

## **LAPORAN UAS PEMROGAMAN WEB**

### **“NORDIC CAFÉ WEBSITE”**

Link Github : <https://github.com/daffafauzan94/UAS-PEMWEB>



**Dosen Pengampu :**

Ari Kurniawan, S.Kom, M.T.

**Disusun Oleh:**

Kelompok 12 2020B

Daffa Fauzan Hadiansyah 20051397024

Mokhammad Brian Yusuf 20051397036

Muhammad Imam Amiruddin 20051397068

**PRODI D4 MANAJAMEN INFORMATIKA**

**FAKULTAS VOKASI**

**UNIVERSITAS NEGERI SURABAYA**

**2024**

## **A. LATAR BELAKANG**

Dalam era digital yang semakin maju ini, website telah menjadi bagian tidak terpisahkan dalam kehidupan sehari-hari. Website-website tersebut tidak hanya memberikan kenyamanan, tetapi juga membuka peluang baru dalam berbagai sektor, termasuk industri retail. Salah satu bentuk website yang semakin populer adalah website. Di era digital yang semakin maju, layanan belanja online telah menjadi bagian integral dalam kehidupan sehari-hari banyak orang. Salah satu tren yang terus berkembang adalah pembelian minuman secara online dan informasi terkait sebuah Cafe sehingga memungkinkan kita mendapat informasi secara online tanpa harus datang ke Cafe terlebih dahulu secara langsung. Website Cafe “Nordic” hadir untuk memenuhi kebutuhan konsumen yang mencari rekomendasi minuman dan bubuk kopi berkualitas tinggi dan lezat dengan kenyamanan dalam informasi yang diberikan dari rumah.

## **B. DASAR PERMASALAHAN**

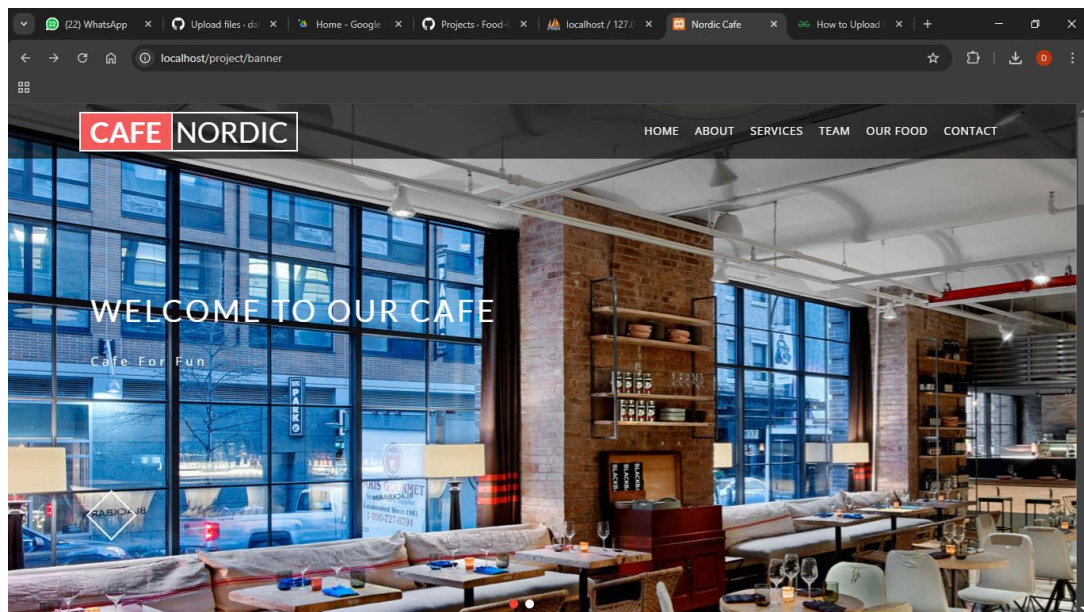
Dasar permasalahan sebagai berikut :

- 1.1 Kebijakan Keamanan dan Perlindungan Data: pengembang harus mempertimbangkan kebutuhan keamanan yang meliputi perlindungan data pengguna, perlindungan terhadap serangan siber, enkripsi data, serta kepatuhan terhadap kebijakan privasi dan regulasi yang berlaku.
- 1.2 Keterbatasan Perangkat Keras : Persyaratan waktu dan memori pada perangkat keras dapat membatasi pilihan pengembang dalam pengembangan perangkat lunak. Misalnya, jika perangkat keras yang tersedia hanya memiliki kapasitas memori terbatas, pengembang harus mempertimbangkan ukuran dan kompleksitas program agar sesuai dengan kapasitas perangkat keras.
- 1.3 Persyaratan Bahasa : Jika pengguna dari berbagai negara akan menggunakan perangkat lunak, pengembang harus mempertimbangkan persyaratan bahasa dan memastikan bahwa perangkat lunak dapat berfungsi dengan baik di berbagai bahasa.
- 1.4 Konvensi Desain Atau Standar Pemrograman : Untuk standar bahasa pemrograman yang di berikan terkait dengan bahasa html, css, dan javascript, kami masih memiliki kendala terhadap tiga bahasa pemrograman tersebut. Sehingga keterbatasan yang disebutkan, berdampak pada kurangnya desain yang di berikan pada website kedai kopi.

## C. DAFTAR FITUR

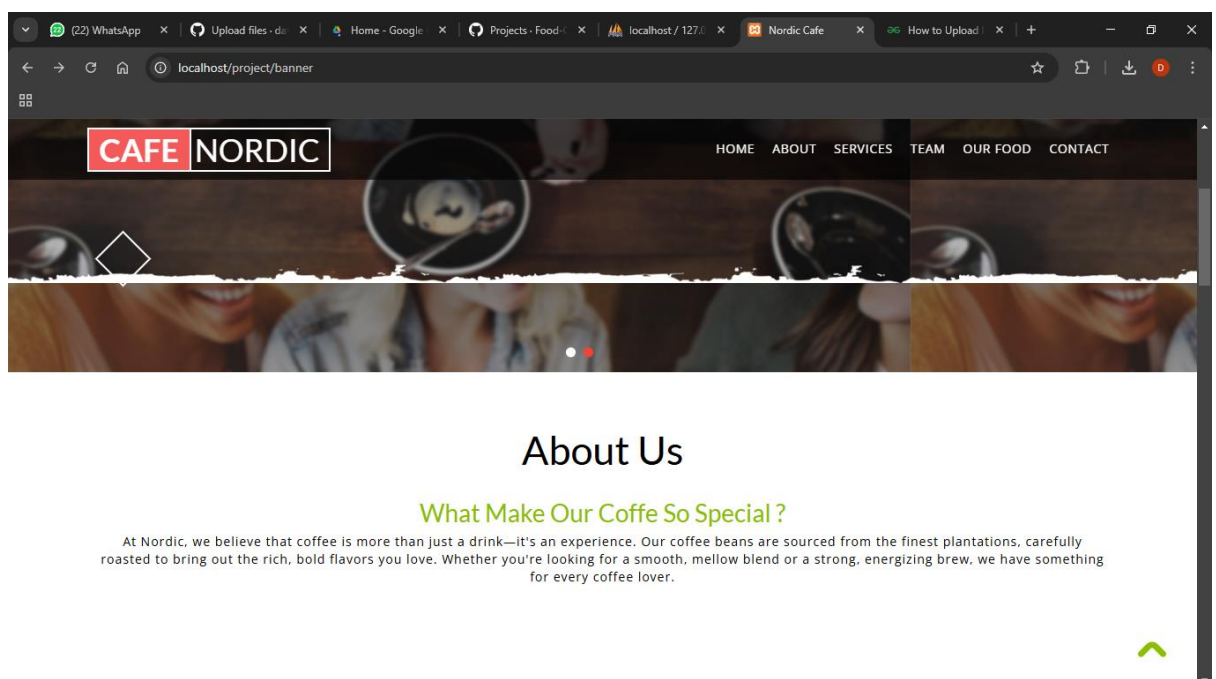
Daftar fitur sebagai berikut :

### 2.1 Fitur Menu Awal / Home



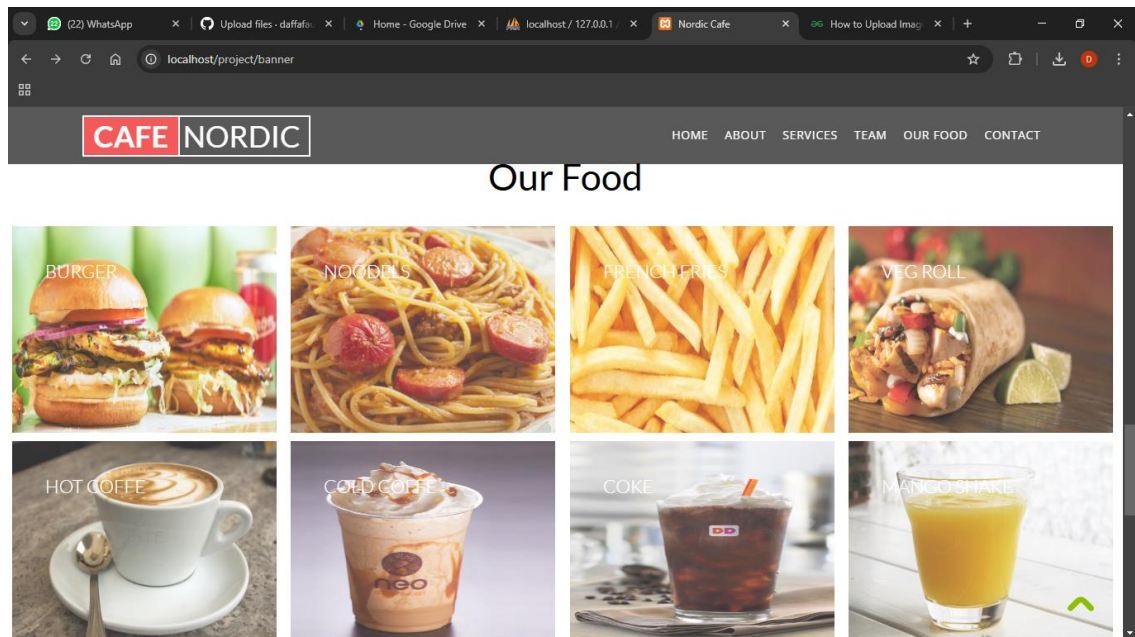
Pada awal pengguna membuka halaman website, pengguna mendapat fitur footer yang terdiri dari home, tentang kami, menu, produk, hubungi kami, dan pilihan bahasa. Selanjutnya pada bagian kanan, terdapat fitur pencarian, yang berguna untuk mencari produk yang diinginkan oleh pengguna, lalu tampilan keranjang, dan tampilah homebar.

### 2.2 Fitur Tentang Kami



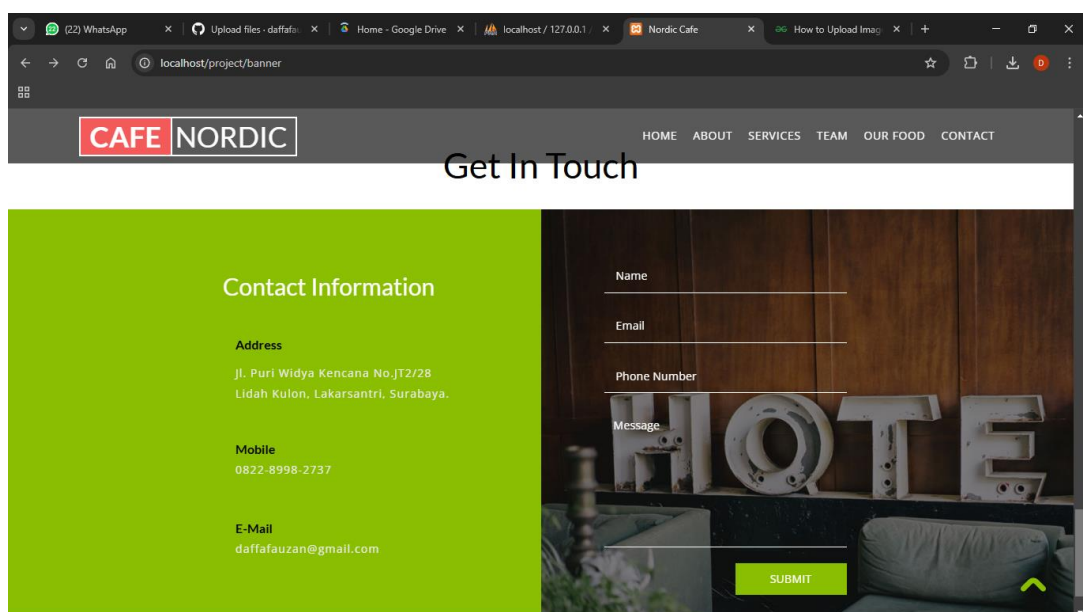
Pada tampilan, terdapat sebuah kalimat yang diambil dari sebuah tembok yang berada di Café Nordic. Dan terdapat gambaran foto Café Nordic. Dimana diharapkan dengan keberadaan tampilan tentang kami pada website, dapat mengundang para pengunjung untuk berkunjung ke kedai kopi.

## 2.3 Fitur Menu



Pada tampilan menu, pengguna dapat terbantu. Bahwa terdapat menu apa saja yang tersedia di dalam Café Nordic. Pada tampilan menu, terdapat foto produk, dan juga harga produk.

## 2.4 Fitur Kontak



Pada tampilan ini, pengguna dapat menghubungi Café Nordic dengan menyatukan nama, email, dan no hp. Dan terdapat tombol contact now di bawahnya. Lalu pada tampilan kiri, terdapat peta menuju Cafe.

#### **D. DAFTAR PENGGUNA**

Berdasarkan tujuan dan fungsi utama dari produk website Café Nordic dan beberapa daftar pengguna yang diantisipasi akan menggunakan produk ini meliputi:

- I. Admin : Daftar pengguna ini akan memiliki hak akses penuh ke sistem dan bertanggung jawab untuk mengelola pengguna, produk, dan laporan. Admin harus memiliki keahlian teknis yang tinggi dalam mengelola website, termasuk pengaturan dan konfigurasi server.
- II. Pembeli : Daftar pengguna ini dapat menjelajahi produk yang ditawarkan oleh penjual. Mereka harus melakukan transaksi melalui aplikasi. Pembeli harus memiliki keahlian teknis tertentu untuk menggunakan aplikasi kami.
- III. Tamu: Daftar pengguna ini adalah pengunjung. Mereka dapat menjelajahi dan melihat produk yang ditawarkan oleh penjual, dan dapat melakukan transaksi atau mengakses fitur lainnya di aplikasi. Dan pengguna dapat bertransaksi secara langsung pada Cafe kami.

Daftar pengguna yang paling penting untuk dipenuhi adalah admin dan pembeli, karena mereka bertanggung jawab untuk mengelola dan menjalankan website. Pembeli dan tamu juga penting, tetapi persyaratan mereka lebih sederhana dan lebih mudah dipenuhi.

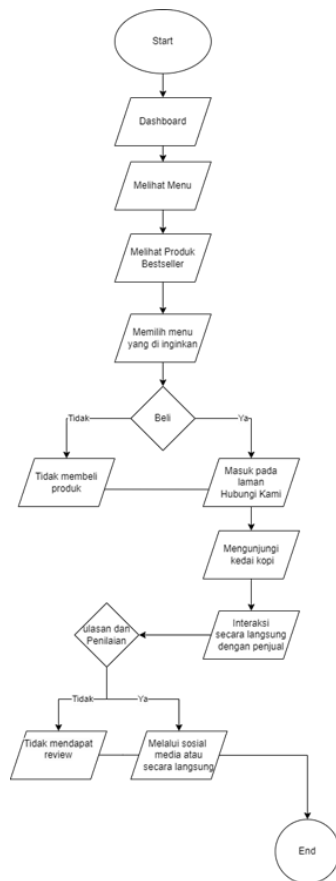
Berikut panduan penggunaan website bagi pengguna tentang fitur-fitur penting pada website cafe nordic :

- i. Kunjungi website Café Nordic : pengguna dapat mengunjungi website terlebih dahulu, jika ingin mengetahui seputar informasi Café Nordic
- ii. Produk yang tersedia pada cafe : pengguna dapat melihat seputar produk apa saja yang terdapat di Café Nordic
- iii. Rekomendasi produk : jika pengguna kesulitan dalam memilih produk yang akan di pesan, pengguna dapat memilih fitur produk. Pada fitur tersebut akan muncul produk rekomendasi Café Nordic

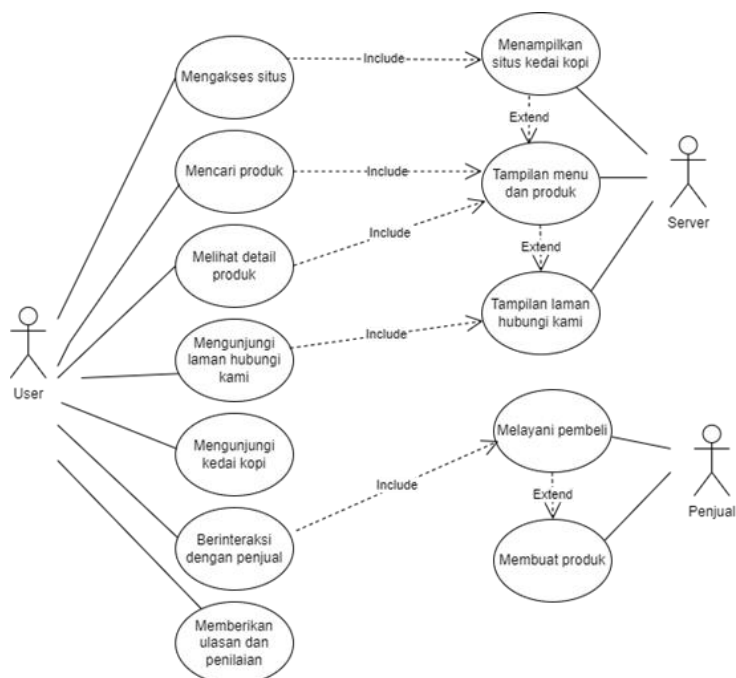
- iv. Komunikasi lebih lanjut : jika pengguna ingin berkomunikasi lebih lanjut terkait dengan Cafe Nordic, pengguna dapat beralih pada fitur hubungi kami. Pada fitur tersebut, pengguna dapat berkomunikasi lebih lanjut terkait partner bisnis, rekan bisnis, dan ulasan cafe atau produk Nordic. Dengan memasukkan nama, email, no hp, dan pesan yang ingin disampaikan
- v. Pengalihan pemesanan melalui aplikasi : pengguna dapat memesan produk kami, dengan menggunakan aplikasi yang tersedia. Dimana nantinya kami akan membuat sebuah aplikasi terkait pemesanan produk pelanggan. Untuk pengambilan pemesanan, dapat kami delivery maupun mengambil produk pada cafe kami(Pick up)
- vi. Pengalihan pemesanan terhadap Cafe : pada kasus ini, kami memiliki dua kasus. Yang pertama jika pelanggan sudah memesan melalui aplikasi dan memilih pengambilan produk secara pick up. Maka pelanggan hanya memberikan bukti pesan dan pembayaran kepada penjual cafe, maka pesanan sudah dapat diproses-menikmati. Lalu pada kasus ke dua, pelanggan dapat datang secara langsung dengan informasi yang tersedia pada website letak Café Nordic. Dan pelanggan memilih produk yang diinginkan. Penjual memproses pemesanan pelanggan.

## **E. BAGAN / GRAFIK ANALISA**

Flowchart :



Usecase :



## F. STRUKTUR KODE SCRIPT & PROGRAM

Struktur kode script dan program :

- 1) HTML
- 2) CSS
- 3) JAVASCRIPT

```
1 // Mengambil elemen dengan class 'navbar' dan menyimpannya di variabel 'navbar'  
2 let navbar = document.querySelector('.navbar');
```

- Kode tersebut menggunakan metode `querySelector` dari DOM (Document Object Model) untuk mengambil elemen pertama yang memiliki class `navbar` dari halaman HTML dan menyimpannya di variabel `navbar`. Jika elemen dengan class `navbar` ada di halaman, maka elemen tersebut akan tersimpan di variabel `navbar`.

```
1 // Mengambil elemen dengan id 'menu-btn' dan menyimpannya di variabel 'menuBtn'  
2 let menuBtn = document.getElementById('menu-btn');
```

- Berfungsi untuk mengambil elemen HTML yang memiliki atribut `id` dengan nilai `menu-btn`, lalu menyimpannya ke dalam variabel `menuBtn`.
- Metode `document.getElementById()` digunakan untuk mengambil elemen HTML berdasarkan ID, dan karena ID harus unik dalam sebuah dokumen HTML, metode ini akan mengembalikan hanya satu elemen (atau `null` jika elemen dengan ID tersebut tidak ditemukan).



```

1 // Menambahkan event listener untuk 'click' pada tombol menu
2 menuBtn.addEventListener('click', () => {
3     // Menambahkan atau menghapus class 'active' pada elemen navbar saat tombol diklik
4     navbar.classList.toggle('active');
5
6     // Menghapus class 'active' dari elemen searchForm saat tombol menu diklik
7     searchForm.classList.remove('active');
8
9     // Menghapus class 'active' dari elemen cartItem saat tombol menu diklik
10    cartItem.classList.remove('active');
11 });

```

- `Menu Btn.addEventListener('click', () => { ... });` , menambahkan event listener pada elemen `menuBtn`. Event listener ini mendeteksi jika pengguna mengklik tombol (elemen `menuBtn`). Saat tombol diklik, fungsi di dalamnya (arrow function) akan dieksekusi.
- `navbar.classList.toggle('active');` , `toggle('active')` akan menambahkan atau menghapus class `active` dari elemen `navbar`. Jika elemen `navbar` belum memiliki class `active`, class tersebut akan ditambahkan. Sebaliknya, jika sudah memiliki class `active`, class itu akan dihapus. Ini sering digunakan untuk membuat elemen muncul atau menghilang, misalnya untuk menampilkan atau menyembunyikan menu navigasi.
- `searchForm.classList.remove('active');` , menghapus class `active` dari elemen `searchForm` (form pencarian) jika ada. Ini memastikan bahwa ketika tombol menu diklik, form pencarian tidak dalam kondisi aktif (tersembunyi atau ditutup).
- `cartItem.classList.remove('active');` , sama seperti `searchForm`, class `active` juga dihapus dari elemen `cartItem` (keranjang belanja). Ini memastikan bahwa ketika tombol menu diklik, elemen keranjang belanja juga tidak dalam kondisi aktif.

a) Fungsi Keseluruhan:

Ketika Tombol menu diklik:

- Navigasi (`navbar`) akan muncul atau hilang dengan menambahkan/menghapus class `active`.
- Elemen pencarian (`searchForm`) dan keranjang belanja (`cartItem`) akan dipastikan dalam keadaan tertutup (class `active` dihapus).

```
1 // Mengambil elemen dengan class 'search-form' dan menyimpannya di variabel 'searchForm'
2 let searchForm = document.querySelector('.search-form');
```

- `document.querySelector('.search-form')`: , Metode ini digunakan untuk mencari elemen HTML berdasarkan selektor CSS. Dalam hal ini, selektor yang digunakan adalah `.search-form`, yang menandakan pencarian elemen yang memiliki class `search-form`.
- Jika ada beberapa elemen dengan class yang sama, metode ini hanya akan mengambil elemen pertama yang ditemukan. Jika tidak ada elemen yang cocok dengan class tersebut, maka nilai yang dikembalikan adalah `null`.

```
1 // Menambahkan event listener untuk 'click' pada tombol search dengan id 'search-btn'
2 document.querySelector('#search-btn').onclick = () => {
3   // Menambahkan atau menghapus class 'active' pada elemen searchForm saat tombol search diklik
4   searchForm.classList.toggle('active');
5
6   // Menghapus class 'active' dari elemen navbar saat tombol search diklik
7   navbar.classList.remove('active');
8
9   // Menghapus class 'active' dari elemen cartItem saat tombol search diklik
10  cartItem.classList.remove('active');
11 }
```

a. `document.querySelector('#search-btn')`

- `document.querySelector('#search-btn')` mengambil elemen HTML yang memiliki ID `search-btn`.
- `onclick` adalah properti yang memungkinkan kita untuk menetapkan fungsi yang akan dieksekusi ketika elemen tersebut diklik.
- Dalam contoh ini, ketika elemen dengan ID `search-btn` diklik, fungsi di dalam arrow function akan dieksekusi.

b. `searchForm.classList.toggle('active');`

- `classList.toggle('active')` menambahkan atau menghapus class `active` pada elemen `searchForm`.
- Jika form pencarian (`searchForm`) tidak memiliki class `active`, class tersebut akan ditambahkan, dan sebaliknya, jika sudah ada, class tersebut akan dihapus.
- Ini sering digunakan untuk menampilkan (open) atau menyembunyikan (close) elemen form pencarian saat tombol search diklik.

c. `navbar.classList.remove('active');`

- Ini menghapus class active dari elemen navbar.
- Saat tombol pencarian diklik, elemen navigasi (navbar) akan dipastikan dalam keadaan tertutup, dengan cara menghapus class active dari elemen tersebut (jika class active ada).

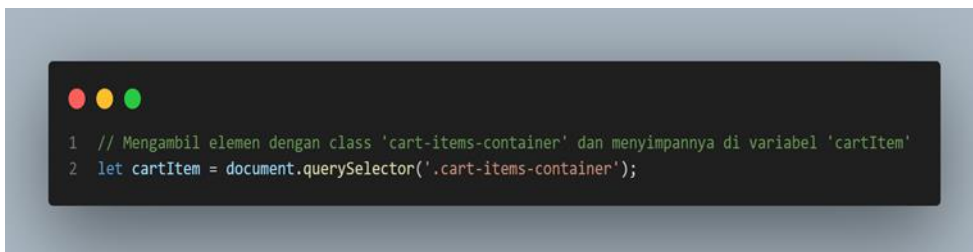
d. `cartItem.classList.remove('active');`

- Menghapus class active dari elemen cartItem.
- Sama seperti elemen navigasi, elemen keranjang belanja (cartItem) akan ditutup saat tombol search diklik, dengan cara menghapus class active.

b) Fungsi Keseluruhan :

Saat tombol search dengan ID search-btn diklik:

- Form pencarian (searchForm) akan tampil atau tersembunyi (dengan menambah/menghapus class active).
- Navigasi (navbar) akan dipastikan tertutup (class active dihapus).
- Keranjang belanja (cartItem) juga akan dipastikan tertutup (class active dihapus).



- `document.querySelector('.cart-items-container')`: , Metode ini mencari elemen HTML yang memiliki class `cart-items-container`.
- Jika ada beberapa elemen dengan class tersebut, metode ini hanya akan mengembalikan elemen pertama yang ditemukan. Jika tidak ada elemen yang cocok, nilai yang dikembalikan adalah `null`.

```
1 // Menambahkan event listener untuk 'click' pada tombol cart dengan id 'cart-btn'
2 document.querySelector('#cart-btn').onclick = () => {
3   // Menambahkan atau menghapus class 'active' pada elemen cartItem saat tombol cart diklik
4   cartItem.classList.toggle('active');
5
6   // Menghapus class 'active' dari elemen navbar saat tombol cart diklik
7   navbar.classList.remove('active');
8
9   // Menghapus class 'active' dari elemen searchForm saat tombol cart diklik
10  searchForm.classList.remove('active');
11 }
```

- a. `document.querySelector('#cart-btn').onclick = () => { ... };`
- `document.querySelector('#cart-btn')` mencari elemen dengan ID `cart-btn`.
  - Properti `onclick` menambahkan event listener yang akan dijalankan ketika tombol dengan ID `cart-btn` diklik.
  - Setelah tombol diklik, fungsi di dalam arrow function akan dieksekusi.
- b. `cartItem.classList.toggle('active');`
- Fungsi `toggle('active')` pada `classList` menambahkan atau menghapus class `active` dari elemen `cartItem`.
  - Jika class `active` tidak ada pada elemen `cartItem`, maka class tersebut akan ditambahkan (misalnya untuk menampilkan keranjang belanja). Jika sudah ada, class itu akan dihapus (misalnya untuk menyembunyikan keranjang belanja).
- c. `navbar.classList.remove('active');`
- Menghapus class `active` dari elemen `navbar`.
  - Ini memastikan bahwa elemen `navbar` (navigasi) akan tertutup ketika tombol `cart` diklik.
- d. `searchForm.classList.remove('active');`
- Sama seperti pada `navbar`, class `active` juga dihapus dari elemen `searchForm`.
  - Ini memastikan bahwa form pencarian akan tertutup ketika tombol `cart` diklik.
- c) Fungsi Keseluruhan :
- Ketika tombol `cart` dengan ID `cart-btn` diklik:
- Elemen keranjang belanja (`cartItem`) akan muncul atau tersembunyi (dengan menambah/menghapus class `active`).
  - Navigasi (`navbar`) akan dipastikan tertutup (class `active` dihapus).
  - Form pencarian (`searchForm`) juga akan dipastikan tertutup.



```

1 // Menambahkan event listener untuk 'scroll' pada window
2 window.onscroll = () => {
3   // Menghapus class 'active' dari elemen navbar saat pengguna melakukan scroll
4   navbar.classList.remove('active');
5
6   // Menghapus class 'active' dari elemen searchForm saat pengguna melakukan scroll
7   searchForm.classList.remove('active');
8
9   // Menghapus class 'active' dari elemen cartItem saat pengguna melakukan scroll
10  cartItem.classList.remove('active');
11 }

```

- a. `window.onscroll = () => { ... };`
  - `window.onscroll` adalah properti yang menambahkan event listener ke objek `window` yang memantau ketika pengguna menggulir halaman (scrolling).
  - Ketika pengguna menggulir, event listener ini akan memicu fungsi (arrow function) yang berisi instruksi-instruksi untuk dijalankan setiap kali scroll terjadi.
- b. `navbar.classList.remove('active');`
  - Saat halaman digulir, class `active` akan dihapus dari elemen `navbar`.
  - Ini memastikan bahwa navigasi (`navbar`) akan tersembunyi jika pengguna menggulir halaman, menutup menu navigasi yang mungkin terbuka.
- c. `searchForm.classList.remove('active');`
  - Sama seperti `navbar`, class `active` dihapus dari elemen `searchForm` saat pengguna menggulir.
  - Ini memastikan bahwa form pencarian (`searchForm`) akan tertutup ketika pengguna menggulir halaman.
- d. `cartItem.classList.remove('active');`
  - Class `active` juga dihapus dari elemen `cartItem` (keranjang belanja) saat pengguna menggulir.
  - Ini memastikan bahwa keranjang belanja akan tersembunyi ketika pengguna melakukan scrolling di halaman.
- d) Fungsi Kesleuruhan :  
Setiap kali pengguna menggulir halaman:
  - Navigasi (`navbar`), form pencarian (`searchForm`), dan keranjang belanja (`cartItem`) akan tertutup jika salah satu dari elemen tersebut terbuka (dengan menghapus class `active` dari elemen-elemen tersebut).

Daffa Fauzan H.	Website, Laporan
M. Imam Amirrudin	Laporan
Brian Yusuf	w