

## **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 ( $\approx \$2,300$ )
- **Backlog Total:** 280 Story Points (SP)
- **Nilai per SP:**  $\approx$  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:

$$PV = 93\% \times Rp\ 35.000.000 = Rp\ 32.550.000$$

### 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

$$PV = 0.93 \times Rp\ 35.000.000 = Rp\ 32.550.000$$

#### 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

<b>Route Agent</b>	100%	Selesai di Sprint 9-10
<b>Component Agent</b>	100%	Selesai di Sprint 9-10
<b>Validator Agent</b>	100%	Selesai di Sprint 11
<b>Move to Project Agent</b>	100%	Selesai di Sprint 11
<b>Backend Polishing</b>	100%	Selesai di Sprint 12
<b>Documentation &amp; Testing</b>	100%	Selesai di Sprint 13
<b>Final Testing &amp; Release</b>	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 =  $0.983 \times \text{Rp } 35.000.000 = \text{Rp } 34.405.000$

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  $\times$  5 sprint = Rp 1.000.000
- **Total per sprint (Sprint 1-8):** Rp 900.000  $\times$  8 = Rp 7.200.000
- **Total per sprint (Sprint 9-13):** Rp 1.100.000  $\times$  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  $\times$  \$0.40 = \$7.20
    - Output: ~9M tokens  $\times$  \$0.80 = \$7.20
  - Total: \$14.40  $\approx$  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  $\times$  \$0.30 = \$0.90
    - Output: ~1.5M tokens  $\times$  \$0.90 = \$1.35
  - Total: \$2.25  $\approx$  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):**

$$\begin{aligned} AC &= Rp\ 12.700.000 + Rp\ 416.000 + Rp\ 1.300.000 + Rp\ 2.450.000 + Rp\ 1.750.000 \\ AC &= Rp\ 18.616.000 \end{aligned}$$

#### **6. Menghitung Indikator Kinerja (CPI & SPI)**

**(indikator dari teori EVM sekarang kita aplikasikan)**

##### **A. Cost Performance Index (CPI)**

$$\begin{aligned} CPI &= EV / AC \\ CPI &= Rp\ 34.405.000 / Rp\ 18.616.000 = 1.85 \end{aligned}$$

**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)

$$\text{SPI} = \text{EV} / \text{PV}$$

$$\text{SPI} = \text{Rp } 34.405.000 / \text{Rp } 32.550.000 = 1.06$$

### 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)

$$\text{CV} = \text{EV} - \text{AC}$$

$$\text{CV} = \text{Rp } 34.405.000 - \text{Rp } 18.616.000 = \text{Rp } 15.789.000$$

### Interpretasi:

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

### Schedule Variance (SV)

$$\text{SV} = \text{EV} - \text{PV}$$

$$\text{SV} = \text{Rp } 34.405.000 - \text{Rp } 32.550.000 = +\text{Rp } 1.855.000$$

### 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:

$$\text{EAC} = \text{AC} + (\text{BAC} - \text{EV}) / \text{CPI}$$

$$EAC = Rp\ 18.616.000 + (Rp\ 35.000.000 - Rp\ 34.405.000) / 1.85$$

$$EAC = Rp\ 18.616.000 + Rp\ 595.000 / 1.85$$

$$EAC = Rp\ 18.616.000 + Rp\ 321.622$$

$$EAC = Rp\ 18.937.622$$

### 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
<b>CPI</b>	1.85	Under Cost	Penggunaan biaya sangat efisien
<b>SPI</b>	1.06	Ahead of Schedule	Progres lebih cepat
<b>CV</b>	+Rp 15.789.000	Positif	Penghematan biaya besar
<b>SV</b>	+Rp 1.855.000	Positif	Lebih cepat dari jadwal
<b>EAC</b>	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 Realistik

- 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:

$\text{Revised EAC} = \text{AC} + (\text{BAC} - \text{EV}) / \text{CPI}$   
 $\text{Revised EAC} = \text{Rp } 10.471.000 + (\text{Rp } 35.000.000 - \text{Rp } 15.750.000) / 1.50$   
 $\text{Revised EAC} = \text{Rp } 10.471.000 + \text{Rp } 12.833.333$   
 $\text{Revised EAC} = \text{Rp } 23.304.333$

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:

$$\text{Revised EAC} = \text{AC} + (\text{BAC} - \text{EV}) / 1.60$$

$$\text{Revised EAC} = \text{Rp } 10.471.000 + (\text{Rp } 35.000.000 - \text{Rp } 15.750.000) / 1.60$$

$$\text{Revised EAC} = \text{Rp } 10.471.000 + \text{Rp } 12.031.250$$

$$\text{Revised EAC} = \text{Rp } 22.502.250$$

### 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 (realistik 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
<b>Schedule</b>	Behind	10% delay, SPI = 0.90
<b>Cost</b>	Under Budget	50.1% saving, CPI = 1.50
<b>Scope</b>	At Risk	Scope creep 20 SP
<b>Quality</b>	On Track	Testing coverage > 70%
<b>Risk</b>	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!)