**LAPORAN PRAKTIKUM PERTEMUAN KE-4**

**JavaScript – Aplikasi To-Do List**

*(Disusun Untuk Tugas Kuliah Praktikum Desain Web (A))*

**

**Dosen Pembimbing :**

**Adi Wahyu Pribadi , S.Si., M.Kom.**

**Disusun Oleh :**

**Kyara Azwa Humayra**

**(4523210058)**

**1 OKTOBER 2024**

**PRODI TEKNIK INFORMATIKA**

**FAKULTAS TEKNIK**

**UNIVERSITAS PANCASILA**

**JAKARTA**

**BAB I**

**Pendahuluan**

* 1. **Tujuan Praktikum**

Dapat mengetahui elemen HTML, belajar cara membuat struktur data menggunkan kelas dan objek, mengetahui bagaimana cara menangani interaksi dari pengguna, serta belajar cara menggunakan JavaScript DOM untuk menambahkan, menghapus, dan memperbarui elemen.

* 1. **Deskripsi Aplikasi To-Do List**

Mahasiswa diminta untuk membuat aplikasi manajemen daftar tugas yang interaktif yang menggunakan JavaScript. Aplikasi ini akan memungkinkan pengguna untuk menambahkan, menandai tugas sebagai selesai, dan menghapus tugas dari daftar. Komponen ini akan menguji pemahaman siswa tentang manipulasi DOM, pengelolaan event, penggunaan class dan object, dan fungsi.

* 1. **Teori Dasar**

JavaScript adalah bahasa pemrograman yang digunakan untuk membuat halaman web

interaktif. Ini adalah bahasa pemrograman yang berbasis prototipe dan bersifat dinamis.

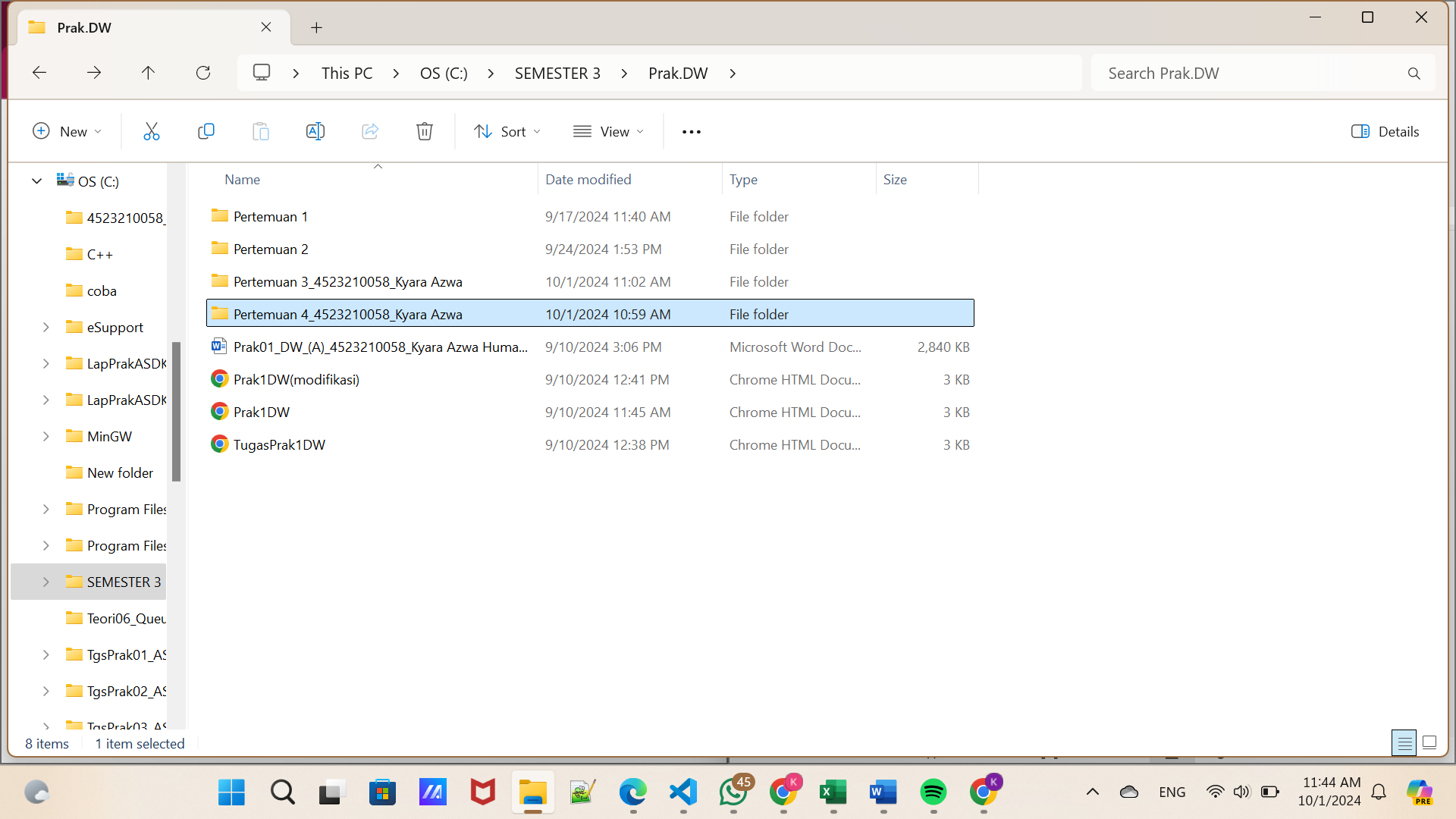
**BAB II**

**Langkah pengerjaan**

**2.1. Tahapan Praktikum**

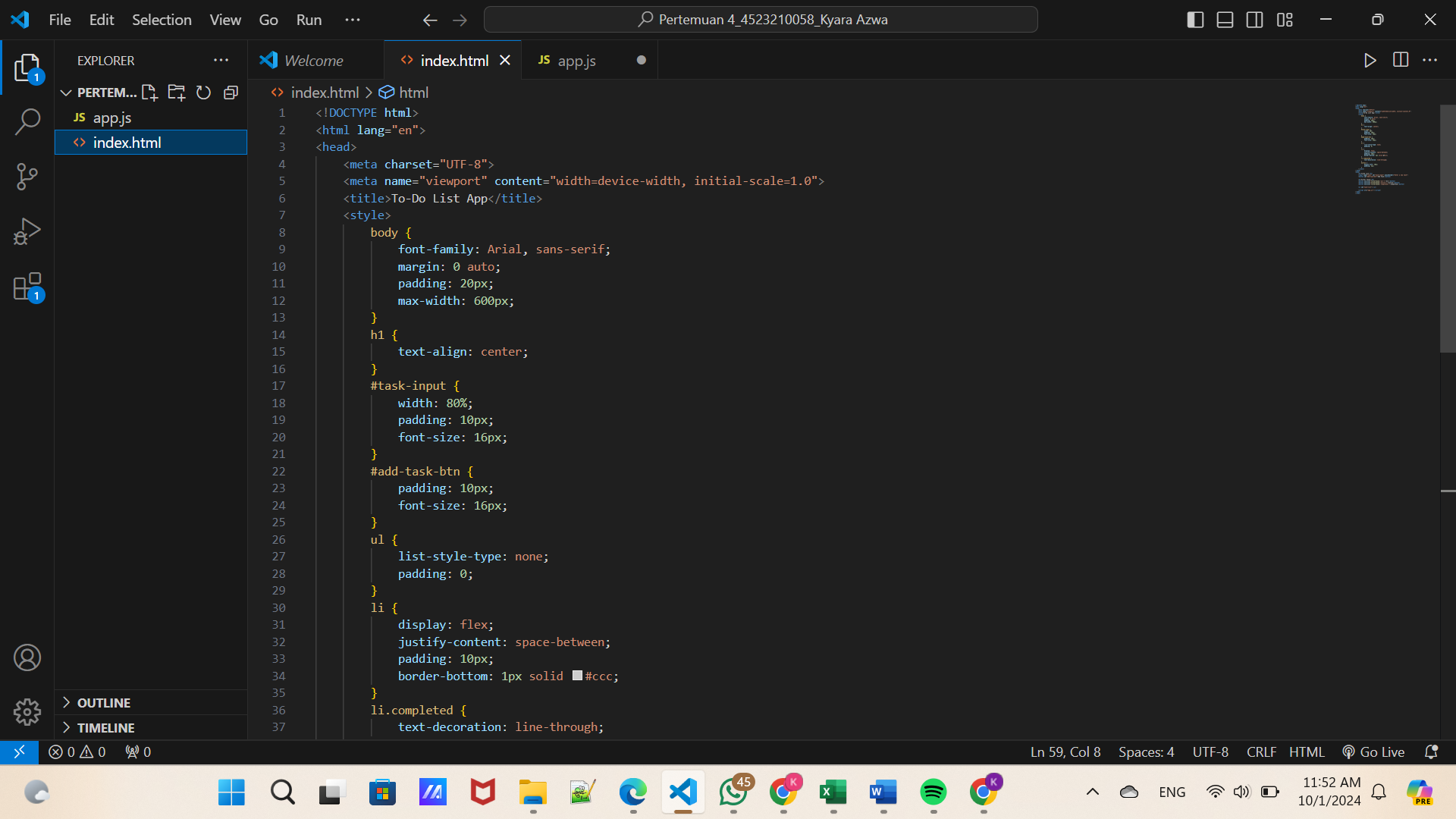
1. Membuat folder file

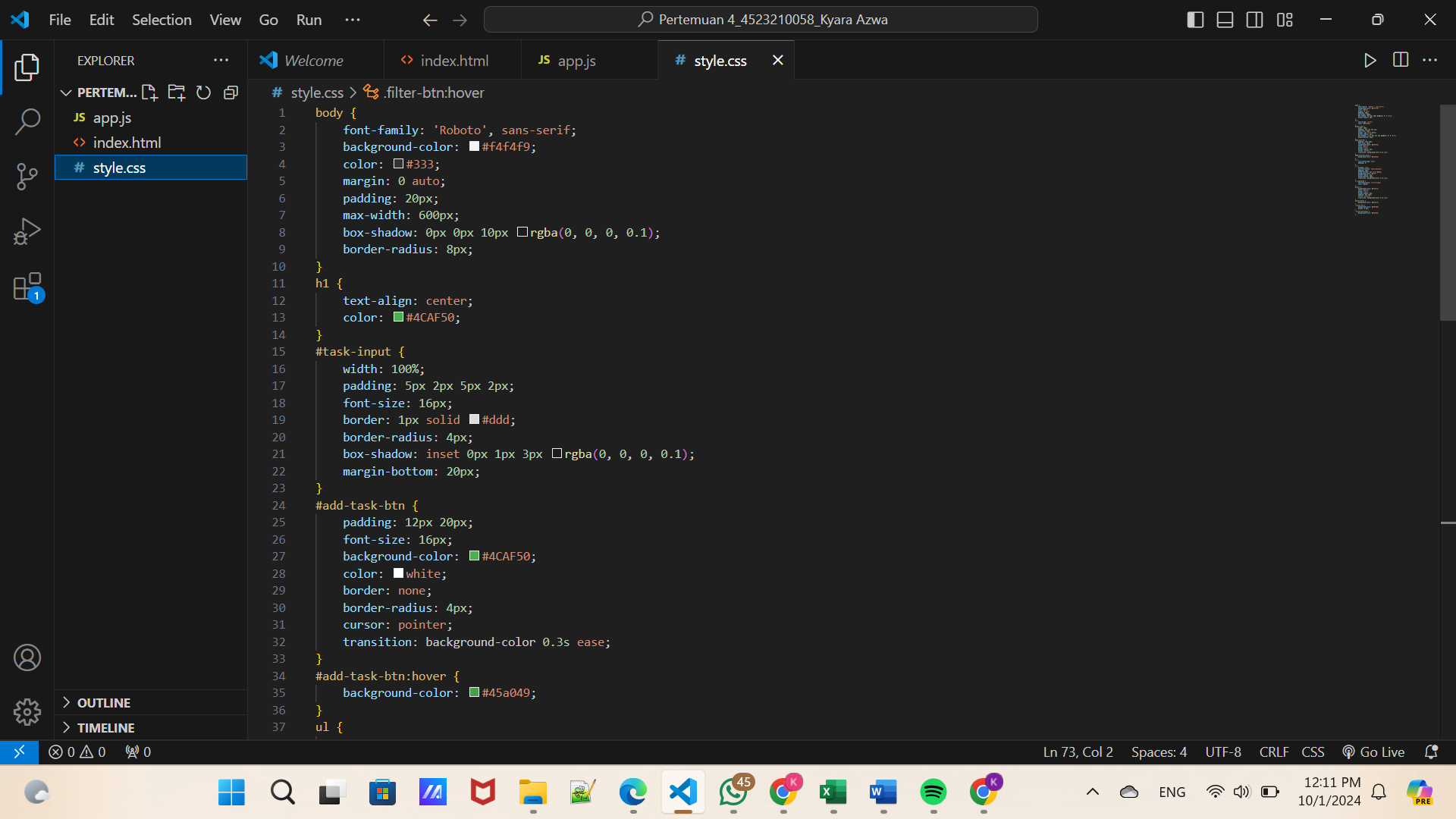
Folder untuk menyimpan file HTML,CSS, dan JavaScript



1. Membuat indeks HTML dan Styling dengan CSS

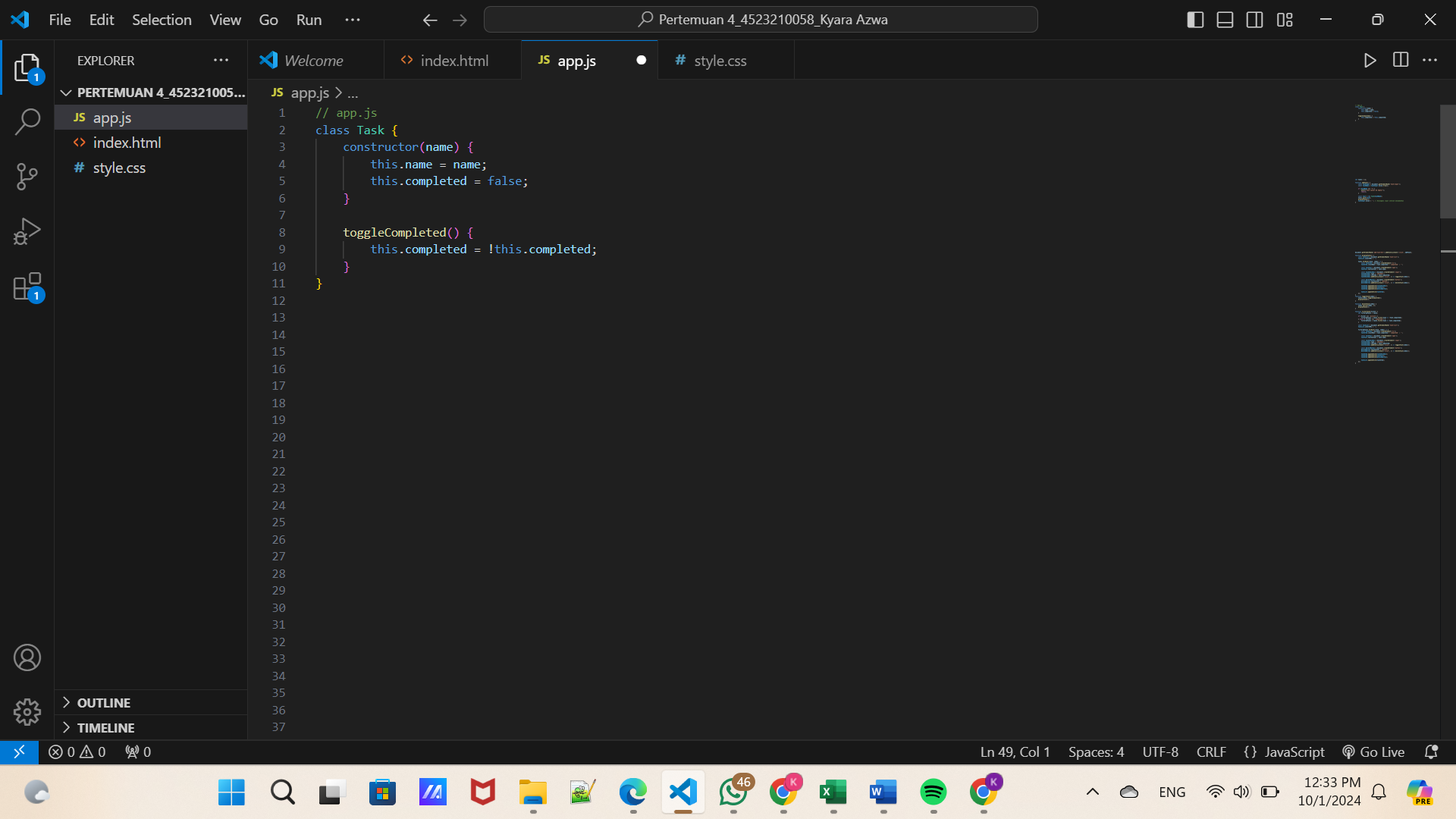
Membuat struktur HTML dasar dan menggunankan CSS untuk styling sehingga tampak menarik dan interaktif





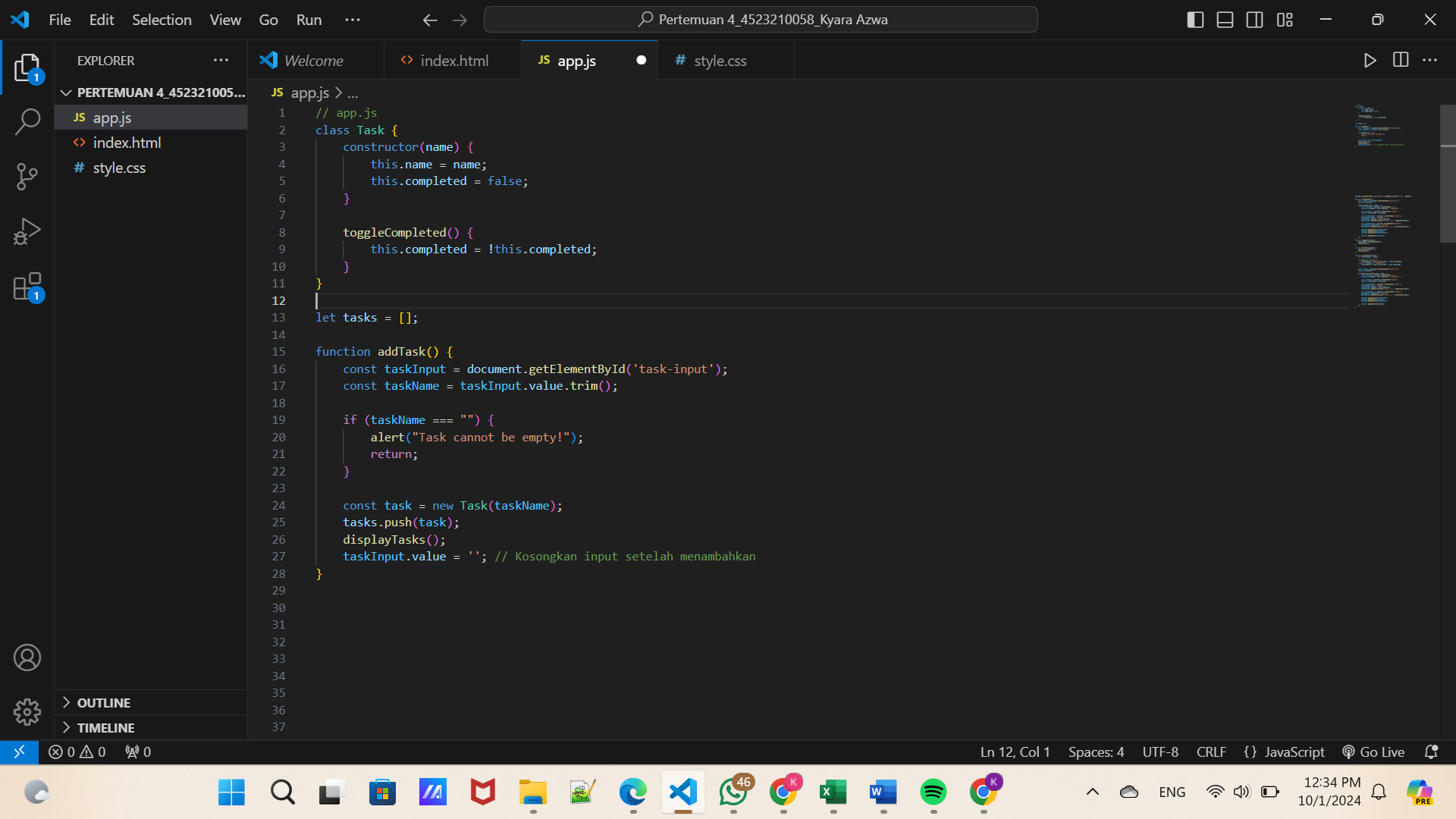
1. Class dan Object untuk tugas

Clas Task untuk mendefinisikan objek tugas dengan properti DOM



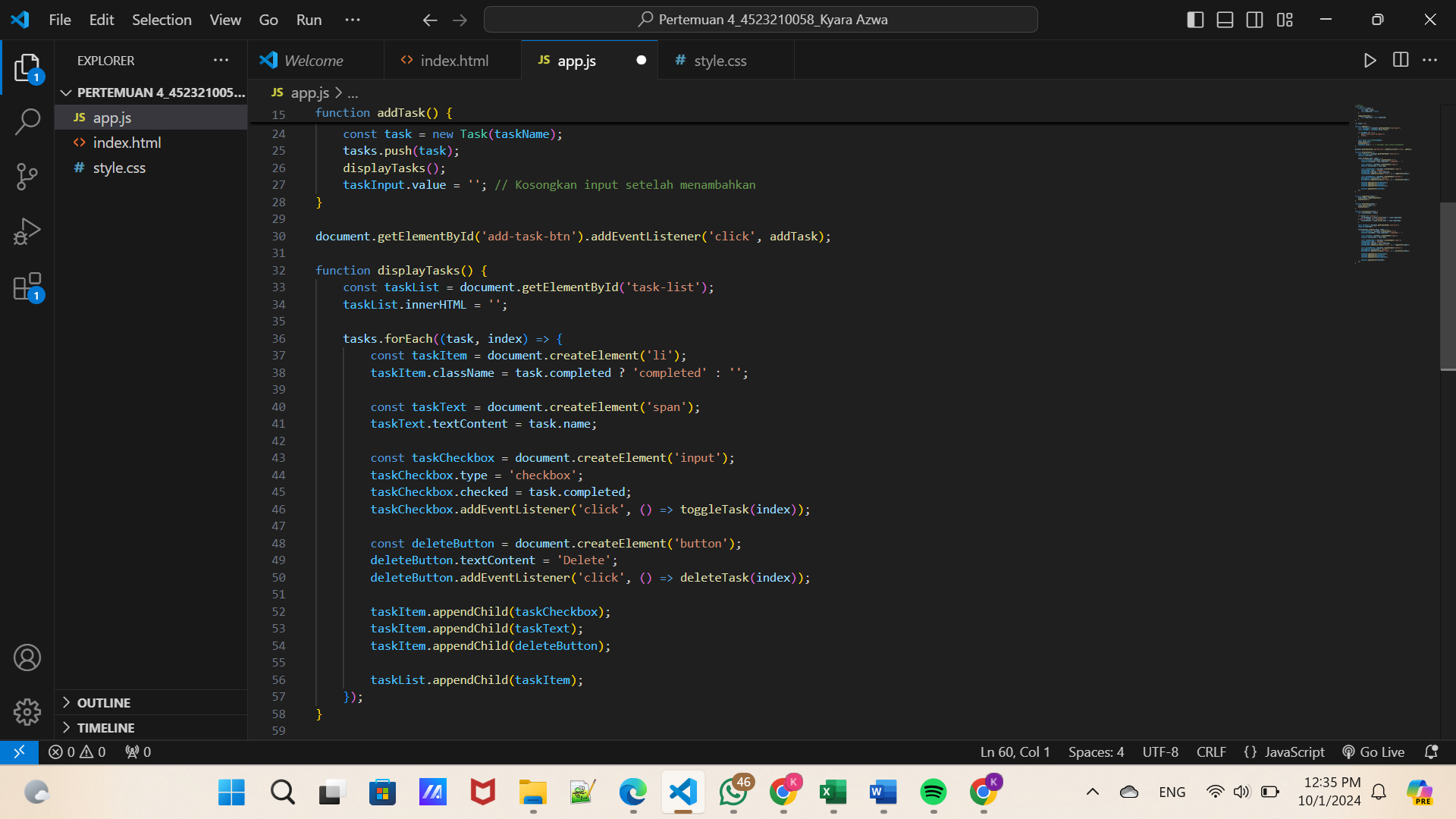
1. Membuat fungsi untuk menambahkan tugas

Fungsi untuk menerima input pengguna dan menambahkan objek



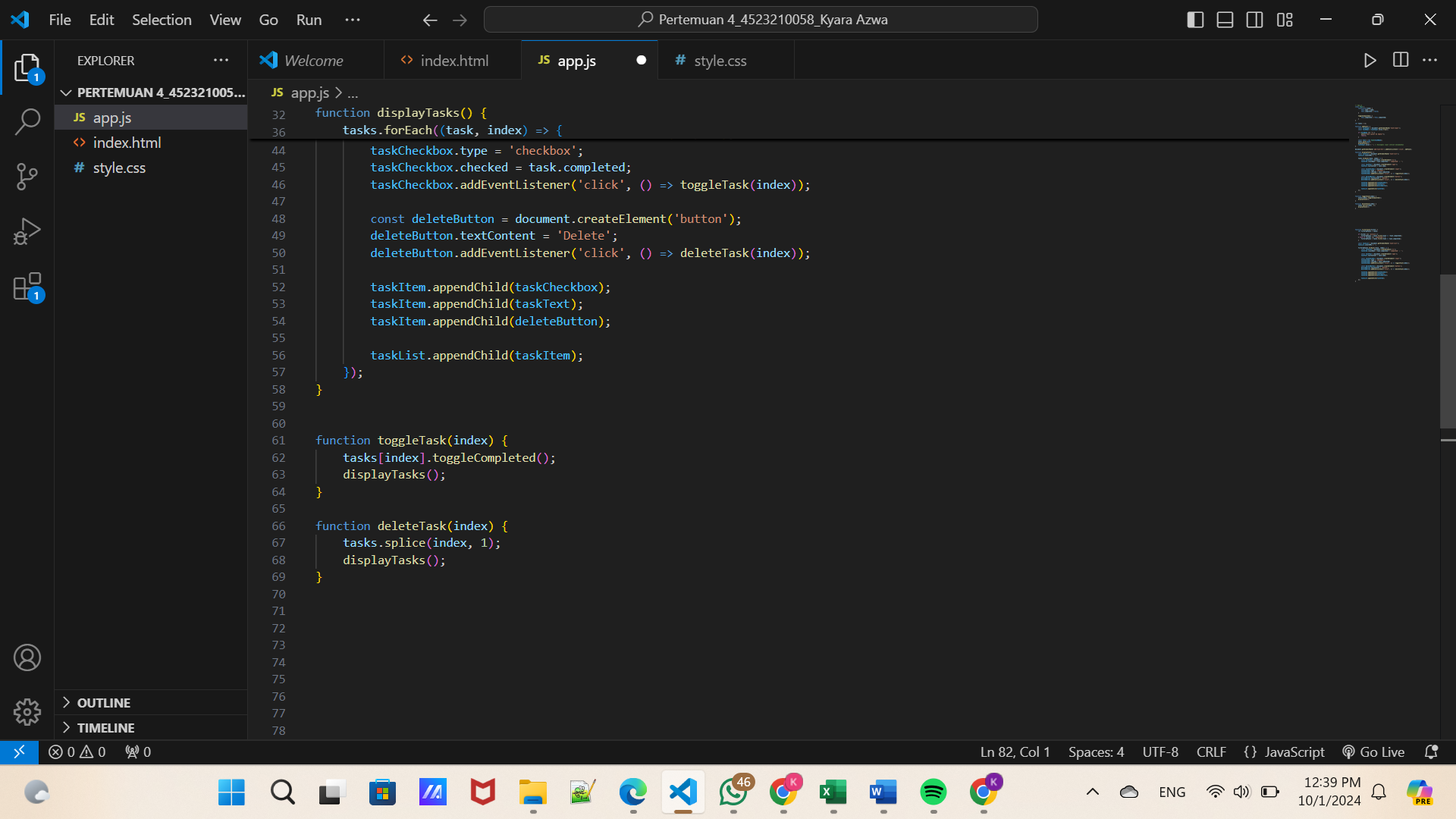
1. Menampilkan Tugas di DOM

Menggunakan DOM manipulation untuk menampilkan daftar tugas di halaman web.

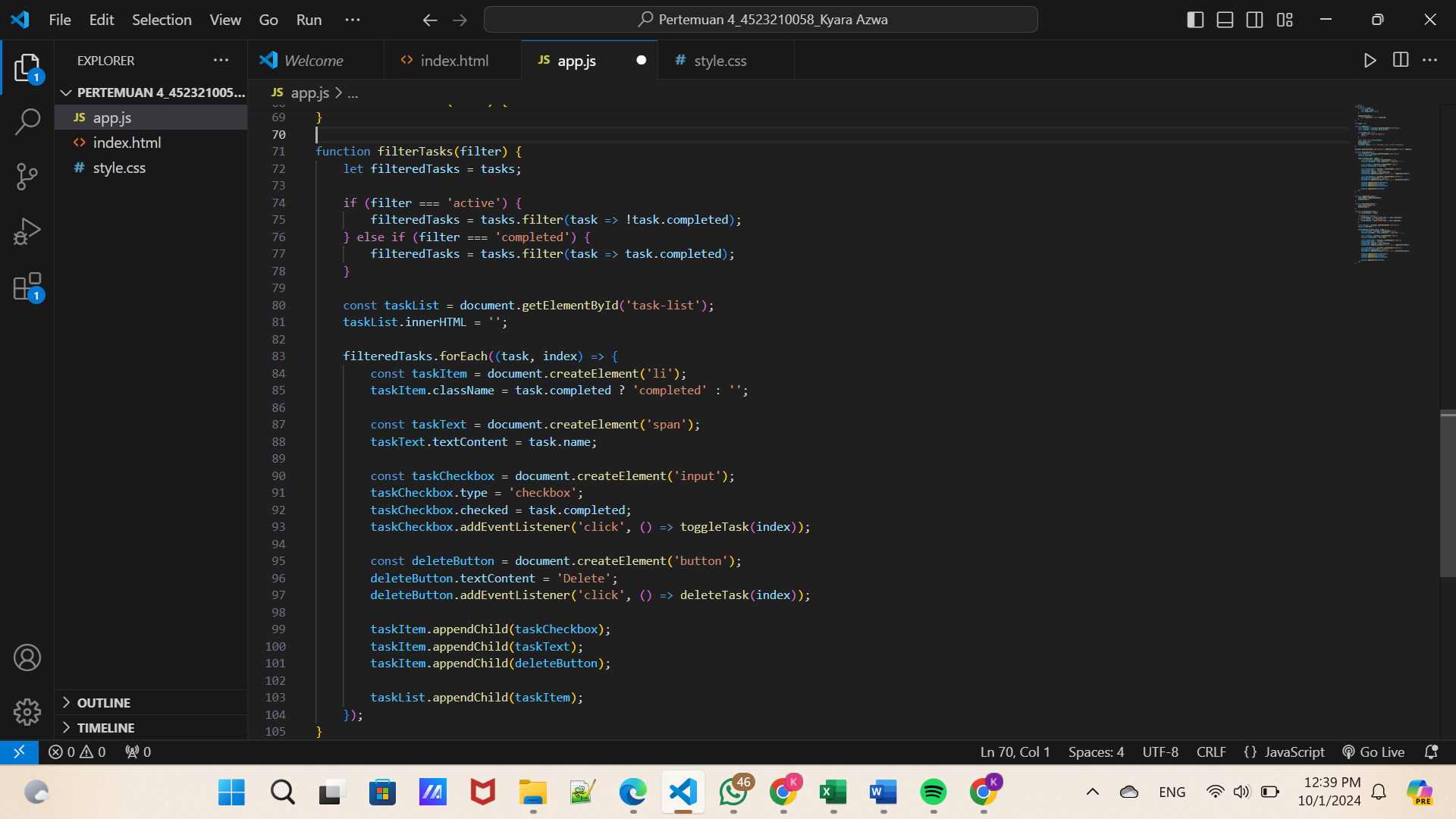


1. Menandai Tugas Selasai dan menghapus tugas

Menambahkan fitur untuk menandai tugas dan menghapus tugas dari daftar



1. Filter tugas bedasarkan status

Menampilkan tugas bedasarkan status 

**BAB III**

**Struktur dan Penjelasan Kode**

**3.1. Kode JavaScript**

// app.js

class Task {

    constructor(name) {

        this.name = name;

        this.completed = false;

    }

    toggleCompleted() {

        this.completed = !this.completed;

    }

}

let tasks = [];

function addTask() {

    const taskInput = document.getElementById('task-input');

    const taskName = taskInput.value.trim();

    if (taskName === "") {

        alert("Task cannot be empty!");

        return;

    }

    const task = new Task(taskName);

    tasks.push(task);

    displayTasks();

    taskInput.value = ''; // Kosongkan input setelah menambahkan

}

document.getElementById('add-task-btn').addEventListener('click', addTask);

function displayTasks() {

    const taskList = document.getElementById('task-list');

    taskList.innerHTML = '';

    tasks.forEach((task, index) => {

        const taskItem = document.createElement('li');

        taskItem.className = task.completed ? 'completed' : '';

        const taskText = document.createElement('span');

        taskText.textContent = task.name;

        const taskCheckbox = document.createElement('input');

        taskCheckbox.type = 'checkbox';

        taskCheckbox.checked = task.completed;

        taskCheckbox.addEventListener('click', () => toggleTask(index));

        const deleteButton = document.createElement('button');

        deleteButton.textContent = 'Delete';

        deleteButton.addEventListener('click', () => deleteTask(index));

        taskItem.appendChild(taskCheckbox);

        taskItem.appendChild(taskText);

        taskItem.appendChild(deleteButton);

        taskList.appendChild(taskItem);

    });

}

function toggleTask(index) {

    tasks[index].toggleCompleted();

    displayTasks();

}

function deleteTask(index) {

    tasks.splice(index, 1);

    displayTasks();

}

function filterTasks(filter) {

    let filteredTasks = tasks;

    if (filter === 'active') {

        filteredTasks = tasks.filter(task => !task.completed);

    } else if (filter === 'completed') {

        filteredTasks = tasks.filter(task => task.completed);

    }

    const taskList = document.getElementById('task-list');

    taskList.innerHTML = '';

    filteredTasks.forEach((task, index) => {

        const taskItem = document.createElement('li');

        taskItem.className = task.completed ? 'completed' : '';

        const taskText = document.createElement('span');

        taskText.textContent = task.name;

        const taskCheckbox = document.createElement('input');

        taskCheckbox.type = 'checkbox';

        taskCheckbox.checked = task.completed;

        taskCheckbox.addEventListener('click', () => toggleTask(index));

        const deleteButton = document.createElement('button');

        deleteButton.textContent = 'Delete';

        deleteButton.addEventListener('click', () => deleteTask(index));

        taskItem.appendChild(taskCheckbox);

        taskItem.appendChild(taskText);

        taskItem.appendChild(deleteButton);

        taskList.appendChild(taskItem);

    });

}

**3.2. Penjelasan Singkat Kode**

* **Class Task dan fungsinya.**

Class Task adalah untuk objek tugas di aplikasi To-Do List. Fungsinya membuat objek tugas baru dengan menginput nama tugas serta mengelola/mengubah status tugas.

* **Fungsi addTask, toggleTask, deleteTask, dan filterTasks.**

Fungsi addTask mengambil input dari pengguna, menambahkan objek task dan memanggil displayTask() agar meng-update tampilan daftar tuga dan input dikosongkan. Fungsi deleteTask untuk menghapus tugas dari daftar. Fungsi filterTasks adalah memfilter tugas bedasrakan status nya berjalan atau selesai dan memperbarui halaman dengan tugas yang difilter

* **Cara penggunaan manipulasi DOM untuk menampilkan tugas.**

displayTasks() untuk mengosongkan elemen daftar tugas, atau task-list, kemudian membuat elemen <li> untuk setiap tugas. Tambahkan elemen checkbox, teks tugas, dan tombol hapus ke dalam elemen <li>, dan kemudian menambahkannya ke dalam daftar.

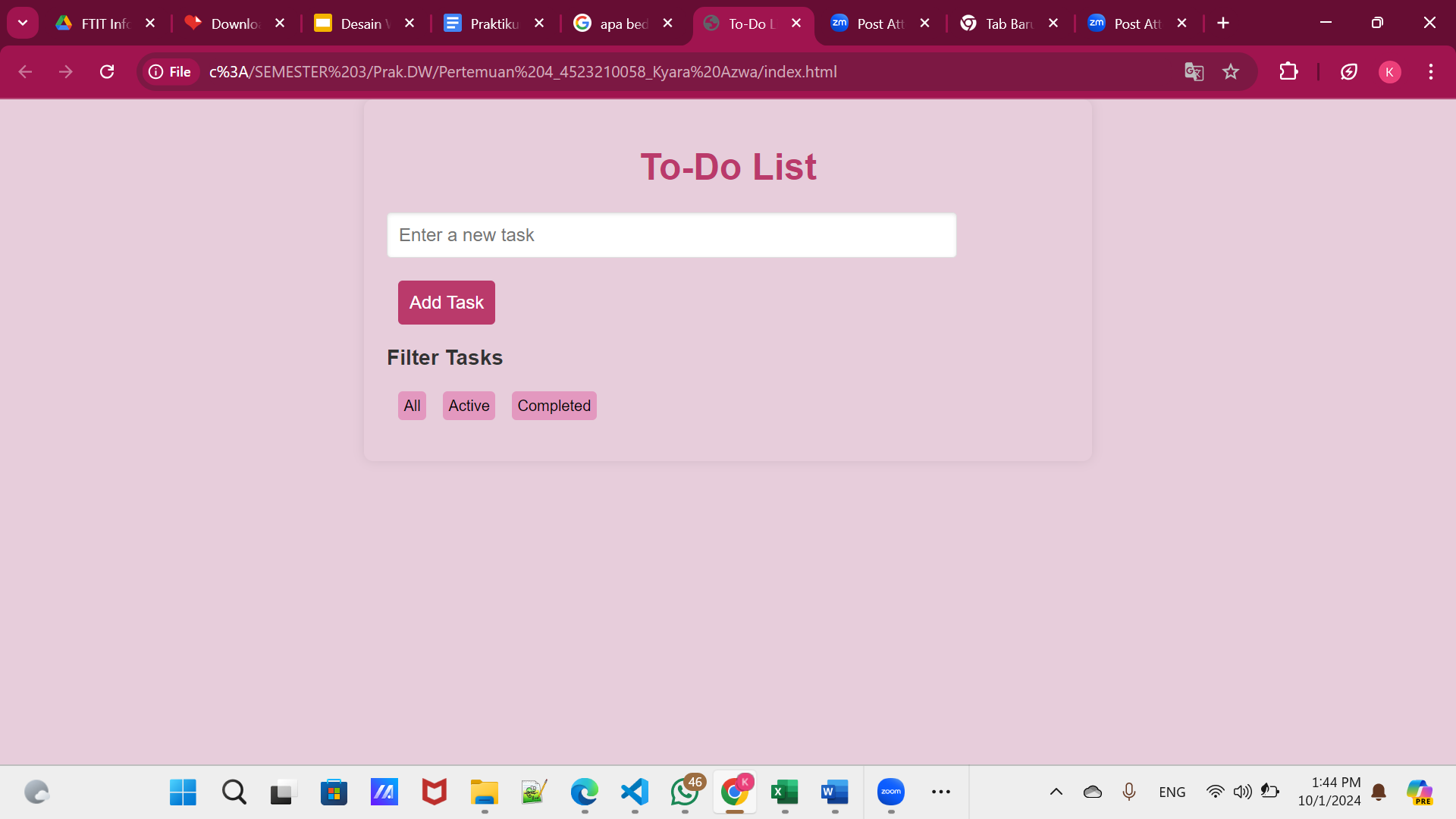
* **Penjelasan event handling yang digunakan, seperti pada tombol “Add Task” dan checkbox.**

Tombol “Add Task” digunakan untuk addEventListener untuk menambahkan event click pada tombol. Ketika tombol diklik, fungsi addTask() akan dipanggil. Checkbox memiliki event listener yang memanggil toggleTask(index) ketika dicentang atau dicopot. Ini memungkinkan pengguna untuk menandai tugas sebagai selesai atau tidak.

**BAB IV**

**Penutup**

**4.1. Hasil Ujicoba**



Aplikasi To-Do List ini merupakan aplikasi untuk memanajemen tugas yang memunggkinkan pengguna untuk menambahkan tugas, memfilter, dan memgelola tugas-tugas dari pengguna.

Fitur-fitur yang ada di Aplikasi :

1. Input Field " Enter a new task", Ini adalah elemen input teks di mana pengguna dapat menambahkan nama atau deskripsi tugas baru ke daftar tugas.
2. Button "Tambah Tugas", Tugas yang telah dimasukkan ke dalam kotak input dapat ditambahkan ke dalam daftar tugas dengan menggunakan tombol ini.
3. Filter Tugas (Semua, Aktiv, Selesai), Dengan menggunakan elemen ini, pengguna dapat memfilter tugas berdasarkan statusnya:  
   all: Menampilkan semua tugas, baik yang aktif maupun yang telah selesai.

Active: Hanya menampilkan tugas yang belum diselesaikan.

Completed: Hanya menampilkan tugas yang sudah selesai.

1. Dekorasi Warna, untuk memberikan kesan lembut dan sederhana, antarmuka pengguna dirancang dengan warna dasar merah muda.
   1. **Kesimpulan**

Aplikasi To-Do List ini membuat mengelola daftar tugas menjadi mudah dan interaktif. Tugas ini memiliki banyak kemampuan, seperti menambah tugas baru, menandai tugas yang sudah selesai, menghapus tugas, dan memfilter tugas berdasarkan statusnya berjalan atau sudah selesai. Dengan fitur dasar yang diperlukan untuk memantau dan mengorganisir aktivitas harian, aplikasi ini memiliki fitur seperti addTask(), toggleTask(), deleteTask(), dan filterTask. Aplikasi ini dapat berjalan secara dinamis di sisi klien berkat JavaScript, yang memungkinkan Anda melihat perubahan pada daftar tugas tanpa memuat halaman lagi.

**Link Github:** https://github.com/Kyaraazwa/Prak04.DW-A-/tree/main/Pertemuan%204\_4523210058\_Kyara%20Azwa