

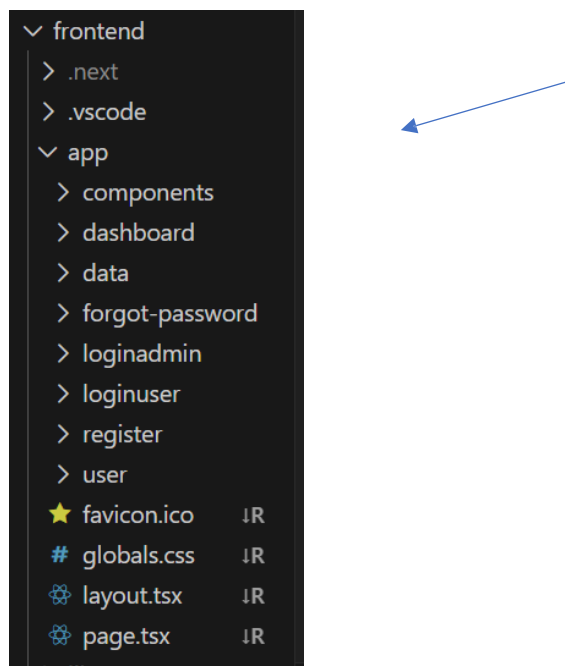
## Laporan dokumentasi proyek

Nama : Nazwa Syakuru

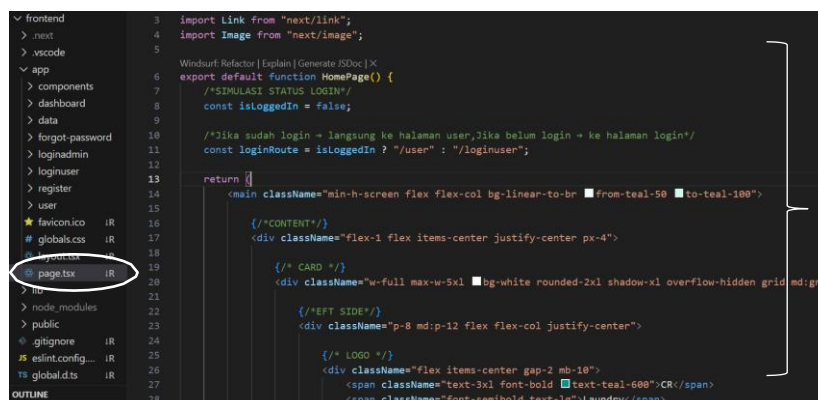
NPM : 23313010

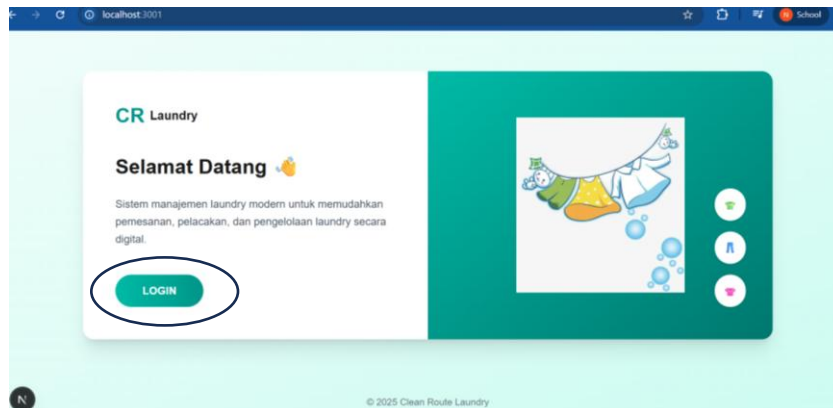
Kelas : TI 23A

1. Menginstall frontend ,setelah menginstall ubah port frontend pada file package.json bagian scripts menjadi 3001 guna agar tidak sama dengan backend nya dan membuat struktur folder seperti gambar dibawah ini:

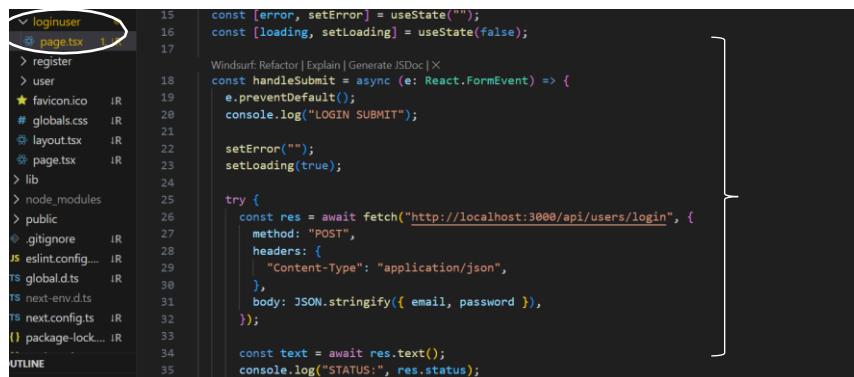


2. Pada file page.tsx di folder app frontend adalah tampilan yang dibuat untuk global atau bisa juga disebut dengan index, di page tersebut berisi halaman utama untuk bisa log in:





- Folder loginuser yang berisi page.tsx adalah jika kita klik button “login” pada index seperti gambar diatas tadi maka akan mengarah pada page login user nya, nanti kita akan langsung diarahkan ke form untuk bisa login sebagai user namun jika belum bisa maka diharuskan untuk daftar terlebih dahulu:



- Berikut adalah jika kita klik “Daftar” tadi pada form login user maka akan mengarah pada page register dan lalu kita bisa untuk mengisi form “Daftar Akun”, setelah sudah mengisi semua nya klik “Daftar”:

```

14 const [error, setError] = useState("");
15 const [loading, setLoading] = useState(false);
16
17 Windurf: Refactor | Explain | Generate JSDoc | X
18 const handleSubmit = async (e: React.FormEvent) => {
19   e.preventDefault();
20   setError("");
21   setLoading(true);
22
23   try {
24     const res = await fetch("http://localhost:3000/api/users/register", {
25       method: "POST",
26       headers: {
27         "Content-Type": "application/json",
28       },
29       body: JSON.stringify({
30         name,
31         email,
32         password,
33       });
34     });
35   }

```

### Daftar Akun

Sudah punya akun? [Login](#)

Setelah register sudah berhasil, kita login kembali menggunakan akun yang sudah dibuat tersebut. Ketika login nya otomatis diarahkan ke dashboard yang sesuai saat kita log in nya yaitu admin atau user.

- Folder loginadmin berisi form untuk admin:

```

3 import { useRouter } from "next/navigation";
4 import { useState } from "react";
5 import { login } from "@lib/auth";
6
7 const API_URL = "http://localhost:3000/api/admin/login";
8
9 export default function AdminLoginPage() {
10   const router = useRouter();
11
12   const [username, setUsername] = useState("");
13   const [password, setPassword] = useState("");
14   const [error, setError] = useState("");
15   const [loading, setLoading] = useState(false);
16
17   Windurf: Refactor | Explain | Generate JSDoc | X
18   const handleLogin = async (e: React.FormEvent) => {
19     e.preventDefault();
20     setError("");
21     setLoading(true);
22
23     try {
24       const res = await fetch("http://localhost:3000/api/admin/login", {
25         method: "POST",
26         headers: {
27           "Content-Type": "application/json",
28         },
29       });
30     }

```

## Login Admin

**Id**

**Password**

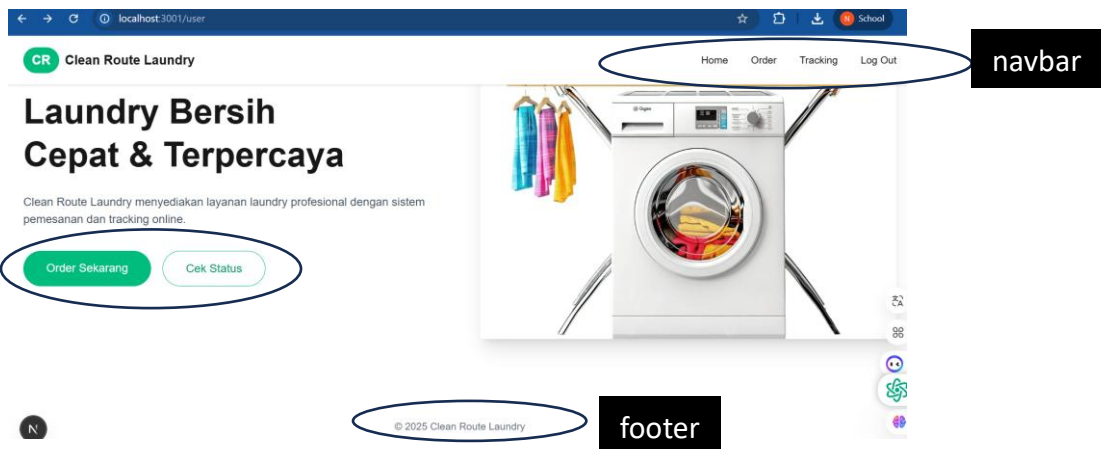
Admin hanya menginputkan ID dan password yang sudah dibuatkan.

## 6. Bagian User

Page.tsx pada folder user adalah tampilan dashboard user nya, dimana user bisa membuat orderan dll:

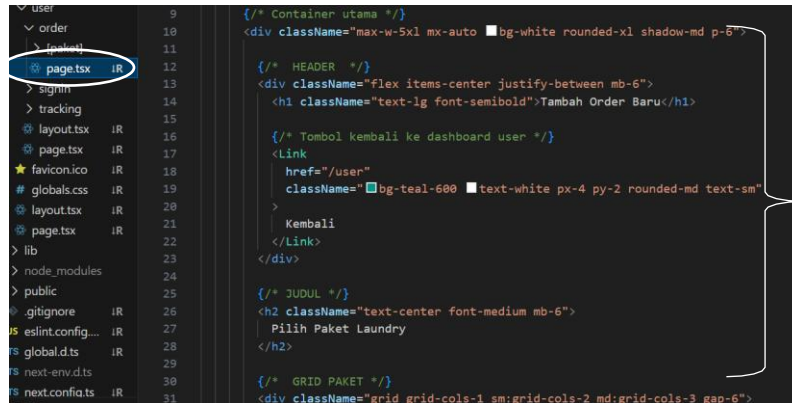
```

83 <div className="max-w-7xl mx-auto grid md:grid-cols-2 gap-14 items-center">
84
85 <div>
86   <h1 className="text-4xl md:text-5xl font-bold mb-6 leading-tight">
87     Laundry Bersih <br /> Cepat & Terpercaya
88   </h1>
89
90   <p className="text-gray-600 mb-8">
91     Clean Route Laundry menyediakan layanan laundry profesional
92     dengan sistem pemesanan dan tracking online.
93   </p>
94
95   <div className="flex flex-wrap gap-4">
96     <Link
97       href={protectedRoute("/order")}
98       className="bg-emerald-500 text-white px-8 py-3 rounded-full"
99     >
100       Order Sekarang
101     </Link>
102
103     <Link
104       href={protectedRoute("/tracking")}
105       className="border border-emerald-500 text-emerald-600 px-8 py-3 rounded-f
106   </div>
  
```

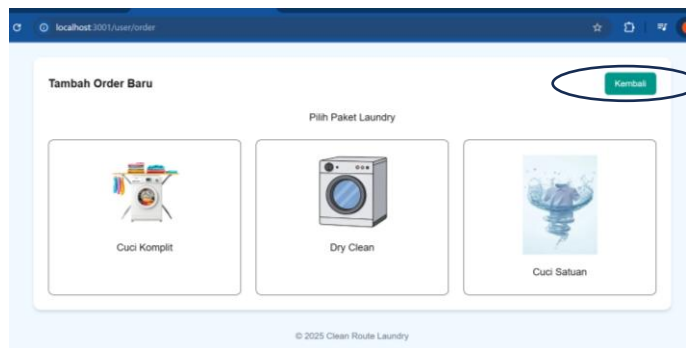


Klik "Order sekarang" untuk membuat pesanan (mengarah pada page order) dan "Cek Status" (mengarah pada page tracking) untuk melihat pesanan kita, di dashboard user juga terdapat navbar Home, Order, Tracking, dan Log Out.

7. Ketika user membuat orderan klik “Order Sekarang”, akan mengarah ke page order nya lalu user bisa memilih paket apa yang diinginkan.

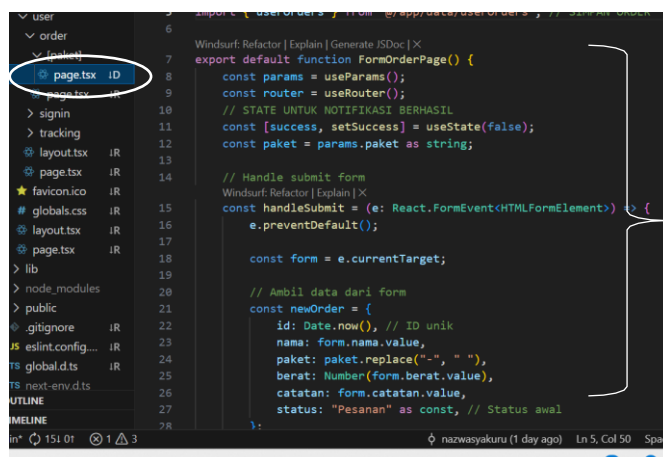


```
9      </div>
10    </div>
11  }
12  </div>
13  </div>
14  </div>
15  </div>
16  </div>
17  </div>
18  </div>
19  </div>
20  </div>
21  </div>
22  </div>
23  </div>
24  </div>
25  </div>
26  </div>
27  </div>
28  </div>
29  </div>
30  </div>
31  </div>
```



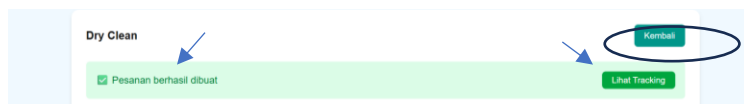
Pilih salah satu paket untuk membuatnya, terdapat button “Kembali” juga yang mengarahkan ke dashboard user.

8. Ketika user sudah memilih paket mana yang diinginkan, user akan diarahkan ke page paket order, lalu mengisi form order nya:



```
6  import { useOrders } from '@app/data/user-orders';
7  export default function FormOrderPage() {
8    const params = useParams();
9    const router = useRouter();
10    // STATE UNTUK NOTIFIKASI BERHASIL
11    const [success, setSuccess] = useState(false);
12    const paket = params.paket as string;
13
14    // Handle submit form
15    const handleSubmit = (e: React.FormEvent<HTMLFormElement>) => {
16      e.preventDefault();
17
18      const form = e.currentTarget;
19
20      // Ambil data dari form
21      const newOrder = {
22        id: Date.now(), // ID unik
23        nama: form.nama.value,
24        paket: paket.replace("-", " "),
25        berat: Number(form.berat.value),
26        catatan: form.catatan.value,
27        status: "Pesanan" as const, // Status awal
28      };
29    };
30  }
31
```

Ketika sudah menginputkan semua data nya, klik “pesan” nanti akan muncul notifikasi pesanan berhasil, terdapat button “kembali” juga yang mengarah ke dashboard user seperti gambar dibawah ini:



9. Klik “Lihat tracking” untuk melihat pesanan kita yang sudah dibuat, nanti akan mengarah ke page tracking nya:

```

export default function TrackingPage() {
  return (
    <main className="min-h-screen bg-sky-50 px-4 py-6">
      <div className="max-w-6xl mx-auto bg-white rounded-xl shadow-md p-6">
        {/HEADER/}
        <div className="flex items-center justify-between mb-6">
          <h1 className="text-lg font-semibold">Tracking Pesanan/!
        </div>
        {/ KEMBALI KE HOME USER /}
        <Link
          href="/user"
          className="bg-teal-600 text-white px-4 py-2 rounded-md text-sm"
        >Kembali
        </Link>
      </div>
      {/DESKTOP (TABLE)/}
      <div className="hidden md:block overflow-x-auto">
        <table className="w-full border-collapse">
          <thead>

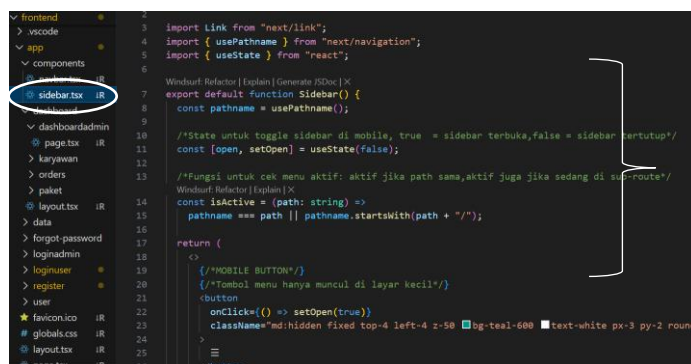
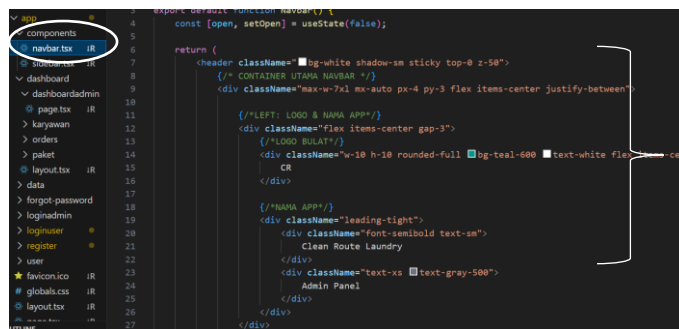
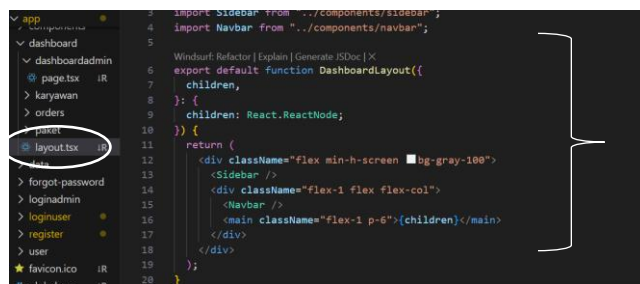
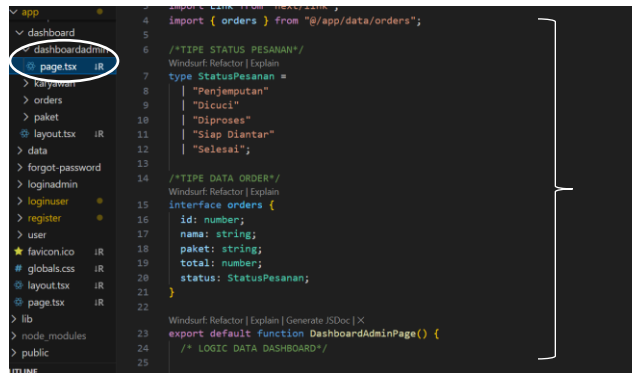
```

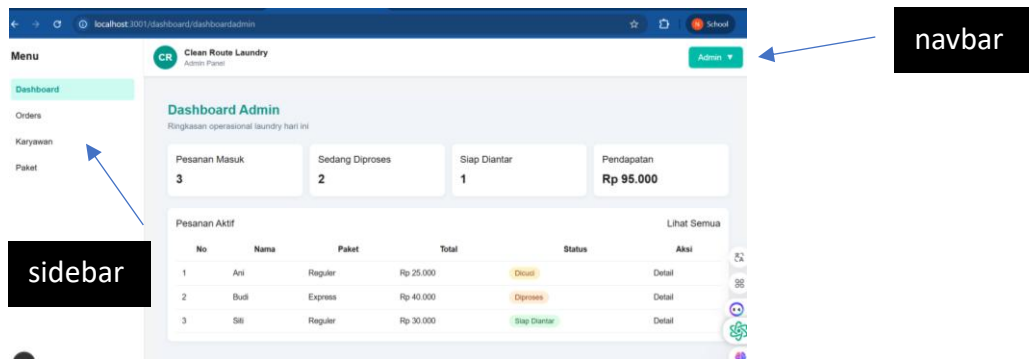
Nama	Paket	Berat	Catatan	Status
rwasykr	dry clean	1 Kg	extra pewangi	Pesanan

Menampilkan pesanan kita dengan status, tersedia button “Kembali” juga yang mengarah ke dashboard user. Jika user sudah selesai melakukan pesanan, bisa “Log Out” pada menu navbar.

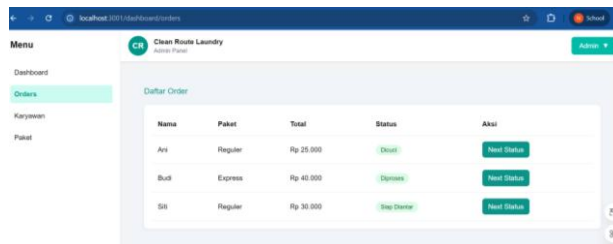
## Bagian Admin

Folder dashboard yang berisi page.tsx adalah page dashboard untuk admin, pada page ini admin bisa melihat daftar pesanan, proses order, dan pendapatan. Kemudian disitu ada file layout.tsx untuk mengatur tampilan di page admin nya dan ada folder components yang berisi navbar dan sidebar untuk dashboard admin:





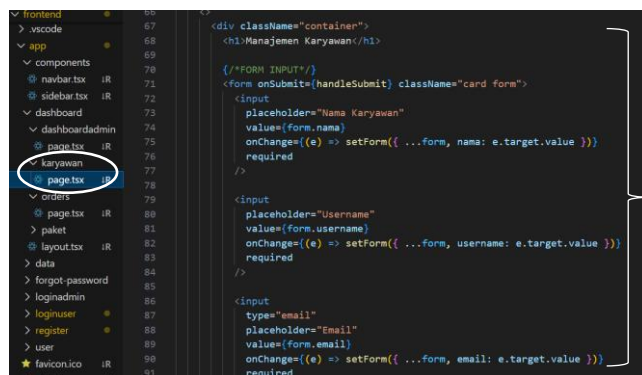
11. Folder orders yang berisi page.tsx adalah page orders untuk admin, pada page ini admin bisa melihat daftar pesanan dan bisa mengganti status pesanan.



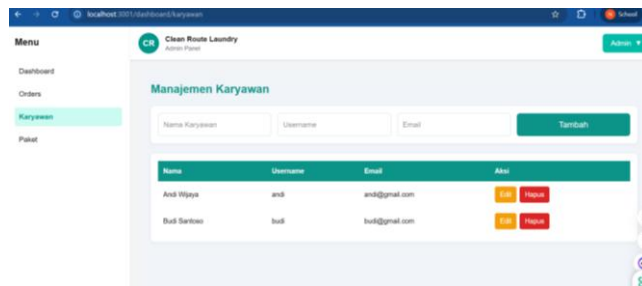
Pada page tersebut admin bisa melakukan perubahan status yang nanti nya akan masuk ke page user nya dengan klik "Next Status".



12. Folder karyawan yang berisi page.tsx adalah page orders untuk admin, pada page ini admin bisa melihat daftar karyawan dan juga bisa menambah list data karyawan.



```
60
61
62 <div className="container">
63   <h1>Manajemen Karyawan</h1>
64
65   {/*FORM INPUT*/}
66   <form onSubmit={handleSubmit} className="card form">
67     <input
68       placeholder="Nama Karyawan"
69       value={form.nama}
70       onChange={e => setForm({ ...form, nama: e.target.value })}
71       required
72     />
73
74     <input
75       placeholder="Username"
76       value={form.username}
77       onChange={e => setForm({ ...form, username: e.target.value })}
78       required
79     />
80
81     <input
82       type="email"
83       placeholder="Email"
84       value={form.email}
85       onChange={e => setForm({ ...form, email: e.target.value })}
86       required
87     />
88   </form>
89
90   <table>
91     <thead>
```

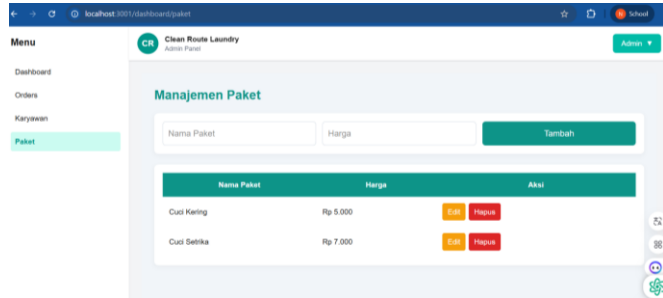


Pada page tersebut admin bisa menambahkan, mengedit ataupun menghapus karyawan.

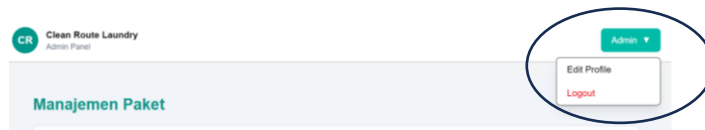
13. Folder paket yang berisi page.tsx adalah page paket untuk admin, pada page ini digunakan admin untuk manajemen paket apa saja yang tersedia.



```
1
2
3 import { useState, FormEvent } from "react";
4
5 /**INTERFACE DATA PAKET*/
6 Windurf Refactor | Explain
7 interface Paket {
8   id: number;
9   nama: string;
10  harga: number;
11 }
12
13 /**INTERFACE FORM*/
14 Windurf Refactor | Explain
15 interface FormPaket {
16   nama: string;
17   harga: string;
18 }
19
20 Windurf Refactor | Explain | Generate JSDoc | X
21 export default function Page() {
22   /*STATE DATA PAKET*/
23   const [data, setData] = useState<Paket[]>([
24     { id: 1, nama: "Cuci Kering", harga: 5000 },
25     { id: 2, nama: "Cuci Setrika", harga: 7000 },
26   ]);
27
28   /* STATE FORM INPUT*/
29   const [form, setForm] = useState<FormPaket>({
```



Pada page tersebut admin bisa menambahkan, mengedit ataupun menghapus paket orderan.



Jika admin ingin Log Out atau mengedit profile, klik button “Admin” pada menu navbar nanti akan muncul seperti gambar diatas.