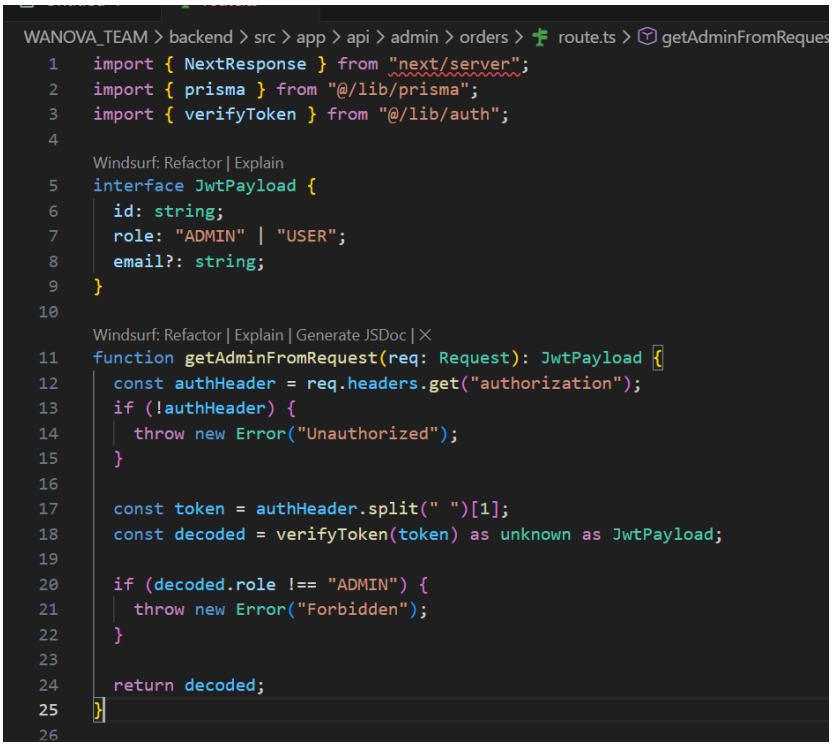
Dokumentasi Novario Dimas Kusuma project wanova_team

1. Bagian backend middleware.ts di backend

Kode middleware ini digunakan untuk mengamankan request di Next.js dengan cara memeriksa token JWT yang dikirim melalui header `Authorization`. Middleware akan mengecek apakah token tersedia dan menggunakan format `Bearer`, jika tidak maka request akan ditolak dengan status 401. Jika token ada, token tersebut diverifikasi menggunakan fungsi `verifyToken`; apabila valid, data user hasil decode token disimpan ke header request agar dapat digunakan oleh proses selanjutnya, lalu request diizinkan untuk dilanjutkan. Namun jika token tidak valid atau verifikasi gagal, middleware akan menghentikan request dan mengembalikan response dengan status 403.

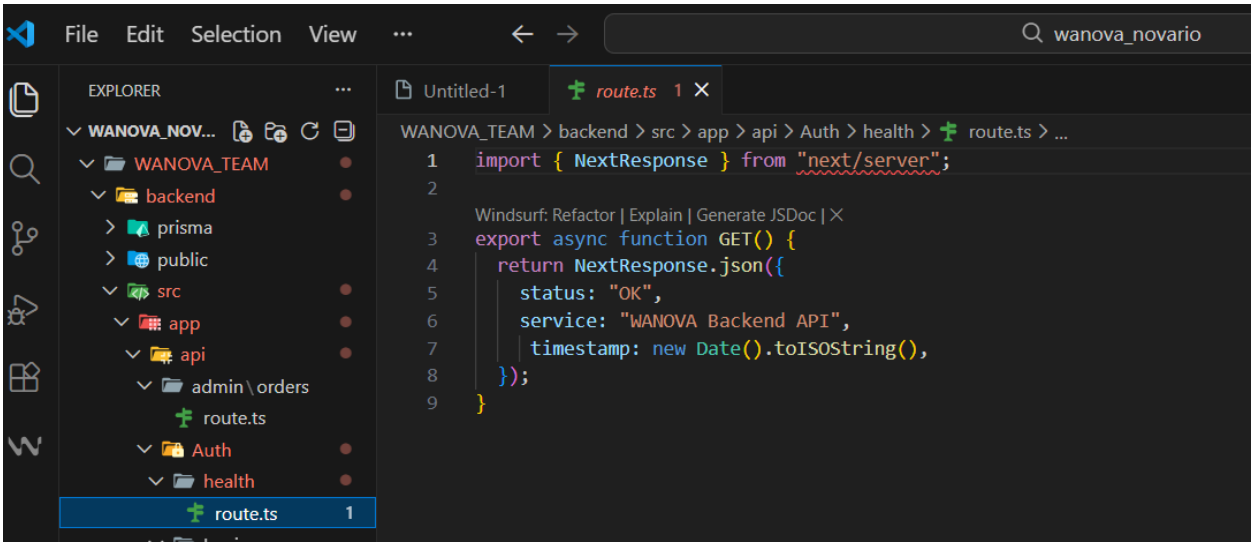
2. code/backend/src/app/api/admin/orders/route.ts



```
/* ========================
   PUT: Admin update status & tracking
======================== */
Windsurf: Refactor | Explain | ×
export async function PUT(req: Request) {
  try {
    getAdminFromRequest(req);

    const { id, status, tracking } = await req.json();

    const updated = await prisma.order.update({
      where: { id },
      data: {
        status,
        tracking,
      },
    });

    return NextResponse.json({
      message: "Pesanan berhasil diperbarui oleh admin",
      data: updated,
    });
  } catch (error) {
    const err = error as Error;

    return NextResponse.json(
      { message: err.message || "Update gagal" },
      { status: err.message === "Forbidden" ? 403 : 401 }
    );
  }
}
```

Kode ini merupakan API Next.js yang hanya bisa diakses oleh **admin** untuk mengelola pesanan. Setiap request akan dicek terlebih dahulu melalui token JWT untuk memastikan pengguna memiliki role **ADMIN**, jika tidak maka akses akan ditolak. Endpoint **GET** digunakan untuk menampilkan seluruh data pesanan beserta informasi user dari database, sedangkan endpoint **PUT** digunakan untuk memperbarui status dan nomor pelacakan pesanan berdasarkan `id`. Semua proses data dilakukan menggunakan Prisma dan hasilnya dikembalikan dalam bentuk JSON.

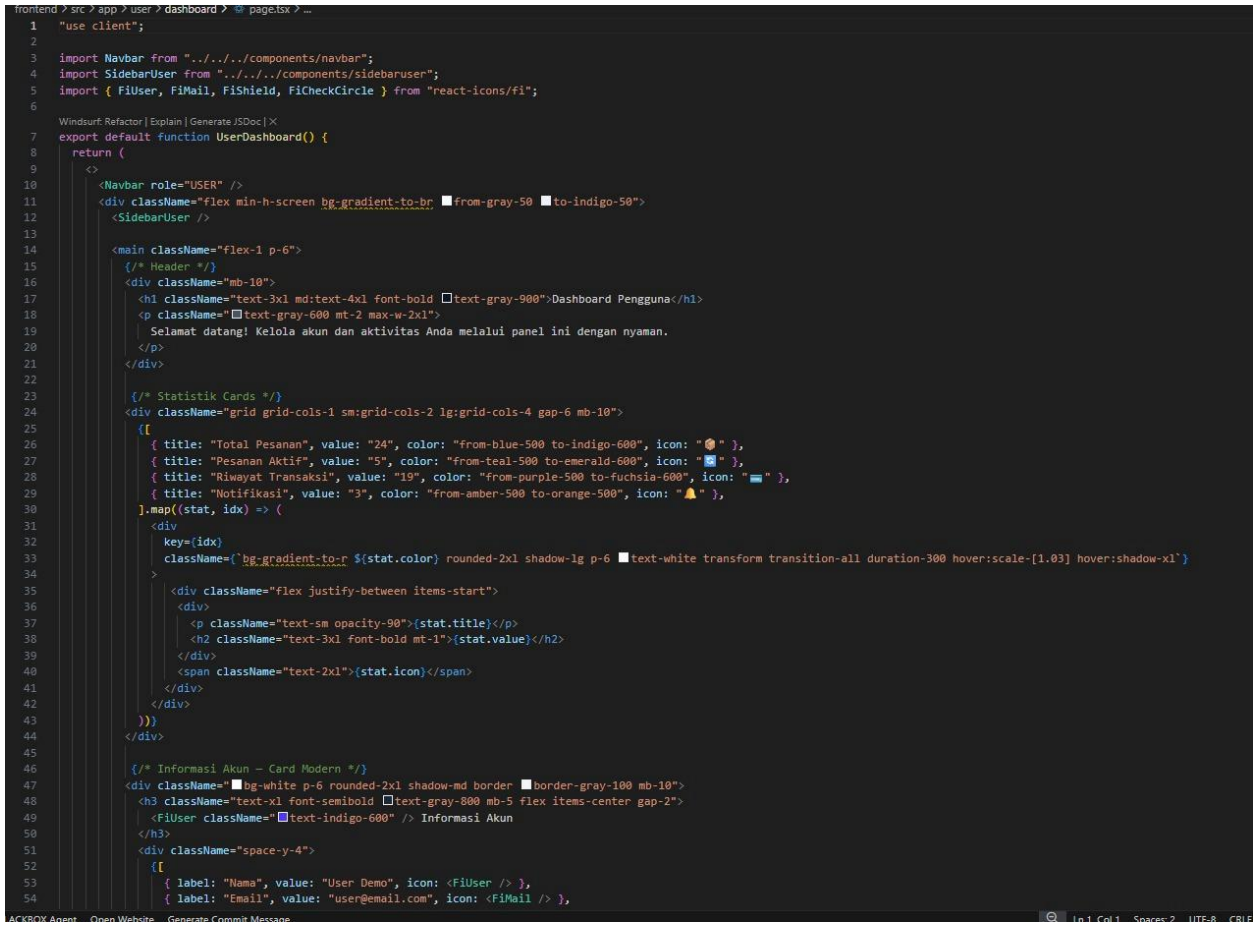
3. Membuat folder helth code/backend/src/app/api/Register/route.ts



Kode pada gambar tersebut adalah **API route health check di Next.js** yang berfungsi untuk memastikan backend berjalan dengan baik. File route.ts ini menyediakan endpoint **GET** yang akan mengembalikan response JSON berisi status "OK", nama service backend, dan waktu saat request diproses (timestamp). Endpoint ini biasanya digunakan untuk **mengecek kesehatan server** (misalnya oleh frontend, DevOps, atau monitoring system) tanpa perlu autentikasi atau proses database yang kompleks.

4. Menampilkan judul halaman dashboard user dengan ukuran besar dan tebal.

code/frontend/src/app/user/dasboard/page.tsx



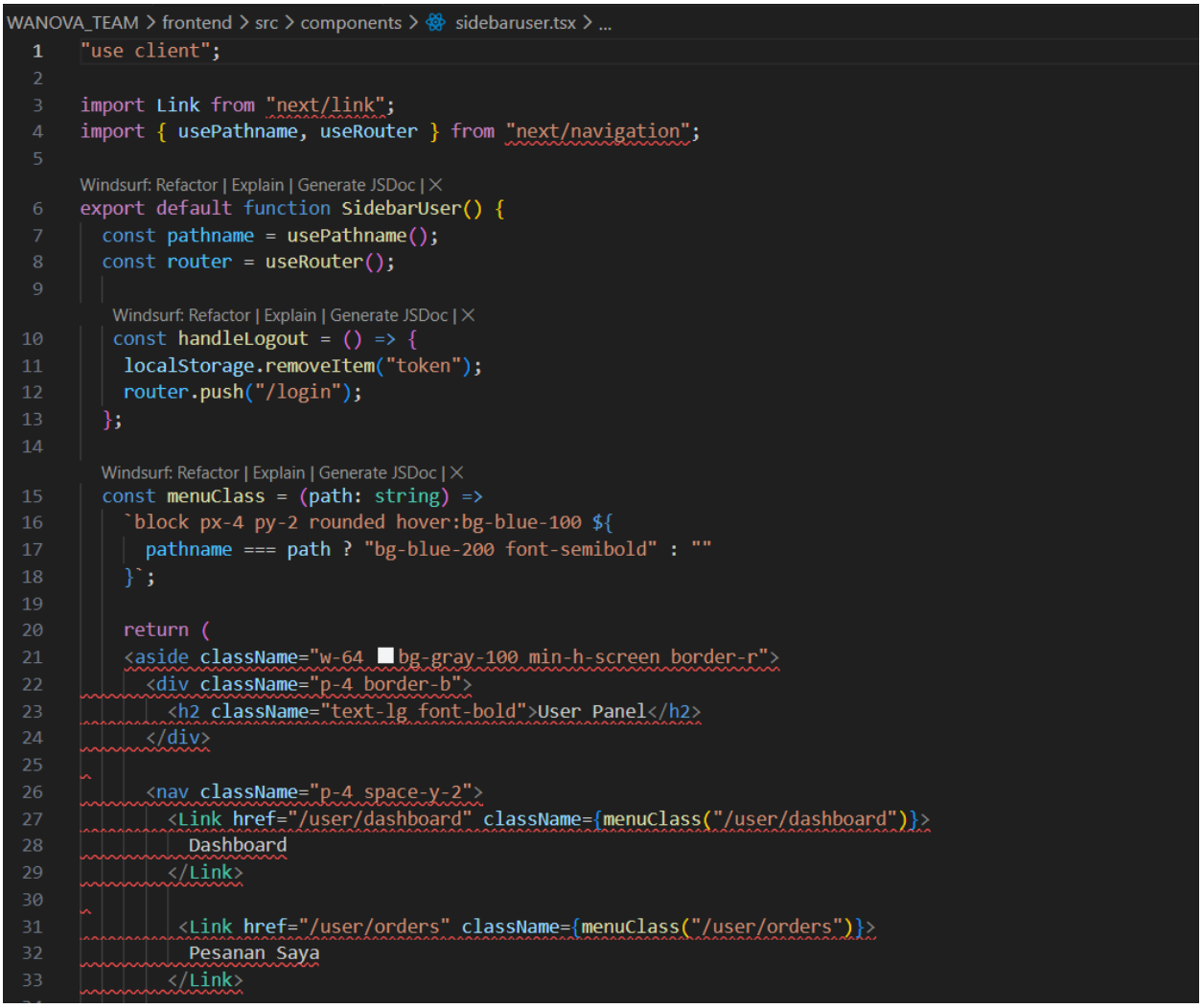
frontend > src > app > user > dashboard > page.tsx > ...

```tsx
"use client";

import Navbar from "../../../components/navbar";
import SidebarUser from "../../../components/sidebaruser";
import { FiUser, FiMail, FiShield, FiCheckCircle } from "react-icons/fi";

Windsurf: Refactor | Explain | Generate JSDoc | ×
export default function UserDashboard() {
  return (
    <>
      <Navbar role="USER" />
      <div className="flex min-h-screen bg-gradient-to-br from-gray-50 to-indigo-50">
        <SidebarUser />

        <main className="flex-1 p-6">
          {/* Header */}
          <div className="mb-10">
            <h1 className="text-3xl md:text-4xl font-bold text-gray-900">Dashboard Pengguna</h1>
            <p className="text-gray-600 mt-2 max-w-2xl">
              Selamat datang! Kelola akun dan aktivitas Anda melalui panel ini dengan nyaman.
            </p>
          </div>

          {/* Statistik Cards */}
          <div className="grid grid-cols-1 sm:grid-cols-2 lg:grid-cols-4 gap-6 mb-10">
            {[
              { title: "Total Pesanan", value: "24", color: "from-blue-500 to-indigo-600", icon: "📦" },
              { title: "Pesanan Aktif", value: "5", color: "from-teal-500 to-emerald-600", icon: "✉️" },
              { title: "Riwayat Transaksi", value: "19", color: "from-purple-500 to-fuchsia-600", icon: "💳" },
              { title: "Notifikasi", value: "3", color: "from-amber-500 to-orange-500", icon: "🔔" },
            ].map((stat, idx) => (
              <div
                key={idx}
                className={`bg-gradient-to-r ${stat.color} rounded-2xl shadow-lg p-6 text-white transform transition-all duration-300 hover:scale-[1.03] hover:shadow-xl`}
              >
                <div>
                  <div>
                    <p>{stat.title}</p>
                    <h2>{stat.value}</h2>
                  </div>
                  <span>{stat.icon}</span>
                </div>
              </div>
            ))}
          </div>

          {/* Informasi Akun — Card Modern */}
          <div>
            <h3>
              <FiUser /> Informasi Akun
            </h3>
            <div>
              {[
                { label: "Nama", value: "User Demo", icon: <FiUser /> },
                { label: "Email", value: "user@email.com", icon: <FiMail /> },
```

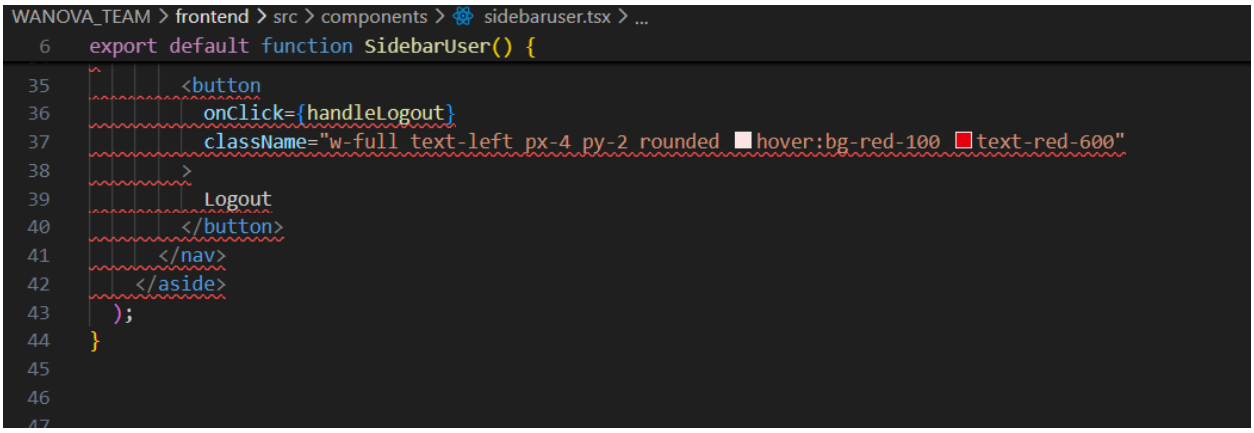
ACKBOX Agent Open Website Generate Commit Message Ln 1, Col 1 Spaces: 2 UTF-8 CRLF

Kode pada file frontend/src/app/user/dashboard/page.tsx ini digunakan untuk **menampilkan halaman dashboard pengguna (USER)** di sisi frontend menggunakan Next.js dan React. Komponen ini bersifat *client component* ("use client") karena berisi interaksi dan tampilan dinamis, seperti Navbar dan Sidebar khusus user. Halaman dashboard menampilkan sambutan pengguna, ringkasan statistik aktivitas (seperti total pesanan, pesanan aktif, riwayat transaksi, dan notifikasi) dalam bentuk kartu visual, serta bagian informasi akun yang menampilkan data dasar pengguna seperti nama dan email. Secara keseluruhan, file ini berfungsi sebagai **pusat informasi dan navigasi utama bagi pengguna setelah login**, dengan tampilan modern menggunakan Tailwind CSS dan ikon untuk meningkatkan pengalaman pengguna.

5. code/frontend/src/components/sidebaruser.tsx



```
WANOVA_TEAM > frontend > src > components > sidebaruser.tsx > ...
"use client";

import Link from "next/link";
import { usePathname, useRouter } from "next/navigation";

Windsurf: Refactor | Explain | Generate JSDoc | X
export default function SidebarUser() {
  const pathname = usePathname();
  const router = useRouter();

  Windsurf: Refactor | Explain | Generate JSDoc | X
  const handleLogout = () => {
    localStorage.removeItem("token");
    router.push("/login");
  };

  Windsurf: Refactor | Explain | Generate JSDoc | X
  const menuClass = (path: string) =>
    `block px-4 py-2 rounded hover:bg-blue-100 ${
      pathname === path ? "bg-blue-200 font-semibold" : ""
    }`;

    return (
    <aside className="w-64 bg-gray-100 min-h-screen border-r">
      <div className="p-4 border-b">
        <h2 className="text-lg font-bold">User Panel</h2>
      </div>

      <nav className="p-4 space-y-2">
        <Link href="/user/dashboard" className={menuClass("/user/dashboard")}>
          Dashboard
        </Link>

        <Link href="/user/orders" className={menuClass("/user/orders")}>
          Pesanan Saya
        </Link>
```

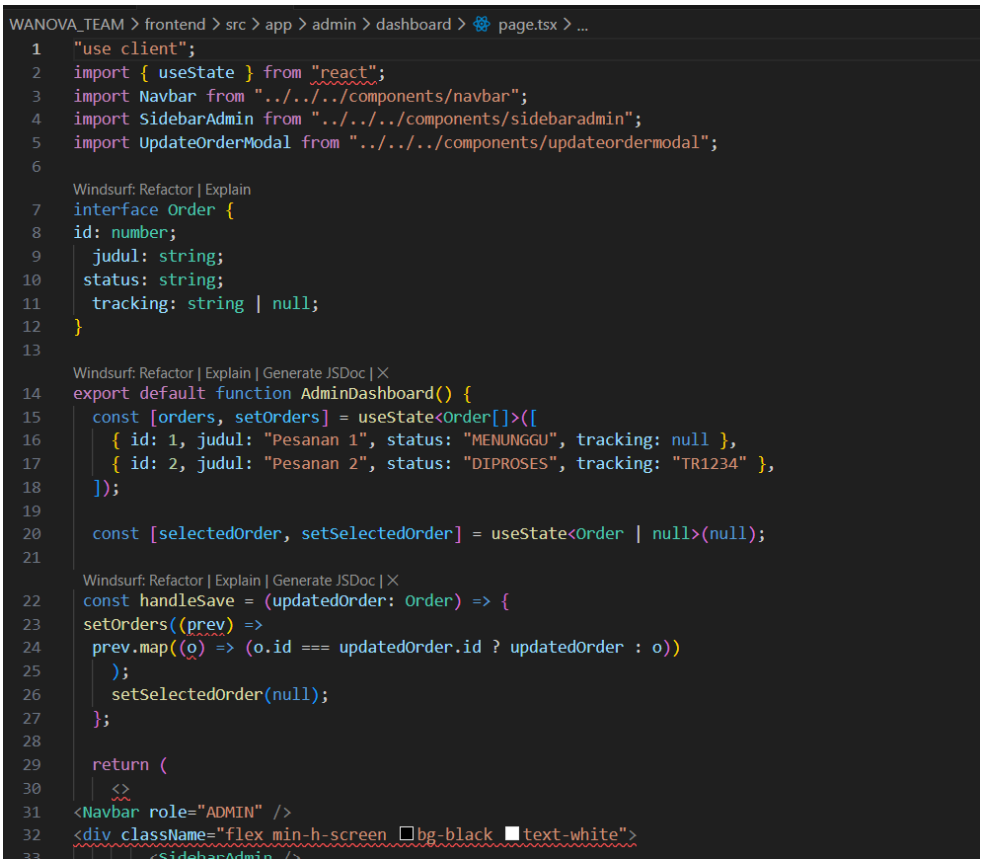


```
WANOVA_TEAM > frontend > src > components > sidebaruser.tsx > ...
export default function SidebarUser() {

        <button
          onClick={handleLogout}
          className="w-full text-left px-4 py-2 rounded hover:bg-red-100 text-red-600"
        >
          Logout
        </button>
      </nav>
    </aside>
  );
}
```

Kode pada file frontend/src/components/sidebaruser.tsx ini digunakan untuk **menampilkan sidebar navigasi khusus pengguna (USER)** di sisi frontend. Komponen ini merupakan *client component* yang memanfaatkan fitur navigasi Next.js untuk mengetahui halaman yang sedang aktif (usePathname) dan melakukan perpindahan halaman (useRouter). Sidebar menampilkan menu seperti **Dashboard** dan **Pesanan Saya**, dengan gaya aktif otomatis ketika menu sesuai dengan halaman yang sedang dibuka. Selain itu, terdapat fungsi **logout** yang menghapus token dari localStorage lalu mengarahkan pengguna kembali ke halaman login. Secara keseluruhan, komponen ini berfungsi sebagai **alat navigasi utama pengguna** agar mudah berpindah halaman dan mengelola sesi login dengan tampilan yang rapi dan interaktif.

6. Menampilkan komponen Navbar dengan peran sebagai ADMIN.

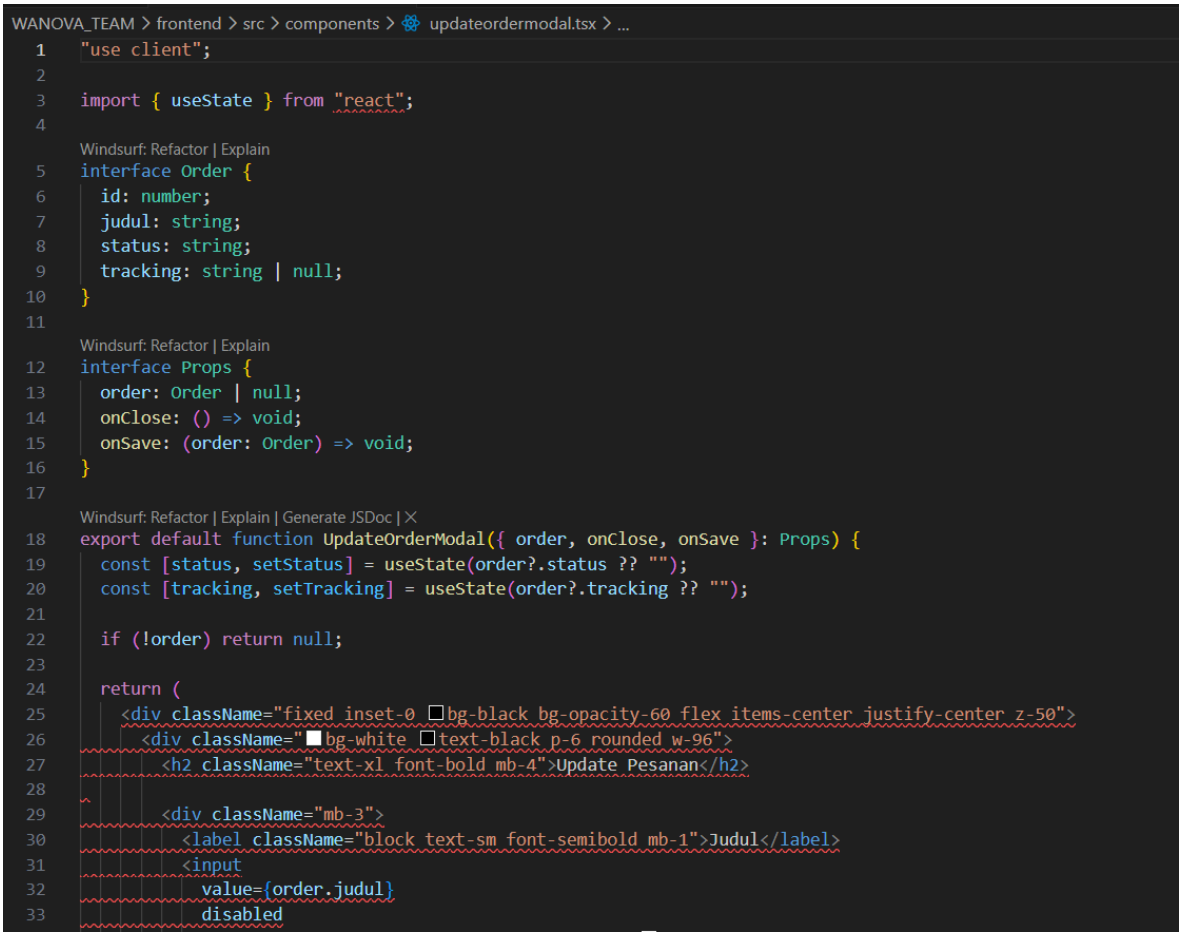
code/frontend/src/app/admin/dashboard/page.tsx



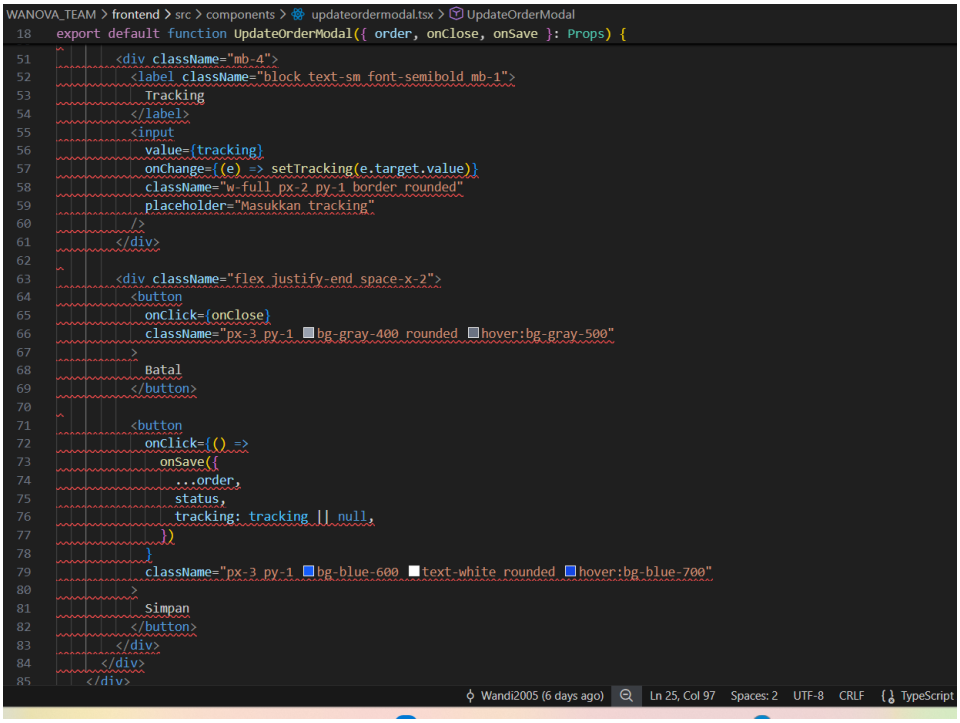
Kode pada file `frontend/src/app/admin/dashboard/page.tsx` ini digunakan untuk **menampilkan halaman dashboard admin** di sisi frontend. Komponen ini merupakan *client component* (`"use client"`) karena menggunakan state React untuk mengelola data secara dinamis. Di dalamnya, terdapat state `orders` yang menyimpan daftar pesanan (contoh data dummy) dan state `selectedOrder` untuk menyimpan pesanan yang sedang dipilih atau diedit. Fungsi `handleSave` berperan untuk memperbarui data pesanan di state ketika admin menyimpan

perubahan, lalu menutup modal edit dengan mengosongkan `selectedOrder`. Komponen ini juga menggunakan `Navbar` dengan role **ADMIN**, `SidebarAdmin` sebagai navigasi admin, serta `UpdateOrderModal` untuk mengubah status dan nomor tracking pesanan, sehingga halaman ini menjadi pusat pengelolaan pesanan oleh admin.

7.perbaikan code menghilangkan code any code/frontend/src/components/updateordermodal.tsx



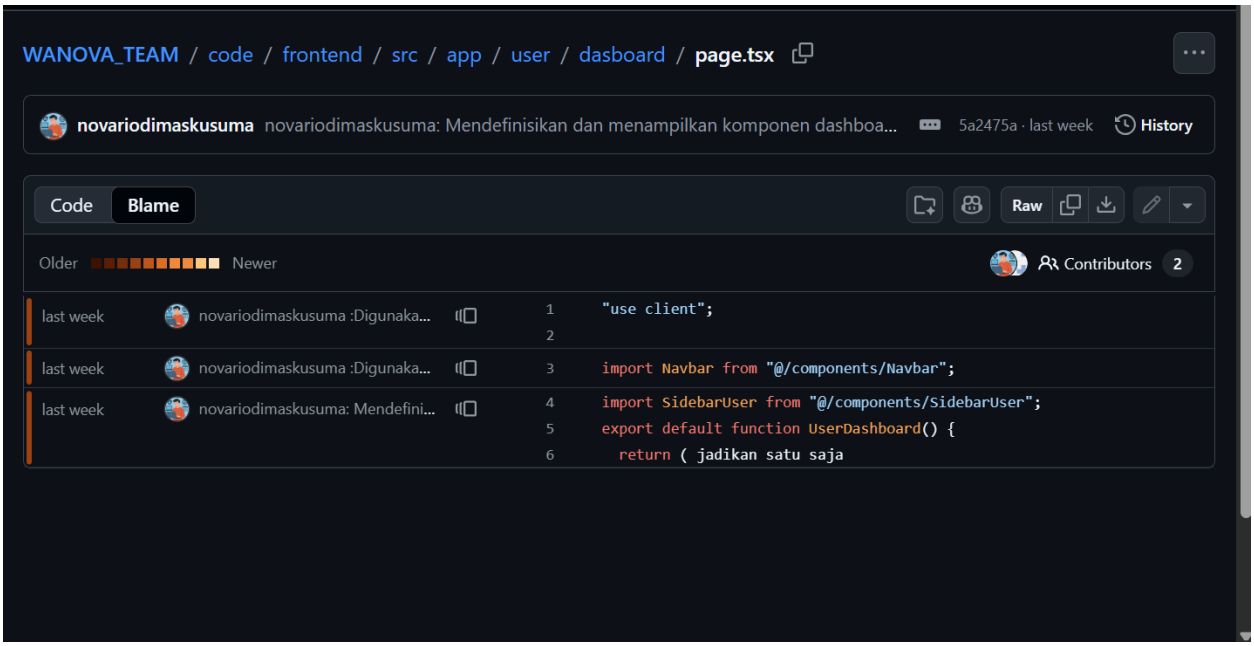
```
WANOVA_TEAM > frontend > src > components > updateordermodal.tsx > ...
1   "use client";
2
3   import { useState } from "react";
4
    Windsurf: Refactor | Explain
5   interface Order {
6     id: number;
7     judul: string;
8     status: string;
9     tracking: string | null;
10  }
11
    Windsurf: Refactor | Explain
12  interface Props {
13    order: Order | null;
14    onClose: () => void;
15    onSave: (order: Order) => void;
16  }
17
    Windsurf: Refactor | Explain | Generate JSDoc | ×
18  export default function UpdateOrderModal({ order, onClose, onSave }: Props) {
19    const [status, setStatus] = useState(order?.status ?? "");
20    const [tracking, setTracking] = useState(order?.tracking ?? "");
21
22    if (!order) return null;
23
24    return (
25      <div className="fixed inset-0 □bg-black bg-opacity-60 flex items-center justify-center z-50">
26        <div className="■bg-white □text-black p-6 rounded w-96">
27          <h2 className="text-xl font-bold mb-4">Update Pesanan</h2>
28
29          <div className="mb-3">
30            <label className="block text-sm font-semibold mb-1">Judul</label>
31            <input
32              value={order.judul}
33              disabled
```

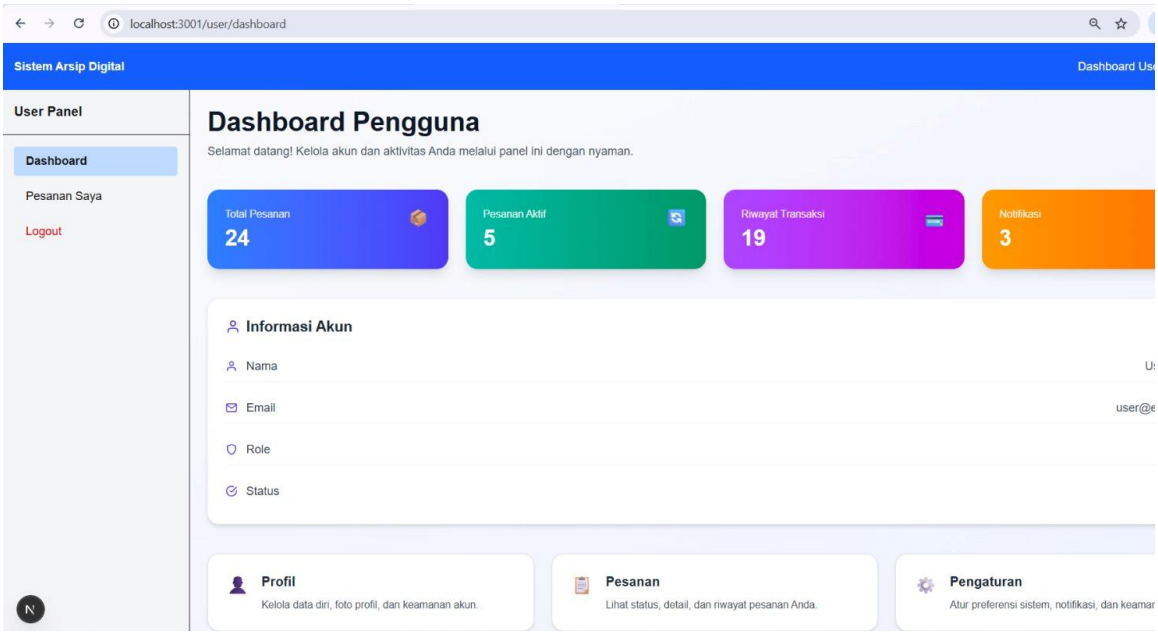


Kode pada file `frontend/src/components/updateordermodal.tsx` ini digunakan untuk **menampilkan modal (popup) yang memungkinkan admin memperbarui data pesanan**. Komponen ini menerima data pesanan (`order`) serta fungsi `onClose` dan `onUpdated` dari parent component. Di dalamnya, React state digunakan untuk menyimpan status pesanan, nomor resi (tracking), dan kondisi loading saat proses update berlangsung. Ketika tombol **Simpan** ditekan, fungsi `handleUpdate` akan mengirim request **PUT** ke API admin dengan membawa data pesanan yang diperbarui dan token autentikasi dari `localStorage`. Setelah proses selesai, modal ditutup dan parent component diberi tahu agar data pesanan diperbarui. Secara keseluruhan, komponen ini berfungsi sebagai **antarmuka interaktif bagi admin untuk mengedit status dan tracking pesanan secara langsung**.

8. code/frontend/src/app/user/dasboard/page.tsx

Mendefinisikan dan menampilkan komponen dashboard user.main





Gambar tersebut menampilkan **halaman Dashboard Pengguna (User Dashboard)** dari aplikasi *Sistem Arsip Digital*. Halaman ini berfungsi sebagai **pusat informasi utama bagi pengguna setelah login**, yang menampilkan ringkasan aktivitas dan status akun. Di bagian atas terdapat **navbar** dengan judul sistem dan tombol logout, sementara di sisi kiri ada **sidebar navigasi** untuk berpindah ke menu Dashboard, Pesanan Saya, dan Logout. Bagian utama dashboard menampilkan **kartu statistik** seperti total pesanan, pesanan aktif, riwayat transaksi, dan notifikasi agar pengguna dapat melihat aktivitasnya secara cepat. Selain itu, terdapat panel **Informasi Akun** yang menampilkan nama, email, role pengguna, dan status akun, serta beberapa menu tambahan seperti Profil, Pesanan, dan Pengaturan. Secara keseluruhan, halaman ini dirancang untuk memberikan **tampilan ringkas, informatif, dan mudah digunakan** bagi pengguna dalam mengelola akun dan aktivitasnya.