

BAB I Pendahuluan

A. Latar Belakang

Management kegiatan yang baik dibutuhkan untuk mendukung kegiatan perkantoran dan kegiatan akademik. Salah satunya adalah management progress kegiatan kerja *staf*, sistem ini berfungsi sebagai sarana yang digunakan untuk mengetahui progres kerja para staf yang telah terselesaikan atau yang sedang berjalan. Sistem informasi management kerja yang baik dan sistematis akan sangat berpengaruh besar dalam menunjang kelancaran kegiatan pengembangan internal kampus yang nantinya diharapkan dapat meningkatkan kualitas informasi kampus.

Kebutuhan tentang progres kerja *staf* ini sangat penting untuk kemajuan kampus, mengingat bahwa banyak sekali tugas staf dosen yang harus dikerjar dan diselesaikan dengan cepat sesuai dengan target yang diberikan.

Sistem management kerja *staf* yang sudah ada di Universitas Ahmad Dahlan masih menggunakan cara manual, yakni Ketika suatu tugas atau job diberikan kepada *staf* maka pimpinan akan cek progres kerjanya secara manual dengan bertanya, sehingga hal ini tidak menjamin seberapa persen progres tugas yang di berikan dan hal ini menyebabkan pekerjaan tidak selesai dan tertumpuk. Sebagai suatu penawaran solusi terhadap permasalahan tersebut, dibuatlah sistem aplikasi web "**Project Management Tools**" untuk progres kerja mengatur dan monitoring kerja staf. Aplikasi ini diharapkan dapat membantu dan mempermudah pekerjaan dalam Biro Sistem Informasi.

B. Personel yang terlibat

NO	NAMA	NIM	PRODI	JOB
1	Brilly Jalu Kumara Biseka	1900018294	Informatika	Ketua

2	Naufal Luthfi Nurhamid	1900018282	Informatika	<i>Backend Developer</i>
3	Ryan Agung Nugraha	1900018285	Informatika	<i>UI/UX</i>
4	Ghaly Alif Dhaifullah	1900018289	Informatika	<i>Tester</i>
5	Hamzah Lopez Nurudin	1900018292	Informatika	<i>Data Analyst</i>
6	Muhammad Khalid H.	1900018293	Informatika	<i>Frontend Developer</i>
7	Anita Mega Budi Anggraini	1900018396	Informatika	Sekretaris, Bendahara, Data <i>Analyst 2</i>

Tabel 1. 1 Daftar Personel yang Terlibat

C. Proses Mendapatkan Proyek (*Lampiran i*).

Proyek ini dibuat dengan tujuan untuk mendukung sistem Internal di Biro Sistem Informasi guna menunjang informasi pada Universitas Ahmad Dahlan. Proyek dapat terealisasi karena adanya permintaan dari Bapak Ahmad Azhari, S.Kom., M.Eng. Melewati Bapak Ahmad Azhari, S.Kom., M.Eng. sebagai dosen pembimbing kami, yang telah menawarkan proyek ini kepada tim kami untuk membuat aplikasi management project yakni "*Project Management Tools*". Hasil yang akan dibuat adalah sebuah aplikasi berbasis mobile dan aplikasi berbasis web untuk mempermudah pengguna dalam pengoprasian aplikasi.

D. Deskripsi Perencanaan Proyek

Project Management Tools merupakan sebuah aplikasi web yang digunakan untuk mengetahui progress kerja para staf yang telah terselesaikan atau yang sedang berjalan dan direncanakan mulai awal Februari dan akan selesai pada bulan Agustus dengan pengerjaan kurang lebih 25 minggu.

BAB II Gambaran Mitra/*Clien*

A. Gambaran Umum

1. Sejarah

Biro Sistem Informasi (BSI) adalah salah satu layanan di Universitas Ahmad Dahlan dalam bidang informasi dan pengembangan *system* informasi.

Universitas Ahmad Dahlan (UAD) terus meningkatkan layanan berbasis *information Technology* (IT). Fungsi-fungsi organisasi utama yaitu manajemen informasi akademik, perpustakaan, SDM, manajemen aset, keuangan, dan gaji telah didukung sepenuhnya dengan sistem informasi untuk meningkatkan efisiensi dan efektifitas kerja. Layanan bagi *stakeholder* mulai dari pendaftaran mahasiswa baru, pembayaran kuliah, aktivitas perkuliahan, bimbingan akademik, wisuda sampai dengan layanan alumni juga telah berbasis IT.

Ada empat aplikasi baru yang diluncurkan tahun ini yaitu Sistem Informasi pengelolaan P2K daring (*online*), Sistem Informasi LSP (Lembaga Sertifikasi Profesi), Sistem Inventaris Artefak untuk media pengelolaan koleksi Museum Muhammadiyah dan Sistem Pembayaran menggunakan Virtual Account. Selain itu telah dilakukan penyiapan infrastruktur e-learning baru yang mendukung aktifitas pembelajaran yang diestimasikan mampu melayani 6400 *concurrent users* atau 32000 *browsing users*. Penggunaan sarana daring menjadi kebutuhan utama pada masa wabah Covid-19. Sehingga secara keseluruhan sudah ada 34 sistem informasi, atau 80% dari sistem yang sudah direncanakan dalam master plan IT, yang telah dibangun untuk mendukung layanan-layanan di Lingkungan UAD

UAD juga selalu berpartisipasi dalam pengembangan sistem informasi di tingkat Pimpinan Pusat Muhammadiyah. Saat ini telah selesai dikembangkan Sistem pembuatan Kartu Tanda Anggota Muhammadiyah Online (KTAM) yang mendukung proses pendaftaran dan pendataan keanggotaan secara daring. Sehingga memudahkan warga Muhammadiyah dalam proses pengurusan keanggotaan.

Pengembangan Sistem Informasi SDM dengan penilaian kinerja berbasis Catur Darma mulai dikembangkan pada bulan November tahun 2020. Pada bulan yang sama, juga mulai dikembangkan Sistem Informasi Penelitian Dan Pengabdian Masyarakat (Simlitabmas UAD) yang memfasilitasi proses penelitian, pengabdian masyarakat, publikasi ilmiah, HKI dan sistem reward publikasi. Untuk mempermudah pelayanan administrasi dan pengarsipan akan dikembangkan e-office yang dapat digunakan seluruh unit internal universitas.

Pengembangan *Learning Management System* (LMS) UAD yang terintegrasi dengan Sistem Informasi Akademik (SIA) untuk mendukung kemudahan layanan administrasi perkuliahan direncanakan akan dilakukan *soft-launching* pada awal semester Genap 2020/2021. *Bookless Library* dengan memanfaatkan teknologi QR Code sebagai identifikasi koleksi pustaka, akan dikembangkan dan diterapkan untuk mempermudah mahasiswa dalam mengakses pustaka digital yang dimiliki oleh perpustakaan UAD.

Data-data dalam seluruh sistem informasi yang ada di UAD diintegrasikan dan divisualisasikan dalam bentuk *Data Executive Summarization* (DES). DES ini digunakan untuk menyajikan rekapitulasi data Catur Darma Perguruan Tinggi yang meliputi AI

Islam Kemuhmadiyah, Pendidikan dan Pengajaran, Penelitian, Pengabdian Masyarakat dan info-info penting lainnya. DES ini ditampilkan dalam format visual, tabel dan grafik sehingga mempermudah pimpinan dalam mendapatkan informasi untuk mengambil keputusan. DES ini dikembangkan seiring dengan Layanan SI juga telah diarahkan menuju layanan terintegrasi dalam suatu portal sistem informasi dan layanan berbasis *mobile application*.

Pengelolaan layanan infrastruktur pendukung teknologi informasi dan komunikasi di lingkungan Universitas Ahmad Dahlan mengalami peningkatan yang signifikan. Penyediaan alokasi *bandwith* di semua lokasi Universitas Ahmad Dahlan sudah mencukupi sejumlah 1,7 Gbps. Rasio *bandwidth* per user saat ini mencapai 65 Mbps. Angka tersebut telah jauh melampaui standar yang ditetapkan Badan Akreditasi Nasional Perguruan Tinggi untuk layanan internet mahasiswa yang mensyaratkan *bandwith* minimum sebesar 2 Kbps per user.

Dalam hal infrastruktur server, UAD telah selesai membangun ruang *data center* di lantai 7 Gedung FK. Data center ini akan dilengkapi dengan server-server dengan teknologi terbaru (*dengan brand Cisco, Sangfor, NutaNix, Hyperflex dll*) dan dengan *storage* yang jauh lebih mumpuni.

2. Visi Misi

Visi dan misi Biro Sistem Informasi :

Visi :

Menjadi biro penyelenggara layanan teknologi informasi komunikasi (TIK) yang handal dalam mendukung pencapaian Sasaran mutu UAD.

Misi :

- a. Menyelenggarakan layanan TIK dengan tingkat kematangan yang meningkat secara berkelanjutan.
- b. Menyediakan layanan sistem informasi yang handal dan terintegrasi dalam mendukung kelancaran proses akademik.
- c. Menyelenggarakan layanan jaringan komunikasi dan internet dengan SLA 99%.
- d. Menyelenggarakan layanan helpdesk dengan waktu respon keluhan maksimum 2 hari.
- e. Mengelola website dan sosial media dalam mendukung pencitraan universitas serta seluruh unit di UAD.

Meningkatkan Kerjasama bidang TIK dengan pihak eksternal dalam rangka meningkatkan daya saing UAD

3. Alamat .

Biro sistem informasi sendiri beralamatkan:

Gedung kedokteran lantai 7 Jalan Ahmad Yani (Ringroad Selatan)
Tamanan, Banguntapan, Bantul, Yogyakarta 55166.

Telepon : (0274) 563515, 511830, 379418, 371120

Faximille : 0274-564604

Email : bsi@uad.ac.id

B. Struktur Organisasi Sumber Daya Manusia dan Sumber Daya Fisik lainnya

1. Pimpinan Biro Sistem Informasi

NO	NAMA	JABATAN
1	Fiftin Noviyanto, S.T., M.Cs.	Kepala Biro Sistem Informasi

2	Sugriyono, M.Kom	Kepala Bidang Perencanaam Sistem Informasi
3	Supriyanto, S.T., M.T.	Kepala Bidang Pengembangan Sistem Informasi
4	Ahmad Azhari, S.Kom., M.Eng.	Kepala Bidang Pengoperasian Sistem Informasi

Tabel 2. 1 Daftar Pimpinan Biro Sistem Informasi UAD

2. Kepala Urusan Biro Sistem Informasi

NO	NAMA	JABATAN
1	Halim Iswan, S.Kom	Kepala Urusan Arsitektur Sistem
2	Muhammad Akhsan, S.Kom	Kepala Urusan Pengembangan Aplikasi dan Web
3	Hamam Sulistyو, S.T	Kepala Urusan Pengelolaan Infrastruktur Jaringan
4	Wahyu Prio Wicaksono, S.Kom	Kepala Urusan Pemeliharaan Sistem
5	Andi Hermanto, S.Kom	Kepala Urusan Layanan Pengguna

Tabel 2. 2 Daftar Kepala Urusan Biro Sistem Informasi UAD

3. Developers

NO	NAMA	JABATAN
1	Sugriyono, M.Kom	<i>Developer</i>
2	Muhammad Akhsan, S.Kom	<i>Developer</i>
3	Farid Ahmad, S.Kom	<i>Developer</i>

4	Halim Iswan, S.Kom	<i>Developer</i>
5	Ahmad Luthfi, S.Kom	<i>Developer</i>
6	Andi Hermanto, S.Kom	<i>Developer</i>
7	Meiliana Sumagita, S.Kom	<i>Developer</i>

Tabel 2. 3 Daftar Developers Biro Sistem Informasi UAD

4. *Operations*

NO	NAMA	JABATAN
1	Hamam Sulisty, S.T	<i>Operations</i>
2	Wahyu Prio Wicaksono, S.Kom	<i>Operations</i>
3	Eddy Irawan Aristianto	<i>Operations</i>
4	Santoso	<i>Operations</i>
5	Dwi Julianto	<i>Operations</i>
6	Denny Supriyanto	<i>Operations</i>
7	Febyan Reza Aditya	<i>Operations</i>
8	Ruba Waluyo	<i>Operations</i>
9	Yusuf Ananda	<i>Operations</i>
10	Tegar Abdullah	<i>Operations</i>
11	Hefry Ramadhani	<i>Operations</i>

Tabel 2. 4 Daftar Operations Biro Sistem Informasi UAD

5. *Administration*

NO	NAMA	JABATAN
1	Hapsari Utami	<i>Administration</i>

Tabel 2. 5 Daftar Administration Biro Sistem Informasi UAD

C. Gambaran Sistem atau Proses Bisnis pada Mitra/*Clien* saat ini

1. *Management Akademik*

Merancang untuk kebutuhan pengelolaan data-data akademik supaya berbasis teknologi komputer, yang mana memudahkan seluruh proses kegiatan akademik dapat terkelola menjadi informasi yang bermanfaat dalam pengelolaan manajemen perguruan tinggi.

2. *Development*

BSI memiliki proses bisnis seperti pemberian project dan kerja sama untuk pembuatan aplikasi atau sistem guna membangun BSI.

3. *Layanan Informasi*

Layanan operasional Informasi juga Panduan untuk membantu kegiatan civitas akademika Universitas Ahmad Dahlan.

BAB III Rencana Pembagian Tugas

A. Rencana Pembagian Tugas

Deskripsi mengenai pembagian tugas dalam pembuatan Aplikasi Project Management Tools sebagai berikut :

NO	JOBDESK	NAMA	KETERANGAN
1.	Ketua	Brilly Jalu Kumara Biseka	<ul style="list-style-type: none">· Menentukan tim manajemen tugas proyek· Membagi jobdesk anggota tim· Mengkoordinasi dan mengatur waktu bimbingan· Merencanakan, melaksanakan, mengawasi, dan mengevaluasi pelaksanaan proyek· Mengkoordinasi dan memimpin setiap pertemuan atau rapat

2.	Backend Developer	Naufal Luthfi Nurhamid	<ul style="list-style-type: none"> · Pengelolaan data project meliputi CRUD (Admin) · Pengelolaan data team (admin) · Transaksi tambah project : <ul style="list-style-type: none"> o API register dan login o API mengelola requirement (CRUD) o API mengelola internal briefing dari setiap project (CRUD) o API mengelola developing dari setiap project (CRUD) o API mengelola testig dari setiap project (CRUD) · Penugasan suatu project kepada team · API mengelola team
3.	UI/UX	Ryan Agung Nugraha	<ul style="list-style-type: none"> · Pembuatan tampilan UI dengan memperhatikan UX · Menentukan layout atau tata letak dari komponen system yang dibangun · Penyusunan rancangan UI pada laporan

4.	Tester	Ghaly Alif Dhaifullah	<ul style="list-style-type: none"> · Melakukan testing berkala pada aplikasi
5.	Data Analyst 1	Hamzah Lopez Nurudin	<ul style="list-style-type: none"> · Menganalisa proyek yang berkaitan dengan perancangan system · Membuat proses bisnis diagram · Membuat ERD (<i>Entity Relationship Diagram</i>) · Membuat <i>Use Case Diagram</i> · Membuat <i>Activity Diagram</i>
6.	Data Analyst 2	Anita Mega Budi Anggraini	<ul style="list-style-type: none"> · Menganalisa proyek yang berkaitan dengan perancangan system · Membuat <i>Activity Diagram (Login, menu Admin)</i>
7.	Frontend Developer	Muhammad Khalid H.	<ul style="list-style-type: none"> · Mengimplementasikan rancangan tampilan yang telah dibuat oleh UI/UX · Membuat validasi input data · Mengintegrasikan backend menggunakan rest api

8.	Sekretaris	Anita Mega Budi Anggraini	<ul style="list-style-type: none"> · Menyusun proposal penawaran proyek · Menyusun surat perjanjian / MOU · Menyiapkan logbook · Notulensi rapat tim · Pembuatan laporan pengerjaan proyek
9.	Bendahara	Anita Mega Budi Anggraini	<ul style="list-style-type: none"> · Pembuatan rincian biaya · Penyusunan pengeluaran biaya

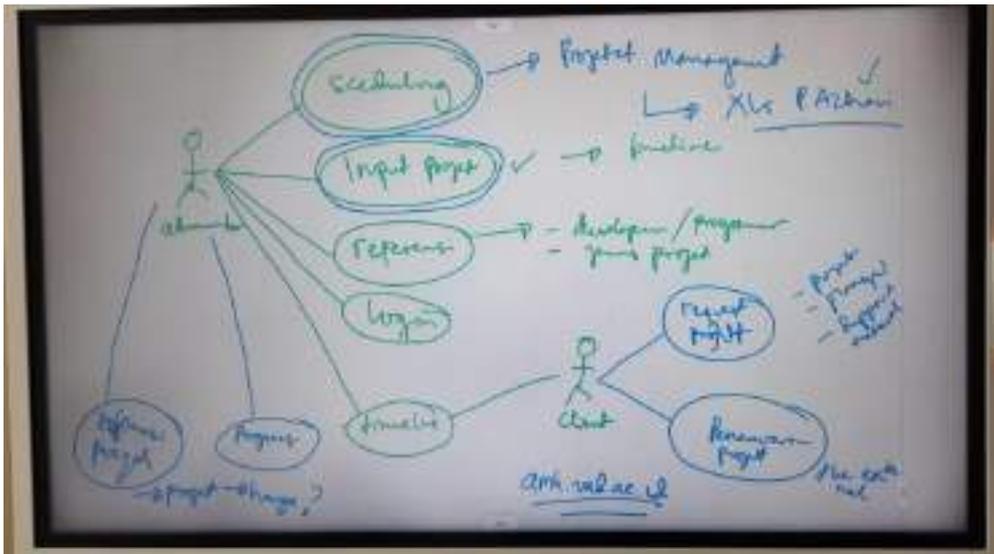
Tabel 3. 1 Daftar Rencana Pembagian Tugas

B. Perancangan Sistem

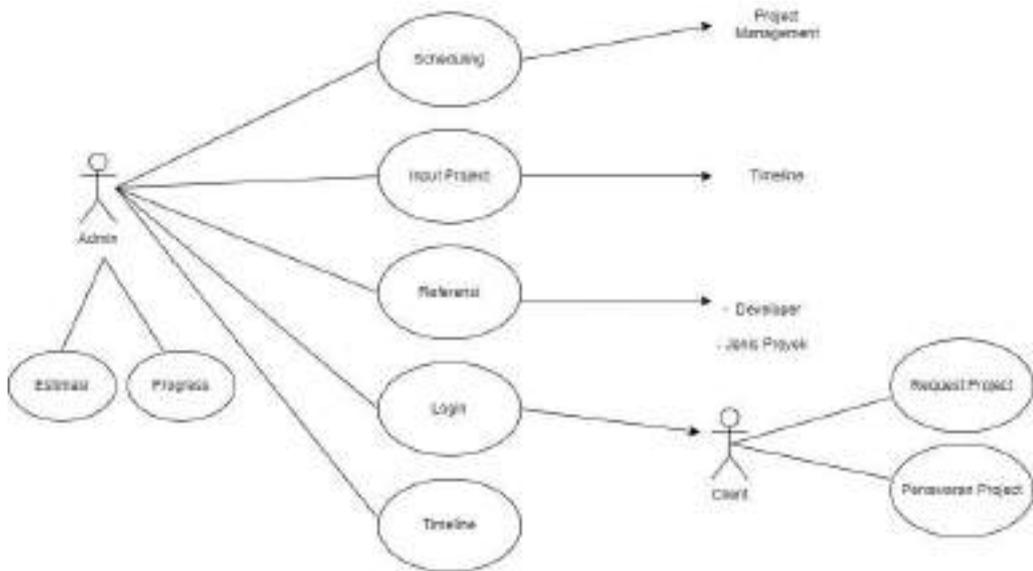
1. Analisis Kebutuhan Sistem

Proses analisis dilakukan dengan cara observasi atau pengamatan langsung terhadap apa yang terjadi untuk mengetahui informasi mengenai alur Biro Sistem Informasi .

Terdapat beberapa kegiatan yaitu dapat dilihat pada gambar :



Gambar 3. 1 Analisis Kebutuhan Sistem



Gambar 3. 2 Hasil Analisis Kebutuhan Sistem yang Akan Dibangun

a. Kebutuhan Fungsional

No	Kode	Deskripsi
1	SKPL-F1	Sistem menampilkan halaman dalam bentuk web
2	SKPL-F2	User bisa mengakses web dengan akunnya dan Registrasi
3	SKPL-F3	Sistem Memiliki Fitur Dashboard
4	SKPL-F4	Fitur user admin untuk memberi akses lebih dalam memanipulasi pada sistem
5	SKPL-F5	Sistem menampilkan Progress Project
6	SKPL-F6	Sistem memberi akses admin untuk menu pembuatan akun user
7	SKPL-F7	Sistem menyediakan fitur internal briefing untuk memuat task dan prioritynya
8	SKPL-F8	Sistem menyediakan fitur Developing untuk pic memberi detail task yang diberikan ke user
9	SKPL-F9	Sistem menyediakan fitur Developing untuk pic/admin memberi detail task yang diberikan ke user
10	SKPL-F10	Sistem menyediakan fitur Functional Test untuk pic / admin menguji fungsional dari aplikasi
11	SKPL-F11	Sistem menyediakan fitur review internal/eksternal untuk pic / admin menentukan status tiap task/requirement
12	SKPL-F12	Sistem menyediakan fitur Security Test untuk pic / admin menguji security dari aplikasi
13	SKPL-F13	Sistem menyediakan fitur upload dokumentasi
14	SKPL-F14	Sistem menyediakan fitur released
15	SKPL-F15	Sistem menyediakan akses client melihat aktivitas progress

Tabel 3. 2 Daftar Kebutuhan Fungsional

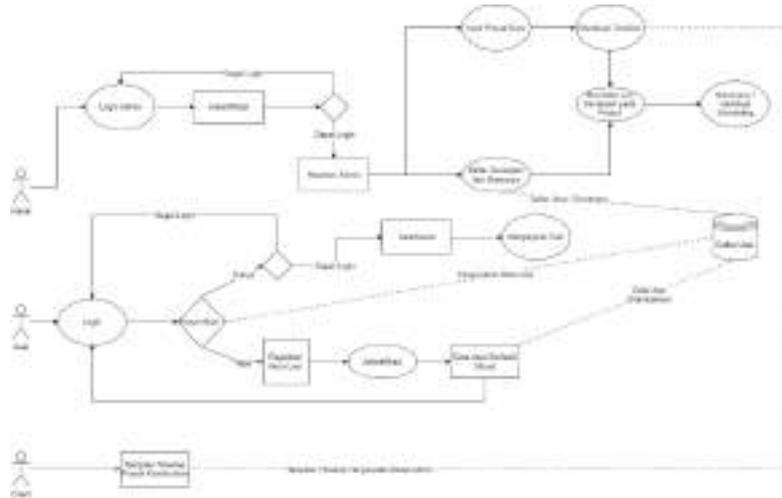
b. Kebutuhan Nonfungsional

No	Kode	Parameter	Deskripsi
1	SKPL-F1	Availability	Disediakan aplikasi yang siap dipakai untuk dapat diakses oleh client
2	SKPL-F2	Reliability	User bisa mengakses web dengan akunnya dan Registrasi
3	SKPL-F3	Ergonomy	Sistem Memiliki Fitur Dashboard memudahkan penggunaan
4	SKPL-F4	Response time	Sistem harus mampu mengolah project data yang melakukan request yang banyak
5	SKPL-F5	Portability	Sistem bisa di akses di berbagai device agar memudahkan user
6	SKPL-F6	Safety	Sistem memberi akses admin untuk menu pembuatan akun user
7	SKPL-F7	Security	Sistem harus menyertakan sebuah prosedur otorisasi dimana pengguna nya harus mengidentifikasi diri dengan sebuah username / email dan password

Tabel 3. 3 Daftar Kebutuhan Nonfungsional

2. Proses Bisnis Diagram

a. Use Case Diagram

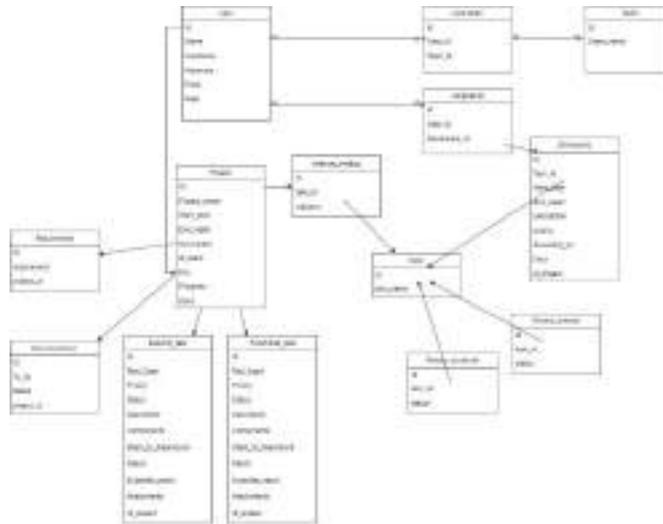


Gambar 3. 3 Use Case Diagram

Use case diagram merupakan aliran kegiatan dan proses bisnis yang dilakukan pengguna (Pratama 2014).

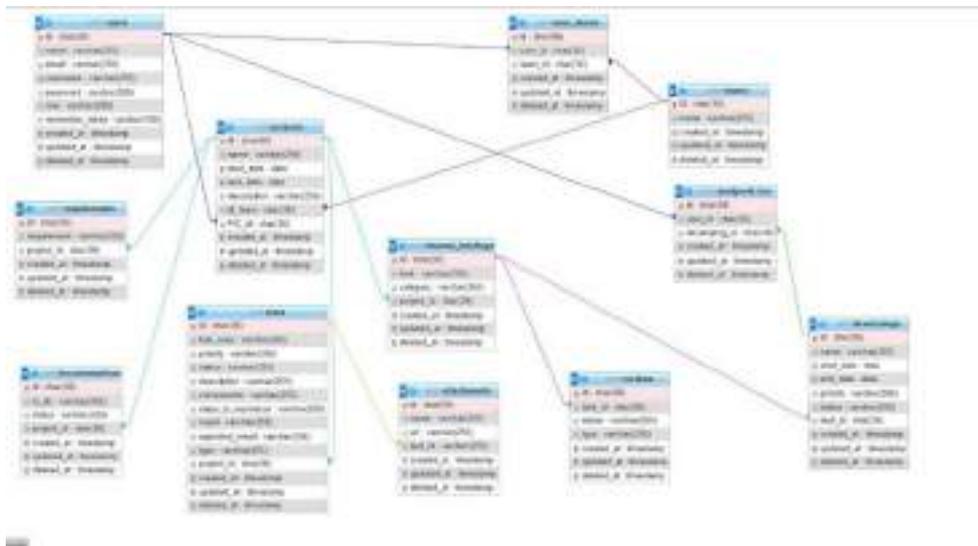
- 1) Admin login untuk mendapatkan autentikasi akun
- 2) Sistem akan menampilkan halaman login admin
- 3) Admin mengontrol dan cek daftar developer dan status pengerjaannya
- 4) Admin dapat menginput proyek baru, membuat *timeline*, mendaftarkan *developer*, membuat *scheduling*
- 5) Lalu *user* juga *login* untuk masuk, jika belum memiliki akun maka dapat mendaftar terlebih dahulu.
- 6) Sistem menampilkan halaman *dashboard user* untuk mengerjakan *task* atau tugasnya
- 7) *Client* masuk untuk melihat *timeline* keseluruhan proyek .

b. Entity Realitionship Diagram



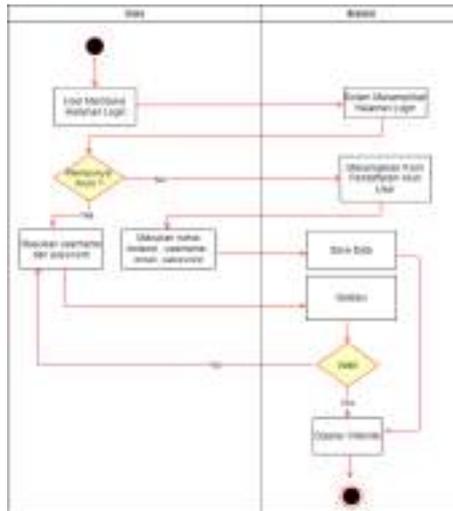
Gambar 3. 4 Entity Realitionship Diagram

c. Database Final



Gambar 3. 5 Rancangan Database

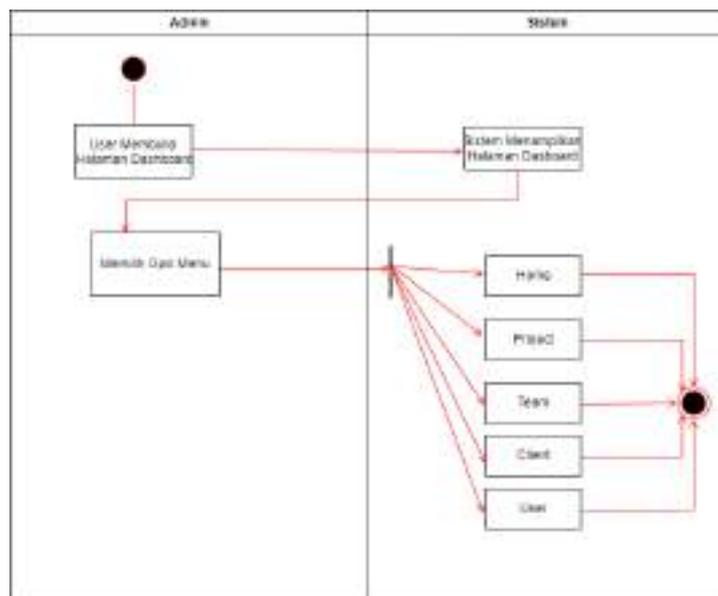
d. Activity Diagram Login



Gambar 3. 6 Activity Diagram Login

Activity diagram menggambarkan workflow (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Activity diagram di atas menjelaskan alur kerja atau kegiatan user untuk menuju ke halaman utama yang didahului dengan login / input username dan password, dan jika tidak memiliki akun sebelumnya maka akan diarahkan ke fitur register.

e. Activity Diagram Menu Admin

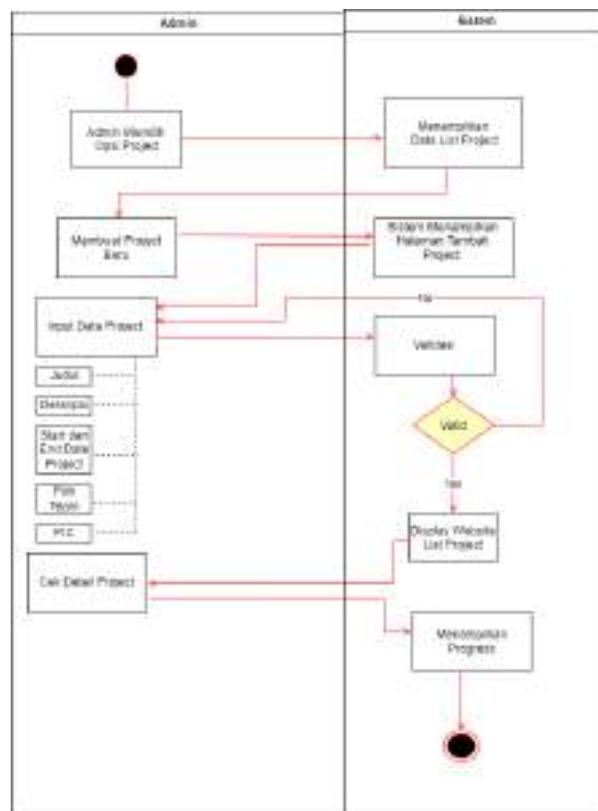


Gambar 3. 7 Activity Diagram Menu Admin

Activity diagram pada Gambar diatas menjeskan mengenai kegiatan user/admin dan alur kerja user/admin ketika meilih opsi menu. Dengan tahapan ringkasnya sebagai berikut:

1. User membuka halaman web.
2. Sistem menampilkan halaman web.
3. User memilih opsi yang ditampilkan halaman utama.

f. Activity Diagram Membuat Project



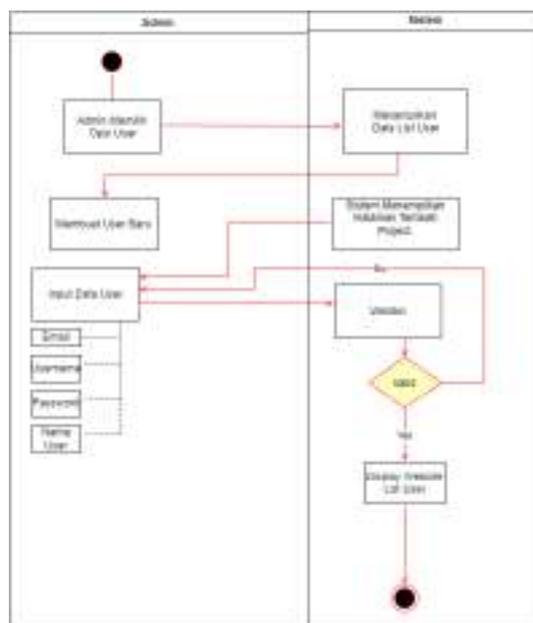
Gambar 3. 8 Activity Diagram Membuat Project

Activity diagram pada gambar diatas menjeskan mengenai kegiatan admin dalam mengelola project , admin memilih opsi buat project baru terlebih dahulu dengan tahapan sebagai berikut:

1. Admin membuka halaman web.
2. Sistem menampilkan halaman web

3. *Admin* memilih fitur *project*
4. Sistem menampilkan halaman data *list project*
5. *Admin* membuat *project* baru
6. Sistem menampilkan halaman tambah *project*
7. *Admin* menginputkan beberapa atribut yang diperlukan untuk detail *project*
8. Sistem memvalidasi datanya, kemudian jika *valid* maka diarahkan ke *display list project* atau jika tidak *valid* maka diberi *attention* mengenai kevalidan data
9. Kemudian barulah *admin* melakukan aktivitas *management*-nya

g. *Activity Diagram* Menambahkan *User* Baru

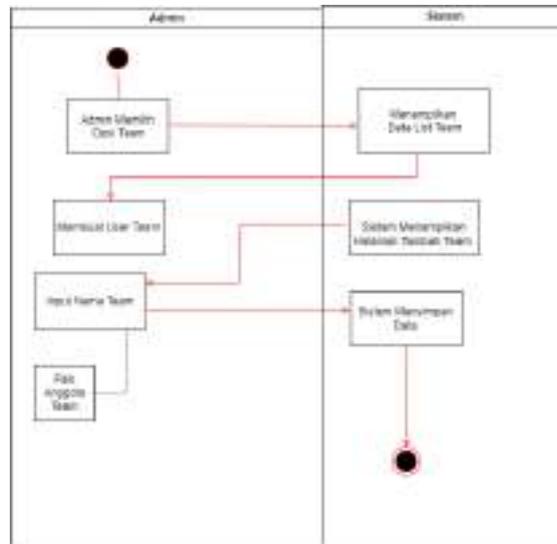


Gambar 3. 9 *Activity Diagram* Menambahkan *User* Baru

Activity diagram pada gambar diatas mendeskripsikan mengenai aktivitas *admin* dalam membuat akun *user*. Selain *user* bisa mendaftarkan akunnya secara mandiri, *admin* juga bisa membuat karena mempunyai kewenangan dalam mengelola atau memanipulasi data *user*, *admin* mempunyai data *list user* untuk melihat semua akun *user* yang terdapat pada *database*, atribut

yang diperlukan untuk data *user* diantaranya *Email*, *Username*, *Password* dan Nama *User*.

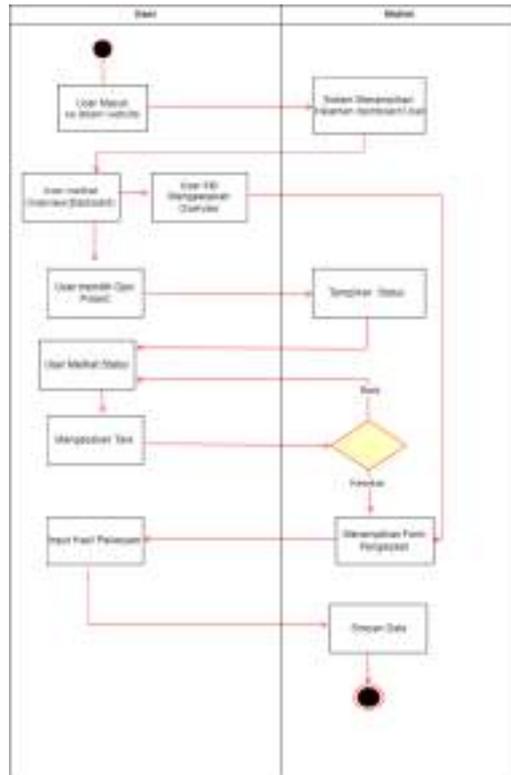
h. Activity Diagram Mengatur Team Project



Gambar 3. 10 Activity Diagram Mengatur Team Project

Activity diagram di atas merupakan pembuatan *team project*, dimulai dari pemilihan menu atau fitur *team* yang dipilih *admin*, kemudian sistem menampilkan data *list team*nya yang sudah ada. Pada tampil data tersebut terdapat beberapa opsi manipulasi data *admin* yang tersedia, jika *admin* ingin menambah *team project* maka *admin* akan memilih fitur tersebut, maka sistem menampilkan *form* untuk menambahkan kan anggota *team* yang akan dimasukan dengan data yang diambil dari *database*. *Database* akan menyimpan hasil simpan-nya

i. Activity Diagram Ketika User Team Mengerjakan Project yang Tersedia



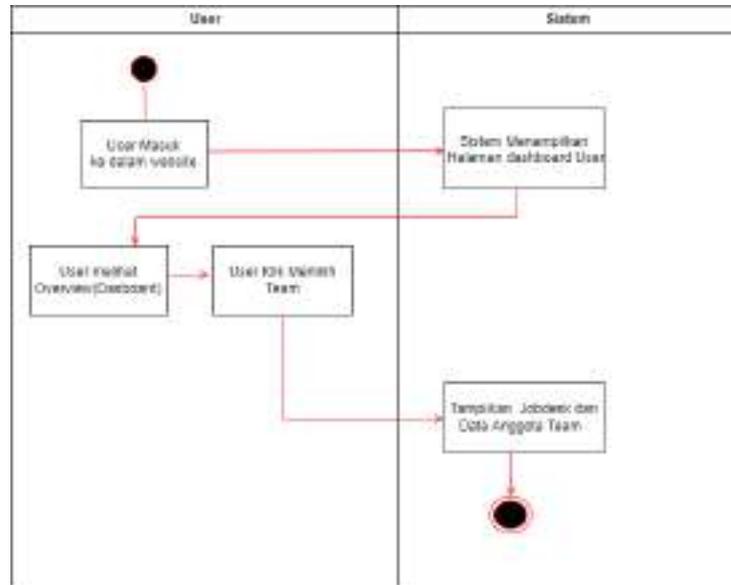
Gambar 3. 11 Activity Diagram Ketika User Team Mengerjakan Project yang Tersedia

Activity diagram pada gambar diatas menggambarkan aktivitas peran *user development*, dengan tahapan sebagai berikut:

1. *User* membuka halaman web.
2. Sistem menampilkan halaman web (*dashboard user*)
3. Ketika *user* mengakses *dashboard*, *user* ditampilkan *task* yang sudah diberikan, *user* bisa mengerjakan langsung pada *overview* atau memilih *opsi project* dahulu
4. Kemudian sistem akan menampilkan status *task*
5. *User* mengerjakan *task*
6. Sistem menampilkan *form* pengerjaan

Ketika *user* memilih pengisian data barang maka *form input* tugas akan ditampilkan dan *user* mengisi data sesuai dengan item yang ada pada *interface*. Setelah semua item diisi maka *user* tinggal mengklik tombol *simpan* dan proses pengerjaan selesai.

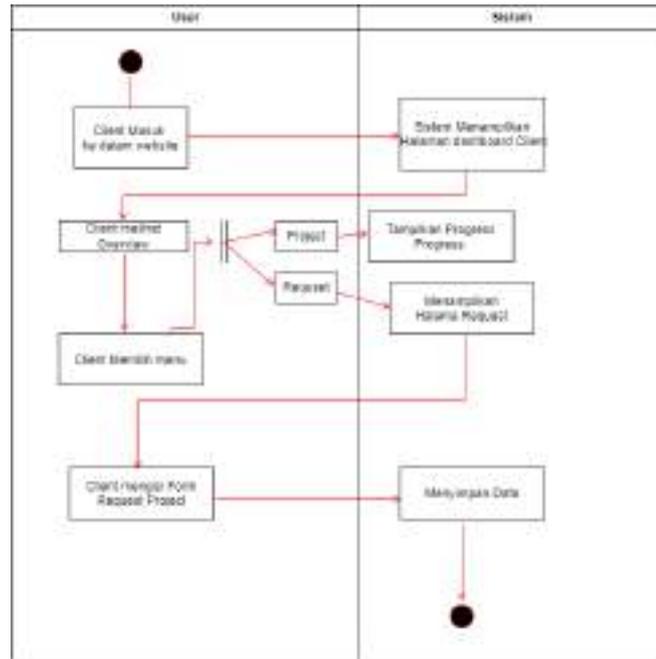
j. *Activity Diagram Ketika User Team Melihat Team*



Gambar 3. 12 *Activity Diagram User Team Melihat Team*

Activity diagram pada gambar diatas mendeskripsikan mengenai aktivitas *user* Ketika memilih opsi *team*, *user* mengakses website kemudian memilih opsi *admin* dan melihat *jobdesk* dan masing-masing anggota *team* termasuk dirinya

k. *Activity Diagram Client Melakukan Request*



Gambar 3. 13 Activity Diagram Client Melakukan Request

Activity diagram di atas merupakan aktivitas *user(client)*, pada *user(client)* hanya dapat mengakses *overview* untuk melihat *progress project* dan *merequest project* pada *admin*.

C. Kesepakatan Proyek

Dalam pelaksanaan proyek ini *Ruby team* telah sepakat dengan pihak *client* untuk mengambil dan membuat proyek sesuai dengan apa yang telah diminta oleh pihak klient. Untuk penjelasannya dapat dilihat sebagai berikut :

1. Waktu Pengerjaan

Kesepakatan proyek berlangsung setelah diskusi bersama dengan dosen pembimbing dengan rentan waktu 5 hari. Dan proyek akan diselesaikan selama 6 bulan sejak kesepakatan dibuat.

2. Fitur Proyek

Terdapat gambaran *wireframe* fitur proyek *Project Management Tools* sebagai berikut :



Gambar 3. 14 Wireframe Aplikasi yang akan dibuat

D. Rencana Jadwal Pelaksanaan Projek

Rencana dan Realisasi Kegiatan Manajemen Proyek Teknologi Informasi

No	Nama Kegiatan	Minggu Pelaksanaan																											
		Februari				Maret				April				Mei				Juni				Juli							
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4				
1	Pembentukan Tim	█	█	█	█																								
2	Pencarian Proyek		█																										
3	Pembuatan Proposal										█	█																	
4	Pembuatan MOU										█	█																	
5	Kesepakatan Pembuatan Proyek			█	█		█	█																					
6	Pengumpulan Data dan Dokumen										█	█																	
7	Desain User Interface (UI)					█	█	█	█		█	█																	
8	Analisis Kebutuhan						█	█	█		█	█																	

BAB IV Pelaksanaan MPTI

A. Progres Pembagian Tugas

Pada pembagian tugas *team* adapun realisasi penyelesaiannya dan juga pembagian tugas per individu. Berikut pembagian dan realisasinya :

1. Pembagian Tugas

NO	JOBDESK	NAMA	KETERANGAN
1.	Ketua	Brilly Jalu Kumara Biseka	<ul style="list-style-type: none">· Menentukan tim manajemen tugas proyek· Membagi jobdesk anggota tim· Mengkoordinasi dan mengatur waktu bimbingan· Merencanakan, melaksanakan, mengawasi, dan mengevaluasi pelaksanaan proyek· Mengkoordinasi dan memimpin setiap pertemuan atau rapat

2.	<i>Backend Developer</i>	Naufal Luthfi Nurhamid	<ul style="list-style-type: none"> · Pengelolaan data <i>project</i> meliputi CRUD (<i>Admin</i>) · Pengelolaan data <i>team (admin)</i> · Transaksi tambah <i>project</i> : <ul style="list-style-type: none"> o API <i>register</i> dan <i>login</i> o API mengelola <i>requirement</i> (CRUD) o API mengelola <i>internal briefing</i> dari setiap <i>project</i> (CRUD) o API mengelola <i>developing</i> dari setiap <i>project</i> (CRUD) o API mengelola <i>testig</i> dari setiap <i>project</i> (CRUD) · Penugasan suatu <i>project</i> kepada <i>team</i> · API mengelola <i>team</i>
3.	UI/UX	Ryan Agung Nugraha	<ul style="list-style-type: none"> · Pembuatan tampilan UI dengan memperhatikan UX · Menentukan <i>layout</i> atau tata letak dari komponen sistem yang dibangun · Penyusunan rancangan UI pada laporan

4.	<i>Tester</i>	Ghaly Alif Dhaifullah	<ul style="list-style-type: none"> · Melakukan <i>testing</i> berkala pada aplikasi
5.	<i>Data Analyst 1</i>	Hamzah Lopez Nurudin	<ul style="list-style-type: none"> · Menganalisa proyek yang berkaitan dengan perancangan sistem · Membuat proses bisnis diagram · Membuat ERD (<i>Entity Relationship Diagram</i>) · Membuat <i>Use Case Diagram</i> · Membuat <i>Activity Diagram</i>
6.	<i>Data Analyst 2</i>	Anita Mega Budi Anggraini	<ul style="list-style-type: none"> · Menganalisa proyek yang berkaitan dengan perancangan sistem · Membuat <i>Activity Diagram (Login, menu Admin)</i>
7.	<i>Frontend Developer</i>	Muhammad Khalid H.	<ul style="list-style-type: none"> · Mengimplementasikan rancangan tampilan yang telah dibuat oleh UI/UX · Membuat validasi <i>input data</i> · Mengintegrasikan <i>backend</i> menggunakan <i>rest api</i>

8.	Sekretaris	Anita Mega Budi Anggraini	<ul style="list-style-type: none"> · Menyusun proposal penawaran proyek · Menyusun surat perjanjian / MOU · Menyiapkan <i>logbook</i> · Notulensi rapat tim · Pembuatan laporan pengerjaan proyek
9.	Bendahara	Anita Mega Budi Anggraini	<ul style="list-style-type: none"> · Pembuatan rincian biaya · Penyusunan pengeluaran biaya

Tabel 4. 1 Daftar Progress Pembagian Tugas Team

2. Realisasi Penyelesaian Tugas

a. Data Analyst

1) Software yang digunakan

- Diagram IO
- Astah UML

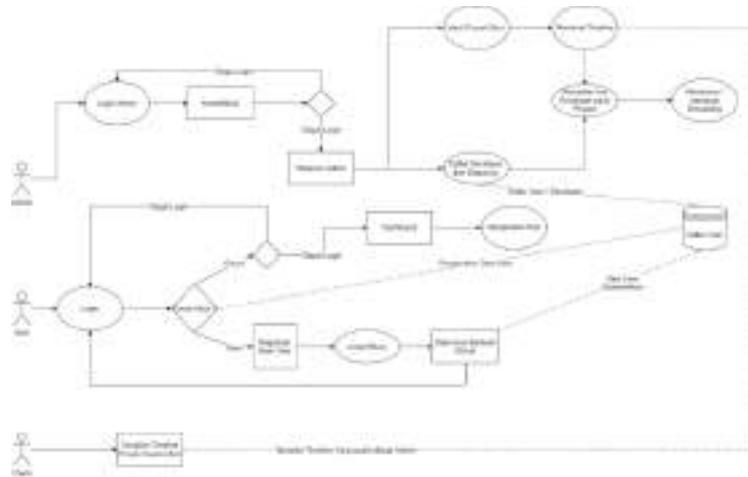
2) Realisasi

Data Analyst	
List Tugas	Realisasi Penyelesaian
Membuat Analisis Kebutuhan Sistem	V
Membuat Diagram Proses Bisnis	V
Membuat ERD (Entity Relationship Diagram)	V
Membuat Use Case Diagram	V
Merancang Database	V
Membuat Activity Diagram	V

Tabel 4. 2 Realisasi Penyelesaian Tugas Data Analyst

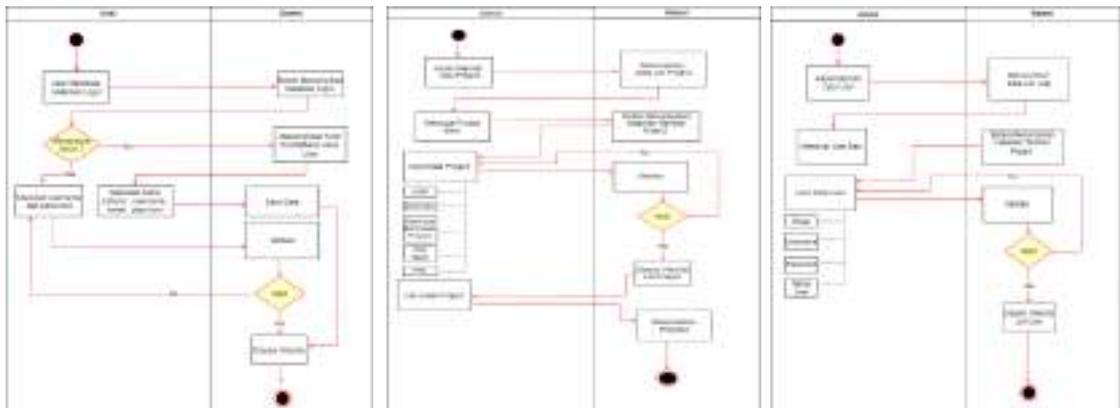
3) Hasil

Use Case Diagram



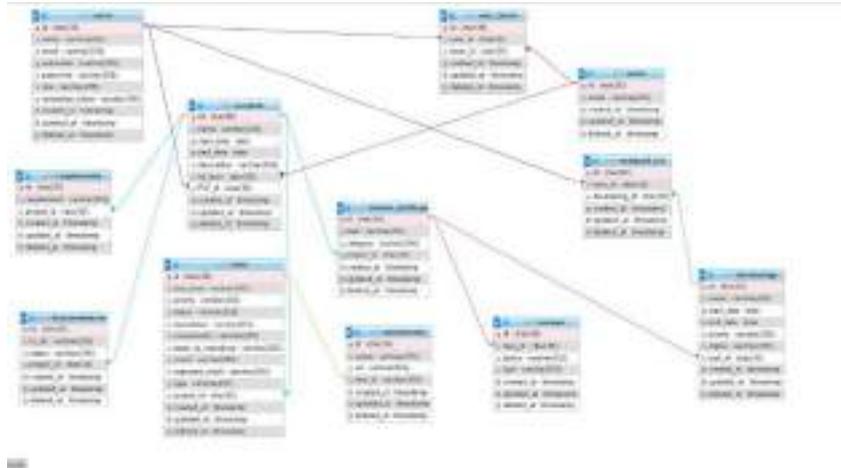
Gambar 4. 1 Use Case Diagram yang dibangun

Activity Diagram



Gambar 4. 2 Daftar Activity Diagram yang dibangun

Rancangan Database



Gambar 4. 3 Rancangan Database yang dibangun

b. Sekretaris dan Bendahara

1) Software yang digunakan

- Ms Office (Word, PDF)
- Kalkulator

2) Realisasi

Sekertaris	
List Tugas	Realisasi Penyelesaian
Membuat Proposal	✓
Membuat MOU	✓
Membuat Log Book	✓
Membuat Laporan	✓

Tabel 4. 3 Realisasi Penyelesaian Tugas Sekertaris

Bendahara	
List Tugas	Realisasi Penyelesaian
Membuat Anggaran Dana	✓

Tabel 4. 4 Realisasi Penyelesaian Tugas Bendahara

3) Hasil

- Membuat Proposal



Gambar 4. 4 Pembuatan Proposal

- Membuat Surat Perjanjian



Gambar 4. 5 Pembuatan Surat Perjanjian

- Pembuatan Logbook

LOG BOOK MANAJEMEN PROYEK TEKNOLOGI INFORMASI (MAMASITP)
PROGRAM STUDI S1 INFORMATIKA UNIVERSITAS MERBAU KARANG LIA 2020/2021

Nama Kelompok BPD :		
No	NIM	NAMA
1	200075292	Nandi Lubri Nohandi
2	200075293	Ryan Agung Nugraha
3	200075291	Muhammad Khalid Hanifanyah
4	200075294	Silly Jala Kusuma Binda
5	200075290	Chay Al Chahidali
6	200075296	Jetta Mega Budi Anggani
7	200075290	Hanzah Lopez Handis

Jadi Manajemen Tegan Proyek : **Adrian** Project Management Task **Subana** Ppt

Dosen Pembimbing : **Rinal (Gabe)** S.Iam, M.Eng

Logbook Minggu 4 dari 10 Bab 10/11

No.	Kegiatan	Waktu Pelaksanaan		Bukti	Sudut, Dampak, Refleksi (jika ada)	Penaf. M. Dan/Asist. P.	Penaf. Dosen Pembimbing MPTI
		Ber/Tgl	Berakhir				
1	Membaca cara menggunakan dan memahami apa itu ERP. Mendeskripsikan apa itu ERP dan apa itu ERP. Mendeskripsikan apa itu ERP.	10/11	11/11		Di sini membahas tentang ERP yang merupakan salah satu sistem yang digunakan dalam bisnis untuk mengelola sumber daya perusahaan.		
2	Menyusun laporan tentang ERP dan apa itu ERP. Mendeskripsikan apa itu ERP dan apa itu ERP.	10/11	11/11		Di sini membahas tentang ERP yang merupakan salah satu sistem yang digunakan dalam bisnis untuk mengelola sumber daya perusahaan.		

Gambar 4. 6 Pembuatan Logbook

- Notulensi Rapat

Bimbingan MPTI 25 Maret 2022

Duan dana : 1.000.000 - 2.000.000

1. Biasanya tidak dikembangkan tidak masalah, namun di lengkapi
2. Melengkapi dan memantapkan tidak, menggunakan tidak boleh
3. Boleh Merespon baru
4. Pake CI backend dan frontend jadi satu
5. Laporan Sistem Client dari dalam dan luar user
6. Admin -> Scheduling, Input Project, Refresh, Login, progress (edit data) estimasi proyek
7. Client -> timeline
8. Developer - mengetahui apakah sistem overload atau tidak
9. Refresh - jenis project, development siapa aja
10. Input project - timeline waktu
11. Scheduling - project management (D.S. Pak Anhari)
12. Timeline - Client - request project, penawaran project
 - Request project = project, harga, support material
 - Penawaran project = ke klien
13. Estimasi proyek - untuk estimasi harga data
14. Estimasi proyek - project, kesepakatan harga
15. Web project = amik user ac id
16. Step2 bisa di modifikasi / bisa di jadikan satu ke refresh
17. Membuat MOU dari proposal
18. Penawaran budget min 1.000.000
19. Project 3-4 bulan, 2 bulan persiapan meeting
20. Mau lanjut atau tidak baru? (Dibuat)

Bimbingan 1 agustus

1. Ppt di lengkapi bagian ketua (realisasi job)
2. Timeline tiap job di kasi bukti (ss)
3. Data analyst Melengkapi permintaan tsj requirement
4. Log book, notulensi rapat,
5. Ppt lebih diringkas
6. Rincian dana pengeluaran di tambah ke ppt
7. UI di tampilan perslide 4/angung
8. Tampilan UI 404 si hapus
9. Backend di demokan langsung
10. Tester di buat skenario / deskripsi tiap langkah dan di jelaskan
11. Data analyst di jelaskan bagian2 yg dikerjakan
12. Wa penguji Pak Ardi
13. Tabel bukti progress kerja
14. Dresscode betas sopan

Gambar 4. 7 Hasil Notulensi Rapat Team

- Laporan



Gambar 4. 8 Pembuatan Laporan

- Anggaran Biaya

1. PEMASUKKAN

NO	AKUN	SATUAN		HARGA	JUMLAH
1	Dana Proyek	-	-	Rp 2.500.000,00	Rp 2.500.000,00
Total Pemasukan					Rp 2.500.000,00

Tabel 4. 5 Biaya Pemasukan

2. PENGELUARAN

NO	AKUN	SATUAN		HARGA	JUMLAH
1	Domain	3	Bulan	Rp 50.000,00	Rp 150.000,00
2	Server	3	Bulan	Rp 200.000,00	Rp 600.000,00
3	Font	1	Paket	Rp 50.000,00	Rp 50.000,00

4	<i>Icon</i>	1	Paket	Rp. 100.000,00	Rp. 100.000,00
5	<i>Fee Team</i>	1	Team	Rp. 1.000.000,00	Rp. 1.000.000,00
6	<i>Figma</i>	3	Bulan	Rp. 100.000,00	Rp. 300.000,00
7	<i>Nation Team</i>	3	Bulan	Rp. 100.000,00	Rp. 300.000,00
Total Pengeluaran					Rp. 2.500.000,00

Tabel 4. 6 Biaya Pengeluaran

3. Penyusunan Biaya Yang Telah Terpakai

No	Akun	Satuan		Harga	Jumlah
1	<i>Domain</i>	3	Bulan	Rp. 175.000	Rp. 525.000
2	<i>Hosting</i>	3	Bulan	Rp. 98.000	Rp. 294.000
3	Biaya Akomodasi (<i>Transport dll</i>)	7	Orang	Rp. 10.000	Rp. 70.000
4	Biaya Seminar MPTI	7	Orang	Rp. 50.000	Rp. 350.000
5	ATK	1	Paket	Rp. 50.000	Rp. 50.000
6	Akses Internet	3	Bulan	Rp. 50.000	Rp. 150.000
Total Pengeluaran					Rp. 1.439.000

Tabel 4. 7 Penyusunan Biaya yang Telah Terpakai

4. Rekap Detail Realisasi Anggaran Dana Perjobdesk

No	Jabatan	Jumlah	Gaji	Total Jam Kerja	Presentasi Jam Kerja	Total Gaji
1	Ketua	1	Rp. 8.000	50 Jam	15 %	Rp. 400.000
2	<i>Backend Developer</i>	1	Rp. 8.000	86 Jam	26 %	Rp. 688.000

3	UI/UX	1	Rp. 8.000	37 Jam	11 %	Rp. 296.000
4	Tester	1	Rp. 8.000	19 Jam	5 %	Rp. 152.000
5	Data Analyst 1	1	Rp. 8.000	23 Jam	7 %	Rp. 184.000
6	Data Analyst 2	1	Rp. 8.000	9 Jam	3 %	Rp. 72.000
7	Frontend Developer	1	Rp. 8.000	65 Jam	19 %	Rp. 520.000
8	Sekretaris	1	Rp. 8.000	38 Jam	12 %	Rp. 304.000
9	Bendahara	1	Rp. 8.000	8 Jam	2 %	Rp. 64.000
	Total	9		335 Jam	100%	Rp. 2.680.000

Tabel 4. 8 Realisasi Anggaran Dana pada Tiap Jobdesk

c. UI/UX

1) Software yang digunakan

- Figma

2) Realisasi

UI / UX	
List Tugas	Realisasi Penyelesaian
Membuat Wire frame	V
Membuat User Interface	V
Membuat Prototype	V

Tabel 4. 9 Realisasi Penyelesaian Tugas UI / UX

3) Hasil

User Interface



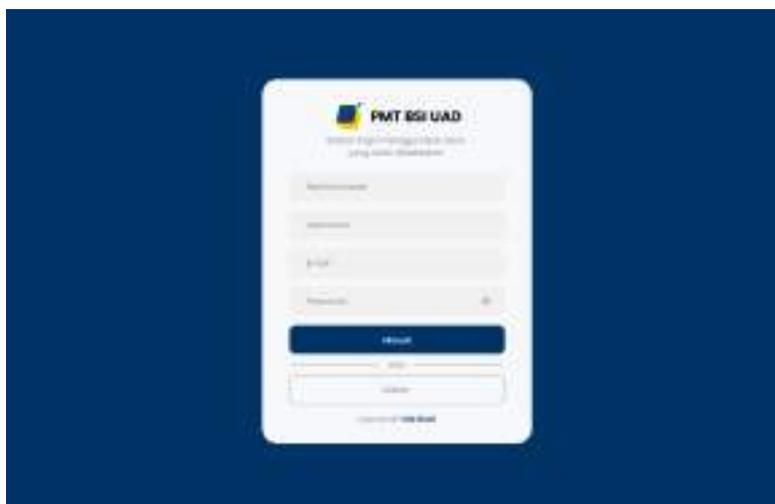
Gambar 4. 9 User Interface yang Telah Dibangun

Halaman *Login*



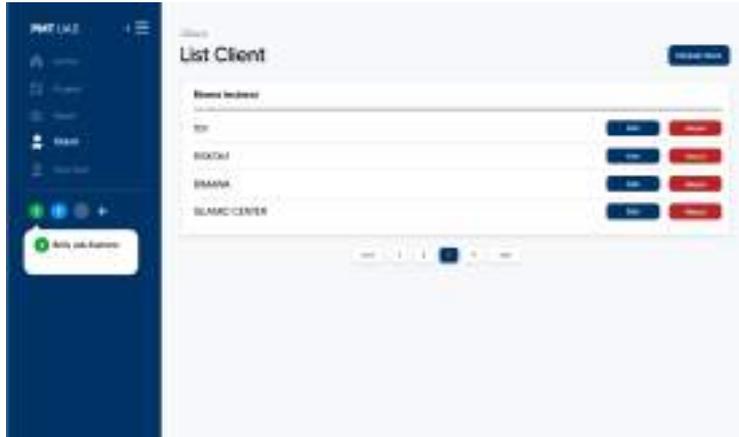
Gambar 4. 10 Halaman Login

Halaman *Register*



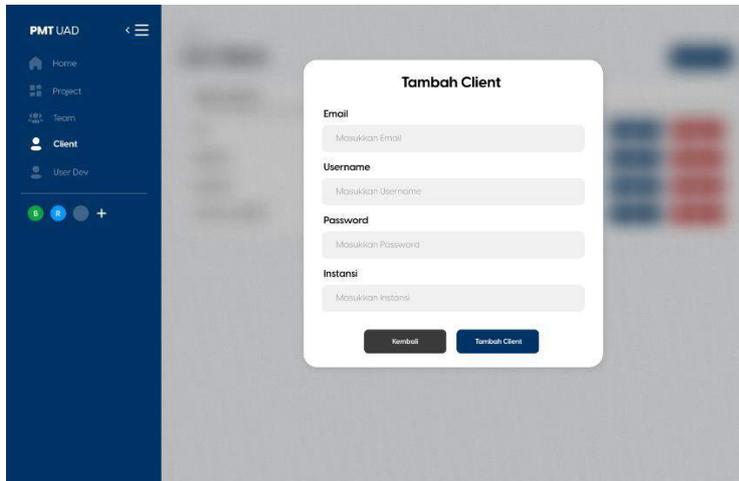
Gambar 4. 11 Halaman Register

Halaman Admin (List Client)



Gambar 4. 12 Halaman Admin (List Client)

Halaman Admin (Tambah Client)



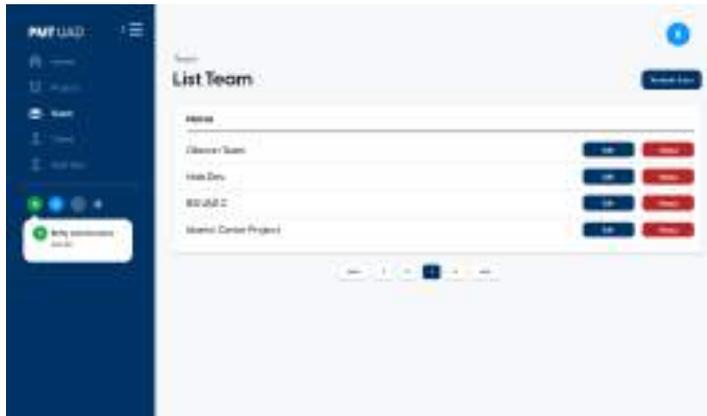
Gambar 4. 13 Halaman Admin (Tambah Client)

Halaman Admin (List Project)



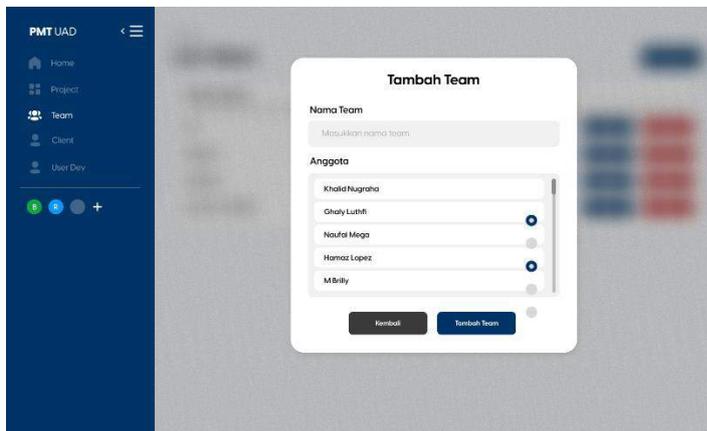
Gambar 4. 14 Halaman Admin (List Project)

Halaman Admin (List Team)



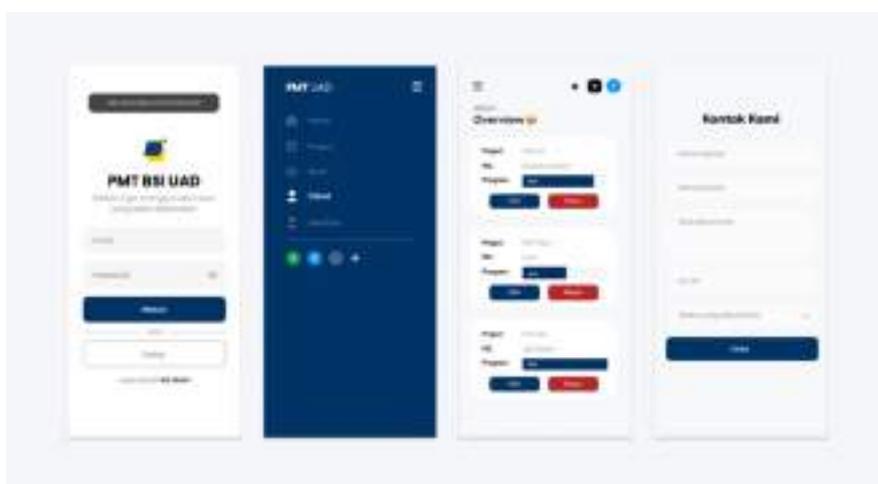
Gambar 4. 15 Halaman Admin (List Team)

Halaman Admin (Tambah Team)

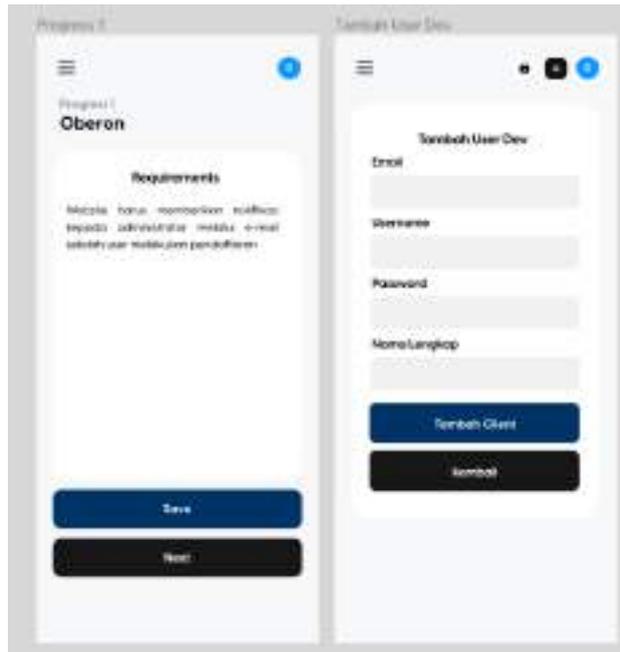


Gambar 4. 16 Halaman Admin (Tambah Team)

Tampilan Responsive



Gambar 4. 17 Tampilan Responsive 1



Gambar 4. 18 Tampilan Responsive 2

Halaman Client



Gambar 4. 19 Halaman Client

Halaman Client (Request Project)



Gambar 4. 20 Halaman Client (Request Project)

Halaman User



Gambar 4. 21 Halaman User

Halaman User (Progress)



Gambar 4. 22 Halaman User (Progress)

d. Backend Developer

1) Software yang digunakan

- Vscod
- Draw io

2) Realisasi

Backend Developer	
List Tugas	Realisasi Penyelesaian
Mengaplikasikan Rancangan Database	✓
Membuat API	✓
Mendeploy API ke hosting	✓

Tabel 4. 10 Realisasi Penyelesaian Tugas Backend Developer

3) Hasil

Pengelolaan Data Project Meliputi CRUD (Admin)

```
public function createproject(Request $request)
{
    $validator = Validator::make($request->all(), [
        "name" => "required|string",
        "description" => "required|string",
        "id_team" => "required|uuid",
        "PIC_id" => "required|uuid",
        "start_date" => "required|string",
        "end_date" => "string|nullable",
    ]);
    if ($validator->fails()) {
        return response()->json(["data" => $validator->errors()], 400);
    };
    $fields = $validator->validated();
    $fields['start_date'] = date('Y-m-d H:i:s', $fields['start_date']);
    if ($fields['end_date']) {
        $fields['end_date'] = date('Y-m-d H:i:s', $fields['end_date']);
    }
    $project = Project::where('name', $fields['name'])->first();
    if ($project) {
        return response()->json(["error" => "Nama project Sudah digunakan"], 400);
    }
    $new_project = Project::create($fields);
    return response()->json([
        "User" => $new_project,
        "Status" => "Create Project Succeed"
    ], 200);
}
```

Gambar 4. 23 Code untuk Pengelolaan Data Project Meliputi CRUD (Admin)

Pengelolaan Data Team CRUD (Admin)

```

public function updateTeamRequest($request)
{
    $validasi = Validator::make($request->all(), [
        'name' => 'required|string',
        'email' => 'required',
        'password' => 'required'
    ]);

    if ($validasi->fails()) {
        return response()->json(['data' => $validasi->errors()], 400);
    }

    $update = $request->all();
    $team = Team::where('name', $update['name'])->first();
    if (!$team) {
        return response()->json(['error' => 'Tidak ada data dengan nama tersebut'], 400);
    }
    $team->update($update);

    if ($update['password']) {
        $password = $update['password'];
        $password = bcrypt($password);
        $update['password'] = $password;
    }

    $team->update($update);

    return response()->json([
        'data' => $team,
        'status' => 'Update data berhasil'
    ], 200);
}

```

Gambar 4. 24 Code untuk Pengelolaan Data Team CRUD (Admin)

Register dan Login

```

public function register($request)
{
    $validasi = Validator::make($request->all(), [
        'name' => 'required|string',
        'email' => 'required|email',
        'password' => 'required|string',
        'password2' => 'required|string',
        'role' => 'required|string'
    ]);

    if ($validasi->fails()) {
        return response()->json(['data' => $validasi->errors()], 400);
    }

    $password = $validasi->all()['password'];
    if ($password['role'] == 'user' || $password['role'] == 'client') {
        return response()->json(['error' => 'Salah Salah'], 400);
    }

    $user = User::where('email', $validasi->all()['email'])->first();
    if ($user) {
        return response()->json(['error' => 'Email Sudah Berdaftar'], 400);
    }

    $password['password2'] = bcrypt($password['password2']);
    $user->update($password);

    return response()->json([
        'data' => $user,
        'status' => 'Register Berhasil'
    ], 200);
}

```

Gambar 4. 25 Code untuk Fungsi Register

```

public function login(Request $request)
{
    $validator = Validator::make($request->all(), [
        'username' => 'required|string',
        'password' => 'required|string'
    ]);
    if ($validator->fails()) {
        return response()->json(['data' => $validator->errors()], 400);
    };
    $fields = $validator->validated();
    $user = User::where('username', $fields['username'])->first();
    if (!$user) {
        return response()->json(['error' => 'Username / password salah'], 400);
    };
    $token = $user->createToken('auth token')->accessToken;
    return response()->json([
        'user' => $user->toArray(),
        'token' => $token,
        'accessToken' => $token,
        'status' => 'Login Berhasil!'
    ], 200);
}

```

Gambar 4. 26 Code Untuk Fungsi Login

Requirement

```

public function createRequirement(Request $request)
{
    $validator = Validator::make($request->all(), [
        'requirement' => 'required|string',
        'project_id' => 'required|string',
    ]);
    if ($validator->fails()) {
        return response()->json(["data" => $validator->errors()], 400);
    };
    $fields = $validator->validated();
    $project = Project::where('id', $fields['project_id'])->first();
    if (!$project) {
        return response()->json(["error" => "Id Project tidak ditemukan"], 400);
    }
    $new_requirement = Requirement::create($fields);
    return response()->json([
        "User" => $new_requirement,
        "Status" => "Create Requirement Succeed"
    ], 200);
}

public function getRequirement($projectId)
{
    $data = Requirement::where('project_id', '=', $projectId)->get();
    if ($data) {
        return response()->json([
            "data" => $data,
            "Status" => "Success"
        ], 200);
    } else {
        return response()->json([
            "Status" => "Failed"
        ], 400);
    }
}

```

Gambar 4. 27 Code untuk Fitur Requirement

Internal Briefing

```
public function createdeveloping(Request $request)
{
    $validator = Validator::make($request->all(), [
        'task_id' => 'required|uid',
        'name' => 'required|string',
        'start_date' => 'required|string',
        'priority' => 'required|string',
        'status' => 'required|string',
        'end_date' => 'string|nullable',
        'user.*' => 'uid'
    ]);

    if ($validator->fails()) {
        return response()->json(['data' => $validator->errors()], 400);
    }

    $fields = $validator->validated();
    $fields['start_date'] = date('Y-m-d H:i:s', $fields['start_date']);
    $fields['end_date'] = date('Y-m-d H:i:s', $fields['end_date']);

    if ($fields['status'] == config('constants.DEV_STATUS.WORK_IN_PROGRESS') || $fields['status'] == config('constants.DEV_STATUS.DONE') || $fields['status'] == config('constants.DEV_STATUS.PENDING') || $fields['status'] == config('constants.STATUS.CANCELED')) {
    } else {
        return response()->json([
            'error' => "Status salah",
            "Status tersedia" => config('constants.DEV_STATUS')
        ], 400);
    }

    if ($fields['priority'] == config('constants.CATEGORY.PRIORITY_1') || $fields['priority'] == config('constants.CATEGORY.PRIORITY_2')) {
    } else {
        return response()->json([
            'error' => "Priority salah",
            "Priority tersedia" => config('constants.CATEGORY')
        ], 400);
    }
}
```

Gambar 4. 28 Code untuk Fitur Internal Briefing

Developing dan penugasan

```
public function createdeveloping(Request $request)
{
    $validator = Validator::make($request->all(), [
        'task_id' => 'required|uid',
        'name' => 'required|string',
        'start_date' => 'required|string',
        'priority' => 'required|string',
        'status' => 'required|string',
        'end_date' => 'string|nullable',
        'user.*' => 'uid'
    ]);

    if ($validator->fails()) {
        return response()->json(['data' => $validator->errors()], 400);
    }

    $fields = $validator->validated();
    $fields['start_date'] = date('Y-m-d H:i:s', $fields['start_date']);
    $fields['end_date'] = date('Y-m-d H:i:s', $fields['end_date']);

    if ($fields['status'] == config('constants.DEV_STATUS.WORK_IN_PROGRESS') || $fields['status'] == config('constants.DEV_STATUS.DONE') || $fields['status'] == config('constants.DEV_STATUS.PENDING') || $fields['status'] == config('constants.STATUS.CANCELED')) {
    } else {
        return response()->json([
            'error' => "Status salah",
            "Status tersedia" => config('constants.DEV_STATUS')
        ], 400);
    }

    if ($fields['priority'] == config('constants.CATEGORY.PRIORITY_1') || $fields['priority'] == config('constants.CATEGORY.PRIORITY_2')) {
    } else {
        return response()->json([
            'error' => "Priority salah",
            "Priority tersedia" => config('constants.CATEGORY')
        ], 400);
    }
}
```

Gambar 4. 29 Code untuk Fitur Developing

- Vscode
- Figma

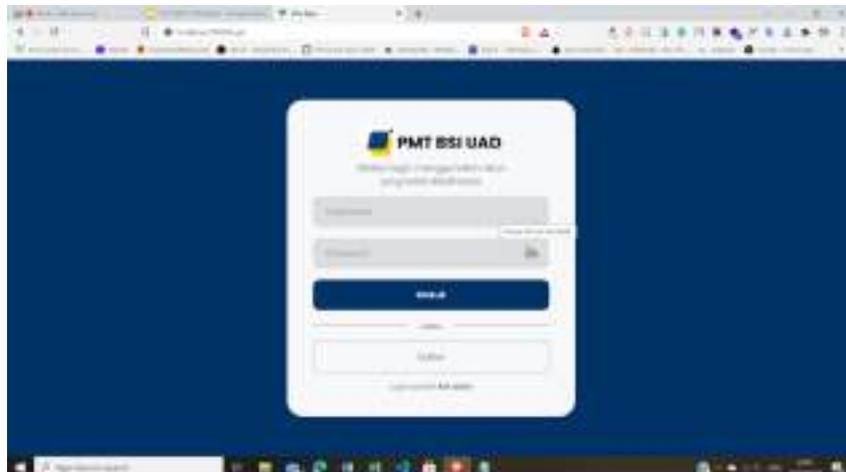
2) Realisasi

Frontend Developer	
List Tugas	Realisasi Penyelesaian
Mengaplikasikan Desain yang dibuat UI /UX	V
Membuat Validasi Website	V
Mendeploy Website ke penyedia hosting	V
Menyambungkan code program backend menggunakan rest api	V

Tabel 4. 11 Realisasi Penyelesaian Tugas Frontend Developer

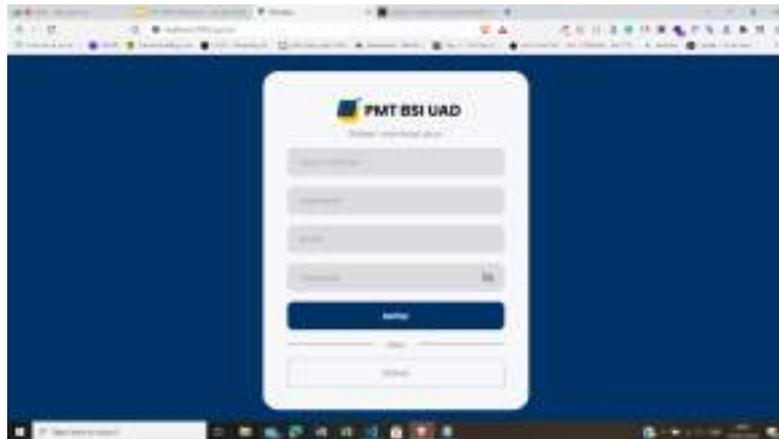
3) Hasil

Halaman *Login*



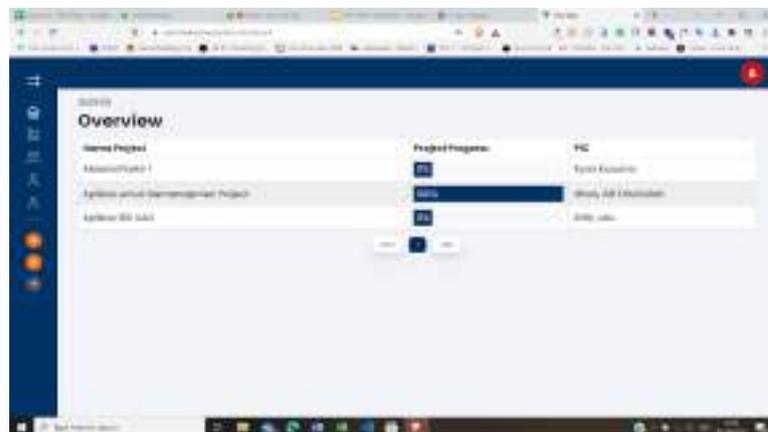
Gambar 4. 32 Tampilan Halaman Login

Halaman *Register*



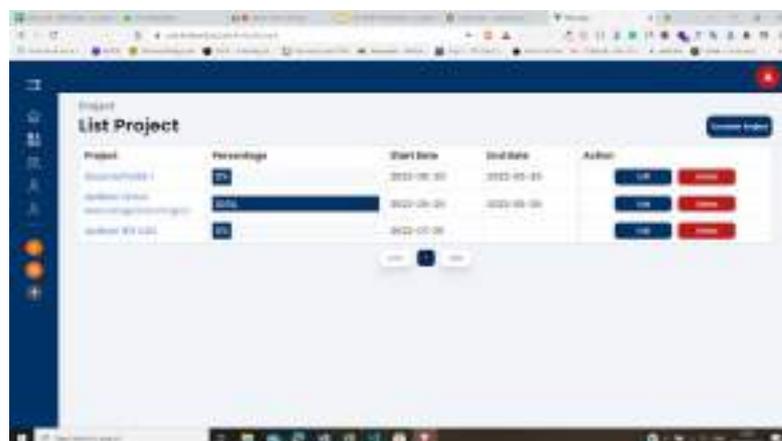
Gambar 4. 33 Tampilan Halaman Register

Dashboard Admin



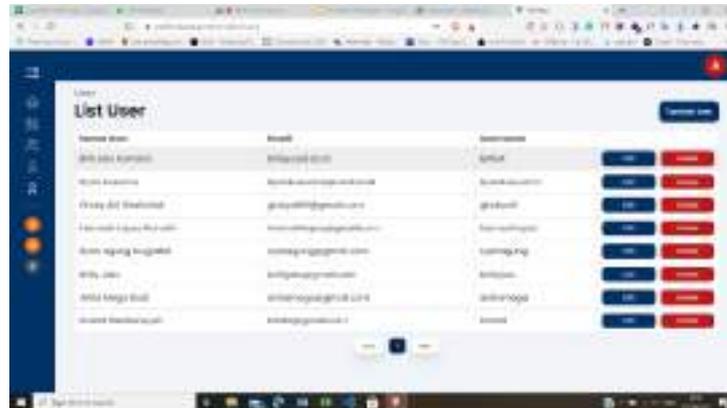
Gambar 4. 34 Tampilan Dashboard Admin

Halaman List Project (Admin)



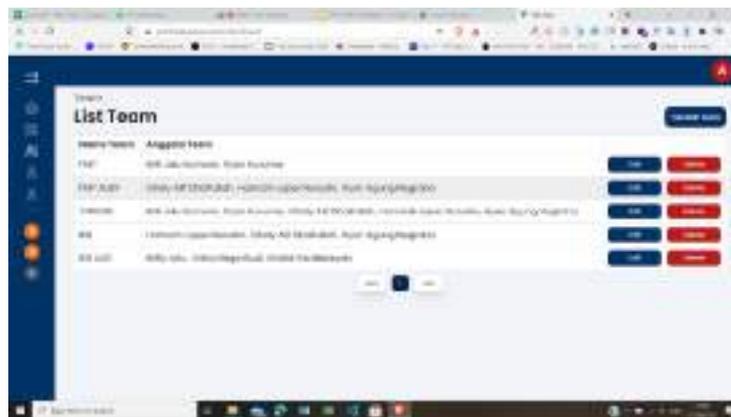
Gambar 4. 35 Tampilan Halaman List Project (Admin)

Halaman *List User (Admin)*



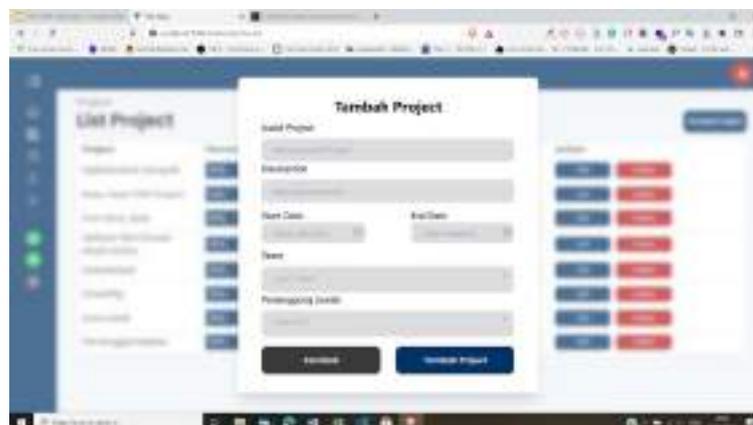
Gambar 4. 36 Tampilan Halaman List User (Admin)

Halaman *List Team (Admin)*



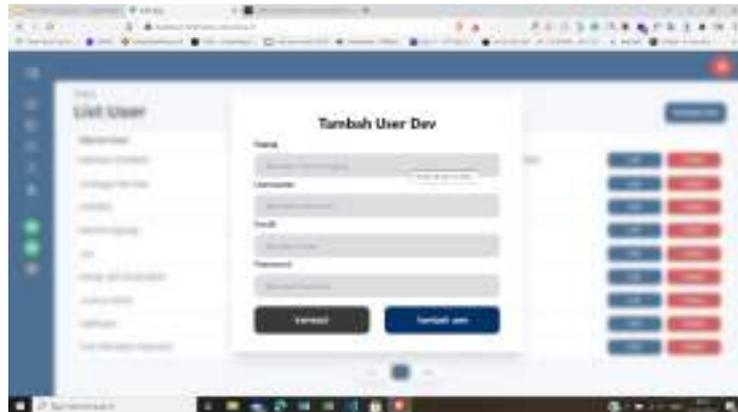
Gambar 4. 37 Tampilan Halaman List Team (Admin)

Modal Edit dan Tambah Data *Project*



Gambar 4. 38 Tampilan Modal Edit dan Tambah Data Project

Modal Edit dan Tambah User



Gambar 4. 39 Tampilan Modal Edit dan Tambah User

Modal Edit dan Tambah Client



Gambar 4. 40 Tampilan Modal Edit dan Tambah Client

f. Tester

1) Software yang digunakan

- Postman

2) Realisasi

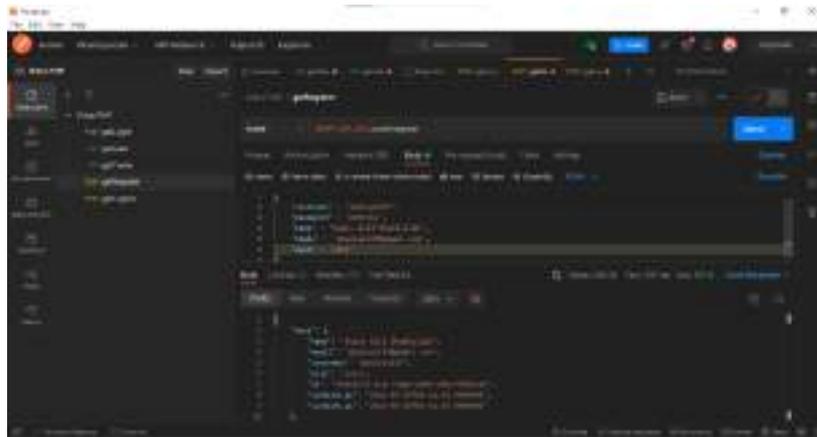
Tester	
List Tugas	Realisasi Penyelesaian
Melakukan Testing API yang telah dideploy	V
Melakukan Fungsional Testing terhadap Website	V

Tabel 4. 12 Realisasi Penyelesaian Tugas Tester

3) Hasil

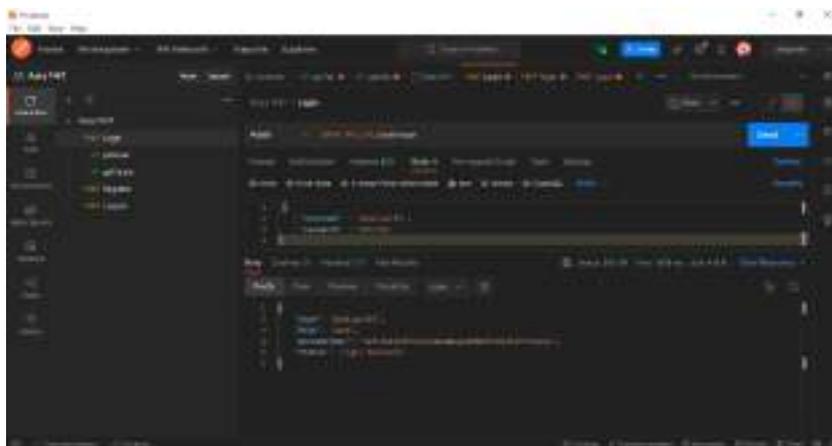
Testing API untuk Auth,

Pertama melakukan register user menggunakan aplikasi postman dengan *method POST Register*.



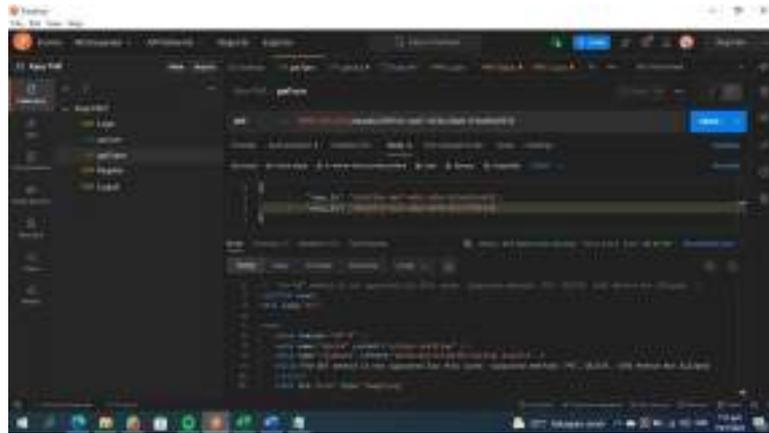
Gambar 4. 41 Testing API untuk Fitur Register

Setelah melakukan register *user*, kita dapat melakukan *login* dengan *request Login*



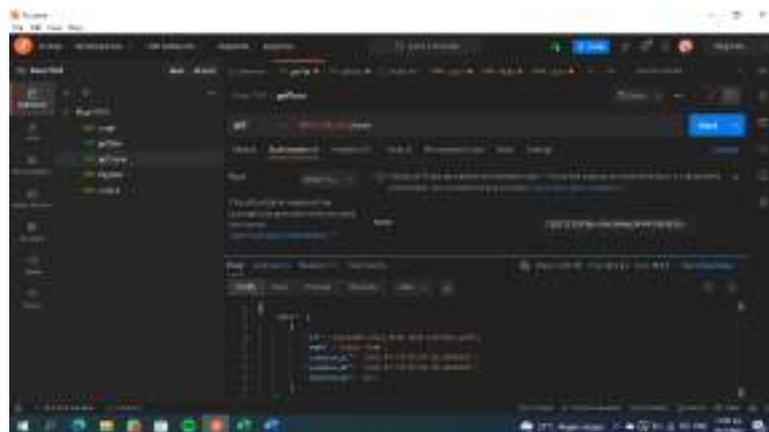
Gambar 4. 42 Testing API untuk Fitur Login

User dapat melihat data team yang ada menggunakan akses token yang tertera dan dimasukkan ke *authorization* pada request *getTeam* dengan *method GET*.



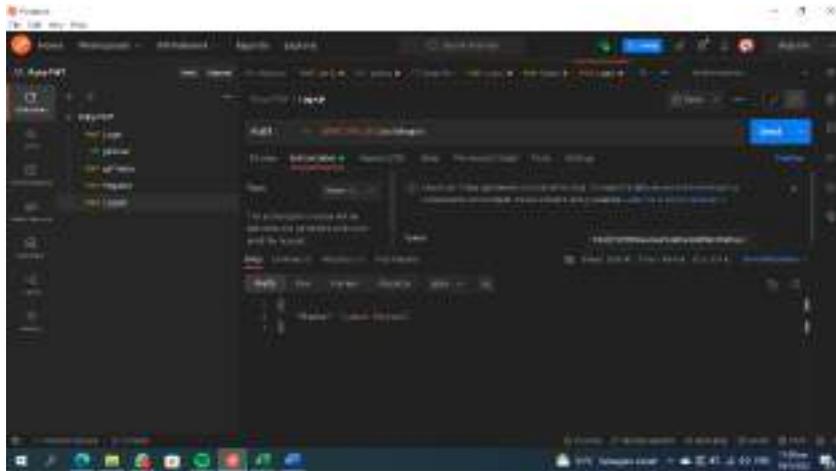
Gambar 4. 43 Testing API untuk Auth Ketika Hasilnya Error

Tampilan diatas masih error

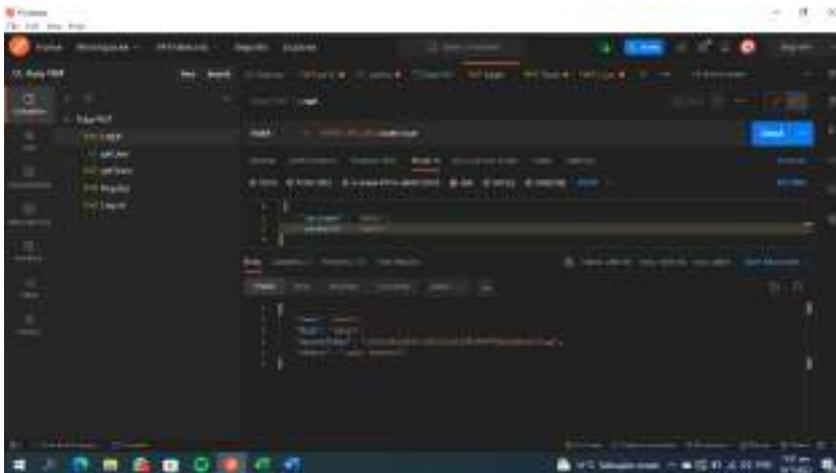


Gambar 4. 44 Testing API untuk Melihat Team

Setelah diperbaiki, kemudian dapat melihat *team* yang ada. kita membuat *Team* baru pada *request getTeam* dengan *method POST* dengan menggunakan akun *admin*, karena pembuatan *team* hanya bisa diakses dengan admin. *Logout* terlebih dahulu dari akun *user* sebelumnya.

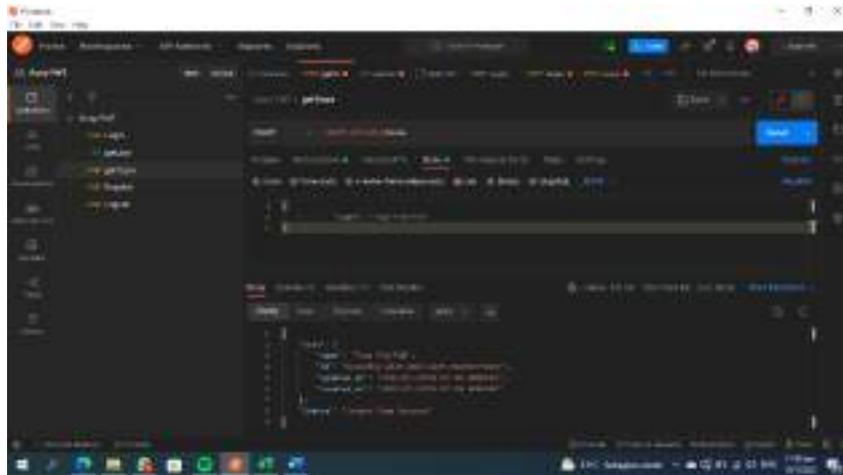


*Gambar 4. 45 Testing API Setelah dilakukan Perbaikan
Kemudian login Kembali dengan akun admin.*

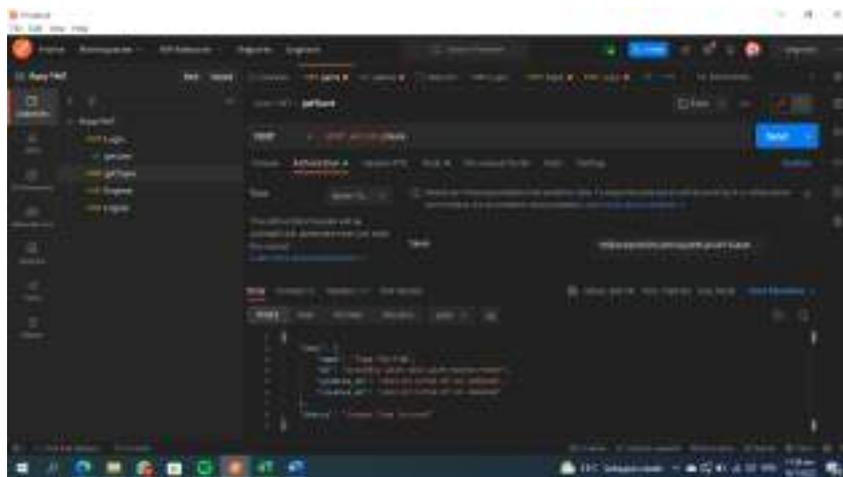


Gambar 4. 46 Testing API untuk Auth Admin

Admin dapat melihat data *user* dan *team* yang ada. Kemudian sekarang kita membuat *team* baru pada *request getTeam* dengan *method POST*. Untuk membuat *team* baru kita membutuhkan akses token lalu membuat *body raw* seperti berikut

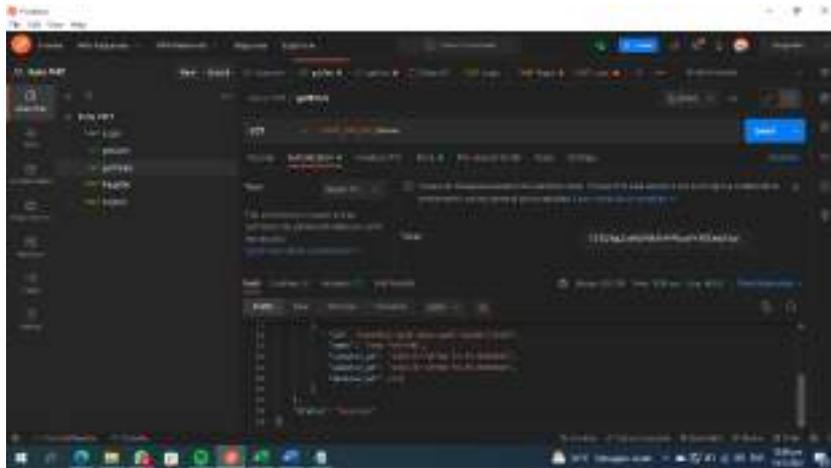


Gambar 4. 47 Testing API untuk Fitur Team Project



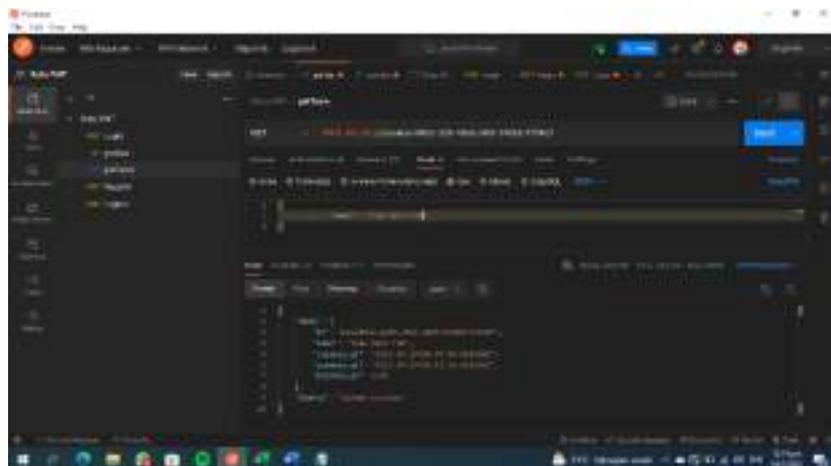
Gambar 4. 48 Testing API untuk Request Team (1)

Setelah itu kita pastikan team yang sudah didaftarkan melalui *method GET* dengan *request getTeam*.



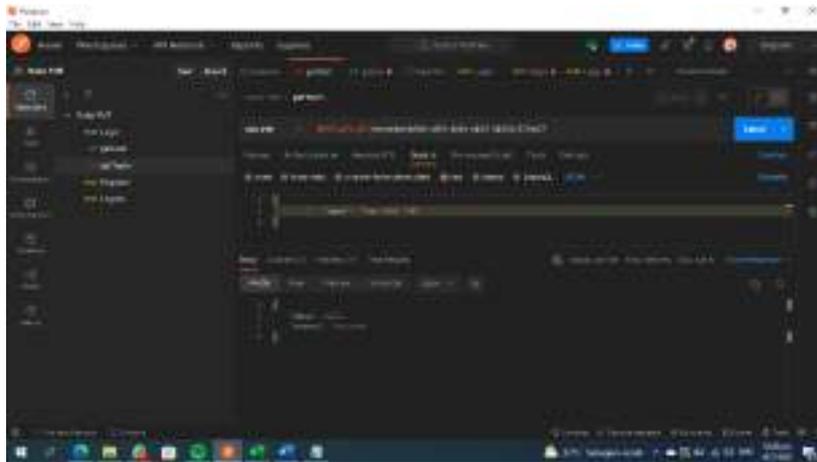
Gambar 4. 49 Testing API untuk Request Team (2)

Setelah kita pastikan data team yang ada terdaftar maka kita akan *update* data tersebut dengan menggantikan nama, caranya ubah *request getTeam* dengan *method PUT* dengan memasukkan id *team* yang sudah ada pada url dan ubah nama di raw dengan nama yang ingin diubah yang sebelumnya “*Team FUN-FUN*” menjadi “*Team HAVE FUN*”



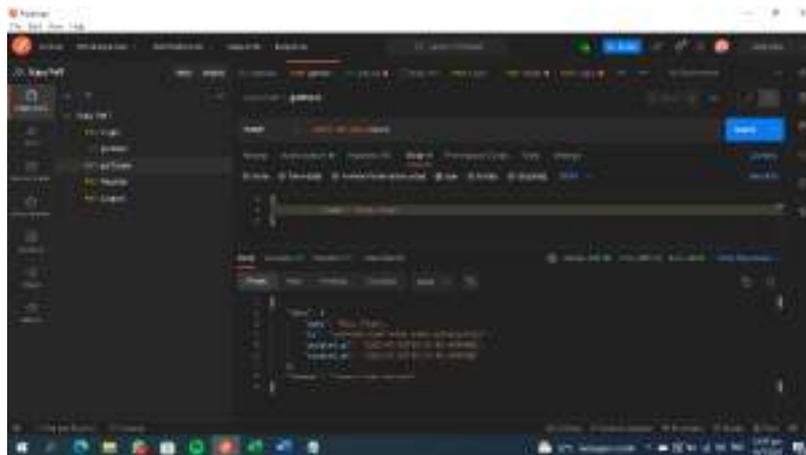
Gambar 4. 50 Testing API untuk Update Team

Lalu kita aplikasikan juga dengan menghapus data *team* tersebut dengan *method DELETE* pada *request getTeam*,



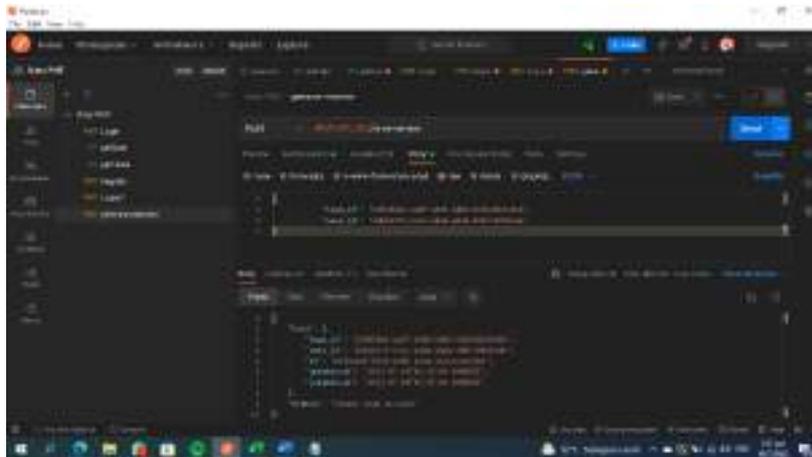
Gambar 4. 51 Testing API untuk Delete Team

Lalu buat terlebih dahulu team karena tadi *team* sudah di *delete*



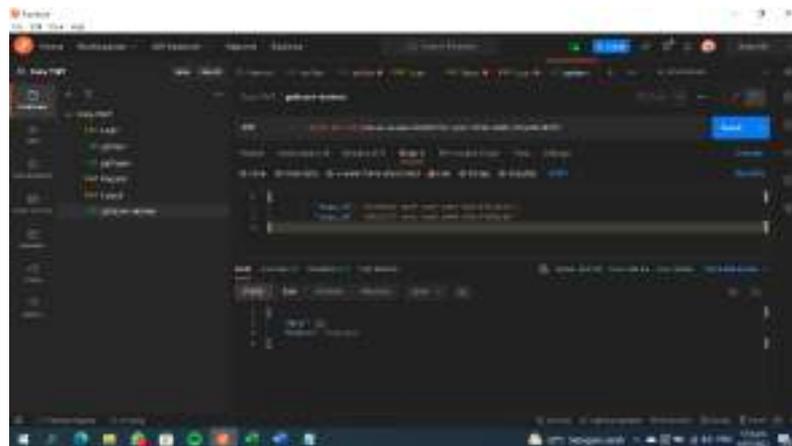
Gambar 4. 52 Testing API Mengecek Team yang tersedia Setelah Team dihapus

Setelah ditambah maka kita akan menambah member pada team yang sudah kita buat, khusus tambah member hanya dapat dilakukan oleh admin maka dari itu kita membutuhkan authorization dari admin dan masukkan *team_id* dan *user_id* pada request *getteam-member* dengan method POST



Gambar 4. 53 Testing API untuk Member Team

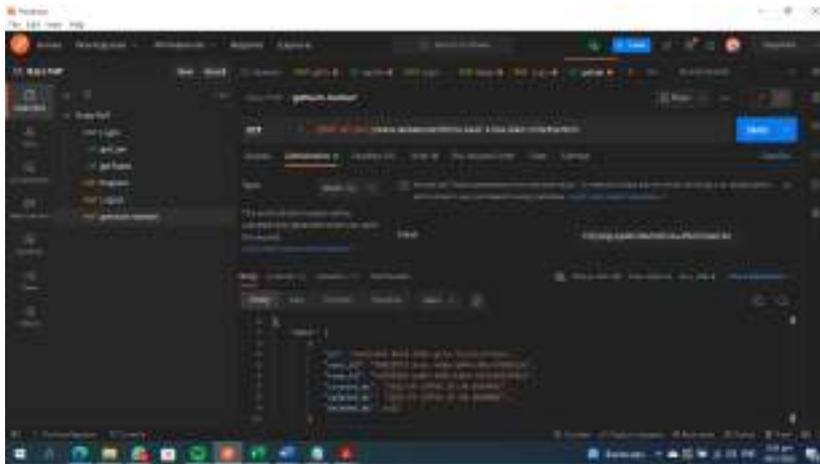
Selanjutnya kita lihat team member pada team yang sudah dibuat menggunakan method GET dengan request getteam-member, masukan team_id pada url dan send



Gambar 4. 54 Testing API menghasilkan Error Ketika Menambahkan Member

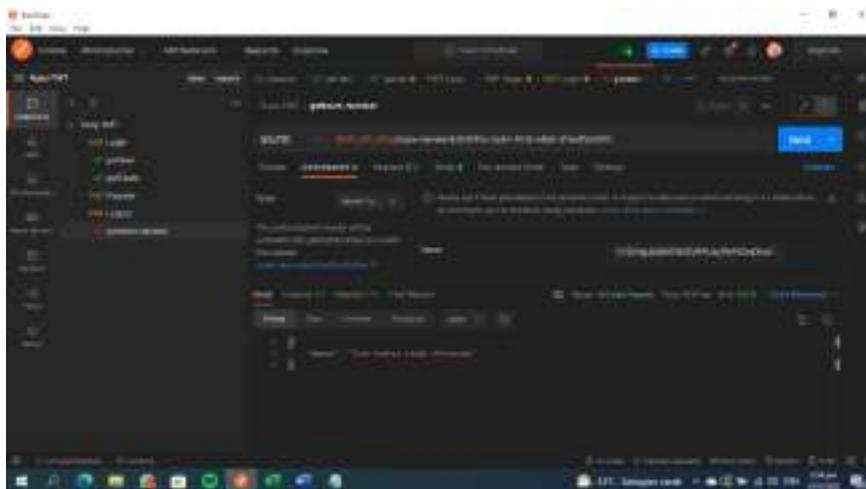
Tampilan diatas masih eror

Setelah source code diperbaiki akan tampil data team-member yang sudah ditambahkan



Gambar 4. 55 Testing API Ketika Team Member sudah diperbaiki

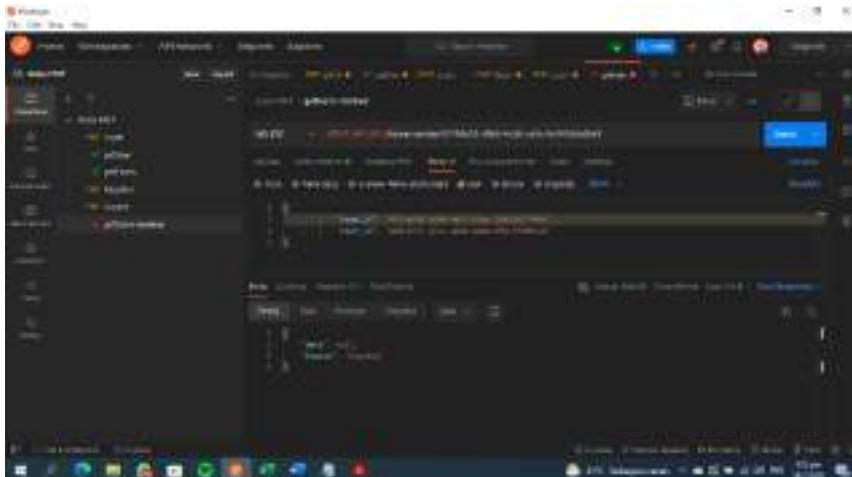
Kemudian kita coba hapus team member yang terdaftar dengan method DELETE pada request getteam-member dan cantumkan id_team-member pada url.



Gambar 4. 56 Testing API menghasilkan Error Ketika Menghapus Member dari Team

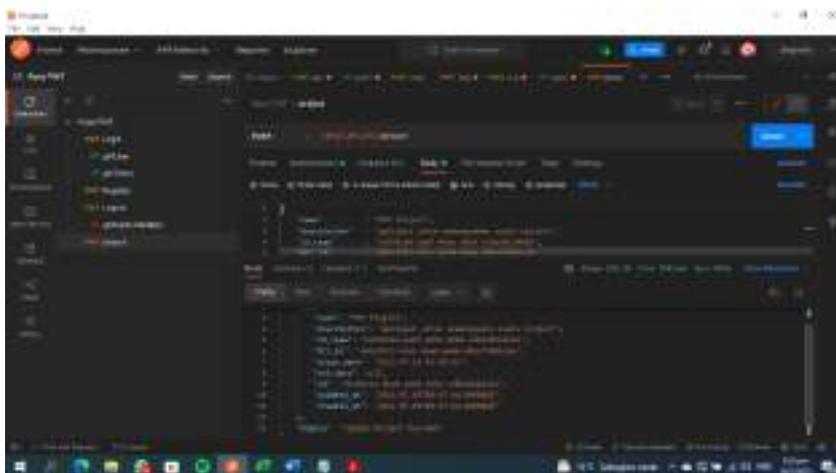
Tampilan diatas masih error

Setelah source code diperbaiki, kita menghapus data team-member yang sudah di buat



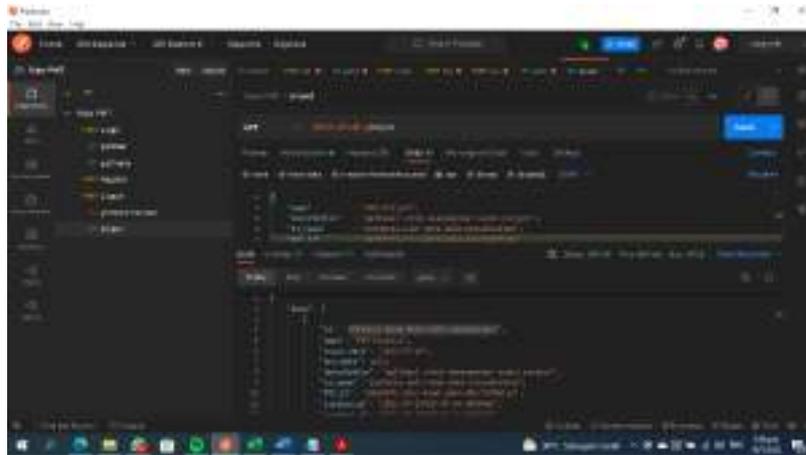
Gambar 4. 57 Testing API Ketika Menghapus Member dari Team

Lalu kita buat new request yaitu project, kemudian kita membuat project baru “PMT Project” dari request project dengan Method POST.



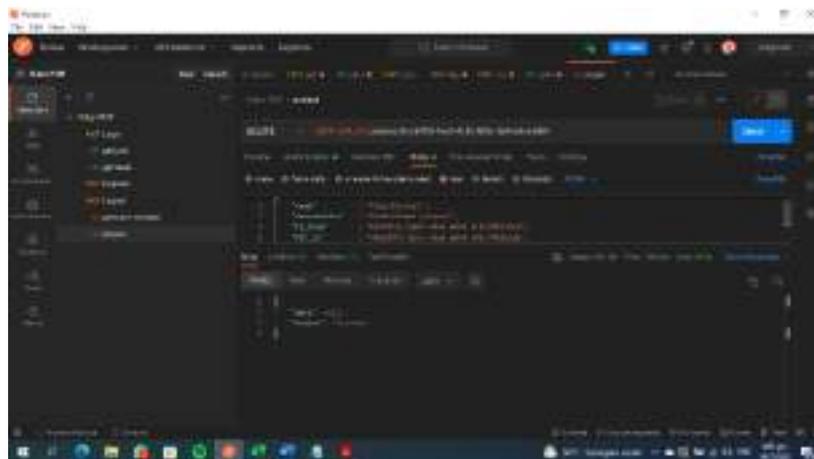
Gambar 4. 58 Testing API untuk Fitur Project

Setelah pembuatan project kita lakukan, kemudian kita akan menampilkan data project yang sudah diinputkan tadi menggunakan method GET dengan request project



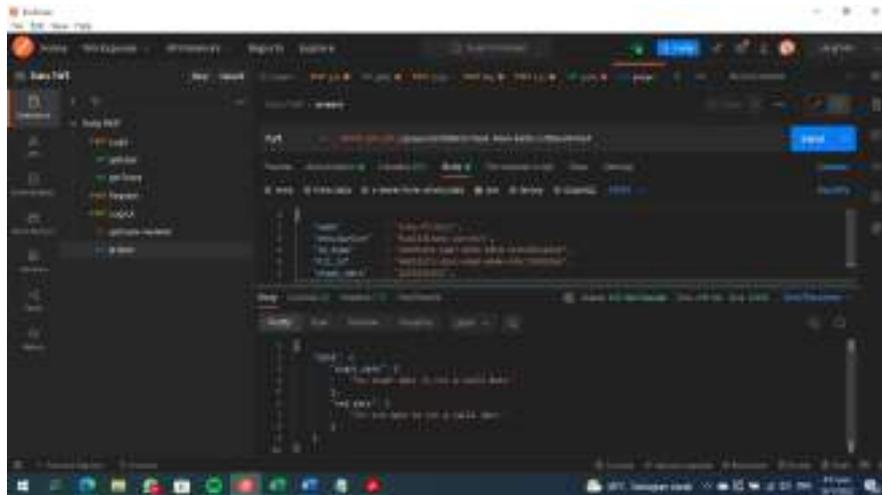
Gambar 4. 59 Testing API untuk Melihat List Project

Setelah tampil data yang sudah dibuat, kita coba menghapus data project yang sudah ada menggunakan method DELETE pada request project dan mencantumkan project_id di url



Gambar 4. 60 Testing API untuk melakukan Hapus Project

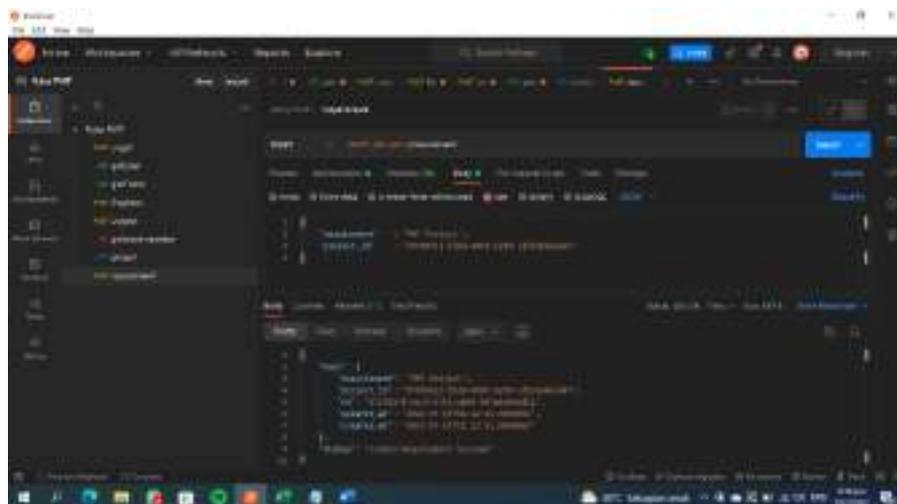
Kemudian kita coba mengupdate data yang ada menggunakan method PUT pada request project. Dari name "PMT project" menjadi "Ruby Team PMT Project"



Gambar 4. 61 Testing API mengalami error Ketika Melakukan Edit Project

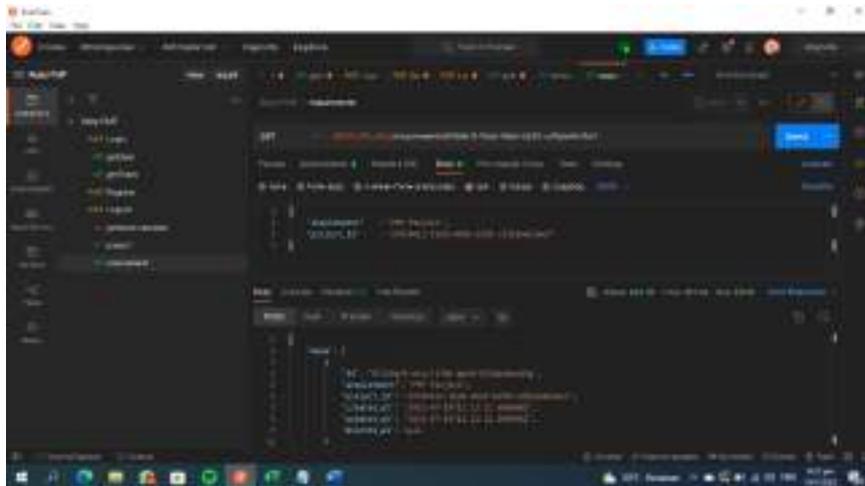
Tampilan diatas masih error

Setelah pembuatan project diatas, kemudian kita membuat request baru yaitu requirement, lalu isi requirement baru menggunakan method POST.



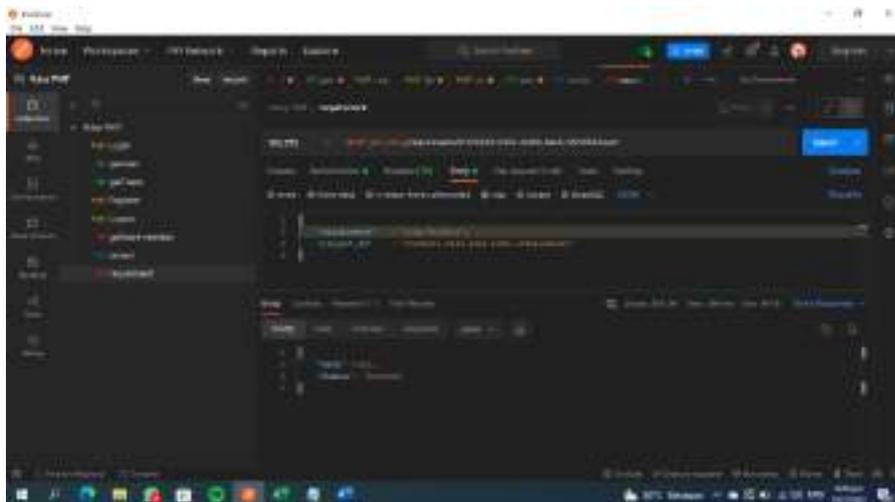
Gambar 4. 62 Testing API untuk Fitur Requirement

Setelah mengisi requirement dengan data selanjutnya kita akan tampilkan data tersebut dengan method GET pada request requirement



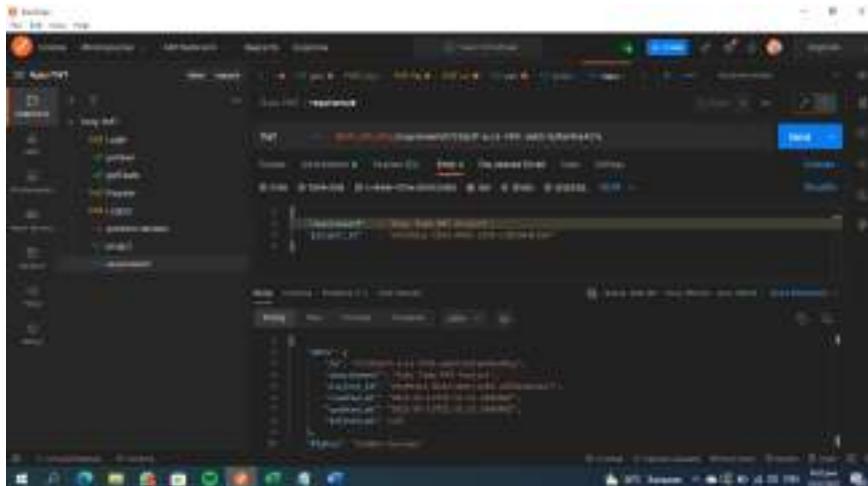
Gambar 4. 63 Testing API untuk Melihat Seluruh Requirement

Setelah itu kita akan menghapus data yang terdaftar pada requirement dengan method DELETE pada request requirement.



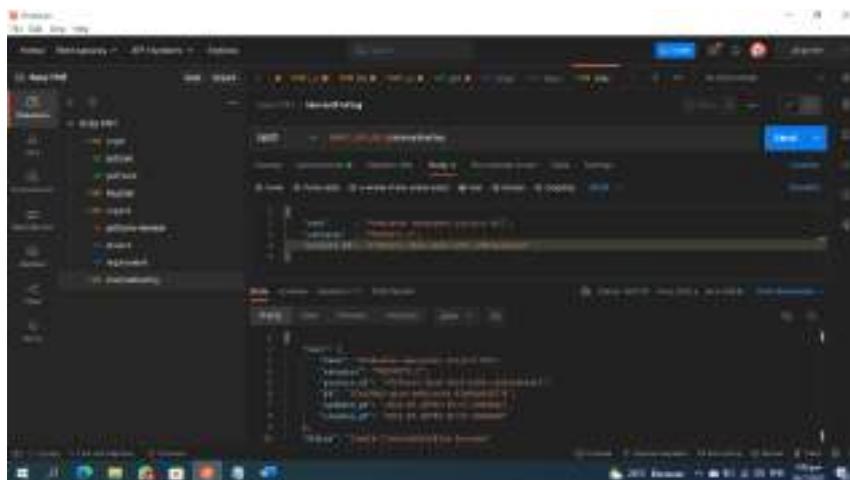
Gambar 4. 64 Testing API untuk Menghapus Requirement

Setelah itu kita mencoba untuk mengupdate data pada requirement, dari requirement “PMT project” menjadi “Ruby Team PMT Project” menggunakan method PUT pada request requirement dan mencantumkan requirement_id pada url



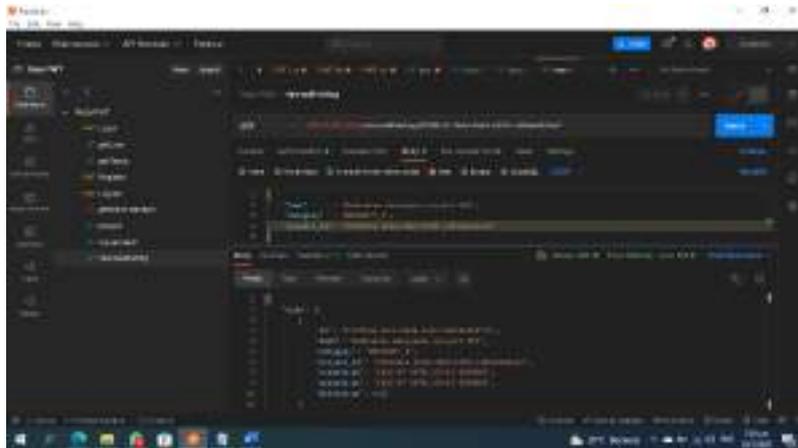
Gambar 4. 65 Testing API untuk Update Requirement

Pembuatan request baru internalBriefing dengan menggunakan method POST untuk membuat isi data internal briefing tersebut,



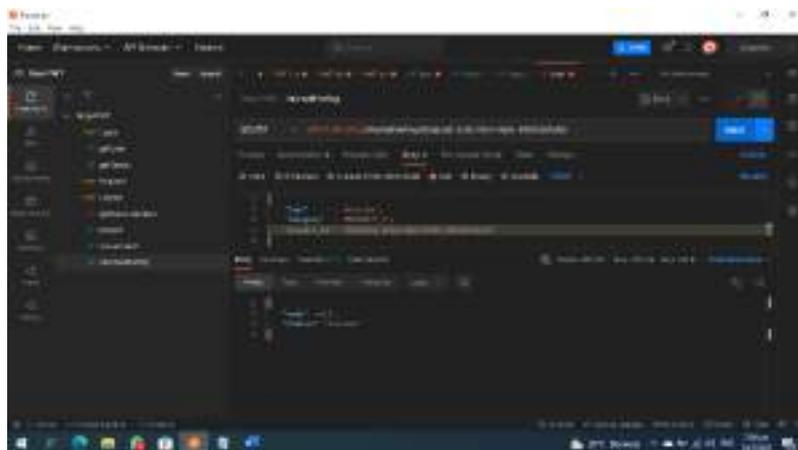
Gambar 4. 66 Testing API untuk Fitur Internal Briefing

Kemudian menampilkan data yang sudah diinputkan tadi menggunakan method POST pada request internalBriefing dan `project_id` di sisipkan pada url



Gambar 4. 67 Testing API untuk Menginput Task Pada Fitur Internal Briefing

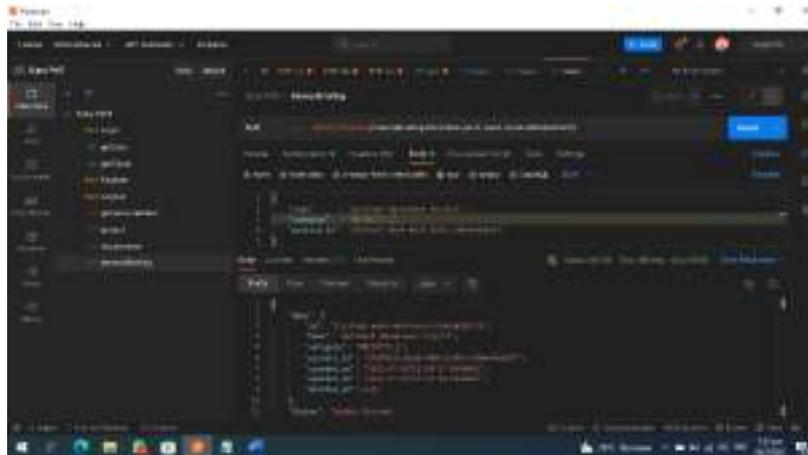
Kemudian kita coba menghapus salah satu data yang sudah diinputkan menggunakan method DELETE pada request internalBriefing dengan menyisipkan id internalBriefing pada url



Gambar 4. 68 Testing API untuk Menghapus Task pada Fitur Internal Briefing

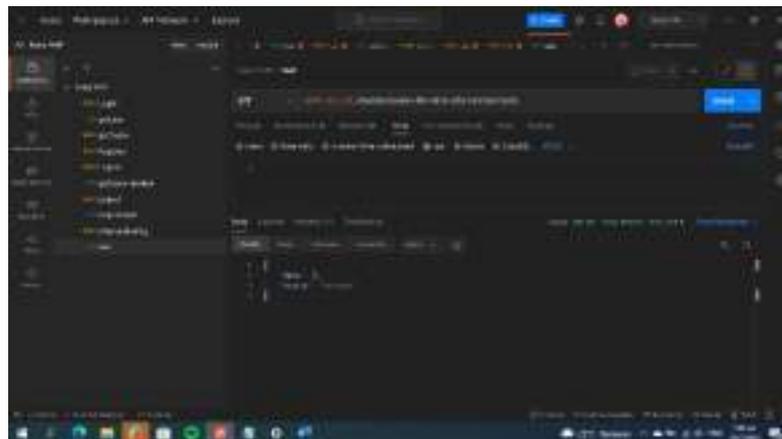
Kemudian kita mencoba mengubah data yang ada menggunakan data yang sudah ada, menggunakan method PUT pada request internalBriefing dan sisipkan id internalBriefingnya pada url

Dari tasknya “pembuatan manajemen project BSI” menjadi “Aplikasi Manajemen Project”



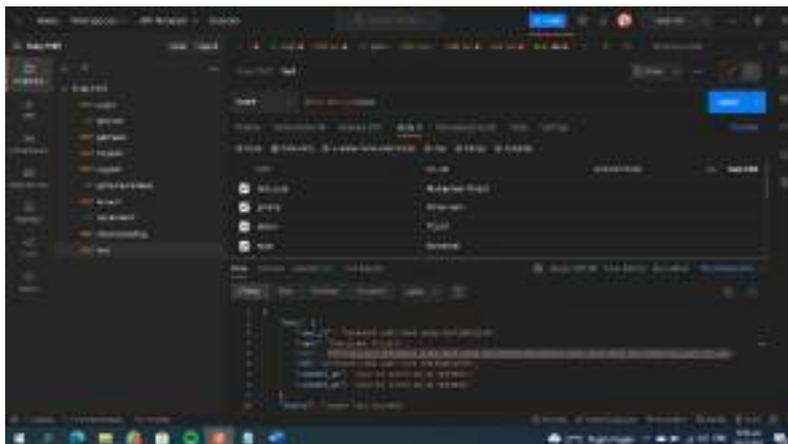
Gambar 4. 69 Testing API untuk Melakukan Edit Task pada Fitur Internal Briefing Update success.

Selanjutnya kita akan membuat request baru yaitu test, mulanya kita akan mengecek dengan request test apakah project success atau tidak, dengan method GET pada request test dan menyisipkan project_id pada url



Gambar 4. 70 Testing API untuk Input Request

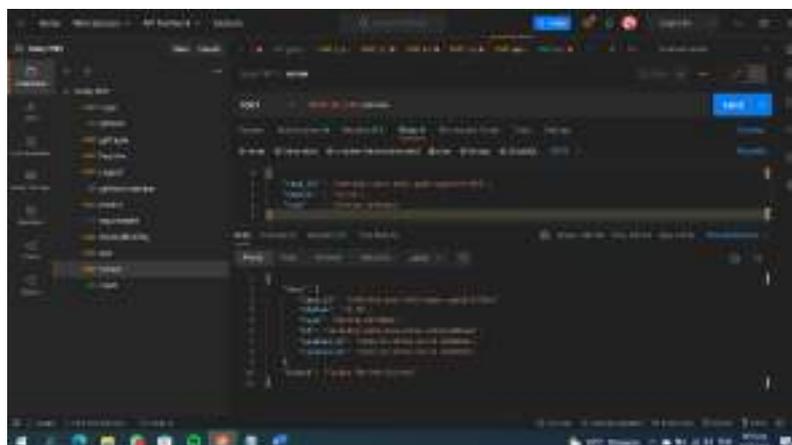
Selanjutnya kita membuat test baru pada request test dengan method POST, di body menggunakan form-data



Gambar 4. 71 Testing API untuk Membuat Test Baru

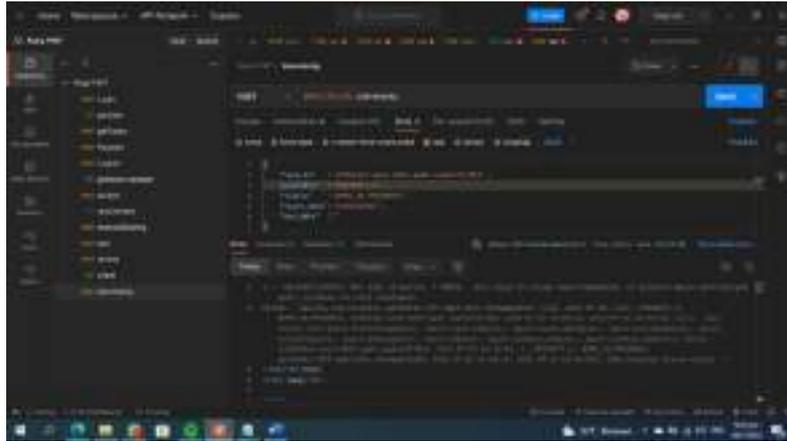
kemudian setelah membuat request test baru dan menginputkan data test kita mencoba untuk melakukan method PUT dan DELETE, yang berfungsi untuk mengubah dan menghapus data test yg sudah diinputkan.

Setelah itu pembuatan request baru yaitu riview, lalu masukan data sesuai dengan docs api.



Gambar 4. 72 Testing API untuk Fitur Riview

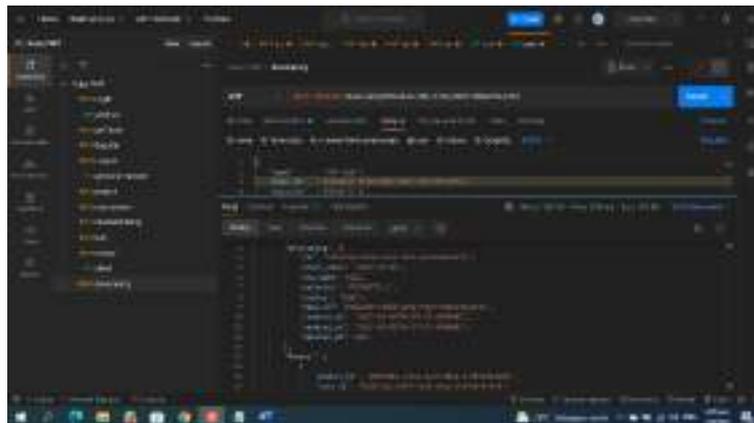
Jika ingin mengubah data tersebut menggunakan method PUT dan masukan id pada url, sama dengan menghapus dengan method DELETE dan juga GET. Setelah itu kita membuat request baru yaitu developing, menggunakan method POST



Gambar 4. 73 Testing API untuk Fitur Developing

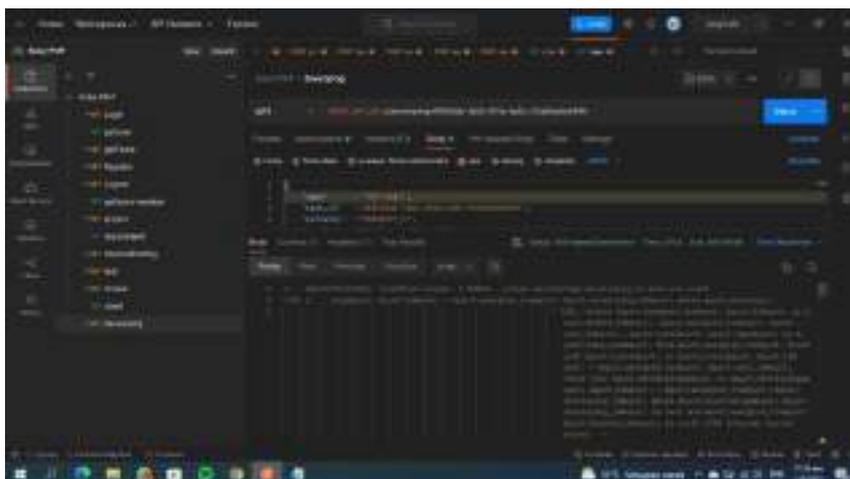
Tampilan diatas masih error ketika memposting data

Setelah di perbaiki



Gambar 4. 74 Testing API untuk Fitur Developing Ketika Setelah Diperbaiki

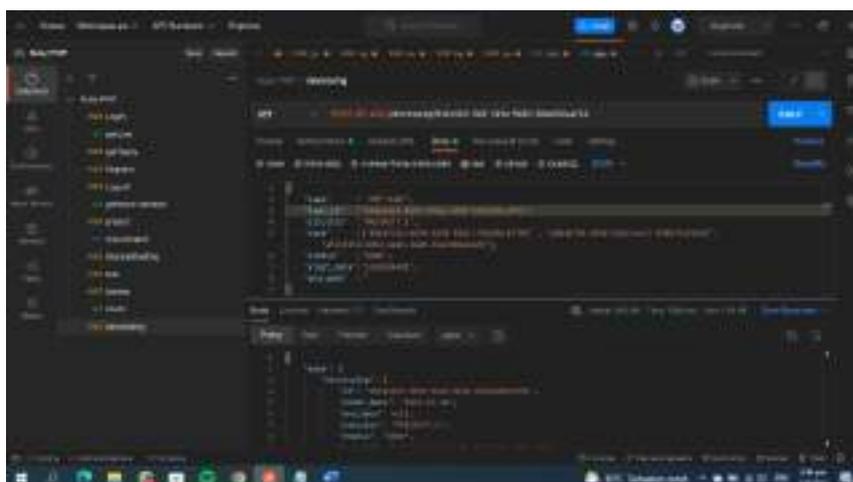
Kemudian menampilkan data developing yang sudah diinputkan



Gambar 4. 75 Testing API untuk Input Data pada Fitur Developing

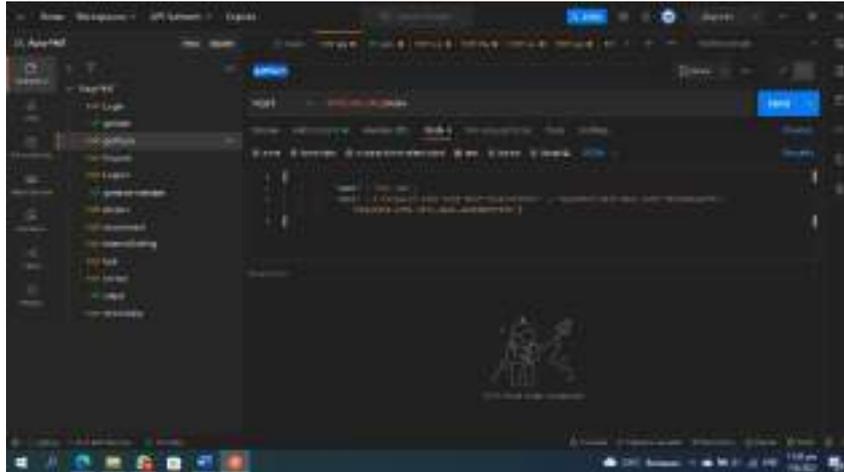
Tampilan diatas masih error, send Method GET tidak menampilkan data developing dan member dari task project

dan setelah diperbaiki



Gambar 4. 76 Testing API Input Data Developing Setelah Diperbaiki

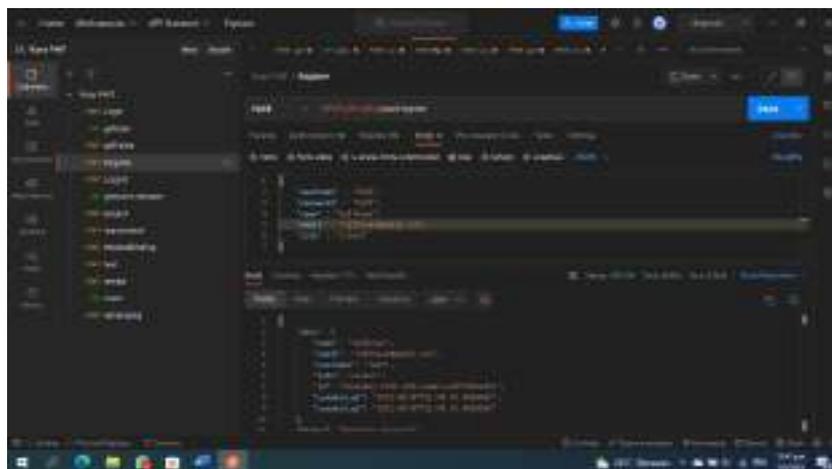
Dari request developing dengan method GET dan menyisipkan developing_id pada url, akan menampilkan data dan member dari team yang mengambil project tersebut.



Gambar 4. 77 Testing API untuk Update Data pada fitur Team

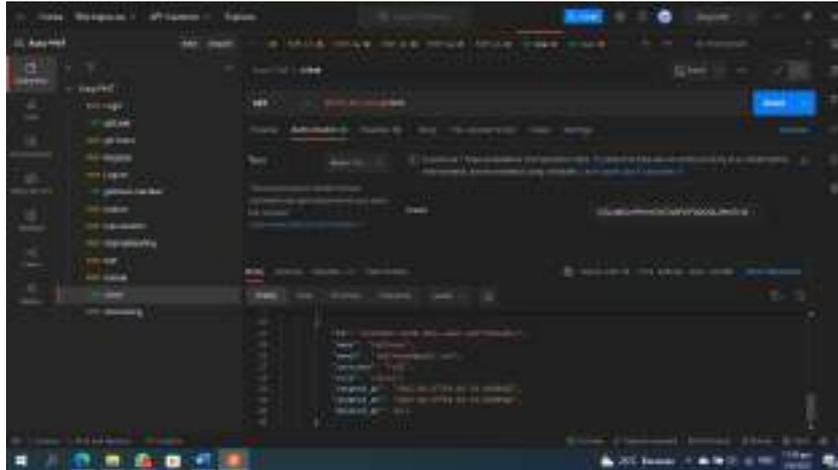
pada request get team terdapat update system, yaitu memasukkan user menggunakan array dengan itu team dapat menampung banyak user.

Kemudian melihat client dengan Method GET pada request client, pertama daftarkan client



Gambar 4. 78 Testing API untuk Melakukan Daftar Akun Client

Setelah itu ke request client dengan method post, dan memasukan token pada authorization lalu send, maka akan tampil seperti berikut



Gambar 4. 79 Testing API untuk Data Client

Jika ingin lebih spesifik masukan client_id pada url dan send, maka akan menampilkan data client sesuai dengan client_id yang di inputkan.

Pengujian URL pada API Test

menampilkan Attachment yang sudah disisipkan. Belum bisa menampilkan data yang disisipkan



Gambar 4. 80 Testing API Test Masih Mengalami error

Dan jika BUG sudah diperbaiki akan menampilkan data yang disisipkan seperti berikut



Gambar 4. 81 Hasil Keseluruhan Testing API

Testing Front end

Menyesuaikan design dari UI/UX

Register



Gambar 4. 82 Testing Tampilan Register

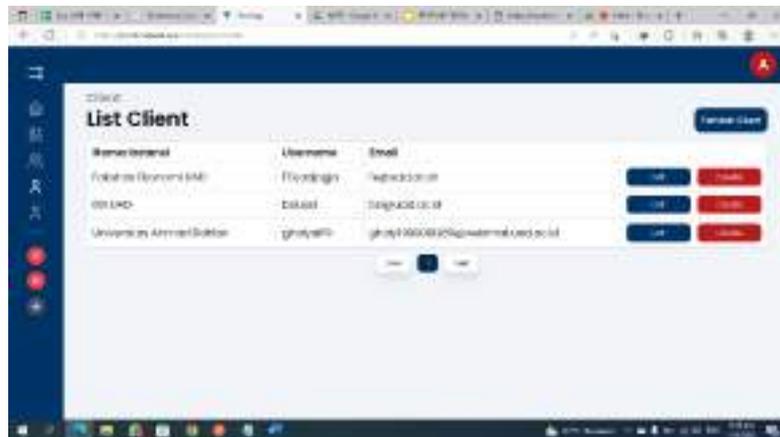
Pendaftaran akun, dari gambar diatas masih belum *redirect* ke laman *login*

Tampilan *List User*



Gambar 4. 83 Testing Tampilan List User

Tampilan List Client



Gambar 4. 84 Testing Tampilan List Client

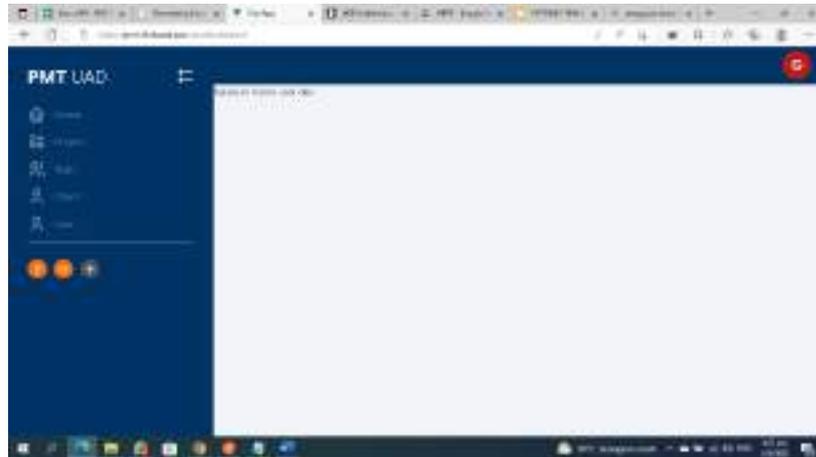
Tampilan Login



Gambar 4. 85 Testing Tampilan Auth Login

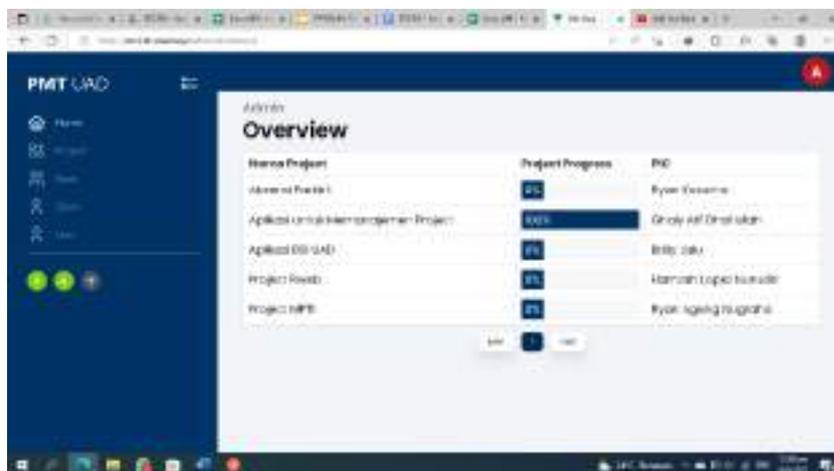
masukkan *username* dan *password* yang sudah di daftarkan, dan tekan masuk. setelah masuk akan *redirect* ke halaman *dashboard admin* atau *user*.

Halaman *dashboard admin* belum menampilkan data atau belum terkoneksi dengan API



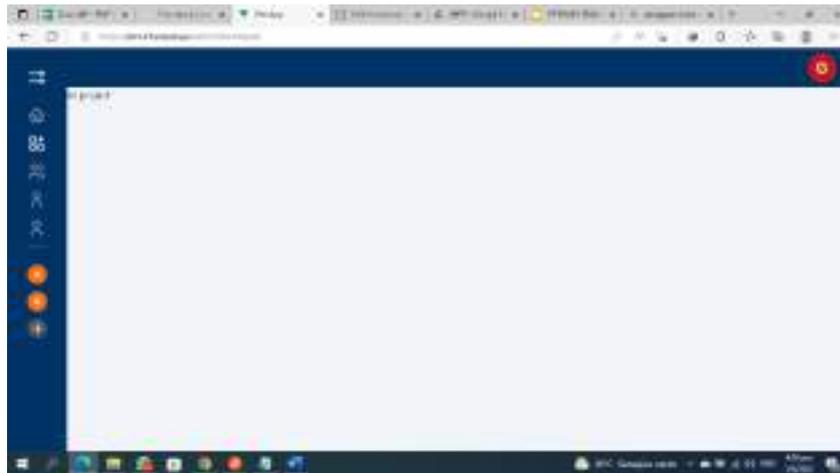
Gambar 4. 86 Testing Tampilan Admin

Halaman *dashboard admin*



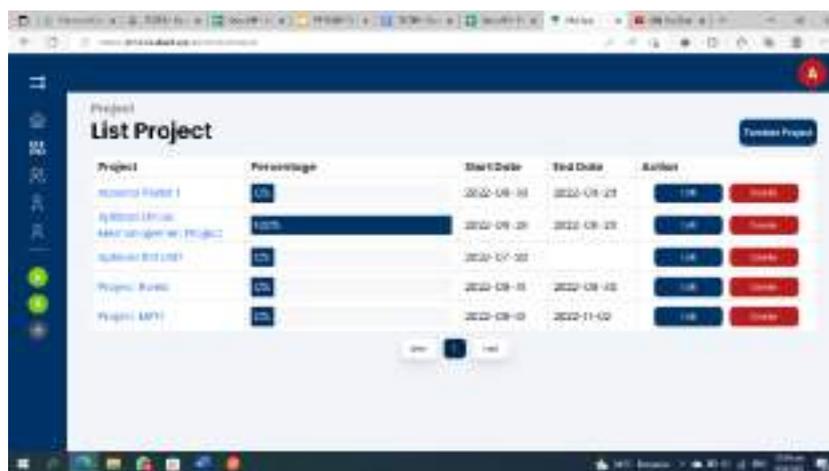
Gambar 4. 87 Testing Tampilan Dashboard Admin

Tampilan *List Project*



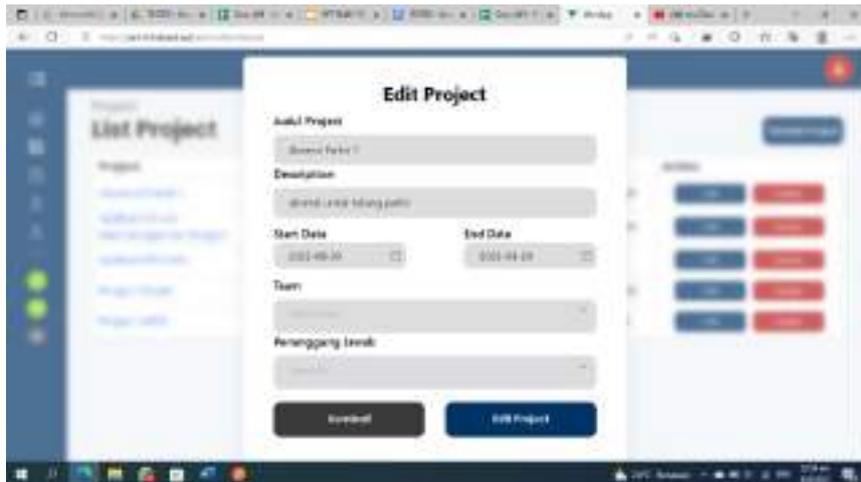
Gambar 4. 88 Testing Tampilan List Project

Tampilan diatas merupakan laman *project* yang belum dikoneksi, kemudian berikut tampilan setelah perbaikan



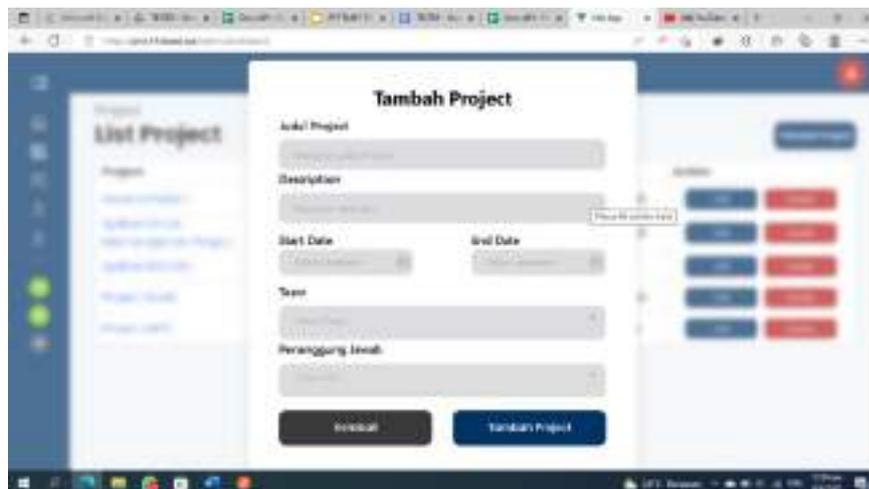
Gambar 4. 89 Testing Tampilan List Project Ketika Sudah diperbaiki

Edit *Project*



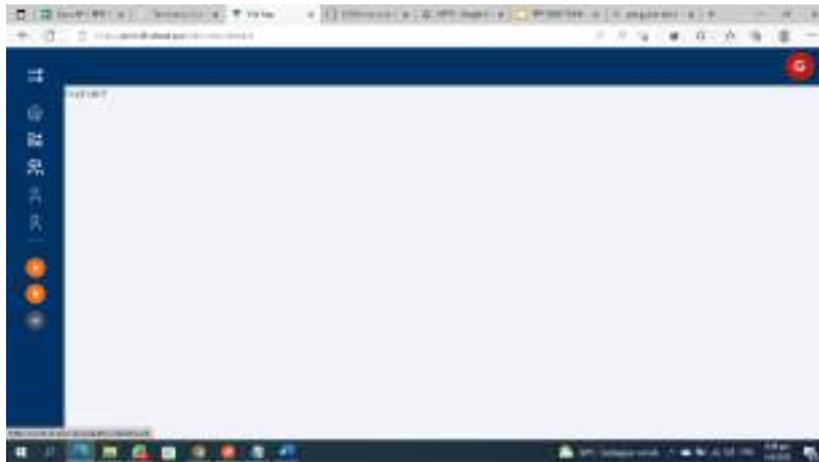
Gambar 4. 90 Testing Tampilan Edit Project

Tambah Project



Gambar 4. 91 Testing Tampilan Tambah Project

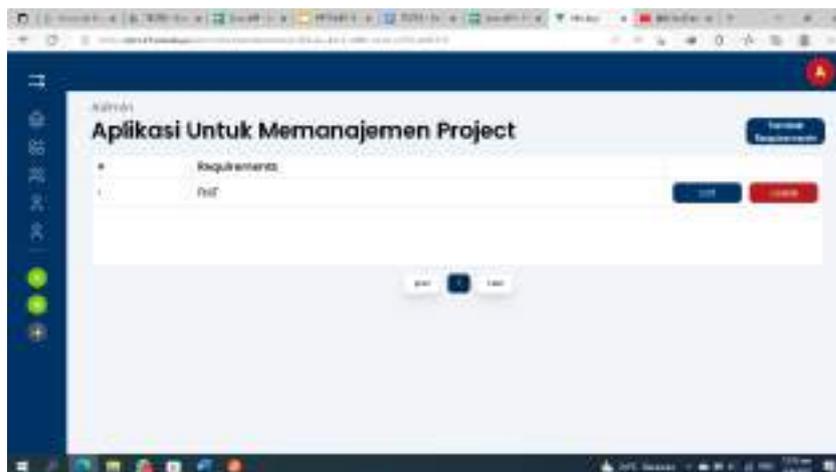
Tampilan Requirement



Gambar 4. 92 Testing Tampilan Requirement Sebelum diperbaiki

tampilan *requirement* terdapat pada laman *project*, dan tampilan belum ada tersedia.

Dan setelah diperbaiki seperti berikut



Gambar 4. 93 Testing Tampilan Requirement Setelah diperbaiki

Edit Requirement

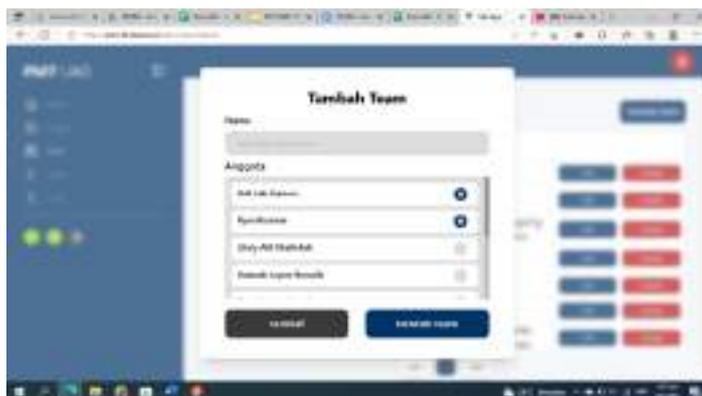
Tambah *team*



Gambar 4. 97 Testing Tampilan Tambah Team

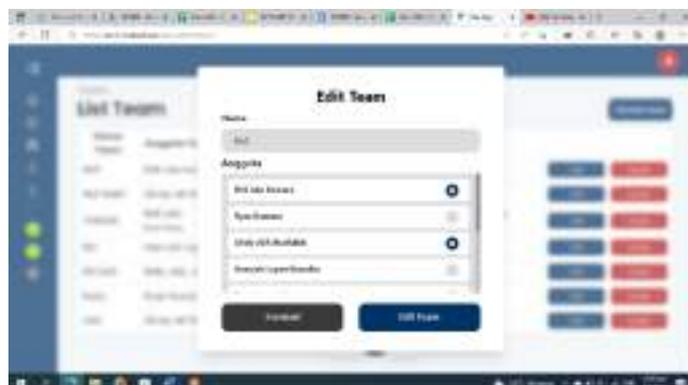
Tampilan diatas *place holder* pada nama *team* tidak sesuai dengan tampilannya

Setelah diperbaiki



Gambar 4. 98 Testing Tampilan Tambah Member

Edit *Team*



Gambar 4. 99 Testing Tampilan Edit Team

Tampilan *List Client*



Gambar 4. 100 Testing Tampilan *List Client*

Tambah *client*



Gambar 4. 101 Testing Tampilan *Tambah Client*

Edit *Client*



Gambar 4. 102 Testing Tampilan *Edit Client*

Tampilan *List User*



Gambar 4. 103 Testing Tampilan List User

Tambah User



Gambar 4. 104 Testing Tampilan Tambah User

Edit User



Gambar 4. 105 Testing Tampilan Edit User

BAB V Penutup

A. Kesimpulan

Kegiatan tugas proyek pembuatan Web Profil telah dilaksanakan di Biro Sistem Informasi, telah menghasilkan aplikasi web yang dapat membantu project yang dikerjakan dari sisi admin, user, client dalam meningkatkan pelayanan untuk Biro Sistem Informasi dan mampu menampilkan proses manajemen kerja dari staf Biro Sistem Informasi.

B. Saran

Aplikasi *Project Management Tools* ini masih perlu pengembangan seperti:

1. Menambahkan notifikasi agar menjadi pengingat dan pemberitahuan
2. Penambahan *reset password* ke email untuk memulihkan kembali *password* yang lupa atau akun terekspos.
3. Menambahkan tingkat keamanan yang lebih kuat sehingga tidak terjadi peretasan atau *hacking* pada sistem.

DAFTAR PUSTAKA

Muhammad, Andi Fadel. 2018. ISI.pdf. <http://eprints.uad.ac.id/11569/14/ISI.pdf>

. 14 Agustus 2018

Biro Sistem Informasi. bsi.uad.ac.id/ . 2021. Profil Biro Sistem Informasi. Diakses pada 19 Juli 2022, dari <https://bsi.uad.ac.id/>