

Bayu Rizki

1. Penjelasan Diagram terkait Integrasi:

Dalam sebuah Perusahaan keuangan/Bank atau fintech mengadopsi sistem middleware sebagai sistem integrasi. Dengan adanya middleware, tercipta transparansi pada seluruh jaringan sehingga Bank dapat dengan mudah melakukan interaksi dan kolaborasi ke aplikasi klien dan server.

Pada kegiatan perbankan atau transaksi keuangan, middleware dengan kapabilitas SOA (Service Oriented Architecture) menghubungkan Bank/transaksi keuangan dengan berbagai kanal elektronik yang ada di Bank, Core Banking, hingga pihak eksternal.

SOA (Service Oriented Architecture) berupaya untuk mengatasi banyak kekurangan dari teknologi integrasi sebelumnya dengan menekankan aspek Service. Service adalah unit tersendiri dari fungsionalitas bisnis yang terbuka dan tersedia untuk (idealnya) semua Enterprise Applications. Service menjadi building block untuk aplikasi yang mendukung SOA.

Setiap Service menjelaskan apa yang dilakukannya, bagaimana panggilannya, dan format pesan yang diharapkan akan diterima dan diproduksi dalam dokumen formatnya. Contoh dari tipe format dokumen yang didukung yaitu XML and JSON. XML menjadi standar format message yang dipergunakan didalam WSDL dan JSON menjadi standar format message yang dipergunakan didalam RESTful API.

Dengan pendekatan ini, service yang di-expose dengan baik dapat dirakit dan digunakan kembali untuk lebih cepat membuat aplikasi yang dapat lebih mendukung perubahan proses bisnis.

Adapun penggunaan OpenAPI 3.1 adalah spesifikasi standar yang digunakan untuk mendefinisikan dan mendokumentasikan API.

2. Berikut adalah detail mengenai rancangan spesifikasi API untuk fitur pinjaman online (Pinjol):

a. API formulir aplikasi pinjaman pribadi dalam format JSON (API FormLoanApplication)

```
openapi: 3.0.0
info:
  title: Personal Loan Application API
  description: API untuk mengirimkan formulir aplikasi pinjaman pribadi melalui halaman pendaratan
  version: 1.0.0

paths:
  /loan-applications:
    post:
      summary: Mengirimkan Aplikasi Pinjaman Pribadi
      description: Mengirimkan formulir aplikasi pinjaman pribadi melalui halaman pendaratan.
      requestBody:
        required: true
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/LoanApplicationForm'
      responses:
        '201':
          description: Aplikasi pinjaman pribadi berhasil dikirim.
        '400':
          description: Kesalahan pada permintaan. Detail kesalahan tersedia dalam respons.

components:
  schemas:
    LoanApplicationForm:
      type: object
      properties:
        fullName:
          type: string
          description: Nama lengkap pemohon pinjaman.
        email:
          type: string
          format: email
          description: Alamat email pemohon pinjaman.
        phoneNumber:
          type: string
          description: Nomor telepon pemohon pinjaman.
        loanAmount:
          type: number
          description: Jumlah pinjaman yang diminta.
        employmentStatus:
          type: string
          description: Status pekerjaan pemohon pinjaman (contoh: karyawan tetap, wiraswasta).
        additionalInformation:
          type: string
          description: Informasi tambahan atau catatan khusus.
```

Request:

POST /FormLoanApplication
Content-Type: application/json

```
{  
  "fullName": "John Doe",  
  "email": "john.doe@example.com",  
  "phoneNumber": "+123456789",  
  "loanAmount": 5000,  
  "employmentStatus": "karyawan tetap",  
  "additionalInformation": "Pemohon memiliki pengalaman kerja lebih dari 5 tahun."  
}
```

Response:

HTTP/1.1 201 Created
Content-Type: application/json

```
{  
  "message": "Aplikasi pinjaman pribadi berhasil dikirim."  
}
```

- b. Berikut adalah contoh spesifikasi API untuk mengirimkan informasi tentang status aplikasi pembukaan pelanggan, termasuk permintaan (request) dan respons (response) dalam format JSON:

```

openapi: 3.0.0
info:
  title: Customer Opening Application Status API
  description: API untuk mengirimkan informasi tentang status aplikasi pembukaan pelanggan
  version: 1.0.0

paths:
  /ApplicationStatus:
    post:
      summary: Mengirimkan Informasi Status Aplikasi Pembukaan Pelanggan
      description: Mengirimkan informasi tentang status aplikasi pembukaan pelanggan.
      requestBody:
        required: true
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/ApplicationStatusInfo'
      responses:
        '200':
          description: Informasi status aplikasi diterima dengan sukses.
        '400':
          description: Kesalahan pada permintaan. Detail kesalahan tersedia dalam respons.

components:
  schemas:
    ApplicationStatusInfo:
      type: object
      properties:
        customerId:
          type: string
          description: ID unik pelanggan.
        status:
          type: string
          description: Status aplikasi (approved, rejected, cancelled).
        validationData:
          type: object
          description: Data validasi status aplikasi.

```

Request (Permintaan):
 POST Openingstatus/UnsecuredLoan
 Content-Type: application/json

```

{
  "customerId": "123456",
  "status": "approved",
  "validationData": {
    "incomeProof": true,
    "creditScore": 750
  }
}

```

Response (Respons):

HTTP/1.1 200 OK

Content-Type: application/json

```
{  
  "message": "Informasi status aplikasi diterima dengan sukses."  
}
```

- c. Berikut adalah contoh spesifikasi API untuk mengambil informasi biaya/tarif (API FeeInformation, termasuk permintaan (request) dan respons (response) dalam format JSON:

openapi: 3.0.0

info:

title: Loan Fee Information API

description: API untuk mengambil informasi biaya/tarif

version: 1.0.0

paths:

/FeeInformation:

get:

summary: Mendapatkan Informasi Biaya/Tarif Loan

description: Mengambil informasi biaya/tarif untuk produk Loan.

responses:

'200':

description: Informasi biaya/tarif berhasil ditemukan.

content:

application/json:

schema:

\$ref: '#/components/schemas/FeeInformation'

'404':

description: Informasi biaya/tarif untuk Loan tidak ditemukan.

components:

schemas:

FeeInformation:

type: object

properties:

productName:

type: string

description: Nama produk.

fees:

type: array

description: Daftar biaya/tarif.

items:

type: object

properties:

name:

type: string

description: Nama biaya/tarif.

amount:

type: number

description: Jumlah biaya/tarif.

currency:

type: string

description: Mata uang biaya/tarif.

Request:

GET /FeeInformation

Content-Type: application/json

Response:

HTTP/1.1 200 OK

Content-Type: application/json

```
{
  "productName": "Loan",
  "fees": [
    {
      "name": "Processing Fee",
      "amount": 200,
      "currency": "USD"
    },
    {
      "name": "Late Payment Fee",
      "amount": 20,
      "currency": "USD"
    }
  ]
}
```

- d. Berikut adalah contoh spesifikasi API untuk mengambil informasi suku bunga untuk produk Loan dalam k, termasuk permintaan (request) dan respons (response) dalam format JSON:

```
openapi: 3.0.0
info:
  title: Interest Rate Information API
  description: API untuk mengambil informasi suku bunga untuk produk Loan
  version: 1.0.0

paths:
  /InterestRate:
    get:
      summary: Mendapatkan Informasi Suku Bunga Loan
      description: Mengambil informasi suku bunga untuk produk Loan.
      responses:
        '200':
          description: Informasi suku bunga berhasil ditemukan.
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/InterestRateInformation'
        '404':
          description: Informasi suku bunga untuk Loan tidak ditemukan.

components:
  schemas:
    InterestRateInformation:
      type: object
      properties:
        productName:
          type: string
          description: Nama produk.
        interestRate:
          type: number
          description: Suku bunga produk Loan.
        term:
          type: string
          description: Periode suku bunga (contoh: annual, monthly).
```

Request:

GET /InterestRate

Content-Type: application/json

Response:

HTTP/1.1 200 OK

Content-Type: application/json

```
{  
  "productName": "Loan",  
  "interestRate": 8.5,  
  "term": "annual"  
}
```

- e. Berikut adalah contoh spesifikasi API untuk membuat akun angsuran pinjaman baru (loan tranche account) untuk produk Loan dalam . Ini mencakup permintaan (request) dan respons (response) dalam format JSON

openapi: 3.0.0

info:

title: Loan Tranche Account API

description: API untuk membuat akun angsuran pinjaman baru untuk produk Loan

version: 1.0.0

paths:

/TrancheAccount:

post:

summary: Membuat Akun Angsuran Pinjaman Loan

description: Membuat akun angsuran pinjaman baru untuk produk Loan.

requestBody:

required: true

content:

application/json:

schema:

\$ref: '#/components/schemas/TrancheAccountCreation'

responses:

'201':

description: Akun angsuran pinjaman berhasil dibuat.

content:

application/json:

schema:

\$ref: '#/components/schemas/TrancheAccountResponse'

'400':

description: Kesalahan pada permintaan. Detail kesalahan tersedia dalam respons.

components:

schemas:

TrancheAccountCreation:

type: object

properties:

customerId:

type: string

description: ID unik pelanggan.

loanAmount:

type: number

description: Jumlah pinjaman.

termInMonths:

type: integer

description: Durasi pinjaman dalam bulan.

interestRate:

type: number

description: Tingkat bunga pinjaman.

TrancheAccountResponse:

type: object

properties:

accountId:

type: string

description: ID unik akun angsuran pinjaman.

message:

type: string

description: Pesan sukses atau informasi tambahan.

Request:

POST /tranche-account

Content-Type: application/json

```
{  
  "customerId": "123456",  
  "loanAmount": 10000,  
  "termInMonths": 12,  
  "interestRate": 8.5  
}
```

Response

HTTP/1.1 201 Created

Content-Type: application/json

```
{  
  "accountId": "abc123",  
  "message": "Akun angsuran pinjaman berhasil dibuat."  
}
```

- f. Berikut adalah contoh permintaan (request) dan respons (response) untuk API melakukan pembayaran tagihan pinjaman tanpa sistem pembalikan dan tanpa perlu inquiry terlebih dahulu dalam format JSON:

openapi: 3.0.0

info:

title: Loan Bill Repayment API

description: API untuk melakukan pembayaran tagihan pinjaman tanpa sistem pembalikan dan tanpa perlu inquiry

version: 1.0.0

paths:

/LoanBillRepayment:

post:

summary: Melakukan Pembayaran Tagihan Pinjaman

description: Melakukan pembayaran tagihan pinjaman tanpa sistem pembalikan dan tanpa perlu inquiry.

requestBody:

required: true

content:

application/json:

schema:

\$ref: '#/components/schemas/LoanRepaymentRequest'

responses:

'200':

description: Pembayaran tagihan pinjaman berhasil.

content:

application/json:

schema:

\$ref: '#/components/schemas/LoanRepaymentResponse'

'400':

description: Kesalahan pada permintaan. Detail kesalahan tersedia dalam respons.

components:

schemas:

LoanRepaymentRequest:

type: object

properties:

customerId:

type: string

description: ID unik pelanggan.

billAmount:

type: number

description: Jumlah tagihan pinjaman yang dibayar.

LoanRepaymentResponse:

type: object

properties:

transactionId:

type: string

description: ID unik transaksi pembayaran.

message:

type: string

description: Pesan sukses atau informasi tambahan.

Request:

POST /loan-bill-repayment

Content-Type: application/json

```
{  
  "customerId": "123456",  
  "billAmount": 1000  
}
```

Response:

HTTP/1.1 200 OK
Content-Type: application/json

```
{  
  "transactionId": "abc123",  
  "message": "Pembayaran tagihan pinjaman berhasil."  
}
```