

# Strategi Arsitektur Otomasi Terintegrasi

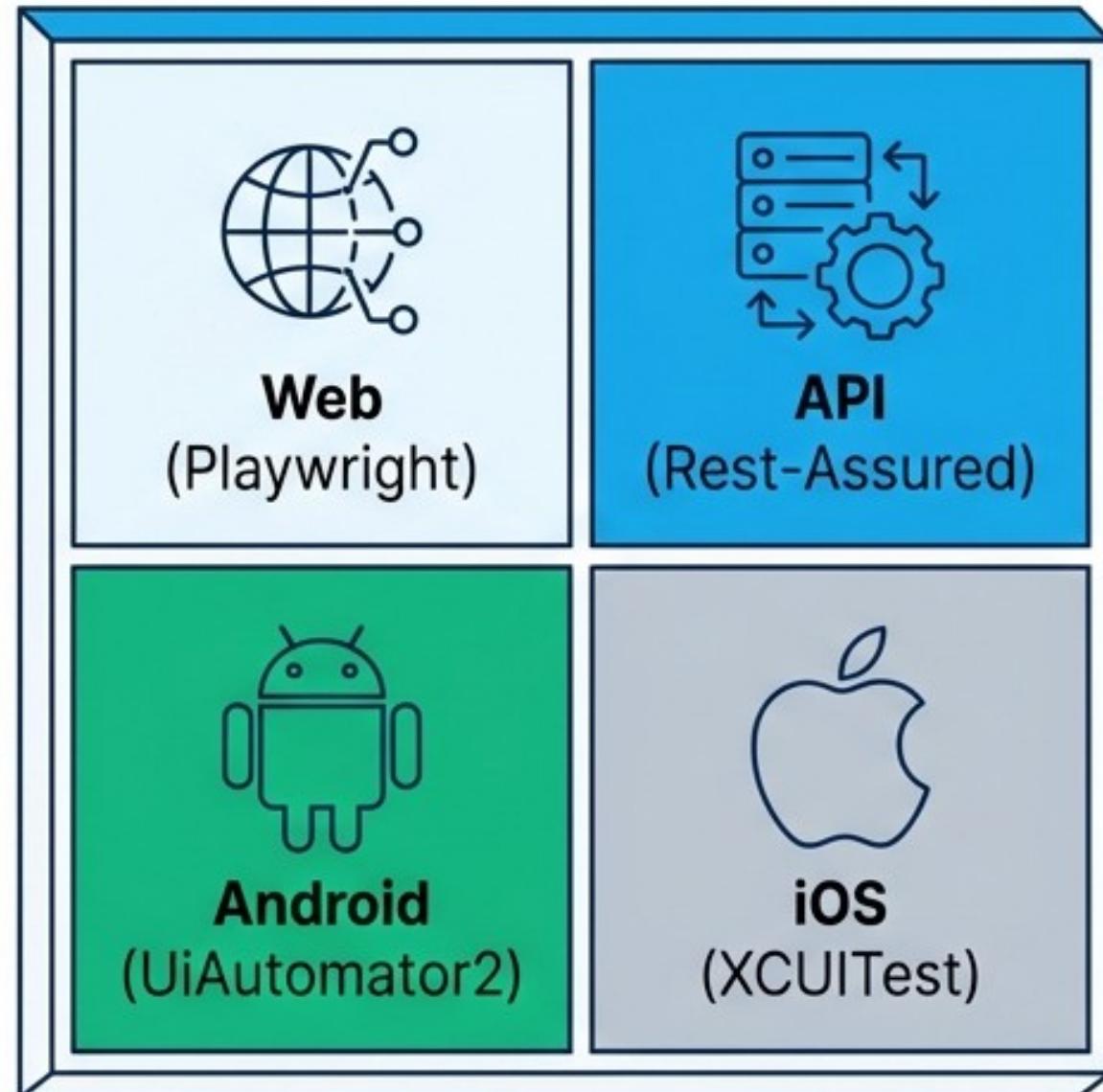
Desain Skalabel untuk Web, API, & Mobile  
dengan Pendekatan Clean Architecture



Disusun Oleh:  
Habib Astari Adi

# Strategi Arsitektur Otomasi Terintegrasi

Desain Skalabel untuk Web, API, Android, & iOS dengan Pendekatan Clean Architecture



## ■ Unified Monorepo

Satu repositori terpadu untuk sinkronisasi ketat antara skrip tes dan kode aplikasi.

## ■ Evolusi Pola Desain

Transisi dari Page Object Model (POM) ke Screenplay Pattern untuk skalabilitas.

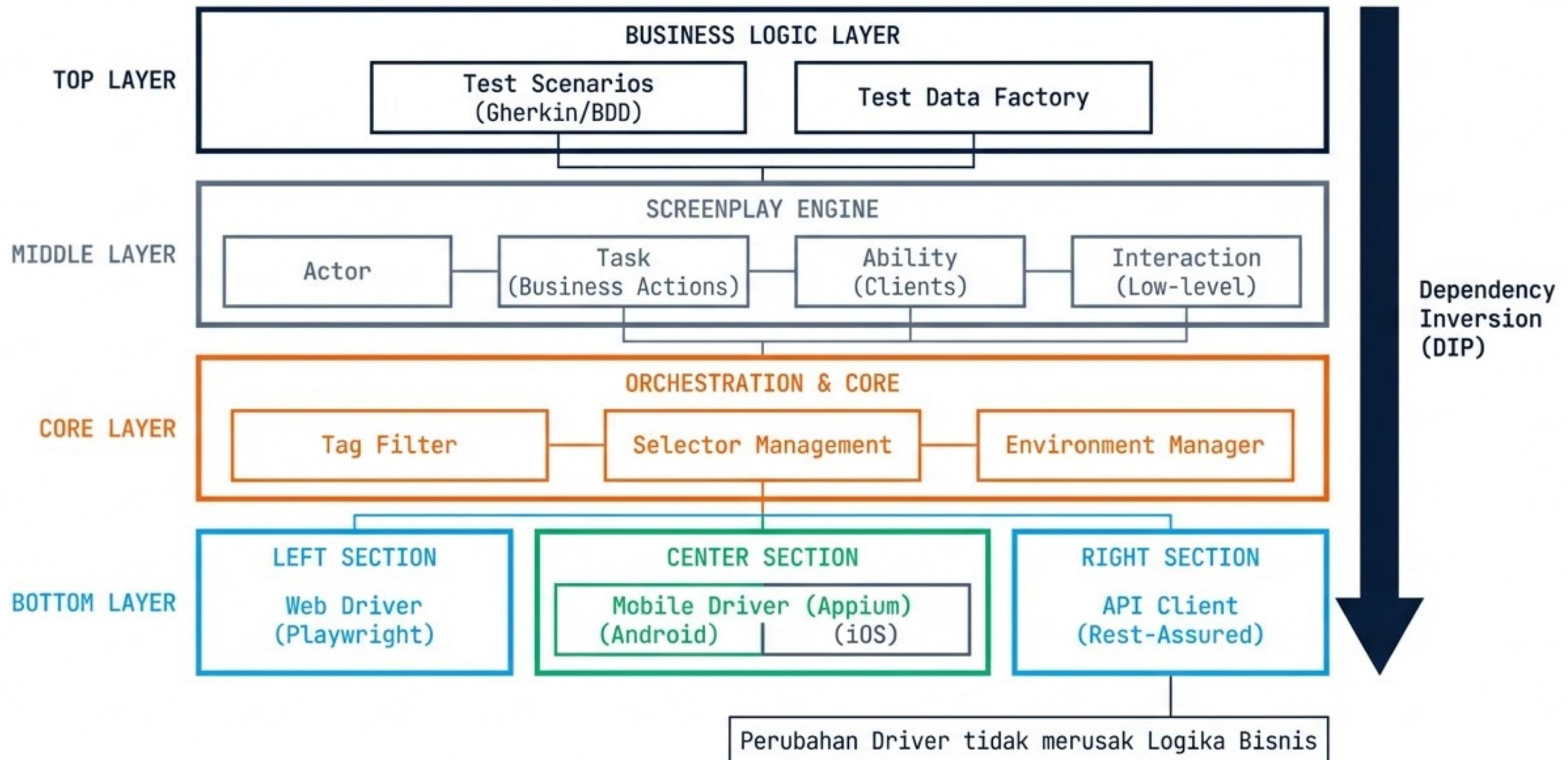
## ■ Infrastruktur Hibrida

Eksekusi Web/API pada VM Cloud dan Mobile pada Local Device Farm.

## ■ AI-Driven Maintenance

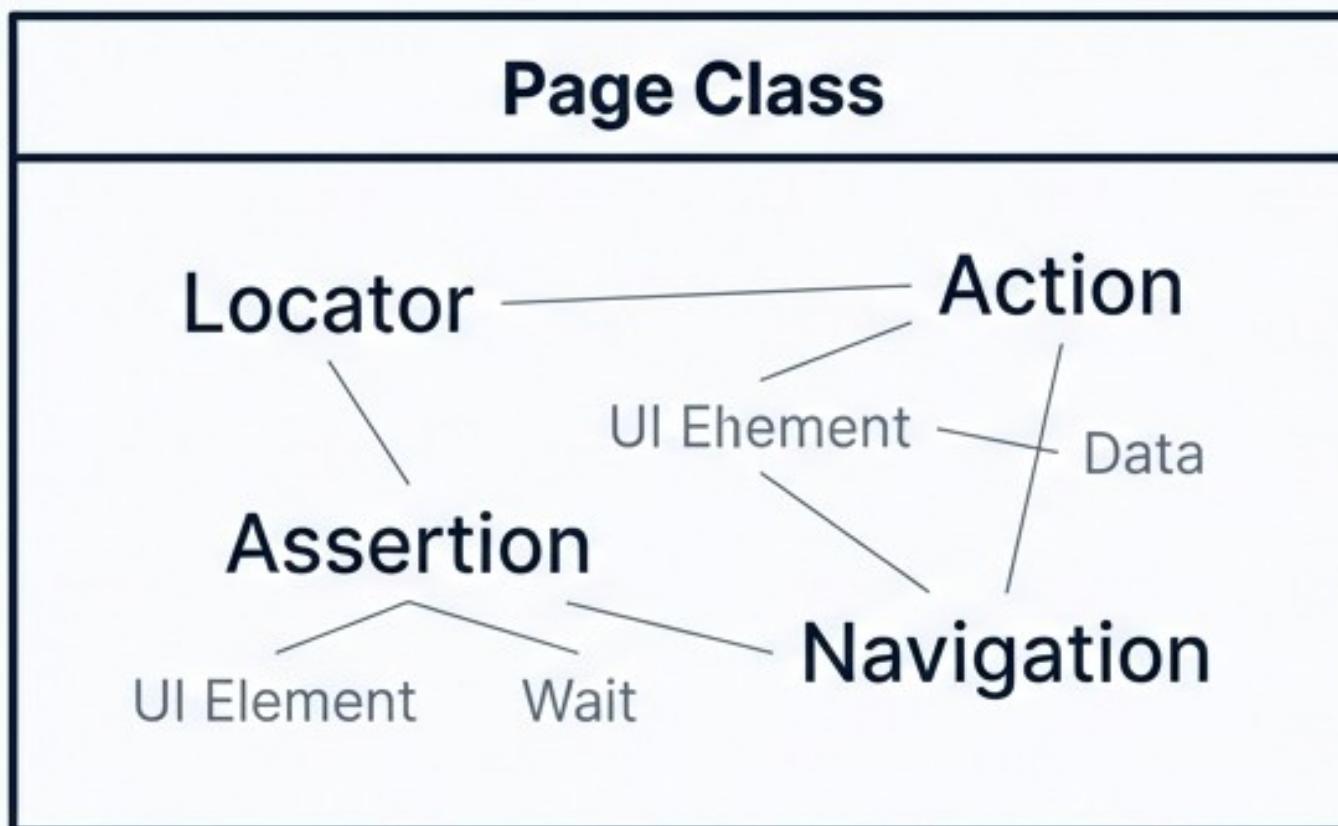
Implementasi Self-Healing dan Test Impact Analysis (TIA).

# PETA ARSITEKTUR: ISOLASI LOGIKA BISNIS DARI DRIVER TEKNIS



# Fondasi Clean Code: Evolusi ke Screenplay Pattern

## Legacy: Page Object Model (POM)



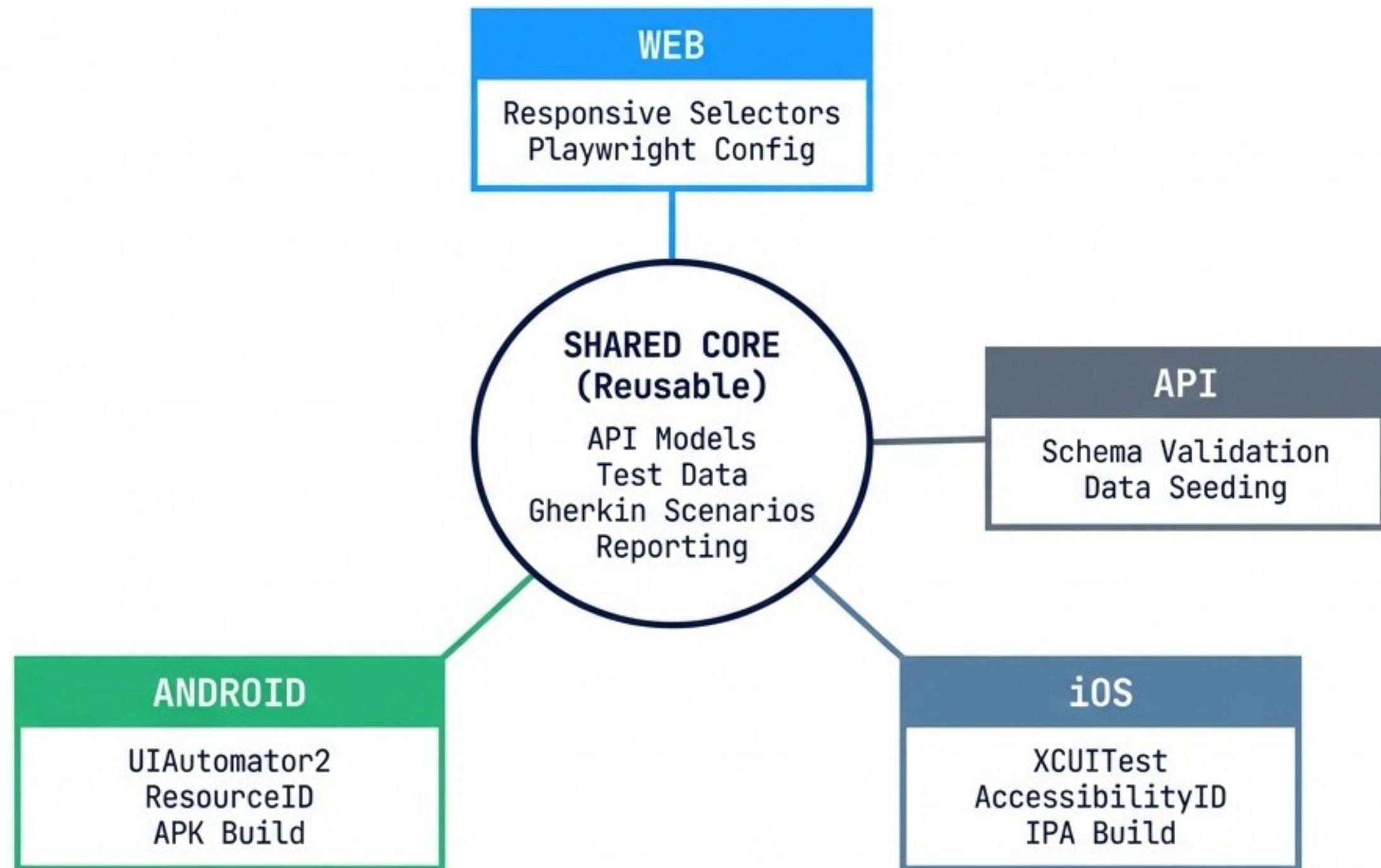
- Melanggar Single Responsibility (SRP)
- Maintenance Nightmare: UI berubah, class besar rusak.

## Modern: Screenplay Pattern



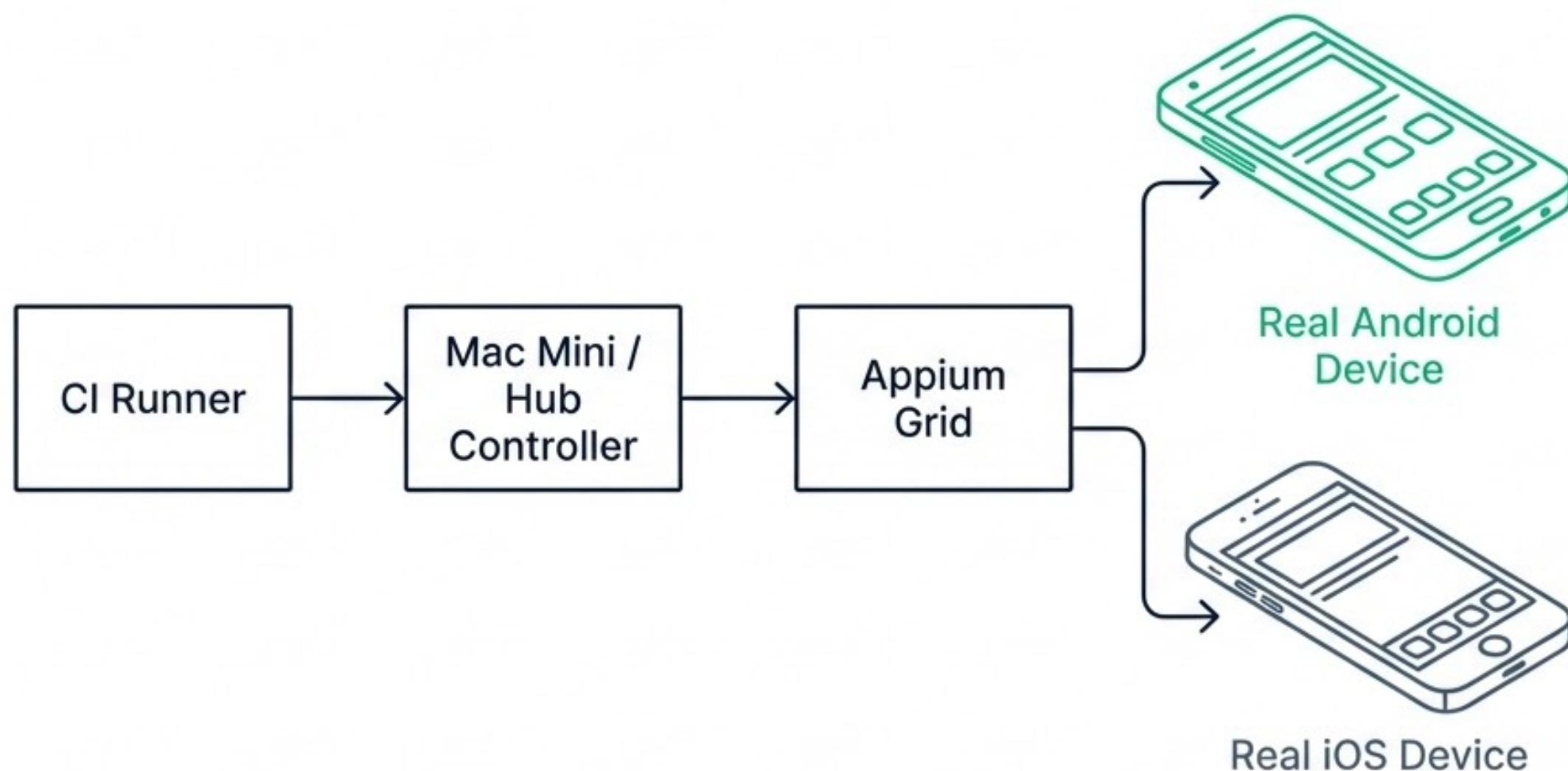
- Modular & Reusable
- Perbaikan bug terisolasi pada satu Interaction.

# Strategi Segregasi Platform dalam Monorepo



Monorepo memfasilitasi sinkronisasi kode, namun implementasi teknis (Driver & Locator) tetap tersegregasi secara ketat.

# Infrastruktur Mobile: Local Device Farm Terorkestrasi



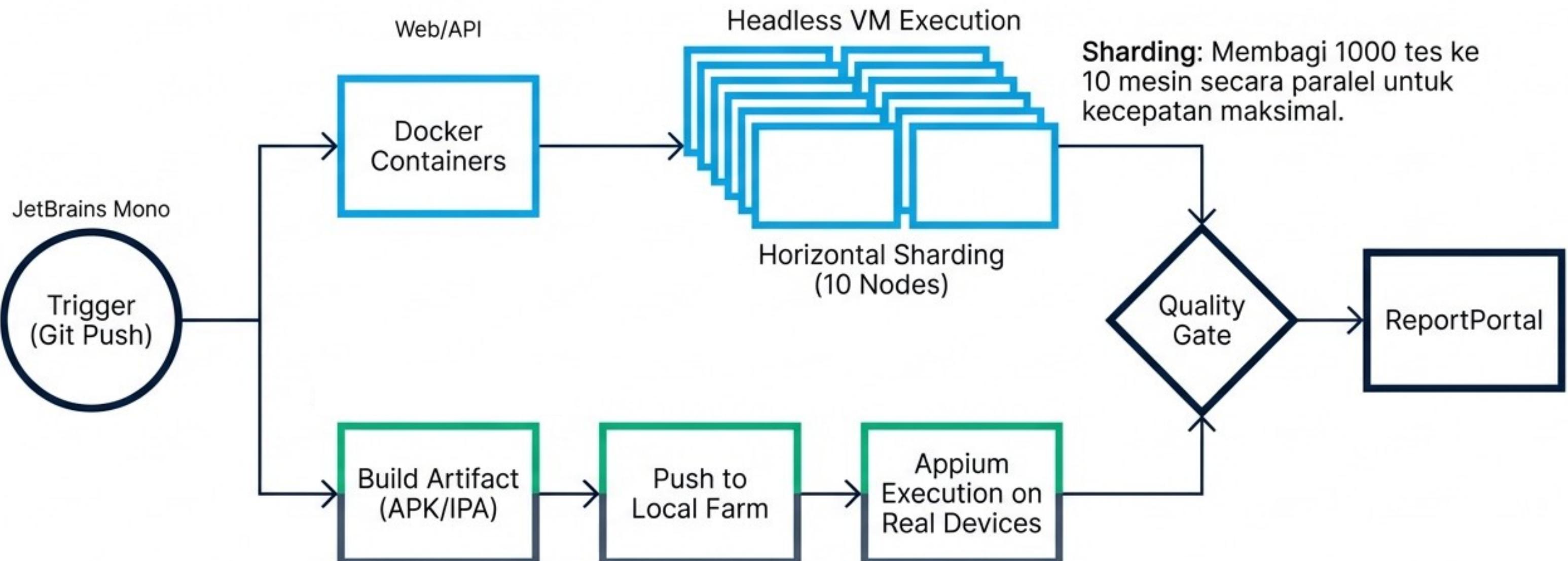
## Mengapa Local Farm?

- Biaya: Nol biaya per menit (vs Cloud).
- Akurasi: Sensor hardware asli & latensi real-time.
- Keamanan: Kontrol penuh privasi data.

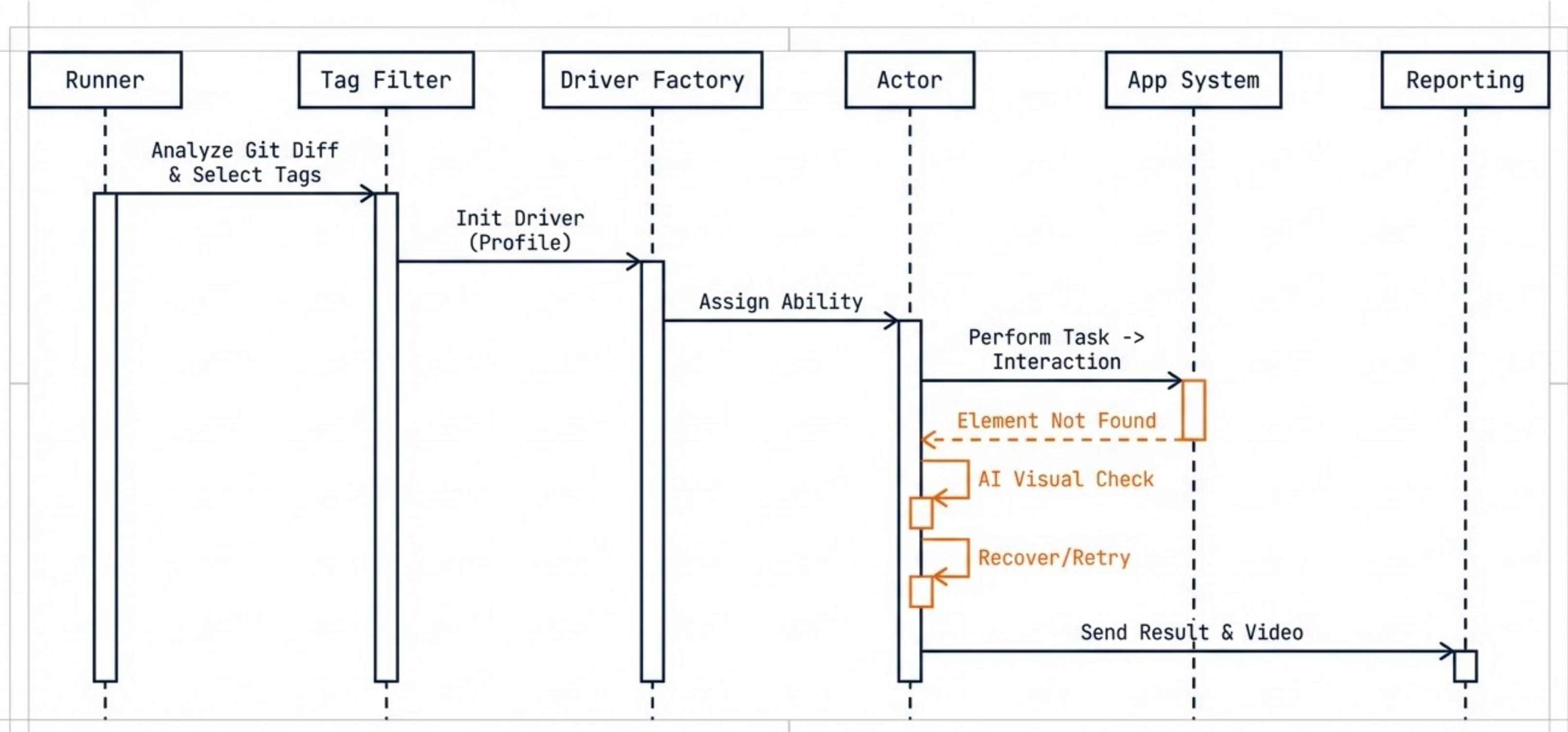
## Mitigasi Isu Fisik

- Auto-health check & monitoring baterai.

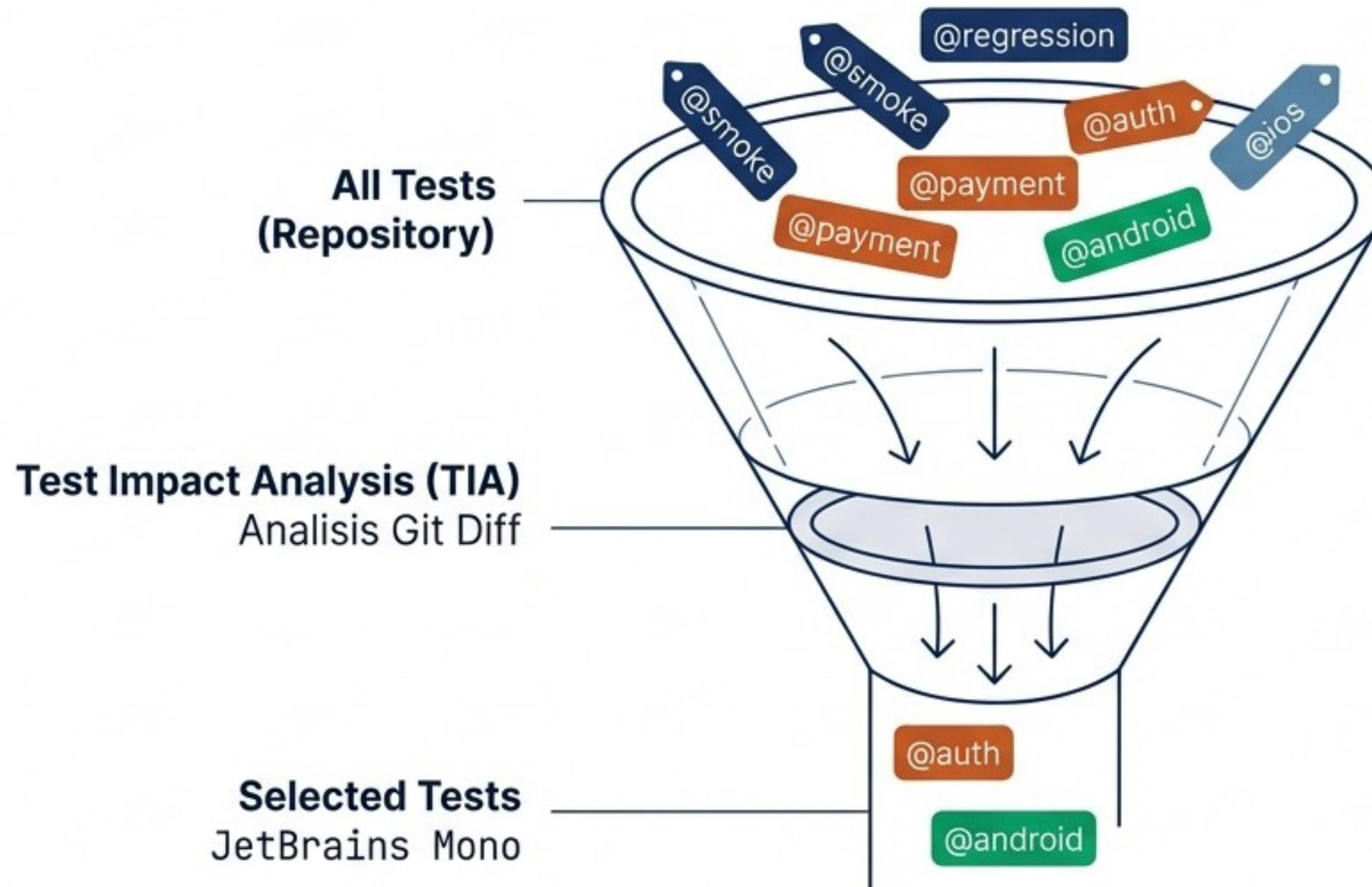
# Pipeline CI/CD: Eksekusi Paralel & Sharding



# Sequence Diagram: Alur Eksekusi Runtime

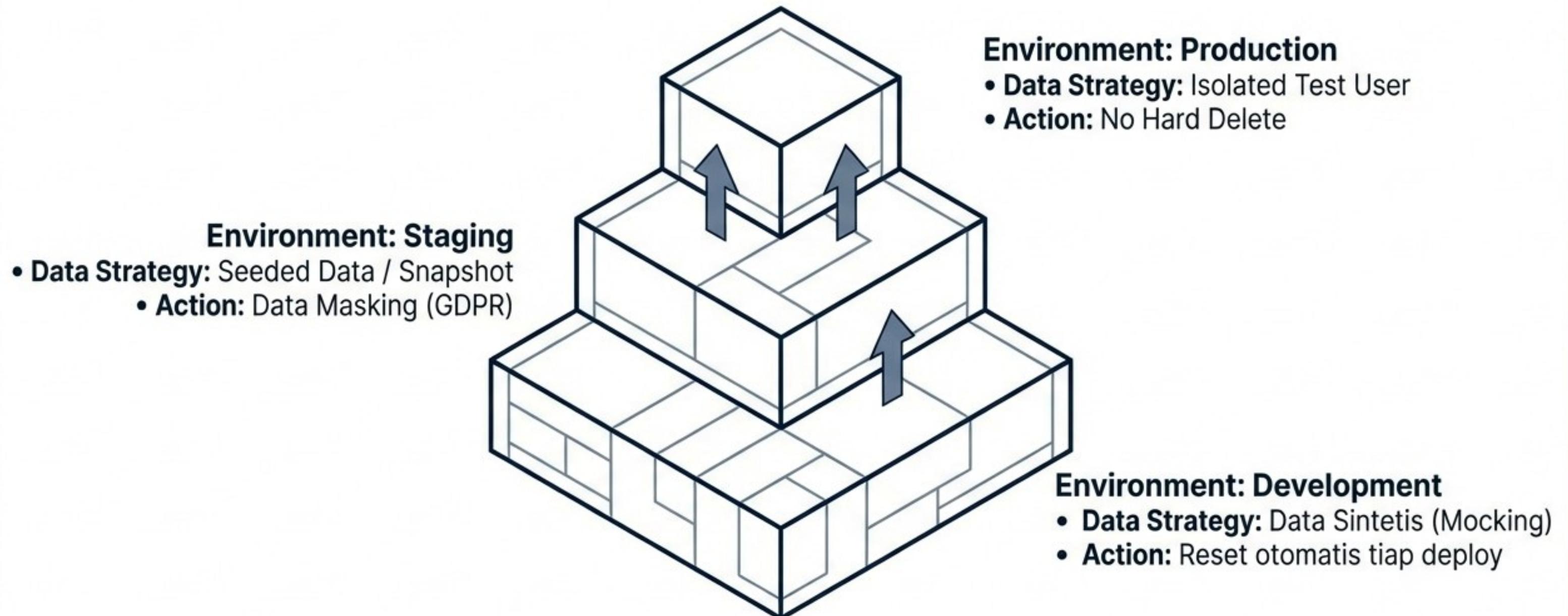


# Strategi Eksekusi Cerdas: Tagging & Impact Analysis



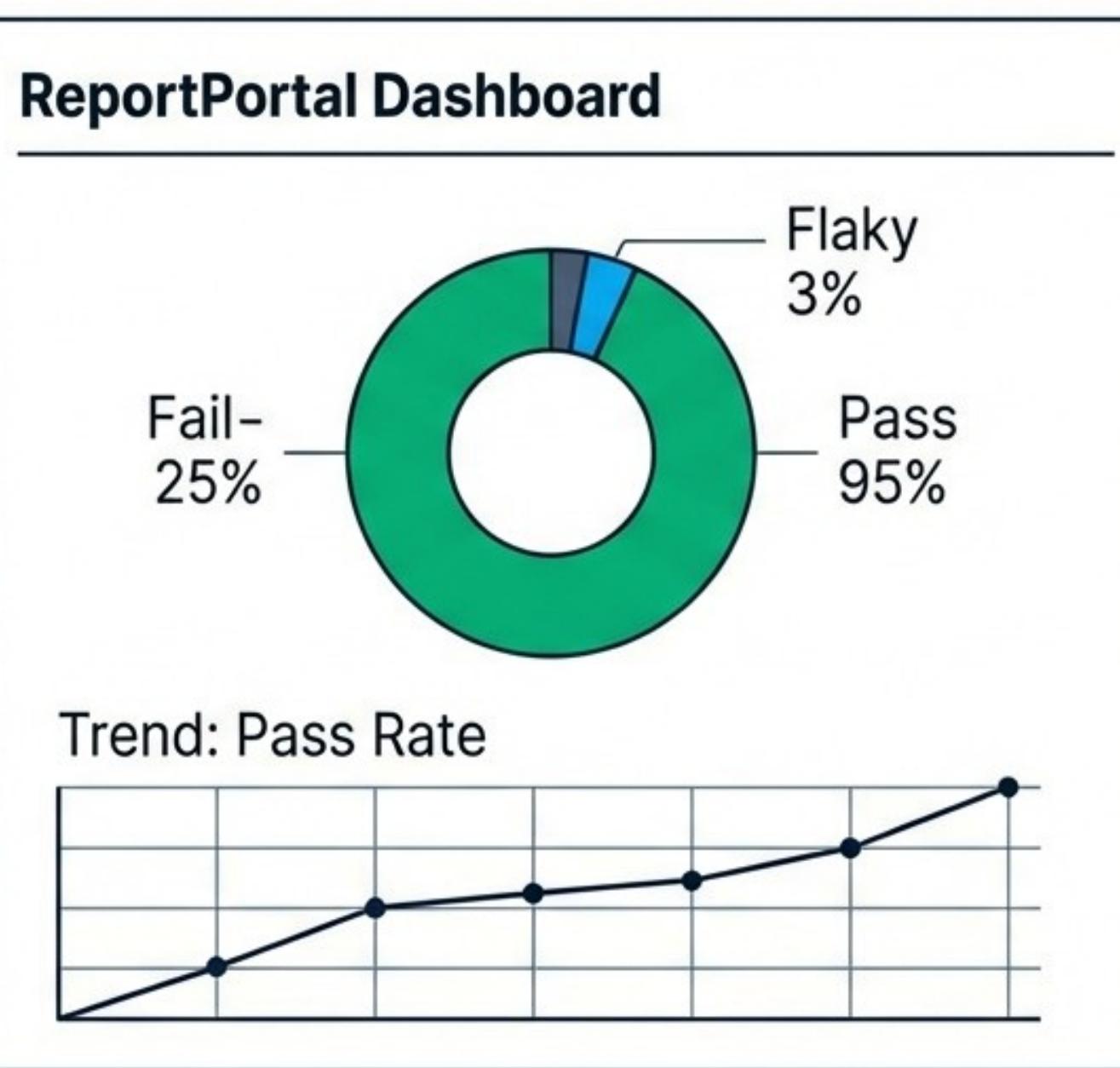
Mengurangi waktu feedback hingga 85%. Hanya menjalankan tes yang relevan dengan perubahan kode (misal: CSS profil berubah -> tes pembayaran dilewati).

# Manajemen Lingkungan & Data (Environment as Code)



Konfigurasi dikelola via .env dan dynamic overrides. Satu skrip untuk semua lingkungan.

# Pelaporan Enterprise & Reduksi Noise



## Analisis Akar Masalah (RCA)

Auto-klasifikasi: Product Bug vs System Issue (**ML-Based**).

## Smart Retry Logic

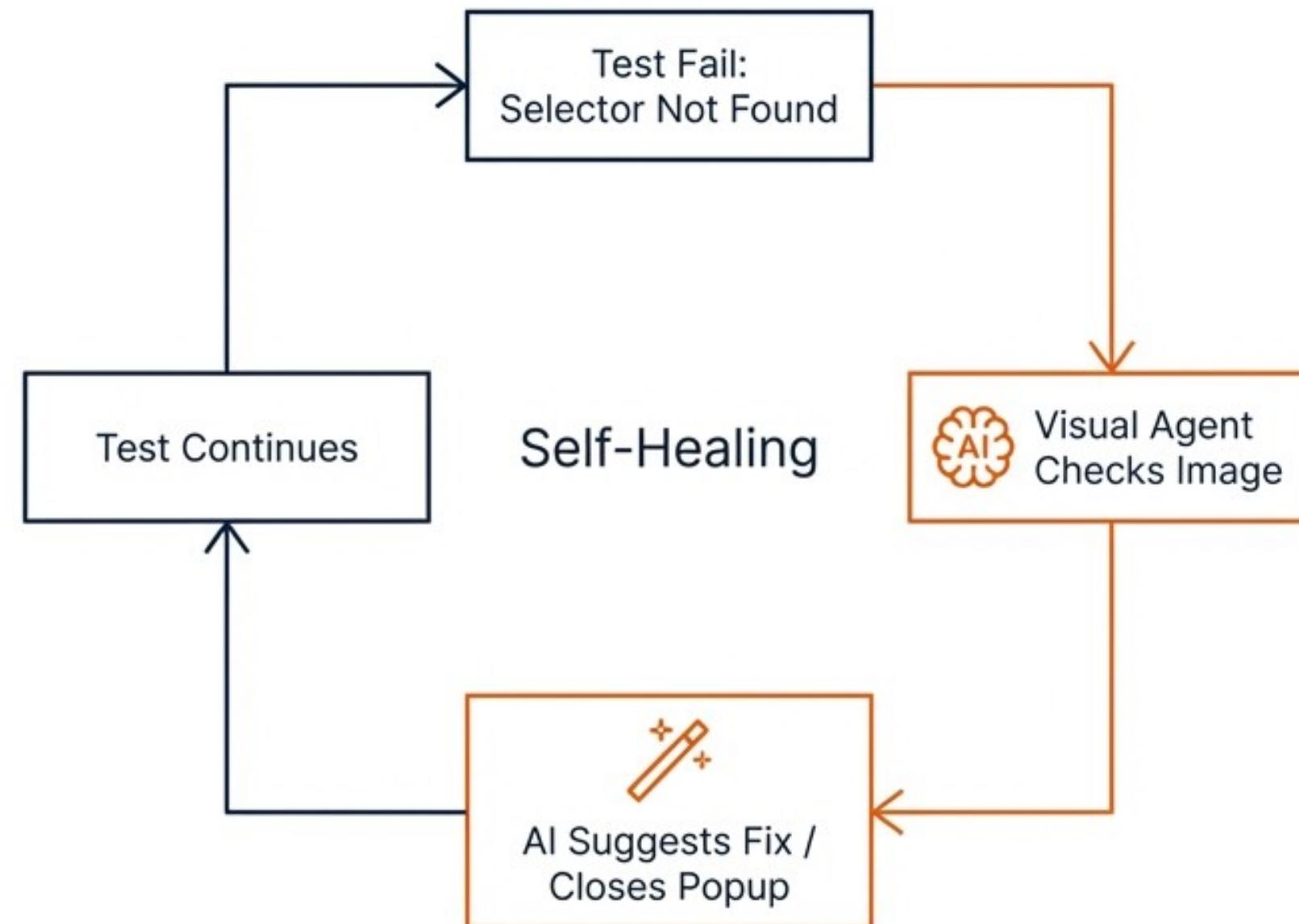
Max 2x retry. Lulus percobaan ke-2 = "Flaky" (Bukan Fail).

## Artefak Lengkap

Screenshot + Video 30 detik + Log Konsol.

# Modernisasi: Mengatasi Bottleneck UI dengan AI

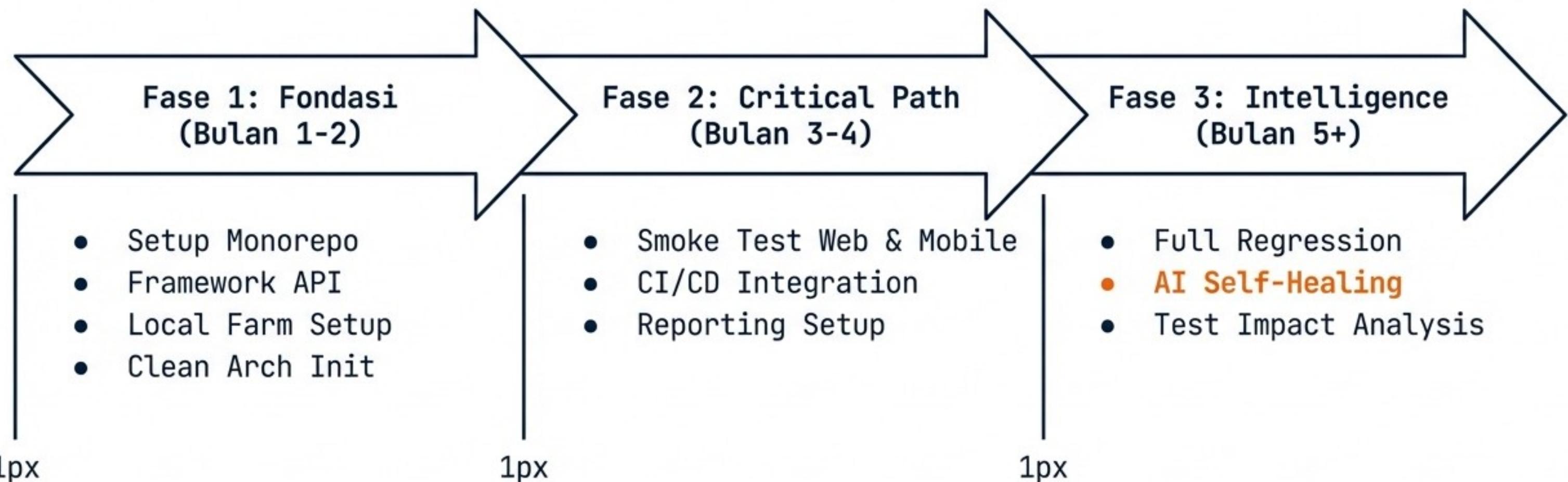
- **Flow Creation:** Generasi kode one-liner dari rekaman user journey.
- **Benefit:** Fokus tim berubah dari "fixing scripts" ke "increasing coverage".



# Ringkasan Keputusan Arsitektur & Trade-offs

Decision	Pros	Cons	Mitigasi
Screenplay Pattern	High Reusability	Learning Curve Tinggi	Template Code & Training
Local Device Farm	Cost Efficiency, Real Hardware	Maintenance Fisik	Auto-health Check
Monorepo	Sync Code & Test Instant	Repo Size Besar	Sparse Checkout

# Peta Jalan Implementasi (Roadmap)



“Membangun ekosistem kualitas yang adaptif, bukan sekadar skrip otomasi.”