

LAPORAN PRAKTIKUM PEMROGRAMAN WEB

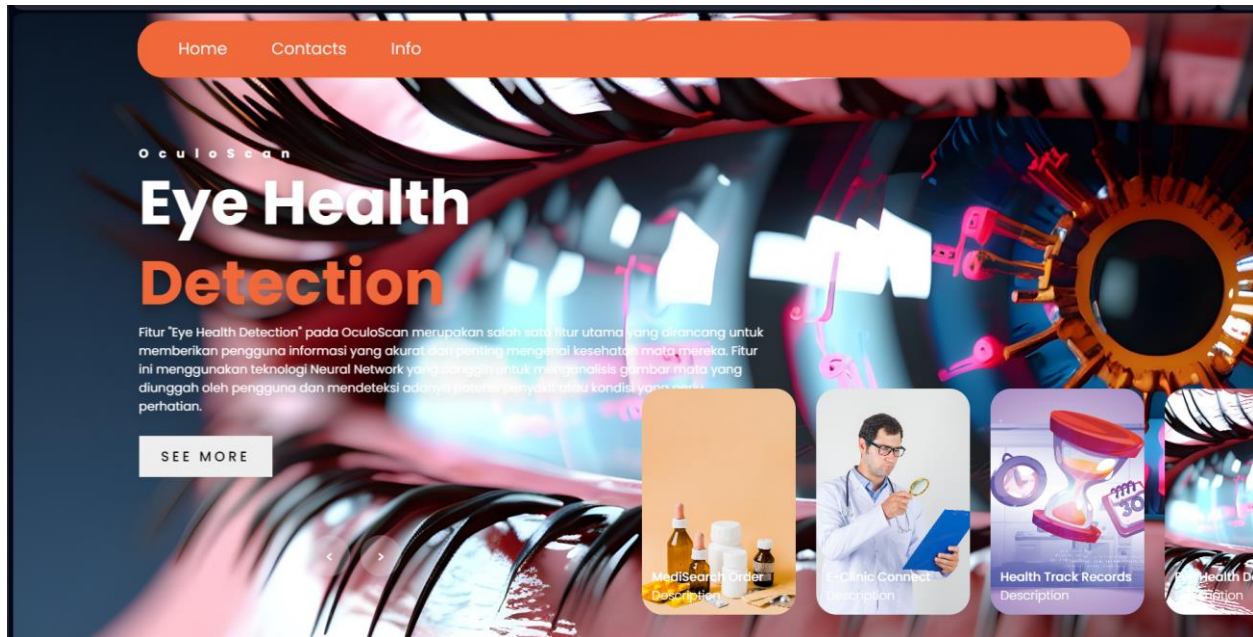
Week: 03



Nama: Eka Putra Meravigliosi
NIM: L0122151

PROGRAM STUDI INFORMATIKA FAKULTAS
TEKNOLOGI INFORMASI DAN SAINS DATA
UNIVERSITAS SEBELAS MARET 2023

Tampilan Awal Web



Judul:

"OculoScan Website"

Deskripsi:

Website ini adalah platform "OculoScan" yang menyediakan berbagai fitur kesehatan online untuk pengguna. Dengan desain yang responsif, pengguna dapat menjelajahi fitur-fitur utama yang ditawarkan oleh OculoScan, mulai dari deteksi kesehatan mata hingga pemesanan produk kesehatan, konsultasi medis virtual, dan pelacakan riwayat kesehatan.

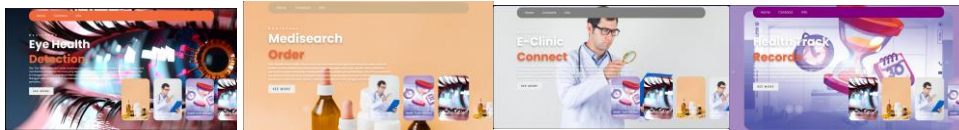
Fungsi-fungsi Web:

1. Navbar:



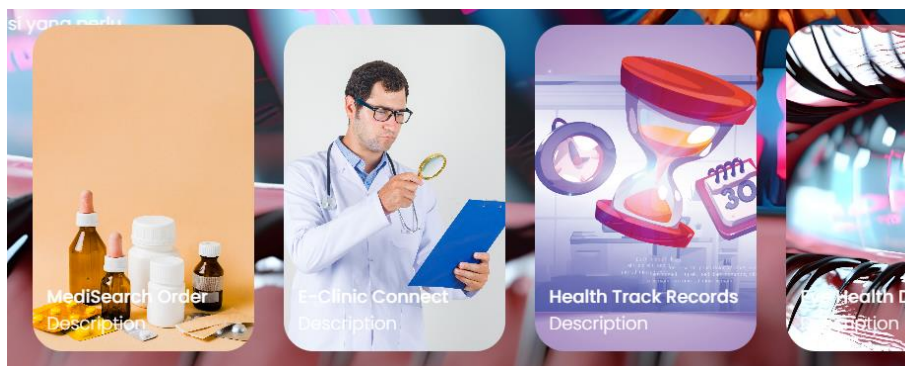
- Menyediakan navigasi sederhana antara halaman "Home", "Contacts", dan "Info".

2. Carousel Slide:



- Menampilkan berbagai fitur unggulan dari OculoScan dalam bentuk slide.
- Setiap slide berisi informasi singkat tentang fitur kesehatan yang ditawarkan, termasuk gambar representatif dan deskripsi fitur.
- Tombol "SEE MORE" pada setiap slide memungkinkan pengguna untuk mendapatkan informasi lebih lanjut.

3. Thumbnail Carousel:



- Menampilkan thumbnail dari setiap slide dengan judul fitur yang sesuai.
- Thumbnail ini memungkinkan pengguna untuk dengan cepat memilih fitur yang ingin mereka lihat tanpa harus menjelajahi seluruh slide.

4. Tombol "Prev" dan "Next":



- Memungkinkan pengguna untuk menggeser slide ke kiri ("Prev") atau ke kanan ("Next") untuk melihat informasi fitur lainnya.
- Memberikan kemudahan navigasi antara slide-slide yang ada.

5. Fitur Detail "SEE MORE":

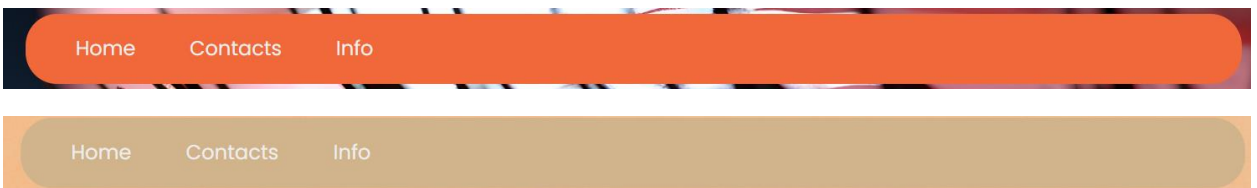


- Ketika tombol "SEE MORE" pada setiap slide di klik, pengguna dapat melihat informasi lebih lanjut tentang fitur yang dipilih.



- Ini membuka opsi untuk mempelajari detail tentang fitur, penggunaan, manfaat, dan cara mengaksesnya.

6. Background Color Navbar Dinamis:





- Navbar di bagian atas halaman akan mengubah warna sesuai dengan fitur yang dipilih oleh pengguna.
- Memberikan tanda visual yang jelas tentang fitur mana yang sedang dilihat pengguna.

7. Animasi Scroll ke Konten:

- Ketika tombol "See More" pada setiap slide di klik, halaman akan melakukan animasi scroll secara otomatis ke bagian konten yang terkait.
- Pengguna langsung diarahkan ke informasi lebih detail tentang fitur yang mereka pilih.

Penggunaan Javascript dalam Web:

1. Deklarasi elemen DOM

```
let nextDom = document.getElementById('next');
let prevDom = document.getElementById('prev');
let carouselDom = document.querySelector('.carousel');
let SliderDom = carouselDom.querySelector('.carousel .list');
let thumbnailBorderDom = document.querySelector('.carousel .thumbnail');
let thumbnailItemsDom = thumbnailBorderDom.querySelectorAll('.item');
```

- nextDom: Menyimpan tombol untuk menggeser carousel ke slide berikutnya.
- prevDom: Menyimpan tombol untuk menggeser carousel ke slide sebelumnya.
- carouselDom: Menyimpan container utama dari slideshow carousel dan thumbnailnya.
- SliderDom: Menyimpan daftar slide dalam carousel.
- thumbnailBorderDom: Menyimpan container untuk thumbnail dari setiap slide dalam carousel.
- thumbnailItemsDom: Menyimpan daftar semua elemen thumbnail dari setiap slide dalam carousel.

2. Deklarasi NextDom, Prev Dom

```
nextDom.onclick = function(){
    showSlider('next');
}

prevDom.onclick = function(){
    showSlider('prev');
}
```

1. **nextDom.onclick = function() {...}:**

- Ketika tombol "Next" (**nextDom**) diklik, fungsi **showSlider('next')** akan dijalankan.
- Ini berarti slideshow akan bergeser ke slide berikutnya.

2. **prevDom.onclick = function() {...}:**

- Ketika tombol "Previous" (**prevDom**) diklik, fungsi **showSlider('prev')** akan dijalankan.
- Ini berarti slideshow akan bergeser ke slide sebelumnya.

3. **Pengubah Background Thumbnail(thumbnailIndex)**

```
let currentIndex = 0;
let previousThumbnailIndex = 0;

function setCurrentIndexToFirst() {
    if (currentIndex === 0) {
        currentIndex = 0;
        console.log("Indeks Saat Ini:", currentIndex);
    }
}

setCurrentIndexToFirst();

thumbnailItemsDom.forEach(function(thumbnail, index) {
    thumbnail.addEventListener('click', function() {
        .....
    })
})
```

- Fungsi ini bertujuan untuk mengubah gambar latar belakang (background) dari carousel berdasarkan indeks thumbnail yang diterima.
- Setiap thumbnail dalam daftar memiliki indeks yang sesuai dengan gambar latar belakang yang akan ditampilkan pada carousel.
- Ketika pengguna mengklik thumbnail tertentu, fungsi ini akan mengubah latar belakang carousel sesuai dengan indeks thumbnail tersebut.

4. Fungsi Transisi Slide

```
function showSlider(type){
    let SliderItemsDom = SliderDom.querySelectorAll('.carousel .list .item');
    let thumbnailItemsDom = document.querySelectorAll('.carousel .thumbnail
.item');

    if(type === 'next'){
        SliderDom.appendChild(SliderItemsDom[0]);
        thumbnailBorderDom.appendChild(thumbnailItemsDom[0]);
        carouselDom.classList.add('next');
    } else {
        SliderDom.prepend(SliderItemsDom[SliderItemsDom.length - 1]);
        thumbnailBorderDom.prepend(thumbnailItemsDom[thumbnailItemsDom.length -
1]);
        carouselDom.classList.add('prev');
    }
    carouselDom.addEventListener('transitionend', function(){
        let activeSlide = SliderDom.querySelector('.item.active');
        if (activeSlide) {
            activeSlide.classList.remove('active');
        }
        let newActiveSlide = SliderDom.querySelector('.item');
        newActiveSlide.classList.add('active');
        carouselDom.classList.remove('next');
        carouselDom.classList.remove('prev');
    }, { once: true });
}
```

Fungsi showSlider(type) digunakan untuk menggeser slideshow carousel ke slide berikutnya atau sebelumnya, tergantung pada parameter type yang diberikan:

- **Jika type adalah 'next', maka:**
 - Slide pertama dipindahkan ke akhir daftar slide dengan `SliderDom.appendChild(SliderItemsDom[0])`.
 - Thumbnail pertama juga dipindahkan ke akhir daftar thumbnail dengan `thumbnailBorderDom.appendChild(thumbnailItemsDom[0])`.
 - Menambahkan kelas 'next' pada `carouselDom` untuk memberikan efek transisi.
- **Jika type bukan 'next' (misalnya, 'prev'), maka:**
 - Slide terakhir dipindahkan ke awal daftar slide dengan `SliderDom.prepend(SliderItemsDom[SliderItemsDom.length - 1])`.
 - Thumbnail terakhir juga dipindahkan ke awal daftar thumbnail dengan `thumbnailBorderDom.prepend(thumbnailItemsDom[thumbnailItemsDom.length - 1])`.
 - Menambahkan kelas 'prev' pada `carouselDom` untuk memberikan efek transisi ke arah sebelumnya.
- **Ketika transisi selesai**
 - Menghapus kelas 'active' dari slide yang sebelumnya aktif (`activeSlide`) menggunakan `activeSlide.classList.remove('active')`.
 - Menetapkan kelas 'active' pada slide yang baru aktif (`newActiveSlide`) menggunakan `newActiveSlide.classList.add('active')`.
 - Menghapus kelas 'next' dan 'prev' dari `carouselDom` untuk membersihkan efek transisi.

5. Fungsi Mendapatkan Index Slide{

```
function getCurrentSlideIndex() {
    let currentSlide = SliderDom.querySelector('.item');
    let currentIndex = Array.from(SliderDom.children).indexOf(currentSlide);

    if (typeof currentIndex === 'undefined' || currentIndex === 0) {
        currentIndex = 1;
    }

    return currentIndex;
}
```


- Pertama, fungsi mencari elemen slide yang sedang aktif dengan menggunakan **SliderDom.querySelector('.item')**.
- Kemudian, fungsi menghitung indeks dari elemen slide tersebut di dalam daftar slide dengan **Array.from(SliderDom.children).indexOf(currentSlide)**.
- Jika indeks tidak terdefinisi (**undefined**) atau indeksinya 0 (misalnya jika tidak ada slide yang aktif), maka fungsi akan mengembalikan nilai **1** sebagai indeks default.
- Akhirnya, fungsi mengembalikan nilai **currentIndex**, yang merupakan indeks dari slide yang sedang aktif dalam carousel.

6. Pengubah Background Slide

```
let firstSlide = SliderDom.querySelector('.item');
firstSlide.classList.add('active');

function changeCarouselBackground(thumbnailIndex) {
  let imageNames = ['img1.jpg', 'img2.jpg', 'img3.jpg', 'img4.jpg'];

  let imageUrl = `image/${imageNames[thumbnailIndex]}`;
  carouselDom.style.backgroundImage = `url('${imageUrl}')`;
}
```

Slide pertama dari slideshow carousel (**firstSlide**) ditetapkan sebagai slide yang aktif dengan menambahkan kelas **'active'** menggunakan **firstSlide.classList.add('active')**.

- Fungsi **changeCarouselBackground(thumbnailIndex)** digunakan untuk mengubah latar belakang carousel berdasarkan thumbnail yang dipilih.
- Terdapat array **imageNames** yang berisi nama-nama file gambar untuk setiap thumbnail.
- Ketika fungsi dipanggil dengan **thumbnailIndex** sebagai argumen, variabel **imageUrl** akan diisi dengan path ke gambar yang sesuai dengan thumbnail tersebut.
- Kemudian, latar belakang dari carousel (**carouselDom**) diubah dengan menggunakan **carouselDom.style.backgroundImage**.

- Ini memungkinkan tampilan latar belakang carousel berubah sesuai dengan thumbnail yang dipilih oleh pengguna.

7. Pengubah Warna Navbar(thumbnailIndex)

```
function changeNavbarColor(thumbnailIndex) {

    var navbar = document.getElementById('headernav');

    switch(thumbnailIndex) {
        case 1:
            navbar.style.backgroundColor = "#f1683a"; // Warna merah muda
            break;
        case 2:
            navbar.style.backgroundColor = "#d2b48c"; // Warna coklat muda
            break;
        case 3:
            navbar.style.backgroundColor = "#808080"; // Warna abu-abu
            break;
        case 4:
            navbar.style.backgroundColor = "#800080"; // Warna ungu muda
            break;
        default:
            navbar.style.backgroundColor = "#f1683a";
    }

    thumbnailItemsDom.forEach(function(thumbnail, index) {
        thumbnail.addEventListener('click', function() {
            let thumbnailIndex = Array.from(thumbnailItemsDom).indexOf(thumbnail);

            changeNavbarColor(thumbnailIndex + 1); // Tambahkan +1 karena indeks
            // dimulai dari 0
        });
    });
}
```

Fungsi `changeNavbarColor(thumbnailIndex)` digunakan untuk mengubah warna background navbar berdasarkan thumbnail yang dipilih:

- Fungsi ini menerima parameter `thumbnailIndex` yang menentukan warna yang akan diubah.
- Dengan menggunakan pernyataan `switch`, fungsi akan menetapkan warna background navbar berdasarkan nilai dari `thumbnailIndex`.

- Jika thumbnailIndex adalah 1, navbar akan memiliki background warna merah muda (#f1683a).
- Jika thumbnailIndex adalah 2, navbar akan memiliki background warna coklat muda (#d2b48c).
- Jika thumbnailIndex adalah 3, navbar akan memiliki background warna abu-abu (#808080).
- Jika thumbnailIndex adalah 4, navbar akan memiliki background warna ungu muda (#800080).
- Jika thumbnailIndex tidak cocok dengan kasus-kasus di atas, maka navbar akan memiliki background warna merah muda (#f1683a) sebagai default.

8. ScrollAtasThumbnail

```
document.addEventListener("DOMContentLoaded", function() {
    // Cari semua thumbnail
    let thumbnails = document.querySelectorAll('.thumbnail .item');

    thumbnails.forEach(function(thumbnail) {
        thumbnail.addEventListener('click', function() {
            // Atur animasi scroll ke bagian atas halaman
            window.scrollTo({
                top: 0,
                behavior: "smooth"
            });
        });
    });
});
```

Kode di atas adalah cara untuk menambahkan event listener pada semua thumbnail dalam halaman. Ketika thumbnail tersebut diklik oleh pengguna, halaman akan di-scroll secara halus (smooth scroll) kembali ke bagian atas halaman.

9. Scroll bawah dan hide extra page sesuai Index

```
document.addEventListener("DOMContentLoaded", function() {
    // Cari tombol "See More" di setiap daftar
    let seeMoreButtons = document.querySelectorAll('.carousel .item .buttons
button');
    let previousContainerCard = null; // variabel untuk menyimpan containerCard
    sebelumnya

    seeMoreButtons.forEach(function(button, index) {
        button.addEventListener('click', function() {
            // Dapatkan nomor daftar dari indeks (dimulai dari 0)
            let listNumber = index + 1;

            // Dapatkan kontainer card sesuai dengan nomor daftar
            let containerCard =
document.querySelector(`.containerCard${listNumber}`);

            // Tambahkan kelas "hidden" pada semua containerCard
            document.querySelectorAll('.containerCard').forEach(function(card) {
                card.classList.add('hidden');
            });

            // Hapus kelas "hidden" pada containerCard yang sesuai dengan nomor
daftar
            containerCard.classList.remove('hidden');

            // Dapatkan posisi offset dari containerCard yang sesuai
            let containerCardOffset = containerCard.offsetTop;

            // Atur animasi scroll ke posisi containerCard yang sesuai
            window.scrollTo({
                top: containerCardOffset,
                behavior: "smooth"
            });

            // Tampilkan nomor daftar yang dipilih di konsol
            console.log("Daftar yang dipilih:", listNumber);

            // Tambahkan kelas "hidden" pada .containerCard sebelumnya
            if (previousContainerCard && previousContainerCard !== containerCard)
            {
                previousContainerCard.classList.add('hidden');
            }
        });
    });
});
```

```

        previousContainerCard = containerCard;
    });
});
});

```

- Saat dokumen telah selesai dimuat ("DOMContentLoaded" event), maka kode berikut akan dijalankan:
- Mengambil semua tombol "See More" pada setiap item daftar di carousel dengan `document.querySelectorAll('.carousel .item .buttons button')`.
- Setiap tombol "See More" akan memiliki event listener yang akan dijalankan ketika diklik.
- Ketika tombol "See More" diklik, fungsi akan:
 - Mendapatkan nomor dari daftar yang diklik dengan `let listNumber = index + 1`.
 - Mengambil kontainer card yang sesuai dengan nomor daftar dengan `let containerCard = document.querySelector('.containerCard${listNumber}')`.
 - Menambahkan kelas "hidden" pada semua kontainer card untuk menyembunyikannya dengan `document.querySelectorAll('.containerCard').forEach(...)`.
 - Menghapus kelas "hidden" pada kontainer card yang sesuai dengan nomor daftar untuk menampilkannya dengan `containerCard.classList.remove('hidden')`.
 - Melakukan animasi scroll halus ke posisi kontainer card yang sesuai dengan `window.scrollTo({ top: containerCardOffset, behavior: "smooth" })`.
 - Menampilkan nomor dari daftar yang dipilih di konsol dengan `console.log("Daftar yang dipilih:", listNumber)`.
 - Menambahkan kelas "hidden" pada kontainer card sebelumnya (jika ada) untuk menyembunyikannya dengan `previousContainerCard.classList.add('hidden')`.
 - Menyimpan kontainer card saat ini sebagai `previousContainerCard` untuk referensi selanjutnya.

Dengan ini, saat pengguna mengklik tombol "See More" pada setiap daftar di carousel, konten yang sesuai akan ditampilkan dengan efek scroll halus ke posisi konten tersebut. Selain itu, konten sebelumnya akan disembunyikan.