

LAMPIRAN

Lampiran 1. Surat Kesepakatan Bimbingan Skripsi

SURAT KESEPAKATAN BIMBINGAN SKRIPSI

Kami yang bertanda tangan dibawah ini:

Pihak Pertama

Nama : Mohamad Khadik
NIM : 21090055
Program Studi : Sarjana Terapan Teknik Informatika

Pihak Kedua

Nama : Ir. Ginanjar Wiro Sasmito, M.Kom., IPM., ASEAN Eng.
Status : Dosen
NIDN : 0613028601
Jabatan Fungsional : Lektor Kepala
Pangkat/Golongan : Penata Tk. I/III-d

Pada hari ini Rabu tanggal 05 Maret 2025 telah terjadi sebuah kesepakatan bahwa Pihak Kedua bersedia menjadi Pembimbing I Skripsi Pihak Pertama dengan syarat Pihak Pertama wajib menyelesaikan Skripsi maksimal pada bulan Juni 2025, jika melebihi batas waktu maka Pihak Kedua berhak untuk tidak melanjutkan proses bimbingan Skripsi. Adapun waktu dan tempat pelaksanaan disepakati antar pihak.

Demikian kesepakatan ini dibuat dengan penuh kesadaran guna kelancaran penyelesaian Skripsi.

Tegal, 17 Maret 2025

Pihak Pertama



Mohamad Khadik

Pihak Kedua



Ir. Ginanjar Wiro Sasmito, M.Kom., IPM.,
ASEAN Eng.

Mengetahui
Ketua Program Studi Sarjana Terapan Teknik Informatika



Dyah Apriliani, S.T., M.Kom.
NIPY. 09.015.225

SURAT KESEPAKATAN BIMBINGAN SKRIPSI

Kami yang bertanda tangan dibawah ini:

Pihak Pertama

Nama : Mohamad Khadik
NIM : 21090055
Program Studi : Sarjana Terapan Teknik Informatika

Pihak Kedua

Nama : Muhammad Fikri Hidayattullah, S.T., M.Kom.
Status : Dosen
NIDN : 0623108801
Jabatan Fungsional : Lektor
Pangkat/Golongan : Penata Tk. I/III-d

Pada hari ini Kamis tanggal 13 Maret 2025 telah terjadi sebuah kesepakatan bahwa Pihak Kedua bersedia menjadi Pembimbing I Skripsi Pihak Pertama dengan syarat Pihak Pertama wajib menyelesaikan Skripsi maksimal pada bulan Juni 2025, jika melebihi batas waktu maka Pihak Kedua berhak untuk tidak melanjutkan proses bimbingan Skripsi. Adapun waktu dan tempat pelaksanaan disepakati antar pihak.

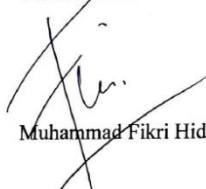
Demikian kesepakatan ini dibuat dengan penuh kesadaran guna kelancaran penyelesaian Skripsi.

Tegal, 17 Maret 2025

Pihak Pertama


Mohamad Khadik

Pihak Kedua


Muhammad Fikri Hidayattullah, S.T., M.Kom

Mengetahui

Ketua Program Studi Sarjana Terapan Teknik Informatika



Dyah Aprilia, S.T., M.Kom.
NIPY. 09.015.225

Lampiran 2. Surat Pernyataan Pengajuan HKI

SURAT PERNYATAAN

Yang bertanda tangan di bawah ini, pemegang hak cipta:

1. N a m a : Mohamad Khadik
Kewarganegaraan : Indonesia
Alamat : Desa Luwungragi RT.04/RW.07, kecamatan Bulakamba, Kabupaten Brebes,
Jawa Tengah 52253
2. N a m a : Ir. Ginanjar Wiro Sasmito, S.Kom., M.Kom
Kewarganegaraan : Indonesia
Alamat : Desa Kluwut RT.003/002, Kecamatan Bulakamba, Kabupaten Brebes,
Provinsi Jawa Tengah, 52253
3. N a m a : Muhammad Fikri Hidayattullah, S.T., M.Kom
Kewarganegaraan : Indonesia
Alamat : Jl. Glatik No. 68 Randugunting, Kota Tegal, Jawa Tengah 52131

Dengan ini menyatakan bahwa:

1. Karya Cipta yang saya mohonkan:
Berupa : Program Komputer
Berjudul : Masjid: Pengembangan Sistem Multi-Tenant pada Manajemen Masjid
menggunakan Sistem Pendukung Keputusan dengan Metode SAW
 - Tidak meniru dan tidak sama secara esensial dengan Karya Cipta milik pihak lain atau obyek kekayaan intelektual lainnya sebagaimana dimaksud dalam Pasal 68 ayat (2);
 - Bukan merupakan Ekspresi Budaya Tradisional sebagaimana dimaksud dalam Pasal 38;
 - Bukan merupakan Ciptaan yang tidak diketahui penciptanya sebagaimana dimaksud dalam Pasal 39;
 - Bukan merupakan hasil karya yang tidak dilindungi Hak Cipta sebagaimana dimaksud dalam Pasal 41 dan 42;
 - Bukan merupakan Ciptaan seni lukis yang berupa logo atau tanda pembeda yang digunakan sebagai merek dalam perdagangan barang/jasa atau digunakan sebagai lambang organisasi, badan usaha, atau badan hukum sebagaimana dimaksud dalam Pasal 65 dan;
 - Bukan merupakan Ciptaan yang melanggar norma agama, norma susila, ketertiban umum, pertahanan dan keamanan negara atau melanggar peraturan perundang-undangan sebagaimana dimaksud dalam Pasal 74 ayat (1) huruf d Undang-Undang Nomor 28 Tahun 2014 tentang Hak Cipta.
2. Sebagai pemohon mempunyai kewajiban untuk menyimpan asli contoh ciptaan yang dimohonkan dan harus memberikan apabila dibutuhkan untuk kepentingan penyelesaian sengketa perdata maupun pidana sesuai dengan ketentuan perundang-undangan.

3. Karya Cipta yang saya mohonkan pada Angka 1 tersebut di atas tidak pernah dan tidak sedang dalam sengketa pidana dan/atau perdata di Pengadilan.
4. Dalam hal ketentuan sebagaimana dimaksud dalam Angka 1 dan Angka 3 tersebut di atas saya / kami langgar, maka saya / kami bersedia secara sukarela bahwa:
 - a. permohonan karya cipta yang saya ajukan dianggap ditarik kembali; atau
 - b. Karya Cipta yang telah terdaftar dalam Daftar Umum Ciptaan Direktorat Hak Cipta, Direktorat Jenderal Hak Kekayaan Intelektual, Kementerian Hukum Dan Hak Asasi Manusia R.I dihapuskan sesuai dengan ketentuan perundang-undangan yang berlaku.
 - c. Dalam hal kepemilikan Hak Cipta yang dimohonkan secara elektronik sedang dalam berperkara dan/atau sedang dalam gugatan di Pengadilan maka status kepemilikan surat pencatatan elektronik tersebut ditangguhkan menunggu putusan Pengadilan yang berkekuatan hukum tetap.

Demikian Surat pernyataan ini saya/kami buat dengan sebenarnya dan untuk dipergunakan sebagaimana mestinya.

Tegal, 20 Juni 2025



(Mohamad Khadik)

(Ir. Ginanjar Wiro Sasmito, S.Kom., M.Kom)

(Muhammad Wikri Hidayattullah, S. T., M. Kom)

Lampiran 3. Surat Pengalihan HKI

SURAT PENGALIHAN HAK CIPTA

Yang bertanda tangan di bawah ini :

N a m a : Mohamad Khadik
Kewarganegaraan : Indonesia
Alamat : Desa Luwungragi RT.04/RW.07, kecamatan Bulakamba, Kabupaten Brebes, Jawa Tengah 52253

N a m a : Ir. Ginanjar Wiro Sasmito, S.Kom., M.Kom.
Kewarganegaraan : Indonesia
Alamat : Desa Kluwut RT.003/002, Kecamatan Bulakamba, Kabupaten Brebes, Provinsi Jawa Tengah, 52253

N a m a : Muhammad Fikri Hidayattullah, S.T., M.Kom.
Kewarganegaraan : Indonesia
Alamat : Jl. Glatik No. 68 Randugunting, Kota Tegal, Jawa Tengah 52131

Adalah **Pihak I** selaku pencipta, dengan ini menyerahkan karya ciptaan saya kepada :

N a m a : Pusat Penelitian dan Pengabdian Masyarakat (P3M)
Politeknik Harapan Bersama
Alamat : Jalan Mataram Nomor 9, Kelurahan Pesurungan Lor,
Kecamatan Margadana, Kota Tegal

Adalah **Pihak II** selaku Pemegang Hak Cipta berupa Program Komputer dengan judul "Masajid: Pengembangan Sistem Multi-Tenant pada Manajemen Masjid menggunakan Sistem Pendukung Keputusan dengan Metode SAW" untuk didaftarkan di Direktorat Hak Cipta dan Desain Industri, Direktorat Jenderal Kekayaan Intelektual, Kementerian Hukum dan Hak Asasi Manusia Republik Indonesia.

Demikianlah surat pengalihan hak ini kami buat, agar dapat dipergunakan sebagaimana mestinya.

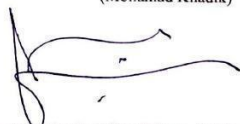
Tegal, 20 Juni 2025

Pemegang Hak Cipta
Kepala P3M


(Muhammad Fikri Hidayattullah, S. T., M. Kom.)

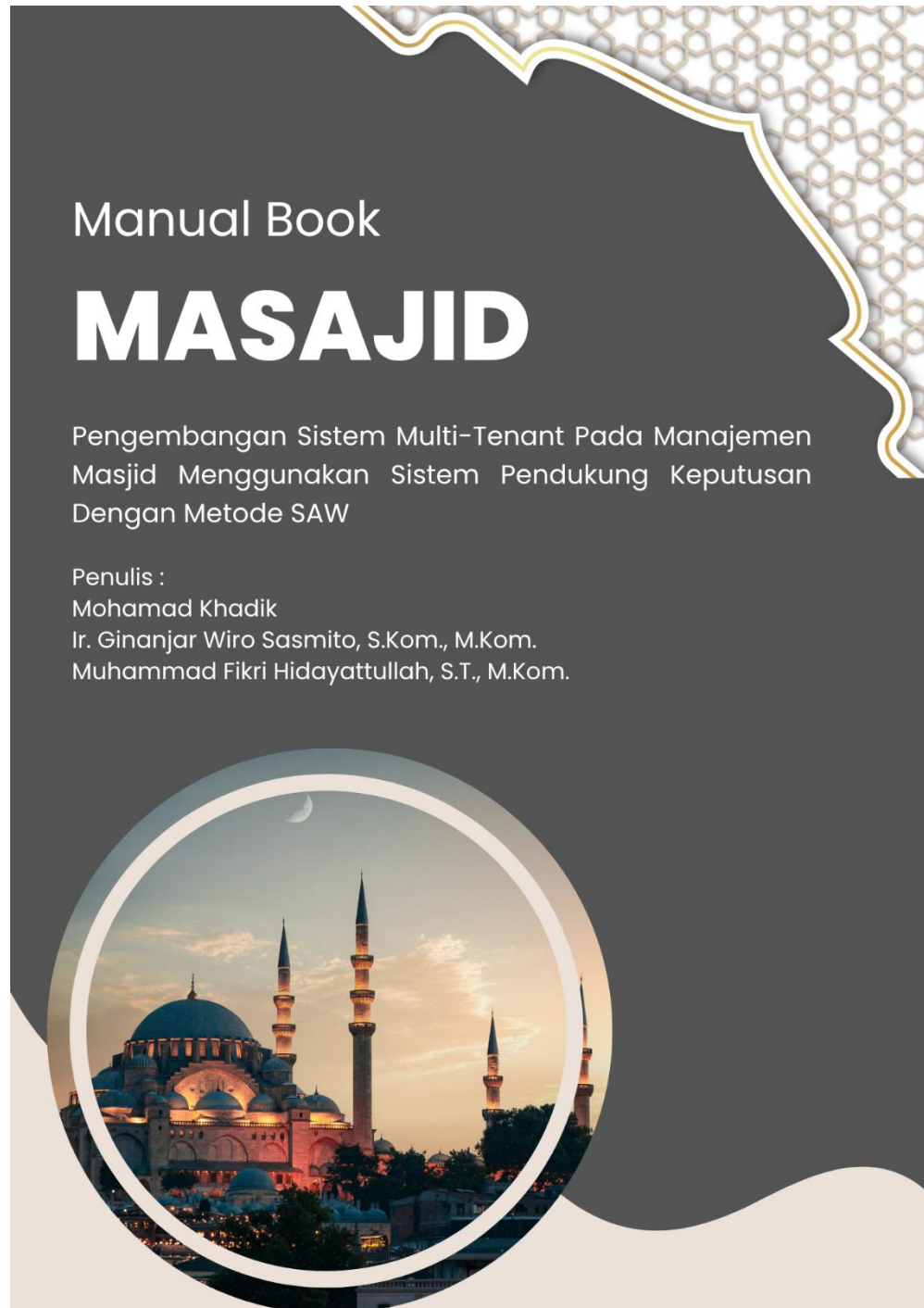
Pencipta


(Mohamad Khadik)


(Ir. Ginanjar Wiro Sasmito, S.Kom., M.Kom.)


(Muhammad Fikri Hidayattullah, S. T., M. Kom.)

Lampiran 4. Manual Book dan Dokumen Teknikal



LATAR BELAKANG

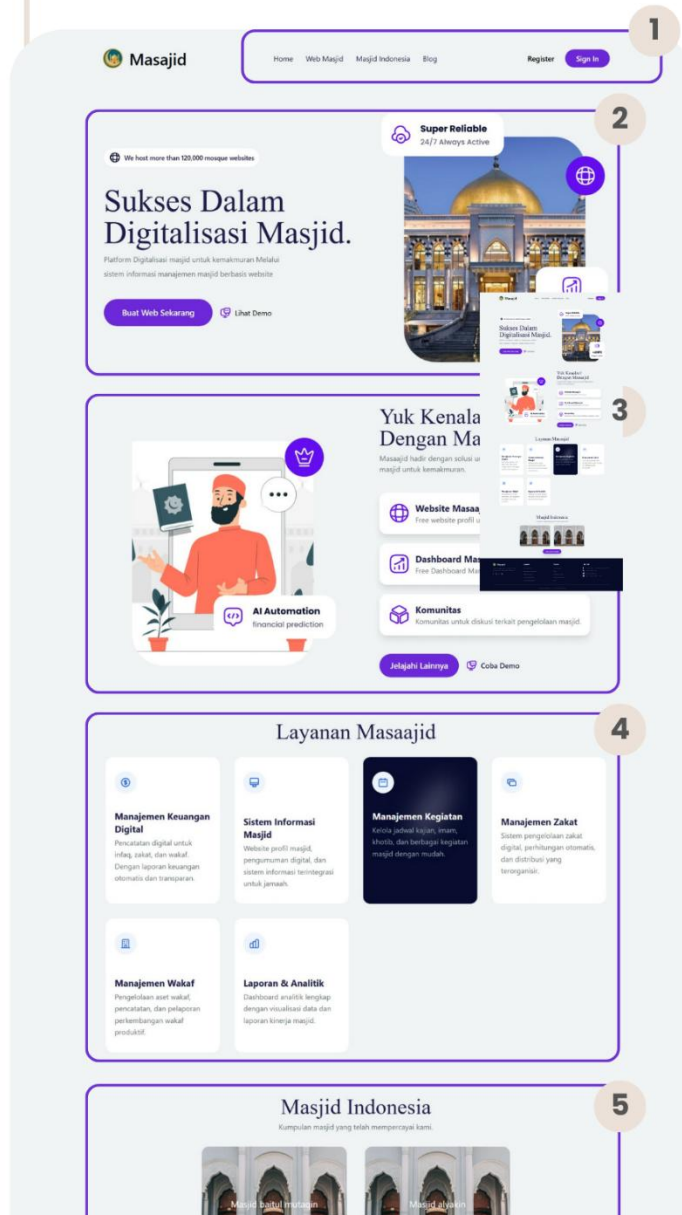
Masjid adalah tempat suci bagi umat Islam yang digunakan untuk melaksanakan ibadah. Selain itu, masjid juga berperan sebagai pusat kegiatan umat muslim, seperti perayaan hari besar Islam, kajian agama, pemberdayaan zakat, infak, dan pendidikan agama. Seiring dengan perkembangan teknologi informasi yang semakin pesat, kebutuhan akan sistem informasi yang efisien dan mudah diakses menjadi sangat penting, termasuk dalam pengelolaan masjid. Sistem manajemen masjid yang terstruktur dan transparan sangat dibutuhkan untuk memudahkan masyarakat dalam mengakses informasi serta mendukung berbagai kegiatan masjid lainnya.

Untuk itu, kami menghadirkan Sistem Informasi Manajemen Masjid berbasis web dengan konsep multi-tenant, yang artinya sistem ini dapat digunakan oleh banyak masjid secara bersamaan dengan pengelolaan data yang aman dan terpisah. Sistem ini dirancang agar mudah digunakan, baik oleh pengurus maupun pengelola program masjid. Fitur unggulan dalam sistem ini adalah Sistem Pendukung Keputusan (SPK) beasiswa dengan metode SAW, yang memudahkan proses seleksi penerima beasiswa secara objektif dan transparan.

Kami berharap sistem ini dapat menjadi solusi praktis dan bermanfaat dalam membantu pengelolaan masjid secara modern, terstruktur, dan lebih profesional. Mari bersama-sama menjadikan masjid lebih maju dan berdampak bagi seluruh umat.

TAMPILAN AWAL

Saat pengguna mengakses sistem Masjid, mereka akan langsung diarahkan ke tampilan awal berupa beranda website yang menyajikan informasi umum dan fitur utama dari sistem.



Langkah 1

Menu navigasi terdiri dari empat pilihan utama, yaitu Home, Web Masjid, Masjid Indonesia, dan Blog. Selain itu, terdapat juga tombol untuk mendaftar (register) dan masuk (sign-in) ke akun pengguna.

Langkah 2

Menampilkan headline utama, deskripsi singkat platform Masjid, dan tombol "Buat Web Sekarang" dan "Lihat Demo".

Langkah 3

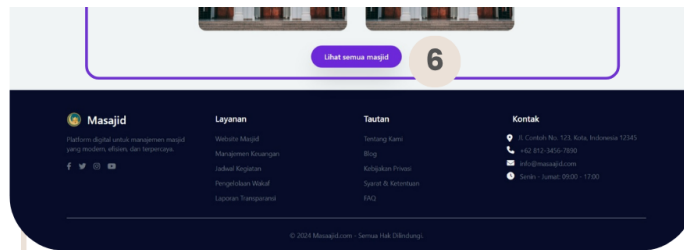
Penjelasan singkat fitur utama yang ditawarkan seperti website masjid, dashboard masjid, komunitas.

Langkah 4

Mengenalkan berbagai layanan digital yang tersedia untuk pengelolaan masjid

Langkah 5

Menampilkan beberapa daftar masjid yang ada di Indonesia.

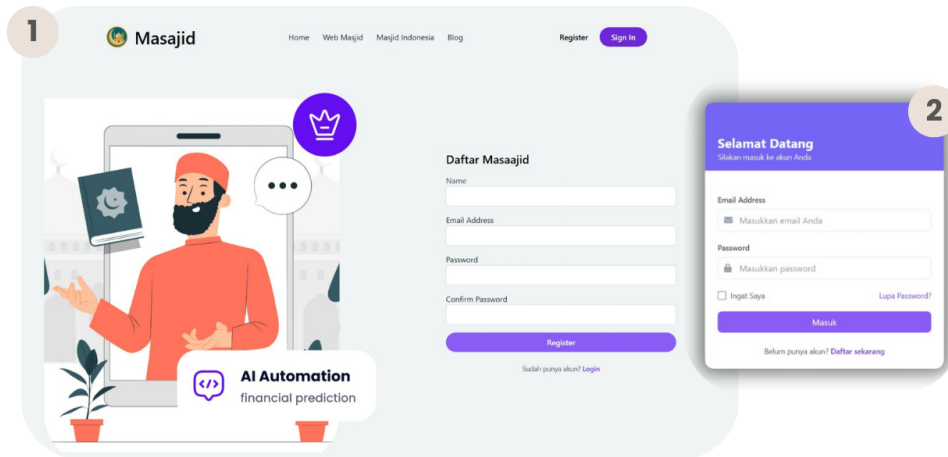


Langkah 6

Tombol lihat semua masjid digunakan jika pengguna ingin melihat seluruh masjid yang ada di Indonesia.

A. PANDUAN PENGGUNA

Registrasi dan Login Pengguna



Langkah 1

Pengguna dapat melakukan registrasi dengan mengisi informasi

- Nama
- Email
- Password
- Konfirmasi password

Setelah proses registrasi selesai, masjid akan langsung terdaftar dalam sistem tanpa perlu melalui proses verifikasi tambahan.

Langkah 2

Untuk mengakses sistem Masajid, pengguna dapat melakukan login dengan memasukkan

- Email
- Password yang telah terdaftar.

Selain itu, Jika pengguna lupa password dapat melakukan pemulihan akun.

03

Daftar Manajemen Masjid

Setelah berhasil melakukan registrasi dan login, pengguna dapat mendaftarkan masjid dengan mengklik menu profil akun di bagian atas. Akan muncul beberapa pilihan dalam dropdown, salah satunya adalah "Daftar Masjid".



Pengguna dapat mendaftarkan masjid dengan memilih menu "Daftar Masjid" yang tersedia pada bagian profile akun. Melalui fitur ini, pengguna dapat membuat halaman web untuk masjid yang dikelola.

A diagram showing three steps of the 'Register Masjid' process. Step 1 shows a form with fields for 'Alamat Lengkap', 'Provinsi' (with a 'Select Provinsi' dropdown), 'Kabupaten' (with a 'Select Kabupaten' dropdown), 'Kecamatan' (with a 'Select Kecamatan' dropdown), and 'Desa' (with a 'Select Desa' dropdown). It has 'Sebelumnya' and 'Selanjutnya' buttons. Step 2 shows a form with fields for 'Nama Masjid', 'Email', 'Nomor HP', and 'Nama Di Ikon Web', with a 'Selanjutnya' button. Step 3 shows a form with a 'Domain Masjid' field containing 'nama-masjid' and a '.masjid.id' suffix, with 'Sebelumnya' and 'Daftarkan Masjid' buttons.

Langkah 1

Pengguna dapat mengisi informasi masjid, seperti Nama Masjid, Email, Nomor HP dan Nama di ikon web. Jika sudah pengguna dapat mengklik tombol "Selanjutnya".

Langkah 2

Pengguna dapat mengisi alamat masjid seperti Alamat Lengkap, Provinsi, Kabupaten dan Desa. Jika pengguna sudah mengisi keseluruhan alamat dapat mengklik tombol "Selanjutnya".

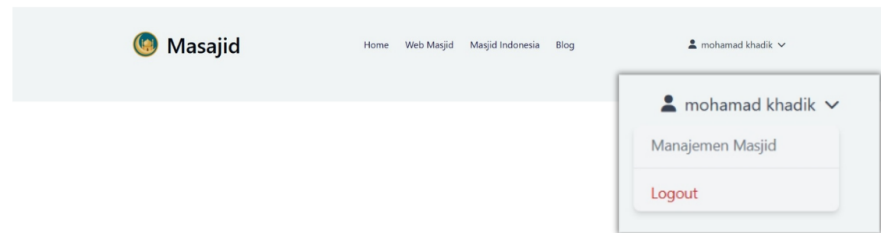
Langkah 3

Pengguna dapat mengisi domain masjid yang akan digunakan kemudian mengklik tombol "Daftarkan Masjid".

04

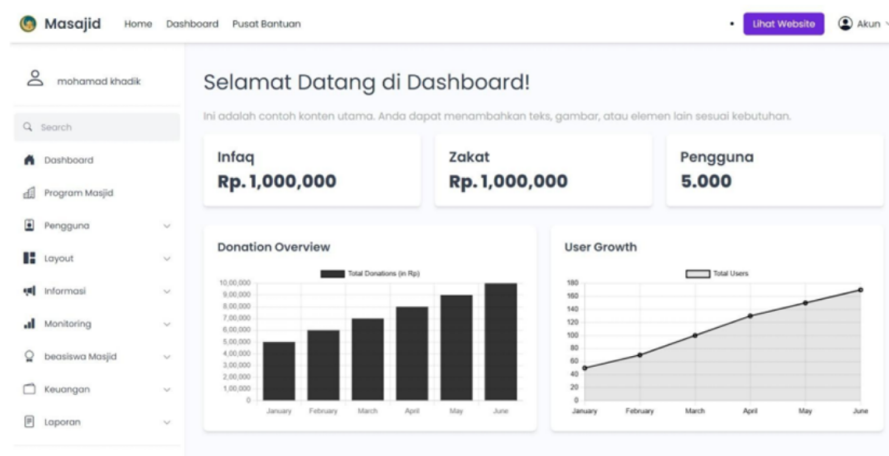
B. PANDUAN MASJID

Dashboard



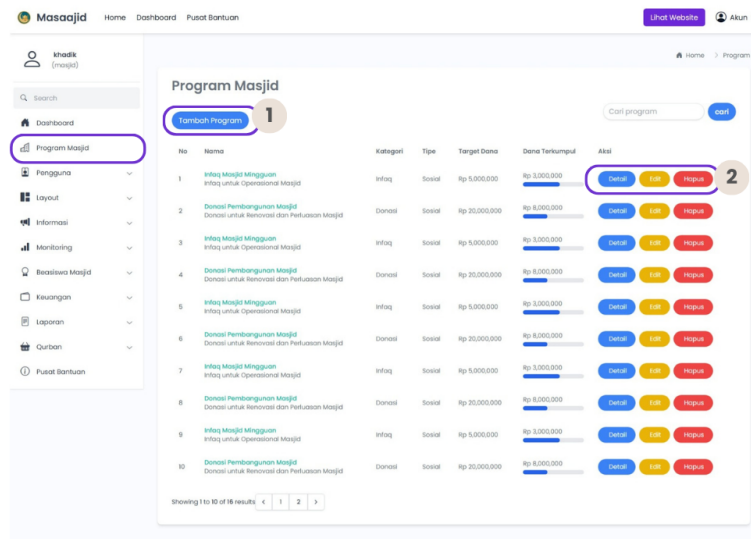
Setelah pengguna berhasil mendaftarkan masjid, menu "Manajemen Masjid" akan muncul pada profil akun pengguna. Untuk mengaksesnya, pengguna cukup mengklik menu tersebut.

Setelah pengguna mendaftarkan masjid dan mengklik menu "Manajemen Masjid", pengguna akan diarahkan ke halaman dashboard. Pada halaman ini, tersedia berbagai fitur yang dapat digunakan untuk mengelola informasi dan aktivitas masjid.



Program Masjid

Fitur ini dapat digunakan oleh pengurus masjid untuk mengelola berbagai program yang dijalankan oleh masjid, seperti infaq mingguan, donasi pembangunan, dan kegiatan sosial lainnya. Melalui fitur ini, pengurus dapat menambahkan program baru, melihat detail program yang telah dibuat, serta melakukan pengelolaan seperti mengedit atau menghapus program.



Langkah 1

Untuk menambahkan program baru, klik tombol “Tambah Program” yang berada di bagian atas halaman. Selanjutnya, isi data program yang dibutuhkan, seperti nama, kategori, tipe, dan target dana. Setelah selesai, simpan data untuk menambahkan program ke daftar.

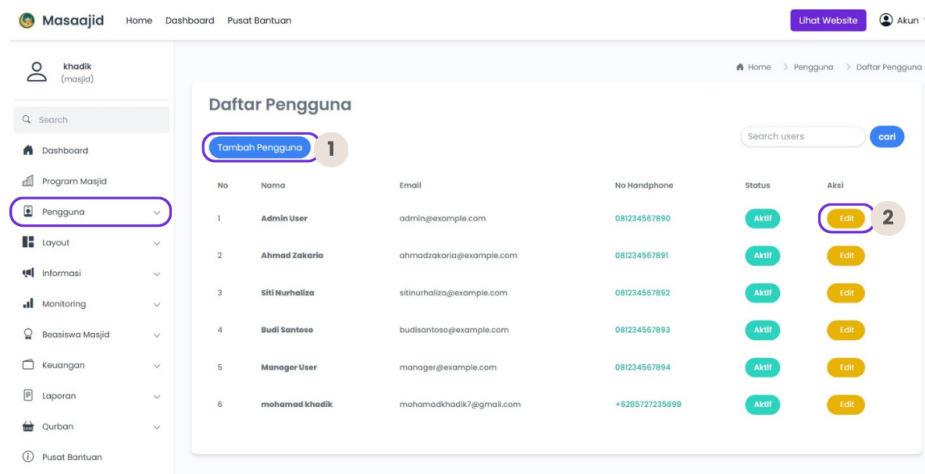
Langkah 2

Setiap program yang telah di tambahkan akan ditampilkan dalam daftar tabel. Pada kolom aksi, terdapat 3 tombol:

- Detail : melihat informasi lengkap dari program tersebut.
- Edit : mengubah data program jika ada informasi yang perlu diperbarui.
- Hapus : menghapus program dari sistem jika tidak lagi dibutuhkan.

Pengguna

Pengurus dapat mengakses fitur Pengguna untuk melihat daftar seluruh pengguna yang terdaftar di website masjid yang telah dibuat. Fitur ini juga dilengkapi dengan fungsi pencarian, sehingga pengurus dapat dengan mudah menemukan pengguna tertentu.



Langkah 1

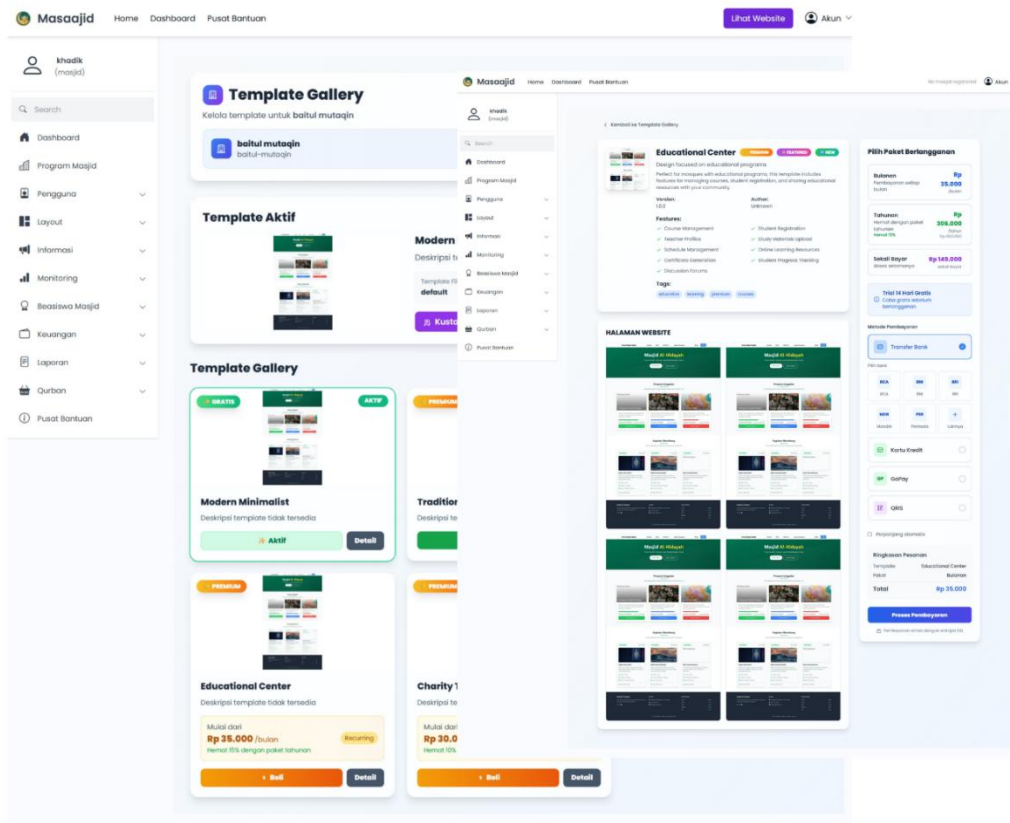
Untuk menambahkan pengguna baru, klik tombol “Tambah Pengguna” yang berada di bagian atas halaman. Selanjutnya, isi data pengguna yang dibutuhkan, seperti nama, email, dan no.handphone. Setelah selesai, simpan data untuk menambahkan pengguna ke daftar.

Langkah 2

Setiap program yang telah di tambahkan akan ditampilkan dalam daftar tabel. Pada kolom aksi, terdapat 3 tombol:

- Status : melihat informasi lengkap dari pengguna tersebut.
- Edit : mengubah data pengguna jika ada informasi yang perlu diperbarui.

Layouts

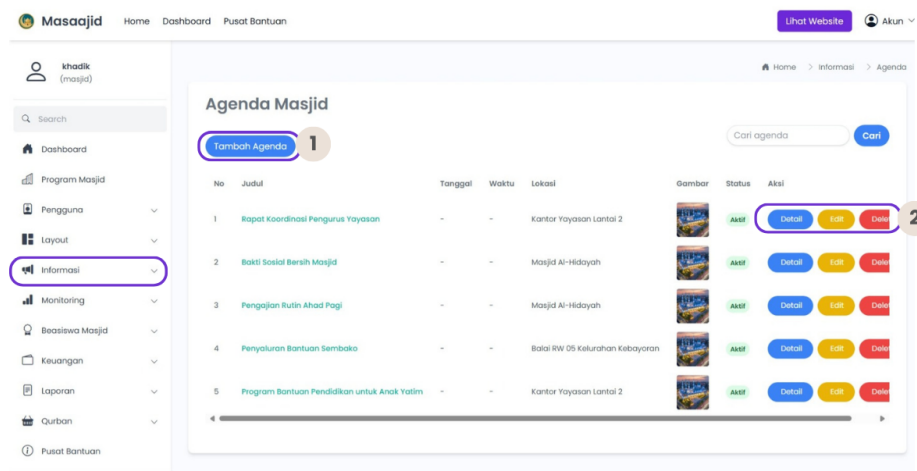


Sistem Masjid menyediakan dua template gratis yang dapat digunakan oleh setiap masjid yang terdaftar. Template ini dirancang untuk mempermudah pengelolaan informasi dan tampilan website masjid, sehingga dapat digunakan tanpa biaya tambahan. Masjid dapat memilih salah satu dari dua template yang tersedia sesuai dengan kebutuhan mereka.

Informasi

Pengurus dapat mengakses fitur informasi, dimana fitur ini memiliki beberapa sub menu di dalamnya untuk melakukan pengelolaan masjid antara lain sebagai berikut.

a. Agenda Masjid



Langkah 1

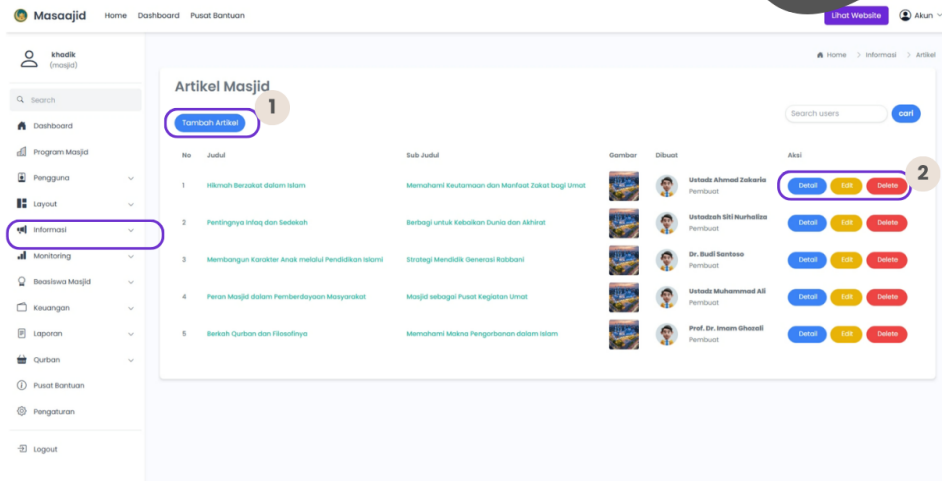
Untuk menambahkan agenda baru, klik tombol “Tambah Agenda” yang berada di bagian atas halaman. Selanjutnya, isi data agenda yang dibutuhkan, seperti judul, tanggal, waktu dan lokasi. Setelah selesai, simpan data untuk menambahkan pengguna ke daftar.

Langkah 2

Setiap agenda yang telah di tambahkan akan ditampilkan dalam daftar tabel. Pada kolom aksi, terdapat 3 tombol:

- Detail : melihat informasi lengkap dari agenda tersebut.
- Edit : mengubah data agenda jika ada informasi yang perlu diperbarui.
- Hapus : menghapus agenda dari sistem jika tidak lagi dibutuhkan.

b. Artikel Masjid



Langkah 1

Untuk menambahkan artikel baru, klik tombol “Tambah Artikel” yang berada di bagian atas halaman. Selanjutnya, isi data agenda yang dibutuhkan, seperti judul, sub judul, gambar dan dibuat. Setelah selesai, simpan data untuk menambahkan pengguna ke daftar.

Langkah 2

Setiap program yang telah di tambahkan akan ditampilkan dalam daftar tabel. Pada kolom aksi, terdapat 3 tombol:

- Detail : melihat informasi lengkap dari program tersebut.
- Edit : mengubah data artikel jika ada informasi yang perlu diperbarui.
- Hapus : menghapus artikel dari sistem jika tidak lagi dibutuhkan.

c. Berita Masjid

The screenshot shows the 'Berita Masjid' page in the MASAjjid application. The sidebar on the left includes a search bar and a menu with 'Informasi' highlighted. The main area displays a table of news items with columns for No, Judul, Gambar, Penulis, Status, and Aksi. A 'Tambah Berita' button is at the top left of the table, and action buttons (Detail, Edit, Delete) are at the bottom of each row.

No	Judul	Gambar	Penulis	Status	Aksi
1	Program Bantuan Pendidikan untuk Anak Yatim		Ahmad Zakaria Penulis	Aktif	Detail Edit Delete
2	Bakti Sosial Ramadhan 2024		Siti Nurhaliza Penulis	Aktif	Detail Edit Delete
3	Pembangunan Masjid Al-Hidayah Tahap 2		Budi Santoso Penulis	Aktif	Detail Edit Delete
4	Santunan Anak Yatim Platu		Manager User Penulis	Aktif	Detail Edit Delete
5	Pelatihan Keterampilan untuk Ibu-Ibu		Ahmad Zakaria Penulis	Aktif	Detail Edit Delete
6	Program Bantuan Pendidikan untuk Anak Yatim		Ahmad Zakaria Penulis	Aktif	Detail Edit Delete
7	Program Bantuan		Ahmad Zakaria Penulis	Aktif	Detail Edit Delete

Langkah 1

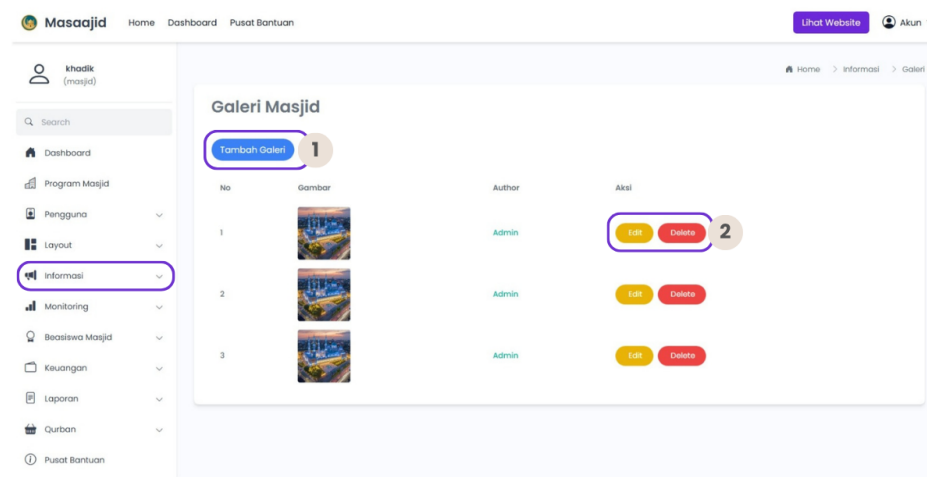
Untuk menambahkan berita baru, klik tombol “Tambah Berita” yang berada di bagian atas halaman. Selanjutnya, isi data berita yang dibutuhkan, seperti judul, gambar, dan penulis. Setelah selesai, simpan data untuk menambahkan pengguna ke daftar.

Langkah 2

Setiap berita yang telah di tambahkan akan ditampilkan dalam daftar tabel. Pada kolom aksi, terdapat 3 tombol:

- Detail : melihat informasi lengkap dari berita tersebut.
- Edit : mengubah data berita jika ada informasi yang perlu diperbarui.
- Hapus : menghapus berita dari sistem jika tidak lagi dibutuhkan.

d. Galeri Masjid



Langkah 1

Untuk menambahkan galeri baru, klik tombol “Tambah Galeri” yang berada di bagian atas halaman. Selanjutnya, isi data galeri yang dibutuhkan, seperti gambar, dan penulis. Setelah selesai, simpan data untuk menambahkan pengguna ke daftar.

Langkah 2

Setiap galeri yang telah di tambahkan akan ditampilkan dalam daftar tabel. Pada kolom aksi, terdapat 2 tombol:

- Edit : mengubah data berita jika ada informasi yang perlu diperbarui.
- Hapus : menghapus berita dari sistem jika tidak lagi dibutuhkan.

Beasiswa Masjid

a. Program Beasiswa

Program Beasiswa

Cari program berdasarkan judul, subjudul, atau deskripsi...

NO	TITLE	SUB TITLE	DESCRIPTION	TOTAL BUDGET	KAPASITAS PESERTA	PESERTA TERDAFTAR	TOTAL PEMENANG	STATUS	AKSI
1	Program Khusus A	Sub Judul Program A	Deskripsi untuk program khusus A.	Rp 10.000.000	100 Orang	2 Orang	25 Orang	Dibuka	Edit Delete
2	Program Khusus B	Sub Judul Program B	Deskripsi untuk program khusus B.	Rp 20.000.000	200 Orang	150 Orang	50 Orang	Dibuka	Edit Delete
3	Program Beasiswa Umum	Beasiswa untuk Semua Jengang	Program beasiswa yang terbuka untuk semua jenjang...	Rp 50.000.000	500 Orang	300 Orang	100 Orang	Proses	Edit Delete
4	Beasiswa Prestasi Akademik	Penghargaan untuk siswa berprestasi	Program beasiswa yang diberikan kepada siswa dengan...	Rp 50.000.000	20 Orang	1 Orang	0 Orang	Dibuka	Edit Delete
5	Program Khusus A	Sub Judul Program A	Deskripsi untuk program khusus A.	Rp 10.000.000	100 Orang	50 Orang	25 Orang	Dibuka	Edit Delete
6	Program Khusus B	Sub Judul Program B	Deskripsi untuk program khusus B.	Rp 20.000.000	200 Orang	150 Orang	50 Orang	Dibuka	Edit Delete
7	Program Beasiswa Umum	Beasiswa untuk Semua Jengang	Program beasiswa yang terbuka untuk semua jenjang...	Rp 50.000.000	500 Orang	300 Orang	100 Orang	Proses	Edit Delete

Langkah 1

Untuk menambahkan program beasiswa baru, klik tombol “Tambah Program” yang berada di bagian atas halaman. Selanjutnya, isi data program beasiswa yang dibutuhkan, seperti title, sub title, deskripsi. Setelah selesai, simpan data untuk menambahkan pengguna ke daftar.

Langkah 2

Setiap program beasiswa yang ditambahkan akan ditampilkan dalam daftar tabel. Pada kolom aksi, terdapat 2 tombol:

- Edit : mengubah data program beasiswa jika ada informasi yang perlu diperbarui.
- Hapus : menghapus program beasiswa dari sistem jika tidak lagi dibutuhkan.

b. Peserta Beasiswa

Peserta Beasiswa

Program Beasiswa: Filter Cepat Pendaftaran Lolos Seksi Reset

Pencarian: Cari peserta berdasarkan nama atau email... Q. Cari

Statistik Tahap: Pendaftaran 4 Seksi 5 Lolos 11 Peninjauan IPJ 2

PROGRAM	EMAIL	NO. TELEPON	TAHAP	STATUS	STATUS DANA	AKSI
Program Beasiswa Anak Yatim	fatimah.uzha@gmail.com	08298765435	Seksi	Memenuhi Syarat	Belum Ters	Detail Edit Delete
Program Beasiswa Anak Yatim	yusuf.brahm@gmail.com	08298765434	Lolos	Memenuhi Syarat	Berhasil Terserim	Nilai Detail Edit Delete
Beasiswa Mahasiswa Kurang Mampu	bambang.prasetyo@gmail.com	08298765433	Peninjauan IPJ	Memenuhi Syarat	Approve IPJ	Detail Edit Delete
Beasiswa Mahasiswa Kurang Mampu	ratna.sari@gmail.com	08298765432	Seksi	Memenuhi Syarat	Belum Terserim	Detail Edit Delete

Langkah 1

Untuk menambahkan peserta baru, klik tombol “Tambah Peserta” yang berada di bagian atas halaman. Selanjutnya, isi data peserta seperti nama lengkap, program, email, no. telpon, tahap, status dan sesatau dana. Setelah selesai, simpan data untuk menambahkan pengguna ke daftar.

Langkah 2

Jika pengurus ingin melihat program beasiswa sesuai jenis itu dapat melakukan filter sesuai kebutuhan, misalkan pendaftaran, lolos, sleksi dan melakukan reset.

Langkah 3

Pengurus dapat melihat total dari peserta pendaftar, sleksi, lolos dan peninjauan.

Langkah 4

Setiap program beasiswa yang ditampilkan dalam daftar tabel terdapat 4 tombol aksi yaitu :

- Nilai : Melihat informasi nilai jika peserta sudah lolos dalam program.
- Detail : Melihat informasi detail dari peserta yang mengikuti program.
- Edit : mengubah data program beasiswa jika ada informasi yang perlu diperbarui.
- Hapus : menghapus program beasiswa dari sistem jika tidak lagi dibutuhkan.

c. Seleksi Peserta

Peserta Seleksi Beasiswa

Program Beasiswa: -- Pilih Program --

Status Seleksi: Tahap Seleksi Memenuhi Syarat

NO	NAMA LENGKAP	PROGRAM	EMAIL	NO. TELEPON	STATUS	AKSI
1	Ahmad Zakaria	Beasiswa Prestasi Ak...	ahmadzakaria@exampl...	+6285727235899	Memenuhi Syarat	Profile Nilai
2	Ahmad Zakaria	Program Khusus A	ahmadzakaria@exampl...	+6285727235899	Memenuhi Syarat	Profile Nilai

Langkah 1

Jika pengurus ingin melihat daftar program beasiswa yang tersedia, pengurus dapat mengklik tombol “Pilih Program”. Status “Tahap Seleksi” atau “Memenuhi Syarat” akan ditampilkan sesuai dengan ketentuan atau tahapan dari program yang dipilih.

Langkah 2

Setiap program beasiswa yang ditambahkan akan ditampilkan dalam daftar tabel. Pada kolom aksi, terdapat 2 tombol:

- Profile : melihat informasi lengkap dari profile pengguna tersebut.
- Nilai : melihat informasi lengkap dari profile pengguna tersebut.

d. Peserta Lolos Beasiswa

Pengurus dapat melihat informasi mengenai total pemenang dan total program beasiswa yang ada

Peserta Lolos Program Beasiswa

Filter Berdasarkan Program: Beasiswa Prestasi Akademik 2025

Beasiswa Prestasi Akademik 2025
Program beasiswa untuk siswa berprestasi. Program beasiswa yang diberikan kepada siswa dengan prestasi akademik terbaik di tingkat SMA. Beasiswa ini mencakup biaya pendidikan, buku, dan biaya...

Total Peserta Lolos: 5 | Total Kriteria: 10 | Total Bobot: 2%

NO	NAMA PESERTA	EMAIL	PROGRAM	SKOR AGRESI SMP	STATUS	Aksi
1	N/A	N/A	Beasiswa Prestasi Akademik 2025	0.0100	Memenuhi_syarat	Detail
2	Admin User	admin@geowample.com	Beasiswa Prestasi Akademik 2025	0.0088	Memenuhi_syarat	Detail
3	Siti Nurhalisa	situnurhalisa@geowample.com	Beasiswa Prestasi Akademik 2025	0.0087	Memenuhi_syarat	Detail
4	Budi Santoso	budisantoso@geowample.com	Beasiswa Prestasi Akademik 2025	0.0086	Memenuhi_syarat	Detail
5	Ahmad Zakaria	ahmadzakaria@geowample.com	Beasiswa Prestasi Akademik 2025	0.0085	Memenuhi_syarat	Detail

Langkah 1

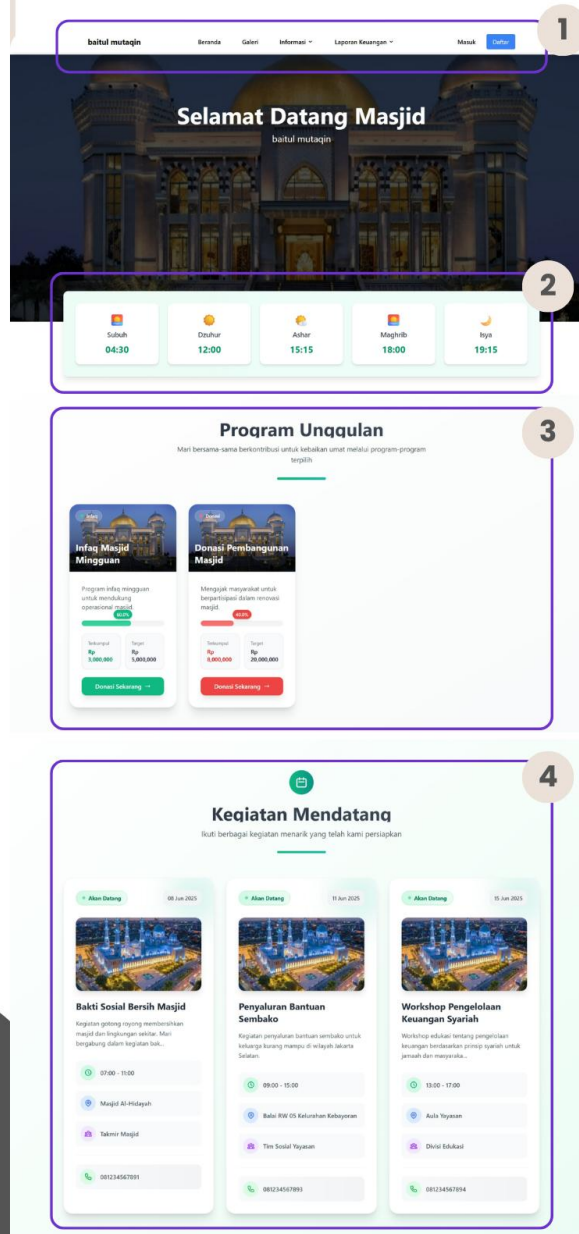
Jika pengurus ingin melihat daftar peserta beasiswa yang telah lolos seleksi dan penerimanya sudah mendapatkan beasiswa, pengurus dapat melihatnya berdasarkan status yang tersedia.

Langkah 2

Seluruh Peserta beasiswa yang ditambahkan akan ditampilkan dalam format tabel. Kolom aksi hanya menyediakan tombol Detail, yang berfungsi untuk menampilkan informasi lengkap terkait Peserta beasiswa yang bersangkutan.

C. PANDUAN JAMAAH MASJID

Beranda



Langkah 1

Menu navigasi terdiri dari empat pilihan utama, yaitu Beranda, Galeri, Informasi, dan Laporan Keuangan. Selain itu, terdapat juga tombol untuk mendaftar dan masuk ke akun pengguna.

Langkah 2

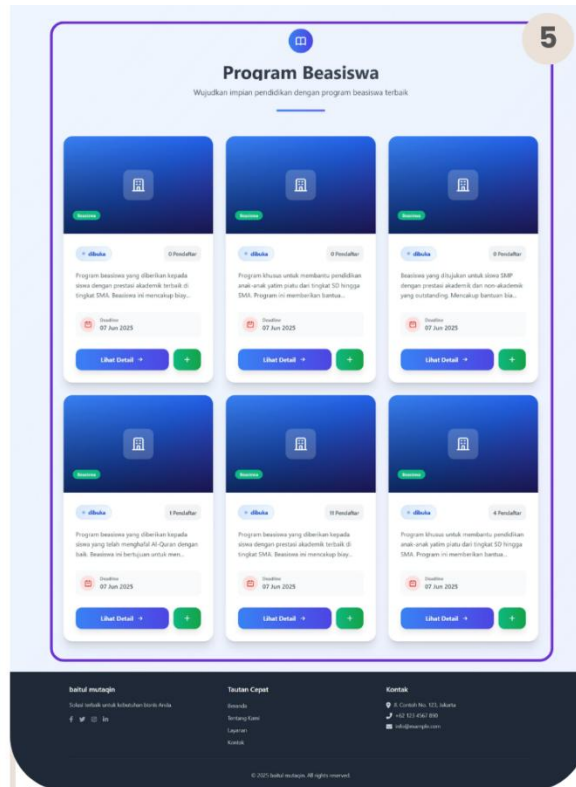
Website menyediakan fitur menu jadwal shalat yang sesuai dengan masjid. Melalui fitur ini, pengguna dapat dengan mudah melihat waktu salat harian yang berlaku di masjid, seperti Subuh, Dzuhur, Ashar, Maghrib, dan Isya.

Langkah 3

Website menghadirkan fitur Program Unggulan yang menampilkan berbagai kegiatan dan inisiatif penting yang diselenggarakan oleh masjid. Beberapa program yang diunggulkan antara lain adalah Infaq Masjid Mingguan dan Donasi Pembangunan Masjid.

Langkah 4

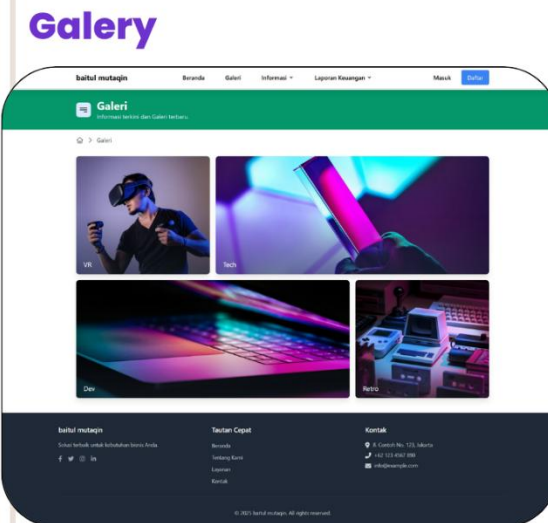
Website menyediakan fitur Kegiatan Mendatang yang berfungsi untuk menampilkan berbagai agenda yang akan dilaksanakan oleh masjid, baik yang bersifat sosial maupun kegiatan rutin keagamaan.



5

Langkah 5

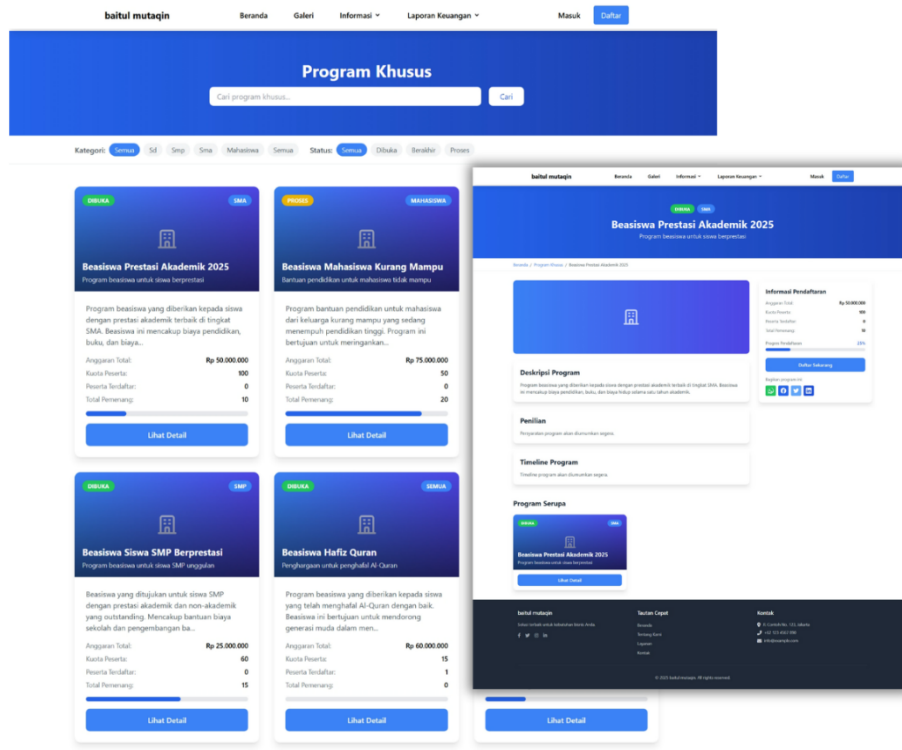
Website ini juga menyediakan fitur Program Beasiswa Masjid yang ditujukan untuk mendukung pendidikan jamaah, khususnya bagi mereka yang memiliki semangat belajar tinggi namun mengalami keterbatasan ekonomi.



Website ini dilengkapi dengan fitur Galeri yang menampilkan berbagai dokumentasi visual terkait kegiatan dan aktivitas yang diselenggarakan oleh masjid. Galeri ini berisi kumpulan gambar yang diambil dari momen-momen penting seperti pengajian, kegiatan sosial, peringatan hari besar Islam, santunan, pembangunan masjid, hingga kegiatan rutin keagamaan lainnya.

20

Beasiswa



Website ini dilengkapi dengan fitur beasiswa yang ditujukan khusus bagi jamaah masjid. Melalui fitur ini, pengguna dapat mengakses informasi lengkap mengenai program beasiswa yang tersedia, termasuk jenis beasiswa, persyaratan, jadwal pendaftaran, serta ketentuan yang berlaku. Selain melihat informasi, pengguna juga dapat langsung melakukan pendaftaran secara online dengan mengisi formulir dan melampirkan dokumen yang dibutuhkan sesuai dengan syarat dan ketentuan yang telah ditetapkan oleh pihak masjid.

Agenda

[baitul mutaqin](#)[Beranda](#)[Galeri](#)[Informasi](#)[Laporan Keuangan](#)[Masuk](#)[Daftar](#)

 **Agenda Kegiatan**
Jadwal dan informasi kegiatan mendatang

Kegiatan Mendatang

Akan Datang08 June 2025



Bakti Sosial Bersih Masjid

⌚ 07:00 - 11:00

📍 Masjid Al-Hidayah

👤 Takmir Masjid

☎ 081234567891

Akan Datang11 June 2025



Penyaluran Bantuan Sembako

⌚ 09:00 - 15:00

📍 Balai RW 05 Kelurahan Kebayoran

👤 Tim Sosial Yayasan

☎ 081234567893

Akan Datang15 June 2025



Workshop Pengelolaan Keuangan Syariah

⌚ 13:00 - 17:00

📍 Aula Yayasan

👤 Divisi Edukasi

☎ 081234567894

[baitul mutaqin](#)
Solusi terbaik untuk kebutuhan bisnis Anda.
f t i n

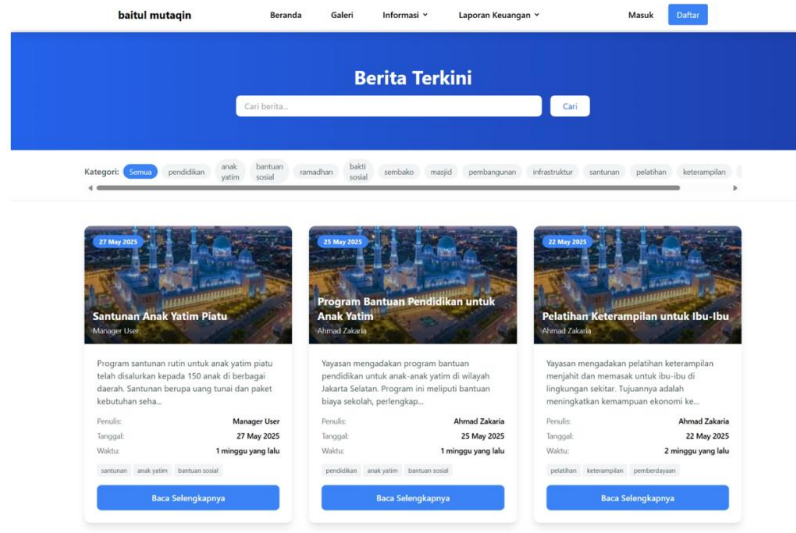
Tautan Cepat
[Beranda](#)
[Tentang Kami](#)
[Layanan](#)
[Kontak](#)

Kontak
Jl. Contoh No. 123, Jakarta
☎ +62 123 4567 890
✉ info@contoh.com

© 2025 baitul mutaqin. All rights reserved.

Website ini menyediakan fitur khusus untuk menampilkan agenda kegiatan masjid secara terstruktur dan mudah diakses. Fitur ini memungkinkan pengguna untuk melihat jadwal kegiatan yang akan dilaksanakan di masjid, termasuk berbagai acara keagamaan, pengajian, kegiatan sosial, dan program rutin lainnya. Dengan adanya agenda kegiatan ini, jamaah dan masyarakat sekitar dapat memperoleh informasi terbaru mengenai waktu dan jenis kegiatan yang akan datang sehingga dapat merencanakan keikutsertaan mereka dengan lebih baik.

Berita



Website ini menyediakan fitur Berita Masjid yang bertujuan untuk memberikan informasi terkini seputar kegiatan, pengumuman, dan perkembangan yang terjadi di lingkungan masjid. Melalui fitur ini, pengguna dapat dengan mudah mengakses berbagai berita terbaru, seperti laporan kegiatan, jadwal acara, pengumuman penting, hingga informasi sosial dan keagamaan lainnya.

Dokumen Teknis

MASAAJID

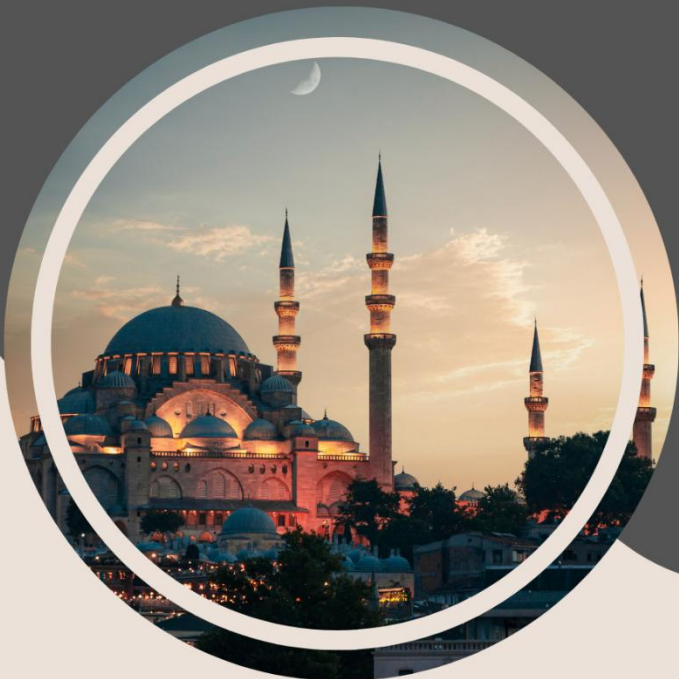
Pengembangan Sistem Informasi Manajemen Masjid
Multi-Tenant Berbasis Web dengan Prediksi
Keuangan Menggunakan Algoritma ARIMA dan
Sistem Pendukung Keputusan Penerima Beasiswa
Menggunakan Metode SAW

Penulis :

Mohamad Khadik

Ir.Ginanjari Wiro S., S.Kom., M.Kom.

Muhammad Fikri H., S.T., M.Kom.



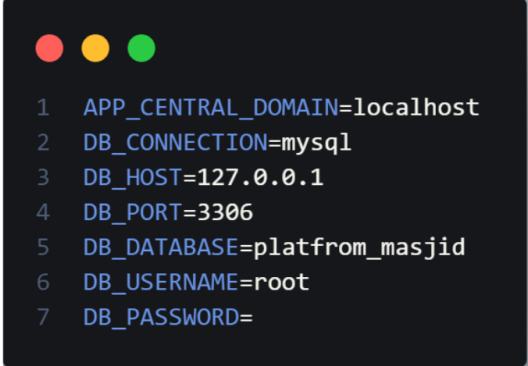
SOURCE CODE

1. LIBRARY

```
1  "require": {
2    "php": "^8.0.2",
3    "barryvdh/laravel-dompdf": "^2.2",
4    "barryvdh/laravel-snappy": "^1.0",
5    "guzzlehttp/guzzle": "^7.2",
6    "laravel/framework": "^9.19",
7    "laravel/sanctum": "^3.0",
8    "laravel/tinker": "^2.7",
9    "laravel/ui": "^4.5",
10   "maatwebsite/excel": "^3.1",
11   "midtrans/midtrans-php": "2.5",
12   "simplesoftwareio/simple-qrcode": "^4.2",
13   "spatie/simple-excel": "^3.2",
14   "stanc1/tenancy": "^3.8"
15 },
16 "require-dev": {
17   "fakerphp/faker": "^1.9.1",
18   "laravel/pint": "^1.0",
19   "laravel/sail": "^1.0.1",
20   "mockery/mockery": "^1.4.4",
21   "nunomaduro/collision": "^6.1",
22   "phpunit/phpunit": "^9.5.10",
23   "spatie/laravel-ignition": "^1.0"
24 },
```

Dependensi proyek Laravel mencakup php untuk runtime, laravel/framework untuk inti aplikasi seperti routing dan ORM Eloquent, laravel/sanctum untuk autentikasi API pada aplikasi SPA atau mobile, laravel/tinker untuk menjalankan perintah PHP di terminal, dan laravel/ui untuk scaffolding antarmuka dengan Bootstrap, Vue, atau React. barryvdh/laravel-dompdf dan barryvdh/laravel-snappy digunakan untuk menghasilkan dokumen PDF dari HTML atau wkhtmltopdf. guzzlehttp/guzzle menangani permintaan HTTP ke API eksternal, maatwebsite/excel dan spatie/simple-excel mengelola ekspor dan impor data Excel/CSV, midtrans/midtrans-php mendukung integrasi pembayaran Midtrans, simplesoftwareio/simple-qrcode menghasilkan kode QR, dan stanc1/tenancy memungkinkan multi-tenancy dengan basis data terpisah. Untuk pengembangan, fakerphp/faker menyediakan data dummy, laravel/pint memformat kode sesuai PSR-12, laravel/sail menawarkan lingkungan Docker, mockery/mockery dan phpunit/phpunit untuk pengujian unit, nunomaduro/collision memberikan output error yang jelas, dan spatie/laravel-ignition mempermudah debugging dengan halaman error interaktif. Dependensi ini mendukung autentikasi, integrasi eksternal, pengelolaan data, dan pengembangan aplikasi yang efisien.

2. KONFIGURASI



```
1 APP_CENTRAL_DOMAIN=localhost
2 DB_CONNECTION=mysql
3 DB_HOST=127.0.0.1
4 DB_PORT=3306
5 DB_DATABASE=platfrom_masjid
6 DB_USERNAME=root
7 DB_PASSWORD=
```

Konfigurasi sistem berbasis website dengan arsitektur multi-tenancy dirancang untuk mengelola pengaturan kunci pada backend. Variabel lingkungan seperti `DB_CONNECTION` menetapkan jenis basis data sebagai MySQL, sementara `DB_HOST` dan `DB_PORT` menentukan alamat dan port server basis data, yaitu 127.0.0.1 dan 3306. Nama basis data diatur melalui `DB_DATABASE` dengan nilai `platfrom_masjid`, dan autentikasi dilakukan menggunakan `DB_USERNAME` dengan nilai `root` serta `DB_PASSWORD` yang dikosongkan. Variabel `APP_CENTRAL_DOMAIN` digunakan untuk menentukan domain utama sistem, yang akan disesuaikan dengan domain hosting saat sistem di-deploy secara online, memastikan fleksibilitas dalam lingkungan produksi.

Untuk mendukung multi-tenancy, sistem dapat menggunakan pendekatan database terpisah dengan membuat database baru seperti `CREATE DATABASE platfrom_masjid_tenant1` untuk setiap tenant, atau skema terpisah dengan perintah seperti `CREATE SCHEMA tenant1` dalam database `platfrom_masjid`. Backend mengarahkan koneksi ke database atau skema yang sesuai berdasarkan identitas tenant, seperti subdomain, dengan logika dinamis seperti pengaturan `database.connections.mysql.database`. Pembuatan tenant baru diotomatisasi melalui dashboard admin, termasuk pembuatan database atau skema serta migrasi tabel, sementara keamanan data antar-tenant dijaga dengan validasi query yang ketat dan skalabilitas ditingkatkan melalui connection pooling serta pemantauan performa, dengan penyesuaian `DB_HOST`, `DB_USERNAME`, dan `DB_PASSWORD` saat di-deploy ke server hosting.

MVC (MODEL, VIEW, CONTROL)

3. MODEL CENTRAL

a. Model User

```
1 <?php
2
3 namespace App\Models;
4
5 use Illuminate\Database\Eloquent\Factories\HasFactory;
6 use Illuminate\Foundation\Auth\User as Authenticatable;
7 use Illuminate\Notifications\Notifiable;
8 use Laravel\Sanctum\HasApiTokens;
9 use Illuminate\Contracts\Auth\MustVerifyEmail;
10
11 class User extends Authenticatable implements MustVerifyEmail
12 {
13     use HasApiTokens, HasFactory, Notifiable;
14
15     protected $fillable = [
16         'name',
17         'email',
18         'password',
19         'role'
20     ];
21
22     protected $hidden = [
23         'password',
24         'remember_token',
25     ];
26
27     protected $casts = [
28         'email_verified_at' => 'datetime',
29     ];
30     public function masjid()
31     {
32         return $this->hasOne(Masjid::class);
33     }
34
35     public function hasMasjid()
36     {
37         return $this->masjid()->exists();
38     }
39 }
```

Model User pada aplikasi Laravel mengelola data pengguna dan autentikasi, mewarisi kelas Authenticatable serta menerapkan antarmuka MustVerifyEmail untuk verifikasi surel. Menggunakan trait HasApiTokens, HasFactory, dan Notifiable, model ini mendukung autentikasi token API, pembuatan data uji, dan notifikasi. Atribut yang dapat diisi meliputi name, email, password, dan role, dengan password dan remember_token disembunyikan dari JSON untuk keamanan. Atribut email_verified_at dikonversi ke format datetime via \$casts. Model ini memiliki relasi satu-ke-satu dengan model Masjid melalui metode masjid() dan hasMasjid() untuk memeriksa data masjid terkait. Operasi basis data dilakukan via Eloquent ORM dengan penanganan kesalahan bawaan Laravel untuk menjaga integritas dan keamanan data.

b. Model Masjid

```
1 <?php
2
3 namespace App\Models;
4
5 use App\Models\Tenant\Param;
6 use App\Models\Tenant\TenantHeroImage;
7 use App\Models\Tenant\TenantTemplateUsage;
8 use App\Models\Tenant\MasjidTemplate;
9 use App\Services\MasjidCreationService;
10 use Illuminate\Database\Eloquent\Factories\HasFactory;
11 use Illuminate\Database\Eloquent\Model;
12 use Illuminate\Support\Facades\Log;
13
14 class Masjid extends Model
15 {
16     use HasFactory;
17
18     protected $fillable = [
19         'name_masjid', 'email', 'nomor_hp', 'nama_ikon_web',
20         'alamat', 'provinsi', 'kabupaten', 'kecamatan',
21         'desa', 'rating', 'domain', 'user_id'
22     ];
23
24     protected static function booted()
25     {
26         static::created(function ($masjid) {
27             Log::info('Creating new Masjid: ' . $masjid->name_masjid);
28
29             try {
30                 $creationService = new MasjidCreationService();
31                 $creationService->handleMasjidCreation($masjid);
32             } catch (\Exception $e) {
33                 Log::error('Error during Masjid creation process: ' . $e->getMessage(), [
34                     'masjid_id' => $masjid->id,
35                     'domain' => $masjid->domain,
36                     'central_domain' => env('APP_CENTRAL_DOMAIN'),
37                     'error' => $e
38                 ]);
39                 throw $e;
40             }
41         });
42     }
43
44     public function user()
45     {
46         return $this->belongsTo(User::class);
47     }
48 }
49
```

Model Masjid pada aplikasi Laravel mengelola data masjid dengan tiga belas atribut fillable (name_masjid, email, nomor_hp, alamat lengkap, domain, user_id, dll) dan mengimplementasikan event handling melalui method booted() yang secara otomatis memicu MasjidCreationService untuk mengupdate role user menjadi 'masjid', membuat tenant baru dengan domain terpisah, dan menginisialisasi delapan template masjid (gratis, trial, dan berbayar) melalui TenantSetupService dan TemplateConfigurationService. Arsitektur multi-tenant ini menggunakan Service Layer Pattern untuk memisahkan logika bisnis yang kompleks, menerapkan comprehensive logging dan exception handling untuk debugging, serta mendukung skalabilitas dengan isolasi data per tenant dan relasi belongsTo ke model User untuk manajemen kepemilikan masjid.

c. Model Masjid Template

```
1 <?php
2
3 namespace App\Models;
4
5 use Illuminate\Database\Eloquent\Factories\HasFactory;
6 use Illuminate\Database\Eloquent\Model;
7 use Illuminate\Database\Eloquent\SoftDeletes;
8
9 class MasjidTemplate extends Model
10 {
11     use HasFactory, SoftDeletes;
12
13     protected $fillable = [
14         'name', 'short_description', 'full_description',
15         'preview_image', 'is_active', 'is_featured',
16         'is_new', 'type', 'price', 'monthly_price',
17         'yearly_price', 'yearly_discount_percentage',
18         'trial_days', 'features', 'screenshots',
19         'usage_limits', 'requires_subscription',
20         'subscription_type', 'version', 'author',
21         'author_url', 'compatibility', 'tags',
22     ];
23
24     protected $casts = [
25         'features' => 'array', 'screenshots' => 'array',
26         'usage_limits' => 'array', 'compatibility' => 'array',
27         'tags' => 'array', 'is_active' => 'boolean',
28         'is_featured' => 'boolean', 'is_new' => 'boolean',
29         'requires_subscription' => 'boolean',
30         'price' => 'decimal:2', 'monthly_price' => 'decimal:2',
31         'yearly_price' => 'decimal:2', 'yearly_discount_percentage' => 'integer',
32         'trial_days' => 'integer',
33     ];
34
35     public function images()
36     {
37         return $this->hasMany(MasjidTemplateImage::class)->ordered();
38     }
39
40     public function primaryImage()
41     {
42         return $this->hasOne(MasjidTemplateImage::class)->primary();
43     }
44 }
```

Model Masjid Template pada website Laravel mengelola data template masjid, mewarisi kelas Model, menggunakan trait HasFactory untuk pembuatan data uji dan SoftDeletes untuk penghapusan lembut, mendefinisikan atribut yang dapat diisi seperti name, type, price, dan features, dengan casting ke tipe data seperti array untuk features dan boolean untuk is_active, memiliki relasi satu-ke-banyak dengan MasjidTemplateImage via images dan satu-ke-satu untuk gambar utama via primaryImage, menyediakan scope seperti active, free, dan premium untuk query efisien, accessor seperti discounted_yearly_price dan preview_image_url untuk manipulasi harga dan URL gambar, metode seperti hasFeature, hasTrial, dan isCompatibleWith untuk validasi fitur dan kompatibilitas, serta metode statis seperti getRecommended dan getLatest untuk mengambil template unggulan dan terbaru, memanfaatkan Eloquent ORM untuk operasi basis data, mendukung relasi dengan model Masjid dan User, serta memastikan integritas dan keamanan data dengan penanganan kesalahan bawaan Laravel.

d. Model Template Image

```
1 <?php
2
3 namespace App\Models;
4
5 use Illuminate\Database\Eloquent\Factories\HasFactory;
6 use Illuminate\Database\Eloquent\Model;
7
8 class MasjidTemplateImage extends Model
9 {
10     use HasFactory;
11
12     protected $fillable = [
13         'masjid_template_id',
14         'image_path',
15         'caption',
16         'sort_order',
17         'is_primary'
18     ];
19
20     protected $casts = [
21         'is_primary' => 'boolean',
22         'sort_order' => 'integer'
23     ];
24
25     public function template()
26     {
27         return $this->belongsTo(MasjidTemplate::class, 'masjid_template_id');
28     }
29 }
```

Model `MasjidTemplateImage` merupakan entitas yang mengelola galeri gambar untuk setiap template masjid dalam sistem. Model ini menyediakan 2 query scope untuk filtering data (primary dan ordered), accessor methods untuk URL generation otomatis termasuk thumbnail handling dengan fallback mechanism, serta utility methods untuk pengelolaan urutan tampilan gambar. Terintegrasi dengan model `MasjidTemplate` melalui relasi `belongsTo`, model ini mendukung fitur gambar utama (primary image) untuk representasi template, sistem pengurutan gambar berdasarkan `sort_order`, dan pengelolaan caption untuk setiap gambar. Dilengkapi dengan automatic casting untuk tipe data boolean dan integer, model ini mengoptimalkan performa dengan thumbnail generation yang intelligent - secara otomatis mendeteksi keberadaan file thumbnail dan memberikan fallback ke gambar original jika thumbnail tidak tersedia, mendukung scalable image management dalam arsitektur multi-tenant sistem manajemen masjid.

e. Model Template Purchase

```
1
2 class TemplatePurchase extends Model
3 {
4     use HasFactory;
5
6     protected $fillable = [
7         'user_id','masjid_template_id','masjid_id','order_id','amount',
8         'currency','purchase_type','transaction_id','payment_type','va_number',
9         'bank','fraud_status','midtrans_response','status','trial_starts_at',
10        'trial_ends_at','subscription_starts_at','subscription_ends_at',
11        'next_billing_date','auto_renewal','billing_cycle','paid_at',
12        'expires_at','payment_method','payment_channel','webhook_logs',
13        'last_webhook_at','notes','failure_reason','metadata',
14    ];
15
16    protected $casts = [
17        'amount' => 'decimal:2','midtrans_response' => 'array',
18        'webhook_logs' => 'array','metadata' => 'array',
19        'auto_renewal' => 'boolean','billing_cycle' => 'integer',
20        'trial_starts_at' => 'datetime','trial_ends_at' => 'datetime',
21        'subscription_starts_at' => 'datetime','subscription_ends_at' => 'datetime',
22        'next_billing_date' => 'datetime','paid_at' => 'datetime',
23        'expires_at' => 'datetime','last_webhook_at' => 'datetime',
24    ];
25
26    // Relationships
27    public function user(): BelongsTo
28    {
29        return $this->belongsTo(User::class);
30    }
31
32    public function masjidTemplate(): BelongsTo
33    {
34        return $this->belongsTo(MasjidTemplate::class);
35    }
36
37    public function masjid(): BelongsTo
38    {
39        return $this->belongsTo(Masjid::class);
40    }
41 }
```

Model TemplatePurchase mengelola transaksi pembelian dan berlangganan template masjid dengan 11 query scope untuk filtering data (pending, paid, failed, expired, active, subscription, oneTime, inTrial, trialExpired, autoRenewal, dueForRenewal), accessor methods untuk kalkulasi status pembayaran dan currency formatting, serta utility methods untuk subscription lifecycle management. Terintegrasi dengan User, MasjidTemplate, dan Masjid melalui relasi belongsTo, model ini mendukung Midtrans payment gateway dengan webhook handling, auto-renewal mechanism, trial period support, dan comprehensive audit logging. Dilengkapi static methods untuk order ID generation dan transaction lookup yang optimized untuk high-volume processing dalam arsitektur event-driven microservices.

f. Model Tenant

```
1 <?php
2
3 namespace App\Models;
4
5 use Stanc1\Tenancy\Database\Models\Tenant as BaseTenant;
6 use Stanc1\Tenancy\Contracts\TenantWithDatabase;
7 use Stanc1\Tenancy\Database\Concerns\HasDatabase;
8 use Stanc1\Tenancy\Database\Concerns\HasDomains;
9
10 class Tenant extends BaseTenant implements TenantWithDatabase
11 {
12     use HasDatabase, HasDomains;
13
14     public function params()
15     {
16         return $this->hasMany(Tenant::Param::class, 'tenant_id', 'id');
17     }
18 }
19
```

Model Tenant merupakan entitas inti yang mengelola arsitektur multi-tenancy dalam sistem dengan implementasi BaseTenant dari Stanc1/Tenancy, menggunakan TenantWithDatabase interface untuk database isolation per tenant, trait HasDatabase untuk automatic database management, dan HasDomains untuk multi-domain support. Model ini terintegrasi dengan relasi hasMany ke Tenant\Param untuk konfigurasi parameter tenant-specific, mendukung automatic tenant switching, database creation/migration, dan domain routing yang optimized untuk scalable SaaS architecture dengan complete tenant isolation.

f. Model Domain

```
1 class Domain extends Model
2 {
3     use HasFactory;
4
5     protected $fillable = [
6         'domain',
7         'tenant_id',
8     ];
9
10    protected $casts = [
11        'tenant_id' => 'string',
12    ];
13
14    // Relationships
15    public function tenant(): BelongsTo
16    {
17        return $this->belongsTo(Tenant::class, 'tenant_id', 'id');
18    }
19
```

Model Domain merupakan entitas yang mengelola routing dan mapping domain dalam arsitektur multi-tenant, menyediakan 3 query scope untuk filtering data (byDomain, byTenant, active), accessor methods untuk URL generation, subdomain detection, dan domain validation, serta utility methods untuk cache management dan domain availability checking. Terintegrasi dengan model Tenant melalui relasi belongsTo dengan cascade delete/update, model ini mendukung automatic cache invalidation, domain format validation dengan regex pattern, dan static methods untuk optimized domain lookup yang di-cache untuk performa tinggi. Dilengkapi dengan event listeners untuk automatic cache clearing dan domain accessibility validation yang optimized untuk high-traffic multi-tenant SaaS architecture.

3. MODEL TENANT

a. Model Tenant User

```
3 namespace App\Models\Tenant;
4
5 use App\Models\Masjid;
6 use Illuminate\Foundation\Auth\User as Authenticatable;
7
8 class TenantUser extends Authenticatable
9 {
10     protected $fillable = [
11         'name',
12         'email',
13         'password',
14         'phone',
15         'status',
16         'role',
17         'tenant_id'
18     ];
19
20     protected $hidden = [
21         'password',
22         'remember_token',
23     ];
24
25     public function masjid()
26     {
27         return $this->belongsTo(Masjid::class, 'tenant_id');
28     }
29 }
```

Model TenantUser adalah entitas yang mengelola data pengguna (user) dalam lingkup tenant tertentu pada arsitektur multi-tenant. Model ini mewarisi fungsi autentikasi dari kelas Authenticatable Laravel, sehingga dapat digunakan untuk autentikasi pengguna seperti login dan manajemen sesi. Model ini terhubung dengan tenant (dalam konteks ini, model Masjid) melalui relasi belongsTo, memastikan setiap pengguna terkait dengan satu tenant. Berikut adalah penjelasan rinci tentang model ini.

e. Model Tenant Program Spesial

```
1 <?php
2
3 namespace App\Models\Tenant;
4
5 use Illuminate\Database\Eloquent\Factories\HasFactory;
6 use Illuminate\Database\Eloquent\Model;
7 use Illuminate\Database\Eloquent\SoftDeletes;
8
9 class TenantProgramSpecial extends Model
10 {
11     use HasFactory, SoftDeletes;
12     protected $fillable = [
13         'title',
14         'sub_title',
15         'description',
16         'total_budget',
17         'participant_capacity',
18         'registered_participant',
19         'total_winners',
20         'category',
21         'status',
22         'visible_fields',
23     ];
24
25     protected $casts = [
26         'visible_fields' => 'array',
27     ];
28
29     public function criteria()
30     {
31         return $this->hasMany(SpkSawCriteria::class, 'tenant_program_special_id');
32     }
33
34     public function alternatives()
35     {
36         return $this->hasMany(SpkSawAlternative::class, 'tenant_program_special_id');
37     }
38
39     public function customFields()
40     {
41         return $this->hasMany(TenantProgramField::class, 'tenant_program_special_id')
42             ->orderBy('field_order');
43     }
44 }
```

Model TenantProgramSpecial adalah entitas yang mengelola program khusus, seperti program beasiswa, dalam arsitektur multi-tenant. Model ini dirancang untuk mendukung pengelolaan program dengan karakteristik khusus, seperti beasiswa, yang melibatkan anggaran, kapasitas peserta, pemenang, kriteria penilaian, alternatif, dan kolom tambahan yang fleksibel (custom fields). Model ini menggunakan SoftDeletes untuk mendukung penghapusan lunak, memungkinkan data yang dihapus tetap tersimpan di database untuk keperluan audit.

e. Model Tenant Field

```
1 <?php
2
3 namespace App\Models\Tenant;
4
5 use Illuminate\Database\Eloquent\Factories\HasFactory;
6 use Illuminate\Database\Eloquent\Model;
7 use Illuminate\Database\Eloquent\SoftDeletes;
8
9 class TenantProgramField extends Model
10 {
11     use HasFactory, SoftDeletes;
12
13     protected $fillable = [
14         'tenant_program_special_id',
15         'field_name',
16         'field_label',
17         'field_type',
18         'field_options',
19         'is_required',
20     ];
21 }
```

```

20     'field_description',
21     'field_order',
22     'is_visible',
23 ];
24
25 protected $casts = [
26     'field_options' => 'array',
27     'is_required' => 'boolean',
28     'is_visible' => 'boolean',
29 ];
30
31 public function program()
32 {
33     return $this->belongsTo(TenantProgramSpecial::class, 'tenant_program_special_id');
34 }
35
36 public function values()
37 {
38     return $this->hasMany(TenantProgramFieldValue::class);
39 }
40 }
41

```

Model TenantProgramField adalah entitas yang mengelola kolom tambahan (custom fields) untuk program khusus, seperti program beasiswa, dalam arsitektur multi-tenant. Model ini memungkinkan pembuatan formulir pendaftaran yang fleksibel untuk program tertentu (misalnya, formulir beasiswa dengan kolom seperti "Nama Lengkap", "IPK", atau "Surat Rekomendasi"). Model ini menggunakan SoftDeletes untuk mendukung penghapusan lunak dan memiliki fitur untuk mengelola tipe kolom, opsi, dan urutan tampilan.

e. Model Tenant Field Value

```

1  <?php
2
3  namespace App\Models\Tenant;
4
5  use Illuminate\Database\Eloquent\Factories\HasFactory;
6  use Illuminate\Database\Eloquent\Model;
7
8  class TenantProgramFieldValue extends Model
9  {
10     use HasFactory;
11
12     protected $fillable = [
13         'tenant_program_field_id',
14         'alternative_id',
15         'field_value',
16     ];
17
18     public function field()
19     {
20         return $this->belongsTo(TenantProgramField::class, 'tenant_program_field_id');
21     }
22
23     public function programField()
24     {
25         return $this->belongsTo(TenantProgramField::class, 'tenant_program_field_id');
26     }
27
28     public function alternative()
29     {
30         return $this->belongsTo(SpkSawAlternative::class, 'alternative_id');
31     }
32 }

```

Model TenantProgramFieldValue adalah entitas yang mengelola nilai yang diisi oleh peserta untuk kolom tambahan (custom fields) dalam program khusus, seperti formulir pendaftaran beasiswa, dalam arsitektur multi-tenant. Model ini digunakan untuk menyimpan data yang dimasukkan oleh peserta (misalnya, nilai IPK atau jurusan studi) yang terkait dengan kolom tertentu di Model TenantProgramField dan kandidat tertentu di Model SpkSawAlternative.

e. Model Tenant SPK SAW Criteria

```

1 class SpkSawCriteria extends Model
2 {
3   use HasFactory, SoftDeletes;
4
5   protected $table = 'spk_saw_criteria';
6
7   protected $fillable = [
8     'tenant_program_special_id',
9     'name',
10    'type',
11    'weight'
12  ];
13
14   protected $casts = [
15     'weight' => 'decimal:2',
16     'type' => 'string'
17  ];
18
19   public function program()
20   {
21     return $this->belongsTo(TenantProgramSpecial::class, 'tenant_program_special_id');
22   }
23
24   public function scores()
25   {
26     return $this->hasMany(SpkSawScore::class, 'criteria_id');
27   }
28 }
29

```

Model SpkSawCriteria mengelola kriteria penilaian untuk program khusus, seperti beasiswa, dalam arsitektur multi-tenant. Kriteria ini digunakan dalam metode SAW (Simple Additive Weighting) untuk mengevaluasi kandidat berdasarkan bobot dan tipe kriteria (benefit atau cost).

e. Model Tenant SPK SAW Alternative

```

1 class SpkSawAlternative extends Model
2 {
3   use HasFactory, SoftDeletes;
4
5   protected $fillable = [
6     'tenant_program_special_id',
7     'user_id',
8     'phone_number',
9     'email',
10    'notes',
11    'name_stage',
12    'status',
13    'fund_distribution_status',
14  ];
15
16   public function program()
17   {
18     return $this->belongsTo(TenantProgramSpecial::class, 'tenant_program_special_id');
19   }
20
21   public function user()
22   {
23     return $this->belongsTo(TenantUser::class, 'user_id');
24   }
25
26   public function scores()
27   {
28     return $this->hasMany(SpkSawScore::class, 'alternative_id');
29   }
30 }
31

```

Model SpkSawAlternative mengelola data kandidat atau alternatif dalam program khusus, seperti peserta beasiswa, dalam arsitektur multi-tenant. Model ini menyimpan informasi tentang kandidat, seperti kontak (nomor telepon, email), status proses seleksi, dan status distribusi dana, serta terhubung dengan sistem penilaian SAW melalui relasi dengan skor dan nilai kolom tambahan dari formulir pendaftaran.

e. Model Tenant SPK SAW Score

```
1 class SpkSawScore extends Model
2 {
3     use HasFactory;
4
5     protected $table = 'spk_saw_scores';
6
7     protected $fillable = [
8         'alternative_id',
9         'criteria_id',
10        'score'
11    ];
12
13    protected $casts = [
14        'score' => 'decimal:2'
15    ];
16
17    public function alternative()
18    {
19        return $this->belongsTo(SpkSawAlternative::class, 'alternative_id');
20    }
21
22    public function criteria()
23    {
24        return $this->belongsTo(SpkSawCriteria::class, 'criteria_id');
25    }
26 }
```

Model SpkSawScore mengelola skor penilaian untuk setiap kandidat (alternatif) berdasarkan kriteria tertentu dalam program khusus, seperti beasiswa, dalam arsitektur multi-tenant. Model ini adalah bagian integral dari sistem penilaian berbasis metode SAW (Simple Additive Weighting), yang digunakan untuk mengevaluasi kandidat secara objektif berdasarkan skor mereka untuk setiap kriteria.

3. CONTROLLER CENTRAL

a. Controller login

```
1  <?php
2
3  namespace App\Http\Controllers\Auth;
4
5  use App\Http\Controllers\Controller;
6  use App\Http\Requests\Auth\LoginRequest;
7  use App\Providers\RouteServiceProvider;
8  use Illuminate\Http\RedirectResponse;
9  use Illuminate\Http\Request;
10 use Illuminate\Support\Facades\Auth;
11 use Illuminate\View\View;
12
13 class AuthenticatedSessionController extends Controller
14 {
15
16     public function create(): View
17     {
18         return view('auth.login');
19     }
20
21     public function store(LoginRequest $request): RedirectResponse
22     {
23         $request->authenticate();
24
25         $request->session()->regenerate();
26
27         return redirect()->intended(RouteServiceProvider::HOME);
28     }
29
30     public function destroy(Request $request): RedirectResponse
31     {
32         Auth::guard('web')->logout();
33
34         $request->session()->invalidate();
35
36         $request->session()->regenerateToken();
37
38         return redirect('/');
39     }
40 }
41
```

Controller `AuthenticatedSessionController` pada aplikasi Laravel mengelola autentikasi sesi pengguna, mewarisi kelas `Controller`. Metode `create` menampilkan form login, metode `store` memproses autentikasi via `LoginRequest` dan `Auth::attempt`, meregenerasi sesi untuk keamanan, lalu mengalihkan ke rute tujuan. Metode `destroy` menangani logout dengan menghapus autentikasi, menginvalidasi sesi, dan meregenerasi token CSRF. Menggunakan `guard web` dan `Eloquent ORM`, controller ini memastikan keamanan dan integritas data pengguna, selaras dengan model `User`, dengan penanganan kesalahan bawaan Laravel.

a. Controller Register

```
1 <?php
2
3 namespace App\Http\Controllers\Auth;
4
5 use App\Http\Controllers\Controller;
6 use App\Models\User;
7 use Illuminate\Foundation\Auth\RegistersUsers;
8 use Illuminate\Support\Facades\Hash;
9 use Illuminate\Support\Facades\Validator;
10
11 class RegisterController extends Controller
12 {
13
14     use RegistersUsers;
15     protected $redirectTo = '/';
16
17     public function __construct()
18     {
19         $this->middleware('guest');
20     }
21
22     protected function validator(array $data)
23     {
24         return Validator::make($data, [
25             'name' => ['required', 'string', 'max:255'],
26             'email' => ['required', 'string', 'email', 'max:255', 'unique:users'],
27             'password' => ['required', 'string', 'min:8', 'confirmed'],
28         ]);
29     }
30
31     protected function create(array $data)
32     {
33         return User::create([
34             'name' => $data['name'],
35             'email' => $data['email'],
36             'password' => Hash::make($data['password']),
37         ]);
38     }
39 }
```

Controller RegisterController pada aplikasi Laravel mengelola registrasi pengguna, mewarisi kelas Controller dan menggunakan trait RegistersUsers. Middleware guest membatasi akses untuk pengguna belum terautentikasi. Metode validator memvalidasi input name, email, dan password dengan aturan seperti keunikan email dan panjang kata sandi. Metode create membuat pengguna baru via model User dengan kata sandi terenkripsi menggunakan Hash. Menggunakan Eloquent ORM, controller ini memastikan keamanan dan integritas data, mendukung relasi dengan model Masjid, dengan penanganan kesalahan bawaan Laravel.

a. Controller Majids

```
1 public function register()
2 {
3     if (auth()->user()->hasMasjid()) {
4         return redirect()->route('home')->with('error', 'You have already registered a masjid.');
```

```
5     }
6     return view('pages.central.masjids.register');
```

```
7 }
8
9 public function registerStore(Request $request)
10 {
11     if (auth()->user()->hasMasjid()) {
12         return redirect()->route('home')->with('error', 'You have already registered a masjid.');
```

```
13     }
14
15     $validatedData = $request->validate([
16         'name_masjid' => 'required|max:255',
17         'email' => 'required|email|max:255',
18         'nomor_hp' => 'required|string|max:15',
19         'nama_ikon_web' => 'nullable|max:255', // Changed to nullable
20         'alamat' => 'required|max:255',
21         'provinsi' => 'required|string|max:255',
22         'kabupaten' => 'required|string|max:255',
23         'kecamatan' => 'required|string|max:255',
24         'desa' => 'required|string|max:255',
25         'domain' => 'required|unique:masjids|max:255',
26     ]);
27
28     // Auto-fill nama_ikon_web if empty
29     if (empty($validatedData['nama_ikon_web'])) {
30         $validatedData['nama_ikon_web'] = $validatedData['name_masjid'];
31     }
32
33     $validatedData['user_id'] = auth()->id();
34
35     $masjid = Masjid::create($validatedData);
36
37     return redirect()->route('home')->with('success', 'Masjid registered successfully.');
```

Metode register dan registerStore pada website Laravel, dalam TenantTemplateController, mengelola pendaftaran masjid untuk sistem multi-tenant, mewarisi Controller, dengan metode register menampilkan halaman pendaftaran masjid melalui view pages.central.masjids.register jika pengguna belum memiliki masjid, memeriksa relasi via hasMasjid dari model User, mengalihkan ke rute home dengan pesan error jika masjid sudah terdaftar, sementara metode registerStore memvalidasi input seperti name_masjid, email, domain (unik di tabel masjids), dan lainnya, mengisi nama_ikon_web secara otomatis jika kosong, menambahkan user_id dari Auth::id(), membuat record masjid baru via model Masjid menggunakan Eloquent ORM, mengalihkan ke home dengan pesan sukses setelah pendaftaran, mendukung relasi satu-ke-satu User dan Masjid, serta memastikan keamanan dan integritas data dengan validasi dan penanganan kesalahan bawaan Laravel.

3. CONTROLLER TENANT

```
1 private function switchToTenantDatabase(Masjid $masjid): void
2 {
3     $tenantDb = 'tenant_' . $masjid->domain;
4     config(['database.default' => 'tenant']);
5     config(['database.connections.tenant.database' => $tenantDb]);
6     DB::purge('tenant');
7     DB::reconnect('tenant');
8 }
9
10 private function switchToCentralDatabase(): void
11 {
12     config(['database.default' => 'mysql']);
13     DB::purge('mysql');
14     DB::reconnect('mysql');
15 }
```

Metode `switchToTenantDatabase` dan `switchToCentralDatabase` pada website Laravel mengelola pergantian koneksi database untuk sistem multi-tenant, dengan `switchToTenantDatabase` mengatur koneksi ke basis data tenant menggunakan `tenant_` berdasarkan domain masjid, memperbarui konfigurasi `database.default` ke tenant, membersihkan koneksi lama via `DB::purge`, dan membuka koneksi baru via `DB::reconnect`, serta `switchToCentralDatabase` mengembalikan koneksi ke basis data pusat `mysql` dengan langkah serupa, memanfaatkan facade `DB`, mendukung relasi Masjid dan User, serta menjaga isolasi, keamanan, dan integritas data dengan penanganan kesalahan bawaan Laravel.

a. Controller Template

```
1 public function index()
2 {
3     $masjids = Masjid::where('user_id', auth()->id()->get());
4     $masjid = $masjids->first();
5
6     if (!$masjid) {
7         return $this->handleMasjidNotFound();
8     }
9
10    try {
11        $availableTemplates = MasjidTemplate::active()->get();
12
13        $paidTemplates = DB::table('template_purchases')
14            ->where('masjid_id', $masjid->id)
15            ->where('status', 'paid')
16            ->where(function ($query) {
17                $query->whereNull('subscription_ends_at')
18                    ->orWhere('subscription_ends_at', '>', now());
19            })
20            ->pluck('masjid_template_id')
21            ->toArray();
22
23        $this->switchToTenantDatabase($masjid);
24
25        $currentTemplate = TenantTemplateUsage::where('is_active', true)->first();
26    }
```

```

27     $currentTemplateDetails = null;
28     if ($currentTemplate) {
29         $currentTemplateDetails = $currentTemplate->getTemplateDetails();
30     }
31
32     $this->switchToCentralDatabase();
33
34     return view('pages.central.masjids.layouts.templates.index', get_defined_vars());
35 } catch (\Exception $e) {
36
37     return redirect()->back()
38         ->with('error', 'Error loading templates. Please try again.');
```

Metode index pada website Laravel menampilkan daftar template masjid dalam sistem multi-tenant, mengambil masjid pengguna via `Masjid::where('user_id', auth()->id())`, mengalihkan ke home jika masjid tidak ditemukan via `handleMasjidNotFound`, mengambil template aktif dari `MasjidTemplate`, memeriksa pembayaran premium di `template_purchases`, beralih ke basis data tenant dengan `switchToTenantDatabase` untuk mendapatkan template aktif dari `TenantTemplateUsage`, mengambil detail template, kembali ke basis data pusat via `switchToCentralDatabase`, merender view `pages.central.masjids.layouts.templates.index`, menggunakan Eloquent ORM, menangani error dengan pengalihan, mendukung relasi User dan Masjid, serta menjaga keamanan dan integritas data dengan penanganan kesalahan Laravel.

```

1 public function switchTemplate(Request $request, int $templateId)
2 {
3     $masjid = $this->getUserMasjid();
4     if (!$masjid) {
5         return $this->handleMasjidNotFound();
6     }
7     $template = MasjidTemplate::active()->find($templateId);
8     if (!$template) {
9         return redirect()->back()
10             ->with('error', 'Template not found or not available.');
```

```

30     if ($existingUsage) {
31         $existingUsage->update([
32             'is_active' => true,
33             'access_type' => $accessType,
34             'activated_at' => Carbon::now(),
35             'updated_at' => Carbon::now(),
36             'access_expires_at' => $accessType === 'trial'
37                 ? Carbon::now()->addDays($template->trial_days)
38                 : null
39         ]);
40     } else {
41         TenantTemplateUsage::create([
42             'template_id' => $templateId,
43             'is_active' => true,
44             'access_type' => $accessType,
45             'customizations' => [],
46             'usage_stats' => [],
47             'usage_limits' => [],
48             'activated_at' => Carbon::now(),
49             'access_expires_at' => $accessType === 'trial'
50                 ? Carbon::now()->addDays($template->trial_days)
51                 : null,
52             'created_at' => Carbon::now(),
53             'updated_at' => Carbon::now()
54         ]);
55     }
56     DB::commit();
57     return redirect()->route('masjid.layouts.templates.index')
58         ->with('success', "Template berhasil diganti ke: {$template->name}");
59 } catch (\Exception $e) {
60     DB::rollback();
61     return redirect()->back()
62         ->with('error', 'Error switching template. Please try again.');
```

Controller TenantTemplateController pada website Laravel, melalui metode switchTemplate, mengganti template masjid dalam sistem multi-tenant, mewarisi Controller, mengambil masjid pengguna via getUserMasjid dengan Auth::id(), mengalihkan ke home jika masjid tidak ditemukan via handleMasjidNotFound, memeriksa template aktif di MasjidTemplate berdasarkan ID, memvalidasi akses premium dengan checkTemplateAccess, beralih ke basis data tenant menggunakan switchToTenantDatabase, menonaktifkan template aktif di TenantTemplateUsage, memperbarui atau membuat record baru dengan access_type (free, paid, trial) dan access_expires_at untuk trial, memanfaatkan Eloquent ORM dan transaksi DB untuk integritas, menangani error dengan rollback dan pesan, kembali ke basis data pusat via switchToCentralDatabase, mendukung relasi User dan Masjid, serta menjaga keamanan dan konsistensi dengan penanganan kesalahan Laravel.

```

1 public function preview(string $templateName)
2 {
3     $masjid = $this->getUserMasjid();
4
5     if (!$masjid) {
6         return $this->handleMasjidNotFound();
7     }
8
9     try {
10         // Get template from central database by name
11         $template = MasjidTemplate::active()
12             ->where('name', $templateName)
13             ->firstOrFail();
14
15         return view('pages.central.masjids.layouts.templates.preview', compact(
16             'masjid',
17             'template'
18         ));
19     } catch (\Illuminate\Database\Eloquent\ModelNotFoundException $e) {
20         return redirect()->back()
21             ->with('error', 'Template not found.');
```

Metode preview pada website Laravel menampilkan pratinjau template masjid dalam sistem multi-tenant, mengambil masjid pengguna via getUserMasjid dengan Auth::id(), mengalihkan ke rute home jika masjid tidak ditemukan melalui handleMasjidNotFound, mengambil template aktif dari MasjidTemplate berdasarkan nama menggunakan active()->where('name', \$templateName)->firstOrFail, merender view pages.central.masjids.layouts.templates.preview dengan data masjid dan template, memanfaatkan Eloquent ORM, menangani error ModelNotFoundException dengan pesan "Template not found" dan error umum dengan logging via Log serta pengalihan, mendukung relasi User dan Masjid, serta memastikan keamanan dan integritas data dengan penanganan kesalahan bawaan Laravel.

a. Controller Template Pembelian Template

```
1 public function payment(Request $request, $templateId)
2 {
3     try {
4         $template = MasjidTemplate::active()->findOrFail($templateId);
5         $user = Auth::user();
6         $masjid = Masjid::where('user_id', $user->id)->first();
7
8         if (!$masjid) {
9             return redirect()->route('home')
10                 ->with('error', 'No masjid found for this user.');
```

Metode payment pada website Laravel menampilkan halaman pembayaran template masjid dalam sistem multi-tenant, mengambil template aktif dari MasjidTemplate berdasarkan ID menggunakan `active()->findOrFail`, mengambil pengguna terautentikasi via `Auth::user()` dan masjid terkait via `Masjid::where('user_id', $user->id)`, mengalihkan ke rute home dengan pesan error jika masjid tidak ditemukan, mencari pembelian tertunda di TemplatePurchase berdasarkan `user_id`, `masjid_template_id`, `masjid_id`, dan status pending, mengalihkan ke rute pembelian jika tidak ada pembelian tertunda, memformat detail pembayaran dengan `formatPaymentDetails`, merender view `pages.central.masjids.layouts.purchases.payment` dengan data template, masjid, pembelian, dan detail pembayaran, memanfaatkan Eloquent ORM, menangani error dengan logging via Log dan pengalihan dengan pesan error, mendukung relasi User dan Masjid, serta memastikan keamanan dan integritas data dengan penanganan kesalahan bawaan Laravel.

a. Controller Program Special

```
1 public function create()
2 {
3     $masjids = Masjid::where('user_id', auth()->id()->get());
4     $masjid = $masjids->first();
5
6     if (!$masjid) {
7         return redirect()->route('home')
8             ->with('error', 'No masjid found for this user.');
```

```
9     }
10
11     $this->switchToTenantDatabase($masjid);
12
13     // Fetch up to 4 existing programs with their criteria and custom fields for recommendations
14     try {
15         $recommendedPrograms = TenantProgramSpecial::with(['criteria', 'customFields'])
16             ->orderBy('created_at', 'desc')
17             ->take(4)
18             ->get();
19
20         // Format recommended programs for JavaScript use
21         $formattedRecommendations = [];
22         foreach ($recommendedPrograms as $program) {
23             $formattedCriteria = [];
24             foreach ($program->criteria as $criteria) {
25                 $formattedCriteria[] = [
26                     'name' => $criteria->name,
27                     'type' => $criteria->type,
28                     'weight' => $criteria->weight,
29                 ];
30             }
31
32             $formattedCustomFields = [];
33             foreach ($program->customFields as $field) {
34                 $formattedCustomFields[] = [
35                     'field_name' => $field->field_name,
36                     'field_label' => $field->field_label,
37                     'field_type' => $field->field_type,
38                     'field_options' => $field->field_options,
39                     'is_required' => $field->is_required,
40                     'field_description' => $field->field_description,
41                     'field_order' => $field->field_order,
42                     'is_visible' => $field->is_visible,
43                 ];
44             }
45
46             $formattedRecommendations[$program->title] = [
47                 'title' => $program->title,
48                 'sub_title' => $program->sub_title,
49                 'description' => $program->description,
50                 'category' => $program->category,
51                 'total_budget' => $program->total_budget,
52                 'participant_capacity' => $program->participant_capacity,
53                 'status' => $program->status,
54                 'criteria' => $formattedCriteria,
55                 'custom_fields' => $formattedCustomFields,
56             ];
57         }
58     } catch (\Exception $e) {
59         $formattedRecommendations = [];
60     }
61
62     $categories = ['sd', 'smp', 'sma', 'mahasiswa', 'semua'];
63     $fieldTypes = ['text', 'number', 'select', 'file', 'date', 'textarea'];
64
65     return view('pages.central.masjids.basiswa.program.create', [
66         'masjids' => $masjids,
67         'masjid' => $masjid,
68         'categories' => $categories,
69         'fieldTypes' => $fieldTypes,
70         'recommendedPrograms' => $formattedRecommendations
71     ]);
72 }
```

Metode create pada website Laravel menampilkan halaman pembuatan program beasiswa masjid dalam sistem multi-tenant, mengambil masjid pengguna via Masjid::where('user_id', auth()->id()), mengalihkan ke rute home dengan pesan error jika masjid tidak ditemukan, beralih ke basis data tenant menggunakan switchToTenantDatabase, mengambil hingga 4 program rekomendasi dari TenantProgramSpecial beserta criteria dan customFields, memformat data program untuk JavaScript dengan detail seperti judul, kriteria, dan field kustom, menyediakan daftar kategori (sd, smp, sma, mahasiswa, semua) dan tipe field (text, number, select, file, date, textarea), merender view pages.central.masjids.beasiswa.program.create dengan data masjid, kategori, tipe field, dan rekomendasi program, memanfaatkan Eloquent ORM, menangani error dengan pengosongan rekomendasi, mendukung relasi User dan Masjid, serta memastikan keamanan dan integritas data dengan penanganan kesalahan bawaan Laravel.

```

1 public function store(Request $request)
2 {
3     $masjids = Masjid::where('user_id', auth()->id())->get();
4     $masjid = $masjids->first();
5
6     if (!$masjid) {
7         return redirect()->route('home')
8             ->with('error', 'No masjid found for this user.');
```

```

9     }
10
11     $this->switchToTenantDatabase($masjid);
12
13     try {
14         DB::beginTransaction();
15
16         // Validate and create program
17         $validatedData = $this->validateProgramData($request);
18         $program = TenantProgramSpecial::create($validatedData);
19
20         // Handle criteria if present
21         if ($request->has('criteria')) {
22             foreach ($request->criteria as $criteriaData) {
23                 $program->criteria()->create([
24                     'name' => $criteriaData['name'],
25                     'type' => $criteriaData['type'],
26                     'weight' => $criteriaData['weight']
27                 ]);
28             }
29         }
30
31         // Handle custom fields if present
32         if ($request->has('custom_fields')) {
33             foreach ($request->custom_fields as $fieldData) {
34                 $program->customFields()->create([
35                     'field_name' => $fieldData['field_name'],
36                     'field_label' => $fieldData['field_label'],
37                     'field_type' => $fieldData['field_type'],
38                     'field_options' => isset($fieldData['field_options']) ? json_encode(explode(',', $fieldData['field_options'])) : null,
39                     'is_required' => isset($fieldData['is_required']) ? $fieldData['is_required'] : false,
40                     'field_description' => $fieldData['field_description'] ?? null,
41                     'field_order' => $fieldData['field_order'] ?? 0,
42                     'is_visible' => isset($fieldData['is_visible']) ? $fieldData['is_visible'] : true
43                 ]);
44             }
45         }
46
47         DB::commit();
48         return redirect()->route('masjid.beasiswa.program.index')
49             ->with('success', 'Program berhasil ditambahkan!');
```

```

50     } catch (Exception $e) {
51         DB::rollback();
52         return redirect()->back()
53             ->with('error', 'Error creating program: ' . $e->getMessage())
54             ->withInput();
55     }
56 }
```


Metode store pada website Laravel menyimpan program beasiswa masjid dalam sistem multi-tenant, mengambil masjid pengguna via `Masjid::where('user_id', auth()->id())`, mengalihkan ke rute home dengan pesan error jika masjid tidak ditemukan, beralih ke basis data tenant menggunakan `switchToTenantDatabase`, memulai transaksi database dengan `DB::beginTransaction`, memvalidasi data program via `validateProgramData`, membuat program baru di `TenantProgramSpecial`, menambahkan kriteria dan field kustom jika ada melalui relasi criteria dan `customFields`, menyimpan field options sebagai JSON, mengatur default untuk `is_required`, `field_description`, `field_order`, dan `is_visible`, menyelesaikan transaksi dengan `DB::commit`, mengalihkan ke rute `masjid.beasiswa.program.index` dengan pesan sukses, menangani error dengan `DB::rollBack` dan pengalihan dengan pesan error serta input sebelumnya, memanfaatkan Eloquent ORM, mendukung relasi User dan Masjid, serta memastikan keamanan dan integritas data dengan penanganan kesalahan bawaan Laravel.

a. Controller Peserta Beasiswa

```

1 class PesertaBeasiswaController extends Controller
2 {
3
4     public function index(Request $request)
5     {
6         $masjids = Masjid::where('user_id', auth()->id())->get();
7         $masjid = $masjids->first();
8
9         if (!$masjid) {
10             return redirect()->route('home')
11                 ->with('error', 'No masjid found for this user.');

```

```

37         // Add stage filter
38         if ($stageFilter) {
39             $query->where('name_stage', $stageFilter);
40         }
41
42         $query->orderBy('id', 'desc');
43
44         $peserta = $query->paginate(10);
45         $programs = TenantProgramSpecial::all();
46
47         // Get stage statistics for display
48         $stageStats = SpkSawAlternative::select('name_stage', DB::raw('count(*) as count'))
49             ->when($programId, function($q) use ($programId) {
50                 return $q->where('tenant_program_special_id', $programId);
51             })
52             ->groupBy('name_stage')
53             ->pluck('count', 'name_stage')
54             ->toArray();
55
56         return view('pages.central.masjids.basiswa.peserta.index', get_defined_vars());
57     } catch (\Exception $e) {
58         return redirect()->back()
59             ->with('error', 'Unable to connect to tenant database: ' . $e->getMessage());
60     }
61 }

```

Metode index pada PesertaBasiswaController di website Laravel menampilkan daftar peserta beasiswa masjid dalam sistem multi-tenant, mengambil masjid pengguna via Masjid::where('user_id', auth()->id()->get()), mengalihkan ke rute home dengan pesan error jika masjid tidak ditemukan, beralih ke basis data tenant menggunakan switchToTenantDatabase, membangun query untuk SpkSawAlternative dengan relasi program, user, dan fieldValues.programField, memfilter berdasarkan program_id, pencarian nama, email, atau nomor telepon, serta tahap (name_stage), mengurutkan hasil berdasarkan ID secara menurun, memaginasikan 10 entri per halaman, mengambil semua program dari TenantProgramSpecial, menghitung statistik tahap dengan name_stage dan jumlah peserta, merender view pages.central.masjids.basiswa.peserta.index dengan data masjid, peserta, program, dan statistik, memanfaatkan Eloquent ORM, menangani error dengan pengalihan dan pesan, mendukung relasi User dan Masjid, serta memastikan keamanan dan integritas data dengan penanganan kesalahan bawaan Laravel.

```

1 public function store(Request $request)
2 {
3     $masjids = Masjid::where('user_id', auth()->id()->get())
4         ->first();
5
6     if (!$masjids) {
7         return redirect()->route('home')
8             ->with('error', 'No masjid found for this user.');
```

```

33     if (!isset($validatedData['status']) || empty($validatedData['status'])) {
34         $validatedData['status'] = 'terdaftar';
35     }
36     if (!isset($validatedData['fund_distribution_status']) || empty($validatedData['fund_distribution_status'])) {
37         $validatedData['fund_distribution_status'] = 'belum_terkirim';
38     }
39
40     // Only check capacity and update participant count if program is selected
41     if (!empty($validatedData['tenant_program_special_id'])) {
42         // Check if program capacity has been reached
43         $program = TenantProgramSpecial::findOrFail($validatedData['tenant_program_special_id']);
44         $currentParticipants = $program->alternatives()->count(); // Fixed: changed from programItems() to alternatives()
45
46         if ($currentParticipants >= $program->participant_capacity) {
47             return redirect()->back()
48                 ->with('error', 'Program capacity has been reached.')
49                 ->withInput();
50         }
51
52         // Update program participants count
53         $program->registered_participant = $currentParticipants + 1;
54         $program->save();
55     }
56
57     // Create peserta
58     $peserta = SpkSawAlternative::create($validatedData);
59
60     // Handle program field values (including file uploads) only if program is selected
61     if (!empty($validatedData['tenant_program_special_id'])) {
62         $this->handleProgramFieldValues($request, $peserta, $validatedData['tenant_program_special_id']);
63     }
64
65     DB::commit();
66     return redirect()->route('masjid.beasiswa.peserta.index')
67         ->with('success', 'Peserta beasiswa berhasil ditambahkan!');
68 } catch (\Exception $e) {
69     DB::rollBack();
70     return redirect()->back()
71         ->with('error', 'Error creating peserta: ' . $e->getMessage())
72         ->withInput();
73 }
74 }

```

Metode store pada PesertaBeasiswaController di website Laravel menyimpan data peserta beasiswa masjid dalam sistem multi-tenant, mengambil masjid pengguna via Masjid::where('user_id', auth()->id()), mengalihkan ke rute home dengan pesan error jika masjid tidak ditemukan, beralih ke basis data tenant menggunakan switchToTenantDatabase, memulai transaksi dengan DB::beginTransaction, memvalidasi data peserta via validatePesertaData, mengatur default name_stage ke 'pendaftaran', status ke 'terdaftar', dan fund_distribution_status ke 'belum_terkirim', memeriksa kapasitas program di TenantProgramSpecial jika program dipilih, mengalihkan dengan pesan error jika kapasitas penuh, memperbarui jumlah peserta terdaftar, membuat peserta baru di SpkSawAlternative, menangani nilai field program dengan handleProgramFieldValues jika program dipilih, menyelesaikan transaksi dengan DB::commit, mengalihkan ke rute masjid.beasiswa.peserta.index dengan pesan sukses, menangani error dengan DB::rollBack dan pengalihan dengan pesan error serta input sebelumnya, memanfaatkan Eloquent ORM, mendukung relasi User dan Masjid, serta memastikan keamanan dan integritas data dengan penanganan kesalahan bawaan Laravel.

a. Controller Peserta Seleksi Beasiswa

```
1 public function index(Request $request)
2 {
3     $masjids = Masjid::where('user_id', auth()->id()->get());
4     $masjid = $masjids->first();
5
6     if (!$masjid) {
7         return redirect()->route('home')
8             ->with('error', 'No masjid found for this user.');
```

```
9     }
10
11     $this->switchToTenantDatabase($masjid);
12
13     try {
14         $programId = $request->input('program_id');
```

```
15
16         // Base query to get only participants in selection phase with "memenuhi_syarat" status
17         $query = SpkSawAlternative::with(['program', 'scores.criteria'])
18             ->where('name_stage', 'seleksi')
19             ->where('status', 'memenuhi_syarat'); // Only participants who meet requirements
20
21         if ($programId) {
22             $query->where('tenant_program_special_id', $programId);
23         }
24
25         $selectedPeserta = $query->orderBy('id', 'desc')->paginate(10);
26         $programs = TenantProgramSpecial::all();
27
28         // Get SPK results if program is selected
29         $spkResults = null;
30         if ($programId) {
31             $spkResults = $this->getSpkSawRanking($programId);
32         }
33
34         return view('pages.central.masjids.beasiswa.seleksi.index', get_defined_vars());
35     } catch (\Exception $e) {
36         Log::error('Selection phase error: ' . $e->getMessage());
37         return redirect()->back()
38             ->with('error', 'Unable to fetch selection data: ' . $e->getMessage());
39     }
40 }
```

Controller `TenantPesertaSeleksiController` pada website Laravel, melalui metode `index`, mengelola tampilan data seleksi peserta beasiswa dalam sistem multi-tenant, mewarisi Controller, mengambil masjid pengguna via `getUserMasjid` dengan `Auth::id()`, mengalihkan ke rute `home` jika masjid tidak ditemukan, beralih ke basis data tenant menggunakan `switchToTenantDatabase`, mengambil peserta dengan status `"memenuhi_syarat"` pada tahap seleksi dari `SpkSawAlternative` dengan relasi program dan `scores.criteria`, memfilter berdasarkan `programId` jika ada, mengurutkan dan paginasi data, mengambil semua program dari `TenantProgramSpecial`, menghitung peringkat SPK jika `programId` dipilih via `getSpkSawRanking`, menangani error dengan logging melalui `Log` dan pengalihan dengan pesan, mendukung relasi User dan Masjid, serta memastikan keamanan dan konsistensi data dengan Eloquent ORM dan penanganan kesalahan Laravel.

```

1 public function storePenilaian(Request $request, $programId, $pesertaId)
2 {
3     $masjids = Masjid::where('user_id', auth()->id()->get());
4     $masjid = $masjids->first();
5
6     if (!$masjid) {
7         return redirect()->route('home')
8             ->with('error', 'No masjid found for this user.');
```

Metode storePenilaian dari TenantPesertaSeleksiController di website Laravel, melalui metode storePenilaian, menyimpan penilaian peserta beasiswa dalam sistem multi-tenant, mengambil masjid pengguna dengan Masjid::where('user_id', auth()->id()), mengalihkan ke rute home jika masjid tidak ditemukan dengan pesan error, beralih ke basis data tenant dengan switchToTenantDatabase, memulai transaksi database menggunakan DB::beginTransaction, mengambil kriteria dari SpkSawCriteria berdasarkan programId, menyimpan atau memperbarui skor di SpkSawScore untuk setiap kriteria berdasarkan input scores dan pesertaId, melakukan commit transaksi jika berhasil, mengalihkan ke rute penilaian dengan pesan sukses, menangani error dengan rollback dan logging melalui Log, mendukung relasi User dan Masjid, serta memastikan keamanan dan integritas data dengan Eloquent ORM dan penanganan kesalahan Laravel.

a. Controller Peserta Lolos Beasiswa

```
1 public function index(Request $request)
2 {
3     $masjids = Masjid::where('user_id', auth()->id()->get());
4     $masjid = $masjids->first();
5
6     if (!$masjid) {
7         return redirect()->route('home')
8             ->with('error', 'No masjid found for this user.');
```

```
9     }
10
11     $this->switchToTenantDatabase($masjid);
12
13     try {
14         $programId = $request->input('program_id');
15
16         $programs = TenantProgramSpecial::all();
17
18         $query = SpkSawAlternative::with(['user', 'program', 'scores.criteria'])
19             ->where('name_stage', 'lolos')
20             ->where('status', 'memenuhi_syarat');
```

```
21
22         if ($programId) {
23             $query->where('tenant_program_special_id', $programId);
24         }
25
26         $pesertaLolos = $query->orderBy('id', 'desc')->paginate(10);
27
28         $spkCalculations = null;
29         $selectedProgram = null;
30
31         if ($programId) {
32             $selectedProgram = TenantProgramSpecial::find($programId);
33             $spkCalculations = $this->getSpkSawCalculations($programId);
34         }
35
36         return view('pages.central.masjids.beasiswa.lolos.index', compact(
37             'pesertaLolos',
38             'programs',
39             'spkCalculations',
40             'selectedProgram',
41             'programId'
42         ));
43     } catch (\Exception $e) {
44         Log::error('Peserta lolos page error: ' . $e->getMessage());
45         return redirect()->back()
46             ->with('error', 'Unable to fetch lolos participants data: ' . $e->getMessage());
47     }
48 }
```

Metode index dari TenantPesertaLolosController di website Laravel menampilkan peserta beasiswa lolos dalam sistem multi-tenant, mengambil masjid pengguna via Masjid::where('user_id', auth()->id()->get()), mengalihkan ke home dengan pesan error jika masjid tidak ditemukan, beralih ke basis data tenant dengan switchToTenantDatabase, mengambil program dari TenantProgramSpecial, mengambil peserta dari SpkSawAlternative dengan status memenuhi_syarat pada tahap lolos beserta relasi user, program, dan scores.criteria, memfilter berdasarkan programId, mengurutkan dan paginasi 10 item, menghitung perhitungan SPK via getSpkSawCalculations jika programId ada, merender view pages.central.masjids.beasiswa.lolos.index dengan data, menangani error dengan logging via Log dan pengalihan, mendukung relasi User dan Masjid, serta memastikan keamanan dan integritas dengan Eloquent ORM dan penanganan kesalahan Laravel.

Lampiran 5. Sertifikat HKI

 REPUBLIK INDONESIA KEMENTERIAN HUKUM	
SURAT PENCATATAN CIPTAAN	
Dalam rangka perlindungan ciptaan di bidang ilmu pengetahuan, seni dan sastra berdasarkan Undang-Undang Nomor 28 Tahun 2014 tentang Hak Cipta, dengan ini menerangkan:	
Nomor dan tanggal permohonan	: EC002025075665, 25 Juni 2025
Pencipta	
Nama	: Mohamad Khadik, Ginanjar Wiro Sasmito dkk
Alamat	: Gang Sujai RT.004/RW.009 Desa Luvungragi, Bulakamba, Kab. Brebes, Jawa Tengah, 52253
Kewarganegaraan	: Indonesia
Pemegang Hak Cipta	
Nama	: Pusat Penelitian dan Pengabdian Masyarakat (P3M) Politeknik Harapan Bersama
Alamat	: Jalan Mataram No. 9, Pesurungan Lor, Kecamatan Margadana, Margadana, Kota Tegal, Jawa Tengah, 52142
Kewarganegaraan	: Indonesia
Jenis Ciptaan	: Program Komputer
Judul Ciptaan	: Masjid: Pengembangan Sistem Multi-Tenant pada Manajemen Masjid menggunakan Sistem Pendukung Keputusan dengan Metode SAW
Tanggal dan tempat diumumkan untuk pertama kali di wilayah Indonesia atau di luar wilayah Indonesia	: 24 Juni 2025, di Kota Tegal
Jangka waktu perlindungan	: Berlaku selama 50 (lima puluh) tahun sejak Ciptaan tersebut pertama kali dilakukan Pengumuman.
Nomor Pencatatan	: 000915926
adalah benar berdasarkan keterangan yang diberikan oleh Pemohon. Surat Pencatatan Hak Cipta atau produk Hak terkait ini sesuai dengan Pasal 72 Undang-Undang Nomor 28 Tahun 2014 tentang Hak Cipta.	
	a.n. MENTERI HUKUM DIREKTUR JENDERAL KEKAYAAN INTELEKTUAL u.b Direktur Hak Cipta dan Desain Industri Agung Damarsasongko,SH.,MH. NIP. 196912261994031001
	Disclaimer: 1. Dalam hal pemohon memberikan keterangan tidak sesuai dengan surat pernyataan, Menteri berwenang untuk mencabut surat pencatatan permohonan. 2. Surat Pencatatan ini telah disegel secara elektronik menggunakan segel elektronik yang diterbitkan oleh Balai Besar Sertifikasi Elektronik, Badan Siber dan Sandi Negara. 3. Surat Pencatatan ini dapat dibuktikan keasliannya dengan memindai kode QR pada dokumen ini dan informasi akan ditampilkan dalam browser.

LAMPIRAN PENCIPTA

No	Nama	Alamat
1	Mohamad Khadik	Gang Sujai RT.004/RW.009 Desa Luwunragi Bulakamba, Kab. Brebes
2	Ginanjari Wiro Sasmito	RT.003/RW.002, Desa Kluwut Bulakamba, Kab. Brebes
3	Muhammad Fikri Hidayattullah, S.T., M.Kom.	Jalan Glatik No.68 Tegal Selatan, Kota Tegal



Lampiran 6. Lembar Bimbingan



D IV TEKNIK INFORMATIKA POLITEKNIK HARAPAN BERSAMA

LEMBAR BIMBINGAN TUGAS AKHIR

Nama : Mohamad Khadik
 NIM : 21090055
 No. Ponsel : 085727235899
 Judul TA : PENGEMBANGAN SISTEM MULTI
 TENANT PADA MANAJEMEN MASJID
 MENGGUNAKAN SISTEM PENDUKUNG
 KEPUTUSAN DENGAN METODE SAW
 Dosen Pembimbing I : Ir. Ginanjar Wiro Sasmito, M.Kom., IPM.,
 ASEAN Eng.

No	Tanggal	Pemeriksaan	Perbaikan Yang Perlu Dilakukan	Paraf Pembimbing
1	15/25 4	Judul Rumus prosedur	ok. ok. Lanjut	f
2.	24/25 4	prosedur	Lanjut	f
3	4/25 6	prosedur	ok.	f
4	20/25 6	Rab 1	akhir	f

J	6/25 /7	Bat I	oke	f.
.6	7/25 /7	Ceporan	oke	f.
		Ace	Pisp yia	f.

--	--	--	--	--

Tegal, 15 Juli 2025
Dosen Pembimbing I



Ir. Ginanjar Wiro Sasmito, S.Kom., M.Kom
NIPY. 09.016.307



**D IV TEKNIK INFORMATIKA
POLITEKNIK HARAPAN BERSAMA**

LEMBAR BIMBINGAN TUGAS AKHIR

Nama : Mohamad Khadik
NIM : 21090055
No. Ponsel : 085727235899
Judul TA : Pengembangan Sistem Multi Tenant Pada
Manajemen Masjid Menggunakan Sistem
Pendukung Keputusan Dengan Metode SAW
Dosen Pembimbing I : Muhammad Fikri H., S.T., M.Kom.

No	Tanggal	Pemeriksaan	Perbaikan Yang Perlu Dilakukan	Paraf Pembimbing
1	21/09/2025	- Progres Aplikasi - Progres & Laporan	- Detail	
2	23/05/2025	- Progres observasi - Foto	- Secure dan Transaksi Uang wajib aman. - kembangkan Sesuai hasil observasi	
3	26/06/2025	- Laporan Bab 1	- Sitasi - Isi Pendahuluan.	

8	1/6	Pengajuan Hki		Xi
4				Xi
7	23/6	Btng Bab 1 Pedah ulan	- pendatu - Data observasi	
8	8/7	lepu	- LB tidak smi - Gay penelitian di tingg pemer blm als - foto observasi fiden als - UML	Xi
8	9/7	lepu	AEC! - Konfirmasi lepuan kesim ygn.	X

--	--	--	--	--

Tegal, 15 Juli 2025.
Dosen Pembimbing II


Muhammad Fikri H., S.T., M.Kom.
NIPY. 10.007.032

Lampiran 7. Foto wawancara

