

TUGAS WEB SERVIS

RESTFUL API E-Commerce Computer dan Laptop

Dosen Pengampu - Adityo Permana Wibowo, S.Kom., M.Cs.



Disusun Oleh :

1. 5210311054_Dwi Aldi
2. 5210311061_Lilik Hananto Putro
3. 5210311063_Riyo Fendi Prakosa
4. 5210311116_Riendra Abkar Sukamto

PROGRAM STUDI SISTEM INFORMASI PROGRAM SARJANA

FAKULTAS SAINS & TEKNOLOGI

UNIVERSITAS TEKNOLOGI YOGYAKARTA

2024

DAFTAR ISI

BAB I	3
1.1 Persiapan Membuat Project.....	3
1.2 Deskripsikan Topik Project	3
1.3 Tentukan Database	4
1.4 Rancangan Database	4
1.5 Desain Endpoint	4
1.6 Framework dan Packages.....	5
BAB 2	6
2.1 Langkah-Langkah Membuat Project.....	6
2.1.1 Backend	6
2.1.2 Fronend.....	7
BAB 3	9
3.1 Test Backend Restful API	9
3.1.1 Test Backend Produk	9
3.1.2 Test Backend User	11
3.2 Test Frontend.....	14

BAB I

1.1 Persiapan Membuat Project

Untuk membuat project restful API untuk tampilan administrasi ecommerce, diperlukan beberapa software dan sistem pendukung, yaitu:

- Visual Studio Code sebagai editor kode
- MySQL sebagai alat desain database
- Postman untuk melakukan pengujian API

1.2 Deskripsikan Topik Project

Proyek ini bertujuan untuk mengembangkan sebuah sistem administrasi e-commerce yang memungkinkan administrator (admin) untuk melakukan operasi CRUD (Create, Read, Update, Delete) pada produk dan pengguna. Dalam konteks e-commerce, memiliki sistem administrasi yang efisien dan dapat diandalkan sangat penting untuk memastikan pengelolaan inventaris produk yang baik dan manajemen pengguna yang efektif.

Alasan memilih topik ini:

- 1) Manajemen Produk: Dengan fokus pada operasi CRUD produk, sistem ini memungkinkan admin untuk dengan mudah menambah, mengedit, atau menghapus produk dari katalog e-commerce.
- 2) Manajemen Pengguna: CRUD pada data pengguna memberikan fleksibilitas dalam manajemen akun pengguna, seperti menambahkan pengguna baru, mengubah informasi pengguna, atau menghapus akun yang tidak aktif.
- 3) Keberlanjutan Bisnis: Dengan adanya sistem administrasi yang handal, bisnis e-commerce dapat memastikan bahwa informasi produk selalu terkini, dan penggunaan data pelanggan dapat dikelola dengan baik.

Manfaat proyek ini:

- 1) Efisiensi Administrasi: Mempermudah admin dalam mengelola produk dan pengguna, menghemat waktu, dan meningkatkan efisiensi operasional.
- 2) Peningkatan Layanan Pelanggan: Data produk yang akurat dan manajemen pengguna yang efisien dapat berdampak positif pada pengalaman pelanggan.

1.3 Tentukan Database

MySQL dipilih sebagai sistem manajemen basis data (DBMS) untuk proyek administrasi e-commerce karena beberapa alasan yang kuat. MySQL dikenal karena keandalan dan kinerjanya yang tinggi, mampu menangani beban kerja tinggi dan memberikan respon cepat. Selain itu, adanya komunitas pengguna yang besar memberikan dukungan, tutorial, dan pembaruan berkala, menjadikannya pilihan yang stabil.

1.4 Rancangan Database

Untuk membuat database MySQL yang mendukung sistem administrasi e-commerce dengan fokus pada CRUD produk dan pengguna, kita dapat merancang skema database berikut:

Product

`name` (String, Nama Produk)
`harga` (Integer, Harga Produk)
`description` (Text, Deskripsi Produk)
`image` (String, Foto Produk)
`url` (String, Link Foto)

Users

`name` (String, Nama Pengguna)
`email` (String, Alamat email Pengguna)
`password` (String, Password)
`role` (String, Kategori Pengguna)

1.5 Desain Endpoint

Endpoint adalah titik akhir untuk mengakses API. Endpoint untuk product terdiri dari empat method, yaitu:

- GET `/products` untuk mendapatkan daftar produk.
- GET `/products/:id` untuk mendapatkan daftar produk sesuai dengan id produk.
- POST `/products` untuk membuat atau menambah produk baru.

- PUT `‘/products/:id’` untuk melakukan perubahan data produk sesuai dengan id produk.
- DELETE `‘/products/:id’` untuk menghapus data barang sesuai dengan id produk.

Endpoint untuk users juga terdiri dari empat method, yaitu:

- GET `‘/users’` untuk mendapatkan daftar user yang terdiri dari pelanggan dan admin.
- GET `‘/users/:id’` untuk mendapatkan data user sesuai dengan id user.
- POST `‘/users’` untuk membuat atau menambah data user baru.
- PUT `‘/users:id’` untuk melakukan perubahan data user sesuai dengan id user.
- DELETE `‘/users:id’` untuk menghapus data user sesuai dengan id user.

1.6 Framework dan Packages

Untuk membuat backend restful API, digunakan framework Express.js dan packages Sequelize, Argon2, CORS, dan Dotenv.

- Express.js adalah framework web yang populer dan mudah digunakan.
- Sequelize adalah library ORM yang memudahkan dalam berinteraksi dengan database.
- Argon2 adalah library hashing yang aman untuk menyimpan password.
- CORS adalah library yang digunakan untuk mengizinkan akses dari sumber daya lain.
- Dotenv adalah library yang digunakan untuk membaca variabel lingkungan.

Sedangkan untuk frontend menggunakan packages berikut ini:

- React & Redux

React adalah library JavaScript yang digunakan untuk membangun user interface (UI) berbasis komponen. Redux adalah library state management yang digunakan untuk mengelola state aplikasi secara global dan terpusat. React digunakan untuk membangun struktur UI dari komponen-komponen kecil yang dapat dikombinasikan untuk membentuk UI yang kompleks. Redux digunakan untuk mengelola state aplikasi, seperti data produk dan pengguna.

- Bootstrap & Bulma

Bootstrap adalah framework CSS populer yang digunakan untuk membangun UI yang responsif dan mobile-first. Bulma adalah framework CSS modern yang berfokus pada modularitas dan kemudahan penggunaan.

Bootstrap dan Bulma digunakan untuk memberikan styling dan komponen UI dasar untuk aplikasi. Bootstrap lebih populer dan memiliki lebih banyak fitur, sedangkan Bulma lebih modern dan minimalis.

- **Axios**
Axios adalah library untuk melakukan request HTTP ke API. Axios digunakan untuk melakukan request ke API backend untuk mengambil data produk dan pengguna.
- **react-router-dom**
react-router-dom adalah library untuk menangani routing (navigasi) dalam aplikasi React. react-router-dom digunakan untuk mengontrol navigasi halaman dalam aplikasi.
- **Testing libraries**
Testing libraries digunakan untuk melakukan pengujian unit dan integrasi. Testing libraries digunakan untuk memastikan bahwa aplikasi berfungsi dengan benar.

BAB 2

2.1 Langkah-Langkah Membuat Project

2.1.1 Backend

Langkah-langkah membuat project restful API untuk ecommerce adalah sebagai berikut:

- 1) Buat folder project dan buka di Visual Studio Code.
- 2) Install packages.json dengan menggunakan perintah berikut:
`npm init -y`
- 3) Install depedensi yang dibutuhkan dengan menggunakan perintah berikut i:
`npm i express mysql2 sequelize argon2 cors dotenv`
- 4) Install express seassion dengan menggunakan perintah berikut:
`npm install express - seassion`
- 5) Pada package.json bagian type ubah menjadi “module”
- 6) Buat file index.js

- 7) Buat file .env
- 8) Install nodemon dengan menggunakan perintah berikut ini :
`npm install --save-dev nodemon`
- 9) Buat beberapa folder yaitu folder config, model, controller, route, middleware
- 10) Didalam folder config buat file database.js
- 11) Buat database pada mysql dengan nama responsi_ws
- 12) Di folder models buat file baru dengan nama UserModel.js dan ProductModel.js
- 13) Di folder controller buat file baru dengan nama Product.js dan Users.js
- 14) Di folder route buat file baru dengan nama UserRoute.js dan ProductRoute.js
- 15) Untuk membuat table users dan product pada database dengan cara menambahkan kode beriku ini pada file index.js

```
import db from "../config/Database.js";
(async () =>{
    await db.sync();
}) ();
```
- 16) Di folder controller buat file baru dengan nama Auth.js
- 17) Di folder routes buat file baru dengan nama AuthRoute.js
- 18) Setelah itu coba jalankan dengan perintah berikut ini :
`nodemon index`
- 19) Install depedensi untuk menyimpan data kedalam database dengan perintah berikut ini :
`npm i connect-session-sequelize`
- 20) Di folder middleware buat file baru dengan nama AuthUser.js

2.1.2 Fronend

Langkah- Langkah membuat fronend adalah sebagai berikut :

- 1) Buka terminal di Vscode, disini kami menggunakan gitbash
- 2) Ketik ``npx create-react-app frontend -template redux``. Tunggu hingga selesai prosesnya hingga folder frontend terbuat.
- 3) Selanjutnya menginstal dependensi yang di perlukan. Ketik ``npm i react-router-dom axios bulma react-icons``. Tunggu hingga selesai prosesnya

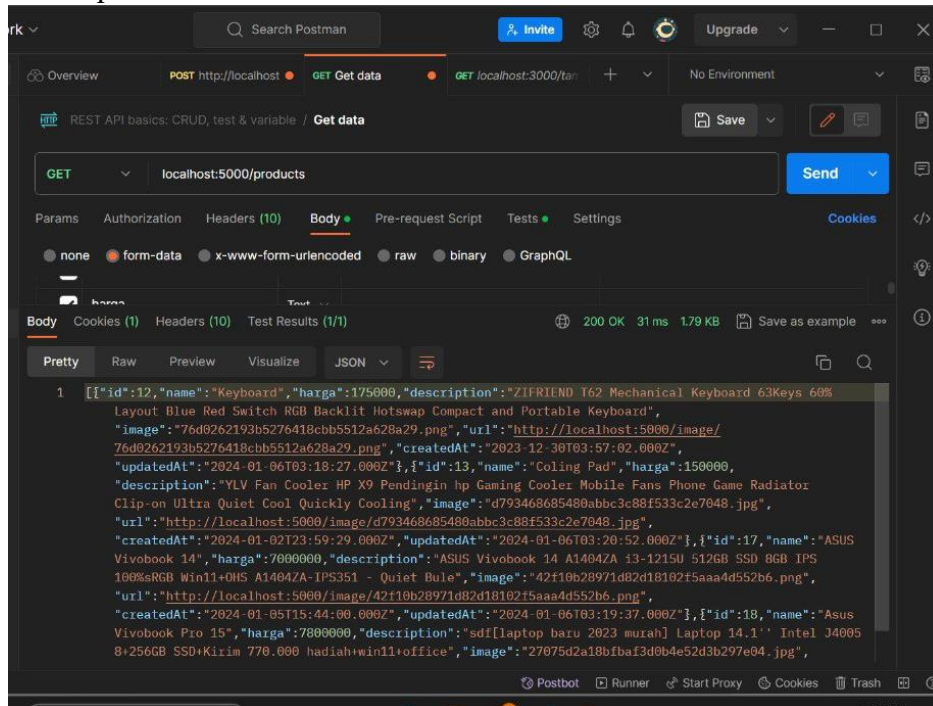
- 4) Pada folder frontend, lalu buat component di `src-components`. Berisi Navbar.jsx, Sidebar.jsx. Untuk tampilan frontend dashboard
- 5) Buat pages di `src-pages`, dengan nama Layout.jsx, Dashboard.jsx
- 6) Buat component `Welcome.jsx`. lalu isi dengan classname title, subtitle.
- 7) Lalu pergi ke `App.js`. impor react-router-dom lalu impor BrowserRouter, dan routes
- 8) Buat element BrowserRouter, lalu isi dengan atribut /dashboard yang telah dibuat
- 9) Buka browser lalu akses `localhost port 5000/dashboard`. Maka tampilan dashboard akan muncul.
- 10) langkah selanjutnya membuat frontend Login, dan frontend lainnya di component, lalu membuat pages sesuai dengan component yang telah dibuat. Seperti membuat tampilan awal dashboard dengan membuat autentikasi dengan backend.
- 11) Membuat Features authSlice dan Redux, di folder features berisi authSlice.js
- 12) Mengisi component Login.jsx agar dapat memasukkan data dari database.
- 13) Membuat get user login & logout, dengan menambahkan query getme,logOut di authSlice.js
- 14) Membuat Perlindungan Dashboard dan semua pages agar mendapat perlindungan akses.
- 15) Mengisi component ProductList.jsx menggunakan useState dan useEffect dan juga melakukan link dari react-router-dom
- 16) Import axios agar dapat melakukan interact dengan API, lalu tambah query getProduct di ProdukList.jsx agar dapat menampilkan data produk dari database.
- 17) Buat endpoint menggunakan axios.get.
- 18) membuat button deleteProduct di ProductList.jsx lalu buat endpoint menggunakan axios.get yang akan melakukan point to /products.
- 19) Membuat akses semua produk agar dapat melakukan perintah saat menekan button CRUD
- 20) Membuat Button Edit product di ProductList.jsx
- 21) Mengisi component UseList.jsx, dengan melakukan impor axios from axios agar dapat melakukan link dari react-router-down
- 22) Mengisi component FormAddUser.jsx dengan perintah Add new user dan update user di tampilan frontend.
- 23) Running frontend menggunakan npm start

BAB 3

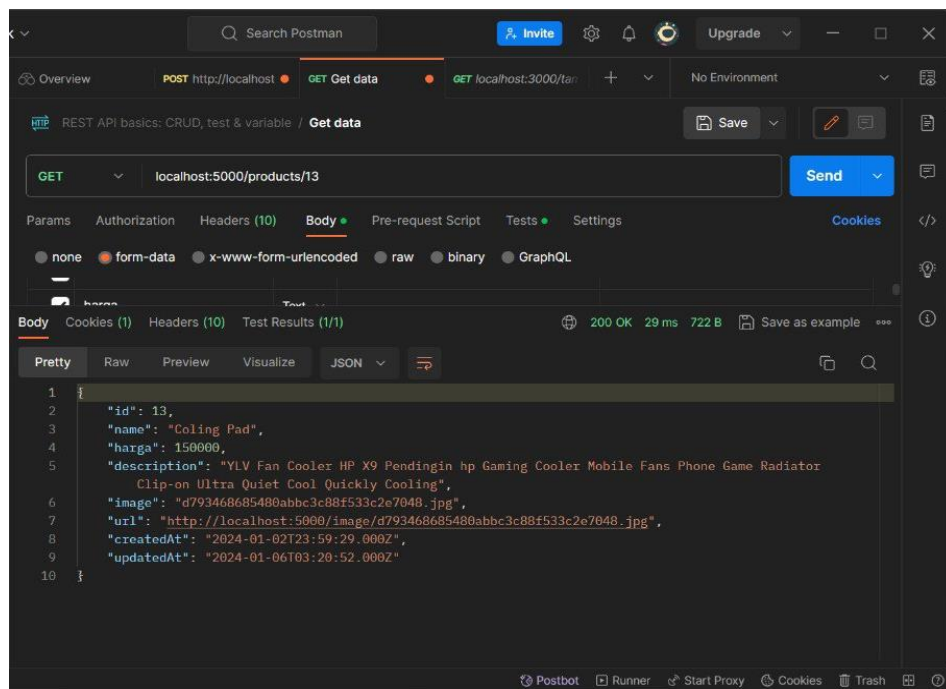
3.1 Test Backend Restful API

3.1.1 Test Backend Produk

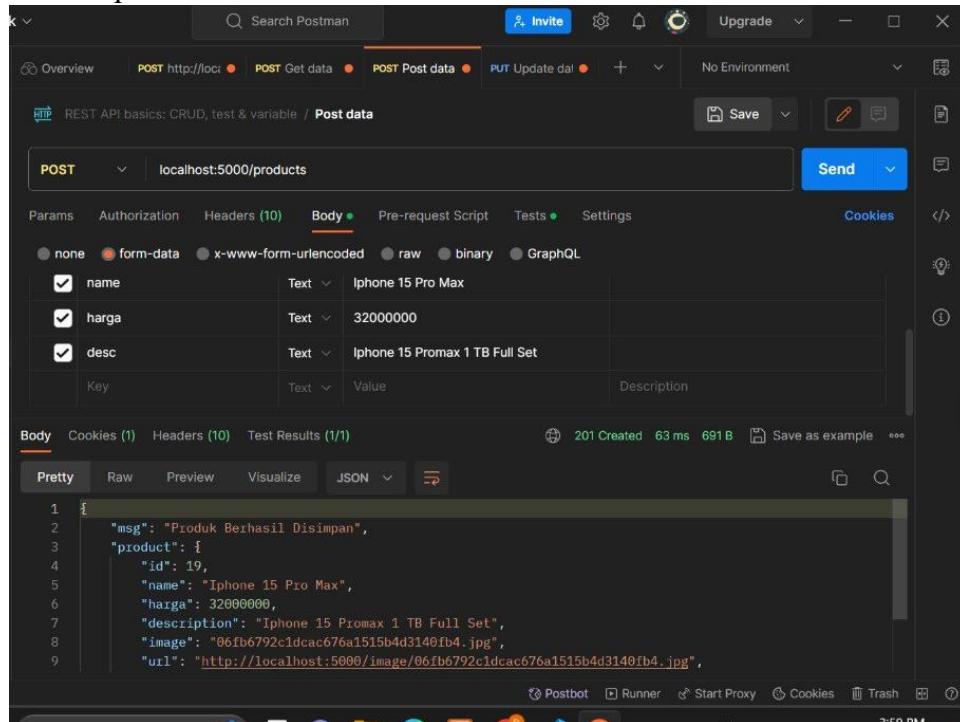
- GET '/products'



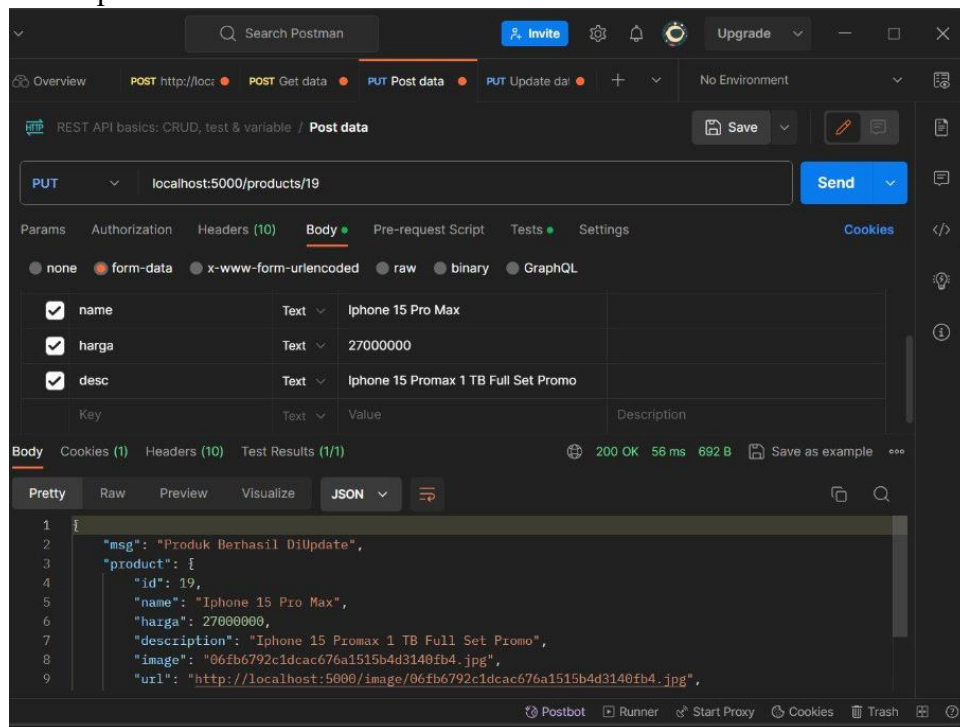
- GET '/products/:id'



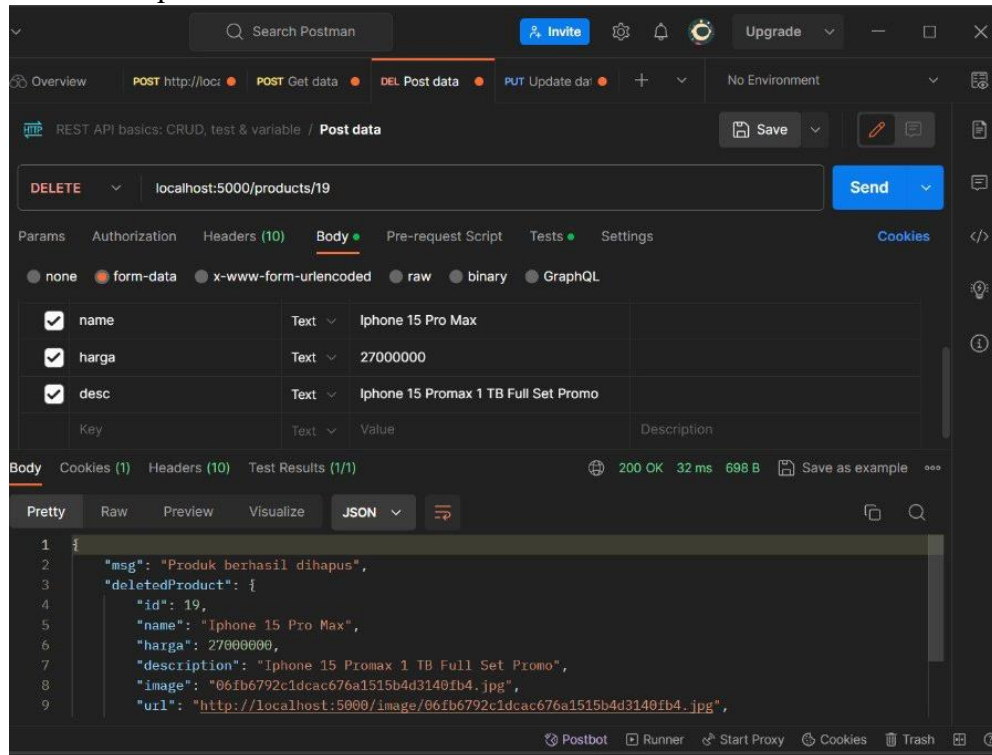
- POST '/products'



- PUT '/products/:id'

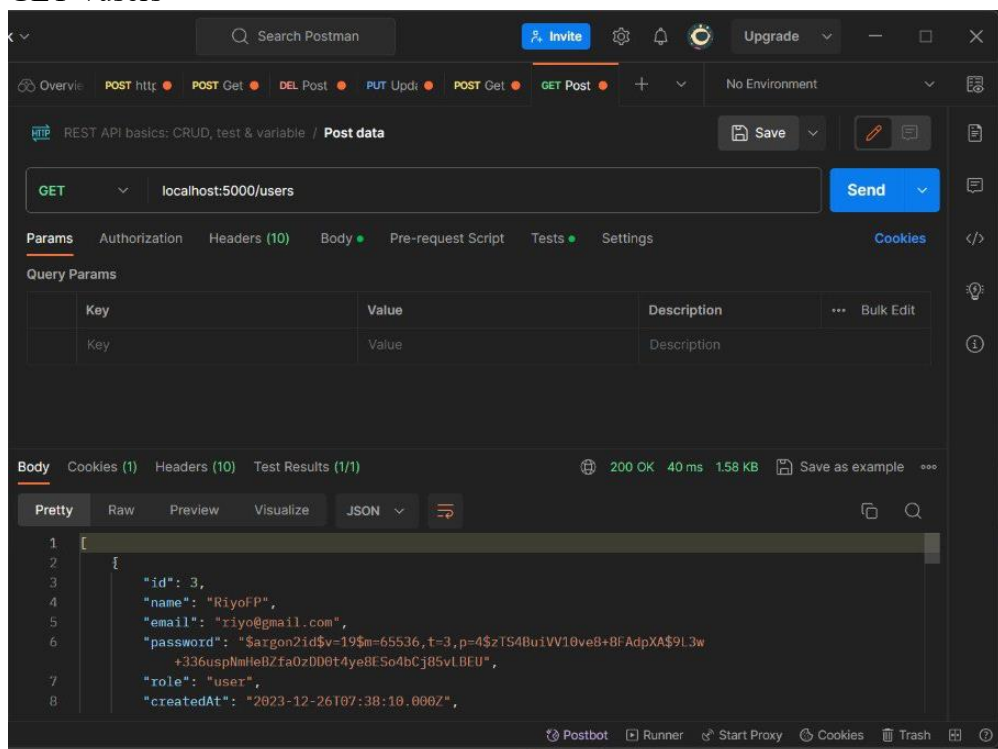


- DELETE '/products/:id'

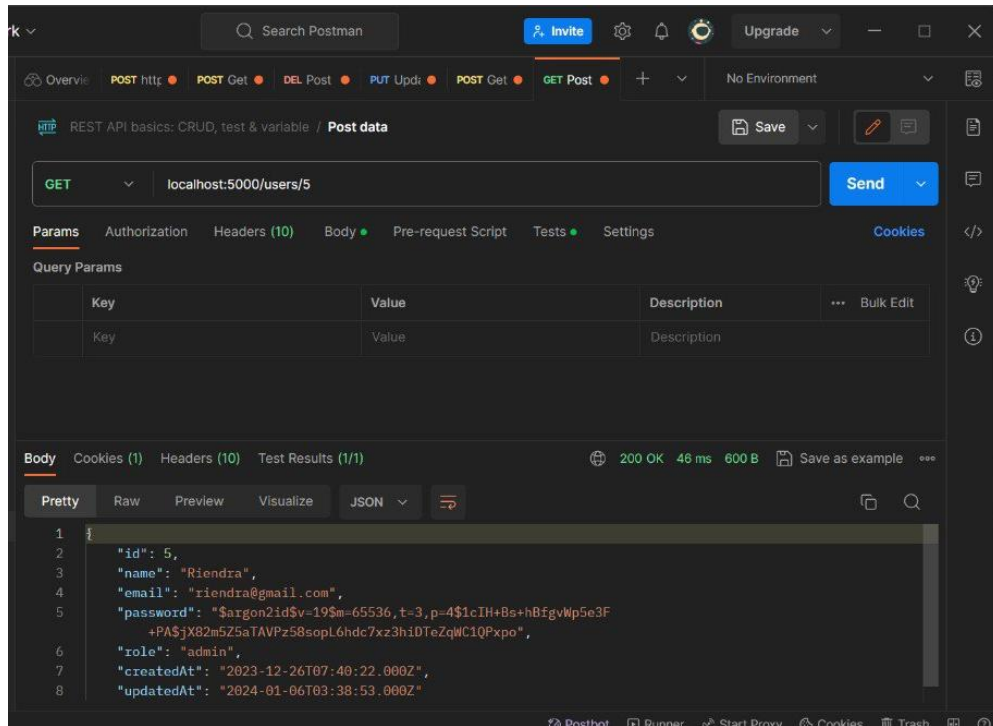


3.1.2 Test Backend User

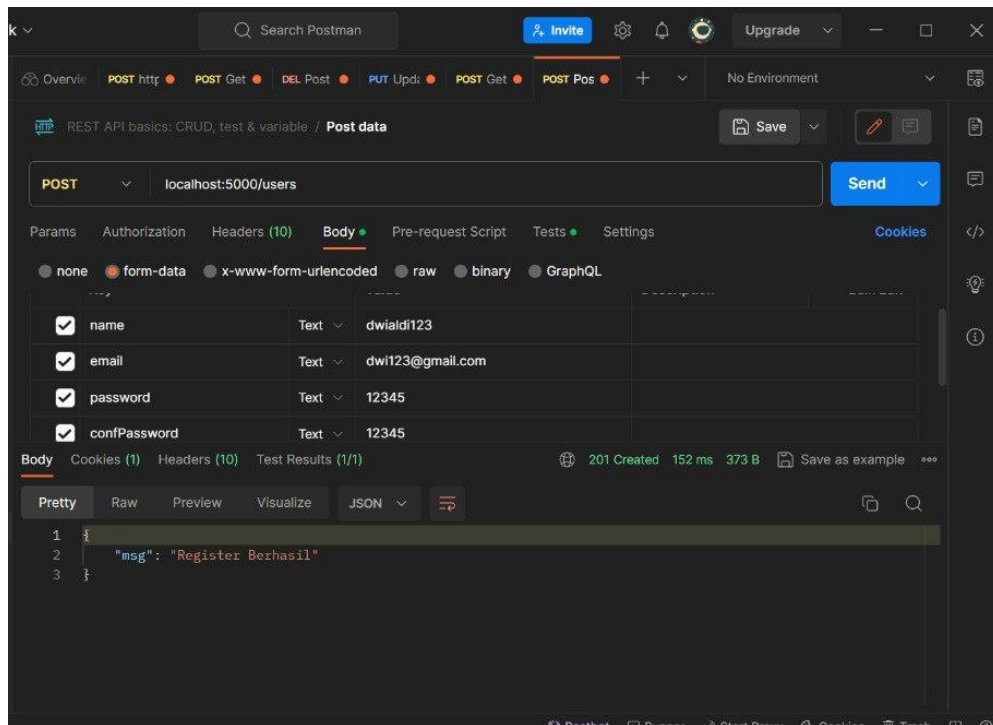
- GET '/users'



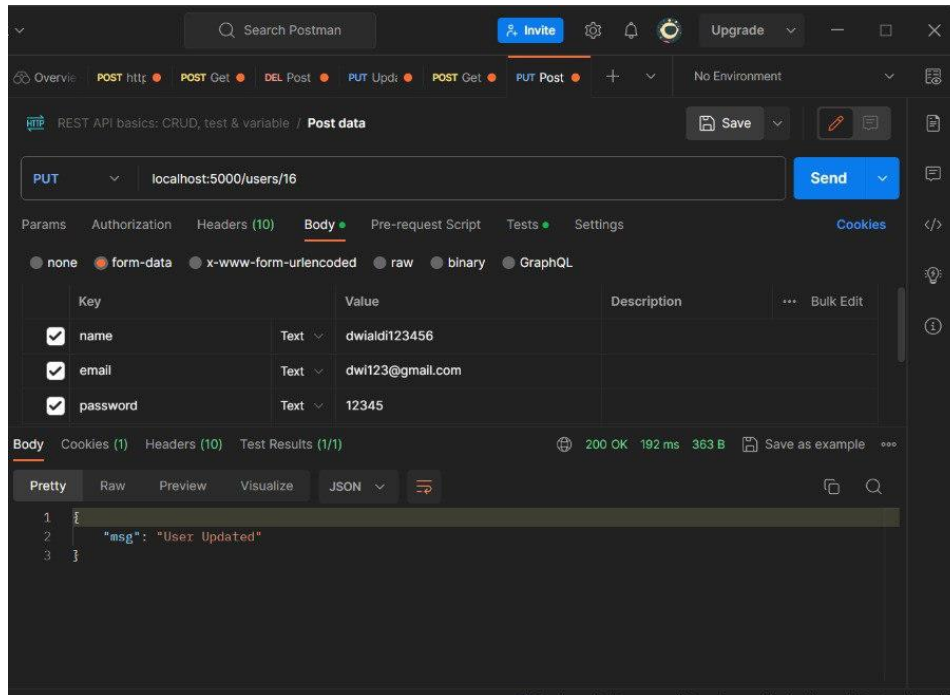
- GET '/users/:id'



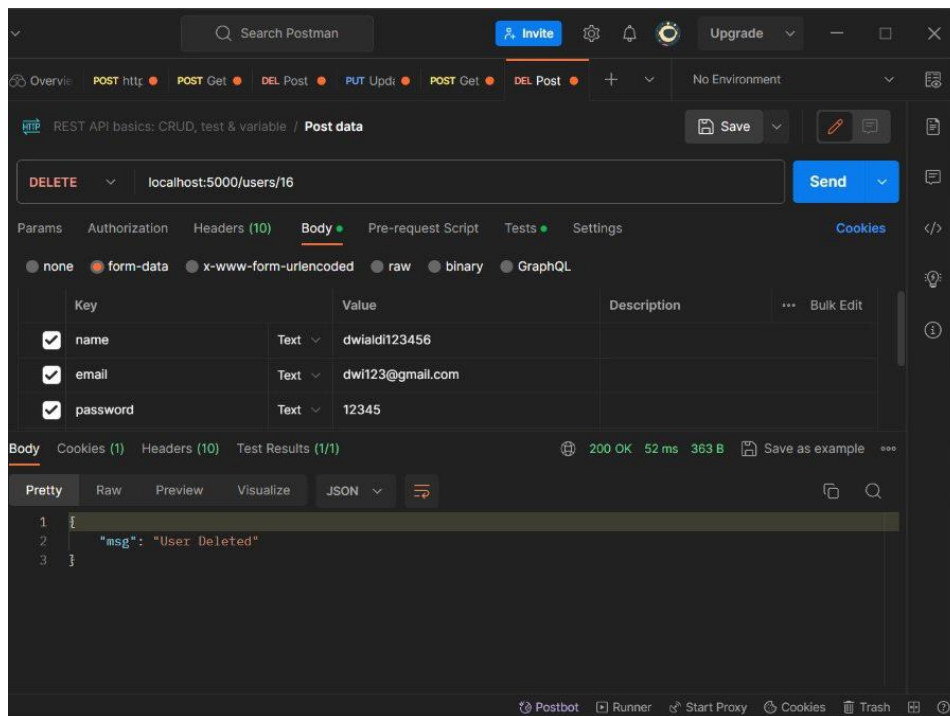
- POST '/users'



- PUT '/users:id'

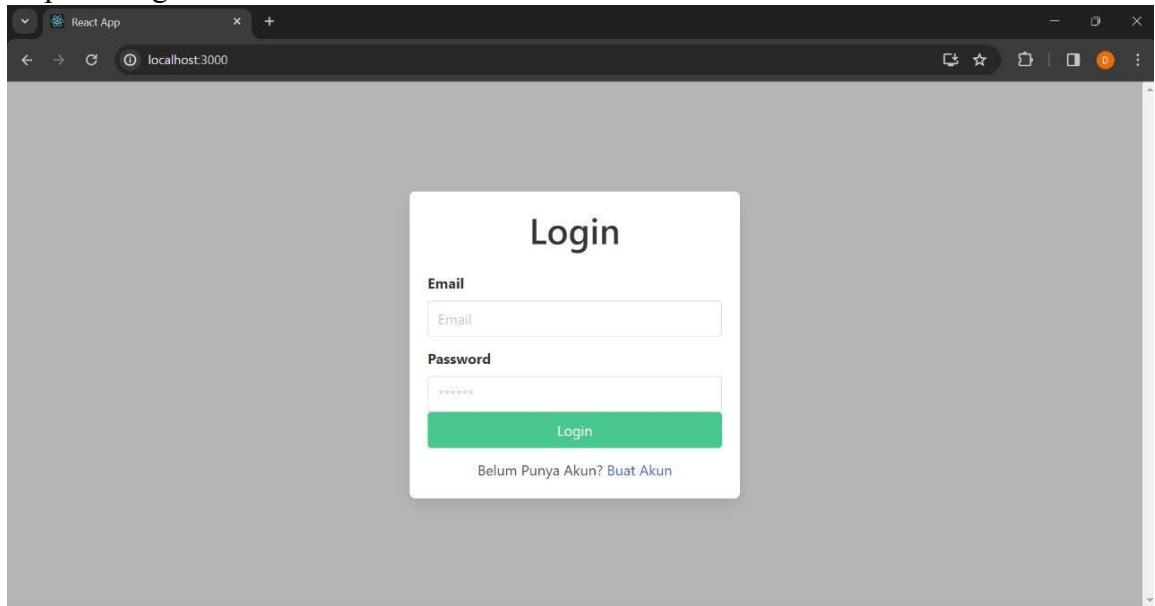


- DELETE '/users:id'



3.2 Test Frontend

- Tampilan Login



A screenshot of a web browser window displaying a login form. The browser's address bar shows 'localhost:3000'. The form is centered on a gray background and has a white title 'Login'. It contains two input fields: 'Email' and 'Password'. Below the password field is a green 'Login' button. At the bottom of the form, there is a link that says 'Belum Punya Akun? Buat Akun'.

React App

localhost:3000

Login

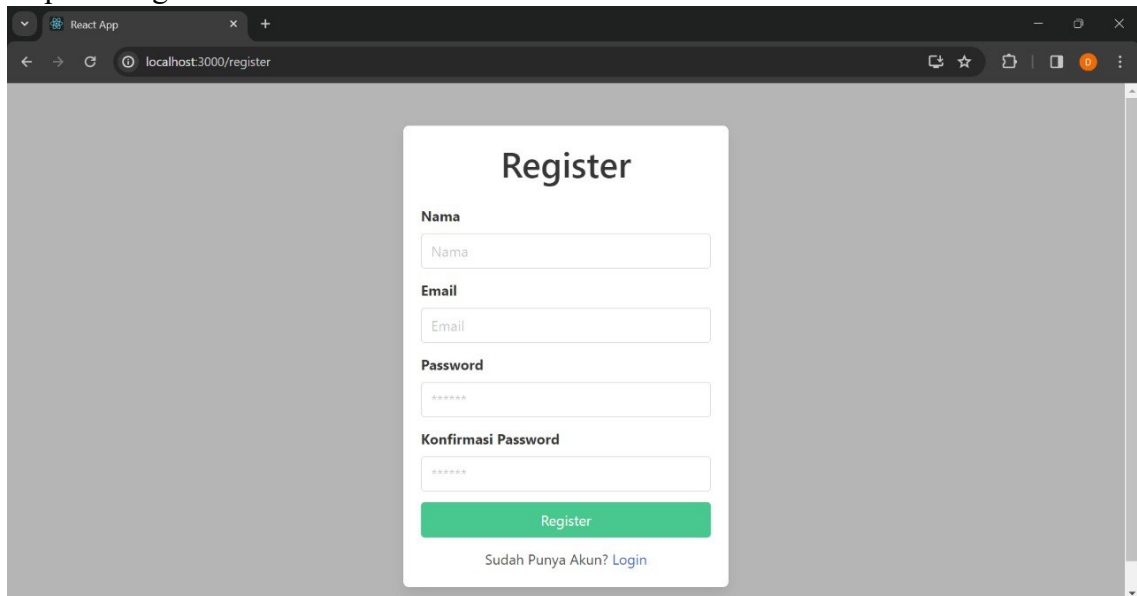
Email

Password

Login

Belum Punya Akun? [Buat Akun](#)

- Tampilan Register



A screenshot of a web browser window displaying a register form. The browser's address bar shows 'localhost:3000/register'. The form is centered on a gray background and has a white title 'Register'. It contains four input fields: 'Nama', 'Email', 'Password', and 'Konfirmasi Password'. Below the last two fields is a green 'Register' button. At the bottom of the form, there is a link that says 'Sudah Punya Akun? Login'.

React App

localhost:3000/register

Register

Nama

Email

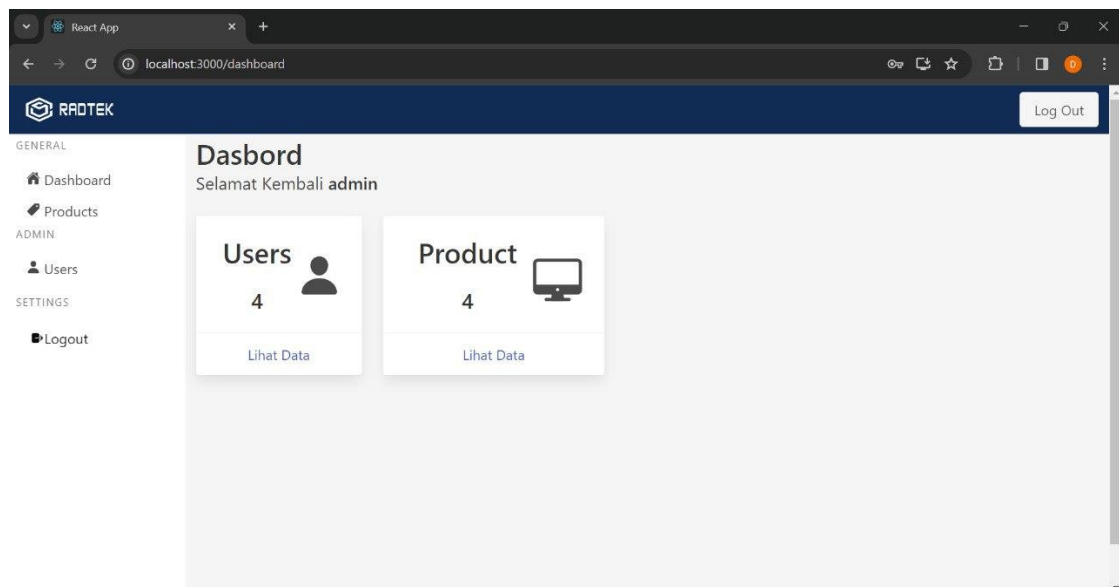
Password

Konfirmasi Password

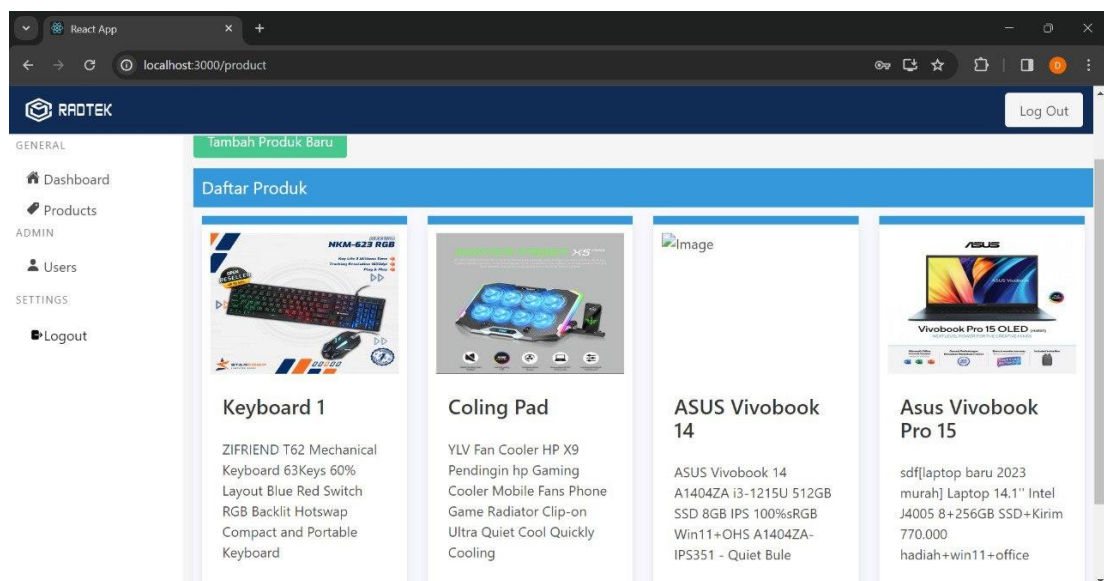
Register

Sudah Punya Akun? [Login](#)

- Tampilan Dashboard



- Tampilan Produk



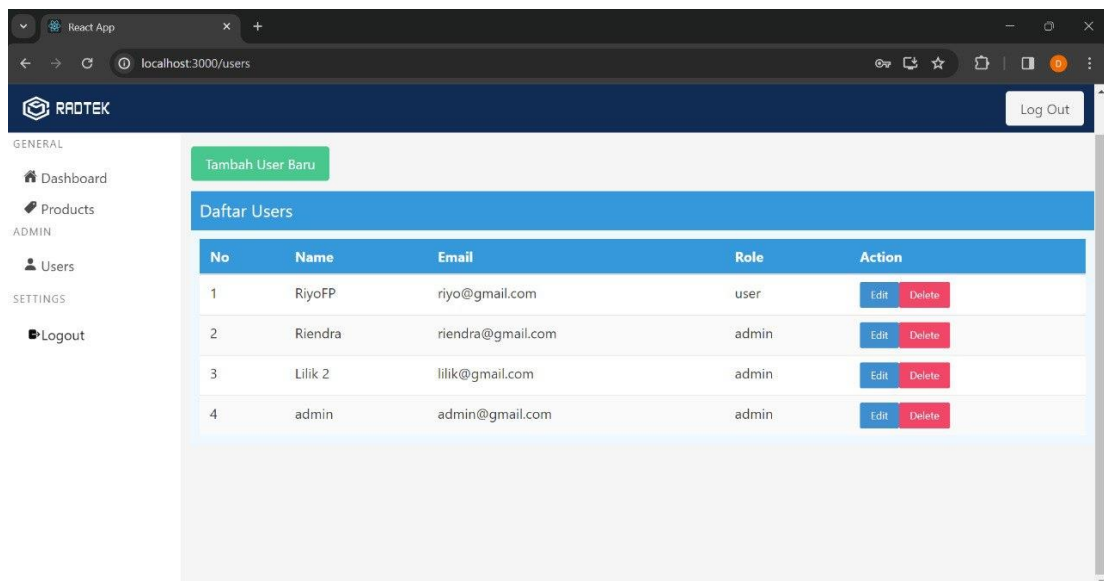
- Tampilan Tambah Produk

The screenshot shows a web browser window with the URL `localhost:3000/product/add`. The application is titled 'RADTEK' and has a 'Log Out' button in the top right corner. The left sidebar contains a menu with the following items: 'GENERAL' (Dashboard, Products), 'ADMIN' (Users), and 'SETTINGS' (Logout). The main content area is titled 'Produk' and 'Tambah Produk Baru'. It contains a form with the following fields: 'Product Name' (placeholder: 'Product Name'), 'Description' (placeholder: 'Description Product'), 'Harga' (placeholder: 'Harga Product'), and 'Image' (placeholder: 'Pilih File...'). A green 'Save' button is located at the bottom of the form.

- Tampilan Edit Produk

The screenshot shows a web browser window with the URL `localhost:3000/product/edit/12`. The application is titled 'RADTEK' and has a 'Log Out' button in the top right corner. The left sidebar contains a menu with the following items: 'GENERAL' (Dashboard, Products), 'ADMIN' (Users), and 'SETTINGS' (Logout). The main content area is titled 'Produk' and 'Edit Produk'. It contains a form with the following fields: 'Product Name' (value: 'Keyboard 1'), 'Description' (value: 'ZIFRIEND T62 Mechanical Keyboard 63Keys 60% Layout Blue Red Switch RGB Backlit Hotswap Compact and Portable Keyboard'), 'Harga' (value: '175000'), and 'Image' (placeholder: 'Pilih File...'). A small image of a keyboard is visible below the 'Image' field.

- Tampilan User



- Tampilan Edit User

