

PT. XYZ Fintech Solutions

NAMA: M Wasilah Hadi

TANGGAL: 22 Januari 2026

SOAL 1: DESKRIPSI DIAGRAM ARSITEKTUR

Diagram menunjukkan arsitektur sistem fintech PT. XYZ yang terdiri dari beberapa komponen terintegrasi untuk mendukung pengembangan aplikasi pinjaman online.

KOMPONEN UTAMA:

1. Mobile Applications

- Tiga aplikasi mobile identik untuk akses pengguna
- Interface utama untuk layanan fintech dan pinjol
- Multi-platform support (iOS/Android)

2. Service Worker

- Mengelola background processes dan task scheduling
- Mengatur komunikasi asinkron antar komponen
- Handle offline capabilities

3. SOA - Service Oriented Architecture

- Hub pusat sebagai orchestrator layanan
- Mengatur komunikasi dan routing antar sistem
- Memfasilitasi loose coupling antar komponen
- Central business logic management

4. Core Banking System

- Sistem perbankan inti untuk pemrosesan transaksi
- Menangani operasi keuangan fundamental
- Database utama untuk data nasabah dan transaksi
- Compliance dan regulatory management

5. OpenAPI 3.1

- Standar dokumentasi dan spesifikasi API
- Interface untuk integrasi dengan sistem eksternal
- Memungkinkan third-party integration
- API versioning dan backward compatibility

6. Kibana

- Dashboard monitoring dan analytics
- Visualisasi data dan logging sistem
- Performance monitoring dan alerting
- Business intelligence reporting

7. Jenkins

- CI/CD automation tool
- Deployment dan testing otomatis
- DevOps pipeline management
- Code quality assurance

ALUR KERJA SISTEM:

Mobile Apps → Service Worker → SOA → Core Banking System

SOA bertindak sebagai central hub yang menghubungkan semua komponen, memungkinkan skalabilitas dan maintainability yang baik untuk pengembangan aplikasi pinjol PT. XYZ.

SOAL 2: SPESIFIKASI API PINJOL

Berikut adalah spesifikasi API dalam format Collection Postman/OpenAPI YAML untuk menangani fitur pengajuan dan monitoring pinjaman online:

```
openapi: 3.0.3
info:
  title: Pinjol API - PT. XYZ
  version: 1.0.0
  description: API untuk layanan pinjaman online PT. XYZ

servers:
  - url: http://localhost:3000
    description: API untuk layanan pinjaman online PT. XYZ
  - url: https://api.xyz-fintech.com/v1
    description: Production Server

paths:
  /auth/login:
    post:
      summary: Login pengguna
      tags:
        - Authentication
      security: []
      requestBody:
        required: true
        content:
          application/json:
            schema:
              type: object
              required:
                - username
                - password
              properties:
                username:
                  type: string
                  example: "user@example.com"
                password:
                  type: string
                  example: "password123"
      responses:
        '200':
          description: Login berhasil
```

```
content:
  application/json:
    schema:
      type: object
      properties:
        access_token:
          type: string
          example:
            "eyJhbGciOiJIUzI1NilsInR5cCl6IkpxVCJ9eyJzdWliOiJ1c2VyMTIzIn0.abc123"
        token_type:
          type: string
          example: "Bearer"
        expires_in:
          type: integer
          example: 3600
        user_id:
          type: string
          example: "user123"
```

```
/loans/apply:
post:
  summary: Mengajukan pinjaman baru
  tags:
    - Loans
  requestBody:
    required: true
    content:
      application/json:
        schema:
          type: object
          required:
            - user_id
            - amount
            - tenor
        properties:
          user_id:
            type: string
            example: "user123"
          amount:
            type: number
            minimum: 500000
            maximum: 20000000
            example: 5000000
          tenor:
```

```
        type: integer
        minimum: 1
        maximum: 24
        example: 12
      purpose:
        type: string
        example: "Modal usaha"
    responses:
      '201':
        description: Pengajuan berhasil
        content:
          application/json:
            schema:
              type: object
              properties:
                loan_id:
                  type: string
                  example: "loan123"
                status:
                  type: string
                  enum: [pending, approved, rejected]
                  example: "pending"
                message:
                  type: string
                  example: "Pengajuan pinjaman berhasil disubmit"
```

```
/loans/loan123:
get:
  summary: Cek status pinjaman (Fixed ID untuk testing)
  tags:
    - Loans
responses:
  '200':
    description: Detail pinjaman
    content:
      application/json:
        schema:
          type: object
          properties:
            loan_id:
              type: string
              example: "loan123"
            status:
              type: string
```

```
        example: "approved"
amount:
  type: number
  example: 5000000
remaining_balance:
  type: number
  example: 4500000
next_payment_date:
  type: string
  format: date
  example: "2024-02-15"

/loans/loan123/payments:
post:
  summary: Bayar cicilan (Fixed ID untuk testing)
  tags:
    - Payments
  requestBody:
    required: true
    content:
      application/json:
        schema:
          type: object
          required:
            - amount
          properties:
            amount:
              type: number
              example: 500000
  responses:
    '200':
      description: Pembayaran berhasil
      content:
        application/json:
          schema:
            type: object
            properties:
              payment_id:
                type: string
                example: "pay123"
              status:
                type: string
                example: "success"
              remaining_balance:
```

```
    type: number
    example: 4000000
```

```
/users/user123/credit-score:
get:
  summary: Cek skor kredit pengguna (Fixed ID untuk testing)
  tags:
    - Credit
  responses:
    '200':
      description: Skor kredit
      content:
        application/json:
          schema:
            type: object
            properties:
              user_id:
                type: string
                example: "user123"
              credit_score:
                type: integer
                example: 750
              risk_level:
                type: string
                enum: [low, medium, high]
                example: "low"
```

```
/loans/loan123/schedule:
get:
  summary: Jadwal pembayaran (Fixed ID untuk testing)
  tags:
    - Loans
  responses:
    '200':
      description: Jadwal cicilan
      content:
        application/json:
          schema:
            type: object
            properties:
              loan_id:
                type: string
                example: "loan123"
              installments:
```

```
        type: array
        items:
          type: object
          properties:
            installment_number:
              type: integer
              example: 1
            due_date:
              type: string
              format: date
              example: "2024-02-15"
            amount:
              type: number
              example: 500000
  /notifications/payment-reminder:
    post:
      summary: Kirim reminder pembayaran
      tags:
        - Notifications
      requestBody:
        required: true
        content:
          application/json:
            schema:
              type: object
              required:
                - loan_id
              properties:
                loan_id:
                  type: string
                  example: "loan123"
                reminder_type:
                  type: string
                  enum: [sms, email, push]
                  example: "email"
      responses:
        '200':
          description: Reminder terkirim
          content:
            application/json:
              schema:
                type: object
                properties:
                  message:
```

```
type: string
example: "Payment reminder sent successfully"
sent_at:
type: string
format: date-time
example: "2024-01-22T10:30:00Z"
```

components:

securitySchemes:

 BearerAuth:

 type: http

 scheme: bearer

 bearerFormat: JWT

security:

- BearerAuth: []

FITUR UTAMA API:

1. Pengajuan Pinjaman (/loan/application)
 - Validasi input (jumlah, tenor, user)
 - Response dengan loan_id untuk tracking
 - Status pengajuan real-time
2. Monitoring Status (/loan/{loan_id}/status)
 - Cek status persetujuan pinjaman
 - Informasi sisa saldo dan jadwal bayar
 - History transaksi

KEAMANAN:

- JWT Bearer token authentication
- Input validation and sanitization
- Rate limiting untuk mencegah abuse

INTEGRASI:

API ini terintegrasi dengan Core Banking System melalui SOA untuk:

- Validasi kredit scoring
- Pemrosesan approval otomatis
- Sinkronisasi data keuangan
- Compliance reporting

