

NAMA : NURHIDAYAT

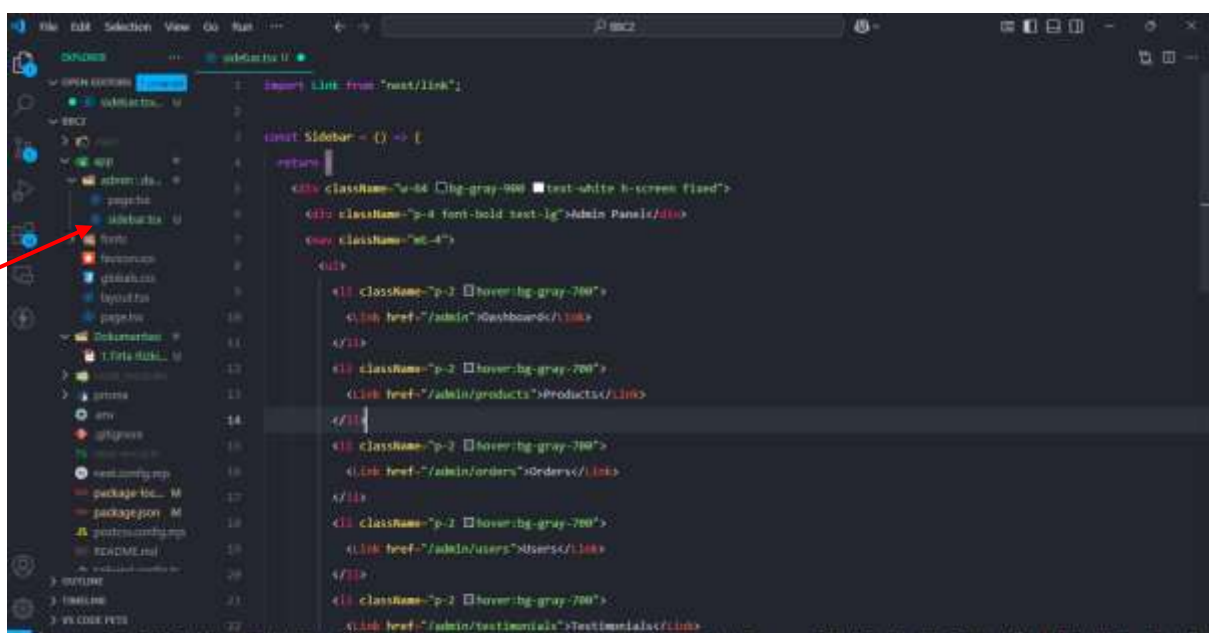
NPM :22312099

## 1. Memindahkan folder dokumentasi

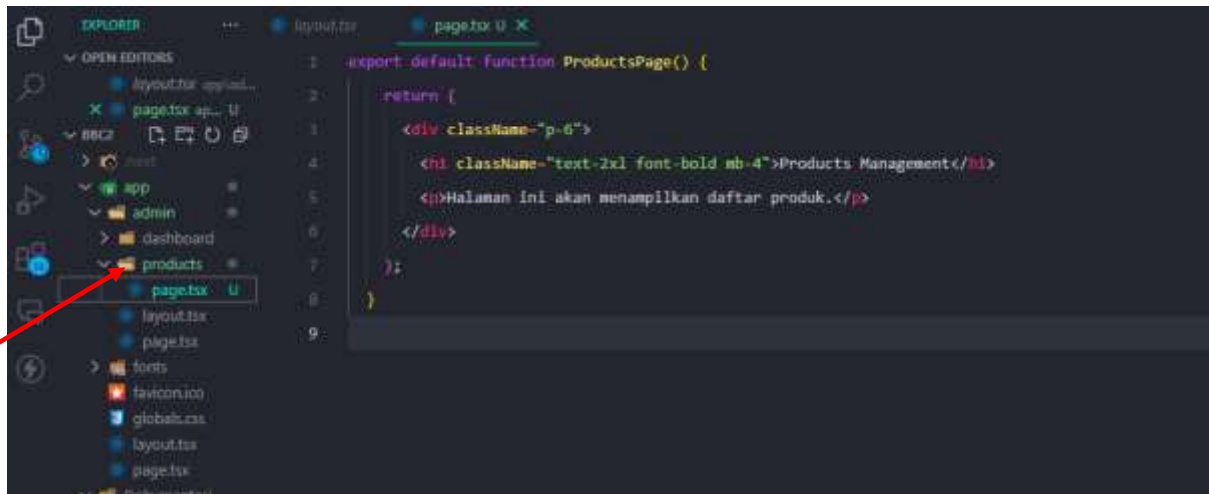


Keterangan: saya memindahkan folder dokumentasi yang sebelumnya ada kesalahan peletakkan didalam folder app, yang seharusnya sejajar/diluar dari app.

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.



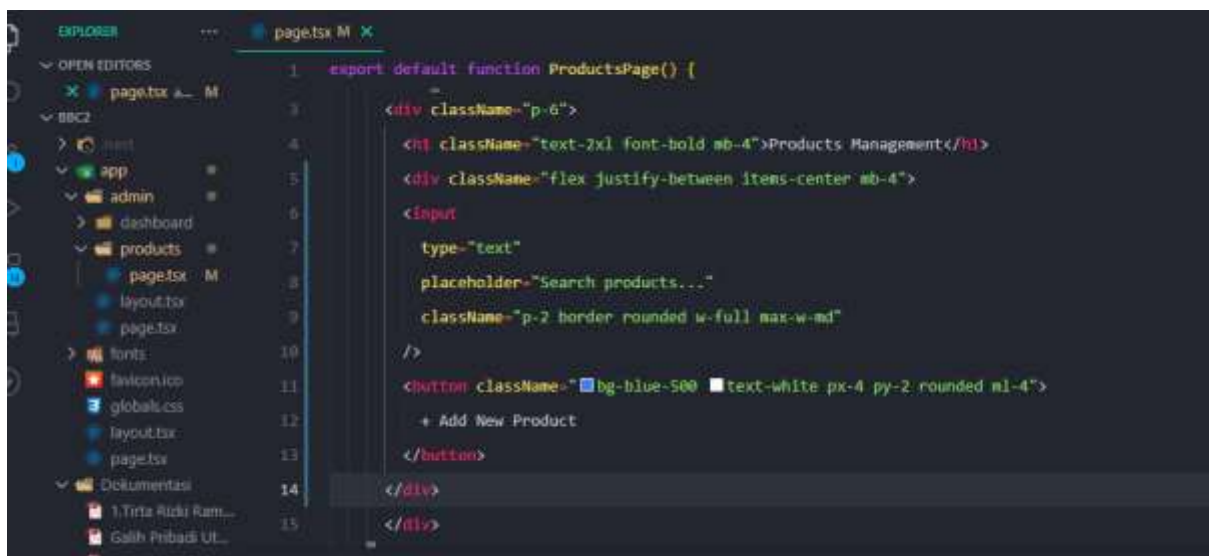
### 3. Membuat Halaman Dasar untuk Products



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.

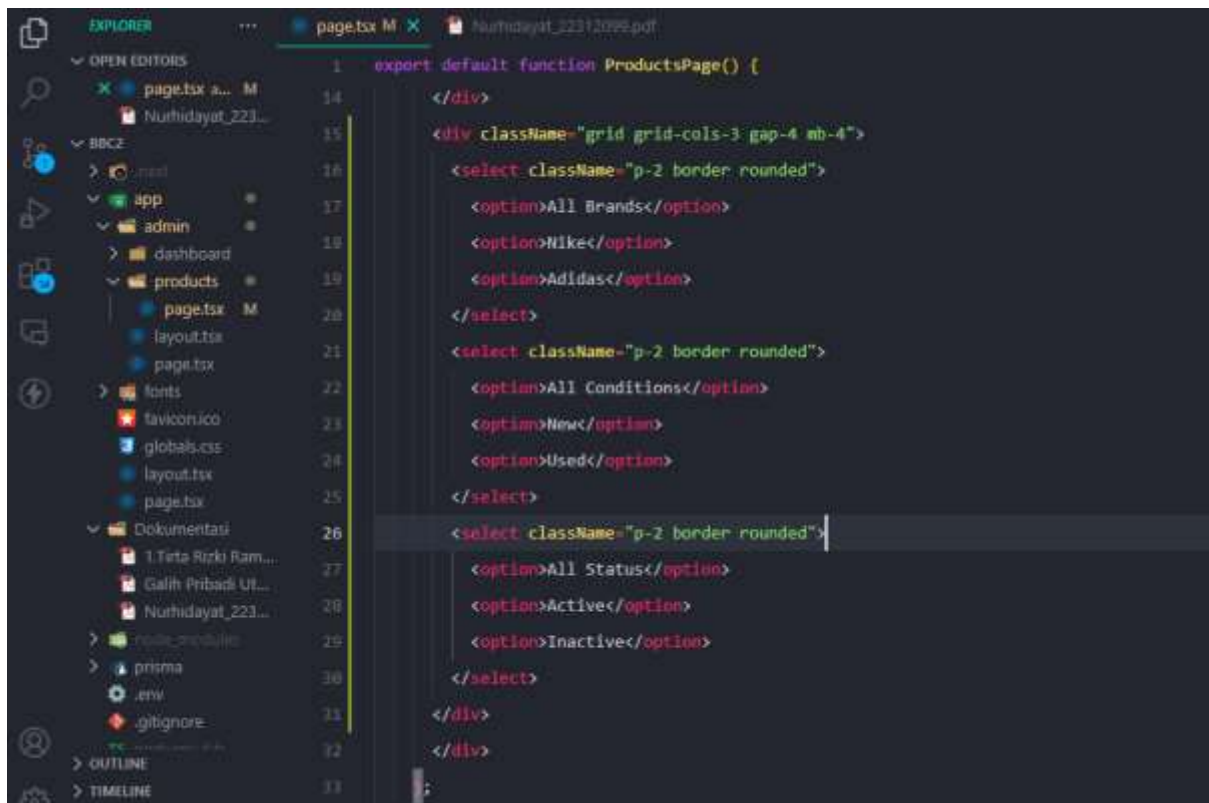
### 4. Menambahkan “Add New Product” dan form pencarian



Keterangan:

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

## 5. Menambahkan Filter Dropdown




The screenshot shows the VS Code editor with the Explorer sidebar on the left and the Editor window on the right. The Explorer sidebar shows the project structure with folders like 'app', 'admin', 'products', 'layout', 'font', 'fonticon', 'global', 'layout', 'page', 'Dokumentasi', 'node\_modules', 'prisma', and '.env'. The Editor window shows the code for 'page.tsx' in the 'products' folder. The code defines a function 'ProductsPage()' which returns a JSX element. It includes two dropdown menus: one for 'All Brands' (Nike, Adidas) and one for 'All Conditions' (New, Used). The code also includes a third dropdown menu for 'All Status' (Active, Inactive).

```
export default function ProductsPage() {  
  </div>  
  <div className="grid grid-cols-3 gap-4 mb-4">  
    <select className="p-2 border rounded">  
      <option>All Brands</option>  
      <option>Nike</option>  
      <option>Adidas</option>  
    </select>  
    <select className="p-2 border rounded">  
      <option>All Conditions</option>  
      <option>New</option>  
      <option>Used</option>  
    </select>  
    <select className="p-2 border rounded">  
      <option>All Status</option>  
      <option>Active</option>  
      <option>Inactive</option>  
    </select>  
  </div>  
</div>  
}
```

Keterangan:

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

## 6. Menambahkan Tabel Produk



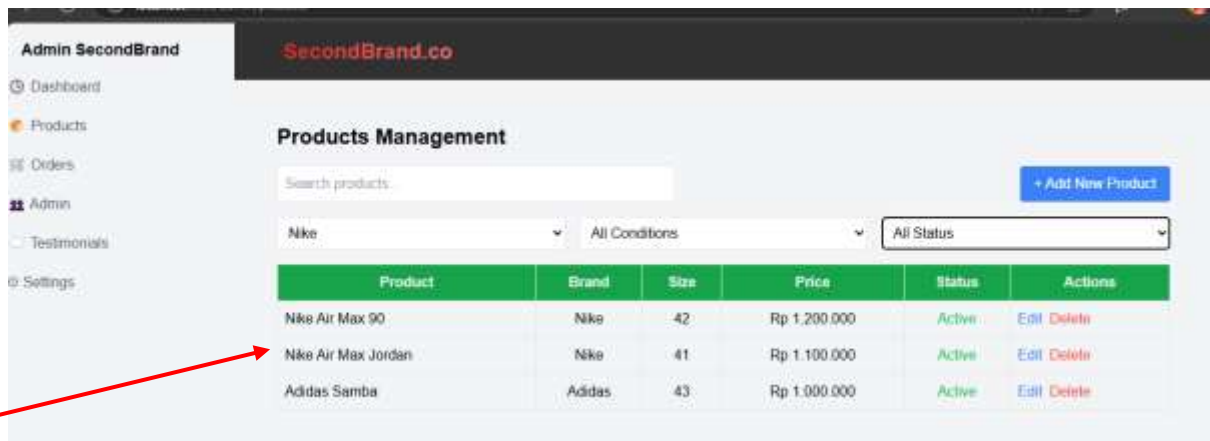
The screenshot shows the VS Code editor with the Explorer sidebar on the left and the Editor window on the right. The Explorer sidebar shows the project structure with folders like 'app', 'admin', 'products', 'layout', 'font', 'fonticon', 'global', 'layout', 'page', 'Dokumentasi', 'node\_modules', 'prisma', and '.env'. The Editor window shows the code for 'page.tsx' in the 'products' folder. The code defines a function 'ProductsPage()' which returns a JSX element. It includes a table with columns: Product, Brand, Size, Price, Status, and Actions. The table contains data for Nike Air Max 90, Nike, and a size of 42. The price is Rp 1.200.000. The status is Active. The actions are Edit and Delete.

```
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>  
    </tbody>  
  </table>  
}
```

Keterangan:

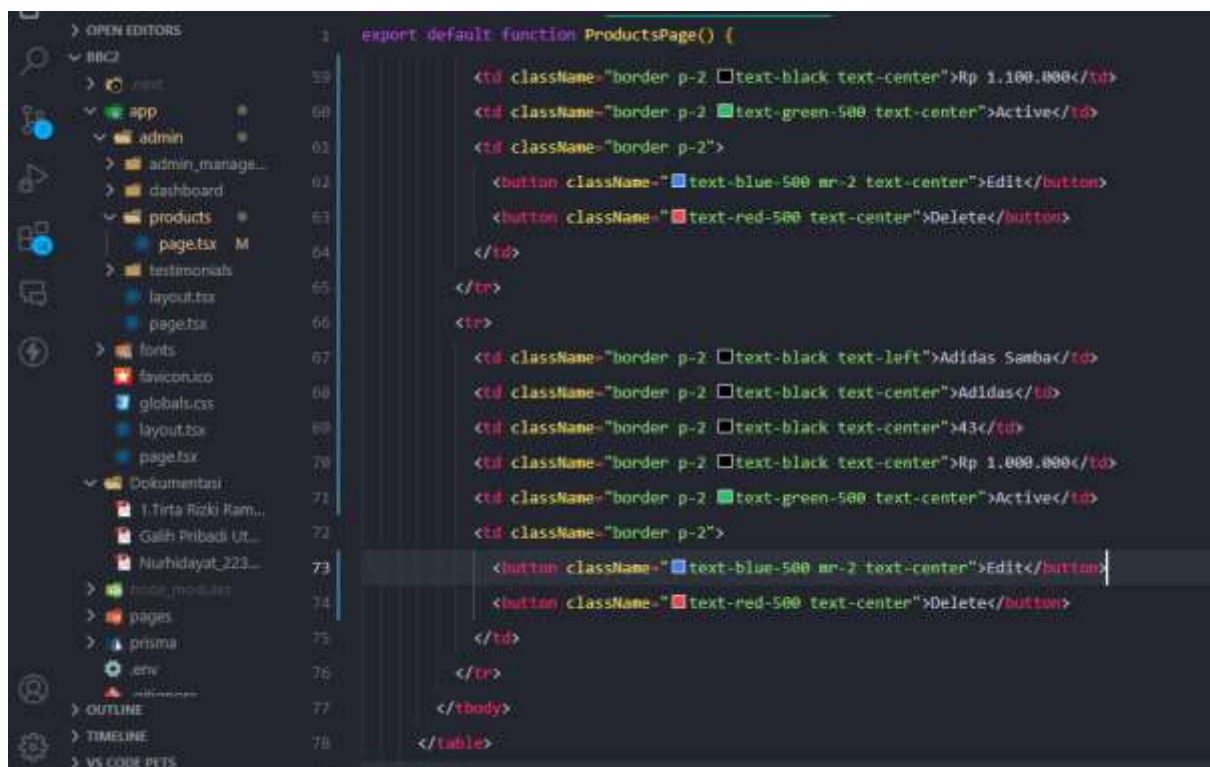
- menambahkan tabel produk dengan data dummy

## 7. Menambahkan data produk

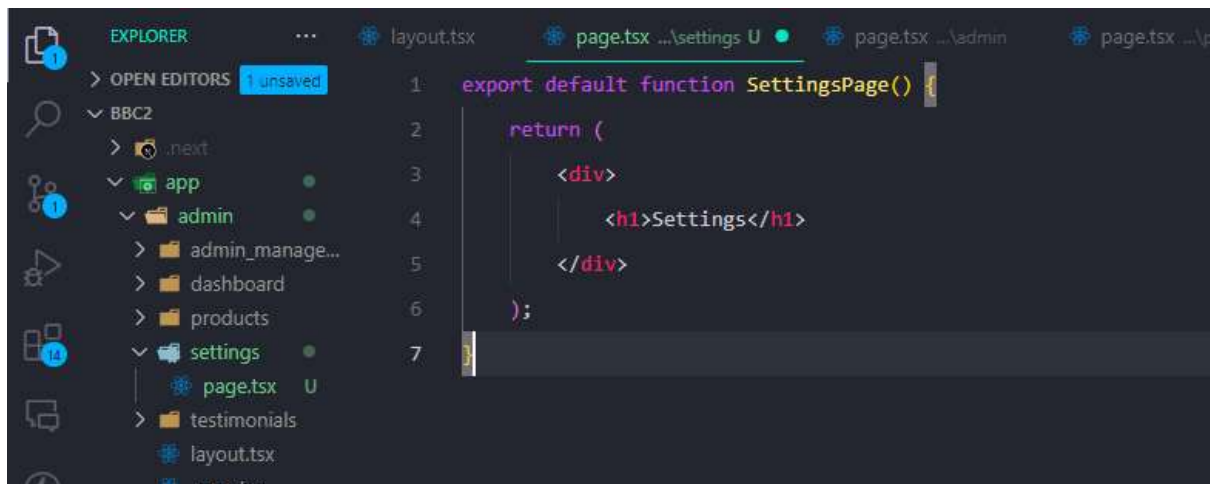


Keterangan:

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



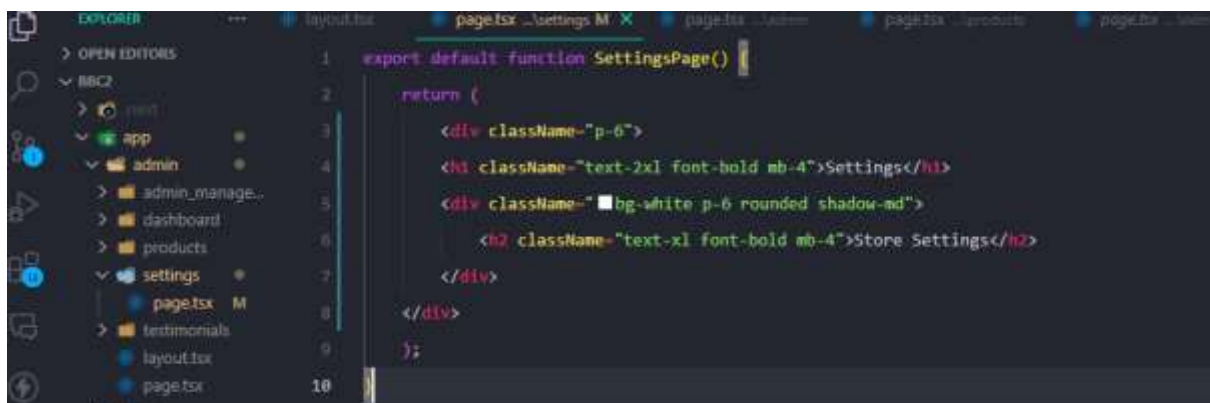
## 9. Membuat folder baru pada admin yaitu folder settings



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

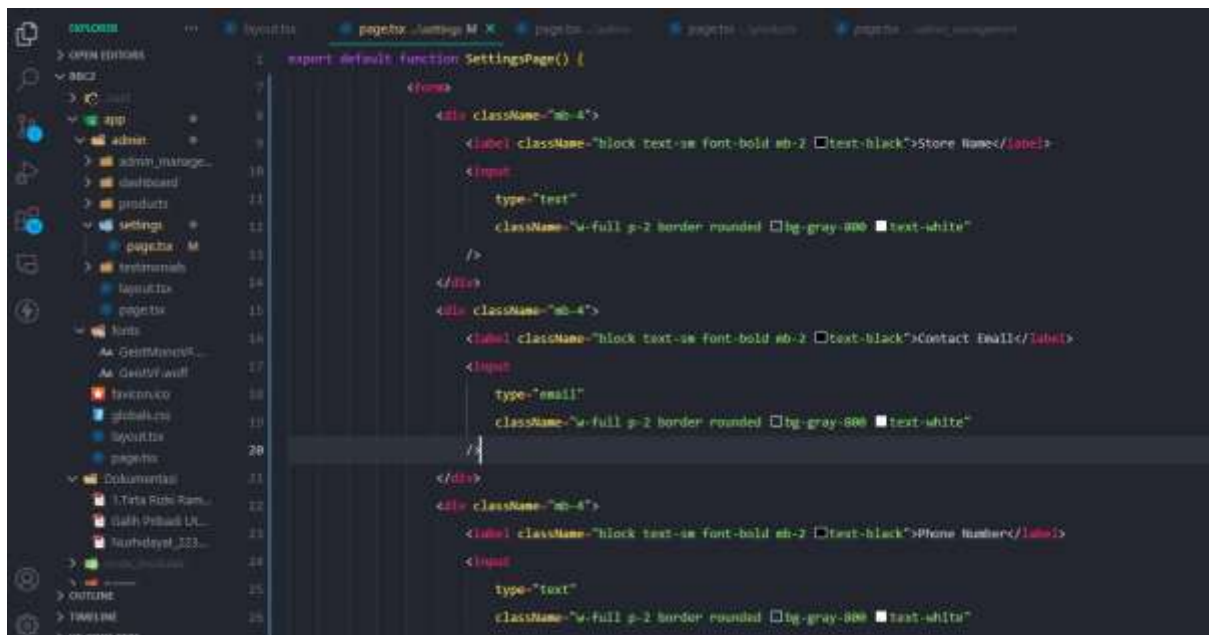


Keterangan:

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



## 11. Menambahkan Form Input di Setting Page

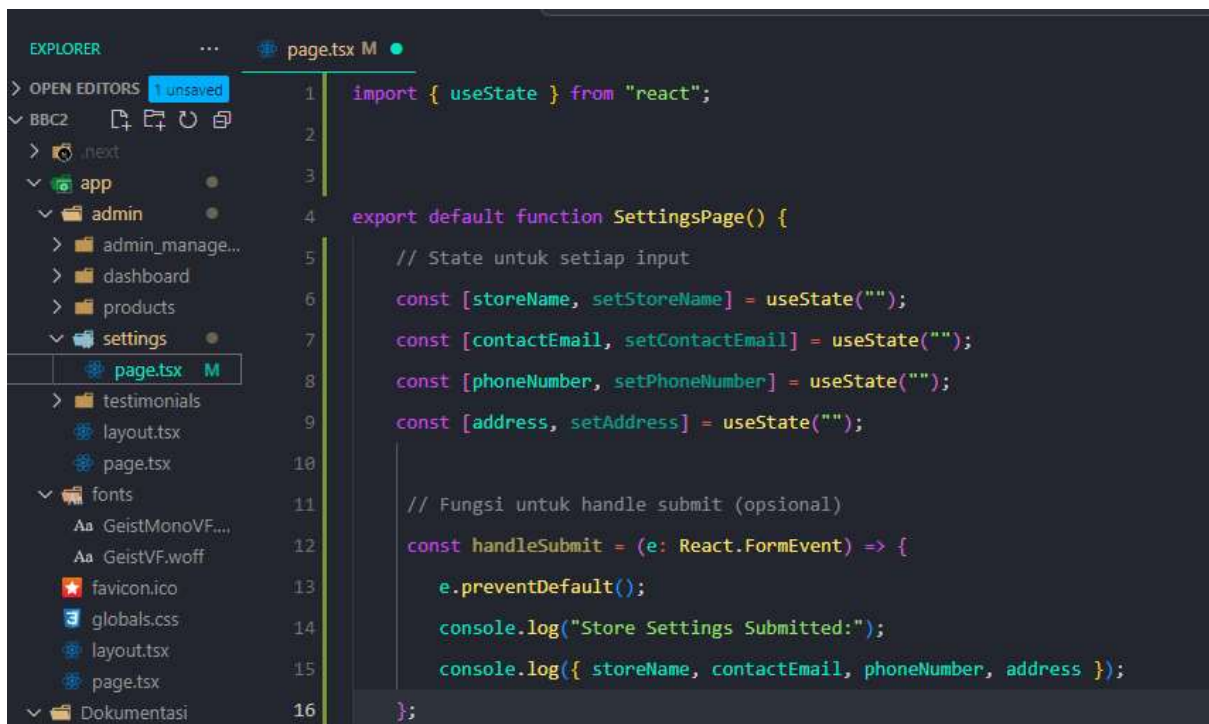


```
export default function SettingsPage() {  
  <form>  
    <div className="mb-4">  
      <label className="block text-sm font-bold mb-2 text-black">Store Name</label>  
      <input  
        type="text"  
        className="w-full p-2 border rounded bg-gray-800 text-white"  
      />  
    </div>  
    <div className="mb-4">  
      <label className="block text-sm font-bold mb-2 text-black">Contact Email</label>  
      <input  
        type="email"  
        className="w-full p-2 border rounded bg-gray-800 text-white"  
      />  
    </div>  
    <div className="mb-4">  
      <label className="block text-sm font-bold mb-2 text-black">Phone Number</label>  
      <input  
        type="text"  
        className="w-full p-2 border rounded bg-gray-800 text-white"  
      />  
    </div>  
  </form>  
}
```

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



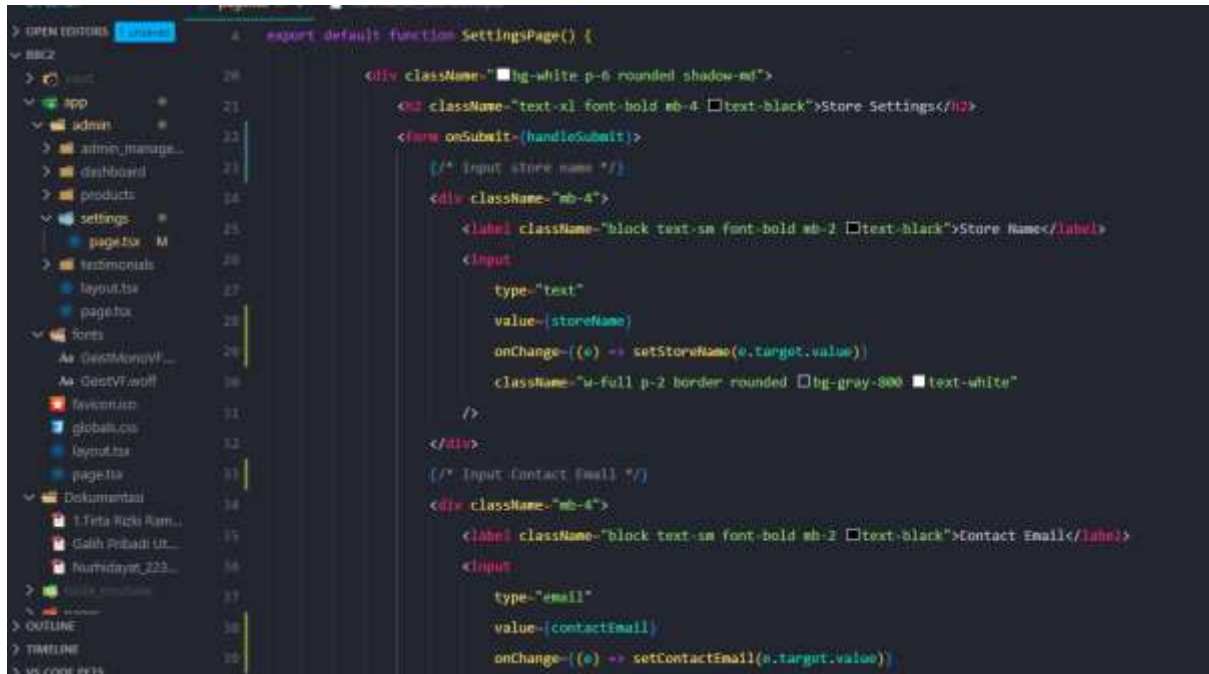
```
import { useState } from "react";  
  
export default function SettingsPage() {  
  // State untuk setiap input  
  const [storeName, setStoreName] = useState("");  
  const [contactEmail, setContactEmail] = useState("");  
  const [phoneNumber, setPhoneNumber] = useState("");  
  const [address, setAddress] = useState("");  
  
  // Fungsi untuk handle submit (opsional)  
  const handleSubmit = (e: React.FormEvent) => {  
    e.preventDefault();  
    console.log("Store Settings Submitted:");  
    console.log({ storeName, contactEmail, phoneNumber, address });  
  };  
}
```

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() {
  // ... (previous code) ...

  <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).

### 14. Menambahkan Submit Button



```

export default function SettingsPage() {
  // ... (previous code) ...

  <div>
    <button
      type="submit"
      className="bg-blue-500 text-white px-4 py-2 rounded"
    >
      Save Settings
    </button>
  </div>
</form>
</div>

```

Keterangan :

- Saya menambahkan submit button / tombol submit untuk menyimpan perubahan

## 15. Menambahkan Notifications Settings

```
0  export default function SettingsPage() {
66      /* Notification Settings */
67      <div className="bg-white rounded-lg shadow p-6">
68          <div className="flex items-center gap-2 mb-4">
69              <div className="h-5 w-5 text-primary text-black" />
70              <div className="text-xl font-bold text-gray-800">Notification Settings</div>
71          </div>
72          <div className="space-y-4">
73              <div className="flex items-center justify-between">
74                  <div>
75                      <p className="font-medium text-gray-800">Order Notifications</p>
76                      <p className="text-sm text-gray-600">Receive notifications for new orders</p>
77                  </div>
78                  <input type="checkbox" className="toggle toggle-primary" defaultChecked />
79              </div>
80              <div className="flex items-center justify-between">
81                  <div>
82                      <p className="font-medium text-gray-800">Review Notifications</p>
83                      <p className="text-sm text-gray-600">Receive notifications for new reviews</p>
84                  </div>
85                  <input type="checkbox" className="toggle toggle-primary" defaultChecked />
86              </div>
87          </div>
88      </div>
89  }
90  // ...
```

Keterangan :

- Menambahkan notification settings pada page 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>

```

Keterangan:

- Field untuk **Current Password**, **New Password**, dan **Confirm Password**.

## Perbaikan pada page settings

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

```

export default function SettingsPage() {
  const [address, setAddress] = useState("");

  // State untuk Notification Settings
  const [orderNotifications, setOrderNotifications] = useState(true);
  const [reviewNotifications, setReviewNotifications] = useState(true);

  // State untuk Security Settings
  const [currentPassword, setCurrentPassword] = useState("");
  const [newPassword, setNewPassword] = useState("");
  const [confirmPassword, setConfirmPassword] = useState("");

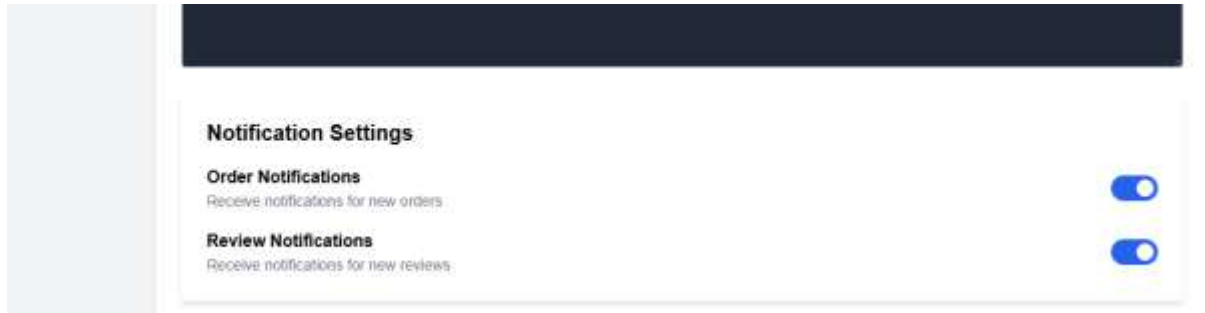
  // Fungsi untuk handle submit
  const handleSubmit = (e: React.FormEvent) => {
    e.preventDefault();
    console.log("Form Submitted:", {
      storeSettings: { storeName, contactEmail, phoneNumber, address },
      notificationSettings: { orderNotifications, reviewNotifications },
      securitySettings: { currentPassword, newPassword, confirmPassword },
    });
  };
}

```

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

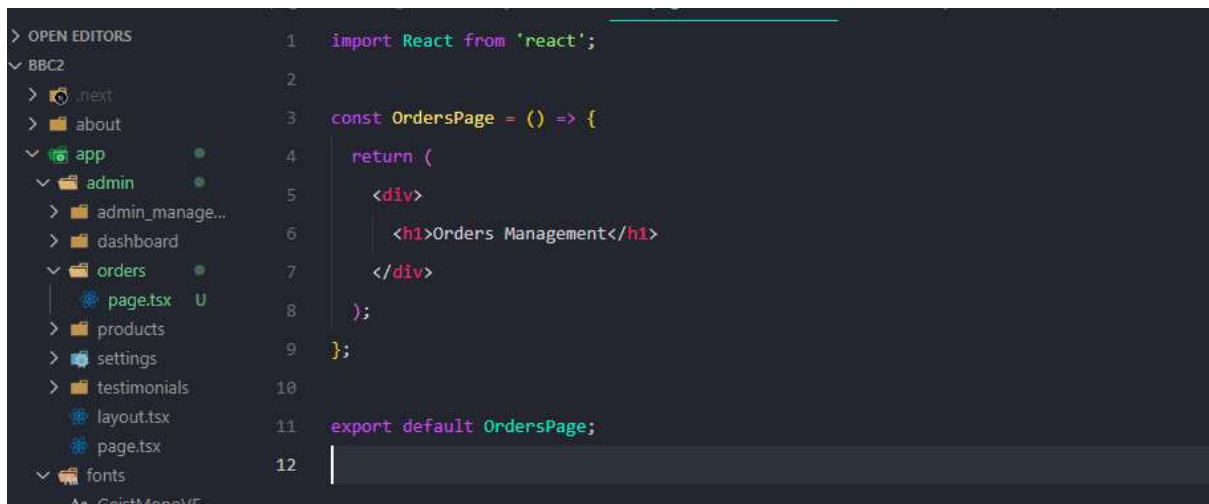
#### 19. Perbaikan pada Security Settings



Keterangan:

- Ini adalah tampilan Security Settings setelah diperbaiki

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



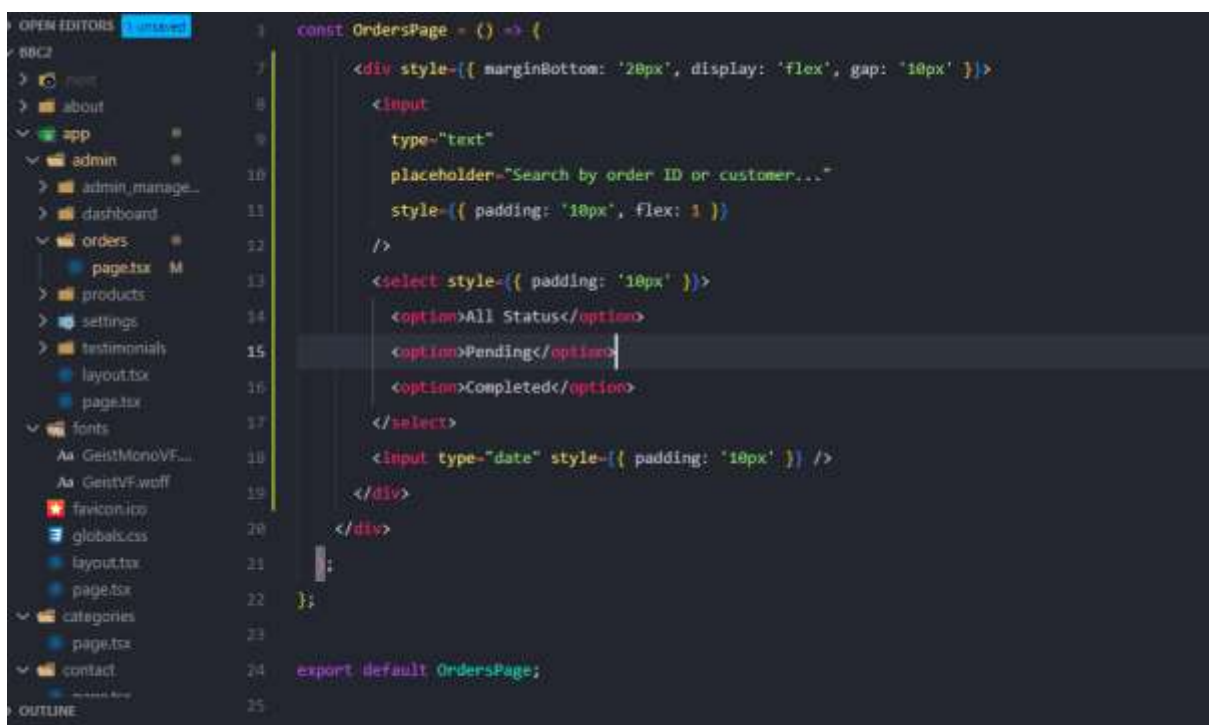
The screenshot shows the VS Code interface with a file explorer on the left and a code editor on the right. The file explorer shows a project structure with folders like .next, about, app, admin, dashboard, orders, products, settings, testimonials, layout.tsx, page.tsx, and fonts. The 'orders' folder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

Keterangan:

- Membuat folder baru pada folder admin yaitu orders
- Menambahkan file page.tsx didalam folder tersebut
- Melakukan setup awal pada page.tsx dengan komponen React dasar

21. Menambahkan struktur halaman layout pada page orders



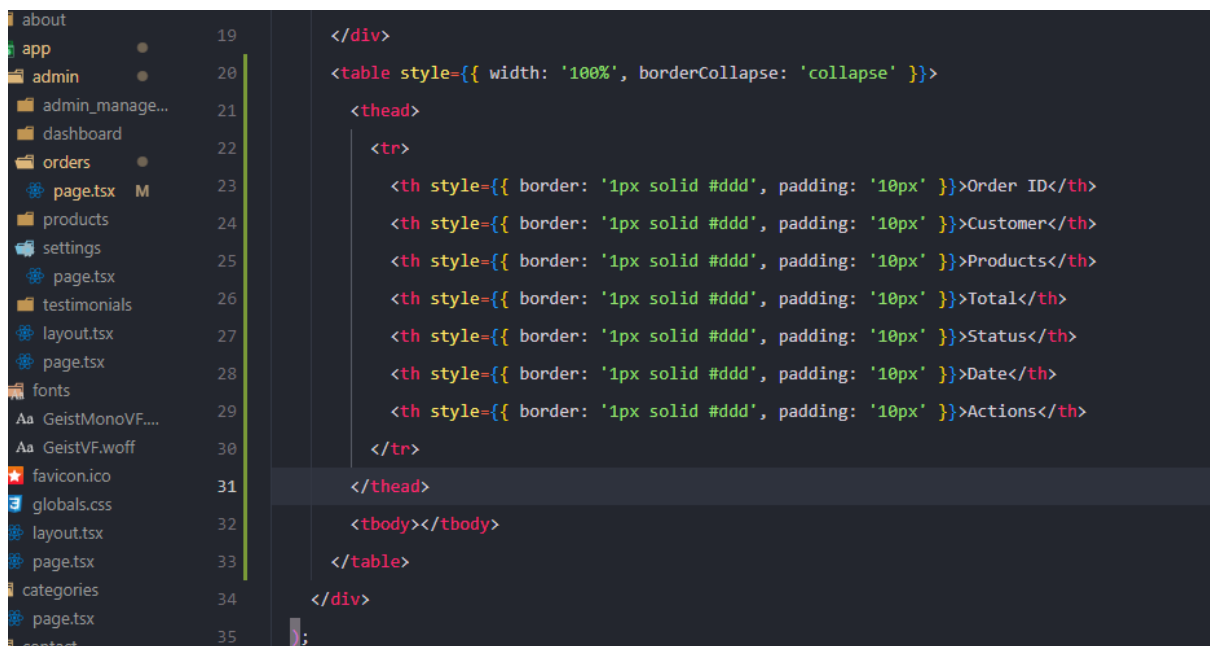
The screenshot shows the VS Code interface with the same file explorer and code editor. The code editor now shows the following code:

```
1 const OrdersPage = () => {
2
3   <div style={{ marginBottom: '20px', display: 'flex', gap: '10px' }}>
4     <input
5       type="text"
6       placeholder="Search by order ID or customer..."
7       style={{ padding: '10px', flex: 1 }}
8     />
9     <select style={{ padding: '10px' }}>
10       <option>All Status</option>
11       <option>Pending</option>
12       <option>Completed</option>
13     </select>
14     <input type="date" style={{ padding: '10px' }} />
15   </div>
16
17   :
18
19   </div>
20 };
21
22 export default OrdersPage;
```

Keterangan :

- Menambahkan input pencarian, dropdown filter status, dan input tanggal di file page.tsx

## 22. Menambahkan Struktur Tabel Kosong pada page orders

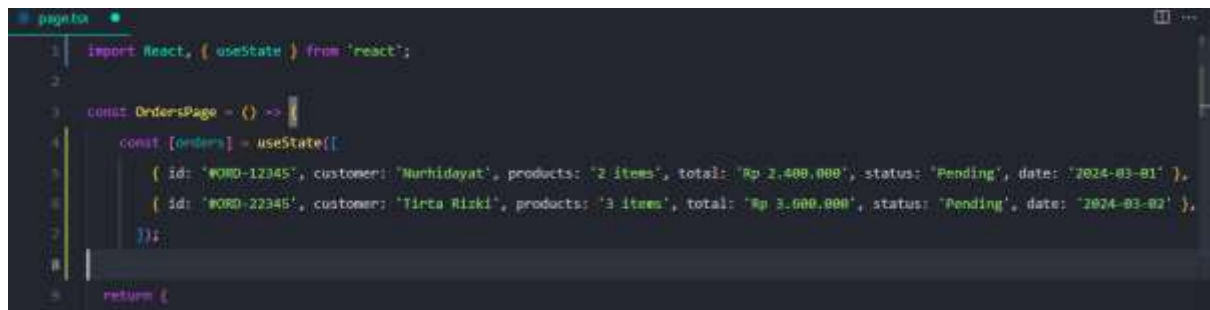


```
19 </div>
20
21 <table style={{ width: '100%', borderCollapse: 'collapse' }}>
22   <thead>
23     <tr>
24       <th style={{ border: '1px solid #ddd', padding: '10px' }}>Order ID</th>
25       <th style={{ border: '1px solid #ddd', padding: '10px' }}>Customer</th>
26       <th style={{ border: '1px solid #ddd', padding: '10px' }}>Products</th>
27       <th style={{ border: '1px solid #ddd', padding: '10px' }}>Total</th>
28       <th style={{ border: '1px solid #ddd', padding: '10px' }}>Status</th>
29       <th style={{ border: '1px solid #ddd', padding: '10px' }}>Date</th>
30       <th style={{ border: '1px solid #ddd', padding: '10px' }}>Actions</th>
31     </tr>
32   </thead>
33   <tbody></tbody>
34 </table>
35 </div>
```

Keterangan:

- Saya menambahkan tabel HTML dasar di bawah elemen filter

## 23. 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 3.000.000', status: 'Pending', date: '2024-03-02' },
7   ]);
8
9   return (
```

Keterangan:

- Menambahkan array data dummy menggunakan useState
- useState digunakan untuk menyimpan data dummy berupa array

## 24. Merender Data Dummy ke Tabel

```

<tbody>
  {orders.map((order) =>
    <tr key={order.id}>
      <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>
  )}
</tbody>

```

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.

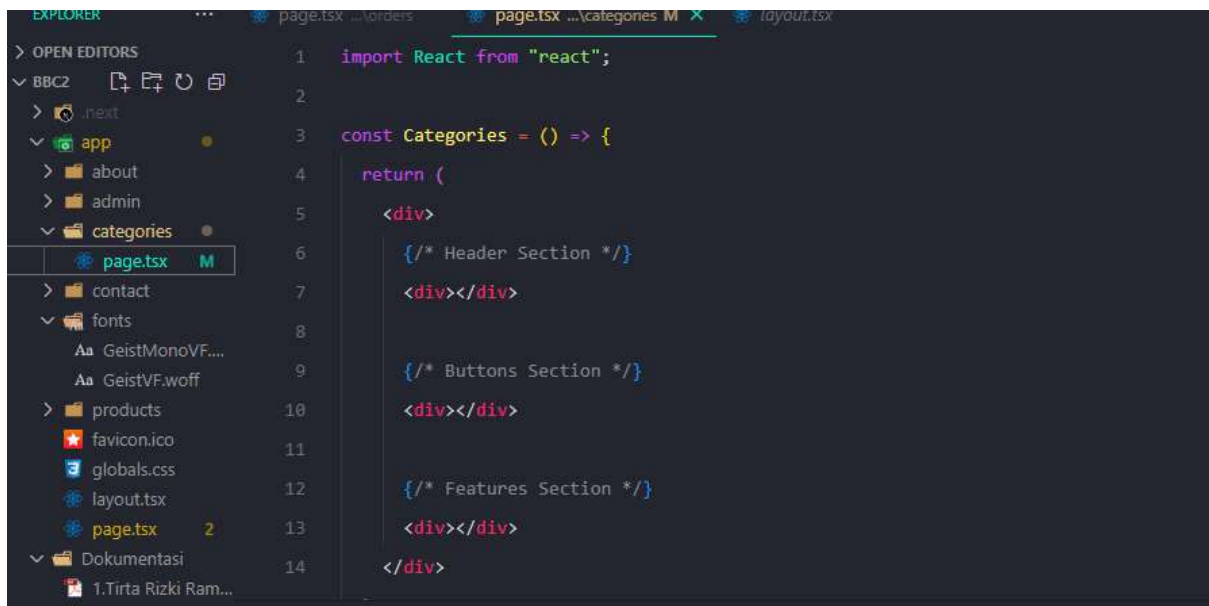
## 25. Menambahkan data dummy pad page orders

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



## 26. Membuat struktur dasar komponen pada page categories

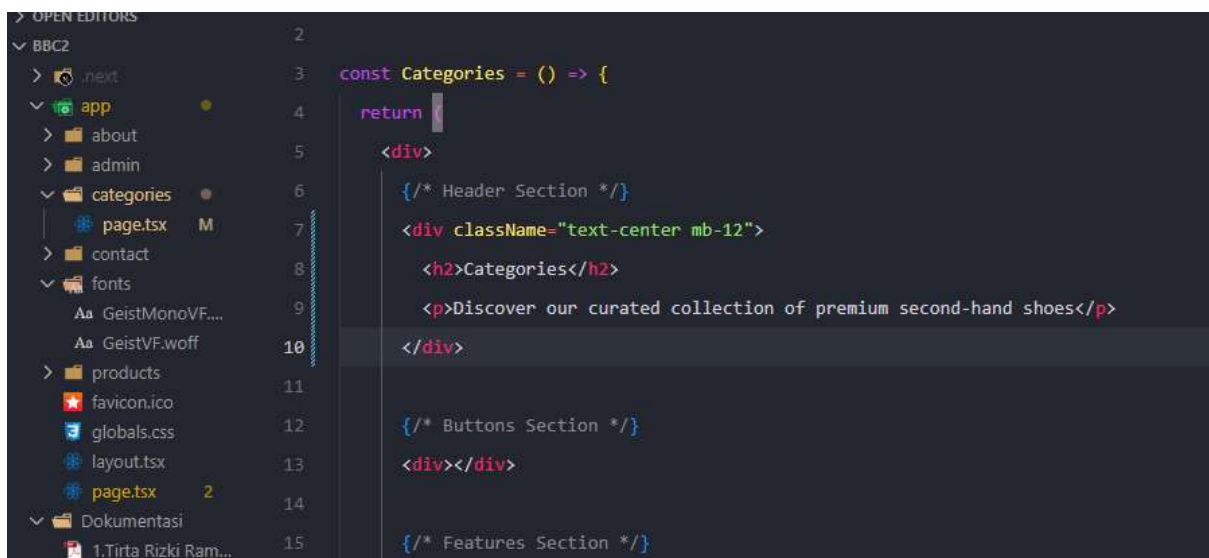


```
1 import React from "react";
2
3 const Categories = () => {
4   return (
5     <div>
6       {/* Header Section */}
7       <div></div>
8
9       {/* Buttons Section */}
10      <div></div>
11
12      {/* Features Section */}
13      <div></div>
14    </div>
15  );
16 }
```

Keterangan:

- Saya membuat struktur komponen awal pada categories. Ini adalah kerangka utama yang akan diisi elemen-elemen pada commit berikutnya.

## 27. Menambahkan Header Section pada page Categories

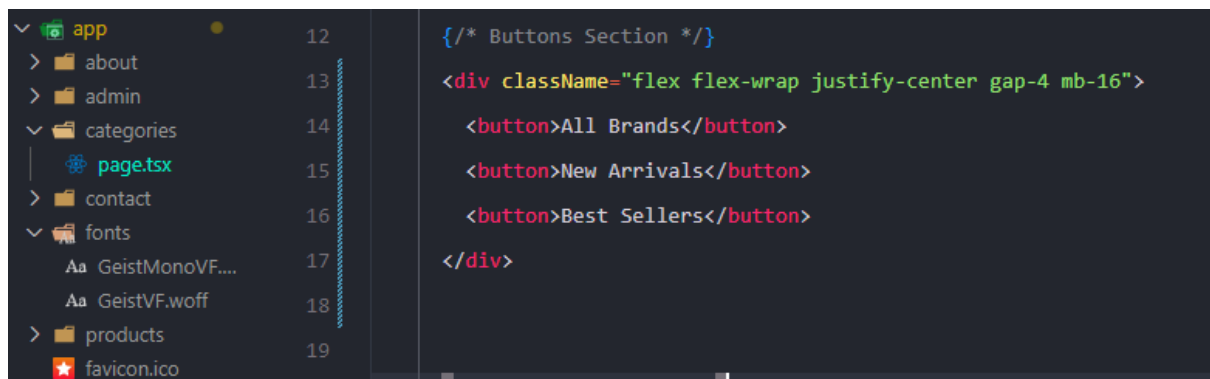


```
2
3 const Categories = () => {
4   return (
5     <div>
6       {/* Header Section */}
7       <div className="text-center mb-12">
8         <h2>Categories</h2>
9         <p>Discover our curated collection of premium second-hand shoes</p>
10      </div>
11
12      {/* Buttons Section */}
13      <div></div>
14
15      {/* Features Section */}
16    </div>
17  );
18 }
```

Keterangan:

- Menambahkan bagian judul (header), termasuk teks utama dan deskripsi.

## 28. 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:

- Menambahkan tombol-tombol untuk kategori produk. Tombol masih tanpa styling pada tahap ini.