**Laporan Praktikum**

**JavaScript – Aplikasi To-Do List**

**Desain Web - A**

****

Oleh :

Dendy Anugrahi Rabbi/4523210034

Dosen :

**Adi Wahyu Pribadi**

**S1 Teknik Informatika**

**Universitas Pancasila**

**01/10**

**2024/2025**

**Pendahuluan**

Tujuan dari mempelajari praktikum ini untuk memberikan pemahaman praktis dalam menggunakan JavaScript DOM untuk memperbaharui elemen HTML, mengimplementasikan JavaScript Class dan Object, dapat menggunakan JavaScript Function untuk mengatur logika aplikasi dan memproses interaksi, dan dapat menerapkan event handling untuk menangani interaksi pengguna dengan aplikasi, seperti klik, input data, atau perubahan lainnya.

Aplikasi To-Do List adalah sebuah alat digital untuk membantu pengguna dalam mengelola tugas harian mereka secara lebih efektif. Dengan aplikasi ini, pengguna dapat membuat daftar tugas, mengatur tenggat waktu, sekaligus untuk pengingat tugas tugas.

# Langkah Pengerjaan

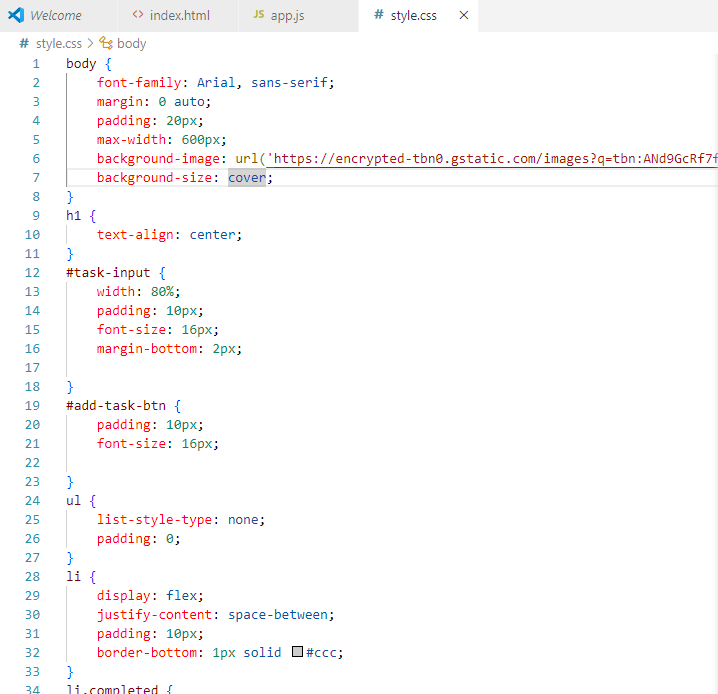
**Menyiapkan File HTML**

HTML adalah file yang menjadi kerangka utama dari halaman web yang berisikan elemen-elemen penting seperti judul, form, tombol, dll.



**Menyiapkan CSS**

File CSS untuk membuat desain tampilan halaman yang meliputi warna, tipografi, tata letak, dana elemen visual lainnya

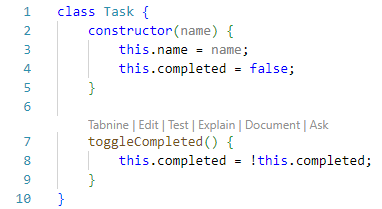
****

**Menyiapkan File JavaScript**

File JavaScript dihubungkan dengan HTML untuk memanipulasi DOM dan menangani interaksi pengguna yang akan mengandung semua logika fungsional aplikasi.



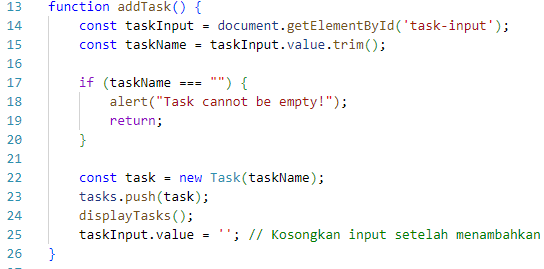
# Struktur dan Penjelasan Kode

****

**Class Task** digunakan untuk mempresentasikan setiap tugas dalam daftar. Setiap tugas memiliki 2 properti:

* name: Nama atau deskripsi tugas
* completed: Status tugas apakah sudah selesai atau belum (default: false)

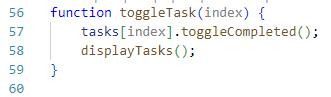
toggleCompleted() adalah metode dalam class yang digunakan untuk mengubah status tugas dari belum selesai menjadi selesai, atau sebaliknya.



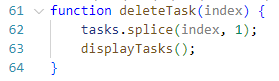
Fungsi **addTask()** untuk menambahkan tugas bar uke dalam daftar. fungsi mengambil nilai input dari pengguna, kemudian membuat objek **Task** baru berdasarkan input tersebut, lalu menambahkannya ke dalam array **tasks**. Setelah itu, fungsi memanggil **displayTasks()** untuk memperbarui tampilan daftar tugas.



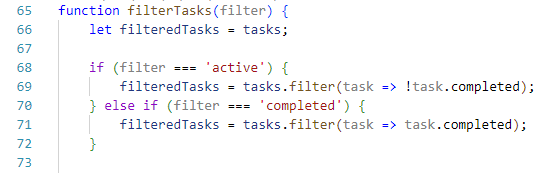
Fungsi **displayTasks()** digunakan untuk memperbarui daftar tugas yang ditampilan di DOM. elemen daftar (ul) dibersihkan, kemudian setiap tugas ditambahkan sebagai elemen baru dengan menggunakan elemen HTML seperti <li>, <span>, <input> (checkbox), dan <button> (delete).



Fungsi **toggleTask()** bertanggung jawab untuk mengubah status tugas (completed atau tidak). Ketika checkbox pada suatu tugas diklik, status tugas tersebut diubah menggunakan metode toggleCompleted(), lalu daftar tugas diperbarui menggunakan displayTasks().



Fungsi **deleteTask()** digunakan untuk menghapus tugas dari array **tasks** berdasarkan indeksnya. Setelah tugas dihapus, daftar tugas diperbarui dengan **displayTasks()**.



Fungsi **filterTasks()** digunakan untuk memfilter daftar tugas yang ditampilkan sesuai status (semua, aktif, atau selesai). Hasil filter ditampilkan dengan memperbarui daftar tugas di DOM.



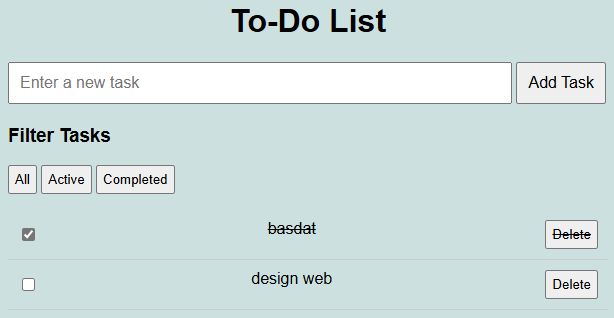
**Tombol "Add Task"**: Menggunakan **event listener** untuk mendeteksi klik tombol, lalu memanggil fungsi addTask() yang mengambil input pengguna dan menambahkannya ke daftar tugas.



**Checkbox**: Checkbox pada setiap tugas menangani event klik untuk mengubah status tugas.

# Hasil Uji Coba

Hasil uji coba yang saya lakukan adalah saya mencoba menambahkan task Kerjaan dan Game dimana Game sudah saya kerjakan sedangkan Kerjaan belum.

****

**KODING KESELURUHAN CSS**

body {

    font-family: Arial, sans-serif;

    margin: 0 auto;

    padding: 20px;

    max-width: 600px;

    background-image: url('https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcRf7fFHN-q7QGF-nWvakUYYeIGIZgC2p-mxnw&s');

    background-size: cover;

}

h1 {

    text-align: center;

}

#task-input {

    width: 80%;

    padding: 10px;

    font-size: 16px;

    margin-bottom: 2px;

}

#add-task-btn {

    padding: 10px;

    font-size: 16px;

}

ul {

    list-style-type: none;

    padding: 0;

}

li {

    display: flex;

    justify-content: space-between;

    padding: 10px;

    border-bottom: 1px solid #ccc;

}

li.completed {

    text-decoration: line-through;

}

button {

    padding: 5px;

}

**KODING KESELURUHAN 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);

    });

}

**Kesimpulan**

Praktikum ini berhasil membangun aplikasi To-Do List yang dapat menambah, menghapus, dan memfilter tugas. Implementasi JavaScript DOM, class, dan event handling memberikan pemahaman tentang bagaimana membuat aplikasi web interaktif.

LINK GITHUB :

<https://github.com/dacamokid/tugas4_dw_A_4523210034_DendyAnugrahiRabbi.git>