

Pengembangan Web Serikat Pekerja Kampus

Framework: Codeigniter 4

Template: Edura

CSS: Tailwind

Database: MySQL

Web dan Sistem Informasi keanggotaan serikat pekerja kampus adalah portal registrasi keanggotaan, pengelolaan keanggotaan, penerbitan nomor anggota, penerbitan kartu anggota, penerimaan anggota, pemberhentian anggota, forum komunikasi anggota, tempat survei anggota, pengumpul dan pengolahan data statistik dari anggota maupun survei, dan informasi-informasi yang relevan bagi anggota. Sebelum pembuatan web dan sistem informasi yang baru ini, sudah ada lebih dari 1700 anggota yang terdaftar, data anggota lama ini akan di bulk upload ke sistem baru ini oleh superadmin, dan statusnya pending jika ada data yang belum terisi lengkap mengikuti kebutuhan data registrasi baru. Setelah melengkapi, baru akan aktif kembali. Keaktifan dapat kembali jika sudah mengupload bukti pembayaran iuran terakhir.

Role:

1) Role yang Digunakan

1. **Super Admin**
2. **Pengurus (Admin)**
3. **Pengurus Wilayah**
4. **Bendahara**
5. **Anggota (Member)**
6. **Calon Anggota (Candidate)**

Catatan: Secara desain, **Role** mengatur jenis pengguna, sedangkan **Status Keanggotaan** (candidate/active/disabled/rejected) mengatur apakah seseorang dianggap anggota aktif atau tidak. Namun karena Anda sudah mendefinisikan “Calon anggota” sebagai role fungsional, maka implementasi bisa:

- **Opsi A (disarankan):** role tetap 4 (super_admin, admin, coordinator, member) dan “calon anggota” ditentukan oleh membership_status=candidate.
- **Opsi B:** tetap pakai 5 role seperti permintaan Anda (lebih sederhana untuk UI), namun tetap simpan membership_status untuk logika verifikasi.

Dokumen ini menuliskan dalam format yang kompatibel dengan **Opsi B**, tetapi tetap bisa dipetakan ke Opsi A.

2) Prinsip Umum Akses

2.1 Prinsip RBAC (Role-Based Access Control)

- Akses fitur ditentukan oleh **role**.
- Tindakan sensitif (approve/disable/download data) dibatasi tambahan dengan **scope wilayah** dan **status keanggotaan**.

2.2 Prinsip Pembatasan Wilayah (Regional Scope)

Berlaku untuk **Pengurus Wilayah**:

- Hanya boleh mengakses/memberi tindakan pada anggota yang wilayahnya sesuai:
 - member.work_province atau member.region_code (disarankan pakai kode wilayah/provinsi)
- Pengurus wilayah **tidak** boleh:
 - mengubah role pengguna
 - mengakses anggota di luar wilayahnya
 - mengakses menu manajemen sistem

2.3 Prinsip Status Keanggotaan

- Jika membership_status != active maka fitur “anggota” tertentu harus dibatasi.
 - “Calon anggota” hanya punya akses konten publik terbatas + status verifikasi.
-

3) Matriks Fitur per Role

3.1 Calon Anggota (Candidate)

Akses Menu/Fitur:

- Profil (view)
- Status keanggotaan (indikator “sudah dikonfirmasi / belum”)
- AD/ART (view)
- Sejarah SPK (view)

- Manifesto SPK (view)
- Informasi Serikat yang bersifat publik (opsional)
- (Disarankan) Upload bukti pembayaran + verifikasi email (kalau modul onboarding terpisah)

Tidak boleh:

- Forum diskusi anggota
- Survei anggota
- Kartu anggota
- Pesan ke pengurus (opsional: boleh, tapi biasanya dibatasi)
- Join grup WhatsApp anggota/wilayah

3.2 Anggota (Member)

Akses Menu/Fitur:

- Profil (view)
- Edit Profil
- Kartu Anggota
- AD/ART
- Manifesto Serikat
- Informasi Serikat
- Sejarah SPK
- Ubah Password
- Forum Diskusi Anggota
- Survei Anggota (mengisi)
- Mengirim pesan ke pengurus
- Tombol gabung grup WhatsApp seluruh anggota
- Tombol gabung grup WhatsApp wilayah (sesuai provinsi domisili bekerja)

Tidak boleh:

- Mengonfirmasi calon anggota

- Menghapus anggota
 - Mengirim notifikasi massal
 - Akses data survei agregat/internal
 - Download data anggota
-

3.3 Pengurus (Admin)

Mewarisi seluruh akses Anggota, ditambah:

Manajemen Anggota & Onboarding:

- Melihat daftar calon anggota
- Mengonfirmasi calon anggota menjadi anggota aktif
- Menolak calon anggota
- Menonaktifkan anggota (sesuai kewenangan organisasi)
- Melihat data anggota (member list + detail)
- Menghapus anggota (*catatan: lebih aman “soft delete”/disable daripada hard delete*)

Konten & Komunikasi:

- Tambah Informasi Serikat
- Kirim tulisan blog
- Membuat informasi untuk notifikasi ke seluruh anggota
- Pengaduan masuk (view, kelola, tanggap)
- Menjawab dan mengirim pesan ke anggota

Survei:

- Membuat survei anggota
- Mengakses data survei anggota (hasil & rekap)

Tidak boleh:

- Role management global
- Menu management / sub menu management (kecuali diberi wewenang khusus)
- Mengubah role pengguna (kecuali Anda izinkan, tapi ini biasanya milik Super Admin)

3.4 Pengurus Wilayah

Mewarisi seluruh akses Pengurus, ditambah **kewenangan wilayah**:

Khusus Wilayah (Scope: wilayahnya saja):

- Download data anggota wilayahnya
- Mengaktifkan/menonaktifkan anggota wilayahnya
- Mengirim informasi ke anggota wilayah
- Update link grup wilayah (WhatsApp wilayah)
- Melihat grafik anggota wilayah
- Melihat dana iuran terkumpul dari anggota wilayah (*hanya wilayahnya*)

Batasan:

- Tidak boleh mengakses data wilayah lain
- Tidak boleh role management
- Tidak boleh menu management
- Tidak boleh mengirim notifikasi “seluruh anggota” (kecuali diberi izin eksplisit)

3.5 Super Admin

Akses FULL (ALL) + manajemen sistem:

Core System:

- Dashboard
- Role Management
- Menu Management
- Sub Menu Management
- Member List
- Member Management (aktif/nonaktif, verifikasi, suspend, delete)
- Mengubah role member
- Content Management (semua konten)
- Semua laporan & data (termasuk semua wilayah)

- Audit log (disarankan)
-

4) Daftar Permission (Rapi untuk Implementasi)

Agar mudah dibuat di DB (RBAC), gunakan permission berbasis aksi:

4.1 Permission: Konten & Informasi

- content.view_public
- content.view_member
- content.manage (buat/edit/hapus konten)
- blog.create
- blog.manage
- notification.send_all
- notification.send_region

4.2 Permission: Profil & Akun

- profile.view_self
- profile.edit_self
- password.change_self
- membercard.view_self

4.3 Permission: Anggota & Onboarding

- member.view_list
- member.view_detail
- member.approve_candidate
- member.reject_candidate
- member.disable
- member.enable
- member.delete (*opsional, biasanya super admin saja*)
- member.change_role (*super admin saja*)

4.4 Permission: Forum & Pesan

- forum.view

- forum.post
- forum.moderate (*opsional*)
- message.send_to_board
- message.reply_to_member
- complaint.view_inbox
- complaint.respond

4.5 Permission: Survei

- survey.fill
- survey.create
- survey.view_results

4.6 Permission: Wilayah (Scope-limited)

- region.member.download
- region.link.update
- region.stats.view
- region.dues.view

5) Mapping Role → Permission

5.1 Calon Anggota

- profile.view_self
- content.view_public
- content.view_member (*opsional: kalau AD/ART/manifesto dianggap member-only, pindahkan ke public khusus calon*)
- membership.status.view (*bisa jadi permission tersendiri atau bagian profil*)

5.2 Anggota

- profile.view_self
- profile.edit_self
- password.change_self
- membercard.view_self

- content.view_public
- content.view_member
- forum.view
- forum.post
- survey.fill
- message.send_to_board
- whatsapp.join_all_members
- whatsapp.join_region

5.3 Pengurus (Admin)

Semua permission Anggota +:

- member.view_list
- member.view_detail
- member.approve_candidate
- member.reject_candidate
- member.disable
- member.enable
- member.delete (*opsional; atau hanya Super Admin*)
- content.manage
- blog.create
- blog.manage
- notification.send_all
- complaint.view_inbox
- complaint.respond
- message.reply_to_member
- survey.create
- survey.view_results

5.4 Pengurus Wilayah

Semua permission Pengurus + (scope wilayah):

- notification.send_region
- region.member.download
- region.link.update
- region.stats.view
- region.dues.view
- member.disable (*dibatasi wilayah*)
- member.enable (*dibatasi wilayah*)

5.5 Super Admin

Semua permission +:

- role.manage
- menu.manage
- submenu.manage
- member.change_role
- audit.view (*disarankan*)

6) Aturan Validasi Akses (Rules Engine)

6.1 Rule: Calon anggota tidak boleh akses fitur anggota

Jika membership_status != active maka blok:

- forum, survei, kartu anggota, join WA, pesan internal (opsional), dsb.

6.2 Rule: Pengurus Wilayah hanya untuk wilayahnya

Saat akses anggota (view/disable/enable/download):

- member.region_code == coordinator.region_code
Jika tidak sama → **403 Forbidden**

6.3 Rule: Pengurus vs Super Admin

- Pengurus boleh kelola konten & onboarding.
- Super Admin mengelola struktur sistem (role/menu).

7) Rekomendasi Struktur Menu Berdasarkan Role

7.1 Calon Anggota

- Beranda
- Profil
- AD/ART
- Manifesto
- Sejarah SPK
- Status Keanggotaan

7.2 Anggota

- Beranda
- Profil
- Kartu Anggota
- Informasi Serikat
- AD/ART
- Manifesto
- Sejarah SPK
- Forum Anggota
- Survei Anggota
- Pesan ke Pengurus
- Join WA (All + Wilayah)
- Ubah Password

7.3 Pengurus

- Semua menu Anggota +
- Manajemen Anggota
- Calon Anggota (Konfirmasi)
- Pengaduan Masuk
- Kelola Informasi/Blog
- Kelola Survei
- Broadcast Notifikasi

7.4 Pengurus Wilayah

- Semua menu Pengurus +
- Anggota Wilayah (list + download)
- Notifikasi Wilayah
- Link Grup Wilayah
- Statistik Wilayah
- Dana Iuran Wilayah

7.5 Super Admin

- Semua +
- Dashboard Admin
- Role Management
- Menu Management
- Sub Menu Management
- Global Member Management
- Content Management (ALL)
- Audit Log

8) Catatan Implementasi Praktis (CI4)

- Simpan role di `sp_members.role` (ENUM) sesuai 5 role di atas.
- Simpan `membership_status` untuk membedakan calon/aktif/nonaktif.
- Terapkan middleware/filter (CI4 Filters) seperti:
 - `auth` (harus login)
 - `role:super_admin|admin|coordinator|member|candidate`
 - `membership:active` (untuk fitur member-only)
 - `region_scope` (untuk coordinator)

Konten public page minimal:

1. Landing Page '/' yang semua kontennya di atur dari Super admin, berisi:

- a. Tentang Serikat Pekerja Kampus (isi konten dinamis di isikan super-admin)
 - b. Statistik anggota beserta demografi
 - c. Publikasi terkini
 - d. Tombol bergabung
 - e. Tombol login
 - f. Pengurus serikat pekerja kampus
 - g. Subscribe informasi
 - h. Footer
2. Page /sejarah
 3. Page /manifesto
 4. Page /visimisi
 5. Page /ad-art
 6. Page struktur /pengurus
 7. Page /publikasi -> berisi pdf yang dapat didownload seperti Policy brief
 8. Page /regulasi -> berisi pdf yang dapat didownload seperti Undang-undang, Peraturan Pemerintah, Peraturan menteri
 9. Page /news Berita SPK -> seperti blog/news berisi teks dan gambar yang diposting pengurus/admin
 10. Page /contact -> kontak ke pengurus SPK

Target halaman:

- / (Landing Page)
- /sejarah
- /manifesto
- /visimisi
- /ad-art
- /pengurus (struktur)
- /publikasi (PDF download)
- /regulasi (PDF download)
- /news (berita/blog, teks + gambar)
- /contact (kontak + form pesan)

1) Konsep CMS: Tipe Konten yang Dikelola

1.1 Content Types

1. **Pages** (halaman statis berbasis teks/HTML)
 - /sejarah, /manifesto, /visimisi, /ad-art, /contact
 2. **Home Sections** (blok/blok konten khusus landing page /)
 - About, Statistik, Publikasi terkini, CTA join/login, Pengurus, Subscribe, Footer
 3. **Documents** (PDF library)
 - /publikasi (Policy brief, dll)
 - /regulasi (UU, PP, Permen, dll)
 4. **News Posts** (blog/news)
 - /news list + detail per artikel
 5. **Officers/Board** (struktur pengurus)
 - data person + jabatan + periode + foto
 6. **Subscribers** (subscribe informasi)
 7. **Contact Messages** (pesan dari halaman kontak)
-

2) Skema Database CMS (Ternormalisasi & Siap Panel Admin)

Semua tabel CMS **terpisah** dari tabel keanggotaan (sp_members) namun dapat mengacu ke sp_members.id sebagai created_by, updated_by, atau member_id.

2.1 Tabel: cms_pages (Halaman Statis)

Menyimpan konten untuk /sejarah, /manifesto, /visimisi, /ad-art, /contact.

Kolom inti:

- id (PK)
- slug (unik) — contoh: sejarah, manifesto, visimisi, ad-art, contact
- title
- content_html (TEXT/LONGTEXT)
- template (ENUM) — default, legal, contact
- status (ENUM) — draft, published, archived
- visibility (ENUM) — public, member_only (opsional)

- primary_document_id (FK nullable → cms_documents.id) (opsional, jika halaman butuh tautan PDF utama)
- published_at (DATETIME nullable)
- created_by (FK nullable → sp_members.id)
- updated_by (FK nullable → sp_members.id)
- timestamps: created_at, updated_at

Index penting:

- UNIQUE(slug)
 - INDEX(status, published_at)
-

2.2 Tabel: cms_page_revisions (Versioning Halaman)

Agar konten legal (AD/ART, manifesto) punya jejak revisi.

- id (PK)
- page_id (FK → cms_pages.id)
- content_html (snapshot)
- note (TEXT nullable) — alasan perubahan
- created_by (FK → sp_members.id)
- created_at

Kebijakan:

- Setiap “Publish” → buat 1 revision snapshot.
 - Bisa “rollback” ke revision tertentu.
-

2.3 Tabel: cms_home_sections (Landing Page Builder untuk /)

Landing page hanya diatur oleh **Super Admin**, jadi kontennya dibuat sebagai section yang terstruktur.

Kolom inti:

- id (PK)
- section_key (UNIQUE) — nilai tetap:
 - about

- stats
- latest_publications
- cta_join
- cta_login
- officers
- subscribe
- footer
- title (nullable, tergantung section)
- body_html (TEXT nullable) — untuk section teks
- config_json (JSON/TEXT) — pengaturan khusus section (contoh di bawah)
- sort_order (INT)
- is_enabled (TINYINT)
- updated_by (FK → sp_members.id)
- updated_at (DATETIME)

Contoh config_json per section:

- stats: { "mode": "dynamic", "cache_minutes": 60, "show_gender": true, "show_province": true }
- latest_publications: { "source": "cms_documents", "type": "publikasi", "limit": 6 }
- cta_join: { "button_text": "Bergabung", "url": "/register" }
- cta_login: { "button_text": "Login", "url": "/login" }
- subscribe: { "double_opt_in": true }
- footer: { "address": "...", "email": "...", "socials": [{ "label": "IG", "url": "..." }] }

2.4 Tabel: cms_documents (PDF Library)

Untuk /publikasi dan /regulasi.

Kolom inti:

- id (PK)
- doc_type (ENUM) — publikasi, regulasi

- title
- slug (unik, opsional) — untuk detail view jika diperlukan
- description (TEXT nullable)
- category_id (FK nullable → cms_document_categories.id)
- file_path (VARCHAR 255) — lokasi file di storage
- original_name (VARCHAR 255)
- mime_type (VARCHAR 100) — harus application/pdf
- file_size (INT)
- checksum_sha256 (CHAR 64 nullable)
- status (ENUM) — draft, published, archived
- published_at (DATETIME nullable)
- created_by, updated_by (FK → sp_members.id)
- timestamps

Index penting:

- INDEX(doc_type, status, published_at)
- INDEX(category_id)

2.5 Tabel: cms_document_categories (Kategori Dokumen)

- id (PK)
- doc_type (ENUM) — publikasi, regulasi
- name
- slug (unik)
- sort_order

2.6 Tabel: cms_news_posts (Berita/Blog)

Untuk /news.

Kolom inti:

- id (PK)

- title
- slug (unik)
- excerpt (TEXT nullable)
- content_html (LONGTEXT)
- cover_image_id (FK nullable → cms_media.id)
- status (ENUM) — draft, published, archived
- published_at (DATETIME nullable)
- author_id (FK → sp_members.id) — Pengurus/Admin yang posting
- timestamps

Index penting:

- UNIQUE(slug)
- INDEX(status, published_at)
- INDEX(author_id)

2.7 Tabel: cms_media (Gambar/Asset)

Dipakai oleh news & profil pengurus.

- id (PK)
- media_type (ENUM) — image
- file_path
- original_name
- mime_type (image/jpeg, image/png, image/webp)
- file_size
- checksum_sha256 (nullable)
- uploaded_by (FK → sp_members.id)
- uploaded_at

2.8 Tabel: cms_officers (Struktur Pengurus /pengurus)

Struktur dapat berdiri sendiri atau menaut ke member.

Kolom inti:

- id (PK)
 - member_id (FK nullable → sp_members.id) — jika pengurus juga punya akun
 - full_name (jika member_id null)
 - position_title (VARCHAR) — Ketua, Sekretaris, dll
 - level (ENUM) — pusat, wilayah
 - region_code (VARCHAR nullable) — wajib jika level wilayah
 - photo_media_id (FK nullable → cms_media.id)
 - bio_html (TEXT nullable)
 - sort_order (INT)
 - is_active (TINYINT)
 - period_start (DATE nullable)
 - period_end (DATE nullable)
 - timestamps
-

2.9 Tabel: cms_subscribers (Subscribe Informasi)

- id (PK)
 - email (unik)
 - status (ENUM) — pending, active, unsubscribed
 - token_hash (CHAR 64 nullable) — untuk double opt-in
 - verified_at (DATETIME nullable)
 - created_at
-

2.10 Tabel: cms_contact_messages (Inbox Kontak /contact)

- id (PK)
- name
- email
- subject (nullable)

- message (TEXT)
 - status (ENUM) — new, in_progress, closed
 - assigned_to (FK nullable → sp_members.id) — pengurus yang menangani
 - admin_reply (TEXT nullable) — bila balas dari panel (opsional)
 - created_at, updated_at
-

3) Hak Akses CMS (Siap Diterapkan di RBAC)

3.1 Aturan Khusus Landing Page

- **Hanya Super Admin** yang boleh CRUD cms_home_sections (Landing /).

3.2 Aturan Page Statis

Disarankan:

- **Super Admin:** CRUD semua pages + publish
- **Pengurus/Admin:** boleh edit **beberapa** pages (mis. /contact) tetapi **tidak** boleh edit page legal (opsional kebijakan)
- **Pengurus Wilayah:** tidak mengelola pages statis global

3.3 Aturan Dokumen

- **Super Admin:** semua dokumen
- **Pengurus/Admin:** CRUD dokumen publikasi & regulasi, publish (atau butuh approval super admin bila ingin ketat)
- **Pengurus Wilayah:** opsional hanya dokumen wilayah (jika ada), default tidak

3.4 Aturan News

- **Pengurus/Admin:** CRUD cms_news_posts + publish
 - **Pengurus Wilayah:** bisa publish news bertag wilayah (jika Anda tambah field wilayah), atau hanya draft
 - **Super Admin:** full
-

4) Struktur Panel Admin (Menu CMS)

4.1 Super Admin

- CMS

- Landing Page Builder (cms_home_sections)
- Pages (cms_pages + revisions)
- Documents
 - Publikasi
 - Regulasi
 - Categories
- News (posts + categories jika ditambah)
- Officers (struktur pengurus)
- Subscribers
- Contact Inbox
- Media Library
- (Optional) Site Settings (footer/social)

4.2 Pengurus/Admin

- CMS
 - Pages (dibatasi)
 - Documents (publikasi/regulasi)
 - News
 - Officers (jika diizinkan)
 - Contact Inbox
 - Media Library

5) Rendering Halaman (Cara Website Menampilkan Konten)

5.1 Landing Page /

Render berdasarkan cms_home_sections (urutan sort_order, hanya yang is_enabled=1):

1. about → title/body_html
2. stats → blok statistik **dinamis** (lihat bagian 6)
3. latest_publications → ambil cms_documents doc_type=publikasi status=published (limit sesuai config)

4. cta_join → tombol join (url dari config)
5. cta_login → tombol login (url dari config)
6. officers → ambil cms_officers level=pusat is_active=1
7. subscribe → form subscribe
8. footer → config_json (alamat, kontak, social)

5.2 Pages Statis

- /sejarah → cms_pages.slug='sejarah' AND status='published'
- /manifesto
- /visimisi
- /ad-art
- /contact (konten + form submit)

5.3 Documents

- /publikasi → list cms_documents.doc_type='publikasi' AND status='published'
- /regulasi → list cms_documents.doc_type='regulasi' AND status='published'
- Download endpoint wajib mengirim header aman (Content-Type: application/pdf).

5.4 News

- /news → list cms_news_posts.status='published' ORDER BY published_at DESC
- /news/{slug} → detail post

6) Modul Statistik Anggota & Demografi (Landing Section “Stats”)

6.1 Sumber Data

Ambil dari sp_members (anggota) dengan filter:

- membership_status='active' (anggota aktif)
- opsional: account_status='active'

Statistik yang disarankan:

- Total anggota aktif
- Distribusi gender (gender)

- Distribusi provinsi domisili bekerja (region_code/province)
- Distribusi status kerja (employment_status)
- Distribusi jabatan akademik (academic_rank)

6.2 Caching (disarankan)

Agar landing tidak berat:

- Buat tabel cms_stats_cache:
 - cache_key (UNIQUE) contoh: member_demographics
 - payload_json (JSON)
 - generated_at
- Update via cron/command (mis. tiap 60 menit) sesuai cache_minutes di config stats.

7) Workflow Konten (Draft → Publish) dan Revisions

7.1 Pages (Sejarah/Manifesto/VisiMisi/AD-ART/Contact)

- Editor membuat perubahan → simpan sebagai draft
- Saat publish:
 - set status='published', set published_at
 - buat snapshot ke cms_page_revisions (untuk audit & rollback)

7.2 Documents

- Upload PDF → draft
- Verifikasi metadata → publish
- Setiap publish update published_at

7.3 News

- Pengurus menulis draft + cover image
- Publish
- Bisa schedule (optional): publish_at di masa depan

8) Endpoint yang Dibutuhkan (Ringkas)

Public

- GET / (landing)
- GET /sejarah, /manifesto, /visimisi, /ad-art
- GET /publikasi, GET /regulasi
- GET /documents/{id}/download (atau /publikasi/{slug})
- GET /news, GET /news/{slug}
- POST /subscribe (double opt-in)
- POST /contact (kirim pesan)

Admin (Protected)

- GET/POST /admin/cms/home (Super Admin only)
- CRUD /admin/cms/pages
- CRUD /admin/cms/documents + categories
- CRUD /admin/cms/news
- CRUD /admin/cms/officers
- GET /admin/cms/subscribers
- GET /admin/cms/contact-inbox
- CRUD /admin/cms/media

9) Keamanan & Validasi (Wajib)

9.1 Upload File

- Dokumen: hanya application/pdf, max size (mis. 5MB)
- Media: hanya image (jpg/png/webp), max size (mis. 2MB)
- Simpan file dengan **random name**, bukan nama asli
- Simpan checksum_sha256 untuk integritas

9.2 Sanitasi Konten HTML

- Gunakan whitelist tag (atau WYSIWYG yang menghasilkan HTML aman)
- Jangan izinkan script tag

9.3 Kontrol Akses

- Landing builder: role=super_admin
- News/Documents: role in (super_admin, admin)
- Contact inbox: role in (super_admin, admin, coordinator) sesuai kebijakan

10) Minimum Deliverables untuk Developer

1. Migrations untuk tabel CMS:
 - cms_pages, cms_page_revisions, cms_home_sections, cms_documents, cms_document_categories, cms_news_posts, cms_media, cms_officers, cms_subscribers, cms_contact_messages, (opsional) cms_stats_cache
2. Model CI4 untuk masing-masing tabel
3. Controller public render:
 - HomeController, PageController, DocumentController, NewsController, ContactController
4. Controller admin CMS + middleware filters untuk role
5. Storage service untuk upload & download file aman
6. Scheduler/command untuk cache statistik demografi

1) Super Admin

A. Sistem & Konfigurasi (Global)

- Konfigurasi seluruh website dan sistem (site settings, integrasi, parameter umum).
- Manajemen Menu & Sub Menu (struktur navigasi dan visibilitas).
- Manajemen CMS (semua halaman, dokumen, news, struktur pengurus, footer, landing builder).

B. Role & Access Control (Global)

- Tambah role.
- Modifikasi role access (permission mapping).

- Mengubah role pengguna (promote/demote): anggota → admin / pengurus wilayah / role lain.

C. Master Data & Migrasi Data (Global)

- Bulk upload anggota lama via template Excel.
- Bulk upload master data dropdown registrasi:
 - Provinsi, Kabupaten/Kota, Universitas, Program Studi, dan master lain.
- Menambah anggota baru secara manual (input satuan).

D. Governance & Audit (Global)

- Melihat aktivitas seluruh anggota (audit log, histori perubahan, aksi admin).
- Akses laporan global (statistik anggota, iuran, konten, forum, survei).

Catatan: Semua kontrol Admin otomatis termasuk di Super Admin.

2) Admin (Pengurus Pusat)

A. Onboarding & Keanggotaan (Global)

- Verifikasi/approve calon anggota menjadi anggota aktif.
- Menolak pendaftaran calon anggota (dengan alasan).
- Mengubah status anggota: aktif, suspend, nonaktif/pengunduran diri.
- Menandai kelengkapan onboarding:
 - status pembayaran/iuran (verifikasi/validasi administrasi),
 - kelengkapan profil (foto, data wajib),
 - status join grup (umum/wilayah/kluster).
- Kelola data anggota (CRUD profil anggota) dengan pembatasan keamanan (riwayat perubahan tercatat).

B. Komunikasi & Notifikasi (Global)

- Mengirim pesan ke anggota (1:1) dan membalas pesan masuk.
- Mengirim informasi/broadcast ke seluruh anggota (notifikasi).
- Mengelola kontak masuk/pengaduan (inbox, respons, status).

C. Konten Publik & Internal (Global)

- Mengelola postingan publik (news/blog) termasuk draft/publish.

- Mengelola dokumen publik (publikasi/regulasi) jika wewenang diberikan oleh Super Admin.

D. Survei (Global)

- CRUD survei anggota (buat, edit, publish, tutup).
- Mengakses data & rekap hasil survei anggota.

E. Forum & Moderasi (Global)

- Moderasi forum (CRUD thread & komentar, hide/delete, pin/lock jika ada).
- Penanganan laporan konten (report/flag).

F. Manajemen Link Grup WhatsApp (Global)

- CRUD link grup WhatsApp:
 - Umum (seluruh anggota),
 - Wilayah,
 - Kluster (opsional).
- Pengaturan visibilitas link (siapa yang boleh lihat).

G. Delegasi Wilayah (Global, dengan kontrol ketat)

- Menetapkan/assign Pengurus Wilayah:
 - memilih anggota → jadikan pengurus wilayah,
 - menentukan provinsi/wilayah scope pengurus wilayah.

Catatan penting: Admin boleh “assign pengurus wilayah”, tetapi tidak mengelola sistem role/permission (itu Super Admin). Admin hanya memilih dari role yang sudah tersedia.

3) Pengurus Wilayah

Prinsip utama: Semua kontrol hanya untuk anggota dalam wilayahnya (scope by region/province).

A. Keanggotaan Wilayah (Scope Wilayah)

- Melihat daftar anggota di wilayahnya + detail profil terbatas (sesuai kebijakan privasi).
- Mengelola status administratif anggota wilayah:
 - menandai “sudah join grup wilayah”,

- o memantau kelengkapan data wilayah (opsional).
- Mengusulkan tindakan ke Admin Pusat:
 - o usulan penonaktifan/suspend/penertiban (dengan alasan dan bukti).

B. Komunikasi Wilayah (Scope Wilayah)

- Mengirim informasi/notifikasi ke anggota wilayah.
- Mengelola pesan dari anggota wilayah (jika fitur pesan wilayah diperlukan).

C. Link Grup WhatsApp Wilayah (Scope Wilayah)

- Membuat/mengubah link WhatsApp wilayah yang dapat diklik anggota wilayah.

D. Survei Wilayah (Scope Wilayah)

- CRUD survei untuk anggota wilayah (jika Anda ingin survei per wilayah).
- Melihat hasil survei wilayah.

E. Forum & Moderasi (Scope Wilayah atau Global-Moderation Terbatas)

Dua opsi desain (pilih salah satu):

- Opsi 1 (lebih ideal): Moderasi hanya untuk konten yang dibuat/berasal dari wilayahnya (thread ber-tag wilayah).
- Opsi 2: Moderasi global terbatas (mis. hide konten, report ke admin).

F. Delegasi Pengurus Wilayah (Scope Wilayah, rekomendasi: usulan saja)

Agar tidak terjadi eskalasi tanpa kontrol:

- Pengurus wilayah mengusulkan anggota wilayah untuk dijadikan pengurus wilayah tambahan.
- Persetujuan final tetap oleh Admin Pusat atau Super Admin.

Permission Key (siap masuk RBAC)

Super Admin

- system.config.manage
- rbac.role.create
- rbac.role.update
- rbac.role.assign
- menu.manage
- cms.manage_all

- masterdata.bulk_import
- members.bulk_import
- audit.view_all

Admin (Pengurus Pusat)

- members.approve
- members.reject
- members.view_all
- members.update_all
- members.suspend
- members.disable
- broadcast.send_all
- news.manage
- docs.manage (*opsional*)
- survey.manage_all
- forum.moderate_all
- whatsapp.manage_all
- region_admin.assign_coordinator

Pengurus Wilayah

- members.view_region
- members.update_region_limited
- broadcast.send_region
- survey.manage_region
- forum.moderate_region
- whatsapp.manage_region
- members.propose_action (*usulan suspend/disable*)

RBAC

- rbac_roles 1—N rbac_role_permissions N—1 rbac_permissions

Menu

- rbac_menus 1—N rbac_submenus
- rbac_menus N—N rbac_permissions via rbac_menu_permissions
- rbac_submenus N—N rbac_permissions via rbac_submenu_permissions

Logika visibilitas

- Item (menu/submenu) **tampil** jika:
 - is_public = 1, atau
 - user punya permission yang dipersyaratkan item itu, sesuai permission_logic:
 - ANY = minimal punya salah satu permission
 - ALL = wajib punya semua permission

Role Baru: Kebendaharaan (Treasurer)

1.1 Tujuan Role

Role **Kebendaharaan** fokus pada:

- pengelolaan **iuran (tagihan, pembayaran, status)**,
- verifikasi bukti pembayaran dan progres bulanan,
- penanganan klaim/dispensasi iuran,
- penegakan aturan keterlambatan (≥ 3 bulan) dengan mekanisme “pending/suspend”.

Prinsip: Kebendaharaan **bukan** pengelola konten, forum, survei, atau manajemen role. Fokusnya finansial & kepatuhan iuran.

2) Cakupan Fitur Kebendaharaan

A. Master Iuran (Rate & Policy)

1. Kelola Master Tarif Iuran

- CRUD tabel sp_dues_rates (golongan & gaji)
- Aktivasi/nonaktif tarif
- Set prioritas/sort order

2. Kebijakan Iuran

- definisi periode iuran (bulanan)

- grace period (mis. sampai tgl 10 tiap bulan)
- denda (opsional)
- aturan “menunggak 3 bulan → pending”

Catatan: perubahan tarif/policy sebaiknya tercatat audit.

B. Billing Bulanan (Generate Tagihan Otomatis)

1. Generate tagihan iuran bulanan

- otomatis per periode (mis. YYYY-MM)
- untuk semua anggota aktif (atau per wilayah)

2. Re-generate / perbaikan tagihan

- jika ada anggota baru masuk di tengah periode
- jika tarif anggota berubah (golongan/gaji berubah)

3. Pengaturan “exclusion”

- anggota tertentu bebas iuran (waived) untuk periode tertentu (opsional, butuh alasan)
-

C. Pembayaran & Verifikasi Bukti Bayar

1. Input pembayaran (manual/transfer)

- unggah bukti pembayaran (PDF/JPG/PNG)
- isi metadata (tanggal bayar, metode, nominal, catatan)

2. Verifikasi pembayaran

- status: submitted → verified / rejected
- set bill: unpaid → paid

3. Resubmission bukti

- jika ditolak, anggota upload ulang bukti
-

D. Monitoring Progres Iuran Bulanan

1. Dashboard progres:

- jumlah anggota wajib iuran vs sudah bayar vs belum bayar
- progress per periode

2. Filter:

- per wilayah/provinsi
- per kategori tarif (golongan/gaji)
- per status anggota

3. Export:

- laporan bulanan (CSV/Excel) untuk rekap internal

E. Klaim Iuran (Claim/Dispute)

Definisi klaim iuran: anggota mengajukan klaim terkait pembayaran yang sudah dilakukan, misalnya:

- “Saya sudah bayar tapi belum terverifikasi”
- “Nominal salah”
- “Double payment”
- “Salah periode”
- “Saya dispensasi (waiver) karena kondisi tertentu”

Fitur:

1. Form klaim dari anggota
2. Verifikasi klaim oleh bendahara
3. Keputusan klaim:
 - approved → koreksi bill/payment
 - rejected → sertakan alasan
4. Jejak koreksi (audit):
 - siapa yang memproses
 - perubahan apa yang dilakukan

F. Penegakan Aturan Tunggakan (≥ 3 Bulan)

1. Auto-flag tunggakan

- sistem menghitung: periode tidak dibayar berturut-turut atau total tunggakan

2. Notifikasi bertahap

- bulan ke-1: reminder
- bulan ke-2: warning
- bulan ke-3: final notice + status pending

3. Tindakan “Pending/Suspend”

- Kebendaharaan dapat mengubah `account_status` atau `membership_status` menjadi “pending/suspended” berdasarkan kebijakan:
 - rekomendasi aman: `account_status=suspended` namun `membership_status` tetap active atau jadi inactive (sesuai AD/ART)
- Sistem menyimpan alasan: “Menunggak > 3 bulan”

4. Re-aktivasi setelah bayar

- jika tunggakan dilunasi → status kembali normal (butuh verifikasi)

Catatan tata kelola: Anda bisa pilih apakah “pending” itu:

- hanya menonaktifkan akses fitur anggota (forum, survei, kartu),
- atau benar-benar suspend login.

G. Integrasi Wilayah (Jika Ada Dana Wilayah)

Untuk Koordinator Wilayah:

- bisa melihat rekap **iuran wilayah** (read-only)
- kebendaharaan mengelola semua transaksi (write)
- jika ingin lebih granular: kebendaharaan wilayah (opsional) untuk scope provinsi

3) Batasan & Hubungan dengan Role Lain (Anti-Redundansi)

Kebendaharaan vs Admin (Pengurus Pusat)

- Admin fokus: verifikasi keanggotaan, konten, forum, survei, broadcast umum, struktur organisasi.
- Kebendaharaan fokus: iuran (billing, payment, klaim, tunggakan).
- Admin boleh melihat ringkasan iuran (read-only), tapi tidak mengubah transaksi.

Kebendaharaan vs Pengurus Wilayah

- Pengurus wilayah: melihat rekap wilayah + follow-up anggota wilayah (read-only, push informasi).
- Kebendaharaan: verifikasi pembayaran & klaim (write), termasuk penetapan pending karena tunggakan.

4) Permission Keys (Siap RBAC)

Kebendaharaan (Treasurer)

Master

- dues.rate.manage
- dues.policy.manage (*opsional, bisa super admin/admin saja*)

Billing

- dues.bill.generate
- dues.bill.view_all
- dues.bill.update

Payments

- dues.payment.view_all
- dues.payment.verify
- dues.payment.reject
- dues.payment.adjust (*koreksi nominal/periode, sangat sensitif*)

Claims

- dues.claim.view_all
- dues.claim.process

Arrears/Enforcement

- dues.arrears.view_all

- dues.arrears.enforce_pending (*set pending/suspend*)
- dues.arrears.reactivate

Reporting

- dues.report.export_all
- dues.dashboard.view_all

Notifikasi finansial

- dues.notify.send (*reminder tunggakan, info verifikasi ditolak*)

Read-only untuk Admin/Pengurus Wilayah (opsional)

- Admin: dues.dashboard.view_all, dues.report.export_all (atau view saja)
- Pengurus wilayah: dues.dashboard.view_region, dues.report.export_region

5) Rancangan Data yang Disarankan (Tambahan Tabel/Field)

Agar fitur kebendaharaan rapi, saya sarankan menambahkan modul iuran bulanan berikut (di luar master rate sp_dues_rates yang sudah ada):

5.1 sp_member_dues_profile (Profil iuran per anggota)

- member_id (PK/FK)
- scheme_type ENUM(golongan,gaji)
- rate_id FK → sp_dues_rates.id
- effective_from (DATE)
- effective_to (DATE nullable)
- is_waived (TINYINT) + waived_reason

5.2 sp_dues_bills (sudah ada, tinggal diperluas)

Tambahkan:

- period_year SMALLINT
- period_month TINYINT
- unique index (member_id,bill_type,period_year,period_month)
- arrears_level TINYINT (0/1/2/3) opsional untuk monitoring

5.3 sp_dues_payments (sudah ada)

Tambahkan:

- ref_no (VARCHAR) opsional (nomor referensi transfer)
- rejected_reason

5.4 sp_dues_claims (baru)

- id, member_id, bill_id, payment_id nullable
- claim_type
ENUM(already_paid,wrong_amount,wrong_period,double_payment,waiver_request,other)
- description
- status ENUM(submitted,in_review,approved,rejected)
- processed_by, processed_at, decision_note

5.5 sp_member_sanctions (opsional, untuk pending/suspend karena iuran)

- id, member_id
 - sanction_type ENUM(arrears_pending,arrears_suspend)
 - start_at, end_at nullable
 - reason, created_by
-

6) UX/Panel Kebendaharaan (Menu)

1. Dashboard Iuran
 - summary: paid/unpaid/overdue
 - grafik tren pembayaran
2. Tagihan Bulanan
 - generate periode
 - list bills + filter
3. Pembayaran Masuk
 - list submitted
 - view bukti bayar
 - verify / reject
4. Klaim Iuran

- list klaim masuk
- proses klaim
- 5. Tunggakan > 3 Bulan
 - daftar anggota menunggak
 - kirim reminder
 - set pending/suspend
 - reaktivasi setelah lunas
- 6. Master Tarif Iuran
- 7. Laporan & Export

7) Aturan Otomasi yang Relevan (Ideal)

1. **Scheduler bulanan:**
 - generate bills untuk bulan berjalan (tgl 1)
2. **Reminder otomatis:**
 - H-3 sebelum due date, H+7 setelah due date
3. **Auto-flag arrears:**
 - setiap hari/hari tertentu, hitung tunggakan
4. **Auto-pending:**
 - jika unpaid \geq 3 periode berturut-turut \rightarrow set pending (dengan log & notifikasi)

Halaman registrasi anggota (Gabung SPK):

1. Klik Bergabung
2. Masukkan email password
3. Isi data registrasi
4. Menyetujui untuk taat pada AD/ART
5. Muncul nominal pembayaran berdasarkan rentang gaji
 - a. 1, 'Golongan I (Ia, Ib, Ic, Id)', iuran yang harus dibayar-> 'Rp.20.000,-'
 - b. 2, 'Golongan II (IIa, IIb, IIc, IId)', iuran yang harus dibayar->'Rp.30.000,-'
 - c. 3, 'Golongan III (IIIa, IIIb, IIIc, Id)', iuran yang harus dibayar->'Rp.35.000,-'
 - d. 4, 'Golongan IV (IVa, IVb, IVc, IVd, IVe)', iuran yang harus dibayar->'Rp.45.000,-'

- e. 5, 'Rp0 - Rp1,500,000', iuran yang harus dibayar-> 'Rp.7.500,-'
 - f. 6, 'Rp1,500,000 - Rp3,000,000', iuran yang harus dibayar-> 'Rp.15.000,-'
 - g. 7, 'Rp3,000,001 - Rp6,000,000', iuran yang harus dibayar-> 'Rp.30.000,-'
 - h. 8, 'Diatas Rp.6,000,000', iuran yang harus dibayar-> 'Rp.60.000,-'
6. Nominal yang harus dibayar disertai gambar QR Code dan nomor rekening
 7. Upload bukti pembayaran.
 8. Klik mendaftar.
 - Setelah membayar/upload bukti pembayaran dan verifikasi email, calon anggota dapat login.
 - Setelah login-> laman login terbatas hanya halaman profile, edit profile, edit password, membaca manifesto, membaca sejarah, membaca AD-ART, membuka dan mengisi survei yang di assign superadmin,
 - Di halaman profile belum dimunculkan nomor anggota sebelum di approve admin
 - Ada akses kontak/chat ke admin/pengurus
 - Ada tombol batalkan pendaftaran/mengundurkan diri jadi anggota (calon anggota maupun anggota yang sudah di approve)
 -

Struktur data registrasi:

// Contoh: \$this->forge->addField([...]);

[

// --- 1. IDENTITAS SISTEM (SYSTEM ID) ---

'id' => [
 'type' => 'INT',
 'constraint' => 11,
 'unsigned' => true,
 'auto_increment' => true,

],

// Public identifier (lebih aman untuk expose di URL/API daripada ID incremental)

'uuid' => [
 'type' => 'CHAR',
 'constraint' => 36,

],

'member_number' => [

```
'type'    => 'VARCHAR',
'constraint' => '50',
'null'     => true,
],
'email' => [
  'type'    => 'VARCHAR',
  'constraint' => '150',
],
'password_hash' => [
  'type'    => 'VARCHAR',
  'constraint' => '255',
],
'account_status' => [
  'type'    => 'ENUM',
  'constraint' => ['pending', 'active', 'suspended', 'rejected'],
  'default'  => 'pending',
],
'role' => [
  'type'    => 'ENUM',
  'constraint' => ['member', 'admin', 'moderator'],
  'default'  => 'member',
],
```

// [BARU] Verifikasi & keamanan akun

```
'email_verified_at' => [
  'type' => 'DATETIME',
  'null' => true,
],
```

```
'last_login_at' => [  
  'type' => 'DATETIME',  
  'null' => true,  
],  
'last_login_ip' => [  
  'type'    => 'VARCHAR',  
  'constraint' => 45, // IPv4/IPv6  
  'null'    => true,  
],  
'last_user_agent' => [  
  'type'    => 'VARCHAR',  
  'constraint' => 255,  
  'null'    => true,  
],  
'failed_login_attempts' => [  
  'type'    => 'TINYINT',  
  'constraint' => 3,  
  'unsigned' => true,  
  'default'  => 0,  
],  
'locked_until' => [  
  'type' => 'DATETIME',  
  'null' => true,  
],  
'password_changed_at' => [  
  'type' => 'DATETIME',  
  'null' => true,  
],
```

```
// Simpan HASH token (bukan token mentah)
```

```
'reset_token_hash' => [
  'type'    => 'VARCHAR',
  'constraint' => 255,
  'null'    => true,
],
'reset_token_expires_at' => [
  'type' => 'DATETIME',
  'null' => true,
],
'remember_token_hash' => [
  'type'    => 'VARCHAR',
  'constraint' => 255,
  'null'    => true,
],
```

```
// --- 2. DATA DEMOGRAFIS (DEMOGRAPHICS) ---
```

```
'full_name' => [
    'type'    => 'VARCHAR',
    'constraint' => '150',
],
'gender' => [
    'type'    => 'ENUM',
    'constraint' => ['L', 'P'],
    'null'    => true,
],
'birth_place' => [
    'type'    => 'VARCHAR',
```



```
'constraint' => 100,
'null'    => true,
],
'birth_date' => [
  'type' => 'DATE',
  'null' => true,
],
// Nomor identitas (opsional; bila dipakai, batasi akses & pertimbangkan enkripsi)
'identity_number' => [
  'type'    => 'VARCHAR',
  'constraint' => 50,
  'null'    => true,
],
'address' => [
  'type' => 'TEXT',
  'null' => true,
],
'province' => [
  'type'    => 'VARCHAR',
  'constraint' => 100,
  'null'    => true,
],
'city' => [
  'type'    => 'VARCHAR',
  'constraint' => 100,
  'null'    => true,
],
'district' => [
```

```
'type'    => 'VARCHAR',
'constraint' => 100,
'null'     => true,
],
'postal_code' => [
  'type'    => 'VARCHAR',
  'constraint' => 10,
  'null'     => true,
],
'phone_number' => [
  'type'    => 'VARCHAR',
  'constraint' => '20',
  'null'     => true,
],
'alt_phone_number' => [
  'type'    => 'VARCHAR',
  'constraint' => '20',
  'null'     => true,
],
// [BARU] Kontak darurat
'emergency_contact_name' => [
  'type'    => 'VARCHAR',
  'constraint' => 150,
  'null'     => true,
],
'emergency_contact_relation' => [
  'type'    => 'VARCHAR',
  'constraint' => 50,
```

```

    'null'    => true,
],
'emergency_contact_phone' => [
    'type'    => 'VARCHAR',
    'constraint' => 20,
    'null'    => true,
],
// Tetap Anda boleh pakai region (kalau masih dibutuhkan)
'region' => [
    'type'    => 'VARCHAR',
    'constraint' => '100',
    'null'    => true,
],

// --- 3. VALIDITAS PROFESI (PROFESSIONAL IDENTITY) ---
'university_name' => [
    'type'    => 'VARCHAR',
    'constraint' => '150',
],
'campus_location' => [
    'type'    => 'VARCHAR',
    'constraint' => 150,
    'null'    => true,
],
'faculty' => [
    'type'    => 'VARCHAR',
    'constraint' => 150,
    'null'    => true,

```

```

],
'department' => [
  'type'    => 'VARCHAR',
  'constraint' => '100',
  'null'    => true,
],
'work_unit' => [
  'type'    => 'VARCHAR',
  'constraint' => '100',
  'null'    => true,
],
'employee_id_number' => [
  'type'    => 'VARCHAR',
  'constraint' => '50',
  'null'    => true,
],
// khusus dosen bila perlu
'lecturer_id_number' => [
  'type'    => 'VARCHAR',
  'constraint' => 50, // NIDN/NIDK, dsb
  'null'    => true,
],
'academic_rank' => [
  'type'    => 'ENUM',
  'constraint' => [
    'Tenaga Pengajar', 'Asisten Ahli', 'Lektor',
    'Lektor Kepala', 'Guru Besar', 'Tendik/Staff', 'Lainnya'
  ],
],

```

```
    'default' => 'Lainnya',
  ],
  'employment_status' => [
    'type'    => 'ENUM',
    'constraint' => [
      'PNS', 'PPPK', 'Tetap Non-PNS',
      'Kontrak/PKWT', 'Dosen Luar Biasa', 'Honoror', 'Lainnya'
    ],
    'default' => 'Lainnya',
  ],
  'employment_start_date' => [
    'type' => 'DATE',
    'null' => true,
  ],
  'contract_end_date' => [
    'type' => 'DATE',
    'null' => true,
  ],
],
```

// --- 4. DATA EKONOMI & PEKERJAAN (ECONOMIC DATA) ---

```
'payroll_source' => [
  'type'    => 'VARCHAR',
  'constraint' => '100',
  'null'    => true,
],
'salary_range' => [
  'type'    => 'VARCHAR',
  'constraint' => '50',
```

```

    'null'    => true,
],
'base_salary' => [
    'type'    => 'DECIMAL',
    'constraint' => '15,2',
    'null'    => true,
    'default'  => null,
],
'take_home_pay' => [
    'type'    => 'DECIMAL',
    'constraint' => '15,2',
    'null'    => true,
    'default'  => null,
],
// [BARU] Persetujuan eksplisit untuk data sensitif (mis. gaji)
'consent_sensitive_data' => [
    'type'    => 'TINYINT',
    'constraint' => 1,
    'default'  => 0,
],
// [BARU] Iuran (opsional kalau ingin di-handle di tabel ini)
'dues_method' => [
    'type'    => 'ENUM',
    'constraint' => ['fixed', 'percentage', 'manual'],
    'default'  => 'manual',
],
'dues_amount' => [
    'type'    => 'DECIMAL',

```

```
'constraint' => '15,2',
'null'    => true,
'default' => null,
],
'dues_percentage' => [
  'type'    => 'DECIMAL',
  'constraint' => '5,2',
  'null'    => true,
  'default' => null,
],
'dues_status' => [
  'type'    => 'ENUM',
  'constraint' => ['unpaid', 'paid', 'overdue', 'waived'],
  'default' => 'unpaid',
],
'dues_last_paid_at' => [
  'type' => 'DATE',
  'null' => true,
],
'expertise' => [
  'type' => 'TEXT',
  'null' => true,
],
```

// --- 5. ADVOKASI & ORGANISASI (ADVOCACY DATA) ---

```
'motivation' => [
  'type' => 'TEXT',
  'null' => true,
```

```
],
'advocacy_interests' => [
  'type' => 'TEXT',
  'null' => true,
],
'is_volunteer' => [
  'type'    => 'TINYINT',
  'constraint' => 1,
  'default'  => 0,
],
// [BARU] Keanggotaan & struktur organisasi internal
'membership_status' => [
  'type'    => 'ENUM',
  'constraint' => ['applied', 'approved', 'inactive', 'resigned', 'expired'],
  'default'  => 'applied',
],
'branch_name' => [
  'type'    => 'VARCHAR',
  'constraint' => 150,
  'null'    => true,
],
'union_position' => [
  'type'    => 'VARCHAR',
  'constraint' => 100,
  'null'    => true,
],
'membership_start_date' => [
  'type' => 'DATE',
```



```
    'null' => true,  
  ],  
  'membership_end_date' => [  
    'type' => 'DATE',  
    'null' => true,  
  ],
```

```
// --- 6. ADMINISTRASI (ADMIN & FILES) ---
```

```
'id_proof_file' => [  
  'type'    => 'VARCHAR',  
  'constraint' => '255',  
  'null'    => true,  
],
```

```
// [BARU] Dokumen pendukung lain (opsional)
```

```
'employee_card_file' => [  
  'type'    => 'VARCHAR',  
  'constraint' => 255,  
  'null'    => true,  
],
```

```
'membership_form_file' => [  
  'type'    => 'VARCHAR',  
  'constraint' => 255,  
  'null'    => true,  
],
```

```
'profile_photo_file' => [  
  'type'    => 'VARCHAR',  
  'constraint' => 255,  
  'null'    => true,
```

],

// [BARU] Alur verifikasi admin

'verification_status' => [

 'type' => 'ENUM',

 'constraint' => ['unverified', 'under_review', 'verified', 'rejected'],

 'default' => 'unverified',

],

'verified_by' => [

 'type' => 'INT',

 'constraint' => 11,

 'unsigned' => true,

 'null' => true,

],

'verified_at' => [

 'type' => 'DATETIME',

 'null' => true,

],

'rejection_reason' => [

 'type' => 'TEXT',

 'null' => true,

],

'agreed_to_terms_at' => [

 'type' => 'DATETIME',

 'null' => true,

],

'agreed_to_privacy_at' => [

```

    'type' => 'DATETIME',
    'null' => true,
  ],
  // [BARU] Catatan internal admin
  'admin_notes' => [
    'type' => 'TEXT',
    'null' => true,
  ],

  // --- TIMESTAMPS ---
  'joined_at' => ['type' => 'DATE', 'null' => true],
  'created_at' => ['type' => 'DATETIME', 'null' => true],
  'updated_at' => ['type' => 'DATETIME', 'null' => true],
  'deleted_at' => ['type' => 'DATETIME', 'null' => true],
]

```

1) Ringkasan Kebutuhan & Alur Bisnis

1.1 Tujuan

Menyediakan proses pendaftaran anggota yang rapi dan terkontrol dengan tahapan:

1. User mendaftar → status **calon anggota (candidate)**
2. User wajib **membayar iuran pendaftaran** dan **upload bukti pembayaran**
3. Setelah bukti pembayaran diupload → user wajib **verifikasi email**
4. Setelah email terverifikasi → **admin/pengurus memverifikasi** menjadi **anggota aktif**
5. Admin dapat **menolak** atau **menonaktifkan** anggota (aktif maupun calon)

1.2 Status & State yang digunakan

A. membership_status (status keanggotaan serikat)

- candidate : calon anggota
- active : anggota aktif

- inactive : anggota tidak aktif (opsional untuk pengelolaan periodik)
- disabled : dinonaktifkan/ditangguhkan pengurus
- rejected : ditolak pengurus

B. onboarding_state (tahapan onboarding / proses daftar)

- registered : baru daftar (belum upload bukti bayar)
- payment_submitted : sudah upload bukti bayar
- email_verified : email sudah diverifikasi setelah upload bukti
- approved : sudah disetujui admin → aktif
- rejected : ditolak admin

C. account_status (akses sistem / login)

- pending : belum aktif login penuh (umumnya untuk calon)
- active : bisa akses penuh
- suspended : akses dibekukan
- rejected : ditolak

Rekomendasi aturan:

- candidate umumnya account_status=pending
- active wajib account_status=active
- disabled wajib account_status=suspended
- rejected wajib account_status=rejected

2) Arsitektur Database (Normalisasi)

2.1 Tabel Utama

- sp_members : akun + profil minimal + status + state
- sp_dues_bills : tagihan iuran (pembayaran terstruktur, bisa diperluas bulanan)
- sp_dues_payments : pembayaran tagihan (verifikasi admin)
- sp_member_documents : semua dokumen (KTP, bukti bayar, dsb)
- sp_email_verifications : token verifikasi email (hash)

- `sp_audit_logs` : jejak tindakan penting (approve/reject/disable, verifikasi pembayaran, dsb)

2.2 Relasi Ringkas

- `sp_members` (1) -> (N) `sp_dues_bills`
- `sp_dues_bills` (1) -> (1) `sp_dues_payments` (disarankan 1 pembayaran utama per tagihan)
- `sp_members` (1) -> (N) `sp_member_documents`
- `sp_members` (1) -> (N) `sp_email_verifications`
- `sp_members` (1) -> (N) `sp_audit_logs` (sebagai actor atau target)
- `sp_dues_payments` -> `sp_member_documents` via `proof_document_id`

2.3 Prinsip Desain

- **Dokumen dipisahkan** agar tidak menumpuk kolom file pada `sp_members`.
- **Pembayaran dipisahkan** agar ada proses verifikasi dan histori (bisa dikembangkan).
- **Audit log wajib** untuk tindakan admin agar akuntabel.

3) Struktur Folder Implementasi (CI4)

Rekomendasi penempatan:

- `app/Database/Migrations/` : 6 migration
- `app/Models/` : model per tabel
- `app/Services/` : `MemberOnboardingService` (state machine + transaksi)
- `app/Controllers/` : endpoint user & endpoint admin
- `app/Config/Routes.php` : definisi rute
- `app/Config/Services.php` : registrasi service (opsional)

4) Langkah Implementasi

4.1 Buat Migrations

Buat migration dalam urutan aman FK:

1. `CreateSpMembers`

2. CreateSpDuesBills
3. CreateSpMemberDocuments
4. CreateSpDuesPayments
5. CreateSpEmailVerifications
6. CreateSpAuditLogs

Jalankan:

php spark migrate

Validasi:

- Pastikan engine InnoDB (untuk FK).
- Pastikan semua index unik untuk email, uuid, dan member_number.

5) Konfigurasi Service

5.1 Registrasi Service (opsional)

Jika ingin memanggil via service('memberOnboarding'), tambahkan di app/Config/Services.php:

```
public static function memberOnboarding($getShared = true)
{
    if ($getShared) {
        return static::getSharedInstance('memberOnboarding');
    }
    return new \App\Services\MemberOnboardingService();
}
```

6) Alur Implementasi Fitur (State Machine)

6.1 Register (Calon Anggota)

Tujuan: Membuat akun calon anggota dan tagihan iuran pendaftaran.

Langkah backend:

1. Validasi input (email unik, password minimal, nama lengkap).

2. password_hash().
3. Insert ke sp_members:
 - membership_status=candidate
 - onboarding_state=registered
 - account_status=pending
4. Buat sp_dues_bills:
 - bill_type=registration
 - bill_status=unpaid
 - due_date opsional
5. Simpan sp_audit_logs: member_registered.

Output ke user:

- Informasi instruksi pembayaran (nominal, rekening, batas waktu).
 - Simpan member_id + bill_id untuk proses upload bukti.
-

6.2 Upload Bukti Pembayaran

Tujuan: Menampung bukti bayar dan menandai pembayaran sebagai submitted.

Langkah backend:

1. Validasi hak akses: user hanya boleh upload untuk bill miliknya.
2. Validasi file (lihat bagian Security).
3. Simpan file (disarankan di writable/uploads/...).
4. Insert dokumen ke sp_member_documents:
 - doc_type=dues_payment_proof
 - review_status=not_reviewed
5. Upsert sp_dues_payments:
 - bill_id unik (1 pembayaran utama per bill)
 - payment_status=submitted
 - proof_document_id mengarah ke dokumen

6. Update `sp_members.onboarding_state=payment_submitted` (jangan menurunkan state yang sudah lebih maju)
7. Simpan `sp_audit_logs: payment_proof_submitted`.

Catatan penting:

- Pada tahap ini, proses verifikasi email baru boleh dibuat (sesuai requirement).
-

6.3 Verifikasi Email (Setelah Upload Bukti)

Tujuan: Mengonfirmasi email user setelah bukti bayar diupload.

Langkah backend: Create token

1. Pastikan `onboarding_state=payment_submitted`.
2. Generate token mentah `random_bytes`, simpan **hash** ke `sp_email_verifications`.
3. Kirim email berisi link verifikasi: `.../verify-email?token=....`
4. `sp_audit_logs: email_verification_token_created`.

Langkah backend: Verify token

1. Ambil token, hash, cari di `sp_email_verifications` (pastikan belum `used_at` dan belum expired).
 2. Set `sp_members.email_verified_at=NOW()`.
 3. Jika state `payment_submitted` → update `onboarding_state=email_verified`.
 4. Set `sp_email_verifications.used_at=NOW()`.
 5. `sp_audit_logs: email_verified`.
-

6.4 Verifikasi Admin → Anggota Aktif

Tujuan: Admin mengubah calon anggota menjadi anggota aktif, sekaligus mengesahkan pembayaran.

Prasyarat (disarankan wajib):

- `sp_members.onboarding_state = email_verified`
- Ada registration bill
- `sp_dues_payments.payment_status = submitted` untuk bill tersebut

Langkah backend:

1. Validasi role admin (admin atau moderator).
 2. Set pembayaran:
 - `sp_dues_payments.payment_status = verified`
 - `verified_by, verified_at, verification_note`
 3. Set tagihan:
 - `sp_dues_bills.bill_status = paid`
 4. Update member:
 - `membership_status=active`
 - `account_status=active`
 - `onboarding_state=approved`
 - generate member_number jika kosong
 - `reviewed_by, reviewed_at, review_note`
 5. `sp_audit_logs: admin_approved_member`.
-

6.5 Admin Menolak / Menonaktifkan

Reject (menolak calon anggota):

- Set:
 - `membership_status=rejected`
 - `account_status=rejected`
 - `onboarding_state=rejected`
 - isi alasan pada `review_note`
- `sp_audit_logs: admin_rejected_member`

Disable (menonaktifkan / suspend):

- Set:
 - `membership_status=disabled`
 - `account_status=suspended`
 - simpan alasan
- `sp_audit_logs: admin_disabled_member`

Enable kembali (opsional):

- Jika sudah approved → kembali active/active.
 - Jika belum approved → kembali candidate/pending.
-

7) Endpoint API / Controller (Rekomendasi)

7.1 Endpoint User

- POST /register
 - input: email, password, full_name, phone, address
 - output: member_id, bill_id, instruksi bayar
- POST /dues/{billId}/upload-proof
 - multipart file + meta pembayaran
- POST /email/verification/request
 - membuat token (hanya jika state payment_submitted)
- GET /email/verification/verify?token=...
 - verifikasi token

7.2 Endpoint Admin

- GET /admin/candidates
 - filter: payment_submitted, email_verified
 - POST /admin/members/{id}/approve
 - POST /admin/members/{id}/reject
 - POST /admin/members/{id}/disable
 - POST /admin/members/{id}/enable
 - GET /admin/payments/pending
 - payment_status=submitted
-

8) Query Dashboard Admin (Contoh SQL)

8.1 Daftar calon anggota yang sudah upload bukti dan verifikasi email

```
SELECT id, full_name, email, onboarding_state, membership_status, created_at
```

```
FROM sp_members  
WHERE membership_status='candidate'  
AND onboarding_state='email_verified'  
ORDER BY updated_at DESC;
```

8.2 Pembayaran pendaftaran yang menunggu verifikasi

```
SELECT p.id, p.bill_id, p.member_id, p.payment_status, p.paid_amount, p.paid_at,  
       m.full_name, m.email  
FROM sp_dues_payments p  
JOIN sp_members m ON m.id = p.member_id  
WHERE p.payment_status='submitted'  
ORDER BY p.updated_at DESC;
```

8.3 Detail bukti pembayaran (file)

```
SELECT p.id AS payment_id, d.file_path, d.original_name, d.mime_type, d.uploaded_at  
FROM sp_dues_payments p  
JOIN sp_member_documents d ON d.id = p.proof_document_id  
WHERE p.bill_id = ?;
```

9) Aturan Validasi & Security (Wajib)

9.1 Password

- Gunakan password_hash(PASSWORD_DEFAULT).
- Jangan pernah simpan password plaintext.

9.2 Token Email

- Simpan **hash token** saja (sha256) di DB.
- Token mentah hanya ada di link email.

9.3 Upload Bukti Pembayaran

Rekomendasi validasi:

- allowed mime: image/jpeg, image/png, application/pdf
- max size (mis. 2–5 MB)

- simpan di writable/uploads/dues_proofs/{member_uuid}/...
- simpan checksum_sha256 untuk integritas file
- jangan gunakan nama file asli sebagai nama file storage (gunakan random name)

9.4 Otorisasi

- User hanya boleh akses bill dan dokumen miliknya.
- Endpoint admin wajib role in (admin, moderator).

9.5 Transaksi Database

- Gunakan transaksi (\$db->transStart() / transComplete()) pada operasi:
 - submit bukti bayar (insert dokumen + upsert payment + update member)
 - verify email (update token + update member)
 - approve/reject/disable (update payment + update bill + update member + audit)

9.6 Audit Log

Wajib catat event:

- register
- submit payment proof
- create token verifikasi email
- verify email
- approve/reject/disable/enable
- verifikasi pembayaran