

## **PROJECT AKHIR PRAKTIKUM WEB SERVICE**

Sistem Informasi Manajemen Inventory di Minimarket Berbasis Website



Disusun oleh :

Adhimas Yoga Pratama	5210311002
Farid Arel Saputra	5210311003
Muhammad Raihan	5210311027
M.Azel Fabian	52103110
Maulana F.	52103110

**UNIVERSITAS TEKNOLOGI YOGYAKARTA**  
**FAKULTAS SAINS & TEKNOLOGI**  
**PRODI SISTEM INFORMASI**

# Daftar Isi

<b>BAB I</b>	2
1.1 Persiapan Membuat Project (System dan beberapa software yang digunakan)	2
1.2 Deskripsi Topik Proyek dan Penjelasan Pemilihan Topik	2-3
1.3 Penentuan Database (MySQL) dan Penjelasan Database Tersebut	3
1.4 Desain Rancangan Database	2
1.5 Desain Endpoint	3-4
1.6 Framework dan Packages yang Digunakan	4-5
<b>BAB II</b>	6
Langkah-langkah membuat project	6
a. Backend	6
b. Frontend	6-7
<b>BAB III</b>	7
PENGETESAN	6
a. Backend	8
a. GET	8
a. GET ID	8
a. POST	9
a. PATCH ID	9
a. DELETE ID	10
b. Frontend	10
a. HOME	10
a. Tampilan CRUD	11
a. Tambah Barang	11
a. Hasil Tambah Barang	12
a. Update Barang	12
a. Hasil Update Barang	13
a. Hasil delete barang	13
Kesimpulan	14

# BAB I

## 1.1 Persiapan Membuat Project (System dan beberapa software yang digunakan)

Pada tahap ini, dilakukan langkah-langkah persiapan proyek sistem inventory barang di Minimarket. Proses dimulai dengan instalasi dan konfigurasi software yang esensial untuk pengembangan. Berikut adalah software yang digunakan:

### 1. Visual Studio Code (VSCode):

Proses dimulai dengan mengunduh dan menginstal Visual Studio Code sebagai editor kode utama. Setelah instalasi, dilakukan konfigurasi ekstensi dan plugin yang mendukung pengembangan web service. Ekstensi seperti "REST Client" mungkin diinstal untuk memudahkan uji coba endpoint API.

### 2. MySQL:

MySQL dipilih sebagai sistem manajemen basis data. Langkah-langkah instalasi MySQL mencakup mengunduh installer resmi, menjalankan proses instalasi standar, dan memasukkan konfigurasi awal seperti nama pengguna dan kata sandi. Selanjutnya, dilakukan konfigurasi tambahan, termasuk pembuatan database untuk menyimpan data inventory.

### 3. Postman:

Proses dimulai dengan mengunduh dan menginstal Postman, sebuah aplikasi yang digunakan untuk pengujian dan dokumentasi endpoint dari web service yang dikembangkan. Setelah instalasi, dilakukan pengaturan awal termasuk pembuatan environment untuk menyimpan variabel-variabel yang akan digunakan dalam pengujian. Postman dapat membantu tim dalam menyusun, menguji, dan mendokumentasikan endpoint API yang terkait dengan manajemen inventaris barang di minimarket.

## 1.2 Deskripsi Topik Proyek dan Penjelasan Pemilihan Topik

Proyek ini memiliki tujuan utama untuk merancang dan mengembangkan sebuah aplikasi web sederhana yang ditekankan pada efektifitas manajemen inventaris barang di lingkungan minimarket. Aplikasi ini dirancang khusus untuk digunakan oleh tim bagian inventory, memberikan kemampuan kepada mereka untuk melaksanakan tiga fungsi pokok dalam pengelolaan inventaris, yakni menambahkan barang baru, mengedit informasi barang yang telah ada, dan menghapus barang dari basis data secara efisien.

Dengan demikian, diharapkan aplikasi ini tidak hanya mempermudah proses administratif, namun juga memberikan kontrol yang lebih baik terhadap pemeliharaan dan akurasi data inventaris. Fokus aplikasi pada tiga fungsi utama tersebut diharapkan akan mempermudah dan mempercepat tugas-tugas harian tim bagian inventory dalam menjaga ketersediaan dan kelangsungan operasional barang di minimarket.

Topik aplikasi ini dipilih karena pada minimarket memiliki sistem inventory yang masih cukup sederhana. Sistem yang dirancang dengan pendekatan yang simple, dengan tujuan untuk memberikan solusi praktis bagi tim bagian inventory, memudahkan proses menambah, mengedit, dan menghapus barang, serta mengurangi potensi kesalahan dalam pengelolaan inventaris.

### 1.3 Penentuan Database (MySQL) dan Penjelasan Database Tersebut

MySQL dipilih sebagai sistem manajemen basis data (RDBMS) untuk proyek ini karena memiliki reputasi yang baik dalam hal manajemen data dan kemampuannya yang terpercaya dalam menyimpan serta mengelola informasi yang krusial bagi sistem inventory. Kelebihan yang dimiliki MySQL, seperti kemampuan transaksi ACID (Atomicity, Consistency, Isolation, Durability) yang menjamin integritas data dan kehandalan, memberikan fondasi yang kuat untuk operasional sehari-hari sistem.

Fleksibilitas struktur data MySQL juga menjadi nilai tambah, memungkinkan penyesuaian optimal terhadap kebutuhan spesifik proyek ini. Dengan kombinasi fitur-fitur tersebut, MySQL menjadi pilihan yang tepat untuk memenuhi tuntutan dan kompleksitas pengelolaan data inventaris di lingkungan minimarket.

### 1.4 Desain Rancangan Database

Pada proyek kali ini, kami memusatkan perhatian pada pengembangan dan implementasi tabel database yang memegang peranan kunci dalam manajemen inventory. Tabel yang menjadi fokus utama adalah tabel "Barang" yang detailnya atributnya akan dijelaskan di bawah ini :

Tabel Barang :

- barang\_id (Primary Key) : ID unik untuk setiap barang.
- nama\_barang : Nama barang untuk identifikasi.
- Kategori : Kategori barang.
- jumlah\_barang : Jumlah stok tersedia.
- harga\_barang : Harga untuk barang yang dijual

### 1.5 Desain Endpoint dengan Minimal 4 Method: GET, POST, PATCH, DELETE

Desain endpoint menjadi aspek krusial dalam implementasi web service sistem inventory barang di minimarket ini. Minimal empat metode HTTP akan diimplementasikan untuk memenuhi kebutuhan akses dan manipulasi data. Berikut adalah penjelasan untuk masing-masing metode:

#### 1. GET (Semua Data):

- **Deskripsi:** Metode ini digunakan untuk mengambil semua data barang atau stok dari sistem.
- **Format Endpoint:** /api/barang

- **Parameter:** Tidak diperlukan parameter tambahan.

## 2. GET (Berdasarkan ID):

- **Deskripsi:** Metode ini digunakan untuk mengambil data barang berdasarkan ID tertentu.
- **Format Endpoint:** `/api/barang/{id}`
- **Parameter:** ID barang yang diinginkan.

## 3. POST:

- **Deskripsi:** Metode ini digunakan untuk menambahkan data barang baru ke dalam sistem.
- **Format Endpoint:** `/api/barang`
- **Parameter:** Data barang yang akan ditambahkan, seperti nama barang, kategori, dan jumlah stok.

## 4. PATCH:

- **Deskripsi:** Metode ini digunakan untuk memperbarui stok barang yang sudah ada dalam sistem.
- **Format Endpoint:** `/api/barang/{id}`
- **Parameter:** ID barang yang akan diperbarui dan data yang akan diubah, seperti jumlah stok baru.

## 5. DELETE:

- **Deskripsi:** Metode ini digunakan untuk menghapus barang dari sistem.
- **Format Endpoint:** `/api/barang/{id}`
- **Parameter:** ID barang yang akan dihapus.

## 1.6 Framework dan Packages yang Digunakan

Dalam proyek pengembangan sistem inventory barang di minimarket, pemilihan framework dan packages memegang peranan penting untuk memastikan kelancaran dan efisiensi dalam pengembangan aplikasi. Berikut adalah framework dan packages yang kami gunakan :

### 1. Visual Studio Code (VSCode):

VSCode diangkat sebagai editor kode utama karena tidak hanya menyediakan antarmuka yang ramah pengguna, tetapi juga mendukung ekstensi yang kaya. Kemampuan ekstensi memungkinkan penambahan fungsionalitas khusus seperti "REST Client", yang sangat berguna untuk melakukan pengujian terhadap endpoint API dengan mudah.

### 2. Express.js:

Express.js dipilih sebagai framework utama untuk membangun aplikasi web server. Dikenal karena kesederhanaan strukturnya dan kemampuan menangani rute dan

permintaan HTTP, Express.js memberikan fondasi yang kokoh untuk pengembangan aplikasi web service.

### **3. MySQL Client:**

Penggunaan klien MySQL sangat penting untuk memastikan koneksi yang efisien dan aman antara aplikasi dan basis data MySQL. Hal ini memungkinkan aplikasi dengan mudah berinteraksi dengan data inventaris di dalam sistem.

### **4. Sequelize:**

Sequelize dipilih sebagai ORM (Object-Relational Mapping) karena mempermudah interaksi dengan basis data MySQL. Dengan Sequelize, proses manipulasi data menjadi lebih abstrak dan terstruktur, sehingga memudahkan pengelolaan dan perubahan struktur database.

### **5. CORS (Cross-Origin Resource Sharing):**

CORS menjadi kunci dalam memastikan keamanan dan aksesibilitas aplikasi web service. Dengan mengizinkan akses lintas domain, CORS memungkinkan aplikasi frontend di domain yang berbeda untuk berkomunikasi dengan aplikasi web service tanpa kendala keamanan di tingkat browser.

## BAB II

### Langkah-langkah membuat project

Langkah awal yang harus dilakukan untuk memulai membuat backend program adalah dengan membuat folder di dokumen dengan nama "MiniMarket". lalu jalankan gitbash untuk membuka aplikasi vscode dengan mengetik query "code ."

#### a. Backend

buat folder baru dengan nama backend di dalam folder MiniMarket pada vscode, lalu buka terminal dan ketikan "cd backend" untuk masuk ke dalam folder backend. kemudian buat file package.json dengan mengetik di terminal "npm init -y", setelah itu dilakukan instalasi package express, mysql client, sequelize untuk relasi database, dan cors agar domain dapat di akses dari luar dengan mengetik di terminal "npm i express mysql2 sequelize cors". Setelah selesai proses instalasi, akan muncul dependencies di package.json dengan isi yang sudah di install sebelumnya.

Di bagian atas entry point index.js akan ditambahkan "type": "module" untuk import dan export module, selanjutnya membuat file "index.js" di dalam folder backend untuk memastikan apakah aplikasi berjalan dengan baik atau tidak. Lalu install nodemon secara global dengan mengetik di terminal "npm i -g nodemon", dan ketik lagi di terminal "nodemon index" karena index adalah entry pointnya. Selanjutnya buat folder baru di dalam backend dengan nama "config" lalu buat file di dalamnya dengan nama "Database.js" untuk membuat koneksi dengan database. Selanjutnya dibuat lah folder lagi di dalam backend dengan nama "models" lalu buat file di dalamnya dengan nama "MiniModel.js" yang berisi database dengan nama tabel dan data di dalamnya dan juga type datanya.

Buat lagi folder di dalam backend dengan nama "controllers" lalu buat file di dalamnya dengan nama "MiniController.js", dan di dalamnya buat method getBarang, buat lagi folder dengan nama "Routes" lalu buat file di dalamnya dengan nama "MiniRoute.js" yang berisi method dan endpoint lalu di panggil dengan "index.js". Setelah di save maka akan terlihat di terminal query yang di eksekusi untuk membuat tabel yaitu tabel "barang" di database, jika dilihat di phpMyAdmin maka akan terlihat tabel baru dengan nama "barang" dengan fieldnya, lalu kami akan melakukan pengujian apakah API berjalan dengan baik dengan menggunakan postman. Di dalam "MiniController.js" kami akan membuat method yang lain yaitu getBarangById, createBarang, updateBarang, deleteBarang lalu dilanjutkan dengan mengulang seperti sebelumnya dan tidak lupa di uji di dalam postman.

#### b. Frontend

Untuk membuat frontendnya, dapat dilakukan dengan cara membuka terminal baru dan mengetikkan "npx create-react-app frontend". Jika instalasi selesai maka akan muncul folder baru di dalam folder "MiniMarket" dengan nama "frontend" lalu untuk masuk ke dalam folder frontend di dalam terminal dapat dilakukan dengan mengetikkan "cd frontend"

kemudian menginstall dependencies yang di butuhkan dengan mengetikkan "npm i react-router-dom axios bulma" karna akan membutuhkan router, dan menggunakan bulma css untuk style nya.

Setelah di install maka di dalam package.json akan terdapat beberapa dependencies baru yang sudah terinstall, lalu untuk mengecek apakah projek berjalan baik dapat mengetik di terminal dengan "npm start". Jika berjalan dengan baik maka akan pindah ke browser dan menampilkan localhost yang telah berjalan, setelah itu baru membuat frontend yang dimau seperti membuat tabel yang berisi data dan aksi untuk edit dan delete data, lalu buat button di atasnya untuk fitur tambah barang.

Selanjutnya buat navigasi ke home yang hanya menampilkan tabel data barang saja dan buat juga navigasi disana dengan nama CRUD yang berisi tampilan sebelumnya di tambah dan update juga buat navigasi untuk form yang di gunakan untuk mengisi data barang baru atau update barang yang telah ada, lalu buat method di dalam "BarangList.js" untuk menghubungkan dengan backend nya dengan axios, buat file baru di dalam components dengan nama "AddBarang.js" dan "EditBarang.js" yang berisi form untuk menambahkan dan mengupdate barang yang ada di dalam tabel, lalu buat file "Home.js" untuk tampilan yang hanya ada tabel barangnya saja, semua file di dalam components ubah bahasanya menjadi JavaScript, lalu hubungkan dengan route di dalam "App.js" agar bisa tampil di web.



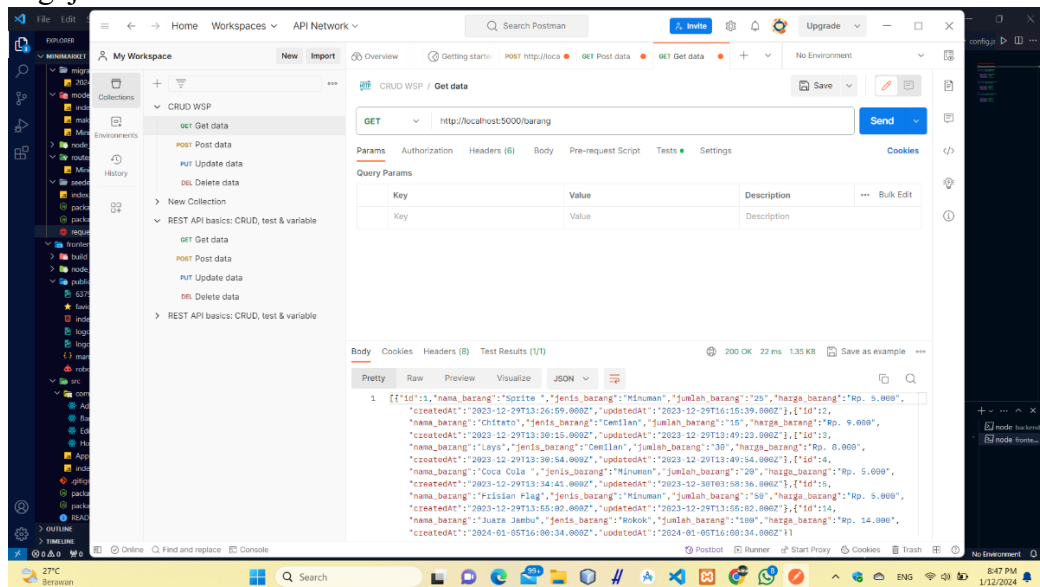
## BAB III

### PENGETESAN

#### a. Backend

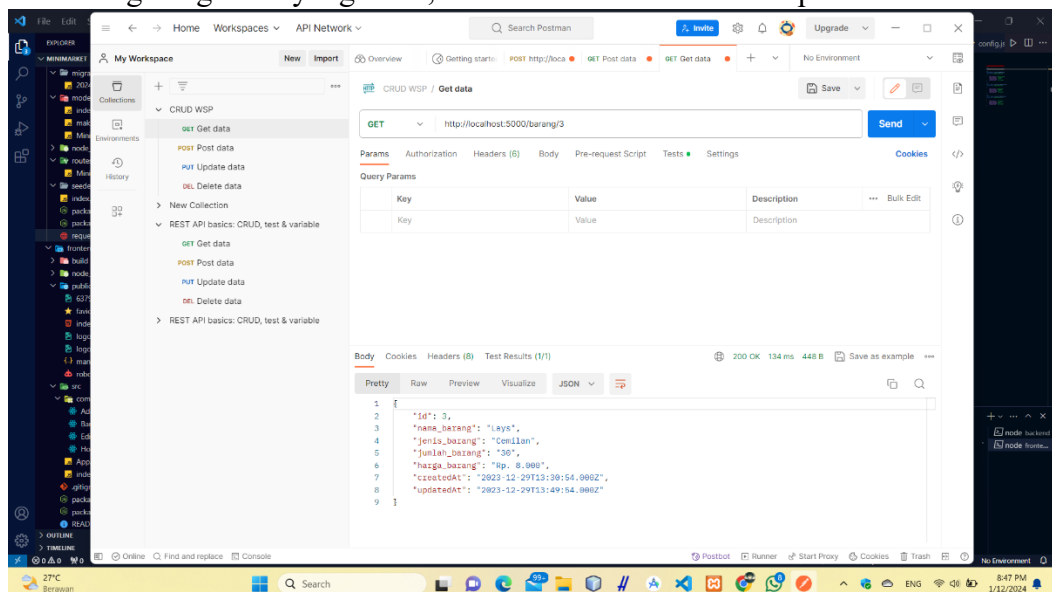
##### - GET

Pengujian method “get” dilakukan untuk menampilkan semua data barang yang tersimpan di dalam database. Dari hasil pengujian ini, didapatkan hasil bahwa method dapat menampilkan data dengan lengkap, yang menandakan bahwa pengujian kali ini berhasil.



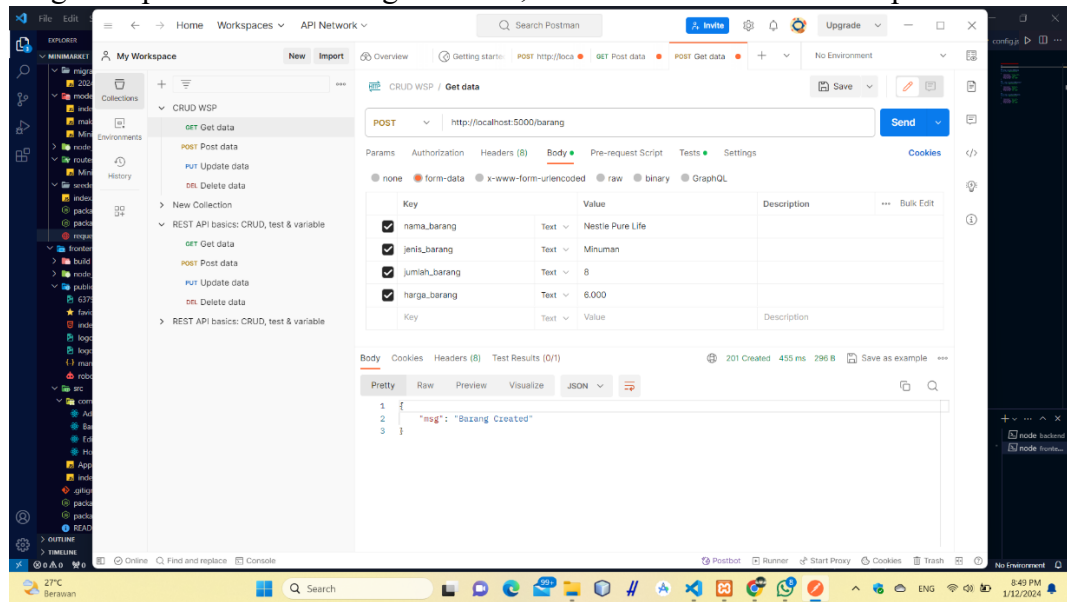
##### - GET ID

Pengujian untuk mengambil data barang berdasarkan ID tertentu dari sistem. Hasil pengujian menunjukkan bahwa metode dapat memperoleh informasi detail dari barang dengan ID yang valid, menandakan keberhasilan operasi.



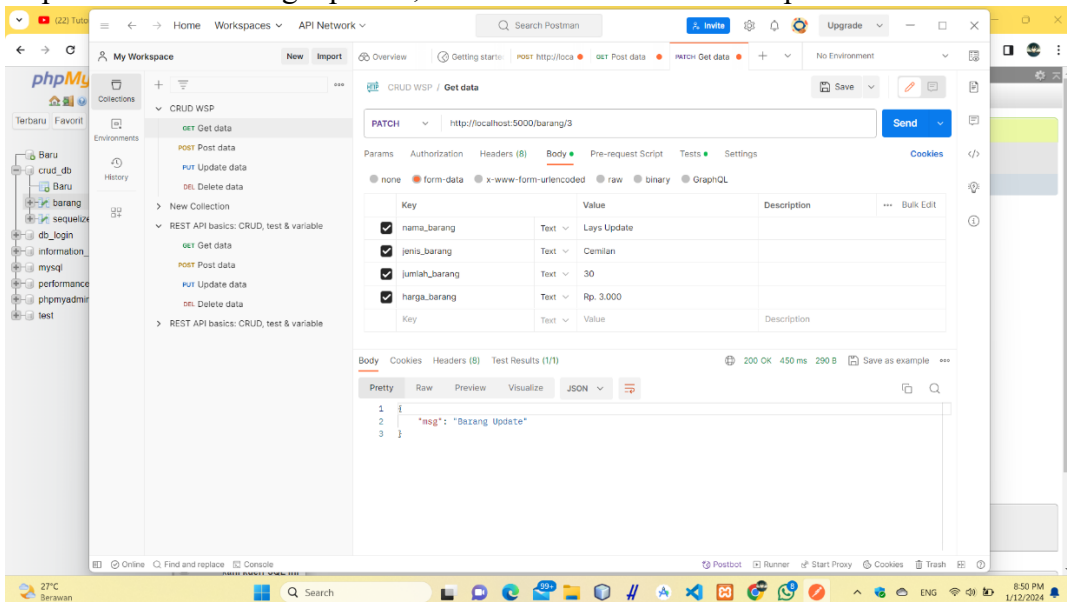
## - POST

Pengujian dilakukan untuk menambahkan data barang baru ke dalam sistem. Hasil pengujian menunjukkan bahwa metode berhasil menambahkan data baru dengan respons status “Barang Created”, menandakan keberhasilan operasi.



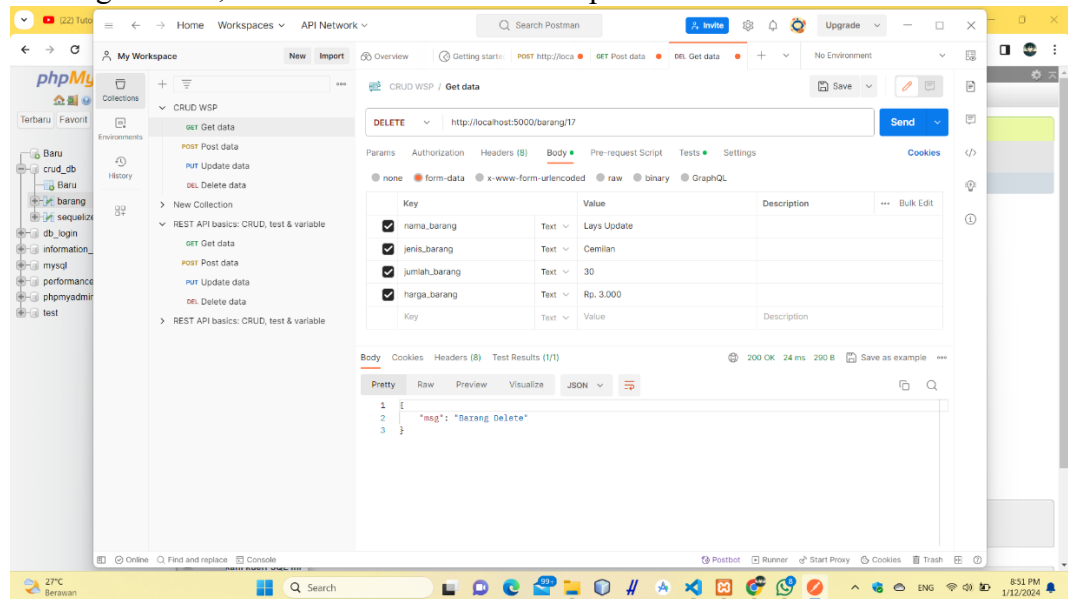
## - PATCH ID

Pengujian untuk memperbarui stok barang yang sudah ada dalam sistem. Hasil pengujian menunjukkan bahwa metode berhasil memperbarui stok barang dengan respons status “Barang Update”, menandakan keberhasilan operasi.



## - DELETE ID

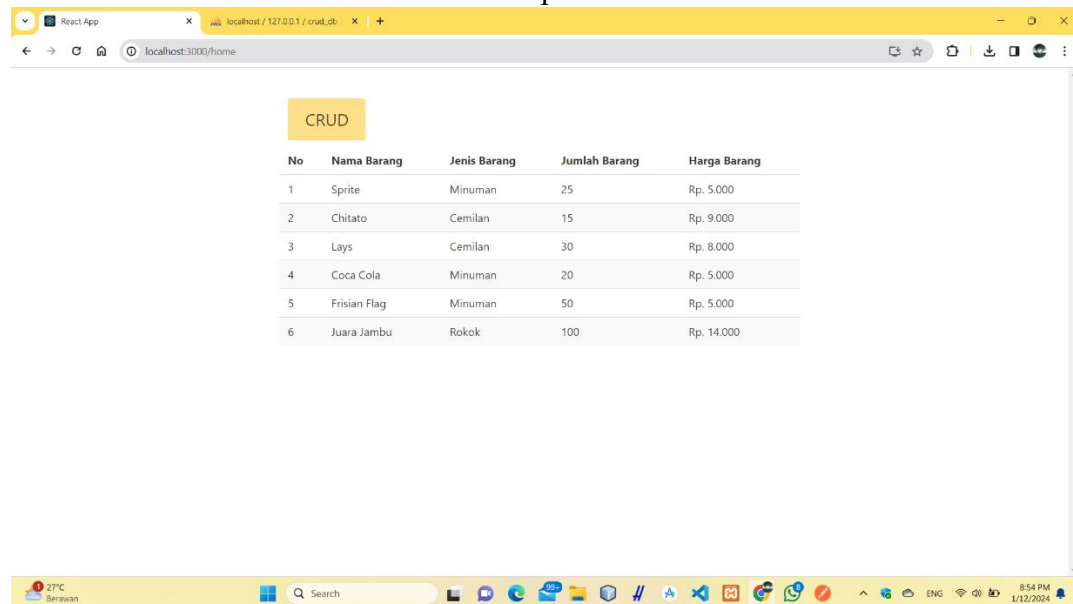
Pengujian dilakukan untuk menghapus barang dari sistem. Hasil pengujian menunjukkan bahwa metode berhasil menghapus barang dengan respons status “Barang Delete”, menandakan keberhasilan operasi.



## b. Frontend

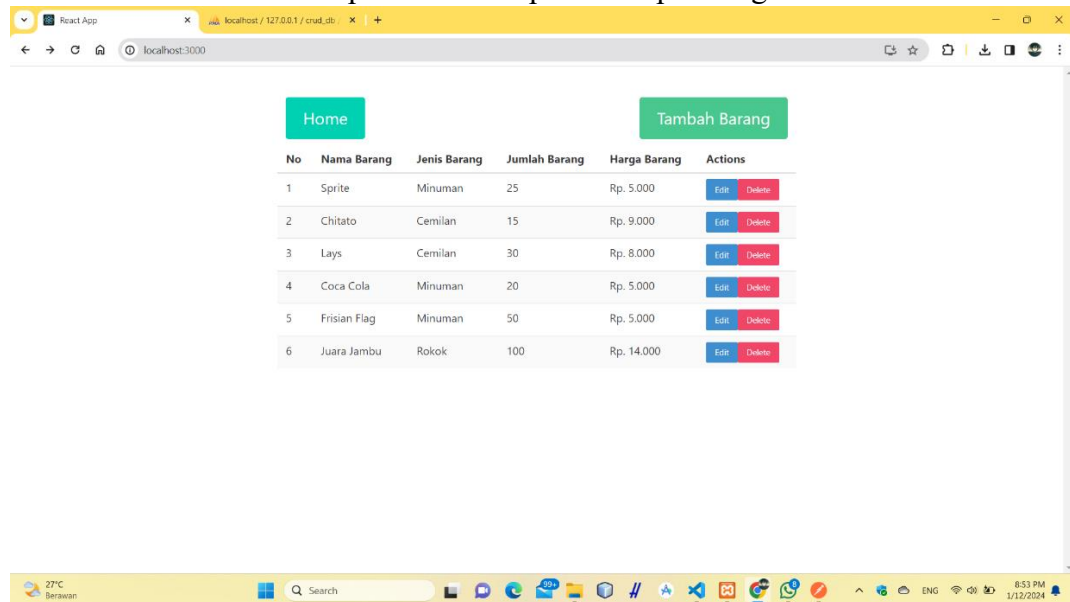
## - HOME

Halaman HOME yang menampilkan informasi barang yang ada di minimarket. Serta memberikan tautan ke halaman tampilan CRUD



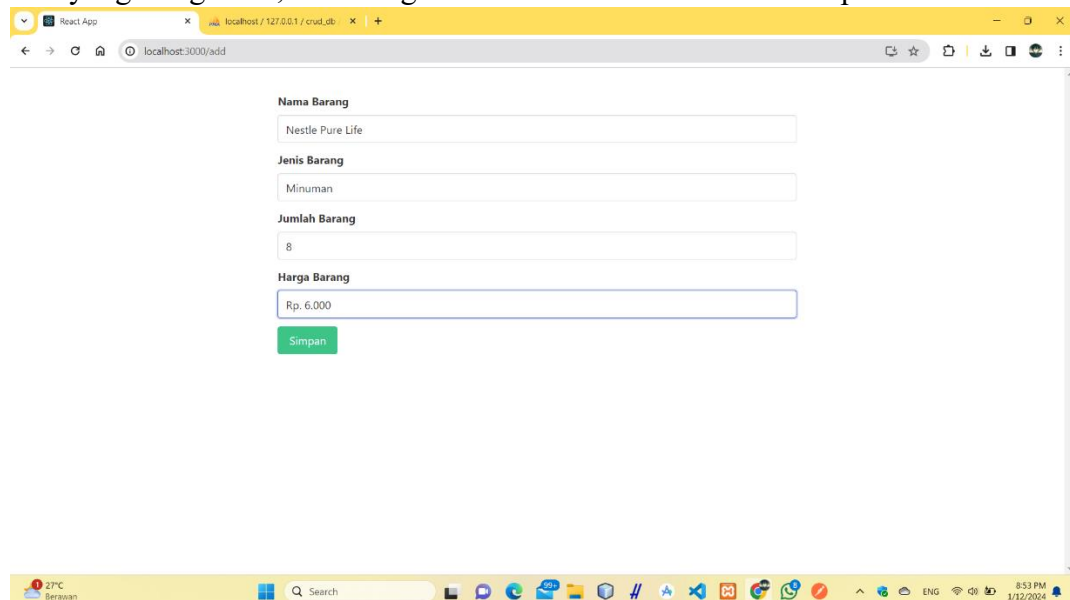
## - Tampilan CRUD

Halaman ini menampilkan daftar lengkap barang yang tersimpan di database. Setiap barang memiliki opsi untuk mengedit, atau menghapus. Memberikan akses mudah untuk melakukan operasi CRUD pada setiap barang.



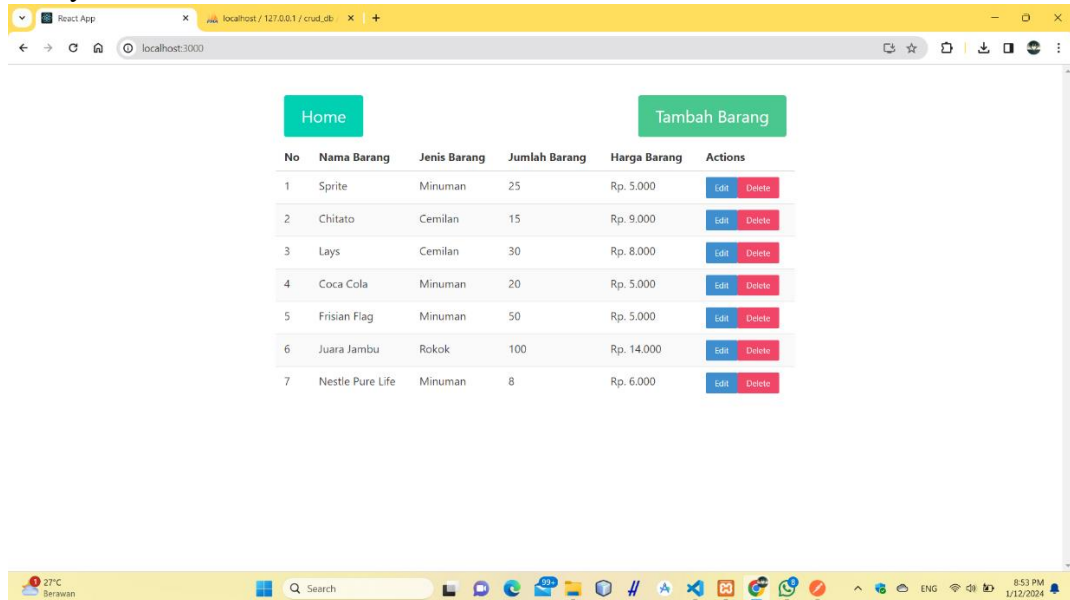
## - Tambah Barang

Halaman formulir ini, berfungsi untuk menambahkan barang baru ke dalam sistem. Pengguna dapat memasukkan informasi seperti nama barang, jenis, jumlah stok yang diinginkan, serta harga sebelum menekan tombol "Simpan".



- Hasil tambah barang

Halaman konfirmasi yang muncul setelah pengguna berhasil menambahkan barang baru. Memberikan informasi tentang berhasilnya operasi penambahan barang dan memberikan opsi untuk kembali ke tampilan CRUD atau melanjutkan operasi lainnya.



- Update Barang

Halaman formulir untuk memperbarui informasi atau stok dari suatu barang. Pengguna dapat melihat nilai-nilai saat ini dan mengedit sesuai dengan kebutuhan sebelum menekan tombol "Update".

**Nama Barang**

**Jenis Barang**

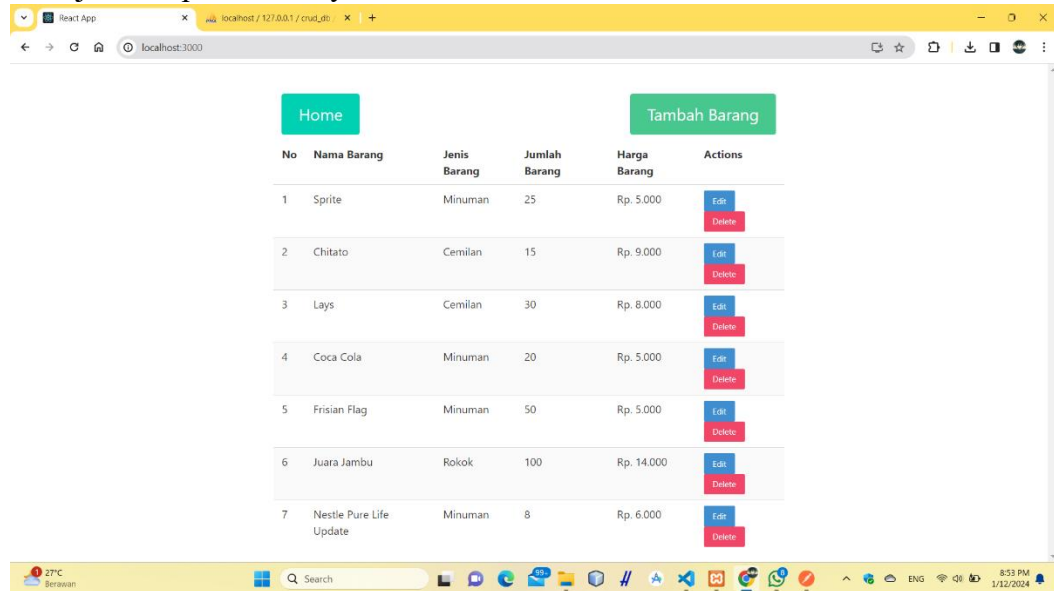
**Jumlah Barang**

**Harga Barang**

[Update](#)

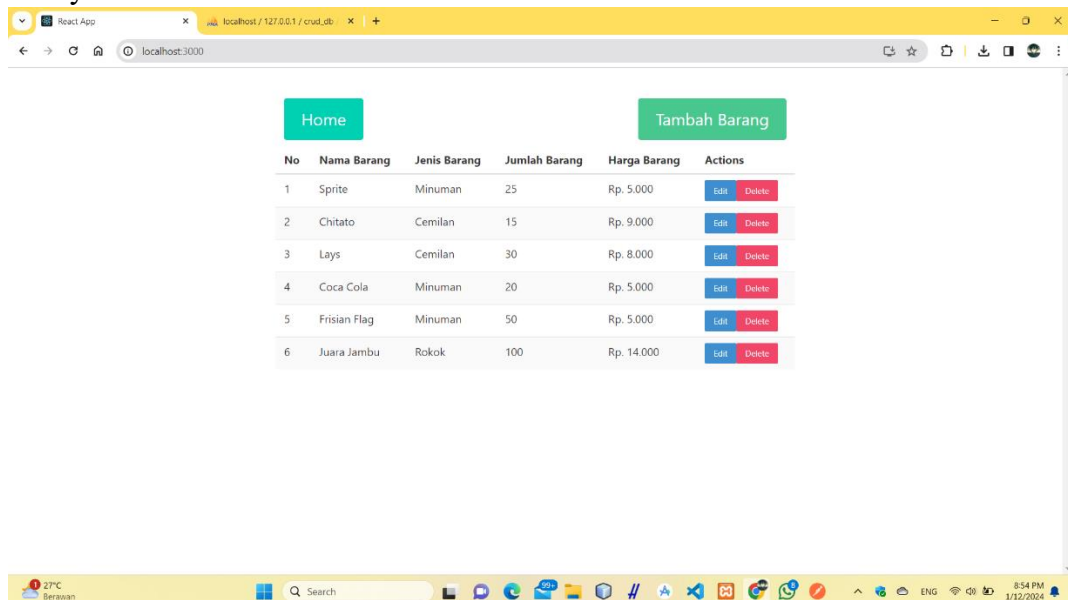
- Hasil update barang

Halaman konfirmasi yang muncul setelah pengguna berhasil memperbarui informasi atau stok barang. Menyediakan informasi tentang berhasilnya operasi pembaruan dan memberikan opsi untuk kembali ke tampilan CRUD atau melanjutkan operasi lainnya.



- Hasil delete barang

Halaman konfirmasi yang muncul setelah pengguna berhasil menghapus suatu barang. Menyajikan informasi tentang berhasilnya operasi penghapusan dan memberikan opsi untuk kembali ke tampilan CRUD atau melanjutkan operasi lainnya.



## Kesimpulan

Dalam project pengembangan sistem inventory barang di minimarket ini, kami mengarahkan fokus pada kesederhanaan dan efisiensi aplikasi. Pemilihan untuk menitikberatkan pada tabel barang dalam desain database membantu menciptakan pemahaman yang jelas dan terorganisir untuk manajemen stok barang. Sementara itu, implementasi endpoint API yang mencakup operasi CRUD berhasil melewati uji coba dengan baik, memberikan respons yang diharapkan dan memvalidasi keberhasilan fungsionalitas sistem.

Antarmuka pengguna yang dirancang dengan simple di bagian frontend memberikan pengalaman yang responsif dan mudah dipahami. Dari halaman Tambah Barang hingga Hasil Hapus Barang, setiap operasi memberikan konfirmasi yang jelas, memastikan pengguna merasa yakin dan terhubung dengan setiap langkah yang mereka ambil. Kami berharap bahwa dengan pendekatan aplikasi yang sederhana ini memberikan kejelasan dalam pengelolaan inventory barang. Dengan penutupan laporan ini, harapan kami adalah solusi yang telah kami kembangkan dapat memenuhi kebutuhan yang diinginkan.