

Summary

Dokumen ini dibuat untuk menjelaskan alur pengembangan aplikasi untuk **Warehouse Management System**. Dokumen ini akan membahas mengenai rumusan masalah yang terjadi pada sistem tersebut, solusi yang akan diajukan, serta detail informasi mengenai *design system* yang akan diajukan untuk membuat **Warehouse Management System**

Problem and Motivation

Terdapat 5 *Warehouse* di seluruh Indonesia yang menyimpan produk-produk siap beli seperti pakan ternak, vitamin, *smartFeeder*, dsb. Masing-masing *Warehouse* akan melayani permintaan yang dilakukan oleh konsumen dalam aplikasi pada cakupan daerah operasionalnya.

Masing-masing *Warehouse* dikepalai oleh seorang kepala departemen pergudangan (*Warehouse Head*) dan dibantu oleh beberapa oleh beberapa petugas gudang (*Warehouse Officer*)

Aplikasi ini dibuat untuk melayani proses bisnis tersebut dalam cakupan *Warehouse Management* dalam bentuk *Backend API Service*. Beberapa masalah yang ingin diselesaikan yaitu:

- Melayani proses permintaan pemesanan yang dilakukan oleh konsumen melalui aplikasi
- Membantu *Warehouse* dalam mengontrol dan memantau ketersediaan produk yang berada di gudang
- Memfasilitasi proses pengiriman barang yang nantinya akan disalurkan kepada konsumen

Detailed Design

1. Entity Relationship Diagram (ERD)

- Daftar Database table

consumer		
id	String	Primary Key
name	String	
status	String	
address	String	

warehouse		
id	String	Primary Key
name	String	
region	String	
status	String	

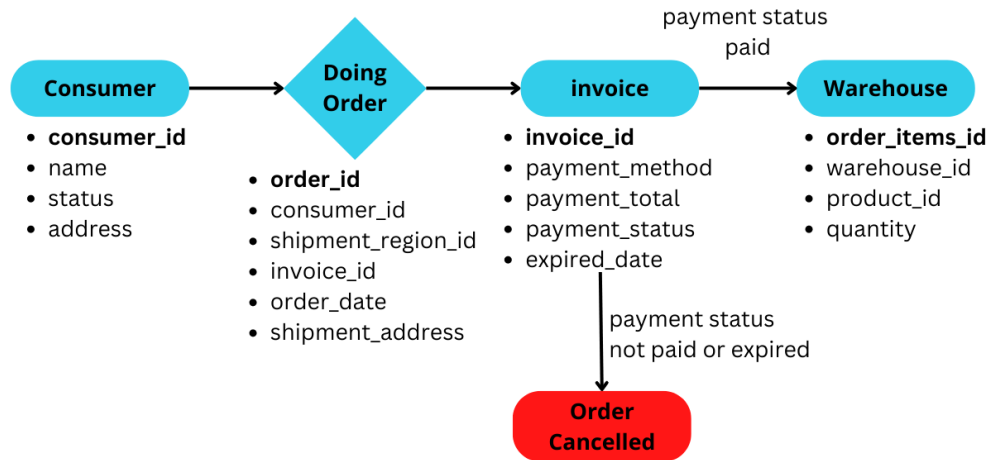
warehouse_product		
id	String	Primary Key
warehouse_id	String	Foreign Key
name	String	
stock	String	
price	String	

order		
id	String	Primary Key
consumer_id	String	Foreign Key
shipment_region_id	String	Foreign Key
invoice_id	String	Foreign Key
order_date	Date	
shipment_address	String	
order_status	String	

order_items		
id	String	Primary Key
order_id	String	Foreign Key
product_id	String	Foreign Key
quantity	Integer	
total_price	Integer	

invoice		
id	String	Primary Key
payment_method	String	
payment_total	Integer	
payment_status	String	
expired_date	Date	

b) Database Relationship



2. API Contract

<u>ADD NEW ORDER</u>
METHOD: POST
ENDPOINT: /order
HEADERS: Authorization: Bearer XXX <JWT Token>, Content-Type: application/json
BODY:

```
{
  "consumer_id": "consumer_id",
  "order_date": "order_date",
  "shipment_address": "shipment_address",
  "order_items": [
    {
      "product_id": "product1",
      "quantity": 5,
      "total_price": 50000
    },
    {
      "product_id": "product2",
      "quantity": 7,
      "total_price": 140000
    }
  ],
  "payment_method": "payment_method",
  "payment_total": 190000
}
```

RESPONSE BODY:

```
{
  "status": 200,
  "order_id": "orderId",
  "invoice_id": "invoiceId",
  "payment_status": "paymentStatus"
}
```

GET LIST OF ORDERS

METHOD: GET

ENDPOINT: /order

HEADERS: Authorization: Bearer XXX <JWT Token>, Content-Type: application/json

BODY:

RESPONSE BODY:

```
{  
  "status": 200,  
  "data": {  
    "order_id": "orderId",  
    "order_date": "orderDate",  
    "consumer_id": "consumerId",  
    "invoice_id": "invoiceId",  
    "shipment_region_id": "shipmentRegion",  
    "shipment_address": "shipmentAddress",  
    "order_status": "orderStatus"  
  }  
}
```