

**Kelompok 7**

# **GENLARAVEL - MCP AGENTS LARAVEL UI GENERATOR**

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



Start : 18 September 2025

# INISIASI PROYEK

## MCP Agents Laravel UI Generator

python v3.7+ laravel v8+ license MIT build passing code style black agents 10 AI Mistral platform windows | linux | macOS

An intelligent multi-agent system that automatically generates Laravel UI components from natural language prompts using AI agents.

### 🚀 Features

- AI-Powered UI Generation: Transform natural language descriptions into complete Laravel Blade components
- Multi-Agent Architecture: Specialized agents handle different aspects of the generation process
- Interactive Draft Preview: Preview HTML drafts in browser before generating Laravel components
- Automatic Laravel Integration: Seamlessly moves generated components to your Laravel project
- Component Validation: Built-in validation system ensures code quality



# GenLaravel

**SPONSOR :**  
**INTERNAL DEVELOPMENT TEAM**

**PROJECT MANAGER:**  
**FIKRI ARMIA FAHMI**

**FRONTEND DEVELOPER:**  
**NADIA**

# INISIASI PROYEK



**GenLaravel**

PROJECT MCP AGENTS LARAVEL UI GENERATOR ADALAH INISIATIF UNTUK MENGELOMONGKAN SISTEM AI MULTI-AGENT YANG DAPAT MENGOTOMATISASI PEMBUATAN KOMPONEN UI LARAVEL DARI NATURAL LANGUAGE PROMPTS. SYSTEMINI DIRANCANG UNTUK MEMPERCEPAT DEVELOPMENT WORKFLOW, MENGURANGI REPETITIVE TASKS, DAN MENINGKATKAN PRODUKTIVITAS DEVELOPER, KHUSUSNYA MAHASISWA DAN DEVELOPER PEMULA.

# Business Case



**GenLaravel**

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.

# Problem Statement



GenLaravel

- 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

# Opportunity



GenLaravel

- MENGOTOMATISASI 70-80% PROSES PEMBUATAN UI COMPONENTS
- STANDARDISASI CODE QUALITY DAN STRUCTURE
- MEMPERCEPAT ONBOARDING DEVELOPER BARU
- MENINGKATKAN PRODUKTIVITAS TIM DEVELOPMENT

# 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.

# TUJUAN UTAMA



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

# STAKEHOLDER



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

# Feasibility Study

| Aspek Evaluasi | Status                                    | Detail   |
|----------------|---|--|
| Teknis         | <input checked="" type="checkbox"/> Layak | Python 3.10.11, Laravel 8.2, Cerebras SDK, Mistral SDK       |
| Ekonomi        | <input checked="" type="checkbox"/> Layak | Open-source MIT License, biaya server minimal                |
| Operasional    | <input checked="" type="checkbox"/> Layak | Developer dapat menjalankan script Python, integrasi Laravel |
| Hukum          | <input checked="" type="checkbox"/> Layak | MIT License, sesuai standar open-source                      |
| Waktu          | <input checked="" type="checkbox"/> Layak | Estimasi 12-16 minggu development agile                      |

## Aspek Teknis

- Stack teknologi yang matang dan terbukti (Python, Laravel)
- SDK yang tersedia (Cerebras SDK, Mistral SDK)
- Infrastruktur development yang memadai

## Aspek Ekonomi

- Menggunakan lisensi open-source (MIT)
- Biaya operasional minimal (server cloud)
- ROI positif melalui peningkatan produktivitas developer

## Aspek Operasional

- Tim developer memiliki skill Python dan Laravel
- Integrasi dengan workflow development yang ada
- Dukungan dokumentasi dan maintenance

## Proyek Layak untuk Di laksanakan

# Feasibility Study

| Aspek Evaluasi | Status                                    | Detail   |
|----------------|---|--|
| Teknis         | <input checked="" type="checkbox"/> Layak | Python 3.10.11, Laravel 8.2, Cerebras SDK, Mistral SDK       |
| Ekonomi        | <input checked="" type="checkbox"/> Layak | Open-source MIT License, biaya server minimal                |
| Operasional    | <input checked="" type="checkbox"/> Layak | Developer dapat menjalankan script Python, integrasi Laravel |
| Hukum          | <input checked="" type="checkbox"/> Layak | MIT License, sesuai standar open-source                      |
| Waktu          | <input checked="" type="checkbox"/> Layak | Estimasi 12-16 minggu development agile                      |

## Aspek Hukum

- Lisensi MIT memungkinkan penggunaan komersial
- Tidak ada konflik dengan intellectual property
- Sesuai dengan standar open-source

## Aspek Waktu

- Timeline 12-16 minggu realistik untuk scope project
- Metodologi agile memungkinkan iterasi cepat
- Milestone yang jelas dan terukur

**Proyek Layak untuk Di laksanakan**

# Main Deliverables

1. AI Agent Modules (10 agents)
2. Laravel UI Generator (Blade files, layouts, components)
3. Validation & Integration System
4. Front-end Preview UI (HTML/css/js)
5. Agent Backend Services & API
6. Documentation & User Guide



**GenLaravel**

# 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  |

# Documentation Deliverables

| # | 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  |

# Human Resources

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

# 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         |
| <b>Total Estimated Cost</b> |              |          | <b>\$292</b> |

# EARNED VALUE MANAGEMENT (EVM)

## Data Proyek

- Budget at Completion (BAC): Rp 35.000.000
- Total Story Point: 350 SP (setelah scope adjustment)
- Durasi: 14 Sprint

## Planned Value (PV)

- Target sampai Sprint ke-13: 93%
- PV: Rp 32.550.000

## Earned Value (EV)

- Progress aktual: 98,3%
- EV: Rp 34.405.000

## Actual Cost (AC)

Total biaya aktual: Rp 18.616.000

## Indikator Kinerja

- Cost Performance Index (CPI): 1,85
  - CPI > 1 → Proyek under budget
  - Biaya digunakan sangat efisien
  - Setiap Rp 1 biaya menghasilkan Rp 1,85 nilai pekerjaan
- Schedule Performance Index (SPI): 1,06
  - SPI > 1 → Proyek lebih cepat dari jadwal
  - Progres pekerjaan melampaui rencana sebesar 6%

## Variance

- Cost Variance (CV): +Rp 15.789.000
  - CV positif → Penghematan biaya
  - Tidak terjadi pembengkakan anggaran
- Schedule Variance (SV): +Rp 1.855.000
  - SV positif → Pekerjaan lebih cepat dari rencana
  - Tidak ada keterlambatan kritis

## Forecasting

- Estimate at Completion (EAC): Rp 18.937.622
- Estimate to Complete (ETC): Rp 321.622
- Variance at Completion (VAC): +Rp 16.062.378

# PROCUREMENT PROYEK

## Tools & Cloud

- Cerebras API (Qwen 3 32B)
- Mistral API (Codestral – fallback)
- Railway (Free Tier)
- Domain & SSL (.my.id)

## Jasa Profesional

- Laravel Expert Review
- Security Audit

## Peralatan

- Laptop tim (amortisasi)
- Software open-source & student license

## Strategi Procurement

- Pay-as-you-go
- Pemanfaatan free tier & student benefits
- Tidak ada pembelian aset baru

# HASIL MONITORING & CONTROLLING

## Metode Monitoring

- Evaluasi EVM per milestone
- Burn Down Chart per sprint
- Daily stand-up
- Sprint review & retrospective
- Tracking biaya aktual per sprint

## Hasil Monitoring Biaya

- CPI > 1 (1,85)
- Proyek under budget
- Efisiensi biaya tinggi

## Hasil Monitoring Jadwal

- SPI > 1 (1,06)
- Proyek ahead of schedule
- Tidak ada keterlambatan kritis

## Temuan Monitoring

- Scope creep +20 SP (Sprint 6)
- Optimasi API menurunkan biaya token 30%
- Penyesuaian velocity sprint ( $40 \rightarrow 32$  SP)
- Risiko proyek rendah

## Hasil Monitoring Jadwal

- Schedule: Ahead of schedule
- Cost: Under budget
- Scope: MVP selesai
- Quality: Testing coverage > 90%
- Risk: Rendah

# **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**

# Internal Stakeholders

| 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      |

# External Stakeholders

| 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  |

# Project Timeline & 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            |
| M7: Documentation Complete | All docs completed                   | Week 13     | Complete documentation  |
| M8: Production Release     | v1.0 launch                          | Week 14     | Production-ready system |

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

# Project Objectives

## Primary Objectives

### 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

# Project Objectives

## 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

# Project Planning & Timeline

## Phase 1: Foundation (Sprint 1-2)

- Sprint 1-2: Project setup, infra, repo templates, env, CI pipelines

## Phase 2: Core Development (Sprint 3-8)

- 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)

## Phase 3: Integration (Sprint 9-11)

- Sprint 9-10: Route Agent & Project Integrator (integration dengan Laravel)
- Sprint 11: Validator Agent (syntax, structure, acceptance criteria)

## Phase 4: Finalization (Sprint 12-14)

- Sprint 12: Agent backend polishing – queueing, retries, logging, security
- Sprint 13: Documentation, examples, repo templates, demo app
- Sprint 14: Final testing, bugfix, release / handover

# Deliverables per Sprint

## Sprint 1-2: Infrastructure

- [x] Repository setup
- [x] Development environment
- [x] CI/CD pipelines
- [x] Project templates

## Sprint 3-4: Prompt Processing

- [x] Prompt Expander Agent
- [x] Planner Agent
- [ ] Backend API foundation

## Sprint 5-6: Draft Generation

- [x] Draft Generator Agent
- [x] HTML preview system
- [x] Frontend scaffold

## Sprint 7-8: UI Components

- [x] Layout Generator Agent
- [x] Component Generator Agent
- [x] Enhanced preview UI
- [x] Interactive controls

## Sprint 9-10: Laravel Integration

- [x] Route Generator Agent
- [x] Project Integrator Agent
- [x] File system integration
- [x] Laravel project updates

## Sprint 11: Validation

- [x] Validator Agent
- [x] Syntax checking
- [x] Structure validation
- [x] Acceptance criteria checks

## Sprint 12: Backend Polish

- [x] Queue management
- [x] Error handling & retries
- [x] Logging system
- [x] Security enhancements

## Sprint 13: Documentation

- [x] README documentation
- [x] User guides
- [x] API documentation
- [x] Example projects
- [ ] Presentation videos

## Sprint 14: Release

- [x] Final testing
- [x] Bug fixes
- [x] Performance optimization
- [x] Release preparation
- [ ] Handover documentation

# Technical Architecture

## Tech Stack

### Frontend

- HTML5, CSS3 (Tailwind atau plain CSS)
- Vanilla JS atau framework kecil (Alpine.js / HTMX) untuk interaktivitas
- Responsive design dengan accessibility basics

### Backend Agents

- Python 3.10.11 untuk agent modules
- FastAPI atau Flask untuk HTTP endpoints
- Redis/RQ atau Celery untuk queueing
- Cerebras SDK & Mistral SDK untuk agent prompts & inference

### Target Integration

- Laravel 8.2 sebagai target project
- Filesystem staging area untuk file management
- GitHub Actions untuk CI/CD

# AI Agent Architecture

## Agent Modules (10 Agents)

### 1. Prompt Expander

Fungsi:

- Expand prompt user
- Ekstrak requirement
- Susun spesifikasi UI

### 2. Prompt Planner

Fungsi:

- Analisis kebutuhan
- Pecah komponen UI
- Rancang alur halaman

### 3. Draft Agent

Fungsi:

- Generate HTML draft
- Buat UI awal
- Siapkan preview

### 4. Page Architect

Fungsi:

- Susun struktur halaman
- Atur layout section
- Kelola hierarki UI

### 5. Component Agent

Fungsi:

- Identifikasi komponen
- Buat Blade component
- Siapkan file komponen

### 6. UI Generator

Fungsi:

- Generate Blade view
- Integrasi komponen
- Jaga konsistensi UI

### 7. Layout Generator

Fungsi:

- Buat layout utama
- Atur yield section
- Susun wrapper app

### 8. Route Agent

Fungsi:

- Tentukan endpoint
- Generate routes
- Sinkron halaman

### 9. Validator Agent

Fungsi:

- Validasi syntax Blade
- Cek struktur UI
- Pastikan siap deploy

### 10. Project Mover

Fungsi:

- Copy file ke Laravel
- Integrasi view & route
- Finalisasi project

# System Architecture

## API Layer

- **Endpoints:**
  - /generate-draft
  - /status/{job}
  - /approve/{job}

## Worker Layer

- Python workers untuk agent execution
- Queue management dengan Redis/Celery
- Job retry mechanism

## Storage Layer

- Staging filesystem (output/)
- SQLite/PostgreSQL untuk job metadata
- Structured logging untuk monitoring

## Security

- API keys untuk access control
- Rate limiting untuk request throttling
- Input validation untuk security
- Secrets management via .env

# Team Structure

## Core Team Members

### Project Manager & Backend Engineer

Fikri Armia Fahmi (2023071018)

- Project management & coordination
- Backend agent development
- API orchestration
- System architecture

### Frontend Developer

Nadia (2024071004)

- Frontend UI development (HTML/CSS/JS)
- Preview interface design
- User experience optimization
- Responsive design implementation

# Team Structure

## Roles & Responsibilities

### Project Manager Responsibilities

- Sprint planning & coordination
- Stakeholder communication
- Progress tracking & reporting
- Risk management
- Quality assurance

### Backend Engineer Responsibilities

- AI agent module development
- API endpoint creation
- Queue management system
- Integration with Laravel projects
- Database design & optimization

### Frontend Developer Responsibilities

- Interactive preview UI creation
- User interface design
- Frontend integration with backend APIs
- Cross-browser compatibility
- Accessibility implementation

# Stakeholders

## Primary Stakeholders

- Sponsor: Internal Research & Development Team
- Development Team: AI/ML Engineers, Laravel Developers
- End Users: Laravel developers & software houses

## Engagement Strategy

- Weekly sprint reviews dengan demo preview UI
- Monthly stakeholder reports via GitHub Project Board
- Issues & backlog management di GitHub
- Demo sessions untuk sample generated Blade files

# Tools & Resources

## Development Tools

- Version Control: GitHub
- Project Management: GitHub Project Board
- Communication: Weekly stand-ups
- Documentation: GitHub Wiki + Notion

## Technical Resources

- AI Service: Mistral API untuk agent inference
- Infrastructure: Docker (optional) untuk containerization
- Testing: Unit tests, integration tests, E2E testing
- Monitoring: Structured logs, simple dashboard

## Budget Considerations

- Open-source MIT License – minimal licensing costs
- Mistral API usage – pay-per-use model
- Server costs – minimal untuk development
- Development time – 14 minggu × 2 developers

# Risk Management (Risk Assessment & Mitigation)

## ● High Impact Risks

### 1. Untrusted HTML Execution in Preview

**Risk:** HTML yang dihasilkan dapat mengeksekusi script berbahaya

**Impact:** Kerentanan keamanan, potensi kompromi sistem

#### Mitigation:

- Render dalam sanitized iframe
- Escape semua scripts
- Tidak mengeksekusi user-generated JavaScript
- Implementasi Content Security Policy

### 2. Agent Output Incorrect (Invalid Blade)

**Risk:** File Blade yang dihasilkan mengandung syntax errors atau struktur yang tidak valid

**Impact:** Aplikasi Laravel rusak, penundaan development

#### Mitigation:

- Validator agent dengan pemeriksaan komprehensif
- Unit testing untuk semua agents
- Validasi sample acceptance criteria
- Integrasi code linting

### 3. Integration Breaks User Laravel Project

**Risk:** Proses integrasi file merusak project Laravel yang sudah ada

**Impact:** Kehilangan data, perlu rollback project

#### Mitigation:

- Buat branch atau staging folder daripada overwrite langsung
- Sistem backup sebelum integrasi
- Implementasi rollback mechanism
- Version control berbasis Git

# Risk Management (Risk Assessment & Mitigation)

## ● Medium Impact Risks

### 4. Queue/Backlog Overload

**Risk:** Terlalu banyak request bersamaan membebani sistem

**Impact:** Sistem melambat, degradasi user experience

#### **Mitigation:**

- Rate-limit requests per user
- Scale workers berdasarkan demand
- Backpressure mechanism ke UI
- Queue monitoring & alerts

### 5. Mistral API Downtime

**Risk:** Layanan AI eksternal tidak tersedia

**Impact:** Fungsionalitas agent sepenuhnya terhenti

#### **Mitigation:**

- Exponential retry mechanism
- Fallback cached templates
- Graceful degradation messages
- Backup alternative AI provider

# Risk Management (Risk Assessment & Mitigation)

## ● Low Impact Risks

### 6. Browser Compatibility Issues

**Risk:** Preview UI tidak berfungsi di browser tertentu

**Impact:** Aksesibilitas user terbatas

#### Mitigation:

- Cross-browser testing (Chrome, Firefox, Safari)
- Progressive enhancement approach
- Fallback mechanisms untuk browser lama

### 7. Performance Issues

**Risk:** File besar atau generasi kompleks memperlambat sistem

**Impact:** User experience buruk

#### Mitigation:

- Batasan ukuran file
- Pagination untuk output besar
- Strategi caching
- Performance monitoring

# Risk Management (Risk Monitoring)

## Key Risk Indicators (KRIs)

- Error rate > 5% dalam agent generation
- Response time > 30 detik untuk generation
- Queue depth > 100 pending jobs
- API failure rate > 10% untuk Mistral calls

## Monthly Risk Review

- Assess risiko baru dari user feedback
- Update strategi mitigasi
- Review incident reports
- Sesuaikan monitoring thresholds

## Contingency Plans

- Emergency rollback procedure terdokumentasi
- Alternative AI provider terkonfigurasi
- Manual fallback process untuk fungsi kritis
- Communication plan untuk major outages

# Success Criteria & KPIs

## Primary Success Metrics

### Technical Performance

- $\geq 90\%$  generasi komponen Blade yang valid secara sintaks (per validator)
- $< 30$  detik waktu respons rata-rata untuk UI generation
- $\geq 95\%$  uptime untuk system availability
- Zero critical bugs dalam production release

### User Experience

- Preview UI memungkinkan pengguna memeriksa dan menyetujui template yang dihasilkan dengan andal
- Integration process memperbarui project Laravel tanpa merusak fungsionalitas yang ada
- $< 5$  clicks dari prompt hingga kode Laravel terintegrasi
- Mobile-responsive preview interface

### Adoption Metrics

- Internal developer team mengadopsi tool untuk setidaknya 3 example pages dalam bulan pertama
- $\geq 80\%$  user satisfaction score dalam internal surveys
- Documentation completeness - semua fitur terdokumentasi dengan contoh

# Success Criteria & KPIs

## Performance Indicators

### Development Metrics

- Sprint velocity – story points diselesaikan per sprint
- Code coverage –  $\geq 80\%$  untuk critical components
- Bug density –  $< 2$  bugs per 1000 lines of code
- Technical debt – dike

### System Metrics

- API response time – 95th percentile  $< 2$  detik
- Queue processing time – rata-rata  $< 10$  detik per job
- Error rate –  $< 3\%$  untuk semua operasi agent
- Resource utilization – CPU  $< 70\%$ , Memory  $< 80\%$

### User Metrics

- Time to first successful generation –  $< 2$  menit
- Learning curve – pengguna produktif dalam  $< 30$  menit
- Feature utilization –  $\geq 70\%$  fitur digunakan oleh active users

# Success Criteria & KPIs

## Measurement Methods

### Automated Monitoring

- Application Performance Monitoring (APM)
- Error tracking dengan structured logging
- User analytics untuk feature usage
- System health dashboards

### Manual Assessment

- Weekly sprint reviews dengan stakeholders
- Monthly user feedback sessions
- Quarterly technical debt assessment
- End-to-end testing scenarios

### Success Validation

- Demo sessions dengan komponen Laravel yang dihasilkan
- Code quality reviews dari senior developers
- Integration testing dengan project Laravel sesungguhnya
- Performance benchmarking terhadap waktu manual development

# Thank You

**Link Notion :**

**[https://www.notion.so/GenLaravel-MCP-Agents-Laravel-UI-Generator-9e1d4a01fad7408da16ce4f3c0c298ae?source=copy\\_link](https://www.notion.so/GenLaravel-MCP-Agents-Laravel-UI-Generator-9e1d4a01fad7408da16ce4f3c0c298ae?source=copy_link)**

**Link Youtube :**

- Part 1 : <https://youtu.be/SLTOWBXYCzc?si=-jbBqZQUm1Nf9sRR>**
- Part 2 : [https://youtu.be/3ZV\\_AVcg82E?si=ECpMfBvGYOQatKwQ](https://youtu.be/3ZV_AVcg82E?si=ECpMfBvGYOQatKwQ)**

