LAPORAN TUGAS AKHIR WEB SERVICE PRAKTIK

FASHIONHUB

Disusun oleh:

5210311051_Ade Maulana Hidayah

5210311068_Fernando Alfredo Simanullang

5210311078_Irfan Zain Azhar

5210311119_Fransiskus Asisi

Dosen Pengampu:

Aditya Ferdiana Arief, S.Kom., M.Kom.

PROGRAM STUDI SISTEM INFORMASI PROGRAM SARJANA

FAKULTAS SAINS & TEKNOLOGI

UNIVERSITAS TEKNOLOGI YOGYAKARTA

2023

Daftar Isi

BAB I 4
Latar Belakang 4
Persiapan Pembuatan projek dari kelompok kami yaitu : 4
Deskripsikan topik Project kalian dan penjelasan mengambil topik tersebut : 5
Tentukan Database yang akan dibuat (SQL/No SQL). Berikan penjelasan dari 5
database tersebut. 5
Desain rancangan Database. 5
Table "orders" (Pesanan): 5
Table "customers" (Pelanggan): 5
Table "products" (Barang): 5
Buat Desain Endpoint dengan minimal 4 Method: GET, POST, PATCH, ... 6
DELETE 6
Menampilkan semua data pelanggan(GET). 6
Menampilkan data orders berdasarkan Id pelanggan(GET by Id) 6
Menginputkan data pelanggan (POST) 7
Update berfungsi untuk mengubah data 8
Delete 8
Framework dan packages yang digunakan. 9
BAB II 10
LANGKAH PENGERJAAN FRONTEND 10
LANGKAH PENGERJAAN BACKEND 10
BAB III 11
Test Backend Restful API Di Postman 11
Get 11
Post 11
Put 12
Delete 12
Test Frontend CRUD Di Web data barang pada Bagian Admin 12
Create data barang 12
Read 13
Update 14
Update berhasil 14

Delete 15
Detele berhasil 15

BAB I

Latar Belakang

FashionHub, sebagai platform inovatif, muncul sejalan dengan transformasi besar-besaran dalam industri fashion yang semakin melibatkan teknologi. Seiring dengan kecenderungan konsumen modern untuk berbelanja secara online, FashionHub menjadi jawaban terhadap kebutuhan akan kenyamanan, variasi produk, dan pengalaman berbelanja yang personal. Dengan fokus pada diversifikasi produk dan desain, FashionHub memberikan kesempatan kepada merek-merek kecil dan desainer independen untuk menjangkau audiens yang lebih besar. Lebih dari sekadar tempat bertransaksi, platform ini juga membangun pengalaman berbelanja yang personal dengan memanfaatkan teknologi cerdas untuk merekomendasikan produk berdasarkan preferensi pengguna.

Melalui kemudahan akses global, FashionHub membuka pintu bagi peritel dan produsen fashion untuk memasarkan produk mereka di seluruh dunia. Dengan keamanan sebagai prioritas utama, platform ini menawarkan sistem pembayaran yang aman, perlindungan bagi pembeli dan penjual, serta kebijakan privasi yang ketat. Selain itu, FashionHub menciptakan kolaborasi dan komunitas di antara penjual, desainer, dan pengguna dengan fitur-fitur seperti forum diskusi, acara khusus, dan promosi bersama. Dengan pendekatan ini, platform ini tidak hanya menjadi wadah untuk bertransaksi, tetapi juga menjadi ekosistem yang mendukung pertumbuhan bersama.

Menggabungkan analisis data untuk memahami perilaku pelanggan, tren belanja, dan preferensi mode, FashionHub membantu penjual merancang strategi pemasaran yang lebih efektif. Sebagai solusi lengkap untuk memperkaya dunia fashion, FashionHub membawa revolusi dalam cara penjualan barang di industri ini, membuka peluang baru dan meredefinisi hubungan antara penjual, pembeli, dan desainer.

Persiapan Pembuatan projek dari kelompok kami yaitu :

Tools yang kami gunakan :

1. Figma
2. Function12
3. Visual code
4. Xampp
5. Phpmyadmin

Framework

```
GET /barang 200 4.946 ms - 660
Terminate batch job (Y/N)?
PS C:\Users\ASUS\Downloads\project-akhir> npm list
Debugger attached.
project-akhir@0.0.0 C:\Users\ASUS\Downloads\project-akhir
├── bcrypt@5.1.1
├── cookie-parser@1.4.6
├── debug@2.6.9
├── dotenv@16.3.1
├── express@4.16.4
├── fastest-validator@1.17.0
├── jsonwebtoken@9.0.2
├── morgan@1.9.1
├── mysql2@3.6.5
├── nodemon@3.0.2
├── sequelize-cli@6.6.2
└── sequelize@6.35.2
```

Deskripsikan topik Project kalian dan penjelasan mengambil topik tersebut :
FashionHub adalah toko fashion daring yang menyediakan pengalaman belanja online yang cepat dan mudah. Dalam sistem ini, penjual dan pembeli dapat melakukan transaksi tanpa kesulitan, dengan fokus pada pemesanan produk fashion berkualitas. FashionHub menawarkan kenyamanan dalam proses pemesanan dan pengiriman yang efisien. Nikmati gaya terkini tanpa hambatan, hanya di FashionHub.

Tentukan Database yang akan dibuat (SQL/No SQL). Berikan penjelasan dari database tersebut.
Kelompok kami menggunakan SQL dipilih karena kemudahan penggunaan, keamanan, fleksibilitas dalam manipulasi data, kemampuan kinerja tinggi, standar industri yang luas, manajemen transaksi yang handal, dan skalabilitas baik untuk data maupun pengguna. Ini menjadikan SQL bahasa yang powerful dan serbaguna dalam manajemen basis data relasional.

Desain rancangan Database.

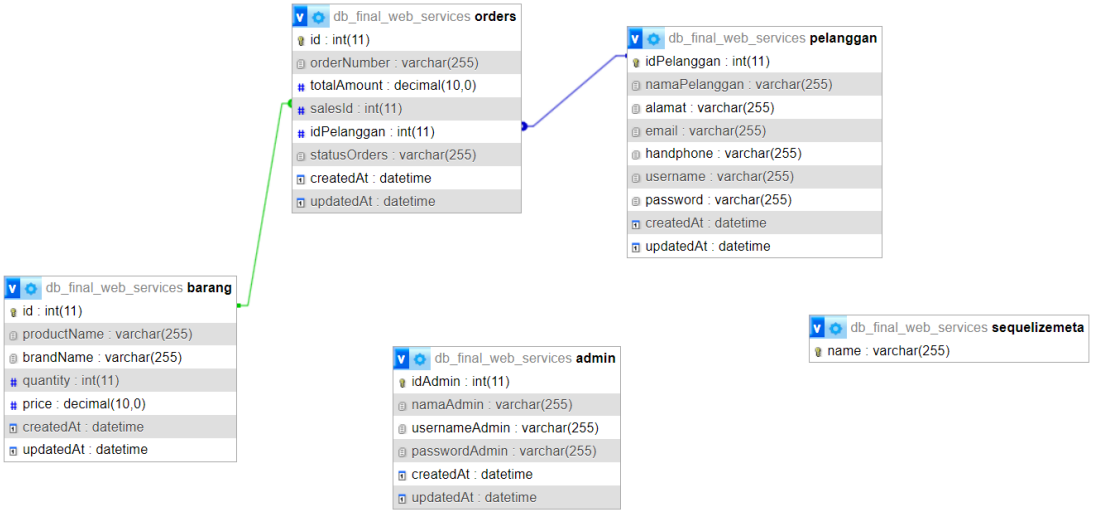


Table orders berrelasi dengan table pelanggan , barang ,admin , sequelizemeta :

Table "orders" (Pesanan):
Tabel ini mewakili informasi tentang pesanan yang ditempatkan oleh pelanggan.Mungkin memiliki kolom-kolom seperti order_id, customer_id, product_id, quantity, dan lainnya.

Table "customers" (Pelanggan):
Tabel ini berisi informasi tentang pelanggan yang melakukan pesanan.

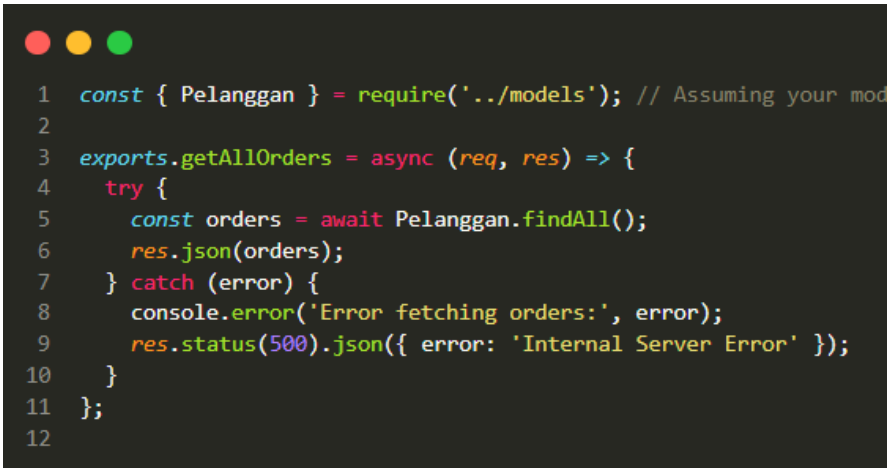
Biasanya memiliki kolom seperti customer_id, name, email, dan lainnya.

Table "products" (Barang):
Tabel ini berisi informasi tentang barang atau produk yang dibeli oleh pelanggan.

Mungkin memiliki kolom seperti product_id, name, price, dan lainnya.

Buat Desain Endpoint dengan minimal 4 Method: GET, POST, PATCH, DELETE.

Menampilkan semua data pelanggan(GET).



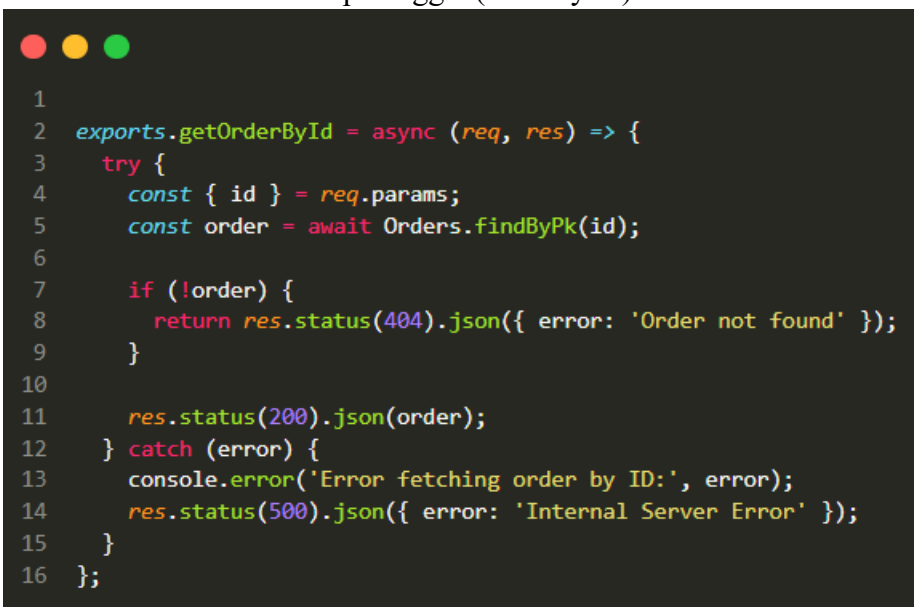
```
const { Pelanggan } = require('../models'); // Assuming your mod

exports.getAllOrders = async (req, res) => {
  try {
    const orders = await Pelanggan.findAll();
    res.json(orders);
  } catch (error) {
    console.error('Error fetching orders:', error);
    res.status(500).json({ error: 'Internal Server Error' });
  }
};

```

Kode tersebut merupakan bagian dari fungsi server Node.js yang menggunakan Sequelize untuk mengambil semua data pelanggan dari tabel "Pelanggan". Jika operasi berhasil, data pelanggan dikirim sebagai respons JSON ke klien. Namun, jika terjadi kesalahan, server memberikan respons status 500 dengan pesan kesalahan. Fungsi ini umumnya digunakan dalam pengembangan aplikasi web untuk menyediakan akses data pelanggan secara efisien.

Menampilkan data orders berdasarkan Id pelanggan(GET by Id)



```

exports.getOrderById = async (req, res) => {
  try {
    const { id } = req.params;
    const order = await Orders.findByPk(id);

    if (!order) {
      return res.status(404).json({ error: 'Order not found' });
    }

    res.status(200).json(order);
  } catch (error) {
    console.error('Error fetching order by ID:', error);
    res.status(500).json({ error: 'Internal Server Error' });
  }
};
```

Fungsi tersebut merupakan bagian dari server Node.js yang menggunakan Sequelize untuk mengambil data pesanan berdasarkan ID dari parameter permintaan (req.params). Jika pesanan ditemukan, server mengirimkan respons sukses (status 200) dengan data pesanan dalam format JSON. Jika pesanan tidak ditemukan, respons mengindikasikan status 404 dengan pesan "Order not found". Jika ada kesalahan selama proses pengambilan data, server memberikan respons status 500 dengan pesan "Internal Server Error". Fungsi ini digunakan untuk mendapatkan detail pesanan berdasarkan ID dalam pengembangan aplikasi web.

Menginputkan data pelanggan (POST)

```
1
2
3  // Create
4  exports.create = async (req, res) => {
5    try {
6      const { productName, brandName, quantity, price } = req.body;
7      const newBarang = await Barang.create({
8        productName,
9        brandName,
10       quantity,
11       price,
12       createdAt: new Date(),
13       updatedAt: new Date(),
14     });
15     res.status(201).json(newBarang);
16   } catch (error) {
17     res.status(500).json({ error: error.message });
18   }
19 };
```

Fungsi tersebut merupakan bagian dari server Node.js dengan Sequelize untuk membuat pesanan baru. Data pesanan diambil dari body permintaan, dan kemudian dilakukan pemeriksaan keberadaan barang dengan salesId yang diberikan. Jika barang tidak ditemukan, server memberikan respons 404 dengan pesan "Barang not found". Jika barang ditemukan, fungsi membuat pesanan baru menggunakan model Orders dan mengirimkannya sebagai respons sukses (status 201). Jika ada kesalahan, server memberikan respons status 500 dengan pesan "Internal Server Error". Fungsi ini digunakan untuk menangani pembuatan pesanan baru dalam aplikasi web.

Update berfungsi untuk mengubah data

```
exports.update = async (req, res) => {
  try {
    const { id } = req.params;
    const { statusOrders } = req.body;

    const updatedOrder = await Orders.update(
      {
        statusOrders,
        updatedAt: new Date(),
      },
      { where: { id } }
    );

    res.status(200).json(updatedOrder);
  } catch (error) {
    res.status(500).json({ error: error.message });
  }
};
```

Fungsi ini merupakan bagian dari server Node.js dan Sequelize untuk memperbarui status pesanan berdasarkan ID. Data statusOrders diambil dari body permintaan, kemudian fungsi menggunakan metode update untuk memperbarui status pesanan di tabel Orders. Respons sukses (status 200) berisi data pesanan yang diperbarui dalam format JSON. Jika ada kesalahan, respons status 500 menyertakan pesan kesalahan dalam format JSON. Fungsi ini digunakan untuk menangani pembaruan status pesanan dalam aplikasi web.

Delete

```
exports.deleteOrder = async (req, res) => {
  try {
    const { id } = req.params;

    // Cek apakah pesanan dengan id yang diberikan ada
    const orderToDelete = await Orders.findByPk(id);
    if (!orderToDelete) {
      return res.status(404).json({ error: 'Order not found' });
    }

    // Hapus pesanan
    await orderToDelete.destroy();

    res.status(200).json({ message: 'Order deleted successfully' });
  } catch (error) {
    console.error('Error deleting order:', error);
    res.status(500).json({ error: 'Internal Server Error' });
  }
};
```

Fungsi ini adalah bagian dari server Node.js yang menggunakan Sequelize untuk menghapus pesanan berdasarkan ID dari parameter permintaan (req.params). Fungsi ini mencari pesanan menggunakan metode findByPk dan menghapusnya menggunakan metode destroy(). Jika pesanan tidak ditemukan, server memberikan respons 404 dengan pesan "Order not found". Jika terjadi kesalahan selama proses penghapusan, respons server adalah status 500 dengan pesan "Internal Server Error". Fungsi ini dirancang untuk menangani penghapusan pesanan dengan singkat dan efisien dalam aplikasi web.

Framework dan packages yang digunakan.

Framework Express, merupakan Framework yang paling banyak dipakai, untuk membuat server jejaring secara sederhana dengan pendekatan minimalis dan berfokus pada inti server.

Packages:

- Nodemon: otomatis restart terhadap aplikasi yang sedang dijalankan.
- Morgan: Packages untuk melihat semua log API kita di nodejs.
- Dotenv: Packages untuk memisahkan konfigurasi dan kode.
- Sequelize: Packages node.js promise-based ORM (Object Relational Mapping)
- untuk Postgre, MySQL
- Mysql2: Packages database (lebih cepat dari Mysql)
- Bcrypt: Packages untuk pengamanan Password
- Fastest validator: Packages untuk membantu memvalidasi objek Javascript.
- Jsonweb token

BAB II

LANGKAH PENGERJAAN FRONTEND

- Pembuatan direktori views
- Menambahkan file html, sesuai data yang dibutuhkan yang ingin didesain
- Mengkonfigurasi koding supaya frontend dapat terhubung dengan backend

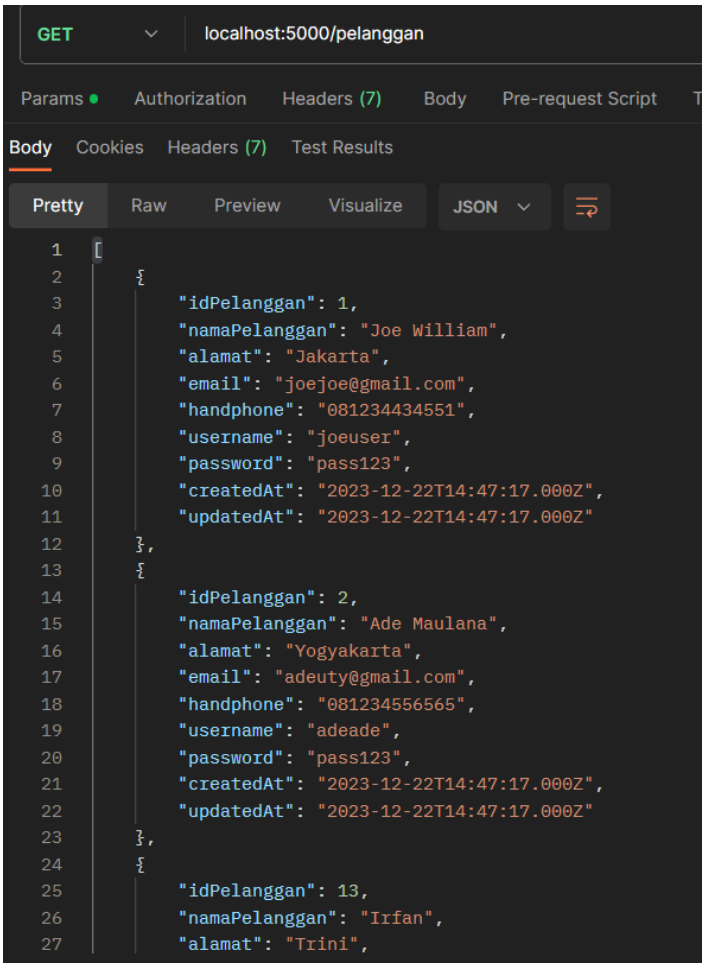
LANGKAH PENGERJAAN BACKEND

- Menginstal package express yang digunakan untuk konfigurasi
- Menginisialisasi npm init
- Konfigurasi environment .env
- Pembuatan router, controller, model
- Pembuatan migration, seeder
- Konfigurasi app.js dengan router controller nya sesuai dengan database yang digunakan

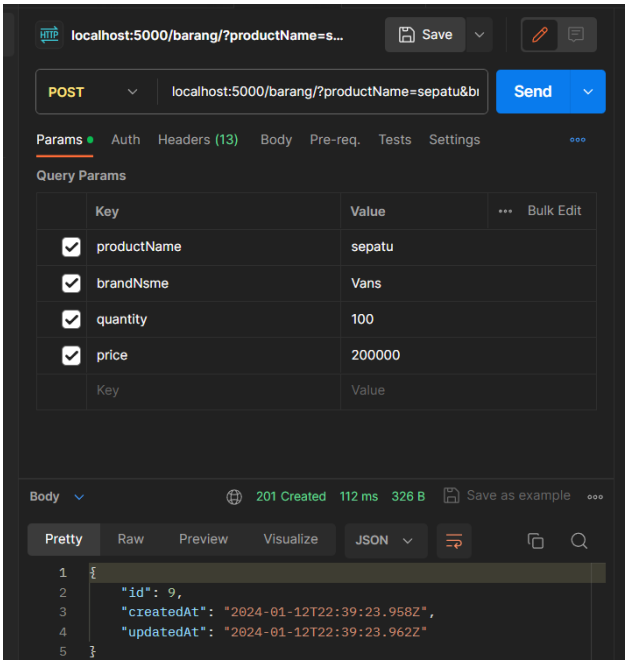
BAB III

Test Backend Restful API Di Postman

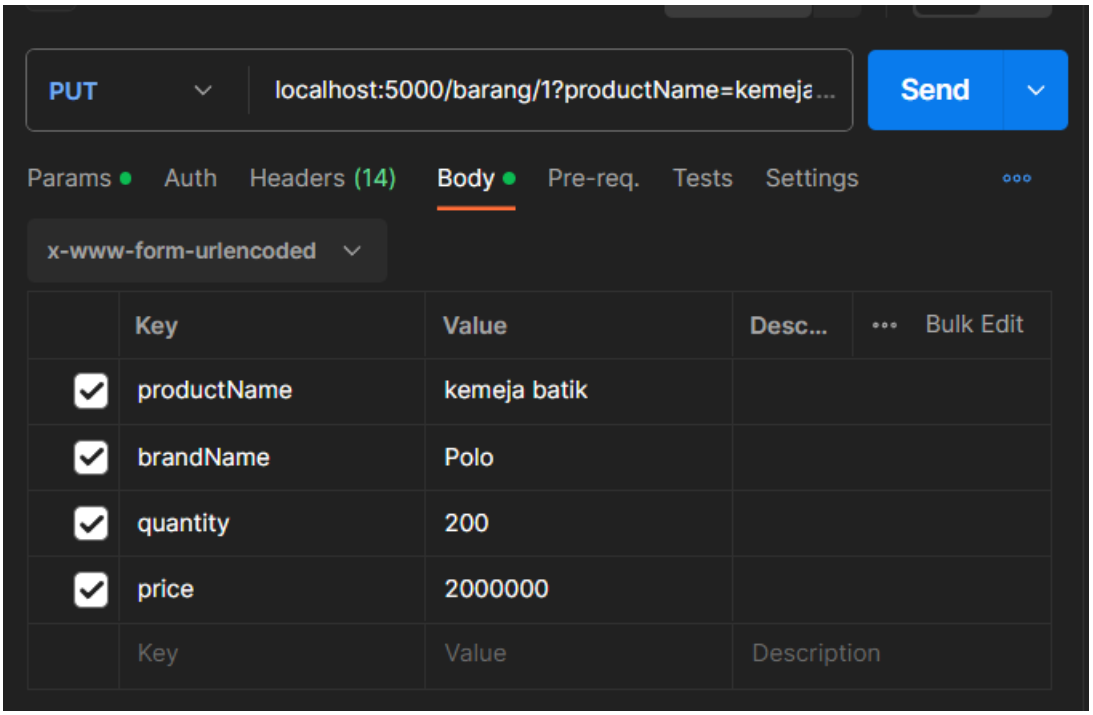
Get



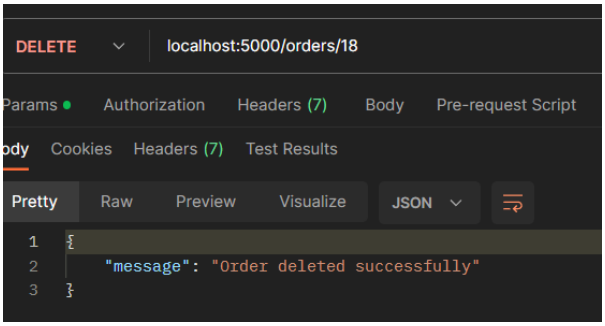
Post



Put

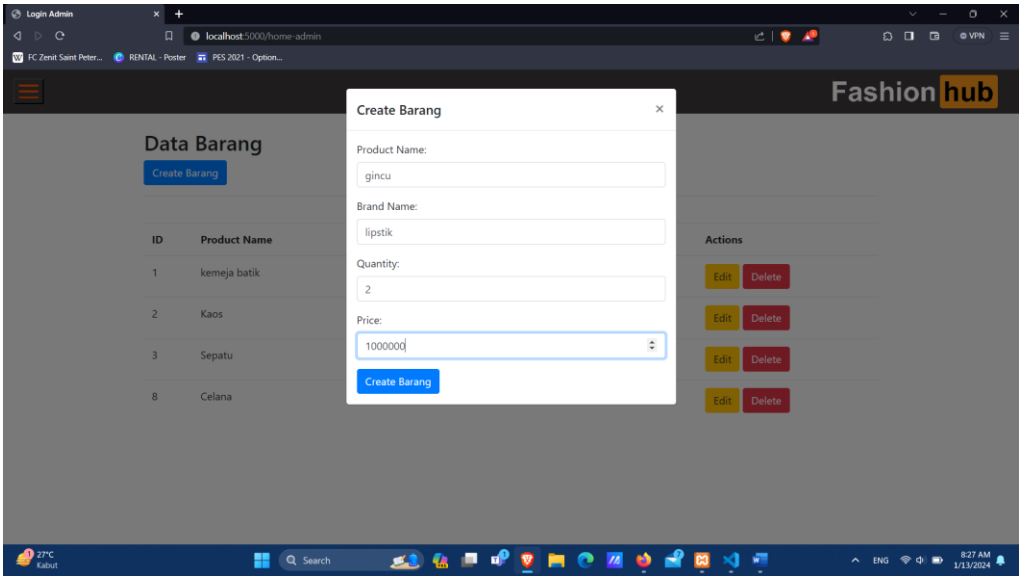


Delete

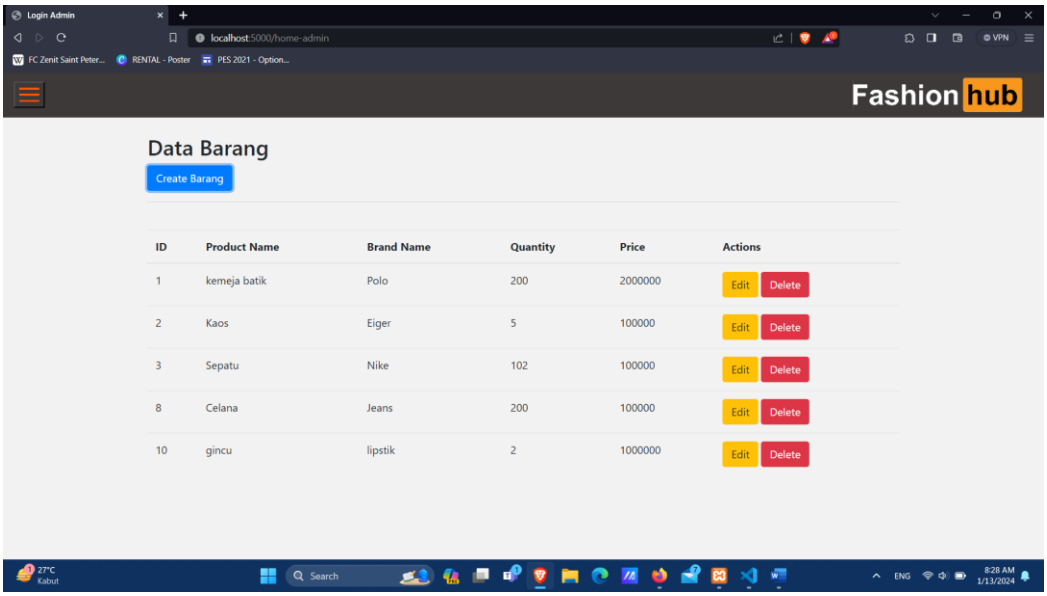


Test Frontend CRUD Di Web data barang pada Bagian Admin

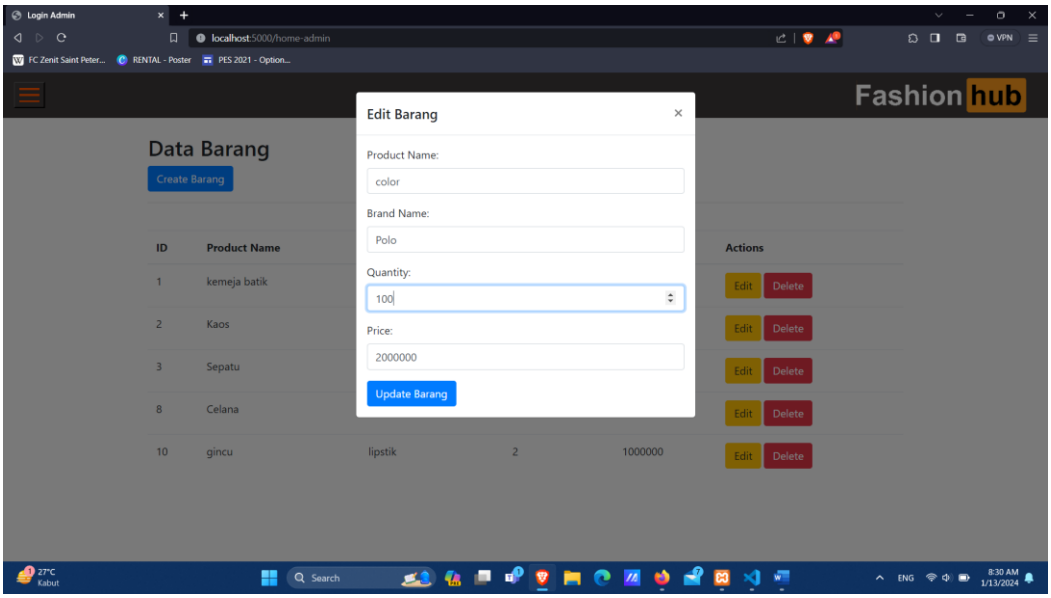
Create data barang


Login Admin
localhost:5000/home-admin
Fashion hub
Data Barang
Create Barang
ID
Product Name
Actions
1
kemeja batik
Edit
Delete
2
Kaos
3
Sepatu
8
Celana
Create Barang
Product Name:
gincu
Brand Name:
lipstik
Quantity:
2
Price:
1000000
Create Barang

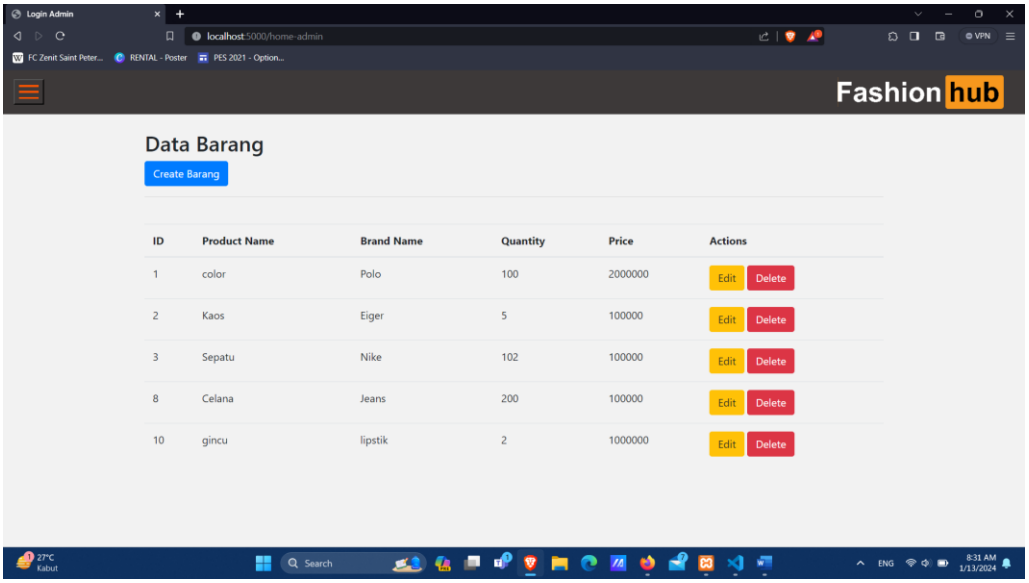
Read


Login Admin
localhost:5000/home-admin
Fashion hub
Data Barang
Create Barang
ID
Product Name
Brand Name
Quantity
Price
Actions
1
kemeja batik
Polo
200
2000000
Edit
Delete
2
Kaos
Eiger
5
100000
3
Sepatu
Nike
102
8
Celana
Jeans
10
gincu
lipstik
2
1000000

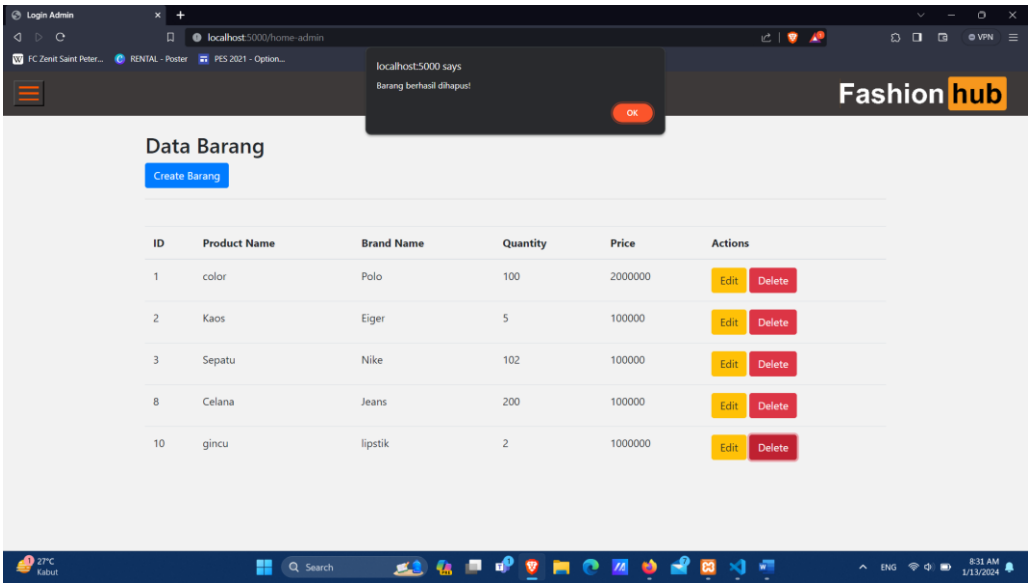
Update



Update berhasil



Delete



Detele berhasil