TCMS

Test Case Management System

Infrastruktur Kualitas Tersentralisasi

NotebookLM

Latar Belakang & Tantangan Bisnis

> ***Berapa pass rate regresi kita minggu lalu? (Tidak ada single source of truth)***

Tantangan Utama Bisnis

- Release confidence rendah
- Less visibility untuk management
- Duplicate effort karena test case tersebar

Sebelum	Sesudah TCMS
Total Test Case: Beda versi/manual	Real-time & filterable
Pass Rate Regresi: Outdated spreadsheet	Auto-calculated
Konteks Bug: Hilang/Lost	Linked langsung ke Jira
Hasil Automation: Cek manual	Auto-report & notifikasi Lark

NotebookLM

TCMS: Infrastruktur & Core Features

Core Engine

- Multi-project support
- Suite hierarchy tak terbatas
- Test plans & eksekusi tervalidasi
- MCP AI Agent Gateway
  Modul besar: 40+ tool operations, 3 domain (TCMS, Jira, Git, Lark)

Platform Scale

- Isolasi data multi-tenant
- Auto-provisioning SSO (Zitadel)
- Module entitlements per tim

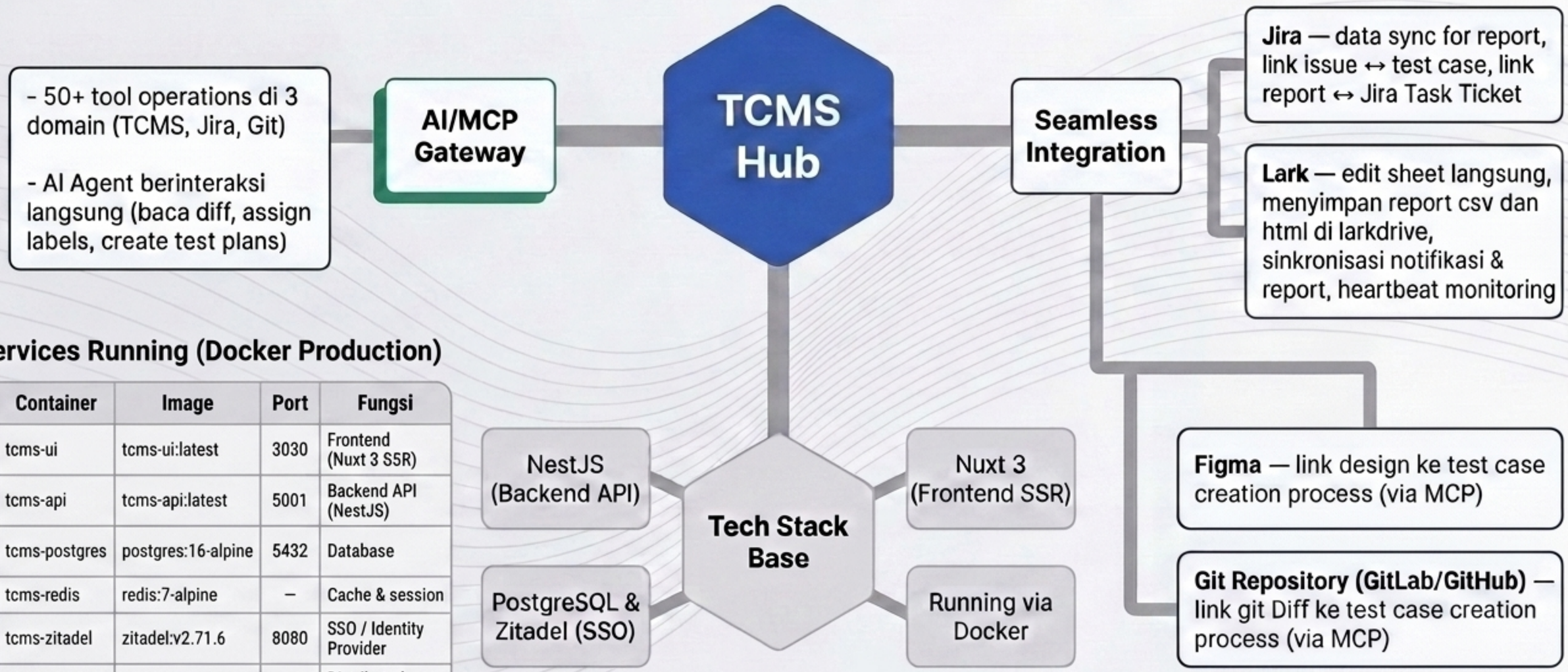
Test Account Management

- Atomic checkout/lease
- Manajemen pool & antrian
- Kredensial terenkripsi aman

Total 42 Module

NotebookLM

Arsitektur, Integrasi & AI Gateway



Services Running (Docker Production)

#	Container	Image	Port	Fungsi
1	tcms-ui	tcms-ui:latest	3030	Frontend (Nuxt 3 S5R)
2	tcms-api	tcms-api:latest	5001	Backend API (NestJS)
3	tcms-postgres	postgres:16-alpine	5432	Database
4	tcms-redis	redis:7-alpine	–	Cache & session
5	tcms-zitadel	zitadel:v2.71.6	8080	SSO / Identity Provider
6	tcms-jaeger	jaeger all-in-one	16686	Distributed tracing

NotebookLM

Standar Keandalan & Keamanan (Enterprise-Ready)



Quality Gate (CI/CD)

Wajib pass Lint, 101 Unit Test files, E2E, & verifikasi migrasi database sebelum merge ke GitLab CI.



Security & Isolation

Login JWT via SSO (tanpa password lokal), kredensial terenkripsi at rest, & isolasi multi-tenant ketat per tim.



Observability

OpenTelemetry & Jaeger untuk distributed tracing, ditambah notifikasi heartbeat realtime ke Lark jika service down.



Recovery Plan

Automated daily backup via pg_dump, rollback scripts instan, dan auto-restart container Docker jika terjadi crash.

NotebookLM

Rencana Adopsi & Implementasi Tim

Fase 1: UI Adoption (Barrier Rendah)



Tim QA login via SSO, input manual, dan manajemen test plan (tanpa perlu coding).

Fase 2: Automation Scale



Integrasi CI/CD menggunakan service tokens & instalasi TCMS reporter pada pipeline otomatis.

Fase 3: Advanced Workflows (TAM)



Migrasi shared test accounts ke pool tersentralisasi dengan manajemen checkout atomik.

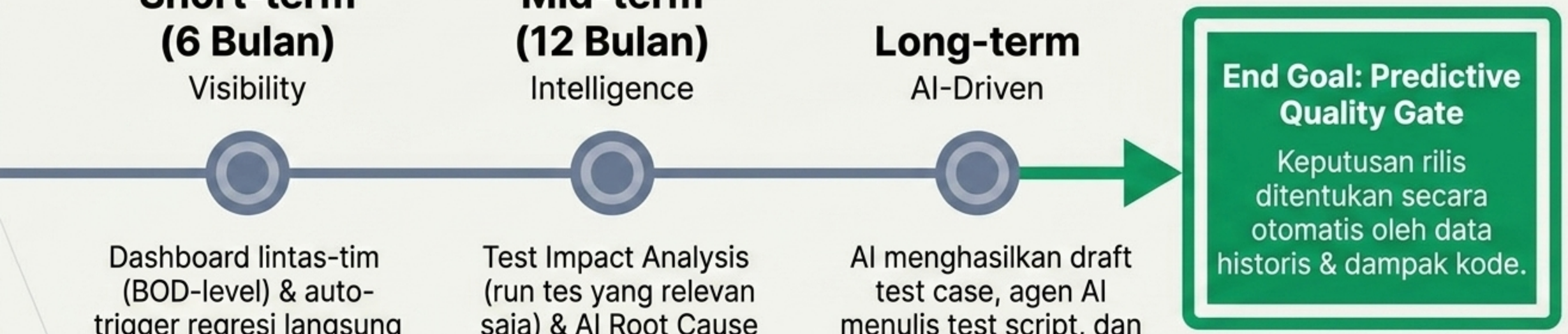
Fleksibilitas Tim



Setiap tim dapat melakukan kustomisasi mandiri untuk format reporter, taksonomi tag, dan mapping project Jira.

NotebookLM

Visi & Roadmap Pengembangan



NotebookLM

“

TCMS secara fundamental mengubah manual tracking yang terfragmentasi menjadi Quality Engineering yang terukur, otomatis, dan siap untuk era AI.

Kesimpulan & Langkah Selanjutnya — Questions?

NotebookLM