

LAPORAN KERJA PRAKTIK
PEMBUATAN WEBSITE SIMCARATA
PIMPINAN CABANG MUHAMMADIYAH TEGALREJO
MENGGUNAKAN FRAMEWORK LARAVEL

Dosen Pembimbing : Ir. Asroni, S.T., M.Eng.



Disusun oleh:

Mochammad Reza Rafli Zamzam Alfarizi

20180140102

PROGRAM STUDI TEKNOLOGI INFORMASI
FAKULTAS TEKNIK
UNIVERSITAS MUHAMMADIYAH YOGYAKARTA
2021/2022

**LEMBAR PENGESAHAN
LAPORAN KERJA PRAKTIK**

1	Judul Kerja Praktik	Pembuatan Website Simcarata Pimpinan Cabang Muhammadiyah Tegalrejo Menggunakan Framework Laravel
2	Tempat Pelaksanaan	Pimpinan Cabang Muhammadiyah Tegalrejo
3	Waktu Pelaksanaan	2 September 2021 – 18 November 2021
4	Pelaksana Kerja Praktik	
	a. Nama Lengkap	Mochammad Reza Rafli Zamzam Alfarizi
	b. No. Mahasiswa	20180140102
	c. Universitas	Universitas Muhammadiyah Yogyakarta
	d. Fakultas	Fakultas Teknik
	e. Program Studi	Teknologi Informasi
	f. No. Telp/HP	089674344282
5	Dosen Pembimbing	
	a. Nama Lengkap dan Gelar	Ir. Asroni, S.T., M.Eng.
	b. NIDN	0526047401

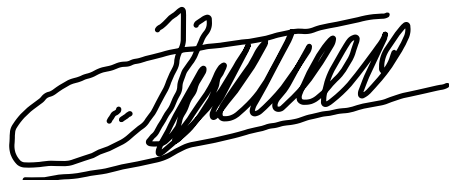
Yogyakarta, 14 December 2021

Dosen Pembimbing Selaku Ketua Jurusan
Teknologi Informasi

Pelaksana Kerja Praktik



Ir. Asroni, S.T., M.Eng.
NIDN : 0526047401



Mochammad Reza Rafli Z.A
NIM : 20180140102

KATA PENGANTAR

Assalamu'alaikum Warahmatullahi Wabarakatuh

Alhamdulillah puji syukur kami panjatkan kehadirat Allah Subhanahu Wa Ta'ala yang telah memberikan rahmat dan karunia-Nya sehingga kami dapat melaksanakan dan menyelesaikan Kerja Praktik ini dengan baik. Sholawat beriring salam tidak lupa kami curahkan kepada Nabi Muhammad Shallallahu 'Alaihi Wasallam yang telah membawa kita umat muslim dari zaman kegelapan menuju zaman terang benderang yaitu agama Islam, semoga kita termasuk umatnya yang kelak mendapatkan syafa'at-Nya.

Pada kesempatan kali ini, kami mengucapkan terimakasih kepada bapak Ir. H. Gani Supriyanto, MP. selaku Ketua Pimpinan Cabang Muhammadiyah Tegalrejo atas kesempatan yang telah diberikan dalam pelaksanaan Kerja Praktik ini, Bapak Ir. Asroni, S.T., M.Eng. selaku Dosen Pembimbing.

Laporan Kerja Praktik ini, kami susun berdasarkan apa yang telah kami laksanakan selama Kerja Praktik ini yang dilaksanakan mulai tanggal 2 September 2021 – 18 November 2021.

Dalam penyusunan laporan kerja praktik ini, kami menyadari masih banyak kekurangan baik dari segi susunan serta cara penulisan laporan ini, karenanya saran dan kritik yang sifatnya membangun demi kesempurnaan laporan ini sangat kami harapkan.

Akhir kata, semoga laporan akhir ini bisa bermanfaat bagi para pembaca pada umumnya dan juga bermanfaat bagi penyusun pada khususnya.

Wassalamu'allaikum Warahmatullahi Wabarakaatuh

Yogyakarta, 10 Oktober 2021

Penyusun,

Mochammad Reza Rafli Z.A

DAFTAR ISI

LEMBAR PENGESAHAN	2
KATA PENGANTAR.....	3
DAFTAR ISI.....	4
DAFTAR GAMBAR.....	6
DAFTAR TABEL	7
BAB 1 PENDAHULUAN	8
A. Latar Belakang.....	8
B. Waktu dan Tempat Pelaksanaan.....	8
C. Maksud dan Tujuan	10
D. Manfaat.....	10
BAB II PROFIL ORGANISASI.....	11
A. Gambaran Umum Instansi.....	11
B. Struktur Organisasi	11
C. Visi.....	11
D. Misi.....	11
E. Tujuan.....	12
F. Sasaran.....	12
BAB III LANDASAN TEORI.....	13
A. Sistem Informasi	13
B. PHP.....	13
C. Framework	13
D. MySQL.....	14
E. Laravel 8.....	14
F. Laravel Voyager.....	14
G. HTML	16
H. CSS	16
BAB IV PERANCANGAN.....	17
A. Analisis Kebutuhan.....	17
B. UML	19
BAB V IMPLEMENTASI.....	23
A. Halaman Public.....	23
B. Halaman Pimpinan Cabang.....	32
C. Halaman Pimpinan Ranting	38

D. Halaman Admin	40
BAB VI PENUTUP	50
A. Kesimpulan	50
B. Saran	50
DAFTAR PUSTAKA	51
LAMPIRAN	52
A. Controller	52
B. Model	61
C. View	75
D. Route	86

DAFTAR GAMBAR

Gambar 1 Use Case Diagram.....	19
Gambar 2 Class Diagram.....	21
Gambar 3 Database Diagram.....	22
Gambar 4 Public – Beranda.....	23
Gambar 5 Public - Post.....	24
Gambar 6 Public - Profil.....	25
Gambar 7 Public - Pimpinan.....	26
Gambar 8 Public – Ranting.....	27
Gambar 9 Public – AUM.....	28
Gambar 10 Public – Anggota.....	29
Gambar 11 Public – Dokumen.....	30
Gambar 12 Public – Login.....	31
Gambar 13 Cabang – Dashboard.....	32
Gambar 14 Cabang – Pimpinan.....	33
Gambar 15 Cabang – Anggota.....	34
Gambar 16 Cabang – AUM.....	35
Gambar 17 Cabang – Aset.....	36
Gambar 18 Cabang - Data Master.....	37
Gambar 19 Ranting – Dashboard.....	38
Gambar 20 Ranting – Anggota.....	39
Gambar 21 Admin – Dashboard.....	40
Gambar 22 Admin – Post.....	41
Gambar 23 Admin – Pimpinan.....	42
Gambar 24 Admin – Anggota.....	43
Gambar 25 Admin – AUM.....	44
Gambar 26 Admin – Aset.....	45
Gambar 27 Admin – Akun.....	46
Gambar 28 Admin - Data Master.....	47
Gambar 29 Admin – Tool.....	48
Gambar 30 Admin – Settings.....	49

DAFTAR TABEL

Tabel 1 – Waktu pelaksanaan 9

BAB 1 PENDAHULUAN

A. Latar Belakang

Pembuatan software Sistem Informasi Cabang Ranting Tegalrejo (SIMCARATA) ini ditujukan untuk mempermudah pendataan anggota Muhammadiyah cabang Tegalrejo sekaligus untuk mempercepat proses pengelolaan data anggota dan aset yang semakin lama semakin berkembang. Pemanfaatan Teknologi Informasi sebagai alat bantu pengelolaan informasi telah lama digunakan dibanyak instansi, namun untuk tingkat Pimpinan Cabang Muhammadiyah masih sedikit yang menggunakan software Sistem Informasi sebagai salah satu sarana untuk pengelolaan data mereka.

Aplikasi Sistem Informasi Cabang Ranting Tegalrejo (SIMCARATA) ini kami buat berbasis website untuk Pimpinan Cabang dan Pimpinan Ranting serta Umum sehingga proses pendataan dapat dilakukan dengan cepat. Pada akhirnya diharapkan dengan menerapkan Sistem Informasi Cabang Ranting Tegalrejo (SIMCARATA) di Tegalrejo ini akan meningkatkan kualitas pelayanan dan efisiensi pengolahan serta pengelolaan data.

B. Waktu dan Tempat Pelaksanaan

1. Tempat pelaksanaan

Pelaksanaan kegiatan kerja praktik ini dilaksanakan di:

Nama Perusahaan : Pimpinan Cabang Muhammadiyah Tegalrejo

Alamat Perusahaan : Pondok Pesantren Tahfidzil Qur'an Muhammadiyah Ibnu Juraimi
Yogyakarta Kricak, Kec.Tegalrejo, Kota Yogyakarta, Daerah Istimewa Yogyakarta
55242

2. Waktu pelaksanaan

Pelaksanaan kegiatan kerja praktik ini dilaksanakan selama 48 hari kerja. Dimulai dari tanggal 2 September 2021 hingga 18 November 2021

No	Item	Agenda	September				Oktober				November	
			Pekan 1	Pekan 2	Pekan 3	Pekan 4	Pekan 1	Pekan 2	Pekan 3	Pekan 4	Pekan 1	Pekan 2
1	Pertemuan dengan PCM	Koordinasi Sistem	V									
2	Meeting Development	Pemilihan Framework		V								
3	Meeting Development	Pembagian tugas			V							
4	Meeting Development	Coding				V						
5	Meeting Development	Coding					V					
6	Pertemuan dengan PCM	Trial Web dan Paparan						V				
7	Meeting Development	Perbaikan masukan dari PCM							V			
10	Pertemuan dengan PCM	Pelatihan dan Implementasi Web								V		
11	Meeting Development	Penyempurnaan Web									V	
12	Pertemuan dengan PCM	Implementasi dan entri data / Final										V

Tabel 1 – Waktu pelaksanaan

C. Maksud dan Tujuan

Dengan dilaksanakan kerja praktik ini, penulis berharap mendapatkan pengalaman dan wawasan ilmu untuk masuk ke dalam dunia kerja. Adapun maksud dan tujuan yang hendak dicapai dalam pelaksanaan Kerja Praktik ini antara lain:

1. Mengamati hubungan antara teori mata kuliah yang telah dipelajari dengan penerapan di lapangan saat Kerja Praktik.
2. Meningkatkan kemampuan untuk menerapkan pengetahuan dan mengasah keterampilan pribadi dalam lingkungan kerja sesungguhnya.
3. Melatih rasa tanggung jawab dan mentalitas mahasiswa dalam bekerja.
4. Meningkatkan kemampuan berkomunikasi, kreatif dan inovatif ketika sedang melaksanakan Kerja Praktik.
5. Mempersiapkan sumber daya manusia yang mampu bersaing dalam dunia kerja kelak setelah selesai menempuh masa studi di Universitas Muhammadiyah Yogyakarta.
6. Memenuhi syarat untuk menyelesaikan program studi Teknologi Informasi di Universitas Muhammadiyah Yogyakarta
7. Membangun Website Simcarata untuk meningkatkan pelayanan PCM Tegalrejo Kota Yogyakarta.

D. Manfaat

Dengan dilaksanakannya kegiatan kerja praktik ini diperoleh manfaat antara lain:

1. Bagi Mahasiswa
 - a. Mendapatkan kesempatan untuk menerapkan ilmu-ilmu yang diperoleh di perkuliahan untuk mendapatkan pemahaman mengenai dunia kerja.
 - b. Menjalin kerjasama yang baik dalam sebuah tim untuk menyelesaikan proyek website.
 - c. Mengembangkan pola pikir serta menganalisa permasalahan yang ada
 - d. Mengembangkan ide yang kreatif serta inovatif.
2. Bagi Universitas
 - a. Meningkatkan kualitas lulusannya melalui pengalaman kerja praktik.
 - b. Meningkatkan kerjasama antar universitas dengan pimpinan cabang muhammadiyah.
3. Bagi Perusahaan
 - a. Mempunyai sistem informasi manajemen Simcarata.
 - b. Dapat meningkatkan pelayanan melalui sistem informasi.

BAB II PROFIL ORGANISASI

A. Gambaran Umum Instansi

Pimpinan Cabang Muhammadiyah Tegalrejo merupakan jenjang struktural Muhammadiyah pada tingkat kecamatan. Dalam level yang lebih tinggi dari Pimpinan Ranting Muhammadiyah, Pimpinan Cabang Muhammadiyah mempunyai fungsi koordinatif bagi seluruh Pimpinan Muhammadiyah yang ada di wilayah kecamatan tersebut, sekaligus mengkoordinasikan berbagai bentuk seperti aktivitas keagamaan, pendidikan kesejahteraan sosial, kesehatan dsb. Pimpinan Cabang Muhammadiyah Tegalrejo mengelola Pimpinan Ranting Muhammadiyah (PRM) dan Pimpinan Ranting Aisyiyah (PRA), yang terdiri dari 10 PRM dan 11 PRA.

PCM Tegalrejo berlokasi di Pondok Pesantren Tahfidzil Qur'an Muhammadiyah Ibnu Juraimi Yogyakarta Kricak, Kec.Tegalrejo, Kota Yogyakarta, Daerah Istimewa Yogyakarta 55242.

B. Struktur Organisasi

Berikut ini struktur organisasi Pimpinan Cabang Muhammadiyah Tegalrejo.

Ketua	: Ir. Gani Supriyanto
Wakil Ketua	: Agus Sutrisno S.Sos.I
Sekretaris	: Ir. Agung Budiarmo
Bendahara	: H. Wahyudi B.A

C. Visi

Visi Pimpinan Cabang Muhammadiyah Tegalrejo.

Muhammadiyah sebagai gerakan Islam yang berlandaskan Al-Qur'an dan As-Sunnah dengan watak tajdid yang dimilikinya senantiasa istiqomah dan aktif dalam melaksanakan dakwah Islam amar ma'ruf nahi munkar di semua bidang dalam upaya mewujudkan Islam sebagai rahmatan lil 'alamin menuju terciptanya atau terwujudnya masyarakat Islam yang sebenar-benarnya.

D. Misi

Misi Pimpinan Cabang Muhammadiyah Tegalrejo. Muhammadiyah sebagai gerakan Islam, dakwah amar ma'ruf nahi munkar yang memiliki misi:

1. Menegakkan keyakinan tauhid yang murni sesuai dengan ajaran Allah SWT yang dibawa oleh para Rasul sejak Nabi Adam as, hingga Nabi Muhammad saw.
2. Memahami agama dengan menggunakan akal fikiran sesuai dengan jiwa ajaran Islam untuk menjawab dan menyelesaikan persoalan - persoalan kehidupan.
3. Menyebarkan ajaran Islam yang bersumber pada Al-Qur'an sebagai kitab Allah SWT terakhir dan Sunnah Rasul untuk pedoman hidup umat manusia.
4. Mewujudkan amalan - amalan Islam dalam kehidupan pribadi, keluarga dan masyarakat.

E. Tujuan

Berikut ini tujuan dari pembuatan *website* SIMCARATA (Sistem Informasi Cabang Ranting Tegalrejo):

1. Mewujudkan tertib administrasi diwilayah Pimpinan Cabang Muhammadiyah Tegalrejo.
2. Membuat mapping digital warga muhammadiyah diwilayah Pimpinan Cabang Muhammadiyah Tegalrejo.
3. Penyediaan database warga muhammadiyah diwilayah Pimpinan Cabang Muhammadiyah Tegalrejo.
4. Terwujudnya pengelolaan sistem informasi Cabang dan Ranting Muhammadiyah Tegalrejo. Mulai data warga muhammadiyah per ranting dll.

F. Sasaran

Berikut ini sasaran dari *website* SIMCARATA:

1. Penggunaan internal dalam ruang lingkup PCM Tegalrejo.
2. Penggunaan eksternal dalam publikasi kegiatan yang ada di PCM Tegalrejo.

BAB III LANDASAN TEORI

A. Sistem Informasi

Sistem informasi (SI) secara umum adalah suatu sistem yang mengkombinasikan antara aktivitas manusia dan penggunaan teknologi untuk mendukung manajemen dan kegiatan operasional. Hal tersebut merujuk pada sebuah hubungan yang tercipta berdasarkan interaksi manusia, data, informasi, teknologi, dan algoritma.

Penerapan sistem informasi tidak hanya diimplementasikan pada bidang Teknologi Informasi dan Komunikasi (TIK) saja, namun kebutuhan proses bisnis lain juga sangat membutuhkan kontrol dari SI. Sehingga, sistem informasi terbentuk sebagai tipe khusus dari proses kerja.

Penggunaan dari SI sendiri ditujukan untuk mengolah berbagai informasi yang dikelola oleh setiap perusahaan atau organisasi, sehingga sumber daya atau *resources* yang dibutuhkan tidak terlalu besar dan dapat mempersingkat waktu penanganan proses. Selain itu, data yang dikelola juga dapat digunakan kapan saja dan dimana saja, serta mampu mempersingkat birokrasi yang ada. (Adani, 2021)

B. PHP

PHP atau *Hypertext Preprocessor* adalah bahasa pemrograman yang berjalan pada *server side scripting* dan bersifat *open source*. Penggunaan dari bahasa ini sering digunakan untuk pembuatan aplikasi berbasis website yang berjalan secara dinamis, sehingga dapat terintegrasi dengan basis data (*database*).

PHP banyak digunakan untuk pembuatan website untuk kebutuhan ecommerce, sistem informasi, maupun landing page. Karena PHP tergolong dalam bahasa pemrograman berbasis *server-side*, maka script yang digunakan akan diproses oleh server. (Adani, 2021)

C. Framework

Framework adalah kerangka kerja untuk mengembangkan aplikasi berbasis website maupun desktop. Kerangka kerja disini sangat membantu developer dalam menuliskan sebuah dengan lebih terstruktur dan tersusun rapi. Kerangka kerja diciptakan untuk mempermudah kinerja dari programmer. Sehingga, seorang programmer tidak perlu untuk menuliskan *code* secara berulang-ulang.

Fungsi *framework* adalah membuat *source code* menjadi lebih terstruktur. Terstruktur disini, berarti program yang akan dimasukkan ke setiap komponen sesuai dengan fungsinya masing-masing. (Adani, 2020)

D. MySQL

MySQL adalah sebuah DBMS (*Database Management System*) menggunakan perintah SQL (*Structured Query Language*) yang banyak digunakan saat ini dalam pembuatan aplikasi berbasis website. MySQL dibagi menjadi dua lisensi, pertama adalah Free Software dimana perangkat lunak dapat diakses oleh siapa aja. Dan kedua adalah Shareware dimana perangkat lunak memiliki batasan dalam penggunaannya.

MySQL termasuk ke dalam RDBMS (*Relational Database Management System*). Sehingga, menggunakan tabel, kolom, baris, di dalam struktur databasenya. Jadi, dalam proses pengambilan data menggunakan metode relational database. Dan juga penghubung antara perangkat lunak dan database server.

Fungsi dari MySQL adalah untuk membuat dan mengelola database pada sisi server yang memuat berbagai informasi dengan menggunakan bahasa SQL. Fungsi lain yang dimiliki adalah memudahkan pengguna dalam mengakses data berisi informasi dalam bentuk String (teks), yang dapat diakses secara personal maupun publik dalam web. (Adani, 2020)

E. Laravel 8

Laravel adalah framework PHP yang bersifat open source dan dirancang agar proses pengembangan aplikasi web menjadi lebih mudah dan cepat karena berbagai fitur bawaan di dalamnya. Fitur-fitur tersebut merupakan salah satu alasan mengapa banyak pengembangan memanfaatkan kerangka kerja Laravel.

Laravel digunakan oleh para pengembang sisi server atau back end. Back end developer dapat mengembangkan fitur-fitur yang dibutuhkan pada sisi server seperti user accounts, export, order management, dan lain-lain. (NKD, 2020)

F. Laravel Voyager

Voyager merupakan Laravel *package* untuk membuat admin interface di aplikasi Laravel. Voyager memudahkan *developer* yang menggunakan Laravel untuk mengelola *media manager, menu builder, database manager, serta bread manager*. (Ribeiro, 2017)

G. HTML

HTML atau *Hypertext Markup Language* adalah bahasa markup yang digunakan untuk membuat struktur halaman website agar dapat ditampilkan pada web browser. HTML dapat dianalogikan sebagai pondasi awal dalam menyusun kerangka halaman web secara terstruktur sebelum membahas terkait tampilan dan sisi fungsionalitas.

Fungsi utama penggunaan HTML sendiri adalah membangun tampilan website yang telah menerapkan metode semantik untuk memudahkan setiap pengembang dalam proses *development* dan *maintenance*. Kemudian, HTML juga dapat dikolaborasikan dengan penggunaan bahasa CSS (*Cascade Style Sheet*) serta JavaScript. (Adani, 2021)

H. CSS

CSS atau *Cascading Style Sheets* adalah kumpulan perintah yang digunakan untuk menjelaskan tampilan sebuah situs web dalam mark-up language. CSS akan membantu para web designer untuk mengubah tampilan teks (baik dari bentuk dan ukuran font maupun warnanya), menambahkan gambar, hingga mengubah latar belakang sebuah halaman HTML. (Dewaweb Team, 2021)

BAB IV PERANCANGAN

A. Analisis Kebutuhan

1. Kebutuhan Functional

a. Admin

- a) Dapat Login
- b) Dapat mengakses Dashboard
- c) Dapat melakukan operasi CRUD dalam Post
- d) Dapat melakukan operasi CRUD dalam Pimpinan
- e) Dapat melakukan operasi CRUD dalam Anggota
- f) Dapat melakukan operasi CRUD dalam AUM
- g) Dapat melakukan operasi CRUD dalam Aset
- h) Dapat melakukan operasi CRUD dalam Akun
- i) Dapat melakukan operasi CRUD dalam Data Master
- j) Dapat melakukan operasi CRUD dalam Tool
- k) Dapat mengakses Settings

b. Pimpinan Cabang

- a) Dapat Login
- b) Dapat mengakses Dashboard
- c) Dapat melakukan operasi CRUD dalam Pimpinan
- d) Dapat melakukan operasi CRUD dalam Anggota
- e) Dapat melakukan operasi CRUD dalam AUM
- f) Dapat melakukan operasi CRUD dalam Aset
- g) Dapat melakukan operasi CRUD dalam beberapa Data Master

c. Pimpinan Ranting

- a) Dapat Login
- b) Dapat Mengakses Dashboard
- c) Dapat melakukan operasi CRUD dalam Anggota

d)

d. Umum

- a) Dapat Melihat Halaman Beranda
- b) Dapat Melihat Halaman Post
- c) Dapat Melihat Halaman Profil

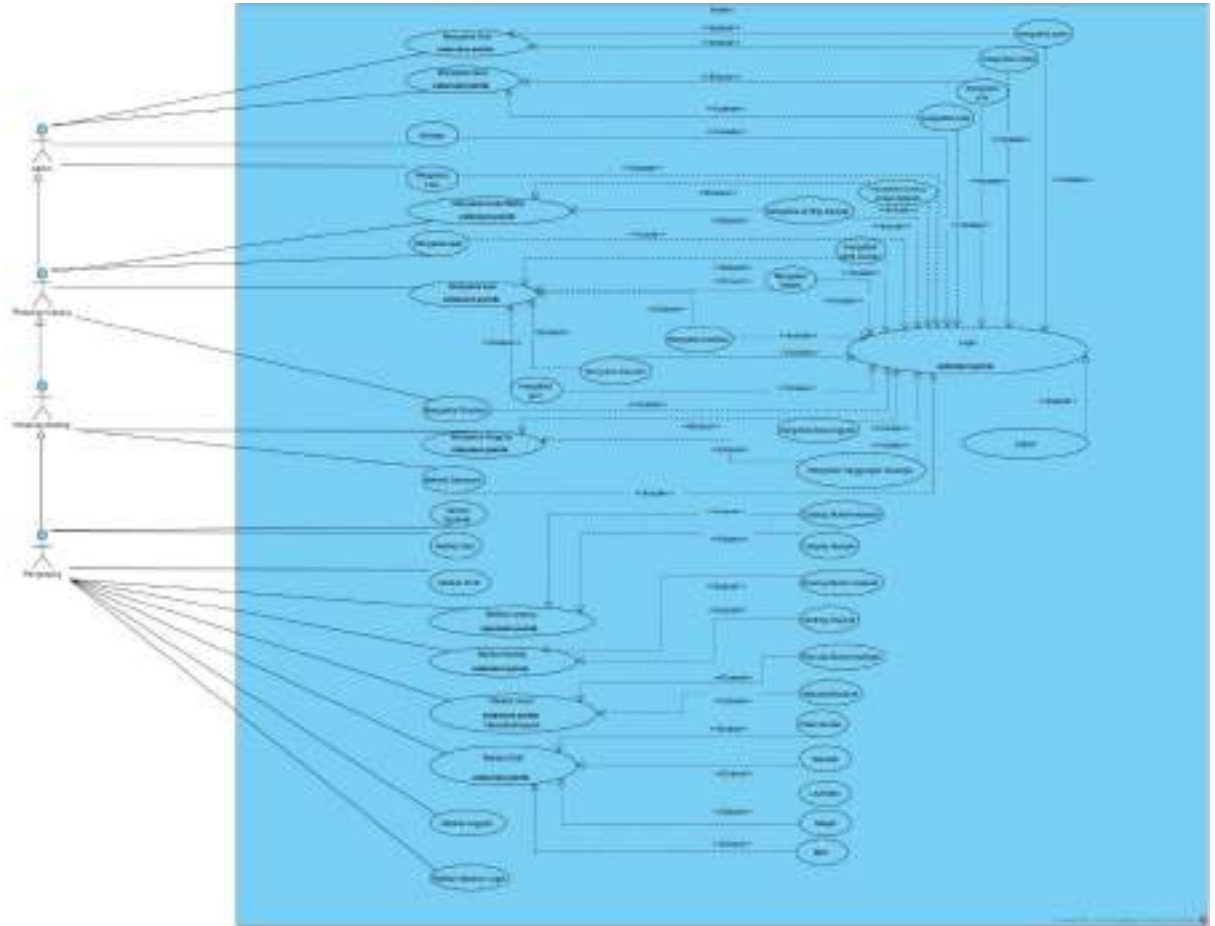
- d) Dapat Melihat Halaman Pimpinan
- e) Dapat Melihat Halaman Ranting
- f) Dapat Melihat Halaman AUM
- g) Dapat Melihat Halaman Anggota
- h) Dapat Melihat Halaman Dokumen
- i) Dapat Melihat Halaman Login

2. Kebutuhan Non-Functional

- a. Keamanan data terjamin
- b. Mudah digunakan

B. UML

1. Use Case Diagram



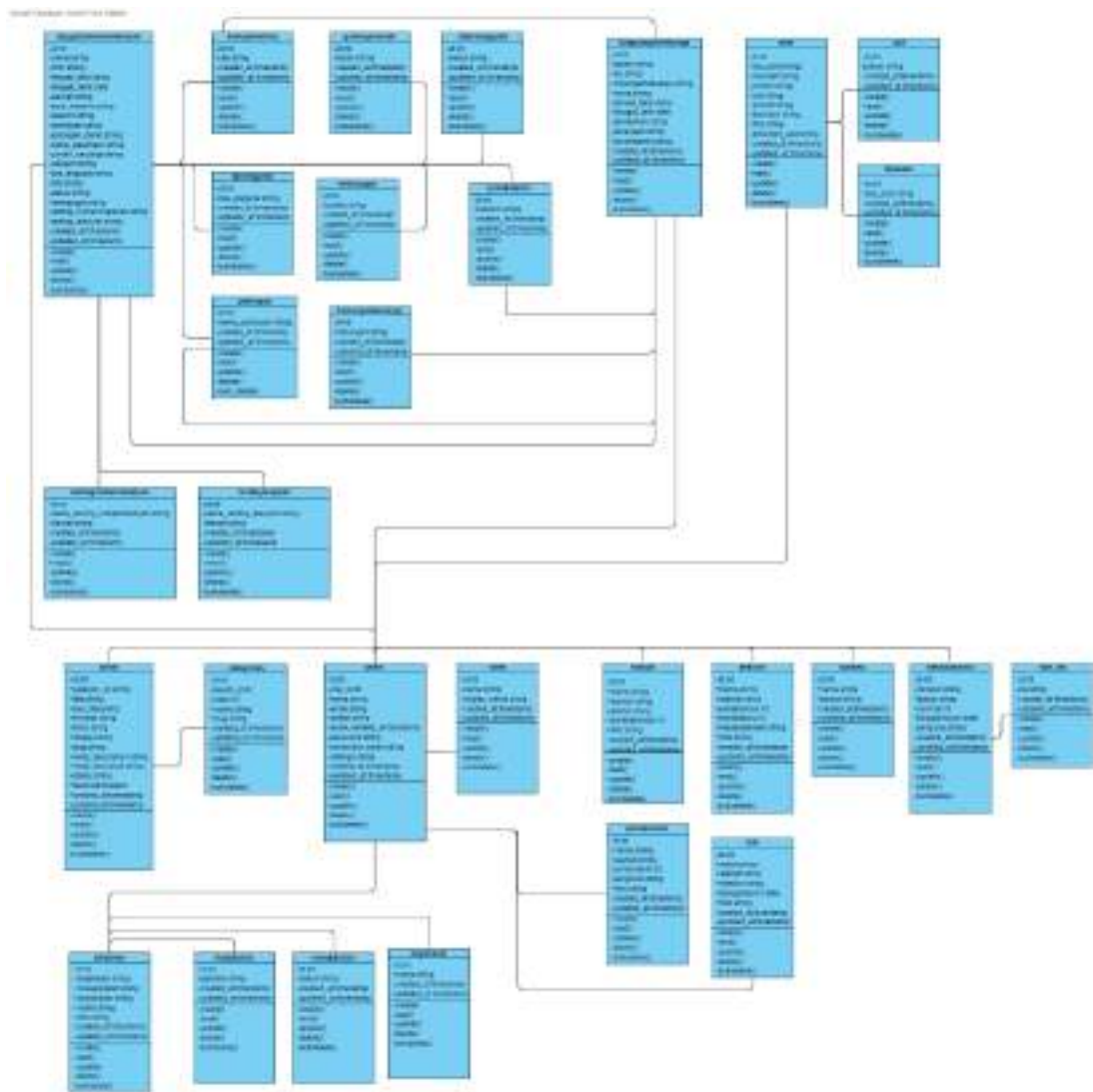
Gambar 1 Use Case Diagram

Berdasarkan use case diagram pada gambar diatas, terdapat 4 aktor yang terlibat dalam software ini, yaitu :

- 1) Admin, disini Admin bertindak sebagai SuperAdmin, yaitu dapat mengakses semua fitur yaitu Mengelola Post, Mengelola Akun, Settings, Mengelola Tool, Mengelola Data Master, Mengelola Aset, Mengelola AUM, Mengelola Pimpinan, Mengelola Anggota, Login.
- 2) Pimpinan Cabang, role ini dipegang oleh Pimpinan Cabang yang dapat melakukan beberapa task seperti Mengelola Data Master, Mengelola Aset, Mengelola AUM, Mengelola Pimpinan dan Login.

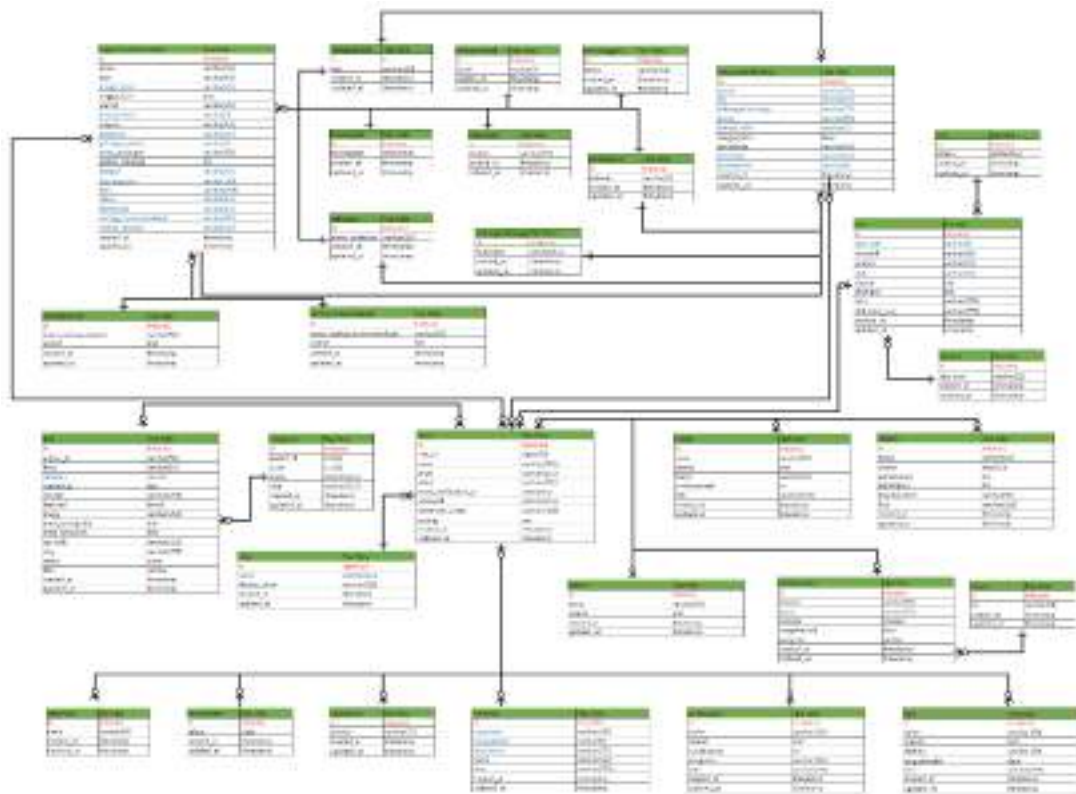
- 3) Pimpinan Ranting, disini Pimpinan Ranting akan dikelola oleh PRM yang dimana dapat melakukan beberapa hal yaitu Mengelola Anggota, Melihat Dashboard dan Login.
- 4) Aktor terakhir adalah Pengunjung, untuk Pengunjung tidak diperlukan Login sehingga Pengunjung hanya dapat Melihat Beranda, Melihat Post, Melihat Profil, Melihat Cabang, Melihat Ranting, Melihat Ortom, Melihat AUM, Melihat Anggota dan Melihat Halaman Login.

2. Class Diagram



Gambar 2 Class Diagram

3. Database Diagram

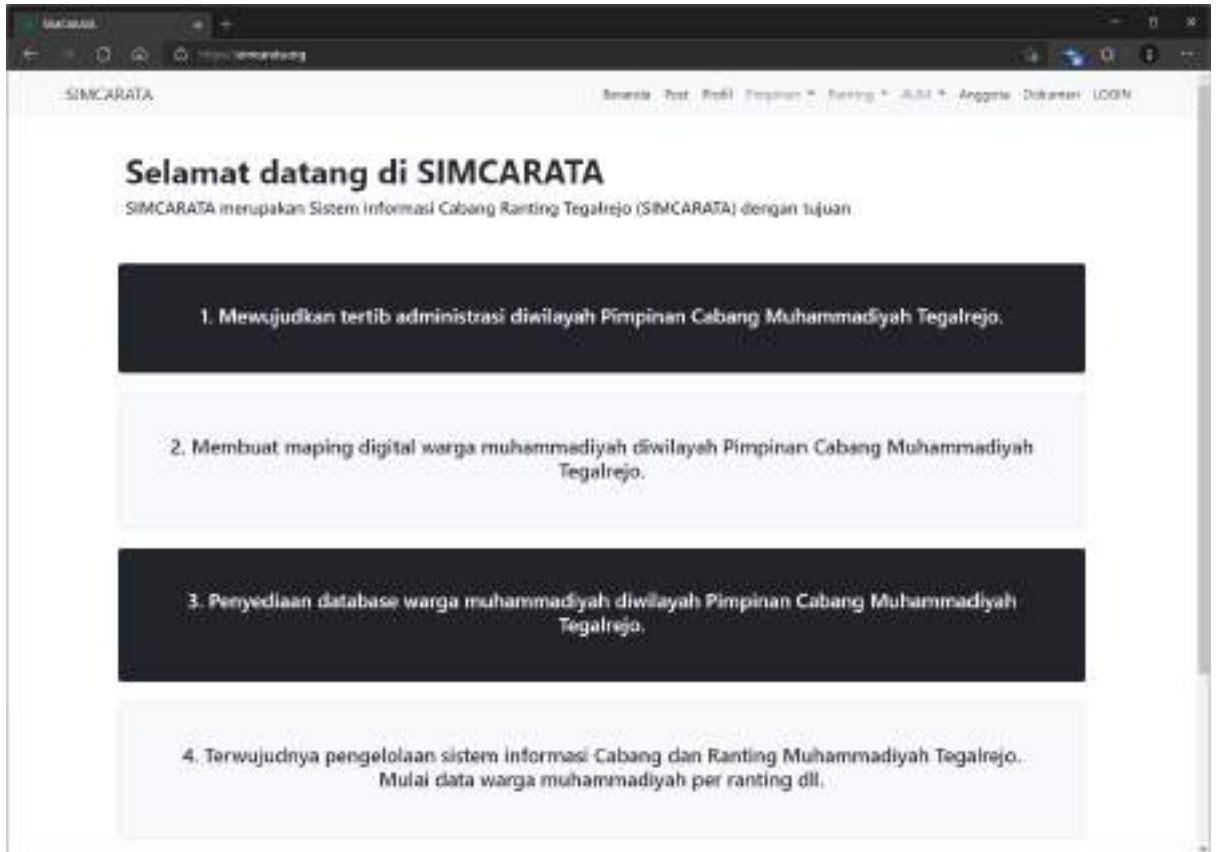


Gambar 3 Database Diagram

BAB V IMPLEMENTASI

A. Halaman Public

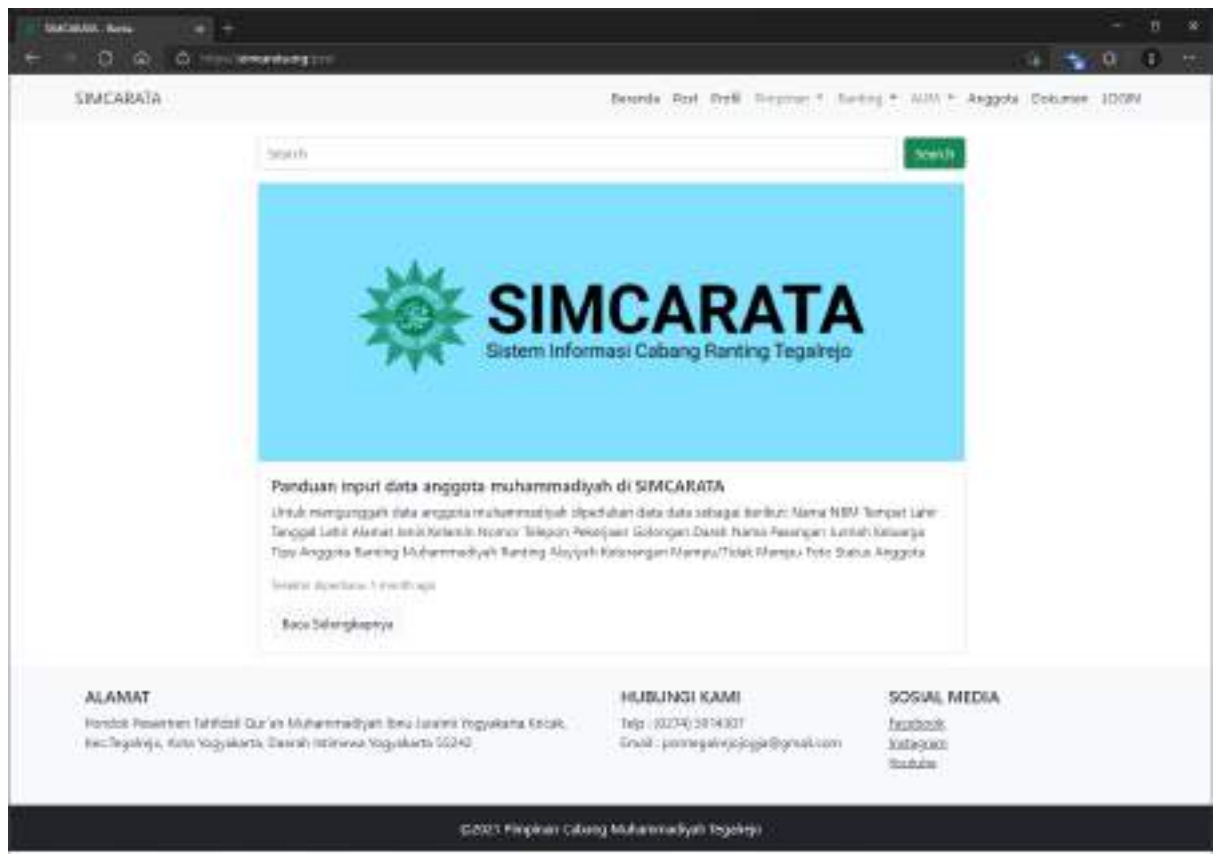
1. Beranda



Gambar 4 Public – Beranda

Halaman Beranda menampilkan informasi berupa tujuan dari website Simcarata.

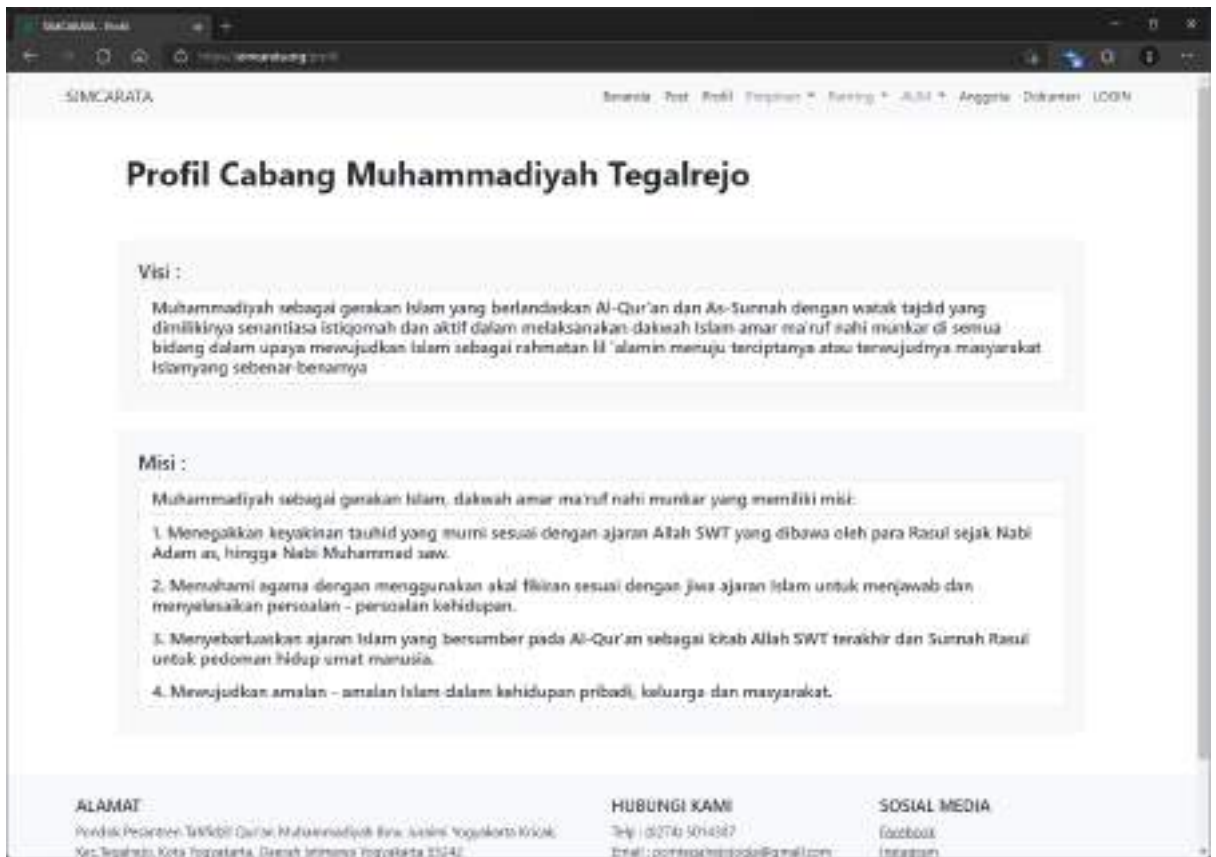
2. Post



Gambar 5 Public - Post

Halaman Post menampilkan informasi atau berita yang diunggah oleh admin. Di dalam halaman juga terdapat fitur untuk mencari berita yang tidak tampil di halaman awal Post.

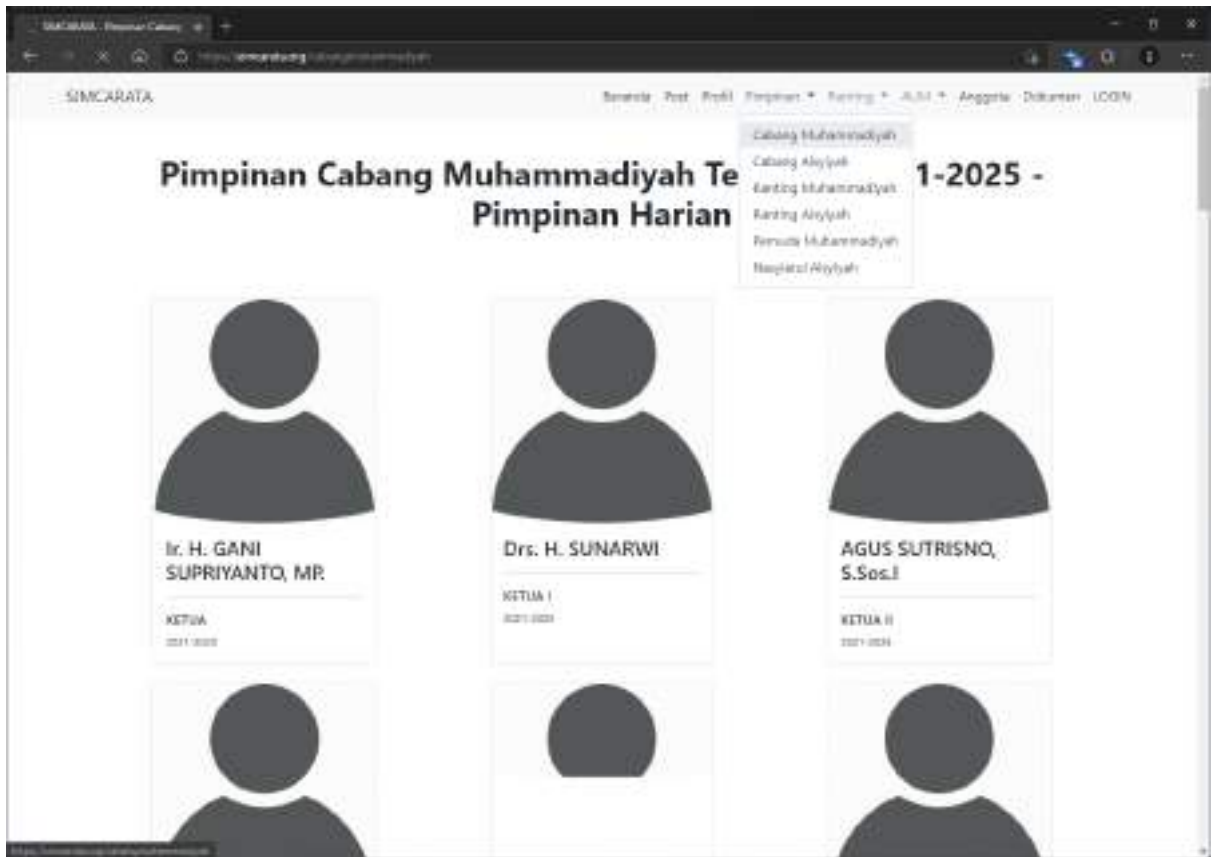
3. Profil



Gambar 6 Public - Profil

Halaman Profil menampilkan informasi visi dan misi yang dimiliki oleh Pimpinan Cabang Muhammadiyah Tegalrejo.

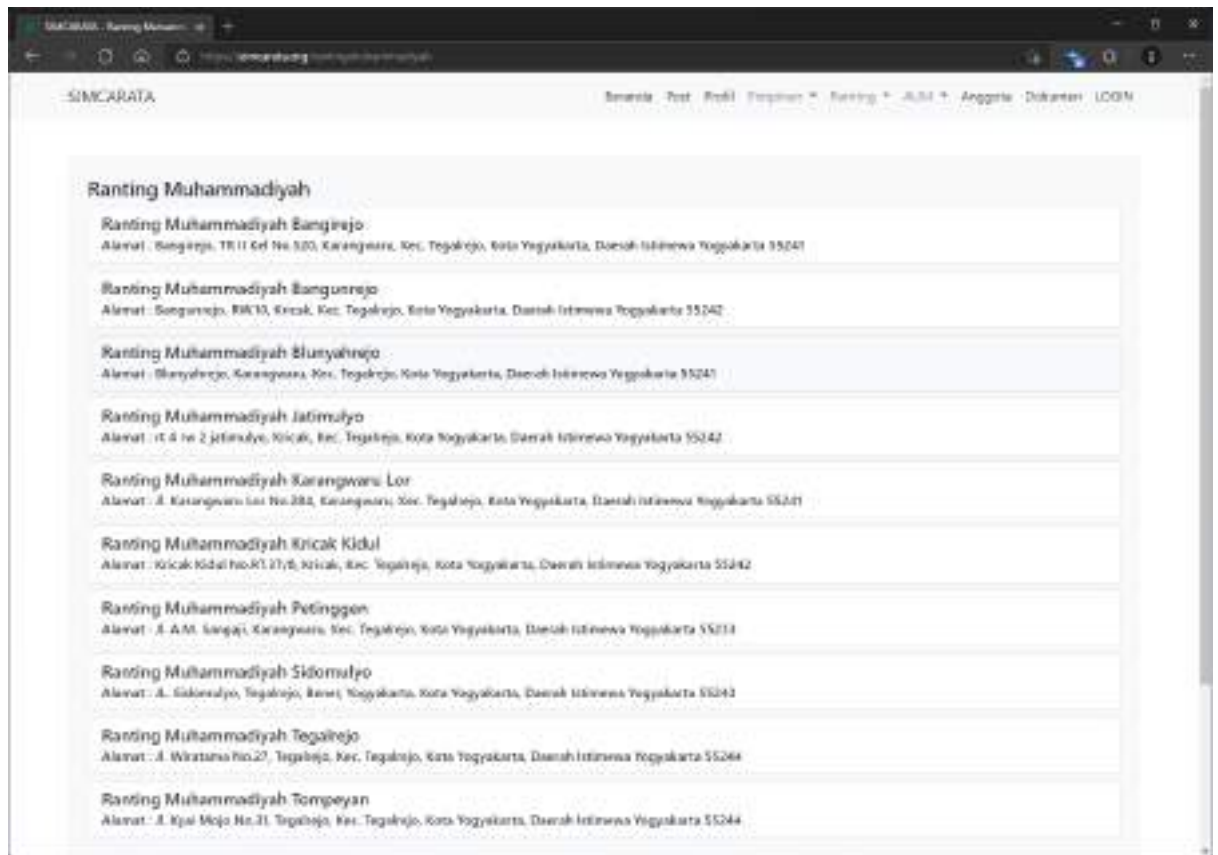
4. Pimpinan



Gambar 7 Public - Pimpinan

Navigation bar Pimpinan berisi halaman yang digunakan untuk menampilkan struktur organisasi Cabang Muhammadiyah Tegalrejo sesuai periode yang berlaku.

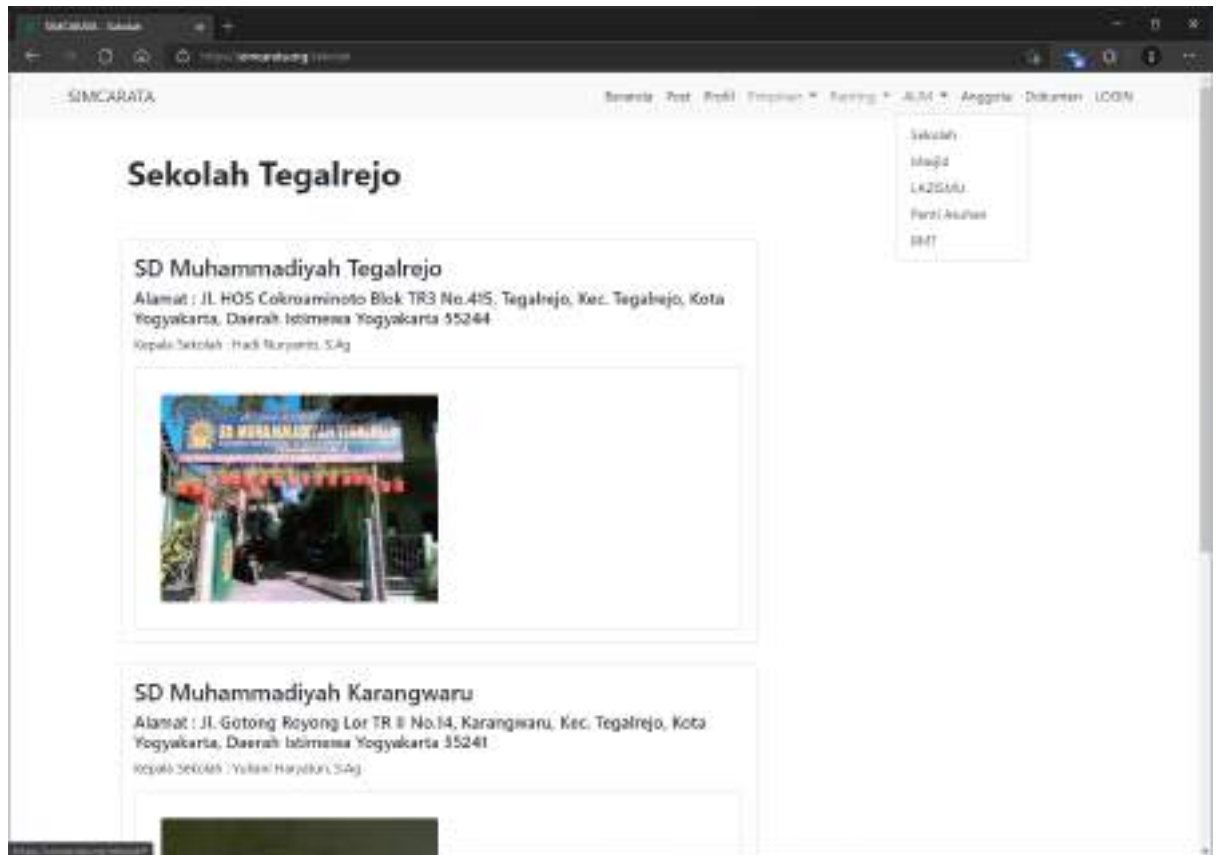
5. Ranting



Gambar 8 Public – Ranting

Navigation bar Ranting berisi halaman yang digunakan untuk menampilkan semua ranting yang ada di wilayah Cabang Tegalrejo.

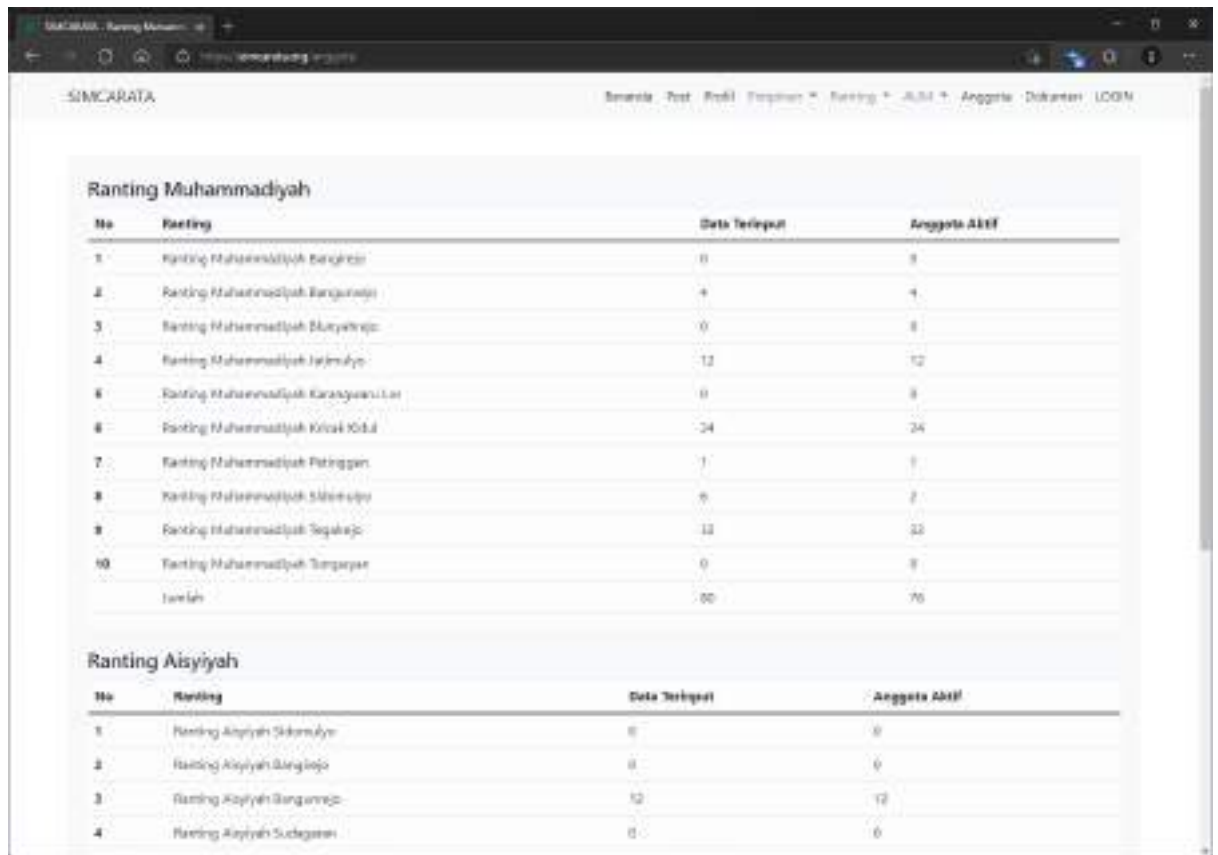
6. AUM



Gambar 9 Public – AUM

Navigation bar AUM terdiri dari berbagai informasi yang menampilkan amal usaha Muhammadiyah di wilayah cabang Tegalrejo.

7. Anggota



The screenshot displays a web interface for 'SINCARATA' with a navigation menu including 'Beranda', 'Post', 'Profil', 'Profilmu', 'Ranting', 'AJK', 'Anggota', 'Diskusi', and 'LOGIN'. The main content is divided into two sections: 'Ranting Muhammadiyah' and 'Ranting Aisyiyah'. Each section contains a table with columns for 'No', 'Ranting', 'Data Terdaftar', and 'Anggota Aktif'.

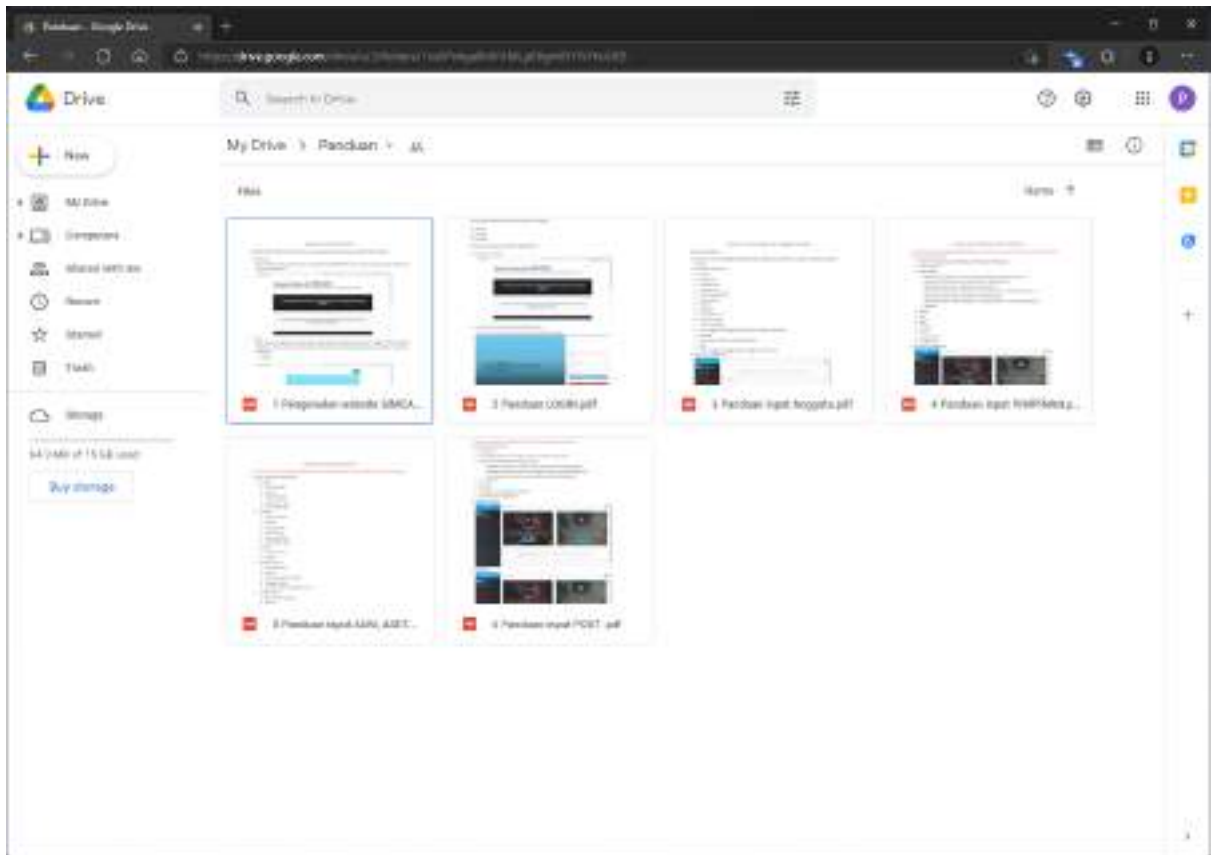
No	Ranting	Data Terdaftar	Anggota Aktif
1	Ranting Muhammadiyah Bangor	0	0
2	Ranting Muhammadiyah Bangor	4	4
3	Ranting Muhammadiyah Bureh	0	0
4	Ranting Muhammadiyah Jember	12	12
5	Ranting Muhammadiyah Karanganyar	0	0
6	Ranting Muhammadiyah Kral Kila	24	24
7	Ranting Muhammadiyah Pitingan	1	0
8	Ranting Muhammadiyah Sidosopo	0	2
9	Ranting Muhammadiyah Tegarejo	12	22
10	Ranting Muhammadiyah Tegalrejo	0	0
	Jumlah	00	70

No	Ranting	Data Terdaftar	Anggota Aktif
1	Ranting Aisyiyah Sukomulyo	0	0
2	Ranting Aisyiyah Bangorejo	0	0
3	Ranting Aisyiyah Bangorejo	12	12
4	Ranting Aisyiyah Sudagaran	0	0

Gambar 10 Public – Anggota

Halaman Anggota menampilkan jumlah anggota disetiap ranting yang ada di cabang Muhammadiyah Tegarejo.

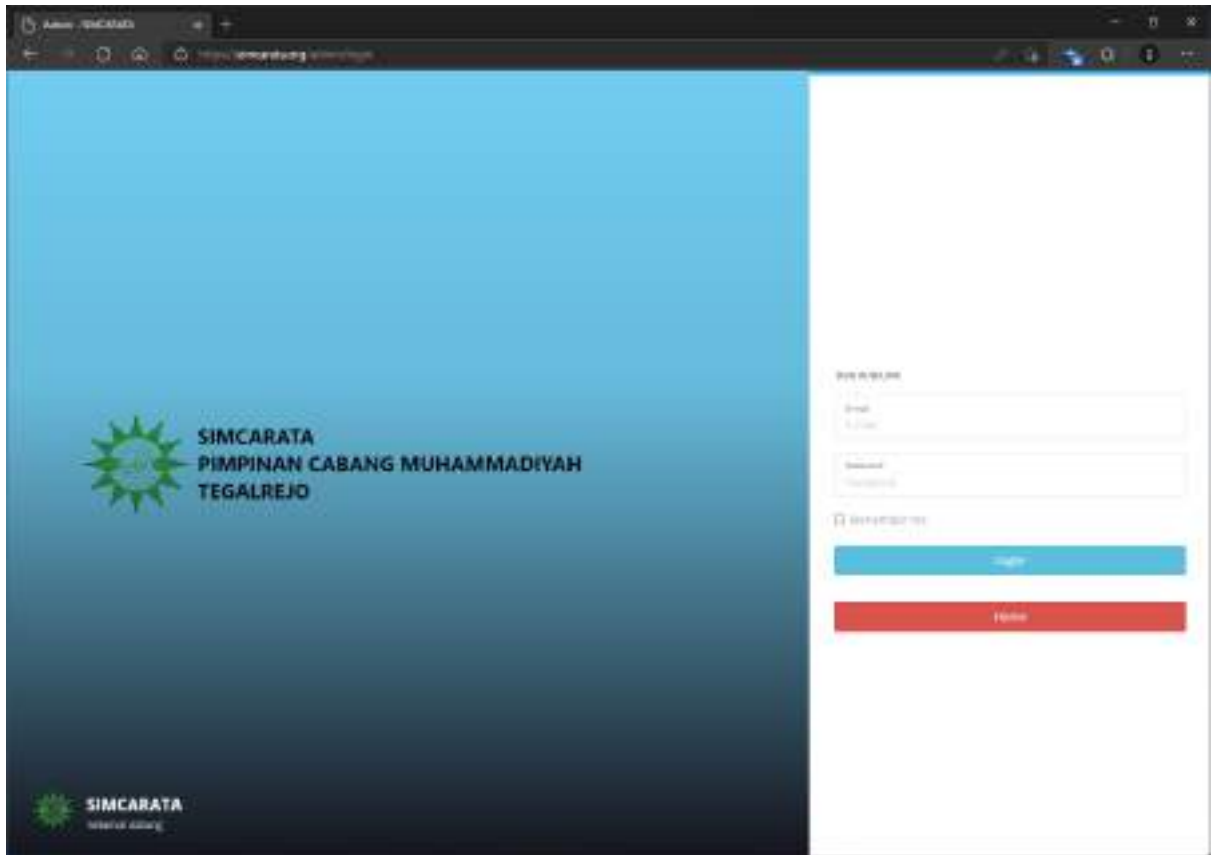
8. Dokumen



Gambar 11 Public – Dokumen

Setelah menekan Dokumen maka akan beralih ke halaman google drive yang berisi beberapa dokumen.

9. Login

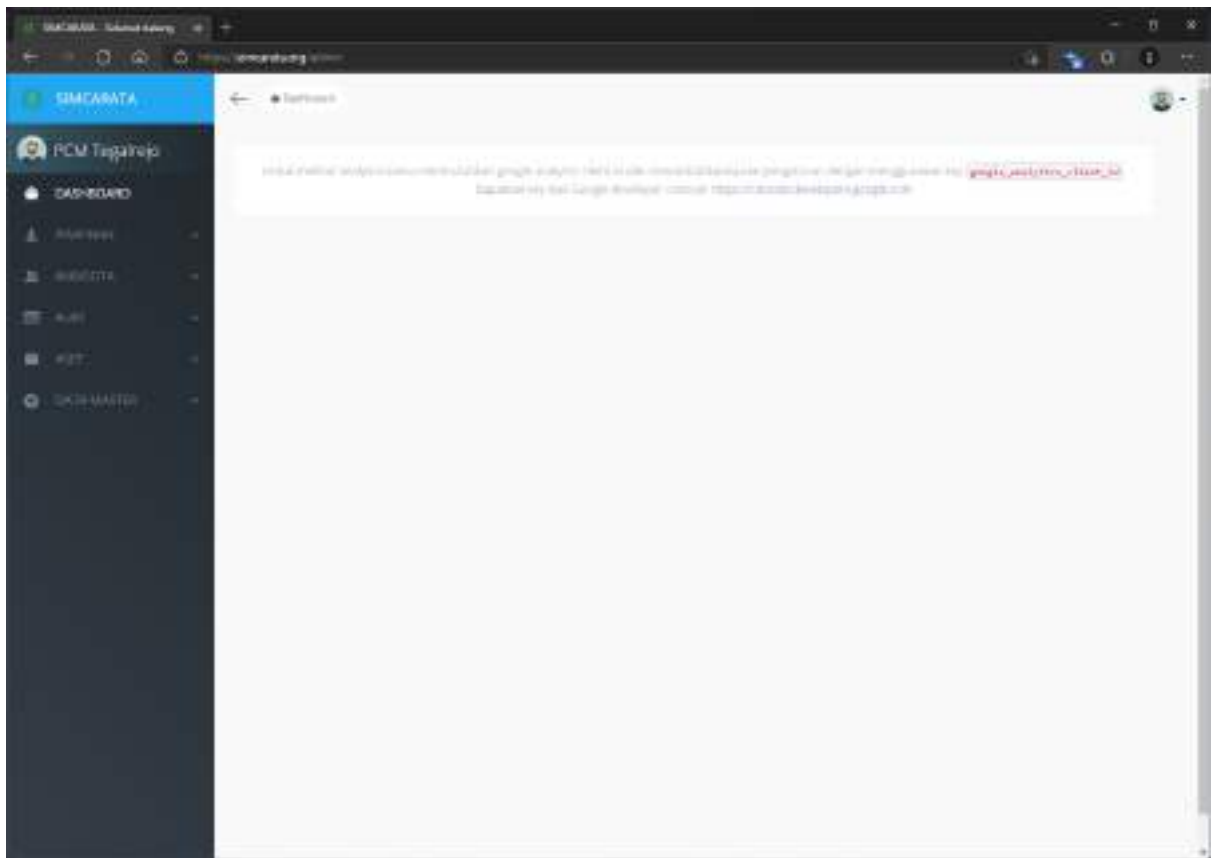


Gambar 12 Public – Login

Halaman Login digunakan untuk masuk ke halaman admin sesuai rolenya setelah memasukkan email dan password serta menekan tombol Login.

B. Halaman Pimpinan Cabang

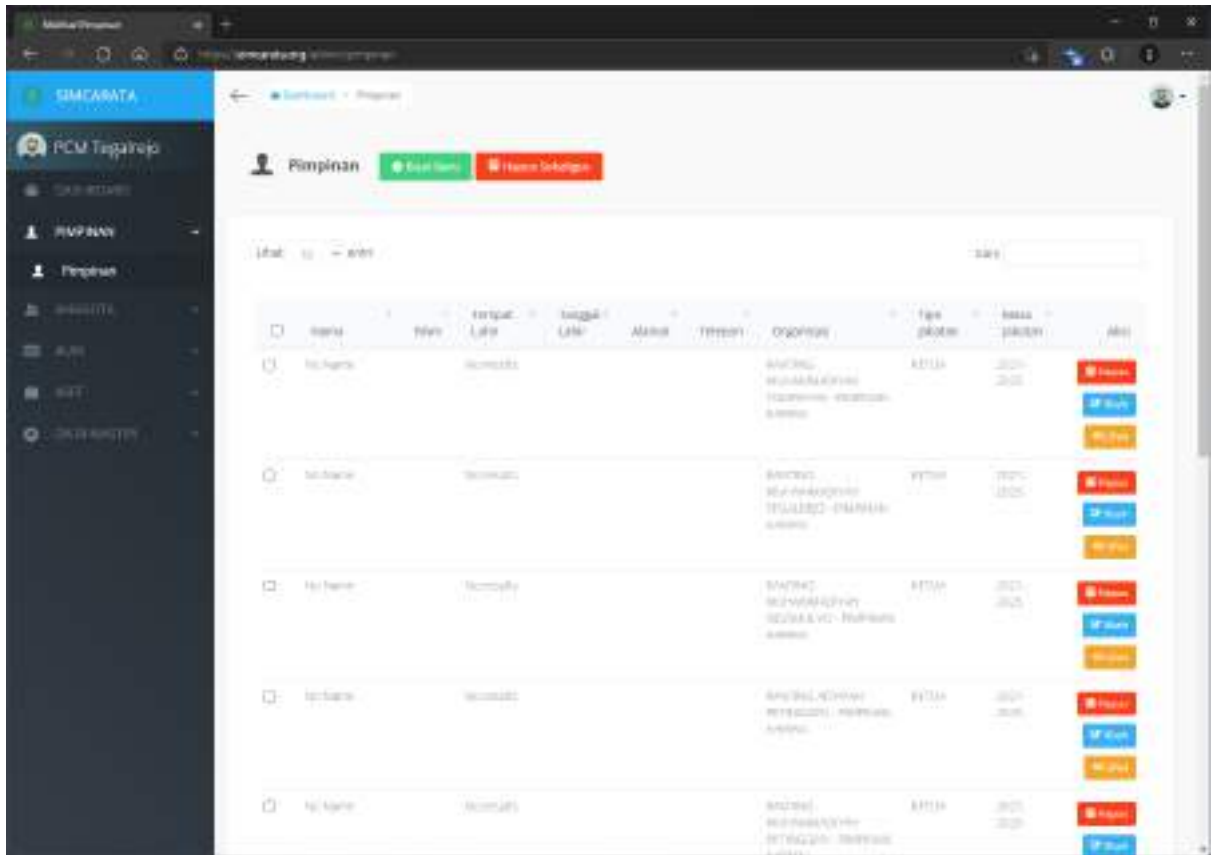
1. Dashboard



Gambar 13 Cabang – Dashboard

Halaman Dashboard merupakan tampilan awal setelah Login.

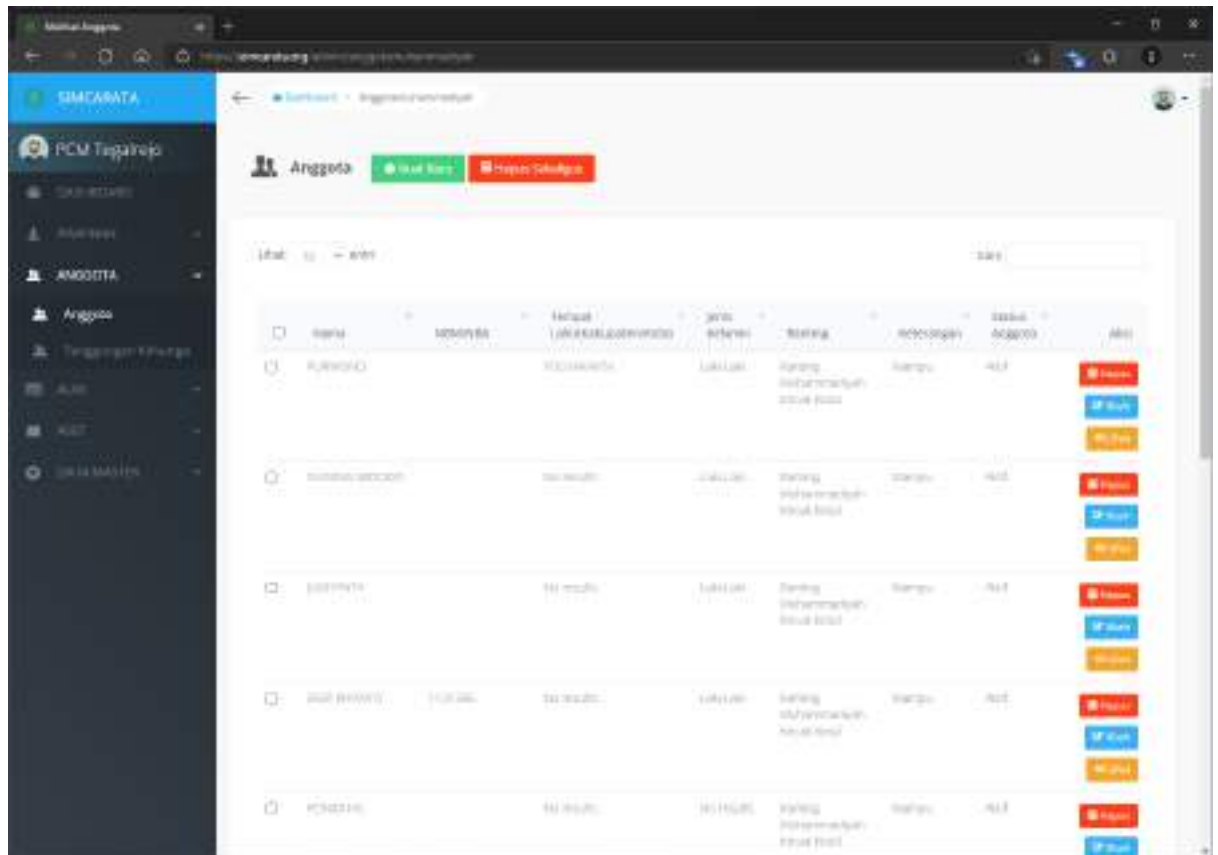
2. Pimpinan



Gambar 14 Cabang – Pimpinan

Halaman Pimpinan dapat melakukan operasi CRUD struktur organisasi yang ada di lingkup cabang Muhammadiyah Tegalrejo.

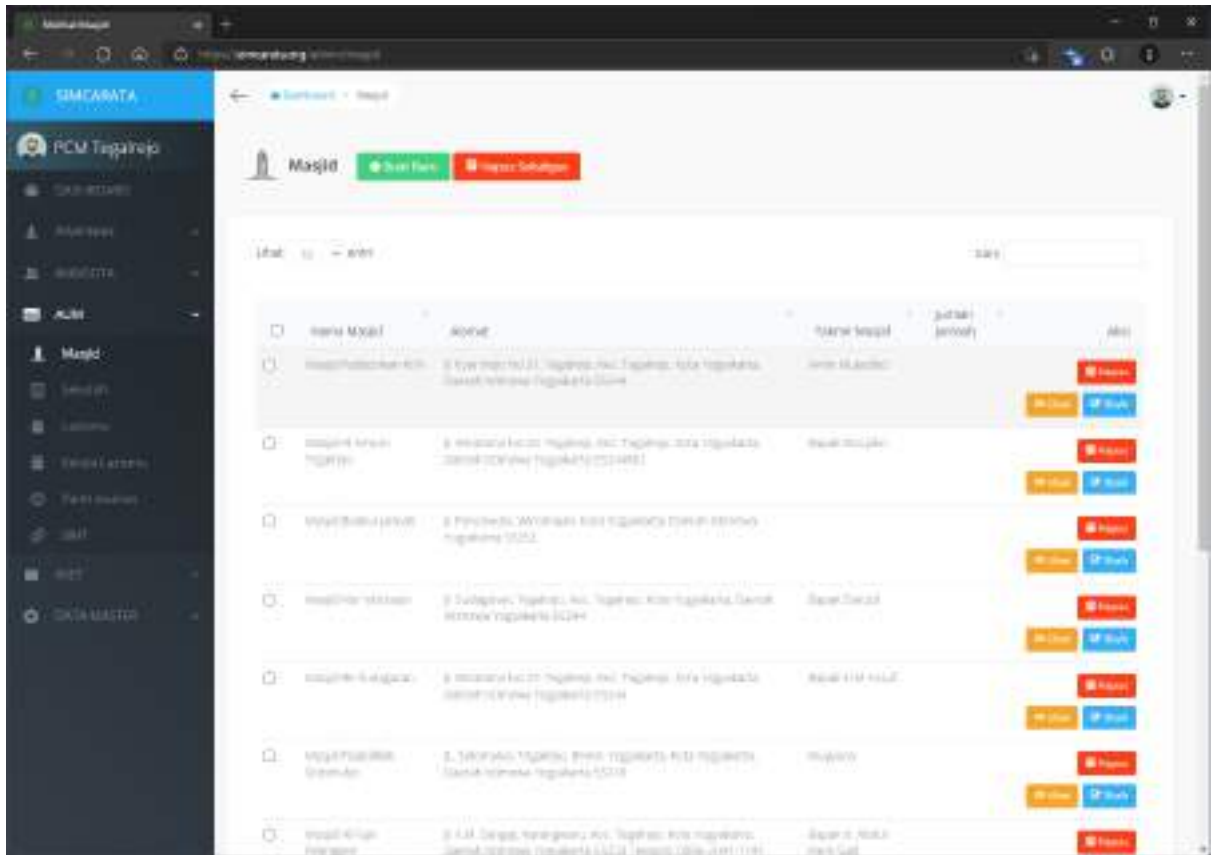
3. Anggota



Gambar 15 Cabang – Anggota

Halaman Anggotan dapat melakukan operasi CRUD data anggota yang merupakan anggota bagian dari cabang Muhammadiyah Tegalorejo.

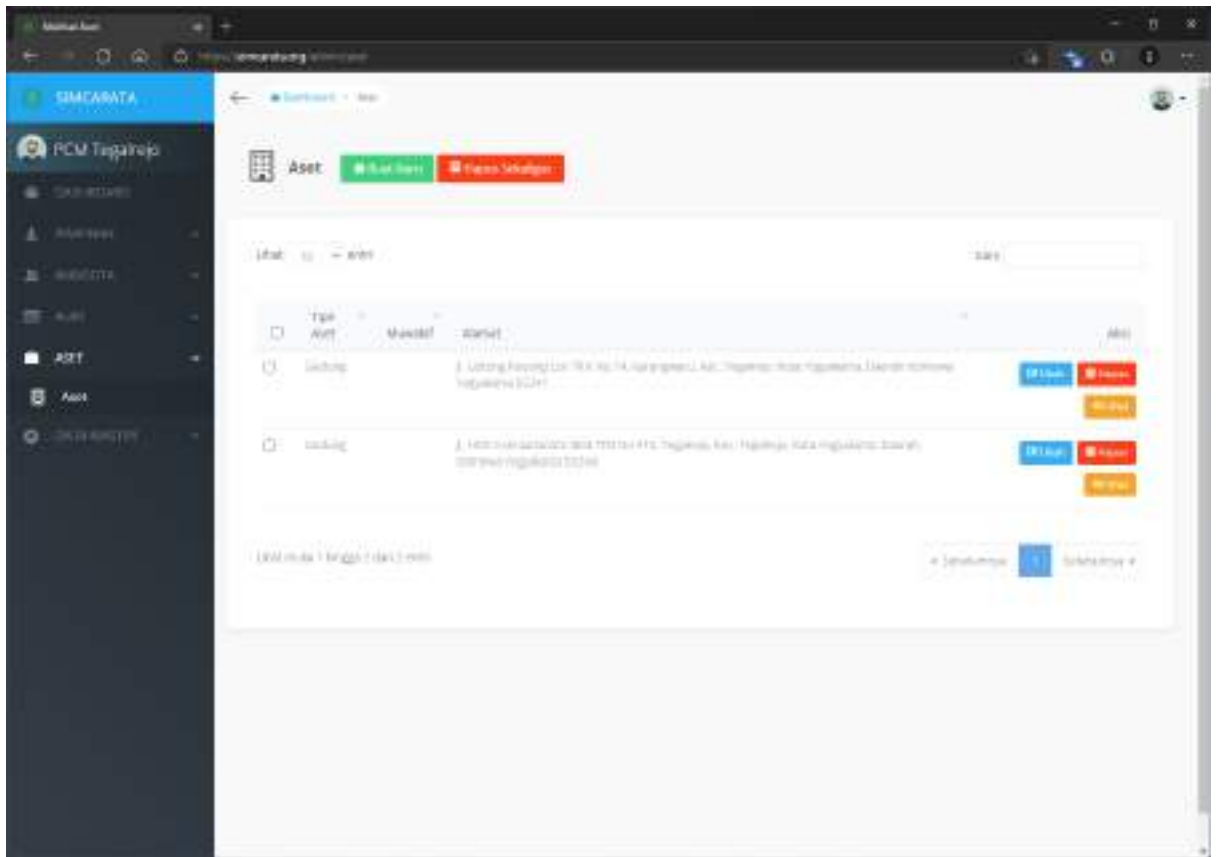
4. AUM



Gambar 16 Cabang – AUM

Halaman AUM dapat melakukan operasi CRUD data amal usaha muhammadiyah yang berada di wilayah cabang Muhammadiyah Tegayun.

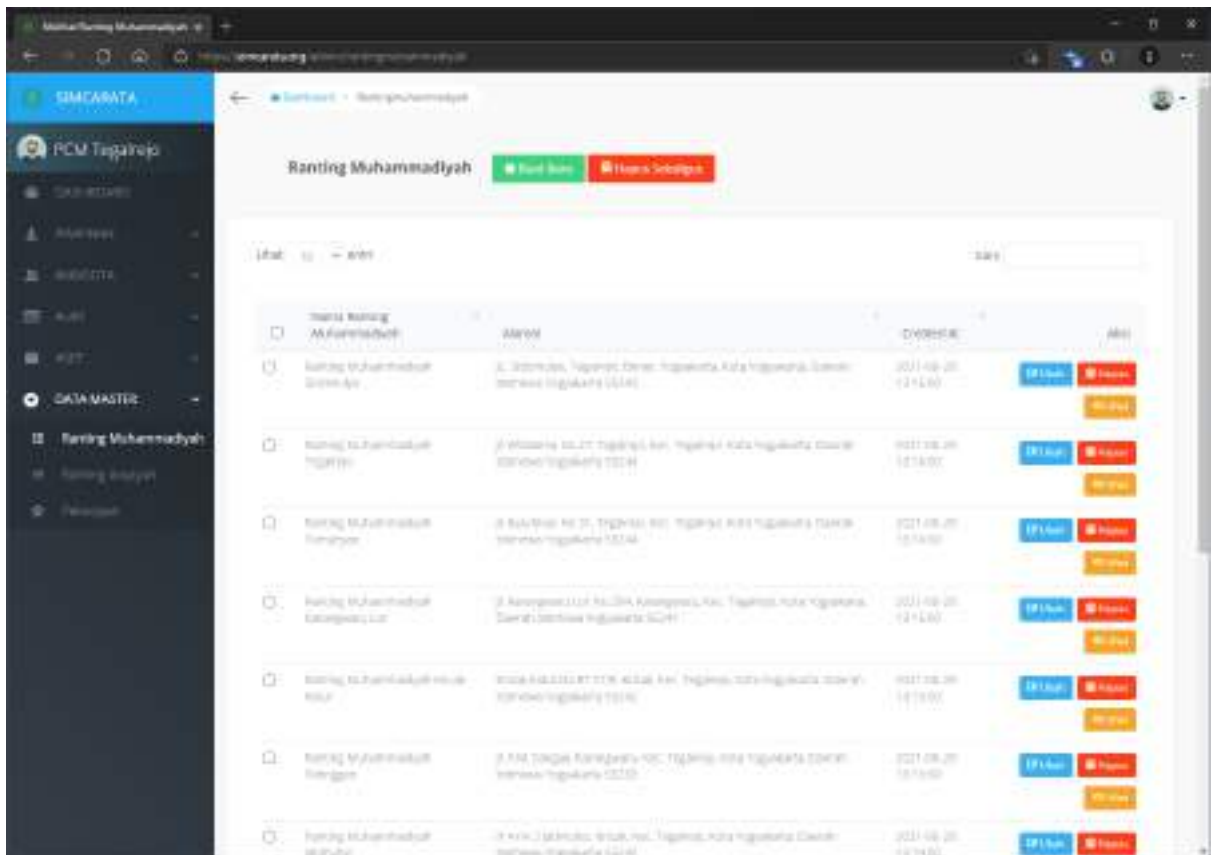
5. Aset



Gambar 17 Cabang – Aset

Halaman Aset dapat melakukan operasi CRUD yang berisi aset cabang Muhammadiyah Tegalrejo.

6. Data Master



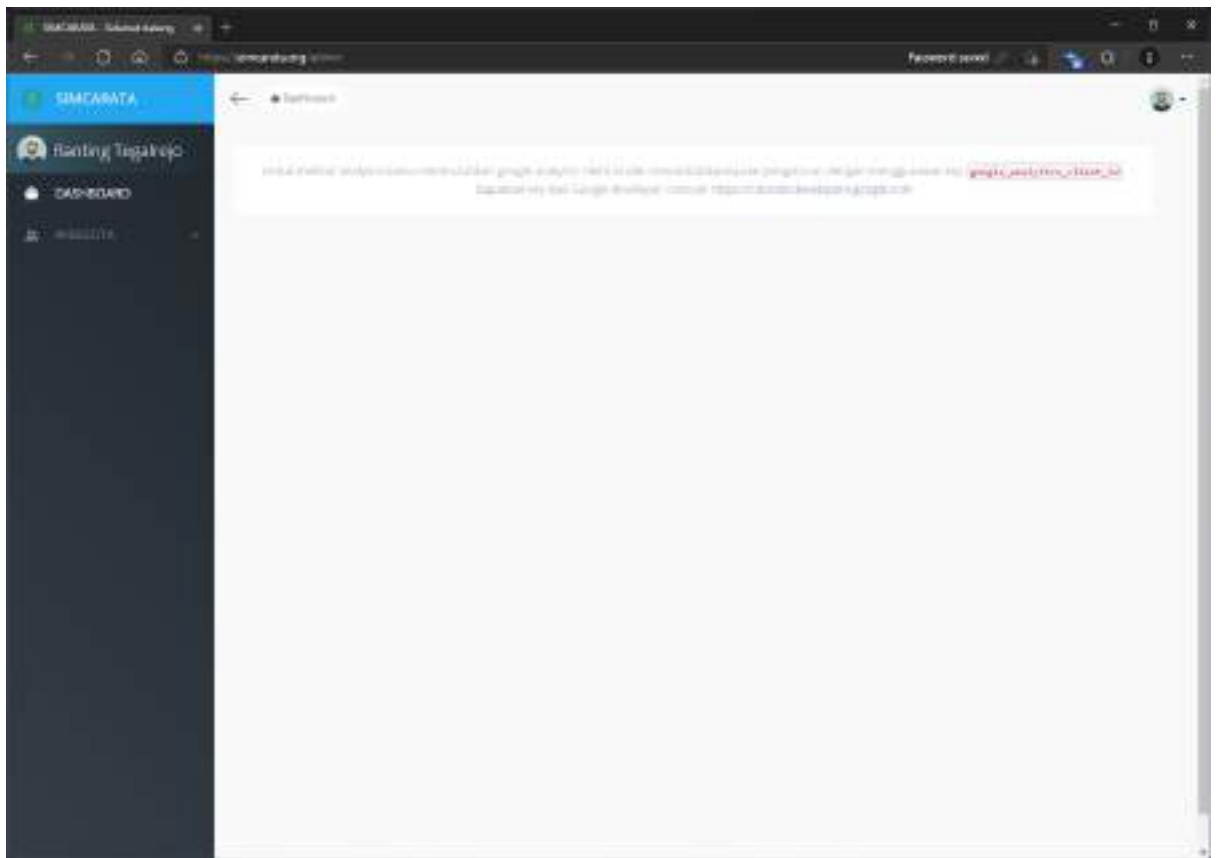
Nama Ranting	Alamat	DIBUAT	Aksi
Ranting Muhammadiyah Soreh Aji	Jl. Setono, Ranting Soreh, Kabupaten, Kota Ngawi, Jawa Timur, Indonesia 62244	2021-08-20 12:14:50	[Edit] [Hapus] [Tambah]
Ranting Muhammadiyah Tegayun	Jl. Widyadarmas, RT. 02, Tegayun, Kec. Tegayun, Kota Ngawi, Jawa Timur, Indonesia 62244	2021-08-20 12:15:07	[Edit] [Hapus] [Tambah]
Ranting Muhammadiyah Kembang	Jl. Bulu Putih No. 27, Tegayun, Kec. Tegayun, Kota Ngawi, Jawa Timur, Indonesia 62244	2021-08-20 12:15:20	[Edit] [Hapus] [Tambah]
Ranting Muhammadiyah Kedungwatu	Jl. Kandang, RT. 01, No. 20A, Kedungwatu, Kec. Tegayun, Kota Ngawi, Jawa Timur, Indonesia 62244	2021-08-20 12:14:50	[Edit] [Hapus] [Tambah]
Ranting Muhammadiyah Kedungwatu	Kedungwatu, RT. 01, No. 20A, Kedungwatu, Kec. Tegayun, Kota Ngawi, Jawa Timur, Indonesia 62244	2021-08-20 12:15:07	[Edit] [Hapus] [Tambah]
Ranting Muhammadiyah Kembang	Jl. PPA, Gedung, Kembang, Kec. Tegayun, Kota Ngawi, Jawa Timur, Indonesia 62244	2021-08-20 12:15:20	[Edit] [Hapus] [Tambah]
Ranting Muhammadiyah Kembang	Jl. Kembang, Gedung, Kembang, Kec. Tegayun, Kota Ngawi, Jawa Timur, Indonesia 62244	2021-08-20 12:15:20	[Edit] [Hapus] [Tambah]

Gambar 18 Cabang - Data Master

Halaman Data Master dapat melakukan operasi CRUD yang berisi data ranting Muhammadiyah, ranting Aisyiyah serta data pekerjaan.

C. Halaman Pimpinan Ranting

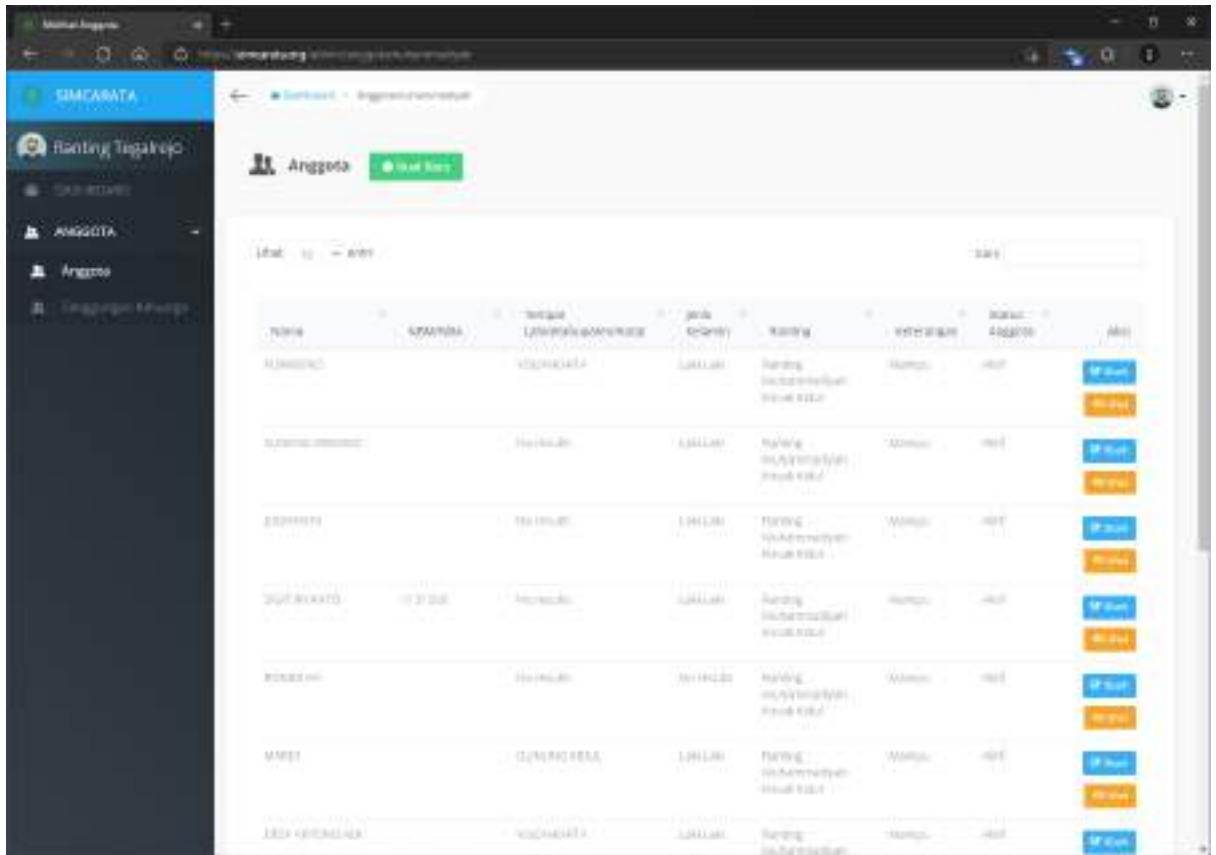
1. Dashboard



Gambar 19 Ranting – Dashboard

Halaman Dashboard merupakan tampilan awal setelah Login.

2. Anggota



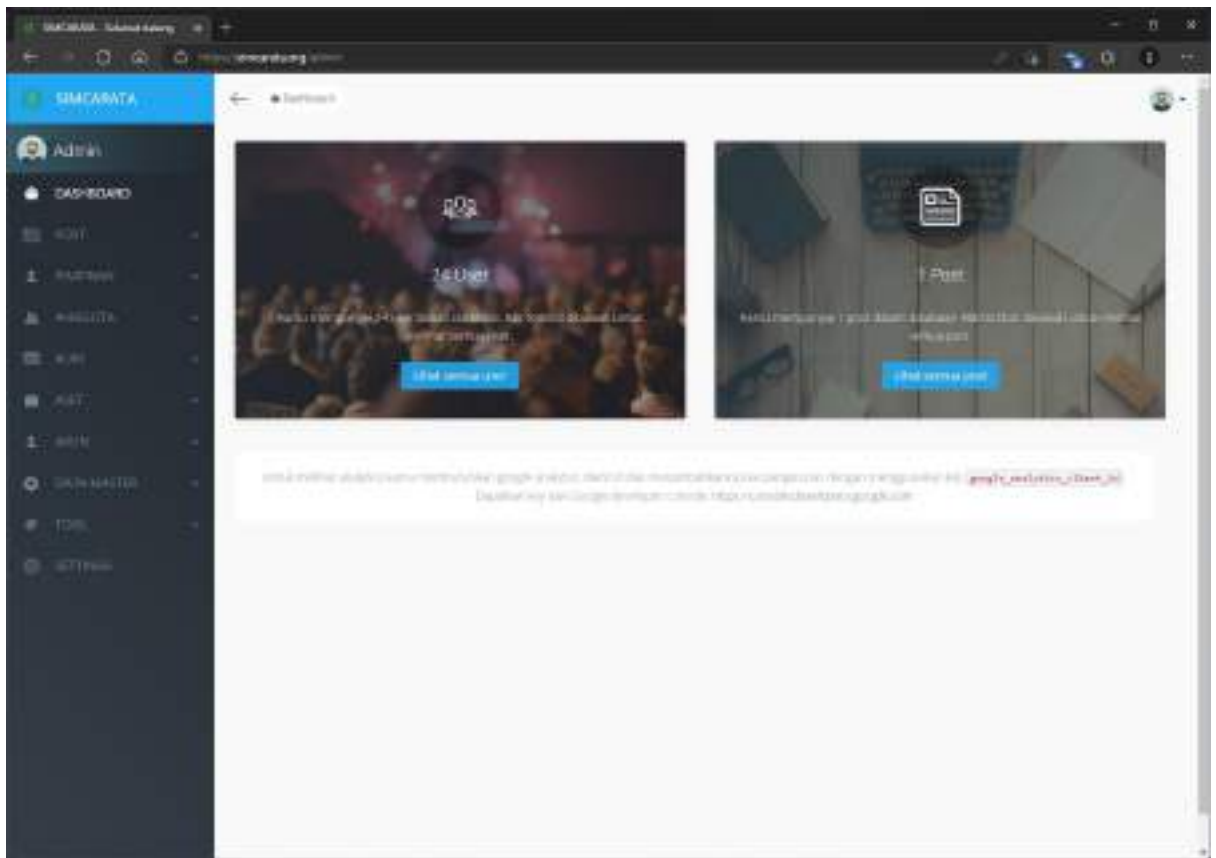
NAMA	KABUPATEN	Jabatan (Jabatan/posisi/kegiatan)	Jenis Kelamin	RANTING	STATUS	Aksi	
MUSLIMAH	KABUPATEN	KORPORASI	LAKILAKIA	RANTING Muhammadiyah Muab KOTA	Wanita	aktif	Edit Hapus
ALYANAH	KABUPATEN	KORPORASI	LAKILAKIA	RANTING Muhammadiyah Muab KOTA	Wanita	aktif	Edit Hapus
EDYATI	KABUPATEN	KORPORASI	LAKILAKIA	RANTING Muhammadiyah Muab KOTA	Wanita	aktif	Edit Hapus
DIYANAH	KABUPATEN	KORPORASI	LAKILAKIA	RANTING Muhammadiyah Muab KOTA	Wanita	aktif	Edit Hapus
ROSELI	KABUPATEN	KORPORASI	LAKILAKIA	RANTING Muhammadiyah Muab KOTA	Wanita	aktif	Edit Hapus
MARI	KABUPATEN	KORPORASI	LAKILAKIA	RANTING Muhammadiyah Muab KOTA	Wanita	aktif	Edit Hapus
LEA KURNIAWA	KABUPATEN	KORPORASI	LAKILAKIA	RANTING Muhammadiyah Muab KOTA	Wanita	aktif	Edit Hapus

Gambar 20 Ranting – Anggota

Halaman Anggota dapat melakukan operasi menambah dan mengubah data anggota yang ada di wilayah cabang Muhammadiyah Tegalrejo.

D. Halaman Admin

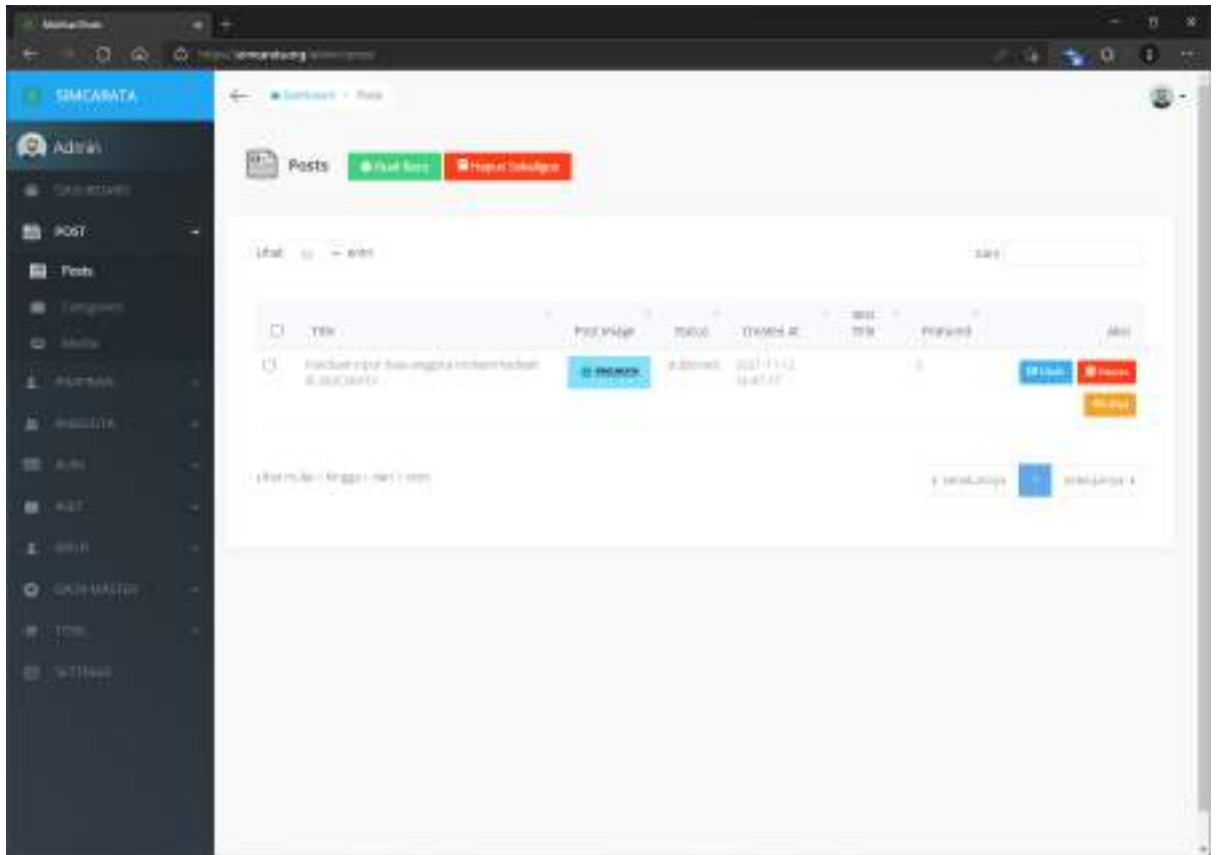
1. Dashboard



Gambar 21 Admin – Dashboard

Halaman Dashboard merupakan tampilan awal setelah Login.

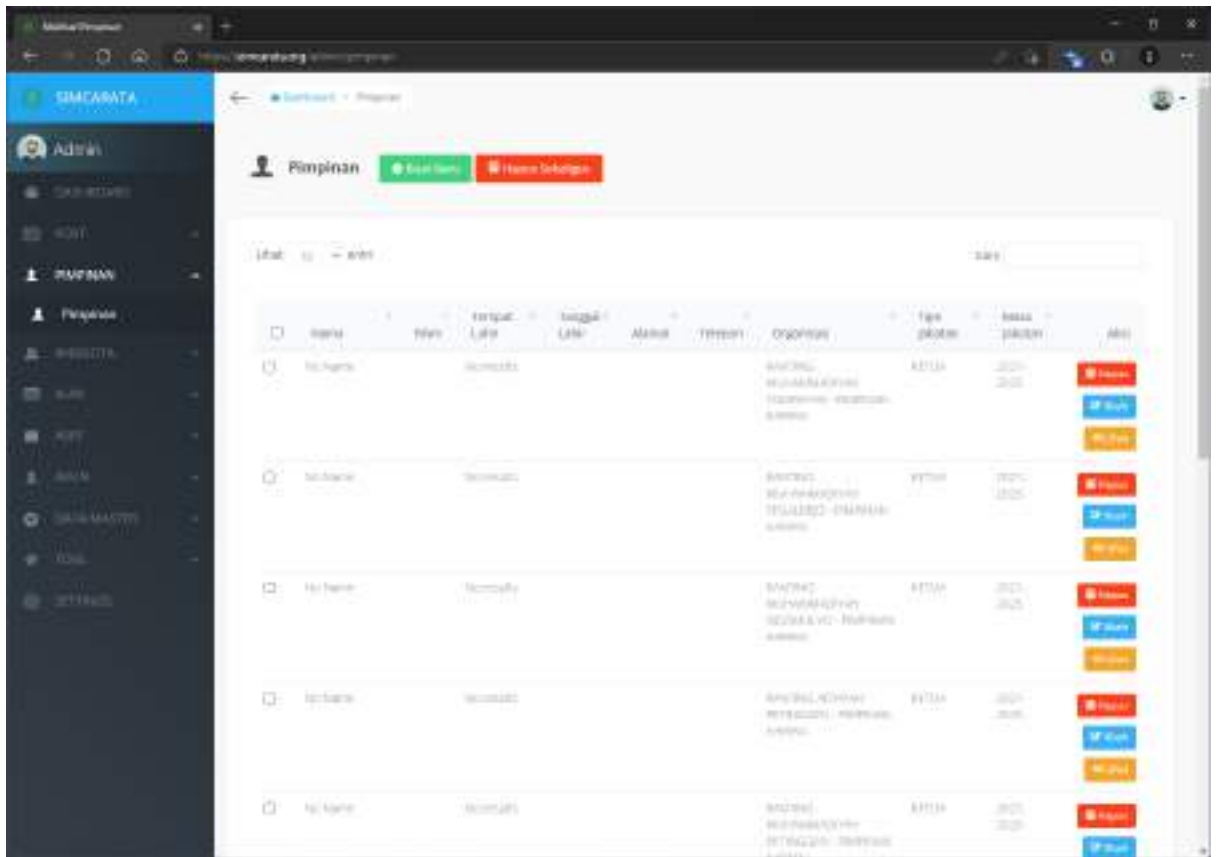
2. Post



Gambar 22 Admin – Post

Halaman Post digunakan untuk mengelola berita yang akan ditampilkan di halaman publik.

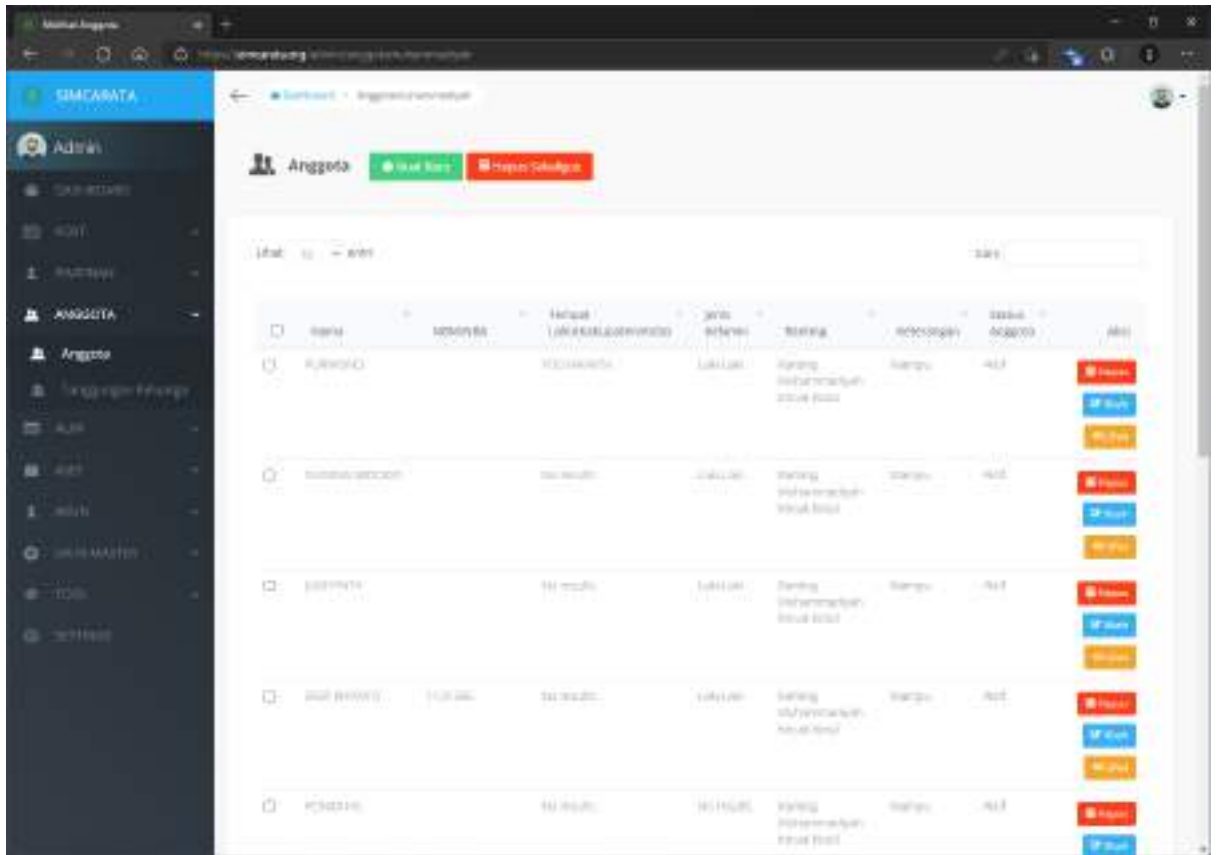
3. Pimpinan



Gambar 23 Admin – Pimpinan

Halaman Pimpinan dapat melakukan operasi CRUD struktur organisasi yang ada di lingkup cabang Muhammadiyah Tegalrejo.

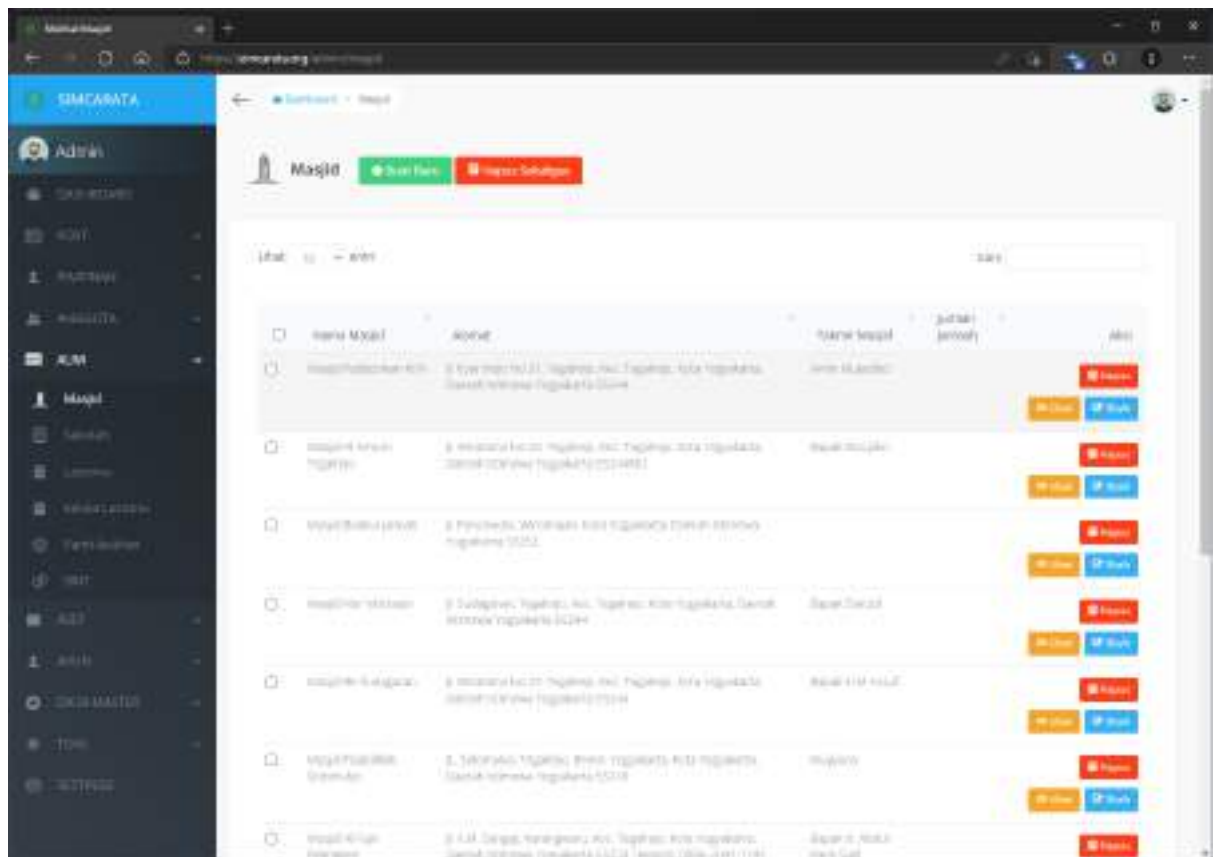
4. Anggota



Gambar 24 Admin – Anggota

Halaman Anggotan dapat melakukan operasi CRUD data anggota yang merupakan anggota bagian dari cabang Muhammadiyah Tegalrejo.

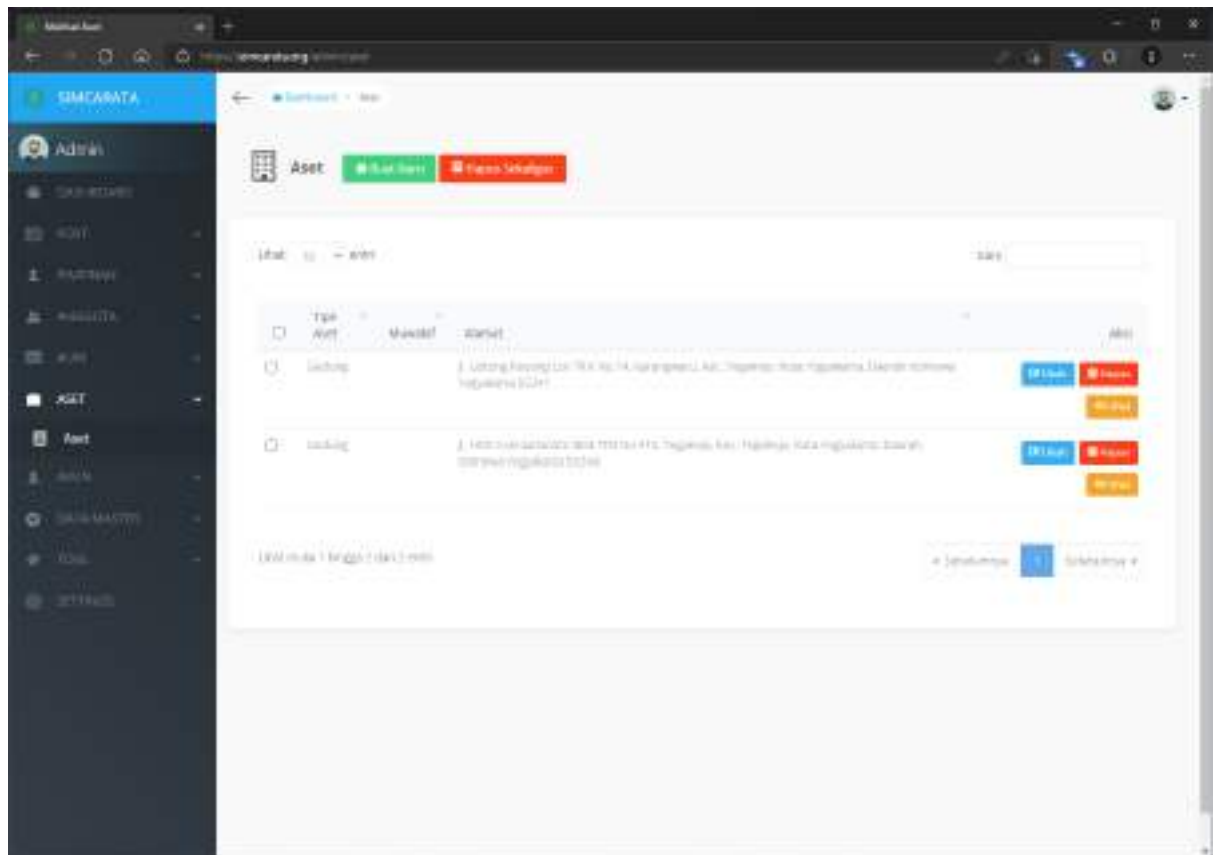
5. AUM



Gambar 25 Admin – AUM

Halaman AUM dapat melakukan operasi CRUD data amal usaha muhammadiyah yang berada di wilayah cabang Muhammadiyah Tegalrejo.

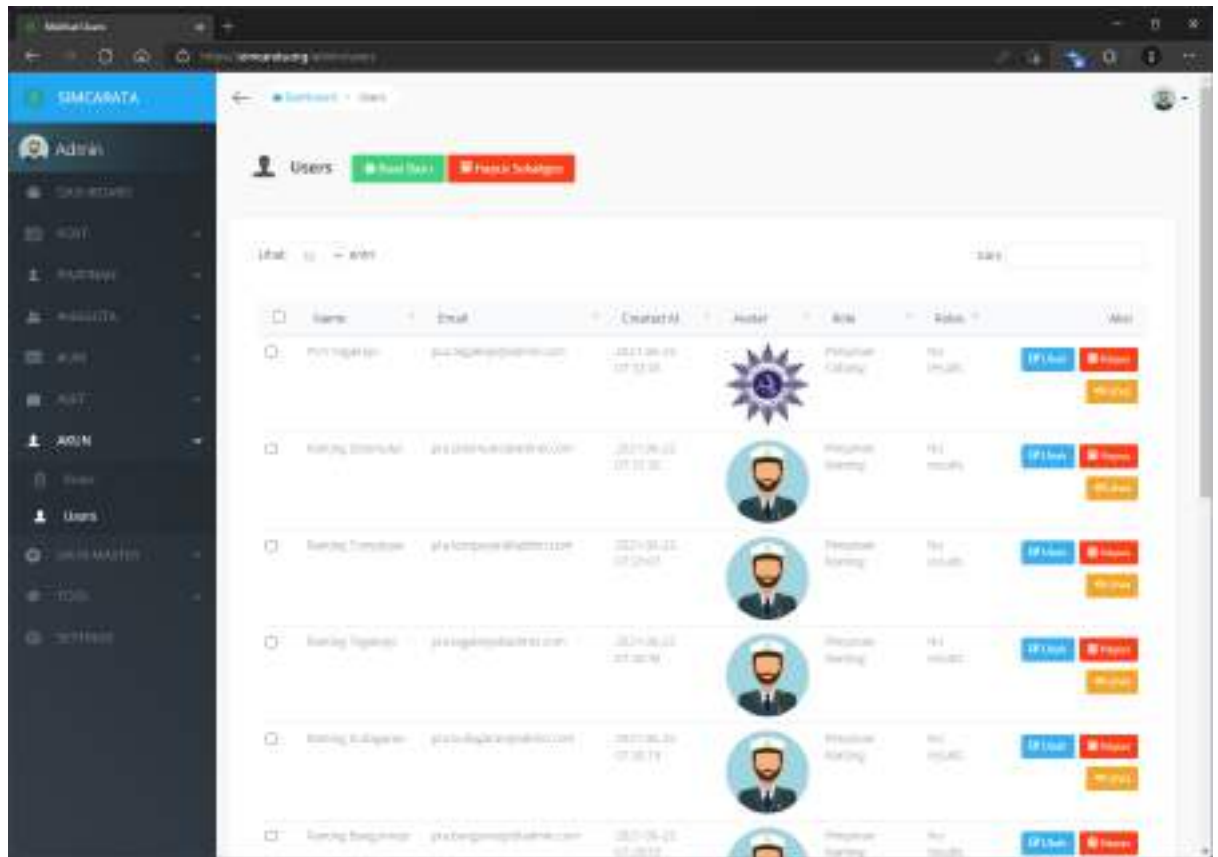
6. Aset



Gambar 26 Admin – Aset

Halaman Aset dapat melakukan operasi CRUD yang berisi aset cabang Muhammadiyah Tegalrejo

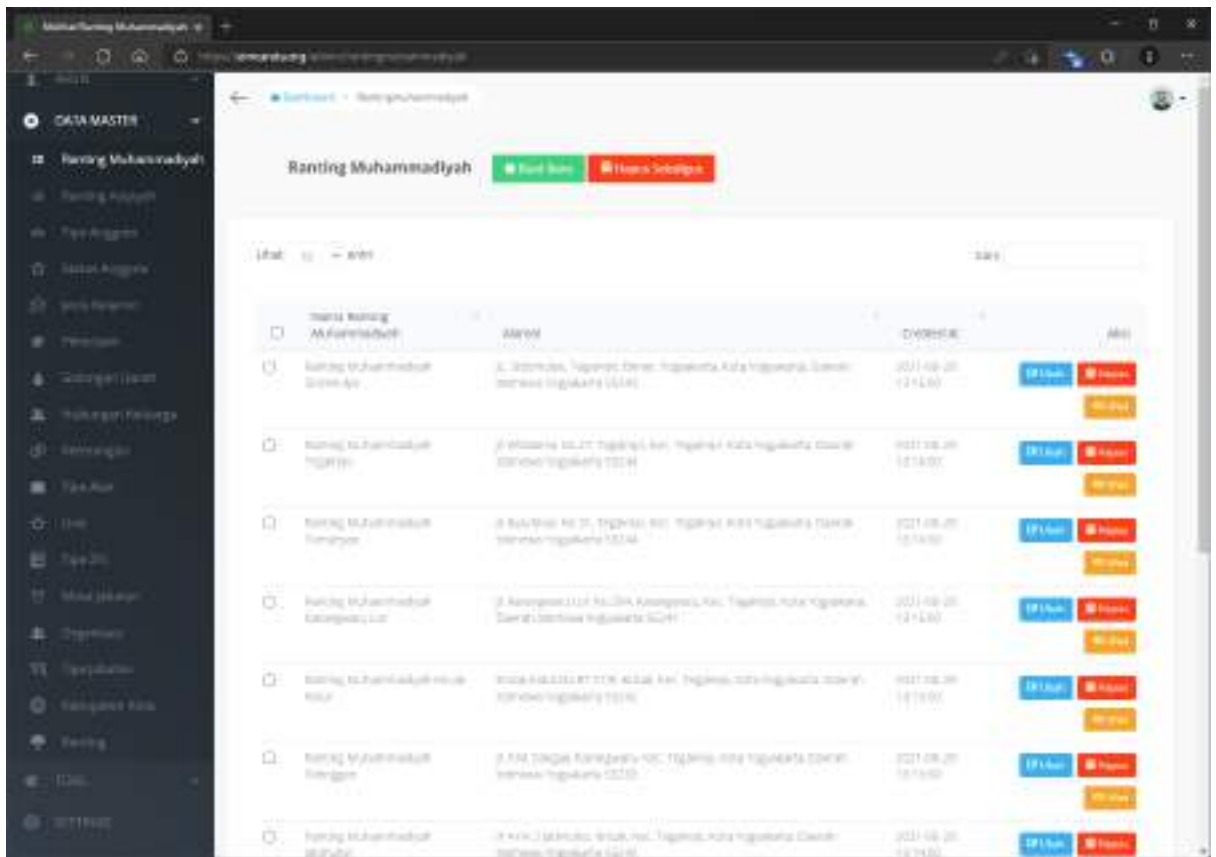
7. Akun



Gambar 27 Admin – Akun

Halaman Akun digunakan untuk mengelola akun serta rolenya yang digunakan untuk Login.

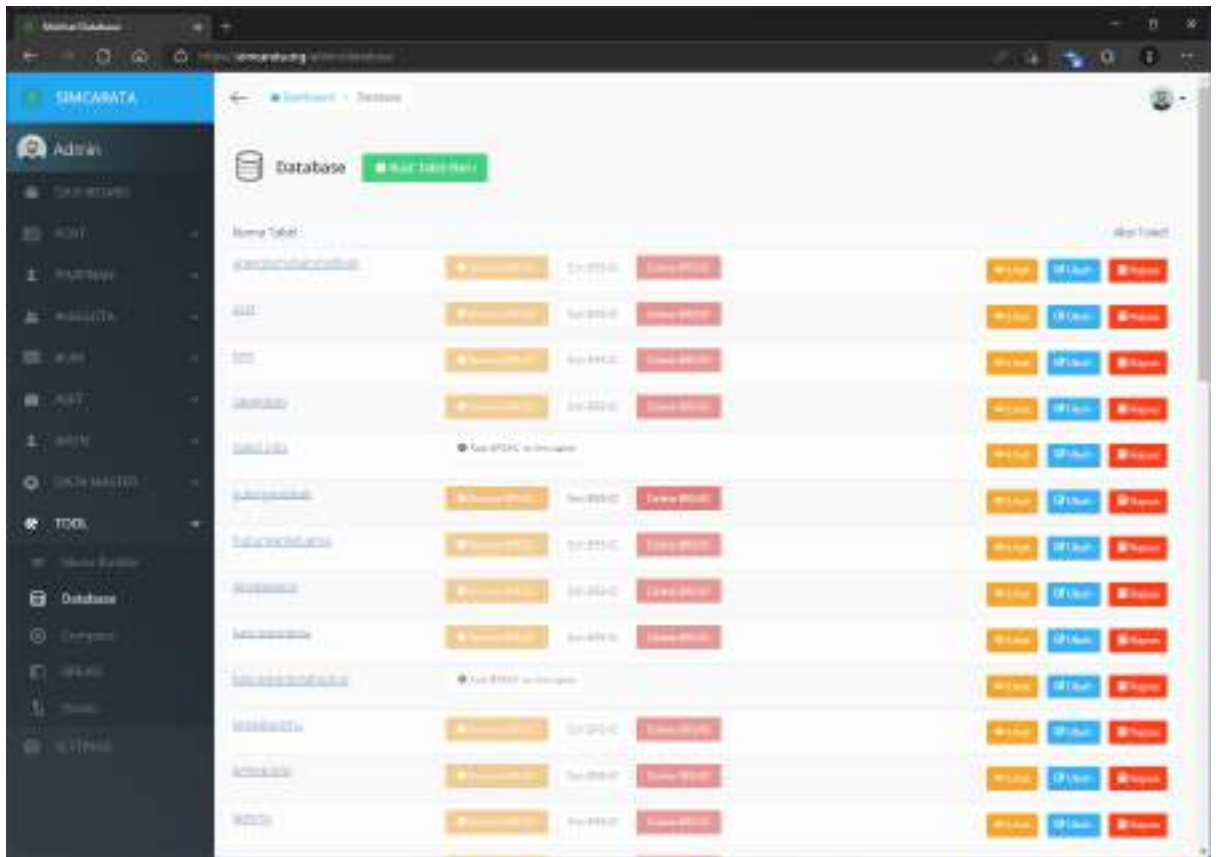
8. Data Master



Gambar 28 Admin - Data Master

Halaman Data Master dapat mengelola data-data yang ada pada sistem informasi Simcarata.

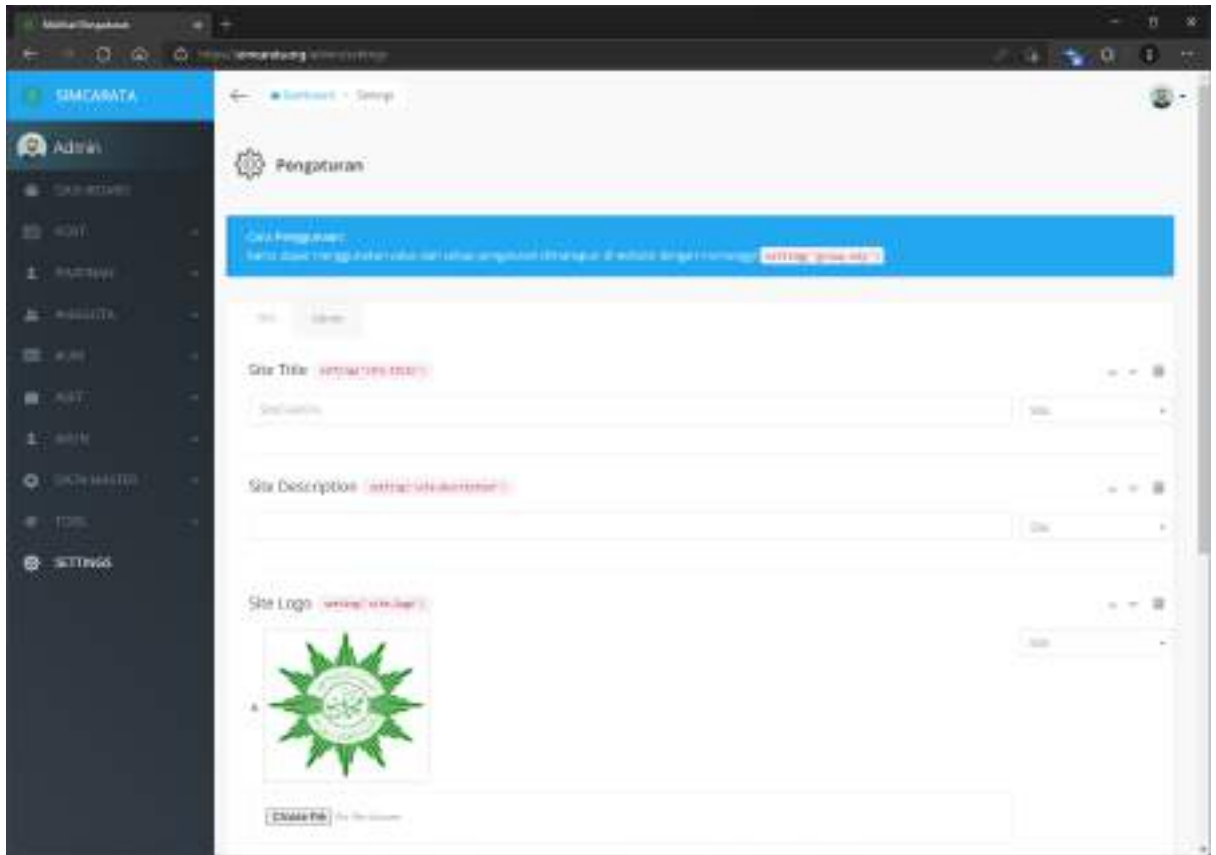
9. Tool



Gambar 29 Admin – Tool

Halaman Tool dapat mengelola database manager serta menu *bread*.

10. Settings



Gambar 30 Admin – Settings

Halaman Settings dapat mengatur tampilan umum website seperti *site logo* dan *site title*.

BAB VI PENUTUP

A. Kesimpulan

Setelah menyelesaikan tahap pembangunan sistem sesuai rancangan yang sudah disepakati, sistem dapat beroperasi seperti semestinya dan dapat memenuhi kebutuhan yang tercantum di Analisis Kebutuhan. Berdasarkan penelitian dapat disimpulkan sebagai berikut :

1. Dengan adanya Sistem Informasi Cabang Ranting Tegalrejo (SIMCARATA) ini, memudahkan admin dalam melakukan olah data anggota dan memberikan informasi kepada anggota Muhammadiyah di daerah Tegalrejo.
2. Dengan adanya SIMCARATA ini memudahkan anggota Muhammadiyah Tegalrejo untuk mengakses informasi terbaru.

B. Saran

Berdasarkan kesimpulan di atas, berikut saran yang mungkin dapat menjadi pertimbangan untuk meningkatkan efektivitas penggunaan sistem SIMCARATA ini :

1. Untuk Admin
 - 1) Dengan adanya sistem ini, diharapkan admin untuk melakukan update data secepatnya jika terdapat perubahan, sehingga data dalam sistem ini adalah faktual.
 - 2) Dengan adanya sistem ini, diharapkan admin untuk lebih aktif memberikan informasi informasi terbaru terkait PCM Tegalrejo sehingga informasi cepat tersampaikan kepada anggota.
2. Untuk Pengguna Non-Admin
 - 1) Dengan adanya sistem ini, diharapkan kepada anggota untuk sering meninjau sistem ini guna melihat pemberitahuan maupun informasi terkini mengenai PCM Tegalrejo.

DAFTAR PUSTAKA

- Adani, M. R., 2020. *Apa itu MySQL: Pengertian, Fungsi, beserta Kelebihan*. [Online]
Available at: <https://www.sekawanmedia.co.id/blog/pengertian-mysql/>
[Accessed Senin Desember 2021].
- Adani, M. R., 2020. *Pengenalan Apa Itu Framework dan Jenisnya untuk Web Development*. [Online]
Available at: <https://www.sekawanmedia.co.id/blog/pengertian-framework/>
[Accessed Senin Desember 2021].
- Adani, M. R., 2021. *Mengulas Penggunaan HTML dan Cara Kerjanya pada Web Development*.
[Online]
Available at: <https://www.sekawanmedia.co.id/blog/pengertian-html/>
[Accessed Senin Desember 2021].
- Adani, M. R., 2021. *Pengertian Sistem Informasi dan Contoh Penerapan pada Dunia Industri*.
[Online]
Available at: <https://www.sekawanmedia.co.id/blog/sistem-informasi/>
[Accessed Senin Desember 2021].
- Adani, M. R., 2021. *Tutorial Mudah Belajar Bahasa Pemrograman PHP untuk Pemula*. [Online]
Available at: <https://www.sekawanmedia.co.id/blog/pengertian-php/>
[Accessed Senin Desember 2021].
- Dewaweb Team, 2021. *Belajar CSS: Pengertian, Peran, Fungsi, dan Macamnya*. [Online]
Available at: <https://www.dewaweb.com/blog/belajar-css-pengertian-peran-fungsi-dan-macamnya/>
[Accessed Senin Desember 2021].
- NKD, F., 2020. *Laravel 8 : Apa Saja Fitur-Fitur Baru yang Ada di Dalamnya ?*. [Online]
Available at: <https://www.logique.co.id/blog/2020/10/01/laravel-8/>
[Accessed Senin Desember 2021].
- Ribeiro, C., 2017. *Voyager – Can an Admin UI Make Laravel Even More Approachable?*. [Online]
Available at: <https://www.sitepoint.com/voyager-can-an-admin-interface-make-laravel-even-more-approachable/>
[Accessed Senin Desember 2021].


```
File Edit Selection View Go Run Terminal Help
CatalogApiController.php - windows - Visual Studio Code

CatalogApiController.php
<?php
namespace App\Http\Controllers;

use Illuminate\Http\Request;
use Illuminate\Support\Facades\DB;

class CatalogApiController extends Controller
{
    public function post()
    {
        $users = DB::table('users')->orderBy('created_at', 'asc')->get();
        return view('catalogApiController', ['data_post' => $users]);
    }
}
```

```
File Edit Selection View Go Run Terminal Help
CatalogAdminApiController.php - windows - Visual Studio Code

CatalogAdminApiController.php
<?php
namespace App\Http\Controllers;

use App\Http\Controllers\Pagination;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\DB;

class CatalogAdminApiController extends Controller
{
    public function post()
    {
        $users = DB::table('users')->orderBy('created_at', 'asc')->get();
        return view('catalogAdminApiController', ['data_post' => $users]);
    }
}
```

```
Controller.php  
namespace App\Http\Controllers;  
  
use Illuminate\Foundation\Auth\Access\AuthorizesRequests;  
use Illuminate\Foundation\Http\DispatchesJobs;  
use Illuminate\Foundation\Validation\ValidatesRequests;  
use Illuminate\Routing\Controller as BaseController;  
  
class Controller extends BaseController  
{  
    use Illuminate\Routing\Controllers\UsesRequest;  
}
```

```
LAZISMController.php  
namespace App\Http\Controllers;  
  
use Illuminate\Routing\Controllers\UsesRequest;  
  
class LAZISMController extends Controller  
{  
    use Illuminate\Routing\Controllers\UsesRequest;  
  
    public function index()  
    {  
        $users = DB::table('users')->orderBy('created_at', 'desc')->get();  
        return view('index', ['data_index' => $users]);  
    }  
}
```

```
MapController.php - src/controllers - Visual Studio Code  
File Edit Selection View Go Run Terminal Help  
MapController.php  
<?php  
namespace App\Controllers;  
use Illuminate\Http\Request;  
use Illuminate\Support\Facades\DB;  
class MapController extends Controller  
{  
    public function render()  
    {  
        $users = DB::table('users')->orderBy('created_at', 'desc')->get();  
        return view('map', ['data_users' => $users]);  
    }  
}
```

```
AsyarakMajalahController.php - src/controllers - Visual Studio Code  
File Edit Selection View Go Run Terminal Help  
AsyarakMajalahController.php  
<?php  
namespace App\Controllers;  
use Illuminate\Http\Request;  
use Illuminate\Support\Facades\DB;  
class AsyarakMajalahController extends Controller  
{  
    public function render()  
    {  
        $users = DB::table('majalah')->orderBy('created_at', 'desc')->get();  
        return view('asyarakmajalah', ['data_maj' => $users]);  
    }  
}
```



```
File Edit Selection View Go Run Terminal Help
PendidikanManajemenKontrol.php - Laravel - Visual Studio Code

PendidikanManajemenKontrol.php
</php>
namespace App\Http\Controllers;

use Illuminate\Http\Request;
use Illuminate\Support\Facades\DB;

class PendidikanManajemenKontrol extends Controller
{
    public function post()
    {
        $users = DB::table('peminjaman')->orderby('created_at', 'desc')->get();
        return view('peminjamanManajemenKontrol', ['data_post' => $users]);
    }
}
```

```
File Edit Selection View Go Run Terminal Help
PeminjamanManajemenKontrol.php - Laravel - Visual Studio Code

PeminjamanManajemenKontrol.php
</php>
namespace App\Http\Controllers;

use Illuminate\Http\Request;
use Illuminate\Support\Facades\DB;

class PeminjamanManajemenKontrol extends Controller
{
    public function post()
    {
        $users = DB::table('peminjaman')->orderby('created_at', 'asc')->get();
        return view('peminjamanManajemenKontrol', ['data_post' => $users]);
    }

    public function post2()
    {
        $users = DB::table('peminjaman')->orderby('created_at', 'asc')->get();
        return view('peminjamanManajemenKontrol', ['data_post' => $users]);
    }
}
```

```

File Edit Selection View Go Run Terminal Help
PodControl.php - windows - Visual Studio Code

PodControl.php
<?php
namespace App\Http\Controllers;

use Illuminate\Http\Request;
use Illuminate\Support\Facades\DB;

class PostController extends Controller
{
    function index()
    {
        //return view('welcome');
        //return view('welcome', ['name' => 'John']);
        $users = DB::table('posts')->orderBy('created_at', 'desc')->get();
        return view('welcome', ['data' => $users]);
    }

    public function detail($id)
    {
        $article = DB::table('posts')->where('id', $id)->first();
        return view('detailpost', ['data' => $article]);
    }

    public function post(Request $request)
    {
        //($request->get('search'))
        $search=$request->search;
        $users = DB::table('posts')->where('title','like','%'.$search.'%')->orderBy('created_at', 'desc')->get();
        $users = DB::table('posts')->orderBy('created_at', 'desc')->paginate(5);
    }
}

```

```

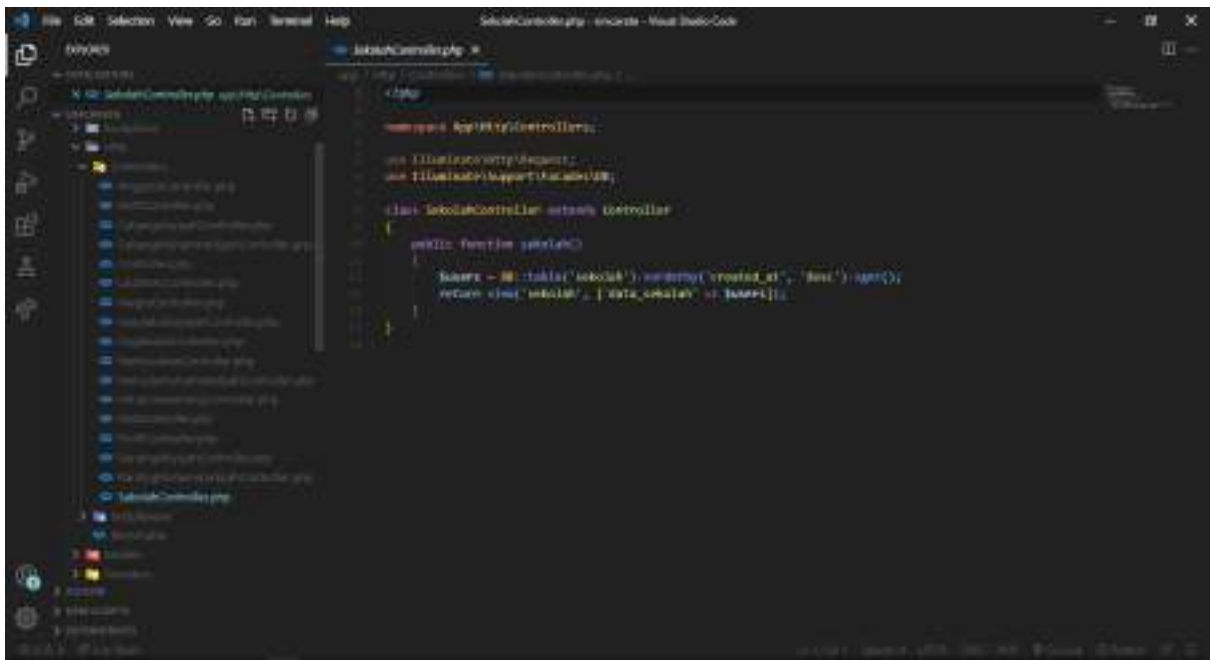
File Edit Selection View Go Run Terminal Help
PodControl.php - windows - Visual Studio Code

PodControl.php
<?php
namespace App\Http\Controllers;

use Illuminate\Http\Request;

class ProfileController extends Controller
{
    public function profile()
    {
        return view('profile');
    }
}

```

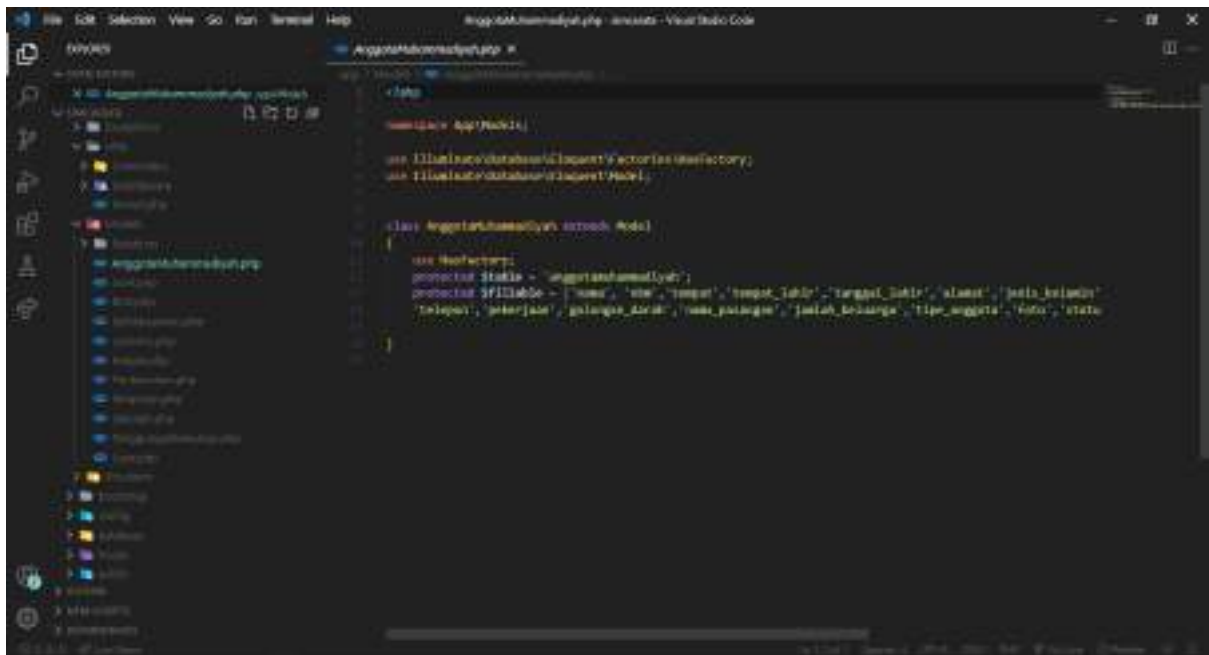
The image shows a screenshot of the Visual Studio Code editor. The interface includes a menu bar at the top with options like File, Edit, Selection, View, Go, Run, Terminal, and Help. The title bar indicates the current file is 'sekolahController.php' in the 'encore - Visual Studio Code' workspace. On the left, the Explorer sidebar shows a project structure for 'sekolah' with various sub-directories and files. The main editor area displays the following PHP code:

```
<?php
namespace App\Http\Controllers;

use Illuminate\Http\Request;
use Illuminate\Support\Facades\DB;

class SekolahController extends Controller
{
    public function sekolah()
    {
        $users = DB::table('sekolah')->orderBy('routed_at', 'desc')->get();
        return view('sekolah', ['data_sekolah' => $users]);
    }
}
```

B. Model

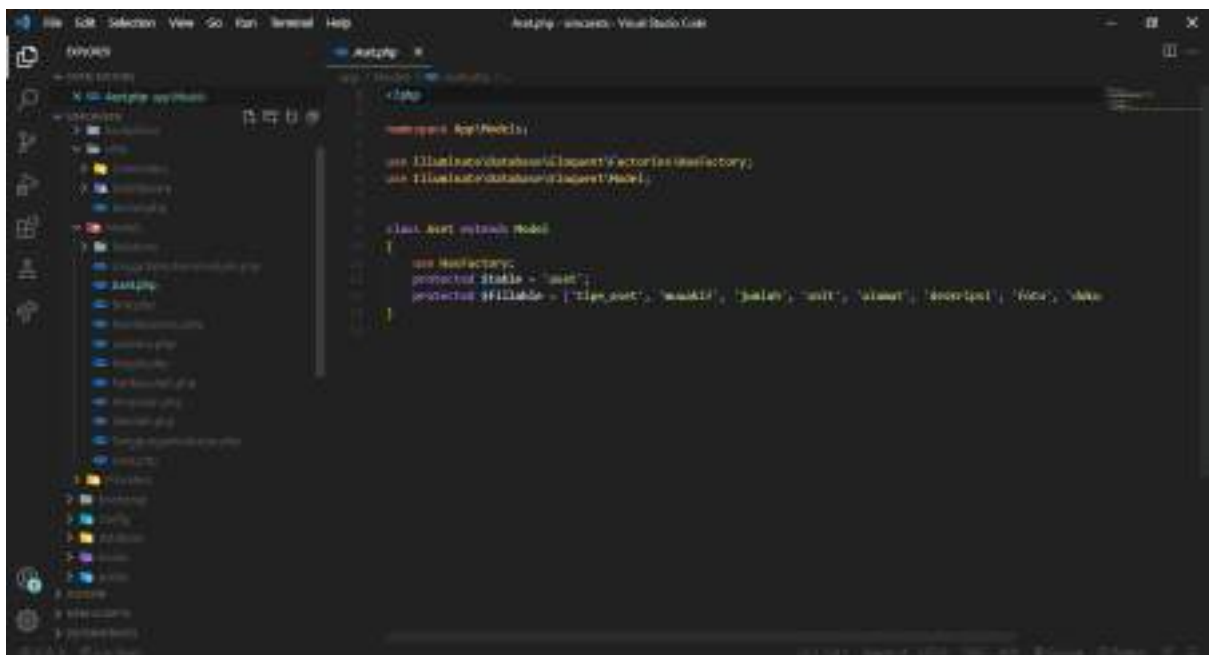


```
File Edit Selection View Go Run Terminal Help
Aggotahasiswa.php - windows - Visual Studio Code

<?php
namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Aggotahasiswa extends Model
{
    use HasFactory;
    protected $table = 'aggotahasiswa';
    protected $fillable = ['nama', 'nim', 'tempat_lahir', 'tanggal_lahir', 'alamat', 'jenis_kelamin',
        'telepon', 'pekerjaan', 'paling_suka', 'nama_pasangan', 'jumlah_anak', 'tipe_sugger', 'foto', 'status'];
}
```



```
File Edit Selection View Go Run Terminal Help
Asst.php - windows - Visual Studio Code

<?php
namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Asst extends Model
{
    use HasFactory;
    protected $table = 'asst';
    protected $fillable = ['tipe_asst', 'waktu', 'jumlah', 'unit', 'alamat', 'deskripsi', 'foto', 'nama'];
}
```

```

inteply
class inteply extends Mode {
  use ModelFactory;
  protected $statis = 'set';
  protected $fillable = ['nama', 'alamat', 'dokter', 'tanggalendir', 'foto'];
}

class inteply extends Mode {
  use ModelFactory;
  protected $statis = 'set';
  protected $fillable = ['nama', 'alamat', 'dokter', 'tanggalendir', 'foto'];
}

```

```

kalsobalok
class kalsobalok extends Mode {
  use ModelFactory;
  protected $statis = 'kalsobalok';
  protected $fillable = ['alamat', 'tipoln', 'nama', 'tanggalawal', 'perguru'];
}

```

```
File Edit Selection View Go Run Terminal Help
LaciumApp - src/main/scala/...

LaciumApp
  <App
    main(args: List[String]): Unit
    use IllustrationDatabaseClientModel
    use IllustrationDatabaseClientFactory

  class LaciumNetworkModel
  {
    use ModelFactory
    protected stable = 'Lacium'
    protected fillable = ['name', 'lastName']
  }
  }

File Edit Selection View Go Run Terminal Help
```

```
File Edit Selection View Go Run Terminal Help
MasjidApp - src/main/scala/...

MasjidApp
  <App
    main(args: List[String]): Unit
    use IllustrationDatabaseClientModel
    use IllustrationDatabaseClientFactory

  class MasjidNetworkModel
  {
    use ModelFactory
    protected stable = 'Masjid'
    protected fillable = ['name', 'lastName', 'tabah', 'jumlahMasuk']
  }
  }

File Edit Selection View Go Run Terminal Help
```

```
File Edit Selection View Go Run Terminal Help PartisanApp - Amarta - Visual Studio Code  
PartisanApp.java  
import java.util.ArrayList;  
import java.util.Observable;  
import java.util.Observer;  
import java.util.ObservableList;  
import java.util.ObservableProperty;  
import java.util.ObservablePropertyFactory;  
import java.util.ObservablePropertyFactoryImpl;  
  
class PartisanApp extends Model  
{  
    use ModelFactory;  
    protected static Partisan[] partisans;  
    protected ObservableList<ObservableProperty> observableProperties;  
}  
  
PartisanApp() {  
    partisans = new Partisan[10];  
    observableProperties = new ObservableList<ObservableProperty>();  
}
```

```
File Edit Selection View Go Run Terminal Help PartisanApp - Geonita - Visual Studio Code  
PartisanApp.java  
import java.util.ArrayList;  
import java.util.Observable;  
import java.util.Observer;  
import java.util.ObservableList;  
import java.util.ObservableProperty;  
import java.util.ObservablePropertyFactory;  
import java.util.ObservablePropertyFactoryImpl;  
  
class PartisanApp extends Model  
{  
    use ModelFactory;  
    protected static Partisan[] partisans;  
    protected ObservableList<ObservableProperty> observableProperties;  
}  
  
PartisanApp() {  
    partisans = new Partisan[10];  
    observableProperties = new ObservableList<ObservableProperty>();  
}
```


The screenshot shows the Visual Studio Code interface with the file explorer on the left and the code editor on the right. The file explorer shows a project structure with folders like 'src' and 'tests'. The code editor displays the content of 'Jalan.php'.

```
app / src / ... / Jalan.php  
  
<?php  
  
namespace App\Model;  
  
use Illuminate\Database\Eloquent\Model;  
use Illuminate\Database\Eloquent\Factories\Factory;  
  
class Jalan extends Model  
{  
    use HasFactory;  
    protected $table = 'jalan';  
    protected $fillable = ['nama', 'alamat', 'jenisjalan', 'jenisgerb', 'kecamatan', 'kota'];  
}
```

The screenshot shows the Visual Studio Code interface with the file explorer on the left and the code editor on the right. The file explorer shows a project structure with folders like 'src' and 'tests'. The code editor displays the content of 'TanggapanKampung.php'.

```
app / src / ... / TanggapanKampung.php  
  
<?php  
  
namespace App\Model;  
  
use Illuminate\Database\Eloquent\Factories\Factory;  
use Illuminate\Database\Eloquent\Model;  
  
class TanggapanKampung extends Model  
{  
    use HasFactory;  
    protected $table = 'tanggapanKampung';  
    protected $fillable = ['nama', 'ibu', 'namaKampung', 'nama', 'tempat_lahir', 'tanggal_lahir'];  
}
```

```
File Edit Selection View Go Run Terminal Help
UserApp - Visual Studio Code

UserApp
├── src
│   ├── database
│   ├── authentication
│   └── user
├── test
├── package.json
├── README.md
└── ...

src
├── database
│   ├── database.js
│   ├── user.js
│   └── ...
├── authentication
│   ├── authentication.js
│   └── ...
└── user
    ├── user.js
    └── ...

src/user/user.js

class User extends UserModel {
  constructor(email, password) {
    super(email, password);
  }

  protected $fillable = [
    'email',
    'password',
  ];
}
```

```
File Edit Selection View Go Run Terminal Help
GolonganApp - Visual Studio Code

GolonganApp
├── src
│   ├── database
│   ├── golongan
│   └── ...
├── test
├── package.json
├── README.md
└── ...

src
├── database
│   ├── database.js
│   ├── golongan.js
│   └── ...
├── golongan
│   ├── golongan.js
│   └── ...
└── ...

src/golongan/golongan.js

class Golongan extends Model {
  constructor(nama, alamat) {
    super(nama, alamat);
  }

  protected $table = 'golongan';
  protected $fillable = ['nama'];
}
```

The screenshot shows the Visual Studio Code editor with the file 'HubungKelembagaan.php' open. The code defines a class 'HubungKelembagaan' that inherits from 'App\Model'. It includes two protected static properties: 'table' set to 'HubungKelembagaan' and 'fillable' set to an array containing 'Hubungan'.

```
<?php
namespace App\Model;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class HubungKelembagaan extends Model
{
    use HasFactory;
    protected $table = 'HubungKelembagaan';
    protected $fillable = ['Hubungan'];
}
```

The screenshot shows the Visual Studio Code editor with the file 'JenisKategori.php' open. The code defines a class 'JenisKategori' that inherits from 'App\Model'. It includes two protected static properties: 'table' set to 'JenisKategori' and 'fillable' set to an array containing 'JenisKategori'.

```
<?php
namespace App\Model;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class JenisKategori extends Model
{
    use HasFactory;
    protected $table = 'JenisKategori';
    protected $fillable = ['JenisKategori'];
}
```


The screenshot shows the Visual Studio Code interface with the file 'Masjid.php' open. The code defines a class 'Masjid' with a namespace 'App\Models' and uses 'Illuminate\Database\Eloquent' and 'Illuminate\Database\Eloquent\Factories'. The class 'Masjid extends Model' has two protected static attributes: 'table' set to 'masjid' and 'fillable' set to ['nama'].

```
namespace App\Models;

use Illuminate\Database\Eloquent;
use Illuminate\Database\Eloquent\Factories;

class Masjid extends Model
{
    use Factory;
    protected $table = 'masjid';
    protected $fillable = ['nama'];
}
```

The screenshot shows the Visual Studio Code interface with the file 'Organisasi.php' open. The code defines a class 'Organisasi' with a namespace 'App\Models' and uses 'Illuminate\Database\Eloquent' and 'Illuminate\Database\Eloquent\Factories'. The class 'Organisasi extends Model' has two protected static attributes: 'table' set to 'organisasi' and 'fillable' set to ['nama'].

```
namespace App\Models;

use Illuminate\Database\Eloquent;
use Illuminate\Database\Eloquent\Factories;

class Organisasi extends Model
{
    use Factory;
    protected $table = 'organisasi';
    protected $fillable = ['nama'];
}
```

The screenshot shows the Visual Studio Code interface for the 'melayan.php' application. The file explorer on the left shows the project structure, with 'Pekerjaan.php' selected under the 'model' directory. The main editor displays the following PHP code:

```
<?php
namespace App\Model;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Pekerjaan extends Model
{
    use HasFactory;
    protected $table = 'pekerjaan';
    protected $fillable = ['nama_pekerjaan'];
}
```

The screenshot shows the Visual Studio Code interface for the 'melayan.php' application. The file explorer on the left shows the project structure, with 'Waiting.php' selected under the 'model' directory. The main editor displays the following PHP code:

```
<?php
namespace App\Model;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class Waiting extends Model
{
    use HasFactory;
    protected $table = 'waiting';
    protected $fillable = ['nama_waiting'];
}
```

The screenshot shows the Visual Studio Code interface with the file explorer on the left displaying a project structure. The main editor window shows the code for the `KartingKategori.php` file. The code includes the following:

```
<?php  
namespace App\Models; use Illuminate\Database\Eloquent\Factories\HasFactory; use Illuminate\Database\Eloquent\Model; class KartingKategori extends Model { use HasFactory; protected $table = 'karting_kategori'; protected $fillable = ['nama_kategori']; }</pre>
```

The screenshot shows the Visual Studio Code interface with the file explorer on the left displaying a project structure. The main editor window shows the code for the `KartingAramaslyah.php` file. The code includes the following:

```
<?php  
namespace App\Models; use Illuminate\Database\Eloquent\Factories\HasFactory; use Illuminate\Database\Eloquent\Model; class KartingAramaslyah extends Model { use HasFactory; protected $table = 'karting_aramaslyah'; protected $fillable = ['nama_aramaslyah']; }</pre>
```

The screenshot shows the Visual Studio Code interface with a project named 'StatoAggregato'. The file explorer on the left shows a directory structure with a 'Stato' folder containing several files. The main editor displays the implementation of the 'Stato' class in 'Stato.cs'. The code includes a namespace declaration, using statements for 'IRepository' and 'IModel', and the class definition with a constructor and a 'GetStato' method.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using IRepository;
using IModel;

namespace App.Models
{
    class StatoAggregato : IModel
    {
        IRepository repository;

        public StatoAggregato(IRepository repository)
        {
            this.repository = repository;
        }

        public void GetStato()
        {
            // Implementation logic
        }
    }
}
```

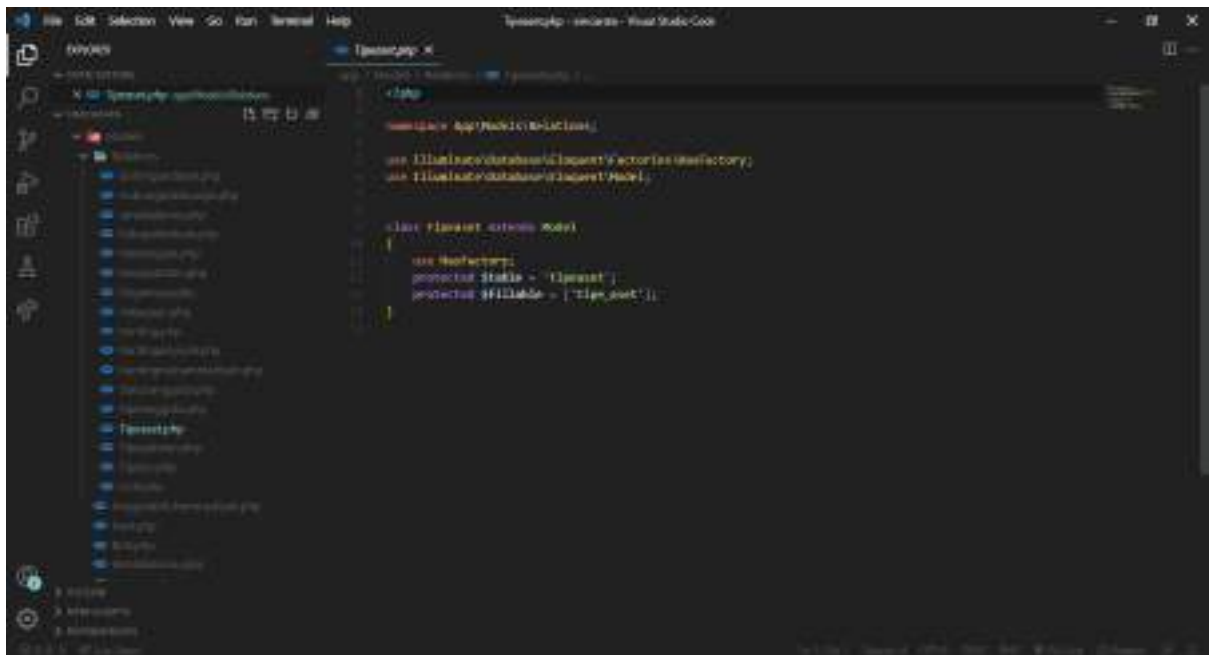
The screenshot shows the Visual Studio Code interface with a project named 'TipoAggregato'. The file explorer on the left shows a directory structure with a 'Tipo' folder containing several files. The main editor displays the implementation of the 'Tipo' class in 'Tipo.cs'. The code includes a namespace declaration, using statements for 'IRepository' and 'IModel', and the class definition with a constructor and a 'GetTipo' method.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using IRepository;
using IModel;

namespace App.Models
{
    class TipoAggregato : IModel
    {
        IRepository repository;

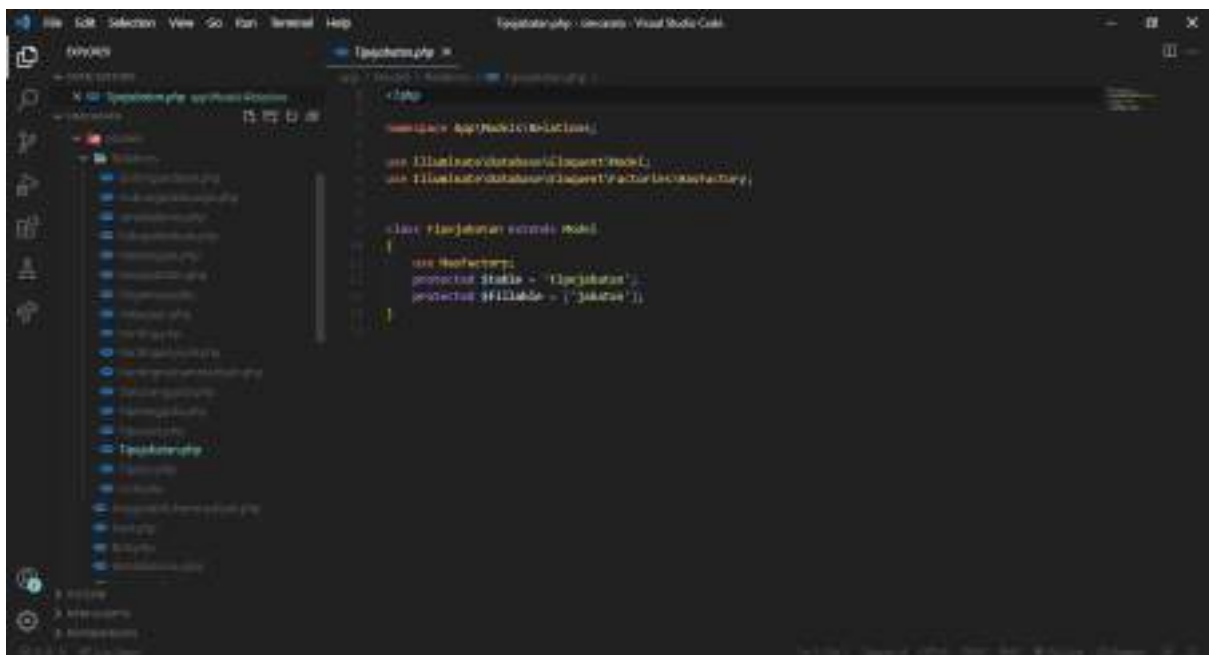
        public TipoAggregato(IRepository repository)
        {
            this.repository = repository;
        }

        public void GetTipo()
        {
            // Implementation logic
        }
    }
}
```

The screenshot shows the Visual Studio Code editor with a project named 'TipesApp'. The file explorer on the left shows a package structure with 'TipesApp' containing 'TipesApp' and 'TipesApp'. The main editor displays the code for 'TipesApp.java'.

```
<?xml version="1.0" encoding="UTF-8" ?>
<code>
</code>
</pre>
```



The screenshot shows the Visual Studio Code editor with a project named 'TipesApp'. The file explorer on the left shows a package structure with 'TipesApp' containing 'TipesApp' and 'TipesApp'. The main editor displays the code for 'TipesApp.java'.

```
<?xml version="1.0" encoding="UTF-8" ?>
<code>
</code>
</pre>
```

```
using App/Model/Relation;  
using IllustrateDatabase/QueryModel;  
using IllustrateDatabase/QueryFactory;  
  
class Tspecify : IModel  
{  
    IFactory Factory;  
    protected static string Title;  
    protected bool Fillable = true;  
}
```

```
using App/Model/Relation;  
using IllustrateDatabase/QueryFactory;  
using IllustrateDatabase/QueryModel;  
  
class Intspecy : IModel  
{  
    IFactory Factory;  
    protected static string Title;  
    protected bool Fillable = false;  
}
```

C. View

```
const express = require('express');
const http = require('http');
const path = require('path');

const app = express();

app.use(express.json());
app.use(express.urlencoded({ extended: true }));
app.use(path.resolve(__dirname, 'public'));

const PORT = 3000;

const server = http.createServer(app);

const startServer = () => {
  server.listen(PORT, () => {
    console.log(`Server is running on port ${PORT}`);
  });
};

const stopServer = () => {
  server.close(() => {
    console.log('Server is stopped');
  });
};

const startClient = () => {
  // Client logic
};

const stopClient = () => {
  // Client logic
};

const start = () => {
  startServer();
  startClient();
};

const stop = () => {
  stopServer();
  stopClient();
};

module.exports = { start, stop };

const { start, stop } = require('./app.js');

start();
```

```
const express = require('express');
const http = require('http');
const path = require('path');

const app = express();

app.use(express.json());
app.use(express.urlencoded({ extended: true }));
app.use(path.resolve(__dirname, 'public'));

const PORT = 3000;

const server = http.createServer(app);

const startServer = () => {
  server.listen(PORT, () => {
    console.log(`Server is running on port ${PORT}`);
  });
};

const stopServer = () => {
  server.close(() => {
    console.log('Server is stopped');
  });
};

const startClient = () => {
  // Client logic
};

const stopClient = () => {
  // Client logic
};

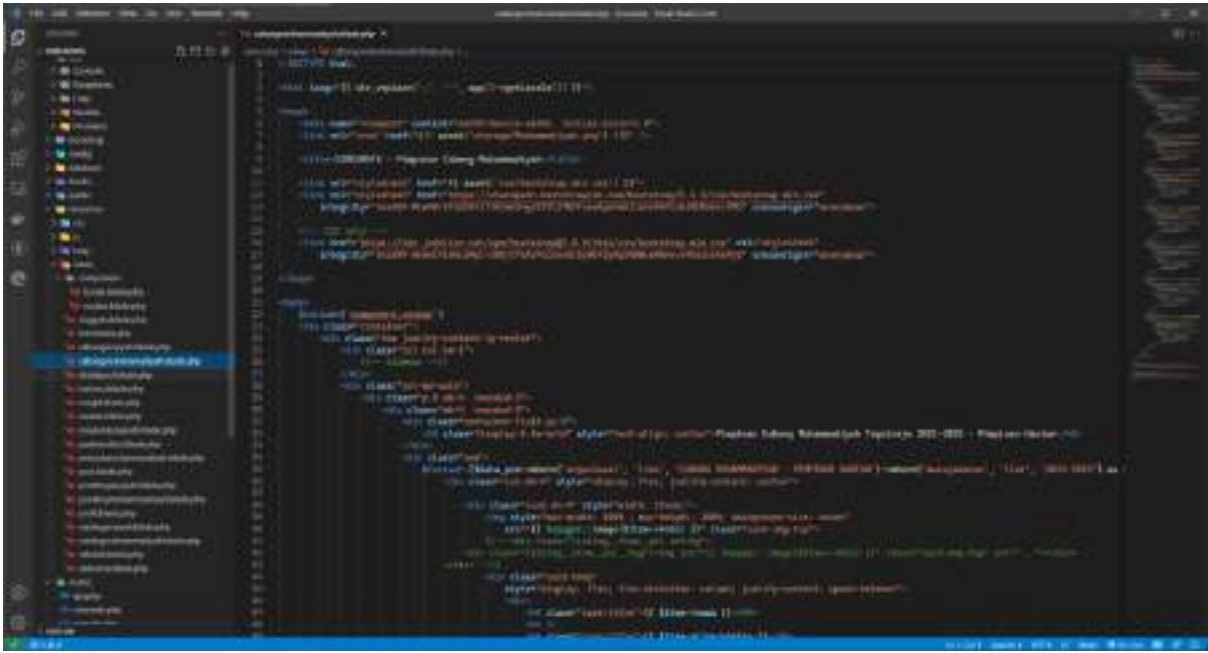
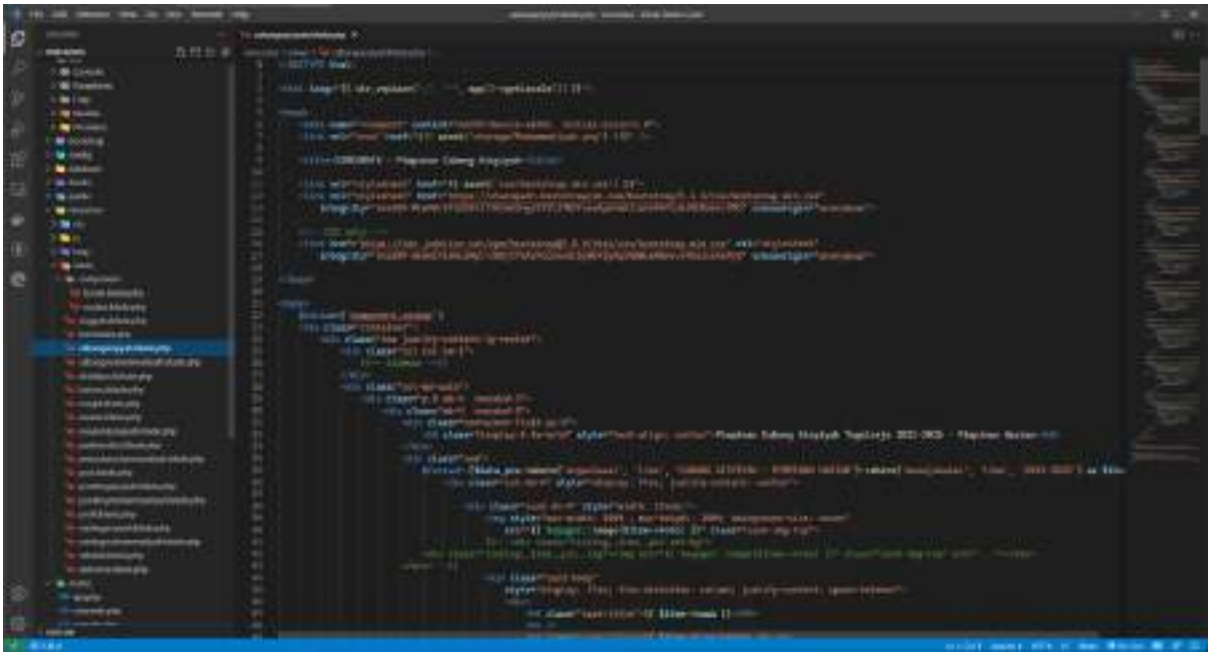
const start = () => {
  startServer();
  startClient();
};

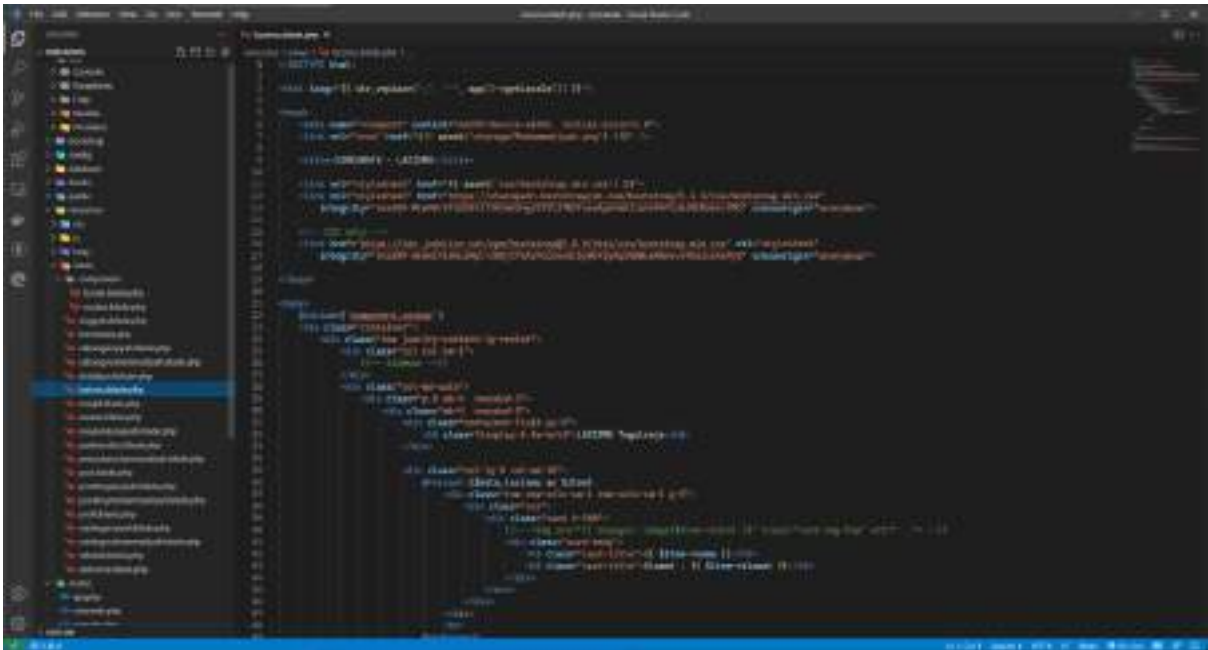
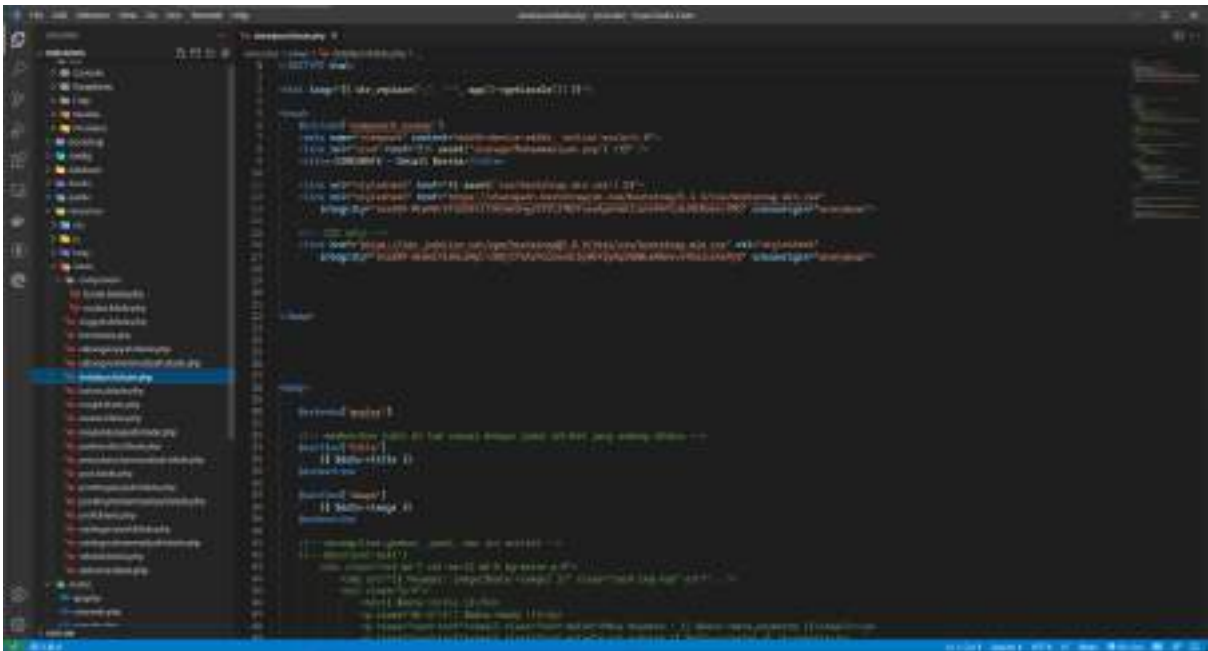
const stop = () => {
  stopServer();
  stopClient();
};

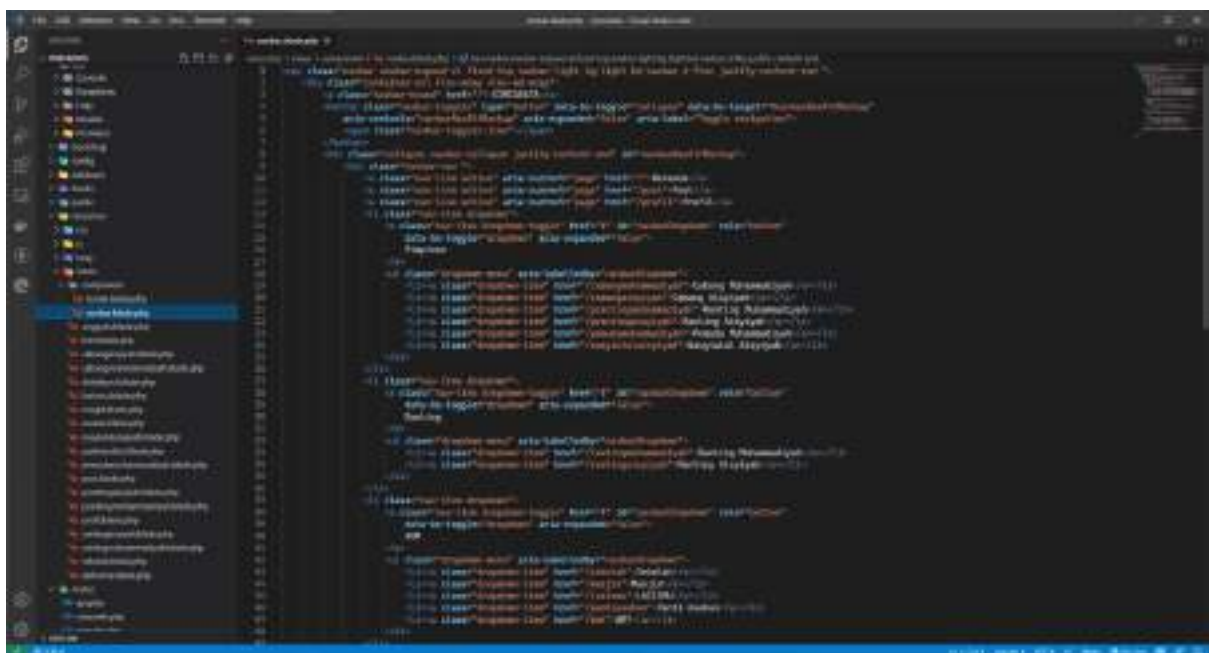
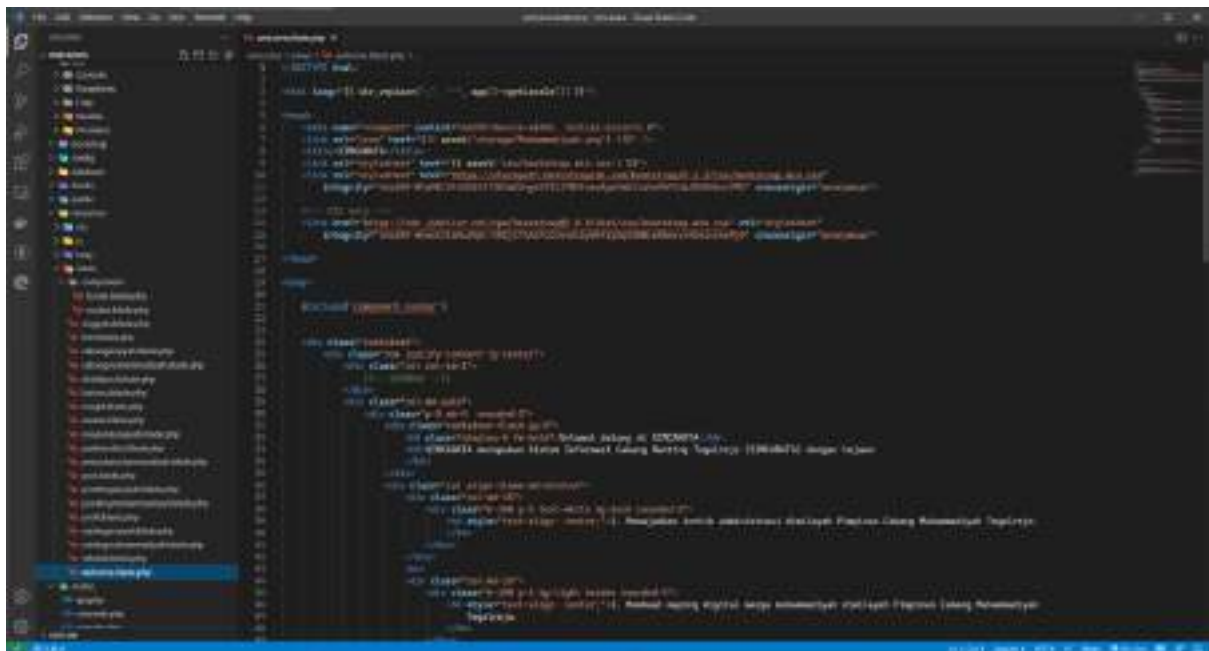
module.exports = { start, stop };

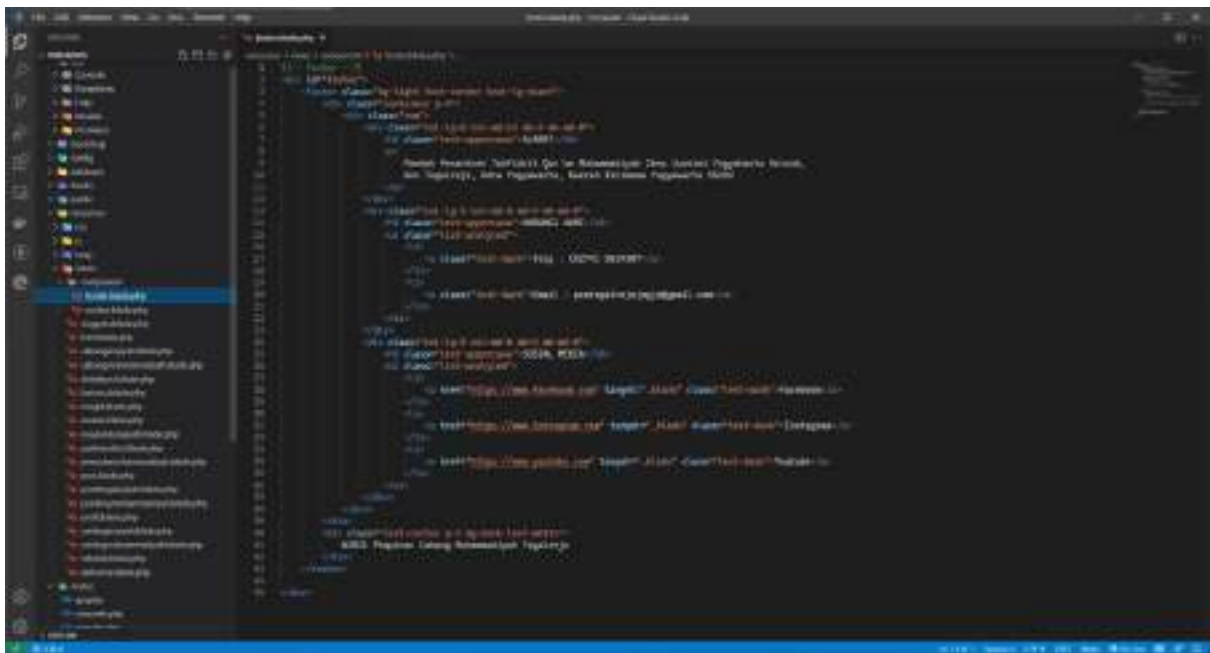
const { start, stop } = require('./index.js');

start();
```

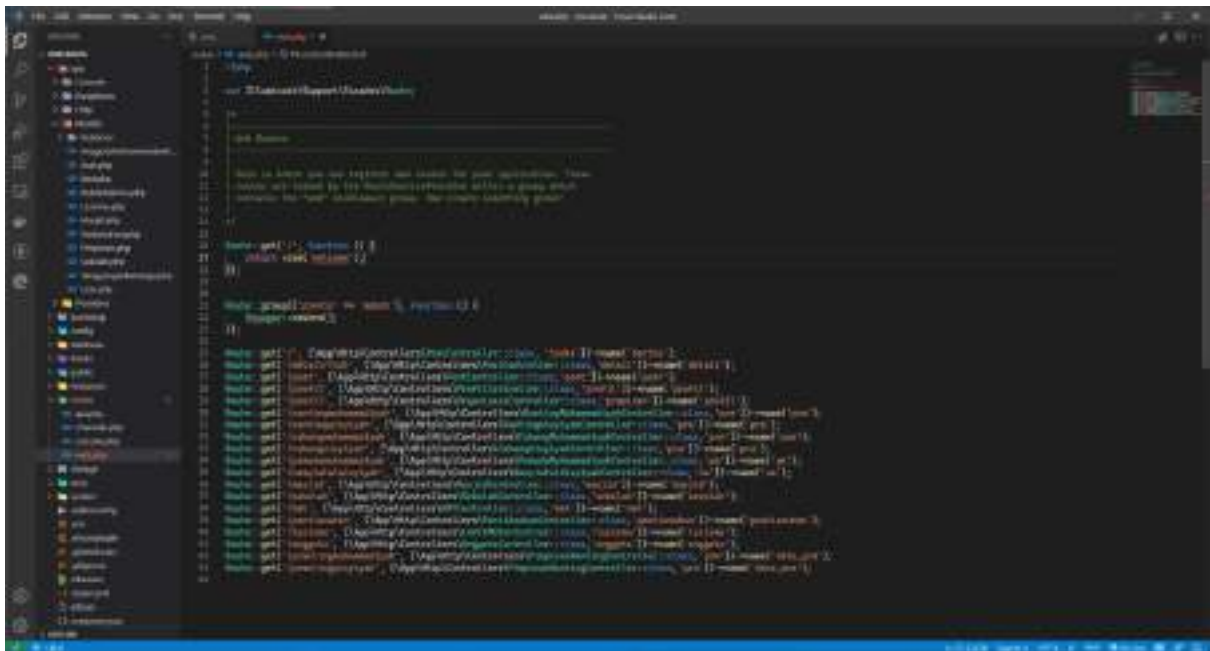








D. Route



```
const express = require('express');
const router = express.Router();

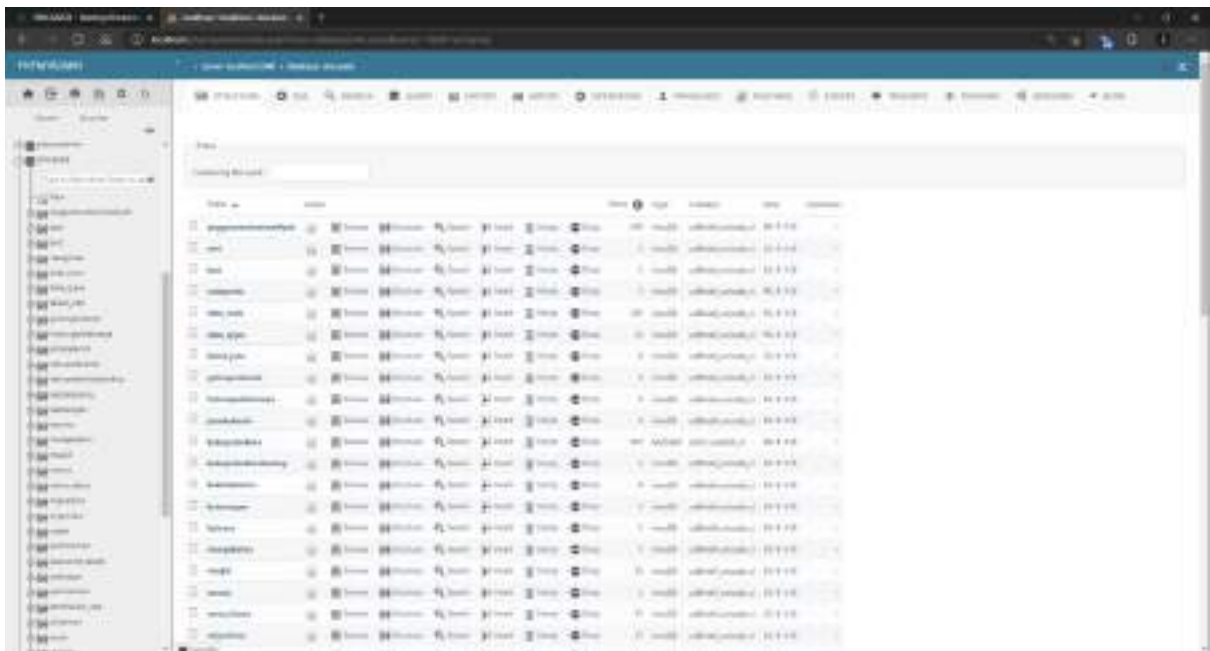
// Routes
// ...

router.get('/', function (req, res) {
  res.render('index');
});

// ...

module.exports = router;
```

E. Database



ID	Name	Email	Password	Role	Status	Created At	Updated At
1	admin	admin@company.com	admin	admin	active	2023-01-01 00:00:00	2023-01-01 00:00:00
2	user	user@company.com	user	user	active	2023-01-01 00:00:00	2023-01-01 00:00:00
3	guest	guest@company.com	guest	guest	inactive	2023-01-01 00:00:00	2023-01-01 00:00:00
4	admin	admin@company.com	admin	admin	active	2023-01-01 00:00:00	2023-01-01 00:00:00
5	user	user@company.com	user	user	active	2023-01-01 00:00:00	2023-01-01 00:00:00
6	guest	guest@company.com	guest	guest	inactive	2023-01-01 00:00:00	2023-01-01 00:00:00
7	admin	admin@company.com	admin	admin	active	2023-01-01 00:00:00	2023-01-01 00:00:00
8	user	user@company.com	user	user	active	2023-01-01 00:00:00	2023-01-01 00:00:00
9	guest	guest@company.com	guest	guest	inactive	2023-01-01 00:00:00	2023-01-01 00:00:00
10	admin	admin@company.com	admin	admin	active	2023-01-01 00:00:00	2023-01-01 00:00:00
11	user	user@company.com	user	user	active	2023-01-01 00:00:00	2023-01-01 00:00:00
12	guest	guest@company.com	guest	guest	inactive	2023-01-01 00:00:00	2023-01-01 00:00:00
13	admin	admin@company.com	admin	admin	active	2023-01-01 00:00:00	2023-01-01 00:00:00
14	user	user@company.com	user	user	active	2023-01-01 00:00:00	2023-01-01 00:00:00
15	guest	guest@company.com	guest	guest	inactive	2023-01-01 00:00:00	2023-01-01 00:00:00
16	admin	admin@company.com	admin	admin	active	2023-01-01 00:00:00	2023-01-01 00:00:00
17	user	user@company.com	user	user	active	2023-01-01 00:00:00	2023-01-01 00:00:00
18	guest	guest@company.com	guest	guest	inactive	2023-01-01 00:00:00	2023-01-01 00:00:00
19	admin	admin@company.com	admin	admin	active	2023-01-01 00:00:00	2023-01-01 00:00:00
20	user	user@company.com	user	user	active	2023-01-01 00:00:00	2023-01-01 00:00:00