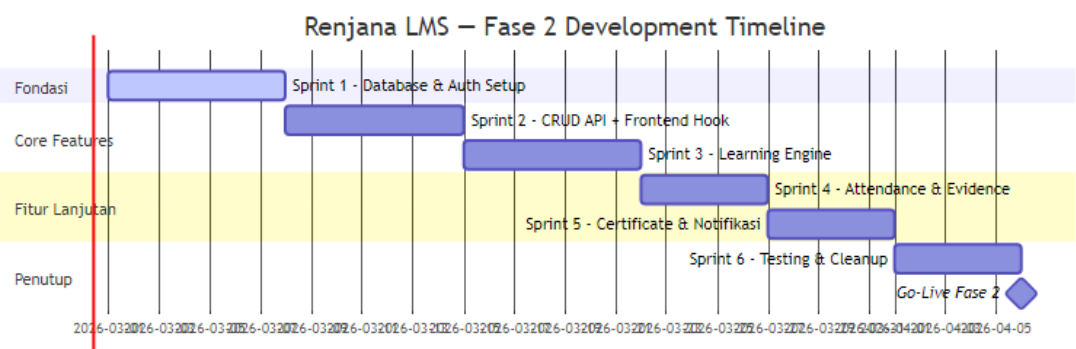


Rencana Pengembangan Fase 2 — Renjana LMS

Project: Renjana LMS **Date:** 28 Februari 2026 **Status:** Planning

Dokumen ini adalah panduan teknis dan sprint plan untuk **Fase 2: Development**. Setiap sprint menghasilkan kode yang berjalan, dokumentasi, dan commit ke GitHub.

Gambaran Umum



Sprint 1 - Fondasi: Database & Authentication

Target: 7 hari | **Commit Tag:** feat/sprint1-database-auth

Tugas Teknis

#	Task	File / Lokasi	Detail
1.0	Setup Docker PostgreSQL	docker-compose.dev.yml	Jalankan PostgreSQL 16 via Docker Desktop (tidak perlu install manual)
1.1	Inisialisasi Prisma	prisma/schema.prisma	npx prisma init + setup DATABASE_URL di .env
1.2	Buat schema DB	prisma/schema.prisma	Model: User, Course, Module, Lesson, Enrollment, Progress, Attendance
1.3	Jalankan migrasi	Terminal	npx prisma migrate dev --name init
1.4	Buat seed script	prisma/seed.ts	Import data dari lib/data/*.ts, hash password pakai bcrypt
1.5	Setup NextAuth	src/app/api/auth/[...nextauth]/route.ts	Credentials Provider + JWT strategy
1.6	Buat middleware RBAC	src/middleware.ts	Matchers sesuai route group (/dashboard, /admin, dsb.)
1.7	Update Login Form	src/components/auth/login-form.tsx	Ganti mock login → signIn() dari NextAuth

Docker Setup Lokal (Dev)

Buat file `docker-compose.dev.yml` di root project untuk PostgreSQL + Prisma Studio:

```
# docker-compose.dev.yml
services:
  postgres:
    image: postgres:16-alpine
    environment:
      POSTGRES_USER: renjana
      POSTGRES_PASSWORD: renjana_dev
      POSTGRES_DB: renjana_db
    ports:
      - "5432:5432"
    volumes:
      - postgres_data:/var/lib/postgresql/data

  adminer:
    image: adminer
    ports:
      - "8080:8080"
    depends_on:
      - postgres

volumes:
  postgres_data:
```

Jalankan dengan: `docker compose -f docker-compose.dev.yml up -d`

Isi `.env` :

```
DATABASE_URL="postgresql://renjana:renjana_dev@localhost:5432/renjana_db"
NEXTAUTH_SECRET="ganti-dengan-random-string-panjang"
NEXTAUTH_URL="http://localhost:3000"
```

Dependencies Baru

```
npm install @prisma/client prisma bcryptjs next-auth
npm install -D @types/bcryptjs
```

Dokumentasi Output

- `docs/sprint1_report.md` — Hasil, screenshot DB, dan catatan masalah

Commit Strategy

```
git add prisma/ docker-compose.dev.yml .env.example src/middleware.ts src/app/api/auth/
src/components/auth/
git commit -m "feat(sprint1): setup prisma + postgres docker + nextauth authentication"
git push origin main
```

Sprint 2 - Core API & Frontend Integration

Target: 7 hari | **Commit Tag:** feat/sprint2-api-crud

Tugas Teknis

#	Task	File / Lokasi	Detail
2.1	API: List Courses	src/app/api/courses/route.ts	GET — filter per role
2.2	API: Course Detail	src/app/api/courses/[id]/route.ts	GET — return modules & lessons
2.3	API: Enrollments	src/app/api/enrollments/route.ts	POST — daftarkan user ke kursus
2.4	API: Users (Admin)	src/app/api/admin/users/route.ts	GET, POST, PUT, DELETE
2.5	Hubungkan Dashboard	src/app/dashboard/page.tsx	Ganti mock data → fetch('/api/...')
2.6	Hubungkan Admin Panel	src/app/admin/programs/page.tsx	Tambah CRUD form yang berfungsi
2.7	Setup React Query	src/providers.tsx	Wrapper QueryClientProvider untuk cache

Dokumentasi Output

- docs/sprint2_report.md — Screenshot API responses, daftar endpoint aktif

Commit Strategy

```
git add src/app/api/ src/app/dashboard/ src/app/admin/
git commit -m "feat(sprint2): core REST APIs and frontend data integration"
git push origin main
```

Sprint 3 - Learning Engine (Progres & Tracker)

Target: 7 hari | **Commit Tag:** feat/sprint3-learning-engine

Tugas Teknis

#	Task	File / Lokasi	Detail
3.1	API: Update Progress	src/app/api/progress/route.ts	PUT — update saat lesson selesai
3.2	API: Get Progress	src/app/api/progress/[enrollmentId]/route.ts	GET — untuk tampil di dashboard
3.3	Continue Learning Card	src/components/dashboard/continue-card.tsx	Widget resume dengan progress bar dinamis
3.4	Visual Timeline	src/app/dashboard/page.tsx	Render milestone per modul (selesai/current/belum)

3.5	Halaman Learning	src/app/learn/[courseId]/page.tsx	Berfungsi dengan data asli, tracker tiap video/kuis
-----	------------------	-----------------------------------	---

Dokumentasi Output

- docs/sprint3_report.md — Screenshot progress tracker, video demo learner flow

Commit Strategy

```
git add src/app/api/progress/ src/components/dashboard/ src/app/learn/
git commit -m "feat(sprint3): learning engine with progress tracking and visual timeline"
git push origin main
```

Sprint 4 - Attendance & Evidence System

Target: 5 hari | Commit Tag: feat/sprint4-attendance

Tugas Teknis

#	Task	File / Lokasi	Detail
4.1	API: Check-In	src/app/api/attendance/route.ts	POST — simpan timestamp + GPS coords
4.2	API: Evidence Upload	src/app/api/evidence/route.ts	POST multipart/form-data → simpan ke storage
4.3	Update Check-in UI	src/app/dashboard/checkin/page.tsx	Integrasikan GPS Geolocation API browser
4.4	Evidence Form	src/app/dashboard/evidence/page.tsx	File uploader → kirim ke API
4.5	Instructor View	src/app/instructor/attendance/page.tsx	Tabel kehadiran peserta per kelas

Dokumentasi Output

- docs/sprint4_report.md — Screenshot attendance log, bukti upload berhasil

Commit Strategy

```
git add src/app/api/attendance/ src/app/api/evidence/ src/app/dashboard/checkin/
src/app/dashboard/evidence/ src/app/instructor/
git commit -m "feat(sprint4): attendance check-in with GPS and evidence upload"
git push origin main
```

Sprint 5 - Certificate Generator & Notifikasi

Target: 5 hari | Commit Tag: feat/sprint5-cert-notif

Tugas Teknis

#	Task	File / Lokasi	Detail
5.1	Install PDF lib	Terminal	<code>npm install @react-pdf/renderer</code> atau <code>puppeteer</code>
5.2	Buat template sertifikat	<code>src/components/certificate/template.tsx</code>	Desain sertifikat sebagai React component
5.3	API: Generate PDF	<code>src/app/api/certificates/[enrollmentId]/route.ts</code>	Render template → return PDF Buffer
5.4	Trigger otomatis	<code>prisma/seed.ts</code> (atau webhook)	Cek 100% completion → generate sertifikat
5.5	Setup Reminder Job	<code>src/app/api/cron/reminder/route.ts</code>	Vercel Cron / node-cron → query idle users
5.6	Integrasi Email	<code>src/lib/mailer.ts</code>	Nodemailer/Resend → kirim email reminder

Dokumentasi Output

- `docs/sprint5_report.md` — Contoh PDF sertifikat, log email reminder terkirim

Commit Strategy

```
git add src/app/api/certificates/ src/components/certificate/ src/app/api/cron/ src/lib/mailer.ts
git commit -m "feat(sprint5): auto certificate generation and email reminder system"
git push origin main
```

Sprint 6 - Testing, Polish & Go-Live Prep

Target: 5 hari | Commit Tag: `feat/sprint6-testing`

Tugas Teknis

#	Task	Detail
6.1	Setup Jest	Unit test API Routes dan utility functions
6.2	Setup Playwright	E2E test: Login, Enroll, Progress, Sertifikat
6.3	Audit Keamanan	Pastikan semua API Route pakai <code>getSession()</code> check
6.4	Environment Staging	Buat <code>.env.production</code> dan test Docker build
6.5	Update README.md	Langkah-langkah setup + ENV yang dibutuhkan
6.6	Final Docs	Update <code>01_Assessment_Report</code> → note "RESOLVED" di setiap issue

Commit Strategy

```
git add tests/ src/ docs/ README.md
git commit -m "chore(sprint6): add tests, security audit, and production readiness"
git push origin main --tags
```

Dokumentasi Per Fase

Setiap sprint akan menghasilkan satu file laporan:

Dokumen	File	Isi
Sprint 1 Report	docs/sprint1_report.md	Setup DB, schema, auth working
Sprint 2 Report	docs/sprint2_report.md	API endpoints, CRUD admin
Sprint 3 Report	docs/sprint3_report.md	Learning engine, progress tracker
Sprint 4 Report	docs/sprint4_report.md	Attendance, evidence upload
Sprint 5 Report	docs/sprint5_report.md	Certificate, email reminder
Sprint 6 Report	docs/sprint6_report.md	Tests, security, Go-Live checklist

Risiko & Mitigasi

Risiko	Kemungkinan	Mitigasi
PostgreSQL tidak tersedia di server	Sedang	Gunakan Docker + .env untuk URL; fallback SQLite untuk dev lokal
SMTP/WA API quota limit	Sedang	Queue reminder, batasi pengiriman 1x/hari per user
Breaking changes saat migrasi mock → real API	Tinggi	Migrasi satu halaman per sprint, jangan rewrite semuanya sekaligus
JWT token expiry race conditions	Rendah	Implementasikan refresh token strategy di NextAuth config

Dokumen ini menjadi panduan eksekusi Fase 2 dan akan diperbarui setiap sprint selesai.