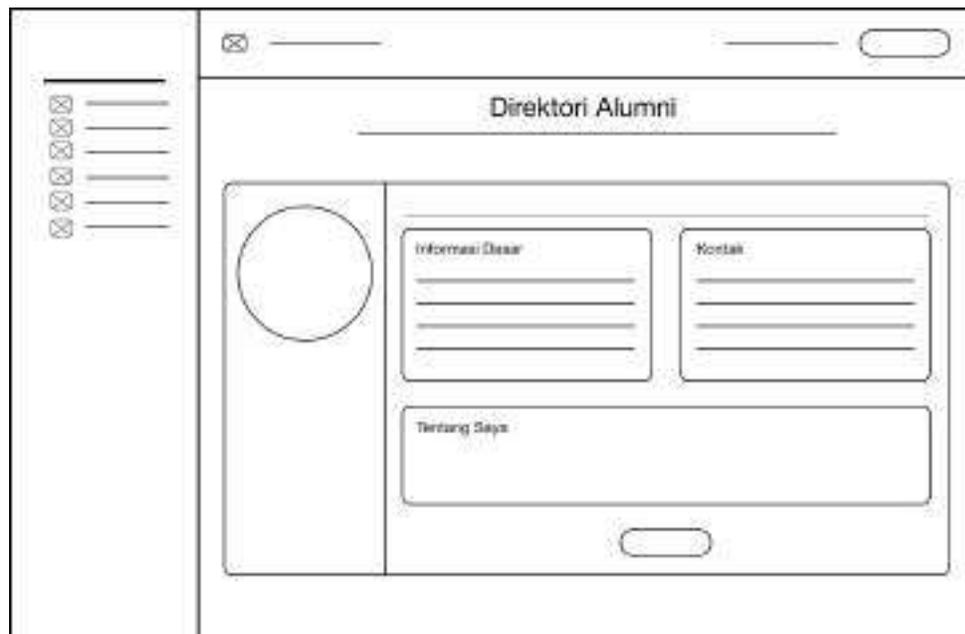


Gambar 3. 59 Wireframe Profil Alumni

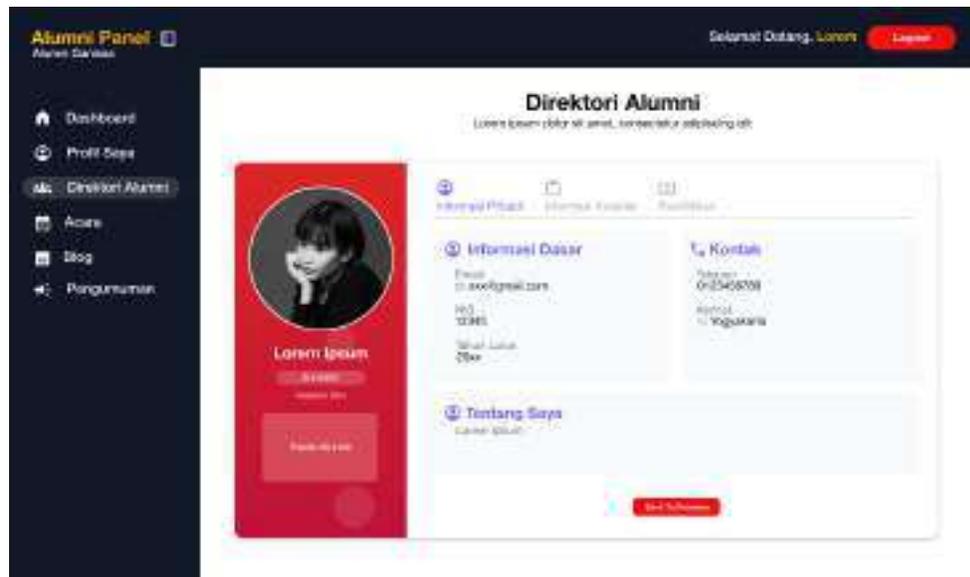


Gambar 3. 60 Mockup Profil Alumni

Halaman Direktori Alumni adalah alat untuk mencari dan menemukan alumni lain dalam komunitas. Fitur yang tersedia memungkinkan pengguna untuk melakukan pencarian berdasarkan nama, serta memfilter hasil berdasarkan "Jurusan" dan "Tahun Lulus". Ini sangat berguna untuk membangun jaringan, mencari kontak lama, atau sekadar melihat profil anggota lain dalam komunitas.

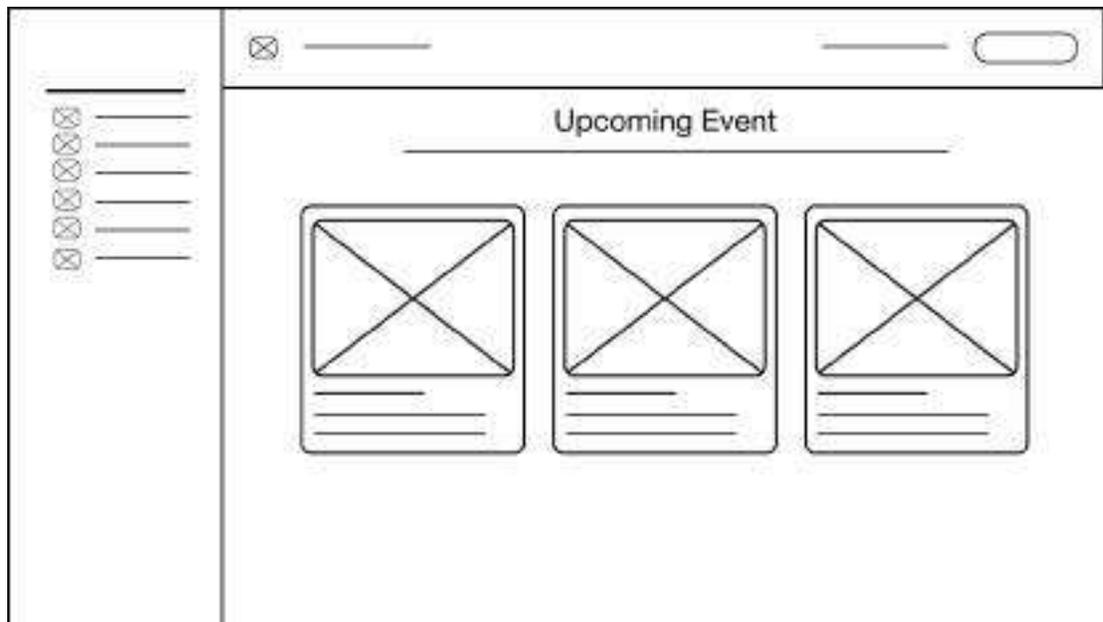


Gambar 3. 61 Wireframe Direktori Alumni

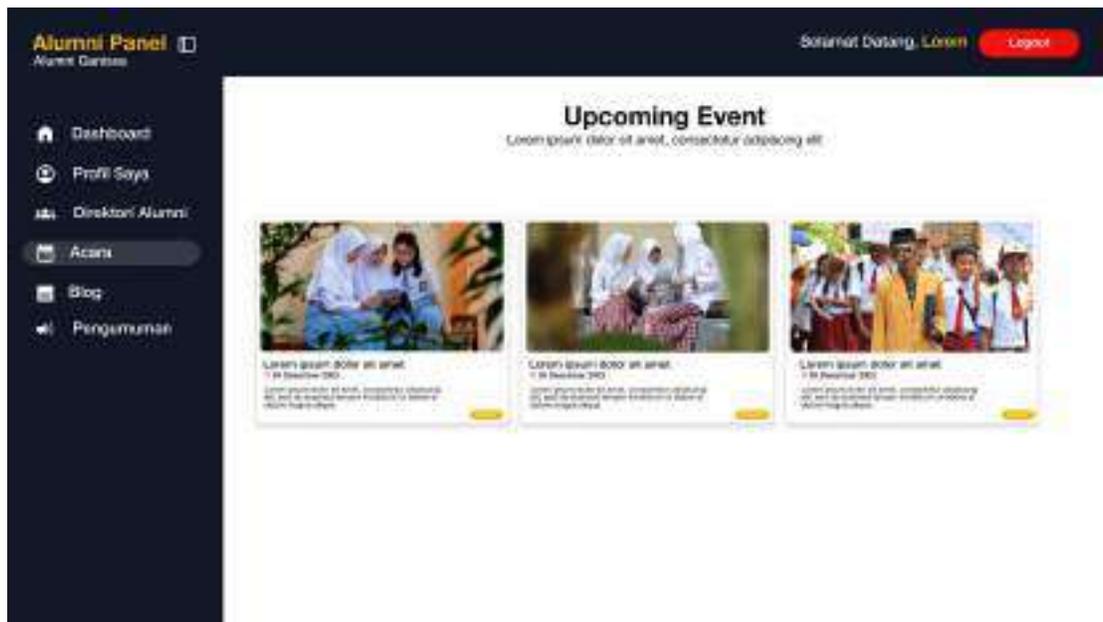


Gambar 3. 62 Mockup Direktori Alumni

Halaman Acara berfungsi untuk memberitahukan alumni tentang berbagai kegiatan yang akan datang. Fitur utamanya adalah menampilkan daftar acara, di mana setiap acara memiliki judul dan tanggal pelaksanaan. Ini berguna agar alumni dapat dengan mudah menemukan dan merencanakan partisipasi mereka dalam kegiatan-kegiatan komunitas.



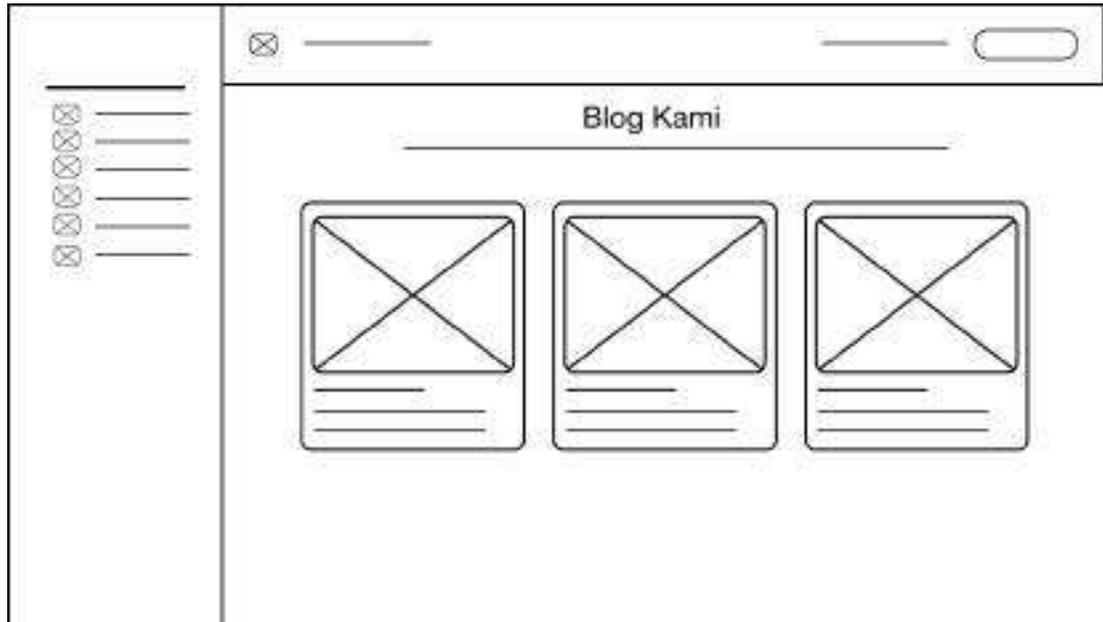
Gambar 3. 63 Wireframe Acara



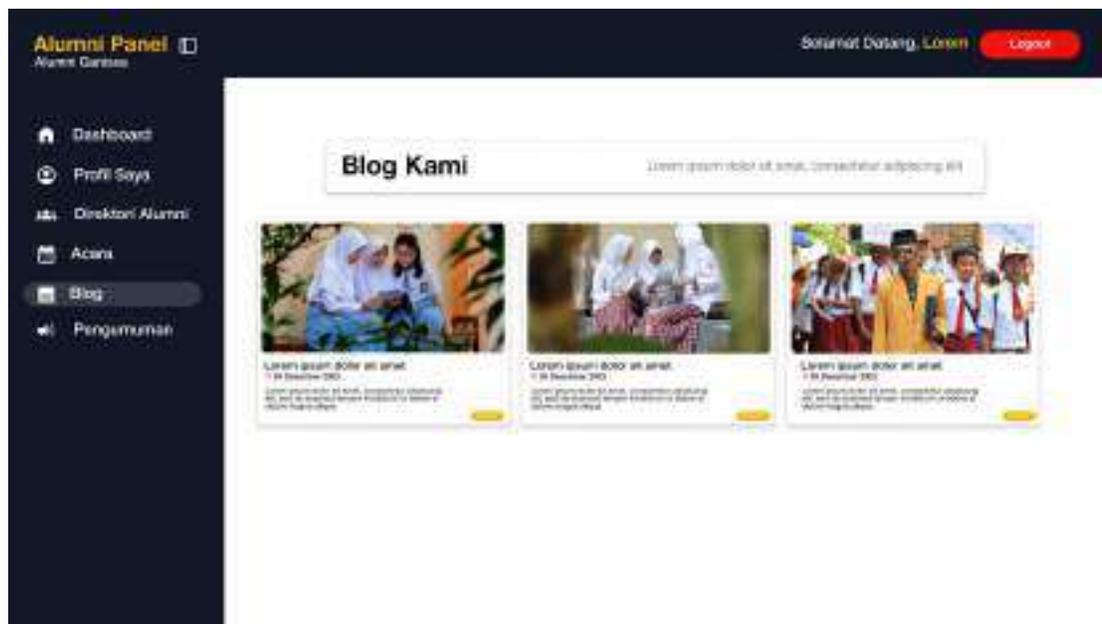
Gambar 3. 64 Mockup Acara

Halaman Blog menyajikan kumpulan artikel atau postingan berita yang relevan dengan komunitas alumni. Fitur utamanya adalah menampilkan daftar konten blog, di mana

setiap entri memiliki judul dan tanggal publikasi. Kegunaannya adalah sebagai sarana informasi dan inspirasi, memungkinkan alumni untuk membaca berbagai topik yang menarik atau mendapatkan *update* terbaru dari organisasi.

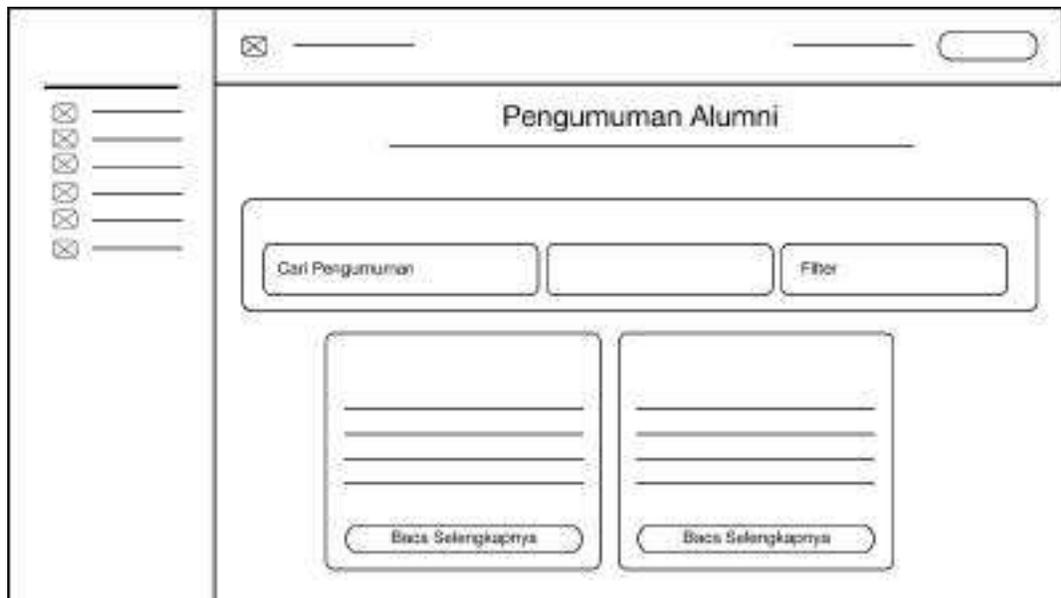


Gambar 3. 65 Wireframe Blog

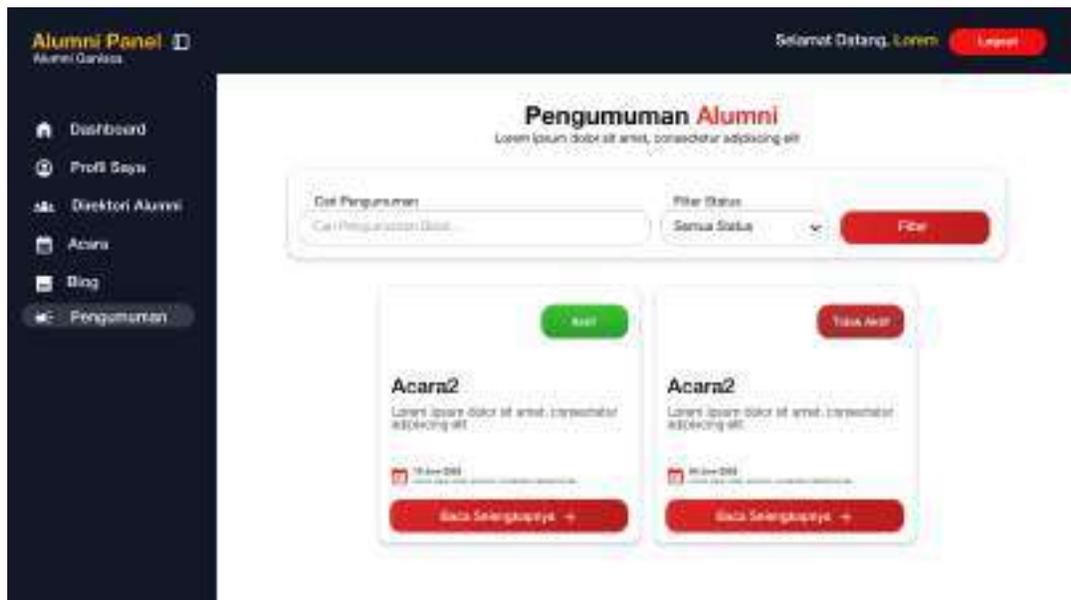


Gambar 3. 66 Mockup Blog

Halaman Pengumuman didedikasikan untuk menyebarkan informasi resmi dan penting kepada seluruh anggota. Fitur-fiturnya memungkinkan pengguna untuk mencari pengumuman tertentu menggunakan *field* pencarian dan memfilter berdasarkan status (Aktif atau Tidak Aktif). Setiap pengumuman ditampilkan secara ringkas dengan judul, deskripsi singkat, dan tanggal, serta opsi "Baca Selengkapnya" untuk melihat detail penuh.



Gambar 3. 67 Wireframe Pengumuman



Gambar 3. 68 Mockup Pengumuman

b. Halaman Admin

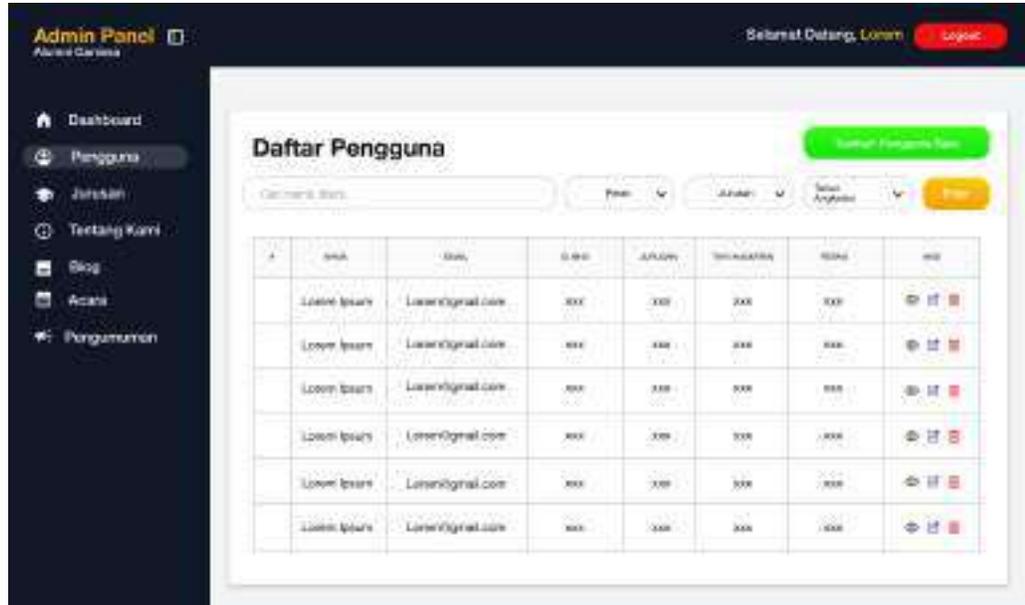
Halaman Dashboard pada Admin Panel berfungsi sebagai pusat kontrol utama yang menampilkan ringkasan data penting. Fitur-fiturnya meliputi statistik jumlah "Blogs", "Events", "Announcements", dan "Users", yang masing-masing dilengkapi dengan tombol untuk pengelolaan cepat. Bagian "Aktivitas Terbaru" menampilkan entri terbaru dari blog, acara, dan pengumuman, sementara "Tindakan Cepat" menyediakan tombol akses langsung untuk membuat konten baru. Kegunaannya adalah untuk memberikan *overview* performa platform dan akses cepat ke fungsi-fungsi administratif.



Gambar 3. 69 Mockup Dashboard Admin

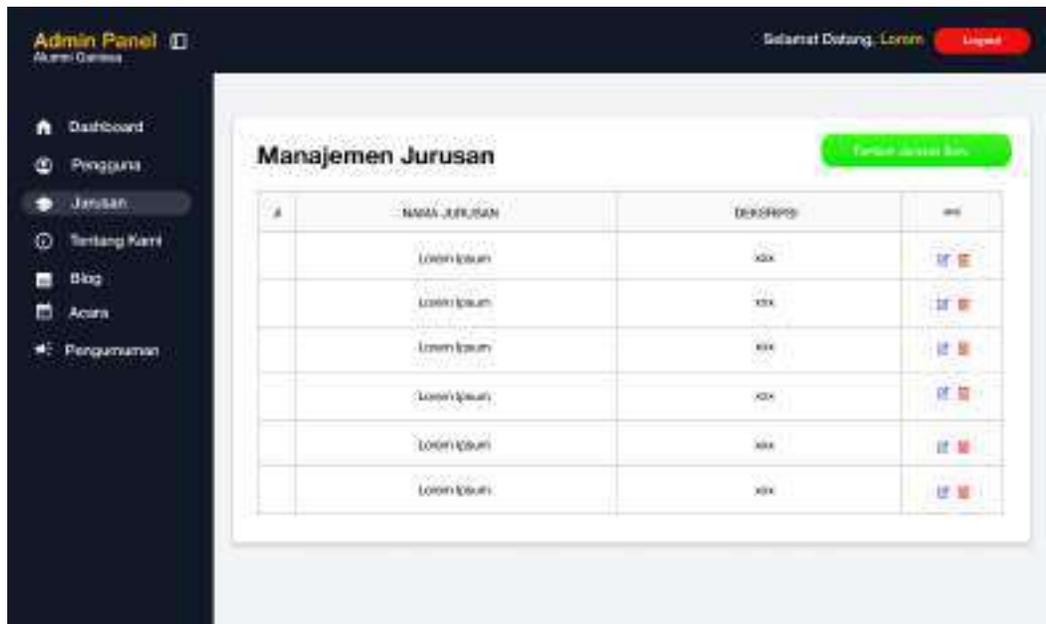
Halaman "Daftar Pengguna" memungkinkan administrator untuk mengelola seluruh akun pengguna dalam sistem. Fitur utamanya adalah tabel yang menampilkan data pengguna seperti nama, email, ID Mahasiswa, Jurusan, Tahun Angkatan, dan Peran. Dilengkapi dengan fungsi pencarian, serta filter berdasarkan "Peran", "Jurusan", dan "Tahun Angkatan", administrator dapat dengan mudah menemukan pengguna tertentu.

Setiap baris memiliki opsi "Aksi" untuk melihat detail, mengedit, atau menghapus pengguna, serta tombol "Tambah Pengguna Baru" untuk registrasi akun baru.



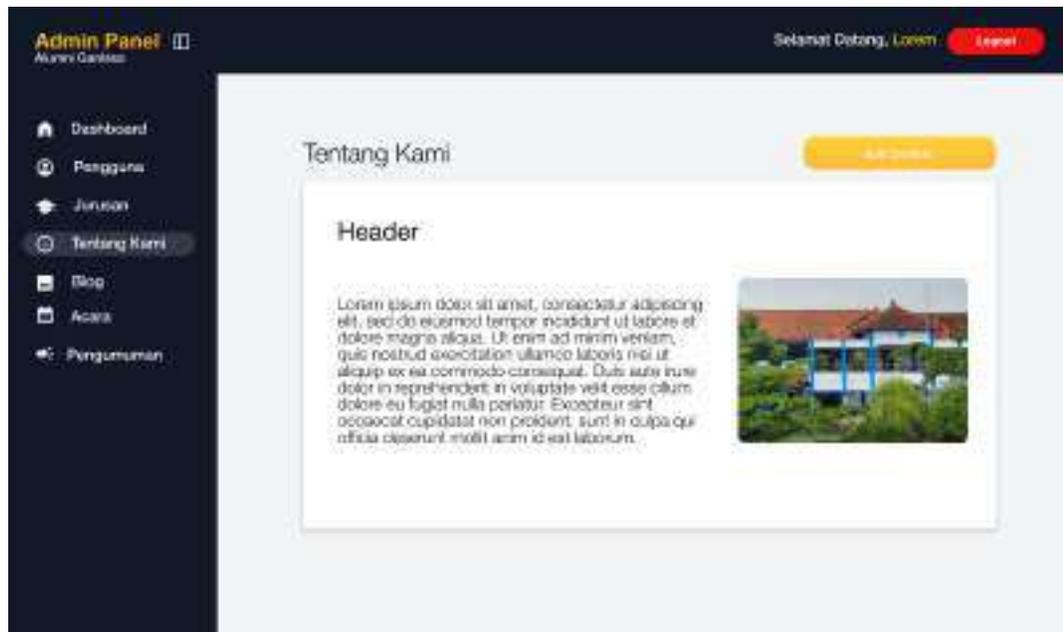
Gambar 3. 70 Mockup Daftar Pengguna

Halaman "Manajemen Jurusan" dirancang untuk administrator dalam mengelola data jurusan yang tersedia. Fitur utamanya adalah tabel yang mencantumkan "Nama Jurusan" dan "Deskripsi" dari setiap jurusan. Kolom "Aksi" menyediakan opsi untuk mengedit atau menghapus entri jurusan. Di bagian atas, tombol "Tambah Jurusan Baru" memungkinkan administrator untuk menambahkan data jurusan baru ke dalam sistem, menjaga informasi jurusan tetap *up-to-date*.



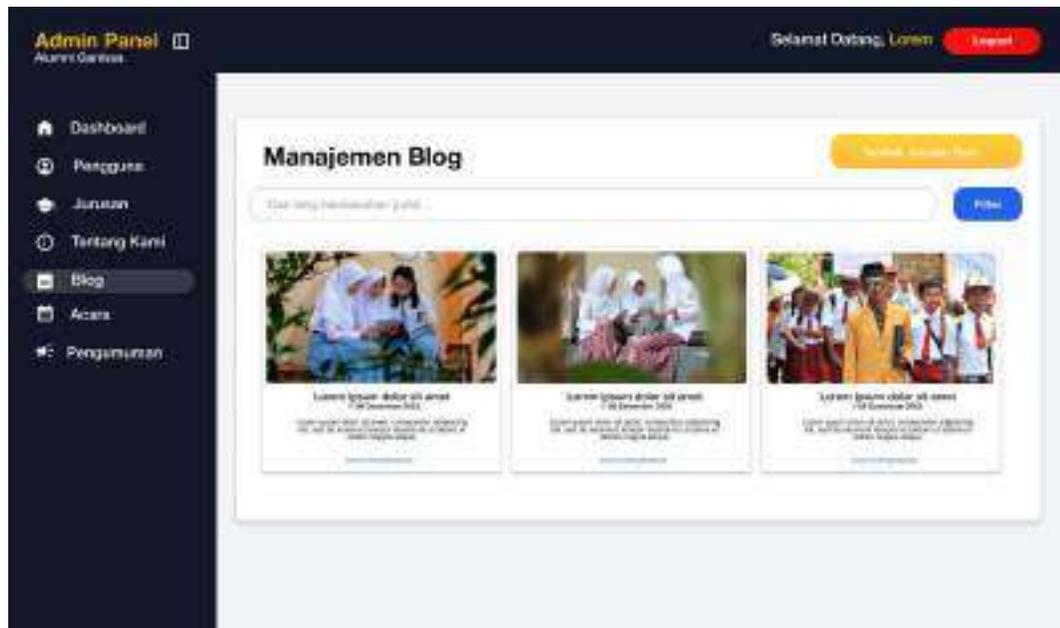
Gambar 3. 71 Mockup Manajemen Jurusan

Halaman "Tentang Kami" memungkinkan administrator untuk mengelola dan memperbarui informasi mengenai organisasi atau platform. Fitur utamanya adalah area konten yang dapat diubah, biasanya berisi teks deskriptif dan gambar. Tombol "Edit Content" disediakan untuk administrator agar dapat dengan mudah mengubah narasi dan informasi yang ditampilkan kepada pengguna akhir mengenai latar belakang, misi, atau visi organisasi.



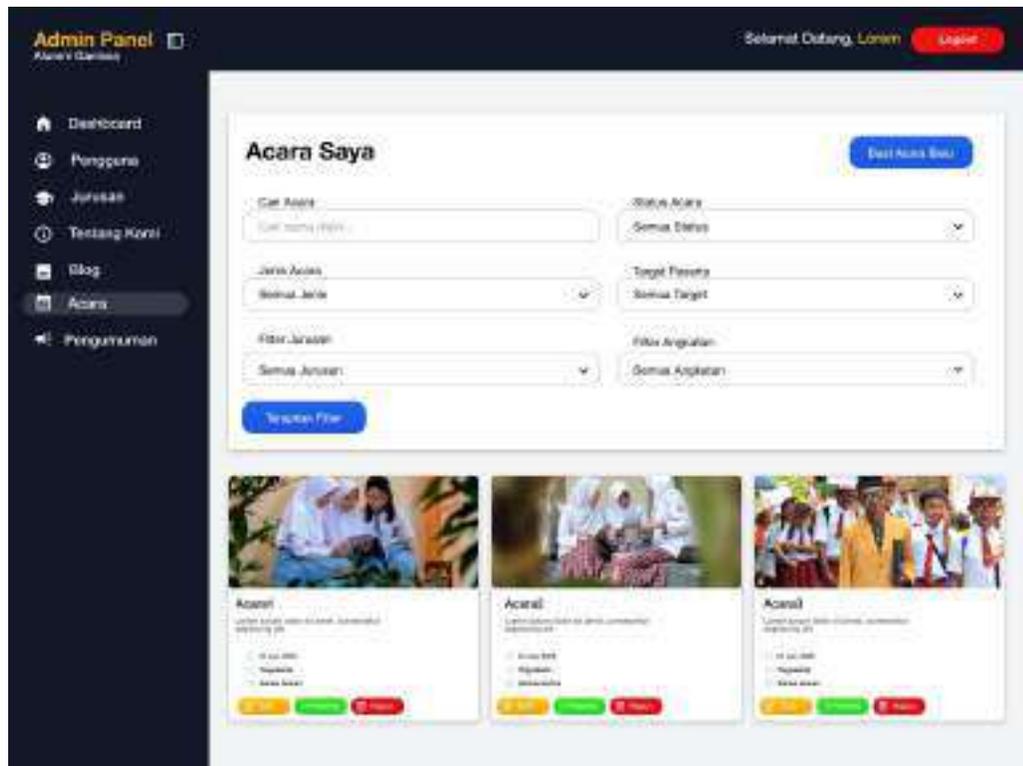
Gambar 3. 72 Mockup Manajemen Tentang Kami

Halaman "Manajemen Blog" adalah tempat bagi administrator untuk mengelola seluruh postingan blog. Fitur utamanya adalah *field* pencarian "Cari blog berdasarkan judul" dan tombol "Filter" untuk membantu menemukan artikel tertentu. Postingan blog ditampilkan dalam format kartu yang mencakup gambar, judul, dan tanggal publikasi, serta opsi "Baca Selengkapnya". Tombol "Tambah Jurusan Baru" (terdapat kesalahan label, seharusnya "Tambah Blog Baru") memungkinkan penambahan artikel baru.



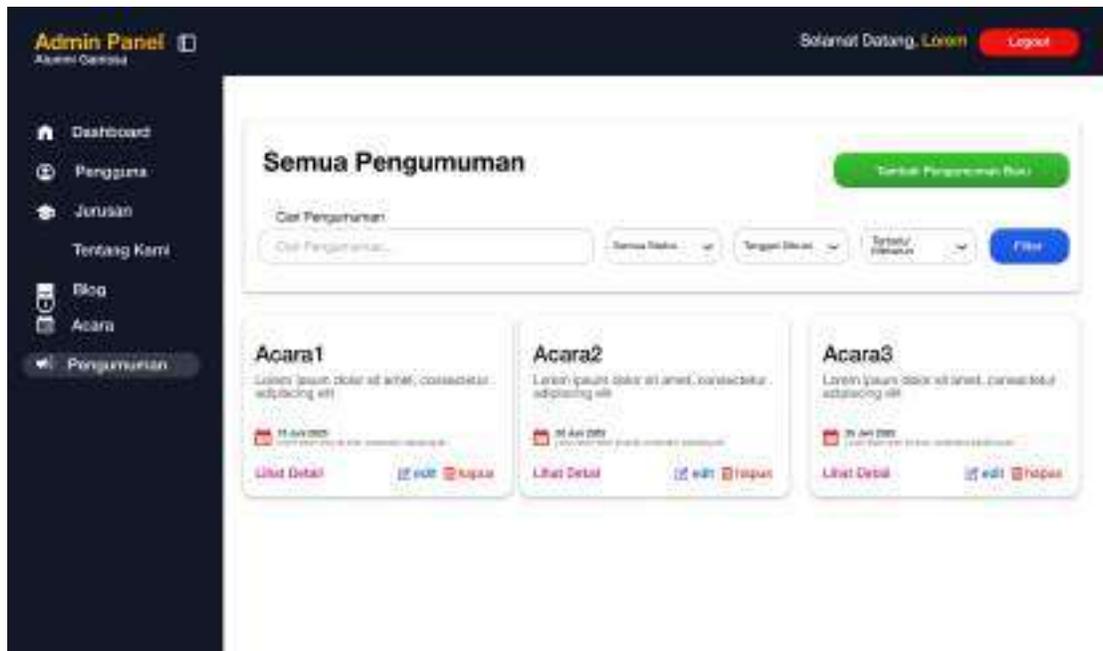
Gambar 3. 73 Mockup Manajemen Blog

Halaman "Acara Saya" memberikan kemampuan kepada administrator untuk mengelola semua acara yang telah dibuat. Fitur-fiturnya meliputi berbagai opsi filter seperti "Cari Acara", "Status Acara", "Jenis Acara", "Target Peserta", "Filter Jurusan", dan "Filter Angkatan", yang dapat diterapkan dengan tombol "Terapkan Filter". Setiap acara ditampilkan dalam bentuk kartu dengan tombol "Edit", "Lihat Peserta", dan "Hapus", serta tombol "Buat Acara Baru" untuk menambah acara.



Gambar 3. 74 Mockup Manajemen Acara

Halaman "Semua Pengumuman" berfungsi sebagai pusat pengelolaan seluruh pengumuman yang diterbitkan. Administrator dapat mencari pengumuman spesifik menggunakan *field* pencarian, serta memfilter berdasarkan "Status" dan "Tanggal Dibuat". Setiap pengumuman disajikan dalam format kartu dengan tanggal, judul, dan opsi "Lihat Detail", "edit", dan "hapus". Tombol "Tambah Pengumuman Baru" memungkinkan administrator untuk menerbitkan informasi baru dengan mudah.



Gambar 3. 75 Mockup Manajemen Pengumuman

4. Realisasi Frontend 1 (NOVAL LIAS RAMADANI)

A. Target User : Guest

1) Halaman profile atau landing page untuk guest



Gambar 3. 76 Tampilan Halaman Landing Page



Gambar 3. 77 Potongan Kode Halaman Landing Page

Penjelasan :

Halaman ini merupakan halaman utama (beranda) dari sebuah situs web portal alumni yang dirancang secara modern dan interaktif. Paragraf pertama, atau bagian hero section, langsung menyambut pengunjung dengan slideshow gambar yang menarik dan slogan yang kuat, yaitu "Terhubung, Tumbuh, Sukses". Tujuannya adalah untuk menarik perhatian dan mengajak pengunjung, khususnya para alumni, untuk mendaftar atau masuk ke dalam sistem guna memperluas jaringan dan kesempatan mereka. Penggunaan Alpine.js untuk carousel dan Tailwind CSS untuk styling menunjukkan pendekatan pengembangan yang modern dan efisien. Selanjutnya, halaman ini menyajikan informasi penting mengenai komunitas alumni dalam beberapa bagian yang terstruktur dengan baik.

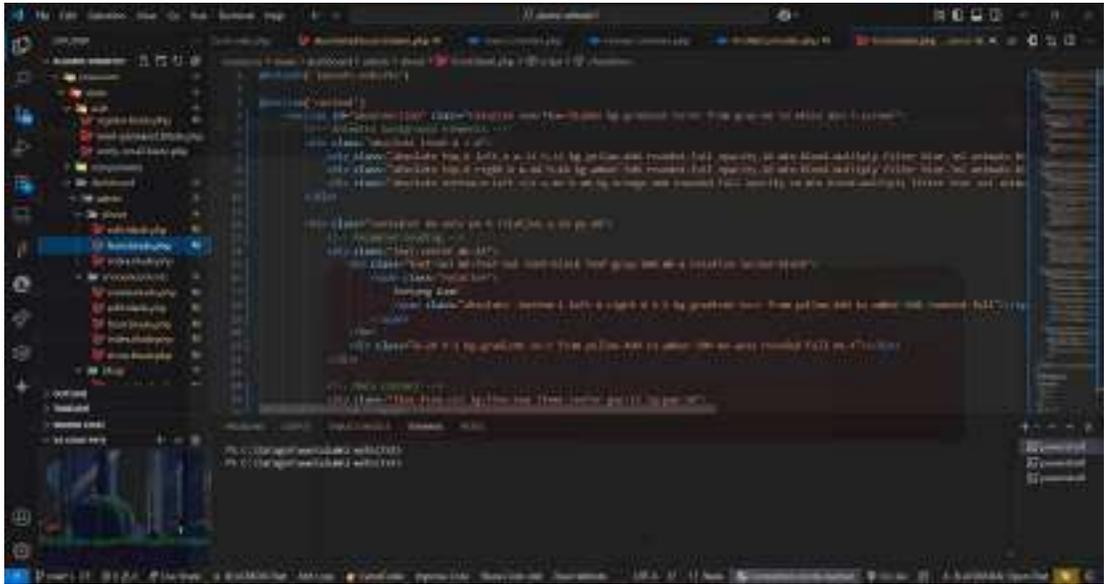
Terdapat bagian "Tentang Kami" yang memberikan gambaran singkat, diikuti dengan "Visi dan Misi" yang menjelaskan tujuan utama dari komunitas tersebut. Halaman

ini juga secara dinamis menampilkan profil beberapa alumni, acara mendatang, artikel blog terbaru, serta pengumuman penting, yang datanya diambil dari backend (misalnya variabel \$alumniUsers, \$events, \$blogs). Ini menunjukkan bahwa halaman tidak hanya statis, tetapi terintegrasi dengan sistem manajemen konten. Terakhir, halaman ini diperkaya dengan berbagai elemen visual dan interaktif untuk meningkatkan pengalaman pengguna (UX). Penggunaan galeri foto dengan Swiper.js, testimoni alumni, serta animasi halus saat scroll dan hover membuat halaman terasa hidup dan profesional. Pilihan desain dengan tema warna kuning dan amber yang konsisten memberikan identitas visual yang kuat. Di beberapa bagian, terdapat ajakan (call-to-action) yang jelas untuk mendorong pengguna agar bergabung dengan komunitas, yang menjadi tujuan akhir dari halaman beranda ini.

2) Halaman tentang kami di bagian landing page untuk guest



Gambar 3. 78 Halaman tentang kami untuk guest



Gambar 3. 79 Potongan kode untuk halaman tentang kami di bagian guest

Penjelasan :

Halaman ini merupakan halaman "Tentang Kami" (About Us) yang dirancang dengan sangat dinamis dan modern untuk sebuah portal alumni. Secara fungsional, halaman ini bertujuan untuk menyajikan informasi inti mengenai komunitas alumni, dimulai dari perkenalan umum yang menampilkan gambar dan deskripsi. Data utama seperti judul ($\$about \rightarrow title$) dan konten ($\$about \rightarrow content$) diambil secara dinamis dari backend Laravel, menunjukkan bahwa halaman ini terintegrasi dengan sebuah sistem manajemen konten (CMS). Jika data tidak tersedia, sistem secara cerdas akan menampilkan konten default sehingga halaman tidak tampak kosong.

Dari sisi desain dan teknologi, halaman ini memanfaatkan Tailwind CSS untuk styling yang responsif dan modern, serta diperkaya dengan animasi CSS kustom (@keyframes) untuk memberikan nuansa visual yang hidup. Elemen seperti blob dan partikel yang bergerak perlahan di latar belakang menciptakan kedalaman visual,

sementara efek transisi pada gambar dan kartu informasi memberikan kesan interaktif saat pengguna mengarahkan kursor. Penggunaan gradient pada teks dan elemen lainnya, dengan palet warna dominan kuning, amber, dan oranye, berhasil menciptakan identitas visual yang hangat dan energik.

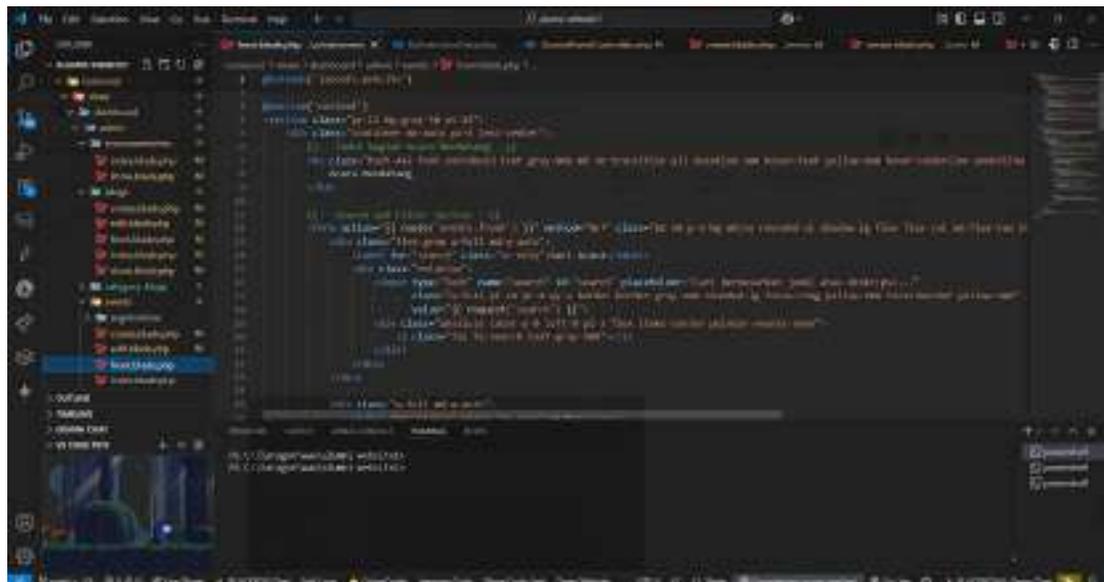
Aspek interaktivitas menjadi salah satu keunggulan utama halaman ini, yang dikelola melalui JavaScript (Vanilla JS). Fitur utamanya adalah tombol "Lebih banyak.." yang menerapkan pola progressive disclosure, di mana konten tambahan tidak dimuat sekaligus. Saat tombol diklik pertama kali, sebuah loading spinner akan muncul untuk menyimulasikan pengambilan data, lalu menampilkan tiga kartu informasi (Visi, Misi, dan Sejarah). Setelah itu, tombol akan berubah teksnya, dan jika diklik lagi, akan memuat set konten berikutnya (Pencapaian dan Komunitas), sebelum akhirnya menghilang setelah semua konten ditampilkan. Secara keseluruhan, halaman ini bukan sekadar halaman informasi statis, melainkan sebuah presentasi yang dipoles dengan baik yang menunjukkan kualitas dan profesionalisme.

Kode ini juga mencakup praktik pengembangan yang baik, seperti penanganan error pada gambar yang gagal dimuat dan penggunaan animasi bertahap (staggered animation) untuk memunculkan setiap paragraf teks secara elegan. Kombinasi antara konten dinamis dari Laravel, styling canggih dari Tailwind CSS, dan fungsionalitas interaktif dari JavaScript menghasilkan sebuah halaman "Tentang Kami" yang menarik, informatif, dan memberikan pengalaman pengguna yang unggul.

3) Halaman acara di bagian landing page untuk guest



Gambar 3. 80 Halaman daftar acara bagian guest



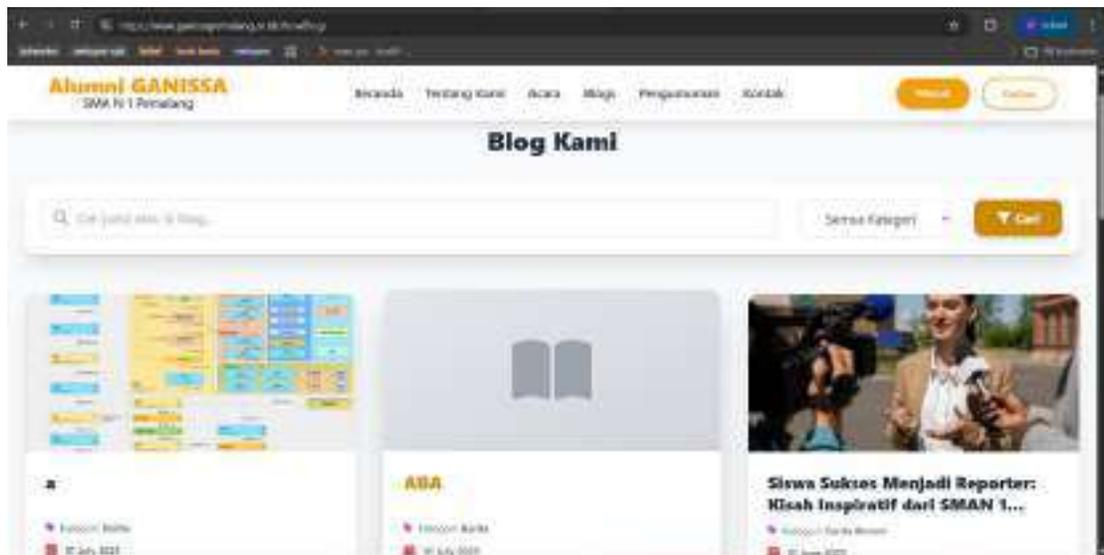
Gambar 3. 81 Potongan kode halaman daftar acara di bagian guest

Penjelasan :

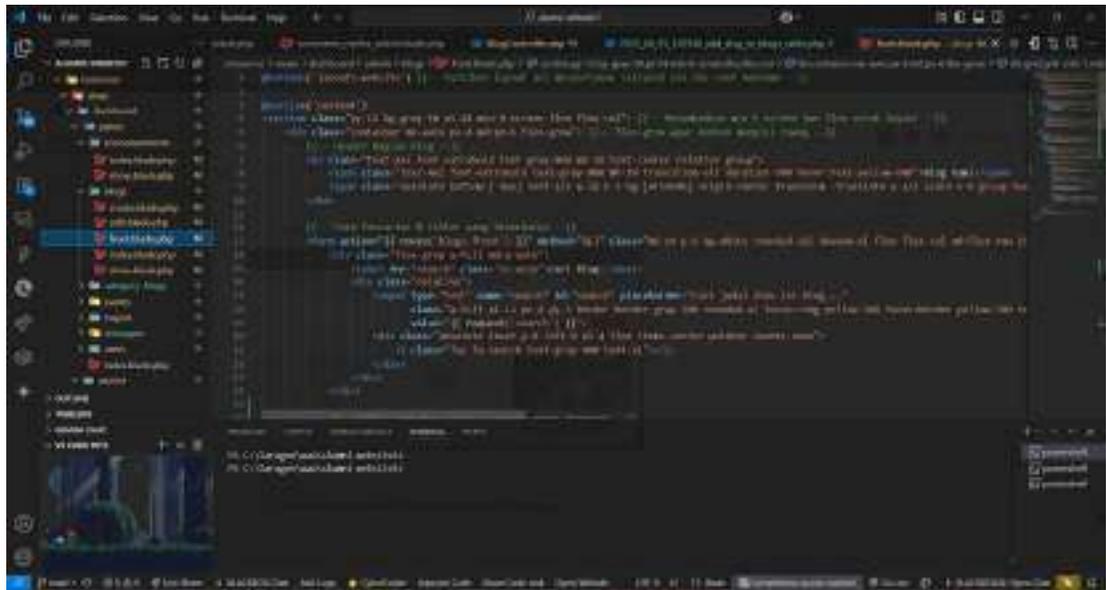
Halaman ini adalah halaman daftar acara (events) yang dirancang untuk ditampilkan kepada publik. Fungsi utamanya adalah menyajikan daftar acara yang akan datang maupun yang telah selesai dalam format kartu (card) yang responsif dan menarik secara visual. Di bagian atas, terdapat fitur pencarian dan penyaringan (filter) yang memungkinkan pengguna mencari acara berdasarkan kata kunci pada judul atau deskripsi, serta memfilter berdasarkan status acara, yaitu "Mendatang" atau "Selesai".

Halaman ini juga secara dinamis akan menampilkan pesan khusus jika tidak ada acara yang ditemukan sesuai kriteria pencarian, dan menyertakan navigasi halaman (pagination) di bagian bawah jika jumlah acara melebihi kapasitas satu halaman. Secara teknis, halaman ini dibangun menggunakan Laravel Blade sebagai templating engine dan Tailwind CSS untuk styling. Konten acara, seperti judul, gambar, deskripsi, tanggal, dan lokasi, diambil dari variabel \$events yang dikirim dari backend (controller). Penggunaan helper seperti asset(), route(), Str::limit(), dan pustaka Carbon menunjukkan integrasi yang kuat antara logika backend untuk memformat data dan presentasi frontend. Setiap kartu acara memiliki tombol "Lihat Detail" yang mengarahkan pengguna ke halaman detail untuk informasi lebih lanjut, menciptakan alur navigasi yang jelas dan fungsional bagi pengunjung.

4) Halaman blog di bagian landing page untuk guest



Gambar 3. 82 Halaman blog bagian guest



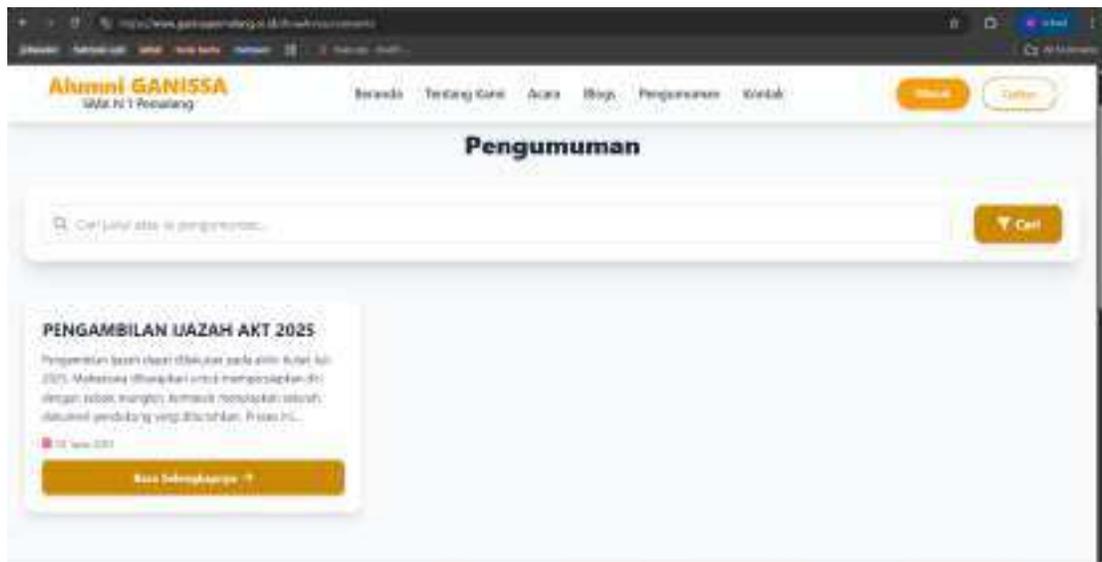
Gambar 3. 83 Potongan kode halaman blog bagian guest

Penjelasan :

Halaman ini merupakan halaman "Blog" yang dirancang untuk menampilkan kumpulan artikel atau tulisan kepada pengunjung website. Secara fungsional, halaman ini menyediakan fitur pencarian dan filter yang memungkinkan pengguna mencari artikel berdasarkan kata kunci dan menyaringnya berdasarkan kategori. Jika tidak ada artikel yang sesuai dengan kriteria pencarian, sebuah pesan informatif akan ditampilkan. Setiap artikel disajikan dalam format kartu (card) yang modern dan responsif, berisi gambar, judul, kategori, tanggal terbit, dan kutipan singkat dari isi artikel. Di bagian bawah setiap kartu, terdapat tombol "Baca Selengkapnya" yang akan mengarahkan pengguna ke halaman detail artikel tersebut. Untuk menangani jumlah artikel yang banyak, halaman ini juga dilengkapi dengan sistem navigasi halaman (pagination).

Secara teknis, halaman ini dibangun menggunakan Laravel Blade untuk struktur dan logika tampilan, serta Tailwind CSS untuk desain visualnya. Data artikel (`$blogs`) dan kategori (`$categories`) dikirim dari backend (controller), yang kemudian ditampilkan secara dinamis menggunakan perulangan `@foreach`. Halaman ini memanfaatkan helper dan pustaka bawaan Laravel seperti `request()` untuk menjaga input filter, `asset()` untuk memuat gambar, `Str::limit()` untuk memotong teks deskripsi, dan pustaka Carbon untuk memformat tanggal. Desainnya yang interaktif dengan efek hover dan transisi bertujuan untuk memberikan pengalaman pengguna yang menarik dan profesional.

5) Halaman pengumuman di bagian landing page untuk guest



Gambar 3. 84 Halaman pengumuman bagian guest



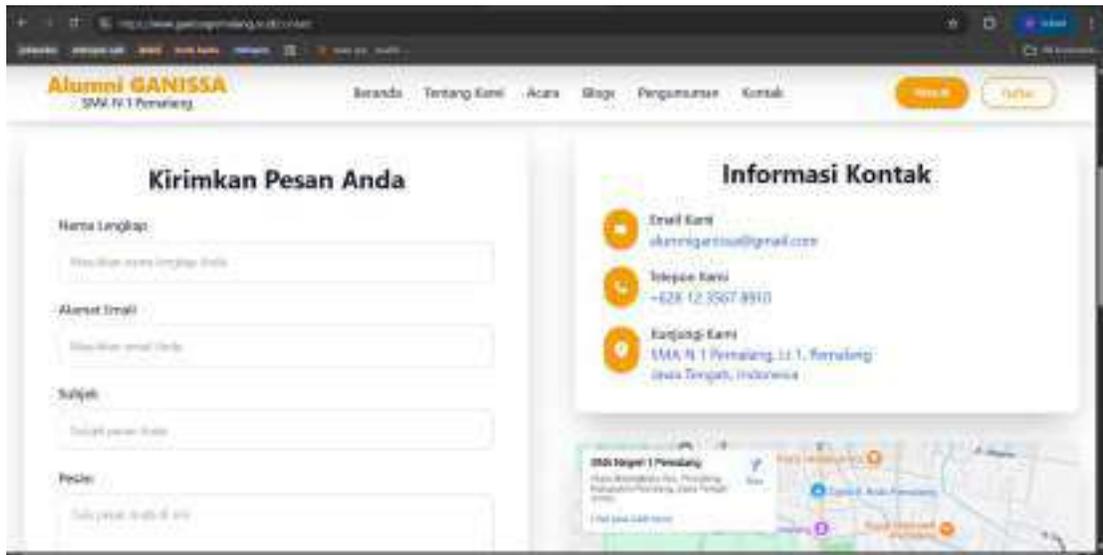
Gambar 3. 85 Potongan kode halaman pengumuman bagian guest

Penjelasan :

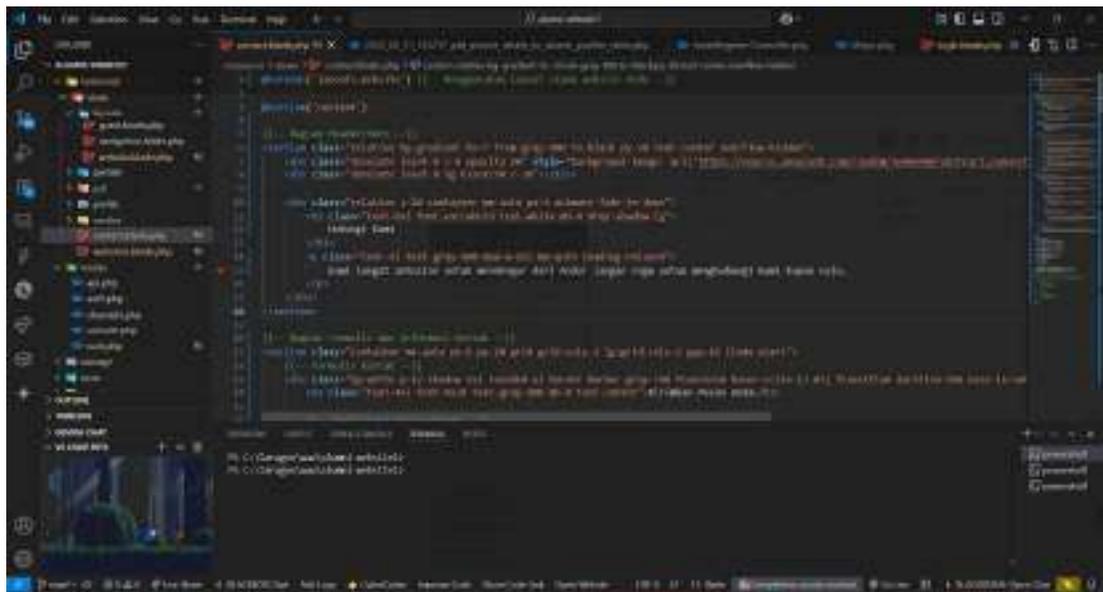
Halaman ini adalah halaman "Pengumuman" yang dirancang untuk diakses oleh publik, berfungsi sebagai pusat informasi di mana pengguna dapat melihat daftar semua pengumuman yang telah diterbitkan. Fitur utamanya adalah form pencarian di bagian atas yang memungkinkan pengunjung untuk dengan mudah mencari pengumuman spesifik berdasarkan kata kunci pada judul atau isinya. Halaman ini secara dinamis menampilkan daftar pengumuman dalam format kartu (card) yang responsif, di mana setiap kartu menyajikan judul, ringkasan singkat dari deskripsi, serta tanggal publikasi. Jika tidak ada pengumuman yang ditemukan, baik secara umum maupun dari hasil pencarian, sistem akan menampilkan pesan yang informatif. Untuk navigasi lebih lanjut, setiap kartu memiliki tombol "Baca Selengkapnya" yang mengarahkan ke halaman detail pengumuman, dan di bagian bawah terdapat navigasi halaman (pagination) untuk

menangani jumlah pengumuman yang banyak, semuanya dibangun menggunakan template Laravel Blade dan di-styling dengan Tailwind CSS.

6) Halaman kontak di bagian landing page untuk guest



Gambar 3. 86 Halaman kontak bagian guest



Gambar 3. 87 Potongan kode halaman kontak bagian guest

Penjelasan :

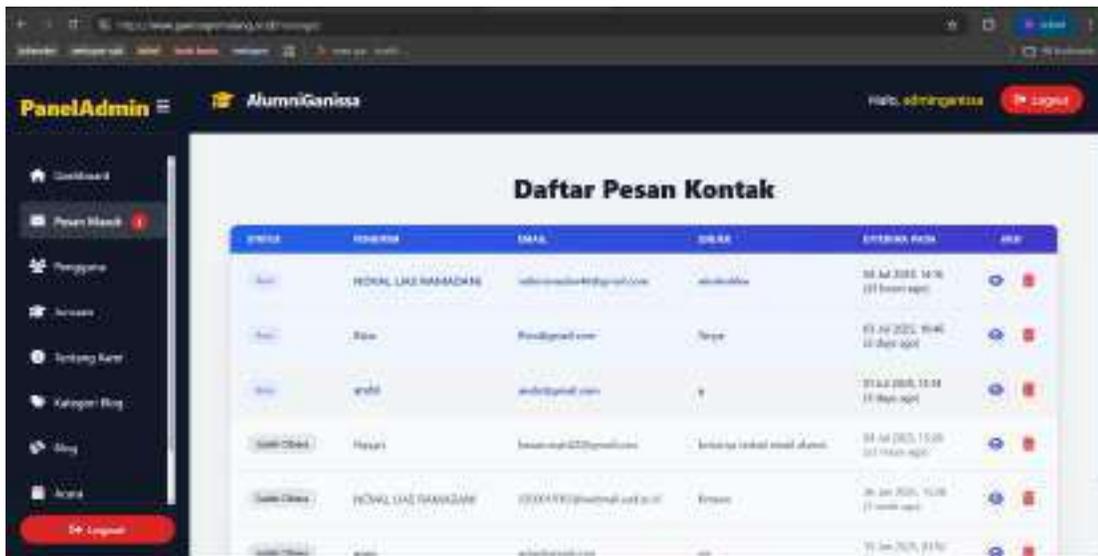
Halaman ini adalah halaman "Hubungi Kami" yang dirancang secara profesional dan interaktif untuk memungkinkan pengunjung berkomunikasi dengan pemilik situs. Secara fungsional, halaman ini terbagi menjadi dua kolom utama: di satu sisi terdapat formulir kontak yang memungkinkan pengguna mengirimkan pesan dengan mengisi nama, email, subjek, dan isi pesan, yang kemudian akan diproses oleh backend Laravel melalui route `contact.submit`. Di sisi lain, disajikan informasi kontak yang jelas seperti alamat email, nomor telepon, dan lokasi fisik yang dilengkapi dengan tautan interaktif serta peta Google Maps yang disematkan untuk memudahkan navigasi. Halaman ini juga sangat ramah pengguna karena secara dinamis menampilkan notifikasi keberhasilan, kegagalan, atau kesalahan validasi setelah formulir dikirim, serta dipercantik dengan animasi modern menggunakan CSS dan styling dari Tailwind CSS untuk pengalaman visual yang menarik.

B. Target User : Admin

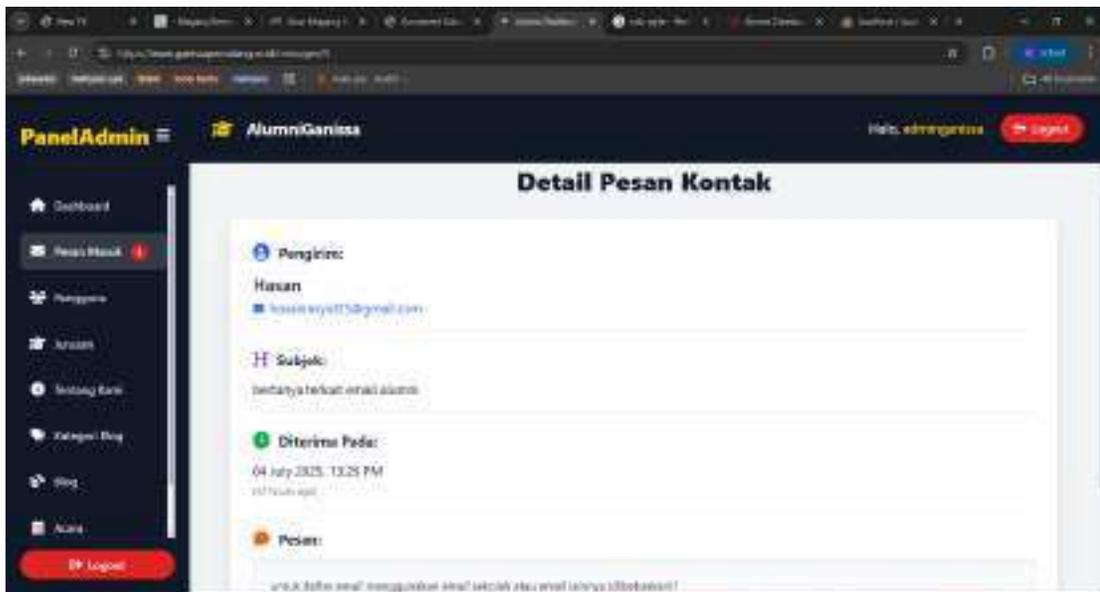
1) Halaman Dashboard Admin

Halaman ini merupakan sebuah dasbor admin yang dirancang secara visual dan interaktif untuk sebuah aplikasi web berbasis Laravel. Halaman ini menyajikan ringkasan data penting dalam bentuk empat kartu statistik berwarna yang menampilkan jumlah total blog, acara, pengumuman, dan pengguna terdaftar. Setiap kartu juga dilengkapi tautan untuk langsung mengelola bagian tersebut. Di bawahnya, terdapat bagian "Aktivitas Terbaru" yang memberikan informasi sekilas tentang konten terbaru yang dipublikasikan, seperti blog, acara, dan pengumuman terakhir. Terakhir, halaman ini menyediakan area "Tindakan Cepat" dengan tombol-tombol yang mudah diakses untuk membuat entri baru, memungkinkan admin untuk menambahkan konten dengan efisien langsung dari dasbor utama. Seluruh tata letak dibuat menggunakan Tailwind CSS untuk menciptakan tampilan yang modern, responsif, dan kaya akan efek visual seperti animasi saat kursor diarahkan ke elemen tertentu.

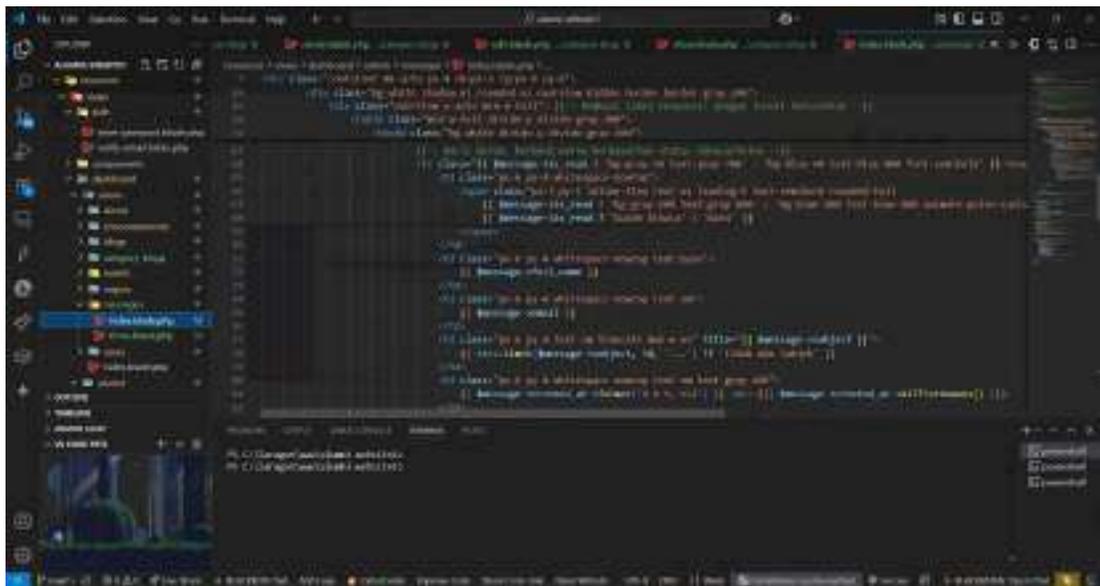
2) Halaman Pesan Masuk



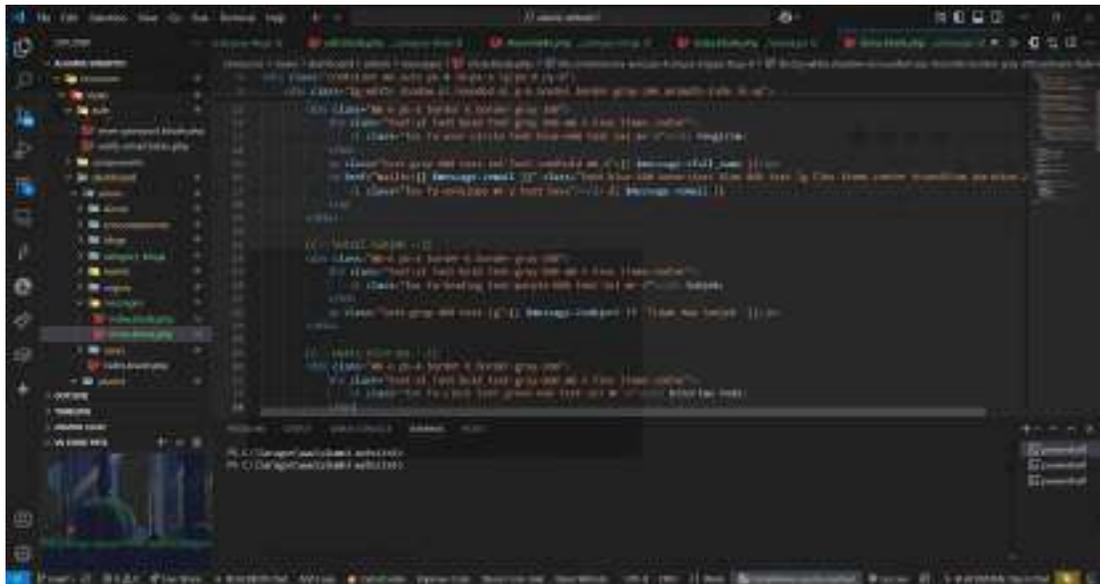
Gambar 3. 90 Halaman pesan masuk bagian admin



Gambar 3. 91 halaman detail pesan pada bagian admin



Gambar 3. 92 Potongan kode untuk halaman pesan masuk di bagian admin



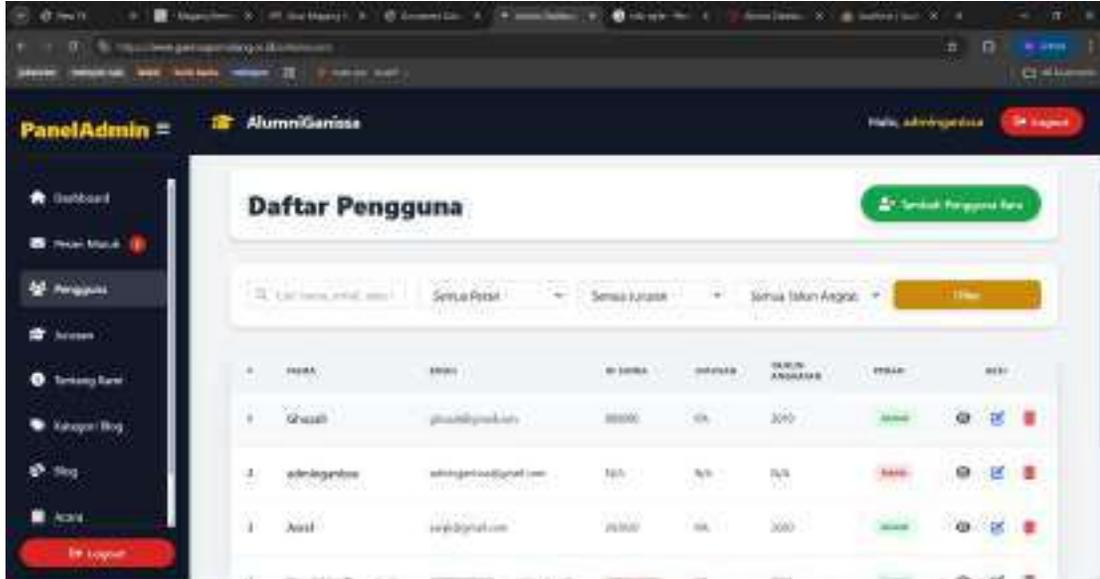
Gambar 3. 93 Potongan kode untuk halaman detail pesan di bagian admin

Penjelasan :

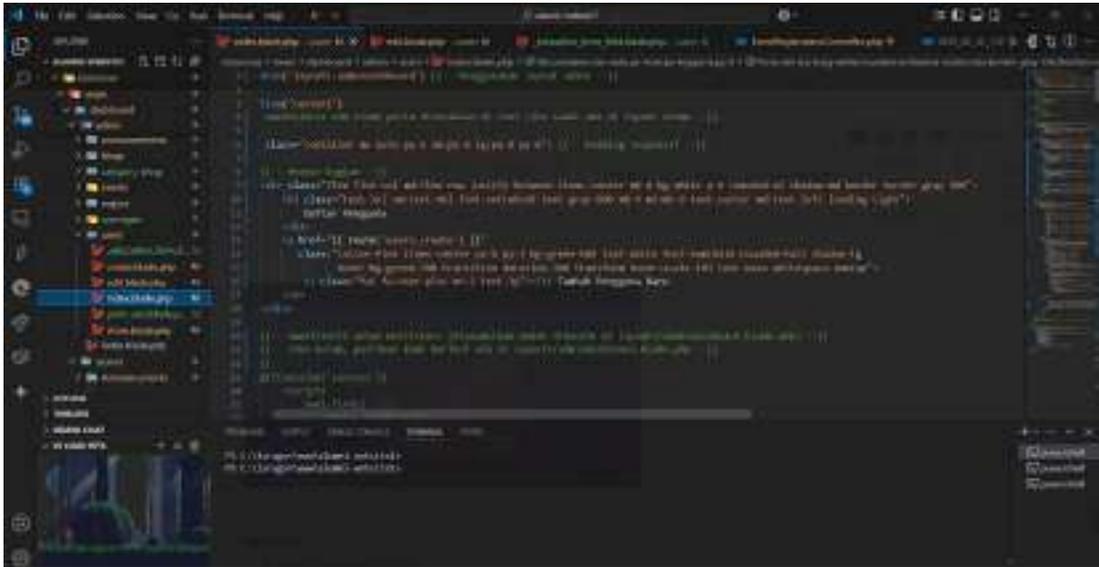
Halaman pertama adalah daftar pesan kontak yang berfungsi sebagai kotak masuk utama bagi admin. Halaman ini dirancang untuk menampilkan semua pesan yang diterima dari formulir kontak dalam format tabel yang rapi dan responsif. Fitur utamanya adalah kemampuan untuk membedakan secara visual antara pesan yang sudah dibaca dan yang belum. Pesan baru ditandai dengan latar belakang yang berbeda dan label "Baru" yang beranimasi untuk menarik perhatian. Setiap baris dalam tabel menyajikan informasi ringkas seperti nama pengirim, email, subjek, dan waktu pesan diterima, serta menyediakan dua tombol aksi: satu untuk melihat detail lengkap pesan dan satu lagi untuk menghapusnya dengan konfirmasi melalui dialog peringatan untuk mencegah kesalahan. Jika tidak ada pesan sama sekali, halaman akan menampilkan ilustrasi dan teks yang informatif.

Halaman kedua adalah halaman detail pesan, yang ditampilkan ketika admin mengklik tombol "lihat" pada salah satu pesan di halaman daftar. Halaman ini fokus untuk menyajikan seluruh informasi dari satu pesan secara jelas dan terstruktur dalam sebuah kartu. Informasi dibagi ke dalam beberapa bagian yang mudah dibaca, seperti detail pengirim (lengkap dengan tautan email yang bisa diklik), subjek, waktu diterima, dan isi pesan lengkap yang ditampilkan dalam area khusus agar mudah dibaca. Di bagian bawah, terdapat dua tombol aksi utama: satu untuk kembali ke halaman daftar pesan dan satu lagi untuk menghapus pesan tersebut secara permanen, yang juga dilindungi oleh fungsi konfirmasi yang sama. Desainnya yang bersih dan terfokus memastikan admin dapat membaca dan mengelola pesan secara efisien tanpa gangguan.

3) Halaman Pengguna Terdaftar



Gambar 3. 94 Halaman Manajemen Pengguna di bagian admin



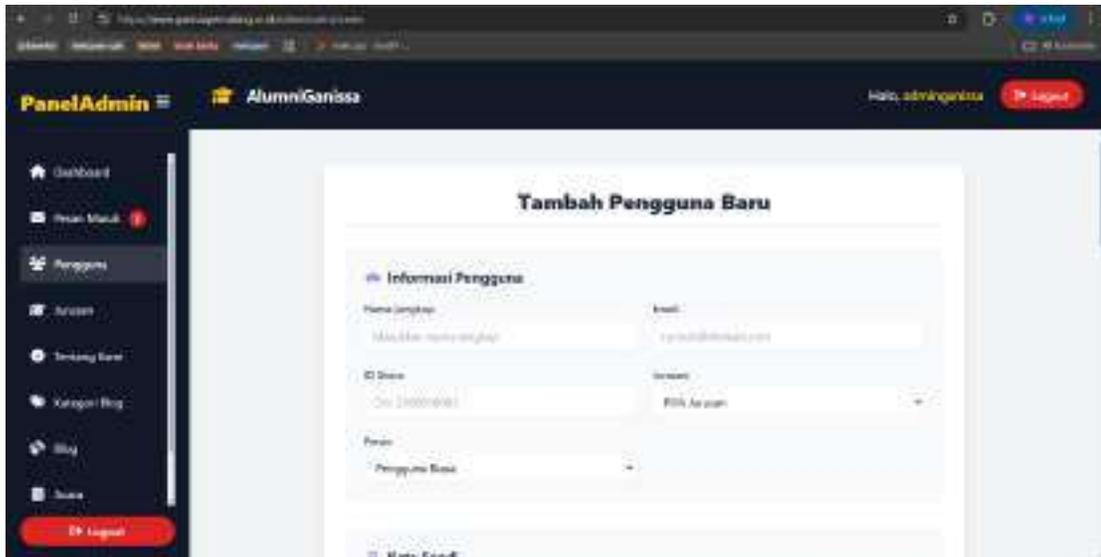
Gambar 3. 95 Potongan kode untuk halaman manajemen pengguna di bagian admin

Penjelasan :

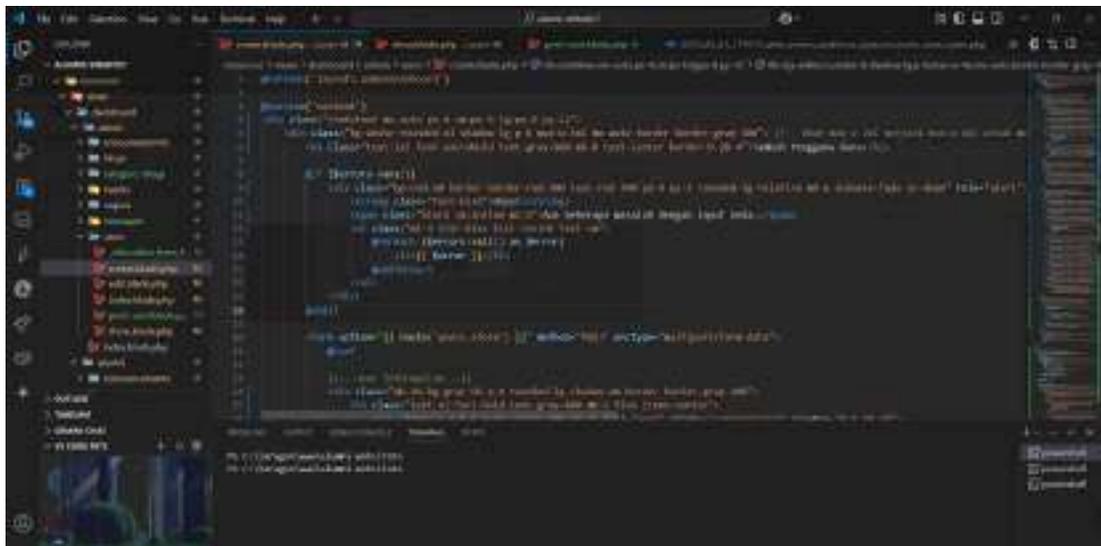
Halaman ini adalah antarmuka manajemen pengguna yang komprehensif untuk panel admin, dirancang dengan tata letak yang bersih dan fungsional menggunakan Tailwind CSS. Di bagian atas, terdapat judul yang jelas dan tombol untuk menambah pengguna baru. Fitur utamanya adalah panel pencarian dan filter yang canggih, memungkinkan admin untuk menyaring daftar pengguna berdasarkan kata kunci (nama, email, atau ID siswa), peran (seperti admin, alumni, atau pengguna biasa), jurusan, serta tahun angkatan. Hasilnya ditampilkan dalam sebuah tabel yang terstruktur rapi, menyajikan detail penting setiap pengguna seperti nama, email, jurusan, dan peran yang ditandai dengan label berwarna. Setiap baris pengguna dilengkapi dengan tombol aksi untuk melihat detail, mengedit, atau menghapus data, di mana fungsi hapus diamankan dengan dialog konfirmasi untuk mencegah penghapusan yang tidak disengaja. Halaman ini juga menangani kasus ketika tidak ada

Penjelasan :

Halaman ini adalah halaman detail pengguna yang sangat terperinci dalam sebuah dasbor admin, dirancang untuk menyajikan profil lengkap seorang pengguna dalam satu kartu besar yang terorganisir. Bagian atas kartu menampilkan ringkasan visual yang kuat, termasuk foto profil pengguna, nama, email, serta label status penting seperti tahun angkatan dan pekerjaan saat ini. Di bawahnya, informasi dibagi secara logis menjadi beberapa bagian: informasi akun dasar (seperti ID siswa dan jurusan), profil alumni (kontak dan kode alumni), informasi karir, dan detail tambahan termasuk bio dan tautan media sosial yang dapat diklik. Fitur yang paling menonjol adalah bagian riwayat pendidikan, yang disajikan dalam format linimasa (timeline) vertikal yang menarik, mengurutkan perjalanan akademis pengguna secara kronologis. Terakhir, di bagian bawah, admin diberikan tombol aksi yang jelas untuk langsung mengedit profil pengguna atau mencetak kartu alumni, membuat pengelolaan data pengguna menjadi sangat efisien dan informatif.



Gambar 3. 98 Halaman tambah pengguna baru bagian admin

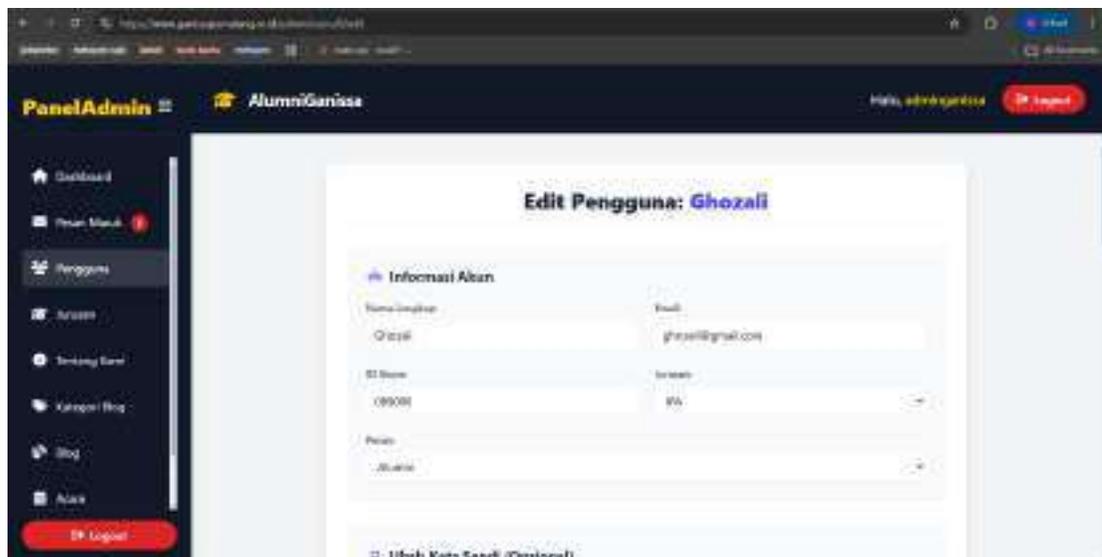


Gambar 3. 99 Potongan kode untuk halaman tambah pengguna baru di bagian admin

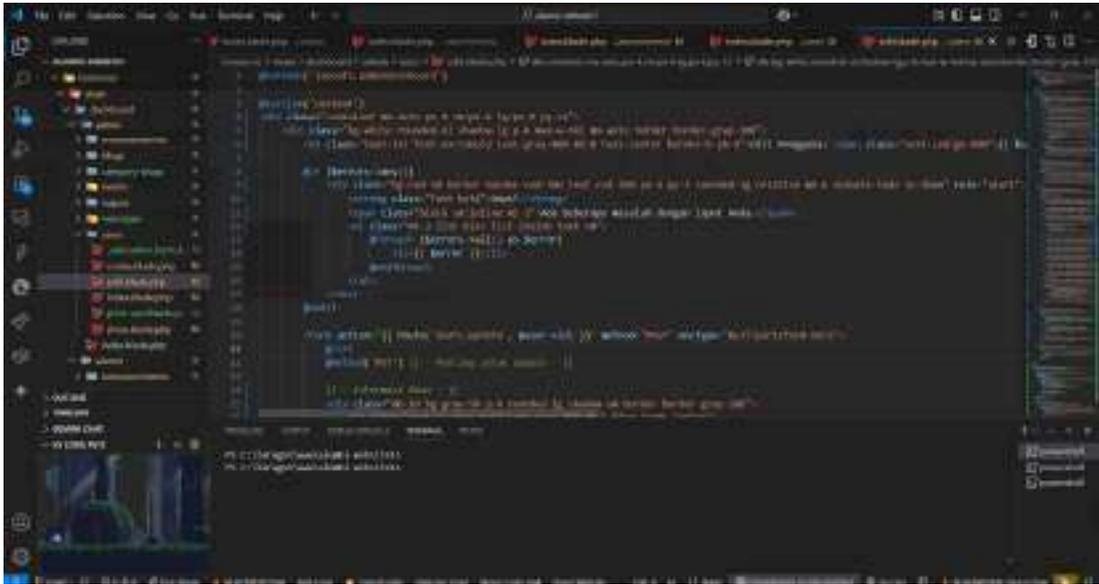
Penjelasan :

Halaman ini merupakan formulir komprehensif untuk menambah pengguna baru dalam sistem panel admin, yang dirancang dengan tata letak yang bersih dan terstruktur. Formulir ini dibagi menjadi beberapa bagian utama yang logis: informasi pengguna yang wajib diisi

(seperti nama, email, ID siswa, jurusan, dan peran), bagian untuk mengatur kata sandi beserta konfirmasinya, dan sebuah profil alumni yang bersifat opsional untuk data yang lebih rinci seperti informasi kontak, riwayat pekerjaan, tautan media sosial, dan foto profil. Fitur paling canggih dari halaman ini adalah bagian riwayat pendidikan yang dinamis, di mana admin dapat secara fleksibel menambah atau mengurangi isian riwayat pendidikan menggunakan JavaScript, memungkinkan pencatatan data akademis yang lengkap dan bervariasi. Halaman ini juga dilengkapi dengan sistem penanganan kesalahan (error handling) yang jelas, menampilkan baik ringkasan error di bagian atas maupun pesan error di bawah setiap kolom isian yang bermasalah, serta tombol aksi "Simpan" dan "Batal" untuk menyelesaikan proses.



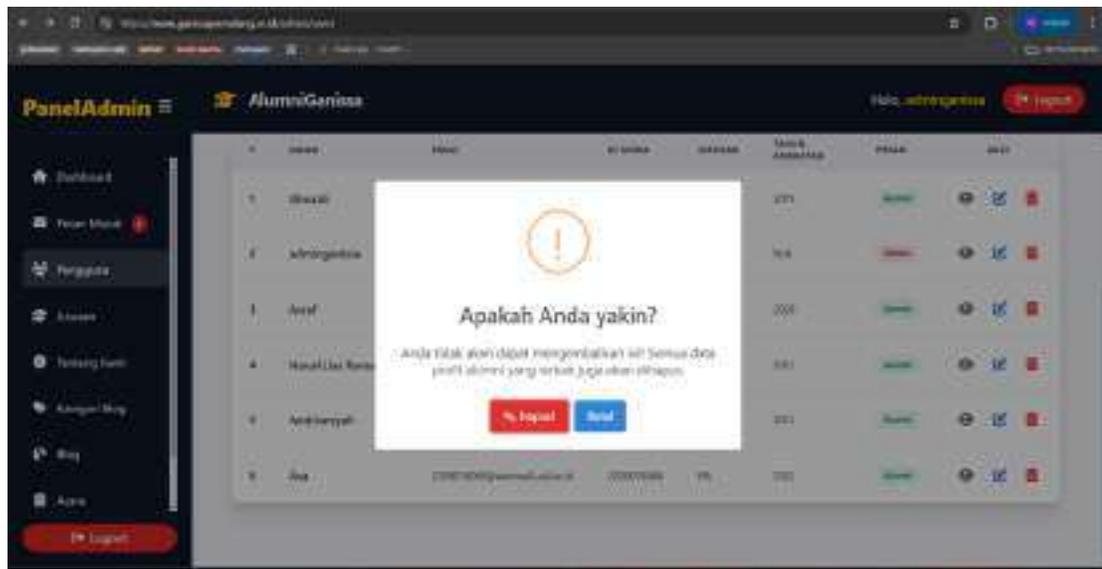
Gambar 3. 100 Halaman edit pengguna di bagian admin



Gambar 3. 101 Potongan kode untuk halaman edit pengguna di bagian admin

Penjelasan :

Halaman ini adalah sebuah formulir edit pengguna yang komprehensif, dirancang untuk memungkinkan administrator memperbarui semua informasi terkait pengguna yang sudah ada. Formulir ini secara cerdas memuat semua data pengguna saat ini ke dalam setiap kolom isian, termasuk informasi akun dasar, profil alumni yang mendetail (seperti data karir, kontak, dan tautan sosial), serta menampilkan pratinjau foto profil yang sedang digunakan. Halaman ini juga menyediakan bagian khusus untuk mengubah kata sandi yang bersifat opsional, sehingga kata sandi lama tetap aman jika tidak ada perubahan. Fitur utamanya adalah bagian riwayat pendidikan yang dinamis, yang tidak hanya menampilkan data pendidikan yang sudah ada tetapi juga memungkinkan admin untuk menambah, mengubah, atau menghapus riwayat pendidikan secara langsung di halaman yang sama menggunakan JavaScript. Dengan tombol "Simpan Perubahan" dan "Batal", halaman ini menyediakan antarmuka yang kuat dan fleksibel untuk mengelola data pengguna secara menyeluruh.



Gambar 3. 102 Tampilan ketika menghapus salah satu user pada bagian admin

Penjelasan :

Tangkapan layar ini secara spesifik menyorot dialog konfirmasi penghapusan, yang muncul setelah admin mengklik ikon tempat sampah. Dialog ini berfungsi sebagai lapisan keamanan, memperingatkan admin bahwa tindakan ini tidak dapat dibatalkan dan akan menghapus semua data terkait, lalu memberikan pilihan untuk melanjutkan dengan menekan tombol "Ya, hapus!" atau membatalkan dengan tombol "Batal".



Gambar 3. 103 Halaman cetak kartu alumni pada bagian admin



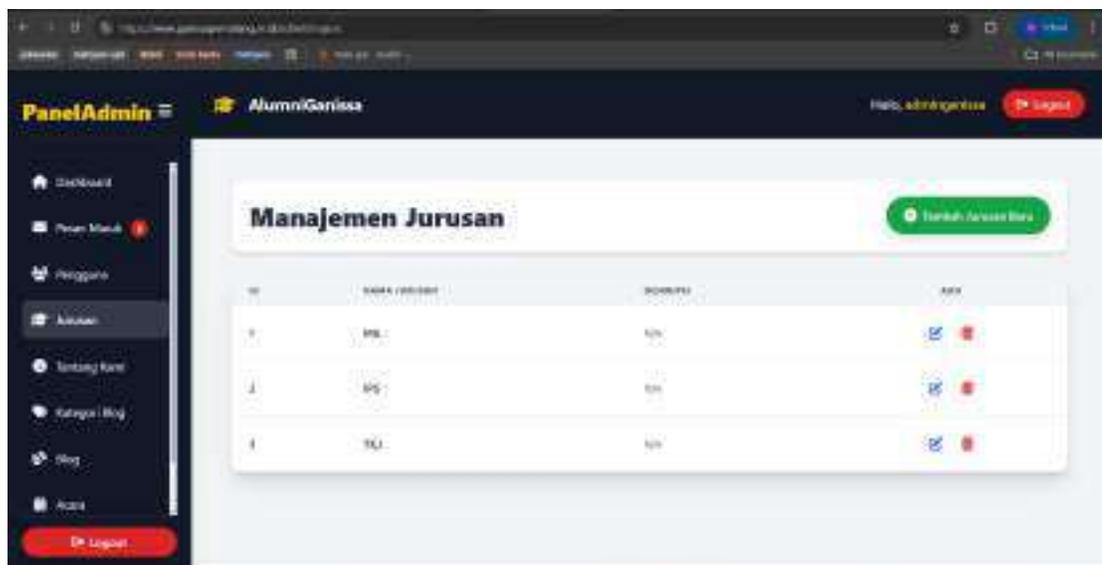
Gambar 3. 104 Potongan kode untuk halaman cetak kartu alumni pada bagian admin

Penjelasan :

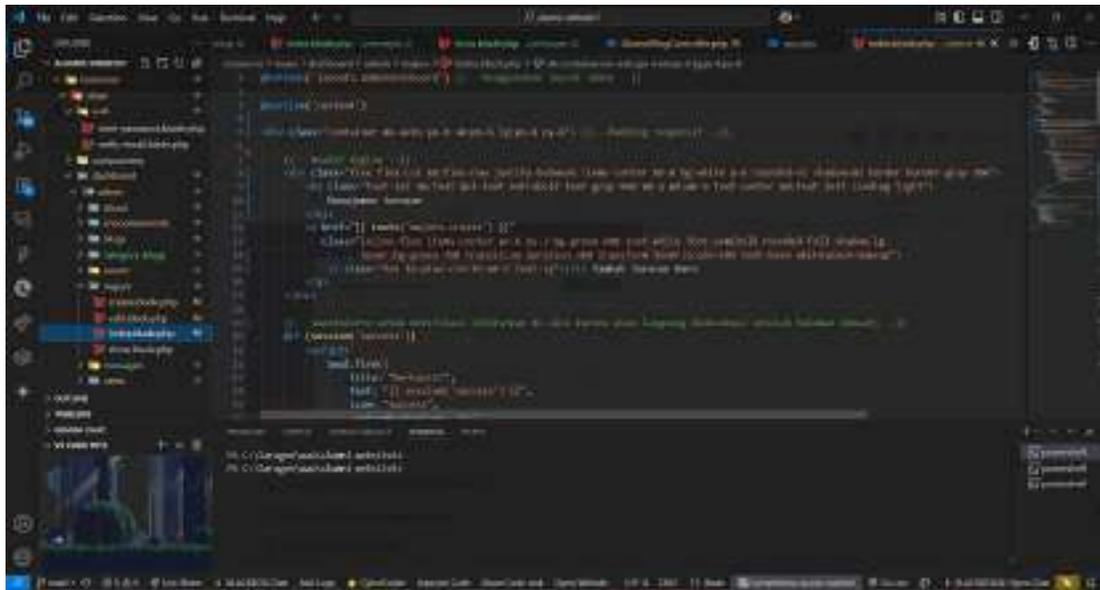
Halaman ini berfungsi sebagai kartu alumni digital dan halaman cetak yang canggih dan interaktif. Secara visual, halaman ini menampilkan sebuah kartu alumni dengan desain premium yang kaya akan animasi, seperti latar belakang gradien yang bergerak, partikel mengambang, dan efek interaktif saat cursor diarahkan ke kartu. Kartu ini memuat informasi penting alumni, termasuk foto, nama, jurusan, tahun angkatan, dan kode alumni unik yang

secara dinamis diubah menjadi gambar barcode menggunakan JavaScript. Fungsi utamanya adalah untuk dicetak saat tombol "Cetak Kartu" diklik, CSS khusus untuk media cetak akan aktif, secara otomatis menyembunyikan semua tombol dan animasi, lalu mengubah ukuran kartu menjadi seukuran kartu kredit standar (85.6mm x 53.98mm) agar siap untuk dicetak menjadi kartu fisik.

4) Halaman Manajemen Jurusan



Gambar 3. 105 Halaman manajemen jurusan bagian admin



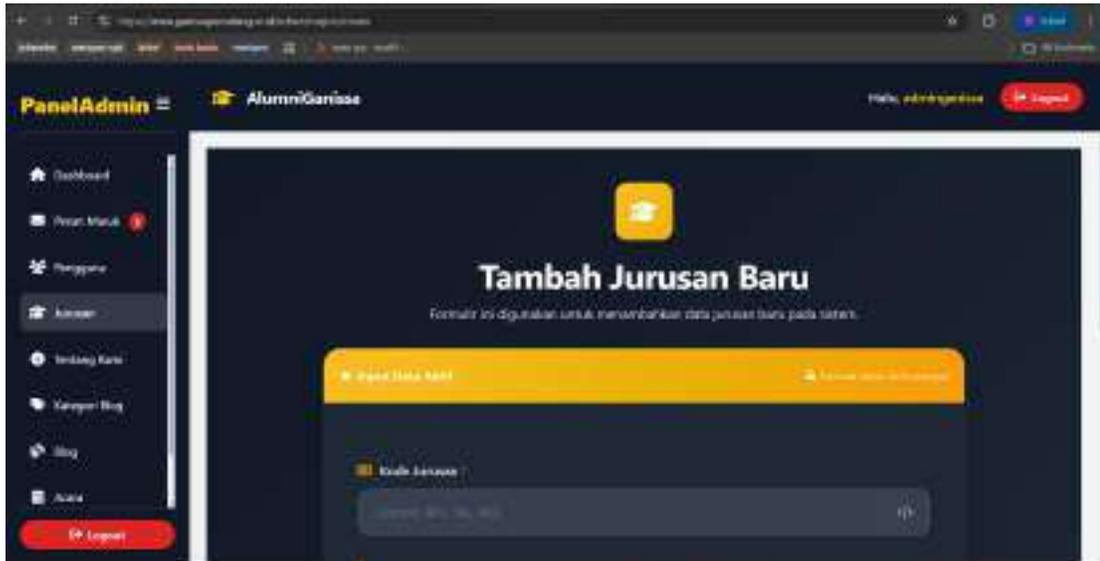
Gambar 3. 106 Potongan kode untuk halaman manajemen jurusan di bagian admin

Penjelasan :

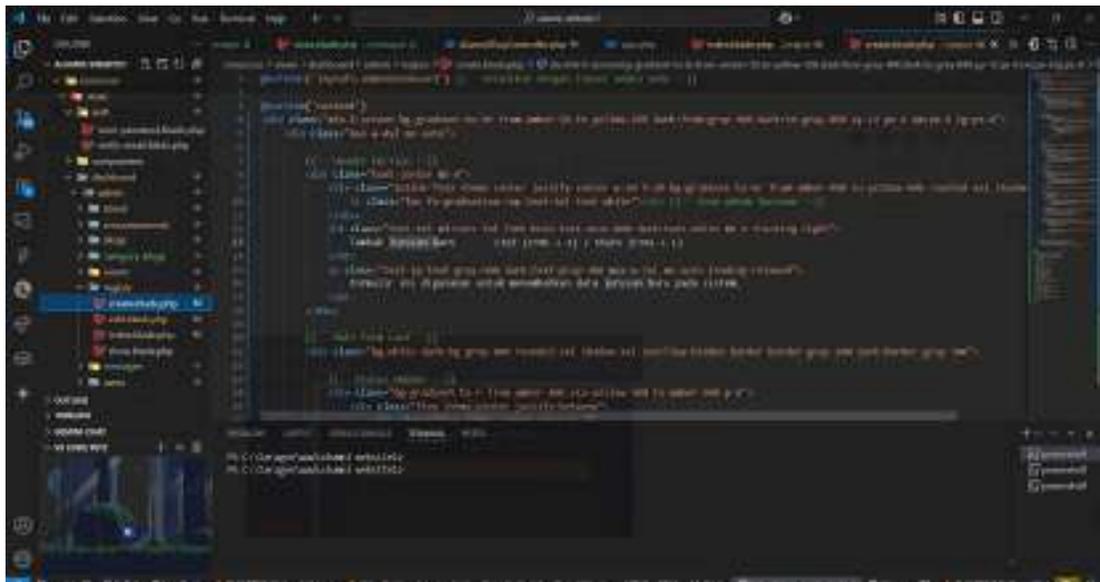
Halaman ini adalah antarmuka Manajemen Jurusan di dalam sebuah panel admin, yang dirancang untuk mengelola data program studi atau jurusan. Tampilannya bersih dan fungsional, dengan judul yang jelas dan tombol "Tambah Jurusan Baru" yang menonjol di bagian atas, memungkinkan admin untuk dengan mudah menambahkan data baru. Halaman ini juga terintegrasi dengan notifikasi SweetAlert2 yang akan menampilkan pesan pop-up elegan untuk memberitahu admin jika suatu operasi (seperti menambah atau menghapus jurusan) berhasil atau gagal.

Inti dari halaman ini adalah sebuah tabel yang menampilkan daftar semua jurusan yang sudah ada. Setiap baris dalam tabel menyajikan informasi penting seperti ID, nama jurusan, dan deskripsi singkatnya. Di ujung kanan setiap baris, terdapat tombol aksi untuk "Edit" dan "Hapus" data. Fungsi hapus dilindungi oleh dialog konfirmasi SweetAlert2 untuk mencegah penghapusan yang tidak disengaja. Jika belum ada jurusan yang terdaftar, halaman

akan menampilkan pesan informatif dengan ilustrasi yang ramah pengguna, lengkap dengan tautan paginasi di bagian bawah untuk navigasi yang mudah jika daftar jurusan sudah sangat banyak.

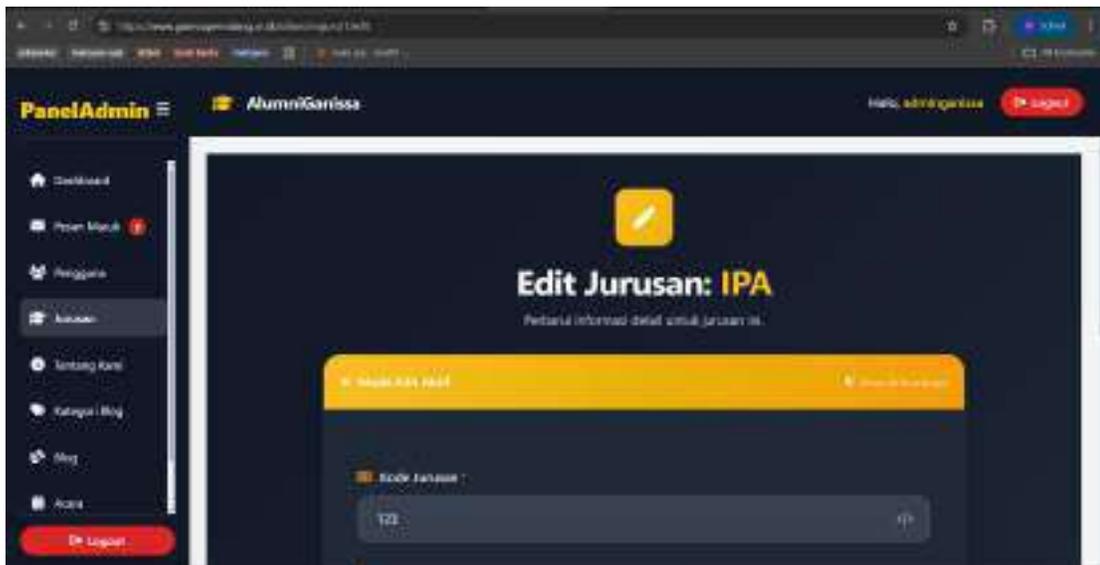


Gambar 3. 107 Halaman tambah jurusan baru di bagian admin

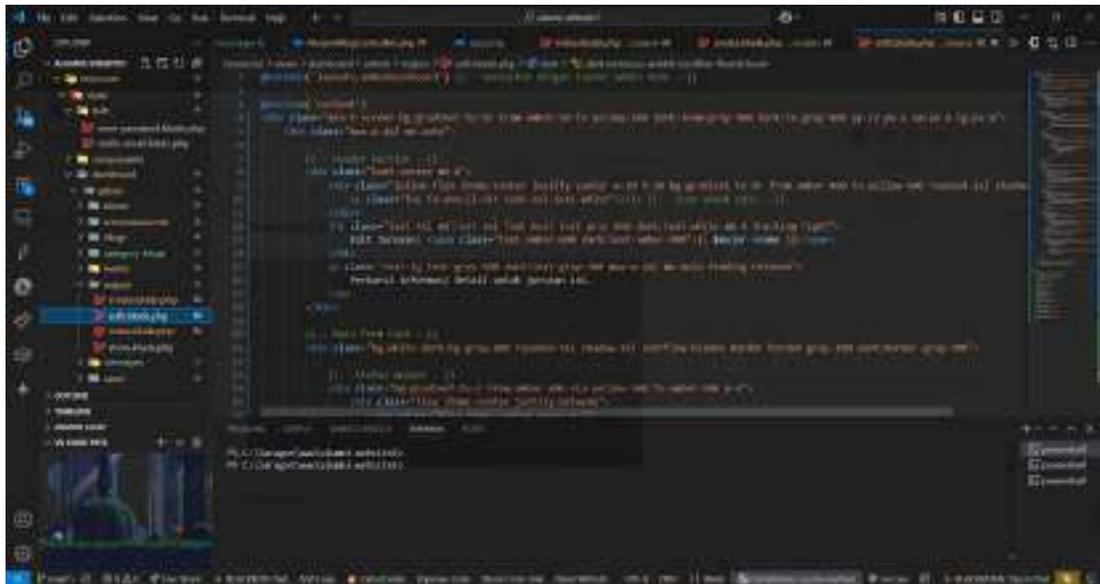


Penjelasan :

Halaman ini merupakan sebuah formulir untuk menambahkan data jurusan baru dengan desain yang sangat modern dan fokus pada pengalaman pengguna. Secara visual, halaman ini menggunakan latar belakang gradien yang lembut, ikon yang informatif, dan elemen kartu yang membulat dengan efek bayangan untuk memberikan tampilan premium, serta telah mendukung mode gelap (dark mode). Formulir itu sendiri dibuat sangat jelas dan sederhana, meminta pengguna untuk memasukkan tiga informasi utama: kode jurusan yang unik, nama lengkap jurusan, dan deskripsi opsional. Halaman ini juga dilengkapi dengan sistem penanganan kesalahan (error) yang canggih, di mana jika terjadi kesalahan validasi, sebuah kotak peringatan yang rapi akan muncul di atas, dan setiap kolom yang salah akan ditandai dengan pesan error di bawahnya, memastikan pengguna dapat dengan mudah memperbaiki input mereka sebelum mengirimkan data.



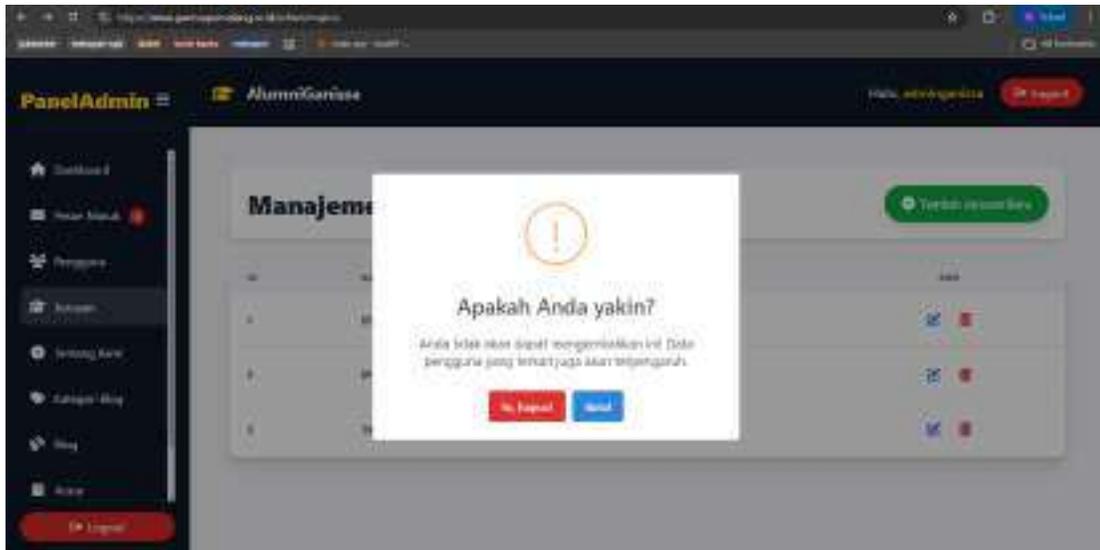
Gambar 3. 109 Halaman edit jurusan di bagian admin



Gambar 3. 110 Potongan kode untuk halaman edit jurusan di bagian admin

Penjelasan :

Halaman ini adalah sebuah formulir untuk mengedit data jurusan yang sudah ada, dirancang dengan antarmuka yang modern dan elegan serta dukungan penuh untuk mode gelap (dark mode). Berbeda dari halaman tambah baru, formulir ini secara otomatis mengisi setiap kolom kode jurusan, nama, dan deskripsi—dengan data yang ada saat ini, memungkinkan admin untuk dengan mudah melihat dan memodifikasi informasi yang diperlukan. Halaman ini sangat fokus pada pengalaman pengguna, menyediakan sistem notifikasi yang jelas untuk pesan keberhasilan (success) atau kesalahan (error) dalam bentuk kotak peringatan yang rapi. Dengan tombol aksi "Perbarui Jurusan" dan "Batal" yang jelas, halaman ini memberikan cara yang efisien dan intuitif untuk mengelola dan memperbarui detail program studi dalam sistem.



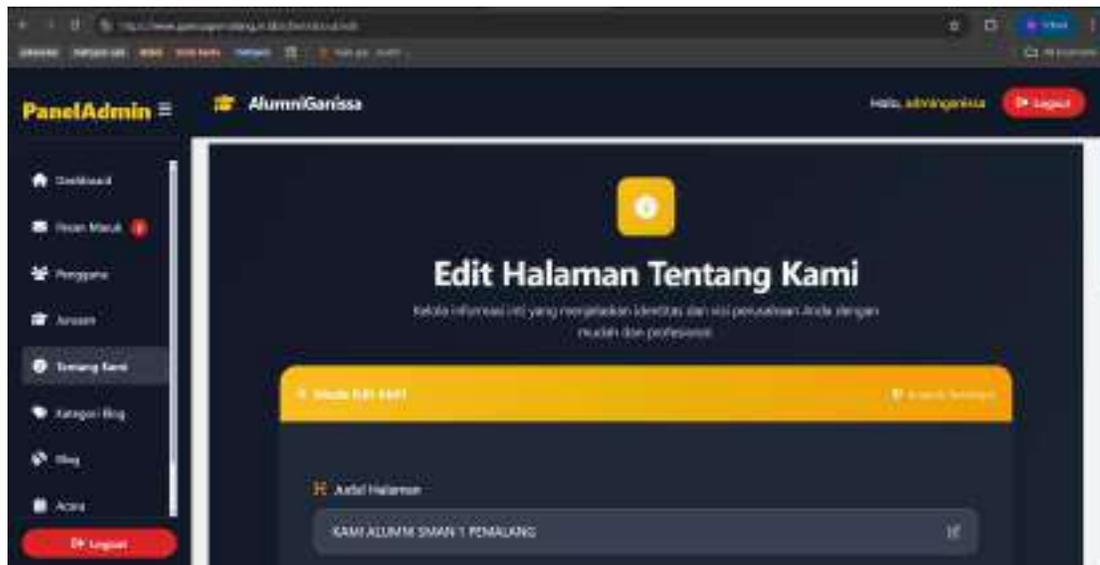
Gambar 3. 111 Tampilan ketika admin menghapus jurusan

Penjelasan :

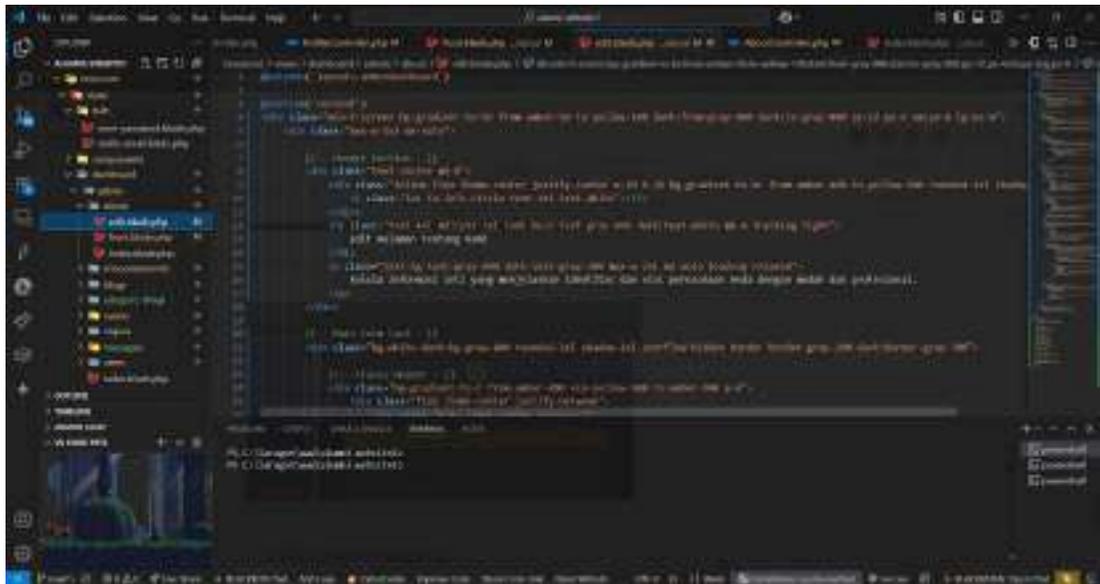
Sebuah dialog konfirmasi muncul di tengah layar untuk mencegah kesalahan. Dialog ini dengan jelas bertanya, "Apakah Anda yakin?", dan memberikan peringatan bahwa tindakan ini tidak bisa dibatalkan dan akan memengaruhi data pengguna yang terkait, sehingga admin harus secara sadar memilih antara melanjutkan dengan tombol "Ya, hapus!" atau membatalkan operasi.

Penjelasan :

Halaman ini berfungsi sebagai dasbor untuk mengelola konten "Tentang Kami" dari sisi admin. Halaman ini dirancang dengan logika kondisional yang cerdas: jika konten sudah ada, halaman akan menampilkannya dalam sebuah kartu yang rapi, lengkap dengan judul, isi teks yang diformat agar mudah dibaca, dan gambar pendukung di sampingnya. Di bagian bawah kartu, terdapat informasi meta seperti tanggal pembuatan dan pembaruan terakhir, serta tombol "Edit Content" untuk mengubahnya. Namun, jika belum ada konten sama sekali, halaman akan menampilkan sebuah "empty state" atau halaman kosong yang informatif, yang mendorong pengguna untuk mulai membuat konten dengan menyediakan tombol "Create Content". Selain itu, halaman ini juga dilengkapi notifikasi sukses yang elegan dan otomatis menghilang untuk memberikan umpan balik kepada admin setelah berhasil melakukan suatu aksi.



Gambar 3. 114 Halaman edit konten tentang kami di bagian admin



Gambar 3. 115 Potongan kode halaman edit konten tentang kami di bagian admin

Penjelasan :

Halaman ini adalah sebuah formulir untuk mengedit konten halaman "Tentang Kami" yang dirancang dengan antarmuka yang sangat modern, profesional, dan mendukung mode gelap (dark mode). Formulir ini memungkinkan administrator untuk dengan mudah memperbarui judul dan isi konten utama halaman, di mana semua kolom isian sudah terisi otomatis dengan data yang ada saat ini. Selain teks, terdapat fitur manajemen gambar yang komprehensif, memungkinkan admin untuk mengunggah gambar pendukung baru, melihat pratinjau gambar yang sedang digunakan, dan bahkan menyediakan opsi kotak centang untuk menghapus gambar yang ada. Dengan desain yang kaya visual, pesan notifikasi keberhasilan dan kesalahan yang jelas, serta tombol aksi yang lugas, halaman ini menyediakan pengalaman pengelolaan konten yang efisien dan intuitif.

Penjelasan :

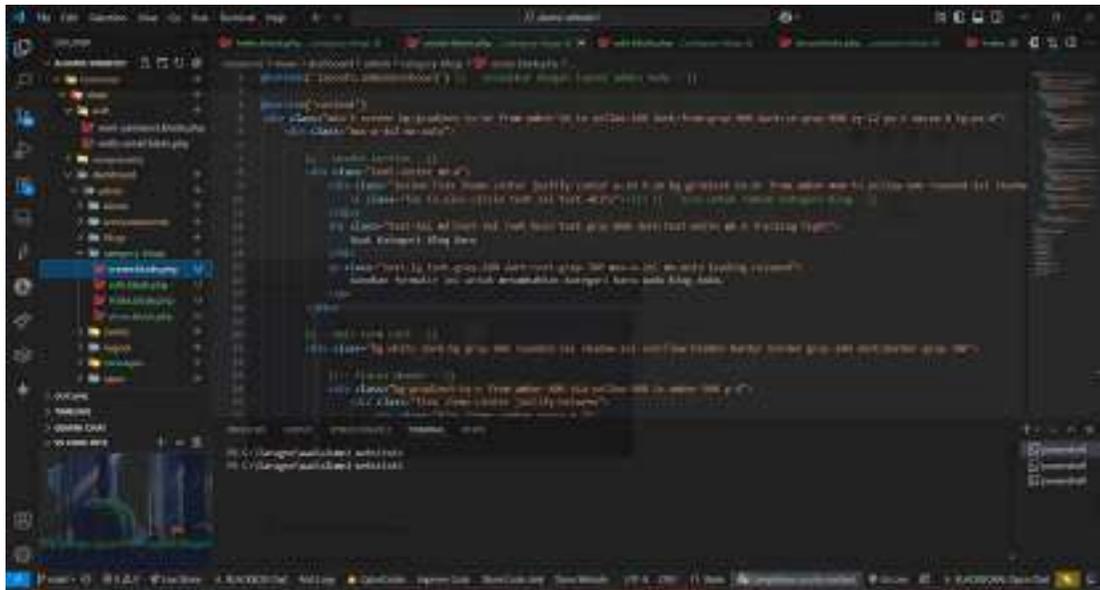
Halaman ini adalah halaman manajemen kategori blog untuk antarmuka admin, yang memungkinkan administrator mengelola kategori blog yang ada di situs web. Tampilannya dirancang agar responsif dan mudah digunakan, dengan fokus pada penyajian informasi yang jelas dan tindakan yang intuitif. Pada bagian atas halaman, terdapat judul besar "Manajemen Kategori Blog" dan tombol mencolok "Tambah Kategori Baru" yang mengarahkan pengguna ke halaman pembuatan kategori baru. Jika belum ada kategori yang terdaftar, halaman akan menampilkan pesan yang ramah pengguna dengan ikon yang relevan, menyarankan administrator untuk mulai menambahkan kategori. Namun, jika kategori sudah ada, halaman ini akan menampilkan tabel interaktif yang mencantumkan detail setiap kategori. Tabel tersebut menyertakan kolom seperti ID kategori, nama kategori, slug (URL ramah SEO), dan deskripsi kategori.

Deskripsi dipersingkat untuk menjaga kerapian tabel, dengan judul lengkap tersedia saat mengarahkan kursor. Setiap baris dalam tabel dilengkapi dengan opsi tindakan, yaitu tombol "Edit" dan "Hapus". Tombol "Edit" mengarahkan administrator ke halaman untuk memodifikasi detail kategori yang dipilih. Sementara itu, tombol "Hapus" dirancang dengan konfirmasi SweetAlert2, yang memberikan peringatan pop-up kepada administrator untuk memastikan mereka benar-benar ingin menghapus kategori, sekaligus memberi tahu mereka bahwa blog yang terkait dengan kategori tersebut akan diatur ke "Tidak Terkategori" setelah penghapusan.

Di bagian bawah tabel, terdapat tautan paginasi yang memungkinkan administrator menavigasi antar halaman jika jumlah kategori melebihi batas tampilan per halaman. Secara keseluruhan, halaman ini berfungsi sebagai pusat kendali yang efisien untuk mengelola dan mengatur konten blog melalui kategorisasi.



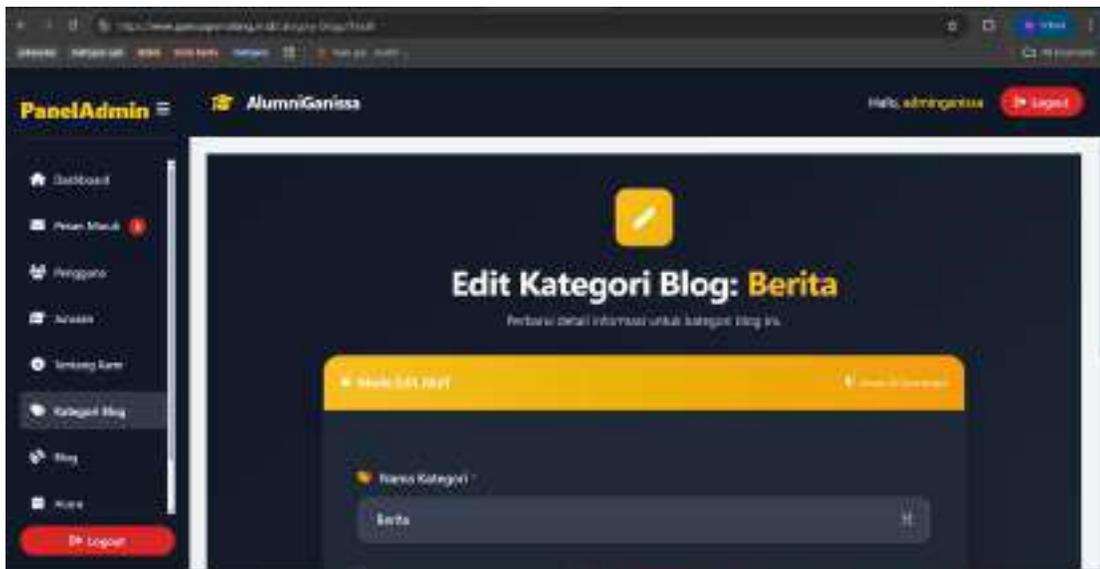
Gambar 3. 118 Halaman tambah kategori blog di bagian admin



Gambar 3. 119 Potongan kode halaman tambah kategori blog di bagian admin

Penjelasan :

Halaman ini berfungsi sebagai antarmuka "Buat Kategori Blog Baru" dalam dashboard admin, dirancang dengan estetika modern menggunakan Tailwind CSS dan mendukung mode gelap. Halaman ini menyediakan formulir intuitif bagi administrator untuk menambahkan kategori blog baru, lengkap dengan bidang untuk nama kategori (wajib) dan deskripsi (opsional). Validasi input disajikan secara dinamis, menampilkan pesan kesalahan yang jelas di bagian atas formulir jika ada masalah. Header formulir menampilkan status input aktif dan menekankan keamanan, sementara tombol "Buat Kategori" dan "Batal" di bagian bawah menawarkan opsi yang jelas kepada pengguna, dengan pesan informatif di bagian footer yang mengingatkan untuk memverifikasi data sebelum menyimpan.



Gambar 3. 120 Halaman edit kategori blog di bagian admin



Gambar 3. 121 Potongan kode halaman edit kategori blog di bagian admin

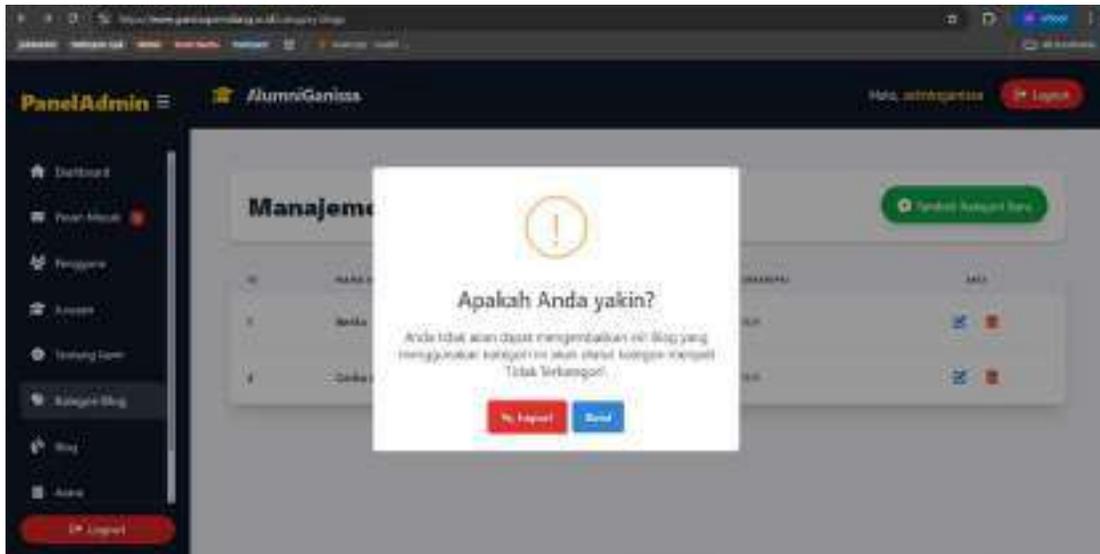
Penjelasan :

Halaman ini merupakan antarmuka "Edit Kategori Blog" dalam panel admin, yang dirancang untuk memungkinkan administrator memperbarui detail kategori blog yang sudah

ada. Tampilan modern dan responsif ini, lengkap dengan dukungan mode gelap, berfokus pada pengalaman pengguna yang efisien. Di bagian atas, halaman ini menampilkan judul "Edit Kategori Blog" yang dinamis, menunjukkan nama kategori yang sedang diedit (`{categoryBlog->name}`), serta pesan deskriptif yang mengarahkan pengguna untuk memperbarui informasi.

Bagian utama halaman adalah formulir pengeditan yang terstruktur dengan rapi, yang mencakup bidang input untuk nama kategori (wajib) dan deskripsi (opsional). Formulir ini secara otomatis mengisi bidang-bidang tersebut dengan data kategori yang sudah ada (`old('name', {categoryBlog->name})`), memudahkan pengeditan. Validasi kesalahan input ditampilkan dengan jelas melalui pesan peringatan yang menarik perhatian jika ada masalah, dan juga terdapat pesan sukses yang akan muncul setelah perubahan berhasil disimpan. Header formulir menandakan "Mode Edit Aktif" dan menjamin keamanan transaksi data.

Di bagian bawah formulir, terdapat tombol aksi "Batal" yang mengarahkan kembali ke daftar kategori, dan tombol "Perbarui Kategori" untuk menyimpan perubahan. Kedua tombol ini didesain menarik dengan ikon dan efek hover. Sebagai penutup, sebuah catatan informatif di bagian footer mengingatkan administrator untuk selalu memastikan kebenaran data sebelum menyimpannya. Halaman ini secara keseluruhan dirancang untuk memberikan pengalaman pengeditan kategori blog yang lancar dan aman.

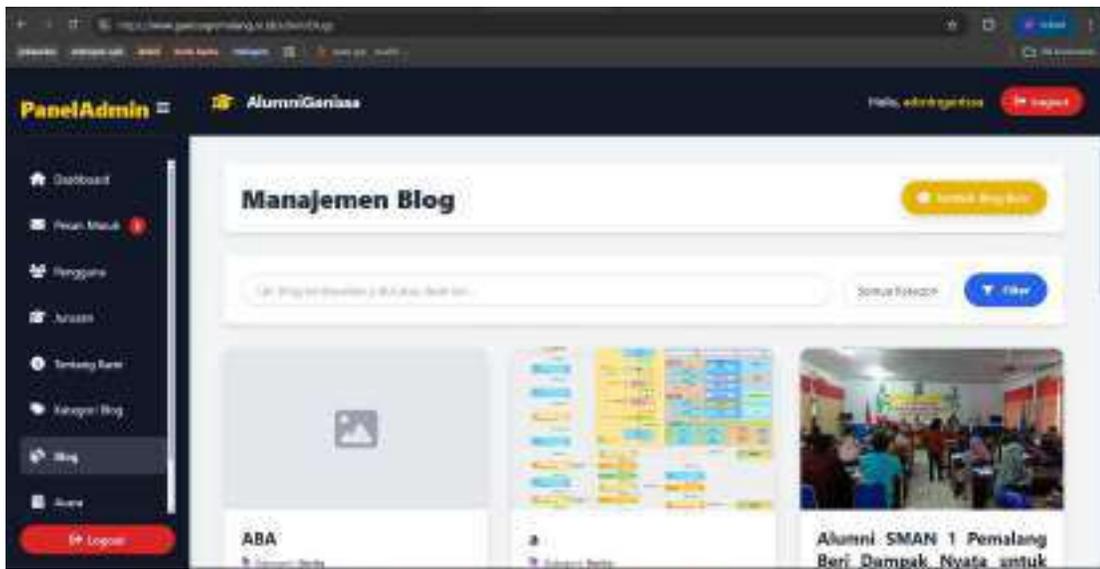


Gambar 3. 122 Tampilan admin ketika menghapus kategori blog yang tersedia

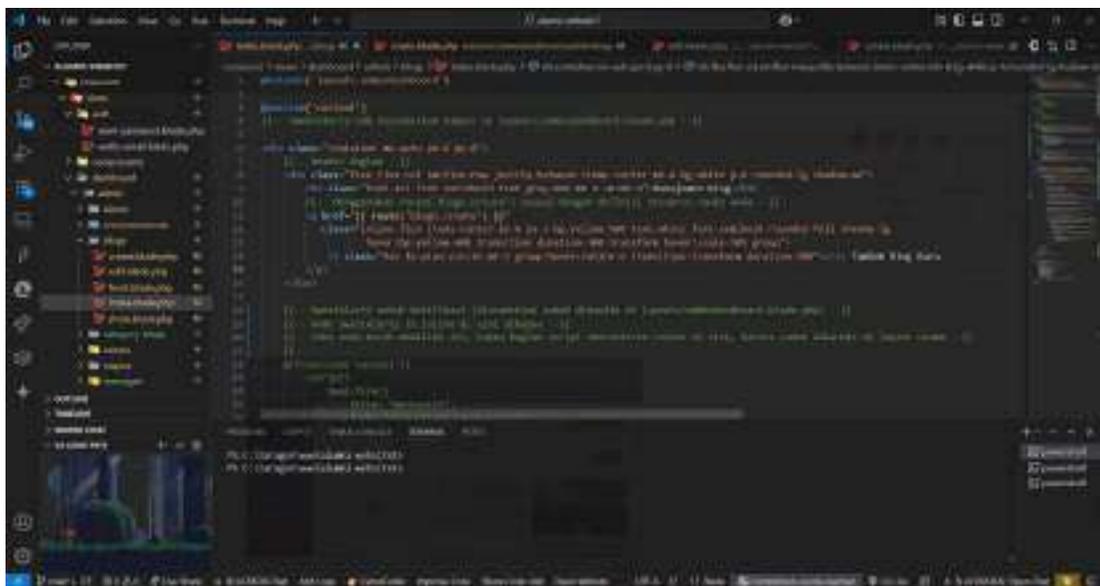
Penjelasan :

Halaman ini menampilkan antarmuka Panel Admin, khususnya bagian "Manajemen Kategori Blog", di mana administrator dapat melihat daftar kategori blog yang ada dalam bentuk tabel, lengkap dengan ID, nama, slug, deskripsi, serta opsi untuk mengedit atau menghapus setiap kategori. Sebuah jendela pop-up konfirmasi muncul di tengah layar, menanyakan "Apakah Anda yakin?" sebelum melanjutkan penghapusan kategori, dan memberikan peringatan bahwa blog yang terkait dengan kategori tersebut akan diatur menjadi "Tidak Terkategori" jika tindakan tersebut dilanjutkan, menawarkan pilihan "Ya, hapus!" atau "Batal".

7) Halaman Manajemen Blog



Gambar 3. 123 Halaman manajemen blog di bagian admin



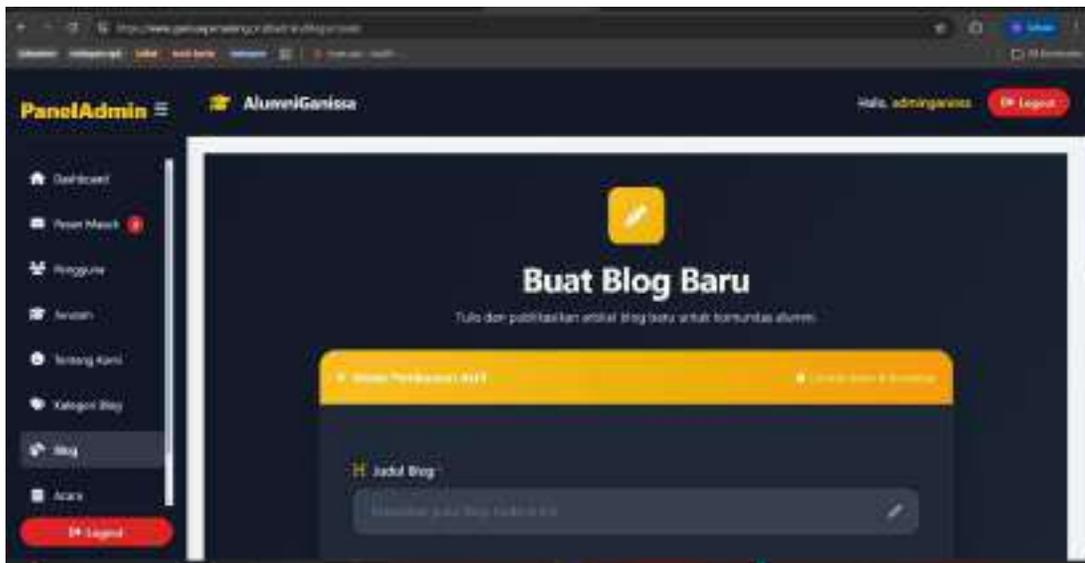
Gambar 3. 124 Potongan kode halaman manajemen blog di bagian admin

Penjelasan :

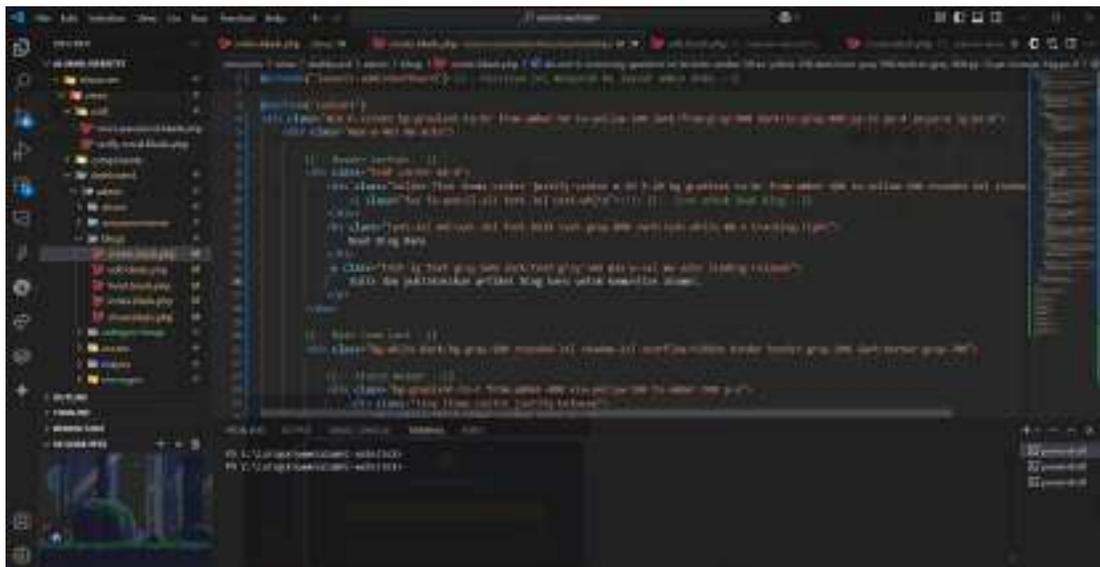
Halaman ini adalah dasbor utama untuk "Manajemen Blog" dalam panel admin, yang memungkinkan administrator mengelola semua postingan blog. Bagian atas halaman menyajikan judul yang jelas dan tombol "Tambah Blog Baru" yang menarik, mempermudah

pembuatan konten baru. Di bawahnya, terdapat fitur pencarian dan filter yang canggih, memungkinkan admin untuk mencari blog berdasarkan judul atau deskripsi, serta memfilter berdasarkan kategori tertentu dari daftar yang tersedia. Tombol "Filter" dan "Reset" memberikan kontrol penuh atas tampilan daftar blog, memastikan admin dapat dengan cepat menemukan dan menyaring informasi yang relevan.

Jika tidak ada blog yang ditemukan (baik karena belum ada atau hasil filter), pesan informatif akan ditampilkan. Sebaliknya, jika ada blog, halaman akan menampilkan daftar postingan dalam format kartu yang responsif. Setiap kartu blog mencakup gambar terkait (atau placeholder), judul yang disingkat, kategori (dengan "Tidak Terkategori" jika tidak ada), tanggal publikasi, status komentar (aktif/nonaktif), dan pratinjau deskripsi. Di bagian bawah setiap kartu, tersedia tautan "Baca Selengkapnya" untuk melihat detail penuh, serta tombol "Edit" dan "Hapus" untuk tindakan manajemen langsung. Fungsi hapus dilengkapi dengan konfirmasi SweetAlert2 untuk mencegah penghapusan yang tidak disengaja, dan di bagian bawah halaman terdapat tautan paginasi untuk menavigasi antar halaman blog.



Gambar 3. 125 Halaman tambah blog baru di bagian admin



Gambar 3. 126 Potongan kode halaman tambah blog baru di bagian admin

Penjelasan :

Halaman ini adalah antarmuka "Buat Blog Baru" dalam panel admin, dirancang dengan desain modern dan responsif, termasuk dukungan mode gelap, untuk memudahkan administrator dalam membuat postingan blog baru. Halaman ini diawali dengan judul yang jelas dan pesan ajakan untuk menulis konten bagi komunitas alumni. Bagian utama adalah formulir yang terstruktur rapi, mencakup bidang wajib seperti judul blog dan konten/deskripsi, serta bidang opsional seperti kategori (melalui dropdown), gambar blog (dengan validasi format dan ukuran), dan tanggal publikasi (yang juga memungkinkan penyimpanan sebagai draft). Fitur penting lainnya adalah tombol toggle untuk mengaktifkan atau menonaktifkan komentar. Validasi input disajikan dengan pesan kesalahan yang informatif, dan halaman diakhiri dengan tombol "Simpan Blog" serta tautan "Kembali" ke daftar blog, dilengkapi dengan pengingat untuk memeriksa data sebelum disimpan.



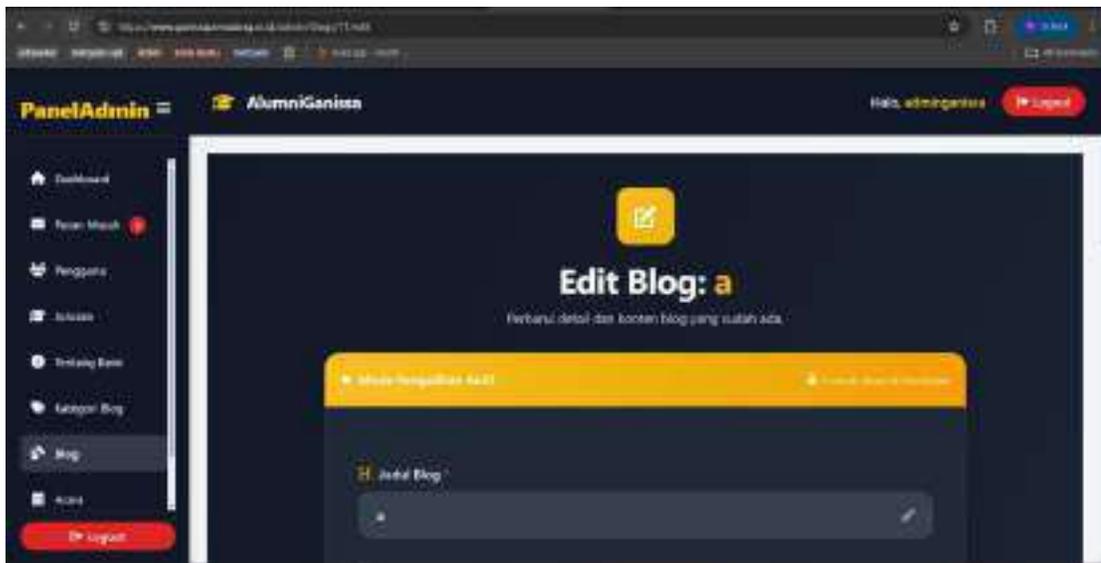
Gambar 3. 127 Halaman detail blog di bagian admin



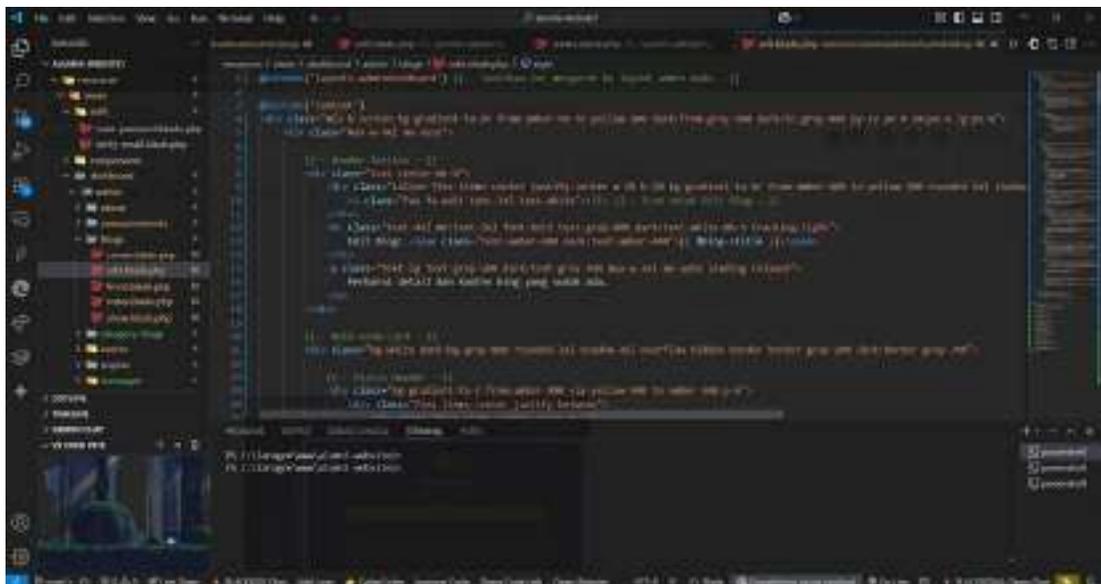
Gambar 3. 128 Potongan kode halaman detail blog di bagian admin

Penjelasan :

Halaman ini adalah tampilan detail untuk sebuah postingan blog dalam panel admin, yang menyajikan informasi lengkap tentang blog tersebut dalam tata letak yang bersih dan modern. Administrator dapat melihat gambar blog (jika ada), judul lengkap, deskripsi konten, serta metadata penting seperti kategori, tanggal publikasi, dan status komentar (aktif atau nonaktif). Bagian bawah detail blog juga menunjukkan kapan blog dibuat dan terakhir diperbarui. Selain itu, halaman ini menyediakan tombol untuk "Edit Blog" dan "Kembali ke Daftar", serta secara kondisional menampilkan bagian manajemen komentar jika komentar diaktifkan untuk blog tersebut. Dalam bagian komentar, admin dapat melihat semua komentar yang masuk, termasuk balasan, serta memiliki opsi untuk membalas setiap komentar atau menghapusnya dengan konfirmasi SweetAlert2.



Gambar 3. 129 Halaman edit blog di bagian admin



Gambar 3. 130 Potongan kode halaman edit blog di bagian admin

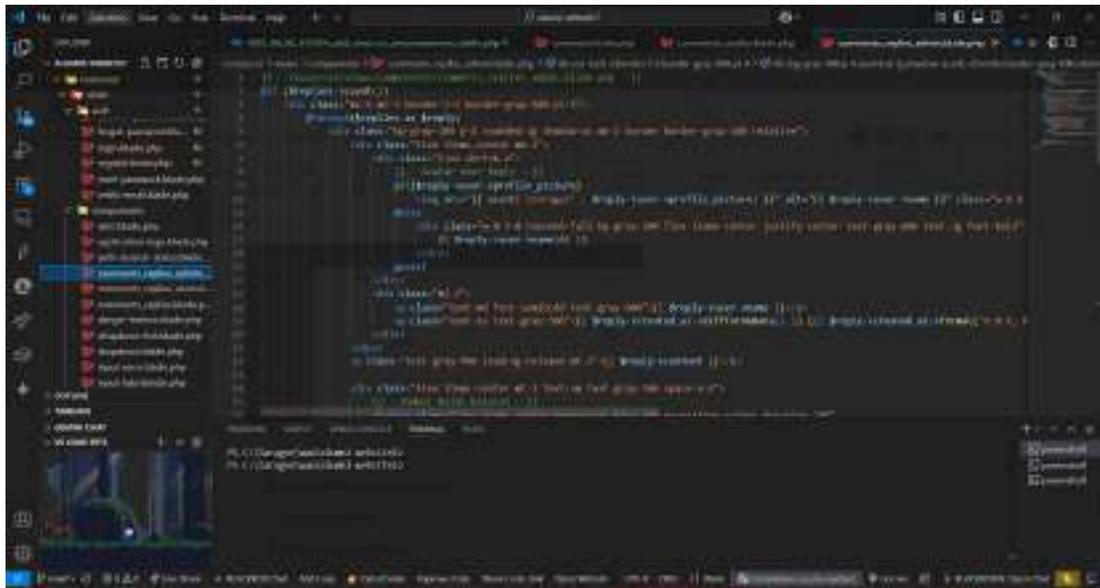
Penjelasan :

Halaman ini adalah antarmuka "Edit Blog" dalam panel admin, yang memungkinkan administrator memperbarui detail dan konten postingan blog yang sudah ada. Desainnya modern dan responsif, dengan dukungan mode gelap, serta menampilkan judul blog yang

sedang diedit. Formulir pengeditan secara otomatis mengisi kolom-kolom seperti judul, kategori (dropdown), deskripsi/konten, gambar blog saat ini (dengan opsi untuk mengganti atau menghapusnya), tanggal publikasi, dan status komentar, memudahkan administrator untuk melakukan perubahan. Validasi kesalahan input ditampilkan secara jelas, dan di bagian bawah halaman terdapat tombol "Perbarui Blog" untuk menyimpan perubahan serta tombol "Kembali" untuk membatalkan pengeditan, dengan pesan peringatan untuk memastikan data sudah benar sebelum diperbarui.



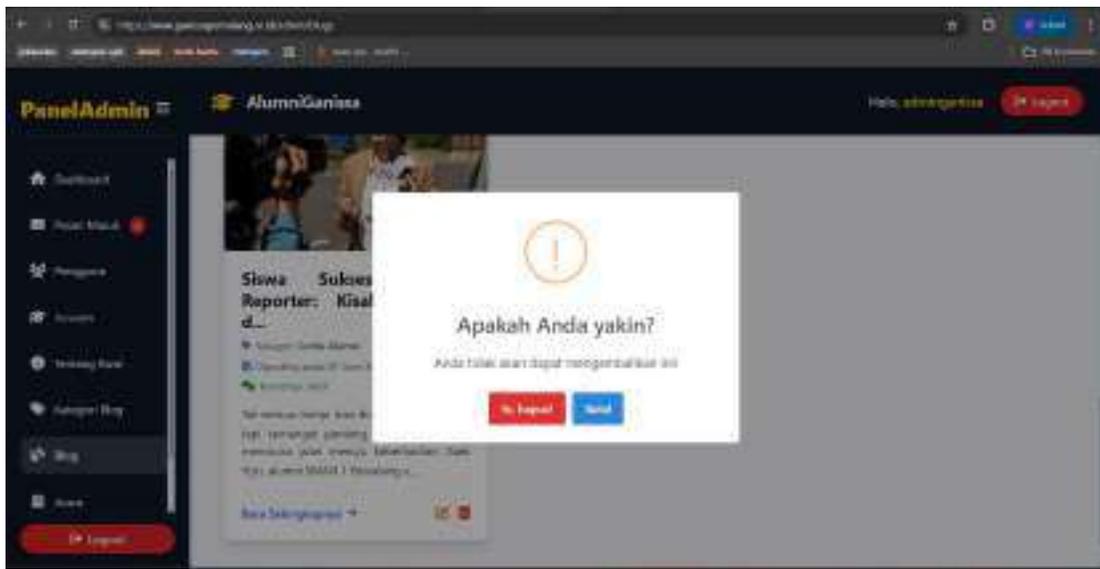
Gambar 3. 131 Tampilan bagian komentar di halaman blog untuk admin



Gambar 3. 132 Potongan kode yang menyusun tampilan bagian komentar di halaman blog untuk admin

Penjelasan :

Halaman ini adalah komponen terpisah, `comments_replies_admin.blade.php`, yang dirancang khusus untuk menampilkan struktur balasan bertingkat pada komentar blog dalam antarmuka admin. Komponen ini secara rekursif menampilkan setiap balasan komentar, lengkap dengan avatar pengguna (atau inisial nama), nama pengguna, dan stempel waktu (baik format relatif maupun absolut). Setiap balasan menyertakan konten balasan itu sendiri, serta opsi interaktif "Balas" yang dapat mengaktifkan formulir balasan tersembunyi, dan tombol "Hapus" yang dilengkapi dengan konfirmasi SweetAlert2 untuk memudahkan moderasi komentar oleh administrator. Struktur indentasi visual diterapkan untuk membedakan antara komentar utama dan balasan-balasan di bawahnya, menciptakan tampilan diskusi yang terorganisir.

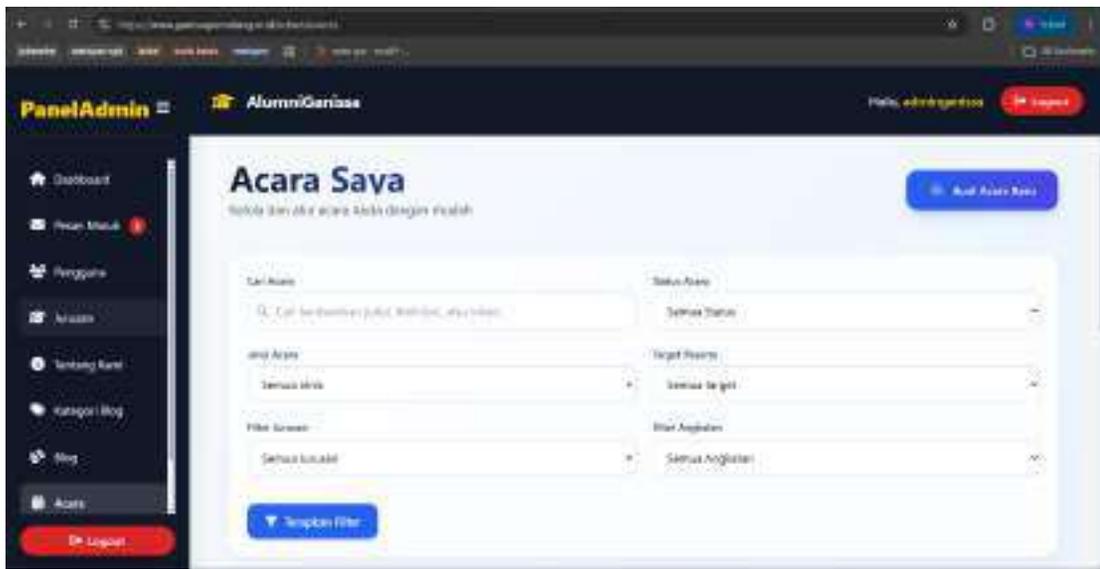


Gambar 3. 133 Tampilan ketika admin menghapus salah satu blog yang ada

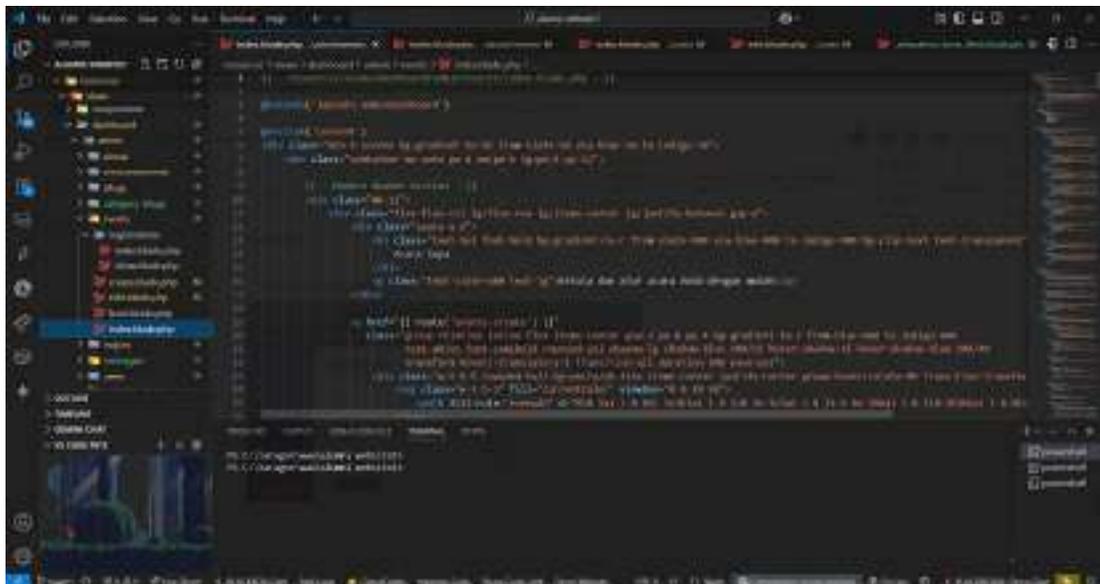
Penjelasan :

Saat tombol hapus ditekan, sebuah pop-up konfirmasi muncul di tengah layar, menanyakan "Apakah Anda yakin?" dengan peringatan bahwa tindakan tersebut tidak dapat dibatalkan, serta menawarkan pilihan "Ya, hapus!" atau "Batal" kepada administrator.

8) Halaman Manajemen Acara



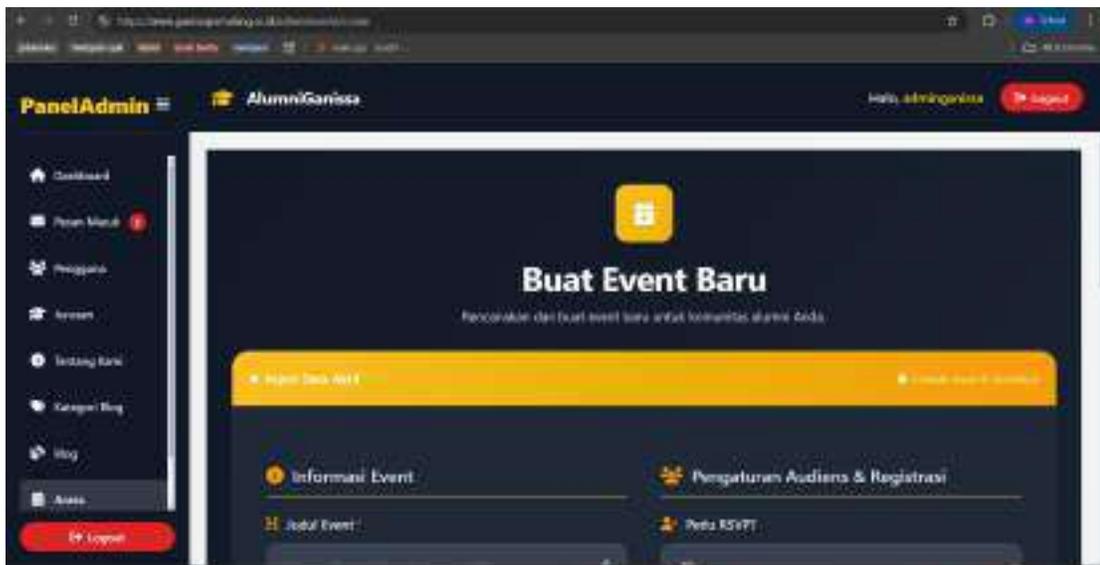
Gambar 3. 134 Halaman manajemen acara di bagian admin



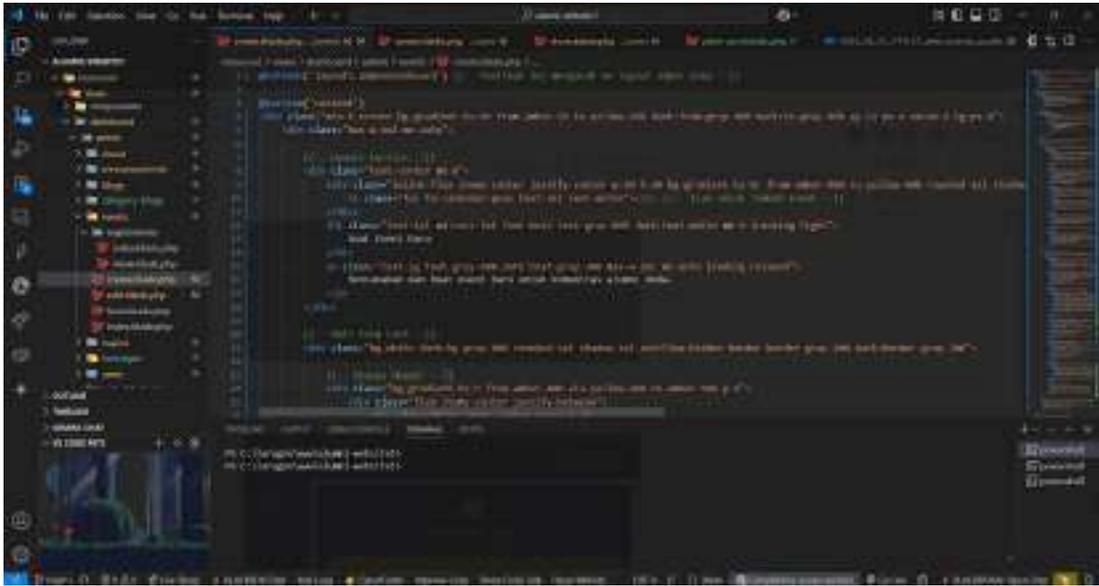
Gambar 3. 135 Potongan kode untuk halaman manajemen acara di bagian admin

Penjelasan :

Halaman ini adalah dasbor "Manajemen Acara" bagi administrator, dirancang dengan antarmuka modern dan responsif yang memungkinkan pengelolaan acara alumni secara komprehensif. Pada bagian atas, halaman menyajikan judul "Acara Saya" yang menonjol dan tombol "Buat Acara Baru" untuk menambah entri baru. Dilengkapi dengan formulir pencarian dan filter multifungsi, administrator dapat dengan mudah menyaring acara berdasarkan judul, deskripsi, lokasi, status (mendatang/selesai), jenis (berbayar/gratis), target peserta (semua, jurusan tertentu, angkatan tertentu), serta jurusan dan angkatan spesifik. Daftar acara ditampilkan dalam tata letak kartu yang menarik, masing-masing menyertakan gambar, detail harga atau status gratis, status RSVP, tanggal, waktu, lokasi, target audiens, dan jumlah peserta terdaftar (jika berlaku), dengan opsi untuk mengedit, melihat detail pendaftaran, atau menghapus acara melalui konfirmasi SweetAlert2. Jika tidak ada acara yang ditemukan, halaman akan menampilkan pesan kosong yang relevan dengan filter yang diterapkan.



Gambar 3. 136 Halaman buat acara baru di bagian admin



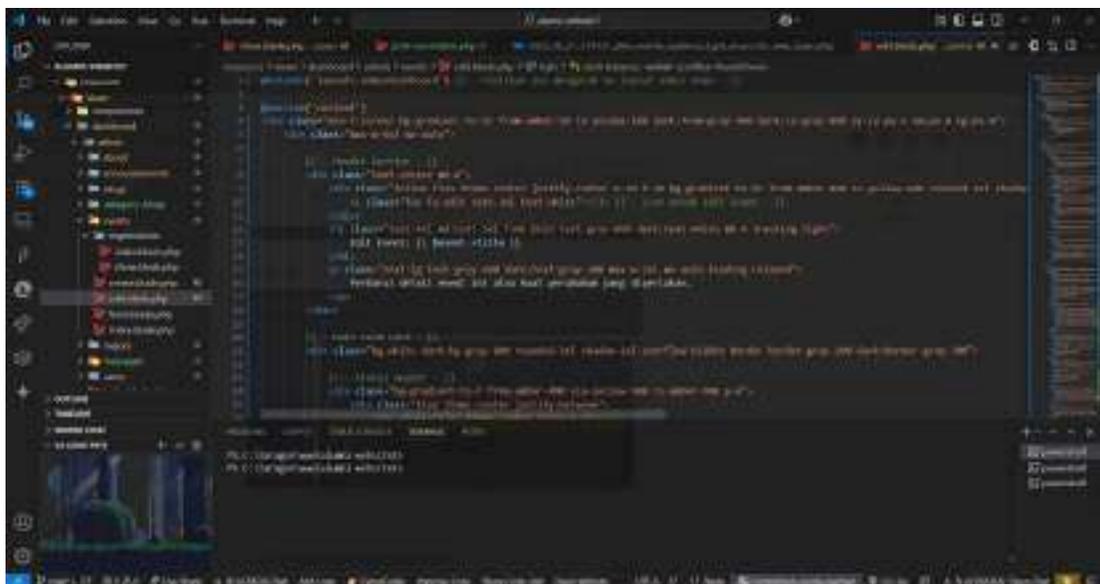
Gambar 3. 137 Potongan kode halaman buat acara baru di bagian admin

Penjelasan :

Halaman ini adalah antarmuka "Buat Event Baru" dalam panel admin, dirancang dengan gaya modern dan responsif untuk memudahkan administrator merencanakan dan membuat acara alumni. Halaman ini dibagi menjadi dua bagian formulir utama: "Informasi Event" yang mencakup detail dasar seperti judul, deskripsi, tanggal, waktu, lokasi, dan gambar event (opsional), serta "Pengaturan Audiens & Registrasi" yang memungkinkan konfigurasi lebih lanjut. Pada bagian kedua, admin dapat menentukan apakah RSVP diperlukan (dengan opsi batasan peserta), jenis audiens target (semua alumni, jurusan tertentu, tahun tertentu, atau kombinasi keduanya yang dinamis menampilkan pilihan jurusan dan tahun lulus), dan tipe pembayaran (gratis atau berbayar beserta harganya). Formulir ini dilengkapi dengan validasi input yang jelas dan dinamis berdasarkan pilihan pengguna, serta tombol "Buat Event" dan "Batal" di bagian akhir, dengan pesan peringatan untuk memastikan data yang dimasukkan sudah benar.



Gambar 3. 138 Halaman edit acara di bagian admin

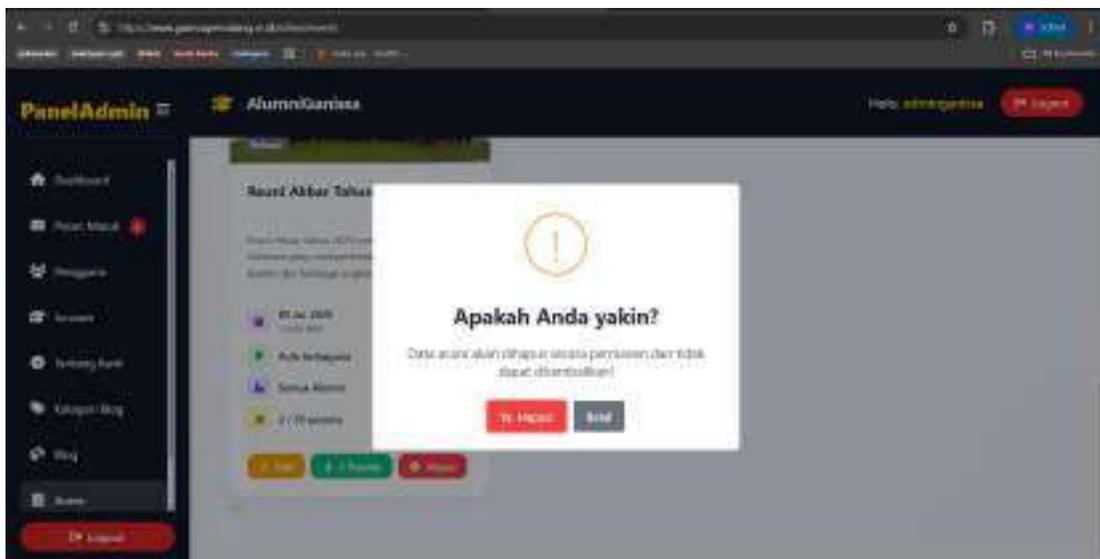


Gambar 3. 139 Potongan kode untuk halaman edit acara di bagian admin

Penjelasan :

Halaman ini adalah antarmuka "Edit Event" dalam panel admin, yang memungkinkan administrator memperbarui detail Event alumni yang sudah ada. Desainnya modern dan responsif, dengan dukungan mode gelap, serta menampilkan judul Event yang sedang diedit.

Formulir pengeditan secara otomatis mengisi kolom-kolom seperti judul, deskripsi, tanggal, waktu, lokasi, dan gambar Event (dengan pratinjau gambar saat ini serta opsi untuk mengganti atau menghapus). Selain itu, admin dapat menyesuaikan pengaturan RSVP (termasuk batas peserta), tipe audiens target (yang secara dinamis menampilkan pilihan jurusan dan/atau tahun lulus yang relevan), dan tipe pembayaran (gratis atau berbayar beserta harganya). Formulir ini dilengkapi dengan validasi input yang jelas dan dinamis berdasarkan pilihan pengguna, serta tombol "Perbarui Event" untuk menyimpan perubahan dan tombol "Kembali" untuk membatalkan pengeditan, dengan pesan peringatan untuk memastikan data yang dimasukkan sudah benar sebelum diperbarui.

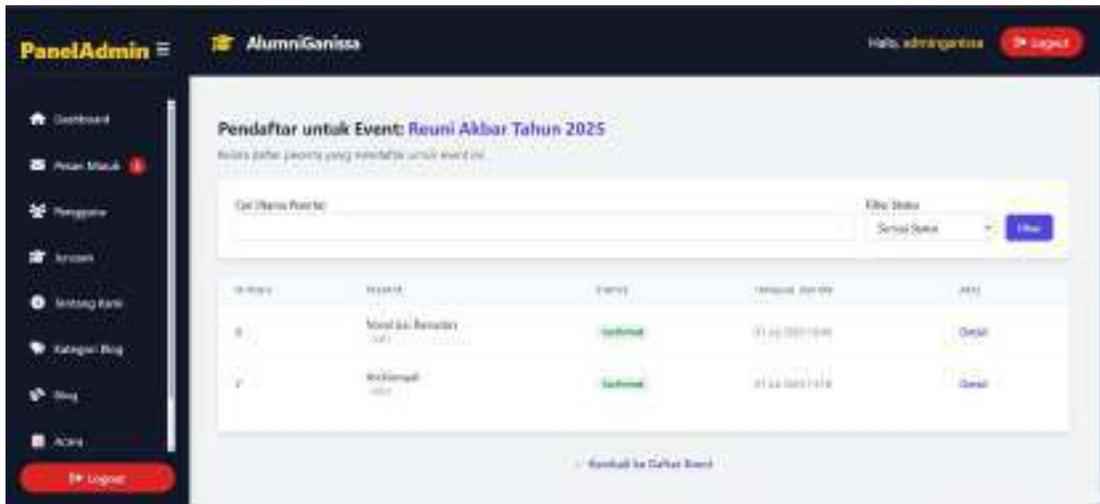


Gambar 3. 140 Tampilan ketika admin ingin menghapus acara yang ada

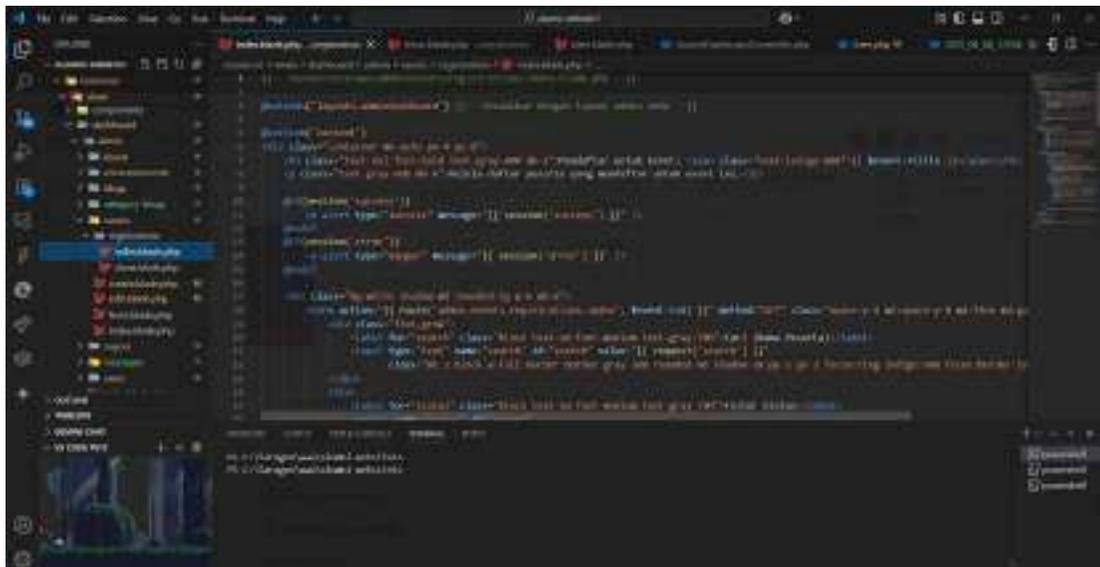
Penjelasan :

Saat mencoba menghapus Event, sebuah pop-up konfirmasi muncul di tengah layar, menanyakan "Apakah Anda yakin?" dengan peringatan bahwa "Data acara akan dihapus

secara permanen dan tidak dapat dikembalikan!", memberikan opsi "Ya, Hapus!" atau "Batal" untuk memastikan tindakan admin.



Gambar 3. 141 Halaman detail partisipan yang terdaftar untuk acara tertentu di bagian admin



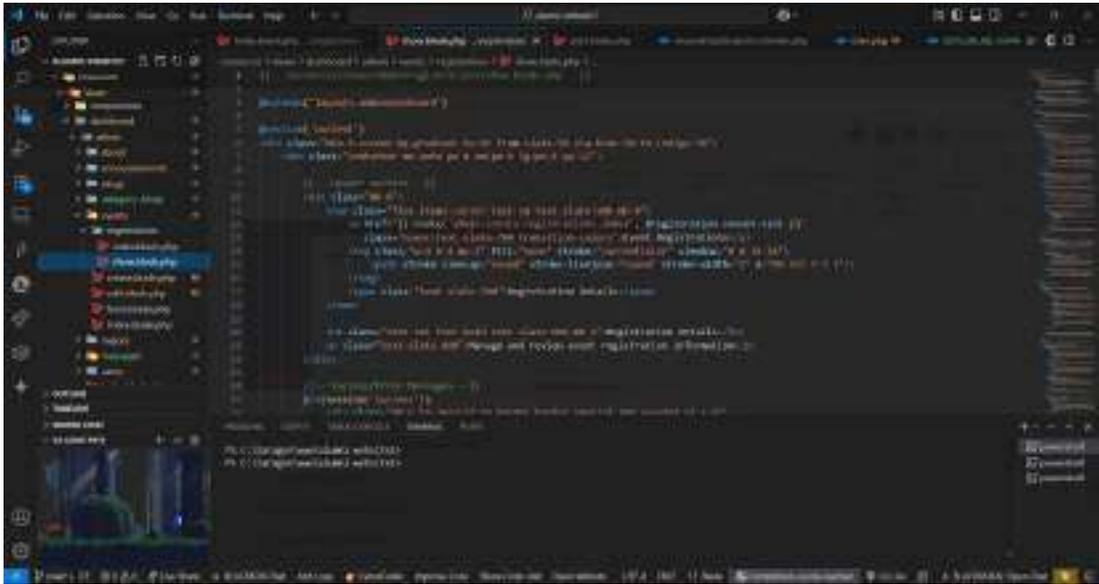
Gambar 3. 142 Potongan kode untuk halaman detail partisipan yang terdaftar untuk acara tertentu di bagian admin

Penjelasan :

Halaman ini adalah antarmuka manajemen pendaftaran peserta untuk sebuah event tertentu di panel admin, yang secara khusus menampilkan judul event yang relevan. Administrator dapat mencari peserta berdasarkan nama dan memfilter daftar berdasarkan status pendaftaran (seperti 'confirmed', 'pending confirmation', 'pending payment', atau 'cancelled'), dengan setiap status ditandai oleh badge berwarna yang berbeda. Daftar pendaftaran disajikan dalam tabel yang rapi, mencakup ID pendaftaran, nama peserta (beserta jurusan dan angkatannya), status, dan tanggal pendaftaran. Untuk setiap pendaftaran, tersedia opsi "Detail", serta tombol aksi cepat seperti "Konfirmasi", "Tolak", atau "Hapus", yang muncul secara kondisional berdasarkan status pendaftaran, memungkinkan manajemen yang efisien. Halaman juga dilengkapi dengan paginasi dan tautan kembali ke daftar event.



Gambar 3. 143 Halaman detail registrasi dari suatu acara di bagian admin



Gambar 3. 144 Potongan kode untuk halaman detail registrasi dari suatu acara di bagian admin

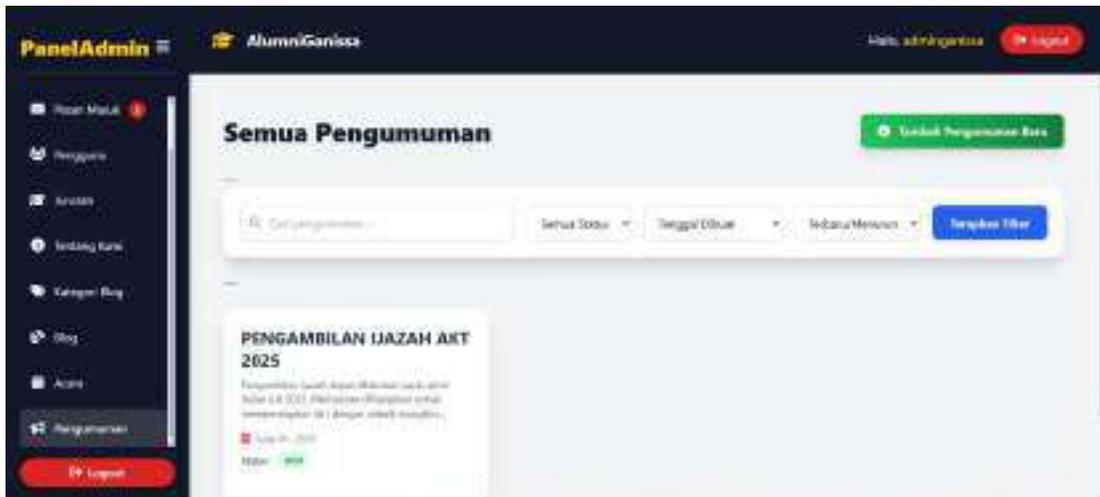
Penjelasan :

Halaman ini adalah detail pendaftaran tunggal untuk sebuah event di panel admin, yang dirancang untuk memberikan tinjauan komprehensif dan kemampuan manajemen atas setiap pendaftaran peserta. Di bagian atas, terdapat breadcrumb navigasi yang memudahkan administrator kembali ke daftar pendaftaran event, diikuti oleh judul "Registration Details" yang jelas. Halaman ini juga dilengkapi dengan notifikasi success atau error yang informatif setelah tindakan admin. Konten utama dibagi menjadi beberapa bagian kartu, termasuk "Participant Information" yang merinci data alumni seperti nama lengkap, email, jurusan, dan tahun kelulusan, serta "Event Information" yang menampilkan judul, tanggal, waktu, lokasi, jenis event (berbayar atau gratis), dan harga event (jika berbayar).

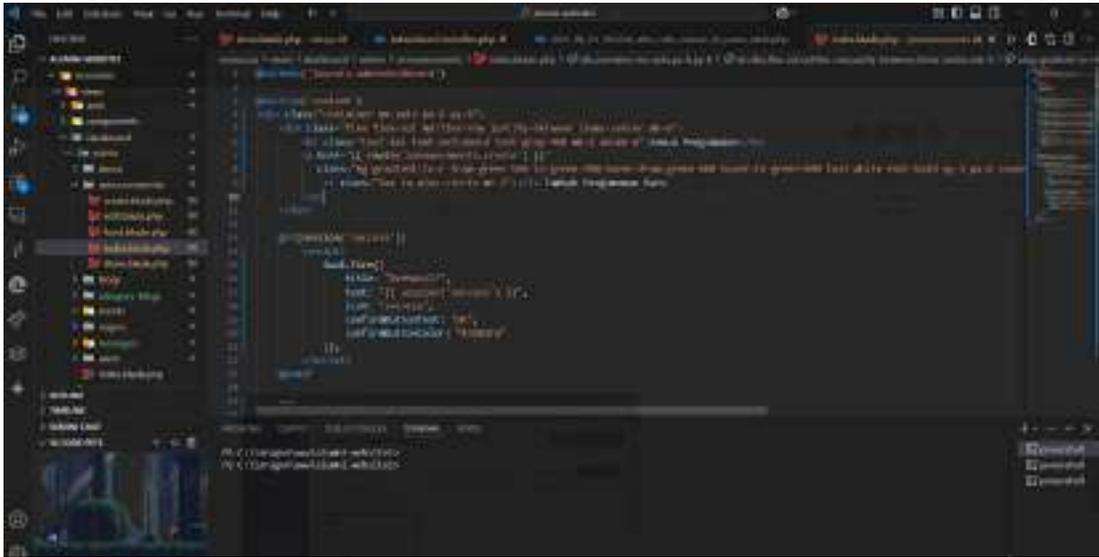
Bagian terpenting dari halaman ini adalah "Payment Proof" (jika event berbayar), di mana admin dapat melihat bukti pembayaran yang diunggah oleh peserta atau pesan jika belum ada bukti yang diunggah. Di kolom samping, terdapat informasi "Registration Status" yang

menampilkan status pendaftaran saat ini (seperti 'Confirmed', 'Pending Confirmation', 'Pending Payment', atau 'Rejected') dengan visual yang jelas. Terakhir, bagian "Actions" menyediakan tombol-tombol fungsional yang kondisional, seperti "Confirm Registration" atau "Reject Registration" jika statusnya 'pending_confirmation', serta tombol "Delete Registration" yang selalu tersedia untuk menghapus pendaftaran dengan konfirmasi melalui pop-up JavaScript. Desain keseluruhan fokus pada kejelasan informasi dan kemudahan interaksi bagi administrator.

9) Halaman Manajemen Pengumuman



Gambar 3. 145 Halaman manajemen pengumuman di bagian admin

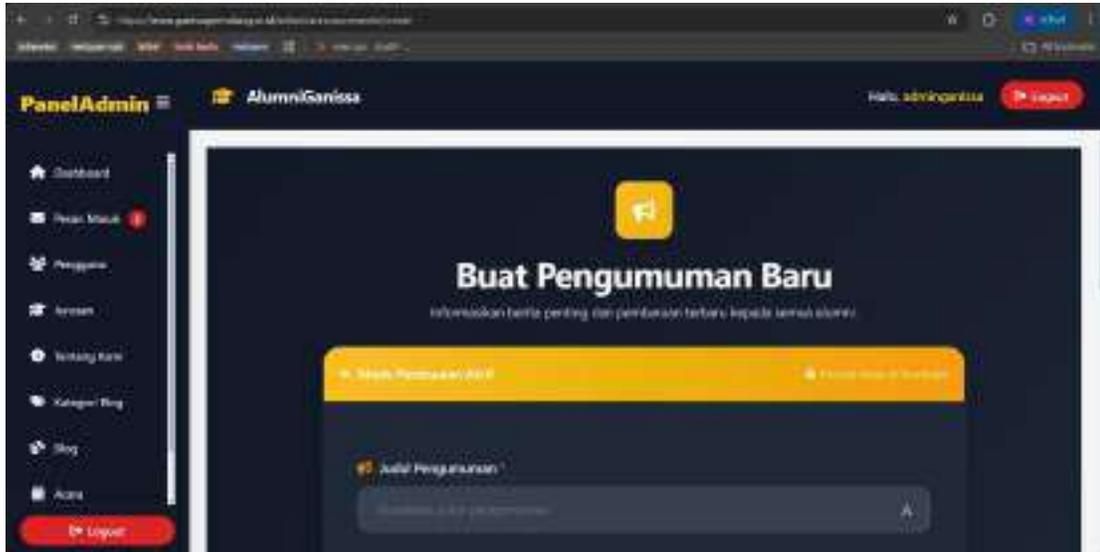


Gambar 3. 146 Potongan kode untuk halaman manajemen pengumuman di bagian admin

Penjelasan :

Halaman ini adalah dasbor "Semua Pengumuman" dalam panel admin, yang memungkinkan administrator untuk mengelola pengumuman yang ada. Bagian atas halaman menampilkan judul yang jelas dan tombol "Tambah Pengumuman Baru" untuk membuat pengumuman baru, disertai notifikasi SweetAlert2 jika ada operasi yang berhasil. Di bawahnya, terdapat formulir pencarian dan filter yang lengkap, memungkinkan admin mencari berdasarkan teks, memfilter berdasarkan status (aktif/tidak aktif), serta mengurutkan pengumuman berdasarkan tanggal dibuat, tanggal publikasi, judul, atau status aktif, dengan opsi untuk mengurutkan secara menaik atau menurun dan menghapus semua filter. Jika tidak ada pengumuman yang ditemukan, pesan informatif akan muncul. Sebaliknya, pengumuman yang ada ditampilkan dalam bentuk kartu grid yang menarik, setiap kartu menyajikan judul, ringkasan deskripsi, tanggal publikasi, dan status aktif/tidak aktif, bersama dengan tombol untuk melihat detail penuh, mengedit, atau menghapus pengumuman

(dengan konfirmasi). Halaman ini juga dilengkapi dengan navigasi paginasi untuk memudahkan penelusuran.



Gambar 3. 147 Halaman buat pengumuman baru di bagian admin



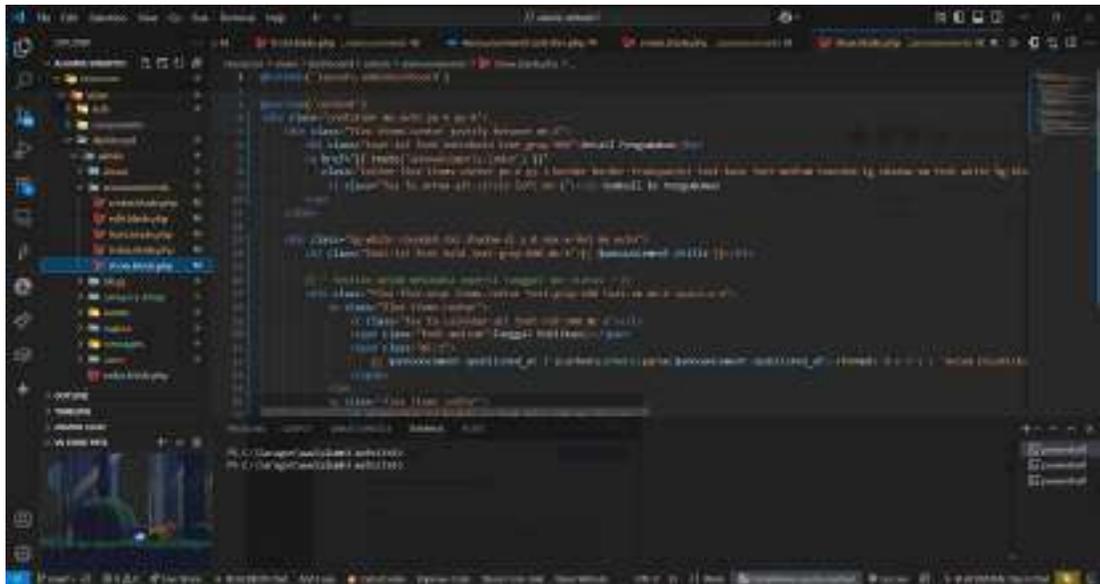
Gambar 3. 148 Potongan kode untuk halaman buat pengumuman baru di bagian admin

Penjelasan :

Halaman ini adalah antarmuka "Buat Pengumuman Baru" dalam panel admin, dirancang dengan desain modern dan responsif, termasuk dukungan mode gelap, yang memungkinkan administrator untuk membuat dan mempublikasikan pengumuman penting bagi komunitas alumni. Di bagian atas, halaman ini menampilkan judul yang jelas dan pesan yang mendorong admin untuk menginformasikan berita terbaru. Formulir utama mencakup bidang wajib seperti judul pengumuman dan deskripsi/konten, serta bidang opsional untuk tanggal publikasi (memungkinkan penjadwalan atau publikasi instan). Fitur penting lainnya adalah tombol toggle "Pengumuman Aktif" yang mengontrol visibilitas pengumuman bagi publik, dengan penjelasan singkat mengenai fungsinya. Halaman ini juga menampilkan pesan sukses atau kesalahan validasi yang jelas, dan diakhiri dengan tombol "Buat Pengumuman" untuk menyimpan, serta tombol "Kembali" untuk navigasi, lengkap dengan pengingat untuk memastikan data telah diisi dengan benar.



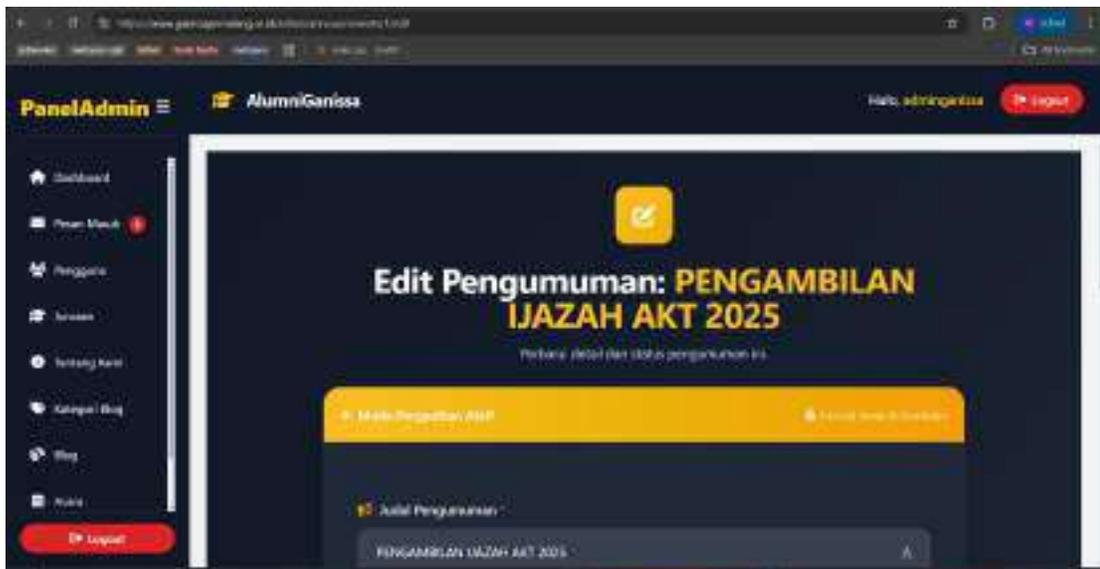
Gambar 3. 149 Halaman detail pengumuman di bagian admin



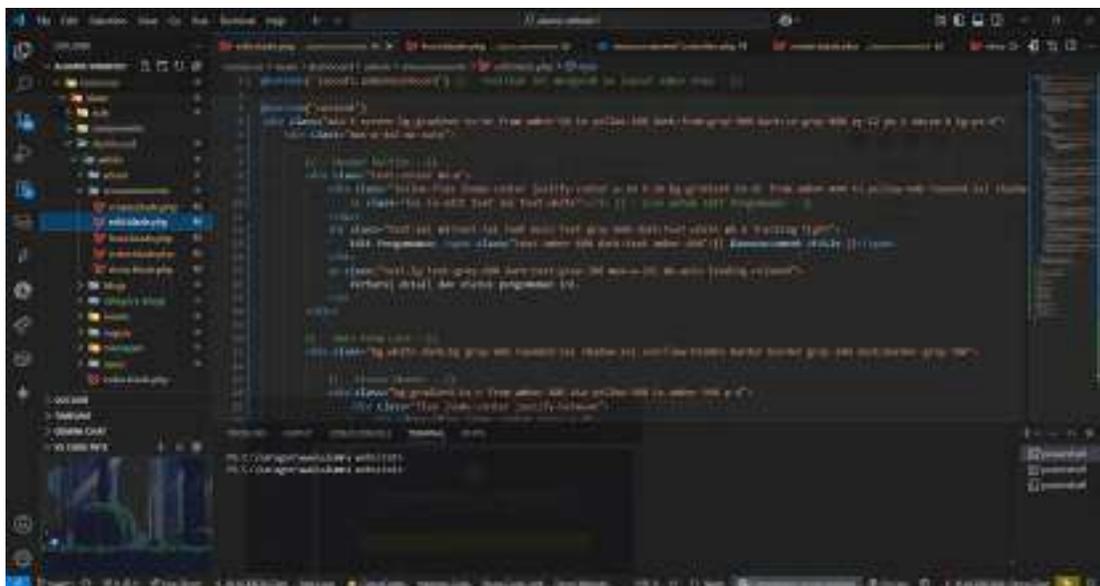
Gambar 3. 150 Potongan kode untuk halaman detail pengumuman di bagian admin

Penjelasan :

Halaman ini adalah tampilan "Detail Pengumuman" dalam panel admin, yang dirancang untuk menampilkan informasi lengkap mengenai sebuah pengumuman spesifik. Bagian atas halaman menyertakan judul "Detail Pengumuman" dan tombol "Kembali ke Pengumuman" untuk navigasi mudah kembali ke daftar pengumuman. Konten utama disajikan dalam kartu yang elegan, dimulai dengan judul pengumuman, diikuti oleh metadata penting seperti tanggal publikasi dan status aktif/tidak aktif (ditampilkan dengan badge berwarna yang jelas). Bagian deskripsi menampilkan isi pengumuman secara penuh, dan di bagian bawah terdapat tombol "Edit Pengumuman" untuk memodifikasi informasi, serta tombol "Hapus Pengumuman" yang memungkinkan administrator menghapus pengumuman setelah konfirmasi.



Gambar 3. 151 Halaman edit pengumuman di bagian admin

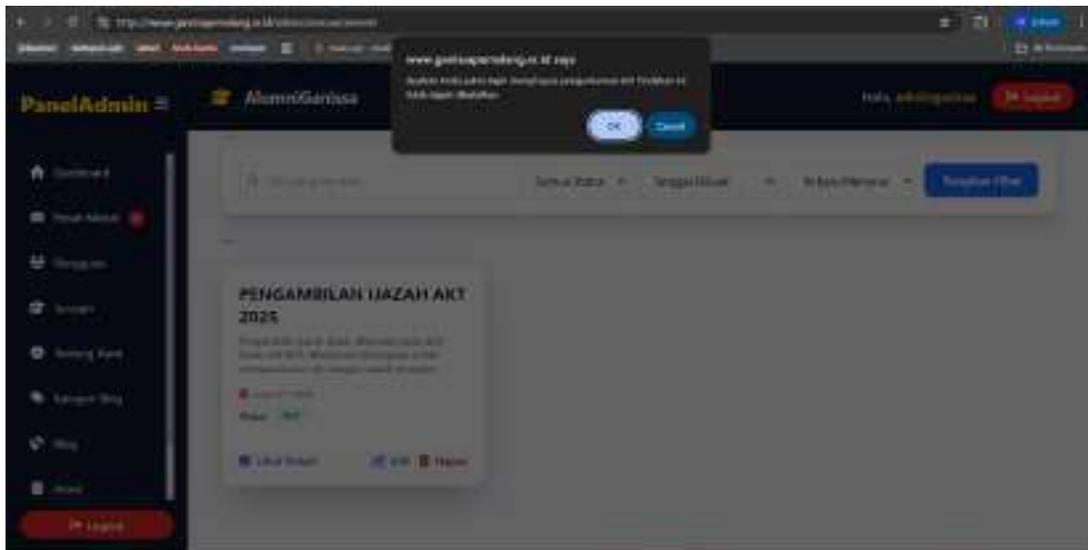


Gambar 3. 152 Potongan kode untuk halaman edit pengumuman di bagian admin

Penjelasan :

Halaman ini adalah antarmuka "Edit Pengumuman" dalam panel admin, yang dirancang untuk memungkinkan administrator memperbarui detail pengumuman yang sudah ada. Tampilannya modern dan responsif, dengan dukungan mode gelap, dan menampilkan

judul pengumuman yang sedang diedit. Formulir pengeditan secara otomatis mengisi kolom-kolom seperti judul pengumuman dan deskripsi/konten, serta bidang opsional untuk tanggal publikasi (yang dapat diatur ulang). Admin juga dapat mengubah status pengumuman menjadi aktif atau tidak aktif melalui tombol toggle. Halaman ini menampilkan pesan sukses atau kesalahan validasi yang jelas, dan diakhiri dengan tombol "Perbarui Pengumuman" untuk menyimpan perubahan serta tombol "Kembali" untuk membatalkan pengeditan, dengan pesan pengingat untuk memastikan data yang dimasukkan sudah benar.



Gambar 3. 153 Tampilan ketika admin mencoba untuk menghapus pengumuman yang ada

Penjelasan :

Saat admin berinteraksi untuk menghapus sebuah pengumuman, sebuah jendela pop-up konfirmasi muncul, meminta persetujuan dengan pesan "Apakah Anda yakin ingin menghapus pengumuman ini? Tindakan ini tidak dapat dibatalkan.", memberikan pilihan "OK" untuk melanjutkan atau "Cancel" untuk membatalkan tindakan.

C. Target User : Alumni

1) Halaman Dashboard Alumni



Gambar 3. 154 Halaman dashboard alumni



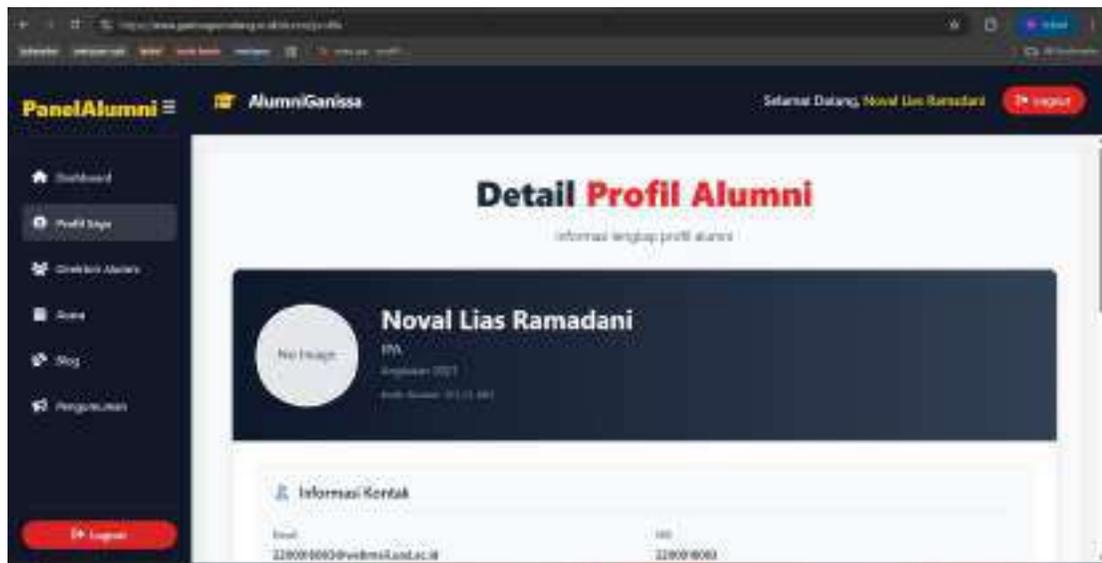
Gambar 3. 155 Potongan kode halaman dashboard alumni

Penjelasan :

Halaman ini adalah dasbor utama untuk alumni, menampilkan sambutan hangat dengan desain hero section modern yang interaktif, lengkap dengan gradient dan animasi visual yang menenangkan. Alumni disambut dengan pesan personal "Selamat Datang Kembali, Alumni Terhormat" dan ajakan untuk terhubung dengan jaringan profesional, menemukan peluang, serta berkontribusi pada almamater, dengan tombol besar "Jelajahi Acara" sebagai call-to-action utama. Di bawahnya, terdapat bagian "Acara Mendatang" yang menyajikan pratinjau tiga acara terdekat dalam format kartu yang menarik, menampilkan gambar, judul, tanggal, waktu, lokasi, dan deskripsi singkat, dengan tautan untuk melihat detail lebih lanjut atau menjelajahi semua acara jika tidak ada acara yang segera akan datang.

Selanjutnya, halaman ini menampilkan bagian "Kisah Alumni" yang menyoroti tiga artikel blog terbaru, dirancang dalam format kartu serupa dengan gambar, judul, tanggal publikasi, dan cuplikan kisah inspiratif, lengkap dengan tautan "Baca Kisah" untuk detail penuh dan opsi untuk melihat semua kisah. Terakhir, bagian "Pengumuman Terbaru" menyajikan tiga pengumuman paling relevan dalam format kartu yang ringkas, menampilkan judul, deskripsi singkat, tanggal publikasi, dan indikator status aktif/tidak aktif, dengan opsi untuk melihat semua pengumuman. Desain keseluruhan halaman ini bertujuan untuk memudahkan alumni dalam mengakses informasi penting, acara, dan kisah inspiratif, serta mendorong interaksi dalam komunitas.

2) Halaman Profile Alumni



Gambar 3. 156 Halaman detail profile alumni

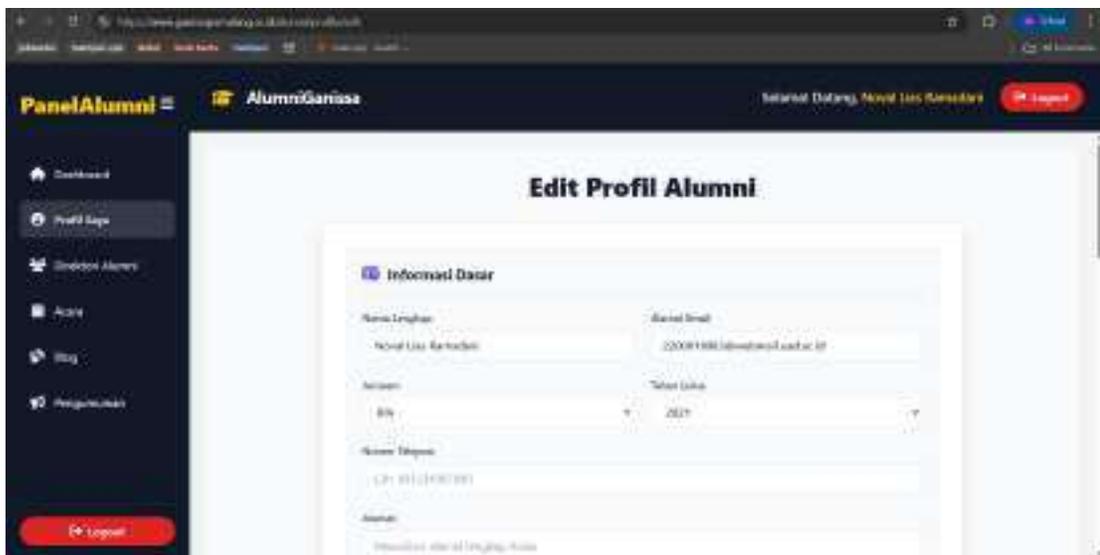


Gambar 3. 157 Potongan kode halaman detail profile alumni

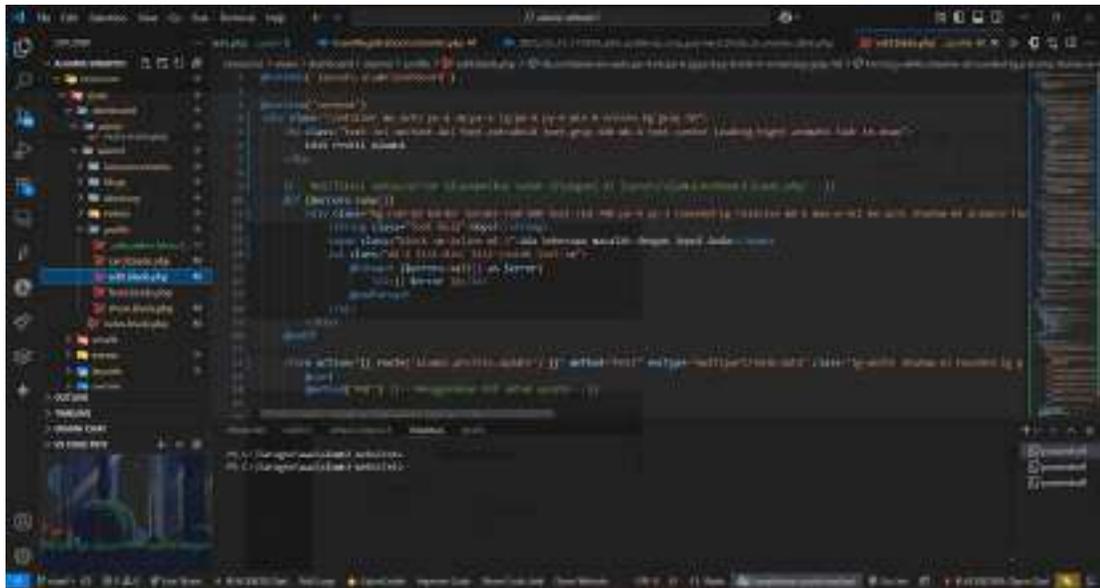
Penjelasan :

Halaman ini menampilkan detail profil lengkap seorang alumni dalam dasbor alumni, dirancang dengan estetika modern dan responsif. Bagian atas halaman menyambut pengguna

dengan judul "Detail Profil Alumni" yang menawan. Konten utama disajikan dalam kartu besar yang terstruktur, diawali dengan header profil alumni yang menampilkan foto profil, nama lengkap, jurusan, angkatan, dan kode alumni. Selanjutnya, profil dibagi menjadi beberapa bagian informatif seperti "Informasi Kontak" (mencakup email, NIS, telepon, tahun lulus, alamat, dan bio), "Informasi Karier" (detail pekerjaan saat ini, perusahaan, posisi, serta tautan LinkedIn dan website pribadi), dan "Riwayat Pendidikan" (daftar jenjang pendidikan, institusi, dan tahunnya). Halaman ini juga dilengkapi dengan notifikasi SweetAlert2 untuk feedback operasi dan dua tombol aksi utama di bagian bawah: "Edit Profil" untuk memodifikasi informasi, dan "Cetak Kartu Alumni" untuk menghasilkan kartu identitas.



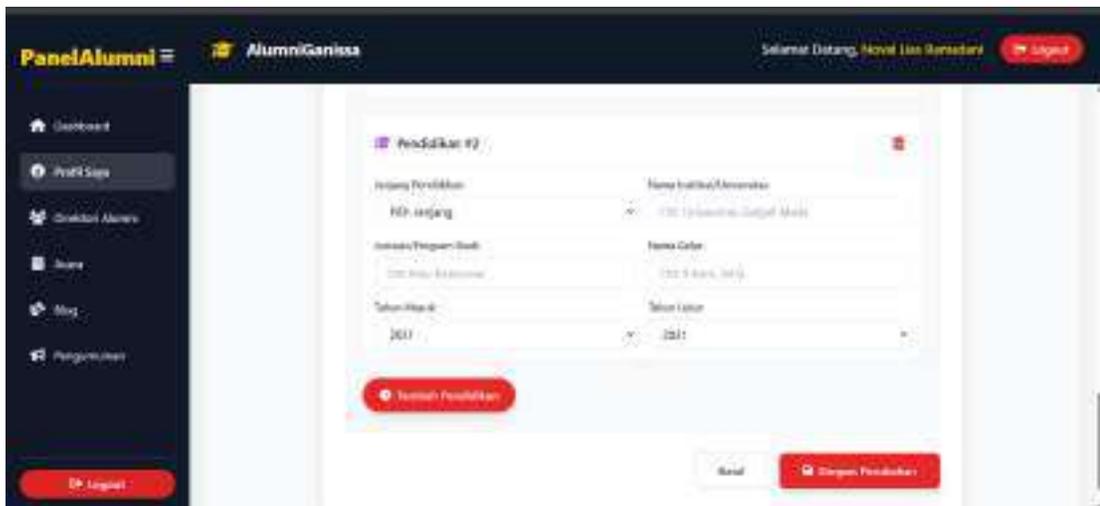
Gambar 3. 158 Halaman edit profile alumni



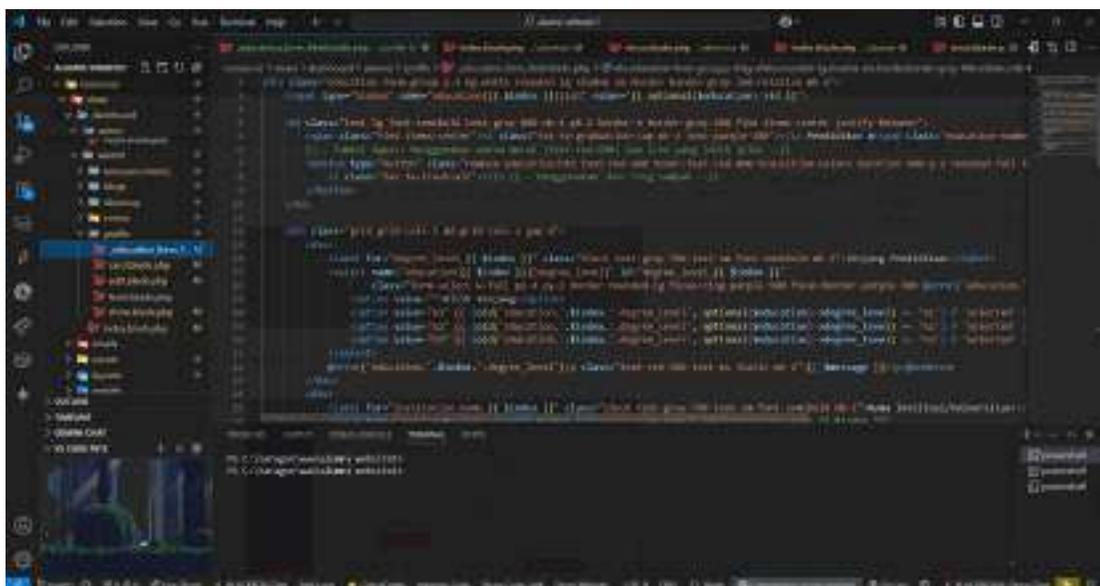
Gambar 3. 159 Potongan kode halaman edit profile alumni

Penjelasan :

Halaman ini adalah antarmuka "Edit Profil Alumni", yang memungkinkan alumni memperbarui informasi pribadi dan riwayat pendidikan mereka dengan desain yang bersih dan modern. Bagian utama halaman adalah formulir yang terbagi menjadi tiga segmen: "Informasi Dasar" untuk data seperti nama, email, jurusan, tahun lulus, nomor telepon, alamat, dan bio; unggah foto profil dengan pratinjau gambar saat ini dan opsi untuk menghapusnya; serta "Informasi Karier" untuk detail pekerjaan, perusahaan, jabatan, tautan LinkedIn, dan website pribadi. Fitur unggulan lainnya adalah bagian "Riwayat Pendidikan" yang dinamis, di mana alumni dapat menambah, mengedit, atau menghapus entri pendidikan (jenjang, institusi, jurusan, gelar, tahun masuk/lulus) secara fleksibel. Formulir ini menampilkan pesan kesalahan validasi yang jelas dan diakhiri dengan tombol "Simpan Perubahan" dan "Batal" untuk mengelola data profil.



Gambar 3. 160 Halaman untuk tambah pendidikan bagian alumni



Gambar 3. 161 Potongan kode halaman untuk tambah pendidikan bagian alumni

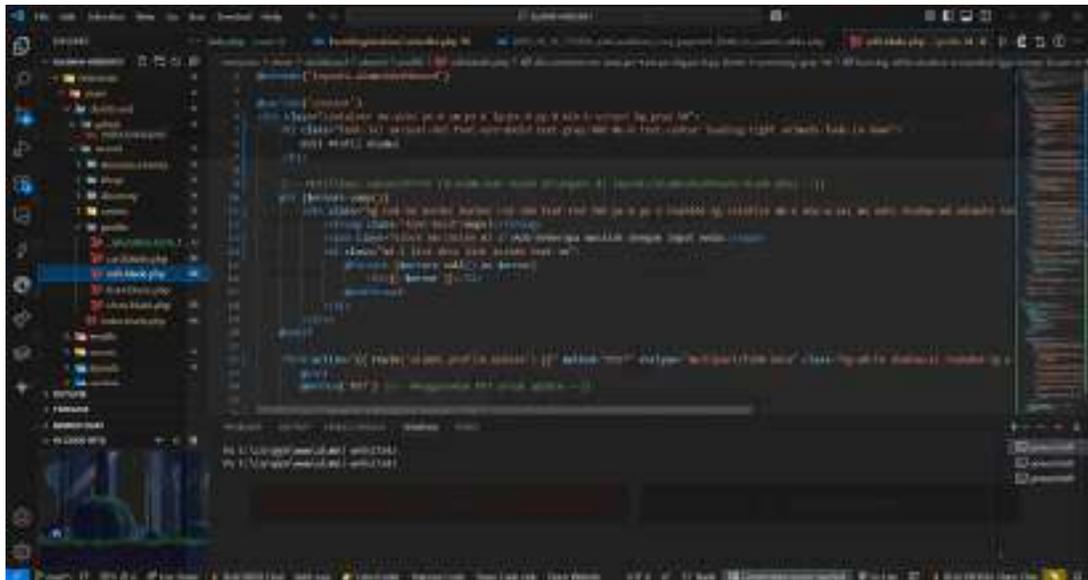
Penjelasan :

Halaman ini adalah komponen parsial (`_education_form_field.blade.php`) yang berfungsi untuk menampilkan dan mengelola satu entri riwayat pendidikan dalam formulir profil alumni yang lebih besar, dengan desain yang bersih dan responsif. Setiap instans

komponen ini mencakup bidang-bidang seperti jenjang pendidikan (S1, S2, S3), nama institusi/universitas, jurusan/program studi, nama gelar, serta tahun masuk dan tahun lulus, yang diisi secara dinamis dengan data yang ada atau nilai old() saat terjadi kesalahan validasi. Setiap entri pendidikan ditandai dengan nomor urut (Pendidikan #) dan dilengkapi dengan tombol "Hapus Pendidikan" yang memungkinkan alumni untuk menghapus entri riwayat pendidikan tersebut dari daftar.



Gambar 3. 162 Halaman cetak kartu alumni



Gambar 3. 163 Potongan kode untuk halaman cetak kartu alumni

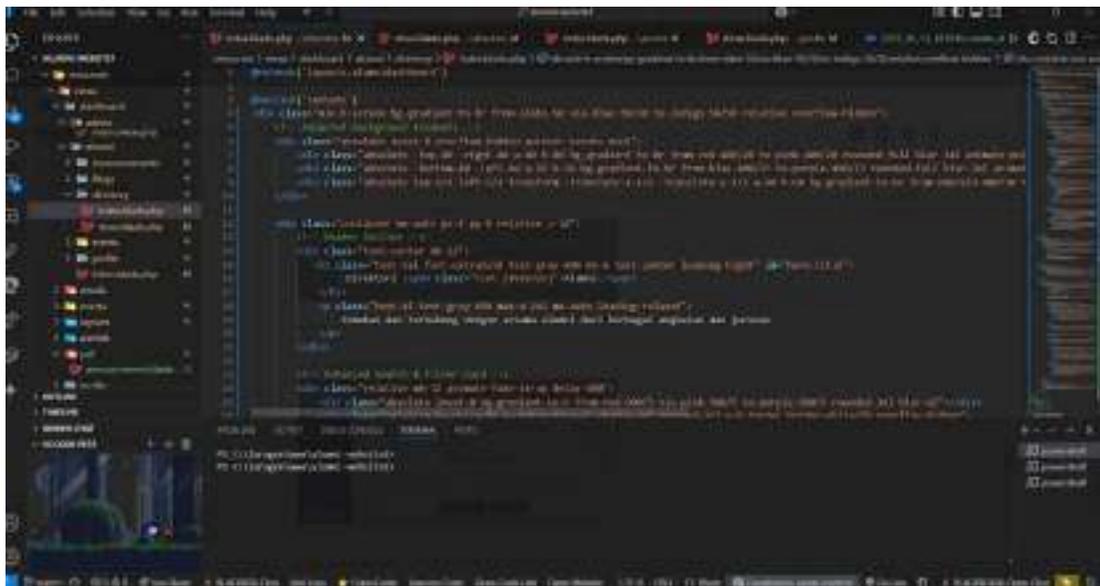
Penjelasan :

Halaman ini berfungsi sebagai tampilan kartu identitas alumni digital yang dapat dicetak, dirancang dengan estetika premium yang kaya akan efek visual modern seperti gradien, animasi shimmering, elemen glassmorphism, dan partikel latar belakang bergerak. Bagian utamanya adalah "Kartu Alumni" yang menampilkan logo "ALUMNI GANISSA", foto profil alumni, nama lengkap, jurusan, angkatan, serta kode alumni unik yang juga direpresentasikan dalam format barcode Code 39 yang dihasilkan secara dinamis. Dua tombol aksi yang jelas, "Cetak Kartu" dan "Tutup", disediakan untuk memudahkan pengguna, dan halaman ini juga mengoptimalkan tata letak serta gaya khusus saat dicetak, memastikan kartu dicetak sesuai ukuran standar identitas.

3) Halaman Direktori Alumni



Gambar 3. 164 Halaman direktori alumni

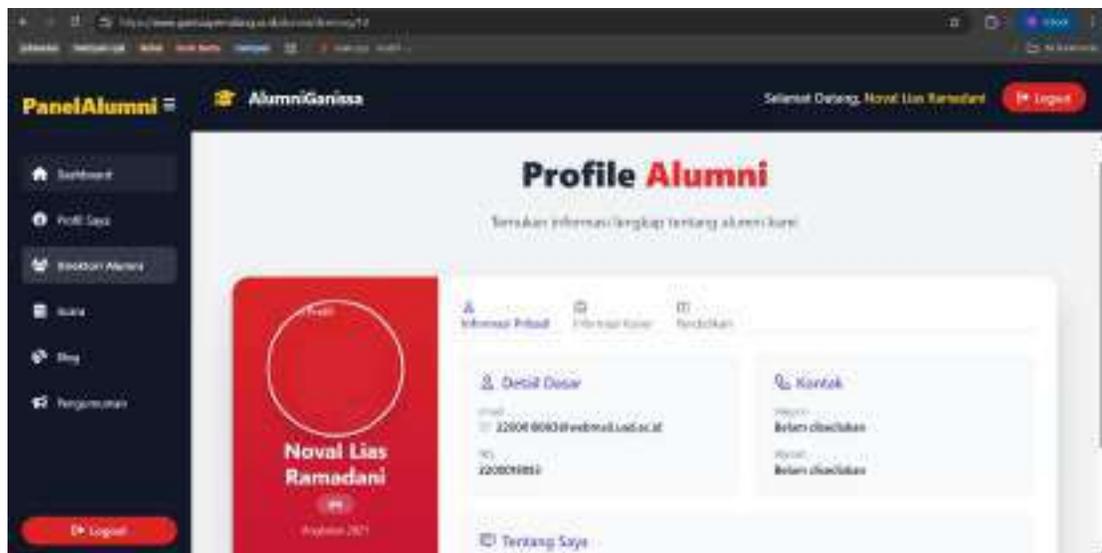


Gambar 3. 165 Potongan kode untuk halaman direktori alumni

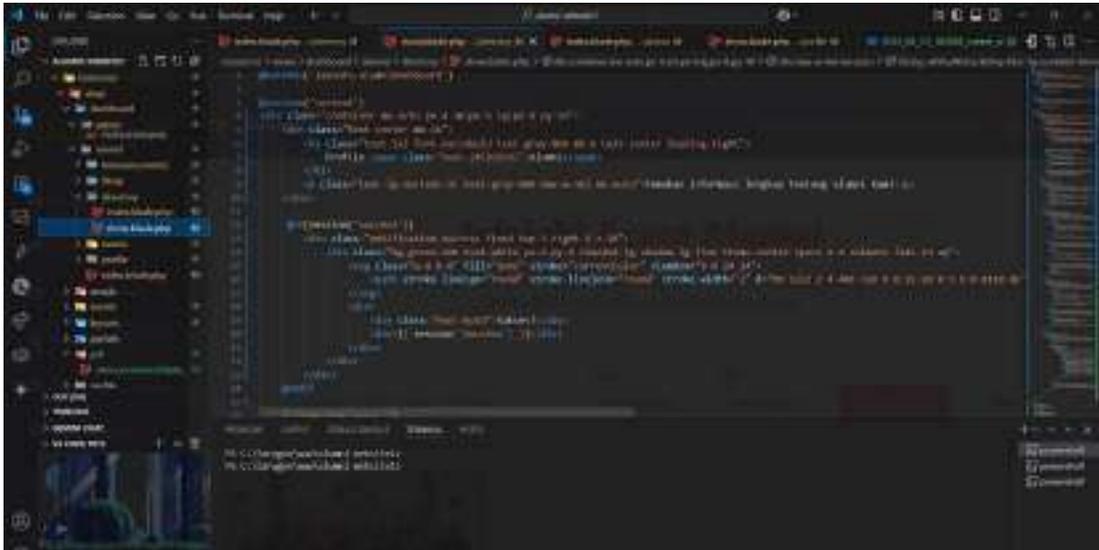
Penjelasan :

Halaman ini adalah Direktori Alumni yang ditujukan untuk alumni, dirancang dengan estetika modern dan interaktif yang menawan, menampilkan latar belakang gradien dengan

elemen animasi lembut dan efek glassmorphism. Di bagian atas, terdapat judul besar "Direktori Alumni" yang mengundang dan pesan singkat yang menjelaskan tujuannya untuk menghubungkan sesama alumni. Bagian terpenting berikutnya adalah panel "Cari & Filter Alumni" yang canggih, memungkinkan pengguna untuk mencari berdasarkan nama, email, bio, atau jurusan, serta memfilter berdasarkan jurusan dan tahun kelulusan dengan dropdown yang stylish. Hasil pencarian ditampilkan dalam tata letak grid yang terdiri dari kartu-kartu alumni yang menarik, masing-masing menampilkan foto profil, nama, jurusan, dan tahun kelulusan, dengan tautan "Lihat Profil" untuk detail lebih lanjut. Seluruh elemen didesain dengan transisi halus dan animasi subtil yang memperkaya pengalaman pengguna, dilengkapi pula dengan paginasi yang estetik untuk navigasi antar halaman hasil.



Gambar 3. 166 Halaman untuk melihat detail profile alumni yang lain



Gambar 3. 167 Potongan kode untuk halaman untuk melihat detail profile alumni yang lain

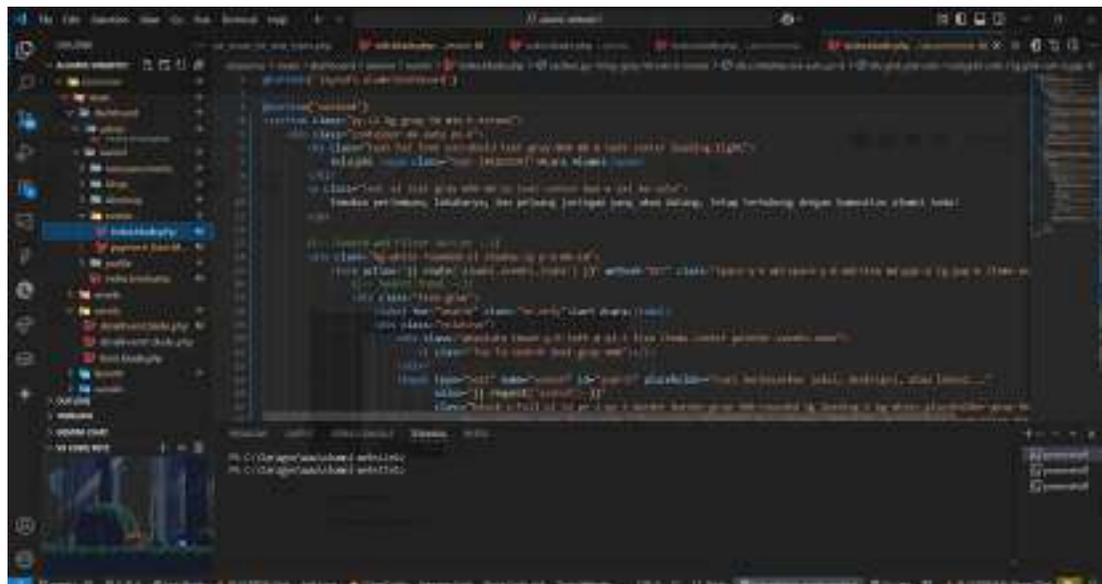
Penjelasan :

Halaman ini menampilkan detail profil alumni dari perspektif direktori, dirancang dengan tata letak modern dan visual yang menarik, termasuk latar belakang dengan gradien dan animasi abstrak yang lembut. Bagian atas halaman menyajikan judul besar "Profile Alumni" dan pesan singkat yang menjelaskan tujuan direktori. Struktur utama halaman dibagi menjadi dua kolom: sidebar kiri menampilkan ringkasan profil alumni dengan foto, nama, jurusan, angkatan, serta tautan ke media sosial dan kode alumni; sementara kolom kanan berisi konten tab yang dinamis. Alumni dapat beralih antara tab "Informasi Pribadi", "Informasi Karier" (termasuk keahlian dalam bentuk skill badge), dan "Pendidikan" (menampilkan riwayat pendidikan dalam format timeline), untuk melihat detail spesifik tentang alumni yang dipilih. Halaman ini juga dilengkapi dengan notifikasi toast untuk umpan balik operasi.

4) Halaman Acara



Gambar 3. 168 Halaman acara alumni



Gambar 3. 169 Potongan kode untuk halaman acara alumni

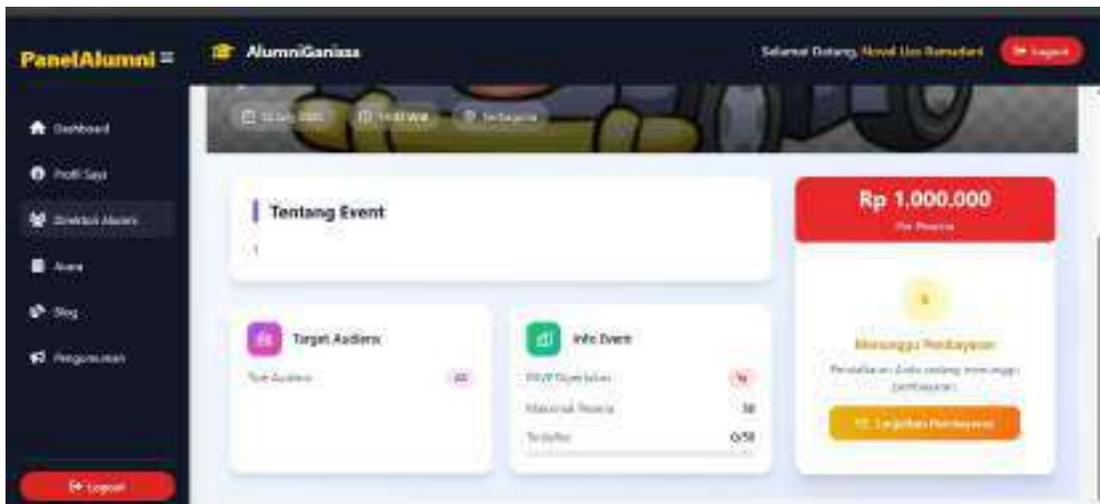
Penjelasan :

Halaman ini adalah direktori "Jelajahi Acara Alumni", yang dirancang untuk memungkinkan alumni menemukan dan berpartisipasi dalam berbagai pertemuan dan

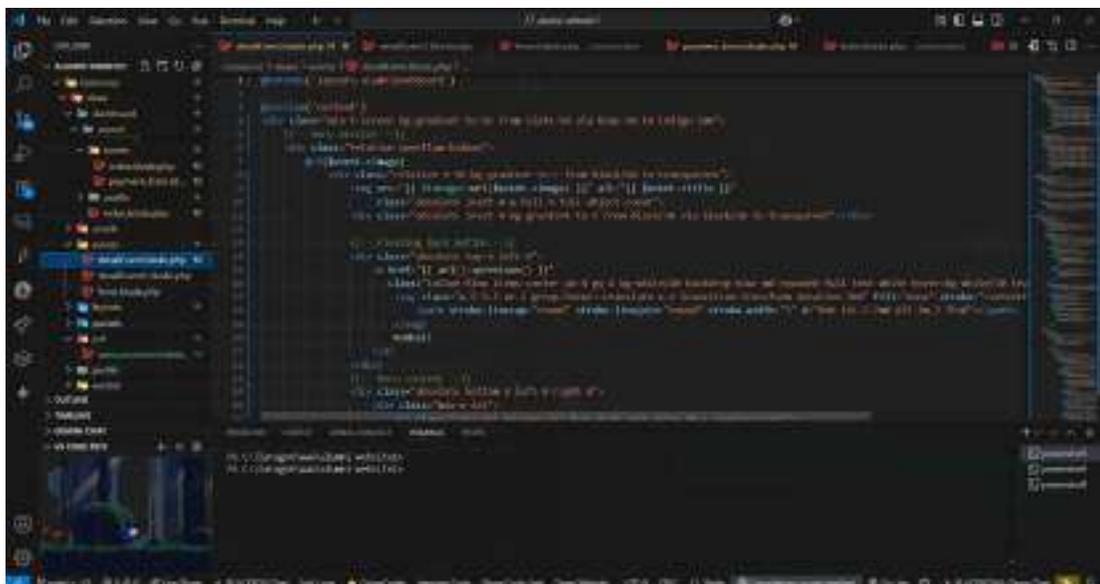
peluang jaringan. Bagian atas halaman menyajikan judul besar yang menarik dan deskripsi yang mengundang, diikuti oleh panel pencarian dan filter yang memungkinkan pengguna untuk mencari acara berdasarkan kata kunci dan menyaringnya berdasarkan status (mendatang, lewat, atau semua) serta tipe (berbayar atau gratis). Jika tidak ada acara yang ditemukan sesuai kriteria, pesan informatif akan ditampilkan. Sebaliknya, acara-acara yang tersedia disajikan dalam tata letak kartu yang rapi, setiap kartu menampilkan gambar acara (atau placeholder), judul, ringkasan deskripsi, tanggal, waktu, lokasi, status RSVP, serta indikasi apakah acara tersebut berbayar atau gratis, dengan tombol "Lihat Detail" untuk informasi lebih lanjut dan fitur paginasi untuk navigasi yang mudah.



Gambar 3. 170 Halaman detail acara di bagian alumni



Gambar 3. 171 Halaman detail acara di bagian alumni 2



Gambar 3. 172 Potongan kode untuk halaman detail acara di bagian alumni

Penjelasan :

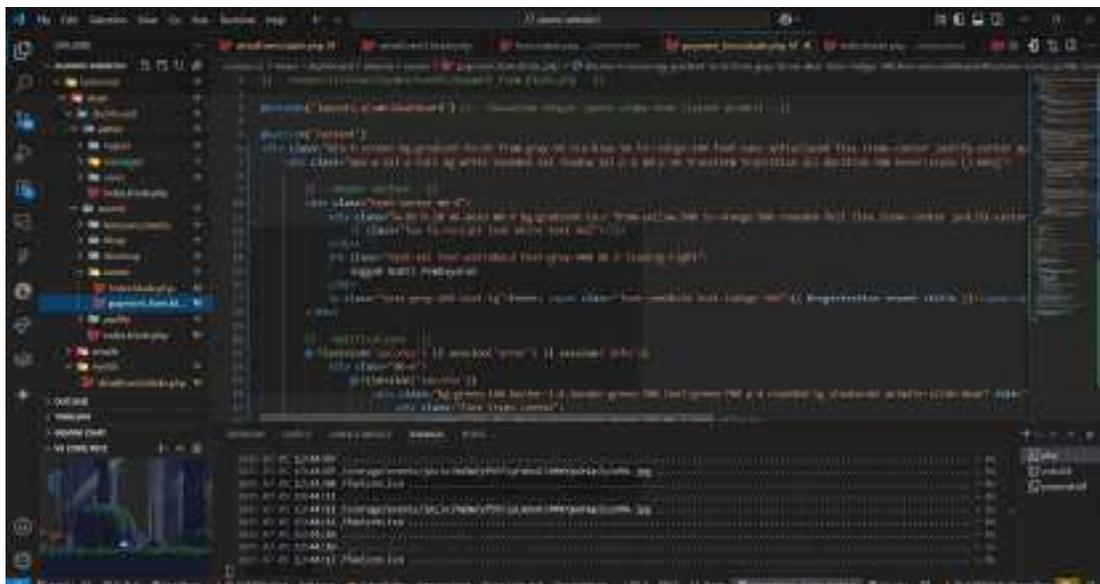
Halaman ini menampilkan detail lengkap suatu event untuk alumni, dimulai dengan bagian hero yang menarik. Bagian hero ini secara dinamis menampilkan gambar event sebagai

latar belakang (atau gradien berwarna jika tidak ada gambar), lengkap dengan judul event yang menonjol, tanggal, waktu, dan lokasi. Terdapat pula tombol "Kembali" yang melayang untuk navigasi yang mudah. Di bawah bagian hero, konten utama terbagi menjadi dua kolom: kolom kiri merinci "Tentang Event" dengan deskripsi lengkapnya, serta panel "Target Audiens" yang menunjukkan jenis audiens yang dituju (misalnya semua alumni, jurusan tertentu, atau angkatan tertentu) dan "Info Event" yang menampilkan status RSVP yang diperlukan, batas maksimal peserta, dan progres pendaftaran yang sudah dikonfirmasi dalam bentuk bilah kemajuan.

Kolom kanan halaman berfungsi sebagai "Kartu Pendaftaran", yang secara jelas menampilkan apakah event berbayar atau gratis. Bagian ini sangat dinamis, menyesuaikan tampilannya dan tombol aksi berdasarkan berbagai kondisi: status login pengguna, peran pengguna (alumni atau lainnya), kelayakan pengguna berdasarkan kriteria audiens event (jurusan/angkatan), ketersediaan slot (apakah kuota sudah penuh), dan status pendaftaran pengguna jika mereka sudah mendaftar (seperti 'menunggu pembayaran', 'menunggu konfirmasi', 'terdaftar', atau 'ditolak/dibatalkan'). Berbagai pesan dan tombol seperti "Daftar Sekarang", "Lanjutkan Pembayaran", atau pesan "Kuota Penuh" akan ditampilkan secara relevan, memandu alumni melalui proses pendaftaran.



Gambar 3. 173 Halaman pembayaran ketika suatu event sifatnya berbayar di bagian alumni



Gambar 3. 174 Potongan kode halaman pembayaran ketika suatu event sifatnya berbayar di bagian alumni

Penjelasan :

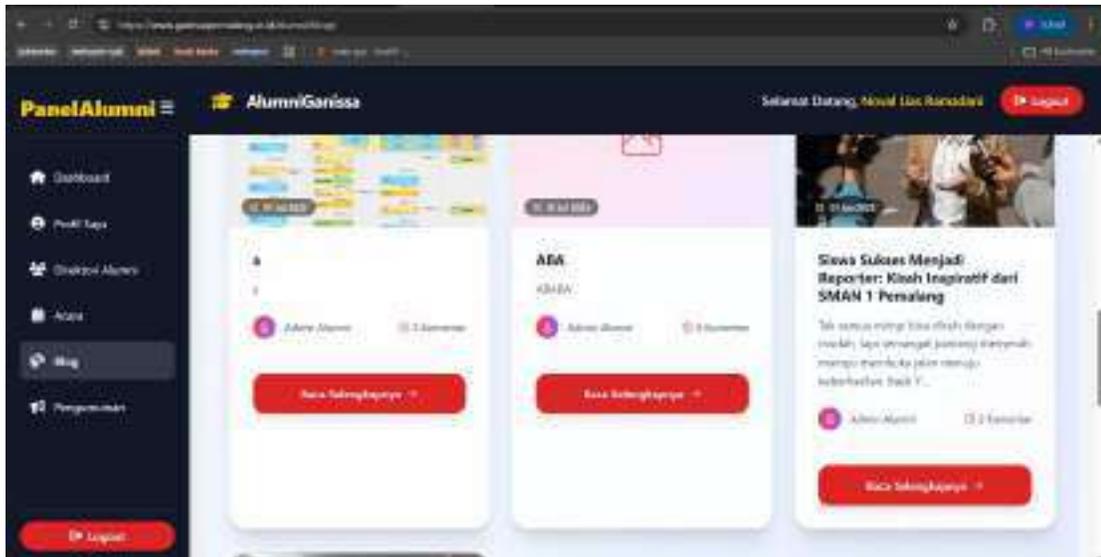
Halaman ini adalah formulir "Unggah Bukti Pembayaran" untuk pendaftaran event berbayar alumni, dirancang dengan tampilan modern dan fokus pada instruksi yang jelas. Di bagian atas, terdapat judul "Unggah Bukti Pembayaran" yang menonjol dan nama event yang

relevan. Alumni akan melihat bagian "Payment Details" yang menunjukkan harga event dan informasi rekening bank tujuan transfer, lengkap dengan tombol "Copy" untuk memudahkan penyalinan nomor rekening. Formulir utama memungkinkan pengguna untuk mengunggah file bukti pembayaran melalui area drag-and-drop atau klik, dengan pratinjau gambar yang dipilih secara instan dan opsi untuk menghapusnya. Terdapat pula pesan sukses/error/info yang dinamis, serta tombol "Unggah Bukti & Selesaikan" untuk mengirimkan bukti dan "Kembali ke Detail Event" untuk navigasi.

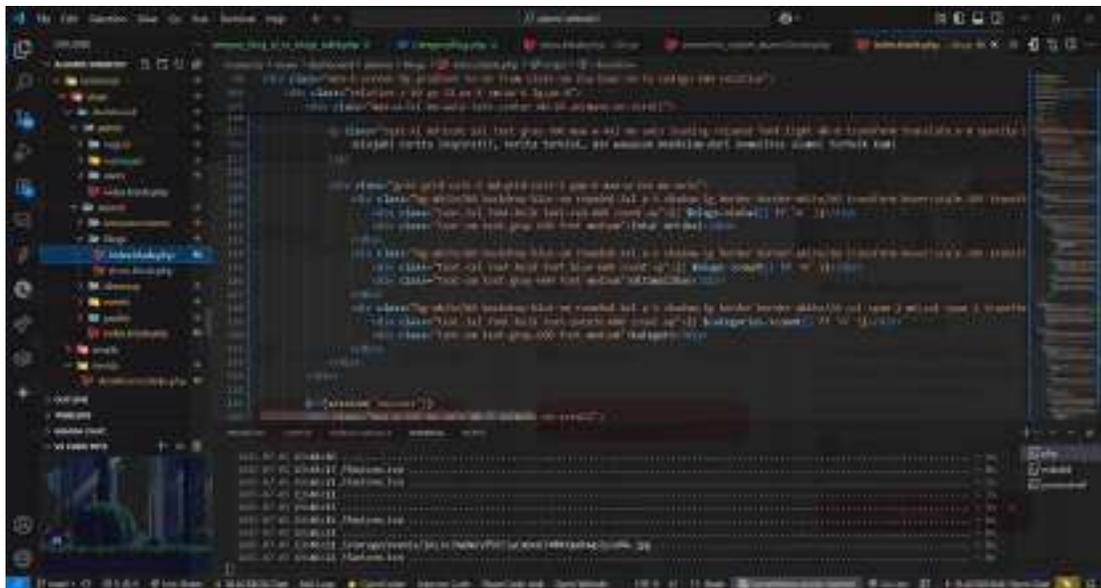
5) Halaman Blog



Gambar 3. 175 Halaman blog alumni



Gambar 3. 176 Halaman blog alumni 2



Gambar 3. 177 Potongan kode halaman blog alumni

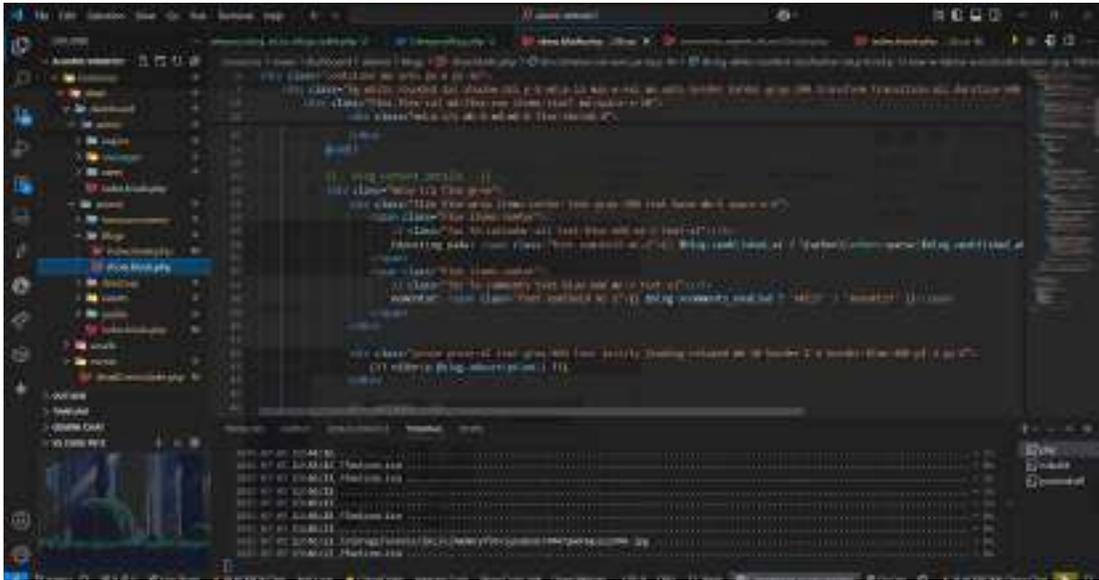
Penjelasan :

Halaman ini adalah direktori "Blog Alumni", dirancang dengan estetika modern dan interaktif yang kaya visual, menampilkan latar belakang dengan gradien lembut dan animasi

abstrak berbentuk *blob* yang bergerak. Bagian atas halaman menyajikan judul besar "Blog Alumni" dan deskripsi yang mengundang untuk menjelajahi cerita inspiratif, diikuti oleh area statistik yang menampilkan total artikel, jumlah artikel yang ditampilkan, dan jumlah kategori, lengkap dengan efek penghitung angka dinamis. Di bawahnya, terdapat panel pencarian dan filter yang canggih, memungkinkan pengguna untuk mencari artikel berdasarkan kata kunci atau memfilter berdasarkan kategori, serta mengurutkan berdasarkan kriteria seperti "Terbaru", "Terlama", atau "Populer". Jika tidak ada artikel yang ditemukan sesuai kriteria, pesan kosong yang *stylish* akan ditampilkan. Sebaliknya, artikel-artikel disajikan dalam tata letak kartu grid yang responsif, menampilkan gambar (dengan efek *skeleton loading*), kategori, tanggal publikasi, judul, cuplikan deskripsi, nama penulis, dan jumlah komentar, dengan tombol "Baca Selengkapnya" untuk melihat detail penuh dan navigasi paginasi di bagian bawah.



Gambar 3. 178 Halaman detail blog di bagian alumni



Gambar 3. 179 Potongan kode halaman detail blog di bagian alumni

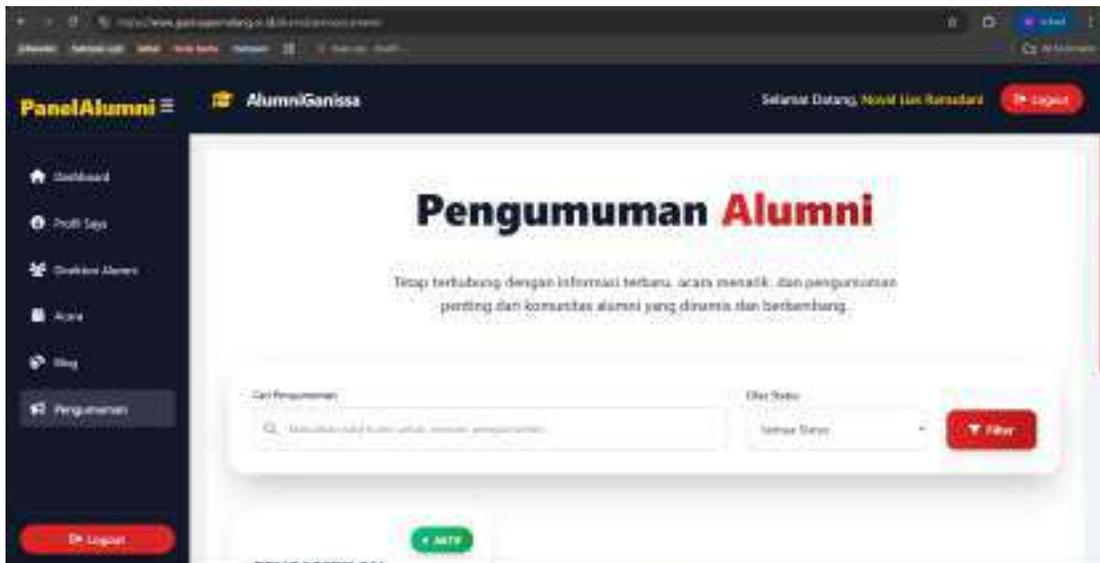
Penjelasan :

Halaman ini menampilkan detail lengkap suatu postingan blog dari perspektif alumni, dengan desain modern dan responsif yang menyoroti konten utama. Bagian paling atas halaman berfungsi sebagai area hero, yang menampilkan gambar blog sebagai latar belakang (atau gradien jika tidak ada) dengan judul blog yang besar, tanggal publikasi, waktu, dan lokasi, serta tombol "Kembali" yang menonjol untuk navigasi. Konten utama blog disajikan dalam format yang mudah dibaca dengan styling yang minimalis. Di bawah konten, jika komentar diaktifkan untuk blog tersebut, akan muncul bagian komentar yang memungkinkan alumni untuk menulis dan mengirim komentar mereka. Selain itu, semua komentar yang sudah ada ditampilkan dalam daftar, lengkap dengan avatar pengguna, nama, waktu komentar dibuat, dan isi komentar, dengan opsi untuk membalas komentar atau menghapus komentar mereka sendiri, yang dikonfirmasi melalui SweetAlert2 untuk tindakan penghapusan.

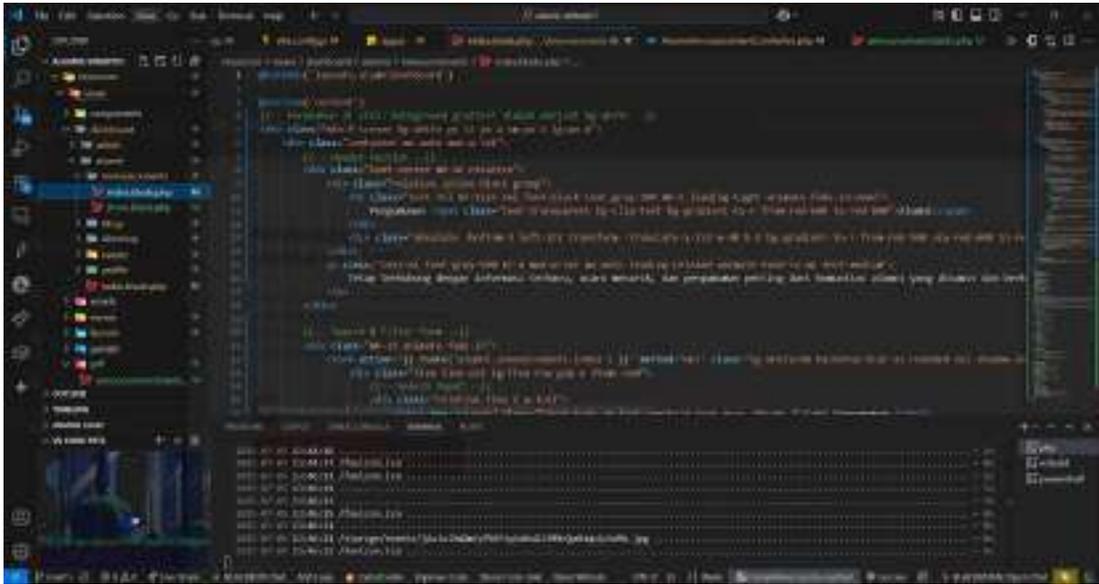
Penjelasan :

Halaman ini adalah komponen parsial (`comments_replies_alumni.blade.php`) yang dirancang untuk menampilkan balasan komentar secara bertingkat dalam konteks dashboard alumni. Setiap balasan ditampilkan dalam kartu yang diindentasi, menyertakan avatar pengguna, nama, dan stempel waktu, diikuti oleh isi balasan. Alumni dapat membalas balasan yang ada melalui formulir yang dapat dibuka/ditutup secara dinamis, atau menghapus balasan mereka sendiri dengan konfirmasi. Komponen ini juga mendukung tampilan balasan yang lebih dalam secara rekursif, menciptakan struktur diskusi yang terorganisir dan mudah diikuti.

6) Halaman Pengumuman



Gambar 3. 182 Halaman pengumuman alumni



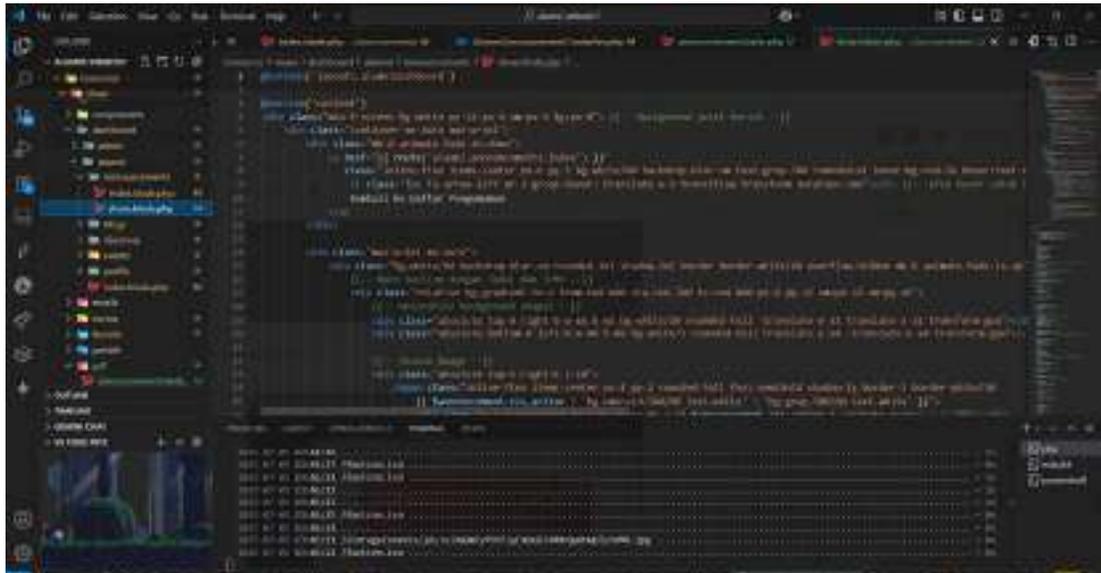
Gambar 3. 183 Potongan kode halaman pengumuman alumni

Penjelasan :

Halaman ini adalah direktori "Pengumuman Alumni", dirancang dengan estetika modern dan interaktif yang kaya animasi visual, menampilkan latar belakang putih bersih yang menonjolkan konten utama. Bagian atas halaman menyajikan judul besar "Pengumuman Alumni" yang menarik dengan efek gradien merah, disertai pesan yang mengundang alumni untuk tetap terhubung dengan informasi terbaru. Di bawahnya, terdapat formulir pencarian dan filter yang canggih dengan efek backdrop blur yang elegan, memungkinkan pengguna untuk mencari pengumuman berdasarkan kata kunci dan menyaringnya berdasarkan status (aktif atau diarsipkan), dengan tombol "Filter" dan "Reset". Pengumuman yang tersedia disajikan dalam tata letak kartu grid yang responsif dan menarik, setiap kartu menampilkan judul, ringkasan deskripsi, tanggal publikasi, dan status aktif/arsip dalam badge yang menonjol, serta tombol "Baca Selengkapnya" untuk detail penuh, dilengkapi dengan navigasi paginasi di bagian bawah.



Gambar 3. 184 Halaman detail pengumuman di bagian alumni



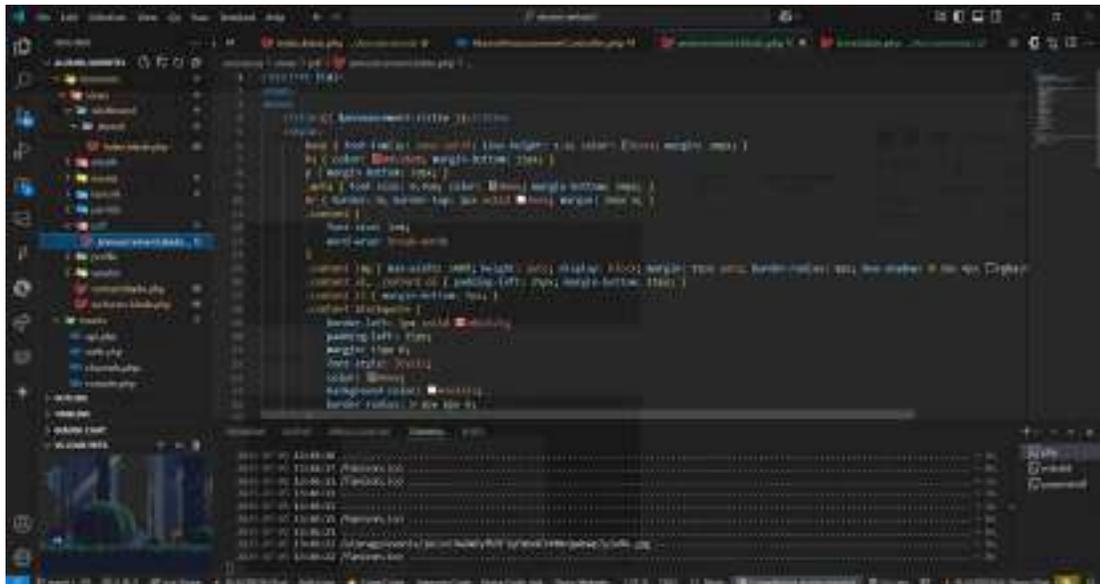
Gambar 3. 185 Potongan kode halaman detail pengumuman di bagian alumni

Penjelasan :

Halaman ini adalah tampilan "Detail Pengumuman" dari perspektif alumni, menampilkan informasi lengkap tentang suatu pengumuman spesifik dengan desain yang bersih, modern, dan visual yang menarik. Bagian paling atas halaman memiliki tombol "Kembali ke Daftar Pengumuman" yang menonjol untuk navigasi yang mudah. Konten utama disajikan dalam kartu yang elegan, diawali dengan judul pengumuman yang besar dan bagian hero dinamis yang menunjukkan status pengumuman (aktif/diarsipkan), tanggal publikasi, waktu pembaruan terakhir, dan jumlah dilihat. Seluruh deskripsi pengumuman ditampilkan dalam format yang mudah dibaca dengan gaya prose yang terstruktur, diikuti oleh pemisah visual. Di bagian bawah, terdapat bagian "Bagikan Pengumuman Ini" yang menyediakan tombol-tombol media sosial populer dan tombol "Salin Tautan", serta opsi "Unduh sebagai PDF", memudahkan alumni untuk menyebarkan informasi penting ini.



Gambar 3. 186 Tampilan ketika alumni mencoba untuk mendownload pengumuman



Gambar 3. 187 Potongan kode untuk tampilan ketika alumni mencoba untuk mendownload pengumuman

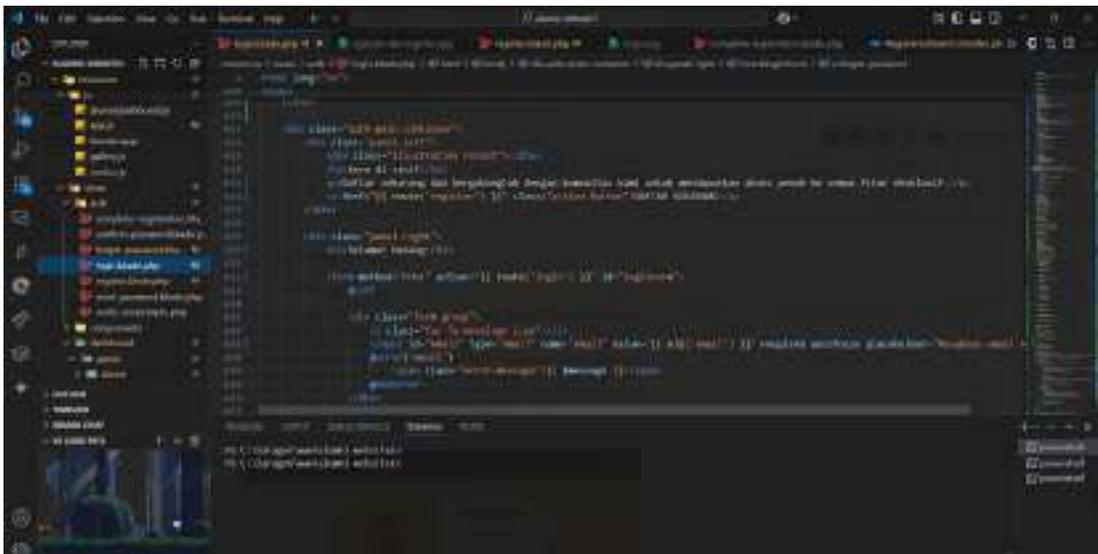
Penjelasan :

Halaman ini adalah tampilan statis dari sebuah pengumuman tunggal, dirancang dengan gaya minimalis dan dasar untuk tujuan pencetakan atau konversi ke format dokumen seperti PDF. Halaman ini menyajikan judul pengumuman, diikuti oleh metadata penting seperti tanggal publikasi dan tanggal terakhir diperbarui. Konten utama pengumuman ditampilkan secara lengkap, dengan styling dasar untuk memastikan keterbacaan yang optimal, termasuk dukungan untuk elemen seperti gambar, daftar, blockquote, dan tabel, menjadikannya ideal untuk distribusi sebagai dokumen formal atau arsip.

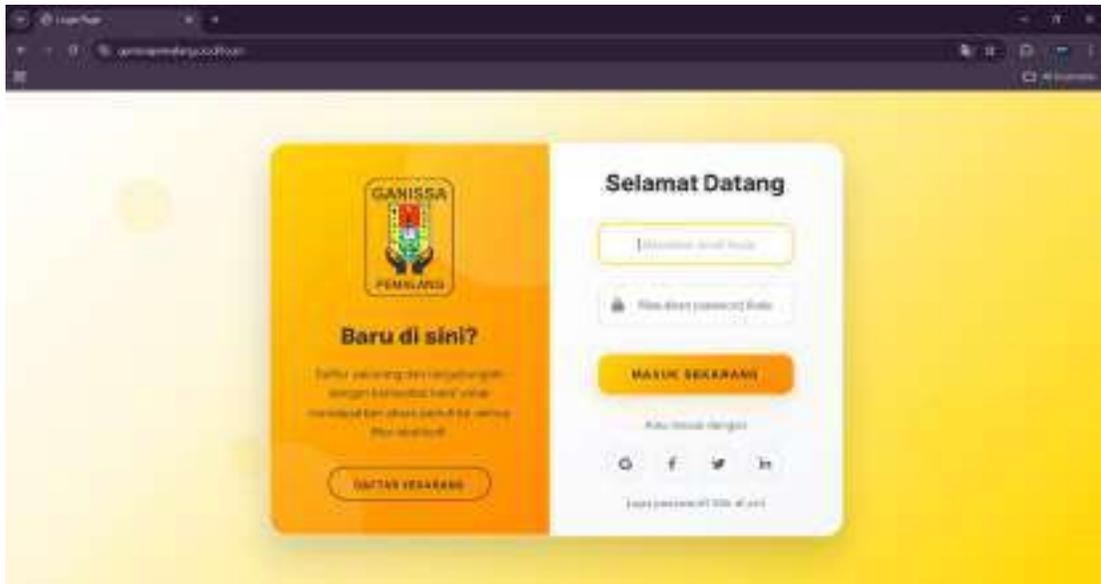
5. Realisasi Frontend 2

1) Halaman Login

Halaman Login dirancang untuk memungkinkan alumni masuk ke sistem dengan aman dan mudah. Mengikuti *wireframe* akan memiliki *field* input untuk *username* atau email dan *password*, serta tombol "Login". Fitur tambahan yang diimplementasikan meliputi opsi "Lupa Kata Sandi" untuk pemulihan akun dan tautan "Daftar Sekarang" yang mengarahkan pengguna ke halaman registrasi. Kegunaan halaman ini adalah sebagai titik akses utama, memastikan hanya pengguna terdaftar yang dapat masuk ke panel alumni.



Gambar 3. 188 Potongan kode halaman login



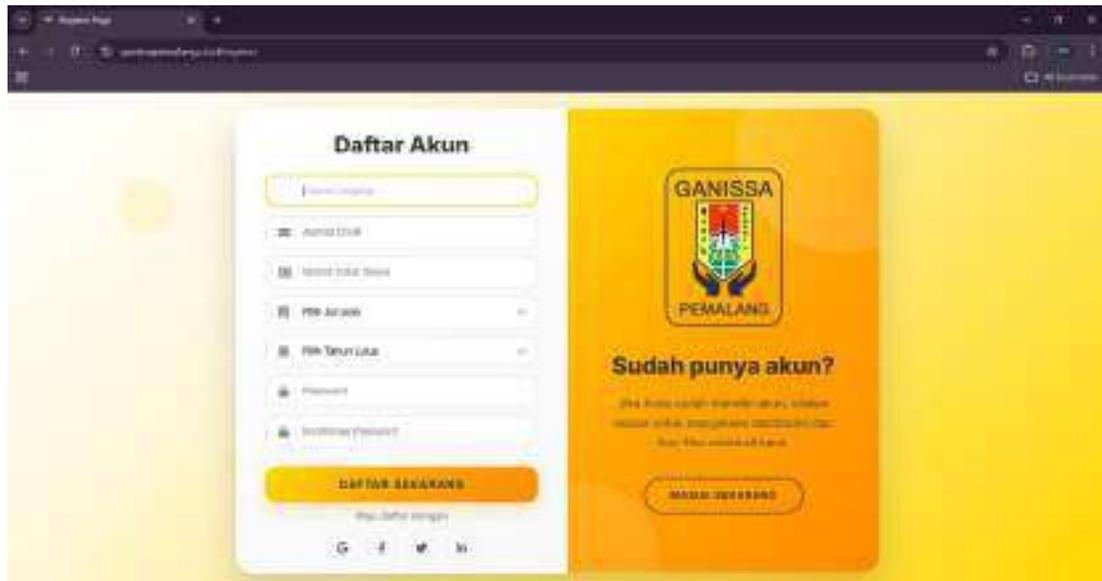
Gambar 3. 189 Tampilan halaman login

2) Halaman Register



Gambar 3. 190 Potongan kode halaman register

Halaman Register dikembangkan untuk memfasilitasi pendaftaran akun baru bagi alumni yang belum terdaftar. Berdasarkan *wireframe* awal desain jadinya akan menampilkan *field-field* yang diperlukan seperti nama lengkap, email, *password*, konfirmasi *password*, dan mungkin data tambahan seperti jurusan atau tahun angkatan. Fitur penting lainnya adalah validasi *input* secara *real-time* untuk memastikan data yang dimasukkan valid dan memenuhi persyaratan. Kegunaan halaman ini adalah untuk memperluas basis pengguna platform, memberikan kesempatan bagi seluruh alumni untuk bergabung dan memanfaatkan fitur-fitur yang tersedia.



Gambar 3. 191 Tampilan halaman register

Tabel "abouts"

The screenshot shows the schema for the 'abouts' table. It has 6 columns: 'id' (BIGINT, PK, AUTO_INCREMENT), 'title' (VARCHAR, 255), 'content' (TEXT), 'image' (VARCHAR, 255), 'created_at' (TIMESTAMP), and 'updated_at' (TIMESTAMP).

#	Name	Database	Length/Set	Unsigned	Allow N.	Zerofill	Default	Comment	Collation	Expression	Virtuality
1	id	BIGINT	20	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	AUTO_INCREMENT				
2	title	VARCHAR	255	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	No default		utf8mb4_unicode_ci		
3	content	TEXT		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	No default		utf8mb4_unicode_ci		
4	image	VARCHAR	255	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL		utf8mb4_unicode_ci		
5	created_at	TIMESTAMP		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL				
6	updated_at	TIMESTAMP		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL				

Tabel "alumni_educations"

The screenshot shows the schema for the 'alumni_educations' table. It has 10 columns: 'id' (BIGINT, PK, AUTO_INCREMENT), 'user_id' (BIGINT, FK), 'degree_level' (ENUM, 'DIPLOMA', 'S1', 'S2', 'S3'), 'institution_name' (VARCHAR, 255), 'major_field' (VARCHAR, 255), 'degree_years' (VARCHAR, 255), 'start_year' (YEAR), 'end_year' (YEAR), 'repeated' (BOOLEAN), and 'updated_at' (TIMESTAMP).

#	Name	Database	Length/Set	Unsigned	Allow N.	Zerofill	Default	Comment	Collation	Expression	Virtuality
1	id	BIGINT	20	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	AUTO_INCREMENT				
2	user_id	BIGINT	20	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	No default		utf8mb4_unicode_ci		
3	degree_level	ENUM	DIPLOMA, S1, S2, S3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL		utf8mb4_unicode_ci		
4	institution_name	VARCHAR	255	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL		utf8mb4_unicode_ci		
5	major_field	VARCHAR	255	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL		utf8mb4_unicode_ci		
6	degree_years	VARCHAR	255	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL		utf8mb4_unicode_ci		
7	start_year	YEAR		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL				
8	end_year	YEAR		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL				
9	repeated	BOOLEAN		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL				
10	updated_at	TIMESTAMP		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL				

Tabel "alumni_profiles"

The screenshot shows the schema for the 'alumni_profiles' table. It has 16 columns: 'id' (BIGINT, PK, AUTO_INCREMENT), 'alumni_code' (VARCHAR, 255), 'user_id' (BIGINT, FK), 'photo' (VARCHAR, 255), 'address' (VARCHAR, 255), 'graduation_year' (VARCHAR, 255), 'bio' (TEXT), 'current_job' (VARCHAR, 255), 'company' (VARCHAR, 255), 'website' (VARCHAR, 255), 'education' (VARCHAR, 255), 'education_start' (VARCHAR, 255), 'education_end' (VARCHAR, 255), 'image' (VARCHAR, 255), 'repeated' (BOOLEAN), and 'updated_at' (TIMESTAMP).

#	Name	Database	Length/Set	Unsigned	Allow N.	Zerofill	Default	Comment	Collation	Expression	Virtuality
1	id	BIGINT	20	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	AUTO_INCREMENT				
2	alumni_code	VARCHAR	255	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL		utf8mb4_unicode_ci		
3	user_id	BIGINT	20	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	No default		utf8mb4_unicode_ci		
4	photo	VARCHAR	255	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL		utf8mb4_unicode_ci		
5	address	VARCHAR	255	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL		utf8mb4_unicode_ci		
6	graduation_year	VARCHAR	255	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL		utf8mb4_unicode_ci		
7	bio	TEXT		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL		utf8mb4_unicode_ci		
8	current_job	VARCHAR	255	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL		utf8mb4_unicode_ci		
9	company	VARCHAR	255	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL		utf8mb4_unicode_ci		
10	website	VARCHAR	255	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL		utf8mb4_unicode_ci		
11	education	VARCHAR	255	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL		utf8mb4_unicode_ci		
12	education_start	VARCHAR	255	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL		utf8mb4_unicode_ci		
13	education_end	VARCHAR	255	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL		utf8mb4_unicode_ci		
14	image	VARCHAR	255	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL		utf8mb4_unicode_ci		
15	repeated	BOOLEAN		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL				
16	updated_at	TIMESTAMP		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL				

Tabel "announcements"

#	Name	Datatype	Length/Set	Unsigned	Allow N.	Zerofill	Default	Comment	Collation	Expression	Virtuality
1	id	BIGINT	28	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	AUTO_INCREMENT				
2	title	VARCHAR	255	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	No default		utf8mb4_unicode_ci		
3	description	TEXT		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	No default		utf8mb4_unicode_ci		
4	published_at	TIMESTAMP		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL				
5	is_active	TINYINT	1	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	1				
6	views	BIGINT	20	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	0				
7	created_at	TIMESTAMP		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL				
8	updated_at	TIMESTAMP		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL				

Tabel "blogs"

#	Name	Datatype	Length/Set	Unsigned	Allow N.	Zerofill	Default	Comment	Collation	Expression	Virtuality
1	id	BIGINT	28	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	AUTO_INCREMENT				
2	image	VARCHAR	255	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL		utf8mb4_unicode_ci		
3	title	VARCHAR	255	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	No default		utf8mb4_unicode_ci		
4	slug	VARCHAR	255	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL		utf8mb4_unicode_ci		
5	published_at	TIMESTAMP		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL				
6	created_at	TINYINT	1	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	1				
7	description	TEXT		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	No default		utf8mb4_unicode_ci		
8	updated_at	TIMESTAMP		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL				
9	updated_at	TIMESTAMP		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL				
10	updated_at	TIMESTAMP	20	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL				

Tabel "category_blogs"

#	Name	Datatype	Length/Set	Unsigned	Allow N.	Zerofill	Default	Comment	Collation	Expression	Virtuality
1	id	BIGINT	28	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	AUTO_INCREMENT				
2	name	VARCHAR	255	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	No default		utf8mb4_unicode_ci		
3	slug	VARCHAR	255	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	No default		utf8mb4_unicode_ci		
4	description	TEXT		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL		utf8mb4_unicode_ci		
5	created_at	TIMESTAMP		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL				
6	updated_at	TIMESTAMP		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL				

Tabel "comments"

#	Name	Datatype	Length/Set	Unsigned	Allow N.	Zerofill	Default	Comment	Collation	Expression	Virtuality
1	id	BIGINT	20	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	AUTO_INCREMENT				
2	user_id	BIGINT	20	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	No default				
3	blog_id	BIGINT	20	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	No default				
4	parent_id	BIGINT	20	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL				
5	content	TEXT		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	No default		utf8mb4_unicode_ci		
6	created_at	TIMESTAMP		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL				
7	updated_at	TIMESTAMP		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL				

Tabel "events"

#	Name	Datatype	Length/Set	Unsigned	Allow N.	Zerofill	Default	Comment	Collation	Charset	Initials
1	id	BIGINT	20	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	AUTO_INCREMENT...				
2	user_id	BIGINT	20	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	No default		utf8mb4_unicode_ci		
3	title	VARCHAR	255	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	No default		utf8mb4_unicode_ci		
4	description	TEXT		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	No default		utf8mb4_unicode_ci		
5	image	VARCHAR	255	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	No default		utf8mb4_unicode_ci		
6	event_date	DATE		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	No default				
7	location	VARCHAR	255	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	No default		utf8mb4_unicode_ci		
8	event_time	TIME		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	No default				
9	req_required	BIGINT	1	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	0				
10	authentication	BIGINT		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	0		utf8mb4_unicode_ci		
11	input_email	BIGINT		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	0				
12	input_pass	CHAR		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	0				
13	max_authentication	TINYINT	10	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	0				
14	is_paid	BIGINT	1	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	0				
15	price	DECIMAL	10,2	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	0				
16	timestamp	TIMESTAMP		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	0				

Tabel "event_registrations"

#	Name	Datatype	Length/Set	Unsigned	Allow N.	Zerofill	Default	Comment	Collation	Charset	Initials
1	id	BIGINT	20	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	AUTO_INCREMENT...				
2	user_id	BIGINT	20	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	No default				
3	event_id	BIGINT	20	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	No default				
4	status	VARCHAR	255	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	pending_registration		utf8mb4_unicode_ci		
5	timestamp_event	VARCHAR	255	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	0		utf8mb4_unicode_ci		
6	timestamp_reg	TIMESTAMP		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	0				
7	timestamp_at	TIMESTAMP		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	0				

Tabel "failed_at"

#	Name	Datatype	Length/Set	Unsigned	Allow N.	Zerofill	Default	Comment	Collation	Charset	Initials
1	id	BIGINT	20	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	AUTO_INCREMENT...				
2	user_id	VARCHAR	255	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	No default		utf8mb4_unicode_ci		
3	connection	TEXT		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	No default		utf8mb4_unicode_ci		
4	ipaddr	TEXT		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	No default		utf8mb4_unicode_ci		
5	password	VARCHAR		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	No default		utf8mb4_unicode_ci		
6	password2	VARCHAR		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	No default		utf8mb4_unicode_ci		
7	failed_at	TIMESTAMP		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	ON NULL DEFAULT...				

Tabel "majors"

#	Name	Datatype	Length/Def	Unsigned	Allow N.	Zeros	Default	Comment	Collation	Expression	Virtuality
1	id	BIGINT	20	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	AUTO INCREMENT				
2	name	VARCHAR	255	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	No default		utf8mb4_unicode_ci		
3	description	TEXT		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	No default		utf8mb4_unicode_ci		
4	score	TINYINT	255	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	No default		utf8mb4_unicode_ci		
5	created_at	TIMESTAMP		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL				
6	updated_at	TIMESTAMP		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL				

Tabel "messages"

#	Name	Datatype	Length/Def	Unsigned	Allow N.	Zeros	Default	Comment	Collation	Expression	Virtuality
1	id	BIGINT	20	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	AUTO INCREMENT				
2	full_name	VARCHAR	255	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	No default		utf8mb4_unicode_ci		
3	email	VARCHAR	255	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	No default		utf8mb4_unicode_ci		
4	subject	VARCHAR	255	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL		utf8mb4_unicode_ci		
5	message_content	TEXT		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	No default		utf8mb4_unicode_ci		
6	is_read	TINYINT	1	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	0				
7	created_at	TIMESTAMP		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL				
8	updated_at	TIMESTAMP		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL				

Tabel "migrations"

#	Name	Datatype	Length/Def	Unsigned	Allow N.	Zeros	Default	Comment	Collation	Expression	Virtuality
1	id	INT	10	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	AUTO INCREMENT				
2	migration	VARCHAR	255	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	No default		utf8mb4_unicode_ci		
3	batch	INT	10	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	No default				

Tabel "password_reset_tokens"

#	Name	Datatype	Length/Def	Unsigned	Allow N.	Zeros	Default	Comment	Collation	Expression	Virtuality
1	email	VARCHAR	255	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	No default		utf8mb4_unicode_ci		
2	token	VARCHAR	255	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	No default		utf8mb4_unicode_ci		
3	created_at	TIMESTAMP		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL				

Tabel "personal_access_tokens"

#	Name	Datatype	Length/Set	Unsigned	Auto IN.	Zero's	Default	Comment	Collation	Expression	Virtuality
1	id	BIGINT	20	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	AUTO_INCREMENT				
2	token_type	VARCHAR	255	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL		utf8mb4_unicode_ci		
3	token_value	BIGINT	20	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL		utf8mb4_unicode_ci		
4	name	VARCHAR	255	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL		utf8mb4_unicode_ci		
5	token	VARCHAR	64	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL		utf8mb4_unicode_ci		
6	expires	TEXT		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL		utf8mb4_unicode_ci		
7	last_used_at	TIMESTAMP		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL				
8	updated_at	TIMESTAMP		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL				
9	created_at	TIMESTAMP		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL				
10	deleted_at	TIMESTAMP		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL				

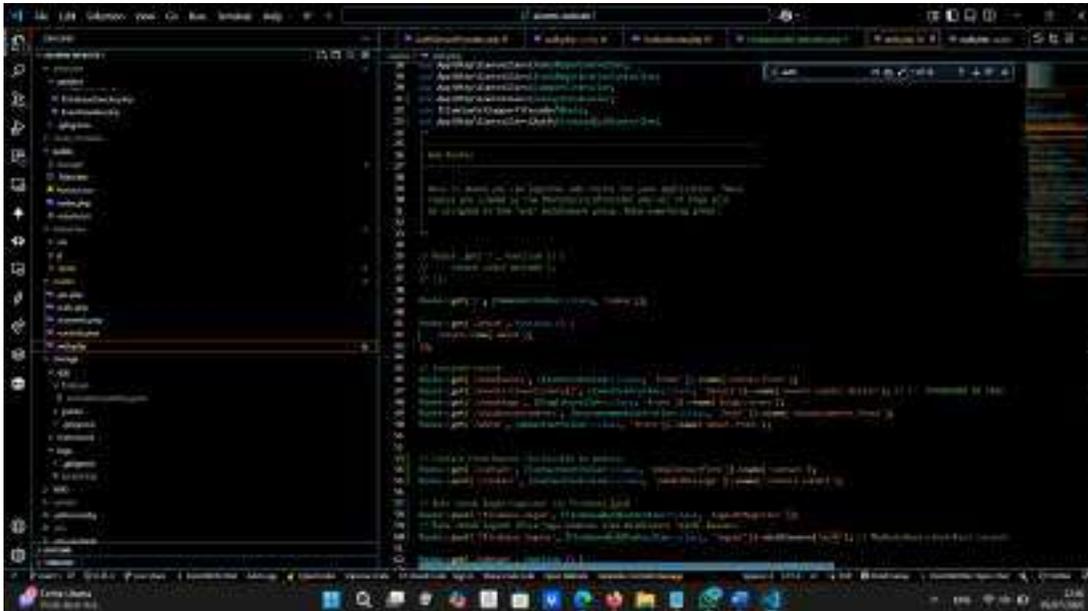
Tabel "users"

#	Name	Datatype	Length/Set	Unsigned	Auto IN.	Zero's	Default	Comment	Collation	Expression	Virtuality
1	id	BIGINT	20	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	AUTO_INCREMENT				
2	name	VARCHAR	255	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL		utf8mb4_unicode_ci		
3	email	VARCHAR	255	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL		utf8mb4_unicode_ci		
4	email_verified_at	TIMESTAMP		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL		utf8mb4_unicode_ci		
5	password	BIGINT	20	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL				
6	remember_token	VARCHAR	100	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL		utf8mb4_unicode_ci		
7	email_verified_at	TIMESTAMP		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL		utf8mb4_unicode_ci		
8	password	VARCHAR	255	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL		utf8mb4_unicode_ci		
9	remember_token	VARCHAR	100	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL		utf8mb4_unicode_ci		
10	remember_token	VARCHAR	100	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL		utf8mb4_unicode_ci		
11	remember_token	TIMESTAMP		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL				
12	updated_at	TIMESTAMP		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL				

Berikut ini realisasinyaa berdasarkan daftar pekerjaan fungsionalitas.

1. F-GUEST-001: Akses Informasi Publik Terbatas

a. Rute-rute Publik (routes/web.php)



Gambar 3. 192 Rute Fitur Akses Informasi Publik Terbatas

Penjelasan: Cuplikan ini menunjukkan definisi rute-rute utama yang dapat diakses oleh publik (Guest) tanpa otentikasi. Rute-rute ini mengarahkan permintaan ke metode index pada HomeController untuk halaman beranda, metode front pada AboutController, EventController, BlogController, dan AnnouncementController untuk menampilkan daftar konten masing-masing. Rute detail event juga tersedia untuk publik. Rute kontak juga terbuka untuk umum.

b. Logika Halaman Beranda (app/Http/Controllers/HomeController.php)

```
1 @test
2
3 namespace App\Http\Controllers;
4 use App\Models\Event;
5 use App\Models\User;
6 use App\Models\Blog;
7 use App\Models\Announcement;
8 use App\Models>About;
9 use Illuminate\Http\Request;
10 use Carbon\Carbon; // --- TAMBAHAN DATE DI
11
12 class HomeController extends Controller
13 {
14     public function index()
15     {
16
17         // PEMERIKSAAN BAGIAN INI
18         // Mengambil 3 event terbaru yang memiliki audience_type 'all'
19         // Ambil 3 event mendatang yang terdapat (event_date lebih besar dari atau sama dengan hari ini)
20         $events = Event::where('event_date', '>=', Carbon::today())
21             ->orderBy('event_date', 'asc') // Urutkan berdasarkan tanggal terdahul
22             ->orderBy('event_time', 'asc') // termasuk berdasarkan waktu
23             ->take(3) // Ambil hanya 3 event
24             ->get(); // Eksekusi query
25
26
27         // Fetch only the latest 3 published blogs for the homepage section.
28         $blogs = Blog::whereNotNull('published_at')
29             ->where('published_at', '<', Carbon::now()) // Ensure it's not a future-dated blog
30             ->latest('published_at') // Order by published_at DESC
31             ->limit(3) // limit to 3 blogs
32             ->get(); // get the results
33
34
35         $announcements = Announcement::latest()->paginate(5);
36
37         // Di dalam metode controller bisa
38         $alumniUsers = User::where('role', 'alumni')->limit(6)->get();
39
40
41         $about = About::first();
42
43         return view('welcome', compact('events', 'alumniUsers', 'blogs', 'announcements', 'about'));
44
45     }
46 }
47
48 }
```

Gambar 3. 193 Logika Halaman Beranda

Penjelasan: Metode index pada HomeController bertanggung jawab mengambil data yang akan ditampilkan di halaman beranda. Ini termasuk 3 event mendatang, 3 blog terbaru yang sudah dipublikasikan, pengumuman, daftar alumni (terbatas), dan konten About Us. Data ini kemudian diteruskan ke view welcome.

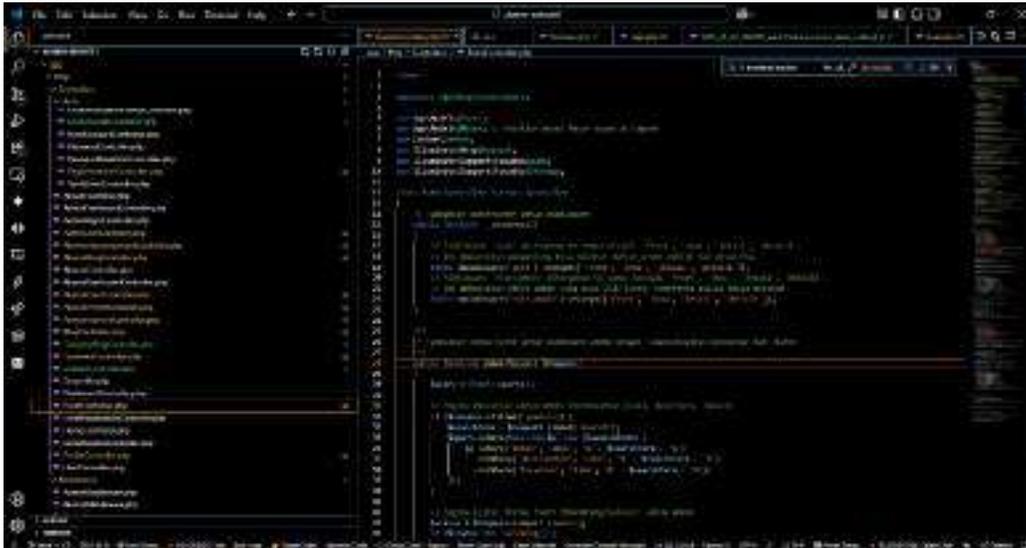
c. Logika Halaman About Us Publik (app/Http/Controllers/AboutController.php)

```
1 <?php
2
3 namespace App\Http\Controllers;
4
5 use Illuminate\Http\Request;
6 use App\Http\Controllers\Controller;
7 use App\Models\About;
8
9 class AboutController extends Controller
10 {
11     public function index()
12     {
13         $about = About::first(); // Get About Us content
14         return view('dashboard.admin.about.index', compact('about'));
15     }
16
17     public function front()
18     {
19         $about = About::first(); // Fetch About Us content
20         return view('dashboard.admin.about.front', compact('about')); // Return the frontend view
21     }
22
23
24     public function edit()
25     {
26         $about = About::first();
27         return view('dashboard.admin.about.edit', compact('about'));
28     }
29
30     public function update(Request $request)
31     {
32         $request->validate([
33             'title' => 'required|string|max:255',
34             'content' => 'required|string',
35             'image' => 'nullable|image|max:2048',
36         ]);
37
38         $about = About::first(); // use about();
39         $about->title = $request->title;
40         $about->content = $request->content;
41
42         if ($request->hasFile('image')) {
43             $about->image = $request->file('image')->store('about', 'public');
44         }
45
46         $about->save();
47
48         return redirect()->route('about.index')->with('success', 'About Us updated successfully');
49     }
50 }
51
```

Gambar 3. 194 Logika Halaman About Us Publik

Penjelasan: Metode front di AboutController mengambil satu-satunya record About dari database (asumsi konten About Us hanya ada satu) dan meneruskannya ke view dashboard.admin.about.front untuk ditampilkan di sisi publik.

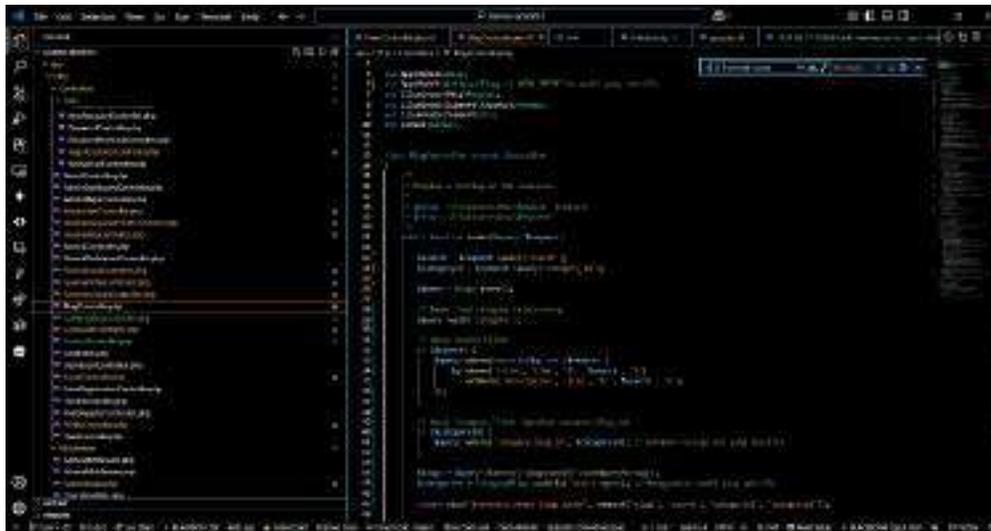
d. Logika Daftar Event Publik (app/Http/Controllers/EventController.php)



Gambar 3. 195 Logika Datar Event Publik

Penjelasan: Metode front pada EventController mengambil daftar event yang audience_type-nya adalah 'all', artinya dapat diakses publik. Metode ini juga mendukung fungsionalitas pencarian dan filter berdasarkan status (mendatang/selesai). Metode detail digunakan untuk menampilkan detail event secara individual kepada publik.

e. Logika Daftar Blog Publik (app/Http/Controllers/BlogController.php)

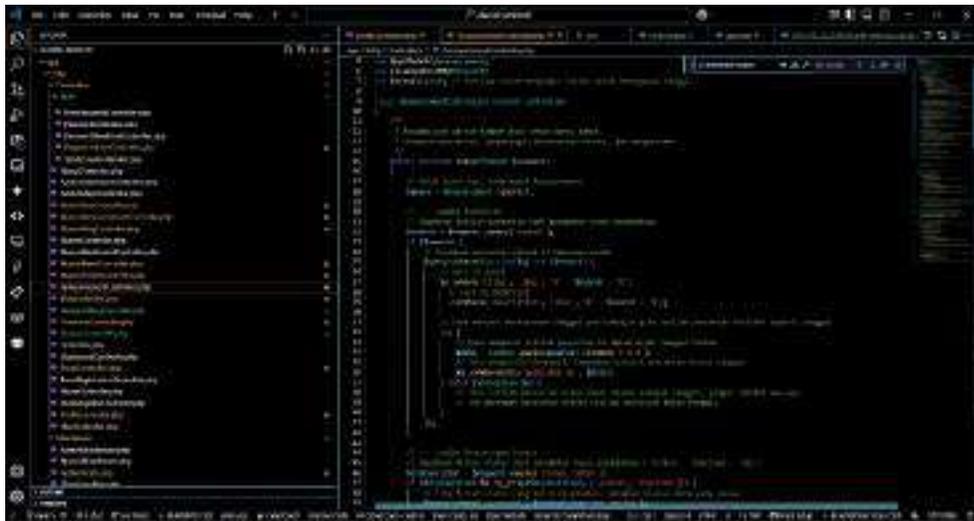


Gambar 3. 196 Logika Daftar Blog Publik

Penjelasan: Metode front di BlogController mengambil daftar blog yang sudah dipublikasikan (published_at tidak null dan tanggal publikasi sudah lewat atau hari ini). Fungsionalitas pencarian dan filter berdasarkan kategori juga disediakan. Hasilnya di-paginate dan dikirim ke view dashboard.admin.blogs.front.

f. Logika Daftar Pengumuman Publik

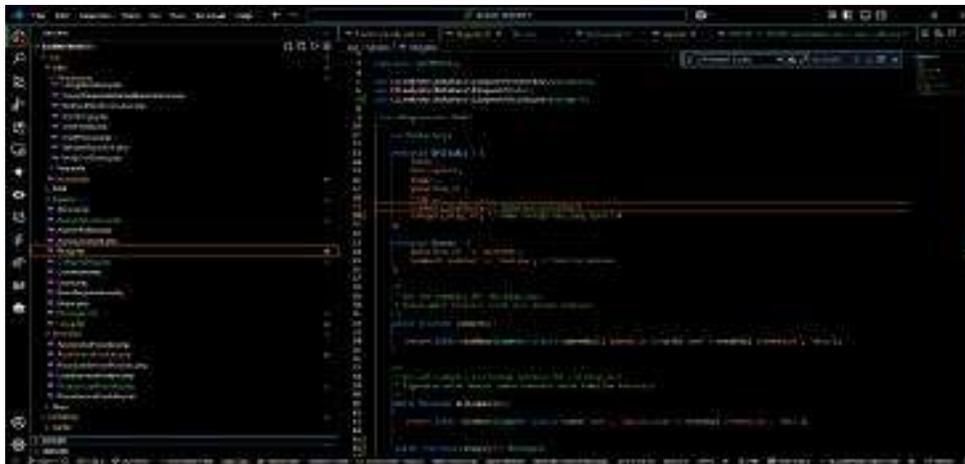
(app/Http/Controllers/AnnouncementController.php)



Gambar 3. 197 Logika Daftar Pengumuman Publik

Penjelasan: Metode front pada AnnouncementController mengambil daftar pengumuman yang is_active (aktif) dan mendukung pencarian berdasarkan judul. Hasilnya di-paginate dan dikirim ke view dashboard.admin.announcements.front untuk tampilan publik.

g. Model Terkait (disini: app/Models/Blog.php)

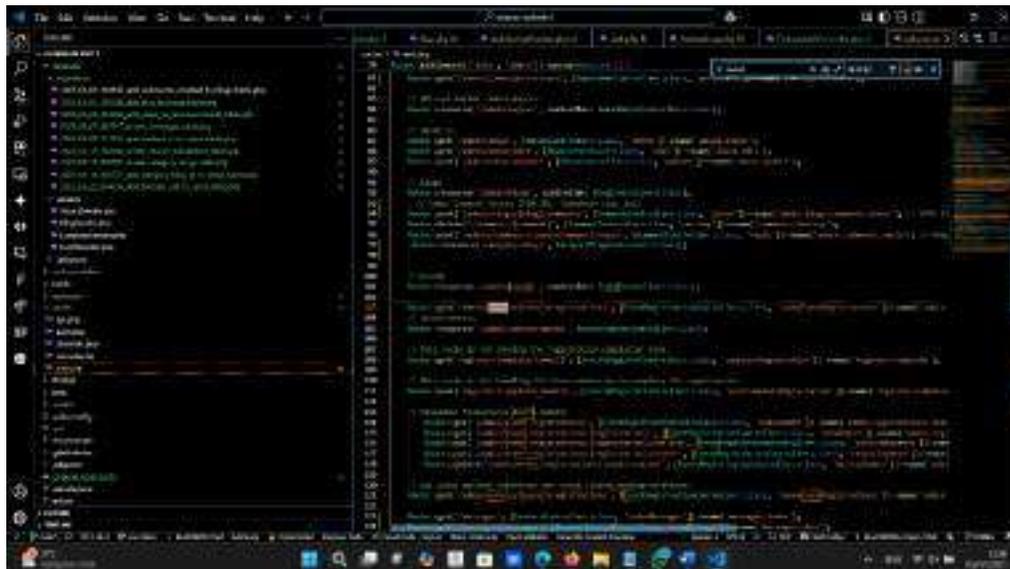


Gambar 3. 198 Model Terkait

Penjelasan: Model Blog menunjukkan atribut seperti title, description, image, published_at, slug, comments_enabled, dan category_blog_id. published_at dan comments_enabled di-cast sebagai tipe data yang sesuai untuk memudahkan penanganan tanggal dan boolean. Terdapat juga relasi belongsTo ke CategoryBlog untuk mengasosiasikan blog dengan kategorinya. Model Event, Announcement, dan About memiliki struktur serupa yang mendukung data yang ditampilkan di halaman publik.

2. F-GUEST-002: Penampil Jadwal & Detail Event Publik

a. Rute-rute Event Publik (routes/web.php)

A screenshot of a code editor showing PHP routes for public events. The editor has a dark theme. On the left, there is a file explorer showing a directory structure with folders like 'resources', 'routes', and 'views'. The main area displays PHP code for routes. The code includes comments in Indonesian and defines two routes: one for showing all public events and another for showing the details of a specific event. The routes are defined using the Laravel Router facade.

```
use Illuminate\Http\Request;
use App\Http\Controllers\EventController;

Route::get('/events', function () {
    // Menampilkan daftar semua event publik
    // menggunakan metode front di EventController
    return EventController::front();
});

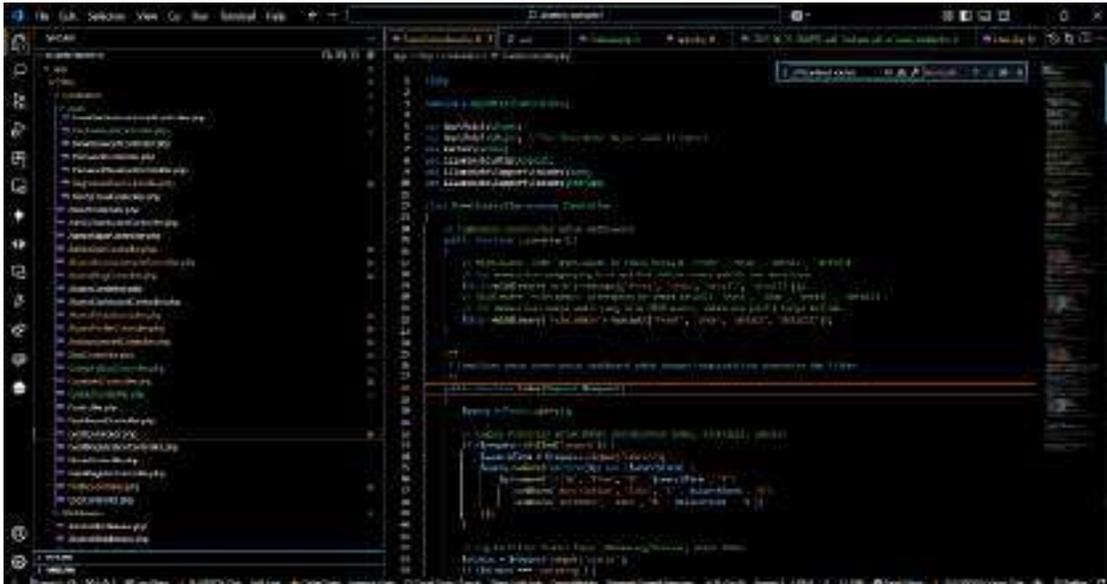
Route::get('/events/{event}/detail', function () {
    // Menampilkan detail spesifik
    // dari sebuah event melalui metode detail
    // di EventController. Parameter {event}
    // pada rute ini akan diisi dengan ID event
    // yang akan ditampilkan.
    return EventController::detail();
});
```

Gambar 3. 199 Rute Event Publik

Penjelasan: Cuplikan ini menunjukkan dua rute utama yang memungkinkan tamu mengakses informasi event. Rute /showEvents menampilkan daftar semua event publik melalui metode front di EventController, sementara rute /events/{event}/detail menampilkan detail spesifik dari sebuah event melalui metode detail di EventController. Parameter {event} pada rute

detail event akan secara otomatis diresolusi menjadi objek Event oleh Laravel (Route Model Binding).

b. Logika Daftar Event Publik (app/Http/Controllers/EventController.php)



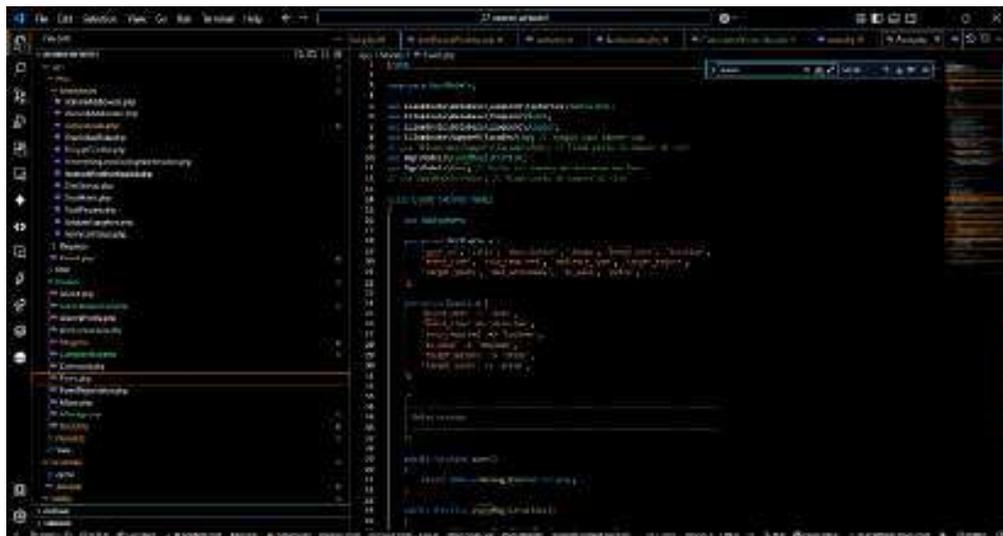
Gambar 3. 200 Logika Daftar Event Publik

Penjelasan:

- Metode front di EventController bertanggung jawab untuk mengambil daftar event yang akan ditampilkan kepada publik. Event difilter agar hanya menampilkan yang memiliki audience_type 'all'.
- Fungsionalitas pencarian berdasarkan judul, deskripsi, atau lokasi juga diimplementasikan.
- Event dapat difilter berdasarkan status 'upcoming' (mendatang) atau 'past' (telah berlalu), dengan default menampilkan event yang akan datang. Hasilnya diurutkan berdasarkan tanggal dan waktu event, kemudian di-paginate untuk memudahkan navigasi.

- Metode detail dan detail1 (jika digunakan sebagai alternatif) mengambil objek Event melalui Route Model Binding dan meneruskannya ke view events.detailEvent atau events.detailEvent1 untuk menampilkan informasi lengkap event.

c. Model Terkait (app/Models/Event.php)



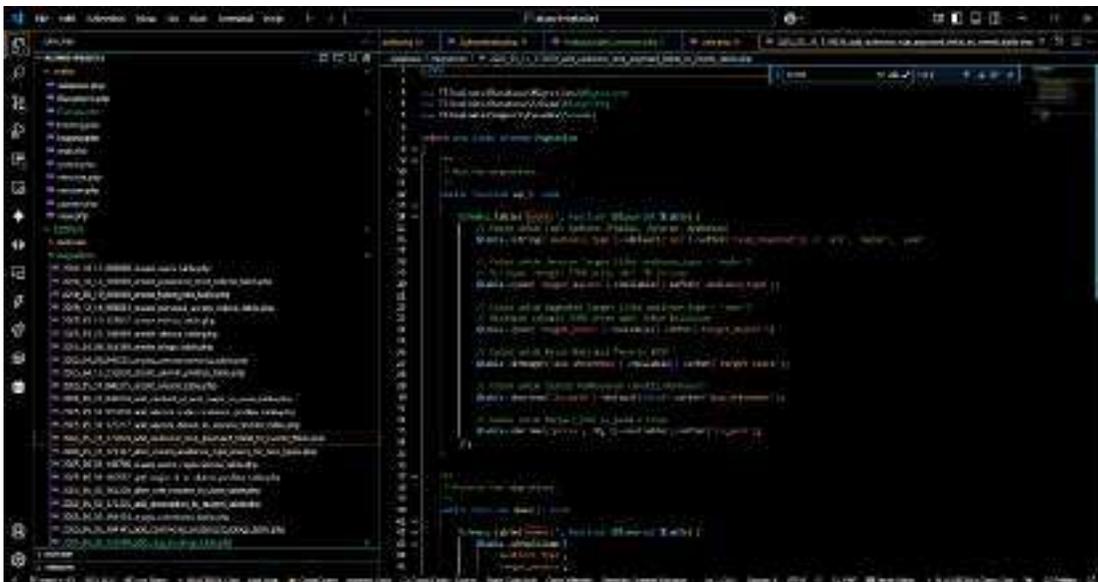
Gambar 3. 201 Model Event

Penjelasan: Model Event mendefinisikan struktur data untuk event, termasuk kolom seperti title, description, event_date, location, audience_type, is_paid, dan price. Atribut event_date dan event_time di-cast sebagai tipe data yang sesuai. Kolom audience_type sangat penting untuk fungsionalitas ini, karena menentukan apakah event dapat diakses oleh publik (all), atau hanya untuk audiens spesifik (misal: major_only, year_only, major_and_year). Metode isEligibleForAlumni juga menunjukkan logika di mana event dengan audience_type 'all' selalu memenuhi syarat untuk diakses publik.

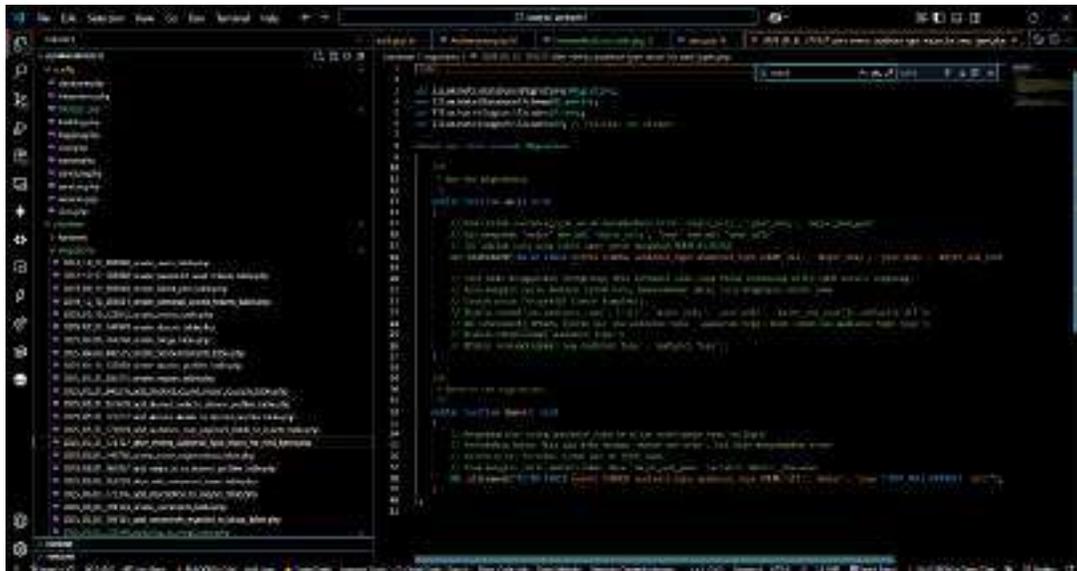
d. Migrasi Database Terkait (database/migrations/...)



Gambar 3. 202 Migrasi Create Event Table



Gambar 3. 203 Migrasi Penambahan Payment di Tabel Event



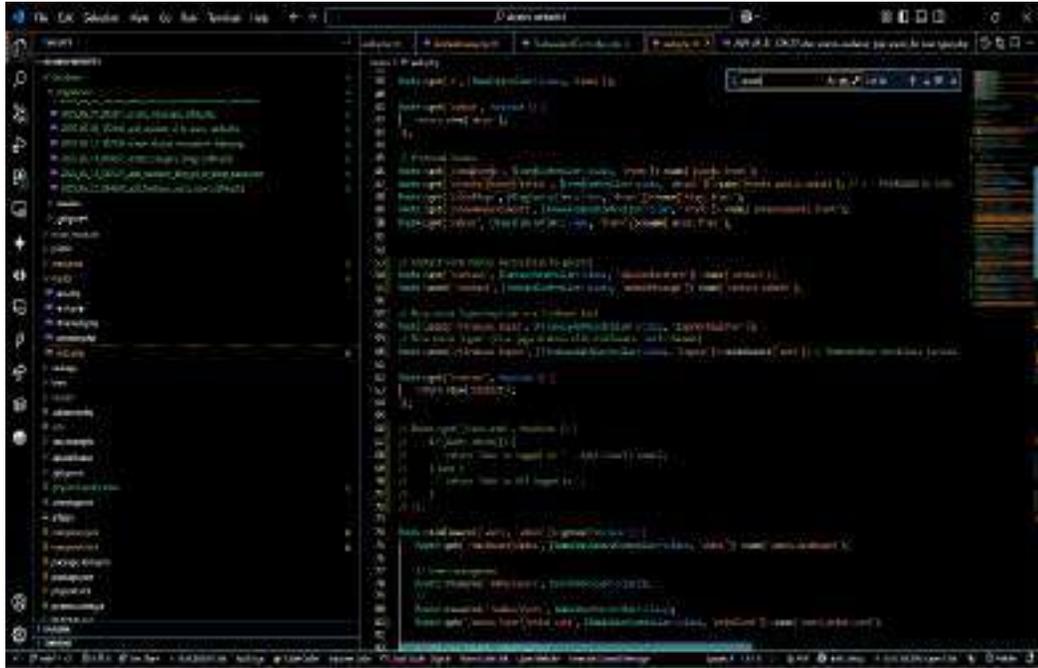
Gambar 3. 204 Migrasi Penambahan Tipe Audiens

Penjelasan: Cuplikan migrasi ini menunjukkan bagaimana tabel events dibangun dan dimodifikasi untuk mendukung fungsionalitas event publik. Awalnya, tabel dasar event dibuat. Kemudian, kolom-kolom seperti audience_type, target_majors, target_years, max_attendees, is_paid, dan price ditambahkan untuk mengelola audiens target dan detail pembayaran event.

Migrasi2025_05_31_174127_alter_events_audience_type_enum_for_new_types.php secara khusus mengubah tipe kolom audience_type untuk mendukung nilai 'all', yang menandakan event publik.

3. F-GUEST-003 & F-ALUM-001: Proses Registrasi Awal Alumni & Aktivasi Akun Mandiri

a. Rute Registrasi (routes/web.php)



```
use Illuminate\Http\Request;
use App\Http\Controllers\Auth\RegisteredUserController;

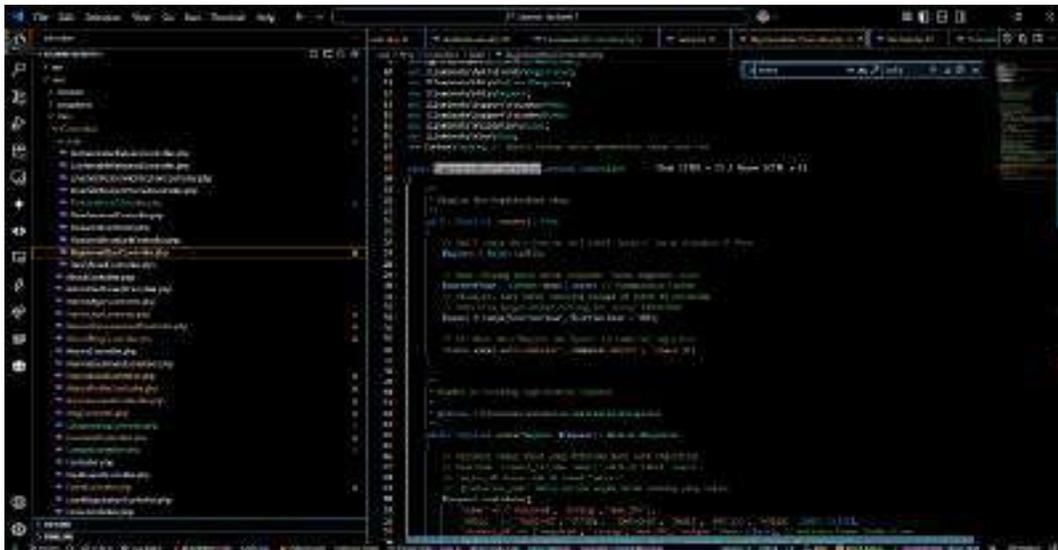
Route::get('/register', function () {
    // ...
});

Route::post('/register', function () {
    // ...
});
```

Gambar 3. 205 Proses Registrasi Awal Alumni & Aktivasi Akun Mandiri

Penjelasan: Cuplikan ini menunjukkan rute GET /register untuk menampilkan form pendaftaran dan POST /register untuk memproses data pendaftaran. Keduanya menggunakan RegisteredUserController.

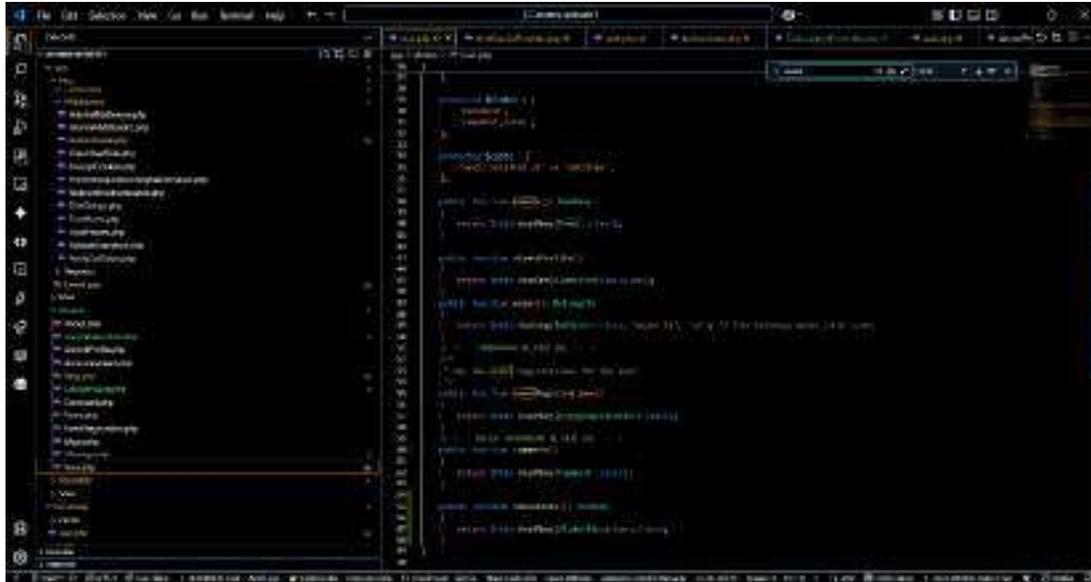
b. Logika Pendaftaran (app/Http/Controllers/Auth/RegisteredUserController.php)



Gambar 3. 206 Logika Pendaftaran

Penjelasan: Metode store ini bertanggung jawab memvalidasi data pendaftaran. Setelah validasi sukses, ia membuat entri baru di tabel users dengan role 'alumni', kemudian secara otomatis membuat AlumniProfile yang terhubung dengan user tersebut, termasuk menghasilkan alumni_code berdasarkan logika kode jurusan, tahun kelulusan, dan nomor induk siswa. Pengguna kemudian otomatis login.

- c. Model Terkait (app/Models/User.php & app/Models/AlumniProfile.php & app/Models/Major.php)



```
class User extends Model {
    protected $table = 'users';
    protected $primaryKey = 'id';
    protected $allowedFields = ['name', 'email', 'password', 'role'];

    public function validate() {
        $this->validateEmail();
        $this->validatePassword();
    }

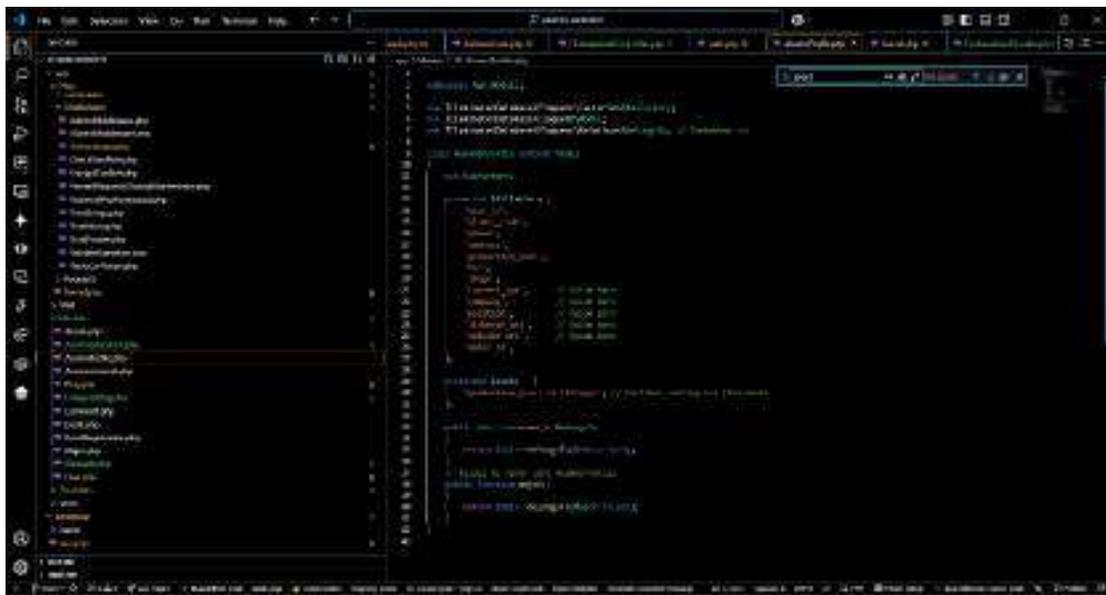
    private function validateEmail() {
        $email = $this->email;
        if (!filter_var($email, FILTER_VALIDATE_EMAIL)) {
            $this->addMessage('Email tidak valid');
        }
    }

    private function validatePassword() {
        $password = $this->password;
        if (strlen($password) < 8) {
            $this->addMessage('Password minimal 8 karakter');
        }
    }

    public function save() {
        $this->saveData();
    }

    private function saveData() {
        $data = [
            'name' => $this->name,
            'email' => $this->email,
            'password' => $this->password,
            'role' => $this->role
        ];
        $this->save($data);
    }
}
```

Gambar 3. 207 Model User



```
class AlumniProfile extends Model {
    protected $table = 'alumni_profiles';
    protected $primaryKey = 'id';
    protected $allowedFields = ['name', 'email', 'phone_number'];

    public function validate() {
        $this->validateEmail();
        $this->validatePhoneNumber();
    }

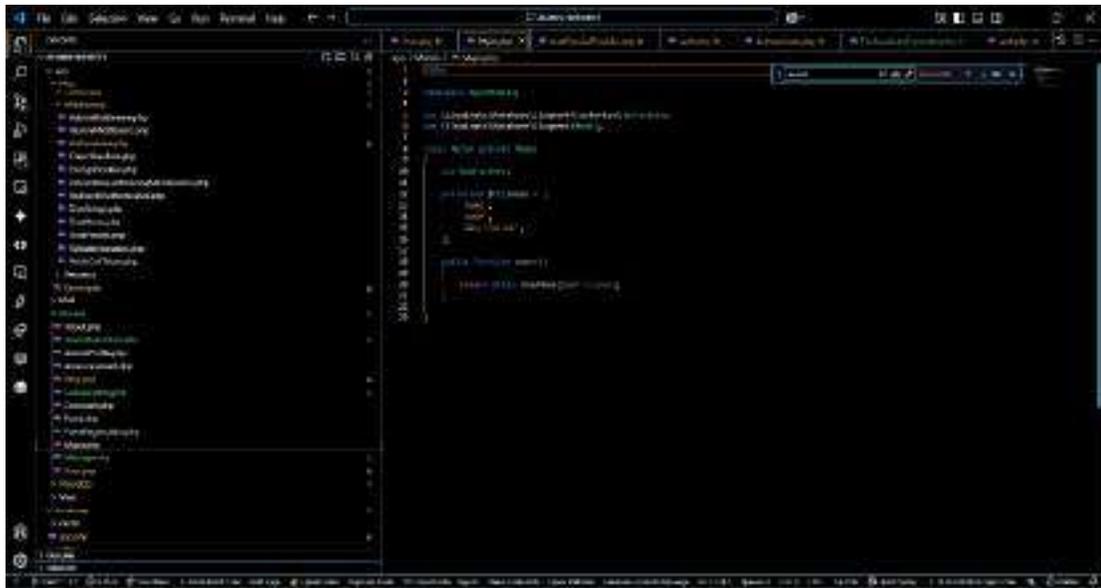
    private function validateEmail() {
        $email = $this->email;
        if (!filter_var($email, FILTER_VALIDATE_EMAIL)) {
            $this->addMessage('Email tidak valid');
        }
    }

    private function validatePhoneNumber() {
        $phone_number = $this->phone_number;
        if (!preg_match('/^08\d{10}$/', $phone_number)) {
            $this->addMessage('Nomor telepon tidak valid');
        }
    }

    public function save() {
        $this->saveData();
    }

    private function saveData() {
        $data = [
            'name' => $this->name,
            'email' => $this->email,
            'phone_number' => $this->phone_number
        ];
        $this->save($data);
    }
}
```

Gambar 3. 208 Model Profile Alumni



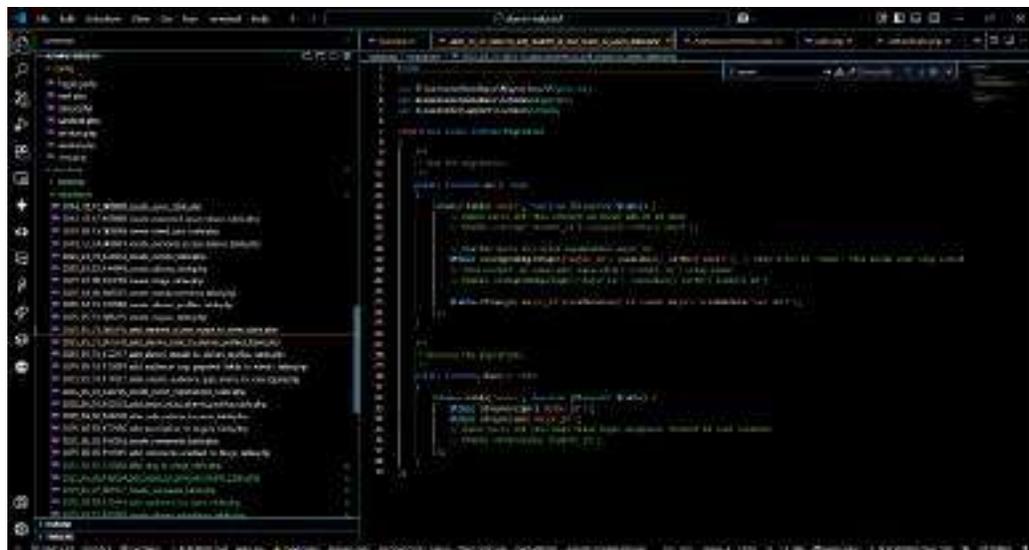
Gambar 3. 209 Model Jurusan

Penjelasan: Ini adalah definisi model untuk User, AlumniProfile, dan Major yang menunjukkan kolom-kolom yang dapat diisi (\$fillable) dan relasi (hasOne, belongsTo) yang menghubungkan mereka di database. Relasi ini penting untuk mengambil data terkait dengan mudah di kontroler.

d. Migrasi Database Terkait (database/migrations/...)

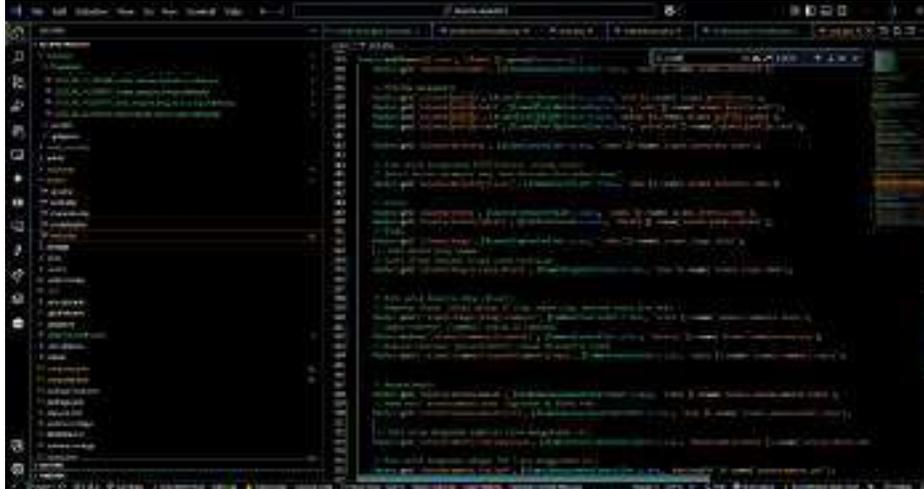


Gambar 3. 210 Migrasi User Table



Gambar 3. 211 Migrasi Penambahan Id dan Jurusan ke User Table

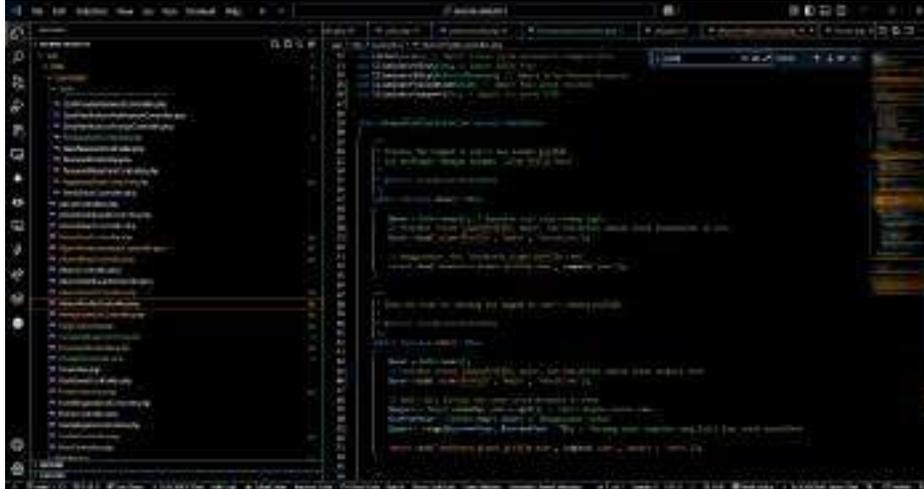
4. F-ALUM-002: Manajemen dan Pembaruan Profil Mandiri
 - a. Rute-rute Manajemen dan Pembaruan Profil (routes/web.php)



Gambar 3. 214 Manajemen dan Pembaruan Profil Mandiri

Penjelasan: Cuplikan ini menunjukkan rute utama yang memungkinkan alumni untuk mengelola informasi profil mereka. Rute `/alumni/profile` menampilkan detail profil melalui metode `show` di `AlumniProfileController`, sedangkan `/alumni/profile/edit` menampilkan formulir pengeditan melalui metode `edit`. Untuk menyimpan perubahan, rute `/alumni/profile` dengan metode `PUT` akan memanggil metode `update`. Terakhir, rute `/alumni/profile/card` memanggil metode `printCard` untuk mencetak kartu profil.

b. Logika (app/Http/Controllers/AlumniProfileController.php)



Gambar 3. 215 Logika Profile Alumni

Penjelasan:

- Metode show dan edit: Metode show bertanggung jawab untuk menampilkan detail profil alumni yang sedang login, sedangkan metode edit menampilkan formulir untuk memodifikasi profil tersebut. Keduanya memuat relasi penting seperti data profil alumni, jurusan, dan riwayat pendidikan untuk memastikan semua informasi yang relevan tersedia di tampilan. Metode edit juga menyiapkan data dropdown untuk jurusan dan rentang tahun.
- Metode update: Metode ini adalah inti dari manajemen profil, berfungsi untuk menyimpan semua perubahan yang dilakukan alumni. Ia memulai dengan melakukan validasi ekstensif terhadap semua input, mulai dari data pengguna dasar, detail profil alumni (seperti nomor telepon, alamat, tahun

kelulusan, info pekerjaan, dan tautan media sosial), hingga riwayat pendidikan. Setelah validasi, ia memperbarui informasi dasar pengguna. Bagian penanganan profil alumni menangani upload gambar (mengganti atau menghapus gambar lama jika diperlukan) dan secara otomatis men-generate alumni_code yang unik jika belum ada untuk profil tersebut. Yang terpenting, metode ini juga mengelola riwayat pendidikan alumni secara dinamis: ia bisa memperbarui entri pendidikan yang sudah ada, membuat entri baru, atau menghapus entri yang tidak lagi ada di formulir. Setelah semua pembaruan berhasil, pengguna akan dialihkan kembali ke halaman profil mereka dengan pesan sukses.

- Metode printCard: Metode ini dirancang khusus untuk menampilkan antarmuka yang memungkinkan alumni mencetak kartu profil mereka, dengan memuat semua informasi profil yang diperlukan untuk tampilan kartu.
- Metode showPublicProfile: Metode ini memungkinkan penampilan profil alumni secara publik, seringkali digunakan dalam direktori alumni. Dengan memanfaatkan *Route Model Binding*, objek AlumniProfile secara otomatis diambil berdasarkan parameter di URL. Kemudian, metode ini memuat relasi user, jurusan, dan riwayat pendidikan yang terkait untuk ditampilkan, memastikan profil lengkap disajikan kepada publik.

c. Model Terkait (app/Models/AlumniProfile.php)



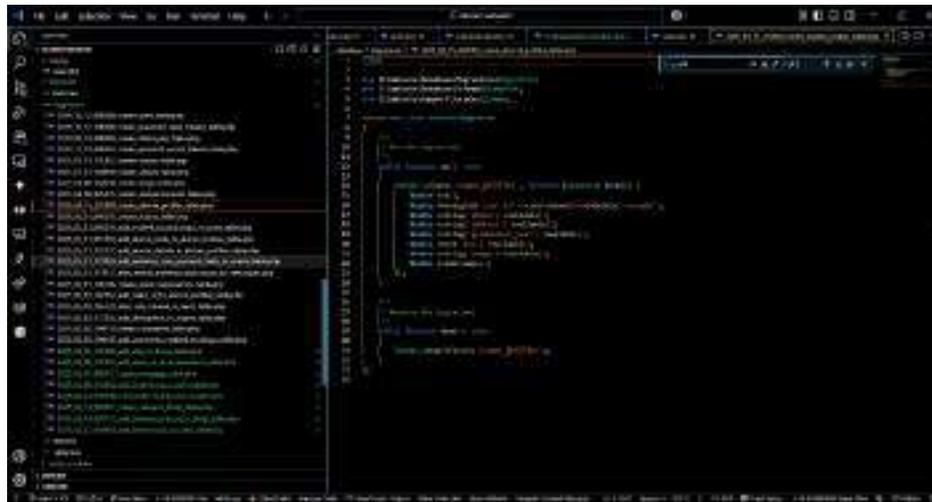
Gambar 3. 216 Model Alumni Profil

Penjelasan Model AlumniProfile mendefinisikan struktur data untuk menyimpan informasi detail profil setiap alumni. Ini mencakup kolom-kolom seperti user_id (untuk menghubungkan ke pengguna), alumni_code (kode unik alumni), informasi kontak (phone, address), data akademik (graduation_year, major_id), deskripsi diri (bio), gambar profil (image), serta detail karir saat ini (current_job, company, position) dan tautan profesional (linkedin_url, website_url). Atribut graduation_year di-cast sebagai integer untuk memastikan tipe data yang benar. Model ini juga memiliki dua relasi penting:

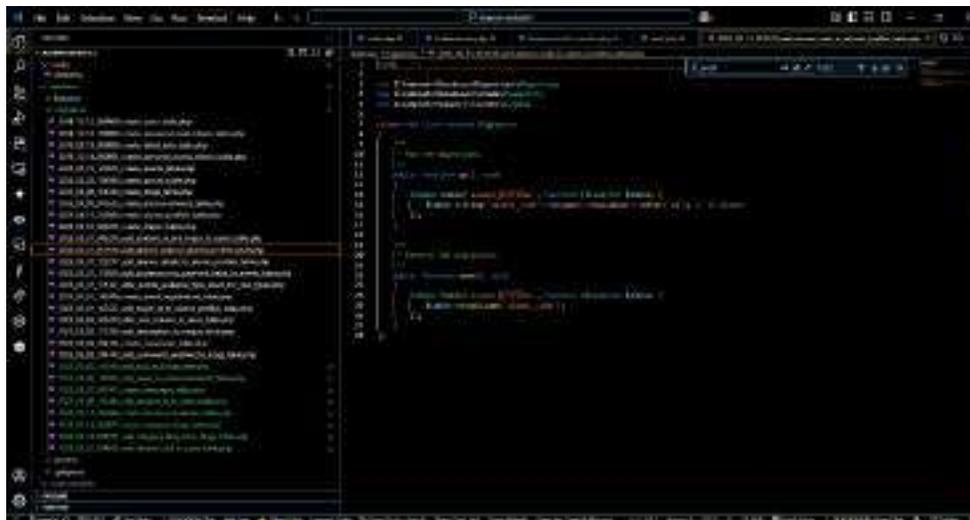
user(): Relasi BelongsTo yang mengaitkan setiap profil alumni dengan satu pengguna (User::class). Ini memungkinkan akses mudah ke data dasar pengguna dari profil alumni.

major(): Relasi BelongsTo lainnya yang menghubungkan profil alumni ke satu jurusan (Major::class), memungkinkan identifikasi jurusan alumni secara langsung dari profil mereka.

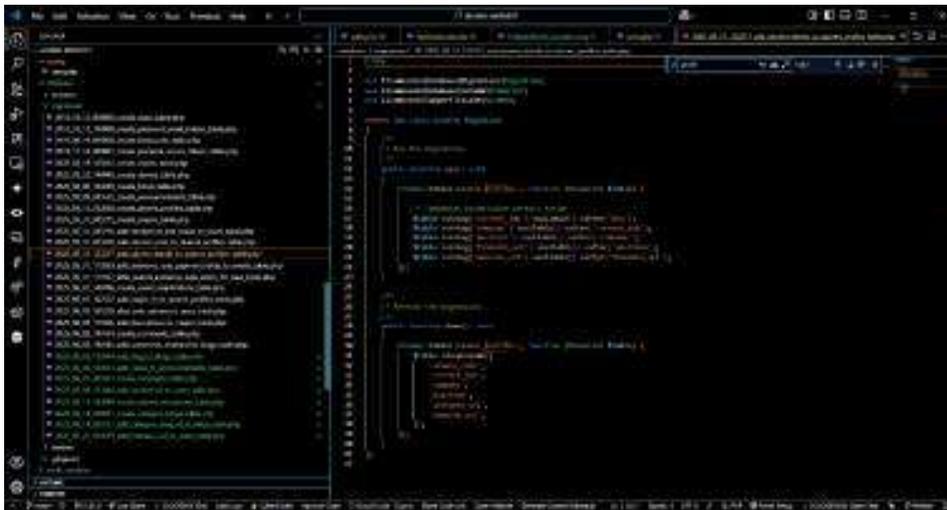
d. Migrasi Database Terkait (database/migrations/...)



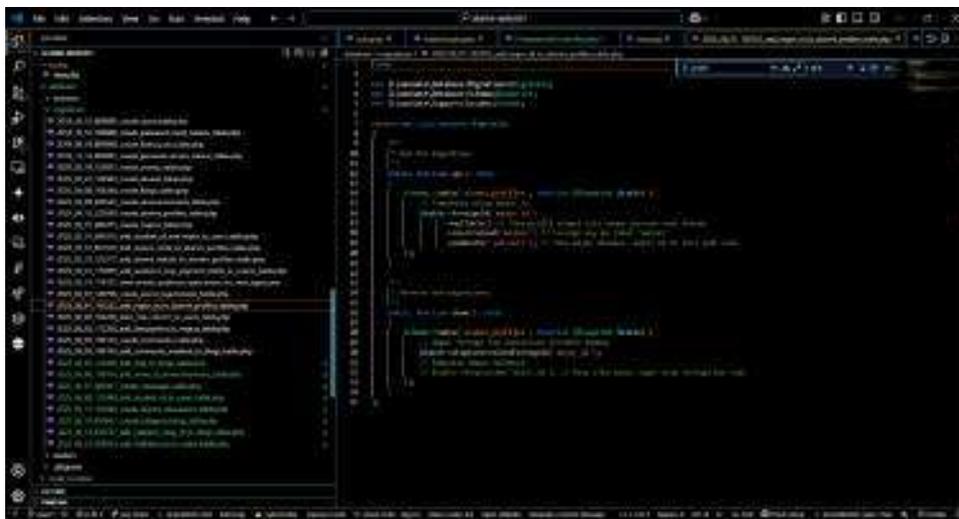
Gambar 3. 217 Migrasi Pembuatan Tabel Alumni Profil



Gambar 3. 218 Migrasi Penambahan Kode Alumni



Gambar 3. 219 Migrasi Penambahan Detail Alumni



Gambar 3. 220 Migrasi penambahan Id untuk Jurusan

Penjelasan: Keempat cuplikan kode migrasi Laravel ini secara berurutan menjelaskan evolusi skema tabel `alumni_profiles`. Migrasi pertama membuat tabel dasar `alumni_profiles` dengan ID unik, `user_id` sebagai *foreign key* ke tabel `users`, dan kolom untuk informasi kontak, alamat, tahun kelulusan, bio, serta gambar. Migrasi kedua menambahkan kolom `alumni_code` yang unik untuk identifikasi alumni. Selanjutnya, migrasi ketiga memperkaya tabel dengan kolom-

kolom terkait informasi karier seperti `current_job`, `company`, `position`, `linkedin_url`, dan `website_url`. Terakhir, migrasi keempat menambahkan `major_id` sebagai *foreign key* ke tabel `majors`, memungkinkan relasi antara profil alumni dengan jurusan mereka. Secara keseluruhan, migrasi ini secara bertahap membangun dan memperluas struktur database untuk menyimpan data profil alumni yang komprehensif.

5. F-ALUM-003: Pencarian & Filter Alumni Lanjutan

a. Rute-rute (`routes/web.php`)



Gambar 3. 221 Rute Pencarian & Filter Alumni Lanjutan

Penjelasan: Rute ini mengarah ke metode `index` pada `AlumniController`. Secara umum, metode `index` dalam sebuah *controller* untuk daftar sumber daya (dalam hal ini, daftar alumni) adalah tempat di mana fungsionalitas **pencarian dan filter** diimplementasikan untuk menampilkan daftar yang relevan kepada pengguna. Rute ini menyediakan titik akses untuk alumni melihat daftar alumni lain, yang dapat menyertakan opsi pencarian dan filter lanjutan di balik layarnya dalam `AlumniController`.

b. Logika (app/Http/Controllers/AlumniController.php)



Gambar 3. 222 Logika Alumni Controller

Penjelasan: AlumniController bertanggung jawab untuk mengelola daftar dan tampilan detail profil alumni. Metode index berfungsi sebagai **direktori alumni**, yang memungkinkan **pencarian dan filter lanjutan** berdasarkan jurusan, tahun kelulusan, serta pencarian umum (nama, email, bio, alamat, telepon, atau nama jurusan). Metode ini mengambil data jurusan dan tahun kelulusan unik untuk filter, kemudian menerapkan logika *query* berdasarkan parameter permintaan sebelum memaginasinya. Sementara itu, metode show berfungsi untuk menampilkan **detail profil seorang alumni spesifik** berdasarkan objek User yang diterima (dengan validasi untuk memastikan keberadaan profil alumni yang valid).

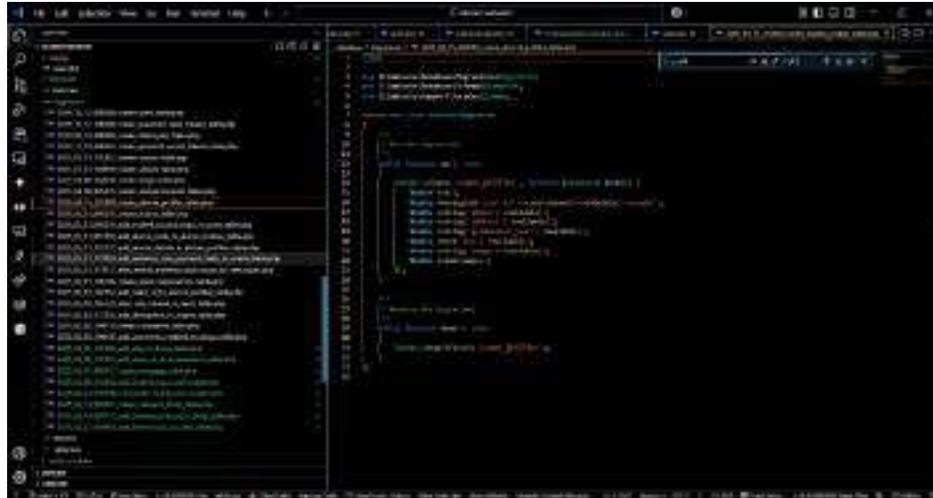
c. Model Terkait (app/Models/Event.php)



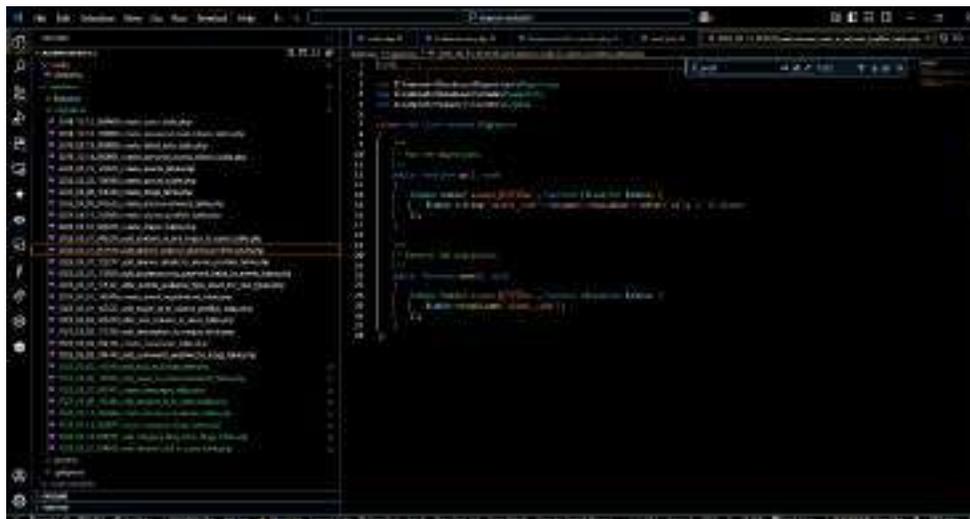
Gambar 3. 223 Model Profil Alumni

Penjelasan: Model Event mendefinisikan struktur data untuk event, termasuk kolom seperti title, description, event_date, location, audience_type, is_paid, dan price. Atribut event_date dan event_time di-cast sebagai tipe data yang sesuai. Kolom audience_type sangat penting untuk fungsionalitas ini, karena menentukan apakah event dapat diakses oleh publik (all), atau hanya untuk audiens spesifik (misal: major_only, year_only, major_and_year). Metode isEligibleForAlumni juga menunjukkan logika di mana event dengan audience_type 'all' selalu memenuhi syarat untuk diakses publik.

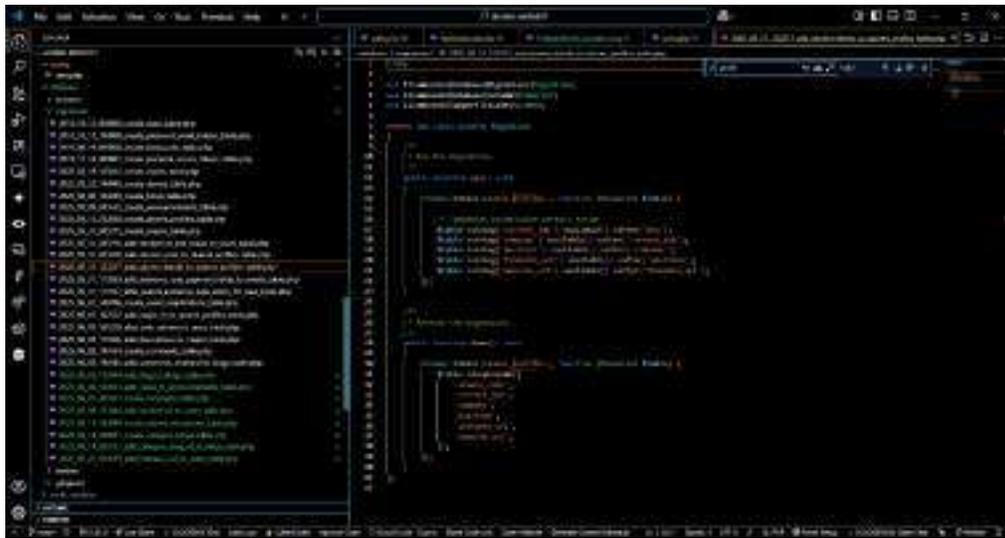
d. Migrasi Database Terkait (database/migrations/...)



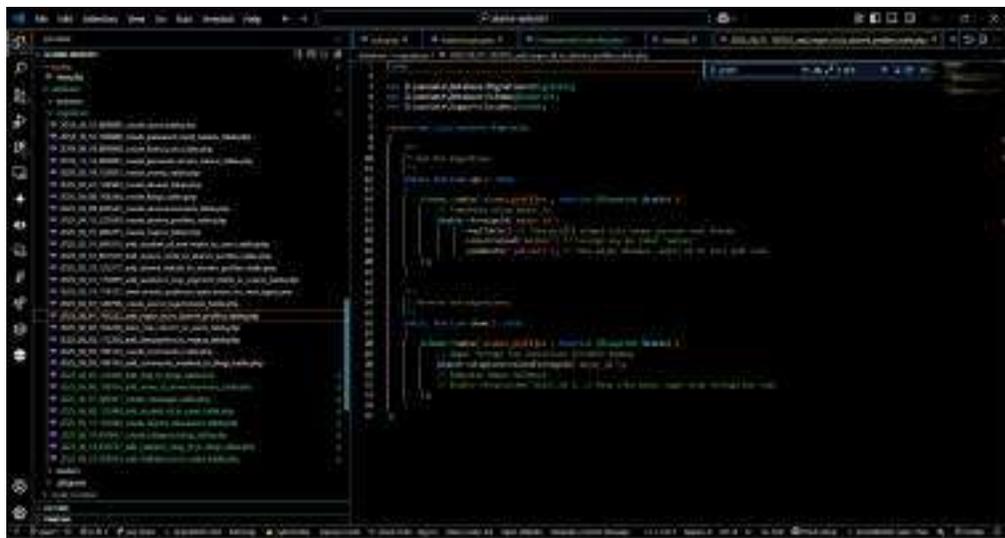
Gambar 3. 224 Migrasi Pembuatan Tabel Alumni



Gambar 3. 225 Migrasi Penambahan Kode Alumni



Gambar 3. 226 Migrasi Penambahan Detail Alumni



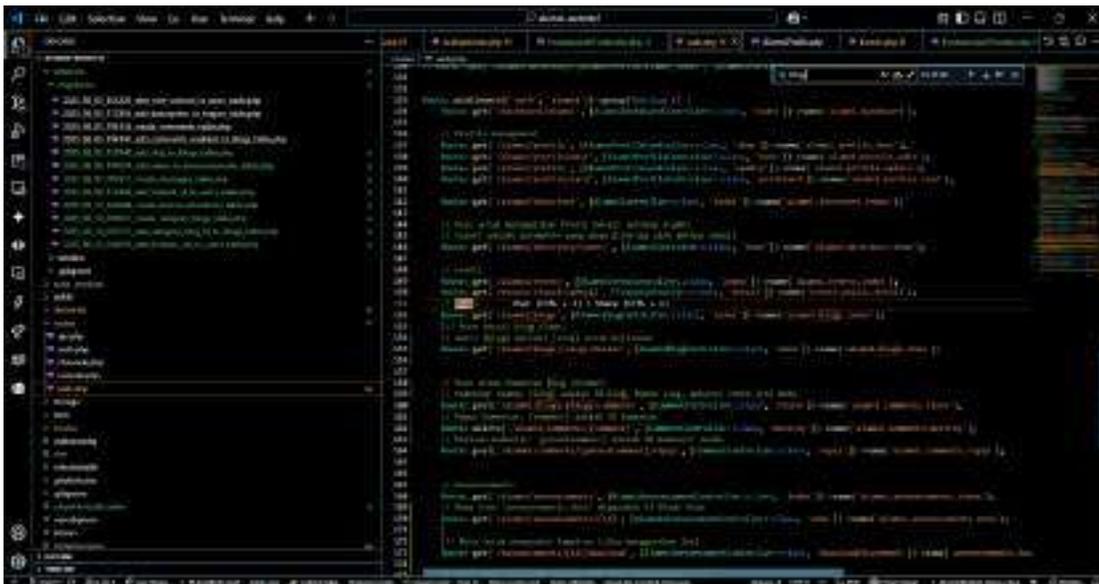
Gambar 3. 227 Migrasi Penambahan Id untuk Jurusan

Penjelasan: Keempat cuplikan kode migrasi Laravel ini secara berurutan menjelaskan evolusi skema tabel `alumni_profiles`. Migrasi pertama membuat tabel dasar `alumni_profiles` dengan ID unik, `user_id` sebagai *foreign key* ke tabel `users`, dan kolom untuk informasi kontak, alamat, tahun kelulusan, bio, serta gambar. Migrasi kedua menambahkan kolom `alumni_code` yang

unik untuk identifikasi alumni. Selanjutnya, migrasi ketiga memperkaya tabel dengan kolom-kolom terkait informasi karier seperti `current_job`, `company`, `position`, `linkedin_url`, dan `website_url`. Terakhir, migrasi keempat menambahkan `major_id` sebagai *foreign key* ke tabel `majors`, memungkinkan relasi antara profil alumni dengan jurusan mereka. Secara keseluruhan, migrasi ini secara bertahap membangun dan memperluas struktur database untuk menyimpan data profil alumni yang komprehensif.

6. F-ALUM-004: Akses & Interaksi Konten (Berita, Artikel)

a. Rute-rute (`routes/web.php`)



```
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Redirect;
use Illuminate\Support\Facades\Route;
use App\Http\Controllers\BlogController;
use App\Http\Controllers\ArticleController;
use App\Http\Controllers\CommentController;

Route::get('/blogs', [BlogController::class, 'index'])->name('blogs.index');
Route::get('/blogs/{id}', [BlogController::class, 'show'])->name('blogs.show');
Route::post('/blogs/{id}/comments', [CommentController::class, 'store'])->name('blogs.comments.store');
Route::delete('/blogs/{id}/comments/{comment_id}', [CommentController::class, 'destroy'])->name('blogs.comments.destroy');
Route::post('/blogs/{id}/comments/{comment_id}/reply', [CommentController::class, 'reply'])->name('blogs.comments.reply');

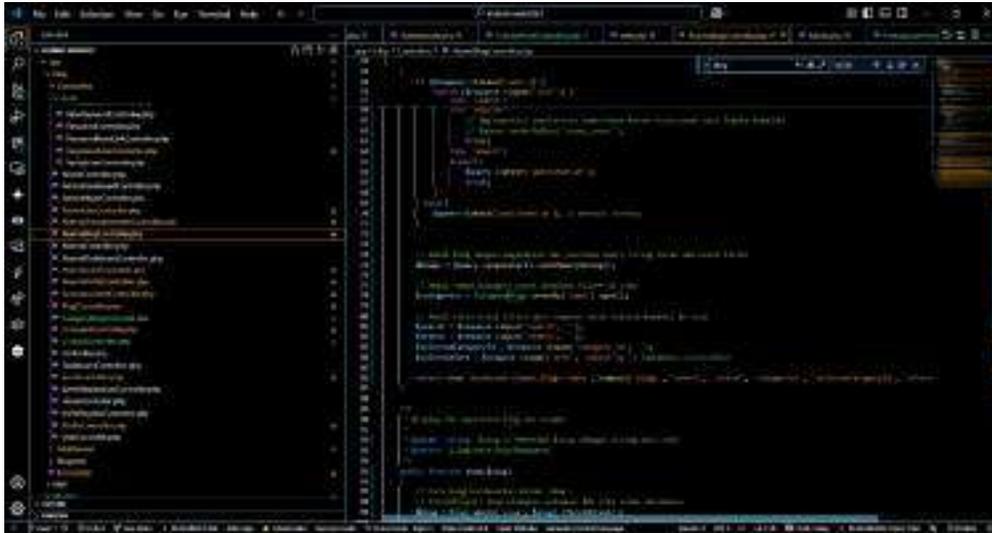
Route::get('/articles', [ArticleController::class, 'index'])->name('articles.index');
Route::get('/articles/{id}', [ArticleController::class, 'show'])->name('articles.show');
```

Gambar 3. 228 Rute Akses & Interaksi Konten (Berita, Artikel)

Penjelasan Rute-rute ini secara spesifik mengelola bagaimana alumni dapat mengakses daftar blog/artikel (`alumni.blogs.index`) dan melihat detail dari sebuah blog tertentu (`alumni.blogs.show`). Lebih lanjut, adanya rute untuk komentar (`alumni.comments.store`, `alumni.comments.destroy`, `alumni.comments.reply`) menunjukkan kemampuan alumni untuk

berinteraksi dengan konten tersebut, seperti menambahkan komentar, menghapus komentar mereka sendiri, atau membalas komentar lain.

b. Logika (app/Http/Controllers/AlumniBlogController.php)



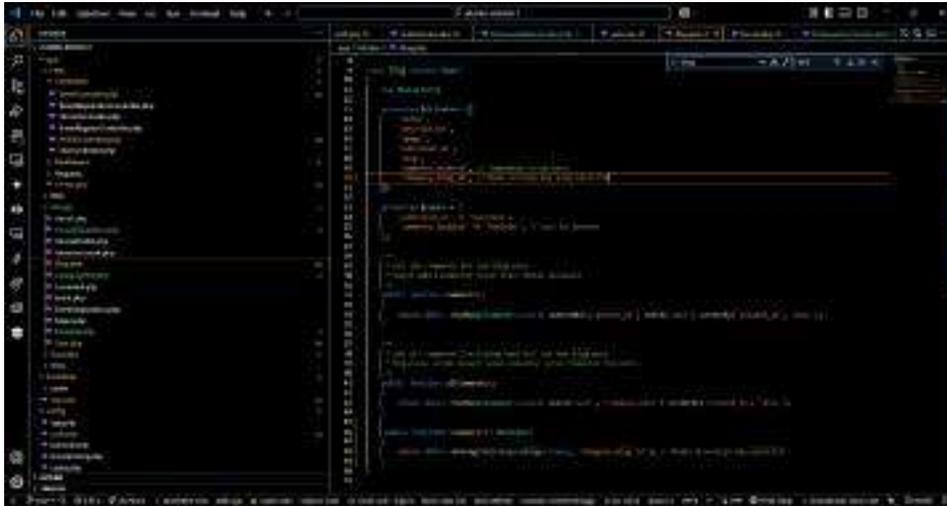
Gambar 3. 229 Logika Alumni Blog Controller

Penjelasan:

AlumniBlogController berfungsi untuk mengelola tampilan daftar dan detail blog yang dapat diakses oleh alumni. Metode index menampilkan daftar blog yang sudah dipublikasikan dengan fungsionalitas **pencarian berdasarkan judul atau deskripsi** dan **filter berdasarkan kategori**, serta opsi pengurutan (terbaru, terlama, dll.), memastikan hanya konten yang relevan dan terkini yang terlihat oleh alumni. Sementara itu, metode show bertanggung jawab untuk menampilkan **detail lengkap sebuah blog** berdasarkan *slug*-nya, memverifikasi status publikasi blog, dan secara kondisional memuat komentar beserta balasannya jika fitur

komentar diaktifkan atau pengguna adalah admin, memungkinkan interaksi alumni dengan konten blog.

iv. Model Terkait (app/Models/Blog.php)



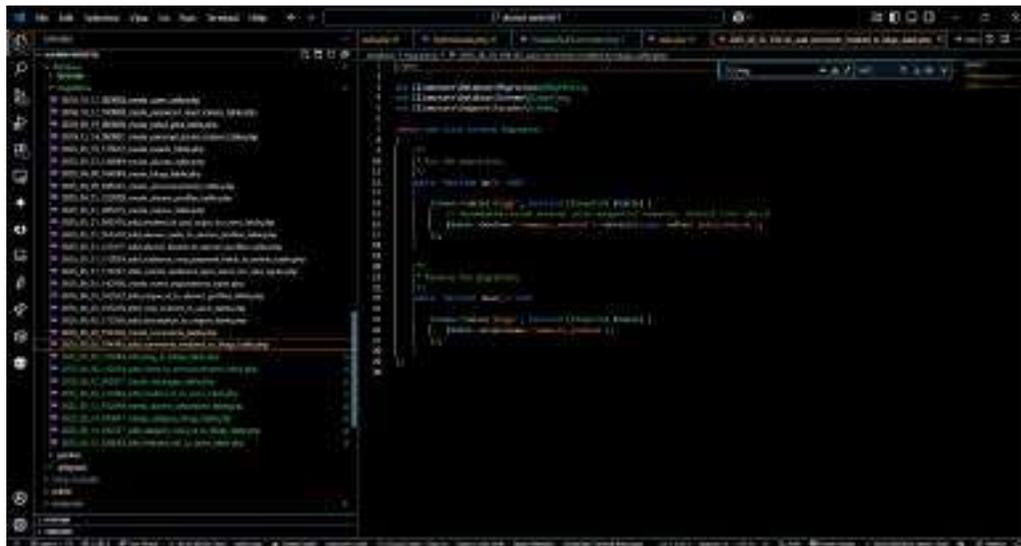
Gambar 3. 230 Model Blog

Penjelasan: Model Blog mendefinisikan struktur data untuk **berita atau artikel**. Ini mencakup kolom seperti title, description, image, published_at (tanggal publikasi), slug (URL yang ramah SEO), dan comments_enabled yang menentukan apakah komentar diizinkan. Kolom published_at di-cast sebagai tanggal/waktu, dan comments_enabled sebagai boolean. Model ini memiliki beberapa relasi penting: **comments()** untuk mengambil komentar tingkat atas (utama) yang terkait dengan blog, **allComments()** untuk memuat semua komentar beserta balasannya dalam struktur hierarkis, dan **category()** yang menghubungkan setiap blog ke kategori blog (CategoryBlog::class) melalui category_blog_id.

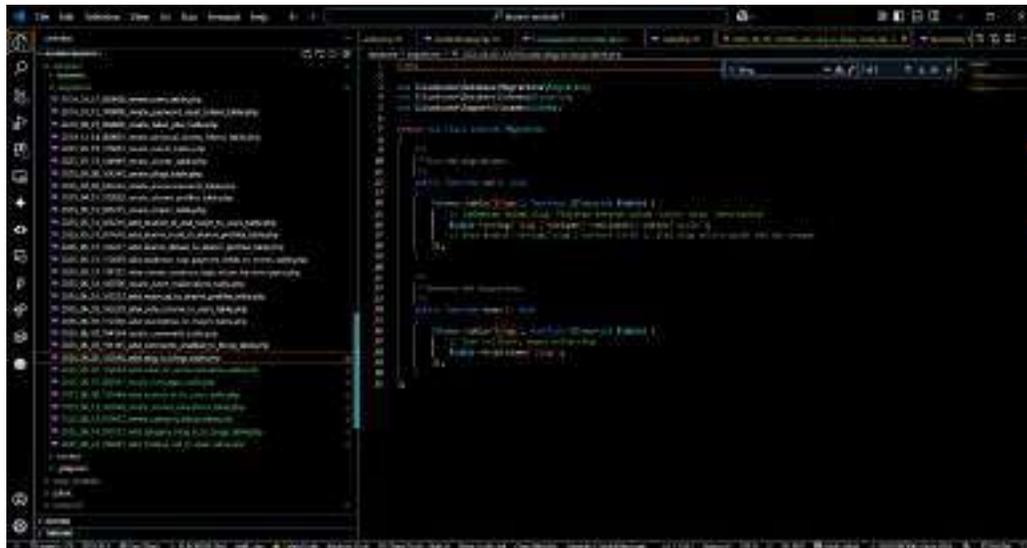
v. Migrasi Database Terkait (database/migrations/...)



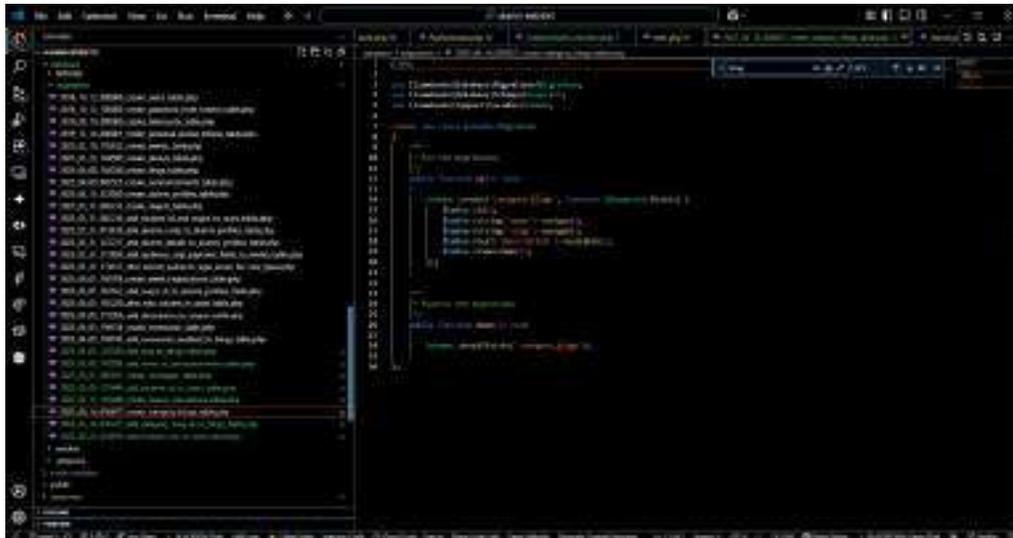
Gambar 3. 231 Migrasi Pembuatan Tabel Blog



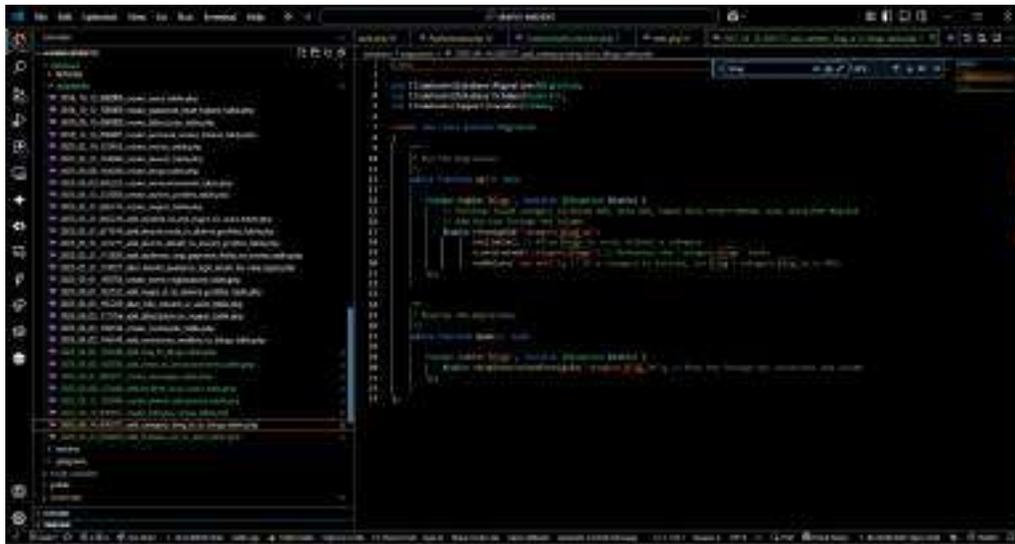
Gambar 3. 232 Migrasi Penambahan Enable Comment



Gambar 3. 233 Migrasi Penambahan Slug



Gambar 3. 234 Migrasi Pembuatan Tabel Kategori Blog



Gambar 3. 235 Migrasi Penambahan Kategori Blog Id

Penjelasan: cuplikan kode migrasi ini secara berurutan menjelaskan pembentukan dan pengembangan skema tabel blogs serta penambahan tabel category_blogs. Migrasi pertama menciptakan tabel blogs dasar dengan ID, gambar, judul, tanggal publikasi, dan deskripsi. Migrasi kedua menambahkan kolom comments_enabled ke tabel blogs untuk mengontrol fitur komentar. Migrasi ketiga memperkenalkan kolom slug yang unik, penting untuk URL ramah SEO. Migrasi keempat membuat tabel terpisah, category_blogs, untuk mengelola kategori blog dengan nama dan slug unik. Terakhir, migrasi kelima menambahkan *foreign key* category_blog_id ke tabel blogs, menghubungkan setiap blog dengan kategorinya dan memungkinkan blog tidak memiliki kategori atau kategori disetel null jika dihapus.

7. F-ALUM-005: Registrasi & Partisipasi Event Online/Offline

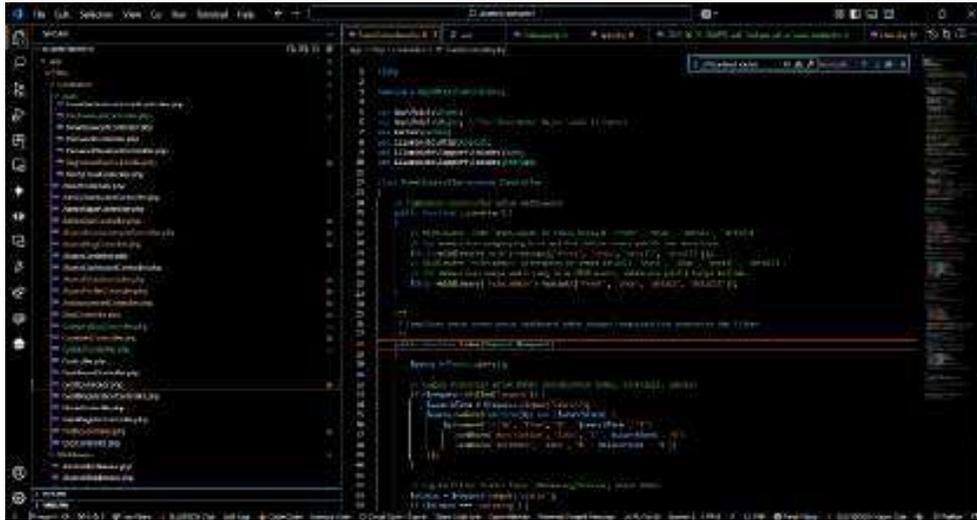
a. Rute-rute (routes/web.php)



Gambar 3. 236 Rute Registrasi & Partisipasi Event Online/Offline

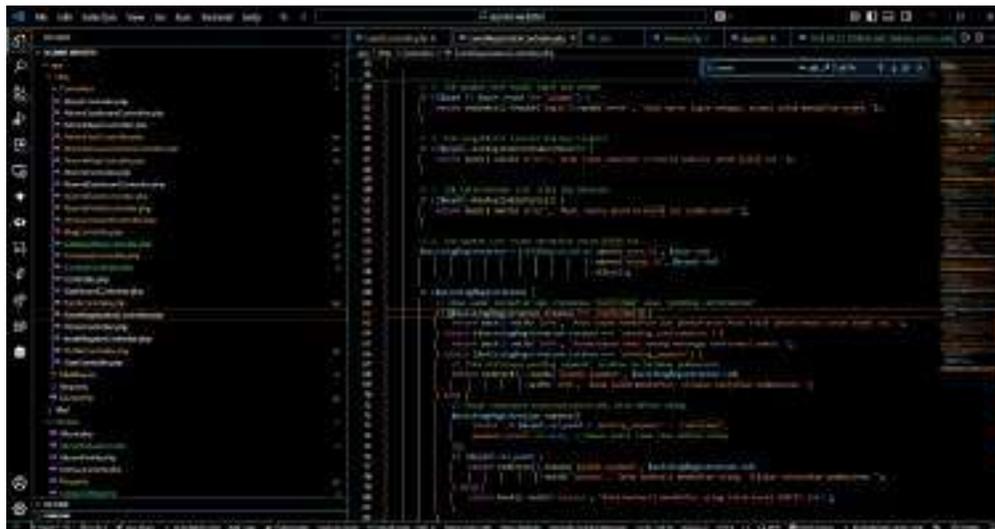
Penjelasan: Rute-rute ini secara spesifik mengelola bagaimana alumni dapat **melihat daftar event** (`alumni.events.index`) dan **melihat detail spesifik event** (`events.public.detail`). Yang terpenting, mereka juga menyediakan jalur untuk **registrasi event** melalui rute `events.rsvp` (untuk mendaftar/RSVP). Setelah mendaftar, proses partisipasi event online/offline didukung oleh rute `events.payment` untuk menampilkan formulir pembayaran dan `events.upload-proof` untuk mengunggah bukti pembayaran, mencakup alur registrasi dan konfirmasi partisipasi dalam event.

- b. Logika (app/Http/Controllers/EventController.php dan EventRegistrationController.php)



The screenshot shows a code editor with a dark theme. On the left, a file explorer displays a directory structure for an application, including folders like 'app', 'config', 'resources', and 'routes'. The main editor area shows the code for 'EventController.php'. The code includes a namespace declaration, use statements for 'Illuminate\Routing\Controller' and 'Illuminate\Http\Request', and a class definition for 'EventController' that inherits from 'Controller'. The class contains several methods, including a constructor, a 'show' method, and a 'create' method. The 'create' method uses 'Request::validate' to check for required fields like 'name', 'description', and 'start_date', and then uses 'Event::create' to save the event to the database. The 'show' method uses 'Event::findOrFail' to retrieve an event and 'Event::getEvents' to get a list of events.

Gambar 3. 237 Logika Event



The screenshot shows a code editor with a dark theme, similar to the previous one. The main editor area shows the code for 'EventRegistrationController.php'. The code includes a namespace declaration, use statements for 'Illuminate\Routing\Controller' and 'Illuminate\Http\Request', and a class definition for 'EventRegistrationController' that inherits from 'Controller'. The class contains several methods, including a constructor, a 'register' method, and a 'cancel' method. The 'register' method uses 'Request::validate' to check for required fields like 'name', 'description', 'start_date', and 'end_date', and then uses 'Event::create' to save the event to the database. The 'cancel' method uses 'Event::findOrFail' to retrieve an event and 'Event::cancel' to cancel the event.

Gambar 3. 238 Logika Pendaftaran Event

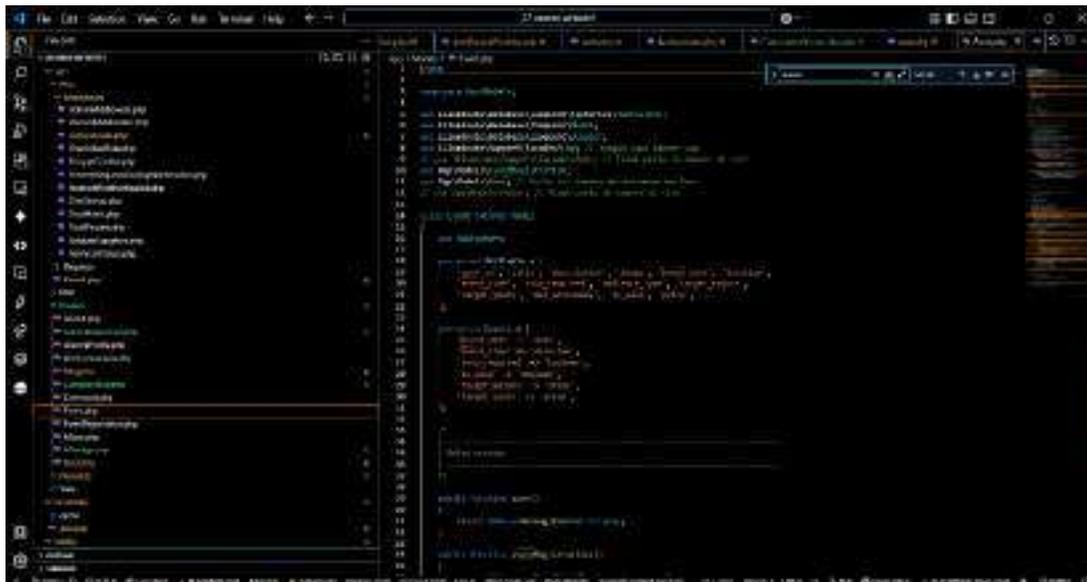
Penjelasan:

EventController bertanggung jawab penuh atas pengelolaan event, baik untuk tampilan publik maupun administrasi. Ia dilengkapi dengan middleware untuk memastikan hanya

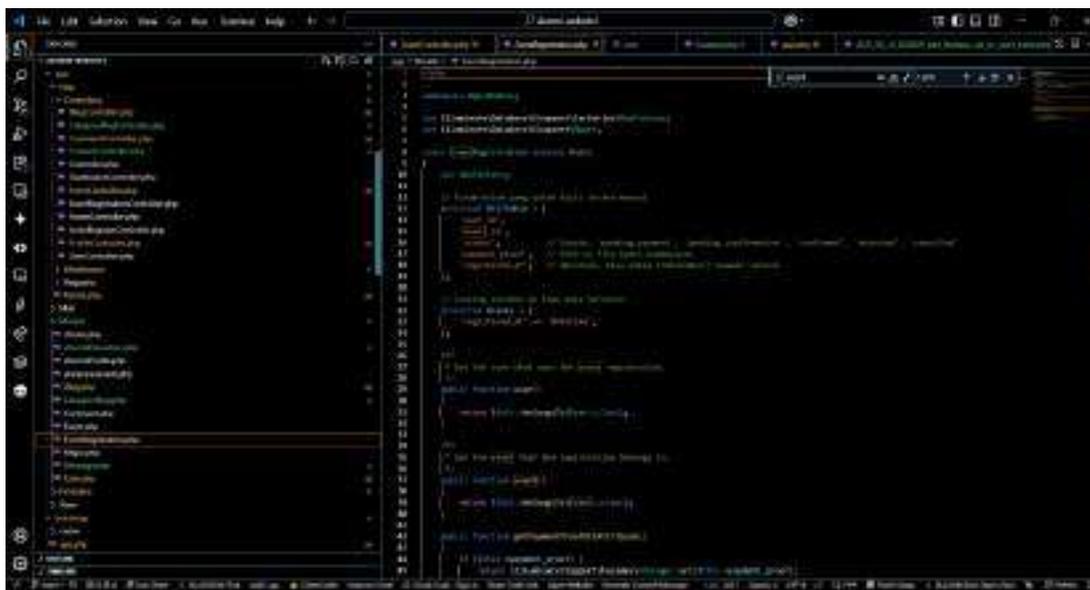
admin yang dapat membuat, mengedit, memperbarui, atau menghapus event (CRUD), sementara pengguna umum dapat melihat daftar event dan detailnya. Metode index menyediakan daftar event untuk admin dengan beragam filter (status, tipe, audiens, jurusan, angkatan) dan fitur pencarian. Metode front menampilkan event untuk publik (hanya yang berstatus 'all' atau 'publik'), juga dengan pencarian dan filter status. Sementara itu, metode create, store, edit, update, dan destroy mengelola siklus hidup event, termasuk validasi data, upload gambar, dan penargetan audiens spesifik (jurusan/angkatan), sedangkan show, detail, dan detail1 hanya menampilkan detail event ke berbagai tampilan.

EventRegistrationController mengelola proses pendaftaran event dan verifikasi pembayaran. Controller ini memisahkan fungsionalitas untuk alumni (RSVP, melihat formulir pembayaran, mengunggah bukti pembayaran) dan admin (melihat daftar pendaftaran, mengonfirmasi/menolak pembayaran, menghapus pendaftaran). Metode rsvp menangani proses pendaftaran awal, memeriksa kelayakan peserta, ketersediaan slot, dan status pendaftaran yang sudah ada, lalu mengarahkan ke proses pembayaran jika event berbayar. Metode showPaymentForm dan uploadPaymentProof memungkinkan alumni untuk menyelesaikan pembayaran event. Sementara itu, metode admin seperti indexAdmin, showAdmin, confirmPayment, rejectPayment, destroyAdmin, dan indexEventRegistrations menyediakan alat lengkap bagi admin untuk mengelola, memantau, dan memverifikasi semua pendaftaran event, termasuk filter pencarian untuk memudahkan administrasi.

c. Model Terkait (app/Models/Event.php dan EventRegistration.php)



Gambar 3. 239 Model Event



Gambar 3. 240 Model Pendaftaran Event

Penjelasan: Model Event mendefinisikan struktur data untuk menyimpan detail acara atau kegiatan. Ini mencakup informasi dasar seperti judul, deskripsi, gambar, tanggal dan waktu

event, lokasi, apakah pendaftaran (*rsvp_required*) diperlukan, apakah berbayar (*is_paid*) beserta harganya, dan kapasitas peserta (*max_attendees*). Yang paling krusial, model ini menentukan **tipe audiens** (*audience_type* seperti 'all', 'major_only', 'year_only', atau 'major_and_year') serta menyimpan *target_majors* dan *target_years* dalam format array JSON, memungkinkan event ditargetkan secara spesifik ke alumni tertentu. Model ini juga menyediakan relasi ke model User (pembuat event) dan EventRegistration (daftar pendaftar), serta metode *isEligibleForAlumni* untuk memverifikasi kelayakan alumni berdasarkan kriteria audiens event dan *hasAvailableSlots* untuk memeriksa ketersediaan kuota.

Model EventRegistration merepresentasikan **pendaftaran seorang pengguna untuk suatu event**. Kolom-kolomnya mencakup *user_id* dan *event_id* sebagai *foreign key* yang menghubungkan pendaftaran ini ke pengguna dan event terkait. Status pendaftaran (*status*) mencatat progresnya, seperti 'pending_payment', 'pending_confirmation', atau 'confirmed', yang esensial untuk melacak status partisipasi. Model ini juga menyimpan *payment_proof* untuk event berbayar dan memiliki relasi *belongsTo* ke model User dan Event untuk memudahkan akses data pendaftar dan detail event. Selain itu, terdapat *accessor* *getPaymentProofUrlAttribute* untuk menghasilkan URL publik dari bukti pembayaran yang tersimpan, mempermudah tampilan bukti di antarmuka.

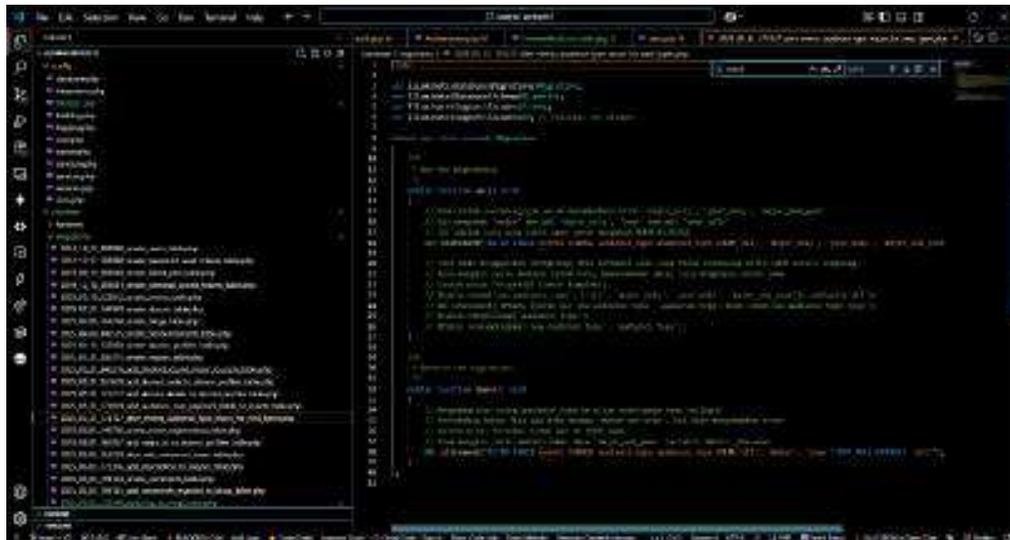
d. Migrasi Database Terkait (database/migrations/...)



Gambar 3. 241 Migrasi Pembuatan Tabel Event



Gambar 3. 242 Migrasi Penambahan Payment

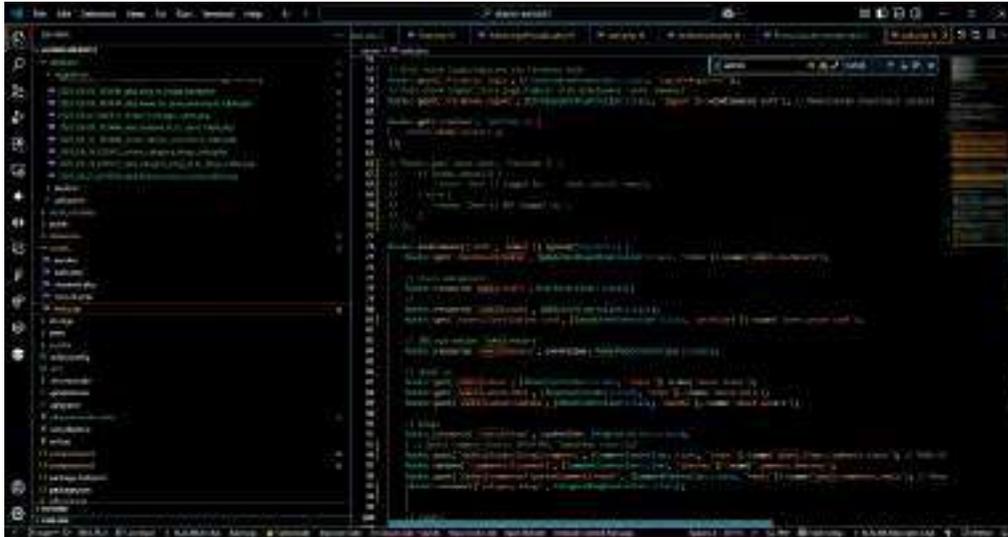


Gambar 3. 243 Migrasi Alter Tipe Audiens

Penjelasan: cuplikan kode migrasi Laravel ini secara bertahap membangun dan menyempurnakan struktur database untuk sistem event dan pendaftarannya. Migrasi pertama membuat tabel events dasar, mencakup detail seperti judul, deskripsi, tanggal, waktu, lokasi, dan apakah RSVP diperlukan, serta menautkannya ke tabel users. Migrasi kedua kemudian memperluas tabel events dengan menambahkan kolom-kolom penting untuk fungsionalitas lanjutan, termasuk **tipe audiens** (audience_type), **target jurusan** (target_majors), **target angkatan** (target_years), **batas peserta maksimal** (max_attendees), dan detail **pembayaran** (is_paid, price). Terakhir, migrasi ketiga membuat tabel event_registrations baru yang berfungsi untuk mencatat setiap pendaftaran, menautkannya ke users dan events, serta menyimpan status pendaftaran dan payment_proof untuk event berbayar, melengkapi keseluruhan sistem manajemen dan partisipasi event.

8. F-ADM-001: Manajemen Akun Admin

a. Rute Manajemen Pengguna (routes/web.php)

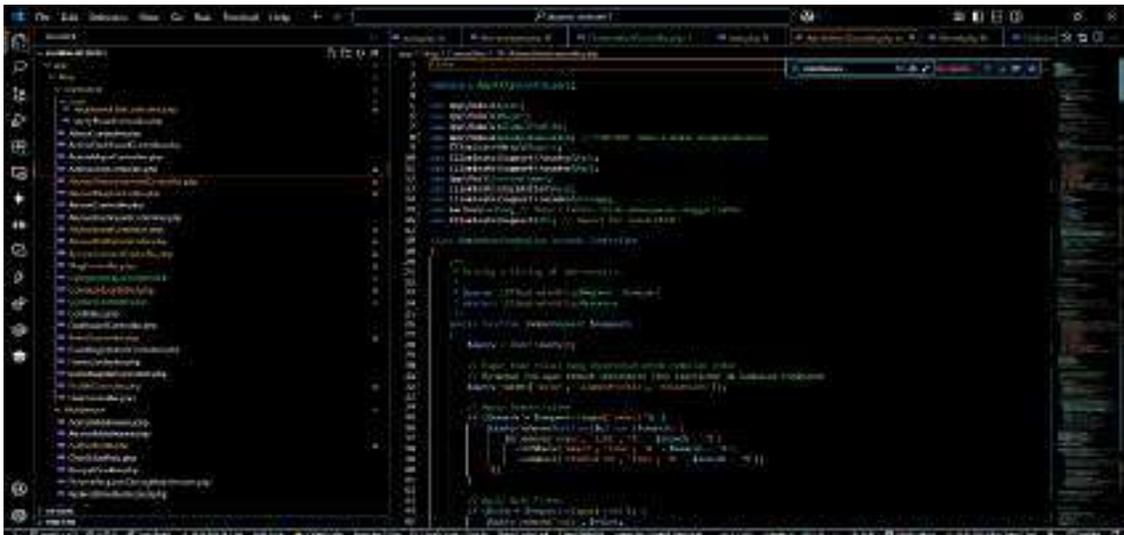


Gambar 3. 244 Rute Manajemen Akun Admin

Penjelasan: Rute-rute ini berada di bawah grup middleware yang mengharuskan pengguna terotentikasi (auth) dan memiliki peran admin (admin). Baris kunci di sini adalah `Route::resource('admin/users', UserController::class);` atau `Route::resource('/admin/users', AdminUserController::class);`. Meskipun ada sedikit kebingungan karena ada dua definisi `Route::resource` yang menunjuk ke URI yang sama (`/admin/users`) namun dengan *controller* yang berbeda (`UserController` dan `AdminUserController`), secara umum, sebuah `Route::resource` akan secara otomatis mendaftarkan tujuh rute RESTful standar, yang mencakup operasi membuat, melihat, mengedit, memperbarui, dan menghapus sumber daya.

Dalam konteks F-ADM-001, rute-rute ini memungkinkan admin untuk sepenuhnya mengelola akun pengguna (user), yang akan mencakup akun admin lainnya. Ini berarti admin dapat membuat akun pengguna baru, mengedit detailnya (termasuk peran/hak akses), dan menghapus akun yang tidak diperlukan, sehingga memenuhi persyaratan manajemen akun admin. Rute printCard juga memungkinkan admin untuk mencetak kartu pengguna tertentu, yang bisa relevan untuk identifikasi internal.

b. Logika Manajemen Pengguna (app/Http/Controllers/AdminUserController.php)



Gambar 3. 245 Logika Manajemen Pengguna

Penjelasan :

Konstruktor: Memastikan hanya pengguna dengan peran 'admin' yang dapat mengakses metode di kontroler ini.

index(): Mengambil daftar pengguna dengan dukungan pencarian (search) berdasarkan nama, email, atau student_id. Terdapat juga filter berdasarkan role (admin/alumni), major_id

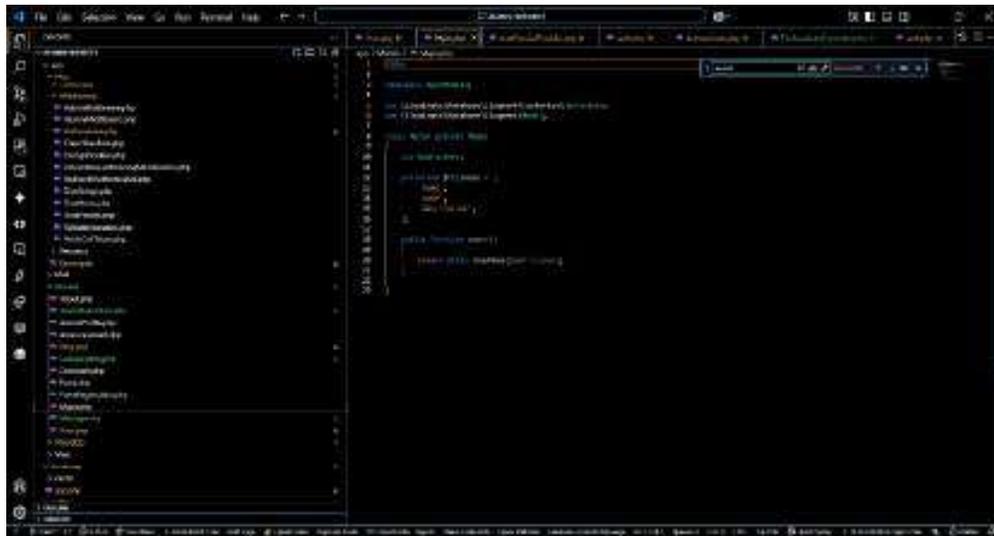
(jurusan), dan `graduation_year` (tahun angkatan). Filter tahun angkatan menggunakan `whereHas` untuk mencari di relasi `alumniProfile`. Hasilnya di-paginate dan dikirim ke view.

create() & store(): Menampilkan form dan memproses penambahan pengguna baru. Logika ini mencakup validasi input, pembuatan entri User dengan peran yang ditentukan, dan secara kondisional membuat `AlumniProfile` baru jika peran yang dipilih adalah 'alumni', lengkap dengan pembuatan `alumni_code` otomatis.

show(): Menampilkan detail lengkap dari satu pengguna, termasuk relasi major dan `alumniProfile`.

edit() & update(): Menampilkan form dan memproses pembaruan data pengguna. Validasi `unique` untuk `email` dan `student_id` diabaikan untuk pengguna yang sedang diedit. Password dapat diubah secara opsional. Logika juga menangani pembuatan/pembaruan/penghapusan `AlumniProfile` berdasarkan perubahan peran pengguna.

destroy(): Menghapus pengguna dari database.



Gambar 3. 248 Model Jurusan

Penjelasan :

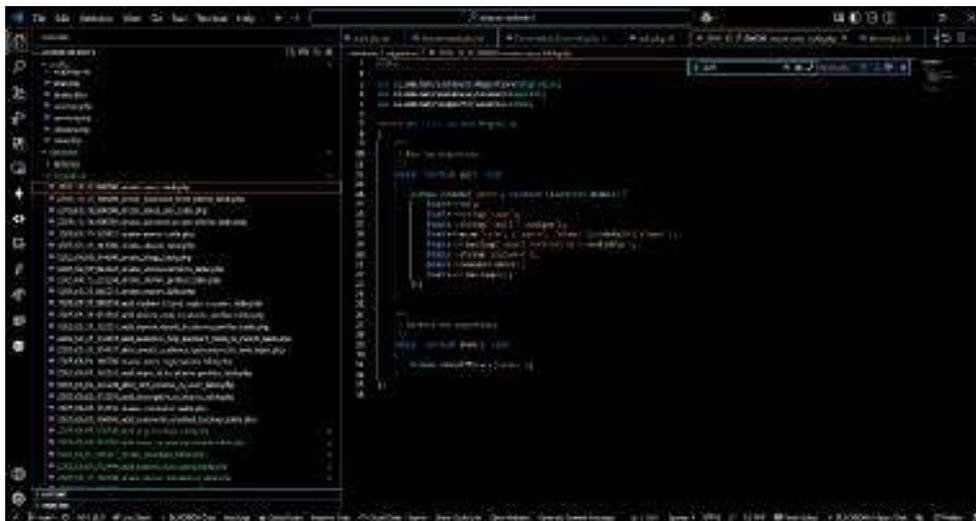
User Model: Mendefinisikan kolom-kolom yang dapat diisi (name, email, student_id, major_id, role, password) dan relasi hasOne ke AlumniProfile serta belongsTo ke Major.

Metode hasRole digunakan untuk otorisasi.

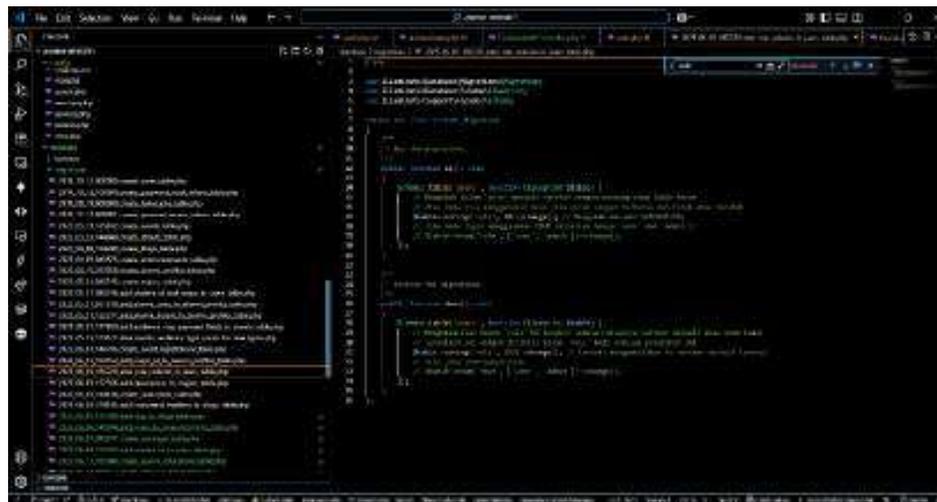
AlumniProfile Model: Mendefinisikan kolom-kolom profil alumni dan relasi belongsTo ke User dan Major. Ini adalah tempat detail spesifik alumni disimpan.

Major Model: Mendefinisikan nama dan kode jurusan, serta relasi hasMany ke User dan AlumniProfile untuk memudahkan pengambilan data terkait.

d. Migrasi Database Terkait (database/migrations/...)



Gambar 3. 249 Migrasi Pembuatan Tabel User



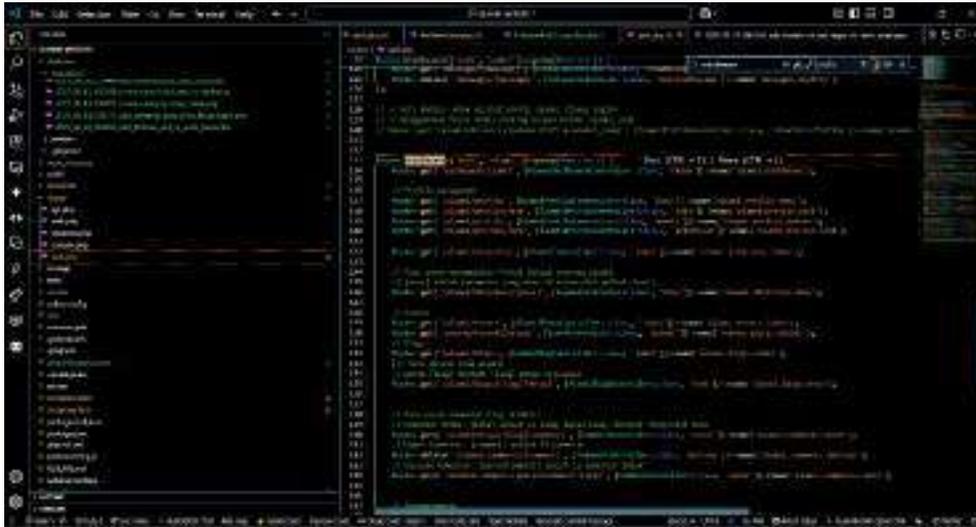
Gambar 3. 250 Migrasi Alter Role

Penjelasan: Cuplikan migrasi ini menunjukkan bagaimana tabel users, alumni_profiles, dan majors dibangun dan dimodifikasi. Ini mencakup penambahan kolom role, student_id, major_id pada tabel users, serta kolom alumni_code, graduation_year, dan detail karier pada

tabel alumni_profiles. Relasi foreign key juga didefinisikan untuk memastikan integritas data antar tabel.

9. F-ADM-006: Manajemen Akun Pengguna & Otorisasi

a. Rute Manajemen Pengguna (routes/web.php)

The image shows a screenshot of a code editor with a dark theme. On the left, there is a file explorer showing a directory structure with folders like 'resources', 'views', and 'routes'. The main area displays PHP code for routes. The code includes a namespace declaration, a use statement for 'Illuminate\Routing\Controllers\Middleware', and a series of route definitions. These routes are grouped under a 'middleware: auth' constraint and are restricted to users with the 'admin' role. The routes cover various actions: 'index' (GET), 'create' (POST), 'store' (POST), 'edit' (GET), 'update' (PUT), 'destroy' (DELETE), and 'show' (GET). Each route is associated with a specific controller and method.

```
namespace App\Http\Controllers;
use Illuminate\Routing\Controllers\Middleware;

class UsersController extends Controller
{
    /**
     * Display a listing of the resource.
     */
    public function index()
    {
        //
    }

    /**
     * Show the form for creating a new resource.
     */
    public function create()
    {
        //
    }

    /**
     * Store a newly created resource in storage.
     */
    public function store()
    {
        //
    }

    /**
     * Show the form for editing the specified resource.
     */
    public function edit($id)
    {
        //
    }

    /**
     * Update the specified resource in storage.
     */
    public function update($id)
    {
        //
    }

    /**
     * Remove the specified resource from storage.
     */
    public function destroy($id)
    {
        //
    }

    /**
     * Show the details of the specified resource.
     */
    public function show($id)
    {
        //
    }
}
```

Gambar 3. 251 Rute Manajemen Akun Pengguna & Otorisasi

Penjelasan: Cuplikan ini menunjukkan rute-rute khusus yang digunakan Admin untuk mengelola akun pengguna. Semua rute ini berada di bawah grup middleware auth dan role:admin, memastikan hanya pengguna yang sudah login dengan peran 'admin' yang dapat mengaksesnya. Rute-rute ini mencakup operasi index (daftar), create (form tambah), store (simpan), edit (form edit), update (perbarui), destroy (hapus), dan show (detail) untuk entitas User.

b. Logika Manajemen Pengguna (app/Http/Controllers/AdminUserController.php)



Gambar 3. 252 Logika Manajemen Pengguna

Penjelasan :

Konstruktor: Memastikan hanya pengguna dengan peran 'admin' yang dapat mengakses metode di kontroler ini.

index(): Mengambil daftar pengguna dengan dukungan pencarian (search) berdasarkan nama, email, atau student_id. Terdapat juga filter berdasarkan role (admin/alumni), major_id (jurusan), dan graduation_year (tahun angkatan). Filter tahun angkatan menggunakan whereHas untuk mencari di relasi alumniProfile. Hasilnya di-paginate dan dikirim ke view.

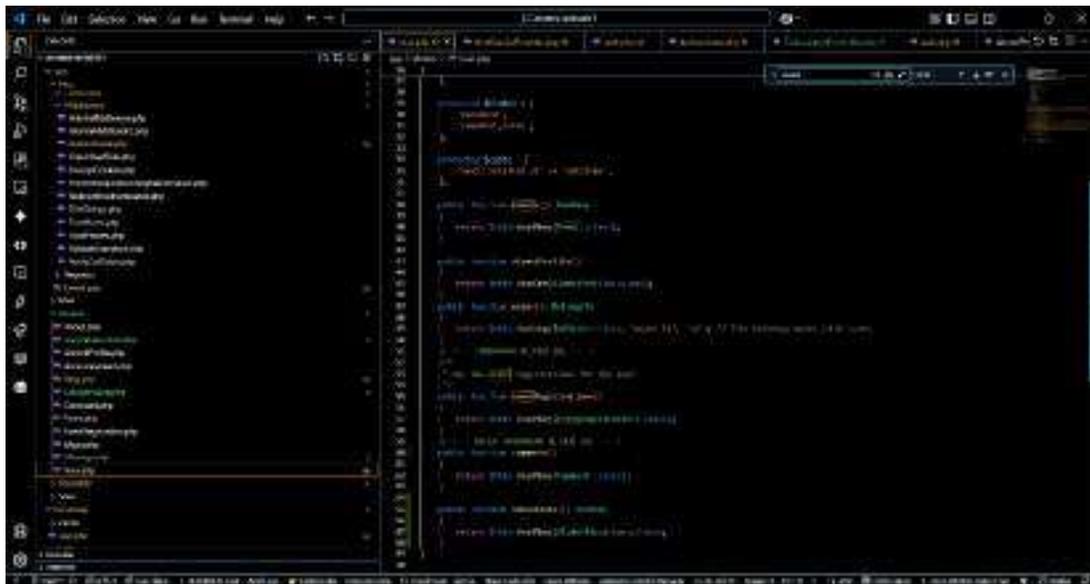
create() & store(): Menampilkan form dan memproses penambahan pengguna baru. Logika ini mencakup validasi input, pembuatan entri User dengan peran yang ditentukan, dan secara kondisional membuat AlumniProfile baru jika peran yang dipilih adalah 'alumni', lengkap dengan pembuatan alumni_code otomatis.

show(): Menampilkan detail lengkap dari satu pengguna, termasuk relasi major dan alumniProfile.

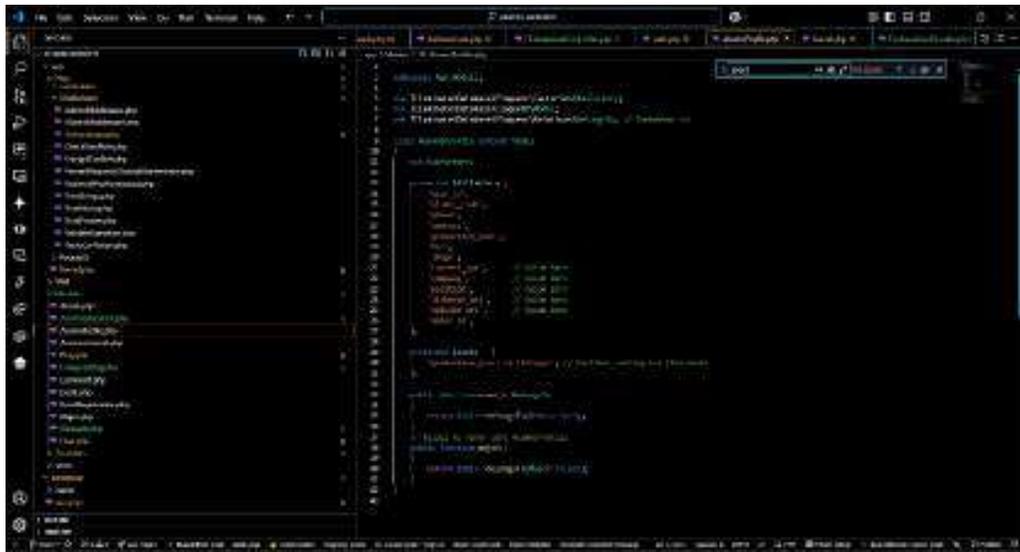
edit() & update(): Menampilkan form dan memproses pembaruan data pengguna. Validasi unique untuk email dan student_id diabaikan untuk pengguna yang sedang diedit. Password dapat diubah secara opsional. Logika juga menangani pembuatan/pembaruan/penghapusan AlumniProfile berdasarkan perubahan peran pengguna.

destroy(): Menghapus pengguna dari database.

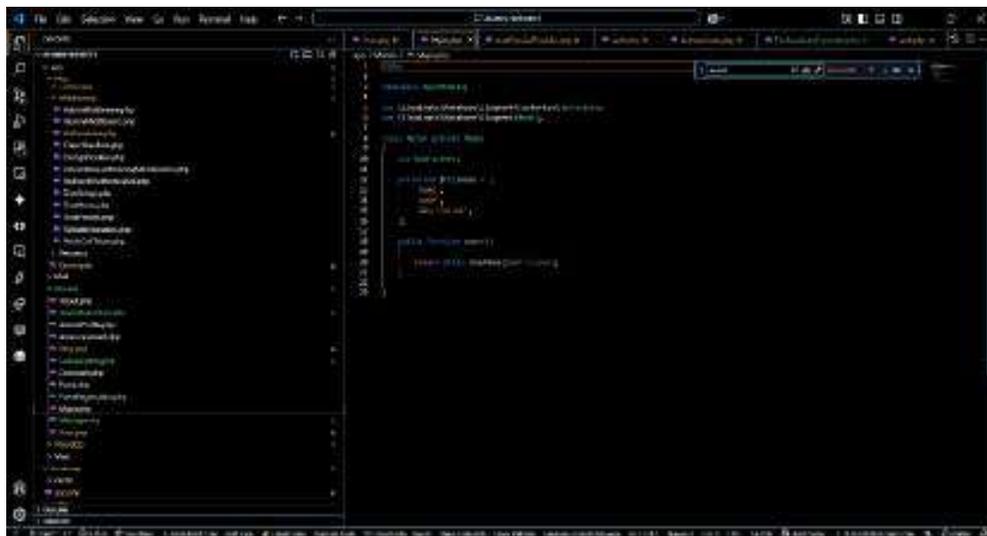
- c. Model Terkait (app/Models/User.php, app/Models/AlumniProfile.php, app/Models/Major.php)



Gambar 3. 253 Model User



Gambar 3. 254 Model Profil Alumni



Gambar 3. 255 Model Jurusan

Penjelasan :

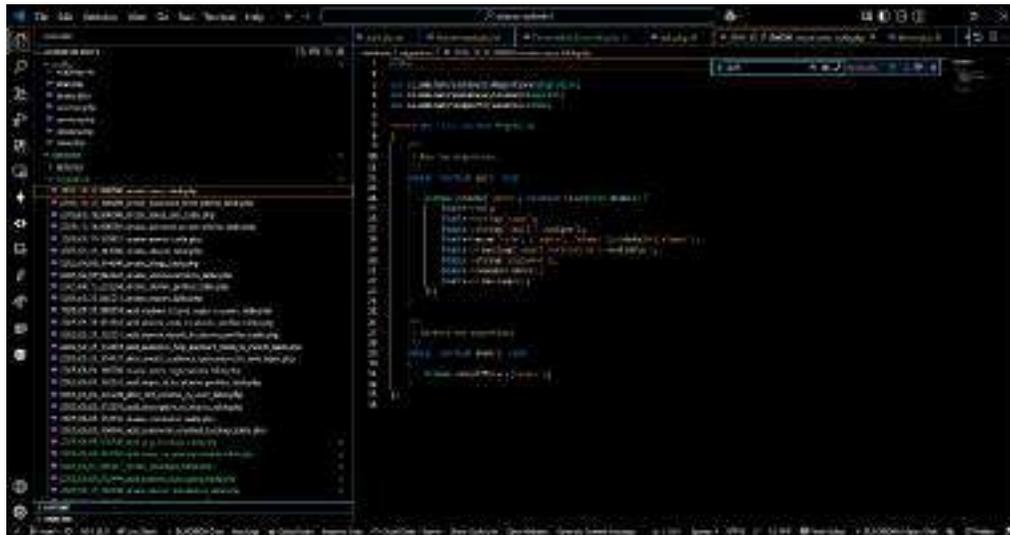
User Model: Mendefinisikan kolom-kolom yang dapat diisi (name, email, student_id, major_id, role, password) dan relasi hasOne ke AlumniProfile serta belongsTo ke Major.

Metode hasRole digunakan untuk otorisasi.

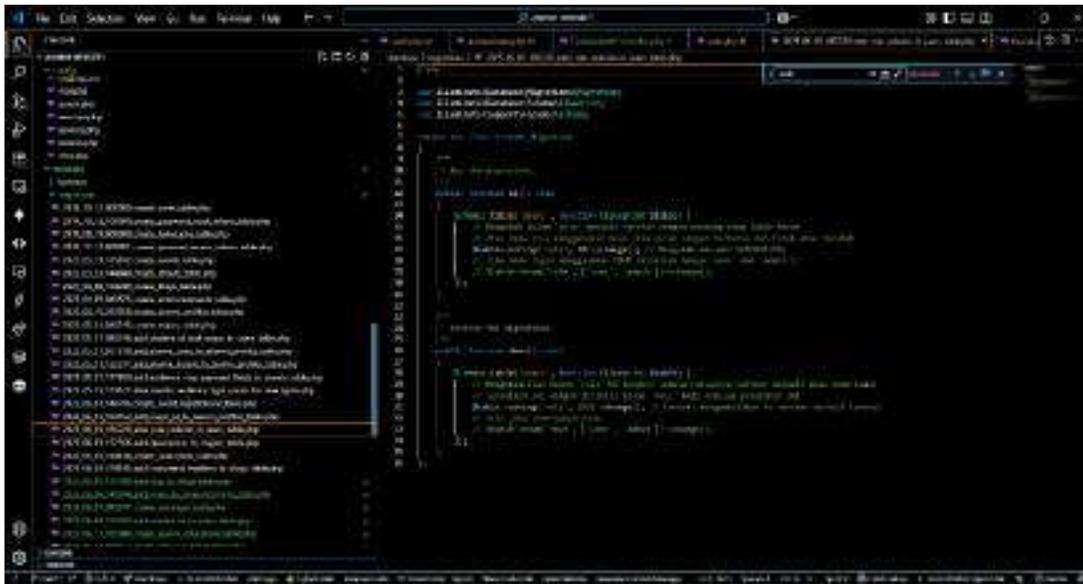
AlumniProfile Model: Mendefinisikan kolom-kolom profil alumni dan relasi belongsTo ke User dan Major. Ini adalah tempat detail spesifik alumni disimpan.

Major Model: Mendefinisikan nama dan kode jurusan, serta relasi hasMany ke User dan AlumniProfile untuk memudahkan pengambilan data terkait.

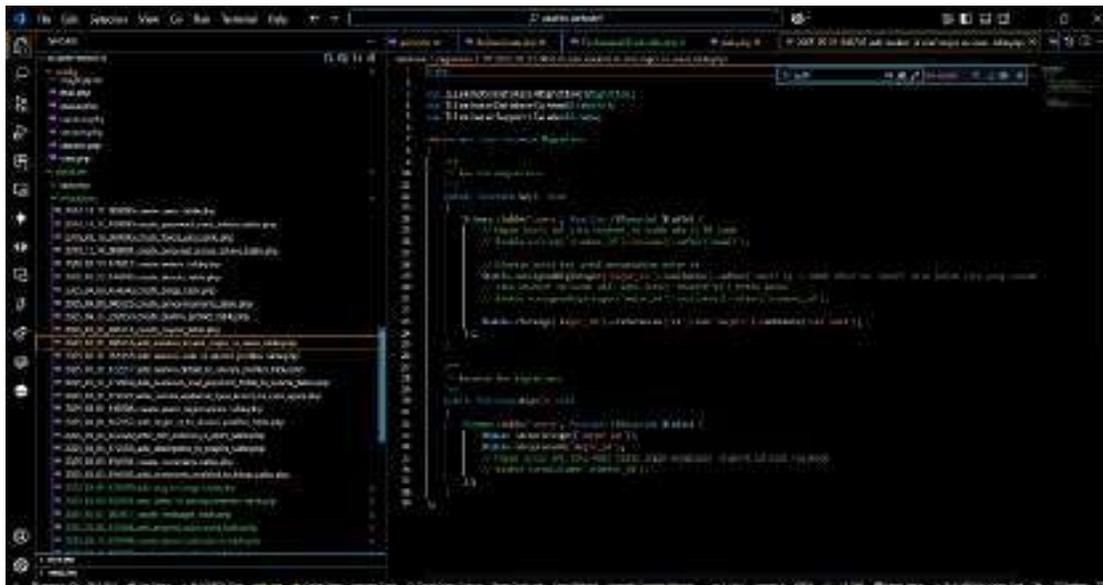
d. Migrasi Database Terkait (database/migrations/...)



Gambar 3. 256 Migrasi Tabel User



Gambar 3. 257 Migrasi Alter Role



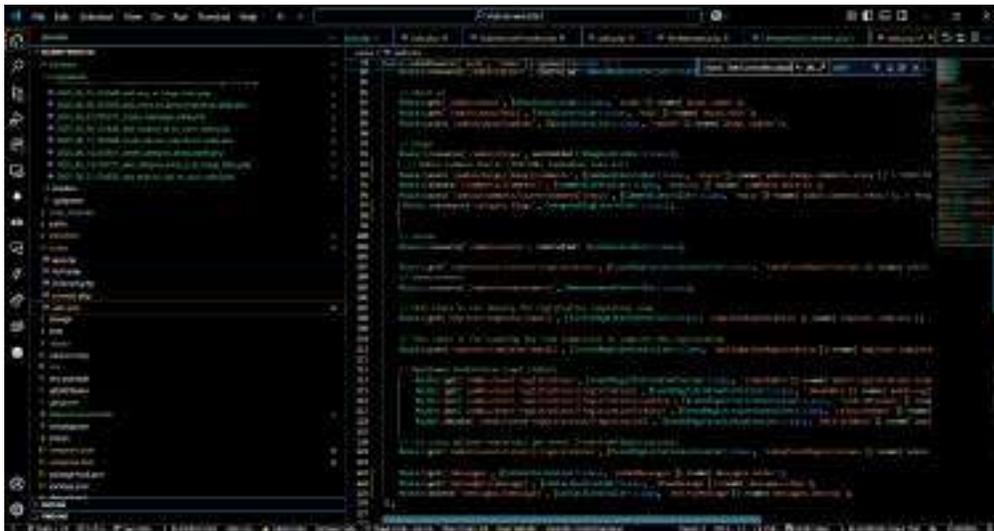
Gambar 3. 258 Migrasi Id Siswa dan Id Jurusan

Penjelasan: Cuplikan migrasi ini menunjukkan bagaimana tabel users, alumni_profiles, dan majors dibangun dan dimodifikasi. Ini mencakup penambahan kolom role, student_id, major_id pada tabel users, serta kolom alumni_code, graduation_year, dan detail karier pada

tabel alumni_profiles. Relasi foreign key juga didefinisikan untuk memastikan integritas data antar tabel.

10. F-ADM-007: Sistem Moderasi Konten & Komunikasi

a. Rute (routes/web.php)

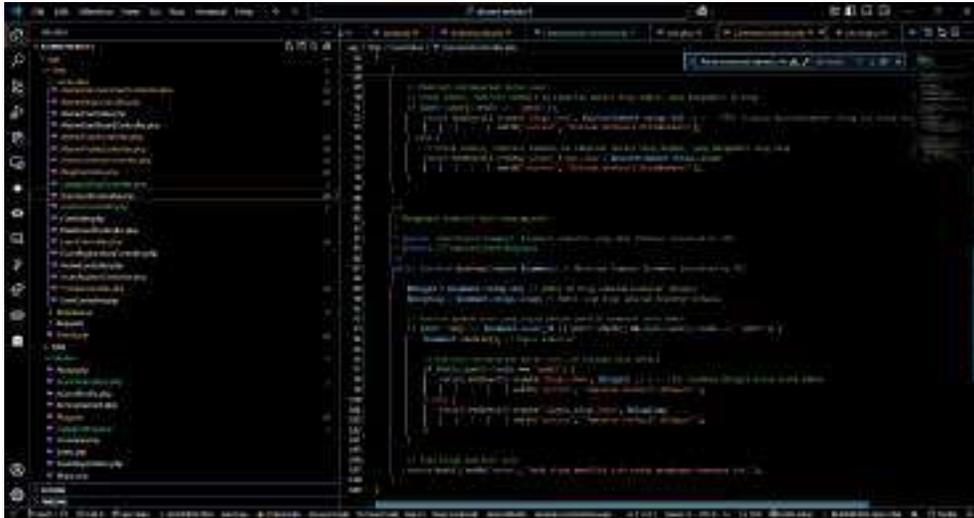


Gambar 3. 259 Rute Sistem Moderasi Konten & Komunikasi

Penjelasan: Admin memiliki kemampuan untuk **memoderasi komentar** melalui `Route::resource('/admin/comments', CommentController::class);`, yang memungkinkan mereka untuk melihat, mengelola, dan terutama **menghapus komentar** yang tidak pantas. Selain itu, sistem juga menyediakan modul moderasi khusus untuk **pendaftaran event**. Rute-rute seperti `/admin/event-registrations` memungkinkan admin untuk melihat daftar semua pendaftaran, memeriksa detail masing-masing pendaftaran, serta secara aktif **menyetujui (confirm)** atau **menolak (reject)** pendaftaran tersebut, khususnya yang melibatkan verifikasi pembayaran. Kemampuan untuk **menghapus (destroy)** pendaftaran event juga tersedia.

Dengan demikian, rute-rute ini secara komprehensif mendukung fungsi moderasi untuk komunikasi (komentar) dan konten (pendaftaran event).

- b. Logika (app/Http/Controllers/CommentController.php dan EventRegistrationController.php)

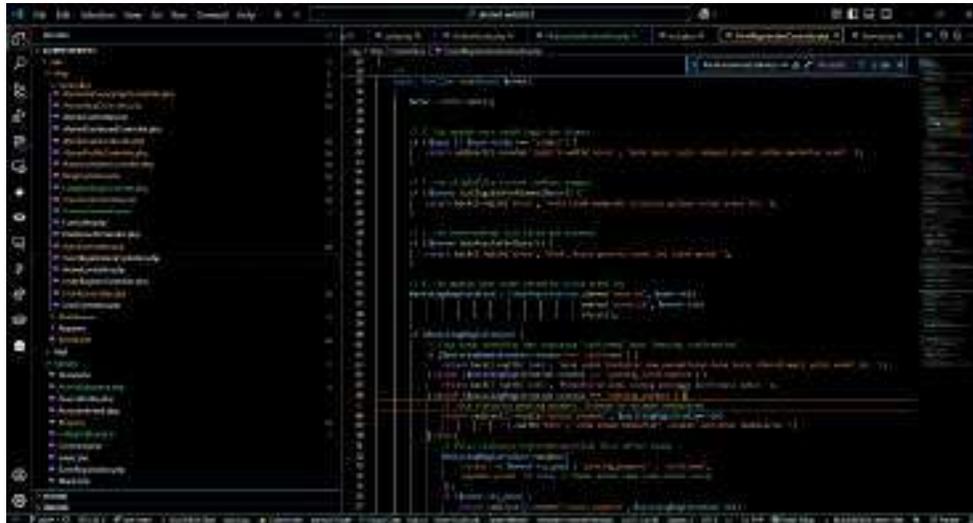


Gambar 3. 260 Logika Komentar

Penjelasan :

CommentController bertugas mengelola fungsionalitas komentar pada blog, dengan memastikan hanya pengguna yang sudah terautentikasi dapat berinteraksi. Metode store memungkinkan pengguna untuk menambahkan komentar baru ke sebuah blog tertentu, bahkan mendukung balasan terhadap komentar lain melalui `parent_id`. Setelah komentar berhasil disimpan, pengguna akan diarahkan kembali ke halaman detail blog yang relevan (baik untuk alumni menggunakan *slug* blog, maupun untuk admin menggunakan ID blog). Metode reply secara spesifik menangani penambahan balasan pada komentar yang sudah ada. Sementara itu, metode destroy memungkinkan pengguna untuk menghapus komentar mereka sendiri, atau admin untuk menghapus komentar mana pun, menjadikannya alat moderasi konten komunikasi

yang penting, sebelum kemudian mengarahkan kembali pengguna ke halaman blog yang bersangkutan.



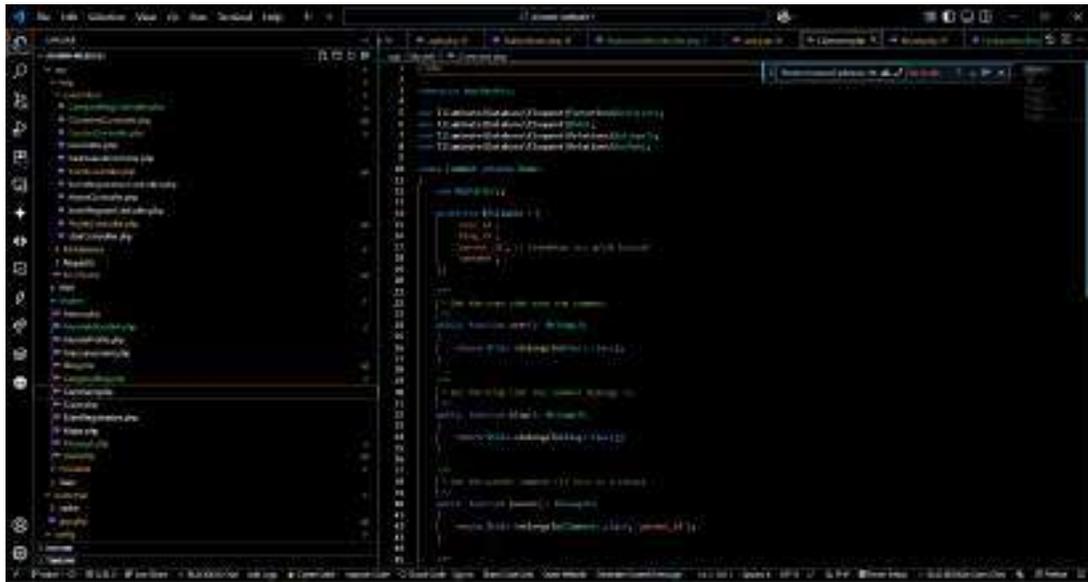
Gambar 3. 261 Logika Pendaftaran Event

Penjelasan:

EventRegistrationController mengelola seluruh alur **pendaftaran dan partisipasi event**, dengan memisahkan hak akses antara alumni dan admin melalui *middleware*. Untuk alumni, metode **rsvp** memungkinkan mereka mendaftar event setelah melewati berbagai validasi seperti status login, kelayakan audiens, dan ketersediaan slot. Jika event berbayar, alumni akan diarahkan ke metode **showPaymentForm** untuk melihat detail pembayaran, dan kemudian metode **uploadPaymentProof** untuk mengunggah bukti pembayaran. Di sisi admin, controller ini menyediakan modul moderasi yang komprehensif. Metode **indexAdmin** dan **showAdmin** memungkinkan admin untuk melihat dan memeriksa semua pendaftaran event, lengkap dengan filter dan pencarian. Metode **confirmPayment** dan **rejectPayment** adalah inti

moderasi, yang memungkinkan admin **menyetujui atau menolak** pendaftaran setelah meninjau bukti pembayaran. Terakhir, metode **destroyAdmin** memberikan kemampuan untuk **menghapus pendaftaran**, sementara `indexEventRegistrations` membantu admin meninjau pendaftar untuk event spesifik, secara keseluruhan mendukung manajemen penuh terhadap partisipasi event.

c. Model Terkait (`app/Models/Comment.php` dan `EventRegistration.php`)

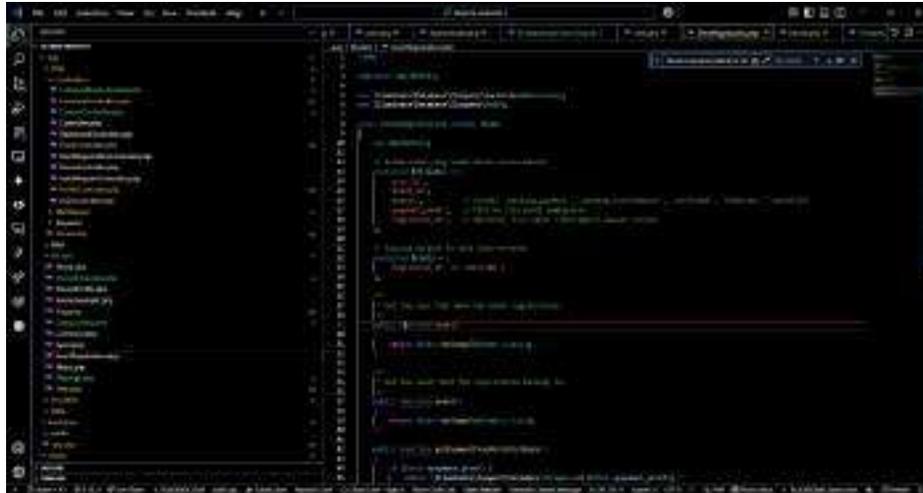


Gambar 3. 262 Model Komentar

Penjelasan :

Model Comment adalah fondasi untuk sistem komunikasi di platform Anda. Ia menyimpan isi komentar (`content`), siapa yang menulisnya (`user_id`), dan pada blog mana komentar itu berada (`blog_id`). Yang membedakannya adalah adanya kolom `parent_id`, yang memungkinkan komentar menjadi balasan dari komentar lain, menciptakan struktur percakapan hierarkis. Model ini memiliki relasi `BelongsTo` ke `User` dan `Blog` untuk mengidentifikasi penulis dan sumber komentar, serta relasi `BelongsTo` ke dirinya sendiri (`parent()`) untuk mengidentifikasi

komentar induk, dan HasMany ke dirinya sendiri (replies()) untuk mengumpulkan semua balasannya, memastikan moderasi dapat dilakukan pada setiap level percakapan.

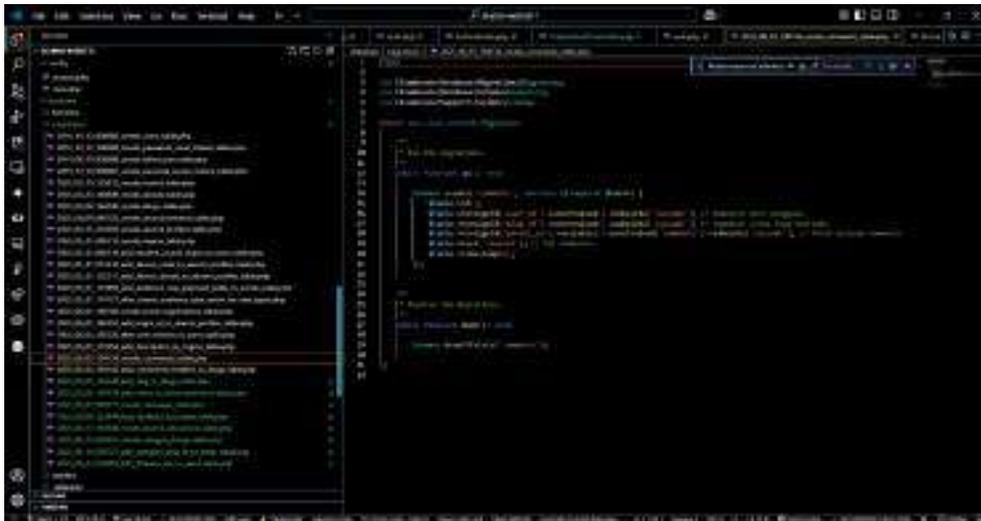


Gambar 3. 263 Model Pendaftaran Event

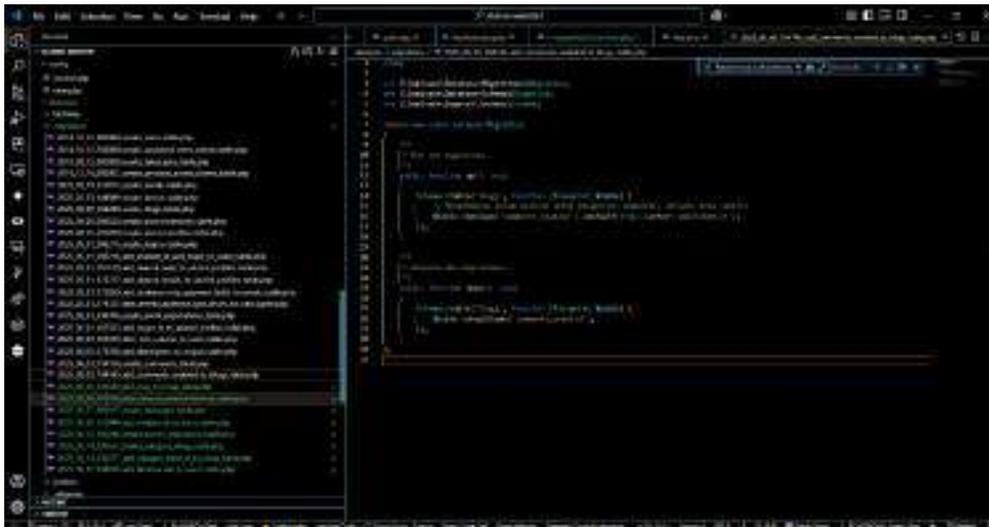
Penjelasan:

Model EventRegistration bertindak sebagai pencatat setiap pendaftaran pengguna untuk suatu acara. Kolom user_id dan event_id adalah kunci yang menghubungkan pendaftaran ini dengan pengguna yang mendaftar dan acara yang didaftar. Status pendaftaran (status), seperti 'pending_payment', 'confirmed', atau 'rejected', sangat penting untuk melacak kemajuan partisipasi. Untuk event berbayar, model ini juga menyimpan payment_proof (bukti pembayaran). Model ini dilengkapi dengan relasi BelongsTo ke User dan Event untuk memudahkan pengambilan data terkait, serta accessor getPaymentProofUrlAttribute yang otomatis menghasilkan URL untuk mengakses bukti pembayaran yang diunggah, mendukung proses moderasi pembayaran oleh admin.

d. Migrasi Database Terkait (database/migrations/...)



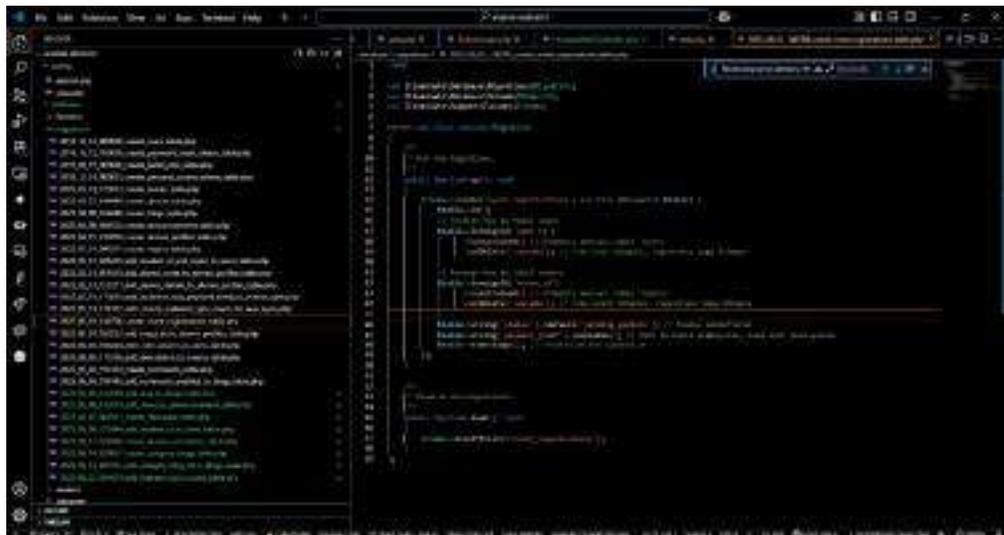
Gambar 3. 264 Migrasi Tabel Komentar



Gambar 3. 265 Migrasi Kolom Comments_enable

Penjelasan: Migrasi pertama menciptakan tabel **comments**, yang menyimpan setiap komentar dengan tautan ke pengguna (**user_id**), blog yang dikomentari (**blog_id**), dan bahkan komentar induk (**parent_id**) untuk mendukung balasan berhierarki, serta isi komentar

(content). Migrasi kedua kemudian memodifikasi tabel **blogs** yang sudah ada dengan menambahkan kolom `comments_enabled`. Kolom boolean ini, yang secara *default* bernilai `true`, berfungsi untuk mengontrol apakah fitur komentar diaktifkan atau dinonaktifkan untuk blog tertentu, memberikan fleksibilitas dalam pengelolaan interaksi konten.



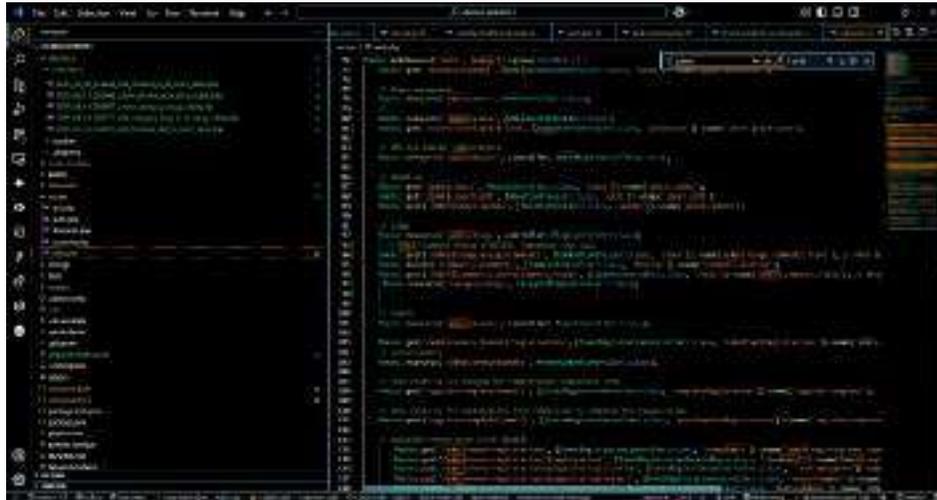
Gambar 3. 266 Migrasi Tabel Pendaftaran Event

Penjelasan: Migrasi Laravel ini bertugas **membuat tabel event_registrations**, yang berfungsi untuk mencatat setiap pendaftaran pengguna ke suatu acara. Tabel ini mencakup id unik, `user_id` dan `event_id` sebagai *foreign key* yang saling terkait dengan tabel `users` dan `events`, dengan pengaturan `onDelete('cascade')` yang berarti pendaftaran akan otomatis terhapus jika pengguna atau acara terkait dihapus. Selain itu, terdapat kolom `status` untuk melacak status pendaftaran (misalnya, `'pending_payment'` sebagai nilai *default*), serta `payment_proof` yang bersifat *nullable* untuk menyimpan jalur bukti pembayaran bagi acara berbayar, dan juga *timestamps* untuk mencatat waktu pembuatan dan pembaruan pendaftaran.

7. Realisasi Backend 2 (Noval Lias Ramadani)

1. F-ADM-002: Manajemen Data Profil Alumni

a. Rute Manajemen Profil Alumni oleh Admin (routes/web.php)



Gambar 3. 267 Rute Manajemen Data Profil Alumni

Penjelasan: Cuplikan ini menunjukkan rute `Route::resource('/admin/users', AdminUserController::class)` yang memberikan serangkaian rute RESTful untuk mengelola pengguna (termasuk alumni) oleh Admin. Rute ini secara otomatis mencakup:

GET `/admin/users` (index): Menampilkan daftar semua pengguna.

GET `/admin/users/create` (create): Menampilkan form tambah pengguna baru.

POST `/admin/users` (store): Menyimpan pengguna baru.

GET `/admin/users/{user}` (show): Menampilkan detail satu pengguna.

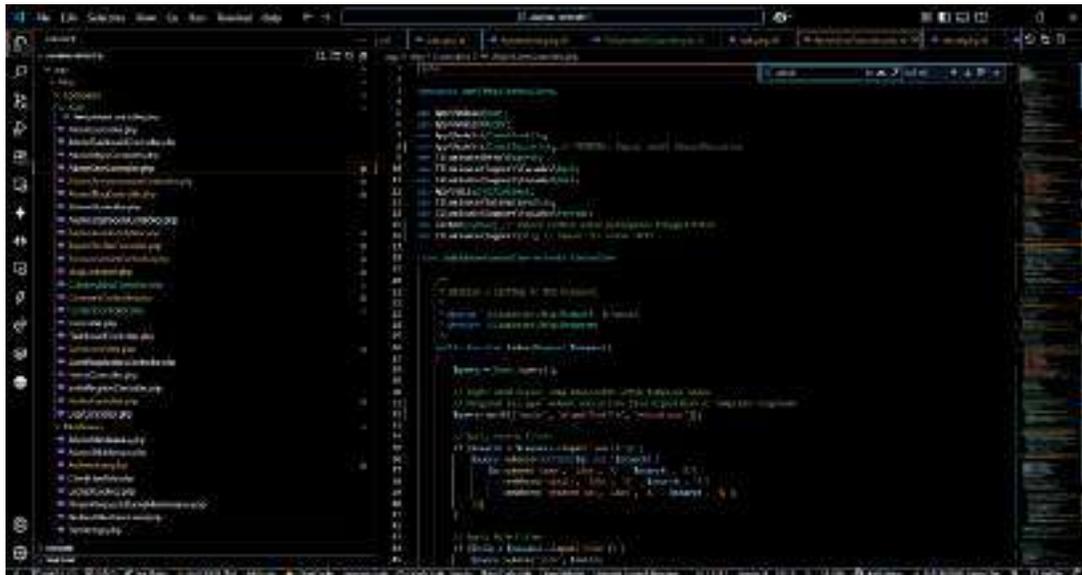
GET `/admin/users/{user}/edit` (edit): Menampilkan form edit pengguna.

PUT/PATCH `/admin/users/{user}` (update): Memperbarui pengguna.

DELETE /admin/users/{user} (destroy): Menghapus pengguna. Semua rute ini dilindungi oleh middleware auth dan admin, memastikan hanya admin yang sudah login yang dapat mengaksesnya. Rute users.print-card juga tersedia untuk mencetak kartu alumni.

b. Logika Manajemen Profil Alumni oleh Admin

(app/Http/Controllers/AdminUserController.php)



Gambar 3. 268 Logika Manajemen Profil Alumni Ole Admin

Penjelasan:

Metode index(): Menampilkan daftar pengguna (termasuk alumni) dengan dukungan filter dan pencarian, dan melakukan eager loading relasi major, alumniProfile, dan educations.

Metode create() dan store(): Menampilkan formulir dan memproses penambahan pengguna baru. Ini mencakup validasi yang komprehensif, pembuatan User dengan peran yang ditentukan, dan secara kondisional pembuatan AlumniProfile serta penanganan riwayat

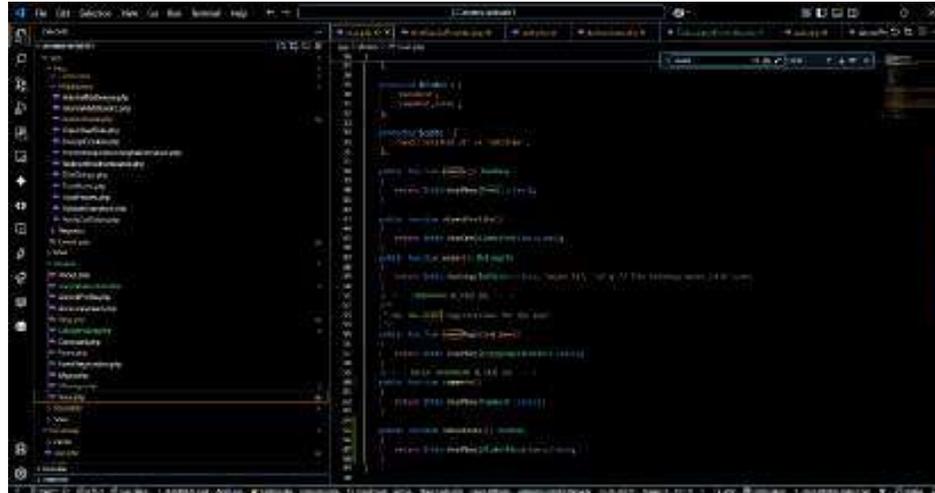
pendidikan (AlumniEducation) jika peran yang dipilih adalah 'alumni'. Logika alumni_code otomatis juga diterapkan.

Metode show(): Menampilkan detail lengkap satu pengguna, memuat semua relasi terkait (profil alumni, jurusan, riwayat pendidikan).

Metode edit() dan update(): Menampilkan formulir dan memproses pembaruan data pengguna. Bagian ini sangat krusial karena mendukung pembaruan data dasar pengguna, detail profil alumni (termasuk upload/hapus gambar dan generate ulang alumni_code jika diperlukan), serta manajemen dinamis riwayat pendidikan (memperbarui yang ada, menambah yang baru, dan menghapus yang tidak lagi ada di form). Validasi unik untuk email dan student_id diabaikan untuk pengguna yang sedang diedit.

Metode destroy(): Menghapus pengguna dan secara berjenjang menghapus profil alumni, gambar profil, dan semua riwayat pendidikan terkait.

- c. Model Terkait (app/Models/User.php, app/Models/AlumniProfile.php, app/Models/AlumniEducation.php, app/Models/Major.php)



```
class User {
    public $table = 'users';
    public $primaryKey = 'id';
    public $timestamps = true;
    public $fillable = ['name', 'email', 'password'];

    public function setPassword($password) {
        $password = bcrypt($password);
        $this->password = $password;
    }

    public function setPasswordAttribute($password) {
        $this->password = bcrypt($password);
    }

    public function email($email) {
        $this->email = strtolower($email);
    }

    public function emailAttribute($email) {
        $this->email = strtolower($email);
    }

    public function name($name) {
        $this->name = trim($name);
    }

    public function nameAttribute($name) {
        $this->name = trim($name);
    }

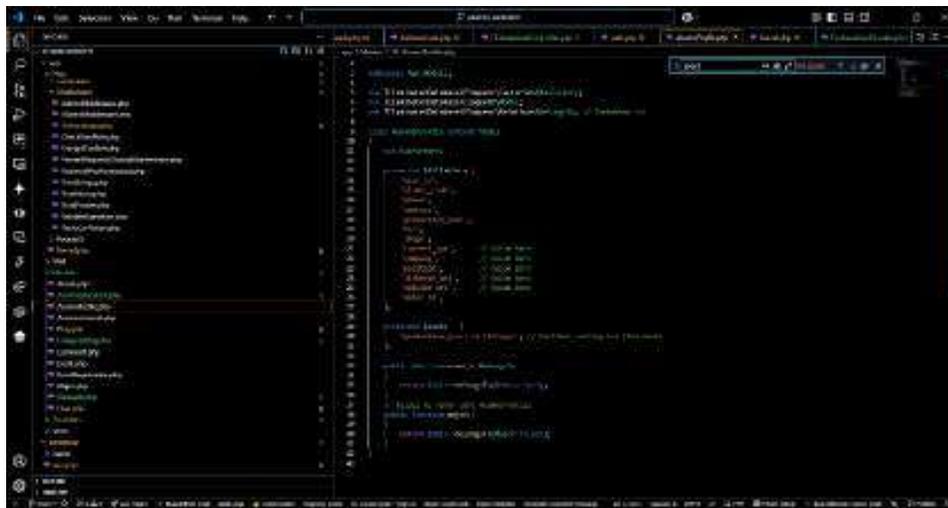
    public function isActive() {
        return $this->status == 'active';
    }

    public function activate() {
        $this->status = 'active';
    }

    public function deactivate() {
        $this->status = 'inactive';
    }

    public function deactivateAttribute($status) {
        $this->status = 'inactive';
    }
}
```

Gambar 3. 269 Model User



```
class AlumniProfile {
    public $table = 'alumni_profiles';
    public $primaryKey = 'id';
    public $timestamps = true;
    public $fillable = ['alumni_id', 'name', 'email', 'phone', 'address'];

    public function alumni() {
        return $this->belongsTo('User', 'alumni_id');
    }

    public function name($name) {
        $this->name = trim($name);
    }

    public function nameAttribute($name) {
        $this->name = trim($name);
    }

    public function email($email) {
        $this->email = strtolower($email);
    }

    public function emailAttribute($email) {
        $this->email = strtolower($email);
    }

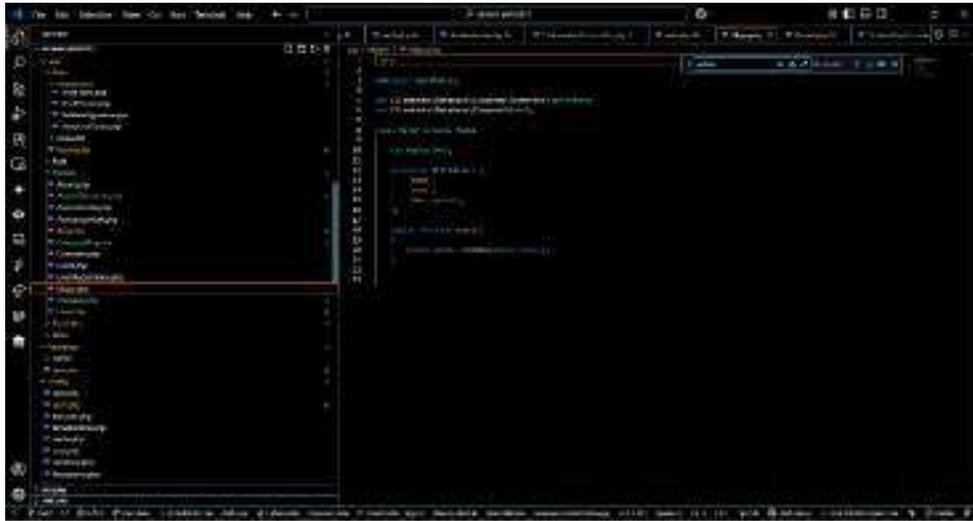
    public function phone($phone) {
        $this->phone = trim($phone);
    }

    public function phoneAttribute($phone) {
        $this->phone = trim($phone);
    }

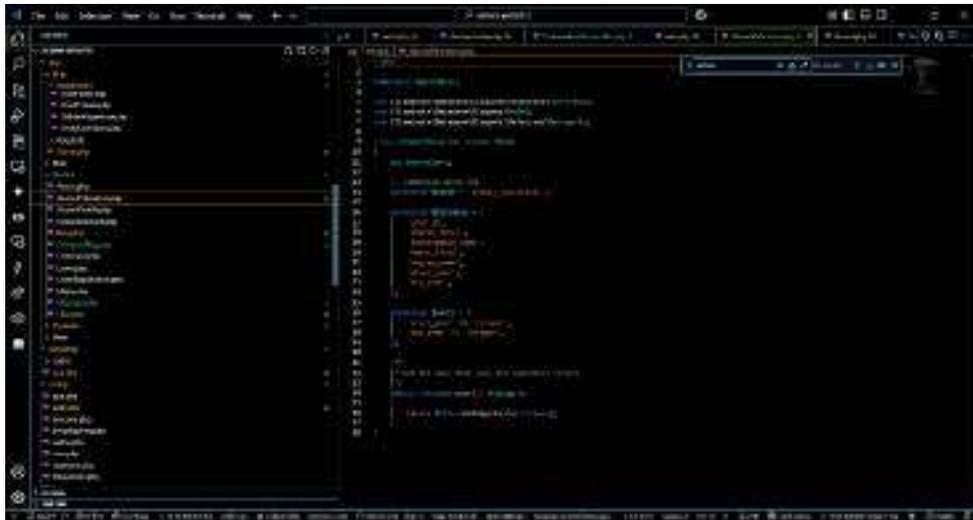
    public function address($address) {
        $this->address = trim($address);
    }

    public function addressAttribute($address) {
        $this->address = trim($address);
    }
}
```

Gambar 3. 270 Model Profil Alumni



Gambar 3. 271 Model Jurusan



Gambar 3. 272 Model Alumni Profil

Penjelasan:

User Model: Relasi hasOne ke AlumniProfile dan hasMany ke AlumniEducation memungkinkan kontroler untuk mengakses dan memanipulasi data profil dan riwayat pendidikan yang terkait dengan setiap pengguna. Kolom student_id dan major_id juga tersedia.

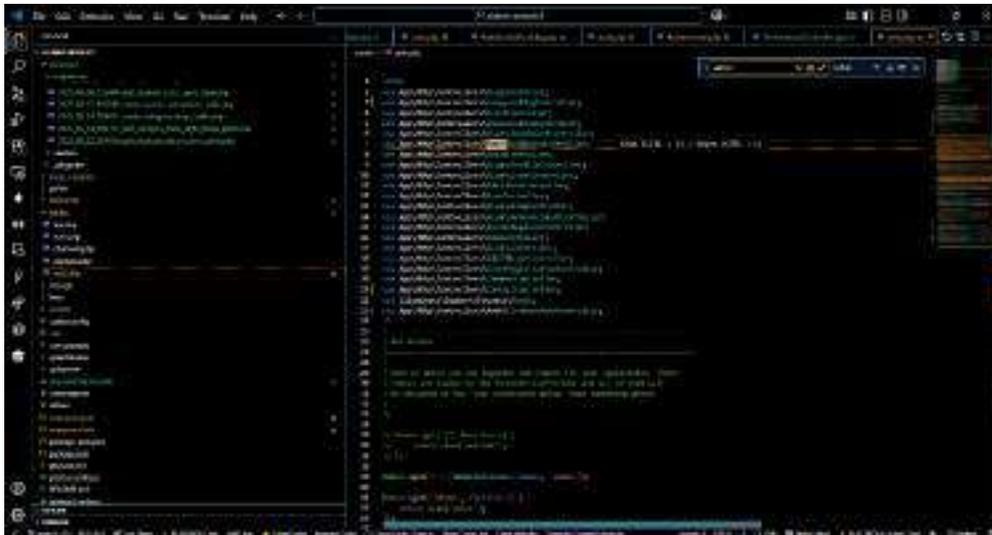
AlumniProfile Model: Menyimpan semua detail profil spesifik alumni, termasuk informasi karier (current_job, company, position, linkedin_url, website_url) dan alumni_code. Relasi belongsTo ke User dan Major serta hasMany ke AlumniEducation memastikan struktur data yang terhubung.

AlumniEducation Model: Merepresentasikan satu entri riwayat pendidikan. Kolom-kolomnya menyimpan detail seperti jenjang, institusi, jurusan, dan tahun masuk/lulus. Relasi belongsTo ke User menghubungkannya dengan pemilik profil.

Major Model: Menyimpan data jurusan dan digunakan dalam dropdown untuk pemilihan jurusan baik di profil pengguna maupun alumni.

C. F-ADM-003: Dashboard & Pelaporan Lanjutan

a. Rute Dashboard Admin (routes/web.php)

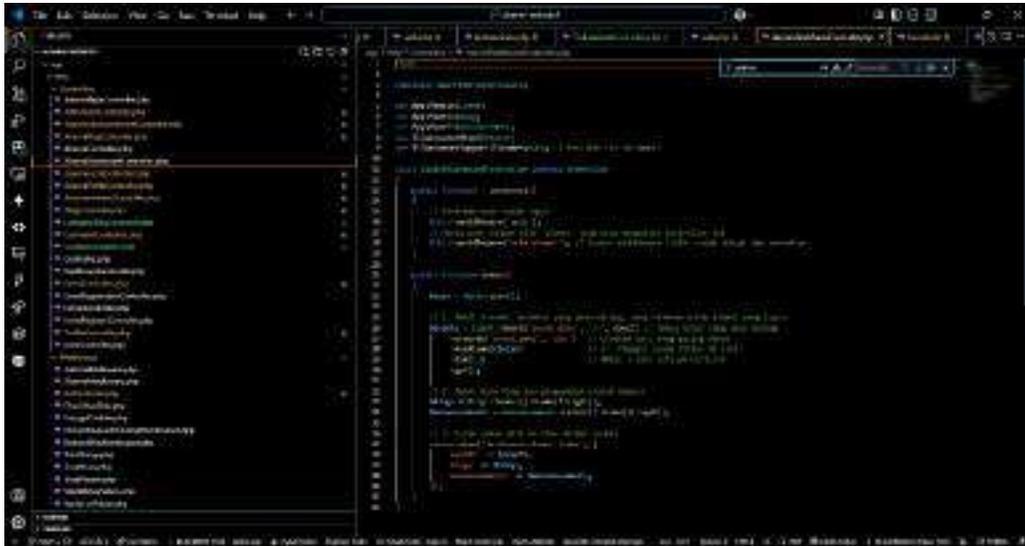
A screenshot of a code editor with a dark theme. The left sidebar shows a file explorer with a tree view of a project structure. The main editor area displays PHP code for routes/web.php. The code includes several route definitions for an admin dashboard, such as 'admin.dashboard.index' and 'admin.dashboard.create'. The code is written in a standard PHP syntax with comments and function calls. The editor has a search bar at the top right and a status bar at the bottom.

Gambar 3. 273 Potongan kode routes/web.php untuk dashboard admin

Penjelasan:

Cuplikan ini menunjukkan rute /dashboard/admin yang mengarahkan ke metode index pada AdminDashboardController. Rute ini dilindungi oleh middleware auth dan admin, memastikan hanya pengguna yang sudah login dengan peran 'admin' yang dapat mengakses dashboard ini.

b. Logika Dashboard Admin (app/Http/Controllers/AdminDashboardController.php)



Gambar 3. 274 app/Http/Controllers/AdminDashboardController.php

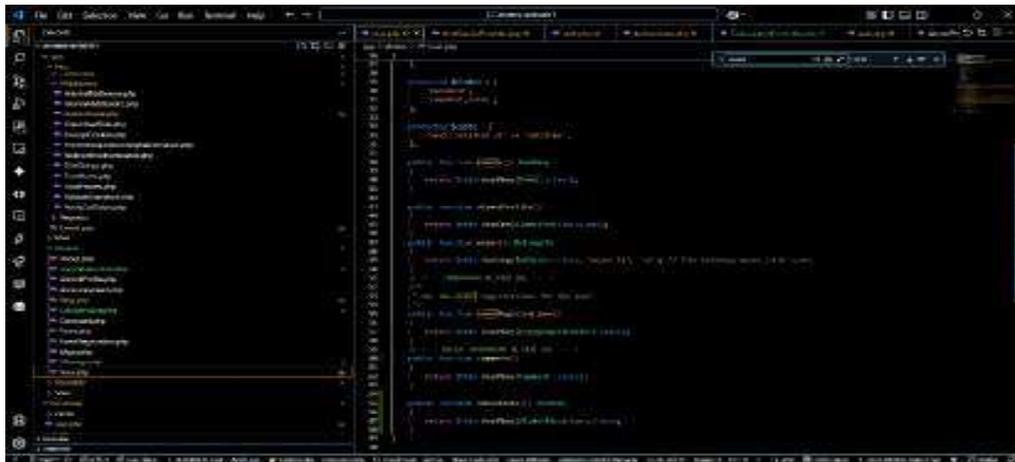
Penjelasan:

Metode index pada AdminDashboardController adalah inti dari fungsionalitas ini.

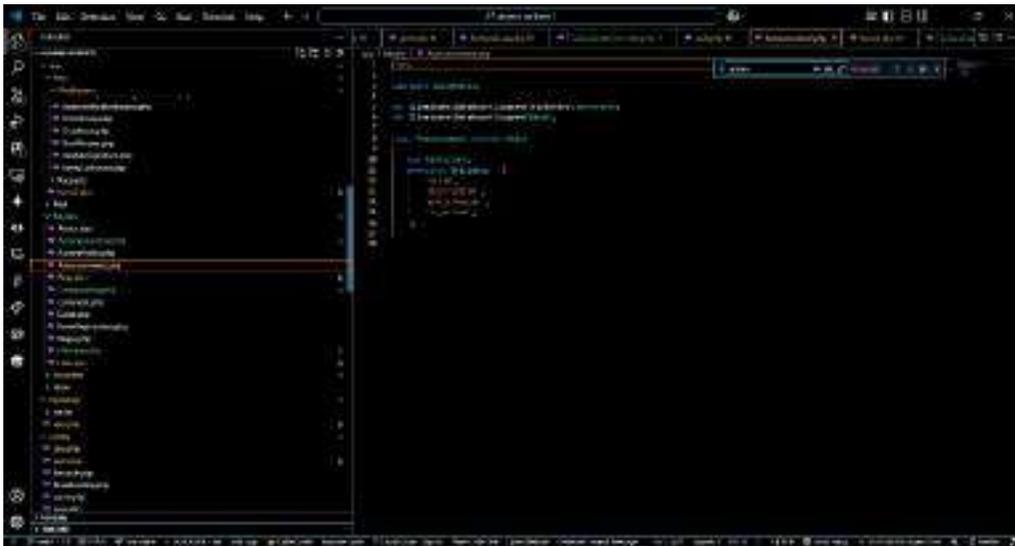
Ia mengambil jumlah total record dari model User, Blog, Event, dan Announcement menggunakan metode count() untuk ditampilkan sebagai statistik. Ia juga mengambil satu record terbaru dari Blog, Event, dan Announcement menggunakan kombinasi latest()->first()

untuk ditampilkan sebagai "item terbaru" di dashboard.Semua data ini kemudian diteruskan ke view dashboard.admin.index untuk ditampilkan.

- c. Model Terkait (app/Models/User.php, app/Models/Blog.php, app/Models/Event.php, app/Models/Announcement.php)



Gambar 3. 275 Potongan kode app/Models/User.php

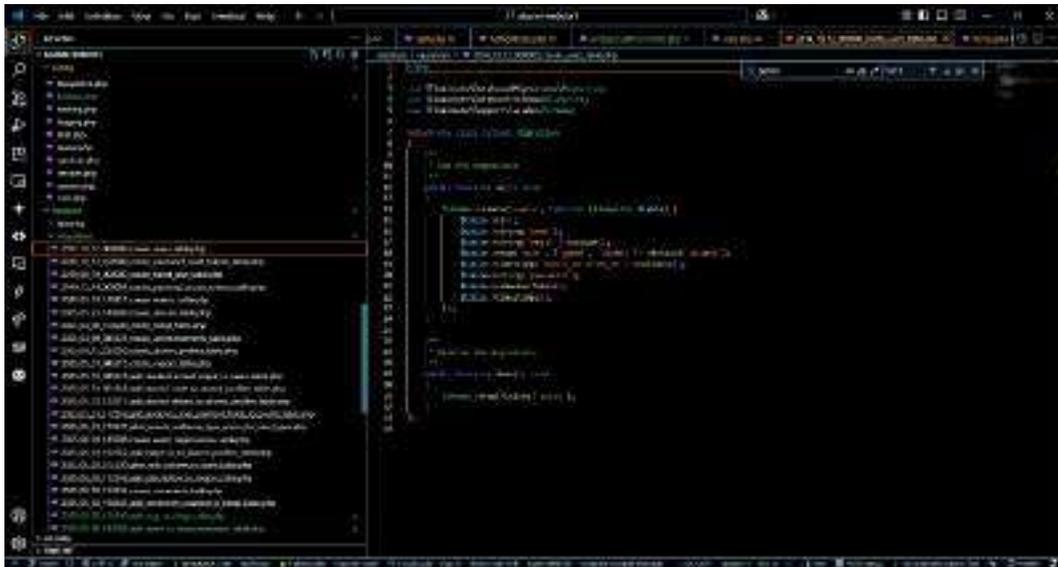


Gambar 3. 276 Potongan kode app/Models/Announcement.php

Penjelasan:

Model-model ini adalah fondasi data untuk dashboard. Penggunaan trait HasFactory pada setiap model memungkinkan penggunaan metode Eloquent seperti `count()` untuk menghitung total record, dan `latest()->first()` untuk mengambil record terbaru. Masing-masing model merepresentasikan tabel yang datanya dihitung dan ditampilkan di dashboard.

d. Migrasi Database Terkait (database/migrations/...)



Gambar 3. 277 Potongan kode migrasi database untuk membuat tabel

Penjelasan:

Cuplikan migrasi ini menunjukkan keberadaan tabel users, blogs, events, dan announcements di database. Struktur tabel ini menyediakan data mentah yang kemudian diolah dan dihitung oleh kontroler untuk disajikan di dashboard admin.

D. F-ADM-004: Manajemen Konten Umum (Berita, Artikel, Pengumuman)

a. Rute Manajemen Konten Admin (routes/web.php)



```
11 use Illuminate\Support\Facades\Route;
12
13 Route::group(['prefix' => 'admin', 'middleware' => 'auth:admin'], function () {
14     // Blogs
15     Route::resource('blogs', Admin\BlogsController::class);
16
17     // Announcements
18     Route::resource('announcements', Admin\AnnouncementsController::class);
19
20     // Comments
21     Route::group(['prefix' => 'blogs', 'middleware' => 'auth:admin'], function () {
22         Route::post('comments/store', Admin\BlogsCommentsController::store);
23         Route::delete('comments/destroy/{comment}', Admin\BlogsCommentsController::destroy);
24         Route::post('comments/reply/{comment}', Admin\BlogsCommentsController::reply);
25     });
26 });
```

Gambar 3. 278 Potongan kode di bagian routes/web.php untuk manajemen konten di bagian admin

Penjelasan:

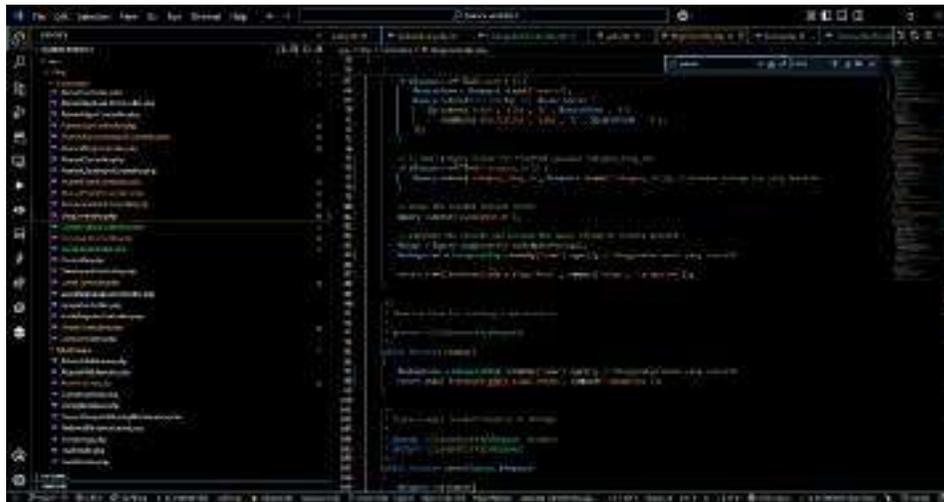
Cuplikan ini menunjukkan rute-rute utama untuk manajemen konten oleh Admin.

Route::resource('/admin/blogs', ...) menyediakan rute CRUD lengkap untuk entitas Blog (artikel). Route::resource('/admin/announcements', ...) menyediakan rute CRUD lengkap untuk entitas Pengumuman.

Rute tambahan untuk komentar (admin.blogs.comments.store, comments.destroy, admin.comments.reply) memungkinkan admin untuk membuat, menghapus, dan membalas komentar langsung dari sisi admin.

Semua rute ini dilindungi oleh middleware auth dan admin, memastikan hanya pengguna yang sudah login dengan peran 'admin' yang dapat mengakses fungsi-fungsi manajemen konten ini.

b. Logika Manajemen Blog (app/Http/Controllers/BlogController.php)



Gambar 3. 279 Potongan kode untuk logika Manajemen Blog

Penjelasan:

Konstruktor: Memastikan hanya admin yang dapat mengakses controller ini untuk CRUD blog.

index(): Menampilkan daftar blog untuk admin, dengan fungsionalitas pencarian berdasarkan judul/deskripsi dan filter berdasarkan kategori. Blog diurutkan dari yang terbaru.

create() & store(): Menampilkan form dan memproses penambahan blog baru. Validasi ketat diterapkan pada input, termasuk judul unik, deskripsi, gambar (opsional), tanggal publikasi, status comments_enabled (dari checkbox), dan kategori. slug blog dibuat otomatis dari judul.

Gambar diunggah dan disimpan di storage.

show(): Menampilkan detail blog. Admin dapat melihat blog meskipun belum dipublikasikan.

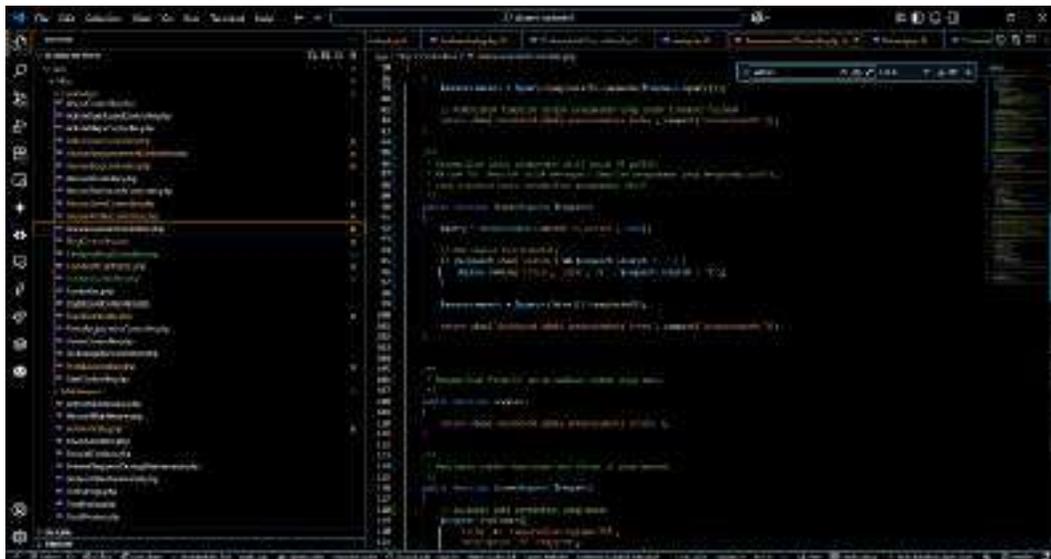
Komentar juga dimuat untuk ditampilkan.

edit() & update(): Menampilkan form dan memproses pembaruan blog. Validasi memastikan judul tetap unik (kecuali untuk blog itu sendiri). Logika penanganan gambar memungkinkan penggantian atau penghapusan gambar yang ada. Status comments_enabled dan category_blog_id juga diperbarui.

destroy(): Menghapus blog dan gambar terkait dari storage.

c. Logika Manajemen Pengumuman

(app/Http/Controllers/AnnouncementController.php)



Gambar 3. 280 Potongan kode logika Manajemen Pengumuman

Penjelasan:

Konstruktor: Memastikan hanya admin yang dapat mengakses controller ini untuk CRUD pengumuman.

index(): Menampilkan daftar pengumuman untuk admin, dengan fungsionalitas pencarian berdasarkan judul/deskripsi/tanggal publikasi dan filter berdasarkan status `is_active` (aktif/tidak aktif). Hasilnya dapat diurutkan dan di-paginate.

create() & store(): Menampilkan form dan memproses penambahan pengumuman baru. Validasi diterapkan pada judul, deskripsi, tanggal publikasi (opsional), dan status `is_active` (dari checkbox).

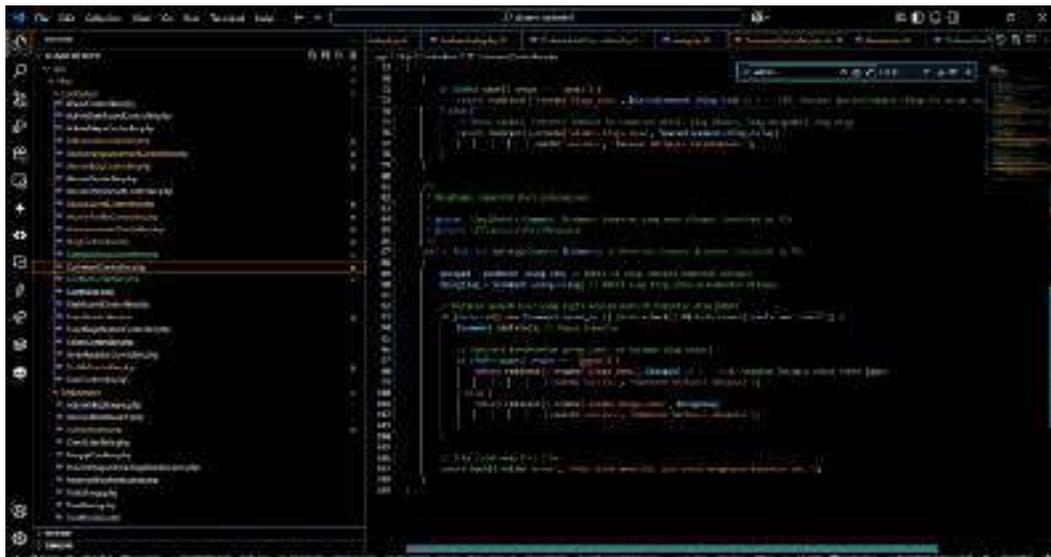
show(): Menampilkan detail pengumuman.

edit() & update(): Menampilkan form dan memproses pembaruan pengumuman. Validasi dan pembaruan data mirip dengan `store()`.

destroy(): Menghapus pengumuman.

d. Logika Manajemen Komentar oleh Admin

(`app/Http/Controllers/CommentController.php`)

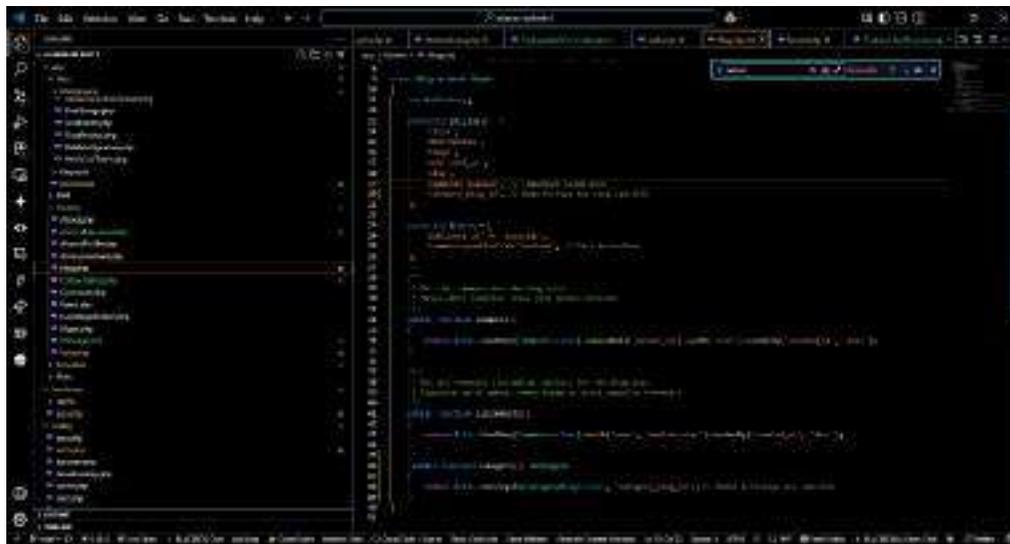


Gambar 3. 281 Potongan kode untuk Logika Manajemen Komentar oleh Admin

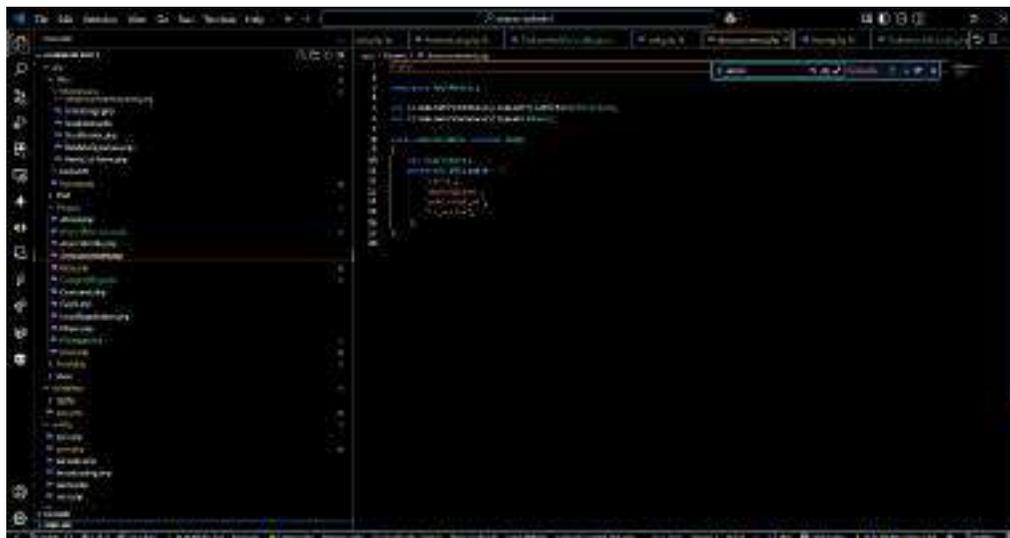
Penjelasan:

CommentController menyediakan fungsionalitas inti untuk mengelola komentar. Meskipun metode store dan reply digunakan oleh alumni, Admin memiliki akses untuk menggunakannya melalui rute admin yang didefinisikan. Metode destroy memungkinkan Admin (selain pemilik komentar) untuk menghapus komentar, yang merupakan bagian dari moderasi konten.

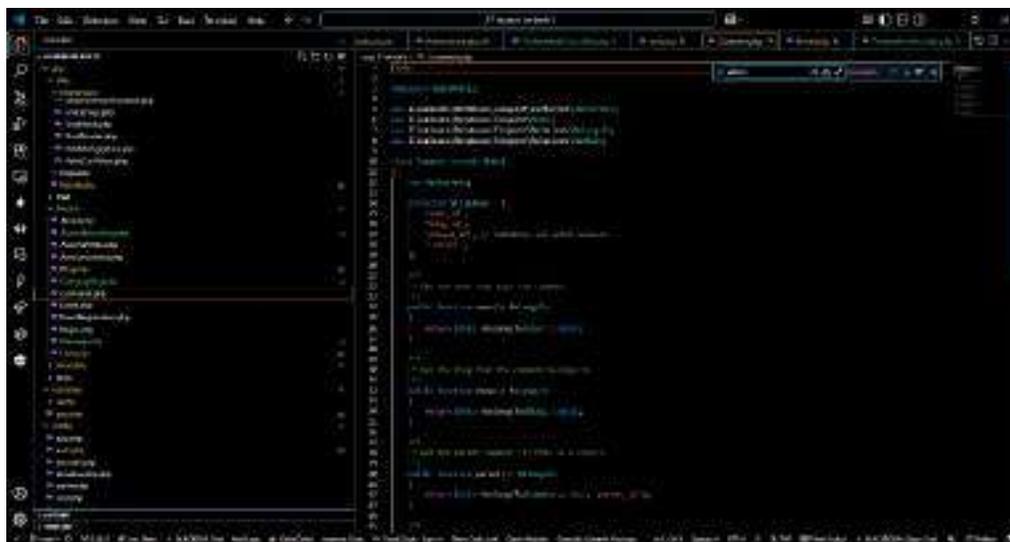
- e. Model Terkait (app/Models/Blog.php, app/Models/Announcement.php, app/Models/Comment.php, app/Models/CategoryBlog.php)



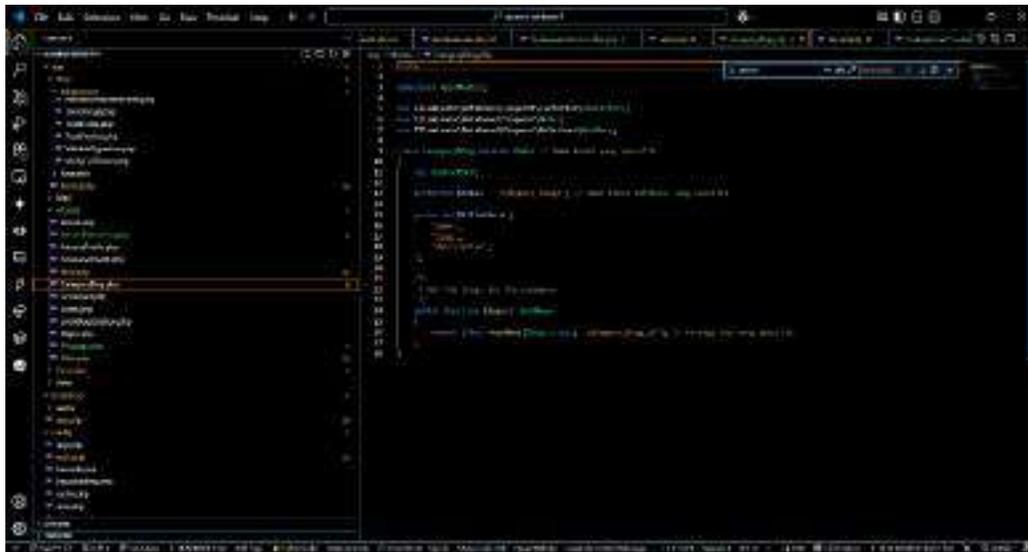
Gambar 3. 282 Potongan kode app/Models/Blog.php



Gambar 3. 283 Potongan kode app/Models/Announcement.php



Gambar 3. 284 Potongan kode app/Models/Comment.php



Gambar 3. 285 Potongan kode app/Models/CategoryBlog.php

Penjelasan:

Blog Model: Mendefinisikan atribut blog, termasuk `comments_enabled` (boolean yang menentukan apakah komentar diizinkan) dan `slug` (untuk URL). Memiliki relasi `hasMany` ke `Comment` dan `belongsTo` ke `CategoryBlog`.

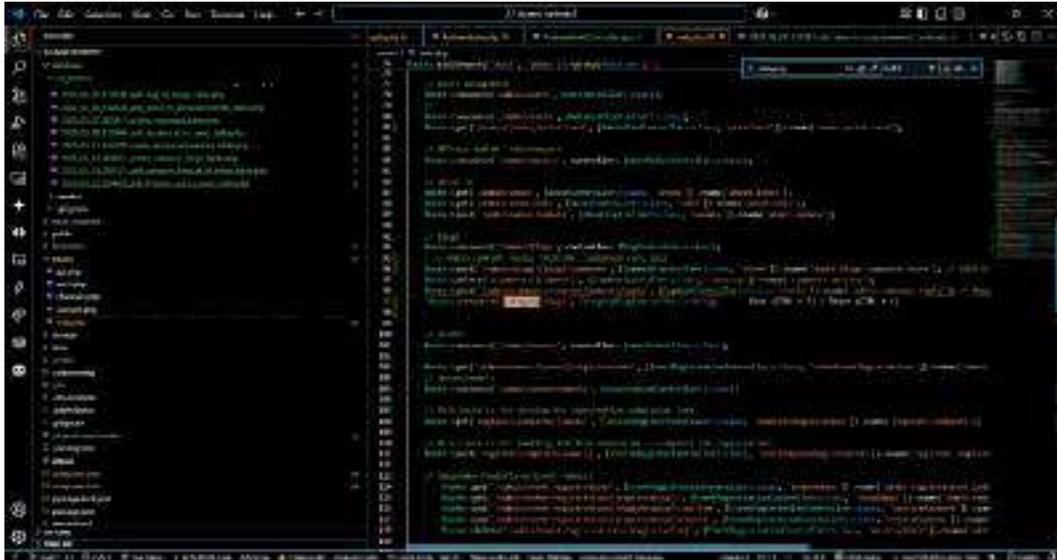
Announcement Model: Mendefinisikan atribut pengumuman, termasuk `is_active` (status publikasi) dan `views` (counter untuk jumlah tayangan).

Comment Model: Merepresentasikan struktur komentar, dengan `parent_id` untuk dukungan balasan, dan relasi `belongsTo` ke `User` dan `Blog`.

CategoryBlog Model: Mendefinisikan kategori untuk blog, dengan `name` dan `slug`. Memiliki relasi `hasMany` ke `Blog`.

E. F-ADM-005: Pengelolaan Kategori & Tag Dinamis

a. Rute Manajemen Kategori Blog Admin (routes/web.php)



Gambar 3. 286 Potongan kode untuk route/web.php kategori blog bagian admin

Penjelasan: Cuplikan ini menunjukkan rute `Route::resource('category-blogs', CategoryBlogController::class)` yang menyediakan serangkaian rute RESTful untuk mengelola kategori blog. Rute ini secara otomatis mencakup:

GET /category-blogs (index): Menampilkan daftar semua kategori blog.

GET /category-blogs/create (create): Menampilkan form tambah kategori baru.

POST /category-blogs (store): Menyimpan kategori baru.

GET /category-blogs/{category-blog} (show): Menampilkan detail satu kategori.

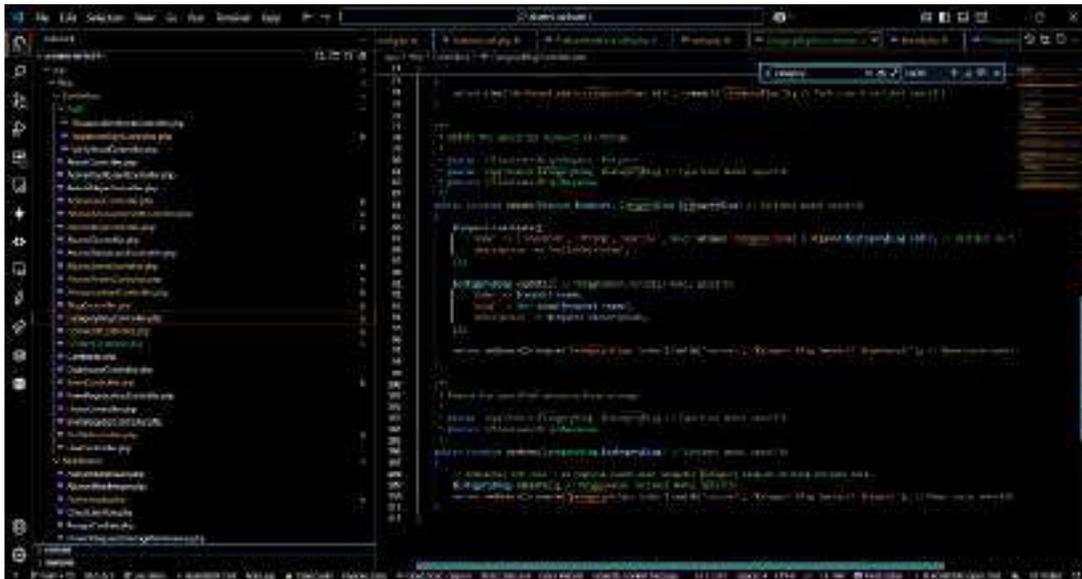
GET /category-blogs/{category-blog}/edit (edit): Menampilkan form edit kategori.

PUT/PATCH /category-blogs/{category-blog} (update): Memperbarui kategori.

DELETE /category-blogs/{category-blog} (destroy): Menghapus kategori. Semua rute ini dilindungi oleh middleware auth dan admin, memastikan hanya admin yang sudah login yang dapat mengaksesnya.

b. Logika Manajemen Kategori Blog

(app/Http/Controllers/CategoryBlogController.php)



Gambar 3. 287 Potongan kode app/Http/Controllers/CategoryBlogController.php

Penjelasan:

Konstruktor: Memastikan hanya admin yang dapat mengakses controller ini untuk CRUD kategori blog.

index(): Mengambil semua kategori blog, mengurutkannya berdasarkan nama, dan memaginasinya hasilnya. Data ini kemudian dikirim ke view untuk ditampilkan dalam daftar.

create() & store(): Menampilkan formulir untuk menambahkan kategori baru dan memproses penyimpanannya. Validasi memastikan nama kategori unik. slug kategori dibuat otomatis dari namanya menggunakan Str::slug().

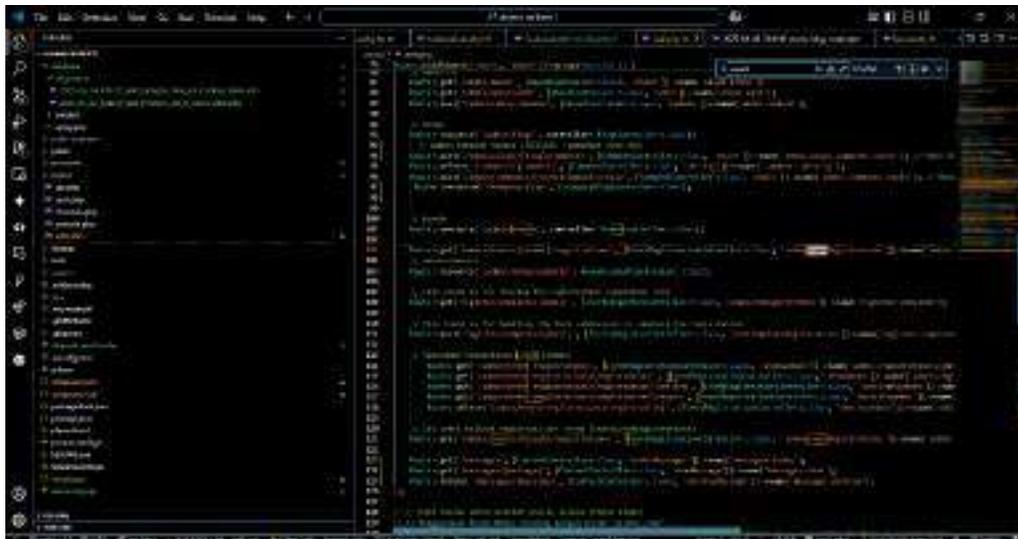
show(): Menampilkan detail satu kategori.

edit() & update(): Menampilkan formulir dan memproses pembaruan kategori. Validasi memastikan nama kategori tetap unik (kecuali untuk kategori yang sedang diedit). slug juga diperbarui jika nama berubah.

destroy(): Menghapus kategori blog. Berkat onDelete('set null') pada foreign key di tabel blogs, setiap blog yang terhubung dengan kategori yang dihapus akan memiliki category_blog_id menjadi NULL, menjaga integritas data tanpa menghapus blog terkait.

F. F-ADM-008: Manajemen Event Komprehensif

a. Rute Manajemen Event Admin (routes/web.php)



Gambar 3. 288 Potongan kode untuk manajemen event admin di bagian routes/web.php

Penjelasan: Cuplikan ini menunjukkan rute `Route::resource('/admin/events', EventController::class)` yang menyediakan serangkaian rute RESTful untuk mengelola event.

Rute ini secara otomatis mencakup:

GET `/admin/events` (index): Menampilkan daftar semua event.

GET `/admin/events/create` (create): Menampilkan form tambah event baru.

POST `/admin/events` (store): Menyimpan event baru.

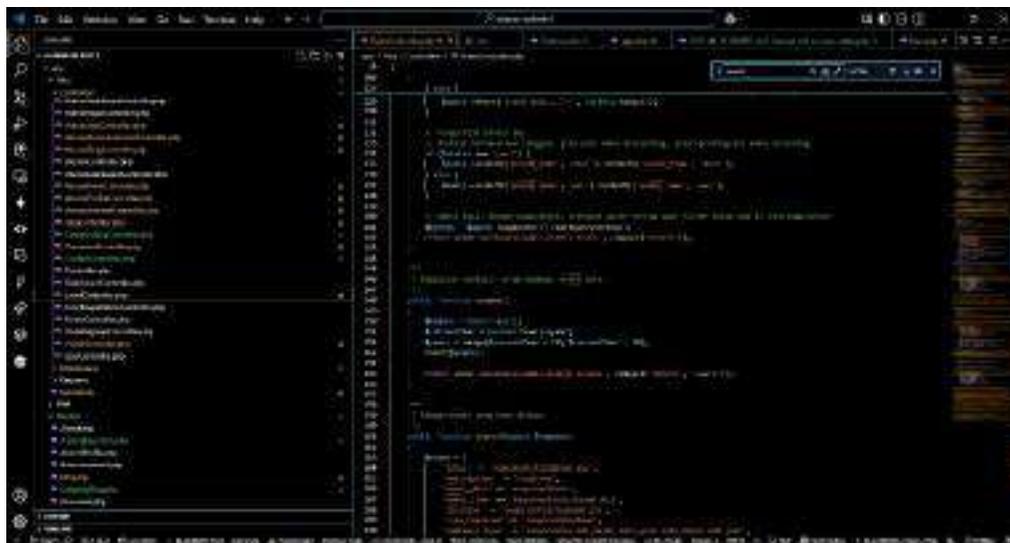
GET `/admin/events/{event}` (show): Menampilkan detail satu event.

GET `/admin/events/{event}/edit` (edit): Menampilkan form edit event.

PUT/PATCH `/admin/events/{event}` (update): Memperbarui event.

DELETE `/admin/events/{event}` (destroy): Menghapus event. Semua rute ini dilindungi oleh middleware `auth` dan `admin`, memastikan hanya admin yang sudah login yang dapat mengaksesnya.

b. Logika Manajemen Event (`app/Http/Controllers/EventController.php`)



Gambar 3. 289 Potongan kode untuk logika manajemen event

Konstruktor: Memastikan hanya admin yang dapat mengakses controller ini untuk CRUD event.

index(): Menampilkan daftar event untuk admin, dengan dukungan pencarian berdasarkan judul/deskripsi/lokasi dan berbagai filter: status (mendatang/selesai), tipe event (berbayar/gratis), tipe audiens (all, major_only, year_only, major_and_year), target jurusan, dan target tahun angkatan. Hasil diurutkan dan di-paginate.

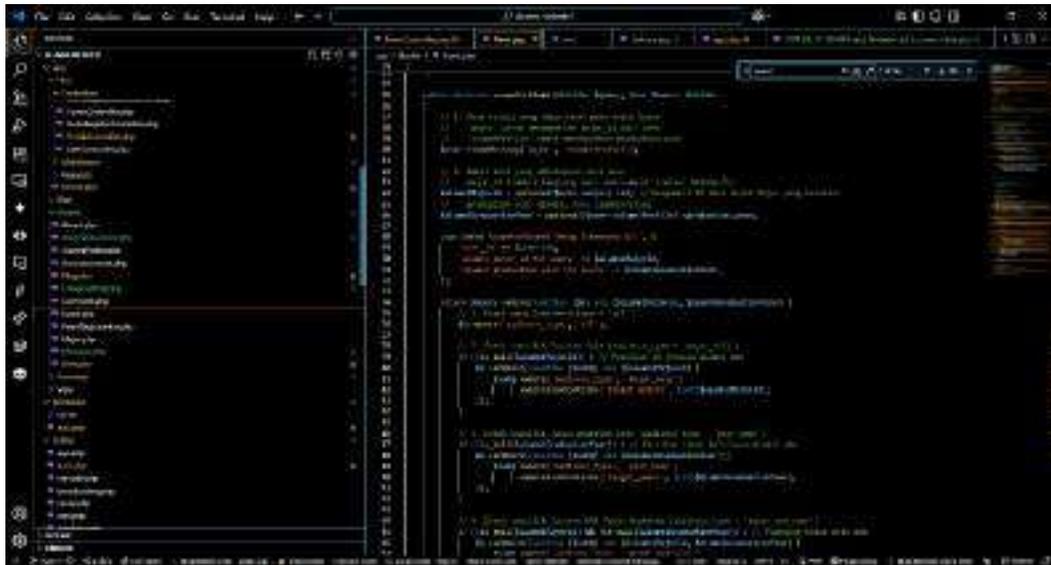
create() & store(): Menampilkan form dan memproses penambahan event baru. Validasi yang ketat diterapkan pada semua detail event, termasuk tanggal, waktu, lokasi, status RSVP, tipe audiens, batas peserta, status berbayar/gratis, dan harga. Penting, validasi target_majors dan target_years bersifat kondisional tergantung pada audience_type. Data target_majors dan target_years disimpan sebagai JSON array di database. Gambar event juga diunggah dan disimpan.

show(): Menampilkan detail satu event.

edit() & update(): Menampilkan form dan memproses pembaruan event. Logika validasi dan pemrosesan data mirip dengan store(), tetapi juga menangani pembaruan gambar event yang sudah ada. target_majors dan target_years juga di-handle dengan benar saat update.

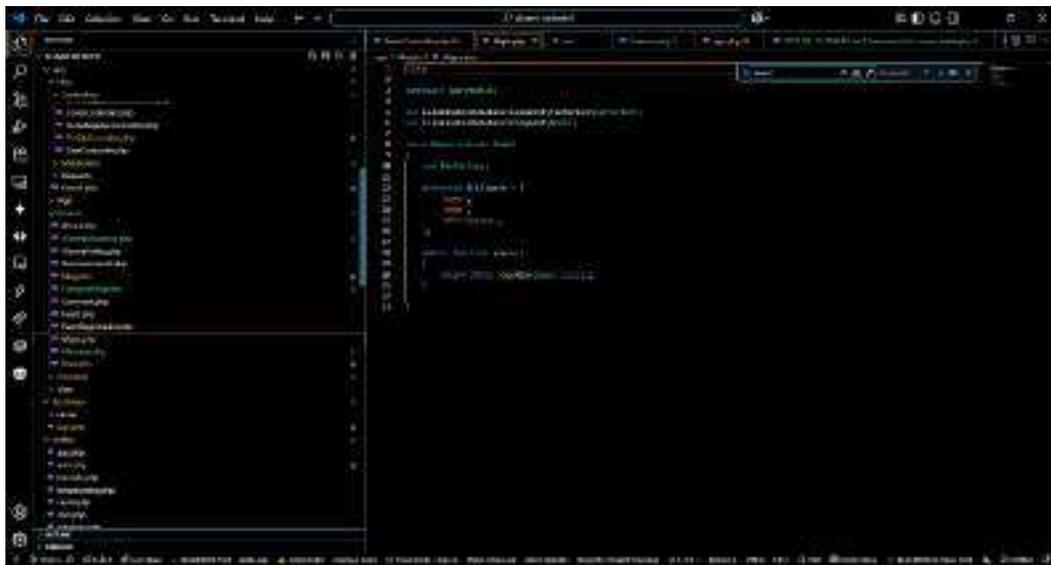
destroy(): Menghapus event dan gambar terkait dari storage.

c. Model Terkait (app/Models/Event.php, app/Models/Major.php)



The screenshot shows an IDE window with a file explorer on the left and a code editor on the right. The file explorer shows a project structure with folders like 'app', 'config', and 'resources'. The code editor displays PHP code for 'app/Models/Event.php'. The code includes a class definition for 'Event' with various methods and attributes. The code is written in a dark theme with syntax highlighting.

Gambar 3. 290 Potongan kode app/Models/Event.php



The screenshot shows an IDE window with a file explorer on the left and a code editor on the right. The file explorer shows a project structure with folders like 'app', 'config', and 'resources'. The code editor displays PHP code for 'app/Models/Major.php'. The code includes a class definition for 'Major' with various methods and attributes. The code is written in a dark theme with syntax highlighting.

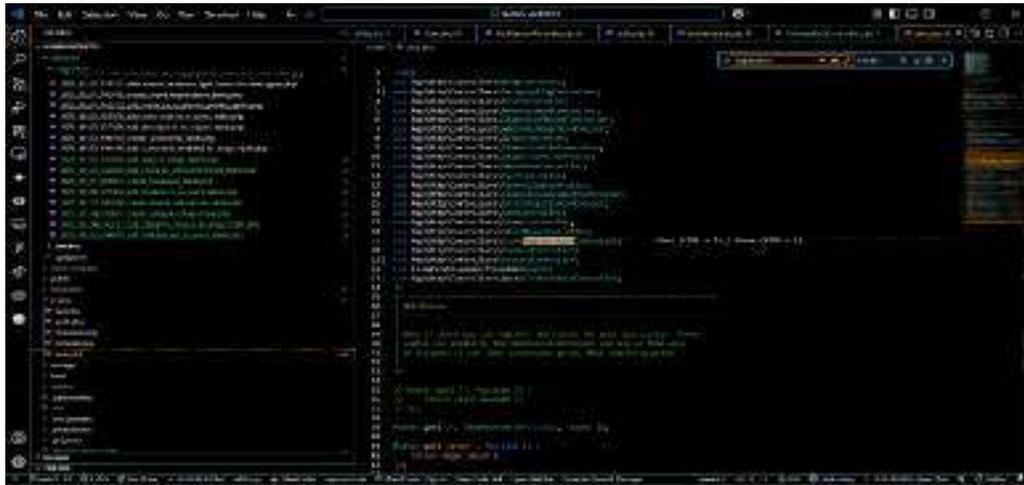
Gambar 3. 291 Potongan kode app/Models/Major.php

Event Model: Mendefinisikan semua atribut event, termasuk `audience_type` (tipe audiens), `target_majors` dan `target_years` (di-cast sebagai array karena disimpan sebagai JSON di DB), `max_attendees` (batas peserta), `is_paid` (status pembayaran), dan `price`. Casting `target_majors` dan `target_years` ke array sangat penting agar data dapat disimpan dan diambil sebagai array PHP, bukan string JSON mentah. Model ini juga memiliki relasi dan metode khusus untuk eligibility event dan slot yang tersedia.

Major Model: Menyediakan daftar jurusan yang digunakan untuk `target_majors` dalam form manajemen event.

G. Manajemen Pendaftaran Event (Admin)

- a. Rute Manajemen Pendaftaran Event Admin (`routes/web.php`)



Gambar 3. 292 Rute Manajemen Pendaftaran Event Admin

Penjelasan: Cuplikan ini menunjukkan rute-rute yang digunakan Admin untuk mengelola pendaftaran event. Semua rute ini dilindungi oleh middleware `auth` dan `admin`.

`admin.registrations.index`: Menampilkan daftar semua pendaftaran event.

admin.registrations.show: Menampilkan detail pendaftaran spesifik.

admin.registrations.confirm: Untuk mengkonfirmasi pembayaran.

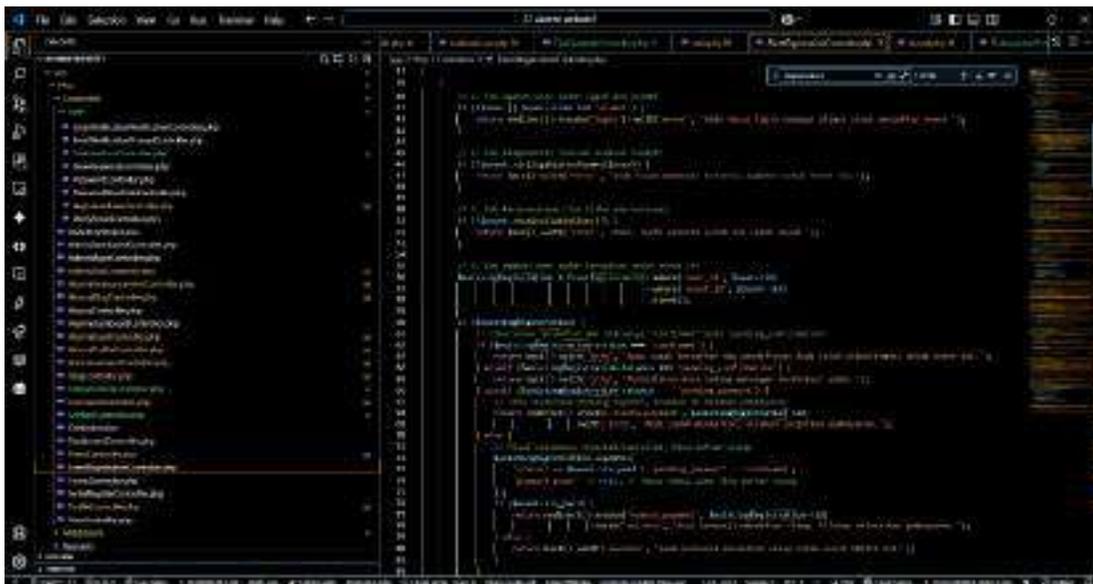
admin.registrations.reject: Untuk menolak pembayaran.

admin.registrations.destroy: Untuk menghapus pendaftaran.

admin.events.registrations.index: Menampilkan pendaftaran untuk event spesifik.

b. Logika Manajemen Pendaftaran Event Admin

(app/Http/Controllers/EventRegistrationController.php)



Gambar 3. 293 Potongan kode app/Http/Controllers/EventRegistrationController

Penjelasan:

Konstruktor: Memastikan hanya admin yang dapat mengakses metode-metode manajemen pendaftaran event ini. indexAdmin(Request \$request): Menampilkan daftar semua pendaftaran event untuk admin. Fungsi ini mendukung pencarian berdasarkan nama pengguna atau judul event, serta filter berdasarkan status pendaftaran (misal:

pending_payment, confirmed). Relasi user.alumniProfile dan event di-eager load untuk tampilan. showAdmin(EventRegistration \$registration): Menampilkan detail spesifik dari satu pendaftaran event. Ia memuat relasi pengguna (termasuk profil alumni dan jurusan) dan event terkait.

confirmPayment(EventRegistration \$registration): Mengubah status pendaftaran menjadi 'confirmed' jika status saat ini adalah 'pending_confirmation'. Sebelum konfirmasi, ada pengecekan ulang ketersediaan slot event. rejectPayment(Request \$request, EventRegistration \$registration): Mengubah status pendaftaran menjadi 'rejected' (ditolak). Ada validasi untuk alasan penolakan (jika ada).

destroyAdmin(EventRegistration \$registration): Menghapus pendaftaran event secara permanen. Jika ada bukti pembayaran yang diunggah, file bukti tersebut juga dihapus dari storage.

indexEventRegistrations(Event \$event, Request \$request): Metode ini mirip dengan indexAdmin, tetapi khusus untuk menampilkan daftar pendaftaran dari *event tertentu*. Ia juga mendukung filter status dan pencarian berdasarkan nama pengguna.

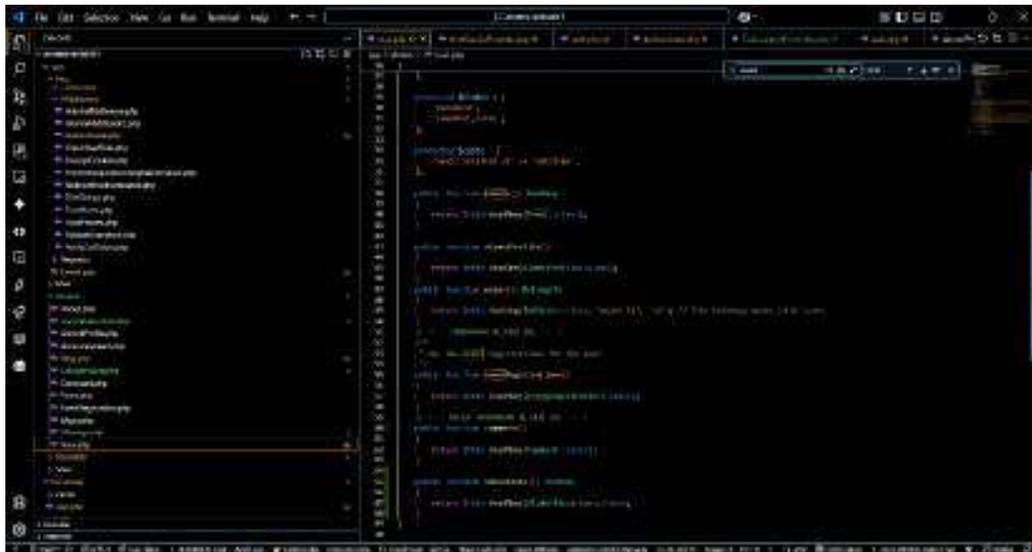
- c. Model Terkait (app/Models/EventRegistration.php, app/Models/Event.php, app/Models/User.php, app/Models/AlumniProfile.php)

This screenshot shows a code editor window with a dark theme. On the left, a file explorer displays a project structure with folders like 'app', 'config', and 'resources'. The main editor area shows PHP code for a class named 'EventRegistration'. The code includes a constructor, a 'register' method, and a 'cancel' method. The 'register' method uses 'Event::create' to create a new event record in the database. The 'cancel' method uses 'Event::update' to update the event's status to 'cancelled'.

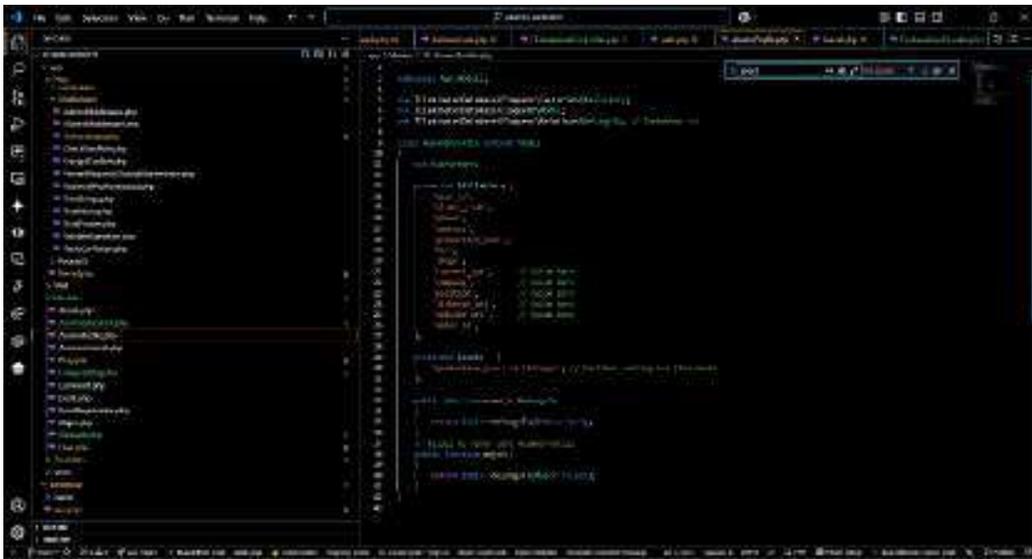
Gambar 3. 294 Potongan kode app/Models/EventRegistration.php

This screenshot shows a code editor window with a dark theme, displaying PHP code for an 'Event' model. The code includes a constructor, a 'create' method, and a 'cancel' method. The 'create' method uses 'Event::create' to create a new event record in the database. The 'cancel' method uses 'Event::update' to update the event's status to 'cancelled'. The code also includes a 'cancel' method that uses 'Event::update' to update the event's status to 'cancelled'.

Gambar 3. 295 app/Models/Event.php



Gambar 3. 296 app/Models/User.php



Gambar 3. 297 , app/Models/AlumniProfile.php

Penjelasan:

EventRegistration Model: Ini adalah model inti yang merepresentasikan satu pendaftaran event. Ia memiliki foreign key ke User dan Event. Kolom status melacak tahapan pendaftaran,

dan `payment_proof` menyimpan `path` ke bukti pembayaran. Accessor `getPaymentProofUrlAttribute` memudahkan pengambilan URL gambar bukti pembayaran.

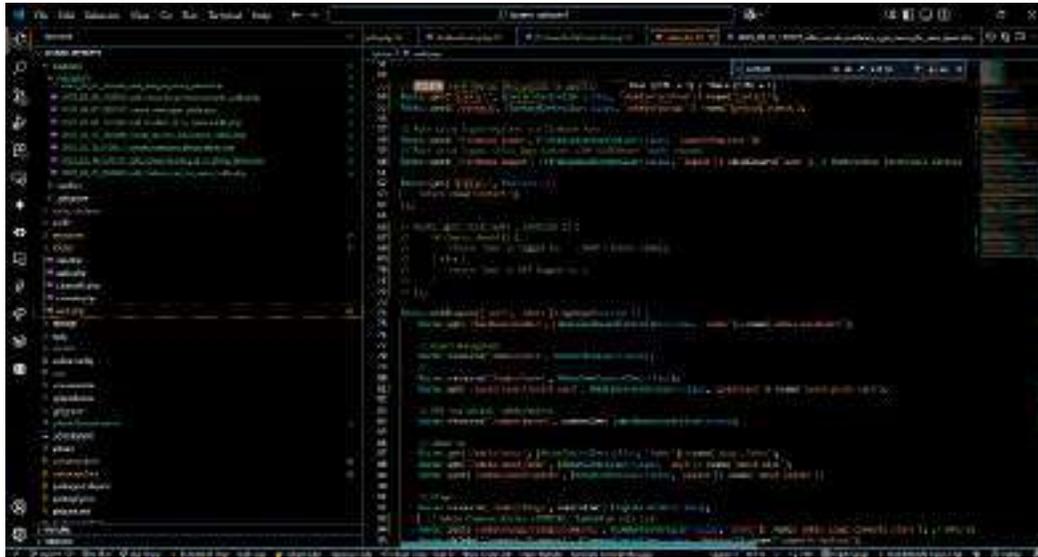
Event Model: Memiliki relasi `hasMany` ke `EventRegistration`, yang memungkinkan untuk menghitung jumlah peserta yang terdaftar (`currentAttendees`) dan memeriksa ketersediaan slot (`hasAvailableSlots`).

User Model: Memiliki relasi `hasMany` ke `EventRegistration`, yang memungkinkan admin untuk mengambil semua pendaftaran yang terkait dengan pengguna tertentu. Relasi ke `AlumniProfile` juga penting untuk menampilkan detail alumni terkait.

AlumniProfile Model: Menyimpan detail profil alumni yang dapat ditampilkan di antarmuka admin saat melihat pendaftaran.

H. Manajemen Pesan Kontak (Admin)

- a. Rute Manajemen Pesan Kontak Admin (`routes/web.php`)



Gambar 3. 298 Rute Manajemen Pesan Kontak Admin

Cuplikan ini menunjukkan rute-rute yang digunakan Admin untuk mengelola pesan kontak.

Semua rute ini dilindungi oleh middleware auth dan admin, memastikan hanya admin yang sudah login yang dapat mengaksesnya.

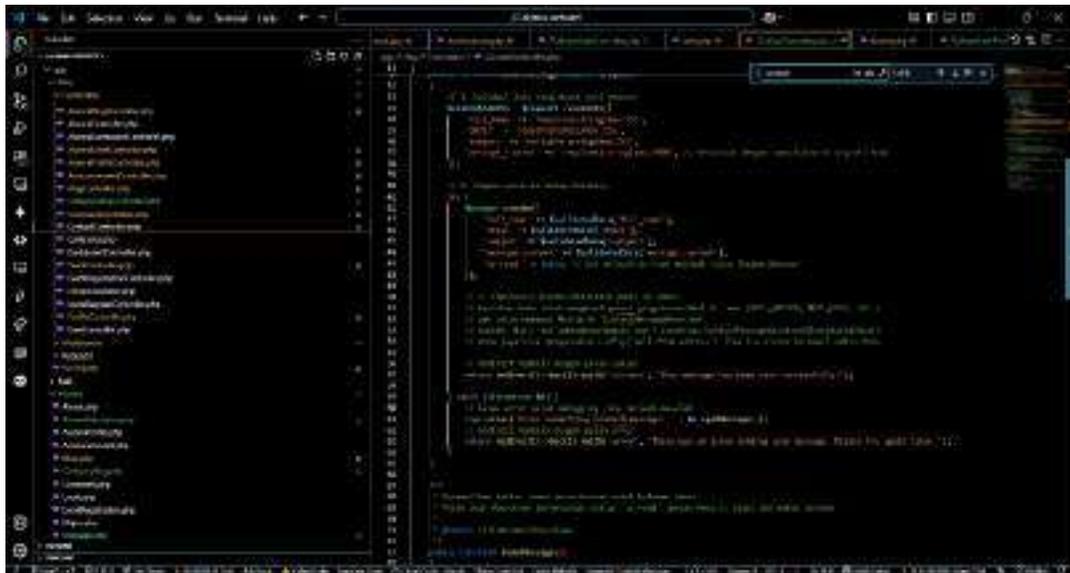
messages.index: Menampilkan daftar semua pesan kontak.

messages.show: Menampilkan detail pesan spesifik.

messages.destroy: Untuk menghapus pesan.

b. Logika Manajemen Pesan Kontak Admin

(app/Http/Controllers/ContactController.php)



Gambar 3. 299 app/Http/Controllers/ContactController.php

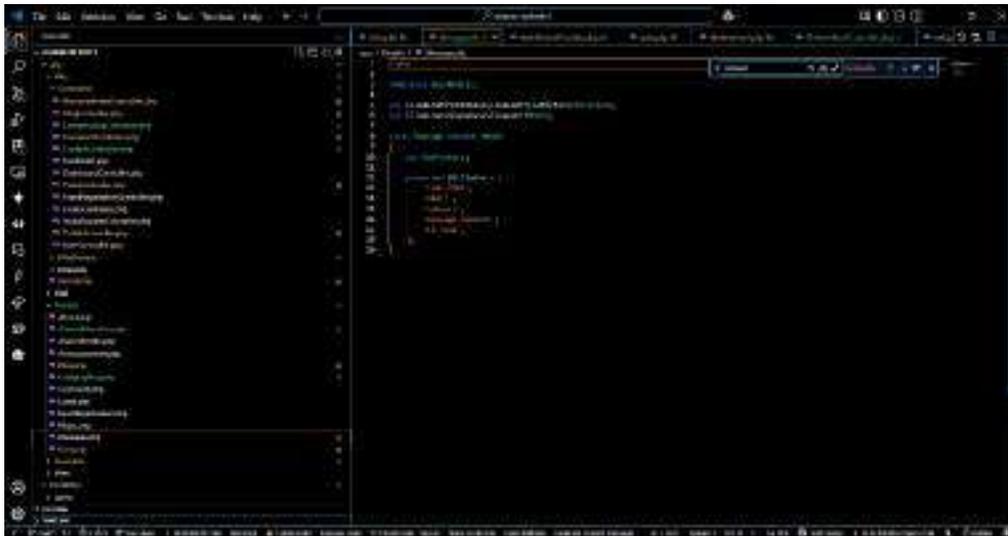
Konstruktor: Memastikan hanya admin yang dapat mengakses metode indexMessages, showMessage, dan destroyMessage.

indexMessages(): Mengambil daftar semua pesan kontak dari database. Pesan diurutkan sehingga pesan yang belum dibaca (is_read = false) muncul lebih dulu, diikuti oleh pesan terbaru. Hasilnya di-paginate untuk tampilan halaman.

showMessage(Message \$message): Metode ini menampilkan detail spesifik dari satu pesan kontak. Setelah pesan ditampilkan, status is_read pesan tersebut akan diubah menjadi true (sudah dibaca) dan disimpan ke database.

destroyMessage(Message \$message): Menghapus pesan kontak dari database. Setelah penghapusan, admin diarahkan kembali ke daftar pesan dengan pesan sukses.

c. Model Terkait (app/Models/Message.php)



Gambar 3. 300 app/Models/Message.php

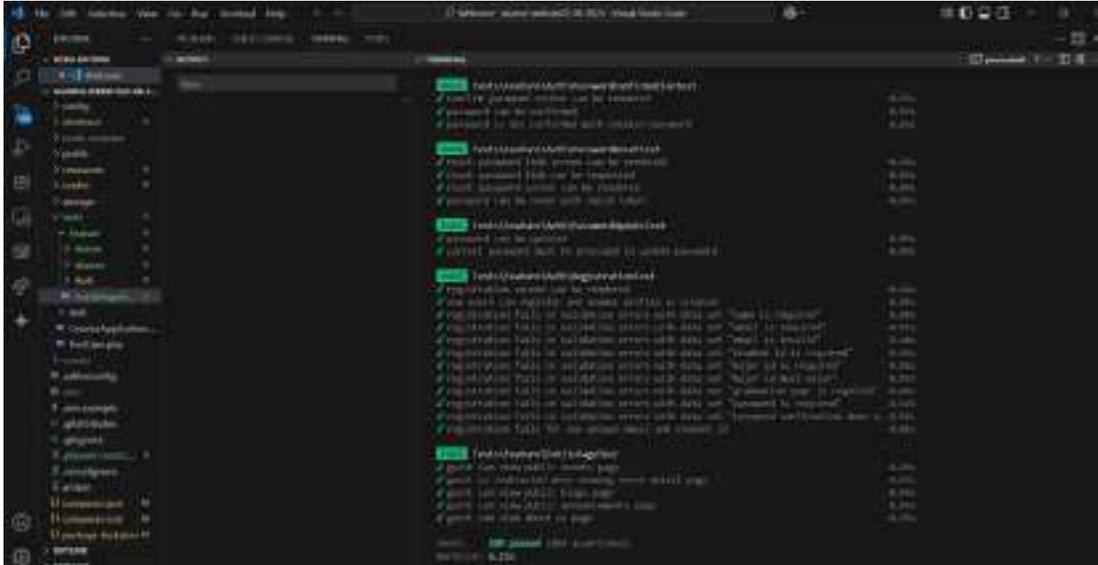
Penjelasan:

Model Message merepresentasikan tabel messages di database. Properti \$fillable mendefinisikan kolom-kolom yang dapat diisi, termasuk full_name, email, subject, message_content, dan is_read. Kolom is_read ini adalah boolean yang digunakan untuk melacak status baca pesan.

8. Realisasi Quality Assurance

Setelah seluruh fungsionalitas sistem selesai diimplementasikan sesuai dengan perancangan pada tahap sebelumnya, langkah selanjutnya adalah melakukan pengujian sistem. Tahap ini, yang juga dikenal sebagai *Quality Assurance* (QA), bertujuan untuk memastikan bahwa setiap fitur dan alur kerja pada aplikasi Website Alumni GANISSA

berjalan sesuai harapan, bebas dari *bug* fungsional, dan memenuhi semua kebutuhan yang telah ditetapkan.



Gambar 3. 301 Contoh Pengujian

a. Metodologi Pengujian

Metode pengujian yang diterapkan adalah Black Box Testing. Metode ini berfokus pada pengujian fungsionalitas dari luar, tanpa perlu mengetahui detail struktur kode internal. Pengujian dilakukan dengan memberikan input pada sistem dan memvalidasi output yang dihasilkan.

Implementasi teknis dari metode ini menggunakan Feature Test yang terintegrasi dalam framework Laravel. Feature Test mensimulasikan interaksi.

b. Lingkungan Pengujian

Pengujian sistem dilaksanakan pada lingkungan pengembangan dengan rincian perangkat lunak sebagai berikut:

Tabel 3. 4 Lingkungan Pengujian

Komponen	Deskripsi
Framework	Laravel
Alat Uji	PHPUnit
Sistem Operasi	Windows
Perangkat Pendukung	Laragon, Composer, PHP, Git
Database pengujian	Menggunakan database terisolasi untuk memastikan proses pengujian tidak mengubah data pada database pengembangan.

c. Skenario Pengujian

Skenario pengujian disusun untuk mencakup semua fungsionalitas utama aplikasi yang telah diimplementasikan. Total terdapat 109 kasus uji yang dirancang untuk memvalidasi modul Admin, modul Alumni, sistem Autentikasi, dan akses Halaman Publik. Rincian skenario pengujian telah dijelaskan pada sub-bab perancangan sistem.

d. Hasil Pengujian

1. Pengujian Mandiri

Tabel 3. 5 Hasil Pengujian

ID	Feature	Expected Result	Status
T001	AdminAboutTest	admin can view the about page	Lulus
T002		admin can update about page when it is empty	Lulus
T003		admin can update existing about content	Lulus
T004	AdminAnnouncementTest	admin can view the announcement page	Lulus
T005		admin can create new announcement	Lulus

T006		admin can update announcement	Lulus
T007		admin can delete announcement	Lulus
T008		admin can search announcements by title	Lulus
T009		admin can filter announcements by status	Lulus
T010	AdminBlogTest	admin can view the blog page	Lulus
T011		admin can create new blog	Lulus
T012		admin can update blog	Lulus
T013		admin can search blogs by title	Lulus
T014		admin can filter blogs by category	Lulus
T015		admin can delete blog	Lulus
T016	AdminCategoryBlogTest	admin can view the category blog page	Lulus
T017		admin can create new category blog	Lulus
T018		admin can update category blog	Lulus
T019		admin can delete category blog	Lulus
T020	AdminEventTest	admin can view the event page	Lulus
T021		admin can create new event	Lulus
T022		admin can update event	Lulus
T023		admin can delete event	Lulus
T024		admin can search events by title	Lulus
T025		admin can filter events by payment type	Lulus
T026		admin can filter events by status	Lulus
T027		admin can filter events by audience type	Lulus
T028		admin can filter events by target major	Lulus
T029		admin can filter events by target year	Lulus
T030		admin can use combined filters for events	Lulus
T031	AdminMajorTest	admin can view the major page	Lulus
T032		admin can create a new major	Lulus
T033		admin can update a major	Lulus
T034		admin can delete a major	Lulus
T035	AdminMessageTest	admin can view the message page	Lulus
T036		admin can view a single message and it is marked as read	Lulus
T037		admin can delete a message	Lulus
T038	AdminUserTest	admin can view user management page	Lulus
T039		admin can create new user with full details	Lulus
T040		admin can update user details and role	Lulus

T041		admin can delete a user	Lulus
T042		admin can search users by name	Lulus
T043		admin can filter users by role	Lulus
T044	AlumniAnnouncementsTest	alumni can view announcement listing page	Lulus
T045		alumni can filter announcements by keyword search	Lulus
T046		alumni can view single announcement detail page	Lulus
T047		guest cannot access alumni announcements index	Lulus
T048	AlumniBlogTest	alumni can view blog listing page	Lulus
T049		alumni can filter blogs by keyword search	Lulus
T050		alumni can view single blog detail page	Lulus
T051		alumni can post comment on blog	Lulus
T052		alumni can reply to a comment	Lulus
T053		alumni can delete their own comment	Lulus
T054		alumni cannot delete others comment	Lulus
T055		blog listing shows no results message correctly	Lulus
T056		guest cannot access alumni blog index	Lulus
T057		guest cannot access alumni blog show page	Lulus
T058	AlumniDashboardTest	alumni dashboard can be rendered	Lulus
T059	AlumniDirectoryTest	alumni can view the directory page	Lulus
T060		alumni can filter directory by keyword	Lulus
T061		alumni can filter directory by major	Lulus
T062		alumni can filter directory by graduation year	Lulus
T063		alumni directory shows no results message correctly	Lulus
T064		alumni can view another alumnus profile page	Lulus
T065		guest cannot access alumni directory	Lulus
T066		non alumni user cannot access alumni directory	Lulus
T067	AlumniEventTest	alumni can view event listing page	Lulus
T068		alumni can filter events by keyword	Lulus
T069		alumni can filter events by status	Lulus

T070		alumni can filter events by type	Lulus
T071		alumni can view single event detail page	Lulus
T072		alumni can register for a free event	Lulus
T073		alumni is redirected to payment page for paid event	Lulus
T074		alumni can view payment proof upload page	Lulus
T075		alumni can upload payment proof	Lulus
T076		alumni can view their own profile page	Lulus
T077		alumni can view the edit profile form	Lulus
T078		alumni can update their profile information	Lulus
T079	AuthenticationTest	login screen can be rendered	Lulus
T080		users can authenticate using the login screen	Lulus
T081		users can not authenticate with invalid password	Lulus
T082		users can logout	Lulus
T083	EmailVerificationTest	user can verified the password	Lulus
T084	PasswordConfirmationTest	confirm password screen can be rendered	Lulus
T085		password can be confirmed	Lulus
T086		password is not confirmed with invalid password	Lulus
T087	PasswordResetTest	reset password link screen can be rendered	Lulus
T088		reset password link can be requested	Lulus
T089		reset password screen can be rendered	Lulus
T090		password can be reset with valid token	Lulus
T091	PasswordUpdateTest	password can be updated	Lulus
T092		correct password must be provided to update password	Lulus
T093	RegistrationTest	registration screen can be rendered	Lulus
T094		new users can register and alumni profile is created	Lulus
T095		registration fails on validation errors with data set "name is required"	Lulus
T096		registration fails on validation errors with data set "email is required"	Lulus

T097		registration fails on validation errors with data set "email is invalid"	Lulus
T098		registration fails on validation errors with data set "student id is required"	Lulus
T099		registration fails on validation errors with data set "major id is required"	Lulus
T100		registration fails on validation errors with data set "major id must exist"	Lulus
T101		registration fails on validation errors with data set "graduation year is required"	Lulus
T102		registration fails on validation errors with data set "password is required"	Lulus
T103		registration fails on validation errors with data set "password confirmation does not match"	Lulus
T104		registration fails for non unique email and student id	Lulus
T105	PublicPageTest	guest can view public events page	Lulus
T106		guest is redirected when viewing event detail page	Lulus
T107		guest can view public blogs page	Lulus
T108		guest can view public announcements page	Lulus
T109		guest can view about us page	Lulus

2. Pengujian berdasarkan/berorientasi pada SRS

Tabel 3. 6 Pengujian berdasarkan/berorientasi pada SRS

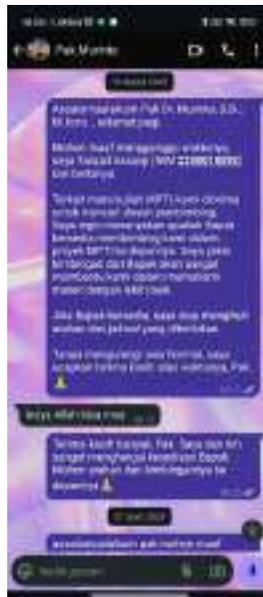
Kode Fungsional	Fungsionalitas Sistem	Deskripsi Fungsionalitas Sistem	Status
F-ADM-001	Manajemen Akun Admin	Sistem harus memungkinkan Admin mengelola akun admin lain, termasuk membuat, mengedit, menghapus, dan mengatur hak akses.	Lulus
F-ADM-002	Manajemen Data Profil Alumni	Sistem harus memungkinkan Admin melakukan CRUD data profil alumni secara lengkap.	Lulus

F-ADM-003	Dashboard & Pelaporan Lanjutan	Sistem harus menyajikan dashboard dengan angka-angka pengguna, konten, dan aktivitas.	Lulus
F-ADM-004	Manajemen Konten Umum (Berita, Artikel, Pengumuman)	Sistem harus menyediakan CMS (Content Mangement System) untuk mengelola berita, artikel, dan pengumuman dengan editor, media, jadwal publikasi, dan riwayat revisi.	Lulus
F-ADM-005	Pengelolaan Kategori & Tag Dinamis	Sistem harus memungkinkan Admin mengelola kategori dan tag konten secara dinamis dan bertingkat.	Lulus
F-ADM-006	Manajemen Akun Pengguna & Otorisasi	Sistem harus mendukung manajemen akun pengguna (Admin, Alumni, Guest) dengan pengaturan peran dan hak akses berbasis RBAC.	Lulus
F-ADM-007	Sistem Moderasi Konten & Komunikasi	Sistem harus menyediakan modul moderasi untuk menyetujui, menolak, atau menghapus konten dan komentar.	Lulus
F-ADM-008	Manajemen Event Komprehensif	Sistem harus memungkinkan Admin mengelola event secara komprehensif, termasuk data peserta.	Lulus
F-ALUM-001	Registrasi & Aktivasi Akun Mandiri	Sistem harus memungkinkan calon alumni mendaftar akun baru secara mandiri melalui formulir.	Lulus
F-ALUM-002	Manajemen & Pembaruan Profil Mandiri	Sistem harus memungkinkan Alumni memperbaiki dan mengelola profil pribadi secara mandiri.	Lulus
F-ALUM-003	Pencarian & Filter Alumni Lanjutan	Sistem harus menyediakan fitur pencarian alumni dengan filter dan urutan hasil berdasarkan kriteria tertentu.	Lulus
F-ALUM-004	Akses & Interaksi Konten (Berita, Artikel)	Sistem harus memungkinkan Alumni mengakses, berinteraksi, dan membagikan konten seperti artikel atau berita.	Lulus
F-ALUM-005	Registrasi & Partisipasi Event Online/Offline	Sistem harus memungkinkan Alumni melihat, mendaftar, dan mengelola partisipasi event secara online/offline.	Lulus
F-ALUM-006	Kontribusi Konten (Artikel, Pengalaman, Portofolio)	Sistem harus memungkinkan Alumni mengirim kontribusi konten seperti artikel, pengalaman kerja, atau portofolio.	Lulus

F-GUEST-001	Akses Informasi Publik Terbatas	Sistem harus memungkinkan Guest mengakses informasi publik tanpa login.	Lulus
F-GUEST-002	Penampil Jadwal & Detail Event Publik	Sistem harus memungkinkan Guest melihat jadwal dan detail event publik.	Lulus
F-GUEST-003	Proses Registrasi Awal Alumni	Sistem harus memungkinkan Guest mengisi formulir pendaftaran awal alumni sebelum aktivasi akun.	Lulus

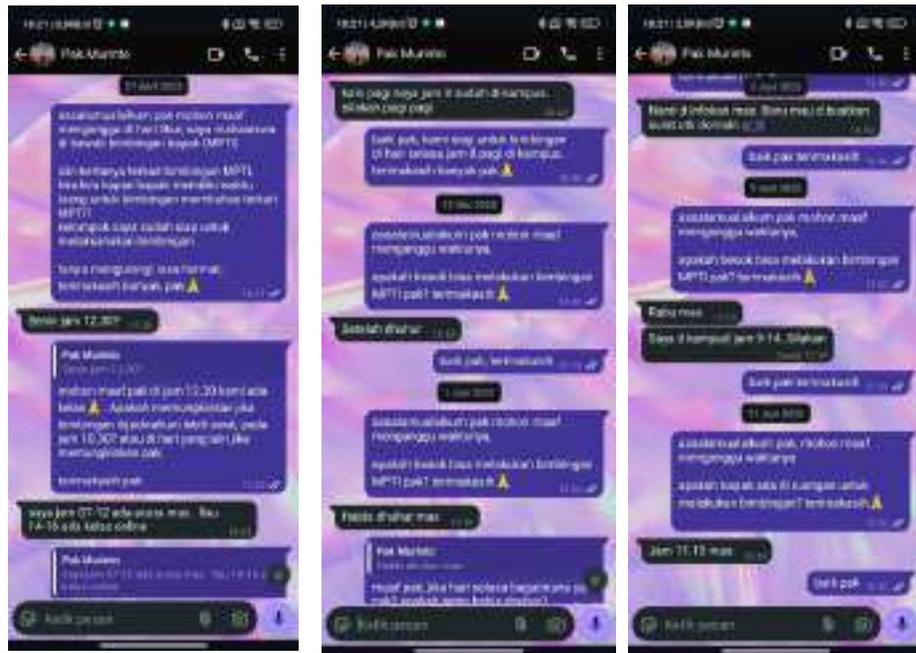
9. Realisasi Sekretaris

a. Dokumentasi Menghubungi/ Mencari Dosen Prmbimbing MPTI

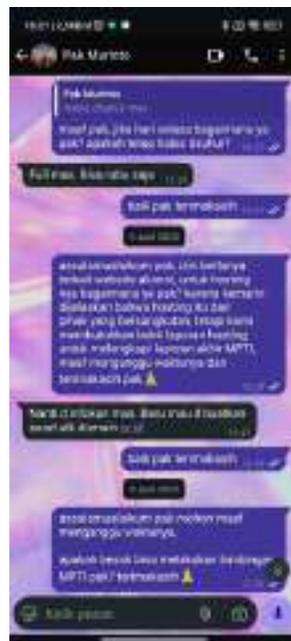


Perwakilan kelompok Zestify menghubungi Dr . Murinto, S.Si, M.Kom, untuk menanyakan kepada beliau untuk berkenan menjadi dosen pembimbing proyek MPTI ini.

b. Dokumentasi Merencanakan Jadwal Bimbingan MPTI



c. Dokumentasi Diskusi MPTI dengan Dosen Pembimbing via WhatsApp



C. Penjaminan Kualitas Proyek

Proses penjaminan kualitas untuk proyek pengembangan Sistem Informasi Alumni SMAN 1 Pemalang (GANISSA) bertujuan untuk memastikan bahwa hasil pengembangan website memenuhi standar spesifikasi yang telah ditetapkan. Penjaminan ini melibatkan anggota tim Zestify dengan SMAN 1 Pemalang. Tujuan dari penjaminan kualitas proyek adalah memastikan website berfungsi dengan baik dan bebas dari bug, serta memastikan perlindungan data pengguna menggunakan SSL untuk mengenkripsi data transfer antara klien dan server.

Tim Zestify memastikan keamanan website menggunakan prepared statements dengan parameter binding untuk mencegah SQL Injection. Kedua, penggunaan session untuk melindungi data pengguna, serta hashing sebelum menyimpan data kata sandi dan jawaban verifikasi.

D. Keberlanjutan Proyek

1. Kelanjutkan program yang belum selesai.
2. Membuat dokumen panduan.
3. Pihak mitra mendapatkan pemeliharaan selama 1 bulan.

BAB IV Penutup

A. Kesimpulan

Proyek Sistem Informasi Alumni SMAN 1 Pemalang (GANISSA) diinisiasi sebagai solusi strategis untuk mengatasi permasalahan mendasar terkait pengelolaan data dan komunikasi alumni yang selama ini masih konvensional, tersebar, dan tidak akurat. Kondisi ini menyebabkan terhambatnya konektivitas, partisipasi, dan solidaritas di antara ribuan alumni SMAN 1 Pemalang.

GANISSA dirancang sebagai platform web terpusat, interaktif, dan mudah diakses, dengan tujuan utama untuk membangun basis data alumni yang akurat, menciptakan saluran komunikasi resmi, mendorong keterlibatan aktif alumni dalam berbagai kegiatan, serta memperkuat jaringan dan solidaritas antar-alumni. Sistem ini akan mengakomodasi tiga aktor utama: Admin (pengelola sistem dan konten), Alumni (pengguna utama yang dapat mengelola profil, berpartisipasi acara, dan berinteraksi), serta Pengunjung (masyarakat umum dengan akses terbatas).

Meskipun proyek ini memiliki kekuatan dalam menyediakan layanan manajemen data yang cepat, fleksibel, dan kompatibel lintas perangkat, terdapat beberapa kelemahan seperti ketergantungan pada SDM teknis, keterbatasan data awal, dan potensi kendala teknis. Namun, peluang besar terbuka melalui dukungan digitalisasi pendidikan, potensi kolaborasi dengan instansi lain, dan tren digitalisasi sekolah. Risiko utama yang dihadapi meliputi kurangnya partisipasi alumni, data tidak akurat, keterbatasan sumber daya, serta isu keamanan dan privasi data. Proyek ini juga dihadapkan pada kendala waktu, anggaran, dan SDM yang terbatas, serta tantangan dalam kompetensi teknologi pengguna.

Strategi yang telah dirumuskan melalui analisis SWOT menunjukkan komitmen untuk memanfaatkan kekuatan dan peluang (misalnya, pengembangan modul tracer study, direktori alumni, dan integrasi dengan pihak ketiga) sambil mengatasi kelemahan dan ancaman (seperti penerapan sistem verifikasi data, pelatihan admin, enkripsi data, dan pembuatan SOP pengelolaan risiko). Secara keseluruhan, GANISSA adalah langkah penting dan relevan untuk memodernisasi ekosistem alumni SMAN 1 Pemalang dan memaksimalkan potensi komunitas yang berharga ini.

B. Saran

Berdasarkan analisis dokumen, berikut adalah beberapa saran untuk keberlanjutan dan optimalisasi proyek Sistem Informasi Alumni SMAN 1 Pemalang (GANISSA):

1. Prioritaskan Sosialisasi dan Edukasi Pengguna:
 - a. Mengingat risiko kurangnya partisipasi alumni dan kesulitan penggunaan oleh pengguna awam, strategi sosialisasi harus lebih masif dan berkelanjutan. Gunakan berbagai media (sosial media, grup alumni, pertemuan langsung) untuk memperkenalkan GANISSA.
 - b. Sediakan panduan penggunaan yang sangat sederhana dan visual (misalnya, tutorial video singkat) untuk membantu alumni yang kurang familiar dengan teknologi. Pertimbangkan sesi pelatihan daring atau luring secara berkala.
2. Perkuat Validasi dan Pembaruan Data Alumni:

- a. Terapkan mekanisme verifikasi data yang kuat sejak awal, seperti verifikasi email/nomor telepon saat pendaftaran, atau validasi oleh admin untuk data-data kunci (misalnya, tahun angkatan, profesi).
 - b. Dorong alumni untuk memperbarui data secara mandiri melalui notifikasi berkala atau insentif kecil. Pertimbangkan fitur pengingat otomatis untuk memperbarui profil.
3. Rencanakan Pemeliharaan dan Pengembangan Berkelanjutan:
- a. Meskipun menggunakan hosting gratis di awal, pertimbangkan rencana jangka panjang untuk infrastruktur yang lebih stabil dan skalabel seiring bertambahnya pengguna dan fitur. Penggalangan dana dari alumni atau sponsor dapat menjadi opsi.
 - b. Susun jadwal pemeliharaan rutin dan pengujian sistem berkala untuk mengidentifikasi dan memperbaiki *bug* atau kerentanan keamanan sesegera mungkin.
 - c. Secara bertahap, kembangkan fitur-fitur yang menjadi strategi S-O dan W-O, seperti modul *tracer study*, direktori alumni yang lebih canggih, atau integrasi API, untuk menambah nilai sistem.
4. Optimalkan Keamanan dan Privasi Data:
- a. Pastikan implementasi SSL menyeluruh. Perkuat penggunaan *prepared statements* dan *hashing* kata sandi.
 - b. Libatkan ahli keamanan siber (jika memungkinkan) untuk melakukan audit keamanan berkala.

- c. Buat kebijakan privasi data yang jelas dan transparan, serta pastikan pengguna memahami bagaimana data mereka dikelola dan dilindungi. Terapkan akses berbasis peran yang ketat untuk setiap aktor.

5. Perkuat Komitmen dan Keterlibatan Stakeholder:

- a. Jaga komunikasi yang efektif dengan seluruh *stakeholder*, terutama pengurus alumni, sekolah, dan alumni senior. Keterlibatan mereka sangat krusial untuk keberhasilan proyek dan adopsi sistem.
- b. Selenggarakan pertemuan rutin untuk mempresentasikan kemajuan dan mendapatkan masukan dari mereka.

Lampiran

A. Proposal Proyek

PROPOSAL PROYEK

PENGANTAR MANAJEMEN DAN PRINSIP PROYEK

PEMBANGUNAN SISTEM INFORMASI ALUMNI SMAN 1 PEMALANG (GANISSA)



Disusun Oleh , :

Noval Lias Ramadani , 2200018083

Hasan Nur Rasyid , 2200018068

Andriansyah , 2200018088

Farqad Assanji , 2200018092

Destyawan Satria Nugraha , 2200018095

UNIVERSITAS AHMAD DAHLAN

FAKULTAS TEKNOLOGI INDUSTRI

PROGRAM STUDI TEKNIK INFORMATIKA

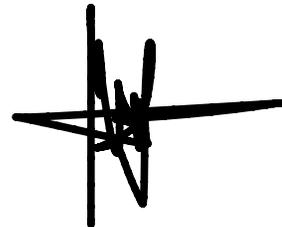
TAHUN AJARAN 2024/2025 GASAL

Kata Pengantar

Puji syukur kami panjatkan ke hadirat Allah SWT yang telah memberikan rahmat dan karunia-Nya, sehingga kami dapat menyelesaikan proposal dengan judul “Sistem Informasi Alumni SMAN 1 Pemalang (GANISSA)” ini dengan baik dan tepat waktu. Proposal ini disusun sebagai langkah awal dalam mewujudkan sistem informasi yang bertujuan untuk mempermudah pengelolaan data alumni serta mempererat komunikasi antar alumni SMAN 1 Pemalang. Kami berharap sistem ini dapat menjadi sarana yang bermanfaat, tidak hanya bagi para alumni, tetapi juga bagi pihak sekolah dalam menjalin sinergi dan kontribusi bersama demi kemajuan almamater.

Kami percaya bahwa dengan memanfaatkan teknologi informasi, dengan memanfaatkan kemajuan dan perkembangan teknologi yang sangat masif seperti perkembangan sistem informasi dalam bentuk website dapat memberikan kemudahan bagi guru, staff, ataupun murid dalam menciptakan minat baca dan literasi di era digital. Dalam proses penyusunan proposal ini, kami banyak menerima dukungan, masukan, dan arahan dari berbagai pihak. Oleh karena itu, kami ingin mengucapkan terima kasih yang sebesar-besarnya kepada Semua pihak yang telah berkontribusi dalam penyusunan proposal ini. Kami menyadari bahwa proposal ini masih memiliki kekurangan. Oleh karena itu, kami sangat terbuka untuk menerima kritik dan saran yang membangun demi perbaikan dan pengembangan proyek ini di masa mendatang. Semoga proposal ini dapat memberikan manfaat nyata dan menjadi kontribusi positif dalam mendukung pertumbuhan literasi di Indonesia.

Yogyakarta, 1 Januari 2025

A handwritten signature in black ink, consisting of several overlapping vertical and horizontal strokes, positioned at the bottom right of the page.

I. Latar Belakang

Komunitas alumni merupakan aset berharga bagi almamater, berfungsi sebagai jaringan pendukung, sumber inspirasi, dan bukti nyata dari keberhasilan pendidikan. Bagi SMAN 1 Pemalang, ikatan yang kuat di antara ribuan alumni yang tersebar di berbagai wilayah dan profesi adalah potensi besar yang perlu dikelola secara optimal. Namun, seiring berjalannya waktu, muncul tantangan signifikan dalam menjaga konektivitas dan mengelola data komunitas ini secara efektif.

Permasalahan utama yang melatarbelakangi kebutuhan akan sebuah sistem terintegrasi adalah data alumni yang tersebar, tidak akurat, dan sulit diperbarui. Pendataan yang masih mengandalkan metode konvensional menyebabkan informasi penting mengenai kontak, domisili, dan karier alumni menjadi usang. Akibatnya, terjalin jurang komunikasi baik antar angkatan alumni maupun antara alumni dengan pihak almamater. Informasi krusial seperti agenda reuni, berita duka, atau pencapaian alumni seringkali tidak tersampaikan secara merata, sehingga menghambat partisipasi dan mengurangi rasa keterlibatan.

Kondisi ini secara langsung berdampak pada melemahnya jaringan dan solidaritas alumni. Potensi untuk kolaborasi profesional, mentoring, maupun kegiatan sosial menjadi hilang karena tidak adanya wadah digital yang terpusat dan terpercaya. Komunikasi yang terbatas dan data yang tidak terkelola dengan baik membuat komunitas alumni SMAN 1 Pemalang belum dapat berfungsi sebagai sebuah ekosistem yang solid dan berkelanjutan.

Oleh karena itu, pengembangan Sistem Informasi Alumni SMAN 1 Pemalang (Ganissa) diinisiasi sebagai solusi strategis untuk mengatasi tantangan tersebut. Ganissa dirancang untuk mentransformasi metode komunikasi dan manajemen data yang konvensional menjadi sebuah platform digital yang terpusat, interaktif, dan

mudah diakses. Proyek ini bertujuan untuk membangun kembali dan memperkuat koneksi dengan menyediakan "rumah digital" bagi seluruh alumni, di mana mereka dapat terhubung, berbagi informasi, dan berpartisipasi aktif dalam kegiatan komunitas, sehingga pada akhirnya dapat mempererat rasa kekeluargaan serta melestarikan warisan dan ikatan emosional terhadap almamater.

II. Tujuan

Tujuan utama dari proyek ini adalah :

1. **Membangun Basis Data Terpusat dan Akurat:** Menciptakan satu sumber data utama (single source of truth) untuk seluruh alumni SMAN 1 Pernalang melalui mekanisme registrasi mandiri dan pengelolaan profil oleh setiap alumni. Ini akan mengatasi masalah data yang usang dan tidak valid.
2. **Menciptakan Saluran Komunikasi Resmi:** Menjadi medium komunikasi yang efisien dan terpercaya dari pengelola alumni kepada seluruh anggota melalui fitur pengumuman dan blog. Hal ini memastikan informasi penting seperti berita duka, pencapaian, atau agenda penting lainnya dapat tersampaikan secara merata.
3. **Mendorong Keterlibatan dan Partisipasi Aktif:** Memfasilitasi dan meningkatkan keterlibatan alumni dalam berbagai kegiatan komunitas dengan menyediakan informasi acara yang terstruktur dan kemudahan untuk mendaftar secara online. Platform ini juga mendorong diskusi melalui fitur komentar pada blog.
4. **Memperkuat Jaringan dan Solidaritas Alumni:** Menjadi wadah digital yang memungkinkan alumni untuk tetap terhubung, berbagi informasi, dan membangun jaringan profesional maupun sosial yang kuat, sehingga

mempererat rasa solidaritas dan kekeluargaan di antara alumni SMAN 1 Pematang.

III. Ruang Lingkup Project

1. Informasi Umum Project				
Nama Project	Sistem Informasi Alumni Ganissa Pematang (SMA N 1 Pematang)			
Nama Mitra	Alumni Ganissa Pematang			
Manfaat Project	Mentransformasi metode komunikasi dan manajemen data komunitas alumni dari konvensional menjadi sebuah platform digital yang terpusat, interaktif, dan mudah diakses. Hal ini bertujuan untuk membangun kembali dan memperkuat koneksi dengan menyediakan "rumah digital" bagi seluruh alumni, di mana mereka dapat terhubung, berbagi informasi, dan berpartisipasi aktif dalam kegiatan komunitas. Pada akhirnya, proyek ini akan mempererat rasa kekeluargaan serta melestarikan warisan dan ikatan emosional terhadap almamater.			
2. Tim Project				
	Name	Jobdesk	NO Hp	E-mail
Project Manager	Noval Lias Ramadani	Project Manager, Front End 1, Back End 2	081268181997	2200018083@webmail.uad.ac.id
Keanggotaan				
1	Hasan Nur Rasyid	Back End 1, Sekretaris		
2	Andriansyah	System Analist		
3	Farqad Assanji	UI/UX, Bendahara		
4	Destyawan Satria Nugraha	Tester/QA	087719284231	2200018095@webmail.uad.ac.id
3. Stakeholders				
Admin : Pihak yang memiliki wewenang tertinggi di dalam sistem. Stakeholder ini bertanggung jawab atas pengelolaan keseluruhan konten (seperti blog, acara, dan pengumuman) serta manajemen data pengguna alumni yang terdaftar. Peran ini kemungkinan besar dipegang oleh pengurus inti organisasi alumni				

<p>Alumni : pengguna utama dan target utama dari sistem ini. Sebagai pengguna terdaftar, mereka memiliki akses ke fitur-fitur interaktif seperti mengelola profil pribadi, mendaftar pada suatu kegiatan, serta memberikan komentar pada konten yang ada untuk membangun jaringan</p>
<p>Pengunjung : Pengguna tidak terdaftar atau masyarakat umum yang memiliki akses terbatas. Peran mereka adalah sebagai audiens pasif yang dapat melihat informasi publik yang dibagikan di platform, seperti berita, pengumuman, atau informasi acara</p>
<p>4. Project Scope</p>
<p>Project Purpose</p>
<p>Pengembangan Sistem Informasi Alumni SMAN 1 Pemalang, yang disebut 'Ganissa', dilatarbelakangi oleh kebutuhan untuk menciptakan ekosistem digital yang terstruktur dan berkelanjutan bagi komunitas alumni. Data alumni yang tersebar, tidak ter-update, dan komunikasi yang terbatas menghambat terbentuknya jaringan yang solid dan hilangnya potensi kolaborasi. Ganissa dirancang sebagai platform yang hidup dan dinamis untuk mentransformasi interaksi dan pengelolaan informasi komunitas alumni SMAN 1 Pemalang.</p>
<p>Scope Definition</p>
<p>Ruang lingkup proyek ini mendefinisikan batasan fungsional dari sistem Ganissa, yang secara spesifik ditentukan oleh interaksi dan kapabilitas masing-masing aktor yang terlibat.</p> <p>Ruang Lingkup Fungsional :</p> <ol style="list-style-type: none"> 4. Aktor Admin : Sebagai pengelola utama, Admin memiliki wewenang penuh atas operasional sistem. Kapabilitasnya meliputi : <ul style="list-style-type: none"> - Melakukan autentikasi untuk mengakses panel administrasi sistem. - Mengelola data dan akun seluruh pengguna yang terdaftar sebagai Alumni. - Memiliki hak akses penuh untuk membuat, membaca, memperbarui, dan menghapus (CRUD) seluruh konten dinamis, yang mencakup artikel blog, informasi acara (events), dan pengumuman resmi. 5. Aktor Alumni : Sebagai anggota inti komunitas, Alumni mendapatkan akses ke fitur-fitur interaktif setelah terautentikasi. Kemampuan mereka mencakup : <ul style="list-style-type: none"> - Melakukan pendaftaran akun baru secara mandiri dan melakukan login untuk masuk ke dalam sistem. - Mengelola informasi pada profil pribadi mereka, seperti data kontak, pekerjaan, dan informasi relevan lainnya. - Melihat daftar acara yang akan datang dan melakukan registrasi partisipasi secara langsung melalui sistem. - Membaca artikel yang dipublikasikan di blog dan berpartisipasi dalam diskusi dengan cara memberikan komentar. - Melihat pengumuman penting serta memiliki opsi untuk mengunduh atau membagikan informasi tersebut. 6. Aktor Pengunjung (Guest) : Aktor ini merepresentasikan publik atau alumni yang belum login. Akses mereka bersifat pasif dan terbatas untuk: <ul style="list-style-type: none"> - Melihat konten-konten publik yang telah disetujui Admin, seperti informasi umum tentang web alumni, artikel blog, daftar acara, dan pengumuman.

- Mengakses halaman yang berisi informasi kontak pengelola alumni.

Objectives

Tujuan fundamental dari pengembangan sistem ini adalah :

- Membangun Basis Data Terpusat dan Akurat : Menciptakan satu sumber data utama (single source of truth) untuk seluruh alumni SMAN 1 Pemalang melalui mekanisme registrasi mandiri dan pengelolaan profil oleh setiap alumni, mengatasi masalah data usang dan tidak valid.
- Menciptakan Saluran Komunikasi Resmi : Menjadi medium komunikasi yang efisien dan terpercaya dari pengelola alumni kepada seluruh anggota melalui fitur pengumuman dan blog, memastikan informasi penting tersampaikan secara merata.
- Mendorong Keterlibatan dan Partisipasi Aktif : Memfasilitasi dan meningkatkan keterlibatan alumni dalam berbagai kegiatan komunitas dengan menyediakan informasi acara yang terstruktur dan kemudahan untuk mendaftar secara online, serta mendorong diskusi melalui fitur komentar pada blog.
- Memperkuat Jaringan dan Solidaritas Alumni : Menjadi wadah digital yang memungkinkan alumni untuk tetap terhubung, berbagi informasi, dan membangun jaringan profesional maupun sosial yang kuat, sehingga mempererat rasa solidaritas dan kekeluargaan di antara alumni SMAN 1 Pemalang.

Deliverables

- Platform Web Ganissa : Sebuah aplikasi web yang dapat diakses melalui peramban web (browser) dari berbagai perangkat (desktop, tablet, mobile) dengan desain responsif.
- Basis Data Alumni Terpusat : Sistem database yang terstruktur dan aman untuk menyimpan seluruh data alumni (profil, informasi kontak, riwayat pendidikan, karier), data artikel blog, detail acara, serta pengumuman.
- Modul Registrasi dan Autentikasi Pengguna :
 - o Fungsionalitas registrasi akun baru secara mandiri bagi alumni.
 - o Sistem login dan logout yang aman untuk setiap aktor (Admin dan Alumni).
- Modul Pengelolaan Profil Alumni:
 - o Alumni dapat melihat dan memperbarui informasi pribadi mereka (kontak, pekerjaan, domisili, angkatan, dll.).
 - o Admin memiliki kemampuan untuk melihat, menambah, memperbarui, dan menghapus (CRUD) data profil alumni.
- Modul Artikel Blog:
 - o Admin dapat membuat, membaca, memperbarui, dan menghapus (CRUD) artikel blog.
 - o Alumni dan Pengunjung dapat membaca artikel blog.
 - o Alumni dapat memberikan komentar pada artikel blog.
- Modul Informasi Acara (Events):
 - o Admin dapat membuat, membaca, memperbarui, dan menghapus (CRUD) informasi acara (nama, deskripsi, tanggal, waktu, lokasi, kuota).
 - o Alumni dan Pengunjung dapat melihat daftar acara yang akan datang.
 - o Alumni dapat melakukan registrasi partisipasi pada acara secara online.
- Modul Pengumuman Resmi:
 - o Admin dapat membuat, membaca, memperbarui, dan menghapus (CRUD) pengumuman resmi.
 - o Alumni dan Pengunjung dapat melihat pengumuman penting.
 - o Pengumuman dapat memiliki opsi untuk diunduh atau dibagikan.
- Antarmuka Admin Panel: Dashboard khusus untuk Admin dengan fitur manajemen pengguna, konten, dan laporan sederhana.
- Laporan Dasar : Laporan mengenai jumlah alumni terdaftar dan jumlah partisipan acara.
- Dokumentasi Proyek : Meliputi dokumen desain sistem, panduan penggunaan untuk Admin, dan panduan dasar untuk Alumni.

IV. Metodologi, Analisis Dan WBS

i. Metodologi

Pembangunan Sistem Informasi Alumni SMAN 1 Pemalang (Ganissa) ini dapat menggunakan metodologi System Development Life Cycle (SDLC). Metodologi ini merupakan pendekatan sistematis yang mencakup beberapa tahapan penting dalam pengembangan sistem informasi. Tahapan System Development Life Cycle (SDLC) :

A) Analisis Kebutuhan (Requirement Analysis)

Pada tahap ini, tim pengembang mengumpulkan dan menganalisis kebutuhan pengguna untuk menentukan fitur dan fungsi yang harus ada dalam sistem informasi Alumni SMAN 1 Pemalang (Ganissa). Ini melibatkan diskusi dengan pemangku kepentingan untuk memahami tujuan dan harapan mereka terhadap sistem.

B) Desain Sistem (Design Analysis)

Setelah kebutuhan diidentifikasi, tahap desain dilakukan untuk merancang arsitektur sistem. Ini mencakup pembuatan prototipe dan dokumentasi yang menjelaskan bagaimana sistem akan berfungsi dan berinteraksi dengan pengguna. Desain ini memberikan gambaran umum kepada pengguna mengenai sistem yang akan dibangun.

C) Pengembangan (Development)

Dalam fase ini, tim pengembang mulai menulis kode berdasarkan desain yang telah disepakati. Proses ini melibatkan pembuatan berbagai komponen sistem sesuai dengan spesifikasi yang telah ditentukan sebelumnya. Tahap ini biasanya merupakan bagian terpanjang dari siklus pengembangan.

D) Pengujian (Testing)

Setelah pengembangan selesai, sistem diuji untuk memastikan bahwa semua fungsi berjalan dengan baik dan memenuhi kebutuhan pengguna. Pengujian meliputi verifikasi kualitas perangkat lunak dan memastikan bahwa sistem dapat menerima input serta menghasilkan output yang sesuai harapan.

E) Implementasi (Implementation)

Setelah pengujian berhasil, sistem diimplementasikan dan mulai digunakan oleh staf perpustakaan dan siswa. Pada tahap ini, pelatihan juga

diberikan kepada pengguna agar mereka dapat memanfaatkan sistem dengan efektif.

F) Pemeliharaan (Evolution/Maintenance)

Setelah sistem berjalan, tahap pemeliharaan dimulai untuk menangani masalah yang mungkin muncul, melakukan perbaikan, dan menambahkan fitur baru sesuai permintaan pengguna atau perubahan lingkungan eksternal. Pemeliharaan penting untuk menjaga agar sistem tetap relevan dan berfungsi dengan baik.

ii. Analisis Kelayakan, Risiko, Rancangan Desain, dan Rencana Pengujian untuk Sistem Informasi Alumni SMAN 1 Pematang Jaya (Ganissa)

a. Analisis Risiko

Strenghts	<ol style="list-style-type: none">1. Menyediakan layanan manajemen data alumni yang cepat, mudah, dan aman.2. Menyajikan informasi alumni secara terstruktur dan real-time, termasuk data pendidikan, pekerjaan, dan kontak.3. Kompatibilitas sistem Ganissa yang berbasis web untuk berbagai perangkat.4. Fitur notifikasi pengingat (misalnya event reuni, update profil) memudahkan keterlibatan pengguna.5. Sistem berbasis web dengan autentikasi pengguna sehingga fleksibel diakses dari mana saja.
-----------	--

Weakness	<ol style="list-style-type: none"> 1. Ketergantungan pada SDM teknis dapat menjadi kendala jika terjadi gangguan sistem. 2. Masih terbatasnya data alumni yang lengkap dan tervalidasi. 3. Kemungkinan kendala teknis seperti bug atau crash dapat mengganggu kenyamanan pengguna. 4. Kurangnya pelatihan rutin bagi operator sekolah atau admin sistem dalam pengelolaan platform. 5. Fitur analisis data alumni belum dioptimalkan untuk kebutuhan sekolah dan organisasi alumni.
Opportunities	<ol style="list-style-type: none"> 1. Berpeluang menjalin kemitraan dengan instansi pendidikan, perguruan tinggi, dan dunia kerja untuk kolaborasi tracer study dan lowongan kerja alumni. 2. Dukungan pemerintah dalam program digitalisasi data pendidikan dan alumni. 3. Potensi integrasi dengan e-sertifikat, atau event alumni nasional. 4. Meningkatnya tren digitalisasi sekolah di seluruh Indonesia. 5. Peluang menarik minat sekolah lain di wilayah Pematang untuk mengadopsi Ganissa sebagai sistem alumni standar.
Threats	<ol style="list-style-type: none"> 1. Ancaman terhadap keamanan data pribadi alumni (seperti nama, alamat, pekerjaan, dll).

	<ol style="list-style-type: none"> 2. Adanya sistem alumni lain yang lebih lengkap dan kompetitif. 3. Regulasi baru tentang privasi dan pengelolaan data pendidikan dari pemerintah. 4. Ketergantungan pada koneksi internet bisa menjadi kendala di wilayah dengan jaringan terbatas. 5. Potensi kesalahan input data yang menyebabkan ketidaktepatan informasi alumni.
--	--

Strategi berdasarkan analisis SWOT :

<p>Strategi S-O (Strength – Opportunities)</p>	<ol style="list-style-type: none"> 1. Menambahkan modul tracer study alumni berdasarkan data pekerjaan mereka untuk mendukung kebutuhan sekolah dan dinas pendidikan. 2. Menyediakan fitur direktori alumni dengan filter berdasarkan angkatan, profesi, atau lokasi. 3. Menyediakan dashboard statistik alumni secara visual (grafik profesi, domisili, pendidikan). 4. Menawarkan katalog profil alumni inspiratif yang bisa diakses publik untuk memperkuat citra sekolah.
--	---

	<ol style="list-style-type: none"> 5. Membuka akses integrasi dengan layanan pihak ketiga (misalnya job portal, universitas, e-learning).
<p>Strategi W-O (Weakness - Opportunities)</p>	<ol style="list-style-type: none"> 1. Mengembangkan fitur pengumpulan data offline (formulir input manual) yang bisa diunggah saat koneksi tersedia. 2. Menyediakan pelatihan digital literasi dan manajemen sistem bagi admin/operator sekolah. 3. Membangun fitur validasi ganda untuk data alumni (verifikasi email/WA). 4. Menyediakan dukungan multi-sekolah jika ingin direplikasi ke SMA lain di Pematang. 5. Menambahkan integrasi form Google/Excel API untuk percepatan input data awal.
<p>Strategi S-T (Strength - Threats)</p>	<ol style="list-style-type: none"> 1. Menerapkan teknologi enkripsi dan two-factor authentication (2FA) untuk menjaga keamanan data alumni. 2. Menyediakan log aktivitas admin dan pengguna untuk pelacakan jika terjadi pelanggaran. 3. Menyediakan fitur unggulan seperti sertifikat digital keaktifan alumni, sistem voting internal (misal pemilihan ketua alumni). 4. Berkoordinasi dengan dinas pendidikan untuk menjaga agar sistem sesuai standar operasional dan kebijakan pemerintah.

<p>Strategi W-T (Weakness - Threats)</p>	<ol style="list-style-type: none"> 1. Menyusun SOP (Standard Operating Procedure) pengelolaan risiko seperti kesalahan input, gangguan server, dan backup data otomatis. 2. Membuat modul disaster recovery untuk pemulihan sistem cepat saat terjadi kendala. 3. Menyediakan dokumentasi lengkap dan panduan penggunaan sistem untuk sekolah dan alumni. 4. Melakukan pengujian berkala terhadap fitur utama dan menerima masukan dari pengguna melalui feedback form. 5. Melakukan sosialisasi intensif kepada alumni agar mereka aktif memperbarui data mandiri.
--	--

b. Rancangan Desain

- Desain Sistem

1. Arsitektur Sistem

- Frontend : React.js untuk pengalaman pengguna yang interaktif.
- Backend : PHP (Laravel) atau Node.js untuk logika sistem.
- Database : MySQL untuk penyimpanan data.

2. Fitur Utama

- Manajemen buku (tambah, edit, hapus).
- Manajemen anggota perpustakaan (siswa dan guru).
- Pencatatan transaksi peminjaman dan pengembalian.
- Laporan keterlambatan dan denda.

3. Integrasi

- Barcode scanner untuk memudahkan input data buku.
 - Kemungkinan integrasi dengan sistem administrasi sekolah.
- Wireframe/Mockup
1. Dashboard Pustakawan :
 - Menu utama untuk navigasi cepat (buku, anggota, transaksi).
 - Tampilan laporan harian, mingguan, dan bulanan.
 2. Form Peminjaman :
 - Input data anggota dan buku.
 - Peringatan jika buku terlambat dikembalikan.
- c. Rencana Pengujian
- Pengujian Fungsional
1. Peminjaman Buku
 - Memastikan fitur pencatatan peminjaman berjalan dengan benar.
 - Verifikasi bahwa data buku dan anggota yang dimasukkan terhubung secara akurat dalam database.
 2. Pengembalian Buku
 - Memastikan sistem dapat menghitung denda otomatis berdasarkan keterlambatan pengembalian.
 - Memastikan pembaruan status buku (tersedia/tidak tersedia) dilakukan dengan benar.
 3. Manajemen Data
 - Pengujian pada fitur tambah, edit, dan hapus data buku dan anggota.
 - Memastikan validasi input data untuk menghindari kesalahan (contoh: ISBN duplikat).

- Pengujian Non-Fungsional
 1. Kinerja
 - Memastikan sistem dapat menangani 100-200 transaksi per hari tanpa adanya lag atau penurunan performa.
 - Simulasi pemuatan data untuk memastikan waktu respon tetap optimal.
 2. Keamanan
 - Pengujian autentikasi dan otorisasi (contoh: pustakawan hanya dapat mengakses fitur tertentu).
 - Uji penetrasi untuk melindungi data sensitif, seperti informasi anggota perpustakaan.
 3. Keandalan
 - Simulasi gangguan listrik untuk memastikan data tidak hilang (menguji fitur auto-save atau backup).
 - Uji ketersediaan sistem saat terjadi gangguan internet dengan mode offline.

- Uji Pengguna
 1. Tahap Uji :
 - Melibatkan pustakawan dan siswa sebagai pengguna utama.
 - Memberikan skenario nyata (peminjaman, pengembalian, pencarian buku) untuk dilakukan oleh pengguna.
 2. Pengumpulan Umpan Balik :
 - Merekam pengalaman pengguna terkait kecepatan, kemudahan navigasi, dan kesesuaian fitur.
 - Mengidentifikasi masalah teknis atau fitur yang sulit dipahami pengguna.

- Dokumentasi Hasil Pengujian

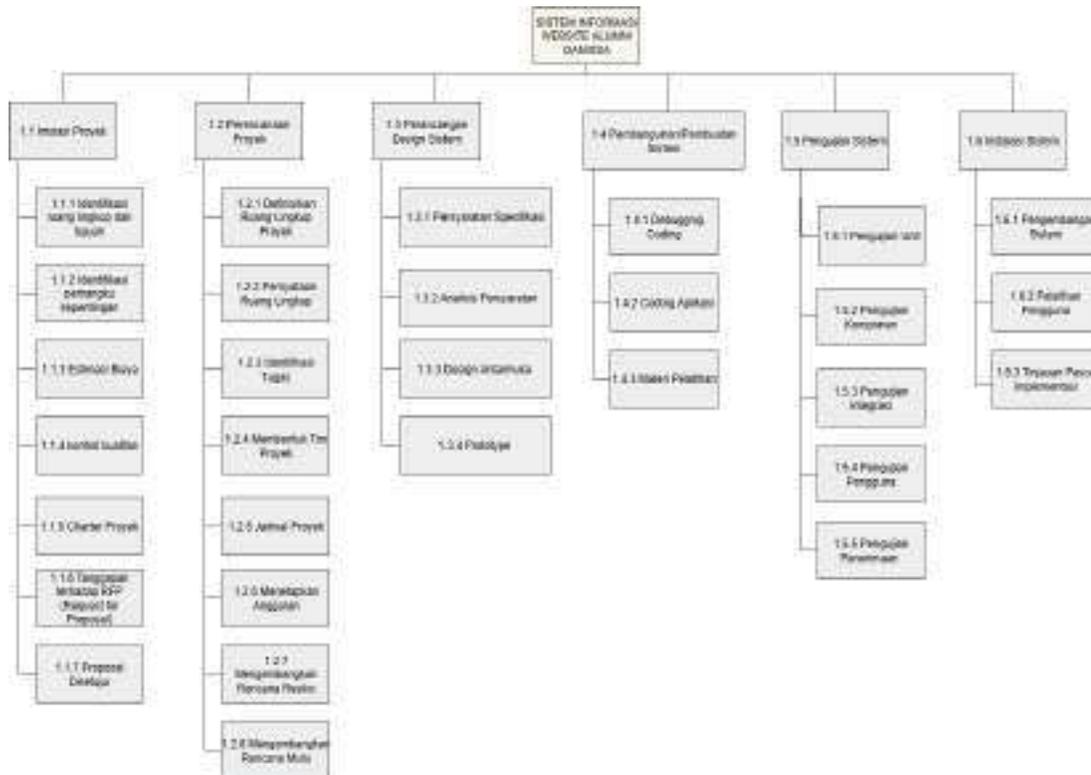
1. Laporan Pengujian

- Setiap bug atau masalah dicatat, termasuk deskripsi dan langkah reproduksi.
- Prioritaskan perbaikan berdasarkan tingkat keparahan (tinggi, sedang, rendah).

2. Tindak Lanjut

- Membuat daftar perbaikan untuk implementasi sebelum peluncuran sistem.
- Uji ulang setelah perbaikan untuk memastikan masalah terselesaikan.

iii. WBS



Tahap ini adalah fondasi awal proyek, di mana ide proyek diresmikan dan kelayakannya dieksplorasi.

1.1 Identifikasi ruang lingkup dan tujuan: Pada sub-bagian ini, tim proyek akan secara jelas mendefinisikan batas-batas proyek. Untuk "Sistem Informasi Alumni SMAN 1 Pemalang," ruang lingkungnya mungkin mencakup fitur-fitur utama seperti pendaftaran alumni, profil alumni (data pribadi, pendidikan, karier), direktori alumni, modul berita/blog, modul acara, dan modul pengumuman. Tujuannya bisa jadi untuk mempermudah komunikasi antar alumni, menyediakan platform informasi, memfasilitasi *networking*, dan mendukung kegiatan sekolah/alumni.

1.2 Identifikasi pemangku kepentingan: Ini melibatkan pengenalan semua individu atau kelompok yang akan terpengaruh oleh proyek atau memiliki kepentingan dalam keberhasilannya. Untuk proyek ini, pemangku kepentingan utama meliputi: manajemen

SMAN 1 Pemalang, staf IT sekolah, alumni SMAN 1 Pemalang (sebagai pengguna akhir), calon alumni (siswa kelas akhir), komite alumni, dan pengembang/tim proyek itu sendiri.

1.3 Estimasi Biaya: Menentukan perkiraan sumber daya finansial yang dibutuhkan untuk menyelesaikan proyek. Ini mencakup perkiraan biaya pengembangan (upah tim, lisensi *software* jika ada), infrastruktur (server, domain), pemeliharaan, pelatihan, dan biaya tak terduga.

1.4 Kontrol kualitas: Mendefinisikan standar dan prosedur untuk memastikan bahwa hasil proyek memenuhi persyaratan yang ditetapkan. Ini bisa berupa metrik kualitas untuk *software* (misalnya, jumlah *bug* yang diizinkan), kecepatan respons sistem, atau kepuasan pengguna.

1.5 Charter Proyek: Dokumen formal yang secara resmi mengotorisasi keberadaan proyek dan memberikan manajer proyek wewenang untuk menerapkan sumber daya organisasi untuk kegiatan proyek. Ini akan merangkum ruang lingkup, tujuan, pemangku kepentingan, anggaran estimasi, dan daftar tim inti.

1.6 Tanggapan terhadap RFP (Request for Proposal): Jika proyek ini merupakan hasil dari permintaan pihak eksternal, tahap ini melibatkan penyusunan proposal rinci yang menjelaskan bagaimana tim proyek akan memenuhi kebutuhan yang digariskan dalam RFP, termasuk metodologi, jadwal, dan biaya.

1.7 Proposal Disetujui: Ini adalah titik balik penting di mana semua pemangku kepentingan kunci menyetujui rencana proyek yang diajukan, memberikan lampu hijau untuk melanjutkan ke tahap berikutnya.

2. Perencanaan Proyek

Tahap ini berfokus pada pengembangan peta jalan yang rinci untuk mencapai tujuan proyek.

2.1 Definisikan Ruang Lingkup Proyek: Memperjelas dan mendokumentasikan tujuan proyek, hasil yang dapat disampaikan, dan batasan-batasannya, seringkali dalam dokumen *Statement of Work (SOW)* atau dokumen ruang lingkup proyek. Ini lebih detail dari pada inisiasi.

2.2 Pernyataan Ruang Lingkup: Dokumen tertulis yang sangat detail mengenai apa yang akan dan tidak akan dilakukan dalam proyek. Ini berfungsi sebagai landasan untuk menghindari *scope creep*.

2.3 Identifikasi Tugas: Memecah ruang lingkup proyek menjadi tugas-tugas yang lebih kecil dan terkelola. Ini bisa menggunakan *Work Breakdown Structure (WBS)* untuk memetakan semua pekerjaan yang perlu dilakukan. Contoh: "Desain database", "Buat modul profil", "Uji coba login".

2.4 Membentuk Tim Proyek: Mengidentifikasi dan mengalokasikan peran serta tanggung jawab untuk setiap anggota tim. Untuk sistem informasi, ini bisa termasuk *Project Manager, Business Analyst, UI/UX Designer, Backend Developer, Frontend Developer, Quality Assurance Tester*, dll.

2.5 Jadwal Proyek: Membuat *timeline* proyek yang merinci kapan setiap tugas akan dimulai dan selesai. Ini bisa menggunakan Gantt chart atau metode *Critical Path Method (CPM)*.

2.6 Menetapkan Anggaran: Mengembangkan anggaran yang terperinci berdasarkan estimasi biaya dan jadwal, mengalokasikan dana untuk setiap tugas atau fase proyek.

2.7 Mengembangkan Rencana Risiko: Mengidentifikasi potensi risiko (teknis, operasional, finansial, keamanan data, dll.) yang dapat memengaruhi proyek dan mengembangkan strategi mitigasi untuk setiap risiko.

2.8 Mengembangkan Rencana Mutu: Mendetailkan bagaimana kualitas akan dikelola di sepanjang proyek, termasuk standar yang akan dipatuhi, proses pengujian, dan bagaimana *feedback* akan diintegrasikan.

3. Perancangan (Desain Sistem)

Fase ini mengubah persyaratan menjadi rencana arsitektur dan desain yang lebih konkret.

3.1 Persyaratan Spesifikasi: Menguraikan semua fungsionalitas dan non-fungsionalitas yang diharapkan dari sistem secara terperinci. Ini adalah dokumen yang sangat detail tentang apa yang harus dilakukan sistem. Contoh: "Sistem harus dapat memverifikasi alumni menggunakan NIS," "Sistem harus dapat menampilkan profil alumni secara publik dengan persetujuan alumni."

3.2 Analisis Persyaratan: Menganalisis persyaratan yang dikumpulkan untuk memastikan konsistensi, kelengkapan, dan kelayakan. Ini mungkin melibatkan pembuatan diagram alur, diagram kasus penggunaan (UML), atau cerita pengguna (*user stories*) untuk memvisualisasikan bagaimana pengguna akan berinteraksi dengan sistem.

3.3 Desain Antarmuka: Membuat sketsa, *wireframes*, dan *mockups* visual tentang bagaimana antarmuka pengguna (UI) sistem akan terlihat dan berfungsi. Ini mencakup tata letak halaman, elemen navigasi, skema warna, dan tipografi.

3.4 Prototype: Membuat versi awal atau model fungsional dari sistem (bisa berupa *low-fidelity* atau *high-fidelity*) untuk mendapatkan *feedback* awal dari pemangku kepentingan dan menguji konsep desain sebelum pengembangan penuh.

4. Pembangunan / Pembuatan Sistem

Ini adalah fase di mana kode dan komponen sistem sebenarnya dibuat.

4.1 Debugging Coding: Proses mengidentifikasi, menganalisis, dan memperbaiki kesalahan (bug) dalam kode program. Ini dilakukan secara iteratif sepanjang proses *coding*.

4.2 Coding Aplikasi: Penulisan kode sumber untuk semua modul dan fitur sistem berdasarkan desain yang telah disetujui. Ini mencakup pengembangan *backend* (API, logika bisnis, database) dan *frontend* (antarmuka pengguna interaktif).

4.3 Materi Pelatihan: Menyiapkan panduan, manual, atau modul pelatihan untuk pengguna akhir (alumni dan admin sekolah) tentang cara menggunakan sistem setelah diimplementasikan.

5. Pengujian Sistem

Fase ini bertujuan untuk menemukan *bug* dan memastikan sistem memenuhi semua persyaratan fungsional dan non-fungsional.

5.1 Pengujian Unit: Menguji komponen individual atau unit kode secara terpisah untuk memastikan mereka berfungsi dengan benar. Dilakukan oleh pengembang.

5.2 Pengujian Komponen: Menguji sekelompok unit yang terintegrasi (misalnya, satu modul penuh) untuk memastikan interaksi antar unit berfungsi.

5.3 Pengujian Integrasi: Menguji interaksi antara berbagai modul atau sub-sistem yang berbeda untuk memastikan mereka bekerja sama dengan mulus. Contoh: Apakah registrasi event berhasil dicatat di profil alumni?

5.4 Pengujian Pengguna (UAT - User Acceptance Testing): Pengguna akhir (perwakilan alumni atau staf sekolah) menguji sistem dalam lingkungan yang realistis untuk memvalidasi bahwa sistem memenuhi kebutuhan bisnis dan persyaratan pengguna.

5.5 Pengujian Penerimaan: Tahap akhir pengujian di mana pemangku kepentingan utama secara formal menerima sistem setelah semua *bug* kritis diperbaiki dan persyaratan terpenuhi.

6. Instalasi Sistem

Fase terakhir proyek, di mana sistem disebar dan mulai digunakan.

6.1 Pengembangan Sistem (Deployment): Proses menginstal dan mengonfigurasi sistem di lingkungan produksi (server live), membuatnya tersedia untuk pengguna akhir. Ini bisa melibatkan migrasi data lama jika ada.

6.2 Pelatihan Pengguna: Melakukan sesi pelatihan untuk alumni dan admin sekolah agar mereka terbiasa dan mahir menggunakan sistem informasi alumni yang baru.

6.3 Tinjauan Pasca Implementasi: Mengevaluasi keberhasilan proyek setelah sistem berjalan selama beberapa waktu, mengidentifikasi pelajaran yang dipetik, dan mengukur apakah tujuan proyek awal telah tercapai. Ini juga bisa menjadi awal dari fase pemeliharaan dan peningkatan berkelanjutan.

V. Jadwal Proyek

Kode Tugas	Tugas	Durasi ()	Ketergantungan
A	Pengumpulan Data	1 Minggu	-
B	Analisis	2 Minggu	A
C	Desain Website	2 Minggu	B
D	Programming	2 Minggu	C
E	Testing	2 Minggu	D
F	Hosting	2 Minggu	E
G	Launching	1 Minggu	F

a) Gantt Chart

NO	Kegiatan	Juni				Juli
		1	2	3	4	1
A	Pengumpulan Data					
B	Analisis					
C	Desain Website					
D	Programming					
E	Testing					
F	Hosting					
G	Launching					

VI. Alokasi dan Sumber Daya

A. Kebutuhan SDM

Name	Role	Responsibilities
Noval Lias Ramadani	Project Manager	Project Manager, Frontend 1, Backend 2
Hasan Nur Rasyid	Project Team	Backend 1, Sekretaris
Andriansyah	Project Team	System Analyst
Farqad Assanji	Project Team	Frontend 2, UI/UX, Bendahara
Destyawan Satria Nugraha	Project Team	Quality Assurance

B. Alokasi Alat

	Nama Alat
Hardware	Komputer Client (jika belum ada)
	LAN
Software	Visual Studio Code
	Database Management System
	Figma
	Flask
	Node js
	GitHub
	Postman
	MS Office

VII. BIAYA DAN LAIN LAIN

A. Biaya Pengembangan Sistem

a) Tim Pengembangan

Komponen	Jumlah Orang	Durasi (bulan)	Gaji per Bulan (Rp)	Total (Rp)
Project Manager	1	4	175,000	700,000
Programmer (Backend)	1	3	130,000	520,000
Programmer (Frontend)	1	3	130,000	520,000
UI/UX Designer	1	1	200,000	200,000
Quality Assurance (QA)	1	2	125,000	250,000
System Analyst	1	3	125,000	375,000
Subtotal				2,565,000

b) Lain-lain

Komponen	Jumlah	Harga per Unit (Rp)	Total (Rp)
Dokumentasi teknis	1 paket	300,000	300,000

Subtotal			300,000
----------	--	--	---------

B. Biaya Perangkat Keras

Komponen	Jumlah	Harga per Unit (Rp)	Total (Rp)
Barcode	1 unit	400,000	400,000
Total Biaya Perangkat Keras	400,000		

C. Biaya Operasional

Komponen	Jumlah		Harga per Bulan (Rp)	Total (Rp)
Hosting/Server Cloud	1 unit		65,000	780,000
Pemeliharaan Website	-		300,000	300,000
Total Biaya Operasional:				1,080,000

D. Total Keseluruhan

Komponen	Total (Rp)
Biaya Pengembangan Sistem	2,865,000
Biaya Perangkat Keras	400,000
Biaya Operasional	1,080,000
Grand Total:	4,345,000

B. Surat Kontrak Kerja dengan Mitra



ZESTIFY CORP
bersama
ALUMNI GANISSA



PERJANJIAN KERJASAMA

PEMBUATAN SISTEM INFORMASI ALUMNI

Yang bertanda tangan di bawah ini :

Nama : Noval Lias R

Jabatan : Project Manager

Alamat : Jl Bausasran, DN3/557

Dalam hal ini bertindak untuk dan atas nama **ZESTIFY** untuk selanjutnya disebut **Pihak Pertama**.

Nama : Yogi Hartono

Jabatan : Ketua Umum

Alamat : Utara, Bojongbata, Kec. Pemalang, Kabupaten Pemalang, Jawa Tengah 52362

Dalam hal ini bertindak untuk dan atas nama **ALUMNI GANISSA** untuk selanjutnya disebut **Pihak Kedua**.

Pihak Pertama dan Pihak Kedua dengan ini menerangkan pembuatan website alumni (untuk selanjutnya disebut perjanjian) dengan ketentuan-ketentuan sebagai berikut:

Pasal 1

Ruang Lingkup Perjanjian

1. Pihak Pertama dalam hal ini bertindak sebagai pengembang dan pembuat kode sumber (source code) Website Alumni Ganissa yang diminta oleh Pihak Kedua.

2. Ruang lingkup pekerjaan Pihak Pertama meliputi desain grafis, template, tata-letak, tipografi, ikon, dan hal lain yang diperlukan dalam lingkup desain Web dan Visual Art sesuai dengan paket yang dipesan oleh Pihak Kedua, serta pengembangan dan penulisan kode program untuk fungsionalitas website.
3. Pihak Pertama tidak bertanggung jawab atas penyediaan layanan hosting, domain, maupun pengelolaan server website. Penyediaan dan pengelolaan hosting serta domain merupakan tanggung jawab Pihak Kedua.

Pasal 2

Hak dan Kewajiban

Dalam perjanjian ini Pihak Pertama dan Pihak Kedua memiliki hak dan kewajiban:

Hak dan Kewajiban Pihak Pertama:

1. Apabila Pihak Kedua menjual kembali atau menggunakan kode website untuk tujuan lain yang berbeda dari yang disepakati, maka harus dengan persetujuan tertulis dari Pihak Pertama.
2. Pihak Pertama berkewajiban untuk menyerahkan program dalam bentuk kode sumber dan aset desain terkait kepada Pihak Kedua setelah terselesainya pengerjaan ini.
3. Pihak Pertama berhak mendapatkan pembayaran sesuai dengan nilai perjanjian yang telah disepakati.
4. Pihak Pertama berhak mendapatkan data dan konten yang lengkap dan tepat waktu dari Pihak Kedua untuk menunjang proses pengerjaan website.

Hak dan Kewajiban Pihak Kedua:

1. Pihak Kedua berkewajiban menyediakan konten, data, dan bahan-bahan lain yang dirasakan perlu untuk pembuatan situs Web secara lengkap dan tepat waktu.
2. Pihak Kedua berkewajiban menyediakan layanan hosting dan domain untuk website alumni.
3. Pihak Kedua berkewajiban melakukan pembayaran sesuai dengan termin yang telah disepakati dalam perjanjian ini.
4. Pihak Kedua berhak mengajukan usulan-usulan dalam pembuatan situs Web yang akan menjadi pertimbangan dari Pihak Pertama.
5. Pihak Kedua berhak menerima kode sumber dan aset desain website setelah seluruh kewajiban pembayaran terpenuhi.

Pasal 3 Nilai Perjanjian

Atas pekerjaannya tersebut maka Pihak Pertama berhak memperoleh pembayaran dari Pihak Kedua berupa uang sejumlah Rp.405.000 sebagai biaya sewa domain dan hosting. Nilai tersebut adalah kesepakatan bersama antara Pihak Pertama dan Pihak Kedua.

Pasal 4

Pembayaran

1. Pada saat perjanjian ini ditandatangani, Pihak Kedua berkewajiban melakukan pembayaran dari nilai yang telah disepakati pada Pasal 3 yaitu sebesar Rp.500.000.
2. Jika terdapat kekurangan pembayaran website yang dipesan oleh Pihak Kedua dan sudah selesai dikerjakan oleh Pihak Pertama maka bisa ditransfer ke rekening 1186085305 a.n. Noval Lias Ramadhani.

Pasal 5

Waktu Pengerjaan

1. Waktu pengerjaan terhitung sejak ditandatanganinya surat perjanjian ini oleh Pihak Pertama dan Pihak Kedua dan pembayaran uang muka telah diterima oleh Pihak Pertama.
2. Jangka waktu pengerjaan kode website oleh Pihak Pertama adalah maksimal dalam **1 bulan** waktu kalender, terhitung pada tanggal **29 Mei** sampai dengan **21 Juni**
3. Bila jangka waktu yang telah disepakati pada pasal 5.2 tidak tercapai karena kelalaian Pihak Pertama, Pihak Kedua berhak membatalkan pembuatan situs Web yang telah dipesan atau mengajukan pengurangan nilai perjanjian yang telah disepakati.
4. Bila Pihak Kedua membatalkan pembuatan situs Web sebelum jangka waktu di atas tanpa alasan yang sah, uang muka yang telah dibayarkan menjadi milik Pihak Pertama. Namun, jika pembatalan karena Pihak Pertama tidak dapat menyelesaikan pekerjaan sesuai jadwal dan kesepakatan, Pihak Pertama berkewajiban mengembalikan uang muka sebesar 25%.
5. Jangka waktu pengerjaan desain dan kode program pada Pasal 5.2 dapat berubah bila Pihak Kedua terlambat memberikan data-data atau persetujuan yang diperlukan kepada Pihak Pertama.

Pasal 6

Garansi

1. Pihak Kedua berhak meminta perbaikan atas fungsionalitas dan bug pada kode website yang telah dibuat oleh Pihak Pertama dalam waktu **1 Bulan** terhitung sejak penyerahan final kode.
2. Garansi pada Pasal 6.1 berlaku apabila dalam batas kewajaran, yaitu perbaikan atas kesalahan atau kerusakan yang timbul dari pengembangan kode oleh Pihak Pertama, bukan karena perubahan yang dilakukan oleh Pihak Kedua atau masalah pada hosting/server.
3. Garansi berlaku selama layout atau desain dan fungsionalitas dasar tetap/sama dengan desain awal dan spesifikasi yang disepakati.

Pasal 7

Ketentuan Lain

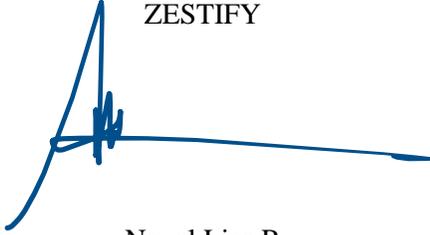
1. Hal-hal yang belum diatur dalam perjanjian ini akan diatur kemudian secara bersama dalam suatu Addendum/Suplemen dengan catatan:
2. Perubahan maupun penambahan dari Perjanjian ini hanya berlaku apabila dituangkan dalam Addendum/Suplemen Perjanjian yang ditandatangani oleh kedua pihak dan merupakan bagian yang tidak terpisahkan dari perjanjian ini.
3. Surat Asli perjanjian ini dibuat rangkap 2 (dua) masing-masing bermaterai cukup dan telah ditandatangani asli oleh kedua belah pihak sehingga masing-masing mempunyai kekuatan hukum yang sama, dengan masing-masing pihak mendapatkan satu eksemplar.

Perselisihan

Apabila timbul perselisihan antara Pihak Pertama dan Pihak Kedua mengenai perjanjian ini yang tidak dapat diselesaikan secara damai dan musyawarah, kedua belah pihak memilih Pengadilan Negeri **Yogyakarta** untuk diselesaikan oleh pengadilan tersebut.

Demikian perjanjian ini disepakati dan ditandatangani di Yogyakarta pada tanggal **30 Mei 2025**.

Pihak Pertama:
ZESTIFY



Noval Lias R
Project Manager

Pihak Kedua:
ALUMNI GANISSA



Edi Sutriyono, S.E., M.Si
Ketua Umum

C. Log Book Kelompok sudah terisi minimal 7 x

1. Sebelum UTS

LOG BOOK MANAJEMEN PROYEK TEKNOLOGI INFORMASIMAHASISWA
PROGRAM STUDI SI INFORMATIKA UNIVERSITAS AHMAD DAHLAN T.A 2024/2025
(WAJIB DISI DAN MASUK DALAM PENILAIAN)

Nama Kelompok NIFT : Zesty		
No	NIM	NAMA
1.	2200018988	Noval Uas Ramadani
2.	2200018988	Haran Nur Rayini
3.	2200018988	Andriangali
4.	2200018982	Fariqul Asyari
5.	2200018995	Destawian Satria Nugraha
Judul Manajemen Tugas Proyek : Sistem Informasi Sekolah Berbasis Web		
Dosen Pembimbing : Dr. Murnito, SSI, Mkom		

Format/Paragraf LogBook

1. Log book ditulis per minggu
2. Log book ditulis terapan
3. Setiap kegiatan dicatat oleh mitra atau Client / dosen pembimbing MP3
4. Log book per minggu di paraf oleh dosen pembimbing kelas MP3
5. Jumlah terisi minimal 4x sebelum UTS

Logbook Minggu 1 of 7 (sebelum UTS)

No	Kegiatan	Waktu Pelaksanaan		Lokasi	Kendala, Rencana Perubahan (jika ada)	Paraf MENTOR/CLIENT	Paraf Dosen Pembimbing MP3
		Hari/Tgl.	Durasi (jam)				
1.	Pembahasan dan pertemuan tim untuk membahas ide proyek yang akan dibuat dan pembagian peran	Senin, 17 Maret 2025	4 jam	Deltafile proyek awal dan pembagian peran sementara	Perbedaan preferensi ide proyek antara anggota		
2.	Pertemuan tim untuk berkolaborasi dan diskusi	Senin, 24 Maret 2025	4 jam	Deltafile fitur awal	Perbedaan pendapat mengenai prioritas fitur		

3	Koordinasi awal dengan mitra terkait isi project dan fitur yang akan dibuat.	Selasa, 20 Maret 2025	2 jam	Pemahaman awal kebutuhan dan harapan mitra terhadap sistem.	Esensi/kebutuhan mitra terkait sistem.		
4	Pembahasan terkait requirements project dan fitur sistem berdasarkan kebutuhan Mediacare mitra.	Selasa, 08 April 2025	2 jam	Perbaikan terhadap isi project dan fitur sistem yang diinginkan oleh mitra.			
5	Proses Persetujuan dengan Mitra - Presentasi final untuk isi project dan fitur sistem untuk persetujuan.	Senin, 14 April 2025	2 jam	Persetujuan mitra terkait project yang akan dibuat dan fitur sistem.			
6	Pertemuan tim untuk finalisasi rencana kerja berdasarkan project yang diberikan dan pembagian pekerjaan.	Senin, 21 April 2025	3 jam	Rencana kerja detail, pembagian tugas software, dan jadwal proses.	Sinkronisasi pemahaman semua anggota terhadap scope yang diberikan.		

Catatan Pembimbing Lapangan/Dosen Pembimbing NPTU/Dosen Pengampu Kelas NPTU.

.....

.....

.....

Dosen Pengampu Kelas NPTU

Guntur Maulana Zamron, S.Sc., M.Eng.
NIP. 60381172

Yogyakarta, 15 Mei 2025

Ketua NPTU

Novel Lis Bernadeti

2. Setelah UTS

LOG BOOK MANAJEMEN PROYEK TEKNOLOGI INFORMASIMAJALISURWA PROGRAM STUDI SE INFORMATIKA UNIVERSITAS AHMAD DAHLAN TA 2024/2025 (MULAI DISI DAN MASUK DALAM PENULISAN)

Nama Kelompok NPTU - Joddy		
No	NIM	NAMA
1.	2200018085	Novel Lis Bernadeti
2.	2200018060	Hutan Nur Rasyid
3.	2200018080	Amriantiah
4.	2200018092	Fanipati Arandi
5.	2200018095	Dedyowati Satria Nugraha

Judul Manajemen Tugas Proyek : Sistem Informasi Akumulasi Garis Perakung

Dosen Pembimbing : Dr. Marelis, S.Si., M.Eng.

Daftar Pekerjaan Individu

1. Log book awal, per minggu
2. Log book akhir semester
3. Sistem laporan akumulasi Garis Perakung / sistem pembimbing NPTU
4. Log book per minggu/akhir each minggu setiap minggu kelas NPTU
5. Laporan perkembangan kinerja Tim Seleksi Sistem Informasi NPTU

Logbook Minggu 3 s.d 11 Setelah UTS

NO	Kegiatan	Waktu Pelaksanaan		Isi	Berkas, Rencana Perubahan (jika ada)	Tipe (MIRAS/CLIENT)	Paraf Dosen Pembimbing NPTU
		Hari/Tgl	Durasi (jam)				
1	Pertemuan tim untuk membahas kendala proyek dengan mitra awal dan konsultasi dengan dosen pembimbing terkait perubahan proyek.	Selasa, 27 Mei 2025	2 jam	Tim capaian awal terkait ke proyek baru "Sistem Informasi Akumulasi Garis Perakung" setelah melakukan analisis kebutuhan dan persetujuan dari dosen pembimbing.	Berkas: Proyek awal diberikan mitra tidak memiliki "Perakung" setelah melakukan analisis kebutuhan. Rencana Perubahan: Mengganti total proyek dan mitra memulai kembali dari tahap awal dan penyiapan untuk sistem baru.		

2	Perencanaan tim untuk kick-off proyek baru re-analisa kebutuhan (System Analysis), perancangan ulang UI/UX dan penyusunan arsitektur teknis (Backend & Frontend)	Junat, 30 Mei 2025	4 jam	Dokumen analisis kebutuhan awal, wireframe dasar, dan struktur proyek yang baru untuk 'Genesa' telah dibuat.	Proses pengorganisasian menjadi lebih terstruktur karena fokus dimulai dari awal. Perlu perhatian kepa tim yang tinggi.		
3	Tahap pengembangan (Development): Tim Backend membangun model, migrasi, dan controller. Tim Frontend mulai mengimplementasikan desain halaman publik dan dashboard. Dan System analis membantu struktur pengembangan agar sesuai dengan fitur apa saja yang sudah dibangun.	Junat, 30 Jan 2025	4 jam per hari tempo dengan 13 Juli	Struktur database terbentuk, dan antarmuka utama (landing page & dashboard) berhasil diimplementasikan secara statis.	Memastikan koordinasi antara Dev dan UI/UX dengan implementasi kode frontend.		
4	Melanjutkan saat pengembangan untuk integrasi frontend dan Backend. Tim Quality Assurance (QA) mulai menyusun skenario pengujian awal untuk fitur baru.	Sabtu, 14 Jan 2025	8 jam	Beberapa fitur - fitur berhasil diintegrasikan. Dokumen rencana pengujian mulai terbentuk.	Tersebut beberapa endpoint backend yang perlu disiapkan untuk memenuhi kebutuhan data di frontend.		
5	Tim QA melakukan pengujian fungsional tahap pertama Black Box Testing. Tim developer melakukan perbaikan bug berdasarkan laporan dari tim QA.	Sabtu, 26 Jan 2025	8 jam	Laporan hasil pengujian awal dibuat.	Monevroni waktu antara perbaikan berbasis bug ini dan melanjutkan dengan pengembangan fitur baru berhasil diidentifikasi dan diperbaiki dengan baik.		

6	Finalisasi seluruh fitur users, dan secara kumulatif menyusun Laporan Akhir Proyek (LAPRO) dan menyiapkan materi untuk presentasi dan bentuk resmi proyek.	Sabtu, 05 Juli 2025	8 jam	Sistem dinyatakan fungsional dan siap digunakan. Laporan akhir dan slide presentasi telah rampung.	Keberhasilan seluruh realisasi terdapat di dalam satu laporan yang bahas.		
7	Tim melakukan hambatan jika dengan dosen pembimbing, diartikan dengan orang tenaga proyek kepada mitra.	13 Juli 2025	1 jam	Proses Sistem Informasi Akutis Genesa Panahang berhasil diartikan dan diartikan oleh mitra. Proyek dinyatakan selesai.			

Catatan Pembimbing Lapangan (Dosen Pembimbing NPT) / Dosen Pengampu kelas NPT :

Dosen Pengampu kelas NPT

Gustur Maulana Zamroli, S.Sc., M. Kom
NIP. 90181117

Pogiparta, 25 Juli 2025

Ketua NPT

Novia Eka Ranaidai

D. Logbook Individu sudah terisi minimal 7x

1. Noval Lias Ramadani (2200018083) – Pm – Be2 – FE1

1) Sebelum UTS

LOG BOOK INDIVIDU MANAJEMEN PROYEK TEKNOLOGI INFORMASI MAHASISWA
PROGRAM STUDI SI INFORMATIKA UNIVERSITAS AHMAD DAHLAN T.A 2024/2025

Nama Kelompok: MFT1 - Zed95		
No	NIM	NAMA
1	2200018083	Noval Lias Ramadani
Judul Manajemen Tugas Proyek ; Sistem Informasi Sekolah Berbasis Web		
Dosen Pembimbing : Dr. Makhjo, S.S., M.Kom.		

- Format Penulisan Log Book
1. Log book ditulis per minggu
 2. Log book ditulis tangan
 3. Setiap kegiatan di paraf oleh/Was atau Dosen / dosen pembimbing MFT
 4. Log book per minggu di paraf oleh dosen pengampu/koby MFT
 5. Jumlah kegiatan minimal 3x Sebelum UTS

Logbook Minggu 1 sd 7 (sebelum UTS)

No	Kegiatan	Waktu Pelaksanaan		Hasil	Keributan, Perkiraan Perubahan (Jika ada)	Paraf NMT/A/CLIENT	Paraf Dosen Pembimbing MFT
		Hari/Tgl	Durasi (jam)				
1	Proyek Manager Memimpin dan melakukan koordinasi dengan tim terkait ide-proyek yang akan dibuat dan pembagian peran	Senin, 17 Maret 2025	4 jam	Daftar ide proyek awal dan pembagian peran sementara	Perbedaan pendapat ide proyek antar anggota.		
2	Proyek Manager Memimpin pertemuan tim terkait pembahasan fitur (sistem)	Selasa, 24 Maret 2025	4 jam	Daftar ide fitur awal	Perbedaan pendapat mengenai prioritas fitur.		
3	Proyek Manager Melakukan koordinasi dengan mitra untuk tahap awal	Selasa, 25 Maret 2025	2 jam	Perencanaan awal kolaborasi dan harapan mitra terhadap sistem.	Beberapa kebutuhan mitra belum jelas.		

4	Project Manager (Memimpin pencapaian proyek, food review, video, dan foto dan lain-lain)	Sabtu, 04 April 2025	2 jam	Pertemuan terhadap ide project dan fitur sistem yang diinginkan oleh MITI			
5	Project Manager (Melakukan perencanaan dan penentuan task final di project dan fitur sistem)	Senin, 14 April 2025	2 jam	Penetapan nama terkait project yang akan dibuat dan fitur sistem			
6	Project Manager (Memimpin pertemuan tim untuk analisis rencana kerja berdasarkan project yang dibuat dan finalisasi pembagian jobdesk)	Senin, 21 April 2025	2 jam	Rencana kerja detail, pembagian tugas keanggota, dan jadwal project	Menyebarkan pemahaman semua anggota terhadap scope yang dituju		
7	Backend Developer 2 (Membuat struktur awal project terkait dan setup database)	Sabtu, 26 April 2025	5 jam	Project launch berhasil diupload, koneksi database aktif			
8	Backend Developer 2 (Ditugaskan untuk migrasi database: guru, berita, file foto, dan lain-lain)	Senin, 28 April 2025	2 jam	Model dan migrasi berhasil dibuat dan dijalankan			
9	Frontend Developer 4 (Mengembangkan kode implementasi desain UI ke dalam Blade Template dan layout UI yang telah dikembangkan oleh tim UI/UX untuk halaman dashboard serta profil sekolah dan juga halaman admin fileman)	Sabtu, 2 Mei 2025	6 jam	Halaman dashboard admin dan beranda sekolah selesai dibuat	Ada sedikit styling karena koneksi CSS		

10	Frontend Developer 3 (Melengkapi halaman profil sekolah dan dashboard UI)	Sabtu, 3 Mei 2025	5 jam	Dashboard admin dan profil sekolah sudah terhubung ke backend	Masih perlu perbaikan pada halaman mobile		
11	Frontend Developer 1 (Mengembangkan kode dan menginputkan data yang telah dibuat oleh backend dan memasukkannya ke halaman awal)	Senin, 5 Mei 2025	5 jam	Data berita, galeri, testimoni berhasil ditampilkan	Ilustrasi sedikit styling tambahan		

Dosen Pengampu Kelas MPII

Darius Maulana Zamani, S.Si., M.Ts.
NIP. 90181117

Yogyakarta, 28 Mei 2025

Kelas MPII


(Novel Usi Romadani)

2) Setelah UTS

LOG BOOK INDIVIDU MANAJEMEN PROYEK TEKNOLOGI INFORMASI MAHASISWA PROGRAM STUDI SI INFORMATIKA UNIVERSITAS AHMAD DAHLAN T.A 2024/2025

Nama Kelompok (MPT) : Zeddy		
No	NIM	NAMA
1	2300038082	Rivali Izzat Ramadhani
Judul Manajemen Tugas Proyek (Sistem Informasi Alumni Graesya Pondok)		
Dosen Pembimbing : Dr. Mulyoto, S.Si, M.Kom.		

- Tetap Diisi dan Isilah Buku
1. Isi buku sesuai dengan materi
 2. Isi buku sesuai dengan
 3. Matrik kapan dipertemuan Mitra atau Client / dosen pembimbing MPT
 4. Isi buku agar mings di isi dan diisi dengan pengumpulan MPT
 5. Isi buku dengan rencana dan pekerjaan

Logbook Minggu 4-14 (setelah UTS - Kegiatan Mitra)

No	Kegiatan	Waktu Pelaksanaan		Hasil	Kendala, Rencana Perubahan (jika ada)	Paraf MITRA/ALUM	Paraf Dosen Pembimbing MPT
		Hari/Tgl	Durasi				
1	Proyek Manajer Melakukan koordinasi awal dengan tim terkait kondisi dari mitra awal, melakukan konsultasi dengan dosen pembimbing (Buddy Mentor) mengenai perubahan struktur proyek.	Senin, 27/05/2025	4 jam	Diselesaikan untuk memulai proyek baru "Sistem Informasi Alumni Graesya Pondok". Tim berhasil mengidentifikasi kendala-kendala yang dihadapi mitra dan tim.	Resiko: Mitra awal (MAM) yang sibuk tidak dapat menganggarkan biaya pengembangan proyek sehingga di sisi lain akan kesulitan. Mitraggiat mitra dan tim proyek secara khusus pembimbing.		
2	Melakukan developer 2. Melakukan struktur awal proyek level baru untuk "Graesya". Melakukan setup database dan konfigurasi lingkungan pengembangan.	Senin, 06 Juni 2025	4 jam (MAM)	Proyek selesai untuk memulai berhasil diimplementasikan, kendala database MySQL diidentifikasi dan siap untuk migrasi.	Perlu penyesuaian cepat dan struktur database proyek lama ke yang baru.		

3	Backend Developer 2: Melakukan Backend 1 dalam merancang & membuat model serta fungsi database utama (User, AlumniProfile, Events, Blog, Announcement).	Jumat, 06 Juni 2025	4 jam (Malam)	Model dan migrasi utama untuk fungsionalitas utama alumni berhasil dibuat dan diimplementasikan struktur data yang sudah terbentuk.	Struktur relasi antar tabel lebih kompleks dibandingkan proyek sebelumnya, memerlukan diskusi intensif dengan System Analyst.		
4	Frontend Developer 1: Mengembangkan implementasi desain UI (Buddy Template) untuk halaman-halaman publik (Guest) seperti Home, About Us, Events dan Blog sesuai dengan desain desain.	Senin, 14 Juni 2025	4 jam X 3 Hari	Halaman-halaman utama untuk pengunjung (guest) selesai diimplementasikan secara statis sesuai desain dari tim UI/UX.	Memastikan desain responsive di berbagai perangkat memerlukan penyesuaian CSS yang detail.		
5	Frontend Developer 2: Mengembangkan implementasi desain UI untuk halaman dashboard alumni dan dashboard Alumni, termasuk struktur menu, komponen statistik, dan layout utama.	Senin, 14 Juni 2025	4 jam X 3 Hari	Layout dasar dan komponen utama untuk dashboard Admin dan Alumni berhasil dibuat dan siap dihubungkan dengan data dari backend.	Perbedaan tata letak antara dashboard Admin dan Alumni memerlukan penyesuaian komponen tampilan.		
6	Frontend Developer 1 & Backend Developer 2: Menghubungkan data dan logika ke tampilan frontend untuk fitur-fitur utama seperti menampilkan daftar pengguna di dashboard admin dan menampilkan profil alumni.	Jumat, 20 Juni 2025	5 jam	Fitur-fitur utama di sisi Admin dan Alumni sudah selesai dan terhubung dengan database data berhasil ditampilkan sesuai dengan fungsinya.	Terdapat beberapa logika yang perlu dioptimalkan agar query tidak efisien saat digunakan di frontend.		
7	Proyek Manajer Melakukan pertemuan tim (koordinasi via Discord/office) untuk meninjau progres, meninjau hasil kerja dan mengidentifikasi kendala teknis dan penyesuaian proyek.	Jumat, 27 Juni 2025	2 jam	Identifikasi beberapa kag mitra pada integrasi Frontend-Backend dan pembagian tugas untuk perbaikan. Program tim terintegrasi sesuai jadwal baru.	Sinkronisasi antara data yang dikirim Backend dan yang diharapkan Frontend memerlukan koordinasi yang lebih intensif.		

1	Project Manager: Melakukan koordinasi dengan tim untuk finalisasi proposal, penyusunan laporan akhir MPT, dan menyusun materi presentasi untuk mitra.	23 Juli 2025	4 jam	Draft laporan akhir selesai dibikin, dan outline presentasi untuk mitra telah disiapkan oleh seluruh anggota tim.	Mengumpulkan semua dokumentasi dari setiap anggota tim dalam format yang seragam dan tepat waktu.		
2	Project Manager: Beribingin dengan Dosen Pembimbing Melakukan penyusunan hasil akhir proyek kepada perwakilan mitra (bukan dari Bapak Mahrobi).	23 Juli 2025	1 jam	Proyek diartikan dengan baik oleh mitra. Mendapatkan masukan untuk pengembangan di masa depan. Laporan diartikan oleh pembimbing.			

Dosen Pengampu Kelas MPT

Guntur Maulana Zamroni, S.Sc., M.Kom.
NIP. 60181172

Yogyakarta, 25 Juli 2025

Ketua MPT


(Novi Lis Ramadani)

2. Hasan Nur Rasyid (2200018068) – Sekretaris – BE1
 1) Sebelum UTS

**LOG BOOK INDIVIDU MANAJEMEN PROYEK TEKNOLOGI INFORMASI MAHASISWA
 PROGRAM STUDI XI INFORMATIKA UNIVERSITAS AHMAD DAHLAN TA 2024/2025**

Nama Kelompok MPPI : Jevifly			
No	NIM	NAMA	
1	2200018068	Hasan Nur Rasyid	
Judul Manajemen Tugas Proyek : Sistem Informasi Sekolah Berbasis Web			
Dosen Pembimbing : Dr. Murnita, S.Si., M.Kom.			

1. Log book dibuat per minggu
 2. Log book dibuat setiap
 3. Setiap kegiatan di paraf oleh Mentor atau Dosen / dosen pembimbing MPPI
 4. Log book per minggu di paraf oleh mentor / pembimbing setiap MPPI
 5. Lembar logbook diserahkan ke sekretaris UTS

Logbook Minggu 1 s.d 7 (sebelum UTS)

No.	Kegiatan	Waktu Pelaksanaan		Hasil	Berkas, Rencana Perubahan (jika ada)	Paraf MITRA/CLIENT	Paraf Dosen Pembimbing MPPI
		Hari/Tgl	Durasi (jam)				
1	Backend Developer 1 (Dokumen awal dengan Backend Dev 1 terkait struktur folder backend dan penyesuaian email berdasarkan analisis kebutuhan)	Jakarta, 28 April 2025	2 jam	Disiapkan pemblokir tanggal awal dan struktur direktori Laravel	Perbaikan preferensi struktur folder dan penamaan		
2	Backend Developer 1 (Implementasi model dan migration bersama backend dev 2, implementasi fitur login/logout admin menggunakan Laravel Auth dan middleware)	Jakarta, 18 Maret 2025	4 jam	Model dan migrasi berhasil dibuat dan diujicoba	Perlu peningkatan validasi		

3	Backend Developer 1 (Implementasi controller MPPI mengatur logic bisnis dan pengendalian data, mendefinisikan route untuk menghubungkan url dengan controller)	Jakarta, 1 Mei 2025	6 jam	Controller dasar berhasil dibuat dan diujicoba	Perlu peningkatan preferensi input user dengan backend dev 2		
4	Backend Developer 1 (penyajian data untuk endpoint backend untuk memastikan data dapat diambil dan disimpan dengan benar, memastikan fitur berjalan sesuai kebutuhan)	Jakarta, 20 Mei 2025	4 jam	Perbaikan terhadap endpoint CRUD berdasarkan pengujian dan implementasi awal dengan frontend			

Dosen Pengampu Kelas MPPI

Berning Robita, S.T., M.T.
 NIM : 0220077803

Kejuruteraan, 16 Mei 2025

Kejuruteraan 1


 Hasan Nur Rasyid
 NIM : 2200018068

2) Setelah UTS

**LOG BOOK INDIVIDU MANAJEMEN PROYEK TEKNOLOGI INFORMASI MAHASISWA
PROGRAM STUDI SI INFORMATIKA UNIVERSITAS AHMAD DAHLAN T.A 2024/2025**

Nama Kelompok MPTI : Destify		Nama	
No	NPM		
1	22100018068	Rafael Nur Rizki	

Judul Manajemen Tugas Proyek : Sistem Akademik Alumni Bekal Berbasis Web - Alumni Garuda Pembela

Dosen Pembimbing : Dr. Muzita, S.Si, M.Kom

Logbook Minggu 8 sd 14 (setelah UAS + Pengisian Matri)

No	Kegiatan	Waktu Pelaksanaan		Hasil	Kendala, Rencana Perubahan (jika ada)	Fase MTSA/CLIENT	Paraf Dosen Pembimbing MPTI
		Hari/Tgl	Durasi (Jam)				
1	Analisa Proyek & Perencanaan Arsitektur Backend: Diskusi awal tim, penentuan struktur proyek Laravel, rust & pemilihan library backend, setup environment, dan instalasi Git repository.	Selasa, 17 Mei 2025	6 jam	Struktur Arsitektur ditetapkan, rekomendasi sup. library dasar teridentifikasi, repo Git aktif.	Peta pemecahan syarat dan struktur database proyek tema ke yang baru...		
2	Desain Database & Implementasi Model: Merancang skema database, implementasi model (User, Admin, Alumni), pembuatan migrasi, dan pendefinisian relasi antar tabel.	Jumat, 04 Juni 2025	6 jam	Skema database final untuk fase awal siap, model dan migrasi bertahap dibuat & dijalankan, relasi terdefinisi.	Struktur relasi antar tabel lebih kompleks dibandingkan proyek sebelumnya, wawancara dilakukan intensif dengan System Analyst.		

3	Pengembangan Modul Login & Otentikasi: Renc. login/logout Admin (Laravel Auth & middleware), pengembangan register Alumni & Guru, serta implementasi sistem IPAC/Role-Based Access Control.	Kamis, 05 Juni 2025	6 jam	Aktivitas Admin & register berhasil berfungsi, sistem RBAC (peran & permission) dasar terdefinisi.	Debugging dan pemrosesan preferensi input user dengan Laravel dev 2.		
4	Pengembangan Modul Pengaturan Data Pengguna & Admin: Pengembangan CRUD untuk manajemen data Admin dan manajemen data pendaftaran profil Alumni.	Kamis, 12 Juni 2025	6 jam	Perbaikan terhadap masalah CRUD berdasarkan pengujian dan implementasi awal dengan frontend.			
5	Pengembangan Modul Pencarian & Akurasi Informasi Publik/Kepeter: Pengembangan logika backend untuk pencarian & filter Alumni terapan, implementasi fungsionalitas atom & interaksi konten berbasis artikel, serta fungsionalitas akurasi informasi publik terapan.	Minggu, 15 Juni 2025	6 jam	Fungsionalitas pencarian & filter Alumni bekerja dengan baik, konten berita/artikel dapat diinput & diterbitkan, referensi publik tambahan berhasil.			
6	Pengembangan Modul Event & Partisipasi Alumni: Implementasi fungsionalitas manajemen event komprehensif oleh Admin, pengembangan fitur registrasi & partisipasi event (web/offline).	Kamis, 20 Juni 2025	6 jam	Admin dapat mengelola event, Alumni dapat mendaftar partisipasi, event publik dapat dilihat.			

Alumni, serta fungsionalitas penampildetail & detail event publik.						
1. Pengembangan Modul Konten & Moderasi Konten: Pengembangan fungsionalitas kontribusi konten (artikel, pengalaman, portfolio) oleh Alumni, dan implementasi sistem moderasi konten & komunikasi oleh Admin.	Senin, 25 Juni 2025	5 jam	Admin dapat mengunggah konten, Admin dapat menyetujui/menolak/menghapus konten & komentar.			
2. Pengujian, Debugging & Optimasi Kinerja: Melakukan pengujian unit/integrasi seluruh ruta web Backend, debugging & perbaikan bug, optimasi kinerja database & logika Backend, serta koordinasi integrasi dengan tim Frontend untuk integrasi.	Sabtu, 28 Juni 2025	6 jam	Sebagian besar fungsionalitas Backend telah diuji, bug kritis berhasil, kinerja awal memadai, integrasi Frontend-Backend berjalan lancar.			

Direktur Pengajaran & Kesiswaan

Bambang Roban, S.T., M.T.
NIDN : 002077901

Yogyakarta, 20 Juni 2025

Backend 1



Hasm Nur Bayat
NIM : 1220026088

3. Andriansyah (2200018088) – System Analyst
 1) Sebelum UTS

105 8601 INDIVIDU MANAJEMEN PROYEK TEKNOLOGI INFORMASI MAHASISWA
 PROGRAM STUDI INFORMATIKA UNIVERSITAS ANHARD DANJANT 4 2024/2025

Nama Kelompok (MFT) : Individu		
No	NIM	NAMA
1	2200018088	Andriansyah
Tipe/Manajemen Tugas Proyek : Sistem Informatika (kecil/kecil/kecil/kecil)		
Dosen Pembimbing : Dr. Marissa, S.Si, M.Kom		

- Skema Diagram (UML)
1. UML Use Case Diagram
 2. UML Activity Diagram
 3. UML Sequence Diagram
 4. UML Class Diagram
 5. UML Package Diagram
 6. UML State Machine Diagram
 7. UML Component Diagram
 8. UML Deployment Diagram

Aspek Minggu 1 s.d 7 (setelah uji)

No. Kegiatan	Waktu Pelaksanaan		Basil	Sudala, Rencana Perubahan (Jika ada)	Pafid MPRADJENT	Pafid Dosen Pembimbing MPR
	Mulai (DD)	Selesai (DD)				
1. Sistem Analisis/Identifikasi kebutuhan sistem dan memvisualisasikan awal	Jan, 17 Maret 2025	2 Jan	Bahan dan kebutuhan awal UML dan diagram	Perencanaan kebutuhan awal belum terlaksana dengan jelas		
2. Sistem Analisis/Identifikasi kebutuhan sistem dengan stakeholder untuk memvisualisasikan awal	Jan, 29 Maret 2025	2 Jan	Caraker memvisualisasikan dan berdiskusi proses bisnis utama			
3. Sistem Analisis/Identifikasi UML untuk proses bisnis utama sistem informasi (sistem)	Jan, 18 Maret 2025	4 Jan	UML untuk proses bisnis dan memvisualisasikan	Perlu pemrosesan pada beberapa hal untuk pada proses bisnis		

4. Sistem Analisis/Identifikasi dengan UML dan memvisualisasikan awal	Jan, 5 April 2025	4 Jan	Proses UML berdasarkan masalah dan diagram			
5. Sistem Analisis/Identifikasi UML dan memvisualisasikan awal	Jan, 14 April 2025	4 Jan	Perencanaan UML dengan beberapa sistem perubahan	Perencanaan proses awal dan kebutuhan sistem permasalahan awal		
6. Sistem Analisis/Identifikasi UML dan memvisualisasikan awal	Jan, 18 April 2025	4 Jan	UML dan diagram untuk sistem informasi sekolah			
7. Sistem Analisis/Identifikasi UML dan memvisualisasikan awal	Jan, 21 April 2025	4 Jan	UML dan diagram untuk sistem informasi sekolah			
8. Sistem Analisis/Identifikasi UML dan memvisualisasikan awal	Jan, 28 April 2025	4 Jan	UML dan diagram untuk sistem informasi sekolah	Perlu pemrosesan pada beberapa hal untuk pada proses bisnis		
9. Sistem Analisis/Identifikasi UML dan memvisualisasikan awal	Jan, 28 April 2025	4 Jan	UML dan diagram untuk sistem informasi sekolah	Perlu pemrosesan pada beberapa hal untuk pada proses bisnis		
10. Sistem Analisis/Identifikasi UML dan memvisualisasikan awal	Jan, 28 April 2025	4 Jan	UML dan diagram untuk sistem informasi sekolah	Perlu pemrosesan pada beberapa hal untuk pada proses bisnis		

2) Setelah UTS

LOG BOOK INDIVIDU MANAJEMEN PROYEK TEKNOLOGI INFORMASI MAHASISWA PROGRAM STUDI SI INFORMATIKA UNIVERSITAS AHMAD DAHLAN T.A 2024/2025

Nama Kelompok: MP11 - Zestdy		
No	NIM	NAMA
1	20780120105	Andrihorsoh
Judul Manajemen Tugas Prasyak : Sistem Informasi Alumni Cabang Pematang		
Dosen Pembimbing : Dr. Nurinta, S.Si, M.Kom.		

- Format Pengisian Log Book
- Log book dibuat per minggu
 - Log book ditulis tangan
 - Setiap kegiatan di tulis oleh Mahasiswa/Chief / dan di tanda tangi MP1
 - Log book per minggu di persil oleh dosen pengampu kelas MP1
 - Jumlah lembar minimal 10 (sepuluh) 10

Logbook Minggu 2 sd 13 (setelah UTS)

No	Kegiatan	Waktu Pelaksanaan		Hasil	Kerangka, Rencana Perubahan (jika ada)	Faul NITRA/CUBIT	Paraf Dosen Pembimbing MP1
		Hari/Tgl	Durasi (jam)				
1	System Analyst(Mengumpulkan kebutuhan sistem dan membuat diagram awal)	Senin, 27/05/2025	2 jam	Dokumen kebutuhan awal FR dan FRK dan diagram	Beberapa kebutuhan masih belum terdefinisi dengan jelas		
2	System Analyst(Melakukan wawancara dengan stakeholder untuk menggali kebutuhan proses bisnis)	Selasa, 28/05/2025	2 jam	Catatan wawancara dan identifikasi proses bisnis utama			
3	System Analyst(Membuat BPMN untuk proses bisnis utama sistem informasi sekolah)	Kamis, 28/05/2025	4 jam	BPMN untuk proses bisnis dan sistem informasi sekolah	Perlu penyesuaian pada beberapa sub proses pada proses bisnis		

4	System Analyst(Membuat use case diagram sistem)	Senin, 29/05/2025	3 jam	Use Case diagram untuk sistem informasi sekolah			
5	System Analyst(Membuat Activity diagram untuk use case utama)	Jum'at, 30/05/2025	4 jam	Diagram activity diagram sesuai dengan use case diagram untuk sistem informasi sekolah			
6	System Analyst(Membuat class diagram awal)	Minggu, 1/06/2025	4 jam	Class diagram versi 1.0 dengan entitas dan relasi utama	Perluasan dalam penambahan entitas dan metode yang lengkap		
7	System Analyst(Dokumen dengan Backend developer untuk penyelesaian class diagram)	Senin, 2/06/2025	2 jam	Review class diagram berdasarkan masukan dari tim backend			
8	System Analyst(Finalisasi class diagram untuk implementasi)	Selasa, 3/06/2025	3 jam	Class diagram final dengan entitas, metode dan relasi lengkap	Wawancara konfirmatif dengan dosen pembimbing mengenai persil di kelas		

Pematang, 13 Mei 2025

Dosen Pengampu Kelas MP1

Bombeng Rob'in, S.T., MT
NIP: 20780120105041002

(Andrihorsoh)

4. Farqad Assanji (2200018092) – UI/UX – FE2
 1) Sebelum UTS

LOG BOOK PEMBIKUN MARIJENSI PROJEK TEKNOLOGI INFORMASI MMHASISMA
 PROGRAM STUDI SI INFORMATIKA UNIVERSITAS AHMAD DAHLAN T.A. 2024/2025

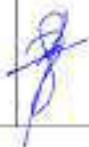
Nama Kelompok: MPTI - Deshifa		
No	NIM	nama
1	2200018092	Farqad Assanji
Judul Manajemen Tugas Proyek : Sistem Informasi Sekolah Berbasis Web		
Dosen Pembimbing : Dr. Murtosa, S.S., M. Kom.		

- Daftar Isi: (Tulis saja jika ada)
1. Log book di akhir minggu
 2. Log book di akhir tugas
 3. Setor laporan di awal dan di akhir (akhir / dosen pembimbing MPTI)
 4. Log book per minggu di pastikan dalam pengumpulan MPTI
 5. Untuk pengiriman email ke Laheba @UG

Logbook Minggu 1 s.d 7 (selain UTS)

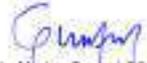
No	Kegiatan	Waktu Pelaksanaan		Hasil	Kendala, Permasalahan (jika ada)	Paraf MPTI/CLIENT	Paraf Dosen Pembimbing MPTI
		Hari/Tgl	Durasi (jam)				
1	UI/UX Designer (Melakukan riset awal dan bentuk forming desain awal untuk informasi sekolah)	Senin, 17 Maret 2025	2 jam	Referensi desain dan draft awal website UI			
2	UI/UX (Menganalisa dan mengidentifikasi desain dan informasi website sekolah lain)	Rabu, 19 Maret 2025	2 jam	mengidentifikasi dan layout desain dalam data dokumen figure	Kendala: menemukan gambar visual yang sesuai dengan layout pengguna		
3	UI/UX Designer (Membuat wireframe halaman utama, profil sekolah, berita, dan galeri)	Kamis, 27 Maret 2025	2 jam	Membuat wireframe website yang sudah disediakan layout	Dibutuhkan gambar dan visual untuk website		

4	UI/UX Designer (Proses final website dan main menu website dan email)	Senin, 24 April 2025	2 jam	Wireframe dan visualisasi kebaruan website	Musik dan email final untuk email, berita, kontak langsung.		
5	UI/UX Designer (Pembuatan Mockup High Fidelity dan panel fitur utama utama serta navigasi)	Jumat, 18 April 2025	2 jam	Desain visual lengkap untuk halaman dashboard dan profil sekolah			
6	UI/UX Designer (Membuat user flow dan gambar untuk tampilan website sekolah)	Rabu, 24 April 2025	2 jam	Menyusun alur dan visualisasi kepada tim pengembang	Kendala: pengantar perlu diskusi ukuran dan formatnya.		
7	UI/UX Designer & Frontend (Diskusi awal antara tim UI/UX dan frontend untuk penyediaan layout di frontend)	Rabu, 2 Mei 2025	2 jam	Layout dan visualisasi agar responsive dan objek awal diimplementasi			
8	Frontend Developer (Membantu frontend untuk mengimplementasikan UI ke dalam Block Template dan hasil UI)	Juma, 1 Mei 2025	4 jam	Hasil dari dashboard admin dan kerangka website sekolah di awal			
9	Frontend Developer (Membantu coding tampilan dan memperbaiki tampilan review ke file frontend)	Senin, 10 Mei 2025	2 jam	Checklist tampilan UI sudah sesuai dengan desain	Masih ada beberapa bagian yang belum selesai (form, berita, profil, galeri, kontak)		

10.	UI/UX Designer (Kategori dengan frontend) terkait penyusunan layout di halaman berita	Desin, 12 Mei 2025	1 jen	layout halaman berita disesuaikan agar lebih rapi dan responsif			
-----	---	--------------------	-------	---	--	--	---

Yogyakarta, 19 Mei 2025

Doori Persempa Gola MPH


Guntur Muliawati Zamroni, B.Sc., M.Com.
NIP. 90120172

(Guntur Muliawati)

2) Setelah UTS

LOG BOKOR INDIVIDU/MANAJEMEN PROYEK TEKNOLOGI INFORMASI MAHASISWA PROGRAM STUDI S1 INFORMATIKA UNIVERSITAS AHMAD DAHLAN T.A 2024/2025

Nama Kelompok NPT: (2022)		
No	NIM	NAMA
1	2200018092	Fariqad Asyraf
Judul Manajemen Tugas Proyek: Sistem Informasi Alumni Gantus Peralang		
Dosen Pembimbing: Dr. Nurinta S.S., M.Kom		

Format Penulisan Logbook

1. Logbook dibuat per minggu
2. Logbook ditulis tangan
3. Setiap kegiatan di paraf oleh Mahasiswa/Dosen / Dosen pembimbing/NPT
4. Logbook per minggu di paraf oleh dosen pengampu mata kuliah
5. Jumlah kegiatan minimal 40 kegiatan/UTS

Logbook Minggu 9 s.d 11 (Setelah UTS)

No	Kegiatan	Waktu Pelaksanaan		Hasil	Kerangka, Revisasi, Perubahan (Jika ada)	Paraf NPT/A/CLIENT	Paraf Dosen Pembimbing NPT
		Hari/Tgl	Durasi (Jam)				
1	Ditandai awal bersama tim terkait pelaksanaan kegiatan sistem baru. Mulai menyusun wireframe halaman utama, login, dan register.	Selasa, 27 Mei 2025	4 jam	Salah wireframe dasar halaman utama dan login/register sebagai Figma.	Alasan ke desain lama tidak relevan, mulai dari awal.		
2	Melengkapi wireframe untuk halaman dashboard alumni & admin. Menyusun struktur navigasi pengguna.	Rabu, 28 Mei 2025	4 jam	Struktur navigasi lengkap, siap uji pengguna terhadap sistem dengan jelas.	Tantangan dalam membedakan peran user alumni dan admin.		
3	Mendesain high fidelity UI untuk struktur halaman berdasarkan wireframe sebelumnya.	Setra, 31 Mei 2025	3 jam	Desain Hi-Fidelity 5 halaman utama selesai.	Warna dan logo disesuaikan dengan branding Gantus.		

4	Implementasi halaman landing page dan layanan ke dalam blog kampus terbaru.	Jumat, 06 Juni 2025	1 jam	Salah satu publikasi selesai & ready dengan responsif. Template selesai desain.			
5	Implementasi UI dashboard alumni & admin struktur sidebar, header, dan footer utama.	Sabtu, 14 Juni 2025	1 jam	Dashboard admin dan alumni mulai selesai desain dan layout selesai.			
6	Mendesain dan mengimplementasikan halaman login & register dengan validasi form.	Senin, 18 Juni 2025	3 jam	Templat login/register selesai, disertai validasi dan alert feedback.			
7	Revisi dan perbaikan desain frontend berdasarkan hasil uji coba dan masalah design.	Jumat, 20 Juni 2025	1 jam	Templat frontend lebih stabil dan sudah diupdate. Desain dipulihkan.	Revisi beberapa icon dan padding antar komponen.		
8	Penyusunan templat sistem dokumentasi style guide dan struktur folder desain.	Sabtu, 28 Juni 2025	2 jam	UI/UX guide selesai. Semua desain Figma document ready untuk dokumentasi sistem.	Format dokumentasi disesuaikan dengan standar universitas.		
9	Menyusun file presentasi UAT & kontribusi sebagai Poinmal 2 ke dalam laporan akhir UTS.	Setra, 19 Juli 2025	6 jam	File laporan presentasi dan dokumen ke-luaran tim. Disesuaikan dengan struktur laporan.	Presentasi templat dilakukan oleh mahasiswa pembimbing.		

5. Destyawan Satria Nugraha (2200018095) – Tester/QA
 1) Sebelum UTS

LUG BOOK INDIVIDU MANAJEMEN PROYEK TEKNOLOGI INFORMASI MAHASISWA PROGRAM
 STUDI SI INFORMATIKA UNIVERSITAS AHMAD DAHLAN T.A 2024/2025

Nama Kelompok MPTI : Zestyly		
No	NIM	NAMA
1	2200018095	Destyawan Satria Nugraha
Judul Manajemen Tugas Proyek : Sistem Informasi Sekolah Berbasis Web		
Dosen Pembimbing : Dr. Marina, S.Si., M.Kom.		

- Daftar Isi Laporan Log Book**
1. Log book dibuat per minggu
 1. Log book ditulis tangan
 1. Setiap kegiatan di jurnal with foto atau video / atau peribahasa MPTI
 1. Log book per minggu di post oleh dosen pengampu kelas MPTI
 1. Jurnal berbentuk minimal 4. sebelum UTS

Logbook Minggu 3 s.d 7 (sebelum UTS)

No	Kegiatan	Waktu Pelaksanaan		Hasil	Sensasi, Rencana Perubahan (jika ada)	Paraf MPTI/CLIENT	Paraf Dosen Pembimbing MPTI
		Hari/Tgl	Durasi (jam)				
1	QA Tester (tugas awal testing, review fitur awal, load test awal)	Senin, 04 Maret 2025	4 jam	Dilakukan test awal dan load test	Fitur yang login dibuat masih belum jalan		
2	QA Tester (Manual testing fitur login, register)	Senin, 24 April 2025	3 jam	Fitur login dan register sudah siap dengan beberapa masalah	login tidak berjalan karena tidak ada data user di database jadi harus register dan register masih ada bug tidak bisa merespon data user ke database dan harus manual		
3	QA Tester (Regresion testing dan responsive layout pada fitur beranda)	Sabtu, 2 Mei 2025	6 jam	Fitur sebelumnya sudah stabil, UI tampil cukup baik di perangkat berbeda	Ada delay yang disebabkan oleh konflik CSS		

4	QA Tester (Testing Regresion dan responsive layout fitur seluler/nya)	Sabtu, 10 Mei 2025	5 jam	Dashboard admin dan profil di hantui masih terhubung dengan baik ke sistem	Halaman mobile kurang responsif dan layout tidak bertautan		
5	QA Tester (Regresion testing dan responsive layout pada fitur Data berita, galeri, partner)	Sabtu, 17 Mei 2025	5 jam	Fitur Data berita, galeri, dan menu berhasil tetapi masih ada masalah di backend	Tabel database konflik saat melakukan migrasi, karena ada duplikasi tabel		

Dosen Pengampu Kelas MPTI

Negeriarts, 16 Mei 2025

QA Tester

Sambang Pribi's, S.T., N.U.I
 HP : 157907202085013002

(Destyawan Satria Nugraha)

2) Setelah UTS

LOG BOOK INDIVIDU MANAJEMEN PROYEK TEKNOLOGI INFORMASI MAHASISWA PROGRAM STUDI SI INFORMATIKA UNIVERSITAS AHMAD DAHLAN T.A 2024/2025

Nama Kelompok (MPT) : Zestify		
No.	NIM	NAMA
1.	2100018095	Devityawan Setria Nugraha
Judul Manajemen Tugas Proyek : Sistem Informasi Alumni Genzisa Pemandang		
Dosen Pembimbing : Dr. Murnito, S.Si, M.Kom.		

Format Penulisan Log Book

1. Log book dibuat per minggu
2. Log book ditulis tangan
3. Setiap kegiatan di paraf oleh mahasiswa/ dosen / atau pembimbing/ MPT
4. Log book per minggu di paraf oleh dosen pembimbing/ MPT
5. Jumlah lembar minimal 4x sebelum UTS

Logbook Minggu 1 s.d 7 (sebelum UAS)

No	Kejelasan	Waktu Pelaksanaan		Hasil	Kendala, Rencana Perubahan (jika ada)	Paraf MPT/ALUMNI	Paraf Dosen Pembimbing MPT
		Hari/TOL	Durasi (jam)				
1.	QA Tester (Sebelum awal testing, review fitur awal, buat test case)	Sabtu, 14 Juni 2025	8 jam	Document test case dan fitur	Fitur yang ingin dibuat masih belum jelas		
2.	Pelaksanaan pengujian awal terhadap fitur-fitur admin dan alumni, termasuk halaman "About", pengumuman, blog serta kategori blog. Pengujian dilakukan berdasarkan perencanaan berdasarkan perencanaan uji fungsional	Selasa, 27 Juni 2025	3 jam	Sebanyak 10 test pertama (T001-T010) berhasil dijalankan dan seluruhnya lulus	Tidak ditemukan kendala		
3.	Pengujian lanjutan dilakukan terhadap fitur-fitur event, kategori blog, dan manajemen jurusan oleh admin. QA memastikan	Rabu, 28 Juni 2025	3 jam	20 test selanjutnya (T020-T030) berhasil dijalankan dan seluruhnya lulus	Tidak ditemukan kendala		

4.	QA menguji fitur alumni termasuk blog dan direktori, serta mencoba akses sebagai tamu untuk memastikan keamanan	Kamis, 29 Juni 2025	3 jam	Terjadi error pada fitur blog Alumni (T040-T050) saat tampilan blog tidak muncul. Test lainnya berhasil dijalankan	Kendala: Ralat antara data blog dan kategori bermasalah. Rencana: Bug diperbaiki oleh tim developer esok hari		
5.	Tim developer memperbaiki error pada fitur Blog Alumni. QA menguji ulang fitur tersebut dan melaporkan pengujian fitur direktori dan dashboard alumni	Jumat, 30 Juni 2025	2 jam	Seluruh fitur blog alumni berhasil diuji ulang (T040-T050) dan dinyatakan lulus. Site test hingga T070 juga lulus	Perbaikan route untuk memastikan blog alumni merespon ke route yang tepat		
6.	Pengujian sistem autentikasi, termasuk login, logout, verifikasi email, reset dan konfirmasi password, dilakukan untuk semua jenis pengguna	Sabtu, 1 Juli 2025	4 jam	Semua test autentikasi pengguna (T070-T080) berhasil dijalankan dengan status lulus	Tidak ditemukan kendala		
7.	QA melakukan pengujian terhadap proses registrasi, validasi input, serta akses publik terhadap halaman blog, event, dan pengumuman. Melakukan Test Ulang kesemua fitur	Senin, 2 Juli 2025	12 jam	Seluruh test dari T080-T100 berhasil dijalankan dan lulus. Dan ketika dilakukan test ulang seluruh test (100 test) semua berjalan dengan baik tanpa adanya kendala	Semua fitur publik berjalan sesuai ekspektasi		

E. Foto dokumentasi kegiatan proyek

Sebelum Pergantian Mitra

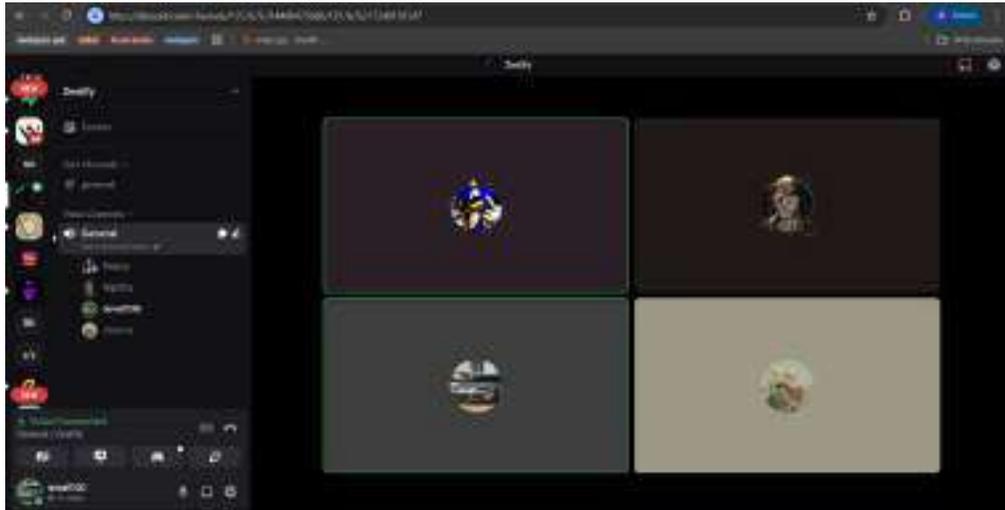
1. Pertemuan internal pertama tanggal 17 Maret 2025 (Bulan Ramadhan)



2. Pertemuan Internal kedua tanggal 08 April 2025



3. Pertemuan Internal ketiga tanggal 22 April 2025



4. Pertemuan Internal keempat tanggal 26 April 2025



5. Pertemuan Internal kelima tanggal 08 Mei 2025



Setelah Pergantian Mitra

6. Pertemuan Internal keenam tanggal 30 Mei 2025



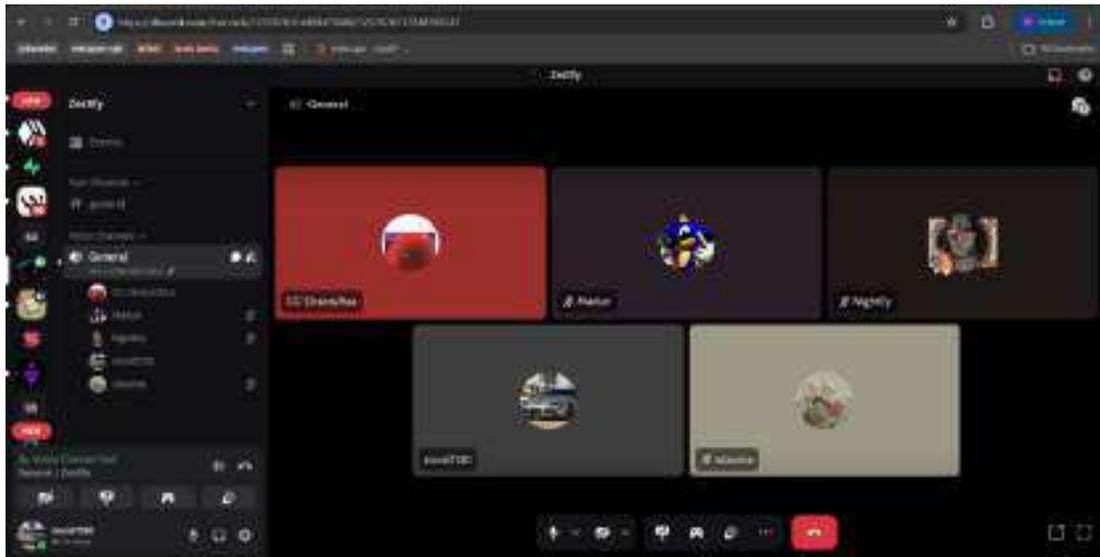
7. Pertemuan Internal ketujuh tanggal 14 Juni 2025



8. Pertemuan Internal Kedelapan tanggal 22 Juni 2025



9. Pertemuan Internal Kesembilan tanggal 2 Juli 2025



10. Pertemuan Internal Kesepuluh tanggal 11 Juli 2025



F. Berita acara / bukti serah terima proyek



ZESTIFY

Jalan bausasran DN3/557 A, Yogyakarta Tlp 6281268181997

Laman : @zestify2025

BERITA ACARA SERAH TERIMA (BAST)

Pada hari ini Rabu tanggal du apuluh tiga bulan juli tahun dua ribu dua puluh lima bertempat di Yogyakarta, yang bertanda tangan dibawah ini :

1. Nama : Noval Lias Ramadani
Instansi : Kelompok MPTI Zestify (Universitas Ahmad Dahlan)
Jabatan : Mahasiswa/Pengembang Sistem
Alamat : Jl. Bausasran, DN3 557/A
Selanjutnya disebut PIHAK PERTAMA.
2. Nama : Edi Sutriyono, S.E., M.Si.
Instansi : GANISSA Pemalang
Jabatan : Ketua Bidang Kominfo GANISSA
Alamat : Jalan A.Yani Pemalang
Selanjutnya disebut PIHAK KEDUA.

PIHAK PERTAMA dan PIHAK KEDUA, selanjutnya disebut PARA PIHAK, dengan ini menyatakan bahwa:

1. Berdasarkan hasil pengembangan sistem, PIHAK PERTAMA telah menyelesaikan pembuatan Sistem Informasi Alumni GANISSA Pemalang sesuai dengan kebutuhan dan kesepakatan awal.
2. Adapun dokumen dan item yang diserahkan oleh PIHAK PERTAMA kepada PIHAK KEDUA adalah sebagai berikut :

No	Nama Dokumen/Item	Keterangan
1	Source Code Sistem	Berbasis Web, disimpan di github atau dalam bentuk file zip yang di upload ke drive
2	Hak Akses Admin & Database	Akun super admin dan database sistem
3	Akun Youtube Ganissa Pemalang	Email/username dan password akun youtube ganissa pemalang

3. Bahwa PIHAK PERTAMA dengan ini mnyerahkan dokumen kepada PIHAK KEDUA dan PIHAK KEDUA dengan ini menerima dokumen tersebut dari PIHAK PERTAMA.
4. Dengan telah dilakukannya serah terima tersebut, maka kewajiban PIHAK PERTAMA dianggap telah selesai, dan PIHAK KEDUA akan melanjutkan pengelolaan serta pemanfaatan sistem sesuai dengan kebutuhan organisasi.

Demikian Berita Acara ini dibuat dan ditandatangani oleh PARA PIHAK dalam keadaan sadar tanpa ada tekanan dari pihak manapun, untuk digunakan sebagaimana mestinya.

Yogyakarta, 23 Juli 2025

PARA PIHAK

PIHAK PERTAMA



Noval Lias Ramadani

PIHAK KEDUA



Edi Sutriyono, S.E., M.Si.

Catatan:

*) Dokumen/File yang diserahkan meliputi:

1. Source Code Sistem
2. Hak Akses Admin & Database
3. Akun Youtube Ganissa Pemaalang

G. Bukti pembiayaan : pemasukan, pengeluaran, pendapatan/keuntungan dan konsultan proyek, bukti pembayaran

1. Bukti Pembayaran Hosting dari Mitra

Uraian	Tipe	Nominal
TRANSFER KE BIMASAKTI MULTI SINERGI 9887888450957783 Winpay-rumahweb - RUMAHWEB INDONE	D	401.043,00
2025-06-22 12:32:23		
TRANSFER DARI MURINTO Pembayaran domain hosting gannisa 2025	K	405.000,00
2025-06-22 10:33:26		

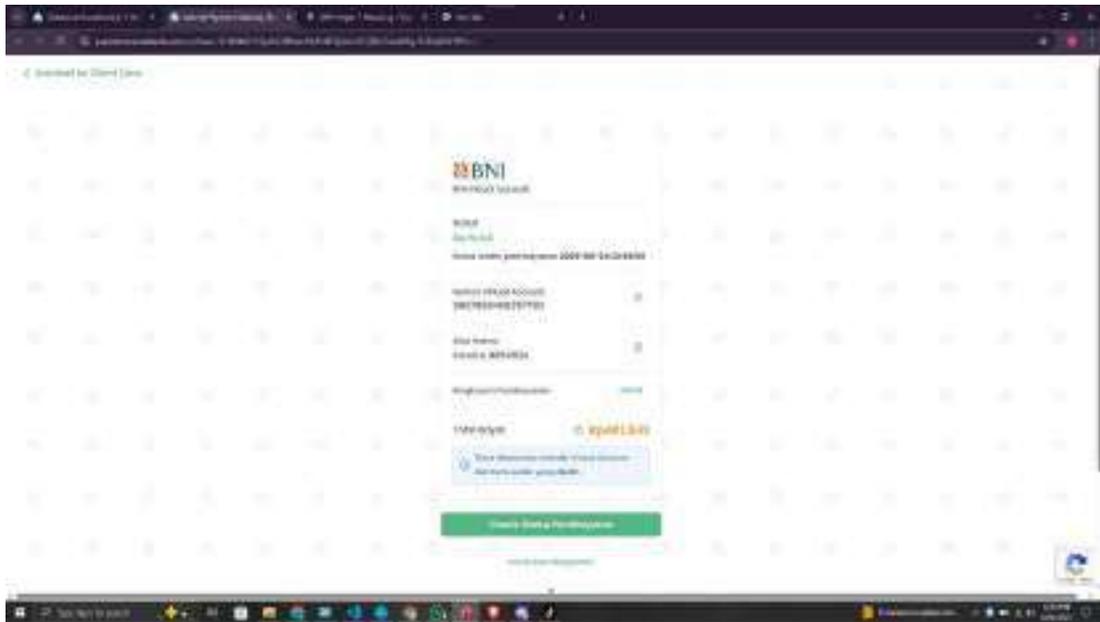
2. Bukti Pembelian Hosting ke layanan penyedia hosting



Transaksi Berhasil

Nomor Referensi	20250622123210993538
Nomor Jurnal	907763
Tanggal Transaksi	22-06-2025
Waktu Transaksi	12:32:23 WIB
Jenis Transaksi	Virtual Account Billing
No.VA	9887888450957783
Nama	Winpay-rumahweb - RUMAHWEB INDONESIA - invoice 4864924

Total Tagihan	Rp 401.043
Biaya admin	Rp0
Total Bayar	Rp401.043,00
Rekening Debet	*****305



H. Tools : source code, hosting, pemegang password, user manual

Untuk Source Code : <https://drive.google.com/drive/folders/1598liBg-i3WRuNUIJXlvY1y4ldmb7OEz?usp=sharing>

Layanan Hosting : Rumahweb

- I. Link video profil produk luaran proyek (video teaser yang berisi informasi proyek, durasi 4-7 menit)

Link Video Youtube : <https://youtu.be/W8zEZPROnEg>

Drive :

<https://drive.google.com/drive/folders/1UaqrpHDSx1NuBIYePM8DvtEU4Oesj6q4?usp=sharing>



J. Poster produk luaran

**TERHUBUNG SUK
TUMBUH SES**

Bersama di satu tempat, untuk masa depan yang lebih hebat!

Temukan Teman **Info Karier**

Lihat Kisah Sukses **Event & Reuni**

<https://ganissapemalang.or.id/>

K. Slide presentasi proyek

Link PPT : https://docs.google.com/presentation/d/1oDDE1fLQEvzChrfWHyEVH-3FiwEFx_iv/edit?usp=sharing&oid=101747819012149820709&rtpof=true&sd=true

1. Informasi umum

MANAJEMEN PROYEK TEKNOLOGI INFORMASI

SISTEM INFORMASI ALUMNI GANISSA



Ketua MPTI : Noval Lias Ramadani - 2200018083 - Project Manager - Backend2 - Frontend1

Tim MPTI :

- Hasan Nur Rasyid - 2200018068 - BachEnd1 - Sekretaris
- Andriansyah - 2200018088 - System Analyst
- Farqad Assanji - 2200018092 - UI/UX Designer - Frontend2
- Destyawan Satra Nugraha - 2200018095 - QA/Tester

Link project : <https://ganissapemalang.or.id>

PEMBIMBING TIM ◆◆◆ **PERKENALAN TIM**



Dr. Muripto, S.Si., M.Kom.
NIDN : 0510077302

- 01 2200018083 - Noval Lias ramadani - PM - BE2 - FE1
- 02 2200018068 - Hasan Nur Rasyid - Sekretaris - BE1
- 03 2200018088 - Andriansyah - System Analyst
- 04 2200018092 - Farqad Assanji - UI/UX - FE2
- 05 2200018095 - Destyawan Satra Nugraha - QA/Tester

PENGENALAN MITRA

Proyek ini bermitra dengan Alumni Ganissa Pemalang, komunitas alumni SMAN 1 Pemalang yang menghadapi tantangan dalam menjaga konektivitas dan mengelola data anggotanya. Sebelumnya, data alumni yang tersebar dan tidak akurat serta komunikasi yang masih konvensional menyebabkan melonjaknya jaringan dan menghambat partisipasi dalam berbagai kegiatan. Oleh karena itu, tujuan utama dari kolaborasi ini adalah untuk membangun sebuah "rumah digital" melalui Sistem Informasi GANISSA. Platform ini dirancang untuk menjadi pusat data alumni yang akurat, saluran komunikasi resmi, serta sarana untuk mendorong keterlibatan aktif, yang pada akhirnya bertujuan memperkuat kembali solidaritas dan rasa kekeluargaan di antara seluruh alumni SMAN 1 Pemalang.



LATAR BELAKANG PROYEK

-  **TANTANGAN** : Komunikasi dan pendataan alumni secara konvensional tidak efisien dan datanya tersebar.
-  **KEBUTUHAN** : Sekolah perlu beradaptasi dengan era digital untuk memberikan layanan informasi yang cepat, akurat, dan transparan.
-  **SOLUSI** : Membangun Ganissa, sebuah platform digital terpusat dan interaktif yang menjadi "rumah digital" bagi semua alumni.

ARSITEKTUR FUNGSIONAL GANISSA

Manajemen Profil & Keanggotaan

Fondasi utama sistem. Alumni dapat mendaftar, login, dan memperbarui profil mereka secara mandiri untuk menjaga akurasi data.

Publikasi & Informasi

Dikelola penuh oleh Admin sebagai corong informasi resmi untuk mempublikasikan artikel, Blog, Acara (Events), dan Pengumuman penting.

Interaksi & Keterlibatan

Menjadikan Ganissa platform yang hidup. Alumni dapat mendaftar acara dan berpartisipasi dalam diskusi melalui kolom komentar yang sehat dan konstruktif.

FOKUS PENGGUNA : MELAYANI TIGA AKTOR UTAMA

Pengelola Alumni (Admin)

Mebutuhkan alat yang efektif untuk menjangkau alumni, menyebar informasi, dan mengelola komunitas secara terstruktur.

Alumni SMAN 1 Permalang

Mebutuhkan platform untuk tetap terhubung, mendapatkan informasi valid, berpartisipasi, dan membangun jaringan profesional.

Masyarakat Umum (Guest)

Pihak eksternal yang ingin mengetahui prestasi dan aktivitas komunitas alumni sebagai citra positif sekolah.

FONDASI TEKNIS: ARSITEKTUR & TEKNOLOGI

Arsitektur Sistem

- Aplikasi Berbasis Web: Responsif dan dapat diakses di semua perangkat.
- Database Terpusat: Menggunakan MySQL untuk menyimpan semua data secara aman.
- Antarmuka Intuitif: Desain UI/UX yang fokus pada kemudahan penggunaan.

Tumpukan Teknologi (Tech Stack):

- Backend: PHP & Laravel Framework.
- Frontend: CSS & Tailwind CSS.
- Database: MySQL.
- Tools: VS Code, Figma, Github, Laragon.

METODE PENGEMBANGAN

◆◆◆

Dalam pengembangan Sistem Ganissa, kami menerapkan metodologi Agile, sebuah pendekatan pengembangan yang bersifat iteratif dan fleksibel. Metode ini memecah proyek besar menjadi siklus-siklus kerja yang lebih kecil—mencakup tahap perencanaan, desain, pengembangan, pengujian, dan peninjauan—yang diulang secara berkala.

Pemilihan metode ini terbukti sangat tepat karena kemampuannya untuk beradaptasi dengan perubahan, yang menjadi kunci saat tim kami harus mengganti proyek di tengah jalan akibat kendala mitra. Selain itu, Agile mendorong kolaborasi yang erat antar anggota tim, sangat mendukung kerja sama antara System Analyst, Backend Developer, dan Frontend Developer kami untuk memberikan umpan balik berkelanjutan dan mengatasi tantangan secara efisien hingga proyek berhasil diselesaikan.

AGILE



2. PM – Noval Lias Ramadani

PROJECT MANAGER - NOVAL LIAS RAMADANI

Jobdesk & Kunci Keberhasilan Project Manager

Tanggung Jawab Utama Sesuai Perencanaan Proyek :

- Bertanggung jawab atas keseluruhan perencanaan, pelaksanaan, pengawasan, dan penyelesaian proyek.
- Menjadi penghubung utama antara tim proyek dan stakeholder.
- Mengelola risiko proyek dan membuat keputusan strategis untuk kelancaran proyek.
- Memastikan seluruh tim bekerja sesuai linimasa dan tujuan proyek tercapai tepat waktu.

Poin Kunci Keberhasilan dalam Proyek Ini:

- Komunikasi Proaktif: Secara konsisten menjadi pusat komunikasi, baik internal dengan tim melalui Discord maupun eksternal dengan dosen pembimbing.
- Adaptasi & Keputusan Strategis: Kemampuan untuk cepat beradaptasi saat menghadapi kendala besar (mitra awal batal) dan mengambil keputusan strategis terbukti krusial untuk menyelamatkan proyek.
- Pengawasan Aktif: Tidak hanya merencanakan, tetapi juga aktif memantau, merevisi, dan memastikan kualitas hasil kerja setiap anggota tim.

REALISASI TUGAS PM - INISIASI & PERENCANAAN AWAL

Koordinasi Pemilihan Pembimbing:

- Memimpin koordinasi tim untuk mencari dan menentukan Dosen Pembimbing Proyek (MPTI) yang paling sesuai dengan keahlian yang dibutuhkan.
- Berhasil mengamankan kesediaan Bapak Dr. Murinto, S.Si., M.Kom sebagai pembimbing setelah diskusi dan pendekatan yang terencana



Definisi Proyek & Pembagian Tugas:

- Memfasilitasi diskusi tim untuk menentukan ide proyek awal (Website SMA N 2 Singkep) dan mengidentifikasi mitra.
- Merancang dan membagikan jobdesk yang jelas untuk setiap anggota tim (System Analyst, UI/UX, Backend, Frontend, QA) guna memastikan alur kerja yang terstruktur sejak awal.

REALISASI TUGAS PM - MANAJEMEN KRISIS & PIVOT PROYEK

- **Tantangan Utama : Kendala dari Mitra Awal :** Setelah progres pengembangan proyek awal mencapai 60%, teridentifikasi masalah krusial bahwa mitra awal tidak mengalokasikan dana untuk kelanjutan pengembangan website.
- **Inisiatif Pengambilan Keputusan Cepat :** Segera mengambil tindakan dengan melakukan pertemuan langsung dengan Dosen Pembimbing untuk mencari solusi strategis atas proyek yang terhenti.
- **Solusi & Pivot Proyek yang Berhasil :**
 1. Berhasil mendapatkan mitra baru (Alumni Ganissa) melalui rekomendasi dan fasilitasi dari pembimbing.
 2. Memimpin reorganisasi tim, melakukan koordinasi cepat untuk membahas rancangan proyek baru, dan mengalihkan fokus pengembangan dengan sigap.



REALISASI TUGAS PM - KOORDINASI & KOMUNIKASI TIM

Menjaga alur komunikasi yang efektif adalah kunci untuk memastikan semua anggota tim bekerja secara sinkron dan produktif.

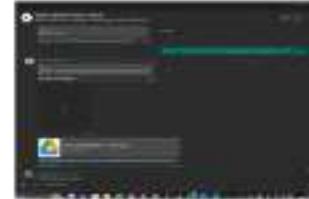
- **Membangun Saluran Komunikasi :** Menginisiasi penggunaan platform komunikasi utama (Discord) untuk diskusi rutin, pembagian tugas, dan koordinasi harian.
- **Memfasilitasi Pertemuan Rutin :** Merencanakan dan memimpin pertemuan tim secara berkala, baik secara online maupun offline di kampus untuk membahas keberlanjutan proyek.
- **Menjaga Alur Informasi :** Bertindak sebagai pusat informasi, memastikan setiap anggota memahami tugasnya, linimasa pengerjaan, dan perubahan-perubahan yang terjadi selama proyek berlangsung, serta secara aktif menanyakan kendala dan memfasilitasi diskusi untuk mencari solusi bersama.



REALISASI TUGAS PM - MONITORING PROGRES & KUALITAS

Selain perencanaan, pengawasan aktif selama tahap pengembangan sangat penting untuk mencapai hasil akhir yang berkualitas.

- **Pemantauan Progres Berkelanjutan:**
 - Secara rutin memantau perkembangan dan hasil kerja dari setiap anggota tim sesuai dengan jobdesk yang telah dibagikan.
 - Memastikan setiap bagian dari proyek, mulai dari analisis, desain, hingga pengkodean, berjalan sesuai linimasa yang direncanakan.
- **Penilaian dan Revisi Hasil Kerja:**
 - Melakukan evaluasi terhadap pekerjaan yang telah diselesaikan oleh anggota tim.
 - Memberikan feedback, penilaian, dan arahan revisi untuk memastikan hasil akhir produk sesuai dengan kebutuhan fungsional dan standar kualitas yang ditetapkan.



REALISASI TUGAS PM - FINALISASI & PENYERAHAN PROYEK

Tahap akhir adalah mengawal proyek hingga tuntas dan diterima dengan baik oleh mitra.

- **Manajemen Tahap Akhir:** Memimpin koordinasi tim untuk finalisasi seluruh fitur proyek, penyusunan laporan akhir, dan persiapan untuk proses deployment.
- **Pengelolaan Proses Deployment:** Mengawasi dan berkoordinasi dalam proses pencarian serta pemilihan layanan hosting yang sesuai dengan kebutuhan dan anggaran proyek.
- **Bimbingan Final dan Penyerahan:**
 - Melakukan sesi bimbingan tahap akhir dengan dosen pembimbing untuk finalisasi produk dan laporan.
 - Memimpin acara penyerahan dan presentasi produk akhir secara resmi kepada pihak mitra, yang menandai keberhasilan penyelesaian proyek.



3. Sekretaris – Hasan Nur Rasyid

KONTRIBUSI SEKRETARIS

Dokumentasi Proyek: Bertanggung jawab mencatat hasil rapat tim, hasil bimbingan, logbook dll.

1

Manajemen Jadwal dan Tugas: Mengelola jadwal rapat, pembagian tugas, dan mengingatkan anggota tim tentang tenggat waktu proyek.

2

Penyimpanan Dokumen: Menyimpan seluruh dokumen proyek secara terstruktur

3

REALISASI SEKRETARIS

Koordinasi dengan Dosen Pembimbing: Mendokumentasikan proses menghubungi dan menjadwalkan bimbingan dengan Dr. Murinto, S.Si., M.Kom.

1

Pencatatan Log Book: Mengisi log book mingguan dan individu untuk mencatat kegiatan, durasi, hasil, dan kendala selama proyek berlangsung, baik sebelum maupun setelah UTS.

2

Dokumentasi Kegiatan Tim: Menyimpan bukti-bukti dokumentasi kegiatan internal tim, seperti foto-foto pertemuan dan tangkapan layar diskusi.

3

DOKUMENTASI KRITIS YANG DIKELOLA

1
 Proposal dan Laporan Proyek: Memastikan dokumen proposal dan laporan akhir proyek terstruktur dengan baik.

2
 Log Book Kelompok dan Individu: Menyajikan catatan harian atau mingguan dari setiap anggota tim sebagai bukti pelaksanaan tugas dan bimbingan.

3
 Surat dan Kontrak: Menyimpan surat kontrak kerja sama dengan mitra (ALUMNI GANISSA) yang mencantumkan ruang lingkup, hak, kewajiban, dan nilai perjanjian.

4
 Berita Acara Serah Terima (BAST): Mendokumentasikan Berita Acara Serah Terima proyek sebagai bukti bahwa kewajiban pengembang telah selesai dan sistem telah diserahkan kepada mitra.

4. System Analyst – Andriansyah

SYSTEM ANALYST

Perancangan Sistem

Perancangan sistem ini bertujuan untuk menyajikan gambaran menyeluruh kepada pengguna mengenai sistem yang akan dikembangkan.

F-ALUM-001	Regulasi & Analisis Alam Mambak	Sistem harus memaparkan data alam melalui alat bantu secara mandiri, real-time, dan akurat.
F-ALUM-002	Manajemen & Penerimaan Profit Mambak	Sistem harus memaparkan Alami, menerima, dan mengelola profil pribadi secara mandiri, sistem harus menyediakan fitur pencarian status dengan filter dan urutan hasil berdasarkan lokasi, tanggal.
F-ALUM-003	Pemasaran & Nilai Ekonomi Lumbung	Sistem harus memaparkan Alami mengelola, berkolaborasi, dan mengelola konten seperti artikel atau berita.
F-ALUM-004	Rilis & Indikator Konten Berita, RTK&B	Sistem harus memaparkan Alami mengelola, berkolaborasi, dan mengelola konten seperti artikel atau berita.

Kebutuhan Fungsional

Kebutuhan fungsional merupakan sekumpulan fungsi yang didokumentasikan untuk menggambarkan kemampuan sistem dalam menjalankan berbagai tugas guna memenuhi kebutuhan pengguna.

ID Kebutuhan Fungsional	Fungsionalitas Sistem	Deskripsi Fungsionalitas Sistem
F-FUN-001	Manajemen Akun Alami	Sistem harus memaparkan Alami mengelola akun Alami dan, menerima, membuat, mengelola, dan mengelola profil Alami.
F-FUN-002	Manajemen Data Profil Alami	Sistem harus memaparkan Alami melakukan CRUD data profil Alami secara lengkap.
F-FUN-003	Search & Rekomendasi	Sistem harus menyajikan rekomendasi dengan algoritma pengujian, analisis, dan statistik.

F-QUEST-001	Akses Informasi Publik Terbuka	Sistem harus memaparkan Alami mengelola informasi publik tanpa login.
F-QUEST-002	Perisipasi (Artikel & Detail Event Publik)	Sistem harus memaparkan Alami melihat, menulis, dan detail event publik.
F-QUEST-003	Proses Registrasi Awal Alami	Sistem harus memaparkan Alami mengelola formulir pendaftaran awal Alami sebelum dikawal Alami.

Kebutuhan Non Fungsional

Kebutuhan non fungsional merupakan jenis kebutuhan yang tidak berhubungan langsung dengan fungsi utama sistem, melainkan berfokus pada aspek-aspek penting lainnya seperti kinerja, keamanan, dan kemudahan penggunaan.

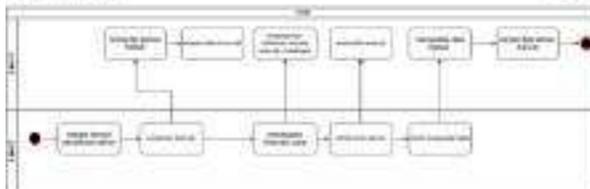
NF-01A	Kompatibilitas	Sistem harus dapat bekerja dengan baik di browser web populer (misal, Chrome, Firefox, Edge, Safari) dan perangkat seluler operasi yang umum.
NF-01B	Kompatibilitas	Sistem harus memiliki kemampuan untuk berinteraksi dengan sistem eksternal penting melalui API yang disediakan dengan baik jika diperlukan.
NF-02	Kinerja	Sistem harus dirancang untuk dapat menangani beban pengguna (skalabilitas) untuk menangani pertumbuhan data dan jumlah pengguna di masa mendatang (misal, hingga 10.000 pengguna terdaftar).
NF-03	Kinerja	Penanganan sumber daya server (CPU, memori) harus dioptimalkan untuk memastikan stabilitas dan kinerja yang konsisten selama operasi.
NF-04	Kepuasan	Antarmuka pengguna (UI) harus intuitif, mudah dinavigasi, dan konsisten di seluruh aplikasi, dengan desain yang menarik untuk berbagai perangkat (desktop, mobile).

NF-05A	Keamanan	Sistem harus memastikan mekanisme autentikasi dan otorisasi yang kuat (misal, password hashing, RBAC) untuk melindungi data pengguna dan data sensitif.
NF-05B	Keamanan	Seluruh transaksi data (baik di transit) harus dienkripsi (HTTPS/TLS), dan data pribadi yang disimpan (data di repositori) harus dienkripsi untuk mencegah akses tidak sah.
NF-06A	Keandalan	Seluruh sistem harus memiliki dan dapat dipulihkan dengan baik untuk memastikan ketersediaan, pemertahanan data, dan pembaruan di masa mendatang.
NF-06B	Keandalan	Sistem harus memiliki logging aktivitas dan monitoring kinerja untuk memudahkan identifikasi masalah dan pemecahan masalah.
NF-07	Kepuasan	Pengguna baru harus dapat dengan mudah mendaftar dan menggunakan fitur-fitur inti sistem secara persis sebelum akses.
NF-08	Kepuasan	Sistem harus menyediakan pesan kesalahan yang jelas dan informatif, serta pilihan bantuan yang terdapat untuk membantu pengguna.

Perancangan Proses Bisnis

Proses Bisnis Sebelum Ada Sistem

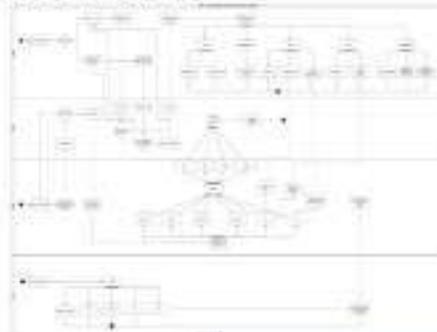
Proses bisnis yang lebih sederhana ini merupakan proses bisnis manual dalam pengolahan data alumni yang terjadi sebelum adanya sistem digital. Aktivitas utama bergantung pada interaksi langsung antara alumni dan admin. Di masa sebelum menyebarkan formulir fisik dan admin secara manual memasukkan data ke dalam spreadsheet Excel.



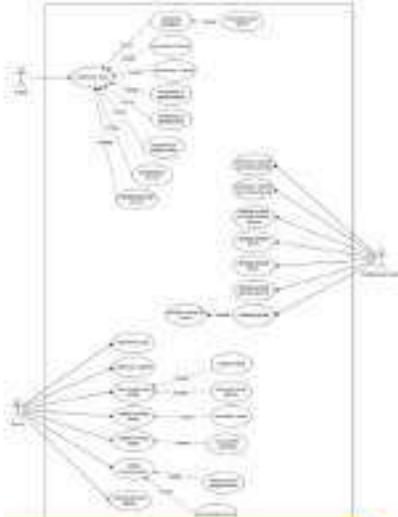
Sebelum adanya sistem, pengolahan data alumni dilakukan secara manual dan tidak terotomatisasi, sehingga proses input, pencarian, dan validasi data memerlukan waktu lama dan rentan kesalahan. Informasi kegiatan serta dokumentasi seperti galeri foto juga dibagikan secara terbatas dan tidak terorganisir dengan baik. Setelah sistem informasi berbasis web diterapkan, seluruh proses menjadi lebih efisien karena data alumni dapat diinput dan dikelola secara digital, pencarian data menjadi cepat, informasi dapat dibagikan secara luas melalui website, dan semua data terdokumentasi dalam satu platform yang mudah diakses oleh admin maupun pengguna.

Proses Bisnis Setelah Ada Sistem

Setelah proses yang dipertanyakan, dapat diidentifikasi bahwa sebelum adanya sistem terapan, proses pencarian dan komunikasi dengan alumni menggunakan buku diadukan secara manual dan berpaginasi. Panitia, alumni, dan alumni tidak memiliki satu platform tunggal untuk berinteraksi dan berbagi informasi. Data alumni tersebar terdistribusi, informasi lowongan kerja dan acara sulit diakses secara efisien, dan ritel adalah salah satu cara komunikasi antara alumni.



Perancangan Use Case Diagram



Admin

- Admin memiliki akses penuh untuk login, mendaftarkan,
- Mengelola pengguna, blog, acara, pengumuman, Instagram, jurusan, dan halaman "Tentang Kami".
- Melihat dan mengelola pesan dari pengunjung.
- Mencetak kartu alumni.

Alumni

- Alumni dapat:
- Registrasi & login.
 - Mengelola profil dan mencetak kartu alumni.
 - Melihat dan mendaftar event.
 - Membaca blog dan memberi komentar.
 - Melihat dan menanggapi pengumuman.
 - Melihat direktori alumni.

Pengunjung

- Pengunjung (tanpa login) dapat:
- Melihat profil alumni, event, blog, pengumuman, dan kontak.
 - Registrasi sebagai alumni.
 - Mengirim pesan ke admin lewat halaman kontak.

Perancangan Class Diagram

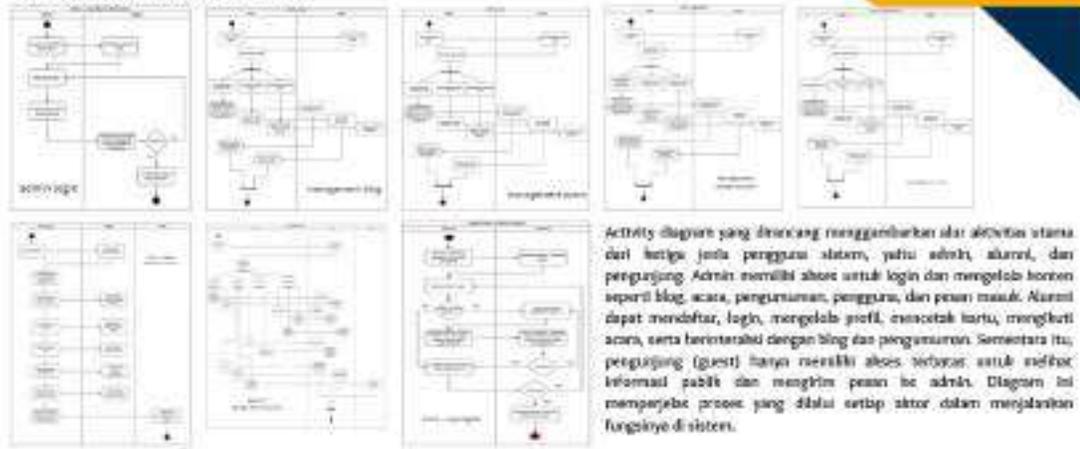


Class diagram ini menggambarkan struktur sistem informasi yang digunakan untuk mengelola data alumni, admin, dan laman website seperti blog, event, dan pengumuman.

- Admin adalah pusat kendali yang memiliki akses untuk menambahkan, melihat, mengedit, dan menghapus data pada hampir seluruh entitas dalam sistem.
- Alumni merupakan pengguna utama yang dapat mendaftar, login, memperbaiki profil, dan mengikuti event.
- Profile berisi data pribadi lengkap dari alumni seperti alamat, pekerjaan, dan kontak.
- Jurusan berfungsi untuk mengelompokkan alumni berdasarkan program studi.
- Blog, Event, dan Pengumuman adalah fitur website yang dapat dikelola oleh admin untuk menyampaikan informasi kepada alumni.
- Tentang Kami menyajikan informasi untuk mengenai institusi.
- Manajemen User digunakan untuk mengelola akun pengguna selain alumni, seperti staf atau pihak jurusan.

Relasi antar class menunjukkan keterkaitan data, seperti data alumni memiliki data profil, satu jurusan memiliki banyak alumni, dan satu event bisa diikuti oleh banyak alumni.

Perancangan Activity Diagram



5. UI/UX – Farqad Assanji

DARI KONSEP MENUJU DESAIN VISUAL (UI/UX)

Merancang Fondasi Antarmuka yang Modern dan Intuitif

Tahap awal berfokus pada perancangan visual sebelum proses coding dimulai. Proses ini mencakup:

1. Penentuan Tema Visual

Timi sepakat menggunakan tema warna kuning yang diambil dari identitas logo Garuda untuk halaman utama, serta kombinasi warna biru dan merah untuk dasbor admin dan alumni.

1

2. Perancangan Wireframe & Mockup

Membuat rancangan awal (kerangka) untuk setiap halaman, seperti Dasbor Alumni, Halaman Profil, dan Direktori untuk memetakan struktur dan fungsionalitas.

2

3. Desain Final (High-Fidelity)

Mengembangkan desain detail di Figma yang menjadi cetak biru (blueprint) untuk implementasi front-end, memastikan setiap elemen visual konsisten dan mudah digunakan.

3

PROSES DESIGN UI/UX



MEMBANGUN ANTARMUKA HOMEPAGE



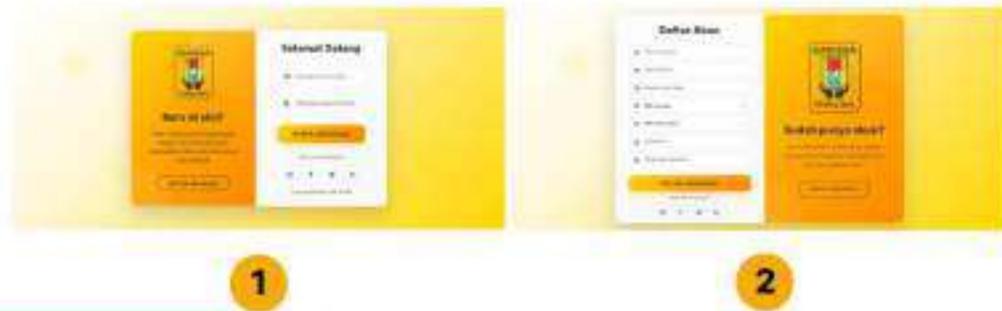
MEMBANGUN DASHBOARD ALUMNI



MEMBANGUN DASHBOARD ADMIN



MERANCANG HALAMAN LOGIN DAN REGISTER



6. BE 1 – Hasan Nur Rasyid

KONTRIBUSI BACKEND DEV

Otak di balik layar yang mengelola data, logika bisnis, dan interaksi kompleks.

Memastikan data profil aman, pencarian efisien, dan semua fitur berjalan sesuai harapan.

Membangun fondasi yang kokoh dan aman.

1

2

3

PENGELOLAAN IDENTITAS & HAK AKSES PENGGUNA

F-ALUM-001: Registrasi & Aktivasi Akun Mandiri: "Membangun API untuk alur pendaftaran alumni yang mandiri dan terverifikasi."

F-GUEST-003: Proses Registrasi Awal Alumni: "Memfasilitasi formulir pendaftaran awal untuk calon alumni."

F-ALUM-002: Manajemen & Pembaruan Profil Mandiri: "Mengembangkan endpoint bagi alumni untuk mengelola data pribadi mereka (update profil, kontak, riwayat)."

1

2

3

Memastikan proses onboarding pengguna yang mulus dan aman, serta kontrol penuh pengguna atas data mereka.

MEMBANGUN SISTEM MANAJEMEN AKUN & OTORISASI

F-ADM-001: Manajemen Akun Admin: "Mengimplementasikan fitur CRUD (Create, Read, Update, Delete) untuk Admin mengelola akun admin lain."

F-ADM-006: Manajemen Akun Pengguna & Otorisasi (RBAC): "Mendesain dan mengimplementasikan sistem Role-Based Access Control (RBAC) yang membedakan hak akses untuk Admin, Alumni, dan Guest."

1

2

Menyediakan lapisan keamanan yang kuat, memastikan setiap peran hanya mengakses fungsionalitas sesuai wewenangnya, dan menjaga integritas sistem.

MEMFASILITASI AKSES INFORMASI & INTERAKSI DATA

F-ALUM-003: Pencarian & Filter Alumni Lanjutan: "Mengembangkan logika backend untuk pencarian dan filter data alumni yang cepat dan akurat (berdasarkan angkatan, jurusan, dll.)."

F-ALUM-004: Akses & Interaksi Konten (Berita, Artikel): "Membangun API untuk pengambilan konten berita/artikel, termasuk kemampuan interaksi seperti like atau share."

F-GUEST-001: Akses Informasi Publik Terbatas: "Memastikan akses aman bagi non-pengguna terhadap informasi publik yang telah ditentukan."

1

2

3

Meningkatkan konektivitas antar alumni melalui pencarian yang efektif, dan memberikan akses mudah ke informasi relevan bagi semua pengunjung.

MENDORONG PARTISIPASI & MENJAGA KUALITAS KONTEN

F-ALUM-005: Registrasi & Partisipasi Event Online/Offline: "Mengimplementasikan sistem pendaftaran event dan pengelolaan data peserta di sisi backend."

F-GUEST-002: Penampil Jadwal & Detail Event Publik: "Menyediakan endpoint untuk menampilkan jadwal dan detail event yang bersifat publik."

F-ALUM-006: Kontribusi Konten (Artikel, Pengalaman, Portofolio): "Membangun API untuk alumni dapat mengirimkan dan mengelola konten kontribusi mereka."

F-ADM-007: Sistem Moderasi Konten & Komunikasi: "Mengembangkan modul backend untuk Admin menyetujui, menolak, atau menghapus konten dan komentar yang dikirimkan."

1

2

3

4

TEKNOLOGI KUNCI & DAMPAK NYATA KONTRIBUSI SAYA

1

Teknologi yang Digunakan:
Bahasa Pemrograman PHP
Framework Laravel
Database MySQL

2

Dampak Keseluruhan:
Berhasil membangun backend yang stabil, aman, dan efisien, menjadi tulang punggung bagi semua fungsionalitas inti sistem alumni.
Memastikan pengelolaan data yang terstruktur dan terintegrasi, siap untuk pengembangan fitur-fitur lanjutan.

3

Pembelajaran Kunci:
Memahami kompleksitas desain database untuk relasi data alumni yang beragam

7. FE 1 – BE 2 – Noval Lias Ramadani

PERAN GANDA : FRONTEND 1 & BACKEND 2

Sebagai Frontend Developer 1:

- Bertanggung jawab mengembangkan seluruh antarmuka pengguna (UI) aplikasi sesuai dengan desain yang telah dirancang oleh tim UI/UX.
- Mengimplementasikan logika tampilan menggunakan HTML, CSS, dan JavaScript, serta memastikan desain yang responsif dan berkinerja baik di berbagai perangkat.
- Berkolaborasi secara intensif dengan tim backend untuk mengambil dan menampilkan data secara dinamis di antarmuka pengguna.

Sebagai Backend Developer 2:

- Mendukung tugas Backend 1 dalam pengembangan server, database, dan integrasi sistem secara keseluruhan.
- Berperan dalam mengoptimalkan performa server dan query database untuk memastikan aplikasi berjalan efisien.
- Membantu proses deployment dan manajemen infrastruktur teknis proyek.

REALISASI FRONTEND 1 : ANTARMUKA PUBLIK (GUEST)

Membangun "wajah" aplikasi yang informatif dan menarik bagi pengunjung umum yang belum login.

- **Landing Page** : Merancang dan mengimplementasikan halaman utama yang interaktif, menampilkan hero section, ringkasan "Tentang Kami", serta pratinjau desain untuk event dan blog terbaru.
- **Halaman Informasi** : Membangun halaman-halaman utama yang dapat diakses publik, seperti halaman detail "Tentang Kami", "Daftar Acara", "Blog", dan "Pengumuman".
- **Fitur Pencarian & Filter** : Mengimplementasikan fungsionalitas pencarian dan filter pada halaman Acara, Blog, dan Pengumuman untuk memudahkan navigasi dan pencarian informasi oleh pengguna.
- **Formulir Kontak** : Membuat halaman kontak yang fungsional, memungkinkan pengunjung mengirimkan pesan langsung kepada admin melalui sistem.



REALISASI FRONTEND 1 : PORTAL INTERAKTIF ALUMNI

Menciptakan pengalaman pengguna yang kaya fitur dan personal bagi alumni yang telah login ke dalam sistem.

- **Dasbor Personal** : Mendesain dasbor utama alumni yang menyajikan ringkasan informasi relevan seperti Acara Mendatang, Kisah Alumni terbaru, dan Pengumuman penting.
- **Manajemen Profil Mandiri** : Mengembangkan fitur lengkap bagi alumni untuk melihat, mengedit, menambahkan riwayat pendidikan, dan mencetak kartu alumni digital mereka.
- **Direktori & Jaringan** : Membangun halaman direktori alumni yang interaktif, dilengkapi fitur pencarian dan filter (berdasarkan jurusan dan tahun lulus) untuk memfasilitasi koneksi antar alumni.
- **Partisipasi Event & Interaksi** : Mengimplementasikan alur lengkap untuk partisipasi acara, mulai dari melihat detail, mendaftar, hingga mengunggah bukti pembayaran untuk event berbayar.



REALISASI FRONTEND 1 : PANEL KONTROL ADMINISTRATIF

Membangun antarmuka yang kuat dan intuitif untuk admin agar dapat mengelola seluruh aspek sistem dengan mudah.

- **Dasbor Analitik:** Merancang dasbor admin yang menampilkan statistik kunci (jumlah pengguna, blog, event, pengumuman) dan ringkasan aktivitas terbaru untuk pemantauan cepat.
- **Manajemen Data (CRUD):** Mengembangkan antarmuka pengguna untuk semua fungsionalitas Create, Read, Update, Delete (CRUD) pada setiap modul, termasuk: Manajemen Pengguna (Admin & Alumni), Manajemen Konten (Blog, Acara, Pengumuman, Tentang Kami), Manajemen Master Data (Jurusan, Kategori Blog).
- **Fitur Moderasi:** Membuat halaman khusus bagi admin untuk melihat dan mengelola pesan masuk serta memoderasi pendaftaran peserta event.



REALISASI BACKEND 2 : LOGIKA PANEL ADMIN

Menyediakan "mesin" yang andal di balik layar untuk semua fungsionalitas yang ada di panel admin.

- **Logika Dasbor & Statistik:** Mengembangkan AdminDashboardController untuk mengambil dan menghitung data agregat (total pengguna, blog, event, dll.) secara efisien dari database untuk ditampilkan di dasbor.
- **Logika CRUD Pengguna:** Membangun AdminUserController yang menangani semua proses manajemen data alumni oleh admin, termasuk validasi data, upload foto, dan pengelolaan relasi antar tabel (users, alumni_profiles, majors).
- **Logika Manajemen Konten:** Membuat controllers (BlogController, AnnouncementController, dll.) untuk menangani logika CRUD pada semua jenis konten, termasuk pembuatan slug otomatis, penjadwalan publikasi, dan manajemen status.
- **Sistem Routing & Otentikasi:** Mendefinisikan semua rute (routes/web.php) untuk fitur admin dan menerapkan middleware auth dan admin untuk melindungi akses dan memastikan hanya admin yang berwenang yang dapat melakukan perubahan.



REALISASI BACKEND 2 : LOGIKA KOMPLEKS EVENT & PENDAFTARAN

Mengimplementasikan salah satu alur bisnis paling kompleks dalam sistem, yaitu manajemen event dari awal hingga akhir.

- **Logika Manajemen Event:** Mengembangkan EventController untuk menangani logika CRUD event, termasuk validasi kompleks pada tipe audiens (jumlah, khusus jurusan/langkahi), batas peserta, dan informasi pembayaran.
- **Logika Pendaftaran:** Membangun EventRegistrationController untuk mengelola alur pendaftaran alumni ke sebuah event, termasuk melakukan pengecekan kelayakan peserta dan ketersediaan kuota secara real-time.
- **Moderasi Pembayaran:** Menciptakan logika di sisi server bagi admin untuk melakukan verifikasi bukti pembayaran, serta mengubah status pendaftaran dari 'pending' menjadi 'confirmed' atau 'rejected'.
- **Struktur Model & Relasi:** Merancang dan mengimplementasikan model Event dan EventRegistration dengan relasi database yang tepat (hasMany, belongsTo) untuk memastikan integritas dan konsistensi data.



REALISASI BACKEND 2 : DUKUNGAN SISTEM & INTEGRASI

Memastikan seluruh komponen backend bekerja secara harmonis dan siap untuk digunakan di lingkungan produksi.

- **Dukungan Integrasi Sistem:** Berperan aktif mendukung Backend 1 dalam memastikan semua controllers, models, dan layanan terintegrasi dengan mulus, sehingga data dapat mengalir lancar dari backend ke antarmuka frontend.
- **Optimasi Performa:** Terlibat dalam proses optimasi, khususnya pada query database, untuk memastikan halaman yang memuat banyak data (seperti dashboard dan daftar pengguna) dapat merespon dengan cepat.
- **Pengembangan Fitur Pendukung:** Membangun ContactController untuk menangani logika penerimaan, penyimpanan, dan pembaruan status 'telah dibaca' pada pesan yang masuk dari formulir kontak publik.
- **Bantuan Deployment:** Sesuai jobdesk Backend 2, turut serta membantu dalam proses deployment dan manajemen infrastruktur teknis, memastikan aplikasi dapat berjalan dengan baik di server hosting.



8. QA/Tester – Destyawan Satria Nugraha

PERAN QA TESTER DALAM PROYEK GANISSA

- Quality Assurance (QA) dilakukan untuk memastikan bahwa semua fitur yang telah dikembangkan berjalan sesuai dengan rancangan sistem yang telah disusun pada tahap sebelumnya.
- Proses ini juga bertujuan untuk menjamin bahwa aplikasi bebas dari kesalahan (bug) yang dapat memengaruhi fungsionalitas utama, baik dari sisi pengguna admin, alumni, maupun tamu.
- Selain itu, QA berfungsi sebagai upaya untuk menilai kesesuaian sistem dengan kebutuhan dan ekspektasi pengguna akhir, sehingga aplikasi dapat digunakan dengan baik dan memberikan pengalaman pengguna yang optimal.

1

2

3

METODOLOGI PENGUJIAN

Metode Pengujian: Black Box Testing

- Fokus utama adalah pada respon sistem terhadap berbagai input, untuk menilai apakah output sesuai dengan yang diharapkan.
- Alat yang Digunakan: Feature Test Laravel
- Laravel menyediakan fitur automated testing yang disebut Feature Test.
- Feature Test digunakan untuk mensimulasikan interaksi pengguna dengan aplikasi, seperti membuka halaman, mengisi formulir, mengklik tombol, dan sebagainya.
- Proses Pelaksanaan:
 - Setiap skenario pengujian ditulis sebagai kode uji otomatis menggunakan PHPUnit, yang kemudian dijalankan untuk mengevaluasi setiap fitur.
 - Pengujian dilakukan secara menyeluruh terhadap modul-modul utama seperti Admin, Alumni, Autentikasi, dan Halaman Publik.
 - Hasil pengujian dibandingkan dengan expected result untuk menentukan apakah sistem lulus uji atau tidak.

1

2

3

LINGKUNGAN PENGUJIAN SOFTWARE DAN TOOLS

- Framework: Laravel
- Alat Uji: PHPUnit
- OS: Windows
- Tools Pendukung: Laragon, Composer, PHP, Git
- Database: Terisolasi khusus pengujian (tidak mengganggu DB dev)

1

2

3

SKENARIO PENGUJIAN

- Total: 109 kasus uji
- Modul yang diuji:
 - Admin
 - Alumni
 - Sistem Autentikasi
 - Halaman Publik
- Skenario disusun untuk mencakup semua fitur utama

1

2

3

