**LAPORAN PRAKTIKUM**

**JAVASCRIPT – APLIKASI TO DO LIST**

****

Disusun oleh:

Nama : Daffa Yassar Ahmad (4523210032)

Mata Kuliah : Praktikum Desain Web – A

Tanggal : 1 Oktober 2024

Dosen Pengampu:

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

**PROGRAM SARJANA TEKNIK INFORMATIKA**

**FAKULTAS TEKNIK**

**UNIVERSITAS PANCASILA**

**JAKARTA**

**2024**

**PENDAHULUAN**

**Deskripsi Praktikum**

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

**Tujuan Praktikum**

Setelah menyelesaikan praktikum ini, mahasiswa diharapkan mampu:

1. Menggunakan JavaScript DOM untuk mengambil, memanipulasi, dan memperbarui elemen HTML.
2. Mengimplementasikan JavaScript Class dan Object dalam aplikasi yang dinamis.
3. Memanfaatkan JavaScript Function untuk mengatur logika aplikasi.
4. Menerapkan event handling dalam interaksi pengguna dengan aplikasi.

**Penjelasan Teori Dasar**

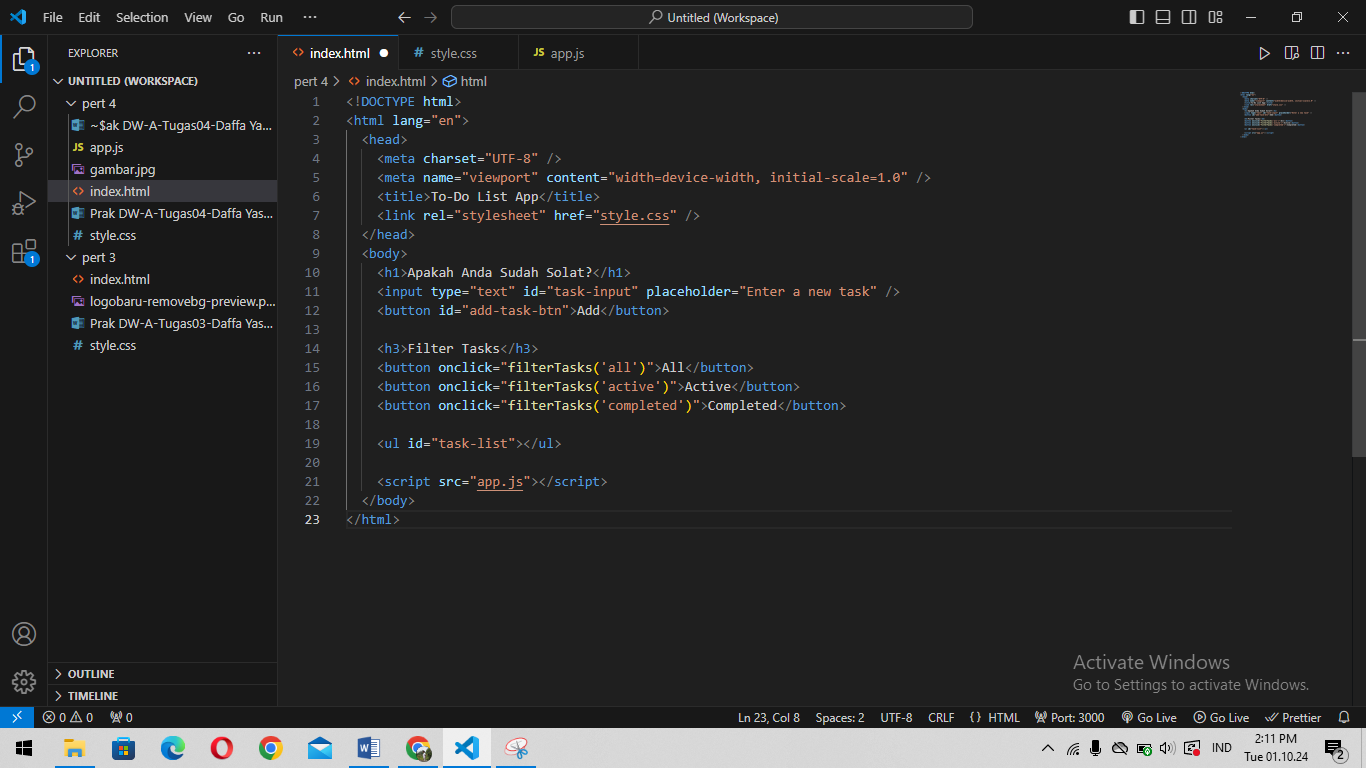
Buat aplikasi To-Do List dengan fitur sebagai berikut:

1. **Menambahkan Tugas**: Pengguna bisa menambahkan tugas baru ke dalam daftar dengan memasukkan teks tugas dan menekan tombol “Add Task”.
2. **Menandai Tugas sebagai Selesai**: Pengguna bisa menandai tugas sebagai selesai dengan mengklik checkbox di sebelah tugas.
3. **Menghapus Tugas**: Pengguna bisa menghapus tugas dari daftar.
4. **Validasi Input**: Jangan biarkan pengguna menambahkan tugas yang kosong.

**Filter Tugas**: Buat tombol untuk menampilkan semua tugas, tugas yang belum selesai, atau tugas yang sudah selesai.

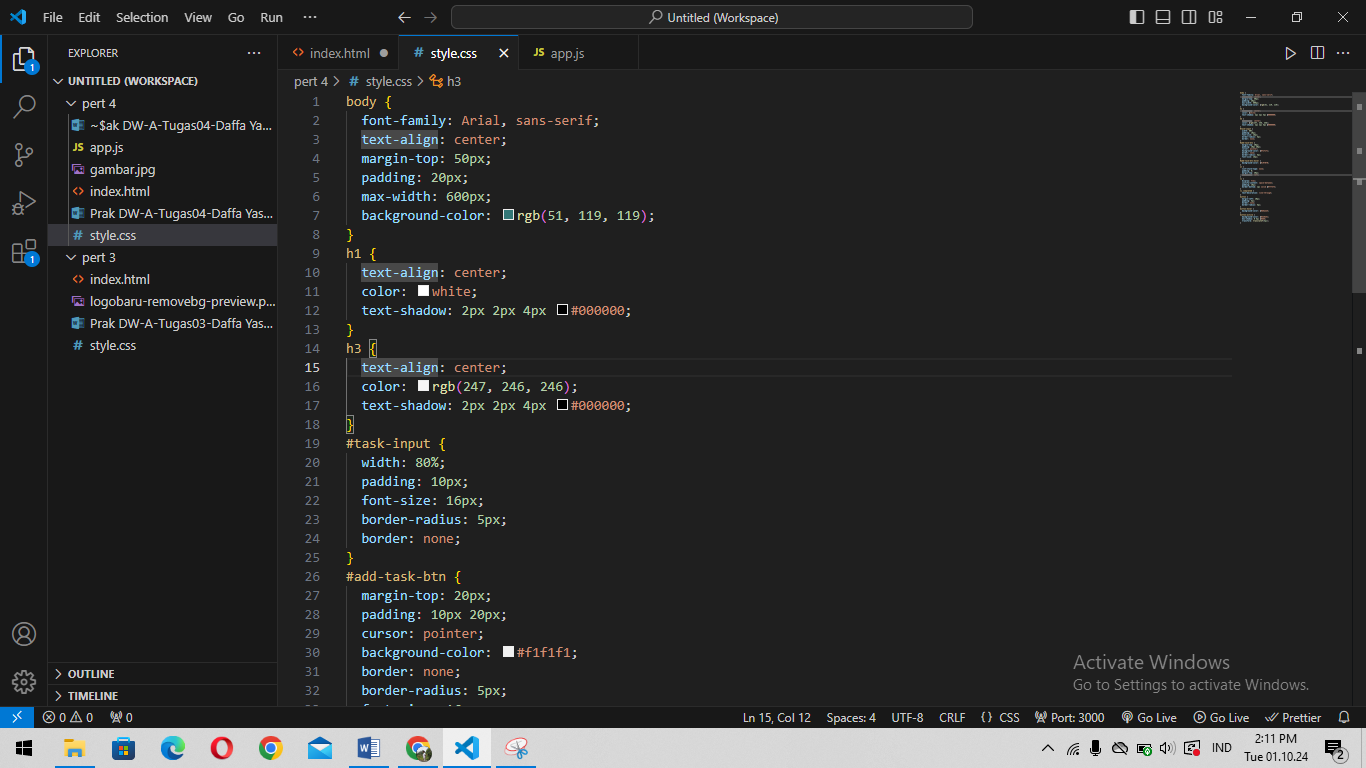
**LANGKAH PENGERJAAN**

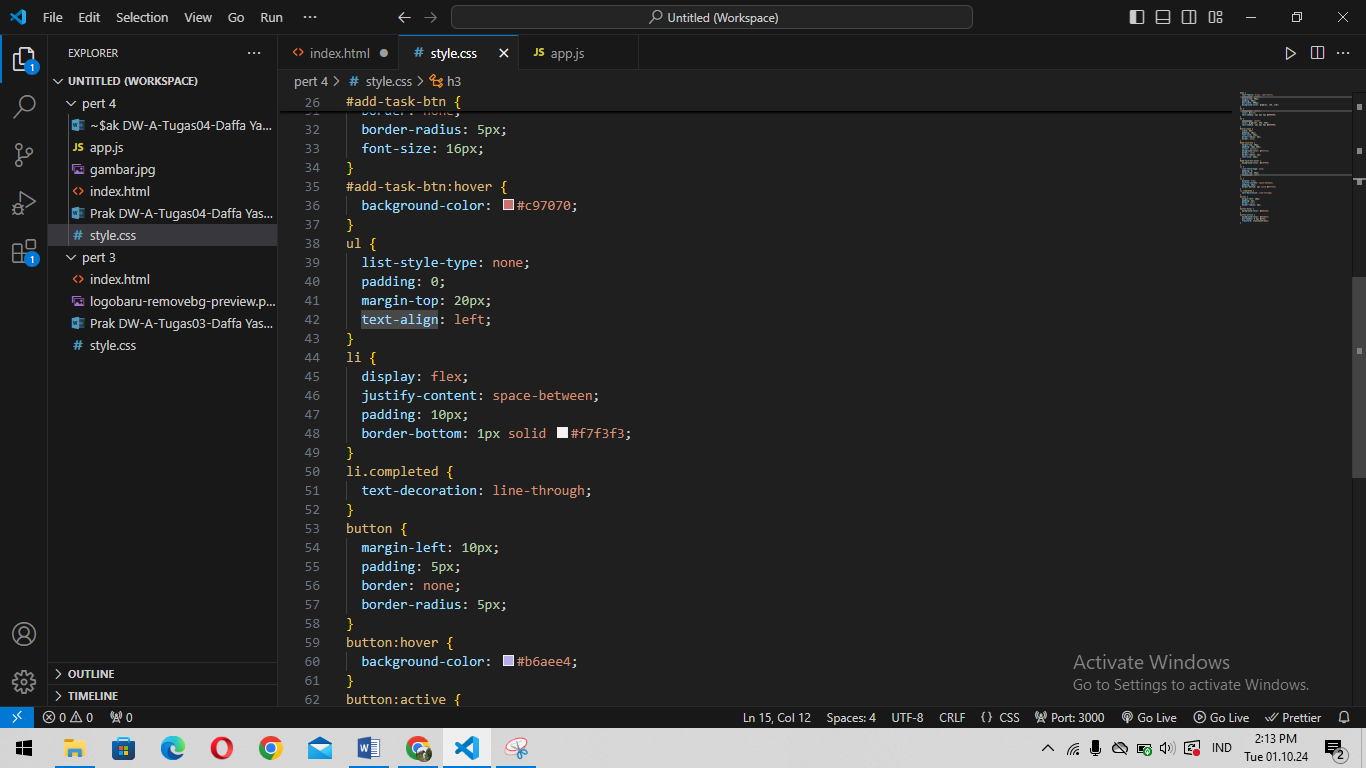
Index.html



Struktur dasar HTML dibuat dengan elemen input dan tombol untuk menambah tugas, serta tombol filter tugas.

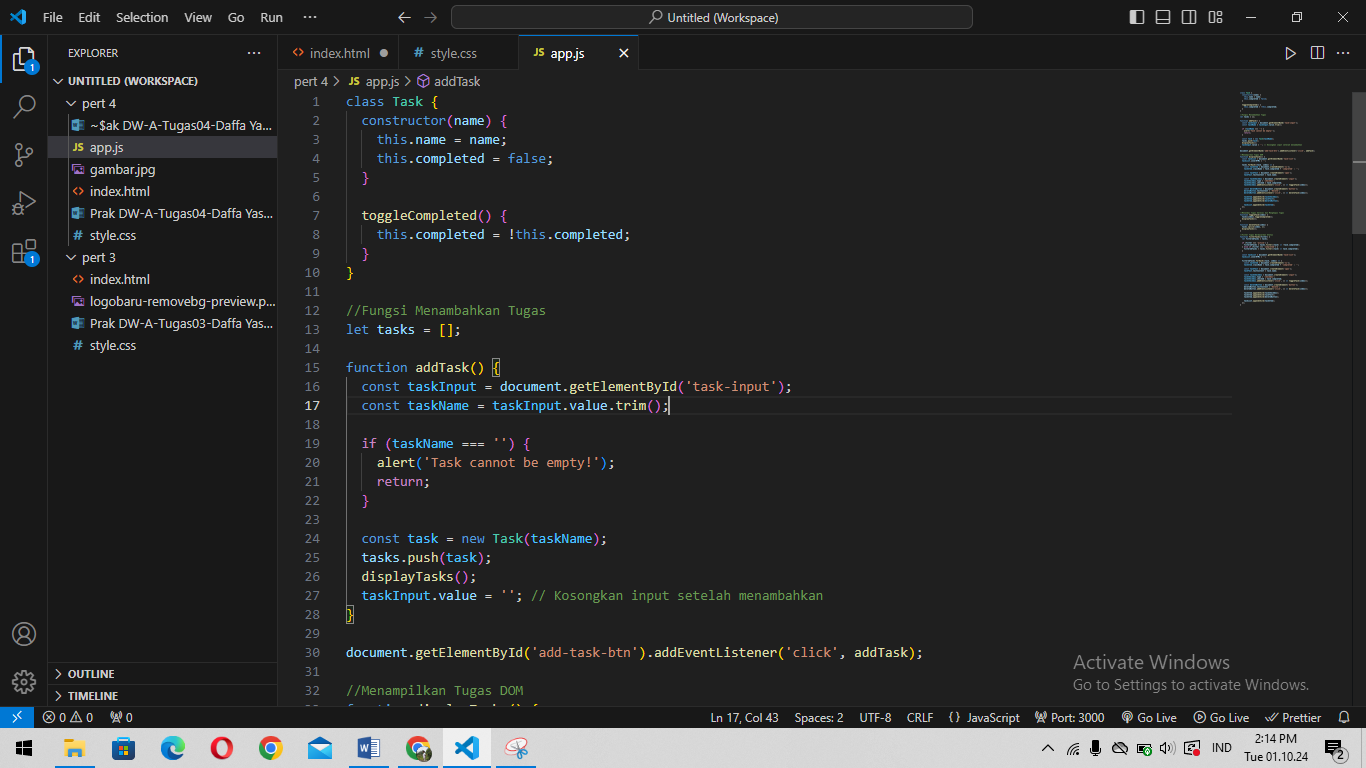
Style.css

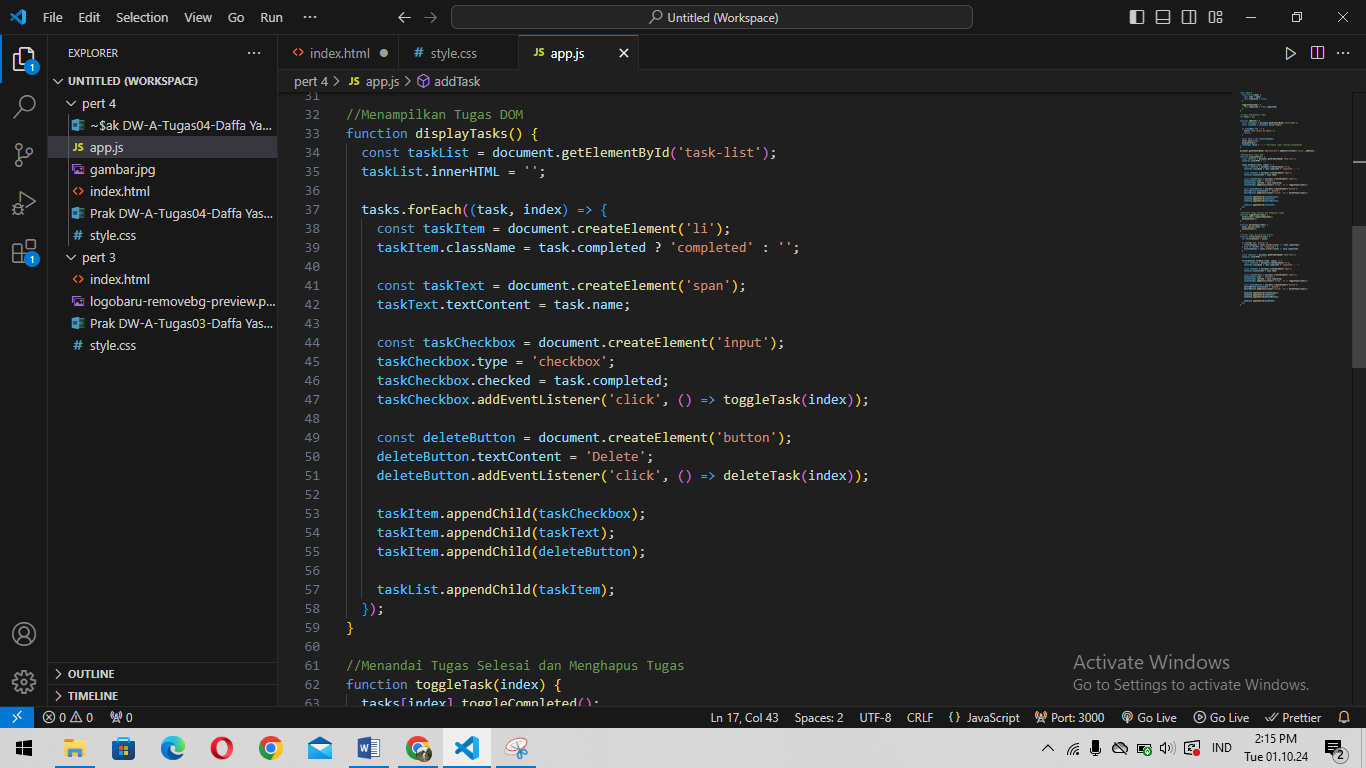


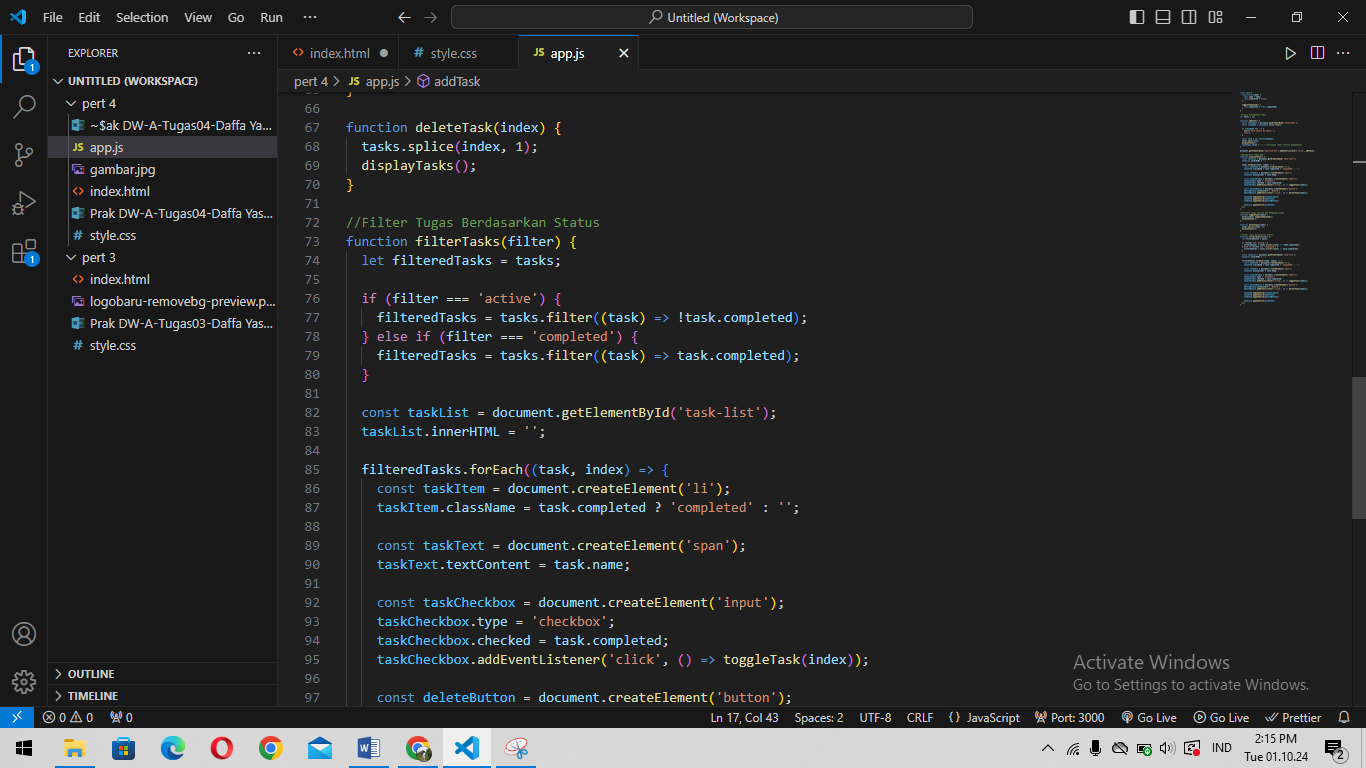


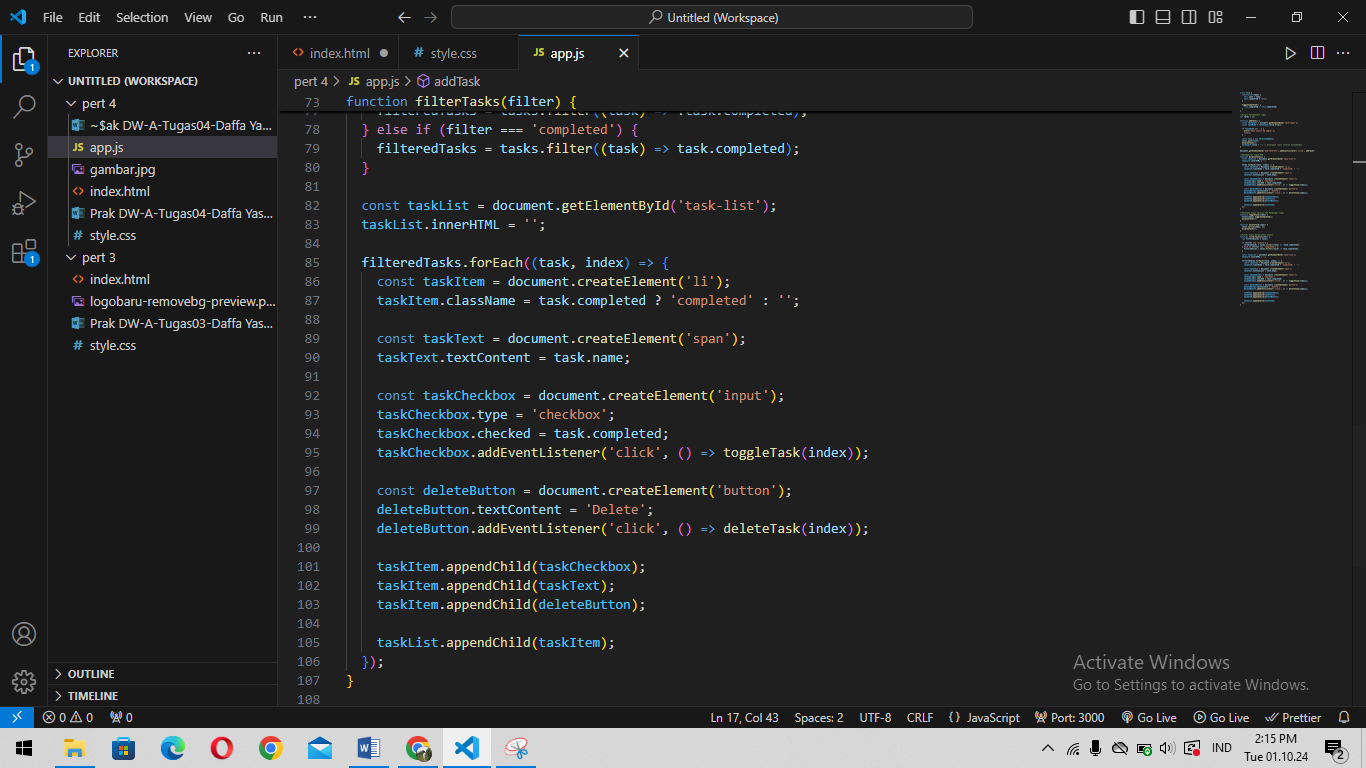
CSS digunakan untuk memberikan tampilan visual yang menarik dan responsif.

App.js









**Class Task**: Digunakan untuk merepresentasikan tugas dengan properti name dan completed.

**Fungsi addTask()**: Menambahkan tugas baru yang dimasukkan pengguna ke dalam array tasks dan menampilkannya di DOM.

**Fungsi toggleTask()**: Menandai tugas sebagai selesai/tidak selesai.

**Fungsi deleteTask()**: Menghapus tugas dari daftar.

**Fungsi filterTasks()**: Memfilter tugas berdasarkan status (semua, aktif, atau selesai).

**STRUKTUR DAN PENJELASAN KODE**

class Task {

  constructor(name) {

    this.name = name;

    this.completed = false;

  }

  toggleCompleted() {

    this.completed = !this.completed;

  }

}

**Fungsi toggleCompleted()**: Digunakan untuk mengubah status tugas dari belum selesai menjadi selesai atau sebaliknya. Ketika fungsi ini dipanggil, status completed akan dibalik dari false ke true, atau sebaliknya.

//Fungsi Menambahkan Tugas

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

}

**Tujuan**: Menambahkan tugas baru ke dalam daftar tugas.

//Menandai Tugas Selesai dan Menghapus Tugas

function toggleTask(index) {

  tasks[index].toggleCompleted();

  displayTasks();

}

**Tujuan**: Mengubah status tugas menjadi selesai atau belum selesai.

function deleteTask(index) {

  tasks.splice(index, 1);

  displayTasks();

}

**Tujuan**: Menghapus tugas dari daftar.

//Filter Tugas Berdasarkan Status

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);

  });

}

**Tujuan**: Menampilkan daftar tugas di elemen DOM.

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

Ketika tombol "Add Task" diklik, fungsi addTask() dipanggil untuk menambahkan tugas baru ke daftar.

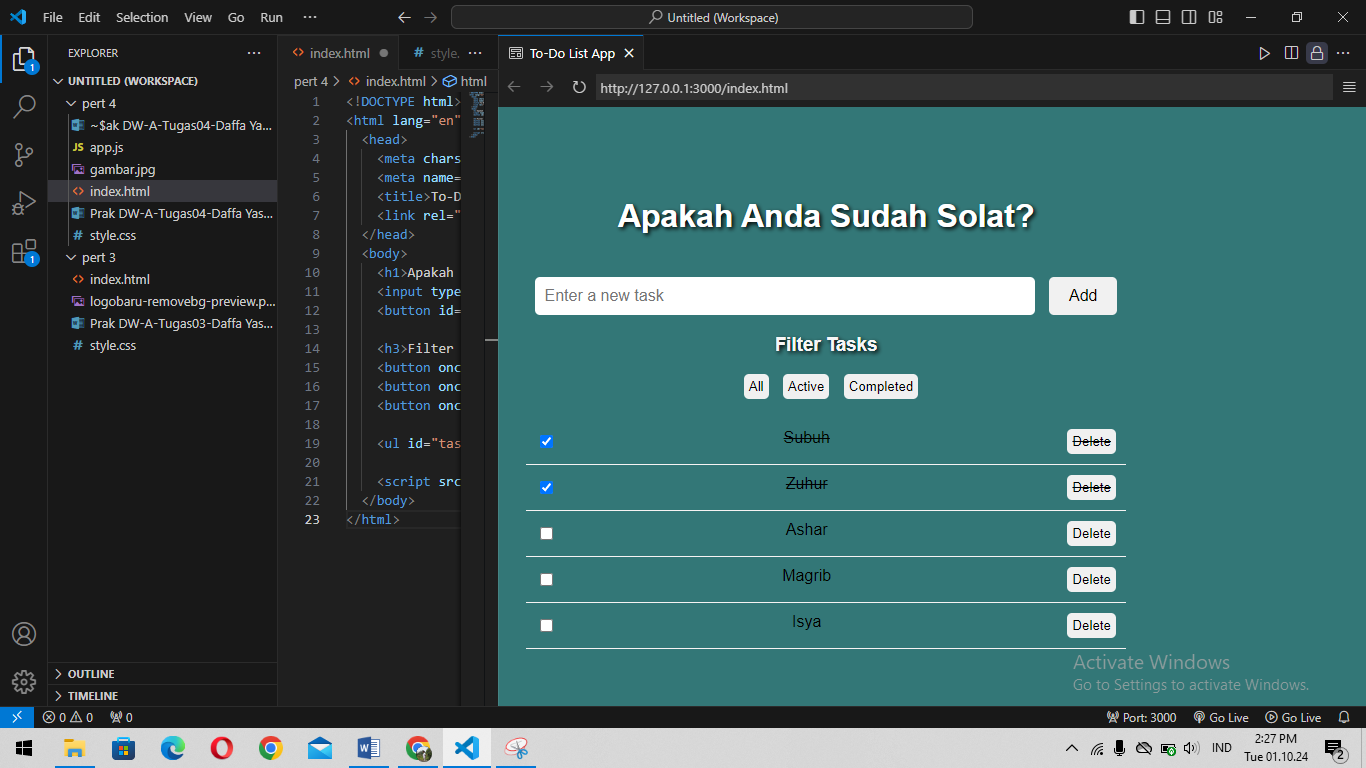
taskCheckbox.addEventListener('click', () => toggleTask(index));

Ketika checkbox diklik, fungsi toggleTask() dipanggil untuk mengubah status tugas menjadi selesai atau belum selesai.

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

Ketika tombol "Delete" diklik, fungsi deleteTask() dipanggil untuk menghapus tugas dari daftar.

**HASIL UJI COBA**



Berikut adalah tampilan dari codingan index.html, style.css dan app.js

**KESIMPULAN**

Aplikasi To-Do List ini mengimplementasikan dasar-dasar JavaScript untuk memanipulasi DOM, menangani event, dan mengatur logika aplikasi secara dinamis. Mahasiswa diharapkan dapat menguasai konsep-konsep penting seperti **Class**, **Object**, **function**, serta penggunaan **event handling**.