

BAB I

Pendahuluan

1.1 Latar Belakang

Lembaga Amil Zakat, Infaq, dan Shadaqah Muhammadiyah (LAZISMU) Yogyakarta merupakan organisasi zakat, infaq, dan shadaqah berskala nasional yang bertujuan untuk memberdayakan masyarakat. Lembaga ini mengelola dan memanfaatkan dana zakat, infaq, wakaf, serta sumbangan lainnya secara produktif, baik yang berasal dari individu, lembaga, perusahaan, maupun instansi lainnya.

Kantor layanan Lazismu tersebar di berbagai wilayah di Indonesia, termasuk salah satunya di Yogyakarta. Dalam menjalankan transaksi, Lazismu DIY menggunakan sistem pelayanan berbasis komputer yang masih mengandalkan aplikasi Microsoft Excel. Berdasarkan hasil pengamatan dan observasi, ditemukan beberapa kendala dalam sistem tersebut, antara lain:

Sistem yang ada saat ini belum mampu mengklasifikasikan perhitungan zakat yang dilakukan dengan metode konvensional, pencatatan data donatur memakan waktu yang lama, serta rekapitulasi laporan terkait informasi klasifikasi untuk Loyalty Badges pada infaq zakat masih kurang efisien. Ketiga hal ini berdampak negatif berupa penurunan efektivitas pengelolaan zakat, potensi keterlambatan dalam pelaporan data donatur, meningkatnya risiko kesalahan pencatatan atau penghitungan, serta beban kerja tambahan bagi admin yang harus melakukan pencatatan dan koreksi manual, sehingga mengurangi produktivitas dan efisiensi operasional secara keseluruhan.

Melihat kondisi tersebut, sistem yang digunakan untuk menangani permasalahan tersebut perlu disesuaikan dengan perkembangan yang ada. Oleh karena itu, diperlukan sebuah sistem informasi yang mampu meningkatkan kualitas pelayanan di Lazismu Yogyakarta. Berdasarkan permasalahan yang telah diidentifikasi, laporan ini diberi judul “Pengembangan *Back*

- End Sistem Kategorisasi Muzakki untuk Optimalisasi Pengelolaan Donasi di Lazimu DIY“

1.2 Batasan Masalah

Adapun Batasan masalah pada kegiatan praktik magang ini dengan membuat perancangan Sistem Klasifikasi Kategorisaasi Muzakki yang ada di Lazizmu DIY.

1. Mengembangkan Sistem informasi berbasis website yang mendukung pengelolaan donasi di Lazizmu DIY dengan fokus pada peningkatan pencatatan, klasifikasi dan pelaporan data donasi
2. Peningkatan efisiensi dan akurasi dalam perhitungan zakat, infaq, dan shadaqah yang sebelumnya dilakukan dengan sistem konvensional , melalui pengembangan sistem back – end yang memadai

1.3 Rumusan Masalah

Pengembangan sistem untuk menyajikan informasi klasifikasi loyalitas (Loyalty Badges) pada donatur berdasarkan data zakat dan infaq yang telah dikumpulkan

Rumusan Masalah

Berdasarkan latar belakang di atas, maka rumusan masalah yang akan dibahas yaitu:

1. Bagaimana membangun sistem back-end yang mampu mengklasifikasikan data muzakki secara otomatis berdasarkan kategori tertentu dengan menggunakan algoritma dan logika pemrograman yang tepat?
2. Bagaimana membangun sistem back-end yang efisien untuk pencatatan data muzakki, sehingga dapat mengurangi waktu yang dibutuhkan dalam proses administrasi serta meningkatkan akurasi data?

1.4 Tujuan Praktik Magang

Tujuan dari Kerja Praktik ini adalah untuk menemukan solusi atas berbagai permasalahan yang ada. Adapun tujuan spesifik dari pelaksanaan Kerja Praktik ini adalah sebagai berikut:

1. Mengembangkan sistem back-end yang mampu mengklasifikasikan data muzakki secara otomatis berdasarkan kategori tertentu dengan menggunakan algoritma dan logika

pemrograman yang tepat.

2. Mengimplementasikan sistem pencatatan data muzakki yang efisien untuk mengurangi waktu proses administrasi dan meningkatkan akurasi data.
3. Mengevaluasi kinerja sistem back-end yang telah dikembangkan melalui pengujian dan analisis terhadap kecepatan, efisiensi, serta keakuratan pengelolaan data muzakki dan transaksi.
4. Mendokumentasikan seluruh proses pengembangan, implementasi, dan evaluasi sistem untuk memberikan laporan yang terstruktur dan dapat digunakan sebagai referensi untuk pengembangan sistem di masa depan.

1.5 Manfaat Praktik Magang

Pelaksanaan kerja praktik (magang) ini diharapkan dapat memberikan manfaat baik bagi perusahaan maupun mahasiswa. Manfaat yang diharapkan dari kegiatan ini adalah sebagai berikut:

1. Bagi Lazismu Yogyakarta
 - a. Memberikan solusi praktis untuk meningkatkan efisiensi dan efektivitas pengelolaan data donasi dan laporan keuangan.
 - b. Mendukung pengelolaan zakat, infaq, dan shadaqah yang lebih produktif dan transparan dengan sistem berbasis teknologi.
2. Bagi Mahasiswa
 - a. Meningkatkan kemampuan mahasiswa dalam merancang dan mengembangkan sistem informasi berbasis teknologi sesuai dengan kebutuhan dunia kerja.

BAB II

Gambaran Instansi

2.1 Profil Instansi

LAZISMU adalah lembaga zakat tingkat nasional yang berkhidmat dalam pemberdayaan masyarakat melalui pendayagunaan secara produktif dana zakat, infaq, wakaf dan dana kederewanan lainnya baik dari perseorangan, lembaga, perusahaan dan instansi lainnya. Didirikan oleh PP. Muhammadiyah pada tahun 2002, selanjutnya dikukuhkan oleh Menteri Agama Republik Indonesia sebagai Lembaga Amil Zakat Nasional melalui SK No. 457/21 November 2002. Dengan telah berlakunya Undang-undang Zakat nomor 23 tahun 2011, Peraturan Pemerintah nomor 14 tahun 2014, dan Keputusan Menteri Agama Republik Indonesia nomor 333 tahun 2015. LAZISMU sebagai lembaga amil zakat nasional telah dikukuhkan kembali melalui SK Menteri Agama Republik Indonesia nomor 730 tahun 2016. LAZISMU adalah lembaga zakat tingkat nasional yang berkhidmat dalam pemberdayaan masyarakat melalui pendayagunaan secara produktif dana zakat, infaq, wakaf dan dana kederewanan lainnya baik dari perseorangan, lembaga, perusahaan dan instansi lainnya.

Latar belakang berdirinya LAZISMU terdiri atas dua faktor. Pertama, fakta Indonesia yang berselimut dengan kemiskinan yang masih meluas, kebodohan dan indeks pembangunan manusia yang sangat rendah. Semuanya berakibat dan sekaligus disebabkan tatanan keadilan sosial yang lemah. Kedua, zakat diyakini mampu bersumbangsih dalam mendorong keadilan sosial, pembangunan manusia dan mampu mengentaskan kemiskinan. Sebagai negara berpenduduk muslim terbesar di dunia, Indonesia memiliki potensi zakat, infaq dan wakaf yang terbilang cukup tinggi. Namun, potensi yang ada belum dapat dikelola dan didayagunakan secara maksimal sehingga tidak memberi dampak yang signifikan bagi penyelesaian persoalan yang ada.

Berdirinya LAZISMU dimaksudkan sebagai institusi pengelola zakat dengan manajemen modern yang dapat menghantarkan zakat menjadi bagian dari penyelesaian masalah (problem solver) sosial masyarakat yang terus berkembang.

2.2 Sumber Daya Penunjang Magang

2.2.1 Struktur organisasi di LAZISMU DIY

Struktur Organisasi LAZISMU DIY dapat dilihat pada Gambar 2.1



Gambar 2. 1 Struktur LAZISMU DIY.

Gambar 2.1 merupakan representasi visual dari struktur organisasi yang menunjukkan seorang pemimpin di puncak, diikuti oleh berbagai divisi dan departemen yang masing-masing diwakili oleh anggota tim, garis yang menghubungkan mereka untuk menunjukkan hubungan hierarki dalam sebuah

organisasi

2.2.2 Sumber Daya Manusia

Lazismu DIY memiliki sumber daya yang mencukupi untuk menjalankan usaha baik dari sumber daya manusia atau sumber daya fisik yang dikembangkan oleh Lazismu DIY. Sumber daya yang dimiliki oleh Lazismu DIY adalah sebagai berikut:

Tabel 2. 1 Sdm LAZISMU DIY

No.	Pengelola	Jumlah
1.	Ketua Badan Pengurus	1 orang
2.	Dewan Pengawas Syari'ah	4 orang
3.	Sekretaris	1 orang
4.	Wakil Sekretaris	1 orang
5.	Wakil Ketua Bidang Audit Kepatuhan dan Keuangan	1 orang
6.	Anggota	2 orang
7.	Wakil Ketua Bidang Perhimpunan dan Kerjasama	1 orang
8.	Anggota	3 orang
	Wakil ketua Bidang Kelembagaan&Sumber Daya Amil	1 orang
	Anggota	3 orang
9.	Wakil ketua Bidang Pendistribusian&Pendayagunaan	1 orang
10.	Anggota	3 orang
11.	Wakil ketua Bidang Transformasi Digital, Monitoring dan Evaluasi	1 orang
12.	Anggota	2 orang

Tabel 2. 1 Sdm LAZISMU DIY

No.	Nama Bagian	Jumlah
1.	Manager Regional	1 orang
2.	Staff Keuangan	2 orang
3.	Staff Administrasi	1 orang
4.	Staff Digital Fundraising	1 orang
5.	Staff Publikasi	1 orang
6.	Staff Fundarising Corporate&Ritail	1 orang
7.	Staff Biro Umum	1 orang
8.	Staff Pendayagunaan&Pendistribusiann	2 orang

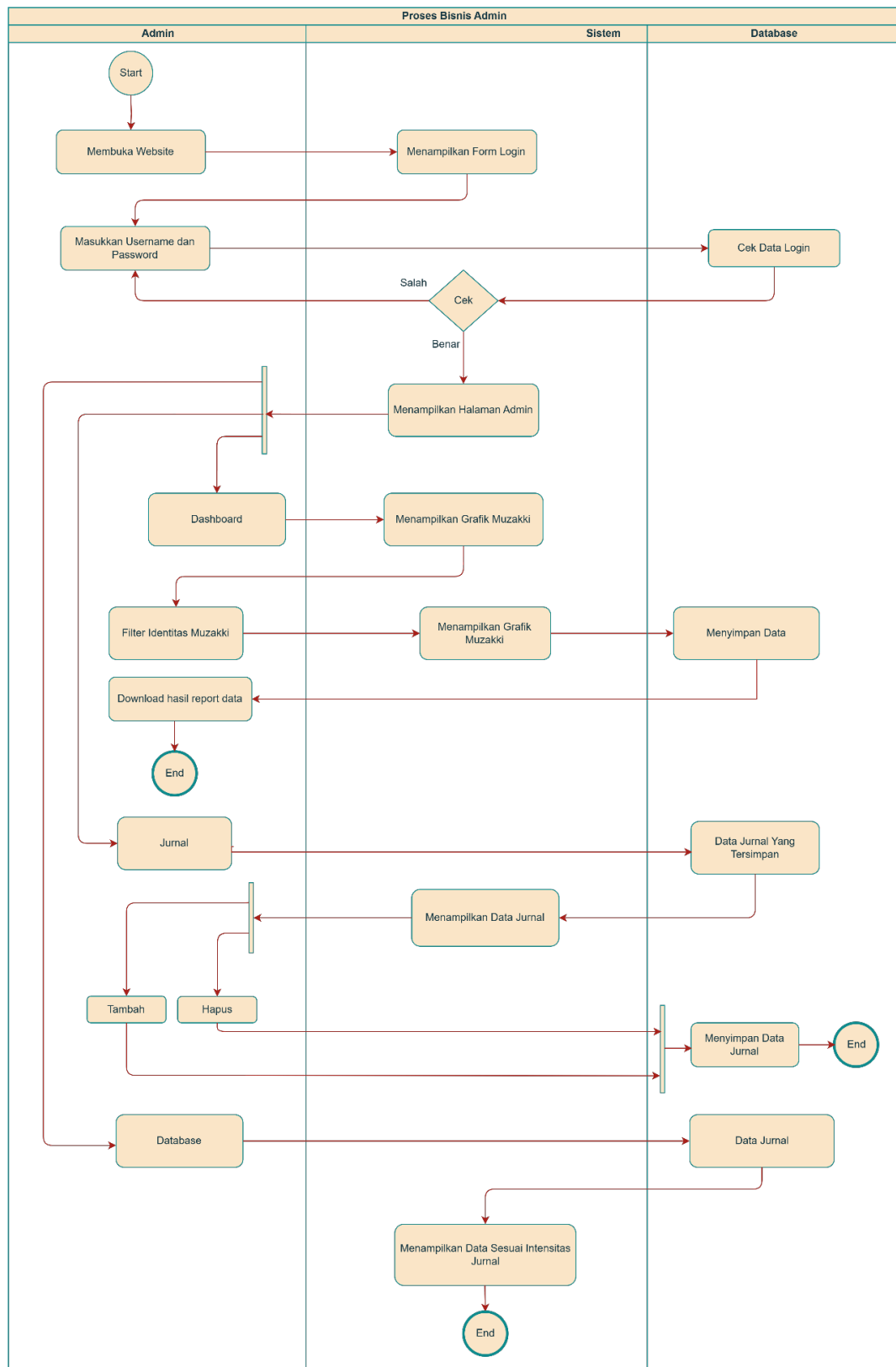
2.2.3 Sumber Daya Fisik

Sumber daya fisik adalah sumber daya yang dapat dilihat dan disentuh, serta dapat digunakan untuk mendukung kegiatan. Salah satu sumber daya fisik di Lazismu DIY adalah fasilitas material yang dimiliki Lazismu DIY. Fasilitas material ini terdiri dari:

- a) Ruang Lazismu
- b) Komputer
- c) Laptop
- d) Printer
- e) Mobil
- f) Proyektor

2.3 Proses Bisnis yang Diharapkan

Proses bisnis yang diharapkan di LAZISMU dapat dilihat pada Gambar 2.2



Gambar 2.2 Proses Bisnis yang diharapkan di LAZISMU DIY

Gambar 2.2 menggambarkan proses bisnis yang diharapkan dalam pencatatan muzakki dalam pengkategorian muzakki di LAZISMU DIY . Proses bisnis dimulai ketika admin membuka website, di mana sistem menampilkan form login untuk memasukkan username dan password. Setelah admin mengisi kredensial, sistem memverifikasi data login dengan database. Jika valid, admin diarahkan ke halaman dashboard yang menampilkan grafik Muzakki sebagai ringkasan data. Di dashboard, admin dapat melakukan berbagai aktivitas, seperti memfilter identitas Muzakki untuk mendapatkan data tertentu, yang kemudian ditampilkan dalam bentuk grafik dan disimpan ke database. Admin juga memiliki opsi untuk mengunduh laporan hasil filter tersebut. Selanjutnya, admin dapat mengelola data jurnal, seperti menambah data baru, menghapus data yang tidak diperlukan, atau melihat data jurnal yang tersimpan berdasarkan intensitas tertentu. Semua perubahan yang dilakukan pada data jurnal disimpan langsung ke database. Setelah semua aktivitas selesai, admin dapat menyelesaikan proses dengan keluar dari sistem.

BAB III

Metode Pelaksanaan Praktik Magang

3.1 Tahapan Persiapan

Tahapan persiapan untuk pembuatan sistem kategorisasi muzakki dimulai dengan identifikasi tujuan, yakni memudahkan staff dari LAZISMU dalam melakukan pencatatan muzakki yang melakukan donasi berdasarkan dari jumlah uang dan juga seringnya dalam melakukan donasi, sehingga dapat di kelompokkan berdasarkan kategori yang ada yang mana nantinya akan tergolong menjadi beberapa kategori yakni :

- a. Kecil Jarang (Orang yang Melakukan infaq kurang dari 3 kali dan jumlah keseluruhan donasi kurang dari Rp. 500.000,00) dan (Orang yang Melakukan zakat kurang dari 3 kali dan jumlah keseluruhan donasi kurang dari Rp. 1.000.000,00)
- b. Besar Jarang (Orang yang Melakukan infaq kurang dari 3 kali dan jumlah keseluruhan donasi sudah lebih dari Rp. 500.000,00) dan (Orang yang Melakukan zakat kurang dari 3 kali dan jumlah keseluruhan donasi lebih dari Rp. 1.000.000,00)
- c. Kecil Sering (Orang yang melakukan infaq lebih dari 3 kali dan jumlah keseluruhan donasi kurang dari Rp. 500.000,00) dan (Orang yang Melakukan zakat lebih dari 3 kali dan jumlah keseluruhan donasi kurang dari Rp. 1.000.000,00)
- d. Besar Sering (Orang yang melakukan infaq lebih dari 3 kali dan jumlah keseluruhan donasi sudah lebih dari Rp. 500.000,00) dan (Orang yang Melakukan zakat lebih dari 3 kali dan jumlah keseluruhan donasi lebih dari Rp. 1.000.000,00)

. Selain itu, persiapan peralatan yang diperlukan, seperti laptop dengan spesifikasi yang memadai untuk membuat sistem. Langkah pertama adalah memikirkan logic yang tepat untuk sistem , sehingga nantinya dapat dikembangkan menjadi sistem yang dapat mengelompokkan kategori muzakki berdasarkan donasi. Setelah membuat logic , langkah selanjutnya adalah

penentuan bahasa pemrograman dan *framework* yang tepat untuk pengembangan sistem. Setelah berdiskusi dan menganalisa rancangan sistem, kami pun memutuskan untuk menggunakan bahasa pemrograman TypeScript dan juga menggunakan beberapa *framework* yakni : *Nextjs*, *Sequelize*, *Exceljs*, *Nodejs* dan *React*. Alasan kami ingin menggunakan bahasa pemrograman dan *framework* tersebut dikarenakan adanya fleksibilitas yang baik dalam pengembangan seperti dengan banyaknya *library* yang digunakan dan mudah dalam digunakan serta dapat menghasilkan bagian *back end* yang mudah diimplementasikan maupun *front end* dan memiliki desain yang responsive.

Setelah semua persiapan dilakukan, tahap berikutnya adalah mengumpulkan, mengolah, dan menyusun data muzakki yang akan digunakan dalam bentuk file *extension* .xlsx (excel). Data tersebut terdiri dari nama muzakki, Nomor HP, Jenis Kelamin, Umur dan Pekerjaan. Proses ini dimulai dengan meminta kepada pihak LAZISMU DIY tentang data yang terkait, namun sayangnya dari pihak LAZISMU tidak bisa memberikan sepenuhnya data yang ada dikarenakan alasan privasi, sehingga kami harus membuat beberapa data *dummy* terutama pada bagian Nomor HP agar bisa diimplementasikan ke sistem. Data yang kami dapat di kelompokkan berdasarkan bulan dan tahun agar nantinya dapat dikategorikan juga berdasarkan waktu. Data yang dikumpulkan dipastikan kesesuaiannya dengan proyek. Selain itu, juga adanya pembuatan alur sistem untuk memudahkan penggunaan aplikasi setelah jadi. Rencana alur dibuat, termasuk bentuk pengguna dapat menggunakan sistem sesuai dengan yang sudah disepakati sebelumnya bersama dengan pihak LAZISMU DIY. Tahapan Pelaksanaan

3.2 Tahapan Pelaksanaan

Tahap Pelaksanaan akan dilakukan dari tanggal 1 Oktober hingga 31 Desember 2024, dengan praktik magang berlangsung selama 3 bulan. Pembuatan jadwal penyusunan aplikasi yang mencakup waktu dan tanggal untuk setiap kegiatan yang akan dilaksanakan mempertimbangkan waktu dari masing – masing anggota dan tingkat kesulitan. Koordinasi antar anggota dengan pihak

terkait, seperti manager maupun staff terkait, perlu dilakukan untuk memastikan kelancaran proses pembuatan sistem. Pastikan data yang didapat sudah sesuai dengan yang akan dikategorikan pada sistem, serta memastikan bahwa *logic* yang akan digunakan sudah tepat.

3.3 Tahapan Evaluasi

Tahap evaluasi pembuatan Sistem Kategorisasi Muzakki di LAZISMU DIY dimulai dengan menguji sistem dengan menggunakan pengujian *black box*. Pengujian ini dilakukan untuk menguji fungsionalitas pada perangkat lunak dan berfokus pada perspektif end – user. Pengujian ini nantinya akan dilakukan oleh calon pengguna langsung yakni staff dari LAZISMU DIY. Evaluasi dilakukan terhadap berbagai aspek seperti kemudahan navigasi, kejelasan informasi, responsivitas fitur interaktif, serta kemudahan dalam mengakses data. Tim menganalisis hasil pengujian untuk mengidentifikasi bagian yang perlu perbaikan atau pembaruan. Setelah melakukan penyesuaian berdasarkan hasil evaluasi, sistem akan diuji kembali untuk memastikan semua masalah yang ditemukan telah diselesaikan. Proses evaluasi ini dilakukan secara berkala untuk memastikan sistem dapat berjalan sesuai dengan fungsionalitas yang diharapkan. Selain itu, evaluasi juga dilakukan dengan berdiskusi secara langsung kepada manajer LAZISMU DIY dan staf terkait yang akan mengoperasikan sistem kategorisasi muzakki.

3.4 Rancangan Jadwal Kegiatan Magang

Rancangan Jadwal Kegiatan Magang dapat dilihat pada Tabel 3.1

Tabel 3.1 Rancangan Jadwal Kegiatan Magang

No.	Kegiatan	Minggu Pelaksanaan											
		1	2	3	4	5	6	7	8	9	10	11	12
1	Analisis kebutuhan Sistem Back -end												
2	Pembagian dan pemahaman jobdesc												
3	Menentukan framework dan bahasa pemrograman												

	yang akan digunakan												
4	Pengembangan backend lanjutan												
5	Integrasi end point API dengan database serta fungsi dasar untuk mengambil datanya												
6	Pengembangan fitur interaktif												
7	Penambahan validasi dan error handling di endpoint												
8	Penambahan Fitur autentikasi dan otorisasi												
9	Integrasi dengan front end												
10	Penyesuaian dan Perbaikan berdasarkan feedback dari pihak LAZISMU DIY												
11	Evaluasi Proyek Bersama dengan pihak LAZISMU DIY												
12	Mengurusi surat menyurat kegiatan magang												

Tabel 3.1 berisi rancangan jadwal kegiatan magang. Beberapa kegiatan yang tercantum antara lain analisis kebutuhan sistem , pembagian tugas dan penentuan bahasa pemrograman dan framework yang akan digunakan. Tabel 3.1 memberikan gambaran jelas tentang kegiatan magang akan berlangsung selama periode tertentu.

BAB IV

Pembahasan Pelaksanaan Praktik Magang

4.1 Hasil Tahapan Persiapan

Hasil dari tahap persiapan pembuatan Aplikasi Kategorisasi Muzakki bertujuan utama untuk membantu pendataan donasi berdasarkan dari Muzakki untuk mempermudah pendataan di Kantor Lazismu DIY agar tidak dilakukan secara manual. Salah satu persiapan dalam pembuatan aplikasi kategorisasi muzakki adalah menentukan tujuan dan sasaran, termasuk mengidentifikasi tujuan utama aplikasi dan menentukan sasaran pengguna. Setelah itu, dilakukan analisis kebutuhan dengan mengumpulkan informasi dari pengguna potensial, menentukan fitur-fitur yang diperlukan, dan membuat dokumen spesifikasi kebutuhan perangkat lunak (SRS). Selanjutnya, pemilihan logic yang tepat menjadi fokus utama, di mana berbagai algoritma yang relevan untuk menentukan yang paling cocok. Setelah itu, perencanaan pengembangan disusun, mencakup timeline pengembangan, sumber daya yang dibutuhkan, dan rencana proyek yang mencakup tahapan pengembangan, pengujian, dan peluncuran.

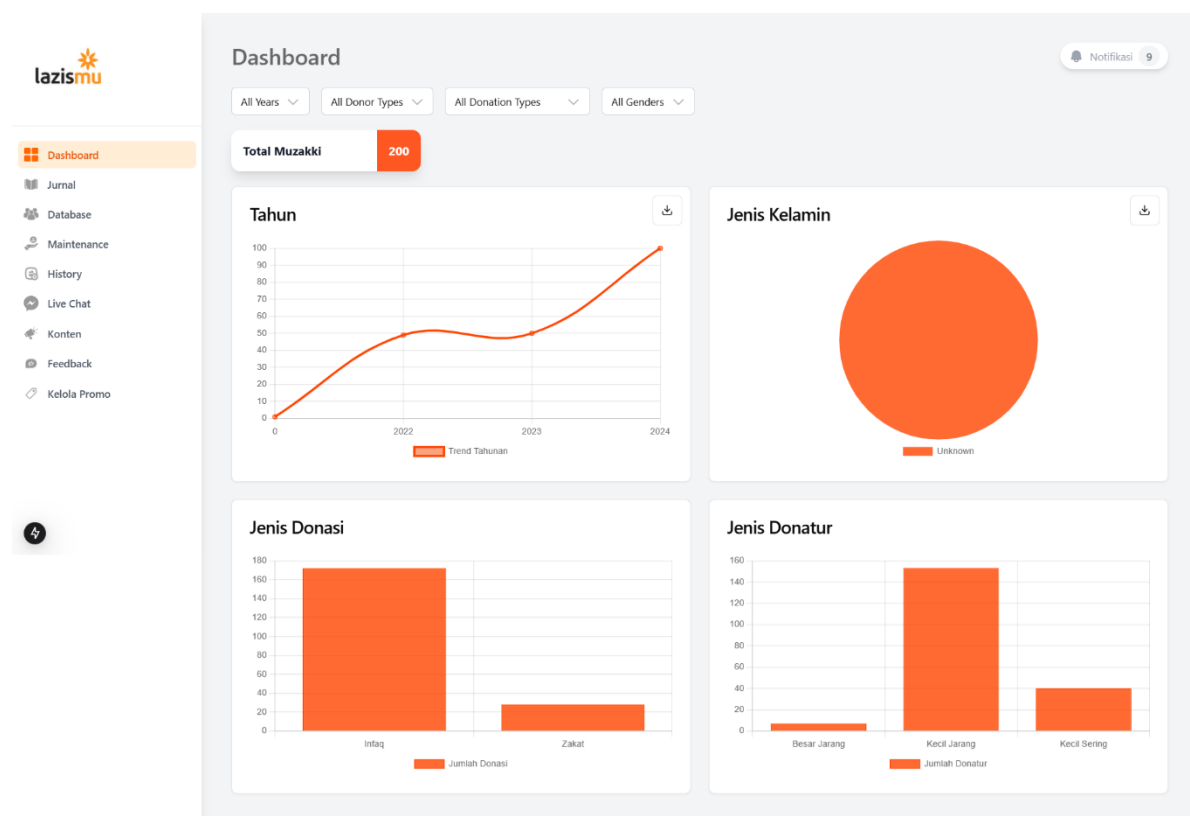
Pada tahap desain sistem, arsitektur aplikasi, termasuk backend dan frontend, dirancang. Wireframe dan mockup untuk antarmuka pengguna dibuat, serta database schema disusun untuk menyimpan data muzakki. Pemilihan teknologi yang tepat juga penting, dengan menentukan bahasa pemrograman, framework, dan database yang akan digunakan serta memastikan kesesuaiannya dengan kebutuhan proyek. Tahapan berikutnya adalah pengumpulan dan persiapan data muzakki yang akan digunakan untuk proses kategorisasi, termasuk pembersihan dan normalisasi data. Semua langkah ini dilakukan dengan cermat untuk memastikan aplikasi yang dihasilkan sesuai dengan kebutuhan pengguna dan berjalan dengan baik

Pengerjaan proyek dilakukan secara tim dengan anggota 3 kelompok yang masing masing terdiri dari : front end developer, back end developer dan system analyst. Pengerjaan proyek ini dikerjakan secara *work from home* (WFH) , dimana pengerjaan proyek dilakukan secara online tanpa melalui tatap muka langsung serta dilaksanakan dengan waktu yang fleksibel. Setiap anggota tim memiliki masing masing jenis pekerjaan yang dibagi sesuai dengan kapasitas dan posisi yang diemban di proyek ini , berikut merupakan pembagian tugas dari proyek ini.

No.	Nama Anggota Tim	Bagian	Tugas dan Tanggung Jawab
1.	Farid Hibaturrahman	Front End	a. Mendesain dan mengimplementasikan antarmuka pengguna yang responsif dan intuitif menggunakan React atau Next.js. b. Menggunakan Tailwind CSS untuk styling komponen. c. Mengintegrasikan API dari back end untuk menampilkan data muzzaki, jurnal, dan lainnya. d. Menangani permintaan HTTP (GET, POST, DELETE) untuk berinteraksi dengan server. e. Melakukan pengujian antarmuka pengguna untuk memastikan fungsionalitas dan pengalaman pengguna yang baik. f. Memperbaiki bug dan masalah yang ditemukan selama pengujian. g. Menulis dokumentasi untuk komponen front end dan panduan penggunaan aplikasi
2.	Naufal Anas ES Abiyyu	Back End	a. Menggunakan Node.js dan framework seperti Express untuk membangun server. b. Mendesain dan mengelola skema basis data untuk menyimpan data muzzaki dan jurnal. c. Menggunakan ORM (Object-Relational Mapping) seperti Sequelize untuk berinteraksi dengan basis data. d. Mengimplementasikan logika bisnis untuk mengklasifikasikan data donasi (besar, kecil, sering, jarang). e. Menangani validasi dan preprocessing data sebelum disimpan ke basis data. f. Melakukan pengujian unit dan integrasi untuk memastikan fungsionalitas API. g. Memperbaiki bug dan masalah yang ditemukan selama pengujian. h. Mengembangkan dan mengelola endpoint API untuk mengelola data muzakki, jurnal dan lainnya
3.	Dola Sri Rejeki	System Analyst	a. Mendesain arsitektur sistem yang mencakup front end, back end, dan basis data.

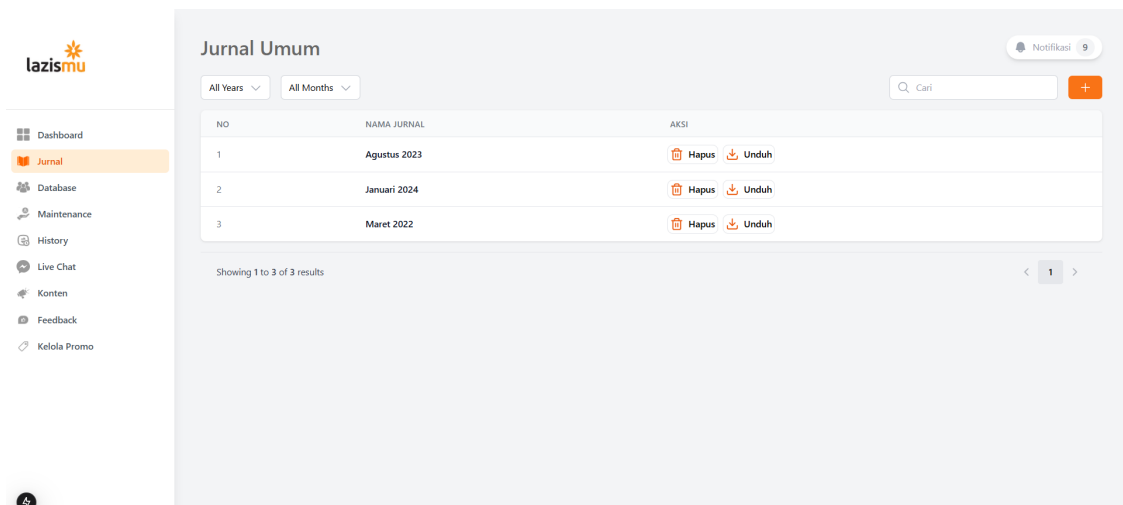
			b. Membuat diagram alur data, diagram ERD (Entity-Relationship Diagram), dan diagram arsitektur sistem. c. Berkoordinasi dengan tim front end dan back end untuk memastikan implementasi sesuai dengan desain. d. Menjadi penghubung antara tim teknis dan pemangku kepentingan bisnis. e. Melakukan validasi akhir terhadap sistem sebelum deployment.
--	--	--	---

4.2 Hasil Tahapan Pelaksanaan



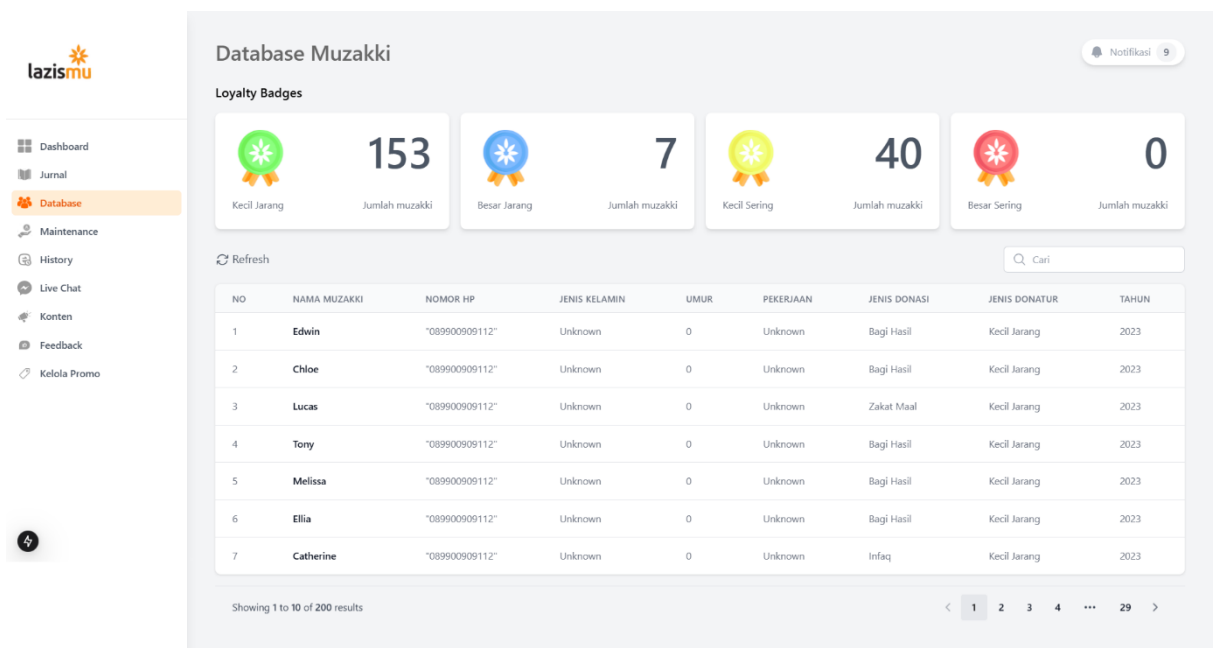
Gambar 4.2.1 Tampilan Menu Dashboard

Sistem berhasil dibuat sesuai dengan ketentuan yang sudah disetujui oleh pihak LAZISMU DIY, yakni untuk memberikan kategori pada setiap muzakki berdasarkan jumlah dan frekuensi donasi. Kategori tersebut terdiri dari 4 golongan muzakki yakni : a) Kecil Jarang, b) Besar Jarang, c) Kecil Sering, dan d) Besar Sering.



Gambar 4.2.2 Tampilan Menu Jurnal

Kategori yang ada tersebut dapat muncul apabila admin dari sistem menginput data dengan format file .xlsx yang isinya terdiri dari berbagai kolom yakni : a)Tanggal, b)Tahun, c)ZIS, d)VIA, e) Sumber dana, f>Nama, g)Donasi dan h)Nomor Hp. Sistem nantinya akan mendefinisikan kategori muzakki tersebut berdasarkan jumlah dan frekuensi donasi.



Gambar 4.2.3 Tampilan Menu Database

Muzakki dikelompokkan berdasarkan kriteria yang berbeda untuk penggolongan pada setiap jenis donasi yang terdiri dari Infaq dan Zakat. Perbedaan donasi Infaq dan Zakat terdiri dari jumlah

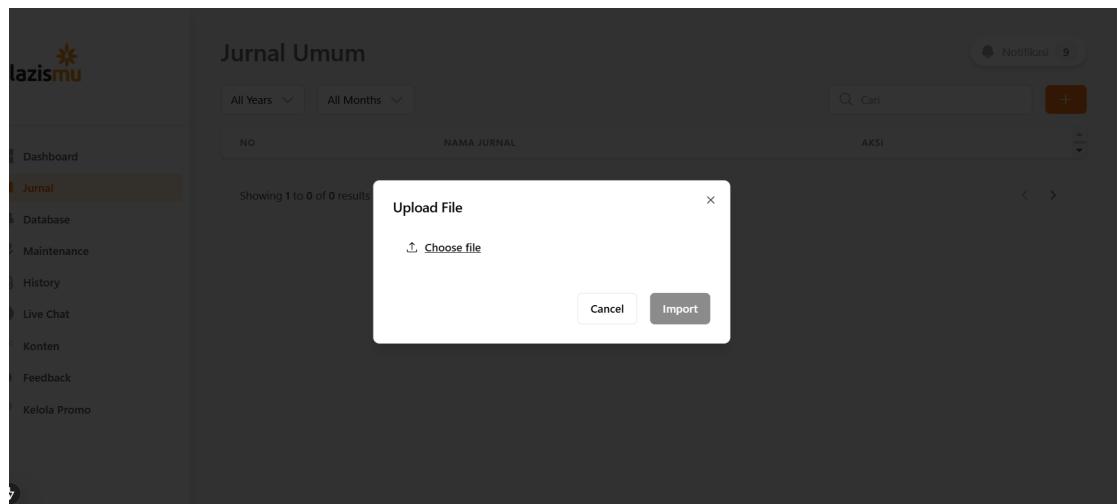
donasi dan frekuensi yang dilakukan dari setiap donasi tersebut. Kategori tersebut memiliki kategori yang dipisahkan oleh jumlah donasi dan frekuensi donasi. Jumlah donasi dikategorikan dengan Besar dan Kecil, sementara frekuensi donasi dikategorikan dengan Jarang dan Sering.

Seorang muzakki masuk kategori **jarang** apabila frekuensi Zakat dan Infaq yang dilakukan oleh muzakki dalam kurun satu tahun berjumlah dibawah 3 kali, sementara itu apabila seorang muzakki melakukan donasi lebih dari 3 kali dalam kurun waktu satu tahun , maka muzakki masuk kategori **sering**.

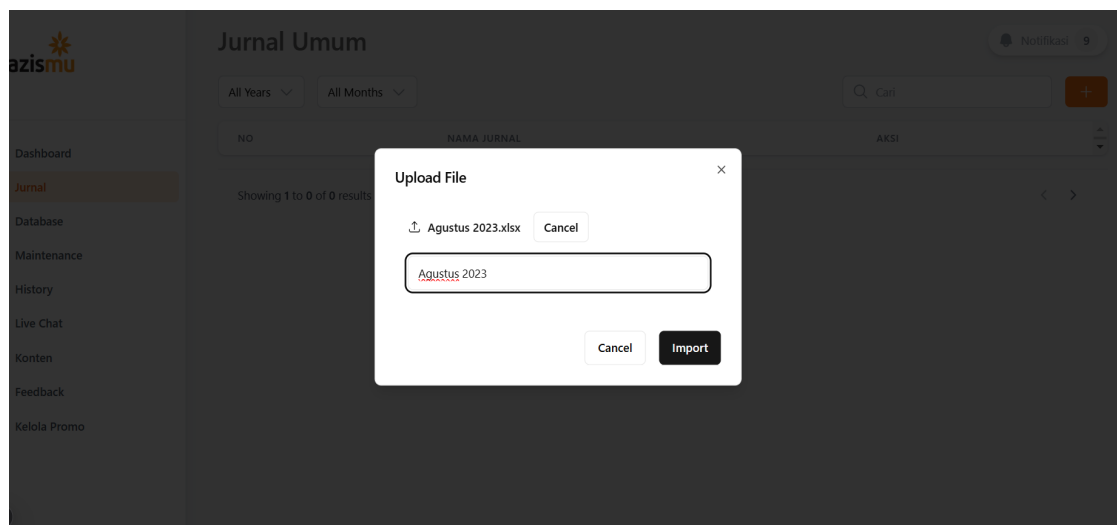
Muzakki juga dikategorikan berdasarkan besarnya jumlah donasi yang dilakukan, dalam hal ini terdapat adanya perbedaan antara besarnya jumlah donasi yang dilakukan antara donasi zakat dan infaq. Seorang muzakki dapat dikategorikan sebagai **besar** dalam donasi zakat apabila memberikan donasi lebih dari Rp. 1.000.000,00 (satu juta rupiah) dalam kurun waktu satu tahun , lalu untuk muzakki yang melakukan donasi zakat yang jumlahnya masih kurang dari yang ditentukan tersebut , maka dapat dikategorikan sebagai **kecil**. Kategori untuk besar infaq dapat didefinisikan dengan jumlah donasi dari muzakki dengan jumlah lebih dari Rp. 500.000,00 (lima ratus ribu rupiah) , maka dapat dikategorikan sebagai **besar**, apabila jumlahnya masih kurang dari angka tersebut maka dapat dikategorikan **kecil**.

Hasil Tahapan Pelaksanaan magang kali ini juga terdiri dari beberapa source code yang berfungsi untuk menyimpan , memproses dan mengirimkan data dari sistem kepada pengguna. Source code dibuat untuk menyediakan server API yang terdiri dari server side dan client side yang saling berkomunikasi dalam memproses data melalui HTTP request. API yang digunakan adalah RESTFUL API menggunakan metode HTTP yang terdiri dari GET, POST dan DELETE.

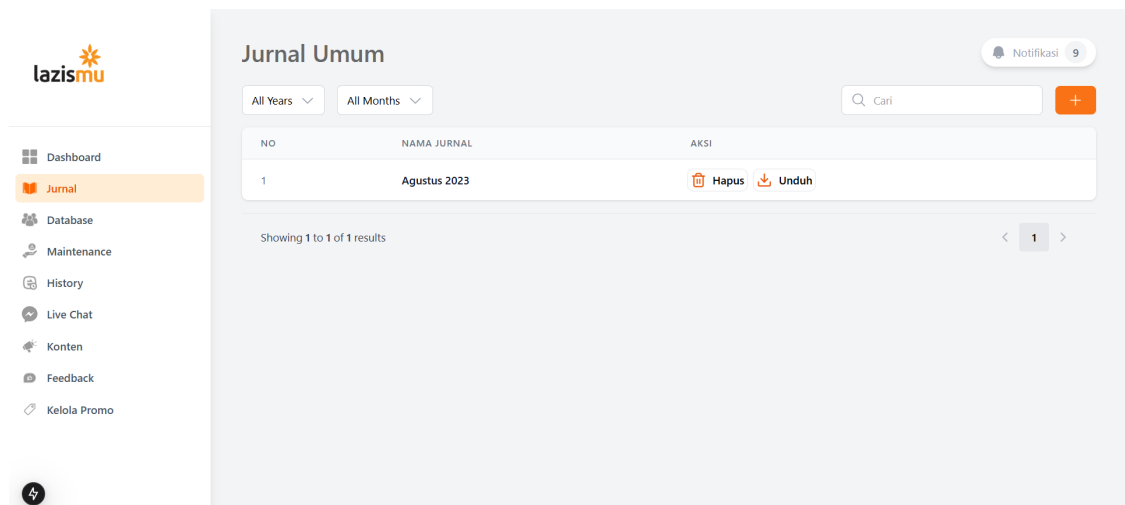
4.2.1 Fungsi untuk mengunggah data jurnal keuangan ke endpoint



Gambar 4.2.4 Tampilan Website untuk Mengunggah Data Excel



Gambar 4.2.5 Tampilan Website untuk Mengunggah Data Excel



Gambar 4.2.6 Tampilan Website untuk Mengunggah Data Excel

Fungsi yang terdapat pada Source code berlokasi di (src/api) ini mendefinisikan beberapa fungsi asinkron untuk berinteraksi dengan API RESTful. Fungsi **getMuzakki**, **getJurnal**, dan **getJurnalDataById** masing-masing digunakan untuk mengambil data muzakki, jurnal, dan jurnal berdasarkan ID dari server. Mereka menggunakan metode HTTP GET dan menangani kesalahan dengan mengembalikan array kosong jika terjadi kesalahan. Fungsi **deleteJurnal** digunakan untuk menghapus data jurnal berdasarkan ID dengan metode HTTP DELETE, dan mengembalikan true jika penghapusan berhasil atau false jika gagal.

Selain itu, tipe **MuzzakiJurnalUploadData** mendefinisikan struktur data untuk mengunggah jurnal, yang terdiri dari nama lampiran dan data lampiran dalam format base64. Fungsi **uploadJurnal** digunakan untuk mengunggah data jurnal ke server dengan metode HTTP POST. Fungsi ini mengembalikan true jika pengunggahan berhasil dan false jika gagal. Semua fungsi ini menggunakan **fetch** untuk melakukan permintaan HTTP dan menangani respons JSON dari server.

4.2.2 Fungsi untuk mendefinisikan sebuah handler API untuk framework Next.js.

```
import type { NextApiResponse, NextApiRequest } from 'next'

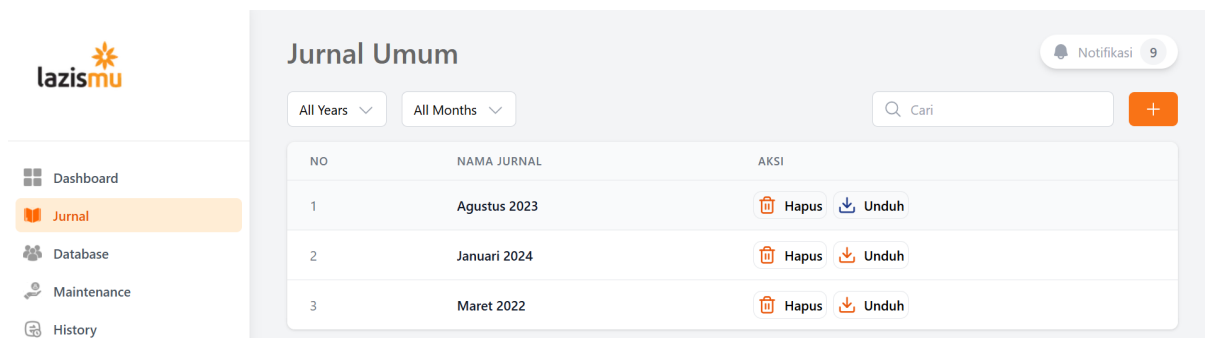
type ResponseData = {
  message: string
}

export default function GET(req: NextApiRequest, res:
NextApiResponse<ResponseData>) {
  res.status(200).json({ message: 'Hello from GET' })
}
```

Source code yang berlokasi di (src/api) ini mendefinisikan sebuah handler API untuk framework Next.js. Handler diimpor dari next dan menggunakan tipe NextApiRequest dan NextApiResponse untuk menentukan tipe parameter yang diterima oleh fungsi. Tipe ResponseData didefinisikan untuk menentukan bentuk data yang akan dikirimkan dalam respons API, yang dalam hal ini adalah sebuah objek dengan properti message bertipe string.

Fungsi handler API ini diekspor sebagai default dan menangani permintaan GET. Ketika fungsi ini dipanggil, ia menerima dua parameter: req (request) dan res (response). Fungsi ini kemudian mengirimkan respons dengan status 200 (OK) dan mengembalikan objek JSON yang berisi pesan Hello from GET. Ini adalah contoh sederhana dari endpoint API yang merespons permintaan GET dengan pesan statis.

4.2.3 Fungsi mendefinisikan handler API framework Next.js yang menangani permintaan GET untuk mengunduh data dalam format tertentu



The screenshot displays the 'Jurnal Umum' interface. At the top, there are filters for 'All Years' and 'All Months', a search bar labeled 'Cari', and a notification bell icon with '9' notifications. Below these is a table with the following data:

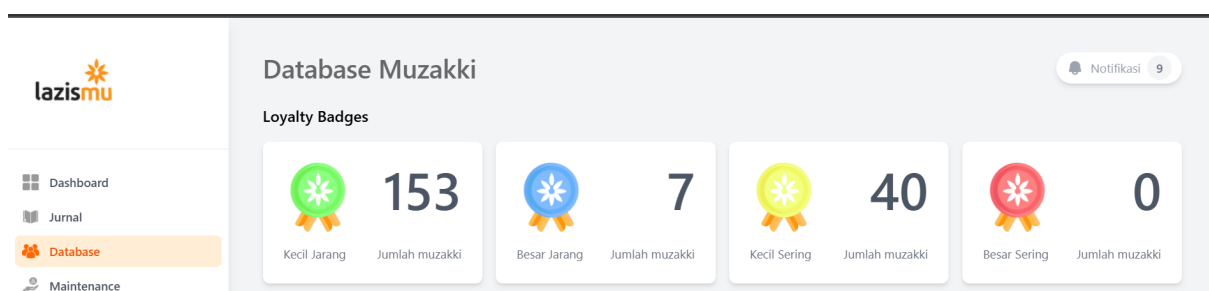
NO	NAMA JURNAL	AKSI
1	Agustus 2023	Hapus Unduh
2	Januari 2024	Hapus Unduh
3	Maret 2022	Hapus Unduh

Gambar 4.2.7 Tampilan untuk mengunduh jurnal dalam format xlsx.

Fungsi tersebut terdapat pada Source code yang berlokasi di (src/app/api/download) mendefinisikan mendefinisikan sebuah handler API untuk framework Next.js yang menangani permintaan GET untuk mengunduh data jurnal dalam format Excel. Fungsi ini pertama-tama mengambil parameter id dari URL permintaan. Jika parameter id tidak ada, fungsi mengembalikan respons dengan status 400 dan pesan error. Jika parameter id valid, fungsi mengambil data jurnal dan data terkait dari database menggunakan model Jurnal dan JurnalData. Data yang diambil kemudian diolah dan dimasukkan ke dalam workbook Excel menggunakan library excel js.

Setelah workbook Excel dibuat, fungsi ini menambahkan baris data ke worksheet dan mengatur gaya untuk header baris pertama. Data jurnal ditambahkan ke worksheet sebagai baris baru. Setelah semua data ditambahkan, workbook ditulis ke dalam buffer dan diubah menjadi blob dengan tipe MIME untuk file Excel. Fungsi kemudian mengembalikan respons dengan blob sebagai lampiran, memungkinkan pengguna untuk mengunduh file Excel yang berisi data jurnal. Respons ini mencakup header yang mengatur tipe konten dan nama file lampiran.

4.2.4 Fungsi untuk mendefinisikan metode untuk mengklasifikasikan data jurnal berdasarkan kategori dan frekuensi donasi



Gambar 4.2.8 Tampilan Klasifikasi data jurnal berdasarkan kategori dan frekuensi

Kategori "Besar":

Untuk donasi dengan kategori "Zakat", nominal donasi harus 1.000.000 atau lebih.

Untuk donasi dengan kategori "Infaq", nominal donasi harus 500.000 atau lebih.

Kategori "Sering":

Untuk donasi dengan kategori "Infaq", diperlukan minimal 3 donasi dalam satu bulan.

Untuk donasi dengan kategori "Zakat", diperlukan minimal 3 donasi dalam satu tahun.

Fungsi yang terletak di Source Code yang berlokasi di (src/app/api/jurnal) ini mendefinisikan sebuah kelas **DonationClassifier** yang digunakan mengklasifikasikan data donasi berdasarkan kategori dan frekuensi donasi. Kelas ini memiliki beberapa metode, termasuk **preprocess_data**, **kategori_muzaki**, **classify_frequency**, dan **classify**. Metode **preprocess_data** membersihkan data input dengan menghapus baris yang memiliki nilai null pada kolom tertentu, menghapus data duplikat, dan menambahkan kategori berdasarkan jenis donasi (zis). Metode **kategori_muzaki** menentukan kategori donatur (Besar atau Kecil) berdasarkan jumlah nominal donasi dan jenis donasi (Zakat atau Infaq).

Metode **classify_frequency** mengklasifikasikan data berdasarkan frekuensi donasi bulanan dan tahunan. Data diolah untuk menambahkan informasi tentang bulan dan tahun donasi, serta menghitung jumlah donasi bulanan dan tahunan untuk setiap donatur. Berdasarkan frekuensi ini, donatur diklasifikasikan sebagai Sering atau Jarang. Metode **classify** menggabungkan semua langkah ini dengan memanggil **preprocess_data** dan **classify_frequency** secara berurutan, dan mengembalikan data yang telah diklasifikasikan. Kelas ini secara keseluruhan bertujuan untuk memberikan label yang lebih bermakna pada data donasi.

4.2.5 Fungsi untuk mendefinisikan handler API framework Next.js yang menangani permintaan untuk menghapus data jurnal dan data terkait dari database.

Fungsi yang terdapat pada source Code yang berlokasi di (src/app/api/jurnal) berfungsi untuk mendefinisikan tiga handler API untuk framework Next.js yang menangani

permintaan GET, POST, dan DELETE terkait data jurnal. Fungsi GET mengambil data jurnal dari database berdasarkan parameter **id** yang diterima dalam URL. Jika **id** diberikan, fungsi mencari jurnal dengan **id** tersebut dan mengembalikan data jurnal beserta data terkait dari tabel **JurnalData**. Jika **id** tidak diberikan, fungsi mengembalikan semua data jurnal yang ada. Jika data jurnal tidak ditemukan, fungsi mengembalikan respons dengan status **error** dan pesan Data not found.

Fungsi POST menerima data dalam format Excel yang dikodekan dalam base64, memprosesnya, dan menyimpannya ke dalam database. Data Excel diubah menjadi array objek, kemudian diklasifikasikan menggunakan kelas **DonationClassifier** sebelum disimpan ke tabel **Jurnal** dan **JurnalData**. Fungsi DELETE menghapus data jurnal dan data terkait dari database berdasarkan parameter **id** yang diterima dalam URL. Jika **id** tidak valid atau data jurnal tidak ditemukan, fungsi mengembalikan respons dengan status **error** dan pesan yang sesuai. Ketiga fungsi ini bersama-sama memungkinkan pengelolaan data jurnal melalui operasi CRUD (Create, Read, Update, Delete) menggunakan API.

4.2.6 Fungsi untuk mendefinisikan sebuah handler API untuk framework Next.js yang menangani permintaan GET untuk mengambil data dari tabel JurnalData

Fungsi yang terletak pada source Code yang berlokasi di (`src/app/api/muzakki`) mendefinisikan sebuah handler API untuk framework Next.js yang menangani permintaan GET untuk mengambil data dari tabel **JurnalData** di database. Fungsi ini pertama-tama memastikan bahwa tabel **JurnalData** sudah sinkron dengan model yang didefinisikan dalam kode menggunakan **await JurnalData.sync()**. Setelah itu, fungsi mengambil semua data dari tabel **JurnalData** dengan **await JurnalData.findAll()**. Data yang diambil kemudian diiterasi dan setiap baris data diproses menjadi format yang lebih terstruktur. Beberapa atribut seperti **gender**, **age**, dan **occupation** diisi dengan nilai default karena tidak ada dalam data asli.

Setelah semua data diproses, hasilnya disimpan dalam array **actual_data**. Fungsi kemudian mengembalikan respons dengan status success dan data yang telah diproses dalam format JSON. Respons ini mencakup header yang mengatur tipe konten sebagai application/json. Secara keseluruhan, fungsi ini bertujuan untuk mengambil data dari tabel **JurnalData**, memprosesnya ke dalam format yang lebih terstruktur, dan mengembalikannya sebagai respons JSON untuk permintaan GET.

4.3 Hasil Tahapan Evaluasi

Tahap Evaluasi Pembuatan Sistem kategorisasi berbasis Website dilaksanakan dengan mempresentasikan sistem secara langsung kepada pihak LAZISMU DIY. Presentasi ditujukan kepada manajer dari pihak LAZISMU , serta staff terkait yang nantinya akan menggunakan sistem tersebut.

Evaluasi dilakukan dari bebragai aspek seperti kemudahan penggunaan, kejelasan informasi, kemudahan navigasi. Evaluasi sistem dilaksanakan menggunakan pengujian blackbox dengan fokus pengujian pada fungsionalitas sistem yang langsung digunakan oleh admin maupun pihak yang nantinya akan menggunakan aplikasi tersebut. Berikut hasil dari pengujian blackbox pada bagian back end yang sudah dilakukan :

Pengujian End Point POST

No.	Test Case Description	Input	Expected Output	Actual Output	Status
1.	Mengunggah file Excel yang valid	File Xlsx. Dengan isi kolom berupa a)Tanggal, b)Tahun, c)ZIS, d)VIA, e) Sumber dana,	File excel terdefinisi dengan baik dan kolom yang ada akan muncul di database	File terupload dan terdeteksi di bagian jurnal	Valid

		f)Nama, g)Donasi dan h)Nomor Hp			
2.	Mengunggah file Excel yang tidak valid	File yang diupload menggunakan format selain .xlsx	File tidak akan terupload dan muncul peringatan file invalid	Muncul peringatan error dan file tidak terupload	Valid
3.	Mengunggah file Excel dengan header yang tidak lengkap	File Xlsx. Dengan isi header yang tidak sesuai dengan ketentuan	File excel tidak terdefinisi , sehingga akan muncul peringatan file tidak sesuai atau error	Muncul peringatan error dan file tidak terupload	Valid
4..	Mengunggah file Excel kosong	File .xlsx dengan isi kosong	File excel tidak terdefinisi , sehingga akan muncul peringatan file tidak sesuai atau error	File terupload namun tidak terdefinisi ke database	Valid

Pengujian End Point GET

No.	Test Case Description	Input	Expected Output	Actual Output	Status
1.	Mendapatkan data jurnal dengan ID yang valid	Id jurnal valid sesuai dengan jumlah data dan header yang ada pada setiap jurnal	Id No. akan terdefinisi pada database dengan urutan	Database terdefinisi sesuai dengan header yang sudah ditentukan	Valid
2.	Mendapatkan data jurnal dengan ID yang tidak valid	Id jurnal invalid yang tidak sesuai dengan jumlah dan header	Id tidak akan mendefinisiikan apapun dan menjadi unknown	Muncul tulisan unknown pada database	Valid
3.	Mendapatkan semua data jurnal	Data sesuai format	Database terbentuk dan	Data tergabung	Valid

		kolom yang sudah diinput	berhasil mendefinsikan jurnal keuangan dan digabungkan menjadi satu kesatuan database	dalam satu table database	
--	--	--------------------------	---	---------------------------	--

Pengujian End Point Delete

No.	Test Case Description	Input	Expected Output	Actual Output	Status
1.	Menghapus data jurnal dengan ID yang valid	Id yang valid sesuai database	Data berhasil dihapus	Data Terhapus	Valid
2.	Menghapus data jurnal dengan ID yang tidak valid	Id yang tidak valid dari databse	Data tidak ditemukan dan tidak berhasil dihapus	Data tidak terhapus dan muncul error	Valid

Tim menerima masukan dari pihak LAZISMU bagian yang perlu diberi perbaikan atau pembaruan. Masukan tersebut berupa penyesuaian filter dan penambahan fitur agar dashboard bisa di download. Setelah melakukan penyesuaian berdasarkan hasil evaluasi, sistem clustering muzakki diuji kembali untuk memastikan semua masalah yang ditemukan telah diselesaikan. Proses evaluasi dilakukan sebelum aplikasi diserahkan kepada pihak LAZISMU , untuk memastikan bahwa sistem siap digunakan sepenuhnya dengan baik dan efisien, selain itu evaluasi juga dilakukan dengan memberikan tata cara penggunaan kepada staf – staf terkait.

Hasil dari kategorisasi muzakki berdasarkan hasil evaluasi secara keseluruhan baik itu pengujian secara blackbox maupun evaluasi dari pihak LAZISMU DIY memiliki hasil kategorisasi yang baik, diantaranya mencakup kategori dari muzakki nya yang sudah berhasil didefinsikan sesuai dengan ketentuan yang ada.

Kategori Berdasarkan Nominal Donasi

Besar:

- Untuk donasi dengan kategori "Zakat", nominal donasi harus 1.000.000 atau lebih.
- Untuk donasi dengan kategori "Infaq", nominal donasi harus 500.000 atau lebih.

Kecil:

- Untuk donasi dengan kategori "Zakat", nominal donasi kurang dari 1.000.000.
- Untuk donasi dengan kategori "Infaq", nominal donasi kurang dari 500.000.

2. Kategori Berdasarkan Frekuensi Donasi

Sering:

- Untuk donasi dengan kategori "Infaq", diperlukan minimal 3 donasi dalam satu bulan.
- Untuk donasi dengan kategori "Zakat", diperlukan minimal 3 donasi dalam satu tahun.

Jarang:

Jika jumlah donasi kurang dari 3 dalam periode yang ditentukan.

Jumlah data yang ada di database juga sudah sesuai dengan jumlah data yang diinputkan melalui jurnal keuangan dan dari jumlah yang ada di database juga sudah sesuai dengan jumlah muzakki yang dikelompokkan berdasarkan jumlah data yang ada di database sesuai dengan id yang ada.

4.4 Realisasi Jadwal Kegiatan Magang

No.	Kegiatan	Minggu Pelaksanaan												Progress
		1	2	3	4	5	6	7	8	9	10	11	12	
1	Analisis kebutuhan Sistem Back - end													Terlaksana (100 %)

2	Pembagian dan pemahaman jobdesc													Terlaksana (100%)
3	Menentukan framework dan bahasa pemrograman yang akan digunakan													Terlaksana (100%)
4	Pengembangan backend lanjutan													Terlaksana (100 %)
5	Integrasi end point API dengan database serta fungsi dasar untuk mengambil datanya													Terlaksana (100 %)
6	Pengembangan fitur interaktif													Terlaksana (100%)
7	Penambahan validasi dan error handling di endpoint													Terlaksana (100%)
8	Penambahan Fitur autentikasi dan otorisasi													Terlaksana (100%)
9	Integrasi dengan front end													Terlaksana (100%)
10	Penyesuaian dan Perbaikan berdasarkan feedback dari pihak LAZISMU DIY													Terlaksana (100%)
11	Evaluasi Proyek Bersama dengan pihak LAZISMU DIY													Terlaksana (100%)
12	Mengurusi surat menyurat kegiatan magang													Terlaksana (100%)

4.5 Kendala

Beberapa kendala yang ditemui selama pelaksanaan magang terdiri dari berbagai kendala, diantaranya seperti kurangnya data real yang diberikan dari pihak LAZISMU, kurangnya waktu untuk mengerjakan secara bersama – sama karena kesibukan masing – masing dan komunikasi dari pihak LAZISMU yang lebih sering dilakukan secara daring. Kendala tersebut tentunya sedikit menghambat proses pengerjaan magang yang seharusnya dapat selesai sebelum periode magang berakhir. Kurangnya data real menjadi salah satu penghambat dari kami untuk menyesuaikan fitur yang akan dibuat berdasarkan gambaran aplikasi yang sudah ada sebelumnya. Kesibukan dari setiap anggota juga menjadi salah satu penghambat dari pelaksanaan magang kali ini, dikarenakan front – end developer dan back – end developer memiliki kesibukan lainnya yakni KKN Alternatif yang dilaksanakan bersamaan dengan periode magang. Kurangnya komunikasi secara langsung dengan pihak LAZISMU DIY, juga menjadi salah satu faktor sedikit terlambat nya pelaksanaan magang ini selesai. Kendala – kendala tersebut berhasil kami lalui sampai saat ini, sehingga proyek yang diharapkan pun sudah bisa diserahkan kepada pihak LAZISMU DIY.

4.6 Solusi

Mengatasi kendala-kendala yang ditemui selama pelaksanaan magang dapat dilakukan melalui berbagai cara, seperti melakukan koordinasi intensif dengan pihak LAZISMU untuk memastikan data real tersedia tepat waktu, dan sementara waktu menggunakan data dummy untuk pengembangan fitur. Manajemen waktu yang baik juga penting dengan membuat jadwal kerja yang jelas. Komunikasi daring dapat dioptimalkan dengan menggunakan platform komunikasi yang efektif seperti Whatsapp Chat atau melakukan Video Call, serta mendokumentasikan setiap pertemuan daring dengan baik. Untuk mengatasi kesibukan anggota tim, prioritas tugas dan delegasi yang baik diperlukan, serta memberikan fleksibilitas waktu pengerjaan agar dapat menyesuaikan dengan kegiatan lain seperti KKN Alternatif. Terakhir, untuk mengatasi kurangnya komunikasi langsung, disarankan mengatur pertemuan tatap muka berkala jika memungkinkan, agar komunikasi dan koordinasi menjadi lebih terorganisir.

BAB V PENUTUP

5.1 Kesimpulan

Setelah melaksanakan praktik magang sebagai Back End Programmer di LAZISMU DIY, saya berhasil mengembangkan bagian back-end dari proyek ini yang mencakup beberapa aspek penting. Pertama, saya telah membuat dan mengintegrasikan endpoint API untuk mengambil data muzakki dan jurnal ke dalam database. Kedua, sistem back-end yang saya kembangkan telah meningkatkan efisiensi dan akurasi dalam pengelolaan data donatur dan transaksi, sehingga mengurangi beban administrasi manual. Ketiga, sistem ini mampu mengklasifikasikan data muzakki berdasarkan kategori tertentu, yang membantu dalam pengelolaan data yang lebih baik. Terakhir, teknologi yang digunakan dalam pengembangan back-end telah dipilih dengan tepat untuk memastikan skalabilitas dan maintainability sistem di masa depan.

5.2 Saran

Berdasarkan kesimpulan kegiatan magang yang telah dilakukan, ada beberapa saran yang dapat disampaikan untuk meningkatkan sistem. Pertama, optimasi kinerja API harus dipastikan bekerja dengan baik, termasuk caching data yang sering diakses dan mengurangi latensi jaringan. Kedua, keamanan data harus dijaga dengan memastikan data yang dikirim dan diterima dienkripsi menggunakan HTTPS. Terakhir, implementasikan mekanisme validasi dan error handling yang komprehensif untuk menangani berbagai jenis kesalahan dan memberikan respons yang informatif.

Lampiran

A. Surat Izin Praktik Magang Dari TU/Fakultas Jika Ada



FAKULTAS TEKNOLOGI INDUSTRI

Nomor : F2.1/84/D.33/IX/2024
Hal : Permohonan Izin Kerja Praktek

2 September 2024

Kepada

Yth. Kepala

LAZISMU DIY

Jl. Gedongkuning No.152, RT.41,
Rejowinangun, Kotagede, Yogyakarta, Daerah Istimewa Yogyakarta 55171

Dengan hormat,

Dalam rangka memberikan kesempatan kepada mahasiswa dalam mengimplementasikan disiplin ilmu yang diperoleh di bangku kuliah dengan keadaan sebenarnya, Pimpinan Fakultas Teknologi Industri mengajukan permohonan kepada Bapak/Ibu untuk berkenan memberikan izin kepada mahasiswa kami tersebut di bawah ini untuk melaksanakan **Kerja Praktek** di instansi yang Bapak/Ibu pimpin.

Adapun data mahasiswa sebagai berikut:

No	Nama	NIM
1.	Naufal Anas ES Abiyyu	2100018465

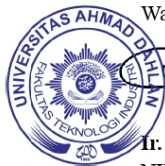
Program Studi : **S1 Informatika**
Fakultas : Teknologi Industri

Sebagai bahan pertimbangan Bapak/Ibu kami sampaikan bahwa sebagai konsekuensinya, mahasiswa yang bersangkutan bersedia memenuhi persyaratan administratif yang diperlukan.

Waktu Pelaksanaan Kerja Praktek Mulai Tanggal 1 Oktober 2024 - 31 Desember 2024.

Atas perhatian dan kerjasama Bapak/Ibu, kami sampaikan terima kasih.

Hormat kami,
a.n Dekan
Wakil Dekan



Ir. Sri Winiarti, S.T., M.Cs.
NIPM.19751216 200103 011 0880702

UAD Kampus IV (Utama)

Jalan Ahmad Yani, Tamanan, Kec. Banguntapan, Kab.
Bantul, Daerah Istimewa Yogyakarta 55191

Telp (0274) 563515
Email: fti@uad.ac.id

**B. Surat Keterangan Telah Menyelesaikan Praktik Magang/Sertifikat/Ttd Stampel
Instansi Di Halaman Persetujuan Instansi**

HALAMAN PERSETUJUAN INSTANSI

Laporan Praktik Magang ini telah disetujui sebagai proses pengajuan seminar bagi:

Nama	: Naufal Anas ES Abiyyu
NIM	: 2100018465
Tempat Praktik Magang	: Lazismu DIY
Realisasi Waktu Pelaksanaan	: 3 Bulan

Yogyakarta, 8 Januari 2025

Menyetujui,



Marzuki, S.E.

C. Logbook Sudah Terisi Minimal 7x

LoogBook UTS



LOG BOOK PRAKTIK MAGANG MAHASISWA
PROGRAM STUDI S1 INFORMATIKA T.A 2024 / 2025
(WAJIB DIISI DAN MASUK DALAM PENILAIAN)

Nim : 2100018465
 Nama Mahasiswa : Naufal Anas ES Abiyyu
 Judul Praktik Magang : Pengembangan Sistem Kategorisasi Muzakki untuk Optimalisasi Pengelolaan Donasi di Lazismu
 Dosen Pembimbing : Dinan Yulianto, S.T., M.Eng.
 Pembimbing Lapangan :

Petunjuk Pengisian Log Book

1. Log book di isi per minggu
2. Log book ditulis tangan
3. Setiap kegiatan di paraf oleh pembimbing lapangan/ dosen pembimbing Praktik Magang
4. Log book per minggu di paraf oleh dosen pengampu kelas Praktik Magang
5. Jumlah bimbingan minimal 4x

Logbook Minggu 3 sd 7 (sebelum UTS)

No	Kegiatan dan Lokasi Praktik Magang	Waktu Pelaksanaan		Hasil	Kendala, Rencana Perubahan (Jika ada)	Paraf Pembimbing Lapangan	Paraf Dosen Pembimbing Praktik Magang
		Hari/TGL	Jam Durasi				
1.	Observasi Proses kerja dan penyerahan tugas pengembangan <i>Customer Relationship Management (CRM)</i> , LAZISMU DIY	24/10/2024	13.00 – 15.00 (2 jam)	Mendapatkan pemahaman tentang alur kerja dan kebutuhan sistem CRM yang ingin dikembangkan oleh LAZISMU			
2.	Menguji fitur klusterisasi data Muzakki dengan menggunakan bahasa	08/10/2024	17.00 – 21.00 (4 jam)	Berhasil melakukan klusterisasi data untuk meningkatkan efisiensi	Kendala : model klusterisasi data yang dihasilkan belum		

	python (WFH)			pengkategorian muzakki	optimal dan juga data yang didapat dari LAZISMU belum sesuai dengan yang diharapkan Hasil : Hasil uji fitur klusterisasi berjalan dengan baik		
3.	Merealisasikan Desain UI pada bagian menu navbar bagian samping menggunakan kerangka kerja <i>next.js</i>	09/10/2024 – 23/10/2024	19.00 – 21.00 (2 jam setiap hari)	Berhasil membangun tampilan menu navbar bagian samping serta mengisi menu – menu nya dan isinya	Kendala : ada beberapa perbedaan dari desain UI yang tidak terlalu signifikan, sehingga tidak terlalu sesuai dengan yang diharapkan		
4.	Membuat grafik diagram batang dan diagram lingkaran berupa data yang nantinya akan di unggah di website untuk nantinya ditampilkan di dashboard	01/11/2024 – 05/11/2024	08.00 – 12.00	Berhasil membuat gambaran grafik diagram batang dan diagram lingkaran	Perubahan : kemungkinan nanti setelah bagian backend dibangun akan menggunakan diagram yang lebih atraktif		

CS Dipindai dengan CamScanner

Catatan Pembimbing Lapangan/Dosen Pembimbing Praktik Magang / Dosen Pengampu Kelas Praktik Magang:

.....

Dosen Pengampu Kelas Praktik Magang



(.....)

Yogyakarta, 15 November 2024

Mahasiswa



(NAUFAL ANAS ES KEMU)

CS Dipindai dengan CamScanner

LogBook UAS

LOG BOOK PRAKTIK MAGANG MAHASISWA
PROGRAM STUDI S1 INFORMATIKA T.A 2024 / 2025
(WAJIB DIISI DAN MASUK DALAM PENILAIAN)

Nim : 2100018465
 Nama Mahasiswa : Naufal Anas ES Abiyyu
 Judul Praktik Magang : Pengembangan Sistem Kategorisasi Muzakki untuk Optimalisasi Pengelolaan Donasi di Lazismu
 Dosen Pembimbing : Dinan Yulianto, S.T., M.Eng.
 Pembimbing Lapangan :

Petunjuk Pengisian Log Book

1. Log book di isi per minggu
2. Log book ditulis tangan
3. Setiap kegiatan di paraf oleh pembimbing lapangan/ dosen pembimbing Praktik Magang
4. Log book per minggu di paraf oleh dosen pengampu kelas Praktik Magang
5. Jumlah bimbingan minimal 4x

Logbook Minggu 8 sd 14 (sebelum UAS)

No	Kegiatan dan Lokasi Praktik Magang	Waktu Pelaksanaan		Hasil	Kendala, Rencana Perubahan (Jika ada)	Paraf Pembimbing Lapangan	Paraf Dosen Pembimbing Praktik Magang
		Hari/TGL	Jam Durasi				
1.	Mengembangkan Endpoint API untuk dikembangkan dan Pengembangan fitur interaktif,	11/11/2024 – 18/11/2024	2 jam setiap sesi	Endpoint API berhasil dikembangkan dan fitur interaktif juga berhasil dibuat	Tidak ada		Paraf: 
2.	Menambahkan validasi dan error handling di endpoint	20/11/2024 – 29/11/2024	2 jam setiap sesi	Validasi dan error handling berhasil ditambahkan untuk fitur tambah jurnal data	Tidak ada		Paraf: 

3.	Penambahan fitur otentikasi	1/12/2024	2 jam	Berhasil menambahkan fitur otentikasi	Tidak ada		Paraf: 
4.	Integrasi dengan front end	2/12/2024 – 12/12/2024	2 jam setiap sesi	Sudah terintegrasi dengan front end	Tidak ada		Paraf: 
5.	Finishing Proyek	13/12/2024 – 31/12/2024	1 jam setiap sesi	Proyek sudah siap untuk digunakan oleh pihak LAZISMU DIY	Tidak ada		Paraf: 
6.	Evaluasi bersama pihak LAZISMU DIY	3/1/2025 – 15/1/2025	30 menit setiap sesi	Proyek mendapatkan beberapa revisi dari pihak LAZISMU DIY	Revisi Sudah dikerjakan terutama pada penambahan beberapa fitur		Paraf: 

Catatan Pembimbing Lapangan/Dosen Pembimbing Praktik Magang / Dosen Pengampu Kelas Praktik Magang:

.....

.....

.....

.....

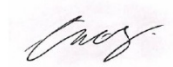
Yogyakarta, 15 Januari 2024

Dosen Pengampu Kelas Praktik Magang

Mahasiswa



(Bambang Rob'in, S.T., M.T.....)



(NAUFAL ANAS ES ABIYYU)

D. Source Code

```
database.ts (src/api)
const API_HOST = '';

export async function getMuzakki() {
  try {
    // Ganti nek wes di deploy
    const res = await fetch(API_HOST + '/api/muzzaki');
    if (!res.ok) throw new Error('Failed to fetch muzzaki data');
    const data = await res.json();
    return data.data;
  } catch (err) {
    console.error(err);
    return [];
  }
}

export async function getJurnal() {
  try {
    // Ganti nek wes di deploy
    const res = await fetch(API_HOST + '/api/jurnal');
    if (!res.ok) throw new Error('Failed to fetch jurnal data');
    const data = await res.json();
    return data.data;
  } catch (err) {
    console.error(err);
    return [];
  }
}

export async function getJurnalDataById(id: number) {
  try {
    const res = await fetch(API_HOST + '/api/jurnal?id=' + id);
    if (!res.ok) throw new Error('Failed to fetch jurnal data');
    const data = await res.json();
    return data.data;
  } catch (err) {
    console.error(err);
    return [];
  }
}

export async function deleteJurnal(id: number) {
  try {
    const res = await fetch(API_HOST + '/api/jurnal?id=' + id, {
```

```

        method: 'DELETE',
    });
    if (!res.ok) throw new Error('Failed to delete jurnal data');
    const data = await res.json();
    return data.status === 'success';
} catch (err) {
    console.error(err);
    return false;
}
}

export type MuzzakiJurnalUploadData = {
    attachment_name: string;
    attachment_base64: string;
}

export async function uploadJurnal(data: MuzzakiJurnalUploadData): Promise<boolean> {
    try {
        const res = await fetch(API_HOST + '/api/jurnal', {
            method: 'POST',
            headers: {
                'Content-Type': 'application/json',
            },
            body: JSON.stringify(data),
        });

        const res_data = await res.json();

        if (res_data.status === 'success') {
            return true;
        }

        return false;
    } catch (err) {
        console.error(err);

        return false;
    }
}

```

Route.ts (src/api)

```

import type { NextApiRequest, NextApiResponse } from 'next'

type ResponseData = {
    message: string
}

export default function GET(req: NextApiRequest, res: NextApiResponse<ResponseData>) {
    res.status(200).json({ message: 'Hello from GET' })
}

```

Route.ts (src/app/api/download)

```
"use server";

import { Jurnal, JurnalData } from "@db/db";
import * as exceljs from 'exceljs';

interface IJournal {
  id: number;
  name: string;
}

interface IJournalData {
  id: number;
  jurnal_id: number;
  nama: string;
  no_hp: string;
  tanggal: Date;
  tahun: number;
  zis: string;
  via: string;
  sumber_dana: string;
  nominal: number;
  jenis_donatur: string;
}

export async function GET(
  request: Request,
) {
  const params_id = new URL(request.url).searchParams.get('id');

  if (!params_id) {
    return new Response(JSON.stringify({
      status: 'error',
      message: 'Invalid parameter'
    }), {
      status: 400,
      headers: {
        'Content-Type': 'application/json'
      }
    });
  }

  let jurnal = await Jurnal.findOne({
    where: {
      id: params_id
    }
  }) as unknown as IJournal;

  if (!jurnal) {
    return new Response(JSON.stringify({
```

```

        status: 'error',
        message: 'Data not found'
    }}, {
        headers: {
            'Content-Type': 'application/json'
        }
    });
}

let jurnal_data = await JurnalData.findAll({
    where: {
        jurnal_id: params_id
    }
}) as unknown as IJournalData[];

let workbook = new exceljs.Workbook();
let worksheet = workbook.addWorksheet('Jurnal');

/**
 *      await JurnalData.create({
        jurnal_id: res_jurnal.id,
        nama: row['nama'],
        no_hp: row['no_hp'],
        tanggal: row['tanggal'],
        tahun: row['tahun'],
        zis: row['zis'],
        via: row['via'],
        sumber_dana: row['sumber_dana'],
        nominal: row['nominal'],
        jenis_donatur: row['jenis_donatur']
    });
 */

worksheet.columns = [
    { header: 'Nama', key: 'nama' },
    { header: 'No HP', key: 'no_hp' },
    { header: 'Tanggal', key: 'tanggal' },
    { header: 'Tahun', key: 'tahun' },
    { header: 'Zis', key: 'zis' },
    { header: 'Via', key: 'via' },
    { header: 'Sumber Dana', key: 'sumber_dana' },
    { header: 'Nominal', key: 'nominal' },
    { header: 'Jenis Donatur', key: 'jenis_donatur' },
];

worksheet.getRow(1).eachCell((cell) => {
    cell.fill = {
        type: 'pattern',
        pattern: 'solid',
        fgColor: { argb: 'FF000000' }, // Black color
    };
});

```

```

        cell.font = {
            color: { argb: 'FFFFFF' }, // White color
            bold: true
        };
    });

    jurnal_data.forEach(row => {
        worksheet.addRow({
            nama: row.nama,
            no_hp: row.no_hp,
            tanggal: row.tanggal,
            tahun: row.tahun,
            zis: row.zis,
            via: row.via,
            sumber_dana: row.sumber_dana,
            nominal: row.nominal,
            jenis_donatur: row.jenis_donatur,
        });
    });

    const buffer = await workbook.xlsx.writeBuffer();
    const blob = new Blob([buffer], { type: 'application/vnd.openxmlformats-officedocument.spreadsheetml.sheet' });

    return new Response(blob, {
        headers: {
            'Content-Type': 'application/vnd.openxmlformats-officedocument.spreadsheetml.sheet',
            'Content-Disposition': `attachment; filename="${jurnal.name}.xlsx"`
        }
    });
}

```

Classifikasi.ts (src/app/api/jurnal)

```

/**
 * Author: TODO
 * Commnet:
 */

export type KeyValue = {
    [key: string]: string | number | boolean | null;
}

export class DonationClassifier {
    preprocess_data(input: KeyValue[]) {
        const column_must_not_null = ['nama', 'zis', 'nominal', 'tanggal'];
        const data = [];
        for (let i = 0; i < input.length; i++) {
            const row = input[i];
            let is_valid = true;

            for (let j = 0; j < column_must_not_null.length; j++) {

```

```

        const field = column_must_not_null[j];
        if (!Object.keys(row).includes(field)) {
            is_valid = false;
            break;
        }
    }

    if (is_valid) {
        data.push(row);
    }
}

// drop duplicate data
const unique_data = [];
const unique_data_map: { [key: string]: boolean } = {};
for (let i = 0; i < data.length; i++) {
    const row = data[i];
    const key = `${row['nama'] ?? ''}${row['zis'] ?? ''}${row['nominal'] ?? ''}${row['tanggal'] ?? ''}`;
    if (!unique_data_map[key]) {
        unique_data.push(row);
        unique_data_map[key] = true;
    }
}

// add new kategori where zis == 'zakat' else 'infaq'
for (let i = 0; i < unique_data.length; i++) {
    const row = unique_data[i];
    if (row['zis'] == 'Zakat') {
        row['kategori'] = 'Zakat';
    } else {
        row['kategori'] = 'Infaq';
    }
}

return unique_data;
}

kategori_muzaki(row: KeyValue): string {
    if (row['kategori'] === 'Zakat') {
        const nominal = row['nominal'];
        if (typeof nominal === 'number' && nominal !== null) {
            return nominal >= 1000000 ? 'Besar' : 'Kecil';
        }
        return 'Tidak Diketahui';
    } else if (row['kategori'] === 'Infaq') {
        const nominal = row['nominal'];
        if (typeof nominal === 'number' && nominal !== null) {
            return nominal >= 500000 ? 'Besar' : 'Kecil';
        }
        return 'Tidak Diketahui';
    }
}

```

```

    }
    else {
        return 'Tidak Diketahui';
    }
}

classify_frequency(data: KeyValue[]): KeyValue[] {
    data.forEach(row => {
        row['c2'] = 'Jarang';
        const date = row['tanggal'] ? new Date(row['tanggal'] as string | number) : new Date();
        row['month'] = date.getMonth() + 1;
        row['year'] = date.getFullYear();
    });

    const donation_counts: { [key: string]: number } = {};
    data.forEach(row => {
        const key = `${row['nama']}-${row['kategori']}-${row['month']}-${row['year']}`;
        if (!donation_counts[key]) {
            donation_counts[key] = 0;
        }
        donation_counts[key]++;
    });

    data.forEach(row => {
        const key = `${row['nama']}-${row['kategori']}-${row['month']}-${row['year']}`;
        row['count'] = donation_counts[key];
    });

    data.forEach(row => {
        if (row['kategori'] === 'Infaq' && typeof row['count'] === 'number' && row['count'] >= 3) {
            row['c2'] = 'Sering';
        }
    });

    const yearly_counts: { [key: string]: number } = {};
    data.forEach(row => {
        const key = `${row['nama']}-${row['kategori']}-${row['year']}`;
        if (!yearly_counts[key]) {
            yearly_counts[key] = 0;
        }
        yearly_counts[key]++;
    });

    data.forEach(row => {
        const key = `${row['nama']}-${row['kategori']}-${row['year']}`;
        row['yearlyCount'] = yearly_counts[key];
    });

    data.forEach(row => {
        row['c1'] = this.kategori_muzaki(row);
    });
}

```

```

        if (row['kategori'] === 'Zakat' && typeof row['yearlyCount'] === 'number' &&
row['yearlyCount'] >= 3) {
            row['c2'] = 'Sering';
        }

        row['jenis_donatur'] = row['c1'] + ' ' + row['c2'];
    });

    return data;
}

classify(data: KeyValue[]): KeyValue[] {
    data = this.preprocess_data(data);
    data = this.classify_frequency(data);

    return data;
}
}

```

Route.ts (src/app/api/jurnal)

```

"use server";

import { Jurnal, JurnalData } from "@/db/db";
import * as exceljs from 'exceljs';
import { Buffer } from 'buffer';
import { DonationClassifier, KeyValue } from "../classifikasi";
import { JurnalRow } from "@/lib/types";

export async function GET(
    request: Request,
) {
    await Jurnal.sync();

    const params_id = new URL(request.url).searchParams.get('id');
    let res_jurnal: JurnalRow | JurnalRow[] | null = null;

    if (params_id) {
        const data = await Jurnal.findAll({
            where: {
                id: params_id
            },
            include: [JurnalData]
        });

        if (data.length === 0) {
            res_jurnal = null;
        } else {
            res_jurnal = data[0].get();
        }
    }
    else {

```

```

    const jurnalData = await Jurnal.findAll();
    res_jurnal = jurnalData.map(j => j.get()) as JurnalRow[];
  }

  if (res_jurnal == null) {
    return new Response(JSON.stringify({
      status: 'error',
      message: 'Data not found'
    })), {
      headers: {
        'Content-Type': 'application/json'
      }
    };
  }

  return new Response(JSON.stringify({
    status: 'success',
    data: res_jurnal
  })), {
    headers: {
      'Content-Type': 'application/json'
    }
  };
}

export async function POST(
  request: Request,
) {
  const body = await request.json();
  const { attachment_name, attachment_base64 } = body;
  let exceldata;

  try {
    exceldata = new exceljs.Workbook();
    const buffer = Buffer.from(attachment_base64, 'base64');
    const arrayBuffer = buffer.buffer.slice(buffer.byteOffset, buffer.byteOffset + buffer.byteLength);
    await exceldata.xlsx.load(arrayBuffer);
  } catch {
    return new Response(JSON.stringify({
      status: 'error',
      message: 'Invalid excel file'
    })), {
      headers: {
        'Content-Type': 'application/json'
      }
    };
  }

  if (exceldata.worksheets.length == 0) {
    return new Response(JSON.stringify({
      status: 'error',

```

```

        message: 'No worksheet found'
    }}, {
        headers: {
            'Content-Type': 'application/json'
        }
    });
}

try {
    const worksheet = exceldata.worksheets[0];
    const data = worksheet.getSheetValues();

    const expected_headers = ['tanggal', 'tahun', 'zis', 'via', 'sumber dana', 'nama', 'donasi', 'no'];
    // let is_header_missing = false;
    // const missing_headers: string[] = [];

    const header = data[1] as string[];
    const header_index: { [key: string]: number } = {};

    // Normalize and map header positions
    for (let i = 1; i <= header.length; i++) {
        let header_name = header[i];
        if (header_name == undefined) {
            continue;
        }

        header_name = header_name.toLowerCase().trim();
        header_index[header_name] = i;
    }

    // Check for missing headers and set to -1 in header_index if missing
    for (const expected_header of expected_headers) {
        if (!(expected_header in header_index)) {
            // is_header_missing = true;
            // missing_headers.push(expected_header);
            header_index[expected_header] = -1; // Set to -1 if header is missing
        }
    }

    const row_data: KeyValue[] = [];
    for (let i = 2; i < data.length; i++) {
        const data_iter = data[i] as string[];
        const tahun = header_index['tahun'] !== -1 ? parseInt(data_iter[header_index['tahun']]) : 0;
        const nominal = header_index['donasi'] !== -1 ? parseInt(data_iter[header_index['donasi']])
: 0;

        const no_hp_key = ['no hp', 'nomor hp', 'hp'].find(key => key in header_index) || '';
        const row: KeyValue = {
            'nama': header_index['nama'] !== -1 ? data_iter[header_index['nama']]?.trim() || " : ",
            'no_hp': header_index[no_hp_key] !== -1 ? data_iter[header_index[no_hp_key]] || " :
",

```

```

        ['tanggal']: header_index['tanggal'] !== -1 ? data_iter[header_index['tanggal']] || " : ",
        ['tahun']: tahun || 0,
        ['zis']: header_index['zis'] !== -1 ? data_iter[header_index['zis']]?.trim() || " : ",
        ['via']: header_index['via'] !== -1 ? data_iter[header_index['via']]?.trim() || " : ",
        ['sumber_dana']: header_index['sumber dana'] !== -1 ? data_iter[header_index['sumber
dana']]?.trim() || " : ",
        ['nominal']: nominal || 0
    };

    row_data.push(row)
}

const classifier = new DonationClassifier();
const classified_data = classifier.classify(row_data);

const res_jurnal = await Jurnal.create({
    'name': attachment_name
}) as unknown as JurnalRow;

// Insert to database
for (let i = 0; i < classified_data.length; i++) {
    const row = classified_data[i];
    await JurnalData.create({
        jurnal_id: res_jurnal.id,
        nama: row['nama'],
        no_hp: row['no_hp'],
        tanggal: row['tanggal'],
        tahun: row['tahun'],
        zis: row['zis'],
        via: row['via'],
        sumber_dana: row['sumber_dana'],
        nominal: row['nominal'],
        jenis_donatur: row['jenis_donatur']
    });
}

return new Response(JSON.stringify({
    status: 'success',
    data: {
        id: res_jurnal.id,
    }
}), {
    headers: {
        'Content-Type': 'application/json'
    }
});
} catch (error) {
    console.log(error);

    return new Response(JSON.stringify({
        status: 'error',

```

```

        message: 'Failed to upload data'
      }, {
        headers: {
          'Content-Type': 'application/json'
        }
      });
    }
  }
}

export async function DELETE(
  request: Request,
) {
  const params_id = new URL(request.url).searchParams.get('id');
  if (!params_id) {
    return new Response(JSON.stringify({
      status: 'error',
      message: 'Invalid parameter'
    }), {
      status: 400,
      headers: {
        'Content-Type': 'application/json'
      }
    });
  }

  try {
    const is_exist = await Jurnal.findOne({
      where: {
        id: params_id
      }
    });

    if (!is_exist) {
      return new Response(JSON.stringify({
        status: 'error',
        message: 'Data not found'
      }), {
        headers: {
          'Content-Type': 'application/json'
        }
      });
    }

    // delete jurnal, but with jurnaldata too
    await Jurnal.destroy({
      where: {
        id: params_id
      }
    });

    await JurnalData.destroy({

```

```

        where: {
            jurnal_id: params_id
        }
    });

    return new Response(JSON.stringify({
        status: 'success',
    }), {
        headers: {
            'Content-Type': 'application/json'
        }
    });
} catch (error) {
    console.error(error)

    return new Response(JSON.stringify({
        status: 'error',
        message: 'Failed to delete data'
    }), {
        status: 500,
        headers: {
            'Content-Type': 'application/json'
        }
    });
}
}
}

```

route.ts (src/app/api/muzakki)

```

"use server";

import { JurnalData } from "@/db/db";
import { JurnalDataRow } from "@/lib/types";

export async function GET() {
    await JurnalData.sync();

    const sql_res = await JurnalData.findAll();
    const actual_data = [];

    for (let i = 0; i < sql_res.length; i++) {
        const row = sql_res[i].get() as JurnalDataRow;

        const data = {
            id: row.id,
            name: row.nama,
            phoneNumber: row.no_hp,
            gender: 'Unknown',
            age: 0,
            occupation: 'Unknown',
            donationType: row.zis,
            donorType: row.jenis_donatur,

```

```

        status: 'Aktif',
        year: row.tahun,
    }

    actual_data.push(data);
}

return new Response(JSON.stringify({
    status: 'success',
    data: actual_data
})), {
    headers: {
        'Content-Type': 'application/json'
    }
}
};
}

```

E. Dokumentasi Kegiatan Magang



