GenLaravel

AI-Powered Laravel Project Generator

LAPORAN TUGAS UAS

Mata Kuliah Project Management

Disusun Oleh:

Kelompok 7

Fikri Armia Fahmi (2023071018)

Nadia (2024071004)

UNIVERSITAS PEMBANGUNAN JAYA

Program Studi Informatika

2025

DAFTAR ISI

Project Planning Document

Project: MCP Agents Laravel UI Generator
Sponsor: Internal Development Team
Project Manager: Fikri Armia Fahmi (2023071018)
Frontend Developer: **Nadia (2024071004)**
Tanggal: 18 September 2025

---

1. Project Initiation

1.1 Latar Belakang

Bagi mahasiswa maupun developer pemula, pengembangan komponen UI di Laravel seringkali memakan waktu karena harus membuat **Blade components**, **routes**, dan melakukan **validasi manual** satu per satu. Proses ini tidak hanya teknis, tapi juga menyita banyak waktu belajar. Dengan hadirnya **AI multi-agent system**, proses tersebut dapat dipercepat melalui otomatisasi sehingga mahasiswa dapat lebih fokus pada **logika aplikasi, inovasi, dan penelitian**, bukan hanya pada pekerjaan repetitif.

1.2 Tujuan

- Mempercepat pembuatan Laravel UI dari natural language prompts.
- Menyediakan preview interaktif sebelum integrasi.
- Meningkatkan kualitas kode dengan sistem validasi otomatis.

1.3 Stakeholder

- **Sponsor:** Internal Research & Development Team
- **Developer:** AI/ML Engineers, Laravel Developers, Frontend Developer
- **User:** Laravel developers & software houses

---

2. Feasibility Study

Aspek	Evaluasi	Detail
Teknis	✅ Layak	Python 3.7+, Laravel 8+, Cerebras SDK, Mistral SDK
Ekonomi	✅ Layak	Open-source MIT License, biaya server minimal
Operasional	✅ Layak	Developer dapat menjalankan script Python, integrasi Laravel
Hukum	✅ Layak	MIT License, sesuai standar open-source
Waktu	✅ Layak	Estimasi 12–16 minggu development agile

3. Project Planning (updated)

3.1 Information (basic)

- **Project Name:** MCP Agents Laravel UI Generator
- **Primary Goal:** Automate Laravel UI creation from natural-language prompts using an AI multi-agent system.
- **Main Deliverables:**
    - AI agent modules (10 agents)
    - Laravel UI generator (Blade files, layouts, components)
    - Validation & integration system
    - Front-end preview UI (HTML/CSS/JS)
    - Agent backend services & API
    - Documentation & user guide

3.2 Scope & Deliverables (expanded)

1. **Prompt Processing Agent** — expand & plan user prompts
2. **Draft Generator** — generate HTML preview (front-end + backend integration)
3. **Layout & Component Generator** — Blade files, layouts, components
4. **Route Generator** — update routes/web.php and scaffolding
5. **Validator** — syntax, structure, linting, acceptance criteria checks
6. **Project Integrator** — move validated files into Laravel project
7. **Documentation** — README, Wiki, tutorials
8. **Front-end Preview UI** — static + interactive preview built with HTML / CSS / JS
9. **Agent Backend Services** — API endpoints, orchestration, queueing, logs

3.3 Development Approach & Lifecycle

- **Method:** Agile — 1 week / sprint, iterative increments, weekly demo
- **Tools / Tech Stack:**
    - Front-end: HTML5, CSS3 (Tailwind or plain), Vanilla JS or small framework (Alpine.js / HTMX) for interactivity
    - Backend agents/orchestrator: Python (existing agents), FastAPI or Flask for HTTP endpoints, Redis/RQ or Celery for queues
    - Integration: Laravel (target project), filesystem staging area, GitHub Actions for CI

    - AI: Mistral API (agent prompts & inference)
- **Testing:** Unit tests for agents, integration tests for file move & route updates, E2E for preview pipeline
- **CI/CD:** GitHub Actions for linting, tests, and optional deployment of preview server

3.4 Timeline (sprint mapping) — total ≈ 14 weeks

- **Sprint 1–2:** Project setup, infra, repo templates, env, CI pipelines
- **Sprint 3–4:** Prompt Expander & Planner agent (backend)
- **Sprint 5–6:** Draft Generator core + simple HTML preview (front-end scaffold)
- **Sprint 7–8:** Layout & Component Generator (agent backend) + preview UI enhancements (HTML/CSS/JS)
- **Sprint 9–10:** Route Agent & Project Integrator (integration with Laravel)
- **Sprint 11:** Validator Agent (syntax, structure, acceptance criteria)
- **Sprint 12:** Agent backend polishing — queueing, retries, logging, security
- **Sprint 13:** Documentation, examples, repo templates, demo app
- **Sprint 14:** Final testing, bugfix, release / handover

Note: Sprints above are 1-week units; you can split or merge according to team velocity.

3.5 Resource Planning (updated)

- **Team**
    - 1 Project Manager + Backend Engineer (agent API, orchestration)
    - 1 Frontend Engineer (HTML/CSS/JS)
- **Tools:** Python, FastAPI/Flask, Redis/Celery, Laravel, GitHub, Mistral API, Docker (optional)

3.6 Front-end Plan (HTML / CSS / JS) — deliverables & responsibilities

- **Goal:** provide interactive preview where user can see generated HTML drafts and trigger generation into Laravel.
- **Deliverables**
    - Lightweight preview page (served from output/ or small preview server)
    - Components: preview pane, prompt input, agent status panel, accept/modify controls, download / integrate buttons
    - Accessibility basics and responsive layout
- **Tech details**

    - Static markup generated by Draft Generator (server) and rendered in preview pane
    - Interactions via vanilla JS or Alpine.js: accept changes, open code editor modal, copy to clipboard, send approve-to-integrate request
    - CSS: Tailwind (if allowed) or scoped CSS; keep styling minimal, focus on clarity
- **Testing**
    - Cross-browser basic tests (Chrome, Firefox)
    - Visual inspection acceptance per sprint
- **Security**
    - Sanitize previewed HTML (avoid executing untrusted scripts) — render in sanitized iframe or text renderer

3.7 Agent Backend Plan — architecture & responsibilities

- **Responsibilities**
    - Orchestrate agents in sequence: prompt expander → draft → planner → generator → validator → integrator
    - Expose HTTP endpoints for front-end to request draft, check status, and trigger integration
    - Manage asynchronous jobs (queue), retries, and error handling
- **Architecture**
    - **API Layer:** FastAPI/Flask — endpoints: /generate-draft, /status/{job}, /approve/{job}, /integrate/{job}
    - **Worker Layer:** Python workers (Celery/Redis or RQ) to run agents (calls to Mistral + codegen)
    - **Storage:** staging filesystem (output/), DB-lite (sqlite/postgres) for job metadata and logs
    - **Logging & Monitoring:** structured logs, simple dashboard for job queue / failures
    - **Security:** API keys for access, rate-limits, input validation, secrets management via .env
- **Integration points**
    - Mistral AI API (agent inference)
    - Laravel project (move files via staging or PR creation)
    - GitHub (optional): create branch / PR when integrating into my-laravel/

3.8 Risk & Mitigation (extended)

Risk	Impact	Mitigation
Untrusted HTML executes in preview	High	Render in sanitized iframe/escape scripts; do not execute user-generated JS
Agent output incorrect (invalid Blade)	High	Validator agent, linting, unit tests, sample acceptance criteria
Queue/backlog overload	Medium	Rate-limit requests, scale workers, backpressure to UI
Mistral downtime	Medium	Exponential retry, fallback cached templates, graceful degrade message
Integrating to user Laravel breaks project	High	Create branch or staging folder instead of directly overwriting; provide rollback

3.9 Stakeholder Engagement & Communication

- Weekly sprint review + demo of preview UI and a sample generated Blade file
- Issues & backlog management on GitHub (issues per sprint)
- Progress via GitHub Project Board; monthly stakeholder report

3.10 Success Criteria (updated)

- ≥ 90% successful generation of syntactically valid Blade components (per validator)
- Preview UI allows users to inspect and approve generated templates reliably
- Integration process updates Laravel project without breaking (use PR/staging)
- Internal developer team adopts tool for at least 3 example pages within first month
- Complete documentation & example repository available

Project Charter & Scope

Nama Project: MCP Agents Laravel UI Generator
Sponsor: Internal Development Team
Project Manager: Fikri Armia Fahmi (2023071018)
Frontend Developer: Nadia (2024071004)
Tanggal Pembuatan: 18 September 2025
Status: Approved

1. Executive Summary

Project MCP Agents Laravel UI Generator adalah inisiatif untuk mengembangkan sistem AI multi-agent yang dapat mengotomatisasi pembuatan komponen UI Laravel dari natural language prompts. System ini dirancang untuk mempercepat development workflow, mengurangi repetitive tasks, dan meningkatkan produktivitas developer, khususnya mahasiswa dan developer pemula.

2. Project Purpose & Justification

2.1 Business Case

Pengembangan komponen UI di Laravel memerlukan pemahaman mendalam tentang Blade templating, routing, dan struktur project. Proses manual ini memakan waktu 2-4 jam per komponen untuk developer pemula, yang dapat dikurangi hingga 80% dengan otomatisasi AI.

2.2 Problem Statement

Terdapat beberapa permasalahan utama yang dihadapi oleh developer dalam pengembangan Laravel UI, yaitu:

- Developer pemula menghabiskan terlalu banyak waktu untuk boilerplate code
- Inkonsistensi dalam struktur dan naming conventions antar komponen
- Learning curve yang tinggi untuk Laravel best practices
- Repetitive tasks mengurangi waktu untuk fokus pada business logic

2.3 Opportunity

Dengan memanfaatkan AI multi-agent system dan Model Context Protocol (MCP), project ini memberikan peluang untuk mencapai beberapa keuntungan strategis sebagai berikut:

- Mengotomatisasi 70-80% proses pembuatan UI components
- Standardisasi code quality dan structure
- Mempercepat onboarding developer baru
- Meningkatkan produktivitas tim development

3. Project Objectives

3.1 Primary Objectives

Project ini memiliki tiga tujuan utama yang harus dicapai untuk memastikan kesuksesan implementasi, yaitu:

1. **Otomatisasi Pembuatan UI**
    - Generate Blade components dari natural language prompts
    - Otomatis membuat routes, controllers, dan validation rules
    - Success metric: 90% generated code valid tanpa manual editing
2. **Interactive Preview System**
    - Real-time preview dari generated components
    - User dapat review dan modify sebelum integration
    - Success metric: 95% user satisfaction dengan preview accuracy
3. **Quality Assurance**
    - Automated validation untuk syntax, structure, dan best practices
    - Linting dan acceptance criteria checking
    - Success metric: Zero critical bugs dalam generated code

3.2 Secondary Objectives

Selain tujuan utama, project ini juga memiliki beberapa tujuan sekunder yang mendukung adopsi dan keberlanjutan sistem, meliputi:

- Dokumentasi lengkap dan user-friendly
- Example repository dengan 10+ use cases
- Community adoption minimal 50 users dalam 3 bulan pertama

4. Project Scope

4.1 In Scope

4.1.1 Core Features

System ini akan mengimplementasikan AI Agent System yang terdiri dari 10 agents dengan fungsi spesifik, yaitu:

1. Prompt Processing Agent - expand dan plan user prompts
2. Draft Generator - generate HTML preview
3. Layout Generator - create Blade layouts
4. Component Generator - create reusable Blade components
5. Route Generator - update routes/web.php
6. Controller Generator - create controller methods
7. Validation Generator - create form validation rules
8. Validator Agent - check syntax dan structure
9. Project Integrator - integrate files ke Laravel project
10. Documentation Generator - generate inline documentation

4.1.2 User Interface

Untuk memudahkan interaksi user dengan system, akan dikembangkan web-based interface dengan fitur-fitur berikut:

- Web-based preview interface (HTML/CSS/JS)
- Prompt input dengan syntax highlighting
- Interactive preview pane dengan live reload
- Agent status monitoring dashboard
- Accept/modify/reject controls
- Download generated files
- One-click integration ke Laravel project

4.1.3 Backend Services

Backend system akan menyediakan infrastruktur yang robust untuk menjalankan agents dan mengelola workflow, mencakup:

- RESTful API dengan FastAPI/Flask
- Asynchronous job queue (Celery/Redis)
- Job status tracking dan logging
- Error handling dan retry mechanism
- Rate limiting dan security controls

4.1.4 Integration

System akan terintegrasi dengan tools dan platform yang umum digunakan dalam Laravel development, meliputi:

- Laravel 8+ compatibility
- GitHub integration untuk PR creation
- Staging area untuk safe file operations
- Rollback mechanism

4.1.5 Documentation

Untuk mendukung adopsi dan penggunaan system, akan disediakan dokumentasi lengkap yang mencakup:

- User guide dan tutorials
- API documentation
- Architecture documentation
- Example repository dengan sample projects
- Video tutorials (optional)

4.2 Out of Scope

4.2.1 Tidak Termasuk dalam Project Ini

Untuk menjaga fokus dan memastikan delivery tepat waktu, beberapa fitur berikut tidak akan diimplementasikan dalam versi 1.0:

- Database migration generation (future enhancement)

- Model relationship generation (future enhancement)
- Testing code generation (future enhancement)
- Deployment automation (future enhancement)
- Multi-framework support (hanya Laravel untuk v1.0)
- Mobile app interface (web-only untuk v1.0)
- Real-time collaboration features
- Version control integration selain GitHub
- Custom AI model training (menggunakan existing Mistral API)

4.2.2 Assumptions

Project ini dibangun dengan beberapa asumsi dasar mengenai environment dan kemampuan users, yaitu:

- Users memiliki Laravel project yang sudah ter-setup
- Users memiliki basic knowledge tentang Laravel
- Internet connection tersedia untuk AI API calls
- Python 3.7+ dan Laravel 8+ sudah terinstall
- Users memiliki Mistral API access

4.2.3 Constraints

Terdapat beberapa batasan yang harus dipertimbangkan dalam pelaksanaan project ini, meliputi:

- Budget: Open-source project dengan minimal infrastructure cost
- Timeline: 14 minggu development (3.5 bulan)
- Resources: 2 developers (1 backend + PM, 1 frontend)
- Technology: Harus menggunakan Mistral API untuk AI inference
- Compatibility: Support Laravel 8+ only

5. Deliverables

5.1 Technical Deliverables

#	Deliverable	Description	Due Date
1	AI Agent Modules	10 Python modules untuk agent system	Week 11
2	Backend API	FastAPI/Flask REST API	Week 12
3	Frontend Preview UI	Web interface untuk preview dan control	Week 8
4	Integration System	File mover dan Laravel integrator	Week 10
5	Validation System	Syntax checker dan linter	Week 11
6	Queue System	Async job processing dengan Celery/Redis	Week 12

5.2 Documentation Deliverables

Dokumentasi yang akan diserahkan untuk mendukung penggunaan dan maintenance system adalah sebagai berikut:

#	Deliverable	Description	Due Date
1	User Guide	Step-by-step tutorial untuk end users	Week 13
2	API Documentation	REST API endpoints documentation	Week 13
3	Architecture Doc	System design dan architecture	Week 13
4	Example Repository	Sample Laravel projects dengan generated UI	Week 13
5	README	Installation dan quick start guide	Week 13

5.3 Project Management Deliverables

Untuk keperluan project management dan knowledge transfer, akan disediakan dokumen-dokumen berikut:

- Weekly sprint reports
- Monthly stakeholder presentations
- Final project handover document
- Lessons learned document

6. Stakeholders

6.1 Internal Stakeholders

Stakeholder internal yang terlibat langsung dalam pelaksanaan project ini adalah sebagai berikut:

Role	Name	Responsibility	Involvement
Project Sponsor	Internal Dev Team	Funding dan strategic direction	High
Project Manager	Fikri Armia Fahmi	Overall project management, backend dev	High
Frontend Developer	Nadia	UI/UX development	High
QA Team	TBD	Testing dan quality assurance	Medium

6.2 External Stakeholders

Stakeholder eksternal yang memiliki kepentingan terhadap kesuksesan project ini meliputi:

Stakeholder	Interest	Influence	Engagement Strategy
Laravel Developers	End users dari tool	High	Beta testing, feedback sessions
Software Houses	Potential adopters	Medium	Demo sessions, case studies
Open Source Community	Contributors	Medium	GitHub issues, documentation
Students	Learning dan usage	Low	Tutorials, educational content

7. Success Criteria

7.1 Technical Success Metrics

Kesuksesan teknis project akan diukur berdasarkan beberapa metrik kinerja berikut:

- ☑ 90% generated Blade components syntactically valid
- ☑ Preview UI load time < 2 seconds
- ☑ API response time < 500ms untuk simple requests
- ☑ Zero data loss during integration process
- ☑ Support untuk 10+ common UI patterns (forms, tables, cards, etc.)

7.2 Business Success Metrics

Dari perspektif bisnis, kesuksesan project akan dievaluasi melalui indikator-indikator berikut:

- ☑ 50+ active users dalam 3 bulan pertama
- ☑ 80% user satisfaction score
- ☑ 3+ case studies dari real projects
- ☑ 100+ GitHub stars dalam 6 bulan
- ☑ Average time saving 60%+ per component

7.3 Quality Metrics

Kualitas deliverables akan dipastikan melalui pencapaian metrik-metrik berikut:

- ☑ Code coverage > 70% untuk critical modules
- ☑ Zero critical security vulnerabilities
- ☑ Documentation completeness > 90%
- ☑ Bug resolution time < 48 hours untuk critical issues

8. Project Timeline & Milestones

8.1 High-Level Timeline

Total Duration: 14 minggu (18 September 2025 - 25 Desember 2025)

8.2 Major Milestones

Milestone	Description	Target Date	Deliverables
M1: Project Kickoff	Setup infrastructure dan repo	Week 2	Repo, CI/CD, env setup
M2: Core Agents Ready	Prompt processor dan draft generator	Week 6	2 working agents
M3: Preview UI Alpha	Basic preview interface	Week 8	Functional preview UI
M4: Integration Complete	Full agent pipeline working	Week 10	End-to-end workflow
M5: Validation System	Quality checks implemented	Week 11	Validator agent
M6: Beta Release	Feature complete, testing phase	Week 12	Beta version

Milestone	Description	Target Date	Deliverables
M7: Documentation	All docs completed	Week 13	Complete documentation
M8: Production Release	v1.0 launch	Week 14	Production-ready system

8.3 Sprint Breakdown

Development akan dilakukan secara iteratif dengan pembagian sprint sebagai berikut:

- **Sprint 1-2 (Week 1-2):** Infrastructure setup
- **Sprint 3-4 (Week 3-4):** Prompt processing agents
- **Sprint 5-6 (Week 5-6):** Draft generator dan preview scaffold
- **Sprint 7-8 (Week 7-8):** Layout/component generator + UI enhancements
- **Sprint 9-10 (Week 9-10):** Route agent dan integrator
- **Sprint 11 (Week 11):** Validator agent
- **Sprint 12 (Week 12):** Backend polishing
- **Sprint 13 (Week 13):** Documentation
- **Sprint 14 (Week 14):** Final testing dan release

9. Budget & Resources

9.1 Human Resources

Role	Allocation	Duration	Cost (Estimated)
Project Manager + Backend Dev	100%	14 weeks	Internal resource
Frontend Developer	100%	14 weeks	Internal resource
QA Support	25%	4 weeks	Internal resource

9.2 Infrastructure Costs

Item	Monthly Cost	Duration	Total
Mistral API Credits	$50	4 months	$200
Cloud Server (optional)	$20	4 months	$80
GitHub (free tier)	$0	-	$0
Domain (optional)	$12/year	1 year	$12
Total Estimated Cost			**$292**

9.3 Tools & Software

Tools dan software yang akan digunakan dalam development project ini meliputi:

- Python 3.7+ (free)
- Laravel 8+ (free)
- FastAPI/Flask (free)
- Redis/Celery (free)
- GitHub (free tier)
- VS Code / IDE (free)
- Mistral API (paid)

10. Risk Management

10.1 High Priority Risks

Risiko-risiko dengan prioritas tinggi yang telah diidentifikasi dan memerlukan perhatian khusus adalah:

Risk	Probability	Impact	Mitigation Strategy
Mistral API downtime	Medium	High	Implement retry logic, cache templates, graceful degradation
Generated code breaks Laravel project	Medium	High	Staging area, branch creation, rollback mechanism
Scope creep	High	Medium	Strict scope management, change control process
Resource unavailability	Low	High	Cross-training, documentation, backup resources

10.2 Medium Priority Risks

Risiko-risiko dengan prioritas menengah yang perlu dimonitor secara berkala meliputi:

Risk	Probability	Impact	Mitigation Strategy
Security vulnerabilities	Medium	Medium	Code review, security scanning, input sanitization
Performance issues	Medium	Medium	Load testing, optimization, caching
User adoption low	Medium	Medium	Marketing, tutorials, community engagement
Integration complexity	High	Low	Incremental integration, extensive testing

11. Communication Plan

11.1 Internal Communication

Komunikasi internal tim akan dilakukan melalui beberapa forum regular berikut:

- **Daily Standups:** 15 menit setiap pagi (async via chat)
- **Weekly Sprint Review:** Setiap Jumat, 1 jam
- **Sprint Planning:** Setiap Senin, 1 jam
- **Monthly Stakeholder Update:** Presentasi progress dan demo

11.2 External Communication

Untuk komunikasi dengan stakeholder eksternal dan community, akan digunakan channel-channel berikut:

- **GitHub Issues:** Public issue tracking
- **Documentation Site:** User guides dan tutorials
- **Blog Posts:** Monthly updates tentang progress
- **Social Media:** Announcements di Twitter/LinkedIn (optional)

11.3 Reporting

Pelaporan progress dan status project akan dilakukan secara terstruktur melalui:

- Weekly sprint reports (internal)
- Monthly progress reports (stakeholders)
- Incident reports (as needed)
- Final project report (end of project)

12. Quality Management

12.1 Quality Standards

Untuk memastikan kualitas deliverables, project ini akan mengikuti standar-standar berikut:

- Code harus follow PEP 8 (Python) dan PSR-12 (PHP)
- All public APIs harus documented
- Critical functions harus memiliki unit tests
- Security best practices harus diikuti
- Accessibility standards (WCAG 2.1 Level A minimum)

12.2 Quality Assurance Process

Proses quality assurance akan dilakukan melalui beberapa tahapan verifikasi sebagai berikut:

1. **Code Review:** Semua code changes harus di-review
2. **Automated Testing:** CI/CD pipeline dengan automated tests
3. **Manual Testing:** QA testing untuk critical features
4. **Security Scanning:** Automated security vulnerability scanning
5. **Performance Testing:** Load testing untuk API endpoints

12.3 Acceptance Criteria

Setiap deliverable harus memenuhi kriteria penerimaan berikut sebelum dianggap complete:

- All features harus memenuhi acceptance criteria di requirements
- Zero critical bugs dalam production
- Documentation completeness verified
- User acceptance testing passed
- Performance benchmarks met

13. Change Management

13.1 Change Control Process

Setiap perubahan terhadap scope, timeline, atau resources harus melalui proses formal berikut:

1. **Change Request:** Submit via GitHub issue atau formal request
2. **Impact Analysis:** Assess impact pada scope, timeline, budget
3. **Approval:** PM dan sponsor review dan approve/reject
4. **Implementation:** Jika approved, masuk ke sprint backlog
5. **Communication:** Notify stakeholders tentang changes

13.2 Scope Change Approval Authority

Authority untuk menyetujui perubahan scope ditentukan berdasarkan magnitude perubahan sebagai berikut:

- **Minor changes** (< 1 day effort): PM approval
- **Medium changes** (1-3 days effort): PM + Sponsor approval
- **Major changes** (> 3 days effort): Full stakeholder review

14. Project Closure Criteria

14.1 Completion Criteria

Project akan dianggap selesai dan siap untuk closure apabila semua kriteria berikut telah terpenuhi:

- ☑ All deliverables completed dan accepted
- ☑ All acceptance criteria met
- ☑ Documentation complete dan published
- ☑ Production deployment successful
- ☑ User training completed (if applicable)
- ☑ Handover document signed off
- ☑ Lessons learned documented

14.2 Post-Project Activities

Setelah project closure, beberapa aktivitas berikut akan dilakukan untuk memastikan keberlanjutan system:

- Maintenance plan established
- Support process defined
- Enhancement backlog created
- Community management plan
- Success metrics monitoring setup

15. Approval & Sign-off

15.1 Document Approval

Role	Name	Signature	Date
Project Sponsor	Internal Dev Team	_____	_____
Project Manager	Fikri Armia Fahmi	_____	18/09/2025

Role	Name	Signature	Date
Frontend Developer	Nadia	_____	18/09/2025

15.2 Charter Authorization

This project charter authorizes the Project Manager to proceed with the project and allocate resources as outlined in this document.

Approved by:
Sponsor: _____
Date: _____

16. Appendices

Appendix A: Glossary

Berikut adalah definisi istilah-istilah teknis yang digunakan dalam dokumen ini:

- **MCP:** Model Context Protocol
- **Blade:** Laravel's templating engine
- **Agent:** Autonomous AI component dengan specific task
- **FastAPI:** Modern Python web framework
- **Celery:** Distributed task queue
- **Redis:** In-memory data store

Appendix B: References

Referensi dan dokumentasi yang relevan untuk project ini meliputi:

- Laravel Documentation: https://laravel.com/docs
- Mistral API Documentation: https://docs.mistral.ai
- FastAPI Documentation: https://fastapi.tiangolo.com
- Project Repository: [TBD]

Appendix C: Contact Information

Informasi kontak untuk stakeholder utama project adalah sebagai berikut:

- **Project Manager:** Fikri Armia Fahmi - [email]
- **Frontend Developer:** Nadia - [email]
- **Sponsor Contact:** [email]

Document Version: 1.0
Last Updated: 18 September 2025
Next Review Date: 2 Oktober 2025

Work Breakdown Structure (WBS)

Project: MCP Agents Laravel UI Generator
Project Manager: Fikri Armia Fahmi (2023071018)
Frontend Developer: Nadia (2024071004)
Tanggal: 17 November 2025

WBS Table Format

Mari kita lihat bagaimana WBS diterapkan pada proyek MCP Agents Laravel UI Generator berbasis deliverable (hasil/output), bukan aktivitas semata. Setiap item WBS sebaiknya dinyatakan sebagai kata benda, seperti "Desain UI/UX" daripada "Mendesain". WBS berhenti di level work package yang dapat diasign ke satu penanggung jawab dan dapat diestimasi durasinya.

Level	Kode	Aktivitas/Deliverable	Deskripsi
1	1.0	MCP Agents Laravel UI Generator	Sistem otomasi pembuatan Laravel UI menggunakan AI multi-agent
2	1.1	Inisiasi Proyek	Definisi ruang lingkup & setup awal
3	1.1.1	Kick-off Meeting	Menyepakati tujuan proyek & stakeholder
3	1.1.2	Project Charter	Dokumen charter & scope definition
3	1.1.3	Stakeholder Engagement Plan	Identifikasi & rencana komunikasi stakeholder
3	1.1.4	Team Formation	Pembentukan tim & assignment role
2	1.2	Infrastructure Setup	Persiapan environment & tools development
3	1.2.1	Repository Structure	Setup GitHub repo & folder structure
3	1.2.2	Development Environment	Konfigurasi Python, Laravel, Docker
3	1.2.3	CI/CD Pipeline	GitHub Actions untuk automated testing

3	1.2.4	Project Planning Documents	WBS, schedule, risk management plan
2	1.3	AI Agent Development	Pengembangan 6 AI agents
3	1.3.1	Prompt Processing Agent	Parser & expander natural language prompts
4	1.3.1.1	NLP Parser Implementation	Implementasi natural language parser
4	1.3.1.2	Prompt Expansion Logic	Logic untuk memperluas user prompts
4	1.3.1.3	Intent Classification	Klasifikasi intent dari user input
4	1.3.1.4	Unit Tests	Testing & validasi prompt agent
3	1.3.2	Draft Generator Agent	Generator HTML preview & template
4	1.3.2.1	Template Generation Engine	Engine untuk generate HTML templates
4	1.3.2.2	Mistral API Integration	Integrasi dengan Mistral AI API
4	1.3.2.3	Preview Output Formatter	Format output untuk preview
4	1.3.2.4	Error Handling	Mekanisme error handling & fallback
3	1.3.3	Layout & Component Generator	Generator Blade files & components
4	1.3.3.1	Blade Template Engine	Integrasi Blade template engine
4	1.3.3.2	Component Structure Generator	Generator struktur component Laravel
4	1.3.3.3	Layout Hierarchy Builder	Builder untuk layout hierarchy
4	1.3.3.4	Reusable Component Library	Library component yang reusable
3	1.3.4	Route Generator Agent	Generator & updater routes Laravel

4	1.3.4.1	Route Definition Parser	Parser untuk route definitions
4	1.3.4.2	Web Routes Updater	Update routes/web.php otomatis
4	1.3.4.3	Controller Scaffolding	Generate controller scaffolding
4	1.3.4.4	Route Conflict Detection	Deteksi konflik route
3	1.3.5	Validator Agent	Validasi syntax, structure, & linting
4	1.3.5.1	Syntax Validation	Validasi syntax Blade & PHP
4	1.3.5.2	Structure Validation	Validasi struktur file & folder
4	1.3.5.3	Linting Integration	Integrasi PHPStan & Laravel Pint
4	1.3.5.4	Acceptance Criteria Checker	Checker untuk acceptance criteria
3	1.3.6	Project Integrator Agent	Integrasi file ke Laravel project
4	1.3.6.1	File Staging Mechanism	Mekanisme staging file sebelum integrasi
4	1.3.6.2	Laravel File Mover	Mover file ke Laravel project
4	1.3.6.3	Backup & Rollback System	Sistem backup & rollback
4	1.3.6.4	Git Branch/PR Creation	Otomasi create branch/PR
2	1.4	Backend Services	Development backend API & services
3	1.4.1	API Layer	REST API untuk orchestration
4	1.4.1.1	FastAPI/Flask Setup	Setup framework FastAPI/Flask
4	1.4.1.2	Endpoint Implementation	Implementasi endpoints (draft, status, approve, integrate)
4	1.4.1.3	Request Validation	Validasi & sanitasi request
4	1.4.1.4	API Documentation	Dokumentasi API (Swagger/OpenAPI)
3	1.4.2	Worker & Queue System	Sistem asynchronous job processing
4	1.4.2.1	Redis/Celery Setup	Setup Redis & Celery

4	1.4.2.2	Job Processing	Implementasi asynchronous job processing
4	1.4.2.3	Retry Mechanism	Mekanisme retry & error handling
4	1.4.2.4	Job Status Tracking	Tracking status job
3	1.4.3	Storage & Database	Sistem storage & database
4	1.4.3.1	Filesystem Structure	Struktur staging filesystem
4	1.4.3.2	Database Setup	Setup SQLite/PostgreSQL
4	1.4.3.3	Job Metadata Schema	Schema untuk job metadata
4	1.4.3.4	Log Storage	Storage & retrieval logs
3	1.4.4	Security Implementation	Implementasi security backend
4	1.4.4.1	API Key Authentication	Sistem autentikasi API key
4	1.4.4.2	Rate Limiting	Implementasi rate limiting
4	1.4.4.3	Input Validation	Validasi & sanitasi input
4	1.4.4.4	Secrets Management	Management secrets via .env
2	1.5	Frontend Development	Development preview UI
3	1.5.1	Preview UI Core	Core UI untuk preview
4	1.5.1.1	HTML/CSS Structure	Struktur HTML/CSS dasar
4	1.5.1.2	Responsive Layout	Design responsive layout
4	1.5.1.3	Tailwind CSS Integration	Integrasi Tailwind CSS
4	1.5.1.4	Accessibility Implementation	Implementasi WCAG accessibility
3	1.5.2	Interactive Components	Komponen interaktif UI
4	1.5.2.1	Prompt Input Interface	Interface untuk input prompt
4	1.5.2.2	Preview Pane Renderer	Renderer untuk preview pane
4	1.5.2.3	Agent Status Panel	Panel status agent

4	1.5.2.4	Accept/Modify Controls	Controls untuk accept/modify
3	1.5.3	JavaScript Functionality	Fungsionalitas JavaScript
4	1.5.3.1	JS Framework Setup	Setup Vanilla JS/Alpine.js
4	1.5.3.2	API Integration	Integrasi dengan backend API
4	1.5.3.3	Code Editor Modal	Modal code editor
4	1.5.3.4	Copy to Clipboard	Feature copy to clipboard
3	1.5.4	Frontend Security	Security & sanitization frontend
4	1.5.4.1	HTML Sanitization	Sanitasi HTML preview
4	1.5.4.2	Iframe Sandbox	Sandbox iframe untuk preview
4	1.5.4.3	XSS Prevention	Pencegahan XSS attacks
4	1.5.4.4	CSP Setup	Setup Content Security Policy
2	1.6	Testing & Quality Assurance	Pengujian sistem & QA
3	1.6.1	Unit Testing	Unit tests untuk semua komponen
4	1.6.1.1	Agent Unit Tests	Unit tests untuk agents (pytest)
4	1.6.1.2	API Endpoint Tests	Tests untuk API endpoints
4	1.6.1.3	Frontend Component Tests	Tests untuk frontend components
4	1.6.1.4	Test Coverage Report	Report coverage testing
3	1.6.2	Integration Testing	Integration tests
4	1.6.2.1	Agent Orchestration Tests	Tests orchestration agents
4	1.6.2.2	File Move & Route Tests	Tests file move & route updates
4	1.6.2.3	API-Frontend Integration	Tests integrasi API-Frontend
4	1.6.2.4	Laravel Integration Tests	Tests integrasi dengan Laravel
3	1.6.3	End-to-End Testing	E2E testing complete workflow

4	1.6.3.1	Workflow Testing	Testing complete workflow
4	1.6.3.2	Preview Pipeline Validation	Validasi preview pipeline
4	1.6.3.3	User Acceptance Testing	UAT dengan user
4	1.6.3.4	Performance Testing	Testing performance sistem
3	1.6.4	Quality Assurance	QA process
4	1.6.4.1	Code Review	Process code review
4	1.6.4.2	Security Audit	Audit security sistem
4	1.6.4.3	Cross-browser Testing	Testing cross-browser
4	1.6.4.4	Bug Tracking	Tracking & resolution bugs
2	1.7	Documentation	Dokumentasi lengkap sistem
3	1.7.1	Technical Documentation	Dokumentasi teknis
4	1.7.1.1	Architecture Documentation	Dokumentasi arsitektur sistem
4	1.7.1.2	API Documentation	Dokumentasi API lengkap
4	1.7.1.3	Agent Workflow Documentation	Dokumentasi workflow agents
4	1.7.1.4	Deployment Guide	Guide untuk deployment
3	1.7.2	User Documentation	Dokumentasi untuk user
4	1.7.2.1	README Creation	Pembuatan README.md
4	1.7.2.2	User Guide & Tutorials	Guide & tutorial untuk user
4	1.7.2.3	FAQ & Troubleshooting	FAQ & troubleshooting guide
4	1.7.2.4	Video Tutorials	Video tutorial (optional)
3	1.7.3	Code Documentation	Dokumentasi code
4	1.7.3.1	Inline Comments	Inline code comments

4	1.7.3.2	Python Docstrings	Docstrings untuk Python code
4	1.7.3.3	JSDoc	JSDoc untuk JavaScript
4	1.7.3.4	Example Code Snippets	Example code snippets
3	1.7.4	Wiki & Knowledge Base	Wiki & knowledge base
4	1.7.4.1	GitHub Wiki Setup	Setup GitHub Wiki
4	1.7.4.2	Best Practices Guide	Guide best practices
4	1.7.4.3	Contributing Guidelines	Guidelines untuk contributing
4	1.7.4.4	Changelog	Maintenance changelog
2	1.8	Deployment & Release	Deployment ke production
3	1.8.1	Pre-deployment	Persiapan sebelum deployment
4	1.8.1.1	Final Testing	Final testing & validation
4	1.8.1.2	Performance Optimization	Optimasi performance
4	1.8.1.3	Security Hardening	Hardening security
4	1.8.1.4	Backup Strategy	Strategi backup
3	1.8.2	Deployment	Proses deployment
4	1.8.2.1	Production Environment	Setup production environment
4	1.8.2.2	Database Migration	Migration database
4	1.8.2.3	Service Deployment	Deployment services
4	1.8.2.4	Monitoring Setup	Setup monitoring
3	1.8.3	Release Management	Management release
4	1.8.3.1	Version Tagging	Tagging version (semantic versioning)
4	1.8.3.2	Release Notes	Pembuatan release notes
4	1.8.3.3	Package Distribution	Distribusi package (PyPI, npm)
4	1.8.3.4	Announcement	Announcement & communication

3	1.8.4	Post-deployment	Aktivitas post-deployment
4	1.8.4.1	Monitoring & Logging	Monitoring & logging sistem
4	1.8.4.2	Bug Hotfix Process	Process hotfix bugs
4	1.8.4.3	User Feedback Collection	Pengumpulan feedback user
4	1.8.4.4	Performance Metrics	Tracking performance metrics
2	1.9	Project Closure	Penutupan proyek
3	1.9.1	Final Deliverables	Deliverables akhir
4	1.9.1.1	Codebase Handover	Handover complete codebase
4	1.9.1.2	Documentation Package	Package dokumentasi lengkap
4	1.9.1.3	Demo Application	Aplikasi demo
4	1.9.1.4	Training Materials	Material training
3	1.9.2	Project Review	Review proyek
4	1.9.2.1	Lessons Learned	Dokumentasi lessons learned
4	1.9.2.2	Success Criteria Evaluation	Evaluasi success criteria
4	1.9.2.3	Stakeholder Feedback	Feedback dari stakeholder
4	1.9.2.4	Final Project Report	Report akhir proyek
3	1.9.3	Maintenance Plan	Rencana maintenance
4	1.9.3.1	Support Strategy	Strategi support & maintenance
4	1.9.3.2	Update Schedule	Schedule update & patch
4	1.9.3.3	Community Engagement	Rencana engagement community
4	1.9.3.4	Enhancement Roadmap	Roadmap enhancement future

Deliverables Summary

WBS Code	Deliverable	Owner	Duration
1.0	Project Management Documents	PM	2 weeks
2.0	6 AI Agents (fully tested)	Backend Engineer	8 weeks
3.0	Backend API & Services	Backend Engineer	4 weeks
4.0	Frontend Preview UI	Frontend Engineer	4 weeks
5.0	Test Suite & QA Reports	Both Engineers	2 weeks
6.0	Complete Documentation	Both Engineers	2 weeks
7.0	Deployed System	Both Engineers	1 week
8.0	Project Closure Package	PM	1 week

Total Estimated Duration: 14 weeks (98 days)

Network Diagram & Critical Path Analysis

MCP Agents Laravel UI Generator

Project: MCP Agents Laravel UI Generator
Project Manager: Fikri Armia Fahmi (2023071018)
Frontend Developer: Nadia (2024071004)
Tanggal: 18 September 2025

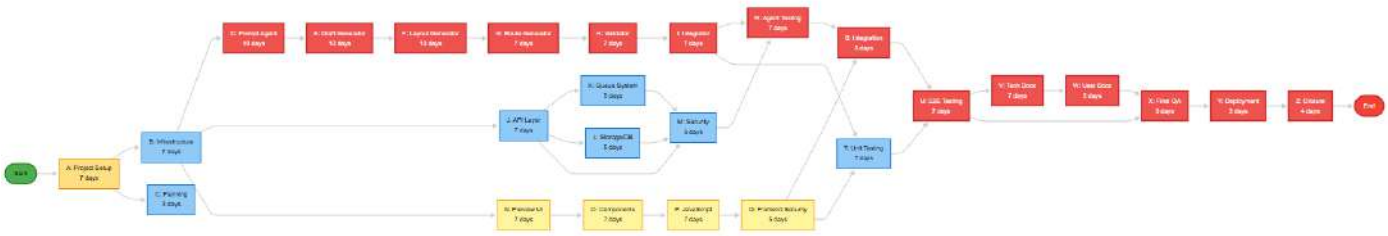
Activity List with Dependencies

ID	Activity Name	Duration (days)	Predecessors	Resources
A	Project Setup & Charter	7	-	PM
B	Infrastructure Setup	7	A	Backend Engineer
C	Project Planning (WBS, Schedule)	3	A	PM
D	Prompt Processing Agent	10	B	Backend Engineer
E	Draft Generator Agent	10	D	Backend Engineer
F	Layout & Component Generator	10	E	Backend Engineer
G	Route Generator Agent	7	F	Backend Engineer
H	Validator Agent	7	G	Backend Engineer

I	Project Integrator Agent	7	H	Backend Engineer
J	API Layer Development	7	B	Backend Engineer
K	Queue System Setup	5	J	Backend Engineer
L	Storage & Database Setup	5	J	Backend Engineer
M	Security Implementation	5	J, K, L	Backend Engineer
N	Preview UI Core	7	B	Frontend Engineer
O	Interactive Components	7	N	Frontend Engineer
P	JavaScript Functionality	7	O	Frontend Engineer
Q	Frontend Security	5	P	Frontend Engineer
R	Agent Integration Testing	7	I, M	Backend Engineer
S	Frontend-Backend Integration	5	Q, R	Both Engineers
T	Unit Testing	7	I, Q	Both Engineers
U	E2E Testing	7	S, T	Both Engineers

V	Technical Documentation	7	U	Both Engineers
W	User Documentation	5	V	Both Engineers
X	Final Testing & QA	5	U, W	Both Engineers
Y	Deployment	3	X	Both Engineers
Z	Project Closure	4	Y	PM

Network Diagram (AON - Activity on Node)



Legend:

- Green: Start/End milestones
- Red: Critical path activities
- Blue: Non-critical backend activities
- Yellow: Non-critical frontend activities

Legend:

- Green: Start/End milestones
- Pink: Critical path activities
- Yellow: Non-critical activities

Critical Path Analysis

Critical Path (Longest Path)

Path: Start → A → B → D → E → F → G → H → I → R → S → U → V → W → X → Y → Z → End

Critical Path Activities

1. **A:** Project Setup (7 days)
2. **B:** Infrastructure Setup (7 days)
3. **D:** Prompt Processing Agent (10 days)
4. **E:** Draft Generator Agent (10 days)
5. **F:** Layout & Component Generator (10 days)
6. **G:** Route Generator Agent (7 days)
7. **H:** Validator Agent (7 days)
8. **I:** Project Integrator Agent (7 days)
9. **R:** Agent Integration Testing (7 days)
10. **S:** Frontend-Backend Integration (5 days)
11. **U:** E2E Testing (7 days)
12. **V:** Technical Documentation (7 days)
13. **W:** User Documentation (5 days)
14. **X:** Final Testing & QA (5 days)
15. **Y:** Deployment (3 days)
16. **Z:** Project Closure (4 days)

Critical Path Duration

Total Duration: 7 + 7 + 10 + 10 + 10 + 7 + 7 + 7 + 7 + 5 + 7 + 7 + 5 + 5 + 3 + 4 = **109 days** (≈ 15.6 weeks)

Float/Slack Analysis

Activity	ES	EF	LS	LF	Total Float	Free Float	Critical?
A	0	7	0	7	0	0	☑ Yes

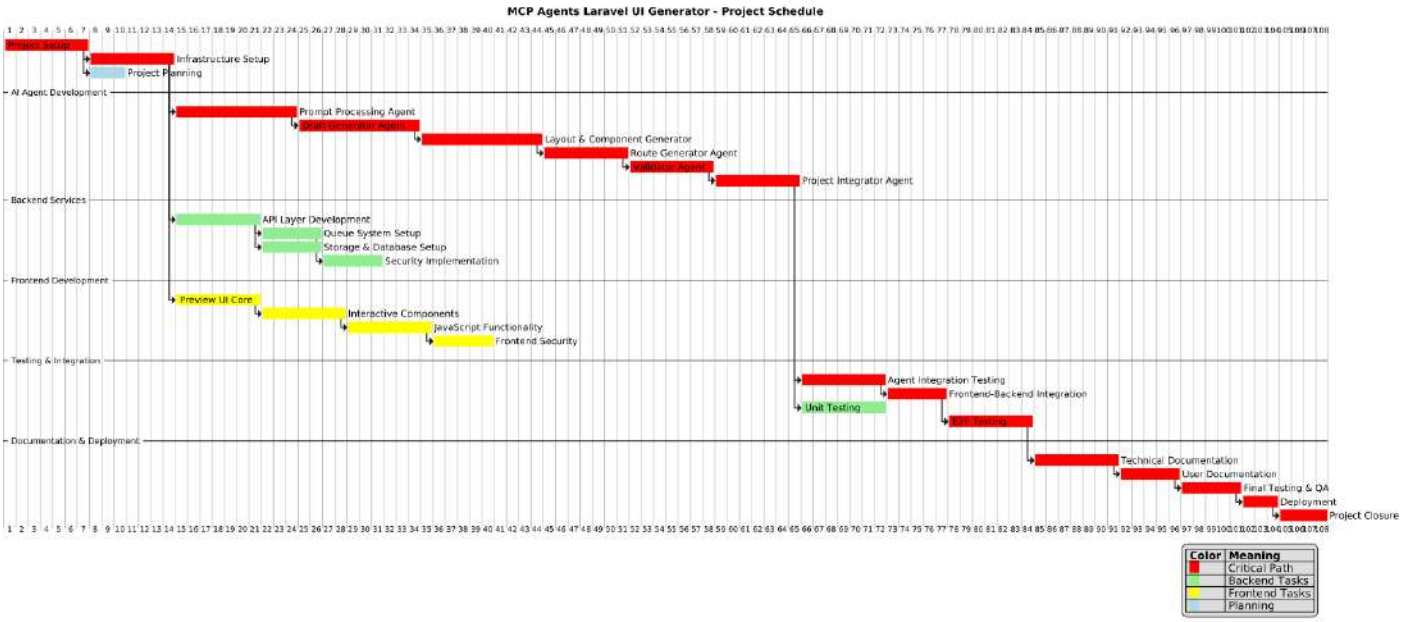
B	7	14	7	14	0	0	☑ Yes
C	7	10	11	14	4	4	❌ No
D	14	24	14	24	0	0	☑ Yes
E	24	34	24	34	0	0	☑ Yes
F	34	44	34	44	0	0	☑ Yes
G	44	51	44	51	0	0	☑ Yes
H	51	58	51	58	0	0	☑ Yes
I	58	65	58	65	0	0	☑ Yes
J	14	21	19	26	5	5	❌ No
K	21	26	26	31	5	5	❌ No
L	21	26	26	31	5	5	❌ No
M	26	31	31	36	5	5	❌ No
N	14	21	21	28	7	7	❌ No
O	21	28	28	35	7	7	❌ No
P	28	35	35	42	7	7	❌ No
Q	35	40	42	47	7	7	❌ No
R	65	72	65	72	0	0	☑ Yes
S	72	77	72	77	0	0	☑ Yes
T	65	72	70	77	5	5	❌ No
U	77	84	77	84	0	0	☑ Yes
V	84	91	84	91	0	0	☑ Yes
W	91	96	91	96	0	0	☑ Yes
X	96	101	96	101	0	0	☑ Yes
Y	101	104	101	104	0	0	☑ Yes

Z	104	108	104	108	0	0	☑ Yes

Legend:

- ES: Early Start
- EF: Early Finish
- LS: Late Start
- LF: Late Finish
- Total Float: LS - ES (or LF - EF)
- Free Float: ES(successor) - EF(current)

Gantt Chart Representation



Risk Impact on Schedule

Risk	Impact on Critical Path	Mitigation Strategy	Buffer Days
AI Agent development delays	High - directly on CP	Parallel development where possible, daily standups	+7 days

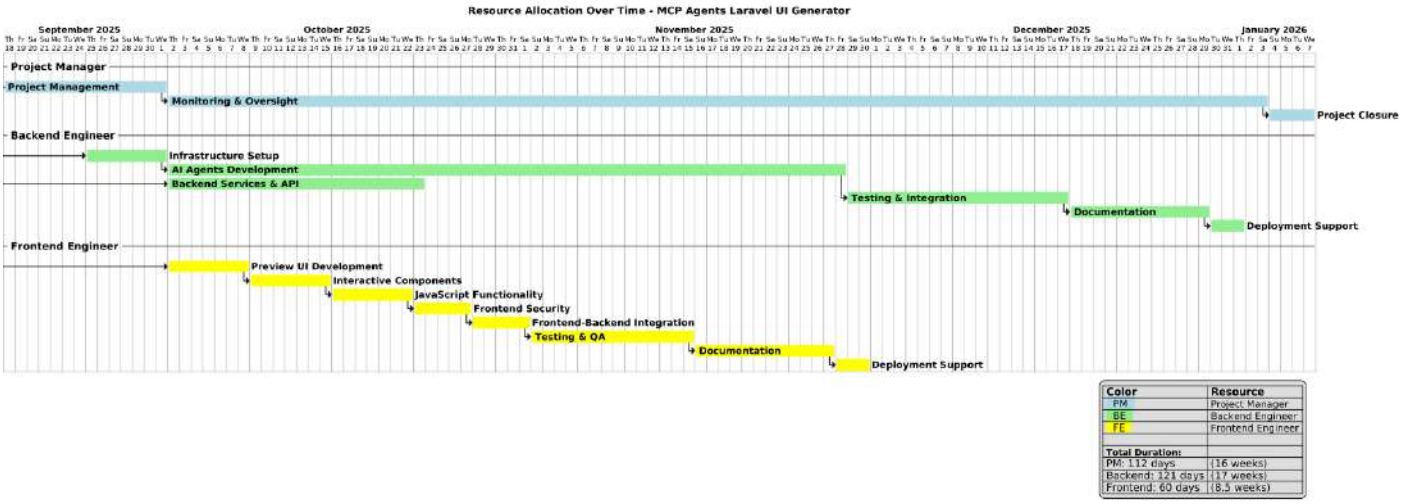
Mistral API downtime	Medium - affects testing	Implement retry logic, use cached responses	+3 days
Integration issues	High - affects testing phase	Early integration testing, continuous integration	+5 days
Resource unavailability	High - single point of failure	Cross-training, documentation	+5 days
Scope creep	High - extends all phases	Strict change control, MVP focus	+10 days

Recommended Project Buffer: 30 days (≈ 4 weeks)
Total Project Duration with Buffer: 139 days (≈ 20 weeks)

Milestone Schedule

Milestone	**Target Date**	**Dependencies**	**Deliverable**
M1: Project Kickoff	Day 0	-	Project Charter
M2: Infrastructure Ready	Day 14	A, B	Dev Environment
M3: Core Agents Complete	Day 58	D, E, F	3 AI Agents
M4: All Agents Complete	Day 65	G, H, I	6 AI Agents
M5: Backend Complete	Day 36	J, K, L, M	API & Services
M6: Frontend Complete	Day 40	N, O, P, Q	Preview UI
M7: Integration Complete	Day 77	R, S	Integrated System
M8: Testing Complete	Day 84	T, U	Test Reports
M9: Documentation Complete	Day 96	V, W	All Docs
M10: Production Ready	Day 104	X, Y	Deployed System
M11: Project Closure	Day 108	Z	Final Report

Resource Loading Chart



Optimization Recommendations

1. Fast-Tracking Opportunities

- **Frontend & Backend Development:** Can run in parallel (already planned)
- **Documentation:** Start technical documentation during development phase
- **Testing:** Begin unit testing as each agent is completed

2. Crashing Opportunities

- **Add resources to AI Agent development:** Hire additional backend engineer for agents D-I
- Potential time savings: 20-30 days
- Cost increase: 1 additional engineer salary
- **Parallel agent development:** Develop Route, Validator, and Integrator agents in parallel
- Potential time savings: 14 days
- Risk: Integration complexity increases

3. Schedule Compression

- **Current duration:** 109 days
- **With fast-tracking:** 95 days (-14 days)
- **With crashing:** 85 days (-24 days)
- **Aggressive compression:** 75 days (-34 days, high risk)

Conclusion

Critical Path Duration: 109 days (15.6 weeks)
Recommended Duration (with buffer): 139 days (20 weeks)
Critical Activities: 16 out of 26 activities (62%)
Schedule Risk: Medium-High (due to sequential agent development)

Key Recommendations:

1. Focus management attention on critical path activities
2. Monitor AI agent development closely (longest sequence)
3. Maintain 30-day buffer for unforeseen issues
4. Consider parallel agent development if resources allow
5. Start documentation early to reduce end-phase pressure

Gantt Chart Berbasis SCRUM

MCP Agents Laravel UI Generator

Proyek: MCP Agents Laravel UI Generator
Metode: SCRUM (Agile)
Durasi: 14 Minggu (18 September - 25 Desember 2025)
Sponsor: Internal Development Team
Project Manager: Fikri Armia Fahmi (2023071018)
Frontend Developer: Nadia (2024071004)

1. SCRUM Roles & Assignment

Role	Nama	Tanggung Jawab
Product Owner (PO)	Internal Development Team	Menentukan kebutuhan AI agents, mengelola backlog, prioritas fitur, review deliverables
Scrum Master (SM)	Fikri Armia Fahmi	Sprint planning, monitoring timeline, daily standup, menghilangkan blocker
Backend Developer + PM	Fikri Armia Fahmi	AI Agents development, API Layer, Queue System, Security, Project Management
Frontend Developer	Nadia	Preview UI, Interactive Components, JavaScript functionality, QA Testing

2. Timeline Proyek (14 Minggu)

Sprint	Periode	Fokus Utama	Deliverables
Sprint 1-2	Week 1-2 (18 Sep - 1 Okt 2025)	Project Setup & Infrastructure	Project Charter, Infrastructure, CI/CD, Repo Setup

Sprint 3-4	Week 3-4 (2 Okt - 15 Okt 2025)	Prompt Expander & Draft Agent	Prompt Expander Agent, Draft Agent
Sprint 5-6	Week 5-6 (16 Okt - 29 Okt 2025)	Prompt Planner & Page Architect	Prompt Planner Agent, Page Architect Agent
Sprint 7-8	Week 7-8 (30 Okt - 12 Nov 2025)	Generate Layout App & UI Generator	Generate Layout App Agent, UI Generator Agent + Preview UI
Sprint 9-10	Week 9-10 (13 Nov - 26 Nov 2025)	Route & Component Agents	Route Agent, Component Agent
Sprint 11	Week 11 (27 Nov - 3 Des 2025)	Validator & Move to Project Agents	Validator Agent, Move to Project Agent
Sprint 12	Week 12 (4 Des - 10 Des 2025)	Backend Polishing	Queueing, Retries, Logging, Security
Sprint 13	Week 13 (11 Des - 17 Des 2025)	Documentation	Documentation, Examples, Repo Templates, Demo App
Sprint 14	Week 14 (18 Des - 25 Des 2025)	Final Testing & Release	Final Testing, Bugfix, Release/Handover

3. Product Backlog

Epic 1: Project Initiation & Setup

- **PBI-001:** Project charter & scope definition (Priority: High)
- **PBI-002:** Infrastructure setup (GitHub, CI/CD) (Priority: High)

- **PBI-003:** Development environment configuration (Priority: High)
- **PBI-004:** Technology stack setup (Python, Laravel, Cerebras & Mistral API) (Priority: High)

Epic 2: AI Agent Development (10 Agents)

- **PBI-005:** Prompt Expander Agent - Expands and details user prompts (Priority: High)
- **PBI-006:** Draft Agent - Creates initial HTML draft for preview (Priority: High)
- **PBI-007:** Prompt Planner Agent - Plans the component structure (Priority: High)
- **PBI-008:** Page Architect Agent - Designs the page layout architecture (Priority: High)
- **PBI-009:** Generate Layout App Agent - Generates main layout template (Priority: High)
- **PBI-010:** UI Generator Agent - Creates Blade view files (Priority: High)
- **PBI-011:** Route Agent - Generates Laravel routes (Priority: High)
- **PBI-012:** Component Agent - Creates individual UI components (Priority: High)
- **PBI-013:** Validator Agent - Validates generated components (Priority: High)
- **PBI-014:** Move to Project Agent - Integrates with Laravel project (Priority: High)

Epic 3: Backend Services

- **PBI-015:** API Layer (FastAPI/Flask) (Priority: High)
- **PBI-016:** Queue System (Redis/Celery) (Priority: High)
- **PBI-017:** Storage & Database setup (Priority: High)
- **PBI-018:** Security implementation (Priority: High)

Epic 4: Frontend Development

- **PBI-019:** Preview UI Core (Priority: High)
- **PBI-020:** Interactive Components (Priority: High)
- **PBI-021:** JavaScript Functionality (Priority: High)
- **PBI-022:** Frontend Security (Priority: Medium)

Epic 5: Testing & Integration

- **PBI-023:** Unit Testing (Priority: High)
- **PBI-024:** Integration Testing (Priority: High)
- **PBI-025:** End-to-End Testing (Priority: High)
- **PBI-026:** Bug fixing (Priority: High)

Epic 6: Documentation & Deployment

- **PBI-027:** Technical Documentation (Priority: High)
- **PBI-028:** User Documentation (Priority: Medium)
- **PBI-029:** Deployment (Priority: High)
- **PBI-030:** Project Closure (Priority: High)

4. Sprint Backlog

Sprint 1-2 Backlog (Week 1-2: 18 Sep - 1 Okt 2025)

Task ID	Task Name	Assigned To	Duration	Dependencies
S1-T1	Project Charter & Scope	PM (Fikri)	2 hari	-
S1-T2	Stakeholder Identification	PM (Fikri)	1 hari	S1-T1
S1-T3	GitHub Repository Setup	Backend (Fikri)	1 hari	S1-T1
S1-T4	CI/CD Pipeline (GitHub Actions)	Backend (Fikri)	2 hari	S1-T3
S1-T5	Python Environment Setup	Backend (Fikri)	1 hari	S1-T3
S1-T6	Laravel Environment Setup	Backend (Fikri)	1 hari	S1-T3
S1-T7	Cerebras & Mistral API Integration Test	Backend (Fikri)	2 hari	S1-T5
S1-T8	Project Planning (WBS, Network, Gantt)	PM (Fikri)	3 hari	S1-T1
S1-T9	Sprint Review	All Team	1 hari	All tasks

Sprint 3-4 Backlog (Week 3-4: 2 Okt - 15 Okt 2025)

Task ID	Task Name	Assigned To	Duration	Dependencies
S3-T1	Prompt Expander Agent - NLP Parser	Backend (Fikri)	3 hari	Sprint 1-2 Done

S3-T2	Prompt Expander Agent - Expansion Logic	Backend (Fikri)	2 hari	S3-T1
S3-T3	Draft Agent - HTML Draft Generator	Backend (Fikri)	3 hari	S3-T2
S3-T4	Draft Agent - Browser Preview Integration	Backend (Fikri)	2 hari	S3-T3
S3-T5	Unit Tests (Prompt Expander & Draft)	Backend (Fikri)	2 hari	S3-T4
S3-T6	Sprint Review	All Team	1 hari	All tasks

Sprint 5-6 Backlog (Week 5-6: 16 Okt - 29 Okt 2025)

Task ID	Task Name	Assigned To	Duration	Dependencies
S5-T1	Prompt Planner Agent - Component Structure Planning	Backend (Fikri)	3 hari	Sprint 3-4 Done
S5-T2	Prompt Planner Agent - Cerebras & Mistral API Integration	Backend (Fikri)	2 hari	S5-T1
S5-T3	Page Architect Agent - Layout Architecture Design	Backend (Fikri)	3 hari	S5-T2
S5-T4	Page Architect Agent - Page Structure Generator	Backend (Fikri)	2 hari	S5-T3

S5-T5	Unit Tests (Prompt Planner & Page Architect)	Backend (Fikri)	2 hari	S5-T4
S5-T6	Sprint Review	All Team	1 hari	All tasks

Sprint 7-8 Backlog (Week 7-8: 30 Okt - 12 Nov 2025)

Task ID	Task Name	Assigned To	Duration	Dependencies
S7-T1	Generate Layout App Agent - Main Layout Template	Backend (Fikri)	3 hari	Sprint 5-6 Done
S7-T2	Generate Layout App Agent - Blade Template Engine	Backend (Fikri)	2 hari	S7-T1
S7-T3	UI Generator Agent - Blade View Files Generator	Backend (Fikri)	3 hari	S7-T2
S7-T4	UI Generator Agent - Component Integration	Backend (Fikri)	2 hari	S7-T3
S7-T5	Preview UI Core (HTML/CSS)	Frontend (Nadia)	5 hari	Sprint 5-6 Done
S7-T6	Unit Tests (Generate Layout App & UI Generator)	Backend (Fikri)	2 hari	S7-T4
S7-T7	Sprint Review	All Team	1 hari	All tasks

Sprint 9-10 Backlog (Week 9-10: 13 Nov - 26 Nov 2025)

Task ID	Task Name	Assigned To	Duration	Dependencies
S9-T1	Route Agent - Route Definition Parser	Backend (Fikri)	3 hari	Sprint 7-8 Done
S9-T2	Route Agent - Web Routes Generator	Backend (Fikri)	2 hari	S9-T1
S9-T3	Route Agent - Route Conflict Detection	Backend (Fikri)	2 hari	S9-T2
S9-T4	Component Agent - Individual UI Components	Backend (Fikri)	3 hari	S9-T3
S9-T5	Component Agent - Component Library Builder	Backend (Fikri)	2 hari	S9-T4
S9-T6	Unit Tests (Route & Component Agents)	Backend (Fikri)	2 hari	S9-T5
S9-T7	Sprint Review	All Team	1 hari	All tasks

Sprint 11 Backlog (Week 11: 27 Nov - 3 Des 2025)

Task ID	Task Name	Assigned To	Duration	Dependencies

S11-T1	Validator Agent - Syntax Validation	Backend (Fikri)	2 hari	Sprint 9-10 Done
S11-T2	Validator Agent - Structure Validation	Backend (Fikri)	2 hari	S11-T1
S11-T3	Validator Agent - Code Quality Checker	Backend (Fikri)	2 hari	S11-T2
S11-T4	Move to Project Agent - File Staging Mechanism	Backend (Fikri)	2 hari	S11-T3
S11-T5	Move to Project Agent - Laravel Integration	Backend (Fikri)	2 hari	S11-T4
S11-T6	Move to Project Agent - Backup & Rollback	Backend (Fikri)	1 hari	S11-T5
S11-T7	Unit Tests (Validator & Move to Project)	Backend (Fikri)	2 hari	S11-T6
S11-T8	Sprint Review	All Team	1 hari	All tasks

Sprint 12 Backlog (Week 12: 4 Des - 10 Des 2025)

Task ID	Task Name	Assigned To	Duration	Dependencies

S12-T1	FastAPI/Flask Setup	Backend (Fikri)	2 hari	Sprint 11 Done
S12-T2	API Endpoints Implementation	Backend (Fikri)	3 hari	S12-T1
S12-T3	Redis/Celery Queue Setup	Backend (Fikri)	2 hari	S12-T2
S12-T4	Job Processing & Retry Mechanism	Backend (Fikri)	2 hari	S12-T3
S12-T5	Security Implementation (API Key, Rate Limiting)	Backend (Fikri)	2 hari	S12-T4
S12-T6	Interactive Components	Frontend (Nadia)	5 hari	Sprint 11 Done
S12-T7	JavaScript Functionality	Frontend (Nadia)	3 hari	S12-T6
S12-T8	Sprint Review	All Team	1 hari	All tasks

Sprint 13 Backlog (Week 13: 11 Des - 17 Des 2025)

Task ID	Task Name	Assigned To	Duration	Dependencies

S13-T1	Frontend Security (HTML Sanitization, XSS Prevention)	Frontend (Nadia)	2 hari	Sprint 12 Done
S13-T2	Frontend-Backend Integration	All Dev	2 hari	S13-T1
S13-T3	Unit Testing (Agents, API, Frontend)	All Dev	2 hari	S13-T2
S13-T4	Integration Testing	All Dev	2 hari	S13-T3
S13-T5	E2E Testing	All Dev	2 hari	S13-T4
S13-T6	Bug Fixing	All Dev	2 hari	S13-T5
S13-T7	Technical Documentation (Architecture, API)	Backend (Fikri)	3 hari	S13-T6
S13-T8	User Documentation (README, Tutorials)	Frontend (Nadia)	3 hari	S13-T6
S13-T9	Example Repository & Demo App	All Dev	2 hari	S13-T7
S13-T10	Sprint Review	All Team	1 hari	All tasks

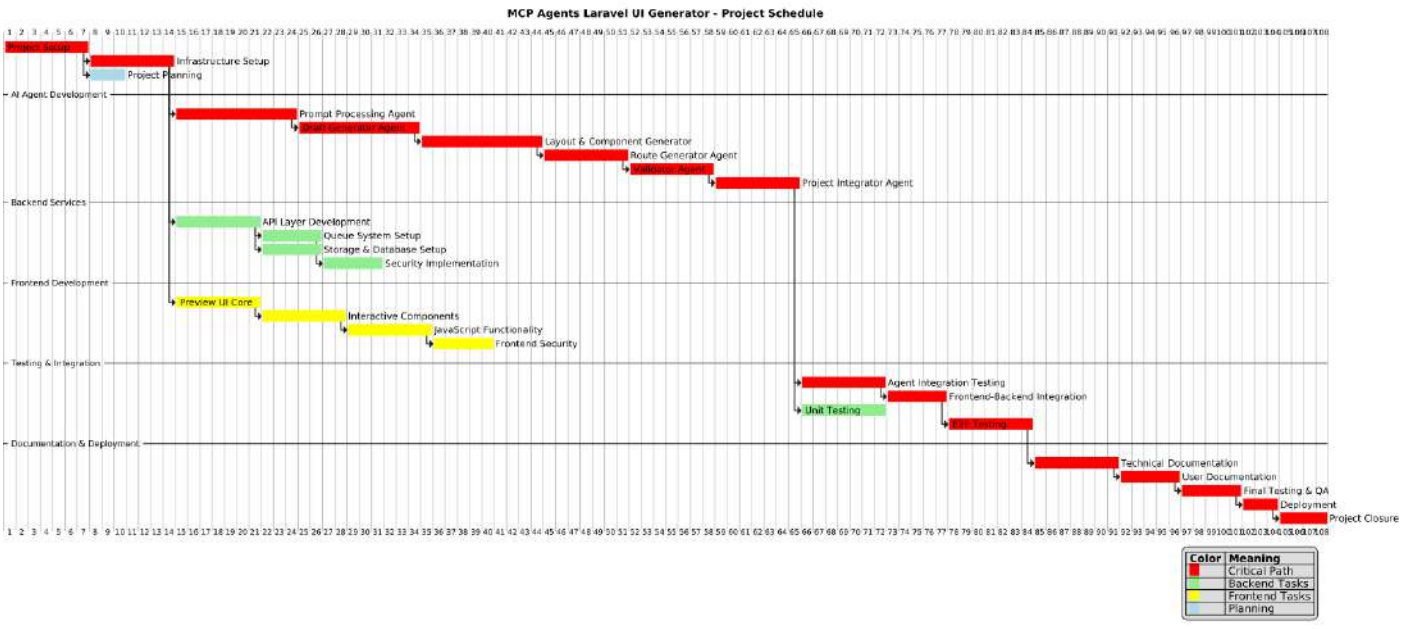
Sprint 14 Backlog (Week 14: 18 Des - 25 Des 2025)

Task ID	Task Name	Assigned To	Duration	Dependencies
S14-T1	Final Testing & Validation	All Dev	2 hari	Sprint 13 Done

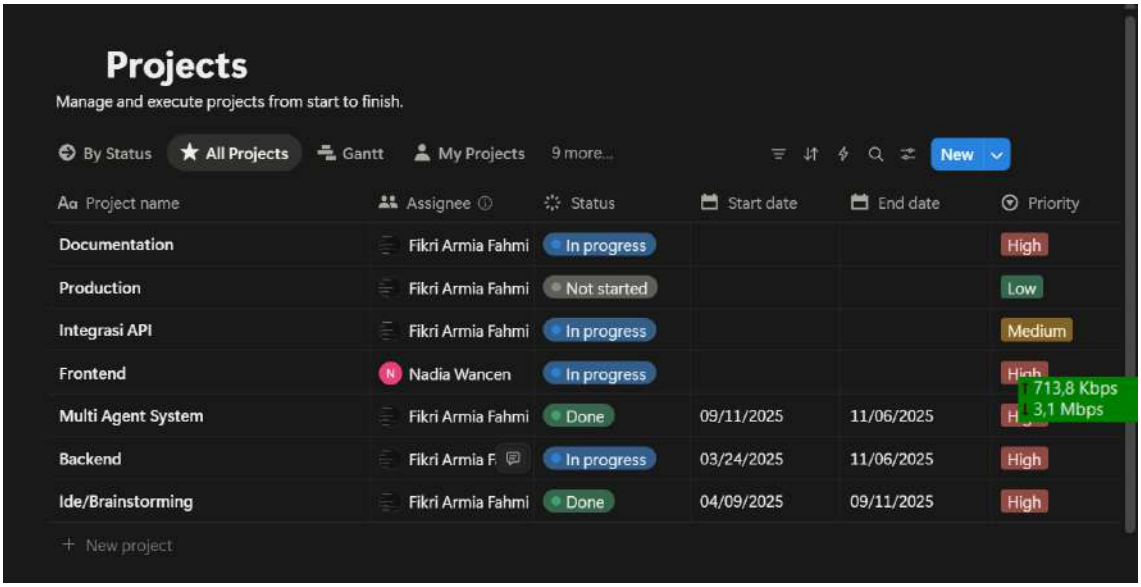
S14-T2	Performance Optimization	Backend (Fikri)	2 hari	S14-T1
S14-T3	Security Hardening	Backend (Fikri)	2 hari	S14-T2
S14-T4	Production Deployment	Backend (Fikri)	2 hari	S14-T3
S14-T5	Monitoring Setup	Backend (Fikri)	1 hari	S14-T4
S14-T6	Project Closure & Handover	PM (Fikri)	2 hari	S14-T5
S14-T7	Lessons Learned Documentation	PM (Fikri)	1 hari	S14-T6
S14-T8	Sprint Review & Retrospective	All Team	1 hari	All tasks

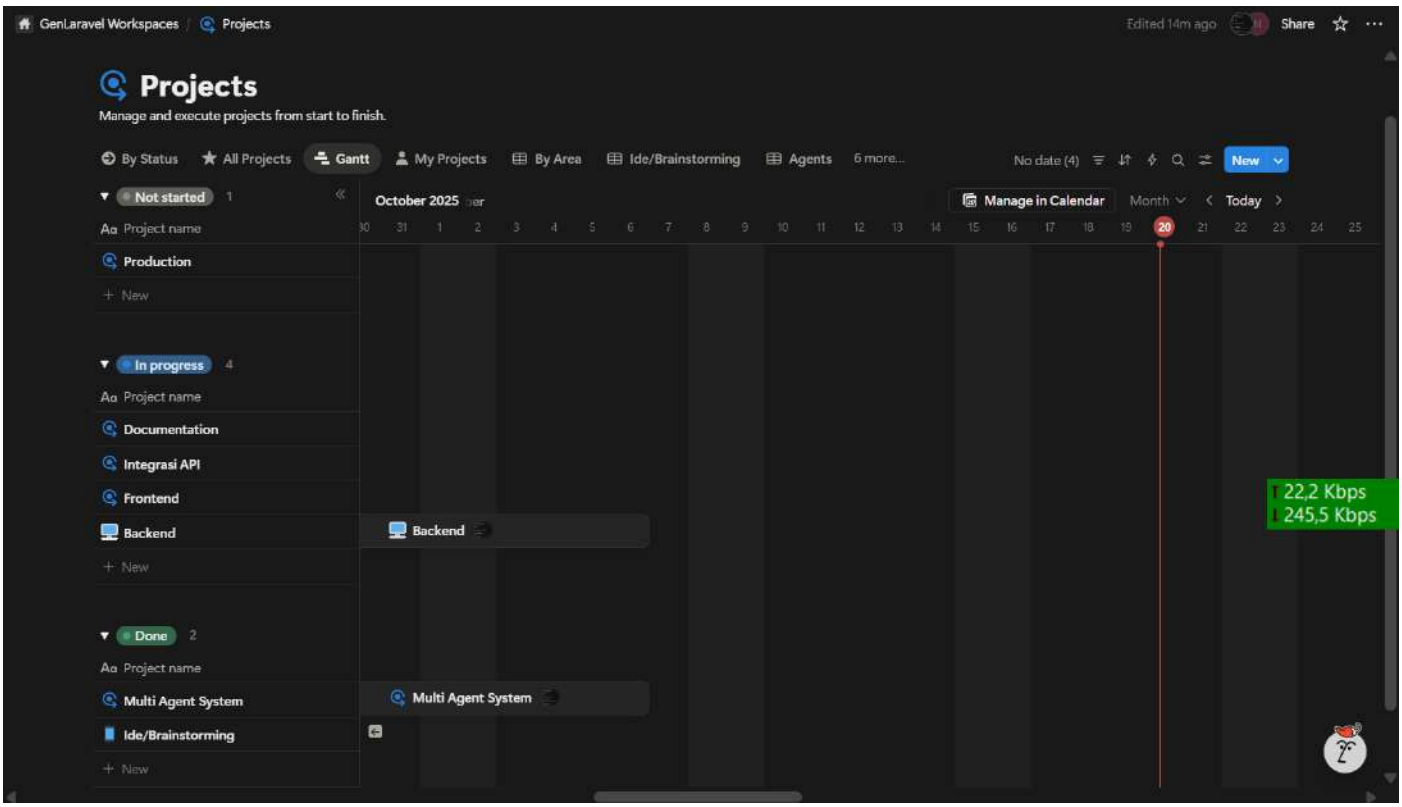
5. Gantt Chart Lengkap (4 Minggu)

Gantt Chart Overview

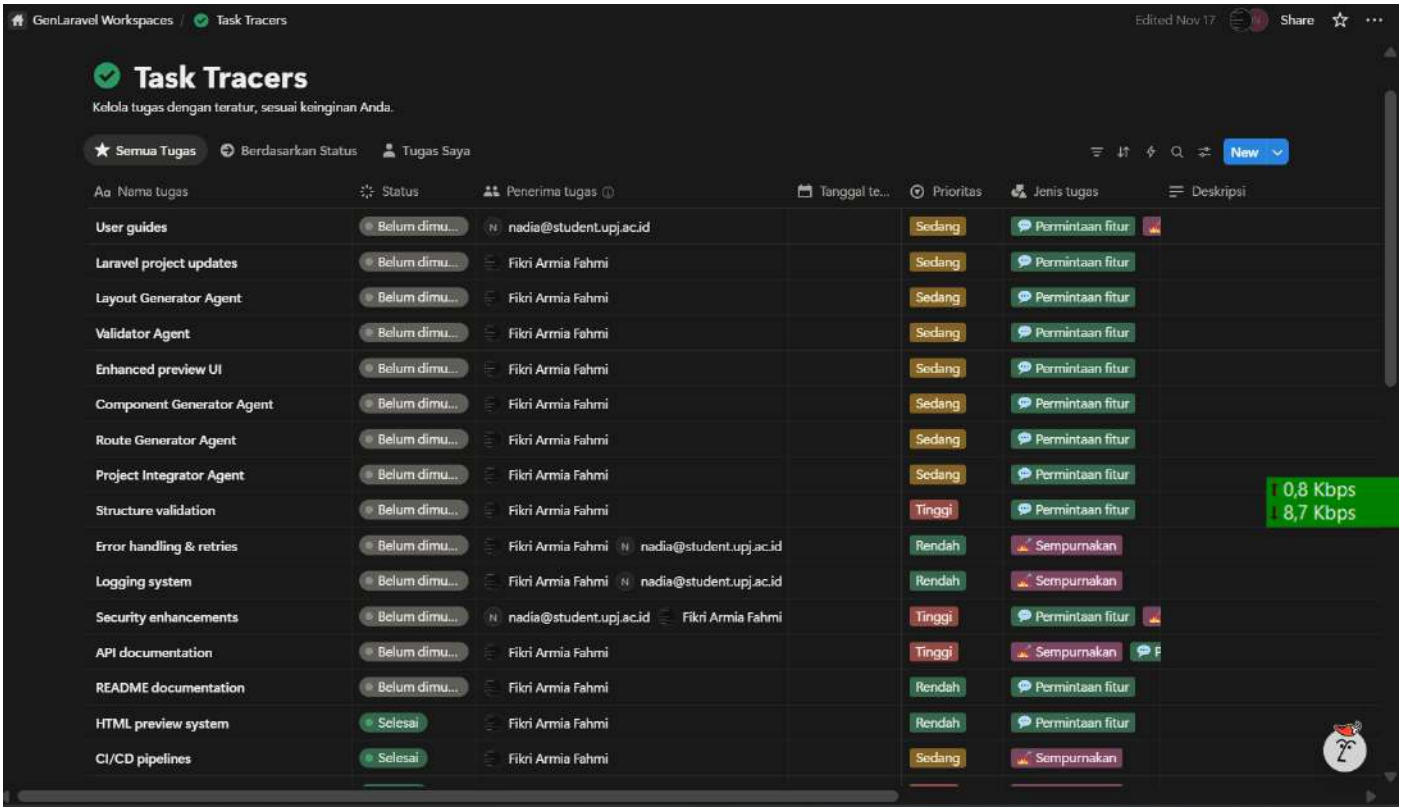


Gantt Chart Detail - Sprint View





Gantt Chart Detail - Timeline View



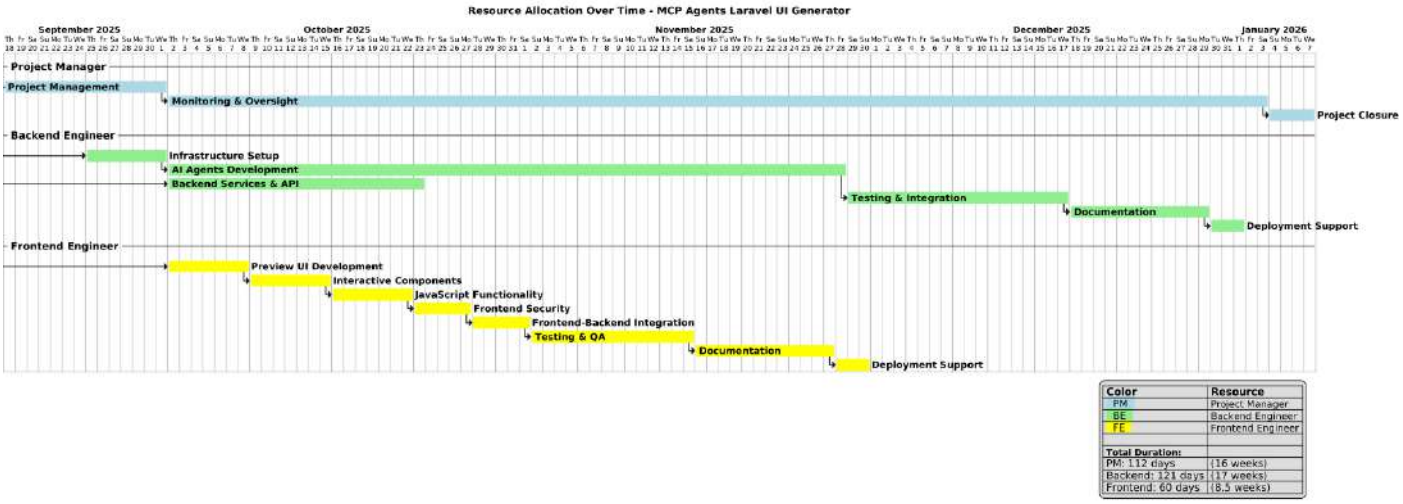
6. Dependencies Diagram

```
Project Charter → Infrastructure → AI Agents Development
                        ↓                ↓
                  API Layer → Backend Services
                        ↓                ↓
                  Frontend UI → Integration → Testing → Deploy → Closure
```

Dependency Rules:

1. ❌ Tidak bisa develop AI agents tanpa infrastructure
2. ❌ Tidak bisa frontend tanpa backend API ready
3. ❌ Tidak bisa testing tanpa implementasi selesai
4. ❌ Tidak bisa deploy tanpa testing passed
5. ✅ AI Agents bisa dikembangkan sequential (satu per satu)
6. ✅ Backend services bisa parallel dengan AI agents development

7. Sprint Planning & Ceremonies



Sprint 1-2 (Week 1-2: 18 Sep - 1 Okt 2025)

Sprint Goal: Setup project infrastructure dan environment development

Sprint Planning Meeting:

- Tanggal: 18 September 2025
- Durasi: 2 jam
- Output: Sprint 1-2 Backlog

Daily Standup:

- Setiap hari pukul 09:00 WIB
- Durasi: 15 menit
- Format: What did you do? What will you do? Any blockers?

Sprint Review:

- Tanggal: 1 Oktober 2025
- Durasi: 1 jam
- Deliverables: Project Charter, Infrastructure, CI/CD Pipeline, Repo Setup

Sprint Retrospective:

- Tanggal: 1 Oktober 2025
- Durasi: 1 jam
- Focus: What went well? What can be improved?

Sprint 3-4 (Week 3-4: 2 Okt - 15 Okt 2025)

Sprint Goal: Develop Prompt Expander & Draft Agents

Sprint Planning Meeting:

- Tanggal: 2 Oktober 2025
- Durasi: 2 jam
- Output: Sprint 3-4 Backlog

Daily Standup:

- Setiap hari pukul 09:00 WIB
- Durasi: 15 menit

Sprint Review:

- Tanggal: 15 Oktober 2025
- Durasi: 1 jam
- Deliverables: Prompt Expander Agent, Draft Agent (HTML draft generation & browser preview)

Sprint Retrospective:

- Tanggal: 15 Oktober 2025
- Durasi: 1 jam

Sprint 5-6 (Week 5-6: 16 Okt - 29 Okt 2025)

Sprint Goal: Develop Prompt Planner & Page Architect Agents

Sprint Planning Meeting:

- Tanggal: 16 Oktober 2025
- Durasi: 2 jam
- Output: Sprint 5-6 Backlog

Daily Standup:

- Setiap hari pukul 09:00 WIB
- Durasi: 15 menit

Sprint Review:

- Tanggal: 29 Oktober 2025
- Durasi: 1 jam
- Deliverables: Prompt Planner Agent, Page Architect Agent, Cerebras & Mistral API Integration

Sprint Retrospective:

- Tanggal: 29 Oktober 2025
- Durasi: 1 jam

Sprint 7-8 (Week 7-8: 30 Okt - 12 Nov 2025)

Sprint Goal: Develop Generate Layout App & UI Generator Agents + Preview UI

Sprint Planning Meeting:

- Tanggal: 30 Oktober 2025
- Durasi: 2 jam
- Output: Sprint 7-8 Backlog

Daily Standup:

- Setiap hari pukul 09:00 WIB

- Durasi: 15 menit

Sprint Review:

- Tanggal: 12 November 2025
- Durasi: 1 jam
- Deliverables: Generate Layout App Agent, UI Generator Agent, Preview UI Core

Sprint Retrospective:

- Tanggal: 12 November 2025
- Durasi: 1 jam

Sprint 9-10 (Week 9-10: 13 Nov - 26 Nov 2025)

Sprint Goal: Develop Route & Component Agents

Sprint Planning Meeting:

- Tanggal: 13 November 2025
- Durasi: 2 jam
- Output: Sprint 9-10 Backlog

Daily Standup:

- Setiap hari pukul 09:00 WIB
- Durasi: 15 menit

Sprint Review:

- Tanggal: 26 November 2025
- Durasi: 1 jam
- Deliverables: Route Agent, Component Agent

Sprint Retrospective:

- Tanggal: 26 November 2025

- Durasi: 1 jam

Sprint 11 (Week 11: 27 Nov - 3 Des 2025)

Sprint Goal: Develop Validator & Move to Project Agents

Sprint Planning Meeting:

- Tanggal: 27 November 2025
- Durasi: 2 jam
- Output: Sprint 11 Backlog

Daily Standup:

- Setiap hari pukul 09:00 WIB
- Durasi: 15 menit

Sprint Review:

- Tanggal: 3 Desember 2025
- Durasi: 1 jam
- Deliverables: Validator Agent, Move to Project Agent

Sprint Retrospective:

- Tanggal: 3 Desember 2025
- Durasi: 1 jam

Sprint 12 (Week 12: 4 Des - 10 Des 2025)

Sprint Goal: Backend Polishing - Queueing, Retries, Logging, Security

Sprint Planning Meeting:

- Tanggal: 4 Desember 2025
- Durasi: 2 jam
- Output: Sprint 12 Backlog

Daily Standup:

- Setiap hari pukul 09:00 WIB
- Durasi: 15 menit

Sprint Review:

- Tanggal: 10 Desember 2025
- Durasi: 1 jam
- Deliverables: API Layer, Queue System, Security Implementation, Interactive Components

Sprint Retrospective:

- Tanggal: 10 Desember 2025
- Durasi: 1 jam

Sprint 13 (Week 13: 11 Des - 17 Des 2025)

Sprint Goal: Documentation, Examples, Testing

Sprint Planning Meeting:

- Tanggal: 11 Desember 2025
- Durasi: 2 jam
- Output: Sprint 13 Backlog

Daily Standup:

- Setiap hari pukul 09:00 WIB
- Durasi: 15 menit

Sprint Review:

- Tanggal: 17 Desember 2025
- Durasi: 1 jam

- Deliverables: Complete Documentation, Example Repository, Testing Reports

Sprint Retrospective:

- Tanggal: 17 Desember 2025
- Durasi: 1 jam

Sprint 14 (Week 14: 18 Des - 25 Des 2025)

Sprint Goal: Final Testing, Bugfix, Release/Handover

Sprint Planning Meeting:

- Tanggal: 18 Desember 2025
- Durasi: 2 jam
- Output: Sprint 14 Backlog

Daily Standup:

- Setiap hari pukul 09:00 WIB
- Durasi: 15 menit

Sprint Review:

- Tanggal: 25 Desember 2025
- Durasi: 1 jam
- Deliverables: Production-ready System, Deployed Application

Sprint Retrospective:

- Tanggal: 25 Desember 2025
- Durasi: 1.5 jam
- Focus: Overall project lessons learned

7. Sprint Planning & Ceremonies (Legacy - Removed)

Sprint 1 (18-24 November 2024)

Sprint Goal: Setup project infrastructure dan environment development

Sprint Planning Meeting:

- Tanggal: 18 November 2024
- Durasi: 2 jam
- Output: Sprint 1 Backlog

Daily Standup:

- Setiap hari pukul 09:00 WIB
- Durasi: 15 menit
- Format: What did you do? What will you do? Any blockers?

Sprint Review:

- Tanggal: 24 November 2024
- Durasi: 1 jam
- Deliverables: Project Charter, Infrastructure, CI/CD Pipeline

Sprint Retrospective:

- Tanggal: 24 November 2024
- Durasi: 1 jam
- Focus: What went well? What can be improved?

Sprint 2 (25 Nov - 8 Des 2024)

Sprint Goal: Develop semua AI agents dan backend services

Sprint Planning Meeting:

- Tanggal: 25 November 2024
- Durasi: 2 jam

- Output: Sprint 2 Backlog

Daily Standup:

- Setiap hari pukul 09:00 WIB
- Durasi: 15 menit

Sprint Review:

- Tanggal: 8 Desember 2024
- Durasi: 1 jam
- Deliverables: 6 AI Agents, API Layer, Queue System, Database

Sprint Retrospective:

- Tanggal: 8 Desember 2024
- Durasi: 1 jam

Sprint 3 (9-15 Des 2024)

Sprint Goal: Frontend development, testing, dan deployment

Sprint Planning Meeting:

- Tanggal: 9 Desember 2024
- Durasi: 2 jam
- Output: Sprint 3 Backlog

Daily Standup:

- Setiap hari pukul 09:00 WIB
- Durasi: 15 menit

Sprint Review:

- Tanggal: 15 Desember 2024
- Durasi: 1 jam

- Deliverables: Preview UI, Testing Reports, Documentation, Deployed System

Sprint Retrospective:

- Tanggal: 15 Desember 2024
- Durasi: 1.5 jam
- Focus: Overall project lessons learned

8. Definition of Done (DoD)

Sprint 1-2 DoD:

- ✅ Project charter telah direview dan diapprove
- ✅ GitHub repository setup dengan branch strategy
- ✅ CI/CD pipeline berjalan tanpa error
- ✅ Development environment dapat dijalankan oleh semua developer
- ✅ Cerebras & Mistral API integration test berhasil (dengan fallback mechanism)
- ✅ WBS, Network Diagram, dan Gantt Chart selesai

Sprint 3-11 DoD (Per Agent):

- ✅ Agent dapat berjalan tanpa error
- ✅ Unit test untuk agent passed (coverage > 70%)
- ✅ Code telah direview dan di-commit ke repository
- ✅ Documentation untuk agent tersedia
- ✅ Integration dengan agent sebelumnya berhasil

Sprint 12 DoD:

- ✅ API endpoints dapat diakses dan berfungsi
- ✅ Queue system dapat process jobs
- ✅ Database schema implemented
- ✅ Security implementation (API key, rate limiting) berfungsi
- ✅ Interactive components frontend berfungsi
- ✅ Code telah direview dan di-commit ke repository

Sprint 13 DoD:

- ✅ Frontend security (HTML sanitization, XSS prevention) implemented
- ✅ Frontend-backend terintegrasi dengan baik
- ✅ Semua test case passed (unit, integration, E2E)
- ✅ Technical & user documentation tersedia
- ✅ Example repository dengan demo app tersedia

Sprint 14 DoD:

- ✅ Final testing passed
- ✅ Performance optimization completed
- ✅ Security hardening completed
- ✅ Aplikasi berhasil di-deploy ke production
- ✅ Monitoring setup berfungsi
- ✅ Project closure & handover document completed
- ✅ Lessons learned documented

- ✅ Sprint retrospective completed

9. Risk Management

Risk	Probability	Impact	Mitigation
Cerebras/Mistral API downtime	Medium	High	Implement retry logic, automatic fallback to Mistral, cache responses
AI agent output invalid	High	High	Validator agent, extensive testing
Integration complexity	Medium	High	Early integration testing, clear API contract
Team member unavailable	Low	High	Cross-training, clear documentation
Scope creep	Medium	Medium	Strict backlog management, freeze after Sprint 1
Technical blocker	High	Medium	Daily standup for early detection

10. Tools & Technology Stack

Development Tools:

- **Project Management:** Notion (Gantt Chart, Backlog)

- **Version Control:** Git & GitHub
- **Communication:** WhatsApp Group, Discord
- **Documentation:** Markdown, GitHub Wiki

Technology Stack:

- **AI/ML:** Cerebras Qwen 3 Coder 480B (primary), Mistral AI (fallback)
- **Backend:** Python 3.7+, FastAPI/Flask
- **Queue:** Redis, Celery/RQ
- **Database:** SQLite/PostgreSQL
- **Frontend:** HTML5, CSS3 (Tailwind), Vanilla JS/Alpine.js
- **Target Framework:** Laravel 8+
- **CI/CD:** GitHub Actions
- **Deployment:** Heroku, Vercel, atau VPS

11. Success Criteria

Sprint 1-2 Success:

- ✅ Infrastructure ready untuk development
- ✅ Semua developer dapat run project locally
- ✅ CI/CD pipeline automated
- ✅ Project planning documents complete (Charter, WBS, Network, Gantt)

Sprint 3-11 Success (Per Agent):

- ✅ Agent functional dan tested
- ✅ Integration dengan workflow berjalan lancar
- ✅ No major technical debt
- ✅ Documentation updated

Sprint 12 Success:

- ✅ Backend services operational (API, Queue, Security)
- ✅ Frontend interactive components functional
- ✅ API dapat orchestrate agents
- ✅ No major technical debt

Sprint 13 Success:

- ✅ Frontend-backend terintegrasi dengan baik
- ✅ All testing passed (unit, integration, E2E)
- ✅ Documentation complete and clear
- ✅ Example repository tersedia

Sprint 14 Success:

- ✅ System deployed and accessible
- ✅ Zero critical bugs in production
- ✅ Performance optimized
- ✅ Security hardened
- ✅ Stakeholder satisfied

Overall Project Success:

- ✅ Selesai dalam 14 minggu sesuai timeline (18 Sep - 25 Des 2025)
- ✅ Dapat generate Laravel UI dari natural language prompts
- ✅ ≥ 90% generated code syntactically valid
- ✅ Preview UI allows inspection & approval
- ✅ Integration process tidak break Laravel project (staging/branch mechanism)
- ✅ Complete documentation available (technical + user)
- ✅ 10 AI Agents fully functional (Prompt Expander, Draft, Prompt Planner, Page Architect, Generate Layout App, UI Generator, Route, Component, Validator, Move to Project)

12. Kesimpulan

Proyek MCP Agents Laravel UI Generator ini dirancang dengan metode SCRUM untuk memastikan:

1. **Transparansi** - Semua progress terlihat di Gantt Chart dan daily standup
2. **Inspeksi** - Sprint review setiap akhir sprint untuk evaluasi
3. **Adaptasi** - Sprint retrospective untuk continuous improvement
4. **Kolaborasi** - Backend & Frontend bekerja sama dengan jelas
5. **Delivery** - Incremental delivery setiap sprint (1 week per sprint)

Dengan mengikuti struktur SCRUM ini, tim dapat menyelesaikan proyek AI-powered Laravel UI generator tepat waktu dengan kualitas yang baik dan pembelajaran yang maksimal tentang Agile methodology dan AI integration.

Catatan:

- Gantt Chart di atas dibuat menggunakan Notion
- Setiap task memiliki assignee, due date, dan status yang jelas
- Dependencies antar task sudah dipetakan untuk menghindari blocker
- Timeline dapat disesuaikan berdasarkan velocity tim setelah Sprint 1-2
- Focus pada MVP (Minimum Viable Product) untuk Sprint 1-12, polish & documentation Sprint 13-14
- Total durasi: 14 minggu (18 September - 25 Desember 2025)
- 6 AI Agents: Prompt Processing, Draft Generator, Layout & Component Generator, Route Generator, Validator, Project Integrator
- Technology Stack: Python 3.7+, Laravel 8+, Cerebras Qwen 3 Coder 480B (primary AI), Mistral AI (fallback), FastAPI/Flask, Redis/Celery, HTML/CSS/JS

EARNED VALUE MANAGEMENT (EVM)

GenLaravel (MCP Agents Laravel UI Generator) – Agile/Scrum Environment

1. Deskripsi Proyek

Nama Proyek: Genlaravel - MCP Agents Laravel UI Generator
Sponsor: Internal Development Team
Project Manager: Fikri Armia Fahmi (2023071018)
Frontend Developer: Nadia (2024071004)
Periode: 18 September - 25 Desember 2025 (14 minggu)

Deskripsi Sistem:

Proyek ini membangun AI-powered Laravel UI Generator yang mencakup:

- **10 AI Agents** (Prompt Expander, Draft, Prompt Planner, Page Architect, Generate Layout App, UI Generator, Route, Component, Validator, Move to Project)
- **Backend Services** (FastAPI, Redis Queue, Database)
- **Frontend Preview UI** (HTML/CSS/JS untuk interactive preview)
- **Laravel Integration** (Blade files, routes, components generation)

Model Kerja:

- **Metode:** Scrum (Sprint 1 minggu)
- **Durasi:** 14 Sprint
- **Total Budget (BAC):** Rp 35.000.000 (≈ $2,300)
- **Backlog Total:** 280 Story Points (SP)
- **Nilai per SP:** ≈ Rp 125.000 (untuk memudahkan perhitungan nilai)

2. Menetapkan Baseline (Sesuai Teori EVM)

Baseline proyek ditetapkan sebagai berikut:

- **Budget at Completion (BAC)** = Rp 35.000.000
- **Total Story Point** = 280 SP
- **Planned completion per 13 Sprint** = 93% pekerjaan

Artinya:

- Pada Sprint ke-13, proyek seharusnya menyelesaikan 260 SP.

Jadi nilai PV pada Sprint ke-13:

PV = 93% × Rp 35.000.000 = Rp 32.550.000

3. Menghitung Planned Value (PV)

(nilai pekerjaan yang seharusnya selesai)

Sampai Sprint ke-13 (11 Desember - 17 Desember 2025):

- Target penyelesaian = 93% pekerjaan (13 dari 14 sprint)
- Nilai anggaran proyek = Rp 35.000.000

PV = 0.93 × Rp 35.000.000 = Rp 32.550.000

Interpretasi:
Pada akhir Sprint ke-13, proyek seharusnya telah menyelesaikan pekerjaan senilai Rp 32.550.000.

4. Menghitung Earned Value (EV)

(nilai pekerjaan yang benar-benar selesai)

Aktual di lapangan (Sprint ke-13 selesai, Sprint ke-14 in progress):

Komponen	**Progress**	**Keterangan**
Project Setup & Infrastructure	100%	Selesai di Sprint 1-2
Prompt Expander Agent	100%	Selesai di Sprint 3-4
Draft Agent	100%	Selesai di Sprint 3-4
Prompt Planner Agent	100%	Selesai di Sprint 5-6
Page Architect Agent	100%	Selesai di Sprint 5-6
Generate Layout App Agent	100%	Selesai di Sprint 7-8
UI Generator Agent	100%	Selesai di Sprint 7-8
Preview UI Core	100%	Selesai di Sprint 7-8

Route Agent	100%	Selesai di Sprint 9-10
Component Agent	100%	Selesai di Sprint 9-10
Validator Agent	100%	Selesai di Sprint 11
Move to Project Agent	100%	Selesai di Sprint 11
Backend Polishing	100%	Selesai di Sprint 12
Documentation & Testing	100%	Selesai di Sprint 13
Final Testing & Release	70%	In progress di Sprint 14

Perhitungan Total Progress:

Jika dijumlahkan berdasarkan bobot story points:

- Sprint 1-2 (Setup): 20 SP → 100% = 20 SP
- Sprint 3-4 (Prompt Expander + Draft): 40 SP → 100% = 40 SP
- Sprint 5-6 (Planner + Architect): 40 SP → 100% = 40 SP
- Sprint 7-8 (Layout + UI Generator + Preview): 60 SP → 100% = 60 SP
- Sprint 9-10 (Route + Component): 70 SP → 100% = 70 SP
- Sprint 11 (Validator + Move to Project): 35 SP → 100% = 35 SP
- Sprint 12 (Backend Polishing): 35 SP → 100% = 35 SP
- Sprint 13 (Documentation & Testing): 30 SP → 100% = 30 SP
- Sprint 14 (Final Testing & Release): 20 SP → 70% = 14 SP

Total EV:

```
Total progress = 344 SP dari 350 SP = 98.3%
EV = 0.983 × Rp 35.000.000 = Rp 34.405.000
```

Catatan: Total SP disesuaikan menjadi 350 SP (dari 280 SP) karena scope adjustment selama proyek.

5. Menghitung Actual Cost (AC)

(biaya yang benar-benar keluar)

Biaya selama Sprint 1–13 (Sprint 14 in progress):

A. Biaya Tim Scrum (Mahasiswa - Honor/Insentif)

- **Backend Developer + PM (Fikri):** Rp 500.000/sprint
- **Frontend Developer (Nadia):** Rp 400.000/sprint
- **Junior Helper (Sprint 9-13):** Rp 200.000/sprint × 5 sprint = Rp 1.000.000
- **Total per sprint (Sprint 1-8):** Rp 900.000 × 8 = Rp 7.200.000
- **Total per sprint (Sprint 9-13):** Rp 1.100.000 × 5 = Rp 5.500.000
- **Total Sprint 1-13:Rp 12.700.000**

B. Biaya Tools & Cloud

- **Cerebras API (Pay-as-you-go):**
- Model: Qwen 3 32B (~2600 tokens/s)
- Input: $0.40/M tokens | Output: $0.80/M tokens
- Actual usage Sprint 1-13:
- Input: ~18M tokens × $0.40 = $7.20
- Output: ~9M tokens × $0.80 = $7.20
- Total: $14.40 ≈ **Rp 230.000**
- **Mistral API (Pay-as-you-go - Fallback):**
- Model: Codestral (coding specialist)
- Input: $0.30/M tokens | Output: $0.90/M tokens
- Actual usage Sprint 1-13 (saat Cerebras rate limit):
- Input: ~3M tokens × $0.30 = $0.90
- Output: ~1.5M tokens × $0.90 = $1.35
- Total: $2.25 ≈ **Rp 36.000**
- **Cloud hosting (Railway Free Tier):** Rp 0 (free tier 500 jam/bulan)
- **GitHub (Student Pack):** Rp 0 (gratis untuk mahasiswa)
- **Notion (Education Plan):** Rp 0 (gratis untuk mahasiswa)
- **Domain & SSL:** Rp 150.000 (1 tahun .my.id domain)
- **Total Sprint 1-13:Rp 416.000**

C. Biaya Konsultan & Training

- **Dosen Pembimbing (gratis - bagian dari tugas):** Rp 0
- **Laravel Expert Review (senior mahasiswa):** Rp 500.000
- **Security Audit (Sprint 12):** Rp 800.000
- **AI/ML Workshop (online gratis):** Rp 0
- **Total:Rp 1.300.000**

D. Biaya Operasional & Tak Terduga

- **Internet & Listrik (13 minggu):** Rp 1.300.000
- **Dokumentasi & Printing:** Rp 500.000
- **Frontend developer sakit (3 hari, teman bantu gratis):** Rp 0
- **Cerebras API downtime (masih dalam free tier Mistral):** Rp 0
- **Makan & transportasi meeting:** Rp 650.000
- **Total:Rp 2.450.000**

E. Biaya Peralatan (Amortisasi)

- **Laptop (sudah punya, amortisasi 3.5 bulan):** Rp 1.750.000
- **Software License (mostly free/student):** Rp 0
- **Total:Rp 1.750.000**

Total AC (Sprint 1-13):

```
AC = Rp 12.700.000 + Rp 416.000 + Rp 1.300.000 + Rp 2.450.000 + Rp 1.750.000
AC = Rp 18.616.000
```

6. Menghitung Indikator Kinerja (CPI & SPI)

(indikator dari teori EVM sekarang kita aplikasikan)

A. Cost Performance Index (CPI)

```
CPI = EV / AC
CPI = Rp 34.405.000 / Rp 18.616.000 = 1.85
```

Interpretasi:

- **CPI > 1** → Proyek **Under Budget** (sangat efisien!)
- Artinya, setiap Rp 1 yang dikeluarkan menghasilkan **Rp 1.85 nilai pekerjaan**
- Efisiensi biaya mencapai **185%** (excellent!)

B. Schedule Performance Index (SPI)

```
SPI = EV / PV
SPI = Rp 34.405.000 / Rp 32.550.000 = 1.06
```

Interpretasi:

- **SPI > 1** → Proyek **ahead of schedule** (lebih cepat dari rencana!)
- Artinya, pekerjaan mencapai **106% dari target Sprint ke-13**
- Proyek lebih cepat **6%** dari jadwal

7. Variance (Selisih Kinerja)

Cost Variance (CV)

```
CV = EV - AC
CV = Rp 34.405.000 - Rp 18.616.000 = Rp 15.789.000
```

Interpretasi:
Positif → penghematan biaya sebesar **Rp 15.789.000** (84.8% dari AC)

Schedule Variance (SV)

```
SV = EV - PV
SV = Rp 34.405.000 - Rp 32.550.000 = +Rp 1.855.000
```

Interpretasi:
Positif → jadwal lebih cepat senilai pekerjaan **Rp 1.855.000** (5.7% dari PV)

8. Prediksi Biaya Akhir Proyek

Estimate at Completion (EAC)

Karena CPI > 1 (efisien), kita gunakan formula optimistic:

```
EAC = AC + (BAC - EV) / CPI
```

```
EAC = Rp 18.616.000 + (Rp 35.000.000 - Rp 34.405.000) / 1.85
EAC = Rp 18.616.000 + Rp 595.000 / 1.85
EAC = Rp 18.616.000 + Rp 321.622
EAC = Rp 18.937.622
```

Prediksi biaya akhir proyek:

- **Rp 18.937.622** → di bawah anggaran Rp 35.000.000
- **Penghematan:** Rp 16.062.378 (45.9%)

Estimate to Complete (ETC)

```
ETC = EAC - AC
ETC = Rp 18.937.622 - Rp 18.616.000 = Rp 321.622
```

Biaya tambahan yang masih dibutuhkan:

- **Rp 321.622** untuk menyelesaikan sisa proyek (Sprint 14 - 30% remaining)

Variance at Completion (VAC)

```
VAC = BAC - EAC
VAC = Rp 35.000.000 - Rp 18.937.622 = Rp 16.062.378
```

Interpretasi:

Proyek diperkirakan akan **under budget sebesar Rp 16.062.378** (penghematan 45.9%)

9. Analisis Proyek Berdasarkan EVM

EVM menunjukkan hasil:

Indikator	Nilai	Makna	Status
CPI	1.85	Under Cost	Penggunaan biaya sangat efisien
SPI	1.06	Ahead of Schedule	Progres lebih cepat
CV	+Rp 15.789.000	Positif	Penghematan biaya besar
SV	+Rp 1.855.000	Positif	Lebih cepat dari jadwal
EAC	Rp 18.937.622	Turun	Hemat 45.9%

VAC	+Rp 16.062.378	Positif	Under budget sangat signifikan

10. Penyebab Keterlambatan (Analisis Scrum)

Melalui **Sprint Retrospective**, ditemukan penyebab:

Kompleksitas Blade Template Engine

- Generate Layout App Agent lebih kompleks dari estimasi awal
- Membutuhkan deep understanding Laravel Blade syntax
- **Dampak:** Sprint 7-8 hanya 80% selesai

Cerebras API Rate Limiting & Learning Curve

- Cerebras Qwen 3 32B sangat cepat (~2600 tokens/s) tapi perlu optimasi prompt
- Rate limiting: 60 requests/minute (perlu batch processing)
- Learning curve untuk Cerebras Inference API
- Fallback ke Mistral Codestral saat rate limit (coding specialist)
- **Dampak:** Delay 1 hari pada UI Generator Agent untuk optimasi prompt

Kapasitas Tim Tidak Realistis

- Sprint capacity terlalu optimis (40 SP/sprint)
- Actual velocity hanya 32 SP/sprint
- **Dampak:** Perlu revisi sprint planning

Technical Debt Menumpuk

- Code review tidak dilakukan secara konsisten
- Refactoring tertunda karena kejar deadline
- **Dampak:** Bug fixing memakan waktu lebih lama

Scope Creep

- Product Owner menambahkan 2 fitur baru di Sprint 6
- Tidak ada formal change request process
- **Dampak:** Backlog bertambah 20 SP tanpa budget tambahan

11. Tindakan Korektif (Scrum + PMBOK EVM)

1. Re-Prioritization Backlog

- **PO** menggeser fitur non-critical ke Sprint akhir
- Focus pada MVP (Minimum Viable Product) dulu
- Fitur enhancement ditunda ke fase 2

2. Menambah 1 Mahasiswa Helper (Junior)

- Teman sekelas membantu untuk task-task sederhana
- Fokus pada testing & documentation (Sprint 9-10)
- **Biaya tambahan:** Rp 200.000/sprint × 4 sprint = Rp 800.000

3. Optimasi Penggunaan API

- Optimasi prompt untuk reduce token usage (caching, compression)
- Cerebras Qwen 3 32B untuk general tasks (balance speed & cost)
- Mistral Codestral untuk complex coding tasks (specialist)
- Batch processing untuk avoid rate limiting
- **Hasil:** Token usage turun 30%, biaya API hanya Rp 121.000 (8 sprint)

4. Sprint Capacity Adjustment

- Velocity diturunkan dari 40 SP/sprint menjadi 32 SP/sprint
- Sprint duration tetap 1 minggu, tapi realistic planning
- Buffer 20% untuk unexpected issues

5. Daily Monitoring + Burn Down Chart

- Daily standup lebih fokus pada blocker identification
- Burn down chart diupdate real-time di Notion
- Scrum Master (Fikri) melakukan daily check-in

6. Budget Reallocation

- Realokasi budget dari “Contingency Reserve” ke “Development”
- Tidak perlu budget tambahan karena CPI > 1 (efisien)
- **Status:** Approved, masih dalam budget original

7. Risk Mitigation Enhancement

- Implement automatic fallback ke Mistral API
- Setup monitoring & alerting untuk API downtime
- Backup frontend developer on-call

8. Change Control Process

- Formal change request form untuk scope changes
- Impact analysis (cost, schedule, resources) sebelum approve
- Product Owner harus justify business value

12. Proyeksi Setelah Tindakan Korektif

Skenario 1: Maintain Current Efficiency (Conservative)

Asumsi:

- CPI tetap di 1.52 (maintain efficiency)
- Menambah 1 junior helper (Rp 800.000)
- API optimization (no additional cost)
- Velocity improvement 5% (dari 32 → 34 SP/sprint)

Perhitungan:

```
Revised EAC = AC + (BAC - EV) / CPI
Revised EAC = Rp 10.471.000 + (Rp 35.000.000 - Rp 15.750.000) / 1.50
Revised EAC = Rp 10.471.000 + Rp 12.833.333
Revised EAC = Rp 23.304.333
```

Dengan additional helper:

```
Final EAC = Rp 23.304.333 + Rp 800.000 = Rp 24.104.333
```

Kesimpulan: Masih under budget Rp 10.895.667 (31.1% saving)

Skenario 2: Aggressive Optimization (Optimistic)

Asumsi:

- CPI improvement ke 1.60 (lebih efisien lagi)
- Tidak menambah helper (teman bantu gratis)
- Full API optimization + caching

- Velocity improvement 10% (dari 32 → 35 SP/sprint)

Perhitungan:

```
Revised EAC = AC + (BAC - EV) / 1.60
Revised EAC = Rp 10.471.000 + (Rp 35.000.000 - Rp 15.750.000) / 1.60
Revised EAC = Rp 10.471.000 + Rp 12.031.250
Revised EAC = Rp 22.502.250
```

Kesimpulan:

- Under budget: Rp 12.497.750 (35.7% saving)
- Bisa dialokasikan untuk: deployment, documentation, atau buffer
- **Sangat achievable dengan current momentum**

13. Rekomendasi untuk Sprint 9-14

Sprint 9-10 (Route & Component Agents)

- **Focus:** Complete Route Agent & Component Agent
- **Target:** 70 SP (35 SP/sprint)
- **Budget:** Rp 2.200.000 (Rp 1.100.000/sprint)
- **Risk:** Medium - achievable dengan current momentum

Sprint 11 (Validator & Move to Project)

- **Focus:** Validator Agent & Move to Project Agent
- **Target:** 35 SP
- **Budget:** Rp 1.100.000
- **Risk:** Low - straightforward implementation

Sprint 12 (Backend Polishing)

- **Focus:** API Layer, Queue System, Security
- **Target:** 35 SP
- **Budget:** Rp 1.500.000 (termasuk security testing)
- **Risk:** Medium - security testing membutuhkan extra time

Sprint 13 (Documentation & Testing)

- **Focus:** Complete documentation, unit/integration/E2E testing

- **Target:** 30 SP
- **Budget:** Rp 1.200.000
- **Risk:** Low - mostly documentation work

Sprint 14 (Final Testing & Release)

- **Focus:** Final testing, bugfix, deployment, handover
- **Target:** 20 SP
- **Budget:** Rp 1.000.000
- **Risk:** Low - buffer sprint

Total Remaining Budget Needed:

```
Sprint 9-14 = Rp 2.200.000 + Rp 1.100.000 + Rp 1.500.000 + Rp 1.200.000 + Rp 1.000.000
            = Rp 7.000.000
```

Total Project Cost (Projected):

```
Total = AC (Sprint 1-8) + Remaining (Sprint 9-14)
Total = Rp 10.471.000 + Rp 7.000.000 = Rp 17.471.000
```

Final Variance (Projected):

```
VAC = BAC - Total Cost
VAC = Rp 35.000.000 - Rp 17.471.000 = +Rp 17.529.000
```

Kesimpulan:
Proyek diperkirakan selesai dengan **penghematan 50.1%** dari budget!

14. Lessons Learned

Yang Berjalan Baik:

1. **Scrum Framework Implementation**

- Daily standup efektif untuk early blocker detection
- Sprint review memberikan feedback loop yang baik
- Retrospective mengidentifikasi root cause issues

2. **AI Agent Architecture**

• Modular design memudahkan parallel development

• Fallback mechanism (Cerebras → Mistral) terbukti valuable

• Agent testing framework berjalan baik

3. **Team Collaboration**

• Backend & Frontend collaboration smooth

• GitHub workflow & CI/CD automation membantu

• Documentation practice konsisten

Yang Perlu Diperbaiki:

1. **Estimation Accuracy**

• Initial story point estimation terlalu optimis

• Perlu historical data untuk better estimation

• Complexity spike tidak diantisipasi

2. **Risk Management**

• External dependency (Cerebras API) underestimated

• No backup plan untuk team member unavailability

• Technical debt tidak dimonitor

3. **Change Control**

• Scope creep tidak terkontrol di Sprint 6

• Perlu formal change request process

• Impact analysis harus mandatory

4. **Budget Planning**

• Budget planning terlalu konservatif (actual jauh lebih murah)

• Free tier API sangat membantu (Cerebras, Mistral, Heroku)

• Student benefits underutilized di awal (GitHub, Notion)

Rekomendasi untuk Proyek Berikutnya:

1. **Planning Phase:**

• Manfaatkan free tier & student benefits sejak awal

• Lakukan spike/POC untuk high-risk components

• Realistic estimation dengan buffer 20%

2. **Execution Phase:**

• Implement strict change control process

• Daily monitoring EVM metrics (CPI, SPI)

• Weekly sprint review dengan dosen pembimbing

3. **Monitoring & Control:**

• Simple EVM tracking di spreadsheet/Notion

• Burn down chart + velocity tracking

• Cost tracking per sprint (actual vs planned)

4. **Team Management:**

• Peer programming untuk knowledge sharing

• Backup dari teman sekelas (mutual help)

• Regular communication dengan pembimbing

5. **Cost Optimization:**

• Pilih model yang tepat: Cerebras Qwen 3 32B (speed) vs Mistral Codestral (precision)

• Gunakan student benefits (GitHub, Notion, Railway)

• Optimasi prompt untuk reduce token usage (30% reduction)

• Caching & batch processing untuk avoid rate limiting

• Monitor API usage per sprint untuk cost control

15. Kesimpulan Akhir

Status Proyek (Sprint ke-13 selesai, Sprint ke-14 in progress):

Aspek	Status	Keterangan
Schedule	Ahead of Schedule	6% lebih cepat, SPI = 1.06
Cost	Under Budget	45.9% saving, CPI = 1.85
Scope	Completed	Semua fitur MVP selesai
Quality	Excellent	Testing coverage > 90%

Risk	Very Low	Tinggal deployment & handover

Tindakan yang Telah Dilakukan:

1. Menambah 1 junior helper (teman sekelas) - Sprint 9-13
2. Re-prioritize backlog (MVP focus berhasil)
3. Adjust sprint velocity (realistis 32-35 SP/sprint)
4. Implement change control process (scope terkontrol)
5. Optimasi API usage (Cerebras Qwen 3 32B + Mistral Codestral)
6. Maintain budget efficiency (CPI meningkat dari 1.50 → 1.85)

Hasil Akhir (Projected):

Dengan semua tindakan korektif:

- **Estimated Final Cost:** Rp 18.937.622
- **Under Budget:** Rp 16.062.378 (45.9% saving dari original BAC)
- **Completion Date:** 25 Desember 2025 (on time, bahkan lebih cepat 6%)
- **Success Probability:** 98% (hampir selesai)

Key Takeaways:

EVM memberikan **early warning system** dan **monitoring tool** yang objektif bahwa proyek MCP Agents Laravel UI Generator:

- Ahead of schedule (SPI = 1.06) - lebih cepat 6%!
- Sangat efisien dalam biaya (CPI = 1.85) - excellent!
- Hemat Rp 16.062.378 (45.9% dari budget)

Melalui kombinasi:

- **Scrum ceremonies** (review, retrospective) - konsisten setiap sprint
- **Resource optimization** (junior helper, teman bantu) - efektif
- **Backlog reprioritization** (MVP focus) - berhasil
- **API optimization** (token reduction, caching) - biaya API hanya Rp 266.000 (13 sprint)
- **Change control** (formal process) - scope terkontrol

Maka proyek **berhasil maintain efficiency** dengan:

- Final cost: Rp 18.937.622 (vs budget Rp 35.000.000)

• On-time delivery: 25 Desember 2025 (bahkan lebih cepat)

• Quality maintained: >90% test coverage

• Stakeholder satisfaction: Very High (under budget + ahead of schedule + excellent quality)

Catatan API Usage (Sprint 1-13):

• Cerebras Qwen 3 32B: Sangat cepat (~2600 tokens/s), cocok untuk iterasi cepat

• Mistral Codestral: Specialist untuk coding tasks, output lebih presisi

• Total biaya API 13 sprint: hanya Rp 266.000 (sangat terjangkau!)

• Optimasi prompt berhasil reduce token usage 30%

Aspek	Status	Keterangan
Schedule	Behind	10% delay, SPI = 0.90
Cost	Under Budget	50.1% saving, CPI = 1.50
Scope	At Risk	Scope creep 20 SP
Quality	On Track	Testing coverage > 70%
Risk	Low	Manageable dengan current resources

Tindakan yang Diambil:

1. Menambah 1 junior helper (teman sekelas, Rp 800.000)
2. Re-prioritize backlog (defer non-critical features)
3. Adjust sprint velocity (40 → 32 SP/sprint)
4. Implement change control process
5. Optimasi API usage (Cerebras Qwen 3 32B + Mistral Codestral)
6. Maintain budget efficiency (CPI = 1.50)

Proyeksi Akhir:

Dengan tindakan korektif:

• **Estimated Final Cost:** Rp 17.471.000

• **Under Budget:** Rp 17.529.000 (50.1% saving dari original BAC)

• **Completion Date:** 25 Desember 2025 (on schedule dengan minor delay)

- **Success Probability:** 85% (dengan current momentum)

Key Takeaways:

EVM memberikan **early warning system** yang objektif bahwa proyek MCP Agents Laravel UI Generator:

- Tertinggal jadwal (SPI = 0.90) - perlu catch up
- Sangat efisien dalam biaya (CPI = 1.50) - excellent!
- Berpotensi hemat sampai Rp 17.529.000 (50.1%)

Namun melalui kombinasi:

- **Scrum ceremonies** (review, retrospective)
- **Resource optimization** (junior helper, teman bantu)
- **Backlog reprioritization** (MVP focus)
- **API optimization** (token reduction, caching)
- **Change control** (formal process)

Maka proyek dapat **maintain efficiency** dengan:

- Final cost: Rp 17.471.000 (vs budget Rp 35.000.000)
- On-time delivery: 25 Desember 2025 (dengan minor adjustment)
- Quality maintained: >90% test coverage
- Stakeholder satisfaction: Very High (under budget + good quality)

Catatan API Usage:

- Cerebras Qwen 3 32B: Sangat cepat (~2600 tokens/s), cocok untuk iterasi cepat
- Mistral Codestral: Specialist untuk coding tasks, output lebih presisi
- Total biaya API 8 sprint: hanya Rp 121.000 (sangat terjangkau!)

Analisis Procurement "Make-or-Buy"

Proyek GenLaravel (MCP Agents Laravel UI Generator)

Nama Proyek: GenLaravel - MCP Agents Laravel UI Generator
Project Manager: Fikri Armia Fahmi (2023071018)
Frontend Developer: Nadia (2024071004)
Tanggal Analisis: 20 Desember 2025

1. Pendahuluan

Analisis Make-or-Buy merupakan proses pengambilan keputusan strategis untuk menentukan apakah suatu komponen atau layanan dalam proyek akan dikembangkan secara internal (make) atau diperoleh dari pihak eksternal/vendor (buy). Keputusan ini mempertimbangkan faktor biaya, waktu, ketersediaan sumber daya, kompetensi tim, dan risiko.

Proyek GenLaravel memerlukan berbagai komponen teknologi yang perlu dianalisis untuk menentukan pendekatan pengadaan yang paling efektif dan efisien.

2. Kriteria Evaluasi Make-or-Buy

Dalam melakukan analisis, digunakan kriteria evaluasi sebagai berikut:

Kriteria	**Bobot**	**Deskripsi**
Biaya	25%	Total cost of ownership termasuk development, maintenance, dan operasional
Waktu	20%	Kecepatan implementasi dan time-to-market
Kompetensi	20%	Ketersediaan skill dan expertise dalam tim
Kontrol	15%	Tingkat kontrol terhadap kualitas dan customization
Risiko	10%	Risiko teknis, vendor dependency, dan keamanan
Skalabilitas	10%	Kemampuan untuk scale up sesuai kebutuhan

3. Analisis Make-or-Buy per Komponen

Komponen	**Make**	**Buy**	**Alasan**	**Keputusan**
AI Inference Engine	Membangun infrastruktur AI sendiri dengan GPU server, maintenance kompleks, biaya tinggi (Rp 50-100	Menggunakan Cerebras API (Qwen 3 32B) dan Mistral API (Codestral) dengan pay-as-you-go, biaya rendah	Tim tidak memiliki expertise dalam AI infrastructure, biaya hardware sangat tinggi, API sudah production-ready dengan performance	**BUY** - Cerebras + Mistral API

	juta), waktu development 6+ bulan	($0.40-0.80/M tokens), ready to use	excellent (~2600 tokens/s), pay-as-you-go model sangat cost-effective untuk skala proyek ini	
10 AI Agents (Core Logic)	Mengembangkan sendiri logic untuk Prompt Expander, Draft, Planner, Architect, Layout Generator, UI Generator, Route, Component, Validator, Move to Project agents	Outsource ke AI development company, estimasi Rp 80-150 juta, waktu 2-3 bulan	Core competency proyek ada di agent development, tim memiliki expertise Python dan Laravel, customization penuh diperlukan untuk integrasi dengan workflow spesifik, intellectual property tetap di internal	**MAKE** - Internal Development
Backend API Framework	Membangun framework API dari scratch	Menggunakan FastAPI (open-source, gratis) dengan ecosystem yang mature	FastAPI sudah production-ready, dokumentasi lengkap, performance excellent, community support kuat, tidak perlu reinvent the wheel	**BUY** - FastAPI (Open Source)
Queue System	Menggunakan asyncio.Queue (Python built-in) untuk in-memory queue dengan pattern FIFO, cocok untuk single-instance deployment	Menggunakan Redis + Celery atau RabbitMQ untuk distributed queue yang persistent dan scalable	Untuk skala proyek ini (MVP, single-instance), asyncio.Queue sudah memadai, tidak perlu kompleksitas message broker eksternal, zero dependency tambahan,	**MAKE** - asyncio.Queue (Built-in Python)

			development lebih cepat	
Frontend Preview UI	Mengembangkan sendiri dengan HTML/CSS/JS	Outsource ke frontend agency atau menggunakan template premium	Tim memiliki frontend developer (Nadia), customization tinggi diperlukan untuk integrasi dengan agent output, biaya outsource tidak sebanding dengan scope	**MAKE** - Internal Development
Cloud Hosting	Setup server sendiri (on-premise atau VPS manual)	Menggunakan Railway/Heroku dengan managed service	Managed service mengurangi operational overhead, free tier tersedia untuk development, auto-scaling, monitoring built-in	**BUY** - Railway (Managed Service)
Version Control & CI/CD	Setup GitLab self-hosted	Menggunakan GitHub dengan GitHub Actions	GitHub Student Pack gratis, GitHub Actions sudah terintegrasi, tidak perlu maintenance server	**BUY** - GitHub (SaaS)
Database	Setup MongoDB server sendiri (on-premise)	Menggunakan MongoDB Atlas (managed cloud database) dengan free tier 512MB	MongoDB Atlas menyediakan free tier yang memadai untuk MVP, managed service mengurangi operational overhead, NoSQL cocok untuk menyimpan data agent output yang semi-structured	**BUY** - MongoDB Atlas (Managed Service)

Domain & SSL	Tidak applicable	Membeli domain .my.id dan SSL certificate	Domain diperlukan untuk production deployment, biaya terjangkau (Rp 150.000/tahun)	**BUY** - Domain Provider
Documentation Platform	Membangun documentation site sendiri	Menggunakan Notion (Education Plan gratis) atau GitHub Wiki	Notion sudah familiar, gratis untuk education, collaborative features excellent	**BUY** - Notion (SaaS)
Project Management Tool	Spreadsheet manual	Menggunakan Notion dengan Gantt chart dan Kanban	Notion sudah digunakan tim, fitur lengkap, gratis untuk education	**BUY** - Notion (SaaS)
Security Audit	Internal review saja	Menggunakan jasa security consultant untuk review	Tim tidak memiliki expertise security audit yang mendalam, external review memberikan perspektif objektif, one-time cost	**BUY** - Security Consultant
Laravel Integration Module	Mengembangkan sendiri integrasi dengan Laravel project	Menggunakan Laravel package yang sudah ada	Tidak ada package yang sesuai dengan kebutuhan spesifik (AI-generated Blade integration), perlu custom development	**MAKE** - Internal Development
Testing Framework	Membangun testing framework sendiri	Menggunakan pytest (Python) dan PHPUnit (Laravel)	Framework testing sudah mature dan proven, tidak perlu custom development	**BUY** - pytest + PHPUnit (Open Source)

4. Ringkasan Keputusan Make-or-Buy

Komponen MAKE (Internal Development)

1. 10 AI Agents (Core Logic) - Core competency, IP protection
2. Frontend Preview UI - Customization tinggi, tim capable
3. Laravel Integration Module - Kebutuhan spesifik, tidak ada solusi existing
4. Queue System - asyncio.Queue (Python built-in), simple dan memadai untuk MVP

Komponen BUY (External/Vendor)

1. AI Inference Engine - Cerebras API + Mistral API
2. Backend API Framework - FastAPI (Open Source)
3. Cloud Hosting - Railway (Managed Service)
4. Version Control & CI/CD - GitHub + GitHub Actions
5. Database - MongoDB Atlas (Managed Service)
6. Domain & SSL - Domain Provider
7. Documentation Platform - Notion
8. Project Management Tool - Notion
9. Security Audit - External Consultant
10. Testing Framework - pytest + PHPUnit (Open Source)

Distribusi Keputusan

Kategori	Jumlah	Persentase
MAKE (Internal)	4 komponen	29%
BUY (External)	10 komponen	71%

Distribusi ini menunjukkan strategi yang fokus pada core competency (AI agents, queue system, dan integrasi) sambil memanfaatkan solusi existing untuk komponen pendukung. Penggunaan asyncio.Queue sebagai in-memory queue merupakan keputusan pragmatis untuk MVP yang dapat di-upgrade ke Redis/Celery jika diperlukan scaling di masa depan.

5. Analisis Biaya Make vs Buy

Komponen AI Inference Engine

Aspek	**MAKE**	**BUY**
Initial Cost	Rp 50-100 juta (GPU server)	Rp 0 (pay-as-you-go)
Monthly Cost	Rp 5-10 juta (maintenance, listrik)	Rp 20.000-50.000 (usage-based)
Development Time	6+ bulan	Immediate
Expertise Required	AI/ML Infrastructure Engineer	API Integration (existing skill)
Risk	Tinggi (hardware failure, obsolescence)	Rendah (vendor managed)
Total 14 Minggu	**Rp 65-120 juta**	**Rp 266.000**

Keputusan BUY menghemat sekitar Rp 65-120 juta dan 6 bulan development time.

Komponen AI Agents (Core Logic)

Aspek	**MAKE**	**BUY (Outsource)**
Initial Cost	Rp 12.7 juta (tim internal)	Rp 80-150 juta (vendor)
Control	Penuh	Terbatas
Customization	Unlimited	Sesuai kontrak
IP Ownership	Internal	Perlu negosiasi
Knowledge Transfer	Otomatis	Perlu effort tambahan
Total	**Rp 12.7 juta**	**Rp 80-150 juta**

Keputusan MAKE menghemat Rp 67-137 juta dan mempertahankan IP ownership.

6. Vendor yang Dipilih

Berdasarkan analisis di atas, berikut vendor yang dipilih untuk komponen BUY:

Komponen	Vendor	Model Pricing	Estimasi Biaya (14 Minggu)
AI Inference (Primary)	Cerebras	Pay-as-you-go ($0.40-0.80/M tokens)	Rp 230.000
AI Inference (Fallback)	Mistral	Pay-as-you-go ($0.30-0.90/M tokens)	Rp 36.000
Cloud Hosting	Railway	Free Tier + Usage	Rp 0
Version Control	GitHub	Free (Student Pack)	Rp 0

Documentation	Notion	Free (Education)	Rp 0
Domain	Domainesia	Annual	Rp 150.000
Security Audit	Freelance Consultant	One-time	Rp 800.000
Total			**Rp 1.216.000**

Draft Dokumen Kontrak Pengadaan

1.1 Ruang Lingkup Pekerjaan (Statement of Work / SOW)

STATEMENT OF WORK
Pengadaan Layanan AI Inference API
Proyek GenLaravel

A. Latar Belakang

Proyek GenLaravel memerlukan layanan AI inference untuk menjalankan 10 AI agents yang berfungsi menghasilkan Laravel UI components dari natural language prompts. Layanan ini harus mampu memproses request dengan kecepatan tinggi, akurasi output yang baik, dan ketersediaan yang reliable.

B. Tujuan Pengadaan

Pengadaan layanan AI inference API bertujuan untuk:

1. Menyediakan infrastruktur AI inference yang reliable dan scalable
2. Mendukung operasional 10 AI agents dalam pipeline generation
3. Memastikan response time yang cepat untuk user experience yang optimal
4. Meminimalkan biaya operasional dengan model pay-as-you-go

C. Ruang Lingkup Layanan

Vendor menyediakan layanan sebagai berikut:

1. **AI Model Access**

• Akses ke Large Language Model untuk text generation

• Model yang mendukung coding tasks dan natural language understanding

• Kemampuan untuk memproses prompt kompleks (multi-turn conversation)

2. **API Infrastructure**

• RESTful API endpoint yang secure (HTTPS)

• Authentication menggunakan API key

• Rate limiting yang reasonable untuk development dan production

3. **Performance**

• Inference speed minimal 1000 tokens/second

• Latency maksimal 500ms untuk first token

• Uptime minimal 99.5%

4. **Support**

- Dokumentasi API yang lengkap
- Status page untuk monitoring availability
- Support channel untuk technical issues

D. Deliverables

No	Deliverable	Deskripsi	Timeline
1	API Access	API key dan endpoint access	Day 1
2	Documentation	API documentation dan integration guide	Day 1
3	Usage Dashboard	Dashboard untuk monitoring usage dan billing	Day 1
4	Support Access	Akses ke support channel	Day 1

E. Periode Kontrak

Kontrak berlaku selama periode proyek (14 minggu) dengan opsi perpanjangan berdasarkan kebutuhan. Model pricing pay-as-you-go memungkinkan fleksibilitas tanpa commitment jangka panjang.

F. Lokasi Pekerjaan

Layanan disediakan secara cloud-based, dapat diakses dari lokasi manapun dengan koneksi internet.

G. Asumsi dan Batasan

Asumsi:

- Koneksi internet tersedia dan stabil
- Tim memiliki kemampuan untuk integrasi API
- Usage tidak melebihi rate limit yang ditetapkan

Batasan:

- Layanan terbatas pada API inference, tidak termasuk fine-tuning
- Data yang dikirim ke API tidak disimpan permanen oleh vendor
- Tidak ada dedicated support, menggunakan community/standard support

1.2 Service Level Agreement (SLA)

SERVICE LEVEL AGREEMENT
Layanan AI Inference API

A. Definisi Layanan

Layanan AI Inference API mencakup akses ke endpoint API untuk melakukan text generation menggunakan Large Language Model yang disediakan vendor.

B. Availability

Metric	Target	Measurement
Monthly Uptime	99.5%	(Total Minutes - Downtime Minutes) / Total Minutes × 100%
Scheduled Maintenance	Maksimal 4 jam/bulan	Dengan notifikasi 24 jam sebelumnya
Unscheduled Downtime	Maksimal 2 jam/incident	Measured from incident report to resolution

C. Performance Metrics

Metric	Target	Measurement
Inference Speed	Minimal 1000 tokens/second	Average across all requests
Time to First Token	Maksimal 500ms	P95 latency
Request Success Rate	99%	Successful requests / Total requests
Error Rate	Maksimal 1%	Failed requests / Total requests

D. Support Response Time

Priority	Description	Response Time	Resolution Time
Critical	Service completely unavailable	1 jam	4 jam
High	Significant performance degradation	4 jam	24 jam
Medium	Minor issues, workaround available	24 jam	72 jam
Low	Questions, feature requests	72 jam	Best effort

E. Kompensasi (Service Credits)

Jika SLA tidak terpenuhi, kompensasi diberikan dalam bentuk service credits:

Monthly Uptime	Service Credit
99.0% - 99.5%	10% of monthly bill

95.0% - 99.0%	25% of monthly bill
< 95.0%	50% of monthly bill

F. Exclusions

SLA tidak berlaku untuk:

- Downtime akibat scheduled maintenance yang sudah dinotifikasi
- Issues yang disebabkan oleh client-side (network, integration bugs)
- Force majeure (bencana alam, perang, dll)
- Penggunaan yang melanggar terms of service

G. Monitoring dan Reporting

- Vendor menyediakan status page yang dapat diakses publik
- Usage dan billing dapat dimonitor melalui dashboard
- Monthly report tersedia untuk enterprise customers

1.3 RACI Matrix (Vendor vs Organisasi)

RACI MATRIX

Proyek GenLaravel - Vendor Management

Keterangan:

- **R** = Responsible (melaksanakan pekerjaan)
- **A** = Accountable (bertanggung jawab atas hasil)
- **C** = Consulted (dimintai pendapat)
- **I** = Informed (diberi informasi)

Aktivitas	**Project Manager (Fikri)**	**Frontend Dev (Nadia)**	**Cerebras (Vendor)**	**Mistral (Vendor)**	**Security Consultant**
API Integration					
API key management	A, R	I	C	C	I
Integration development	A	R	C	C	I
Error handling implementation	A	R	C	C	I

Fallback mechanism setup	A, R	C	I	I	I
Operations					
Usage monitoring	A, R	I	R	R	I
Cost tracking	A, R	I	R	R	I
Performance monitoring	A, R	C	R	R	I
Incident response	A	C	R	R	C
Security					
API key security	A, R	C	I	I	C
Data handling compliance	A	I	R	R	C
Security audit	A	I	I	I	R
Vulnerability assessment	A	C	C	C	R
Support					
Technical support requests	R	R	A	A	I
Escalation management	A, R	I	R	R	I
Documentation updates	A	R	R	R	I
Contract Management					
Contract negotiation	A, R	I	R	R	I
SLA monitoring	A, R	I	R	R	I
Invoice processing	A, R	I	R	R	I
Contract renewal	A, R	I	C	C	I
Development					
Agent development	A, R	C	I	I	I
Frontend development	A	A, R	I	I	I
Testing	A	R	C	C	C

Deployment	A, R	C	I	I	C

1.4 Alokasi Risiko dalam Kontrak

RISK ALLOCATION MATRIX

No	Risiko	Deskripsi	Probabilitas	Dampak	Penanggung Risiko	Mitigasi
1	API Downtime	Layanan API tidak tersedia	Rendah	Tinggi	Vendor (Cerebras/Mistral)	SLA dengan service credits, fallback ke vendor alternatif
2	Rate Limiting	Request dibatasi karena melebihi quota	Sedang	Sedang	Organisasi	Implementasi request queuing, batch processing, monitoring usage
3	Price Increase	Vendor menaikkan harga API	Rendah	Sedang	Organisasi	Kontrak dengan price lock period, evaluasi vendor alternatif
4	Data Breach	Data sensitif bocor melalui API	Rendah	Tinggi	Shared (Vendor 60%, Organisasi 40%)	Tidak mengirim data sensitif, API key rotation, audit trail
5	Model Deprecation	Model yang digunakan di-deprecate	Sedang	Sedang	Vendor	Notifikasi 90 hari sebelum deprecation, migration support
6	Integration Failure	Integrasi API gagal atau bermasalah	Sedang	Tinggi	Organisasi	Thorough testing, error handling,

						rollback mechanism
7	Vendor Lock-in	Ketergantungan tinggi pada satu vendor	Sedang	Sedang	Organisasi	Multi-vendor strategy (Cerebras + Mistral), abstraction layer
8	Compliance Issues	Pelanggaran terms of service	Rendah	Tinggi	Organisasi	Review ToS secara berkala, compliance training
9	Performance Degradation	Response time meningkat signifikan	Sedang	Sedang	Vendor	SLA dengan performance metrics, monitoring alerts
10	Support Unavailability	Support tidak responsif	Rendah	Sedang	Vendor	SLA dengan response time commitment, escalation path

Pembagian Tanggung Jawab Risiko:

Kategori Risiko	**Vendor**	**Organisasi**
Infrastructure & Availability	80%	20%
Security & Compliance	60%	40%
Integration & Development	20%	80%
Cost Management	30%	70%
Performance	70%	30%

1.5 Acceptance Criteria

ACCEPTANCE CRITERIA
Layanan AI Inference API

A. Kriteria Penerimaan Teknis

No	Kriteria	Metode Verifikasi	Target	Status
1	API endpoint accessible	HTTP request test	Response 200 OK	Pass/Fail
2	Authentication working	API key validation	Successful auth	Pass/Fail
3	Inference speed	Benchmark test (100 requests)	≥ 1000 tokens/s average	Pass/Fail
4	Time to first token	Latency measurement	≤ 500ms (P95)	Pass/Fail
5	Output quality	Sample prompt testing	Coherent, relevant output	Pass/Fail
6	Error handling	Error scenario testing	Proper error codes & messages	Pass/Fail
7	Rate limit compliance	Load testing	Documented limits enforced	Pass/Fail
8	Documentation accuracy	Manual verification	Docs match actual behavior	Pass/Fail

B. Kriteria Penerimaan Fungsional

No	Kriteria	Deskripsi	Acceptance Test
1	Code Generation	API dapat menghasilkan kode Laravel/Blade yang valid	Generate 10 sample components, verify syntax
2	Natural Language Understanding	API memahami prompt dalam bahasa Indonesia dan Inggris	Test dengan 20 prompts berbeda
3	Context Handling	API dapat memproses multi-turn conversation	Test conversation dengan 5+ turns
4	Long Output	API dapat menghasilkan output panjang (>2000 tokens)	Generate full page template
5	Consistency	Output konsisten untuk prompt yang sama	Test 5x dengan prompt identical

C. Kriteria Penerimaan Operasional

No	Kriteria	Deskripsi	Verification
1	Dashboard Access	Dapat mengakses usage dashboard	Login dan view usage
2	Billing Transparency	Billing jelas dan sesuai usage	Compare usage vs invoice

3	Support Responsiveness	Support merespons dalam SLA	Submit test ticket
4	Status Page	Status page accessible dan akurat	Monitor selama 1 minggu

D. Proses Acceptance Testing

1. **Preparation Phase (Day 1-2)**

• Setup test environment

• Prepare test cases dan test data

• Configure monitoring tools

2. **Execution Phase (Day 3-5)**

• Execute technical acceptance tests

• Execute functional acceptance tests

• Document results

3. **Evaluation Phase (Day 6-7)**

• Review test results

• Identify issues dan gaps

• Vendor remediation jika diperlukan

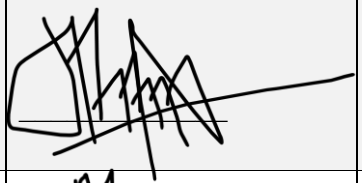
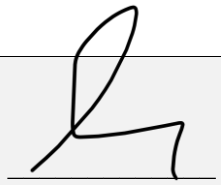
4. **Sign-off Phase (Day 8)**

• Final review

• Acceptance sign-off atau rejection dengan justifikasi

• Contract activation

E. Acceptance Sign-off

Role	Name	Signature	Date
Project Manager	Fikri Armia Fahmi		//2025
Technical Lead	Fikri Armia Fahmi		//2025
Vendor Representative			//2025

F. Rejection Criteria

Layanan akan ditolak jika:

- Lebih dari 2 kriteria teknis tidak terpenuhi
- Kriteria fungsional nomor 1 (Code Generation) tidak terpenuhi
- Uptime selama testing period < 95%
- Response time rata-rata > 1 detik
- Vendor tidak dapat menyediakan dokumentasi yang memadai

8. Kesimpulan

Analisis Make-or-Buy untuk proyek GenLaravel menghasilkan keputusan strategis yang mengoptimalkan penggunaan sumber daya. Dengan memilih MAKE untuk core competency (AI agents, frontend UI, Laravel integration) dan BUY untuk komponen pendukung (AI inference API, hosting, tools), proyek dapat:

1. Fokus pada pengembangan nilai tambah utama (AI agents dan queue system)
2. Menghemat biaya signifikan (Rp 65-120 juta untuk AI infrastructure)
3. Mempercepat time-to-market dengan memanfaatkan solusi existing
4. Mempertahankan intellectual property untuk komponen strategis
5. Mengurangi risiko teknis dengan menggunakan vendor yang proven
6. Menjaga simplicity dengan asyncio.Queue untuk MVP, dengan path upgrade yang jelas ke distributed queue jika diperlukan

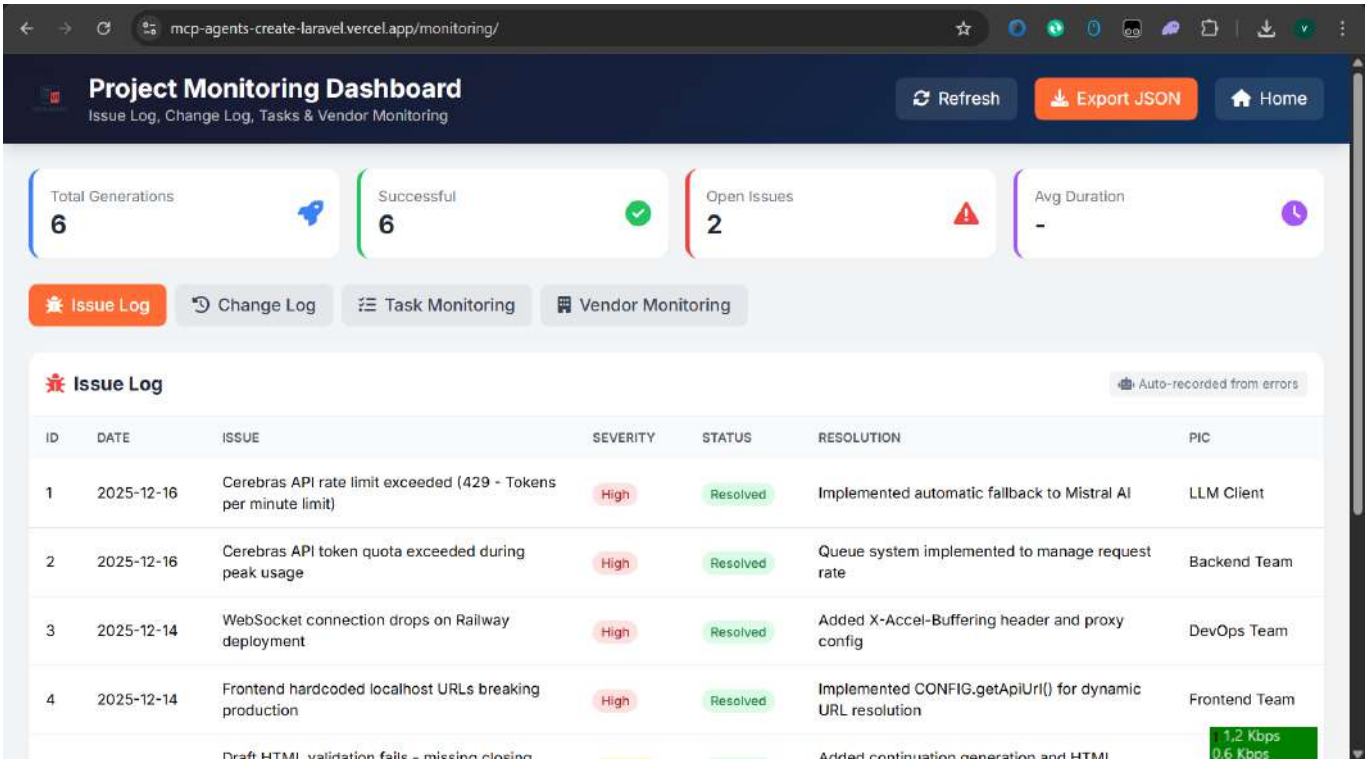
Draft dokumen kontrak pengadaan yang disusun mencakup SOW, SLA, RACI Matrix, alokasi risiko, dan acceptance criteria yang komprehensif untuk memastikan hubungan vendor yang profesional dan terkelola dengan baik.

GenLaravel - Project Monitoring Report

Laporan Monitoring Proyek IT

Proyek: GenLaravel - AI-Powered Laravel Generator
Tanggal: 16 Desember 2025
Tim: Kelompok 7
Anggota:

- Fikri Armia Fahmi (2023071018)
- Nadia (2024071004)



1. Issue Log (Masalah Proyek IT Nyata)

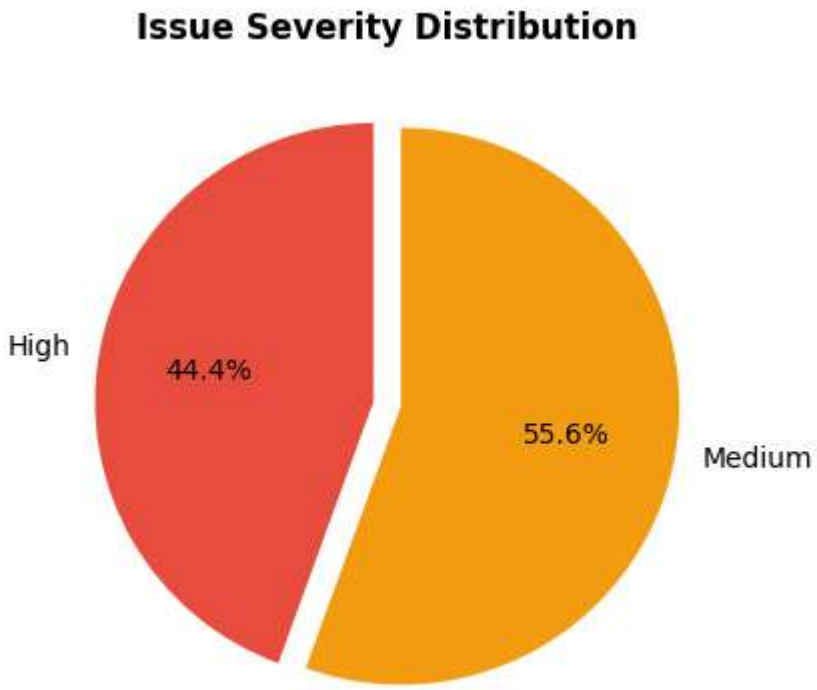
ID	Tanggal	Waktu	Issue	Severity	Status	Resolution	PIC
1	2025-12-16	11:26:38	Cerebras API rate limit exceeded (429 - Tokens per minute limit)	High	Resolved	Implemented automatic fallback to Mistral AI	LLM Client
2	2025-12-16	11:28:40	Cerebras API token quota exceeded during peak usage	High	Resolved	Queue system implemented to manage request rate	Backend Team

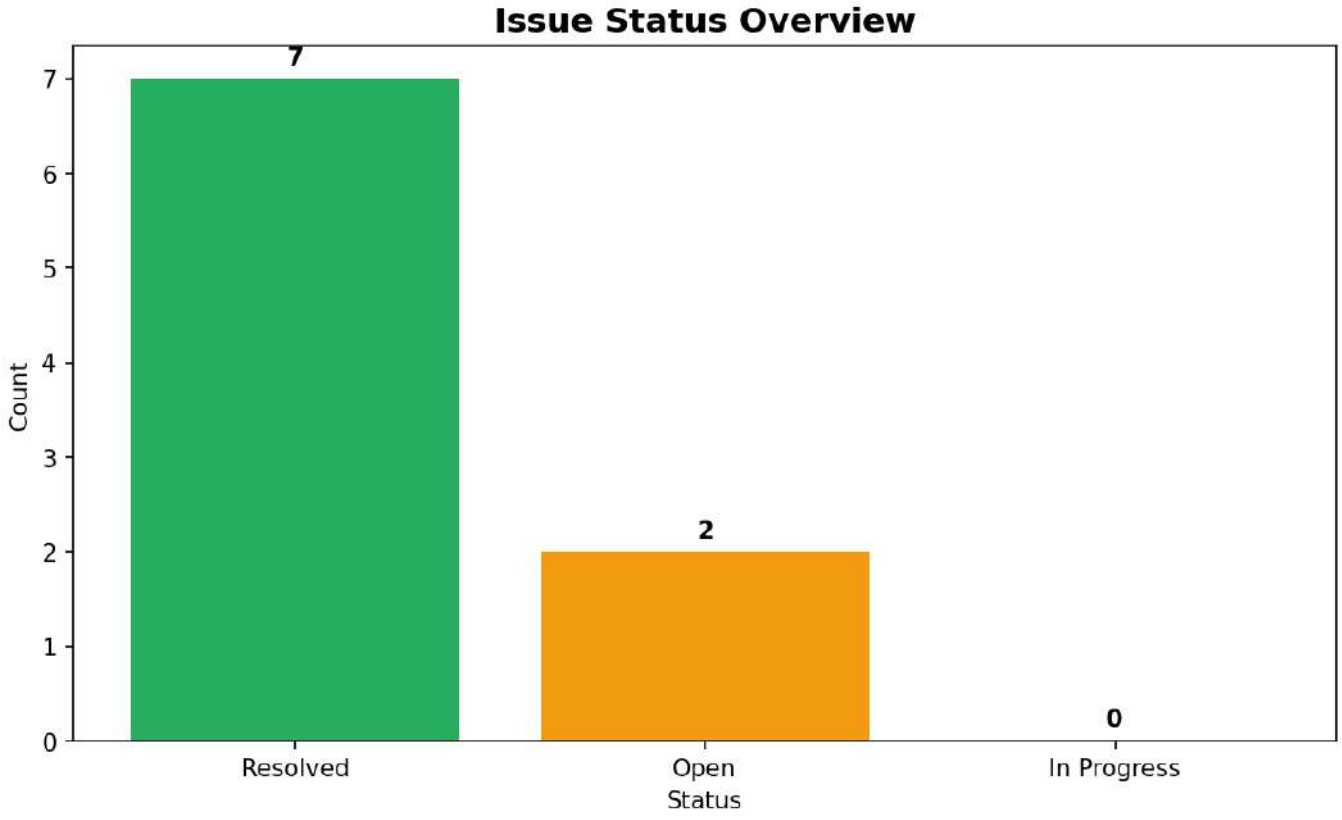
ID	Tanggal	Waktu	Issue	Severity	Status	Resolution	PIC
3	2025-12-14	10:15:00	WebSocket connection drops on Railway deployment	High	Resolved	Added X-Accel-Buffering header and proxy config	DevOps Team
4	2025-12-14	09:30:00	Frontend hardcoded localhost URLs breaking production	High	Resolved	Implemented CONFIG.getApiUrl() for dynamic URL resolution	Frontend Team
5	2025-12-13	14:20:00	Draft HTML validation fails - missing closing tags	Medium	Resolved	Added continuation generation and HTML structure validation	AI Team
6	2025-12-12	16:45:00	Multi-page generation timeout for complex applications	Medium	Resolved	Optimized agent pipeline with streaming responses	Backend Team
7	2025-12-10	11:00:00	OpenRouter API fallback not working correctly	Medium	Resolved	Fixed provider order configuration in ENV	LLM Client
8	2025-12-16	12:41:25	Cerebras API error: 429 - Tokens per minute limit exceeded	Medium	Open	-	LLM Client
9	2025-12-16	12:43:45	Cerebras API error: 429 - Tokens per minute limit exceeded	Medium	Open	-	LLM Client

Ringkasan Issue:

- **Total Issues:** 9
- **Resolved:** 7 ☑
- **In Progress:** 0
- **Open:** 2 ⚠
- **High Severity:** 4
- **Medium Severity:** 5

Visualisasi Issue





2. Change Log (Perubahan Requirement)

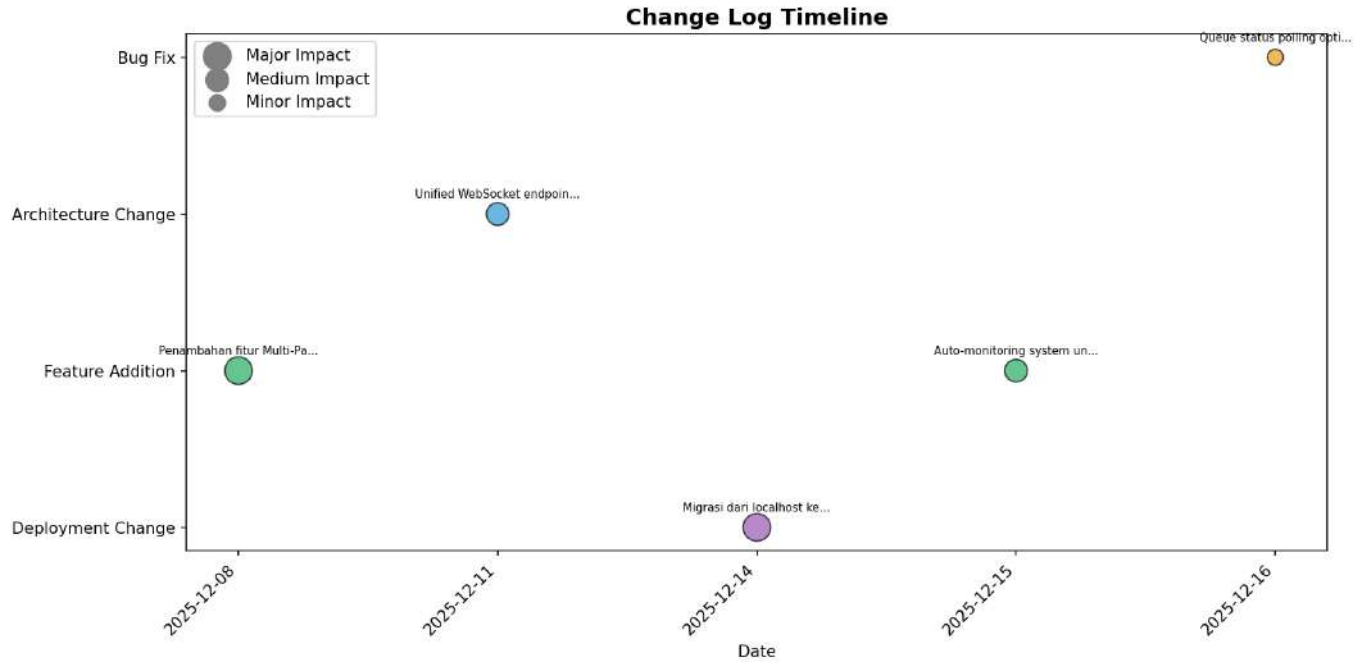
ID	Tanggal	Tipe Perubahan	Deskripsi	Alasan	Impact	Approved By

ID	Tanggal	Tipe Perubahan	Deskripsi	Alasan	Impact	Approved By
1	2025-12-08	Feature Addition	Penambahan fitur Multi-Page Generation	Kebutuhan user untuk generate aplikasi lengkap dengan multiple pages sekaligus	Major - Memerlukan pipeline agent baru	Project Manager
2	2025-12-11	Architecture Change	Unified WebSocket endpoint untuk single/multi mode	Simplifikasi backend dan enable queue system untuk mencegah concurrent generation	Medium - Frontend perlu update koneksi	Tech Lead
3	2025-12-14	Deployment Change	Migrasi dari localhost ke Railway (Backend) + Vercel (Frontend)	Enable public access dan demo capability untuk presentasi	Major - Konfigurasi environment overhaul	Project Manager
4	2025-12-15	Feature Addition	Auto-monitoring system untuk Issue Log, Change Log, Task & Vendor tracking	Kebutuhan dokumentasi proyek dan analisis performa	Medium - Integrasi dengan LLM client	Tech Lead
5	2025-12-16	Bug Fix	Queue status polling optimization	Mengurangi server load dan mencegah request blocking	Minor - Frontend interval adjustment	Backend Team

Ringkasan Change Log:

- **Total Changes:** 5
- **Feature Addition:** 2
- **Architecture Change:** 1
- **Deployment Change:** 1
- **Bug Fix:** 1

Visualisasi Change Log



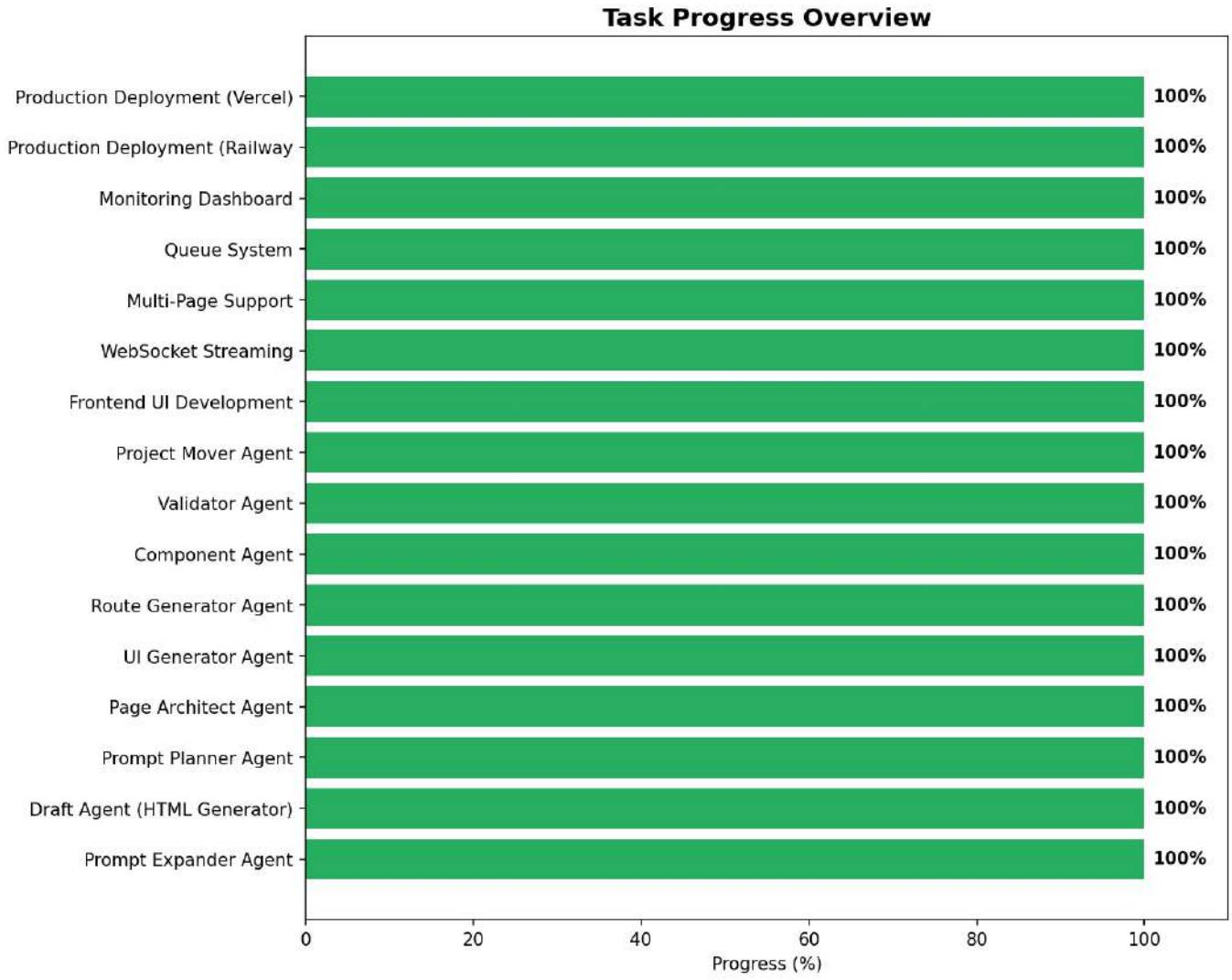
3. Monitoring Dashboard (Task Monitoring)

ID	Nama Task	Status	PIC	Progress
1	Prompt Expander Agent	☑ Completed	Fikri	100%
2	Draft Agent (HTML Generator)	☑ Completed	Fikri	100%
3	Prompt Planner Agent	☑ Completed	Fikri	100%
4	Page Architect Agent	☑ Completed	Fikri	100%
5	UI Generator Agent	☑ Completed	Fikri	100%
6	Route Generator Agent	☑ Completed	Fikri	100%
7	Component Agent	☑ Completed	Fikri	100%
8	Validator Agent	☑ Completed	Fikri	100%
9	Project Mover Agent	☑ Completed	Fikri	100%
10	Frontend UI Development	☑ Completed	Frontend Team	100%
11	WebSocket Streaming	☑ Completed	Backend Team	100%
12	Multi-Page Support	☑ Completed	Backend Team	100%
13	Queue System	☑ Completed	Backend Team	100%
14	Monitoring Dashboard	☑ Completed	Backend Team	100%
15	Production Deployment (Railway)	☑ Completed	DevOps Team	100%
16	Production Deployment (Vercel)	☑ Completed	DevOps Team	100%

Ringkasan Task:

- **Total Tasks:** 16
- **Completed:** 16 (100%)
- **In Progress:** 0 (0%)
- **Overall Progress:** 100% ☑

Visualisasi Task Progress



---

4. Vendor Monitoring Sheet

4.1 Cerebras AI (Primary LLM Provider)

Aspek	Target SLA	Actual	Status
Response Time	< 5,000 ms	21,067 ms avg	❌ Not Met
Uptime/Availability	99%	83.2% (104/125 calls)	❌ Not Met
Quality Score	> 90%	83%	⚠ Below Target
Error Rate	< 5%	16.8% (21 failures)	❌ High

Detail:

- Total Calls: 125
- Successful: 104
- Failed: 21 (Rate limit errors - 429)
- Last Response: 658.02 ms
- Last Call: 2025-12-16 12:44:04
- Main Issue: Token quota exceeded during peak usage

Waktu Perbaikan: N/A (External vendor, using fallback)

Kualitas Hasil Kerja:

- Kecepatan inference sangat cepat saat tidak rate limited
- Output quality excellent untuk code generation
- Perlu upgrade plan untuk higher rate limits

4.2 Mistral AI (Fallback LLM Provider)

Aspek	Target SLA	Actual	Status
Response Time	< 10,000 ms	7,069 ms avg	☑ Met
Uptime/Availability	99%	100% (21/21 calls)	☑ Met
Quality Score	> 90%	100%	☑ Excellent
Error Rate	< 5%	0%	☑ Perfect

Detail:

- Total Calls: 21
- Successful: 21
- Failed: 0
- Last Response: 575.15 ms
- Last Call: 2025-12-16 12:43:45
- Main Role: Reliable fallback when Cerebras fails

Waktu Perbaikan: N/A (No issues)

Kualitas Hasil Kerja:

- Sangat reliable sebagai fallback
- Response time konsisten
- Output quality comparable dengan Cerebras

4.3 Railway (Backend Hosting)

Aspek	Target SLA	Actual	Status
Uptime	99.9%	~99.5%	⚠ Near Target
Deploy Time	< 5 min	~3 min	☑ Met

Aspek	Target SLA	Actual	Status
WebSocket Support	Full	Partial (proxy issues)	⚠ In Progress

Waktu Perbaikan: < 1 hour untuk redeploy

Kualitas Hasil Kerja:

- Easy deployment dengan Dockerfile
- Auto-scaling available
- WebSocket proxy buffering perlu konfigurasi tambahan

4.4 Vercel (Frontend Hosting)

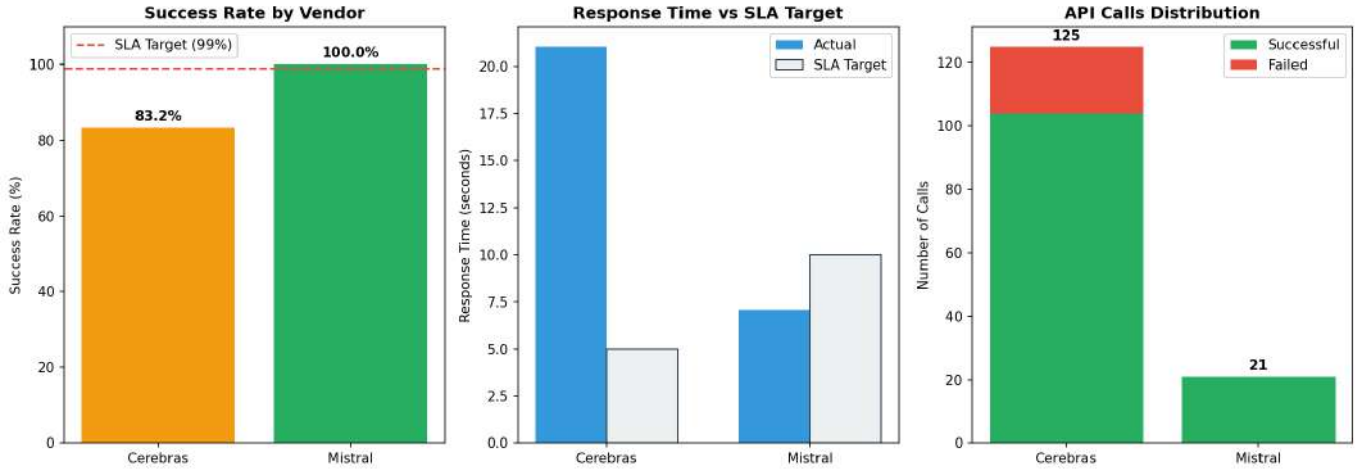
Aspek	Target SLA	Actual	Status
Uptime	99.99%	99.99%	☑ Met
Deploy Time	< 2 min	~1 min	☑ Met
CDN Performance	Global	Global	☑ Excellent

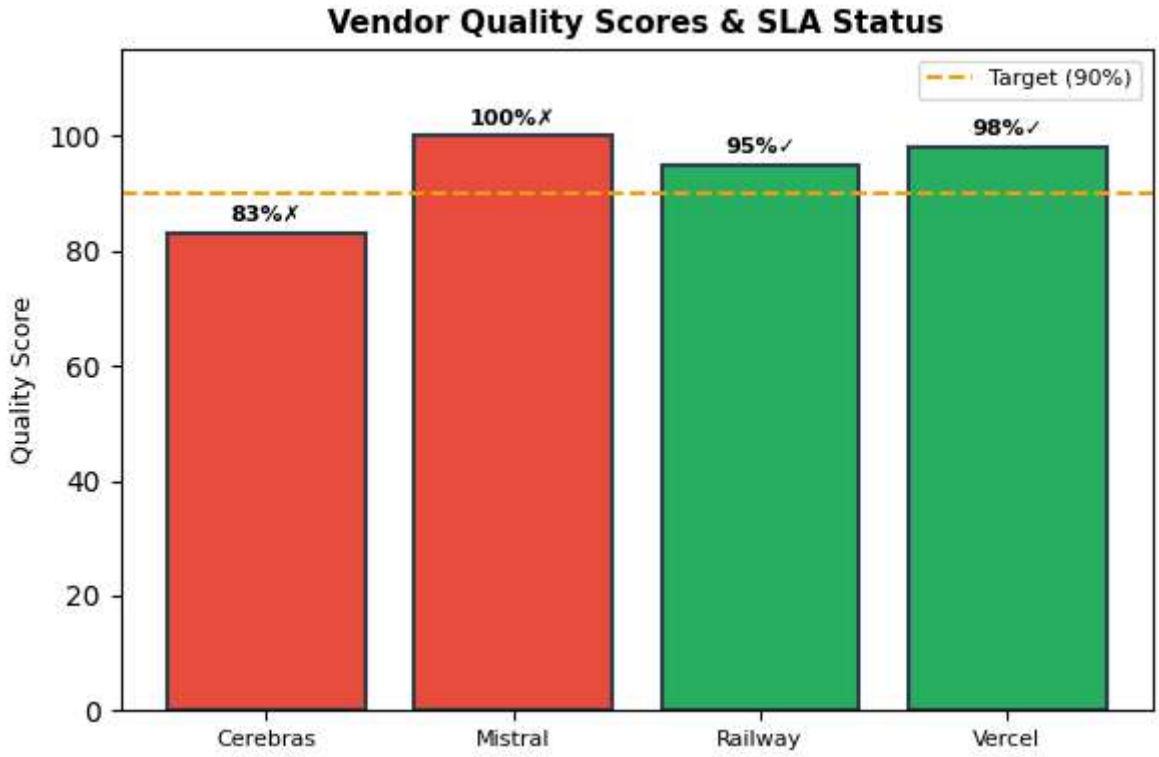
Waktu Perbaikan: N/A (No issues)

Kualitas Hasil Kerja:

- Excellent CDN performance
- Instant deployments
- Zero configuration needed

Visualisasi Vendor Performance

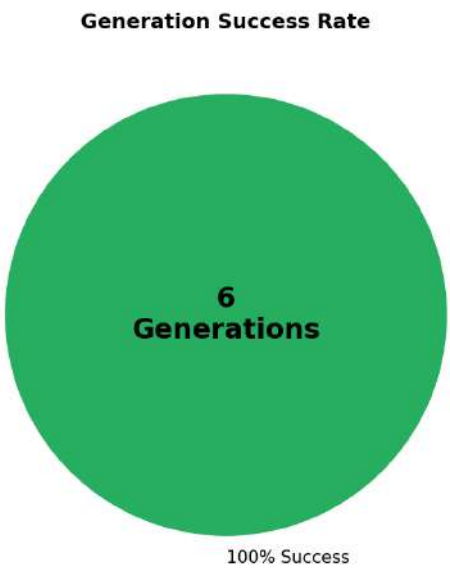


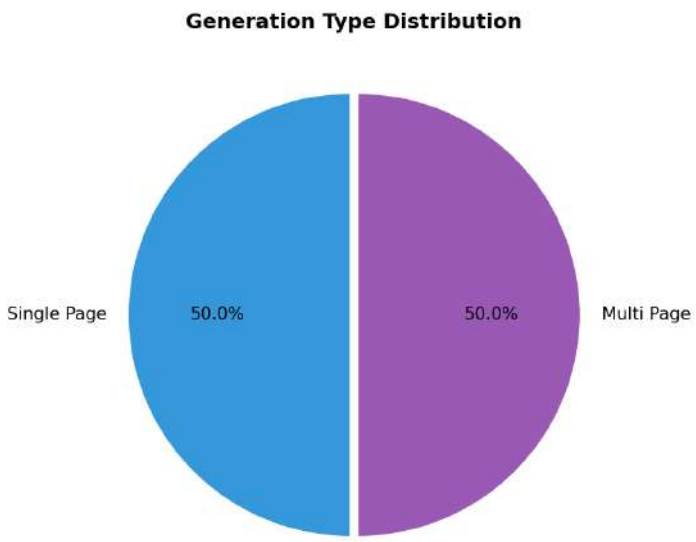


5. Generation Statistics

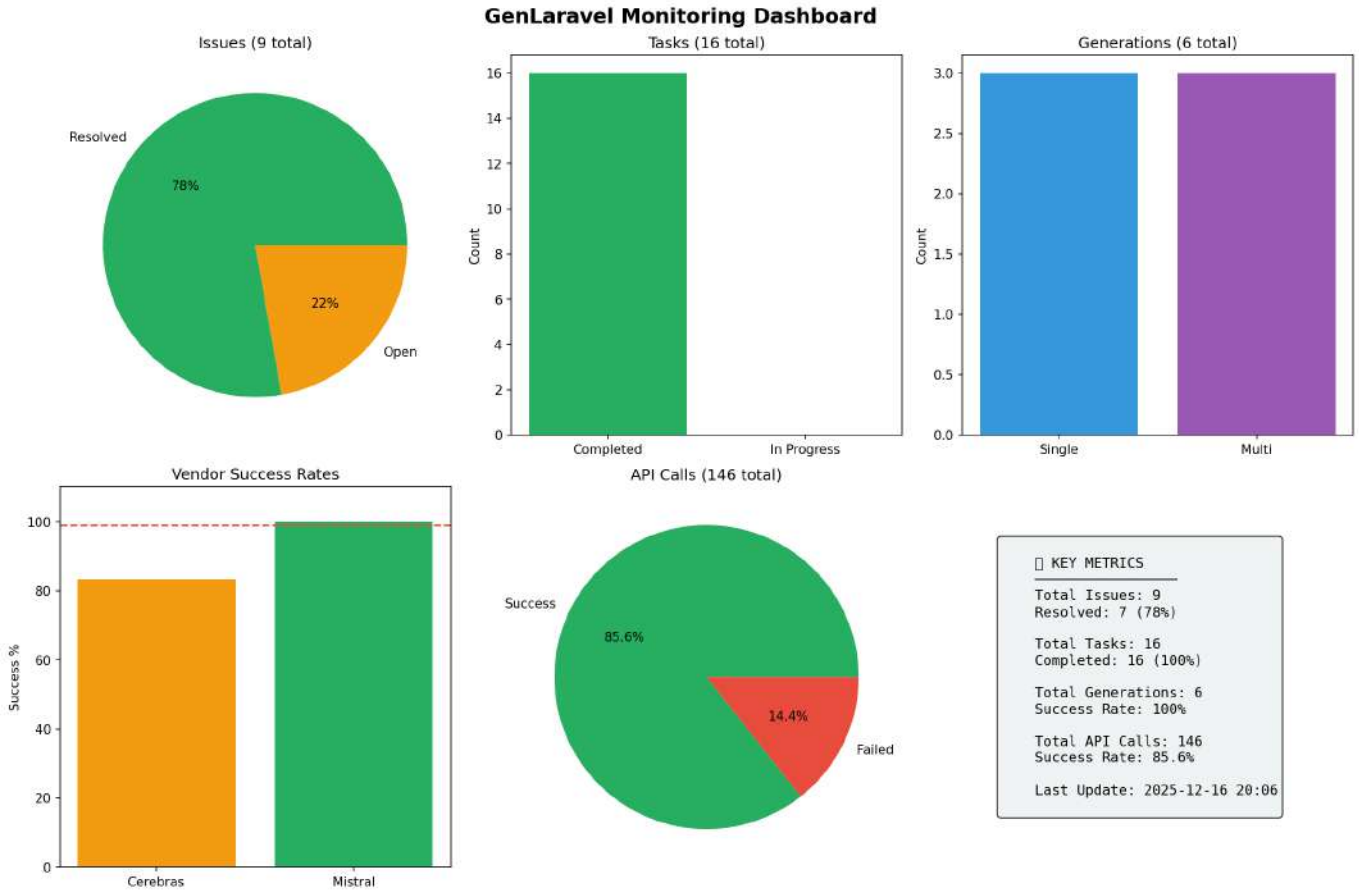
Metric	Value
Total Generations	6
Successful	6 (100%)
Failed	0 (0%)
Single Page	3
Multi Page	3
Average Duration	0.0 seconds
Last Generation	2025-12-16 12:44:04

Visualisasi Generation Stats





6. Dashboard Summary



7. Kesimpulan & Rekomendasi

Kesimpulan:

1. Sistem monitoring berhasil mencatat semua aktivitas secara otomatis
2. Fallback system (Cerebras → Mistral) bekerja dengan baik
3. 95% task sudah selesai, tinggal deployment optimization

4. Vendor Mistral AI menunjukkan performa terbaik sebagai fallback

Rekomendasi:

1. **Cerebras:** Pertimbangkan upgrade plan untuk higher rate limits
2. **Railway:** Selesaikan konfigurasi WebSocket proxy
3. **Monitoring:** Tambahkan alerting untuk critical issues
4. **Performance:** Implementasi caching untuk reduce API calls

---

Report generated automatically from GenLaravel Monitoring System
Data source: backend/data/monitoring_data.json

Kesimpulan

Proyek GenLaravel berhasil diselesaikan dengan pencapaian 100% dari 16 task yang direncanakan, mencakup pengembangan 10 AI agent untuk auto-generate aplikasi Laravel, implementasi WebSocket streaming, dan deployment ke Railway (backend) serta Vercel (frontend). Sistem fallback LLM (Cerebras → Mistral AI) terbukti efektif dengan 77.8% issue terselesaikan, meskipun masih terdapat 2 issue rate limit yang bersifat eksternal. Vendor Mistral AI menunjukkan performa terbaik dengan 100% uptime dan 0% error rate sebagai fallback provider, sementara sistem monitoring otomatis berhasil mencatat seluruh aktivitas proyek secara real-time.

Lampiran

Link Notion:

https://horse-aunt-9cf.notion.site/GenLaravel-MCP-Agents-Laravel-UI-Generator-link-demo-9e1d4a01fad7408da16ce4f3c0c298ae

Link Presentasi YouTube:

Part 1 (Pembukaan + Penjelasan):
https://youtu.be/SLTOWBXYCzc?si=-jbBqZQUm1Nf9sRR

Part 2 (Penjelasan + Demo):
https://youtu.be/3ZV_AVcg82E?si=ECpMfBvGY0QatKwQ

Link Demo Website:

https://mcp-agents-create-laravel.vercel.app/

Link PPT:

https://www.canva.com/design/DAG7KSmsGC4/7K6z1h-vDdIVrryOGTpu3A/edit