TECHNICAL DESIGN DOCUMENT (TDD)

---

1. Document Control

Item	Description
Document Title	Technical Design Document (TDD)
Project Name	Mobile App Pinjaman Online
Version	1.0
Document Status	Draft for Technical Review
Prepared By	System Architect / Backend Engineer
Reviewed By	IT Project Manager
Approved By	Head of Technology
Date	17/12/2025
Confidentiality	Internal & Client Confidential

---

2. Purpose

Dokumen ini mendefinisikan desain teknis detail untuk sistem **Mobile App Pinjaman Online PT. XYZ** sebagai acuan implementasi, integrasi, pengujian, deployment, dan audit sistem.

Dokumen ini digunakan oleh:

- Backend Engineer
- Mobile Engineer
- DevOps / Infrastructure Team
- QA & Automation Team

- Security & Compliance
- Auditor Internal & Eksternal

---

3. Scope of System

Sistem mencakup:

- Aplikasi Mobile (Android & iOS)
- Backend Microservices
- API Gateway
- Database & Cache
- Event-driven Communication
- Notification System
- Monitoring & Logging
- Security & Compliance

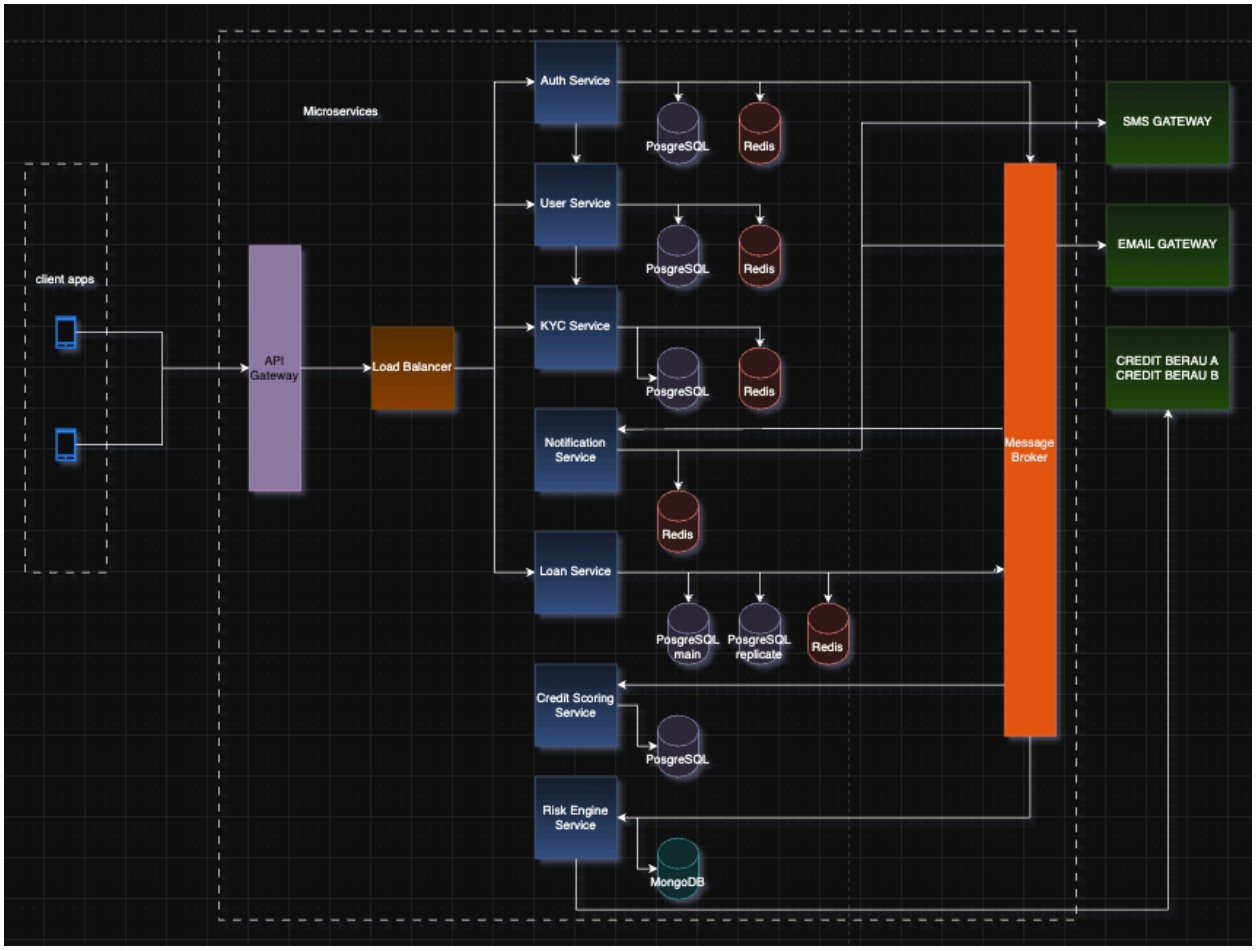
Tidak termasuk:

- Sistem pihak ketiga (hanya integrasi)
- Infrastruktur internal vendor SMS/Email

---

4. Architecture Overview

4.1 High-Level Architecture



Komponen utama sistem:

- Mobile App (Android / iOS)
- API Gateway (Kong)
- Microservices (Golang / Java / ASP .NET)
- PostgreSQL Database
- MongoDB Database
- PostgreSQL Database
- Redis Cache
- Message Broker (RabbitMQ)

- Auth Service
- User Service
- Loan Service
- Notification Service
- KYC Service
- Credit Scoring Service
- Risk Engine Service
- Cloud Infrastructure

4.2 Architecture Style

- Microservices Architecture
- Stateless Backend
- API-first Design
- Event-driven (Asynchronous Processing)

4.3 Architecture Principles

- Scalability & Elasticity
- Security by Design
- Fault Tolerance
- Loose Coupling
- Observability & Auditability

---

5. Technology Stack

Layer	Technology
Mobile App	Dart (Flutter)
Backend	Go / Java / ASP .NET
API Protocol	REST (JSON)
Authentication	JWT / Token-based
Database	PostgreSQL, Redis, ELK, MongoDB
Cache	Redis
Message Broker	RabbitMQ
Container	Docker
Orchestration	Kubernetes
CI/CD	GitHub Actions / GitLab CI
Cloud	AWS / GCP
Monitoring	Prometheus, Grafana
Logging	ELK Stack
Secrets	Kubernetes Secret / Vault

---

6. Service Decomposition (Microservices)

6.1 List of Services

Service	Responsibility
API Gateway	Routing, auth validation, rate limiting
Auth Service	Registrasi, login, token, biometric
User Service	Profil user & status akun
KYC Service	Upload & verifikasi KTP & selfie
Loan Service	Pengajuan pinjaman & status
Credit Scoring Service	Hitung kelayakan pinjaman

Risk Engine Service	Integrasi skor kredit pihak ke-3
Notification Service	Email & SMS

---

7. Backend Internal Design

7.1 Layered Architecture (Golang)

Setiap service mengikuti struktur:

- **Handler** – HTTP request/response
- **Service** – Aplikasi pihak ketiga
- **Domain / Entity** – Core business object
- **Repository** – Database access
- **DTO** – Request & response mapping
- **Infra** – Core system (Database, Message Broker, Email, Config, dll)
- **Use case** – Business logic

7.2 Coding Principles

- Clean Architecture
- SOLID Principles
- Dependency Injection
- Event Driver Architecture
- Unit-testable components
- Integration testing

---

8. API Design

8.1 API Standards

- RESTful
- gRPC
- Stateless
- JSON, Protobuf
- Versioned endpoint (`/api/v1`)
- Idempotency untuk transaksi penting
- Rate limiter
- CORS
- Api-Key
- Authentikasi & Authorization
- Enkripsi
- Validation input

8.2 Sample API Endpoints

Method	Endpoint	Description
POST	/api/v1/register	Registrasi user
POST	/api/v1/login	Login user
POST	/api/v1/kyc/upload	Upload KYC
POST	/api/v1/loan/apply	Ajukan pinjaman
GET	/api/v1/loan/status	Status pinjaman
GET	/api/v1/billing	Tagihan & hutang

---

9. Event-Driven Design (RabbitMQ)

9.1 Event Communication

Digunakan untuk proses asynchronous:

- KYC verification
- Credit scoring
- Risk Engine Credit

- Notification

9.2 Sample Event Contract

Event Name: LoanAppliedEvent

Payload:

- event_id
- user_id
- loan_id
- amount
- timestamp

---

10. Database Design

10.1 Core Tables

- users_auth (Auth Service)
- auth_sessions (Auth Service)

- user_profiles (User Service)

- user_kyc (KYC service)
- user_kyc_histories (KYC service)

- loans (Loan Service)
- loans_payments (Loan Service)
- loans_histories (Loan Service)

- credit_scores (Credit Scoring Service)
- credit_scores_list (Credit Scoring Service)

- risk_evaluations_collection (Risk Engine Service)

- notifications_collection (Notification Service)

10.2 Data Design Principles

- UUID sebagai primary key
- Soft delete
- Immutable transaction records
- Referential integrity enforced
- Data Confidential (Encrypted)

---

11. Security Design

11.1 Authentication & Authorization

- JWT / Access Token
- Refresh Token
- Role-based Access Control (RBAC)

11.2 Data Security

- TLS (HTTPS)
- Password hashing (bcrypt)
- Encryption at rest
- Secure secret management
- AES data pribadi

11.3 Audit & Compliance

- Semua transaksi finansial tercatat
- Log tidak dapat dimodifikasi

- Timestamp & user traceability

---

12. Deployment Architecture

12.1 Containerization

- One service per Docker image
- Immutable artifact

12.2 Kubernetes Design

- Deployment per service
- Horizontal Pod Autoscaler
- Ingress Controller
- ConfigMap & Secrets

12.3 Autoscaling Strategy

Metric	Threshold
CPU	>70%
Memory	>75%
Request Rate	Configurable

---

13. Logging & Monitoring

13.1 Logging Strategy

- Structured logging (JSON)
- Correlation ID per request

- Centralized log aggregation

13.2 Monitoring Metrics

- Service uptime
- API latency
- Error rate
- Resource utilization

---

14. Error Handling Strategy

- Standard error format
- Validation error
- Business rule violation
- System failure
- Retry & dead-letter queue

---

15. Performance & Reliability

- Stateless service
- Circuit breaker
- Timeout & retry
- Graceful shutdown

---

16. Traceability Matrix (BRD → SRS → TDD)

BRD / SRS ID	TDD Component
UC-01	Auth Service
UC-02	KYC Service
UC-03	Loan Service
UC-04	Credit Scoring
UC-05	Notification

---

17. Assumptions & Risks

Assumptions

- Cloud infrastructure tersedia
- Third-party service stabil

Risks

- Traffic spike → autoscaling
- Third-party downtime → fallback

---

18. Approval & Sign-off

Role	Name	Signature	Date
Head of Technology			
IT Project Manager			
System Architect			