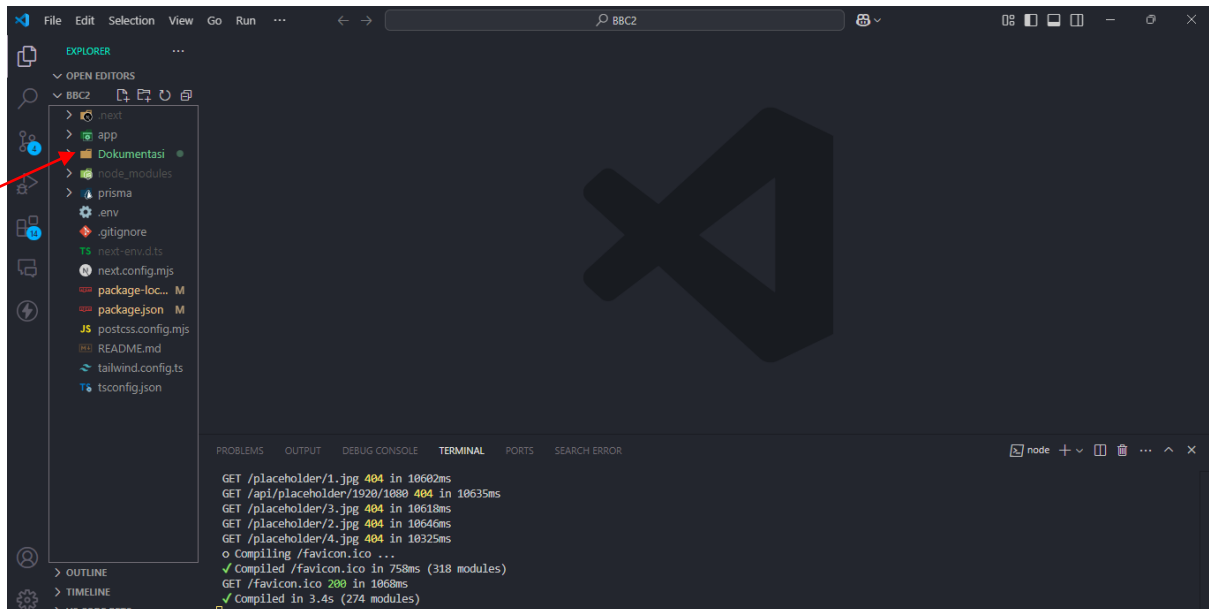


NAMA : NURHIDAYAT

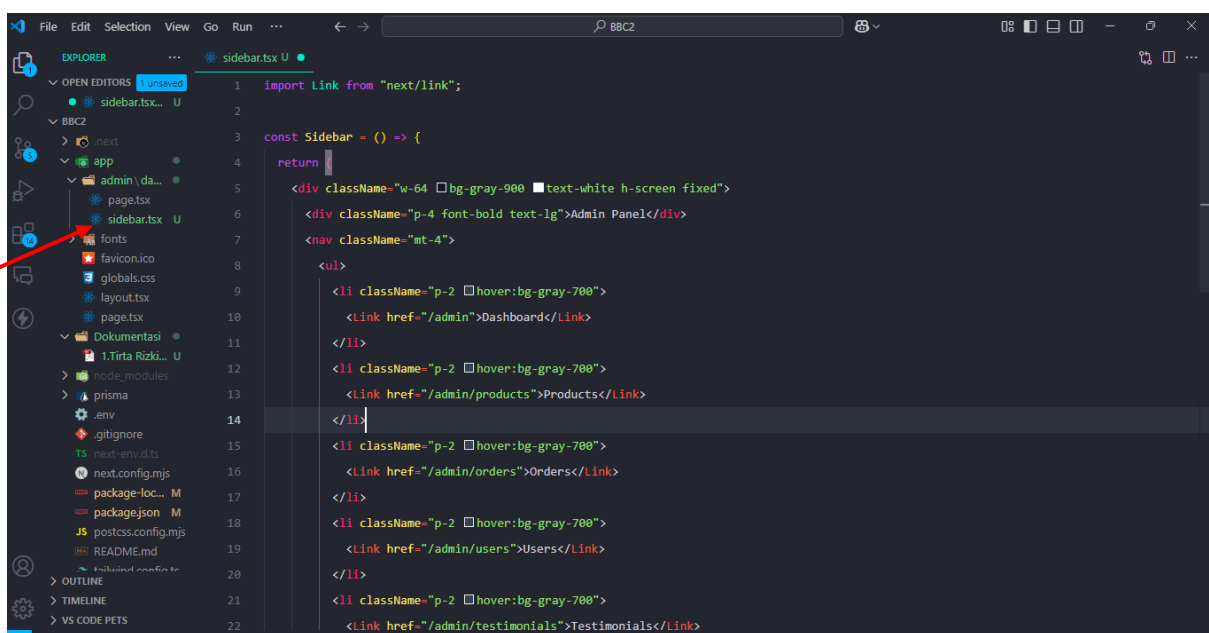
NPM :22312099

### 1. Memindahkan folder dokumentasi

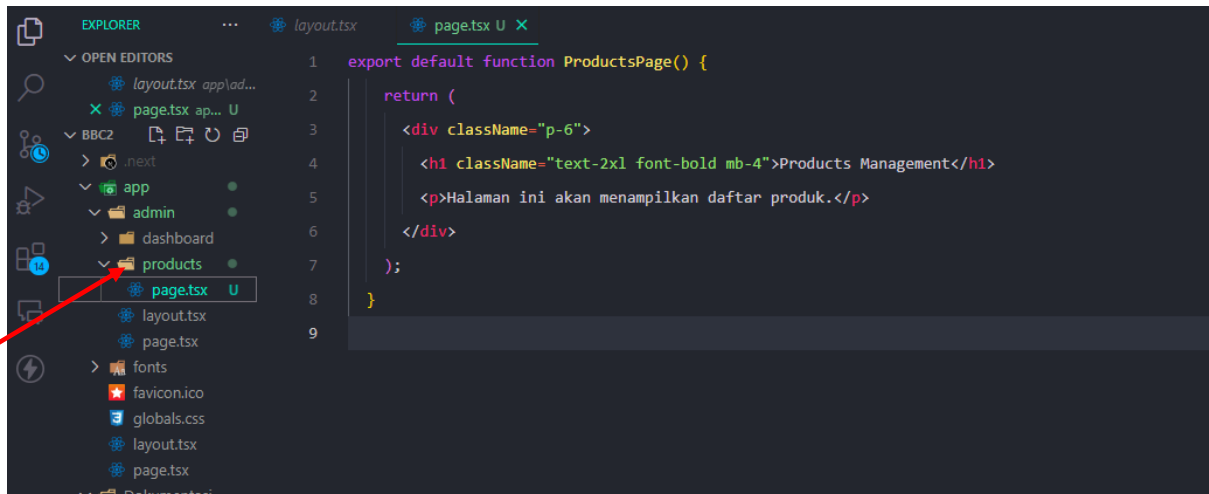


Keterangan: saya memindahkan folder dokumentasi yang sebelumnya ada kesalahan peletakkan didalam folder app, yang seharusnya sejajar/diluar dari app.

2. Saya membuat file baru pada folder admin yaitu sidebar.tsx, file ini adalah komponen navigasi utama di admin panel yang mempermudah pengguna berpindah antar halaman seperti Dashboard, Products, dan Orders yang nantinya akan saya buat. Komponen ini dibuat agar dapat digunakan ulang di semua halaman, menjaga konsistensi tampilan, dan mempermudah pengelolaan kode. Menggunakan Link dari Next.js untuk navigasi antar halaman.



### 3. Membuat Halaman Dasar untuk Products

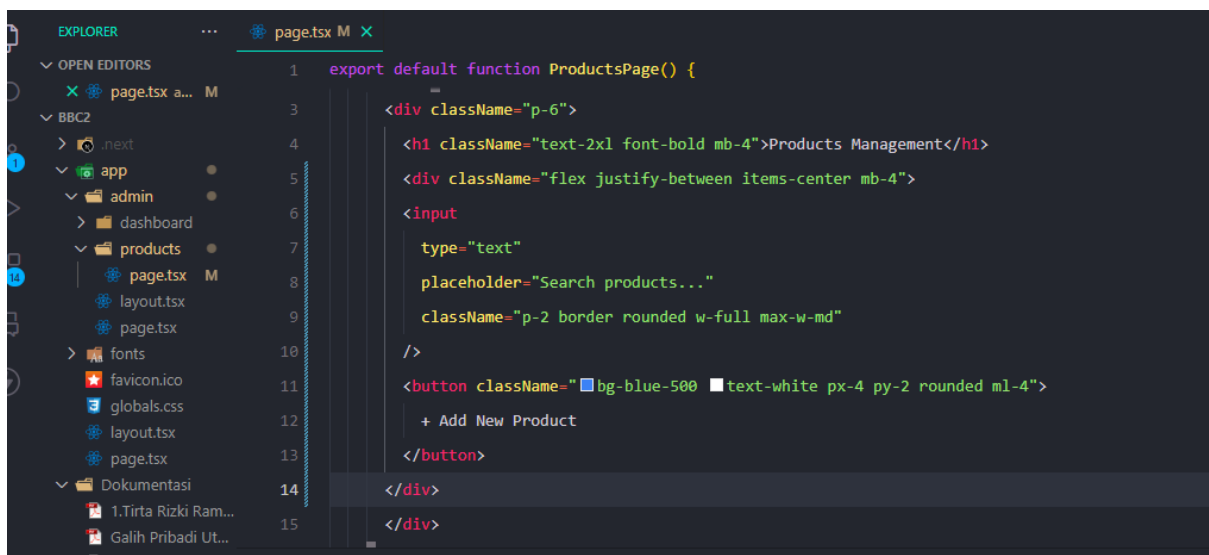


```
export default function ProductsPage() {
  return (
    <div className="p-6">
      <h1 className="text-2xl font-bold mb-4">Products Management</h1>
      <p>Halaman ini akan menampilkan daftar produk.</p>
    </div>
  );
}
```

Keterangan:

- Saya membuat folder baru dalam admin yaitu folder product yang didalamnya terdapat file page.tsx
- File ini adalah halaman utama untuk Products. Untuk sementara, hanya ada judul dan teks placeholder.

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



```
export default function ProductsPage() {
  return (
    <div className="p-6">
      <h1 className="text-2xl font-bold mb-4">Products Management</h1>
      <div className="flex justify-between items-center mb-4">
        <input
          type="text"
          placeholder="Search products..."
          className="p-2 border rounded w-full max-w-md"
        />
        <button
          className="bg-blue-500 text-white px-4 py-2 rounded ml-4"
        >
          + Add New Product
        </button>
      </div>
    </div>
  );
}
```

Keterangan:

- Bagian ini menambahkan input pencarian produk dan tombol untuk menambah produk.

## 5. Menambahkan Filter Dropdown

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 'app', 'admin', 'products', 'fonts', 'Dokumentasi', and 'node\_modules'. The code editor shows the 'page.tsx' file with a React component named 'ProductsPage'. The component uses a 'div' with 'className="grid grid-cols-3 gap-4 mb-4"' and contains two 'select' elements. The first 'select' has options 'All Brands', 'Nike', and 'Adidas'. The second 'select' has options 'All Conditions', 'New', and 'Used'. The third 'select' is partially visible and has options 'All Status', 'Active', and 'Inactive'.

Keterangan:

- Penambahan dropdown untuk memfilter data produk, pada file `app/admin/product/page.tsx`

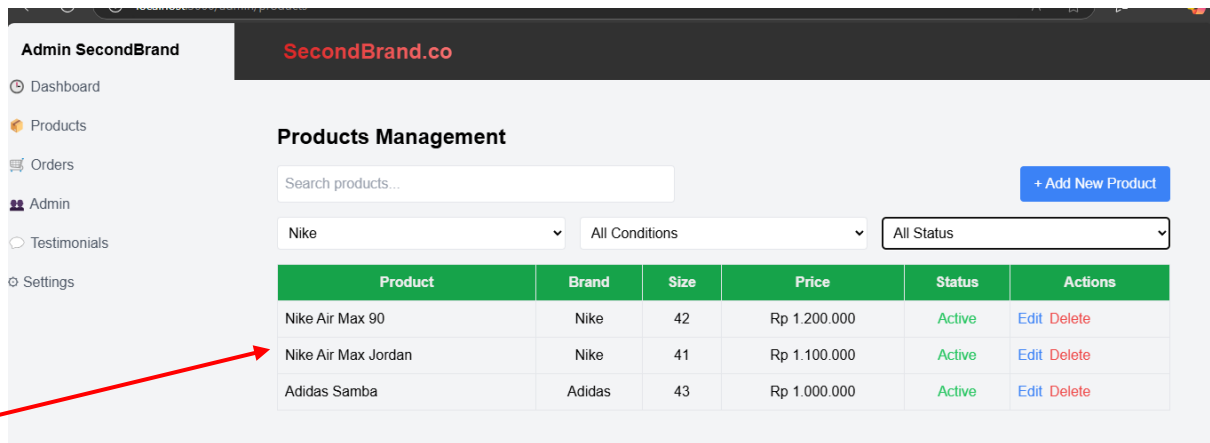
## 6. Menambahkan Tabel Produk

The screenshot shows the VS Code interface. The file explorer on the left lists the project structure, including folders like 'admin', 'products', 'testimonials', 'fonts', and 'Dokumentasi'. The 'products' folder is expanded, showing 'page.tsx' as the active file. The main editor displays the content of 'page.tsx', which is a React component for a product list. It uses a table to display product data, with columns for Product, Brand, Size, Price, Status, and Actions. The table is styled with a gray background and white text. The 'Status' column shows 'Active' in green text. The 'Actions' column has an 'Edit' button in blue text.

Keterangan:

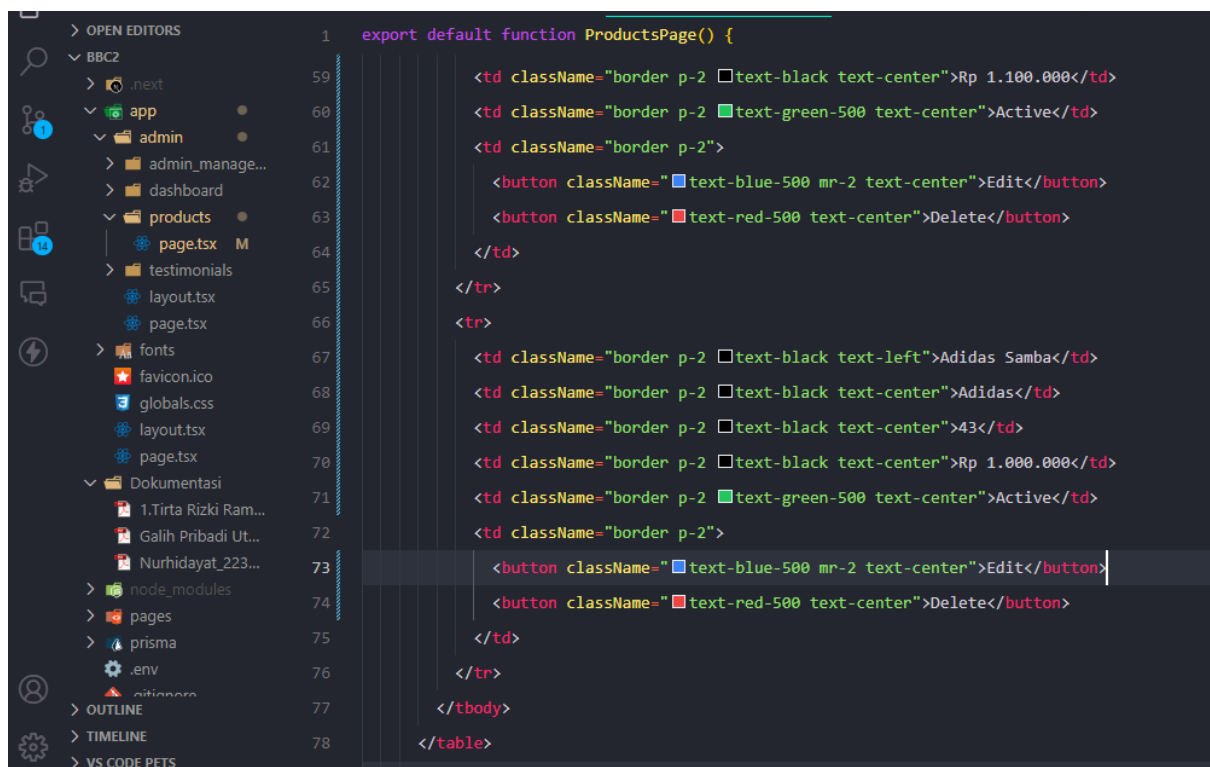
- menambahkan tabel produk dengan data dummy

## 7. Menambahkan data produk

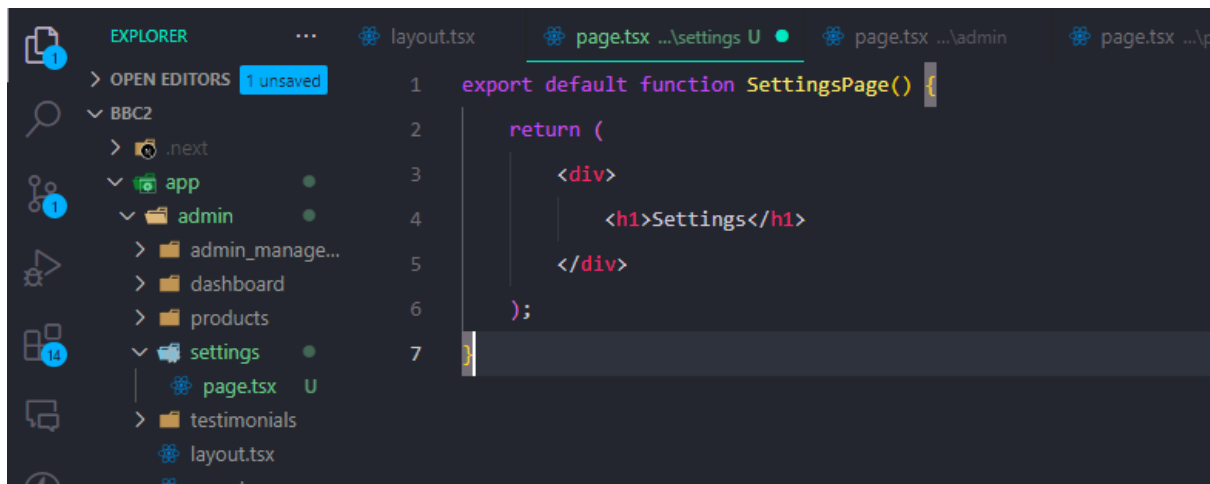


Keterangan:

- Menambahkan data pada product, tampilannya seperti gambar diatas
- Dan kodingannya seperti dibawah ini



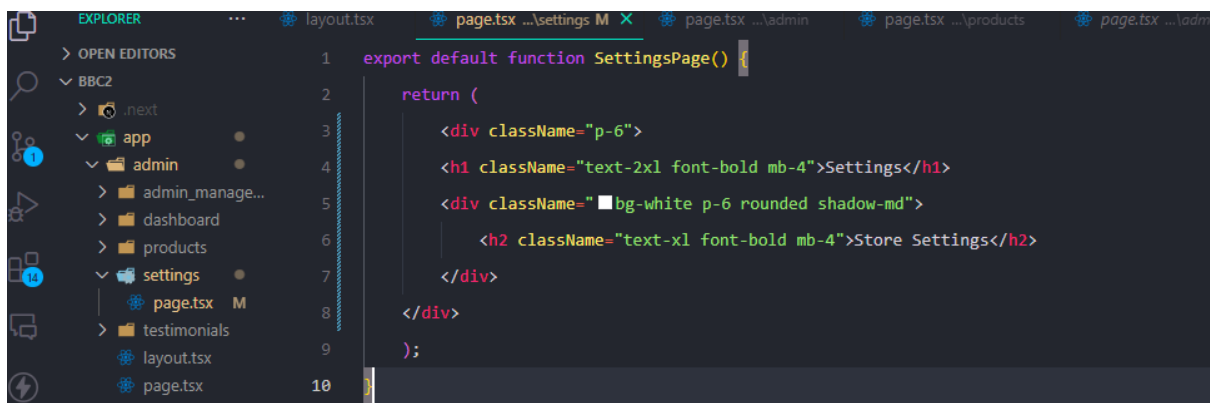
## 9. Membuat folder baru pada admin yaitu folder settings



Keterangan :

- Ini adalah langkah awal untuk membuat halaman **Settings**.
- Dan pada page ini nantinya akan digunakan untuk membuat tampilan Settings

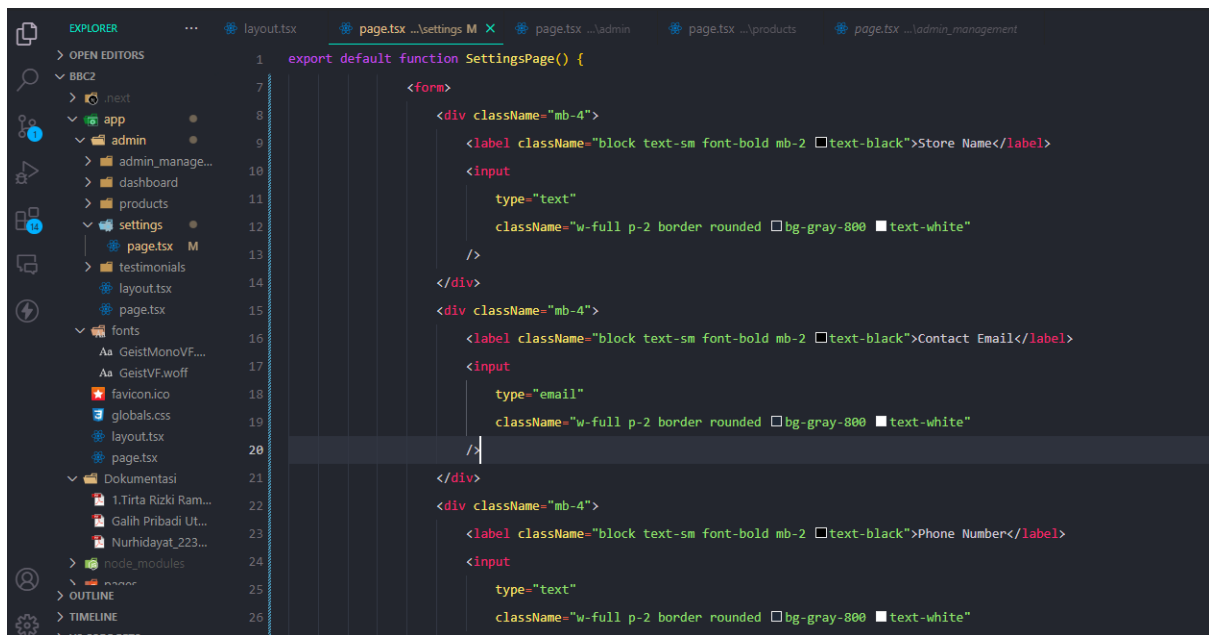
## 10. Struktur Dasar dan Header pada Settings



Keterangan:

- Menambahkan elemen dasar seperti judul halaman dan bagian header **Store Settings**.

## 11. Menambahkan Form Input di Setting Page

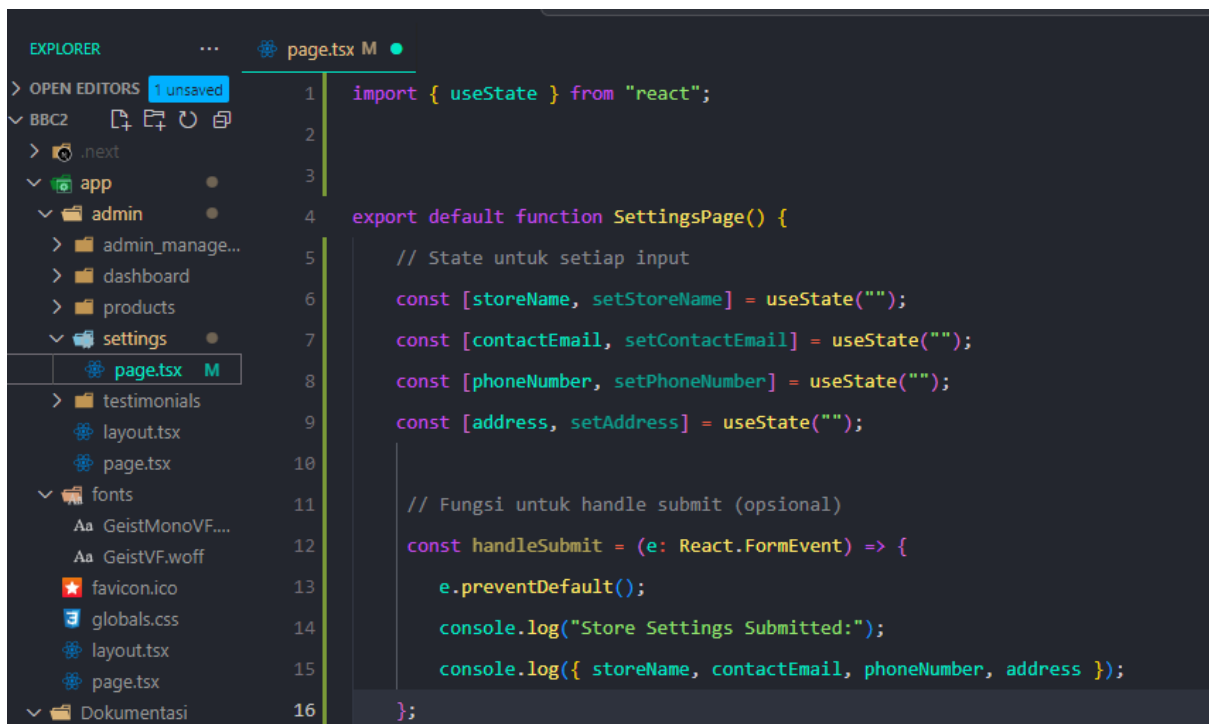


```
1 export default function SettingsPage() {
2
3   <form>
4     <div className="mb-4">
5       <label className="block text-sm font-bold mb-2 text-black">Store Name</label>
6       <input
7         type="text"
8         className="w-full p-2 border rounded bg-gray-800 text-white"
9       />
10    </div>
11    <div className="mb-4">
12      <label className="block text-sm font-bold mb-2 text-black">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 text-black">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 text-black">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 :

- Menambahkan empat input field (Store Name, Contact Email, Phone Number, dan Address).
- Semua input memiliki desain sesuai tema menggunakan className.

## 12. 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.

### 13. Memanggil State pada masing-masing Input

```

export default function SettingsPage() {
  <div className="bg-white p-6 rounded shadow-md">
    <h2 className="text-xl font-bold mb-4 text-black">Store Settings</h2>
    <form onSubmit={handleSubmit}>
      /* input store name */
      <div className="mb-4">
        <label className="block text-sm font-bold mb-2 text-black">Store Name</label>
        <input
          type="text"
          value={storeName}
          onChange={(e) => setStoreName(e.target.value)}
          className="w-full p-2 border rounded bg-gray-800 text-white"
        />
      </div>
      /* Input Contact Email */
      <div className="mb-4">
        <label className="block text-sm font-bold mb-2 text-black">Contact Email</label>
        <input
          type="email"
          value={contactEmail}
          onChange={(e) => setContactEmail(e.target.value)}
        />
      </div>
    </form>
  </div>
}

```

Keterangan :

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