

NAMA : NURHIDAYAT

NPM :22312099

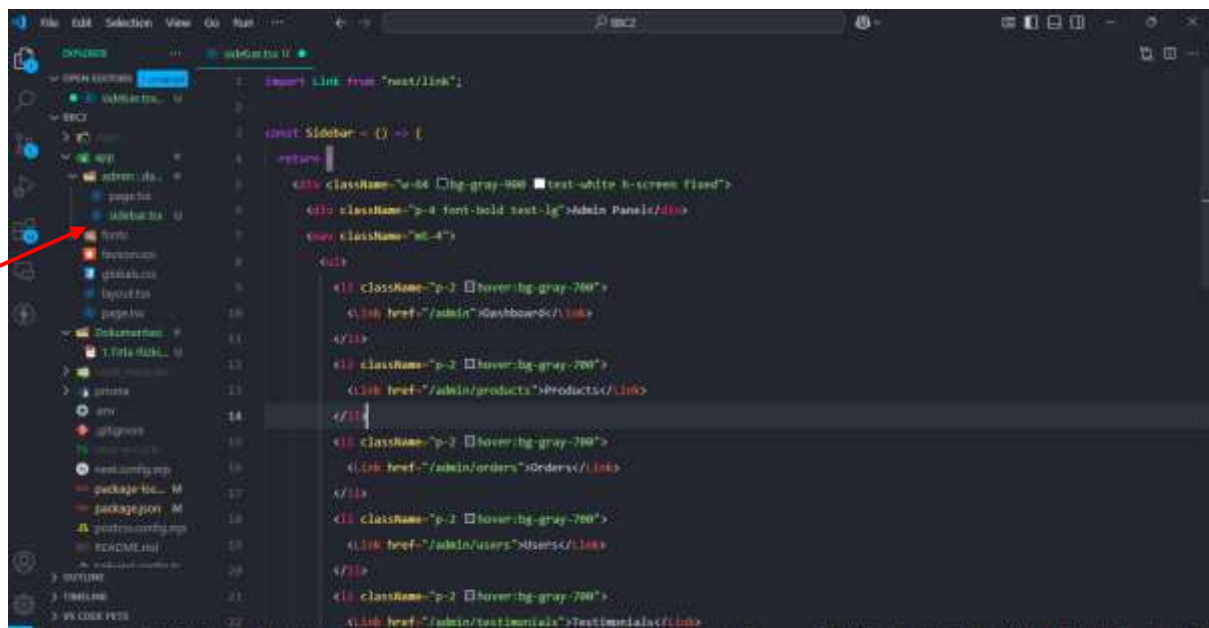
## 1. Memindahkan folder dokumentasi



Deskripsi:

- **Tujuan:** Memperbaiki kesalahan struktur folder agar lebih terorganisir.
- **Langkah Teknis:** Memindahkan folder "dokumentasi" dari dalam folder "app" ke direktori utama.
- **Hasil:** Struktur folder menjadi sejajar antara "app" dan "dokumentasi".

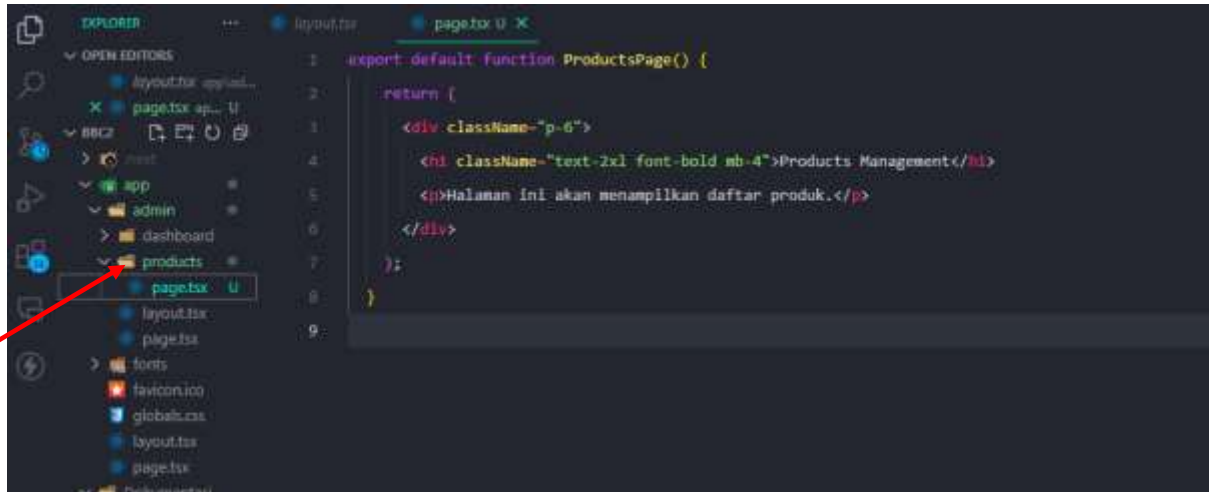
## 2. . Membuat file sidebar.tsx pada folder admin



Deskripsi:

- **Tujuan:** Membuat komponen navigasi utama untuk admin panel agar lebih mudah berpindah antarhalaman.
- **Langkah Teknis:** Menggunakan Link dari Next.js untuk navigasi antarhalaman.
- **Hasil:** Komponen navigasi yang reusable di seluruh halaman admin.

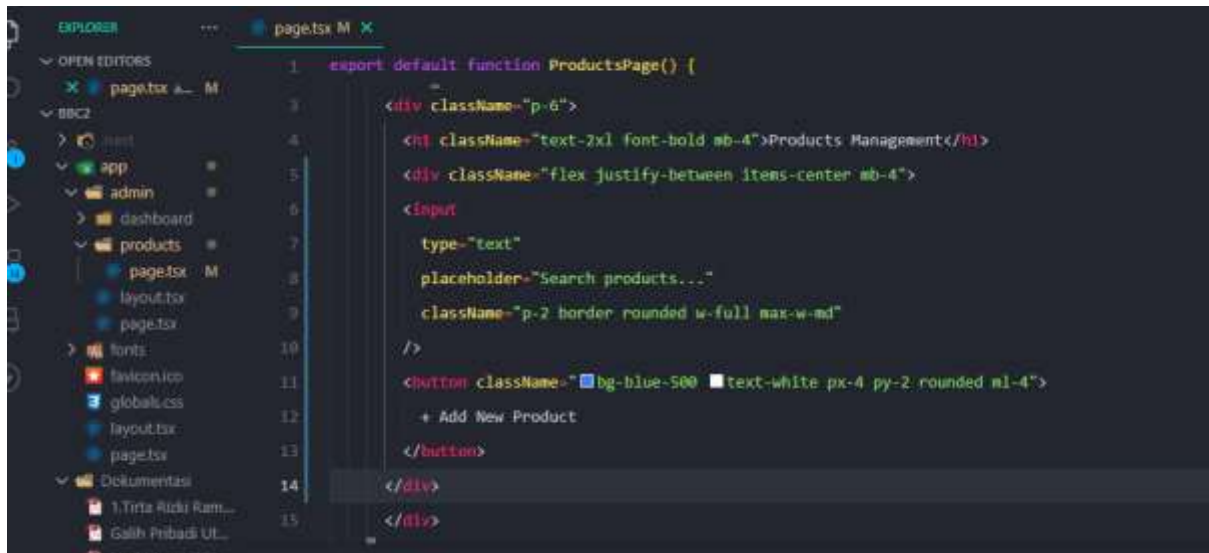
### 3. Membuat Halaman Dasar untuk Products



Deskripsi:

- **Tujuan:** Menyediakan struktur awal untuk halaman Products.
- **Langkah Teknis:** Membuat folder "product" di dalam admin dan menambahkan file "page.tsx" dengan judul dan placeholder teks.
- **Hasil:** Halaman awal untuk Products.

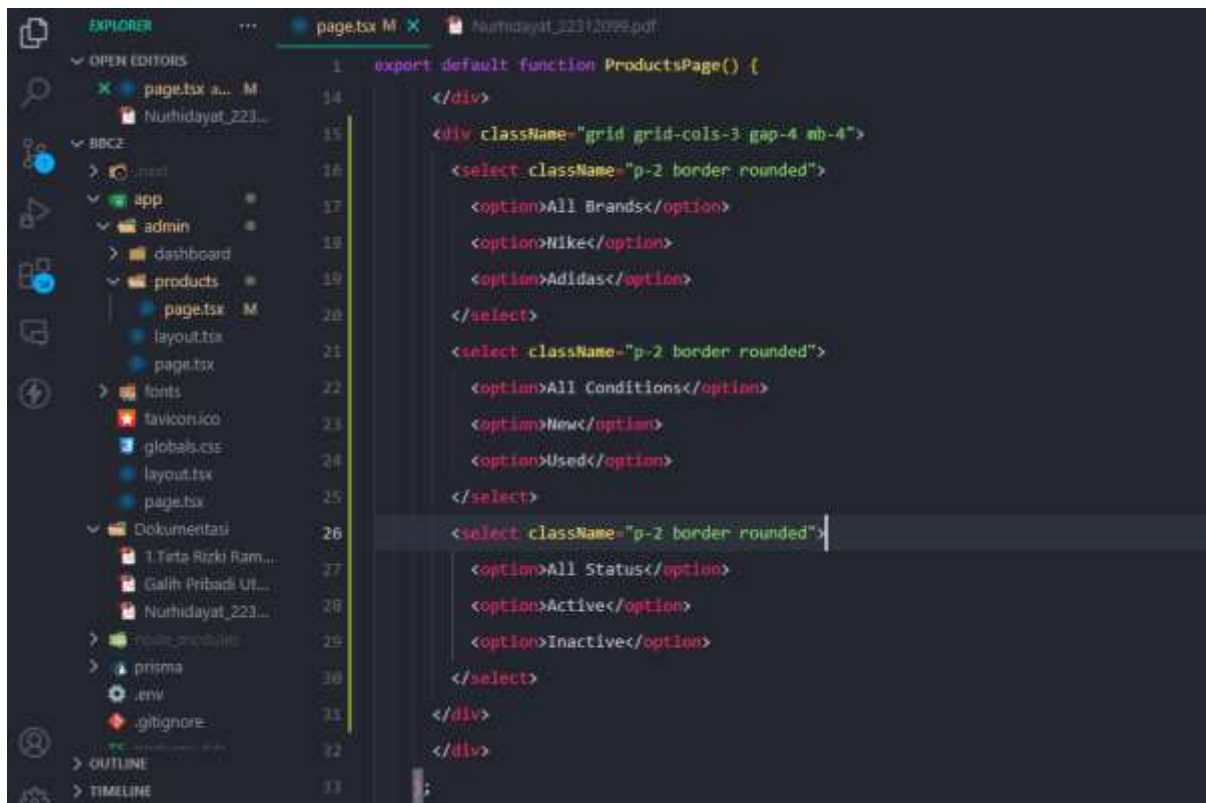
### 4. Menambahkan “Add New Product” dan form pencarian



Keterangan:

- **Tujuan:** Mempermudah pengguna dalam menambahkan produk baru dan mencari produk yang ada.
- **Langkah Teknis:** Menambahkan input pencarian dan tombol "Add New Product".
- **Hasil:** Fitur pencarian dan tombol untuk menambah produk.

## 5. Menambahkan Filter Dropdown



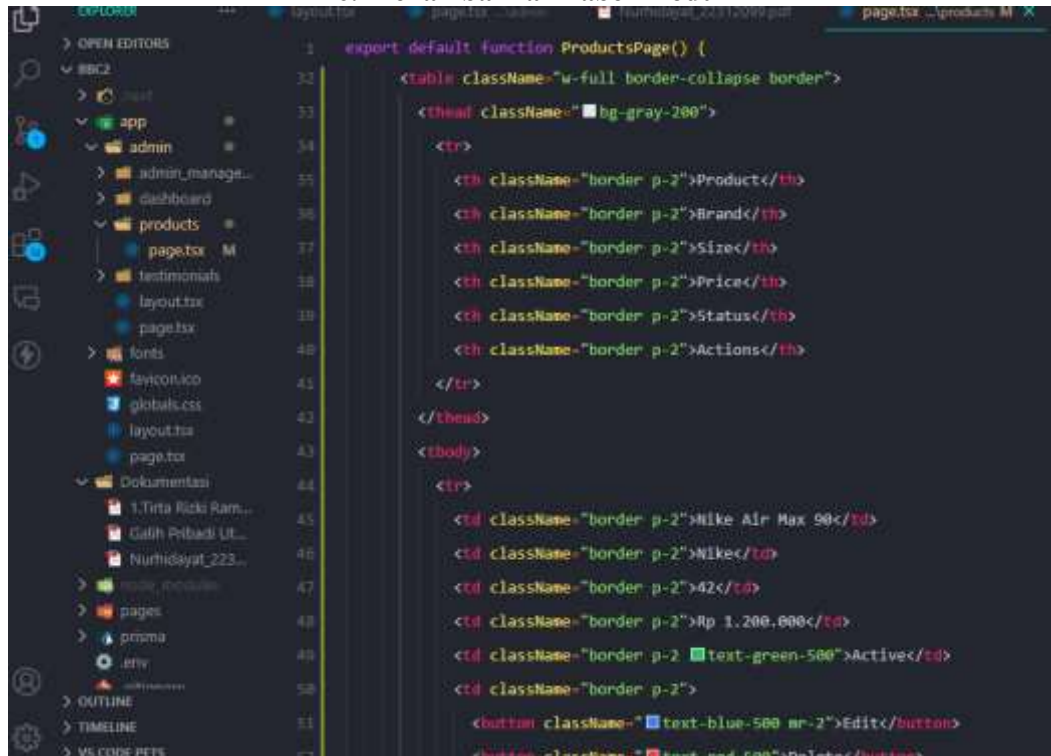
The screenshot shows a code editor with the Explorer sidebar on the left and the Editor window on the right. The Explorer sidebar shows a file tree with folders like 'admin' and 'products', and files like 'page.tsx'. The Editor window shows the code for 'page.tsx' with the following content:

```
1 export default function ProductsPage() {
14   </div>
15   <div className="grid grid-cols-3 gap-4 mb-4">
16     <select className="p-2 border rounded">
17       <option>All Brands</option>
18       <option>Nike</option>
19       <option>Adidas</option>
20     </select>
21     <select className="p-2 border rounded">
22       <option>All Conditions</option>
23       <option>New</option>
24       <option>Used</option>
25     </select>
26     <select className="p-2 border rounded">
27       <option>All Status</option>
28       <option>Active</option>
29       <option>Inactive</option>
30     </select>
31   </div>
32 </div>
```

Keterangan:

- **Tujuan:** Mempermudah penyaringan data produk berdasarkan kategori tertentu.
- **Langkah Teknis:** Menambahkan dropdown filter di file "page.tsx" dalam folder admin/product.
- **Hasil:** Dropdown filter produk.

## 6. Menambahkan Tabel Produk

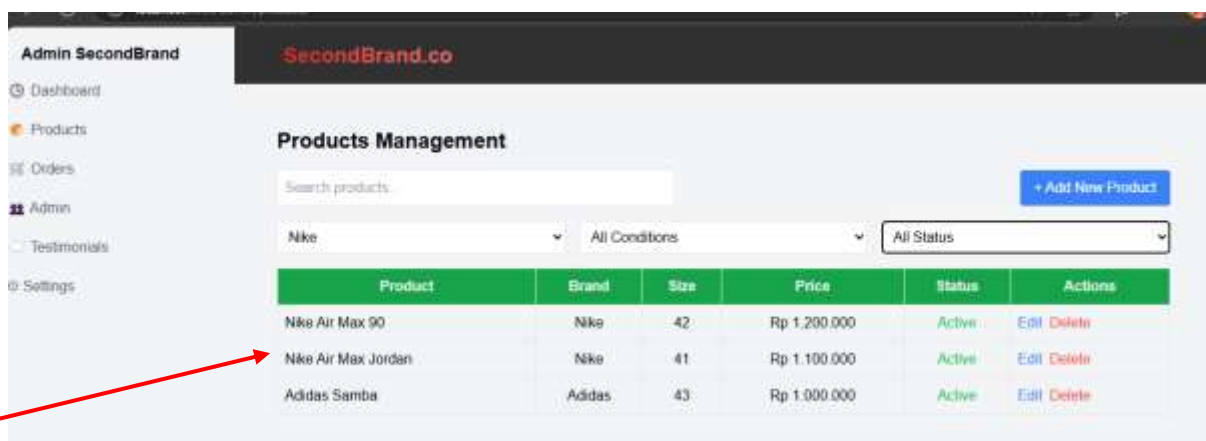


```
export default function ProductsPage() {  
  <table className="w-full border-collapse border">  
    <thead className="bg-gray-200">  
      <tr>  
        <th className="border p-2">Product</th>  
        <th className="border p-2">Brand</th>  
        <th className="border p-2">Size</th>  
        <th className="border p-2">Price</th>  
        <th className="border p-2">Status</th>  
        <th className="border p-2">Actions</th>  
      </tr>  
    </thead>  
    <tbody>  
      <tr>  
        <td className="border p-2">Nike Air Max 90</td>  
        <td className="border p-2">Nike</td>  
        <td className="border p-2">42</td>  
        <td className="border p-2">Rp 1.200.000</td>  
        <td className="border p-2 text-green-500">Active</td>  
        <td className="border p-2">  
          <button className="text-blue-500 mr-2">Edit</button>  
          <button className="text-red-500">Delete</button>  
        </td>  
      </tr>  
      <tr>  
        <td className="border p-2">Nike Air Max Jordan</td>  
        <td className="border p-2">Nike</td>  
        <td className="border p-2">41</td>  
        <td className="border p-2">Rp 1.100.000</td>  
        <td className="border p-2 text-green-500">Active</td>  
        <td className="border p-2">  
          <button className="text-blue-500 mr-2">Edit</button>  
          <button className="text-red-500">Delete</button>  
        </td>  
      </tr>  
      <tr>  
        <td className="border p-2">Adidas Samba</td>  
        <td className="border p-2">Adidas</td>  
        <td className="border p-2">43</td>  
        <td className="border p-2">Rp 1.000.000</td>  
        <td className="border p-2 text-green-500">Active</td>  
        <td className="border p-2">  
          <button className="text-blue-500 mr-2">Edit</button>  
          <button className="text-red-500">Delete</button>  
        </td>  
      </tr>  
    </tbody>  
  </table>  
}
```

Deskripsi:

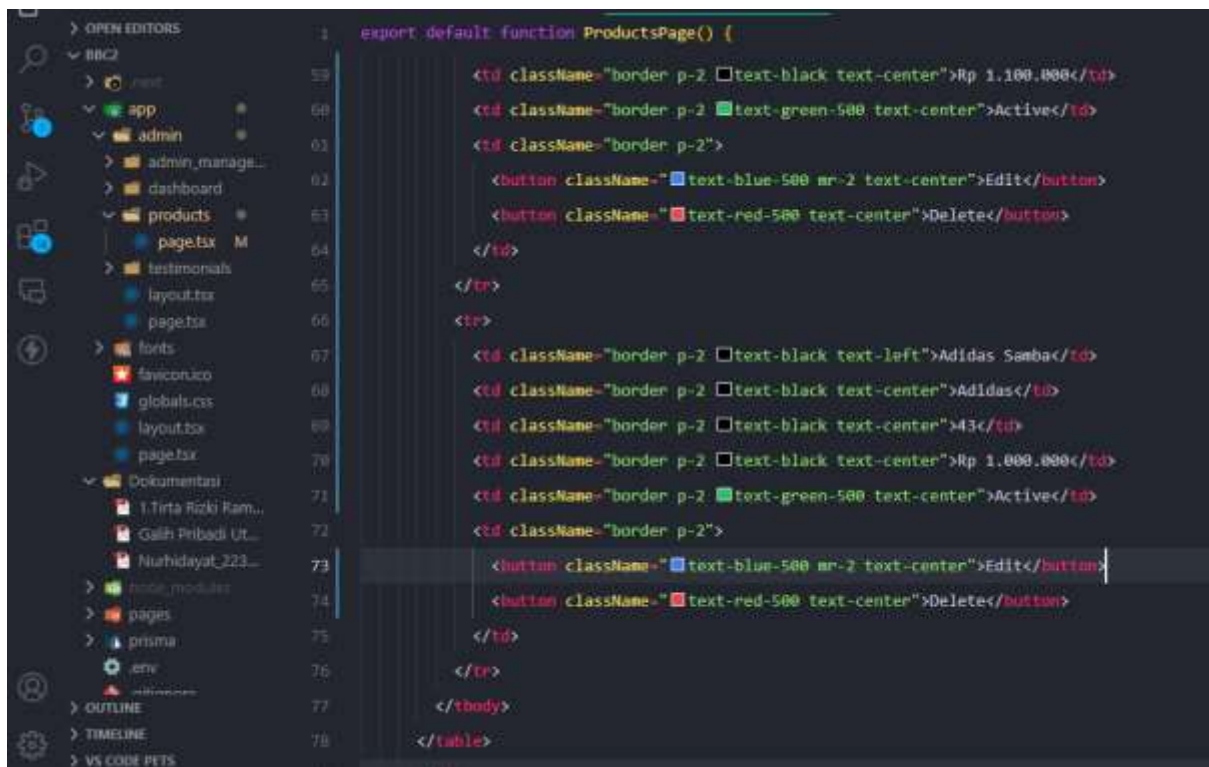
- **Tujuan:** Menampilkan data produk secara terstruktur.
- **Langkah Teknis:** Membuat tabel dengan data dummy.
- **Hasil:** Tabel produk dengan data dummy.

## 7. Menambahkan data produk



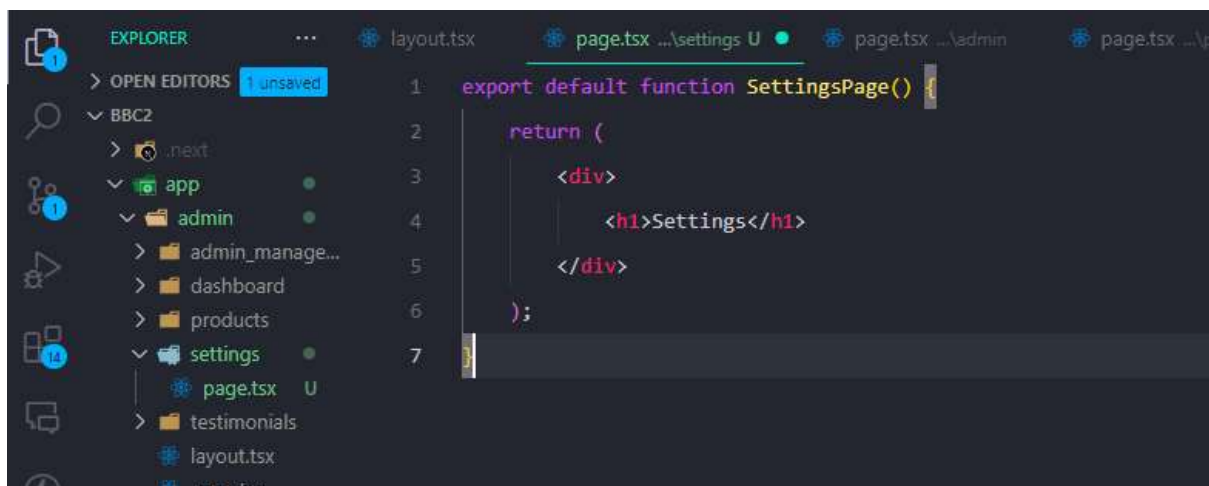
Keterangan:

- **Tujuan:** Menguji tampilan tabel dengan data produk.
- **Langkah Teknis:** Menambahkan data dummy ke tabel produk.
- **Hasil:** Tampilan tabel yang berisi data dummy produk.
- Dan kodingannya seperti dibawah ini



```
export default function ProductsPage() {  
  <td className="border p-2 text-black text-center">Rp 1.100.000</td>  
  <td className="border p-2 text-green-500 text-center">Active</td>  
  <td className="border p-2">  
    <button className="text-blue-500 mr-2 text-center">Edit</button>  
    <button className="text-red-500 text-center">Delete</button>  
  </td>  
</tr>  
<tr>  
  <td className="border p-2 text-black text-left">Adidas Samba</td>  
  <td className="border p-2 text-black text-center">Adidas</td>  
  <td className="border p-2 text-black text-center">43</td>  
  <td className="border p-2 text-black text-center">Rp 1.000.000</td>  
  <td className="border p-2 text-green-500 text-center">Active</td>  
  <td className="border p-2">  
    <button className="text-blue-500 mr-2 text-center">Edit</button>  
    <button className="text-red-500 text-center">Delete</button>  
  </td>  
</tr>  
</tbody>  
</table>
```

## 8. Membuat folder baru pada admin yaitu folder settings



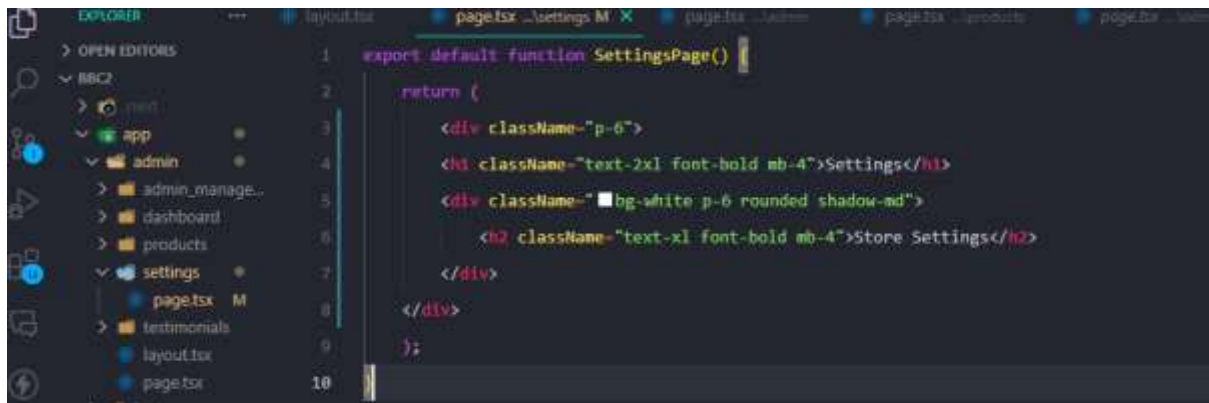
```
export default function SettingsPage() {  
  return (  
    <div>  
      <h1>Settings</h1>  
    </div>  
  );  
}
```

Keterangan :

- **Tujuan:** Membuat halaman khusus untuk pengaturan (Settings).
- **Langkah Teknis:** Membuat folder "settings" dalam admin dan menambahkan file "page.tsx".
- **Hasil:** Struktur awal halaman Settings.



## 9. Struktur Dasar dan Header pada Settings

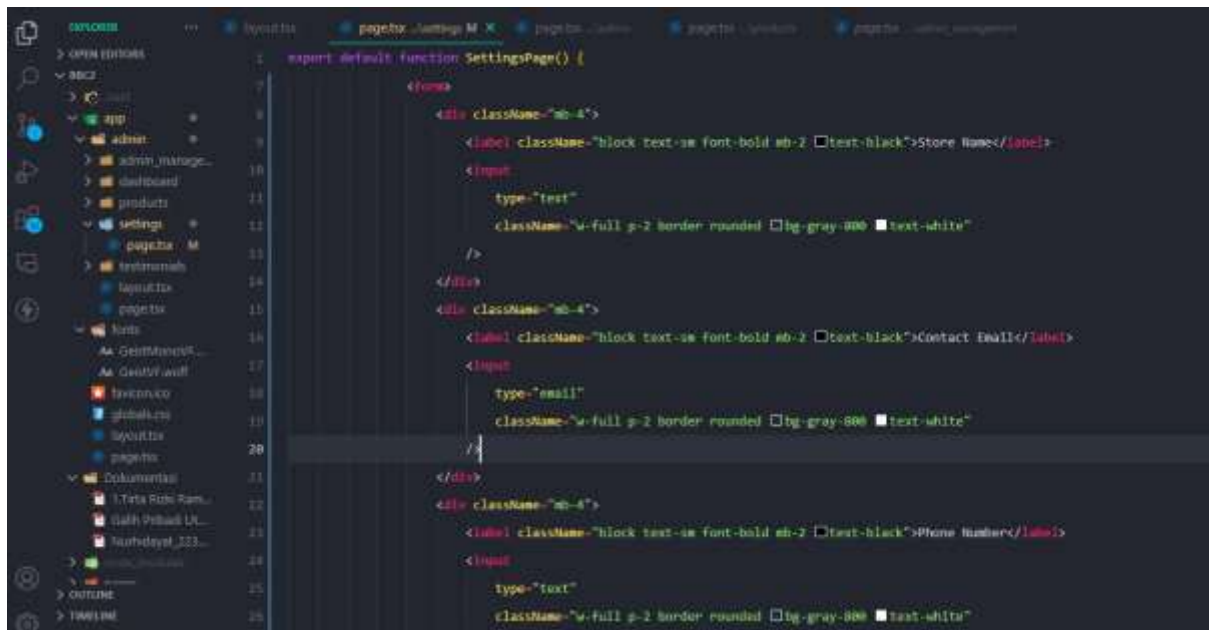


```
1 export default function SettingsPage() {
2
3   return (
4     <div className="p-6">
5       <h1 className="text-2xl font-bold mb-4">Settings</h1>
6       <div className="bg-white p-6 rounded shadow-md">
7         <h2 className="text-xl font-bold mb-4">Store Settings</h2>
8       </div>
9     </div>
10  );
11 }
```

Keterangan:

- **Tujuan:** Menyediakan kerangka dasar dan header untuk halaman Settings.
- **Langkah Teknis:** Menambahkan judul halaman dan header Store Settings.
- **Hasil:** Halaman Settings dengan header.

## 10. Menambahkan Form Input di Setting Page

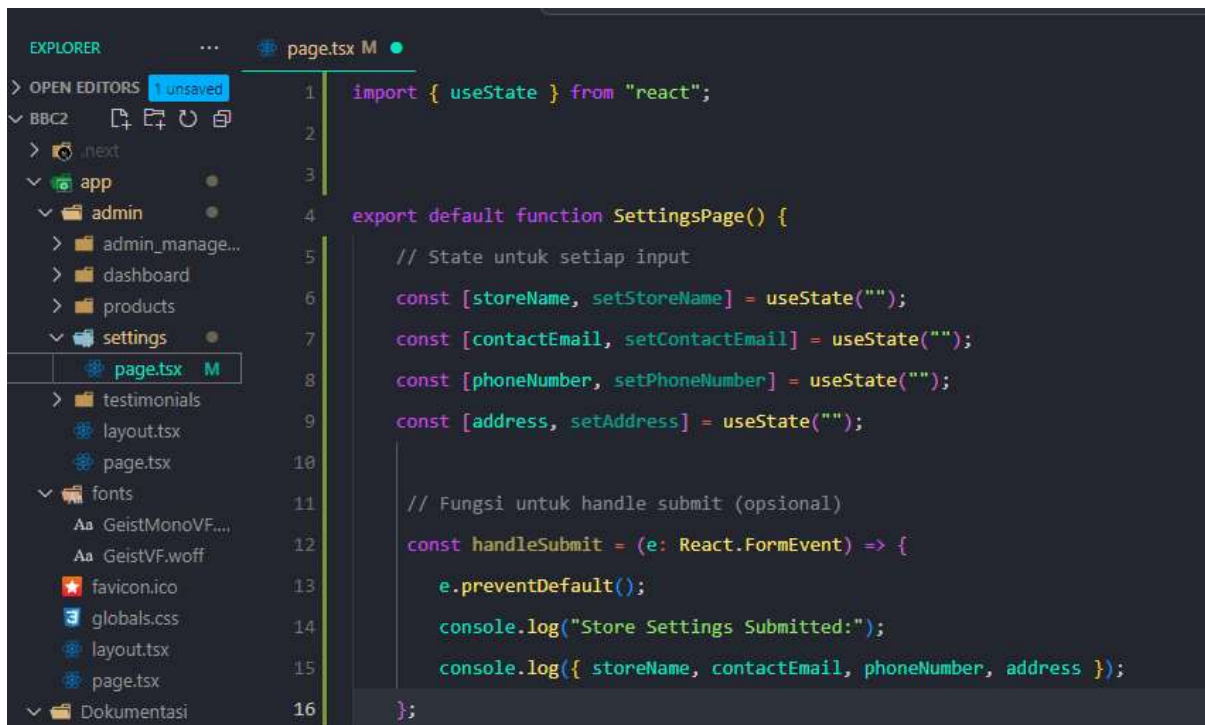


```
1 export default function SettingsPage() {
2
3   <form>
4     <div className="mb-4">
5       <label className="block text-sm font-bold mb-2">Store Name</label>
6       <input
7         type="text"
8         className="w-full p-2 border rounded bg-gray-800 text-white"
9       />
10    </div>
11    <div className="mb-4">
12      <label className="block text-sm font-bold mb-2">Contact Email</label>
13      <input
14        type="email"
15        className="w-full p-2 border rounded bg-gray-800 text-white"
16      />
17    </div>
18    <div className="mb-4">
19      <label className="block text-sm font-bold mb-2">Phone Number</label>
20      <input
21        type="text"
22        className="w-full p-2 border rounded bg-gray-800 text-white"
23      />
24    </div>
25    <div className="mb-4">
26      <label className="block text-sm font-bold mb-2">Address</label>
27      <input
28        type="text"
29        className="w-full p-2 border rounded bg-gray-800 text-white"
30      />
31    </div>
32  </form>
33 }
```

Keterangan :

- **Tujuan:** Menambahkan input untuk pengaturan informasi toko.
- **Langkah Teknis:** Menambahkan empat input field (Store Name, Contact Email, Phone Number, dan Address) dengan desain sesuai tema.
- **Hasil:** Form input untuk pengaturan toko.

## 11. Menambahkan State untuk setiap input dan Fungsi untuk handle Submit pada Settings

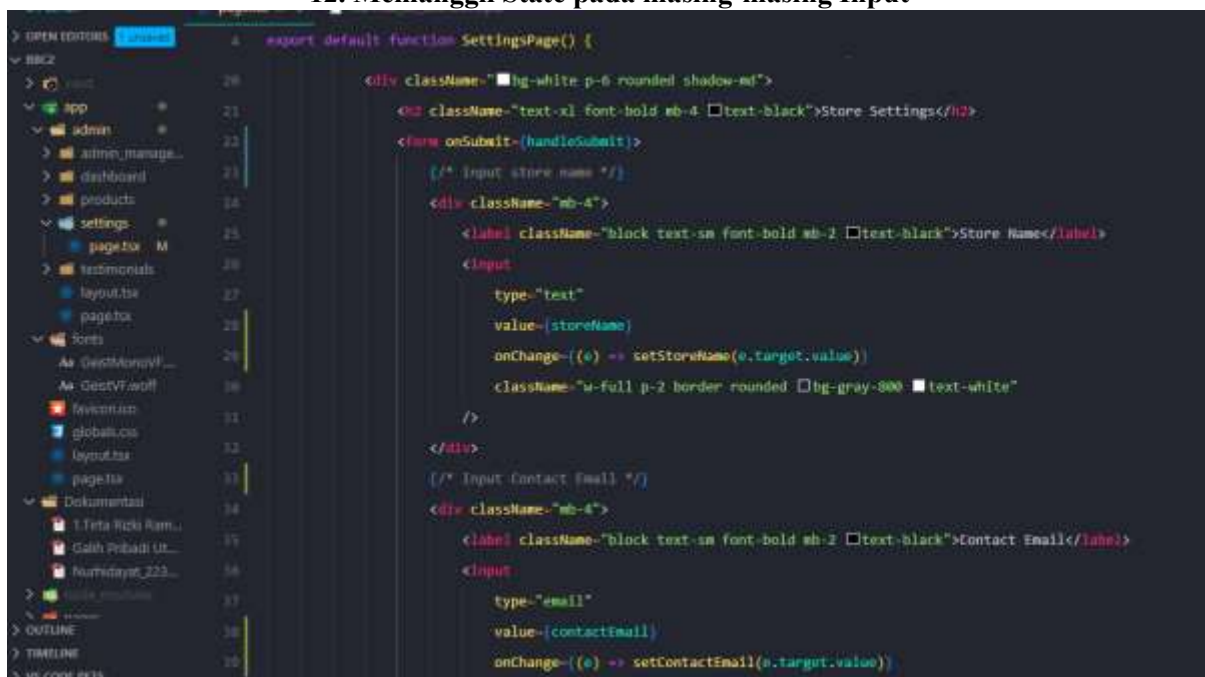


```
1  import { useState } from "react";
2
3
4  export default function SettingsPage() {
5      // State untuk setiap input
6      const [storeName, setStoreName] = useState("");
7      const [contactEmail, setContactEmail] = useState("");
8      const [phoneNumber, setPhoneNumber] = useState("");
9      const [address, setAddress] = useState("");
10
11     // Fungsi untuk handle submit (opsional)
12     const handleSubmit = (e: React.FormEvent) => {
13         e.preventDefault();
14         console.log("Store Settings Submitted:");
15         console.log({ storeName, contactEmail, phoneNumber, address });
16     };
17 }
```

Keterangan:

- useState digunakan untuk mengelola nilai setiap input field (storeName, contactEmail, phoneNumber, dan address).
- Nilai/isi akan diperbarui setiap kali pengguna mengetikkan sesuatu di input.
- Fungsi handleSubmit menangani saat form disubmit. Data form ditampilkan di console untuk memastikan form bekerja dengan baik.

## 12. Memanggil State pada masing-masing Input

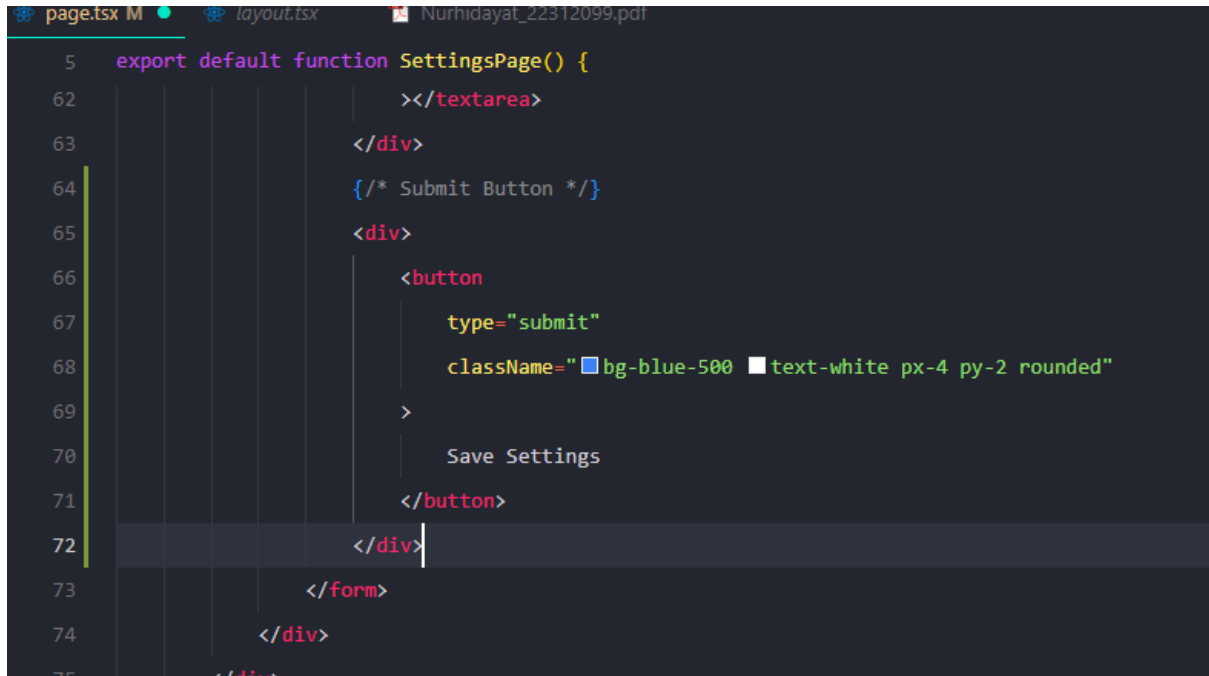


```
1  export default function SettingsPage() {
2
3      <div className="bg-white p-6 rounded shadow-md">
4          <h2 className="text-xl font-bold mb-4 text-black">Store Settings</h2>
5          <form onSubmit={handleSubmit}>
6              {/* Input store name */}
7              <div className="mb-4">
8                  <label className="block text-sm font-bold mb-2 text-black">Store Name</label>
9                  <input
10                      type="text"
11                      value={storeName}
12                      onChange={(e) => setStoreName(e.target.value)}
13                      className="w-full p-2 border rounded bg-gray-800 text-white"
14                  />
15              </div>
16              {/* Input Contact Email */}
17              <div className="mb-4">
18                  <label className="block text-sm font-bold mb-2 text-black">Contact Email</label>
19                  <input
20                      type="email"
21                      value={contactEmail}
22                      onChange={(e) => setContactEmail(e.target.value)}
23                  />
24              </div>
25          </form>
26      </div>
27  }
```

Keterangan :

- Menambahkan state untuk semua fields pada settings (storeName, contactEmail, phoneNumber, dan address).

#### 14. Menambahkan Submit Button



```
5  export default function SettingsPage() {
62      ></textarea>
63  }
64  { /* Submit Button */ }
65  <div>
66      <button
67          type="submit"
68          className="bg-blue-500 text-white px-4 py-2 rounded"
69      >
70          Save Settings
71      </button>
72  </div>
73  </form>
74  </div>
75  </div>
```

Keterangan :

- Saya menambahkan submit button / tombol submit untuk menyimpan perubahan



## 15. Menambahkan Notifications Settings

```
export default function SettingsPage() {  
  /** Notification Settings */  
  <div className="bg-white rounded-lg shadow p-6">  
    <div className="flex items-center gap-2 mb-4">  
      <Shield className="h-5 w-5 text-primary" text="black" />  
      <h2 className="text-xl font-bold text-gray-800">Notification Settings</h2>  
    </div>  
    <div className="space-y-4">  
      <div className="flex items-center justify-between">  
        <div>  
          <p className="font-medium text-gray-800">Order Notifications</p>  
          <p className="text-sm text-gray-600">Receive notifications for new orders</p>  
        </div>  
        <input type="checkbox" className="toggle toggle-primary" defaultChecked />  
      </div>  
      <div className="flex items-center justify-between">  
        <div>  
          <p className="font-medium text-gray-800">Review Notifications</p>  
          <p className="text-sm text-gray-600">Receive notifications for new reviews</p>  
        </div>  
        <input type="checkbox" className="toggle toggle-primary" defaultChecked />  
      </div>  
    </div>  
  </div>  
}
```

Keterangan :

- **Tujuan:** Menyediakan opsi untuk mengatur notifikasi.
- **Langkah Teknis:** Menambahkan pengaturan notifikasi pada halaman Settings.
- **Hasil:** Seksi Notifications Settings.

## 16. Menambahkan Security Settings

```
export default function SettingsPage() {  
  /** Security Settings */  
  <div className="bg-white rounded-lg shadow p-6">  
    <div className="flex items-center gap-2 mb-4">  
      <Shield className="h-5 w-5 text-primary" text="black" />  
      <h2 className="text-xl font-bold text-gray-800">Security Settings</h2>  
    </div>  
    <div className="space-y-4">  
      <div>  
        <label className="block text-sm font-medium text-gray-600 mb-2">  
          Current Password  
        </label>  
        <input  
          type="password"  
          className="input input-bordered w-full"  
          placeholder="Enter current password"  
        />  
      </div>  
      <div>  
        <label className="block text-sm font-medium text-gray-600 mb-2">  
          New Password  
        </label>  
      </div>  
    </div>  
  </div>  
}
```

Keterangan:

- **Tujuan:** Menambahkan pengaturan keamanan untuk mengubah kata sandi.
- **Langkah Teknis:** Menambahkan field Current Password, New Password, dan Confirm Password.
- **Hasil:** Seksi Security Settings.

### Perbaikan pada page settings

17. Melakukan penambahan state untuk Notification Settings, Security Settings, dan Fungsi untuk handle Submit

```
export default function SettingsPage() {  
  10   const [address, setAddress] = useState("");  
  11  
  12   // State untuk Notification Settings  
  13   const [orderNotifications, setOrderNotifications] = useState(true);  
  14   const [reviewNotifications, setReviewNotifications] = useState(true);  
  15  
  16   // State untuk Security Settings  
  17   const [currentPassword, setCurrentPassword] = useState("");  
  18   const [newPassword, setNewPassword] = useState("");  
  19   const [confirmPassword, setConfirmPassword] = useState("");  
  20  
  21   // Fungsi untuk handle submit  
  22   const handleSubmit = (e: React.FormEvent) => {  
  23     e.preventDefault();  
  24     console.log("Form Submitted:", {  
  25       storeSettings: { storeName, contactEmail, phoneNumber, address },  
  26       notificationSettings: { orderNotifications, reviewNotifications },  
  27       securitySettings: { currentPassword, newPassword, confirmPassword },  
  28     });  
  29   };  
}
```

Keterangan:

- **State pada Notification Settings:** Untuk mengelola status notifikasi (aktif/nonaktif).
- **State pada Store Settings:** Untuk menyimpan data terkait toko, seperti nama, email, nomor telepon, dan alamat.
- **State pada Security Settings:** Untuk menangani input kata sandi saat pengguna ingin mengubahnya.

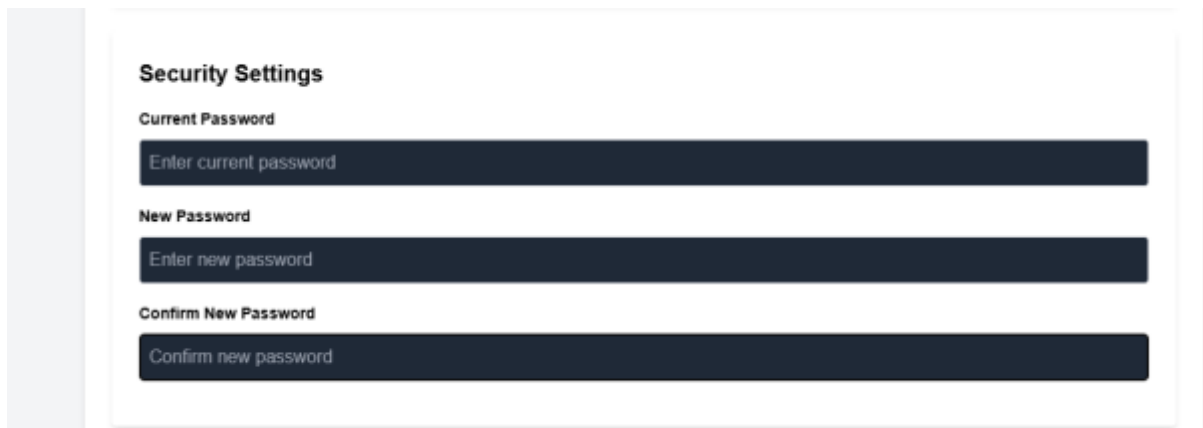
### 18. Melakukan perbaikan pada Notification Settings



Keterangan:

- Ini adalah tampilan Notification Settings setelah diperbaiki

## 19. Perbaikan pada Security Settings

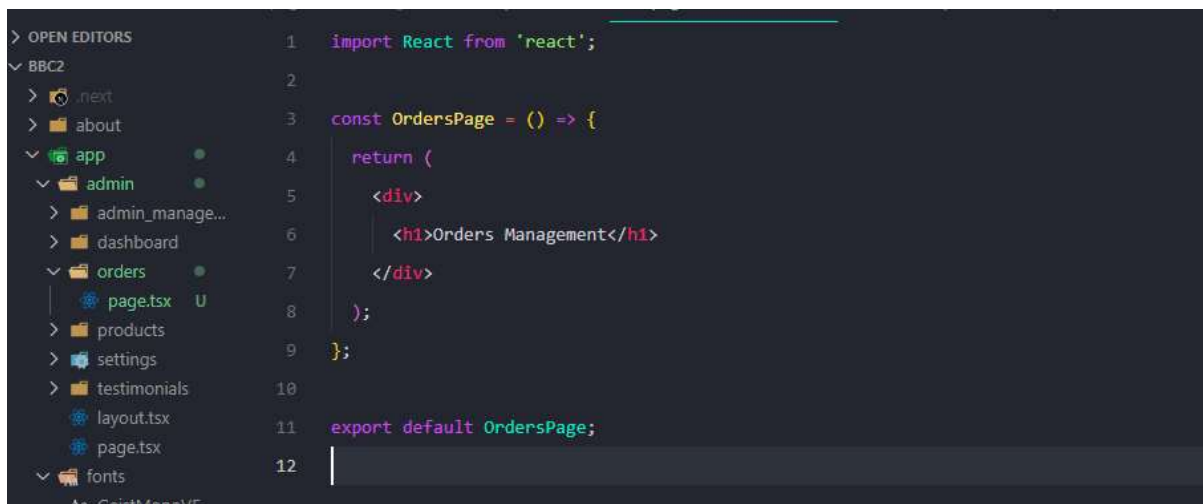


The screenshot shows a 'Security Settings' form with three input fields. The first field is labeled 'Current Password' and contains the placeholder text 'Enter current password'. The second field is labeled 'New Password' and contains the placeholder text 'Enter new password'. The third field is labeled 'Confirm New Password' and contains the placeholder text 'Confirm new password'. All fields are dark blue with white text.

Keterangan:

- Ini adalah tampilan Security Settings setelah diperbaiki

Membuat folder orders di dalam folder admin dan file page.tsx dengan komponen React dasar



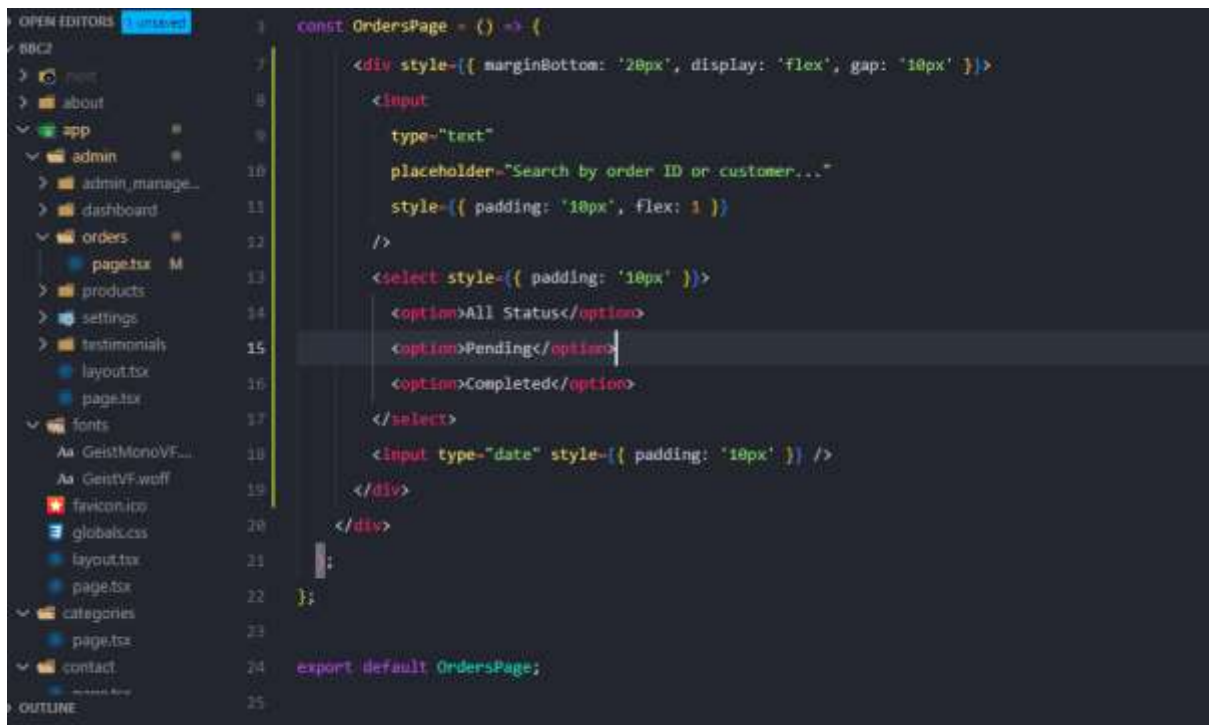
The screenshot shows a code editor with a file explorer on the left and a code editor on the right. The file explorer shows a project structure with folders like .next, about, app, admin, dashboard, orders, products, settings, testimonials, layout.tsx, page.tsx, and fonts. The 'orders' folder is highlighted. The code editor shows the following code:

```
1  import React from 'react';
2
3  const OrdersPage = () => {
4    return (
5      <div>
6        <h1>Orders Management</h1>
7      </div>
8    );
9  };
10
11  export default OrdersPage;
12
```

Keterangan:

- **Tujuan:** Membuat halaman untuk mengelola pesanan.
- **Langkah Teknis:** Membuat folder "orders" dan file "page.tsx" dengan komponen React dasar.
- **Hasil:** Struktur awal halaman Orders.

## Menambahkan struktur halaman layout pada page orders

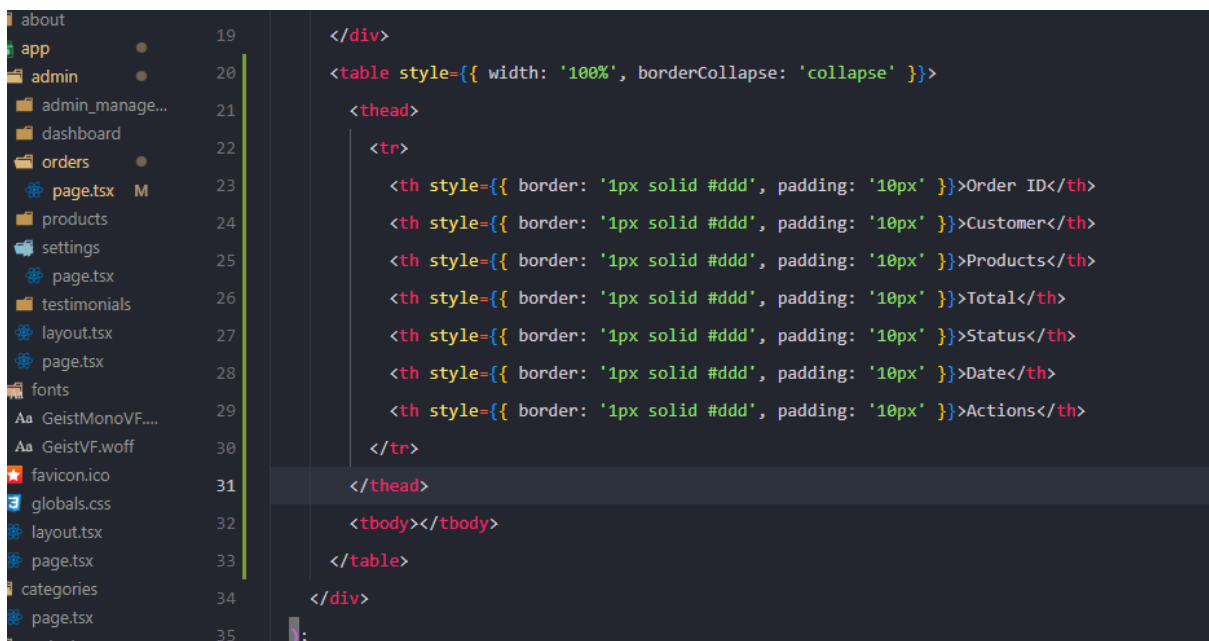


```
const OrdersPage = () => {  
  <div style={{ marginBottom: '20px', display: 'flex', gap: '10px' }}>  
    <input  
      type="text"  
      placeholder="Search by order ID or customer..."  
      style={{ padding: '10px', flex: 1 }}  
    />  
    <select style={{ padding: '10px' }}>  
      <option>All Status</option>  
      <option>Pending</option>  
      <option>Completed</option>  
    </select>  
    <input type="date" style={{ padding: '10px' }} />  
  </div>  
};  
  
export default OrdersPage;
```

Keterangan :

- **Tujuan:** Membuat halaman Orders lebih fungsional.
- **Langkah Teknis:** Menambahkan input pencarian, dropdown filter status, dan input tanggal.
- **Hasil:** Halaman Orders dengan elemen filter.

## Menambahkan Struktur Tabel Kosong pada page orders

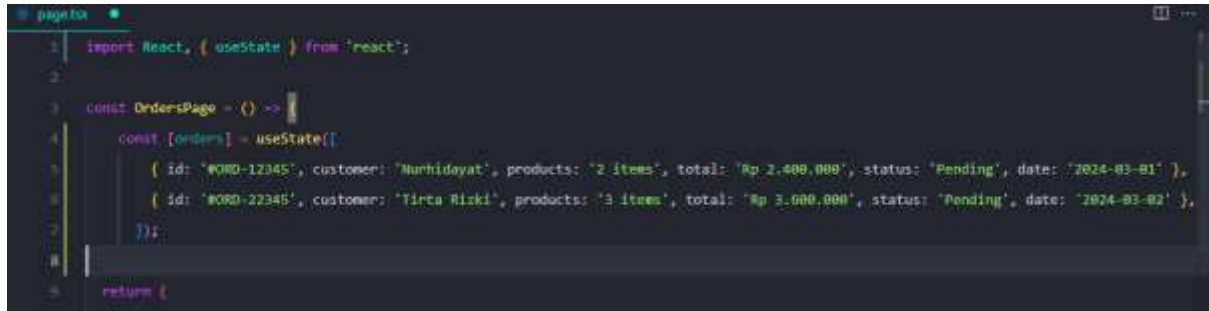


```
</div>  
  
<table style={{ width: '100%', borderCollapse: 'collapse' }}>  
  <thead>  
    <tr>  
      <th style={{ border: '1px solid #ddd', padding: '10px' }}>Order ID</th>  
      <th style={{ border: '1px solid #ddd', padding: '10px' }}>Customer</th>  
      <th style={{ border: '1px solid #ddd', padding: '10px' }}>Products</th>  
      <th style={{ border: '1px solid #ddd', padding: '10px' }}>Total</th>  
      <th style={{ border: '1px solid #ddd', padding: '10px' }}>Status</th>  
      <th style={{ border: '1px solid #ddd', padding: '10px' }}>Date</th>  
      <th style={{ border: '1px solid #ddd', padding: '10px' }}>Actions</th>  
    </tr>  
  </thead>  
  <tbody></tbody>  
</table>  
</div>  
;
```

Keterangan:

- **Tujuan:** Menampilkan data pesanan secara terstruktur.
- **Langkah Teknis:** Menambahkan tabel HTML dasar di bawah elemen filter.
- **Hasil:** Tabel kosong pada halaman Orders.

### Menambahkan array data dummy menggunakan useState di file page.tsx



```

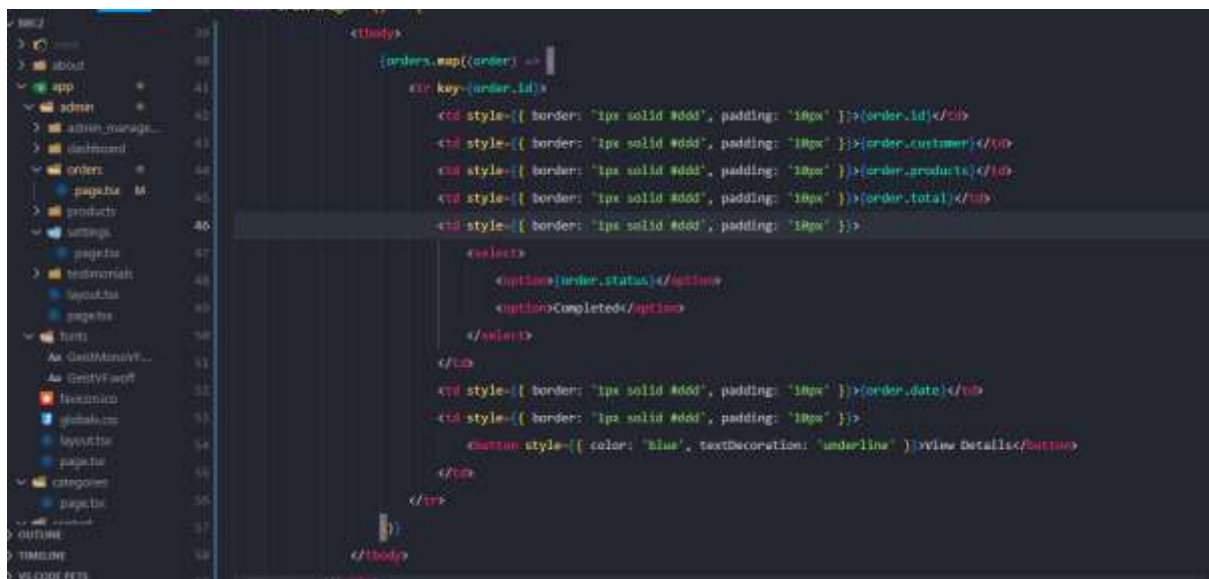
1 import React, { useState } from 'react';
2
3 const OrdersPage = () => {
4   const [orders] = useState([
5     { id: 'ORD-12345', customer: 'Nurhidayat', products: '2 items', total: 'Rp 2.400.000', status: 'Pending', date: '2024-03-01' },
6     { id: 'ORD-22345', customer: 'Tirta Riski', products: '3 items', total: 'Rp 3.600.000', status: 'Pending', date: '2024-03-02' },
7   ]);
8
9   return (

```

Keterangan:

- **Tujuan:** Mengisi tabel dengan data contoh untuk pengujian.
- **Langkah Teknis:** Menambahkan array data dummy menggunakan useState dan merendernya ke tabel dengan fungsi map.
- **Hasil:** Tabel dengan data dummy pesanan.

### Merender Data Dummy ke Tabel



```
|  |
| --- |
| {orders.map((order) => {   <tr>     <td style={{ border: '1px solid #ddd', padding: '10px' }}>{order.id}</td>     <td style={{ border: '1px solid #ddd', padding: '10px' }}>{order.customer}</td>     <td style={{ border: '1px solid #ddd', padding: '10px' }}>{order.products}</td>     <td style={{ border: '1px solid #ddd', padding: '10px' }}>{order.total}</td>     <td style={{ border: '1px solid #ddd', padding: '10px' }}>       <select>         <option>{order.status}</option>         <option>Completed</option>       </select>     </td>     <td style={{ border: '1px solid #ddd', padding: '10px' }}>{order.date}</td>     <td style={{ border: '1px solid #ddd', padding: '10px' }}>       <button style={{ color: 'blue', textDecoration: 'underline' }}>View Details</button>     </td>   </tr> })} |

```

Keterangan:

- Menggunakan fungsi map untuk merender data dummy ke dalam elemen <tbody>
- Menambahkan logika untuk membuat baris tabel berdasarkan data dummy.
- Menambahkan elemen interaktif seperti dropdown status dan tombol "View Details" untuk setiap baris.



## Menambahkan data dummy pada page orders

Orders Management						
Search by order ID or customer				All Status	dd/mm/yyyy	
Order ID	Customer	Products	Total	Status	Date	Actions
#ORD-12345	Nurhidayat	2 items	Rp 2.400.000	Pending	2024-03-01	<a href="#">View Details</a>
#ORD-22345	Tirta Rizki	3 items	Rp 3.600.000	Pending	2024-03-02	<a href="#">View Details</a>
#ORD-22346	Afrando	2 items	Rp 2.600.000	Pending	2024-03-03	<a href="#">View Details</a>
#ORD-22347	Udin	1 items	Rp 1.600.000	Pending	2024-03-03	<a href="#">View Details</a>
#ORD-22346	Umar Al-Faruq	2 items	Rp 2.600.000	Pending	2024-03-01	<a href="#">View Details</a>
#ORD-22347	Galih Pribadi Utomo	3 items	Rp 3.600.000	Pending	2024-03-02	<a href="#">View Details</a>

Keterangan: Saya menambahkan data dummy pada page orders dan gambar diatas merupakan tampilan page orders

## Membuat struktur dasar komponen pada page categories

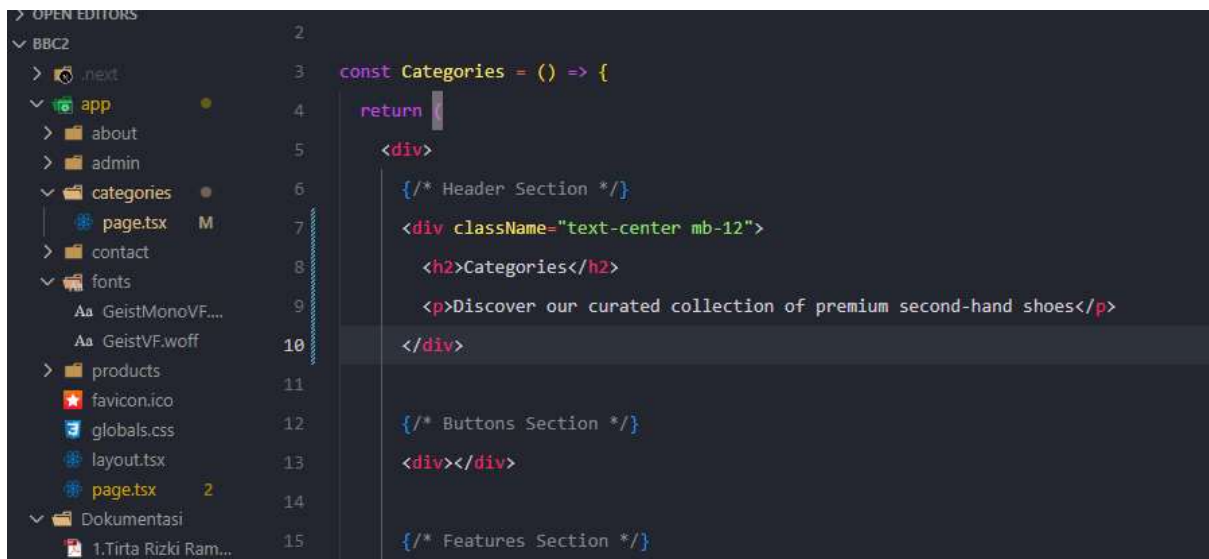
```
EXPLORER
  > OPEN EDITORS
  > BBC2
  > .next
  > app
    > about
    > admin
    > categories
      page.tsx
    > contact
    > fonts
      GeistMonoVF....
      GeistVF.woff
    > products
      favicon.ico
      globals.css
      layout.tsx
      page.tsx
    > Dokumentasi
      1.Tirta Rizki Ram...
```

```
1 import React from "react";
2
3 const Categories = () => {
4   return (
5     <div>
6       {/* Header Section */}
7       <div></div>
8
9       {/* Buttons Section */}
10      <div></div>
11
12      {/* Features Section */}
13      <div></div>
14    </div>
```

Keterangan:

- **Tujuan:** Menyediakan kerangka awal untuk halaman Categories.
- **Langkah Teknis:** Membuat struktur komponen awal pada Categories.
- **Hasil:** Kerangka awal halaman Categories.

## Menambahkan Header Section pada page Categories

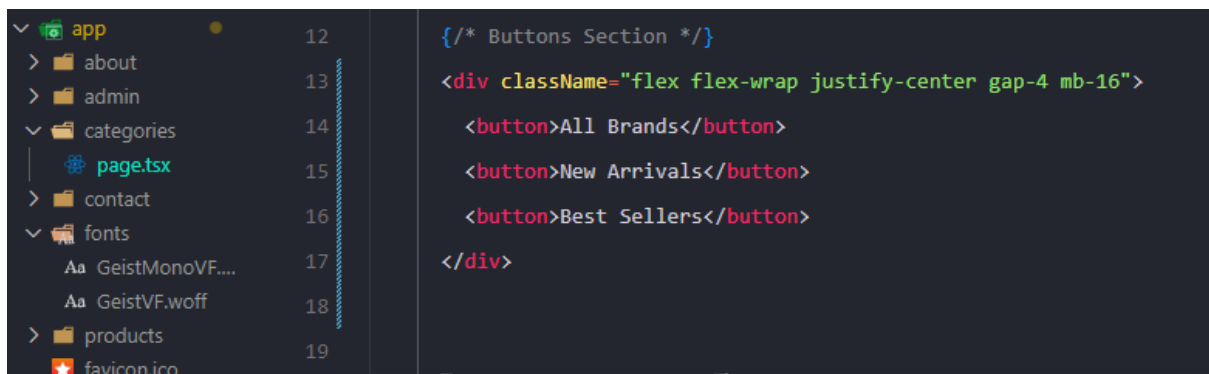


```
2
3 const Categories = () => {
4   return
5     <div>
6       {/* Header Section */}
7       <div className="text-center mb-12">
8         <h2>Categories</h2>
9         <p>Discover our curated collection of premium second-hand shoes</p>
10      </div>
11
12      {/* Buttons Section */}
13      <div></div>
14
15      {/* Features Section */}
```

Keterangan:

- **Tujuan:** Menyediakan bagian judul dan deskripsi untuk Categories.
- **Langkah Teknis:** Menambahkan elemen header pada halaman Categories.
- **Hasil:** Header section pada Categories.

## Menambahkan Button Section

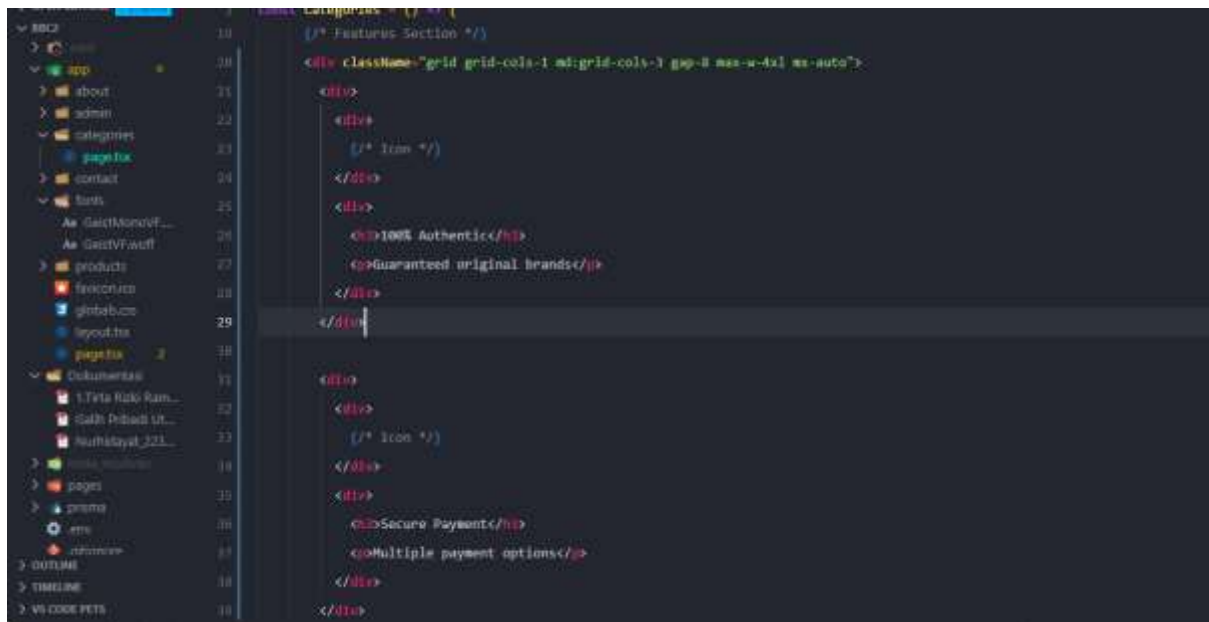


```
12      {/* Buttons Section */}
13
14      <div className="flex flex-wrap justify-center gap-4 mb-16">
15        <button>All Brands</button>
16        <button>New Arrivals</button>
17        <button>Best Sellers</button>
18      </div>
19
```

Keterangan:

- **MeTujuan:** Menyediakan tombol-tombol untuk pengelolaan kategori.
- **Langkah Teknis:** Menambahkan tombol tanpa styling.
- **Hasil:** Button section pada Categories

## Menambahkan Features Section

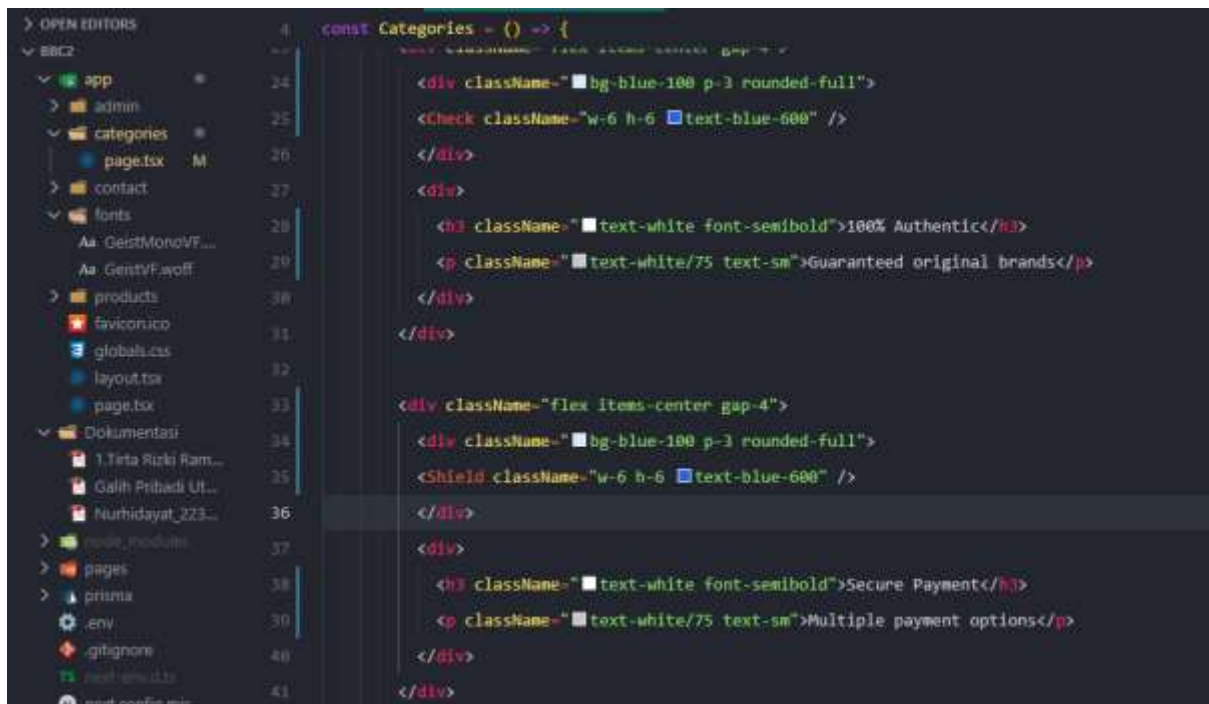


```
const Categories = () => {  
  /* Features Section */  
  <div className="grid grid-cols-1 md:grid-cols-1 gap-8 max-w-4xl mx-auto">  
    <div>  
      /* Icon */  
    </div>  
    <div>  
      <h3>100% Authentic</h3>  
      <p>Guaranteed original brands</p>  
    </div>  
  </div>  
  
  <div>  
    <div>  
      /* Icon */  
    </div>  
    <div>  
      <h3>Secure Payment</h3>  
      <p>Multiple payment options</p>  
    </div>  
  </div>  
}
```

Keterangan:

- **Tujuan:** Menyediakan bagian fitur untuk Categories.
- **Langkah Teknis:** Menambahkan struktur dasar fitur tanpa ikon.
- **Hasil:** Features section pada Categories.

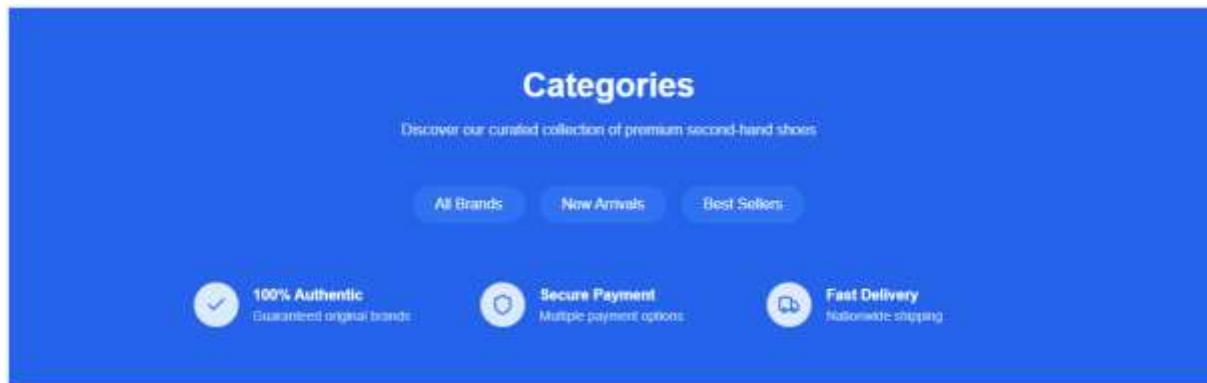
## Menambahkan Styling dan Icons pada page Categories



```
const Categories = () => {  
  /* Features Section */  
  <div className="bg-blue-100 p-3 rounded-full">  
    <Check className="w-6 h-6 text-blue-600" />  
  </div>  
  
  <div>  
    <h3 className="text-white font-semibold">100% Authentic</h3>  
    <p className="text-white/75 text-sm">Guaranteed original brands</p>  
  </div>  
  
  <div>  
    <div>  
      <div className="bg-blue-100 p-3 rounded-full">  
        <Shield className="w-6 h-6 text-blue-600" />  
      </div>  
  
      <div>  
        <h3 className="text-white font-semibold">Secure Payment</h3>  
        <p className="text-white/75 text-sm">Multiple payment options</p>  
      </div>  
    </div>  
  </div>  
}
```

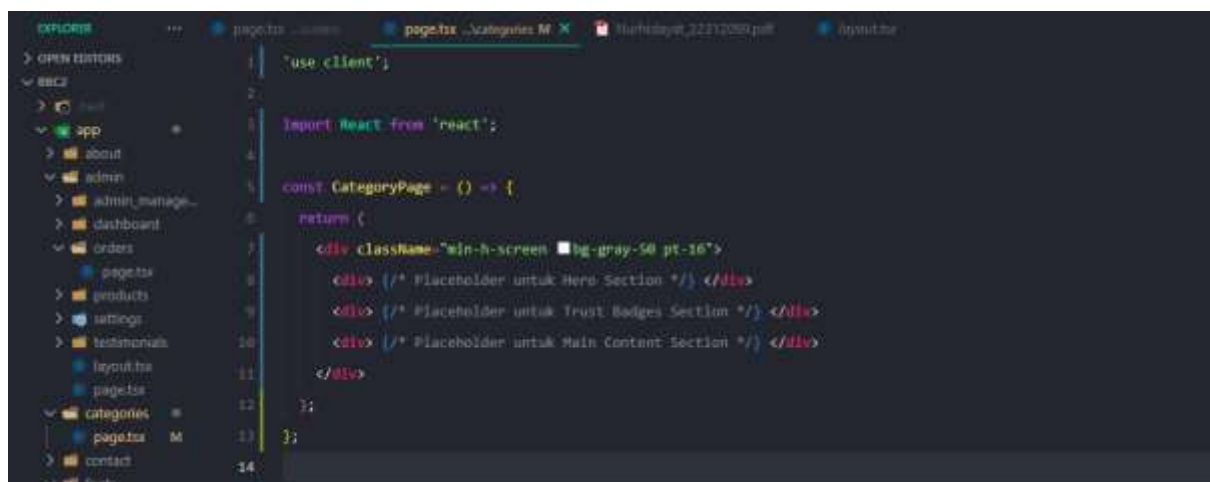
Deskripsi:

Pada tahap ini saya menambahkan styling dan ikon seperti ikon Check, Shield dan Truck. Komponen siap digunakan dengan tampilan lengkap. Dan tampilannya seperti gambar dibawah ini



Perbaikan pada page Category

Setup struktur dasar untuk komponen CategoryPage



Deskripsi:

Saya membuat struktur awal komponen pada category page dan memisahkan menjadi tiga bagian utama yaitu Hero Section (header utama), Trust Badges Section (ikon kepercayaan), dan Main Content Section (konten utama).

## Implementasi layout Hero Section



```
import React from 'react';

const CategoryPage = () => {
  return (
    <div className="min-h-screen bg-gray-50 pt-16">
      <div className="bg-gradient-to-r from-blue-600 to-blue-800 text-white">
        <div className="max-w-7xl mx-auto px-4 py-16">
          <h1 className="text-4xl font-bold mb-4">Categories</h1>
          <p className="text-blue-100">
            Temukan koleksi sepatu second-hand premium kami
          </p>
        </div>
      </div>
    </div>
  );
}
```

Deskripsi :

Hero Section ditambahkan untuk menjadi pengantar utama halaman kategori. Bagian ini memiliki latar belakang dengan gradasi biru, teks judul besar, dan deskripsi singkat untuk memberikan gambaran tentang isi halaman.

## Menambahkan tombol Navigasi di Hero Section



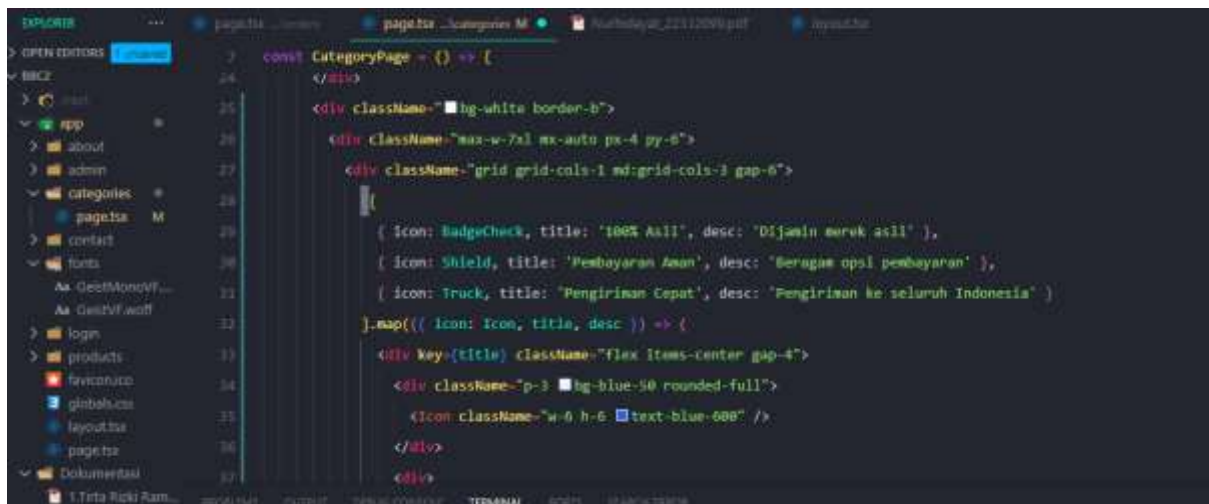
```
</p>
</div>
<div className="flex gap-4">
  {['All Brands', 'New Arrivals', 'Best Sellers'].map((item) => (
    <button key={item} className="px-6 py-2 bg-white/10 rounded-full hover:bg-white/20">
      {item}
    </button>
  ))}
</div>
```

Deskripsi :

Saya menambahkan tombol navigasi di Hero Section untuk memudahkan pengguna berpindah ke kategori seperti "All Brands", "New Arrivals", dan "Best Sellers". Tombol ini dilengkapi efek hover dengan transisi warna agar tampilannya lebih interaktif.



## Membuat Trust Badges Section untuk informasi tambahan



```
const CategoryPage = () => {  
  </div>  
  <div className="bg-white border-b">  
    <div className="max-w-7xl mx-auto px-4 py-6">  
      <div className="grid grid-cols-1 md:grid-cols-3 gap-6">  
        {  
          [ { icon: BadgeCheck, title: '100% Asli', desc: 'Dijamin merek asli' },  
            { icon: Shield, title: 'Pembayaran Aman', desc: 'Beragam opsi pembayaran' },  
            { icon: Truck, title: 'Pengiriman Cepat', desc: 'Pengiriman ke seluruh Indonesia' }  
        ].map(( { icon, title, desc } ) => {  
          <div key={title} className="flex items-center gap-4">  
            <div className="p-3 bg-blue-50 rounded-full">  
              <Icon className="w-6 h-6 text-blue-600" />  
            </div>  
            <div>  
              <h3>{title}</h3>  
              <p>{desc}</p>  
            </div>  
          </div>  
        })  
      </div>  
    </div>  
  </div>  
}
```

Deskripsi:

Bagian ini bertujuan untuk menonjolkan keunggulan layanan dengan menambahkan Trust Badges, seperti "100% Asli", "Pembayaran Aman", dan "Pengiriman Cepat". Masing-masing dilengkapi ikon dan deskripsi singkat untuk membangun rasa percaya pengguna terhadap layanan.

## Menambahkan daftar kategori dengan state management



```
const CategoryPage = () => {  
  <div className="md:w-64">  
    <div className="bg-white rounded-xl shadow-sm p-6">  
      <h3 className="font-bold mb-4">Browse Categories</h3>  
      <div className="space-y-2">  
        {categories.map((cat) => {  
          <button  
            key={cat.name}  
            onClick={() => setActiveCategory(cat.name)}  
            className={`w-full flex items-center justify-between px-4 py-2 rounded-lg transition-colors ${activeCategory === cat.name  
              ? 'bg-blue-50 text-blue-600'  
              : 'bg-gray-50'}`  
          >  
            {cat.name}  
          </button>  
        })  
      </div>  
    </div>  
  </div>  
}
```

Deskripsi:

Daftar kategori dibuat untuk menampilkan nama kategori dan jumlah produk di setiap kategori. Dengan menggunakan state `activeCategory`, kita bisa melacak kategori mana yang sedang aktif dipilih oleh pengguna, sehingga tampilan bisa disesuaikan secara dinamis.

## Menambahkan navigasi breadcrumb untuk kategori aktif

```
/* Category content */  
  
<div className="flex items-center gap-2 text-sm text-gray-600 mb-6">  
  <span>Home</span>  
  <ChevronRight className="w-4 h-4" />  
  <span>Categories</span>  
  <ChevronRight className="w-4 h-4" />  
  <span className="text-blue-600">{activeCategory}</span>  
</div>
```

Deskripsi :

Breadcrumb ditambahkan untuk membantu pengguna memahami posisi mereka di dalam hierarki navigasi. Breadcrumb ini menunjukkan jalur seperti "Home > Categories > [Kategori yang Dipilih]" agar pengalaman pengguna lebih terarah.

## Menambahkan grid produk untuk kategori aktif

```
<div className="grid grid-cols-2 md:grid-cols-3 gap-6">  
  {[1, 2, 3, 4, 5, 6].map((item) => {  
    <div  
      key={item}  
      className="group relative overflow-hidden rounded-lg transition-all hover:shadow-md">  
      <div className="aspect-square bg-gray-100"></div>  
      <div className="absolute inset-0 bg-gradient-to-t from-black/60 to-transparent flex items-end">  
        <div className="p-4 w-full transform translate-y-4 group-hover:translate-y-0 transition-transform">  
          <p className="text-white font-medium">Nike Air Max</p>  
          <p className="text-white/80 text-sm">Mulai dari Rp 880.000</p>  
        </div>  
      </div>  
    )}</div>
```

Deskripsi:

Grid produk dibuat untuk menampilkan daftar produk dalam kategori yang sedang aktif. Setiap produk memiliki efek hover dan dilengkapi detail sederhana seperti nama produk dan harga awal, memberikan kesan modern dan informatif.

The screenshot shows the VS Code interface. On the left, the file explorer displays a project named 'petstore' with a 'pages' folder containing 'featured.vue'. The main editor shows the content of 'featured.vue', which is a Vue.js component template. The template uses a grid layout to display a list of featured collections. Each collection item is represented by a card with a title, description, and two buttons: 'View Collection' and 'Add Item'. The template uses Vue.js directives like 'v-for' and 'v-bind' to iterate over a list of collections and bind data to the template elements.

Saya menambahkan bagian koleksi unggulan untuk menarik perhatian pengguna pada kategori istimewa seperti "Limited Edition" dan "New Arrivals". Setiap koleksi memiliki deskripsi singkat dan tombol untuk melihat koleksi tersebut secara lebih detail.