

Backend Developer Technical Test

Anda diminta untuk membangun sebuah sistem payment gateway menggunakan bahasa Go. Sistem ini akan mengelola transaksi pembayaran untuk sebuah toko online. Anda harus membuat API yang dapat digunakan oleh front-end aplikasi untuk melakukan transaksi pembayaran, mengecek status pembayaran, dan melihat riwayat transaksi. Anda juga harus memastikan bahwa sistem ini aman, scalable, dan reliabel.

Kebutuhan Fungsional

1. Membuat Transaksi Pembayaran:

- Endpoint: `POST /transactions`

Request Body:

json

```
{
  "amount": 1000,
  "currency": "USD",
  "payment_method": "credit_card",
  "description": "Payment for order #1234",
  "customer_id": "cust_001"
}
```

Response:

json

```
{
  "transaction_id": "trans_001",
  "status": "pending",
  "created_at": "2024-06-07T12:34:56Z"
}
```

2. Mengecek Status Pembayaran:

- Endpoint: `GET /transactions/{transaction_id}`

Response:

json

```
{
  "transaction_id": "trans_001",
  "status": "completed",
  "amount": 1000,
  "currency": "USD",
  "payment_method": "credit_card",
  "description": "Payment for order #1234",
  "customer_id": "cust_001",
  "created_at": "2024-06-07T12:34:56Z",
  "updated_at": "2024-06-07T12:45:00Z"
}
```

3. Melihat Riwayat Transaksi Pelanggan:

- Endpoint: `GET /customers/{customer_id}/transactions`

Response:

json

```
[
  {
    "transaction_id": "trans_001",
    "status": "completed",
    "amount": 1000,
    "currency": "USD",
    "payment_method": "credit_card",
    "description": "Payment for order #1234",
    "created_at": "2024-06-07T12:34:56Z",
    "updated_at": "2024-06-07T12:45:00Z"
  },
  {
    "transaction_id": "trans_002",
    "status": "failed",
    "amount": 2000,
    "currency": "USD",

```

```
"payment_method": "paypal",  
"description": "Payment for order #1235",  
"created_at": "2024-06-07T13:00:00Z",  
"updated_at": "2024-06-07T13:05:00Z"  
}  
]
```

Kebutuhan Non-Fungsional

1. Keamanan:

- Semua endpoint harus dilindungi oleh autentikasi berbasis token (misalnya, JWT).
- Data sensitif seperti informasi kartu kredit tidak boleh disimpan dalam sistem.
- Gunakan enkripsi untuk data sensitif yang perlu disimpan sementara.

2. Skalabilitas:

- Sistem harus mampu menangani setidaknya 1000 transaksi per detik.
- Gunakan mekanisme seperti worker pool untuk memproses transaksi dalam background.

3. Reliabilitas:

- Sistem harus memiliki mekanisme untuk menangani kegagalan transaksi dan memungkinkan retry.
- Gunakan log yang baik dan sistem monitoring.

4. Ketersediaan:

- Sistem harus dirancang untuk memiliki waktu operasional (uptime) minimal 99.9%.

Tugas Anda

1. Desain Basis Data:

- Buatlah skema basis data yang mendukung kebutuhan fungsional di atas. Gunakan Mysql.
- Skema harus mencakup tabel untuk transaksi, pelanggan, dan metode pembayaran.

2. Implementasi API:

- Implementasikan API menggunakan Golang dan framework Gin.
- Pastikan untuk mengimplementasikan autentikasi JWT pada setiap endpoint.
- Gunakan gRPC untuk komunikasi internal antar layanan.

3. Penggunaan Fitur Go:

- Gunakan goroutines dan channels untuk concurrency.
- Implementasikan error handling yang baik dengan penggunaan **errors** package.
- Gunakan **context** untuk mengatur lifecycle dari request.

4. Pengujian:

- Tulis tes unit dan integrasi untuk memastikan setiap endpoint berfungsi sesuai dengan spesifikasi.
- Sertakan skrip untuk menguji beban (load testing) pada API Anda menggunakan k6.

5. Review

Deadline pada tanggal 13 Juni 2024. Siapkan Project untuk di Presentasikan via Google Meet