

1. Struktur Dasar HTML

- `<!DOCTYPE html>` → Menentukan bahwa dokumen ini menggunakan HTML5.
- `<html lang="en">` → Menentukan bahasa dokumen sebagai bahasa Inggris.
- `<head>` → Berisi metadata, termasuk charset, viewport, dan referensi ke file eksternal.
- `<body>` → Bagian utama yang menampilkan model 3D dan elemen UI lainnya.

2. Bagian `<head>`

a. Metadata

Kode:

```
html
CopyEdit
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>3D Model Viewer</title>
```

Penjelasan:

- `<meta charset="UTF-8">` → Mengatur encoding karakter ke UTF-8.
- `<meta name="viewport" content="width=device-width, initial-scale=1.0">`
→ Membuat tampilan responsif di berbagai perangkat.
- `<title>3D Model Viewer</title>` → Menentukan judul halaman.

3. Styling dengan CSS

a. Styling `body`

Kode:

```
css
CopyEdit
body {
  margin: 0;
  font-family: Arial, sans-serif;
  background-color: #f4f4f4;
  height: 100vh;
  overflow: hidden;
  display: flex;
  flex-direction: column;
}
```

Penjelasan:

- `margin: 0;` → Menghapus margin bawaan browser.
- `font-family: Arial, sans-serif;` → Menggunakan font Arial atau alternatif sans-serif.
- `background-color: #f4f4f4;` → Warna latar belakang abu-abu terang.
- `height: 100vh;` → Menggunakan tinggi penuh dari viewport.
- `overflow: hidden;` → Menyembunyikan scroll jika ada elemen yang melampaui ukuran layar.
- `display: flex; flex-direction: column;` → Menyusun elemen secara vertikal.

b. Container Model 3D

Kode:

```
css
CopyEdit
#model-container {
  width: 100%;
  height: calc(100vh - 160px);
  background: #fff;
  position: relative;
}
```

Penjelasan:

- `width: 100%;` → Lebar penuh.
- `height: calc(100vh - 160px);` → Menyesuaikan tinggi agar tidak terlalu memenuhi layar.
- `background: #fff;` → Latar belakang putih.
- `position: relative;` → Menyesuaikan posisi relatif terhadap elemen lain.

c. Tombol Pilihan Model

Kode:

```
css
CopyEdit
#controls {
  position: absolute;
  bottom: 0;
  left: 50%;
  transform: translateX(-50%);
  display: flex;
  gap: 10px;
  justify-content: center;
  z-index: 3;
}
```

Penjelasan:

- `position: absolute; bottom: 0;` → Menempatkan elemen di bagian bawah.

- `left: 50%; transform: translateX(-50%);` → Memastikan elemen berada di tengah.
- `display: flex; justify-content: center;` → Menyusun elemen secara horizontal.

d. Tombol Home

Kode:

```
css
CopyEdit
.home-button {
  position: absolute;
  top: 10px;
  left: 10px;
  display: flex;
  align-items: center;
  background-color: #fff;
  border: 2px solid #000;
  border-radius: 15px;
  padding: 5px 10px;
  text-decoration: none;
  color: #000;
  font-size: 16px;
  font-weight: bold;
  box-shadow: 2px 2px 5px rgba(0, 0, 0, 0.2);
  z-index: 1000;
}
```

Penjelasan:

- `position: absolute; top: 10px; left: 10px;` → Menempatkan tombol di sudut kiri atas.
- `display: flex; align-items: center;` → Membuat tampilan lebih fleksibel.
- `background-color: #fff;` → Latar belakang putih.
- `border: 2px solid #000;` → Garis tepi hitam.
- `border-radius: 15px;` → Sudut melengkung.
- `padding: 5px 10px;` → Memberikan ruang dalam tombol.
- `box-shadow: 2px 2px 5px rgba(0, 0, 0, 0.2);` → Efek bayangan untuk tampilan lebih elegan.

4. JavaScript untuk 3D Viewer

Kode:

```
js
CopyEdit
import * as THREE from
"https://cdn.skypack.dev/three@0.129.0/build/three.module.js";
import { GLTFLoader } from
"https://cdn.skypack.dev/three@0.129.0/examples/jsm/loaders/GLTFLoader.js";
```

```
import { OrbitControls } from
"https://cdn.skypack.dev/three@0.129.0/examples/jsm/controls/OrbitControls.js
";
```

Penjelasan:

- **THREE.js** digunakan untuk menampilkan model 3D.
- **GLTFLoader** digunakan untuk memuat model dalam format .gltf.
- **OrbitControls** memungkinkan pengguna memutar model 3D dengan mouse.

a. Inisialisasi Scene dan Kamera

Kode:

```
js
CopyEdit
let scene, camera, renderer, controls;
scene = new THREE.Scene();
camera = new THREE.PerspectiveCamera(75, window.innerWidth /
window.innerHeight, 0.1, 1000);
renderer = new THREE.WebGLRenderer({ alpha: true });

const container = document.getElementById('model-container');
renderer.setSize(container.offsetWidth, container.offsetHeight);
container.appendChild(renderer.domElement);
```

Penjelasan:

- `scene = new THREE.Scene();` → Membuat scene 3D.
- `camera = new THREE.PerspectiveCamera(75, window.innerWidth / window.innerHeight, 0.1, 1000);` → Mengatur kamera dengan sudut pandang 75 derajat.
- `renderer = new THREE.WebGLRenderer({ alpha: true });` → Membuat renderer WebGL dengan latar belakang transparan.
- `renderer.setSize(container.offsetWidth, container.offsetHeight);` → Mengatur ukuran renderer sesuai dengan container.

b. Memuat Model 3D

Kode:

```
js
CopyEdit
function loadModel(modelKey) {
  Object.values(models).forEach(model => {
    scene.remove(model);
  });

  if (!models[modelKey]) {
    const { path, scale, title, description } = modelData[modelKey];
```

```

const loader = new GLTFLoader();
loader.load(
  path,
  (gltf) => {
    const model = gltf.scene;
    model.scale.set(...scale);
    model.rotation.y = Math.PI;
    models[modelKey] = model;
    scene.add(model);
  },
  undefined,
  (error) => console.error('Error loading model:', error)
);
} else {
  scene.add(models[modelKey]);
}

const details = document.getElementById('details');
details.querySelector('h1').textContent = modelData[modelKey].title;
details.querySelector('p').textContent = modelData[modelKey].description;
}

```

Penjelasan:

- **Menghapus model lama** sebelum menambahkan model baru.
- **Menggunakan GLTFLoader** untuk memuat model berdasarkan path.
- **Menampilkan detail model** di bagian deskripsi.