POLITEKNIK NEGERI BANYUWANGI

Laporan: UAS Interopabilitas

Mobile Application Development
Semester Ganjil 2024/2025

Disusun Oleh Kelompok 03:

Tahun 2025

BAB 1
Pendahuluan & Latar Belakang

1.1 Deskripsi Sistem

Proyek ini adalah sistem **Middleware** yang dibangun menggunakan Node.js dan Express. Tujuan utamanya adalah menyelesaikan masalah interoperabilitas antar sistem vendor yang heterogen. Middleware ini mengagregasi data dari tiga vendor (Vendor A, B, dan C) yang memiliki format data berbeda menjadi satu format JSON standar (`integratedData`).

1.2 Tabel Pembagian Tanggung Jawab

Berikut adalah detail pembagian tugas sesuai instruksi ujian:

Anggota 1: Vendor A — Implementasi Controller Vendor A — (nama: Stefano)

Anggota 2: Vendor B — Implementasi Controller Vendor B — (nama: Rizwar Ardian)

Anggota 3: Vendor C — Implementasi Controller Vendor C — (nama: Joan Ilham)

Anggota 4: Integrator — Logic Aggregation & Middleware — (nama: Darius Bagaskara)

BAB 2
Spesifikasi Data Vendor

Kami menghubungkan tiga vendor dengan karakteristik data yang berbeda:

2.1 Vendor A
- **Format**: `kd_produk`, `nm_brg`, `hrg` (String), `ket_stok`.
- **Tantangan**: Harga dalam format string dan perlu parsing.
- **Controller**: `vendorA_Controllers.js`.

2.2 Vendor B
- **Format**: `sku`, `productName`, `price` (Integer), `isAvailable` (Boolean).
- **Tantangan**: Penamaan field berbeda (`sku` vs `kd_produk`) dan status stok boolean.
- **Controller**: `vendorB_Controllers.js`.

2.3 Vendor C
- **Format**: Nested Object (`details.name`, `pricing.base_price`, `stock`).
- **Tantangan**: Struktur data bertingkat dan harga perlu perhitungan pajak (`tax`).
- **Controller**: `vendorC_Controller.js`.

BAB 3
Arsitektur Sistem Middleware

3.1 Teknologi yang Digunakan

Sistem ini dibangun menggunakan tech stack berikut:

- **Runtime**: Node.js
- **Framework**: Express.js
- **Format Data**: JSON (JavaScript Object Notation)

3.2 Alur Data (Data Flow)

1. **Client Request**: Client (Postman/Browser) mengirim request GET ke `/api/products`.
2. **Router**: `routes/api.js` meneruskan request ke `integratorController.js`.
3. **Fetching**: Integrator mengambil data mentah dari `vendorA`, `vendorB`, dan `vendorC`.
4. **Normalization**: Data mentah di-mapping ke format standar (`id`, `nama`, `harga_final`, `status`, `sumber`).
5. **Response**: JSON terintegrasi dikirim kembali ke Client.

BAB 4
Implementasi Logika Integrator

Logika integrasi terdapat pada file `controllers/integratorController.js`.

4.1 Normalisasi Data

Setiap vendor memiliki penamaan field yang berbeda. Middleware melakukan mapping sebagai berikut:

- `kd_produk` / `sku` / `id` -> `id`
- `nm_brg` / `productName` / `details.name` -> `nama`
- `hrg` / `price` / `pricing` -> `harga_final`

4.2 Business Logic (Logic Trap)

1. **Vendor A**: Diskon 10% dihitung dengan `Math.round(parseInt(item.hrg) * 0.9)`.
2. **Vendor B**: Status `isAvailable` (boolean) diubah menjadi string "Tersedia" atau "Habis".
3. **Vendor C**: Harga basis ditambah pajak (`base_price + tax`). Produk makanan diberi label "(Recommended)".

BAB 5
Implementasi API Endpoints

Aplikasi mengekspos beberapa endpoint melalui `routes/api.js`.

5.1 Endpoint Vendor Individual

- `GET /api/vendor-a`: Mengembalikan data mentah Vendor A.
- `GET /api/vendor-b`: Mengembalikan data mentah Vendor B.
- `GET /api/vendor-c`: Mengembalikan data mentah Vendor C.

5.2 Endpoint Terintegrasi

- `GET /api/products`: Endpoint utama yang menggabungkan seluruh data vendor.

5.3 Kode Implementation

Menggunakan `router.get()` dari Express untuk mendefinisikan rute dan menghubungkannya dengan fungsi controller yang sesuai.

BAB 6
Hasil Pengujian (Interoperabilitas)

Pengujian dilakukan dengan membandingkan response dari vendor asli dengan response terintegrasi dari Middleware.

6.1 Test Case 1: Vendor A

Input: `{"kd_produk": "A001", "hrg": "15000"}` Output Integrator:

```
{
  "id": "A001",
  "harga_final": 13500, // Diskon 10%
  "sumber": "Vendor A"
}
```

6.2 Test Case 2: Vendor C

Input: `{"details": {"name": "Nasi Tempong"}, "stock": 50}` Output Integrator:

```
{
  "nama": "Nasi Tempong (Recommended)",
  "status": "Tersedia",
  "sumber": "Vendor C"
}
```

BAB 7
Penutup & Kesimpulan

7.1 Kesimpulan

Middleware Interoperabilitas yang dibangun berhasil mengintegrasikan data dari tiga vendor berbeda tanpa harus mengubah sistem asli masing-masing vendor.

1. **Heterogenitas Data Teratasi**: Perbedaan format nama field dan tipe data berhasil dinormalisasi.
2. **Business Logic Terpusat**: Logika diskon dan pajak diterapkan di layer middleware, memudahkan maintenance.

7.2 Saran Pengembangan

- Menambahkan fitur caching (Redis) untuk mengurangi load ke vendor.
- Implementasi autentikasi (JWT) pada endpoint middleware.