

KELOMPOK 3
PROJECT AKHIR PRAKTIKUM WEB SERVICE



Disusun Oleh :
5210311106_ Mohamad Ilham Septianto
5210311109_ Rizky Kurniawan
5210311129_ Christo Wendly Haron De Fretes
5210311135_ Naufal Ammar Dzakiy

PROGRAM STUDI SISTEM INFORMASI PROGRAM SARJANA
FAKULTAS SAINS & TEKNOLOGI
UNIVERSITAS TEKNOLOGI YOGYAKARTA
2024

Daftar Isi

BAB 1	1
1.1 Persiapan	1
1.2 Deskripsi	1
1.3 Database	1
1.4 Endpoint	2
1.5 Framework dan Packages	3
BAB 2	4
2.1 BACKEND	4
Membuat Folder	4
Membuat File Package.json	4
Menginstall Dependensi	4
Mengkonfigurasi File index.js	4
Mengkonfigurasi File Databases.js	4
Membuat Product Models	4
Membuat Routes	4
2.2 FRONTEND	4
Menginstall Dependensi	4
Menkonfigurasi Indeks.js	4
Membuat File ProductList	5
Membuat File AddProduct	7
Membuat File EditProduct	9
Membuat File SignIn	11
Membuat File SignUp	13
Menjalankan Project	15
BAB 3	16
3.1. Pengujian Backend	16
Method Get	16
Method Get byId	16
Method Post	17
Method Patch	17
Method Delete	18
3.2. Pengujian Frontend	19

BAB 1

1.1 Persiapan

Persiapan untuk membuat proyek Restful API melibatkan beberapa langkah dan penggunaan perangkat lunak tertentu. Pertama-tama, kita perlu merencanakan desain sistem dan menentukan kebutuhan fungsional dan non-fungsional proyek. Ini melibatkan identifikasi sumber daya yang akan diakses dan pengaturan *endpoint*.

Setelah perencanaan, langkah selanjutnya adalah memilih teknologi dan alat yang sesuai untuk pengembangan. Untuk membuat Restful API, dapat menggunakan bahasa pemrograman seperti *JavaScript* (Node.js). *Framework* seperti *Express.js* dapat digunakan untuk mempercepat pengembangan. Penggunaan *database* juga penting, dan pilihan *database* tergantung pada kebutuhan aplikasi seperti MySQL. Selain itu, *software* yang dapat digunakan dalam pembuatan project ini antara lain ialah Visual Studio Code serta Postman.

1.1 Deskripsi

Project Restful API dengan tema penjualan *hardware* merupakan solusi yang efektif untuk mengelola dan menyediakan data terkait penjualan produk perangkat keras secara efisien. Dalam konteks ini, API akan dirancang untuk mendukung operasi yang melibatkan informasi produk. API ini akan memungkinkan pengguna untuk mengakses data produk, termasuk detail spesifikasi.

Adapun alasan memilih tema penjualan *hardware* adalah karena industri penjualan perangkat keras merupakan bagian yang krusial dari dunia teknologi dan bisnis. Produk perangkat keras meliputi berbagai jenis seperti komputer, laptop, server, komponen elektronik, dan perangkat jaringan.

1.2 Database

Dalam proyek Restful API, *database* memainkan peran krusial sebagai penyimpan data yang diperlukan untuk aplikasi tersebut. *Database* digunakan untuk menyimpan informasi seperti produk atau entitas lainnya yang dikelola oleh API. Dalam proyek yang dibuat menggunakan SQL. SQL atau *Structured Query Language* yang merupakan suatu bahasa yang digunakan untuk mengelola data di dalam database relasional. SQL pada awalnya tidak digunakan untuk menyebut jenis database, namun seiring berjalannya waktu SQL menjadi jenis *database* dikarenakan munculnya jenis *database* baru yang memiliki *query* yang berbeda.

Database yang akan digunakan dalam proyek ini memiliki dua tabel yaitu tabel product dan users. Berikut ini struktur rancangan tabel untuk *database* yang akan digunakan dalam proyek ini;

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/> 1	id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
<input type="checkbox"/> 2	name	varchar(255)	utf8mb4_general_ci		Yes	NULL			Change Drop More
<input type="checkbox"/> 3	image	varchar(255)	utf8mb4_general_ci		Yes	NULL			Change Drop More
<input type="checkbox"/> 4	url	varchar(255)	utf8mb4_general_ci		Yes	NULL			Change Drop More
<input type="checkbox"/> 5	createdAt	datetime			No	None			Change Drop More
<input type="checkbox"/> 6	updatedAt	datetime			No	None			Change Drop More

Gambar 1. Struktur Tabel Product

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/> 1	id	int(11)			No	None		AUTO_INCREMENT	Change Drop More
<input type="checkbox"/> 2	fullName	varchar(128)	utf8mb4_general_ci		No	None			Change Drop More
<input type="checkbox"/> 3	password	varchar(128)	utf8mb4_general_ci		No	None			Change Drop More
<input type="checkbox"/> 4	email	varchar(128)	utf8mb4_general_ci		No	None			Change Drop More
<input type="checkbox"/> 5	createdAt	datetime			No	None			Change Drop More
<input type="checkbox"/> 6	updatedAt	datetime			No	None			Change Drop More
<input type="checkbox"/> 7	refresh_token	varchar(255)	utf8mb4_general_ci		Yes	NULL			Change Drop More

Gambar 2. Struktur Tabel Users

1.3 Endpoint

Endpoint dalam RESTful API adalah suatu titik akhir (endpoint) atau URL tertentu yang digunakan untuk berinteraksi dengan layanan atau sumber daya pada sistem. Dalam arsitektur REST, setiap *endpoint* merepresentasikan suatu operasi atau akses terhadap sumber daya tertentu. Setiap endpoint memiliki metode HTTP yang terkait, seperti GET untuk mendapatkan informasi, POST untuk membuat sumber daya baru, PUT atau PATCH untuk memperbarui sumber daya, dan DELETE untuk menghapusnya.

Untuk endpoint yang akan digunakan dalam project ini antara lain;

```

1 router.post("/auth/sign-up", isLogout, signUp);
2 router.post("/auth/sign-in", isLogout, signIn);
3 router.post("/auth/sign-out", isLogout, signOut);
4
5 router.get("/products", isLogin, getProducts);
6 router.get("/products/:id", isLogin, getProductById);
7 router.post("/products", isLogin, saveProduct);
8 router.patch("/products/:id", isLogin, updateProduct);
9 router.delete("/products/:id", isLogin, deleteProduct);

```

Berdasarkan endpoint diatas:

- POST(/auth/sign-up) digunakan untuk menambahkan akun user
- POST(/auth/sign-in) digunakan untuk melakukan login dan masuk kesistem
- POST(/auth/sign-out) digunakan untuk keluar dari sistem
- GET(/products) digunakan untuk menampilkan seluruh data yang ada ditabel database

- GET(/products/:id) digunakan untuk menampilkan data berdasarkan id
- POST(/products) digunakan untuk menambahkan data ke dalam database
- PATCH(/products/:id) digunakan untuk mengedit data berdasarkan id
- DELETE(/products/:id) digunakan untuk menghapus data berdasarkan id

1.4 Framework dan Packages

Framework adalah kerangka kerja perangkat lunak yang menyediakan struktur dan alat bantu untuk membangun aplikasi. Dalam pengembangan RESTful API, framework menyediakan struktur dasar, aturan konvensi, dan sering kali fungsi utilitas yang membantu pengembang dalam membuat, mengelola, dan menyediakan API dengan lebih efisien.

Packages adalah kumpulan modul atau pustaka perangkat lunak yang menyediakan fungsionalitas tambahan untuk aplikasi. Dalam konteks RESTful API, packages sering digunakan untuk menambahkan fitur-fitur tertentu, seperti autentikasi, otentikasi, manajemen izin, dokumentasi API, dan lainnya.

Dalam project yang dibuat *framework* yang digunakan ialah .sedangkan untuk *packages* yang diguakan dalam project ini antara lain cors, express, express-fileupload, morgan, mysql2, nodemon, dan sequelize.

BAB 2

2.1 BACKEND

2.2 FRONTEND

Pada bagian frontend ini menggunakan react.js. Untuk membuat project ini dapat mengetik perintah `npx create-react-app frontend` pada terminal. Maka folder frontend akan muncul.

Menginstall Dependensi

Beberapa dependensi yang diinstal antara lain react , axios, dan bulma. Untuk menginstal semua dependensi tersebut sekaligus caranya ialah dengan mengetik `npm i react-router-dom axios bulma` pada terminal lalu tekan enter. Maka semua dependensi yang diketik akan terinstal. Untuk mengecek apakah dependensi terinstal bisa melihat file package.json seperti gambar dibawah ini

```
1  {
2    "name": "frontend",
3    "version": "0.1.0",
4    "private": true,
5    "dependencies": {
6      "@testing-library/jest-dom": "^5.17.0",
7      "@testing-library/react": "^13.4.0",
8      "@testing-library/user-event": "^13.5.0",
9      "axios": "^1.6.3",
10     "bulma": "^0.9.4",
11     "jwt-decode": "^4.0.0",
12     "react": "^18.2.0",
13     "react-dom": "^18.2.0",
14     "react-router-dom": "^6.21.1",
15     "react-scripts": "5.0.1",
16     "web-vitals": "^2.1.4"
17   },
```

Menkonfigurasi Indeks.js

File indeks.js ini terdapat pada folder src.. Salah satu yang diubah di file ini ialah menambahkan import untuk bulma. Berikut ini kode lengkap dari file indeks.js

```
1  import React from "react";
2  import ReactDOM from "react-dom";
3  import App from "./App";
4  import "bulma/css/bulma.css";
5
6  const root = ReactDOM.createRoot(document.getElementById("root"));
7  root.render(
8    <React.StrictMode>
9      <App />
10    </React.StrictMode>
11  );
12
```

Membuat File ProductList

Sebelum membuat file ini buat dulu folder components pada folder src. Folder ini digunakan untuk menyimpan file ProductList. File ProductList ini digunakan untuk menampilkan seluruh data yang ada didatabase. Berikut ini source code nya

```
1 import React, { useState, useEffect } from "react";
2 import axios from "axios";
3 import { Link, useNavigate } from "react-router-dom";
4
5 // ... (import statements)
6
7 const ProductList = () => {
8   const [products, setProducts] = useState([]);
9   const navigate = useNavigate();
10
11   useEffect(() => {
12     getProducts();
13   }, []);
```

```

1  const getProducts = async () => {
2    try {
3      const response = await axios.get("http://localhost:3000/products", {
4        withCredentials: true,
5      });
6      setProducts(response.data);
7    } catch (error) {
8      if (error.response.status === 401) {
9        navigate("/sign-in");
10       return;
11     }
12     console.error("Error fetching products:", error);
13   }
14 };
15
16 const deleteProduct = async (productId) => {
17   try {
18     await axios.delete(`http://localhost:3000/products/${productId}`, {
19       withCredentials: true,
20     });
21     getProducts();
22   } catch (error) {
23     if (error.response.status === 401) {
24       navigate("/sign-in");
25       return;
26     }
27     console.log(error);
28   }
29 };
30
31 const signOut = async () => {
32   try {
33     const res = await axios.post(
34       "http://localhost:3000/auth/sign-out",
35       {},
36       {
37         withCredentials: true,
38       }
39     );
40     navigate("/sign-in");
41   } catch (error) {
42     console.log(error);
43   }
44 };
45
46 return (
47   <div className="container mt-5">
48     <div className="columns">
49       <div className="column">
50         <h1 className="title">Product List</h1>
51       </div>
52       <div className="column is-narrow">
53         <Link to="/add" className="button is-success ml-auto">
54           Add New
55         </Link>
56       </div>
57       <div className="column is-narrow">
58         <button onClick={signOut} className="button is-danger ml-auto">
59           Sign-out
60         </button>
61       </div>
62     </div>
63
64     <div className="columns is-multiline mt-2">
65       {products.map((product) => (
66         <div className="column is-one-quarter" key={product.id}>
67           <div className="card">
68             <div className="card-image">
69               <figure className="image is-4by3">
70                 <img src={product.url} alt="Image" />
71               </figure>
72             </div>
73             <div className="card-content">
74               <div className="media">
75                 <div className="media-content">
76                   <p className="title is-4">{product.name}</p>
77                 </div>
78               </div>
79             </div>
80
81             <footer className="card-footer">
82               <Link

```


Membuat File AddProduct

File ini dibuat dan disimpan seperti file ProductList yaitu disimpan di folder components. File ini digunakan untuk menampilkan halaman yang berisi formulir untuk menambahkan barang. Berikut ini Source Code dari file ini;

```
1  import React, { useState } from "react";
2  import axios from "axios";
3  import { useNavigate } from "react-router-dom";
4
5  const AddProduct = () => {
6    const [title, setTitle] = useState("");
7    const [desc, setDesc] = useState("");
8    const [harga, setHarga] = useState("");
9    const [file, setFile] = useState("");
10   const [preview, setPreview] = useState("");
11   const navigate = useNavigate();
12
13   const loadImage = (e) => {
14     const image = e.target.files[0];
15     setFile(image);
16     setPreview(URL.createObjectURL(image));
17   };
18
19   const saveProduct = async (e) => {
20     e.preventDefault();
21     const formData = new FormData();
22     formData.append("file", file);
23     formData.append("title", title);
24     formData.append("desc", desc);
25     formData.append("harga", harga);
26     try {
27       await axios.post("http://localhost:3000/products", formData, {
28         headers: {
29           "Content-type": "multipart/form-data",
30         },
31         withCredentials: true,
32       });
33       navigate("/");
34     } catch (error) {
35       if (error.response.status === 401) {
36         navigate("/sign-in");
37         return;
38       }
39       console.log(error);
40     }
41   };
42
43   return (
44     <div className="columns is-centered mt-5">
45       <div className="column is-half">
46         <form onSubmit={saveProduct}>
47           <div className="field">
48             <label className="label">Product Name</label>
49             <div className="control">
50               <input
51                 type="text"
52                 className="input"
53                 value={title}
54                 onChange={(e) => setTitle(e.target.value)}
55                 placeholder="Product Name"
56               />
57             </div>
58           </div>
59           <div className="field">
60             <label className="label">Product Description</label>
61             <div className="control">
62               <input
63                 type="text"
64                 className="input"
65                 value={desc}
66                 onChange={(e) => setDesc(e.target.value)}
67                 placeholder="Product Description"
68               />
69             </div>
70           </div>
71         </form>
72       </div>
73     </div>
74   );
75 }
```

```

1  <div className="field">
2      <label className="label">Product Price</label>
3      <div className="control">
4          <input
5              type="text"
6              className="input"
7              value={harga}
8              onChange={(e) => setHarga(e.target.value)}
9              placeholder="Price"
10         />
11     </div>
12 </div>
13
14 <div className="field">
15     <label className="label">Image</label>
16     <div className="control">
17         <div className="file">
18             <label className="file-label">
19                 <input
20                     type="file"
21                     className="file-input"
22                     onChange={loadImage}
23                 />
24                 <span className="file-cta">
25                     <span className="file-label">Choose a file...</span>
26                 </span>
27             </label>
28         </div>
29     </div>
30 </div>
31
32 {preview ? (
33     <figure className="image is-128x128">
34         <img src={preview} alt="Preview Image" />
35     </figure>
36 ) : (
37     ""
38 )}
39
40 <div className="field">
41     <div className="control">
42         <button type="submit" className="button is-success">
43             Save
44         </button>
45     </div>
46 </div>
47 </form>
48 </div>
49 </div>
50 );
51 };
52
53 export default AddProduct;
54

```

Membuat File EditProduct

File ini dibuat dan disimpan seperti file ProductList dan AddProduct yaitu disimpan di folder components. File ini digunakan untuk mengedit barang misalnya nama, deskripsi, harga, maupun gambar dari barang yang diedit. Berikut ini Source Code dari file ini;

```
1 import React, { useState, useEffect } from "react";
2 import axios from "axios";
3 import { useParams, useNavigate } from "react-router-dom";
4
5 const EditProduct = () => {
6   const [title, setTitle] = useState("");
7   const [desc, setDesc] = useState("");
8   const [harga, setHarga] = useState("");
9   const [file, setFile] = useState("");
10  const [preview, setPreview] = useState("");
11  const { id } = useParams();
12  const navigate = useNavigate();
13
14  useEffect(() => {
15    getProductById();
16  }, []);
17
18  const getProductById = async () => {
19    try {
20      const response = await axios.get(`http://localhost:3000/products/${id}`, {
21        withCredentials: true,
22      });
23      setTitle(response.data.name);
24      setDesc(response.data.description);
25      setHarga(response.data.harga);
26      setFile(response.data.image);
27      setPreview(response.data.url);
28    } catch (error) {
29      if (error.response.status === 401) {
30        navigate("/sign-in");
31        return;
32      }
33    }
34  };
35
36  const loadImage = (e) => {
37    const image = e.target.files[0];
38    setFile(image);
39    setPreview(URL.createObjectURL(image));
40  };
41
42  const updateProduct = async (e) => {
43    e.preventDefault();
44    const formData = new FormData();
45    formData.append("file", file);
46    formData.append("title", title);
47    formData.append("desc", desc);
48    formData.append("harga", harga);
```

```

1  try {
2    await axios.patch(`http://localhost:3000/products/${id}`, formData, {
3      headers: {
4        "Content-type": "multipart/form-data",
5      },
6      withCredentials: true,
7    });
8    navigate("/");
9  } catch (error) {
10   if (error.response.status === 401) {
11     navigate("/sign-in");
12     return;
13   }
14   console.log(error);
15 }
16 };
17
18 return (
19   <div className="columns is-centered mt-5">
20     <div className="column is-half">
21       <form onSubmit={updateProduct}>
22         <div className="field">
23           <label className="label">Product Name</label>
24           <div className="control">
25             <input
26               type="text"
27               className="input"
28               value={title}
29               onChange={(e) => setTitle(e.target.value)}
30               placeholder="Product Name"
31             />
32           </div>
33         </div>
34
35         <div className="field">
36           <label className="label">Product Description</label>
37           <div className="control">
38             <input
39               type="text"
40               className="input"
41               value={desc}
42               onChange={(e) => setDesc(e.target.value)}
43               placeholder="Product Description"
44             />
45           </div>
46         </div>
47
48         <div className="field">
49           <label className="label">Product price</label>
50           <div className="control">
51             <input
52               type="text"
53               className="input"
54               value={harga}
55               onChange={(e) => setHarga(e.target.value)}
56               placeholder="Product Price"
57             />
58           </div>
59         </div>

```

```

1  <div className="field">
2      <label className="label">Image</label>
3      <div className="control">
4          <div className="file">
5              <label className="file-label">
6                  <input
7                      type="file"
8                      className="file-input"
9                      onChange={loadImage}
10                 />
11                 <span className="file-cta">
12                     <span className="file-label">Choose a file...</span>
13                 </span>
14             </label>
15         </div>
16     </div>
17 </div>
18
19 {preview ? (
20     <figure className="image is-128x128">
21         <img src={preview} alt="Preview Image" />
22     </figure>
23 ) : (
24     ""
25 )}
26
27 <div className="field">
28     <div className="control">
29         <button type="submit" className="button is-success">
30             Update
31         </button>
32     </div>
33 </div>
34 </form>
35 </div>
36 </div>
37 );
38 };
39
40 export default EditProduct;
41

```

Membuat File SignIn

File ini dibuat dan disimpan di folder components. File ini digunakan untuk menampilkan tampilan login. Berikut ini Source Code dari file ini;

```

1  import React, { useState } from "react";
2  import axios from "axios";
3  import { Link, useNavigate } from "react-router-dom";
4
5  export default function SignIn() {
6      const [formData, setFormData] = useState({
7          email: "",
8          password: "",
9      });
10
11      const [error, setError] = useState("");
12      const navigate = useNavigate();
13
14      const handleChange = (e) => {
15          const { name, value } = e.target;
16          setFormData({ ...formData, [name]: value });
17      };
18
19

```

```

1  const handleSubmit = async (e) => {
2    e.preventDefault();
3
4    if (!formData.email || !formData.password) {
5      setError("Please fill in all required fields");
6      return;
7    }
8
9    try {
10     const response = await axios.post(
11       "http://localhost:3000/auth/sign-in",
12       JSON.stringify(formData),
13       {
14         headers: {
15           "Content-Type": "application/json",
16         },
17         withCredentials: true,
18       }
19     );
20     setError("");
21     navigate("/");
22   } catch (error) {
23     if (error.response.status === 403) {
24       navigate("/");
25       return;
26     }
27     console.error("Error submitting form:", error);
28     setError(
29       error.response.data.message ||
30       "An error occurred while processing your request"
31     );
32   }
33 };
34
35 return (
36   <div className="hero is-fullheight">
37     <div className="hero-body is-flex-direction-column is-justify-content-center is-align-items-center">
38       <form onSubmit={handleSubmit}>
39         <div className="columns is-flex is-flex-direction-column box">
40           <p className="title">Sign in</p>
41           <hr className="my-2" />
42           {error && (
43             <p className="has-text-white has-background-danger py-0 px-2">
44               {error}
45             </p>
46           )}
47           <div className="column">
48             <label htmlFor="email">Email</label>
49             <input
50               id="email"
51               className="input is-primary"
52               type="text"
53               placeholder="Email address"
54               name="email"
55               value={formData.email}
56               onChange={handleChange}
57             />
58           </div>
59           <div className="column">
60             <label htmlFor="password">Password</label>
61             <input
62               id="password"
63               className="input is-primary"
64               type="password"
65               placeholder="Password"
66               name="password"
67               value={formData.password}
68               onChange={handleChange}
69             />
70           </div>
71           <div className="column">
72             <button className="button is-primary is-fullwidth" type="submit">
73               Sign in
74             </button>
75           </div>
76           <div className="has-text-centered">
77             <p>
78               Don't have an account yet?{" "}
79               <Link to="/sign-up" className="has-text-primary">
80                 Sign up
81               </Link>
82             </p>
83           </div>
84         </div>
85       </form>
86     </div>
87   </div>
88 );
89 }
90

```

Membuat File SignUp

File ini dibuat dan disimpan di folder components. File ini digunakan untuk menampilkan halaman formulir registrasi user. Berikut ini Source Code dari file ini;

```
1  import React, { useState } from "react";
2  import axios from "axios";
3  import { Link } from "react-router-dom";
4
5  export default function SignUp() {
6    const [formData, setFormData] = useState({
7      fullname: "",
8      email: "",
9      password: "",
10   });
11
12   const [message, setMessage] = useState({
13     success: "",
14     error: "",
15   });
16
17   const handleChange = (e) => {
18     const { name, value } = e.target;
19     setFormData({ ...formData, [name]: value });
20   };
21
22   const handleSubmit = async (e) => {
23     e.preventDefault();
24
25     if (!formData.fullname || !formData.email || !formData.password) {
26       setMessage({ success: "", error: "Please fill in all required fields" });
27       return;
28     }
29
30     try {
31       const response = await axios.post(
32         "http://localhost:3000/auth/sign-up",
33         JSON.stringify(formData),
34         {
35           headers: {
36             "Content-Type": "application/json",
37           },
38         }
39       );
40       setMessage({ success: response.data.message, error: "" });
41     } catch (error) {
42       console.error("Error submitting form:", error);
43       setMessage({
44         success: "",
45         error:
46           error.response.data.message ||
47           "An error occurred while processing your request",
48       });
49     }
50   };
51 }
```

```

1  return (
2    <div className="hero is-fullheight">
3      <div className="hero-body is-flex-direction-column is-justify-content-center is-align-items-center">
4        <form onSubmit={handleSubmit}>
5          <div className="columns is-flex is-flex-direction-column box">
6            <p className="title">Sign Up</p>
7            <hr className="my-2" />
8            {message.error && (
9              <p className="has-text-white has-background-danger py-0 px-2">
10                {message.error}
11              </p>
12            )}
13            {message.success && (
14              <p className="has-text-white has-background-success py-0 px-2">
15                {message.success}, please{" "}
16                <u>
17                  <Link to="/sign-in" className="has-text-white">
18                    log in
19                  </Link>
20                </u>
21              </p>
22            )}
23            <div className="column">
24              <label htmlFor="name">Full Name</label>
25              <input
26                id="name"
27                className="input is-primary"
28                type="text"
29                placeholder="Enter Name"
30                name="fullname"
31                value={formData.fullname}
32                onChange={handleChange}
33              />
34            </div>
35            <div className="column">
36              <label htmlFor="email">Email</label>
37              <input
38                id="email"
39                className="input is-primary"
40                type="text"
41                placeholder="Email address"
42                name="email"
43                value={formData.email}
44                onChange={handleChange}
45              />
46            </div>
47            <div className="column">
48              <label htmlFor="password">Password</label>
49              <input
50                id="password"
51                className="input is-primary"
52                type="password"
53                placeholder="Password"
54                name="password"
55                value={formData.password}
56                onChange={handleChange}
57              />
58            </div>
59            <div className="column">
60              <button className="button is-primary is-fullwidth" type="submit">
61                Create an account
62              </button>
63            </div>
64            <div className="has-text-centered">
65              <p>
66                Already have an account?{" "}
67                <Link to="/sign-in" className="has-text-primary">
68                  Sign in
69                </Link>
70              </p>
71            </div>
72          </div>
73        </form>
74      </div>
75    </div>
76  );
77 }
78

```


Menjalankan Project

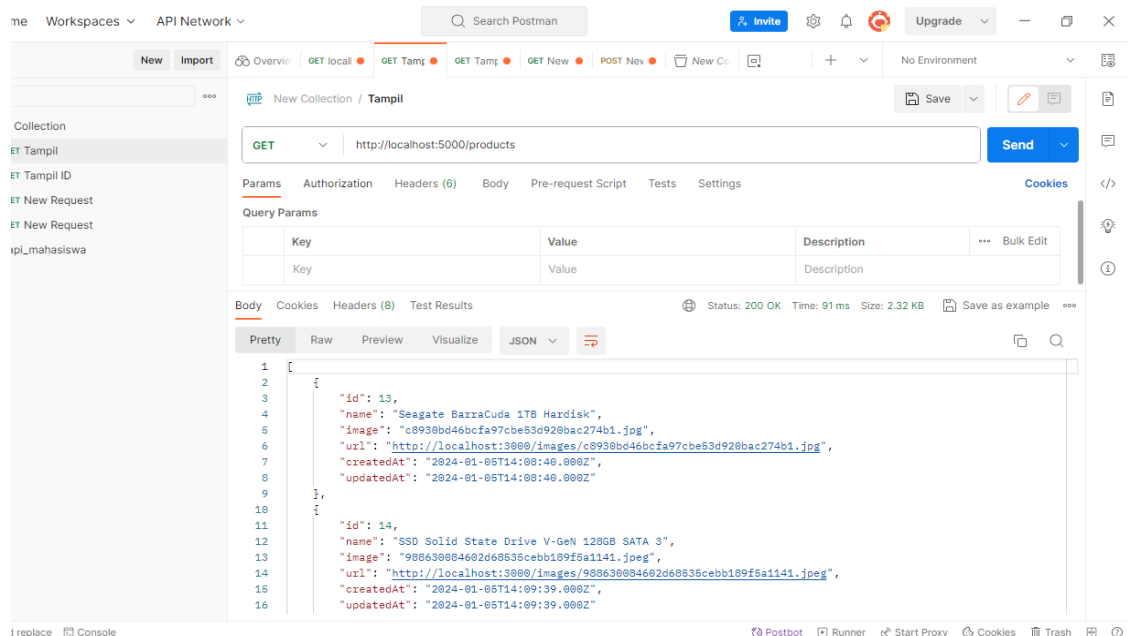
Setelah langkah-langkah diatas dilakukan. Hal yang terakhir dilakukan ialah menjalankan project tersebut. Untuk menjalankan project tersebut dilakukan dengan cara menuliskan npm start pada terminal lalu enter. Setelah itu buka browser dan ketikkan <http://localhost:7200/> maka project akan ditampilkan di browser.

BAB 3

3.1. Pengujian Backend

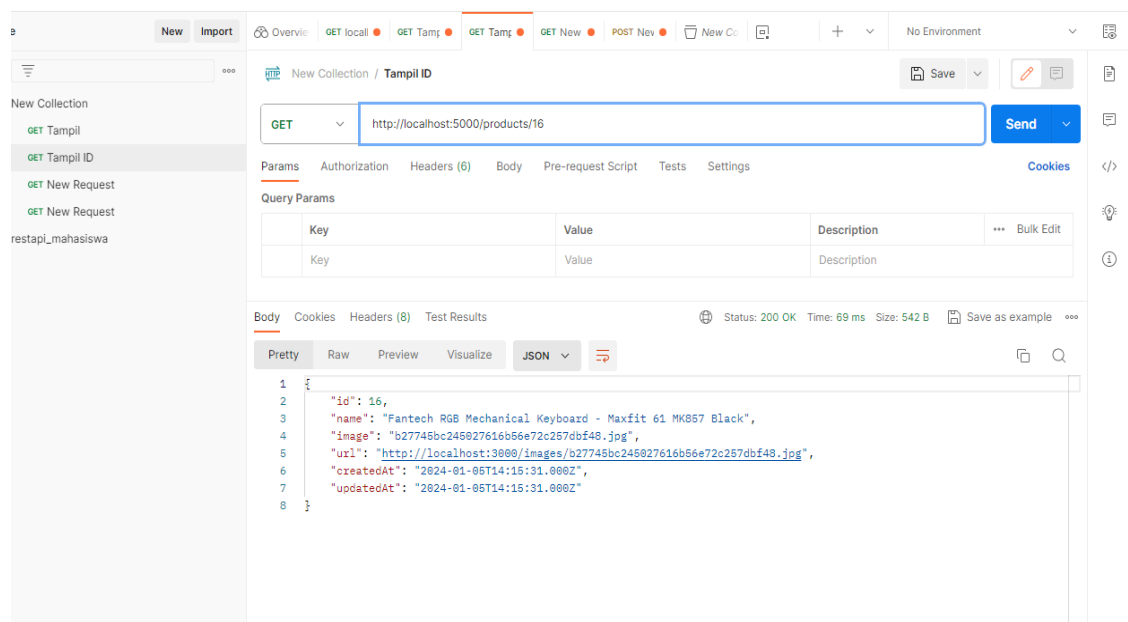
Method Get

Method ini digunakan untuk menampilkan seluruh data yang ada di tabel. Berikut ini pengujian yang dilakukan di Postman untuk menampilkan seluruh data yang ada di tabel;



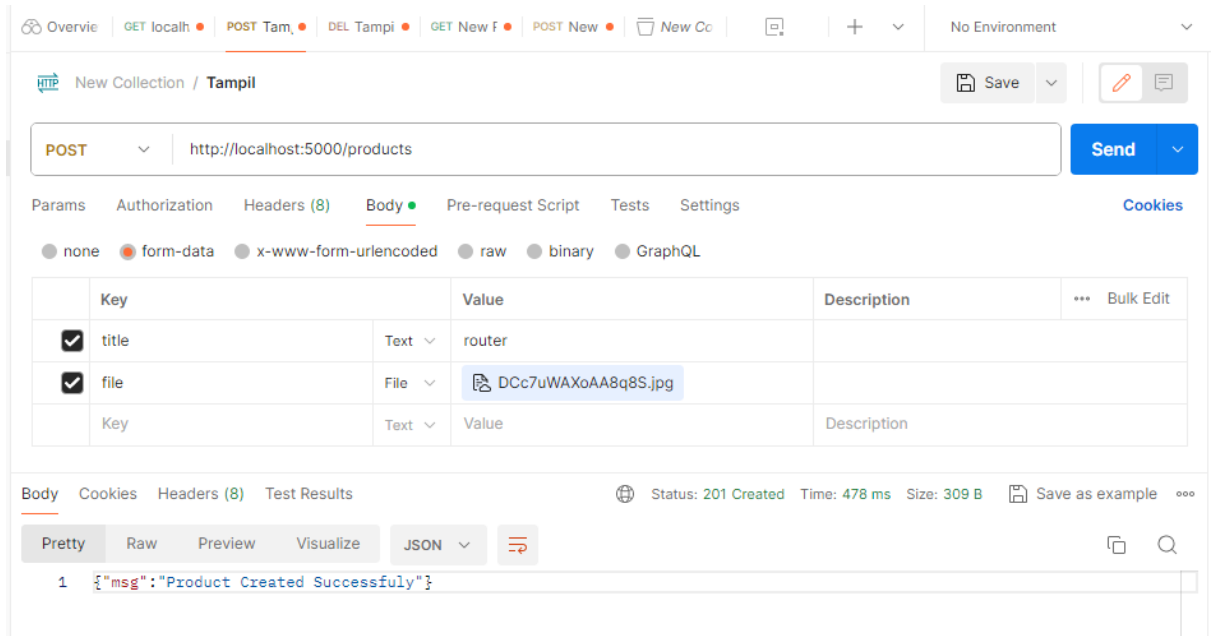
Method Get by Id

Method ini digunakan untuk menampilkan data yang ada di tabel berdasarkan Id. Berikut ini pengujian yang dilakukan di Postman untuk menampilkan data yang ada di tabel berdasarkan id;

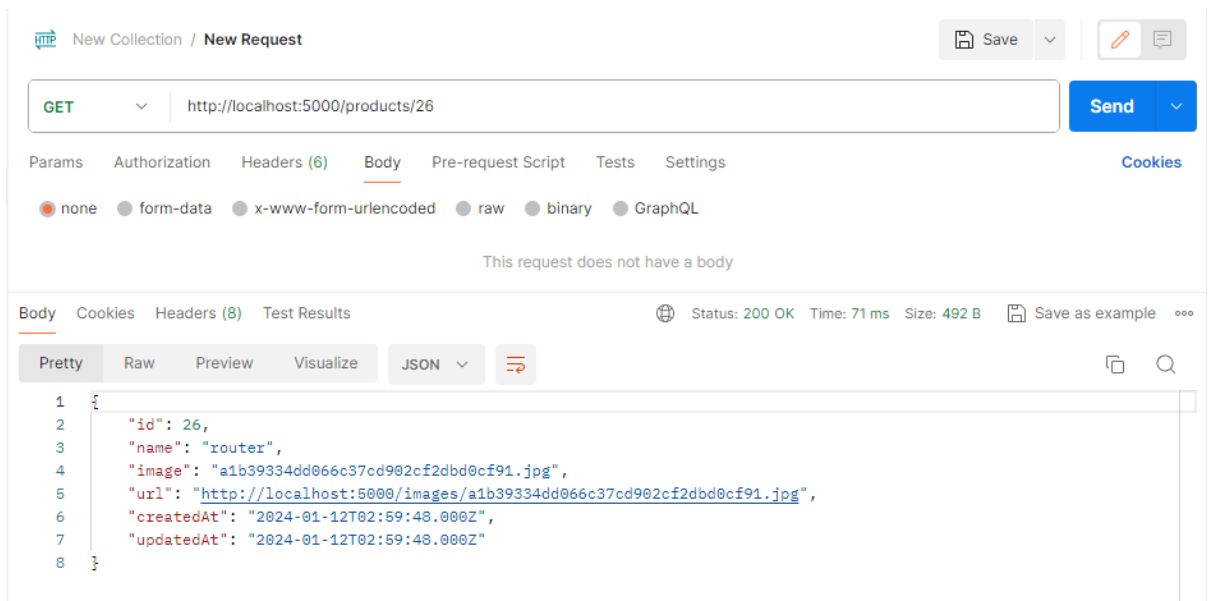


Method Post

Method ini digunakan untuk menambah data. Berikut ini pengujian di Postman untuk method Post



Hasilnya



Method Patch

Method ini digunakan untuk mengedit data yang ada di database. Berikut ini pengujian di Postman untuk method ini;

New Collection / **UPDATE** Save

PATCH http://localhost:5000/products/26 Send

Params Authorization Headers (8) **Body** Pre-request Script Tests Settings Cookies

☐ none ☒ form-data ☐ x-www-form-urlencoded ☐ raw ☐ binary ☐ GraphQL

	Key		Value	Description	...	Bulk Edit
<input checked="" type="checkbox"/>	title	Text	Power Supply			
<input checked="" type="checkbox"/>	file	File	DCc7uWAXoAA8q8S.jpg			
	Key	Text	Value	Description		

Body Cookies Headers (8) Test Results Status: 200 OK Time: 195 ms Size: 304 B Save as example

Pretty Raw Preview Visualize **JSON**

```

1 {
2   "msg": "Product Updated Successfully"
3 }

```

Hasilnya

New Collection / **Tampil ID** Save

GET http://localhost:5000/products/27 Send

Params Authorization Headers (6) **Body** Pre-request Script Tests Settings Cookies

☒ none ☐ form-data ☐ x-www-form-urlencoded ☐ raw ☐ binary ☐ GraphQL

This request does not have a body

Body Cookies Headers (8) Test Results Status: 200 OK Time: 39 ms Size: 498 B Save as example

Pretty Raw Preview Visualize **JSON**

```

1 {
2   "id": 27,
3   "name": "Power Supply",
4   "image": "a1b39334dd066c37cd902cf2dbd0cf91.jpg",
5   "url": "http://localhost:5000/images/a1b39334dd066c37cd902cf2dbd0cf91.jpg",
6   "createdAt": "2024-01-12T03:08:07.000Z",
7   "updatedAt": "2024-01-12T03:08:07.000Z"
8 }

```

Method Delete

Method ini digunakan untuk mengedit data yang ada di database. Berikut ini pengujian di Postman untuk method ini;

New Collection / **Hapus** Save

DELETE http://localhost:5000/products/27 Send

Params Authorization Headers (6) **Body** Pre-request Script Tests Settings Cookies

Query Params

	Key	Value	Description	...	Bulk Edit
	Key	Value	Description		

Body Cookies Headers (8) Test Results Status: 200 OK Time: 102 ms Size: 304 B Save as example

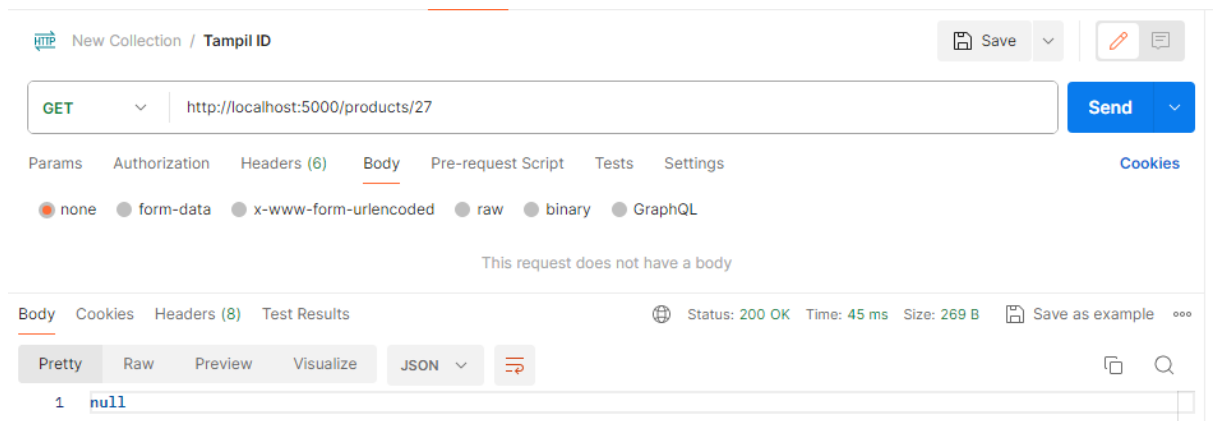
Pretty Raw Preview Visualize **JSON**

```

1 {
2   "msg": "Product Deleted Successfully"
3 }

```

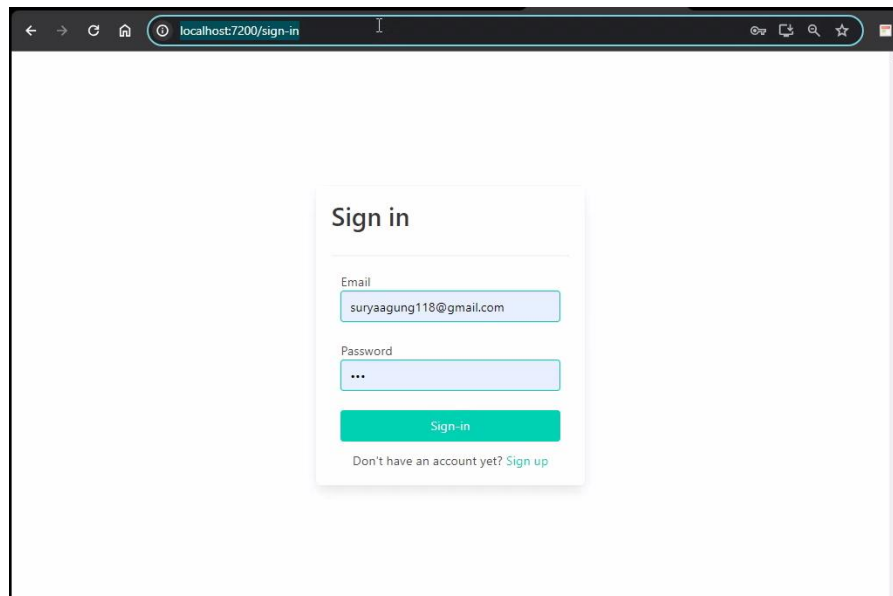
Hasilnya



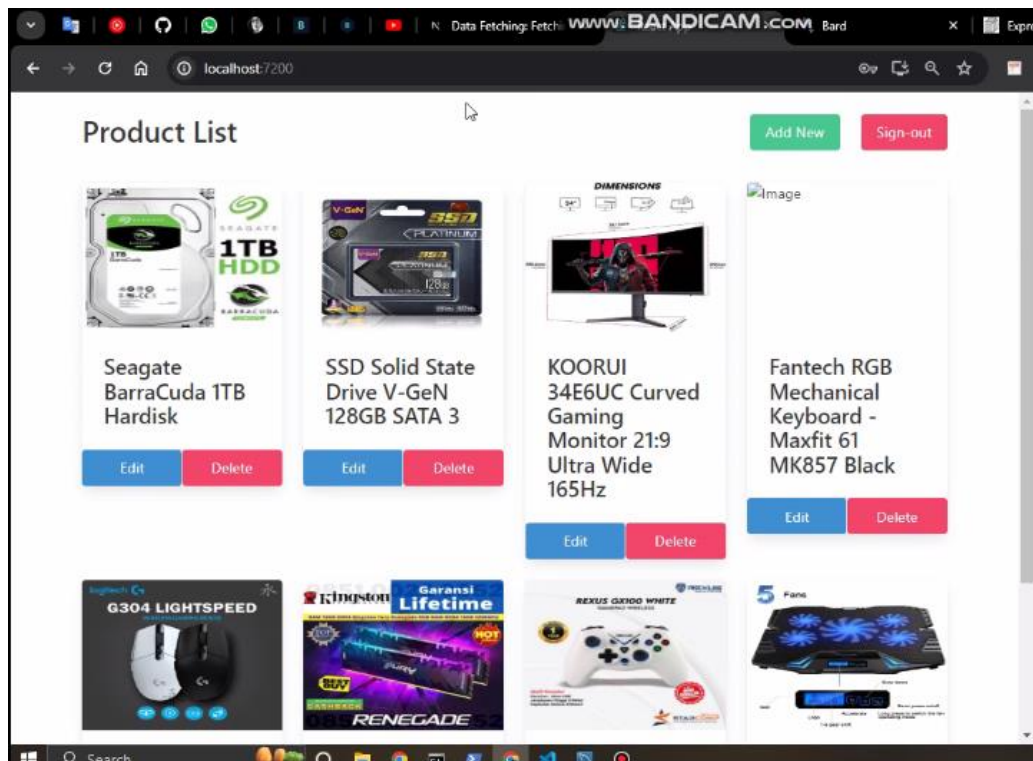
Hasilnya null dikarenakan data telah dihapus sehingga data tidak bisa ditampilkan

3.2.Pengujian Frontend

Menampilkan halaman login



Menampilkan Seluruh Barang



Menambahkan Barang

localhost:7200/add

Product Name

Product Description

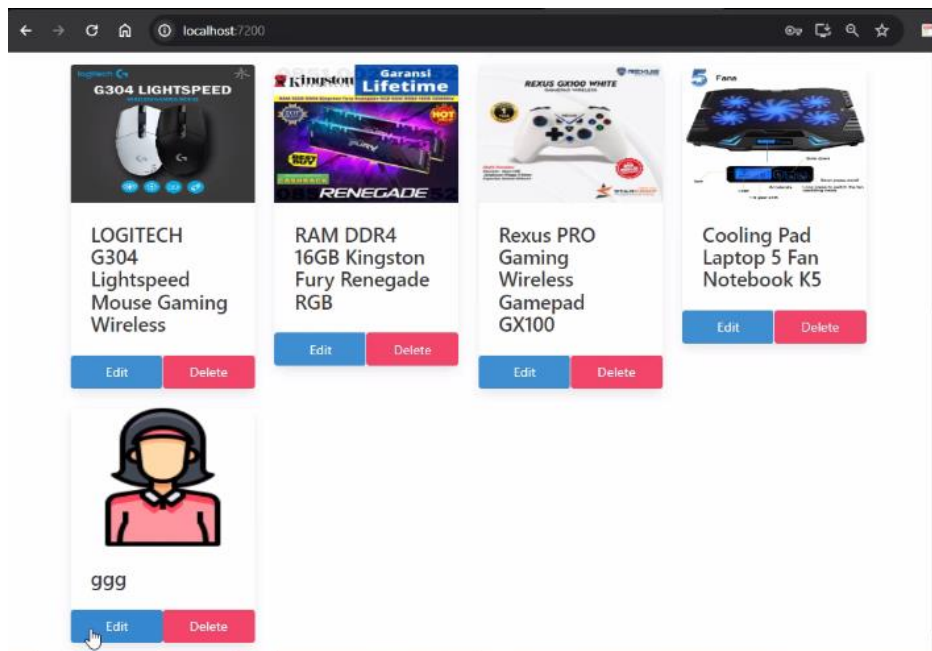
Product Price

Image

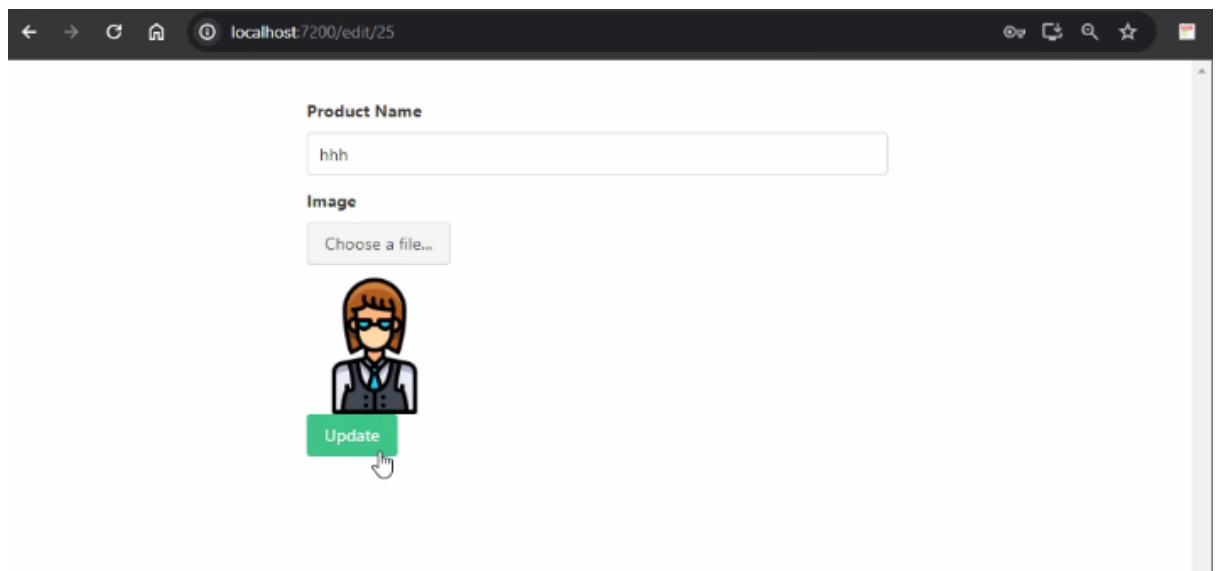
Choose a file...

Save

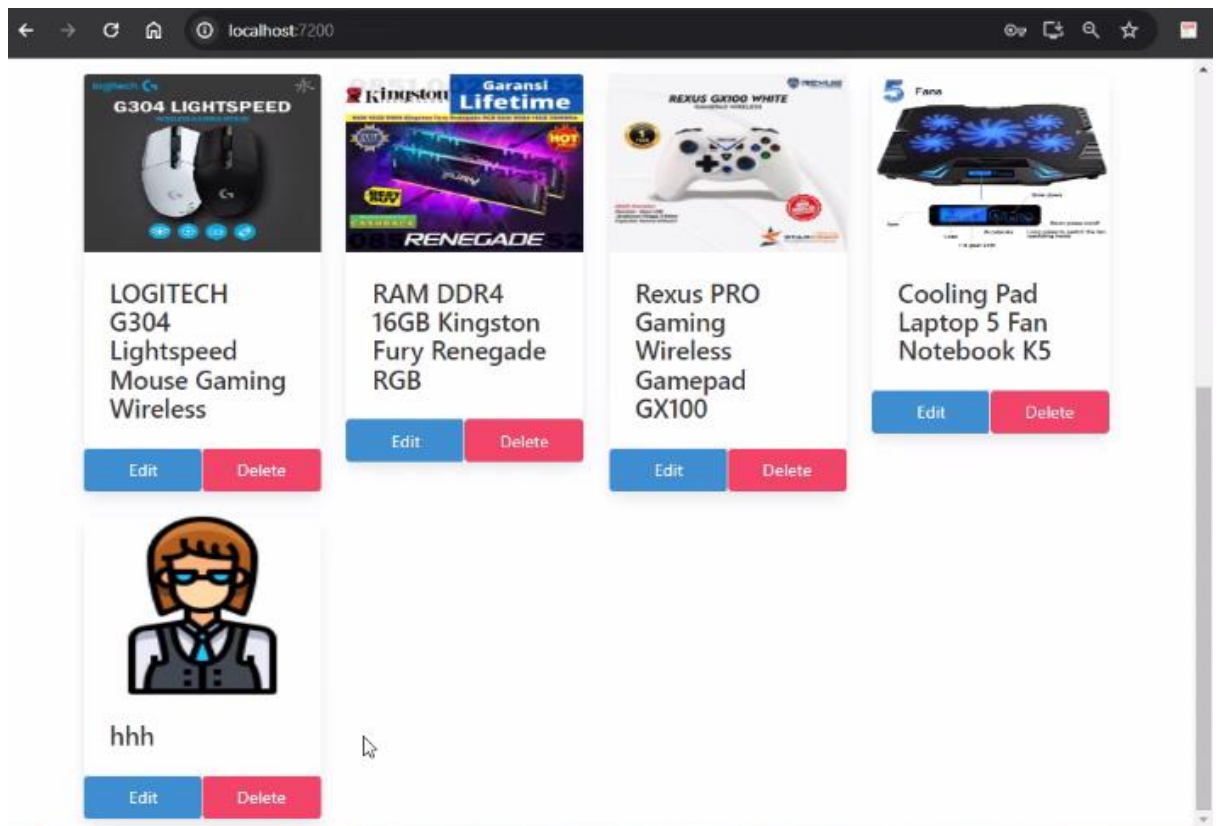
Hasil



Mengedit



Hasil



Menghapus

Saat menekan tombol delete data langsung terhapus

