**PRAKTIKUM 04**

**Praktikum JavaScript – Aplikasi To-Do List**

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

****

**Karin Lina Safitri**

**4523210056**

Dosen Pembimbing :

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

**01 Oktober 2024**

**FAKULTAS TEKNIK**

**UNIVERSITAS PANCASILA**

**JAKARTA**

**2024**

**BAB I**

**PENDAHULUAN**

* 1. **Tujuan**

Mahasiswa dapat memahami cara menggunakan DOM (Document Object Model) di JavaScript untuk mengakses dan mengubah konten dan struktur elemen HTML secara dinamis. Mahasiswa dapat menerapkan konsep Class dan Object dalam JavaScript untuk membuat kode lebih terstruktur dan modular saat mengembangkan aplikasi yang dapat disesuaikan dengan interaksi pengguna atau data.

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

Alat digital yang disebut aplikasi to-do list membantu pengguna mencatat, mengelola, dan melacak tugas yang harus dilakukan. Aplikasi ini memiliki fitur yang memungkinkan pengaturan prioritas, pengingat, dan pencapaian target harian atau jangka panjang untuk meningkatkan produktivitas.

**1.3 Penjelasan teori dasar praktikum**

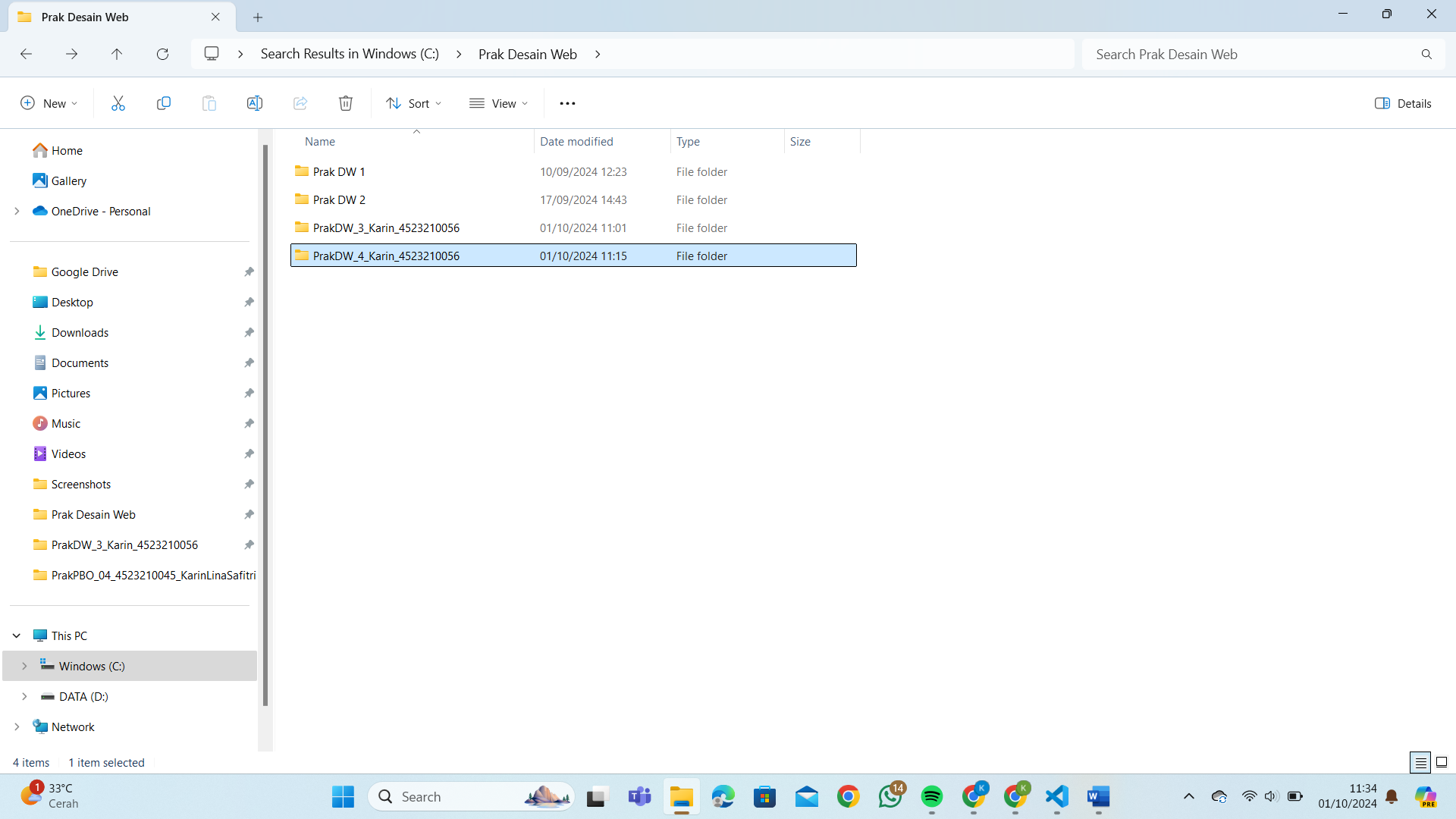
JavaScript adalah bahasa pemrograman yang digunakan untuk membuat halaman web lebih dinamis, interaktif, dan responsif. Itu bekerja di sisi browser, memungkinkan pengembang mengubah konten HTML/CSS, menangani peristiwa pengguna, membuat animasi, dan berinteraksi dengan server melalui AJAX atau Fetch API.

**BAB II**

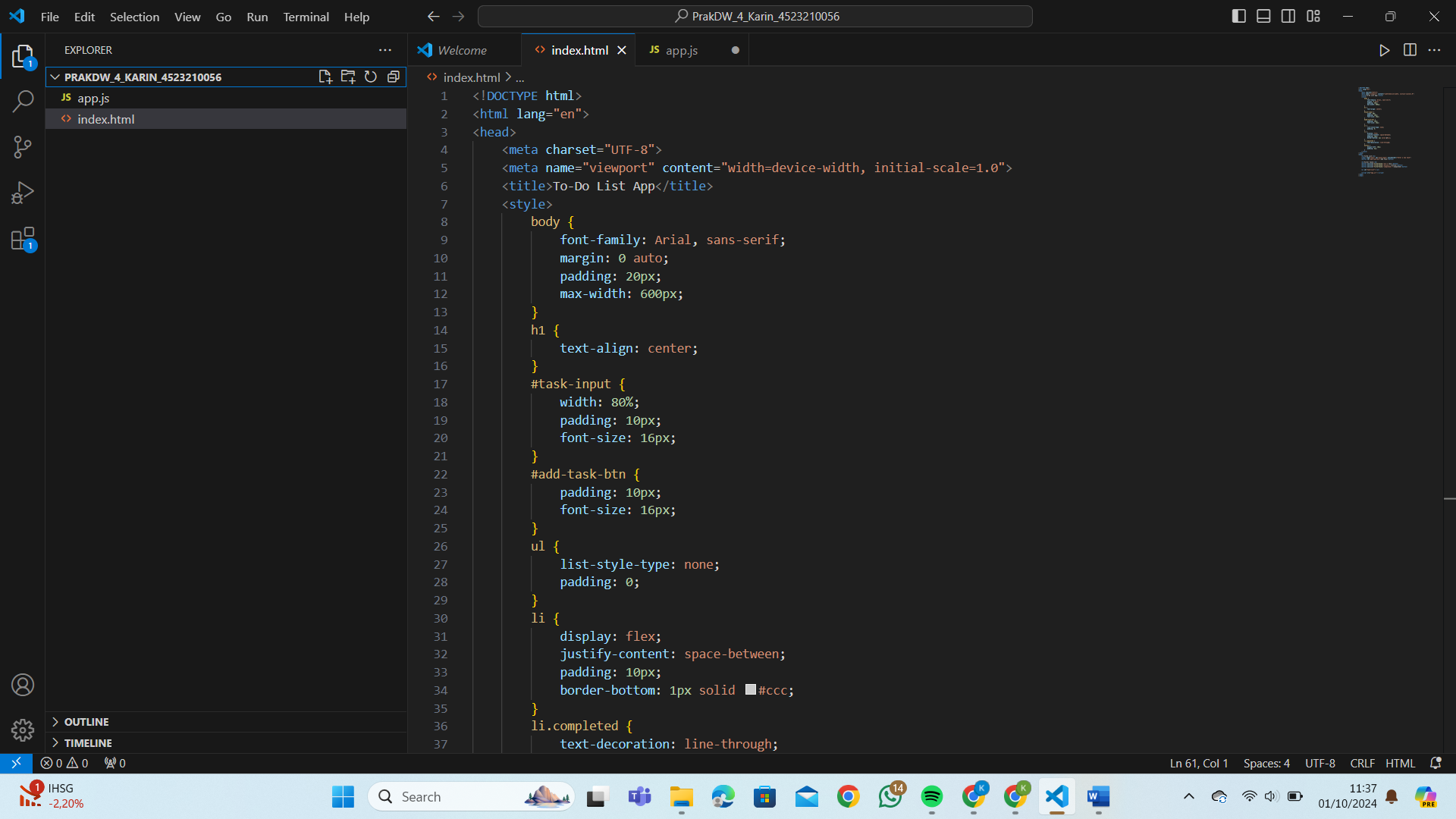
**LANGKAH PENGERJAAN**

* 1. **Tahapan Praktikum**

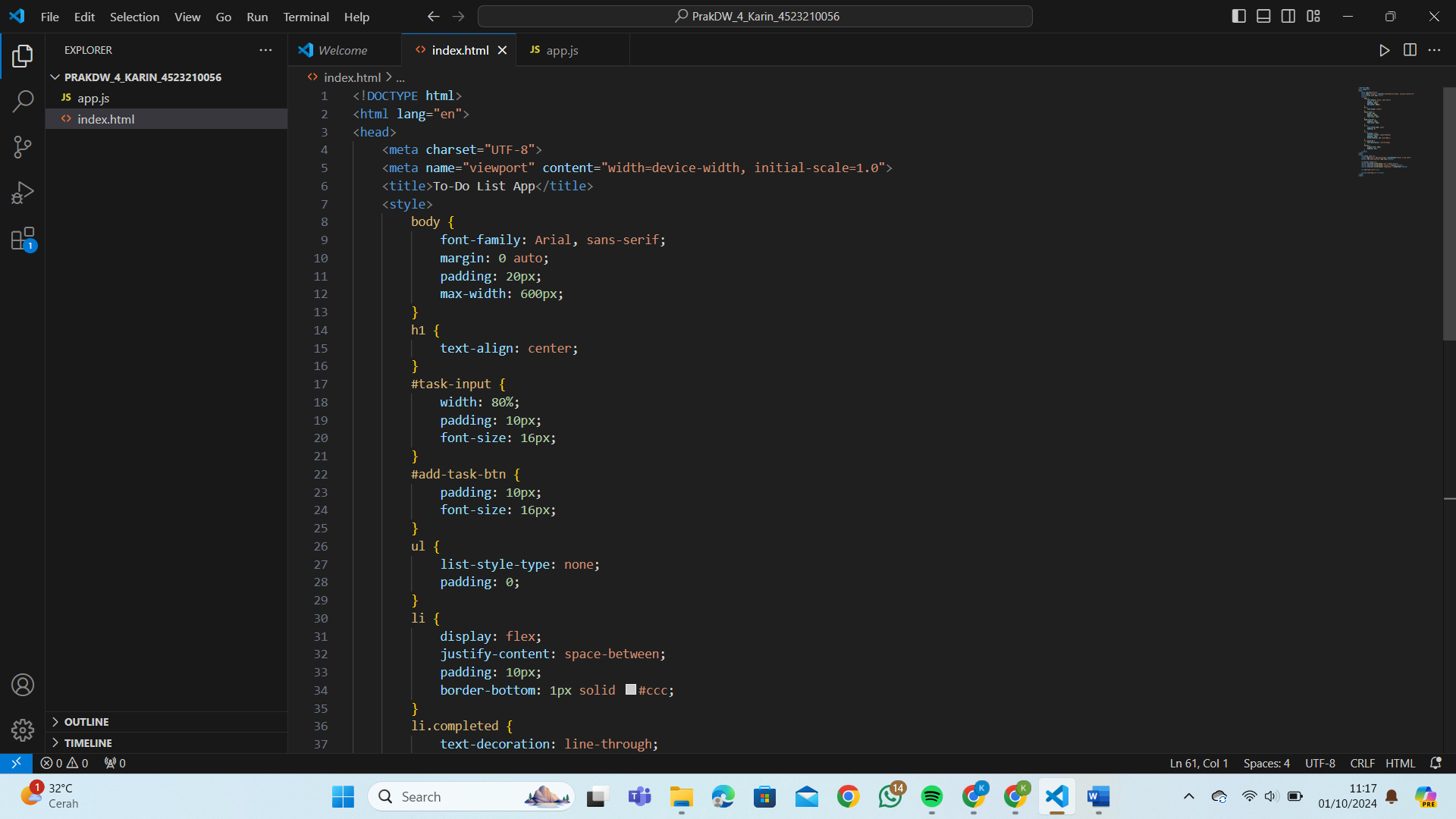
1. Membuat folder untuk menyimpan data

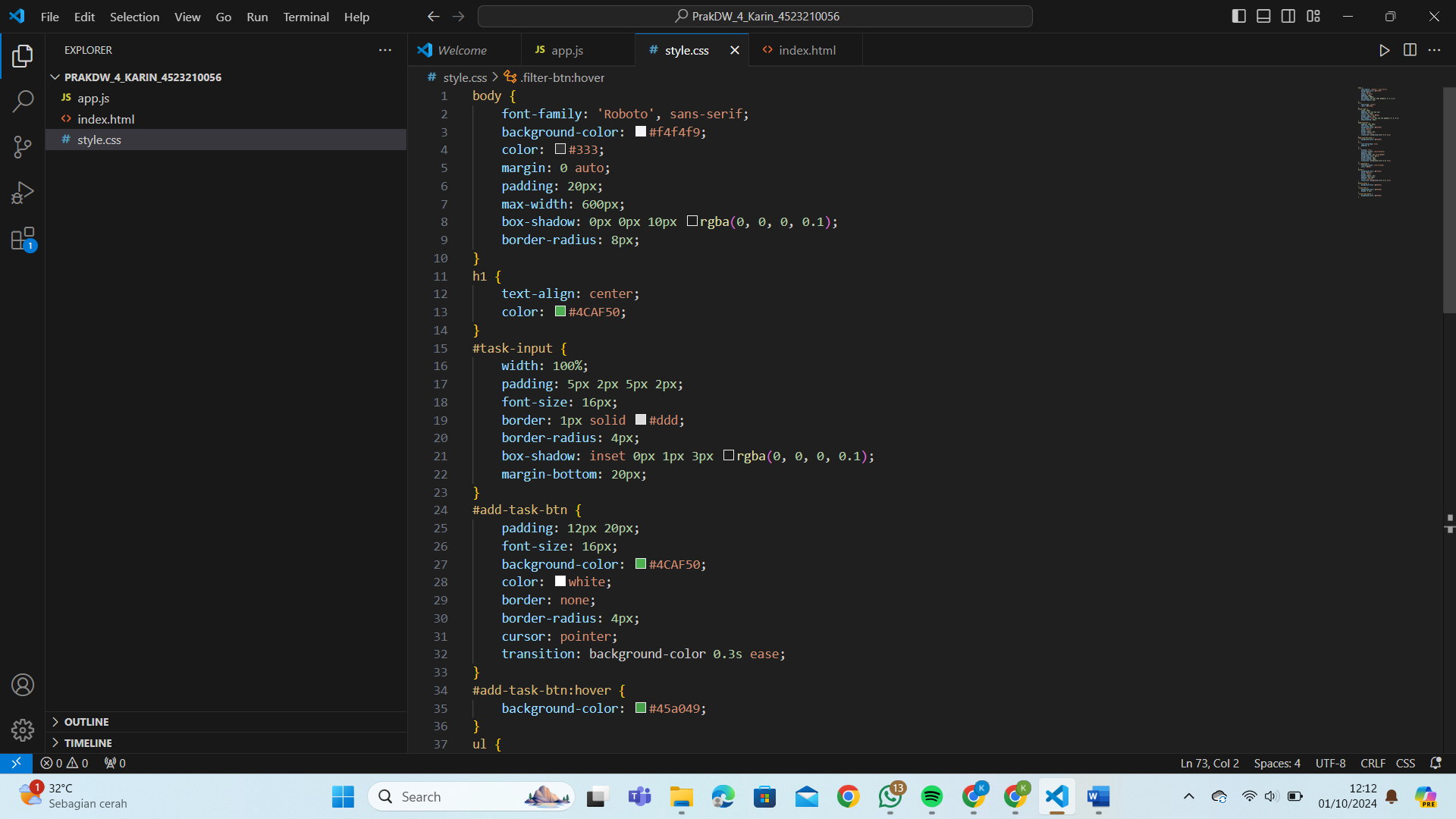


1. Membuat file tiap data

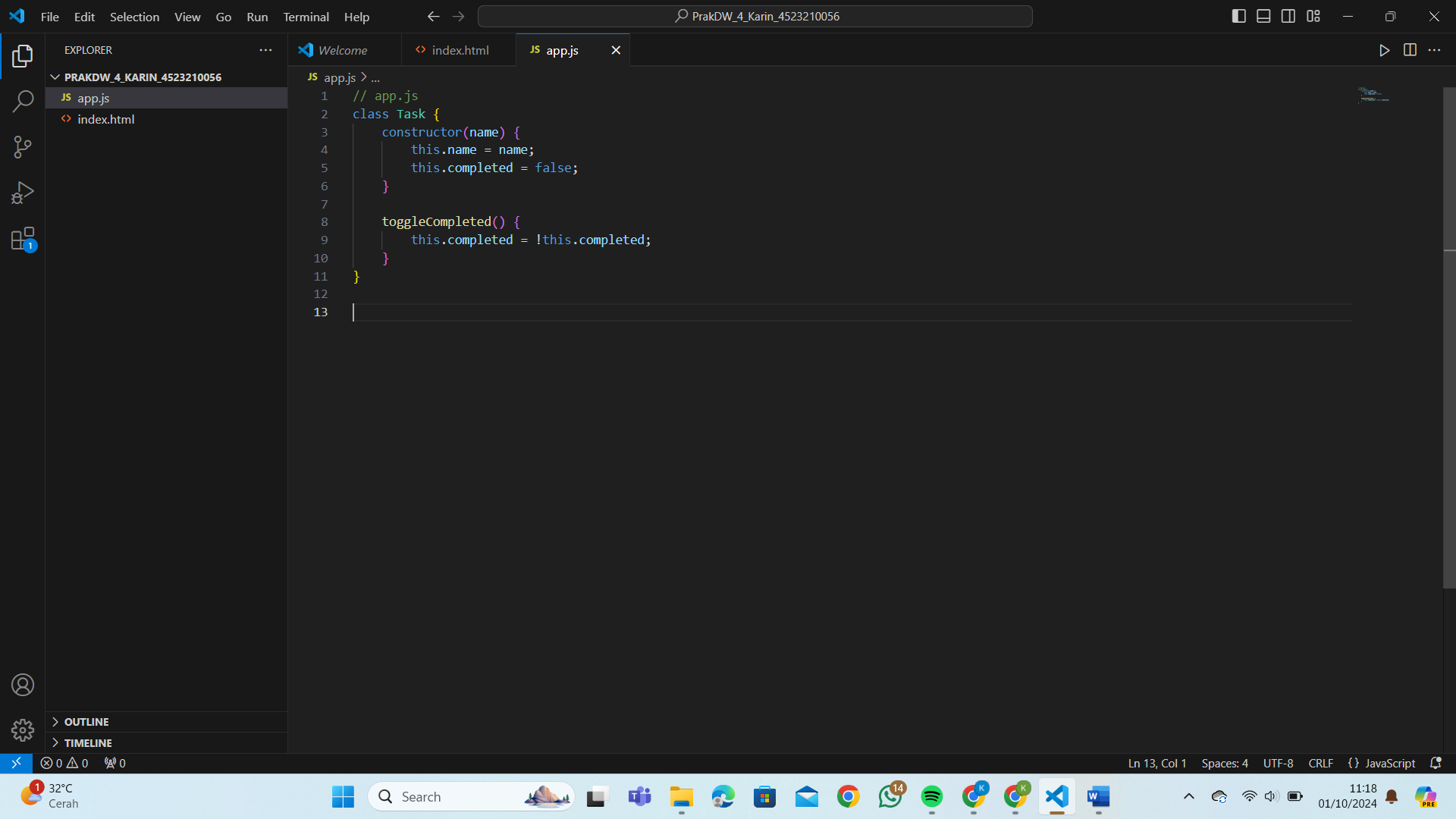


1. Memasukkan Index.html dan style.css

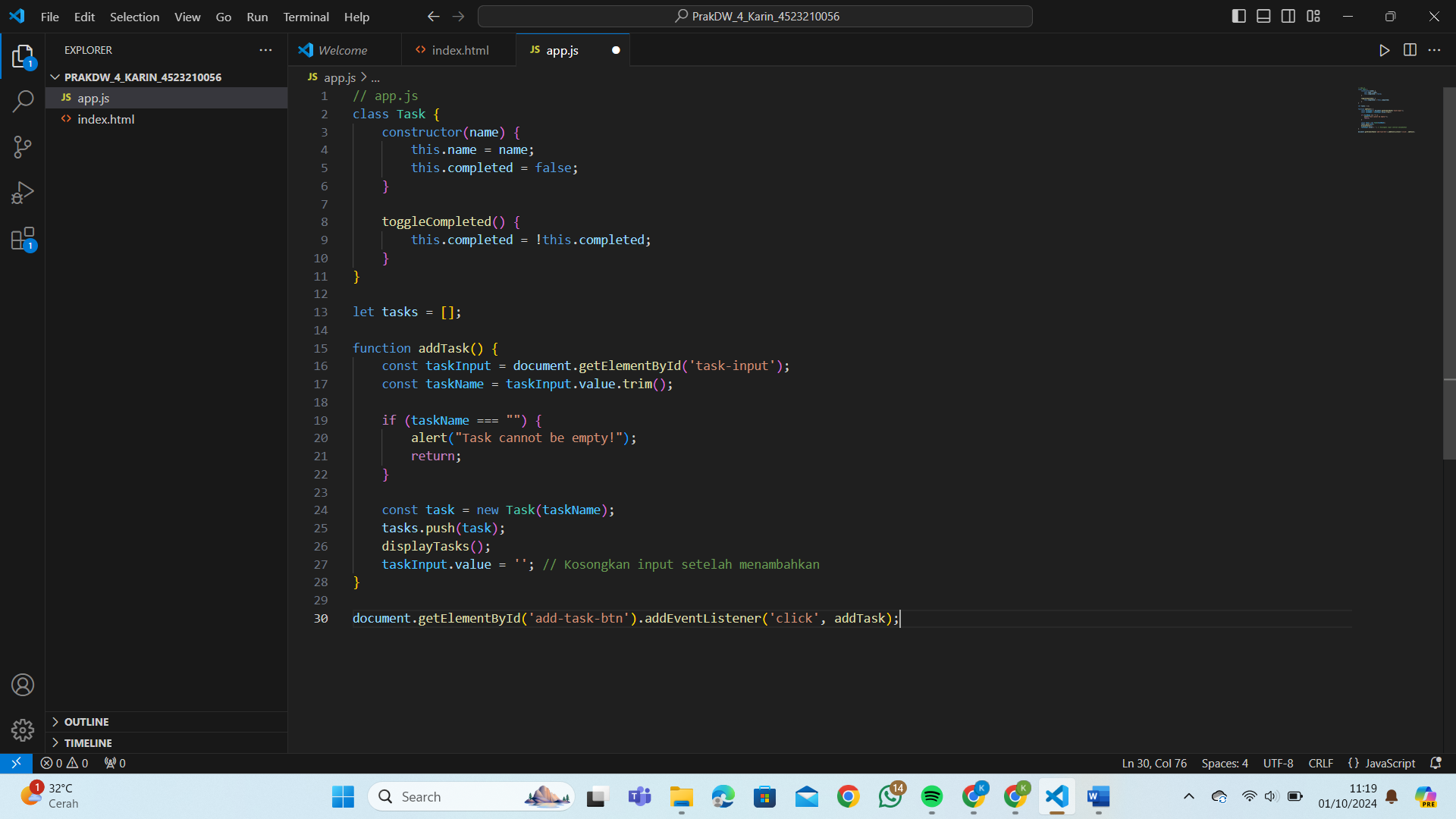




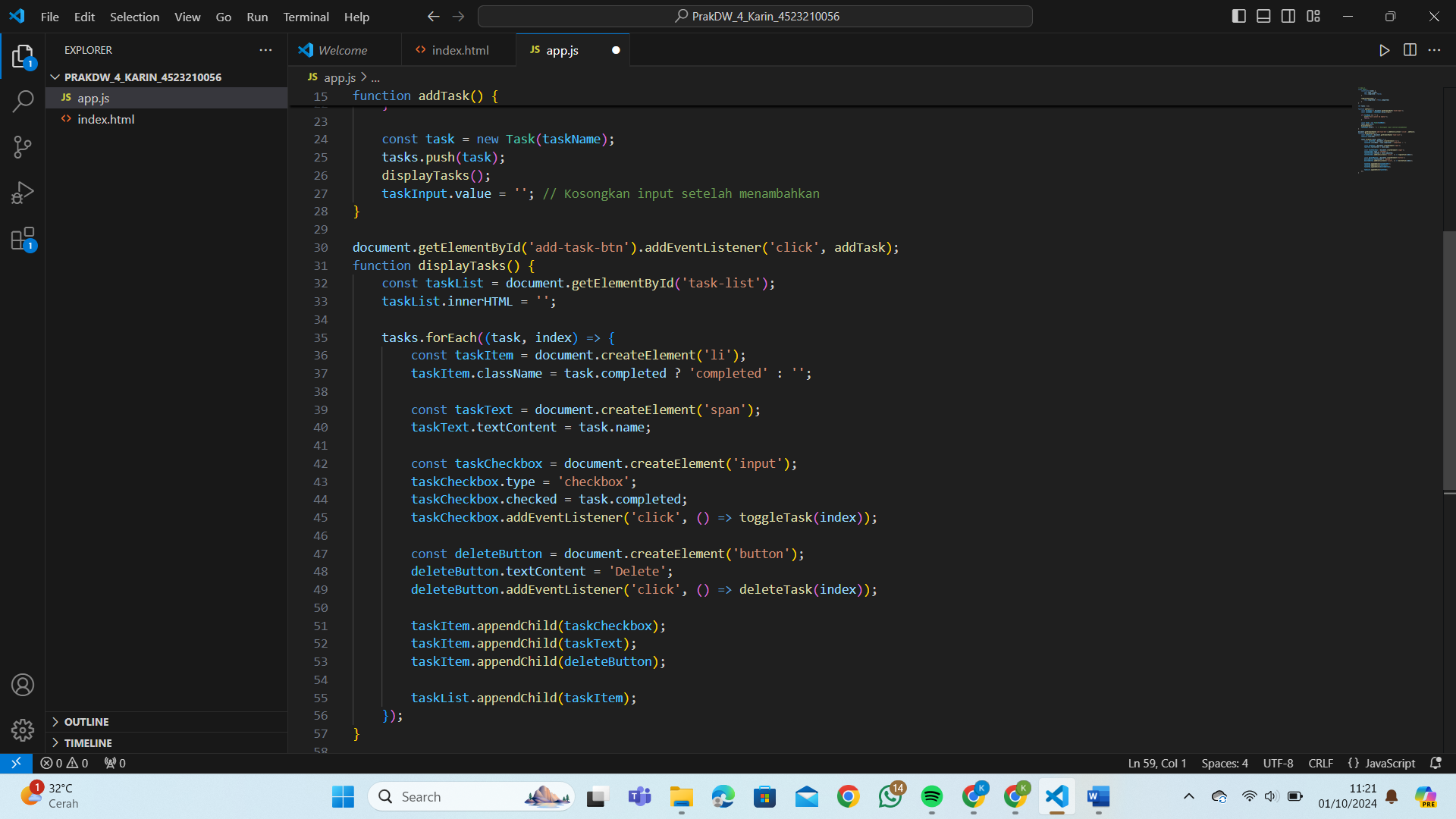
1. Memasukkan App.js



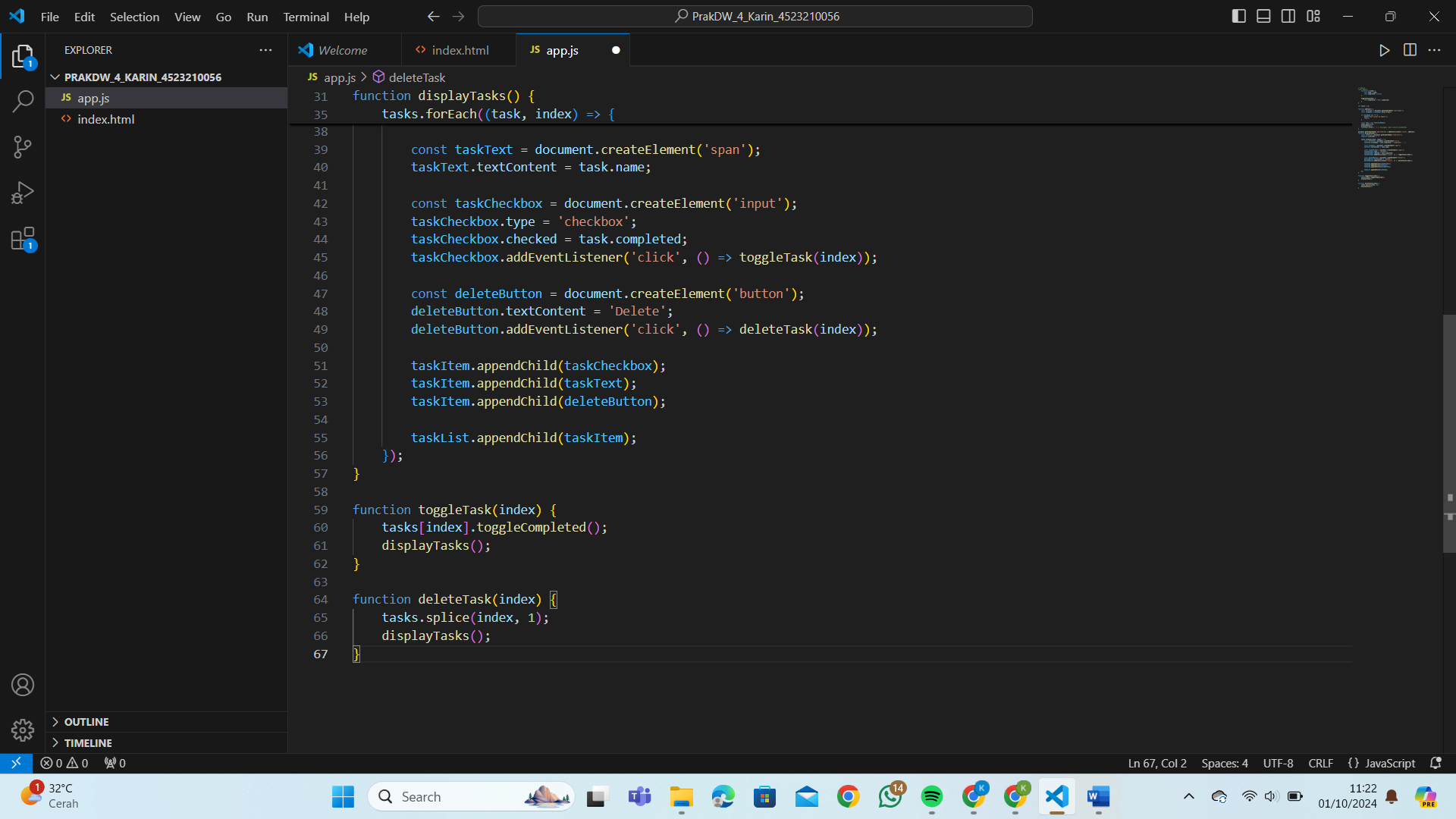
1. Membuat Fungsi untuk Menambah Tugas



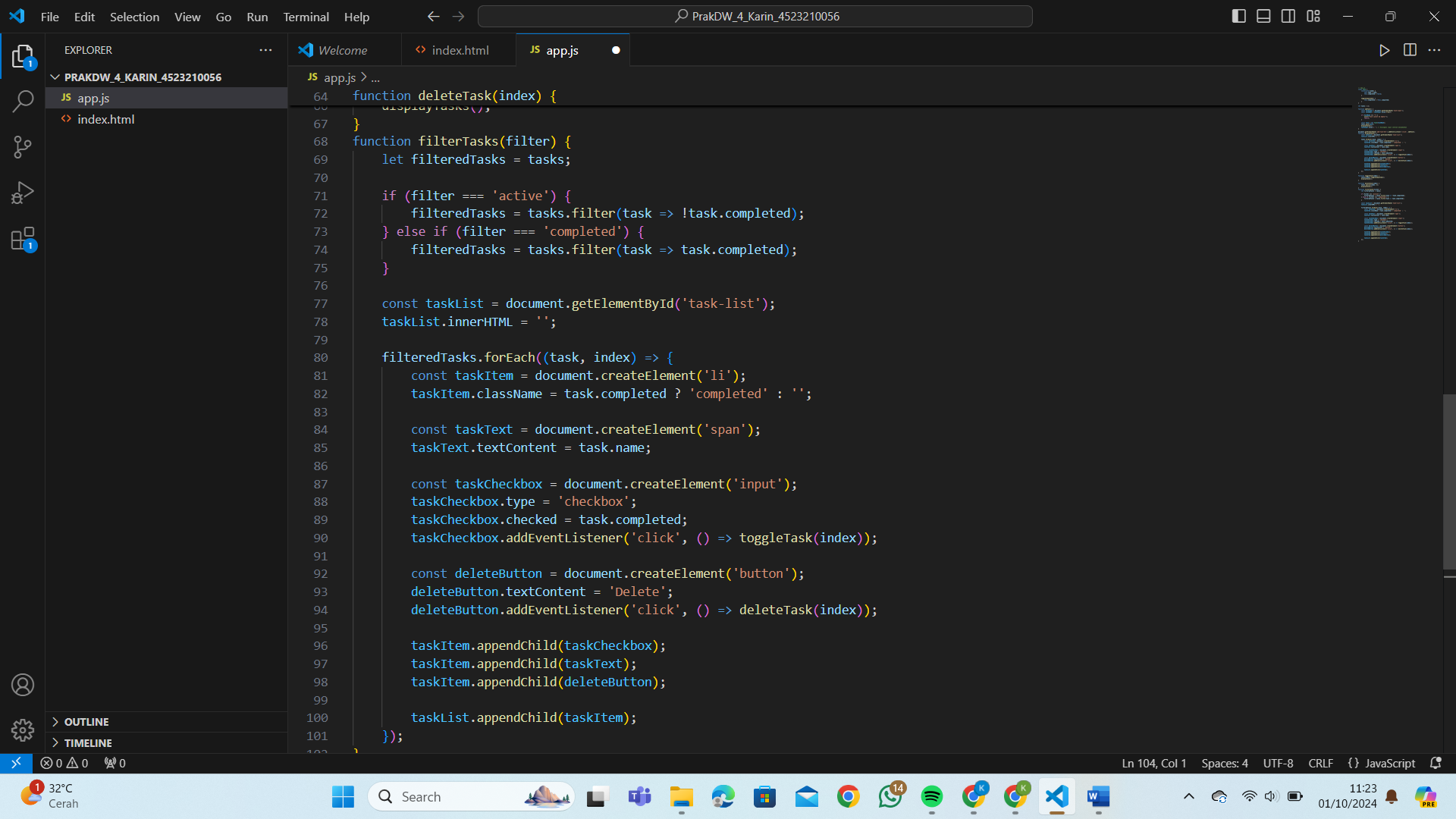
1. Menampilkan Tugas di DOM



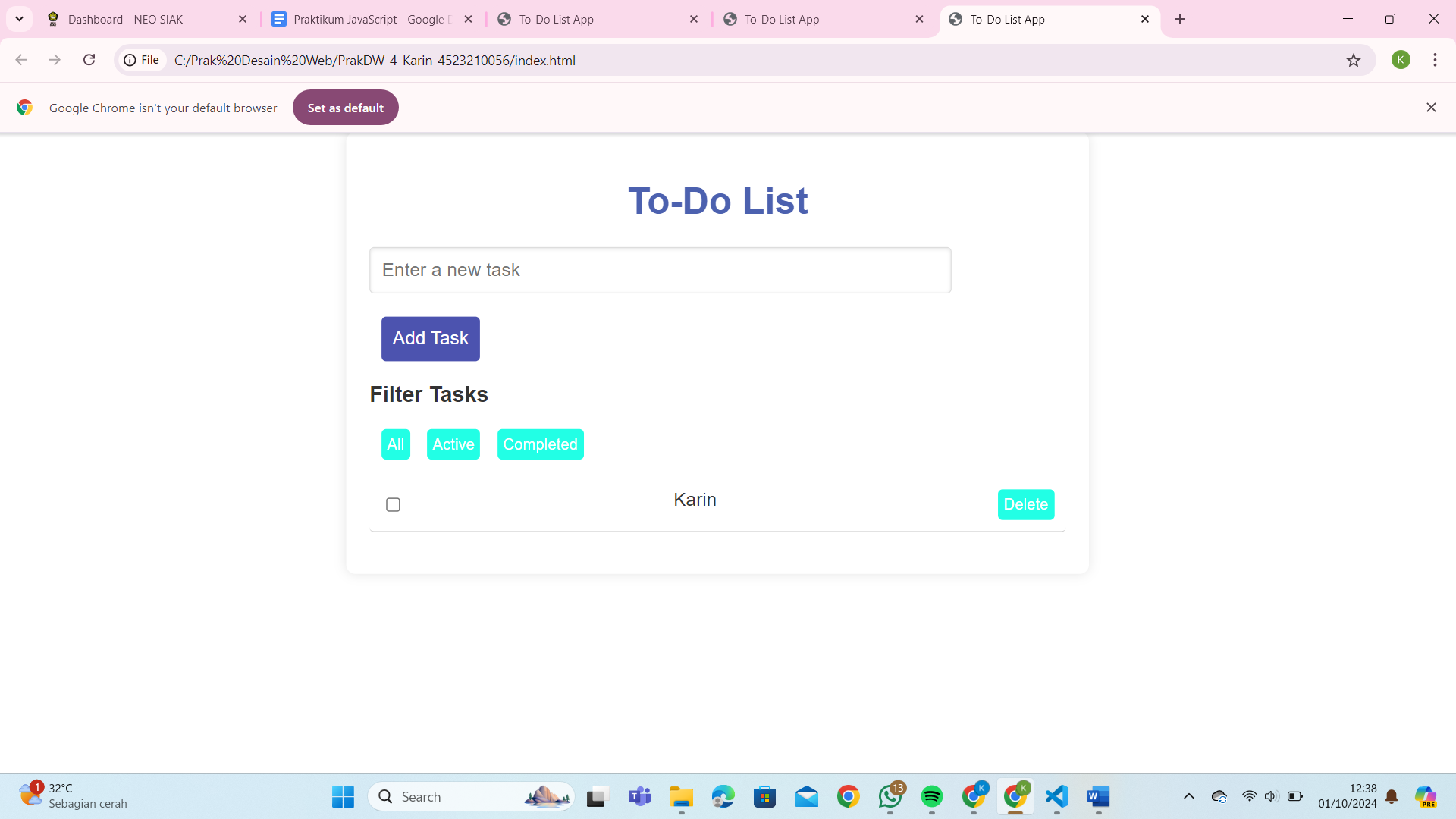
1. Menandai Tugas Selesai dan Menghapus Tugas



1. Filter Tugas Berdasarkan Status



1. Hasil Running



**BAB III**

**STRUKTUR DAN PENJELASAN KODE**

* 1. **Sertakan kode JavaScript lengkap yang digunakan dalam aplikasi To-Do List.**

// 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);

    });

}

* 1. **Jelaskan secara singkat setiap bagian dari kode:**
     1. **Class Task dan fungsinya.**

Class Task berfungsi untuk merepresentasikan sebuah tugas yang memiliki nama (name) dan status penyelesaian (completed). Secara default, tugas belum selesai (completed = false), dan method toggleCompleted() digunakan untuk mengubah status penyelesaian tugas antara selesai dan belum selesai.

* + 1. **Fungsi addTask**

Fungsi addTask() digunakan untuk menambahkan tugas baru ke dalam daftar tugas (tasks). Fungsi ini mengambil nilai input dari elemen dengan ID task-input, memvalidasi apakah input tidak kosong, lalu membuat objek baru dari kelas Task. Objek tugas tersebut ditambahkan ke array tasks, ditampilkan menggunakan fungsi displayTasks(), dan input teks dikosongkan setelah tugas ditambahkan.

**3.2.3 Fungsi toggleTask**

Fungsi toggleTask(index) digunakan untuk mengubah status penyelesaian (completed) dari tugas pada indeks tertentu dalam array tasks. Fungsi ini memanggil method toggleCompleted() pada tugas yang dipilih, lalu memperbarui tampilan tugas dengan memanggil displayTasks().

**3.2.4 Fungsi deleteTask**

Fungsi deleteTask(index) digunakan untuk menghapus tugas pada indeks tertentu dalam array tasks. Fungsi ini menggunakan metode splice() untuk menghapus tugas dari array, lalu memperbarui tampilan daftar tugas dengan memanggil displayTasks().

**3.2.5 Fungsi filterTask**

Fungsi filterTasks(filter) digunakan untuk memfilter dan menampilkan tugas berdasarkan status penyelesaiannya. Jika parameter filter adalah 'active', fungsi ini hanya menampilkan tugas yang belum selesai. Jika filter adalah 'completed', hanya tugas yang selesai yang ditampilkan. Jika tidak ada filter, semua tugas ditampilkan. Fungsi ini juga memperbarui tampilan daftar tugas (task-list) sesuai dengan hasil filter tersebut.

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

displayTasks():

Mengosongkan elemen daftar tugas (task-list), kemudian membuat elemen <li> untuk setiap tugas.

Menambahkan elemen checkbox, teks tugas, dan tombol hapus ke dalam <li>, dan menambahkannya ke dalam daftar.

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

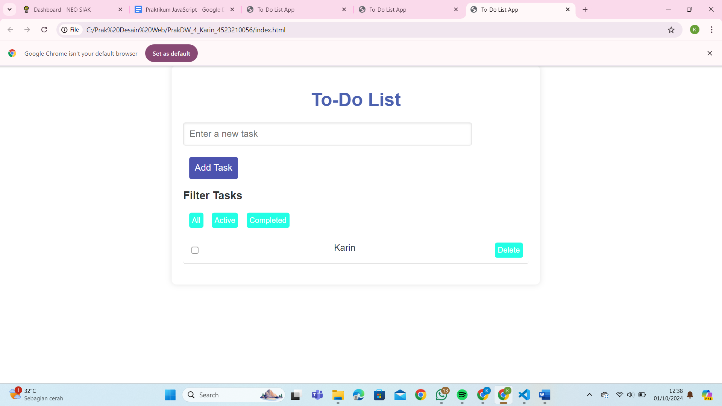
Tombol “Add Task”:Menggunakan addEventListener untuk menambahkan event click pada tombol. Ketika tombol diklik, fungsi addTask() akan dipanggil.

Checkbox:Setiap checkbox memiliki event listener yang memanggil toggleTask(index) ketika dicentang atau dicopot. Ini memungkinkan pengguna untuk menandai tugas sebagai selesai atau tidak.

**BAB IV**

**HASIL DAN KESIMPULAN**

* **HASIL**

****

Elemen-elemen yag terdapat pada gambar diatas :

* + 1. Input field (“ Enter a new task”) : Pada bagian kontak input ini pengguna bisa memasukkan atau mengetik teks ,nama,atau deskripsi tugas yang ingin ditambahkan
    2. “Add Task” : Setelah mengetik pada bagian input , pengguna dapat menekan tombol “Add Task” untuk menambahkan tugas tersebut ke dalam daftar.
    3. Filter Tasks:
* All: Filter ini menampilkan semua tugas yang telah ditambahkan, baik yang masih aktif maupun yang sudah diselesaikan.
* Active: Filter ini menampilkan hanya tugas-tugas yang belum diselesaikan.
* Completed: Filter ini menampilkan hanya tugas-tugas yang sudah diselesaikan.
  + 1. Checkbox: Di samping setiap tugas yang sudah atau akan ditambahkan terdapat kotak centang (checkbox) yang memungkinkan pengguna untuk menandai tugas sebagai sudah selesai atau aktif.
    2. Tombol "Delete": Tombol ini berwarna hijau muda dan digunakan untuk menghapus tugas-tugas yang dipilih oleh pengguna.
* **KESIMPULAN**

Aplikasi ini memungkinkan pengguna mengelola daftar tugas secara mudah dan interaktif. Tugas ini memiliki banyak kemampuan, termasuk menambah tugas baru, menandai tugas yang sudah selesai, menghapus tugas, dan memfilter tugas berdasarkan statusnya (aktif atau selesai). Aplikasi ini memiliki fitur dasar yang diperlukan untuk memantau dan mengorganisir aktivitas harian secara efektif, seperti addTask(), toggleTask(), deleteTask(), dan filterTasks. Aplikasi ini dapat berjalan secara dinamis di sisi klien berkat JavaScript, yang memungkinkan Anda melihat perubahan pada daftar tugas tanpa memuat ulang halaman.