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

*Laporan ini disusun untuk memenuhi Tugas Keempat mata kuliah Desain Web*

**Dosen Pengampu : Adi Wahyu Pribadi, S.Si., M.Kom.**



oleh:

Nama : Naufal Maulana Saputra

NIM : 4523210083

Tanggal : 1 Oktober 2024

**PROGRAM STUDI S1-TEKNIK INFORMATIKA**

**FAKULTAS TEKNIK**

**UNIVERSITAS PANCASILA**

**2024/2025**

1. **Pendahuluan**

Aplikasi To-Do list adalah aplikasi yang Menggunakan HTML, CSS dan JavaScript DOM untuk mengambil, memanipulasi, dan memperbarui elemen HTML. Mengimplementasikan JavaScript Class dan Object dalam aplikasi yang dinamis. Memanfaatkan JavaScript function untuk mengatur logika aplikasi. Menerapkan event handling dalam interaksi pengguna dengan aplikasi.

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.

1. **Langkah Pengerjaan**

1. Membuat halaman HTML dan Styling dengan css (index.html)

<!DOCTYPE *html*>

<html *lang*="en">

<head>

<meta *charset*="UTF-8">

<meta *name*="viewport" *content*="width=device-width, initial-scale=1.0">

<title>To-Do List App</title>

<style>

body {

font-family: Arial, sans-serif;

margin: 0 auto;

padding: 20px;

max-width: 600px;

}

h1 {

text-align: center;

}

*#task-input* {

width: 80%;

padding: 10px;

font-size: 16px;

}

*#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 {

margin-left: 10px;

padding: 5px;

}

</style>

</head>

<body>

<h1>To-Do List</h1>

<input *type*="text" *id*="task-input" *placeholder*="Enter a new task">

<button *id*="add-task-btn">Add Task</button>

<h3>Filter Tasks</h3>

<button *onclick*="filterTasks('all')">All</button>

<button *onclick*="filterTasks('active')">Active</button>

<button *onclick*="filterTasks('completed')">Completed</button>

<ul *id*="task-list"></ul>

<script *src*="app.js"></script>

</body>

</html>

2. Class dan Object untuk Tugas (app.js)

// app.js

class Task {

constructor(*name*) {

*this*.name = *name*;

*this*.completed = false;

}

toggleCompleted() {

*this*.completed = !*this*.completed;

}

}

* Task memiliki atribut name dan completed.
* ToggleCompleted digunakan untuk menandai tugas sebagai selesai atau belum.

3. Membuat Fungsi untuk Menambah 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

}

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

4. Menampilkan Tugas di DOM

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);

});

}

5. Menandai Tugas Selesai dan Menghapus Tugas

function toggleTask(*index*) {

tasks[*index*].toggleCompleted();

displayTasks();

}

function deleteTask(*index*) {

tasks.splice(*index*, 1);

displayTasks();

}

6. File 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);

});

}

1. **Struktur dan Penjelasan Kode**

1. Class Task dan fungsinya

// app.js

class Task {

constructor(*name*) {

*this*.name = *name*;

*this*.completed = false;

}

toggleCompleted() {

*this*.completed = !*this*.completed;

}

}

Untuk mendeklarasikan atribut name, completed, dan merepresentasikan tugas

2. Fungsi addTask, toggleTask, deleteTask, dan filterTasks

* addTask

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

}

addTask sebagai penambah atau menambahkan task pada website To-Do List. tetapi pada fungsi ini terdapat pengecekan apabila task itu kosong maka tidak boleh untuk mengosongkan task. Setelah menuliskan task, task akan dibuat ke list dan akan ditampilkan list nya.

* toggleTask

function toggleTask(*index*) {

tasks[*index*].toggleCompleted();

displayTasks();

}

toggleTask sebagai penanda atau menandakan untuk task yang sudah dikerjakan.

* deleteTask

function deleteTask(*index*) {

tasks.splice(*index*, 1);

displayTasks();

}

deleteTask sebagai penghapus atau menghapus task yang sudah ada di dalam list.

* filterTasks

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);

});

}

filterTasks sebagai filter atau membatasi task pada list, seperti task yang sudah selesai atau belum selesai.

3. Cara penggunaan manipulasi DOM untuk menampilkan tugas.

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);

});

}

Cara penggunaannya dengan menampilkan task ketika sudah mengisi lalu mengklik button add task lalu akan dimasukkan kedalam list. Pada list kita bisa men-toggle apakah sudah selesai atau menghapus task.

4. Penjelasan event handling yang digunakan

* Button addTask

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

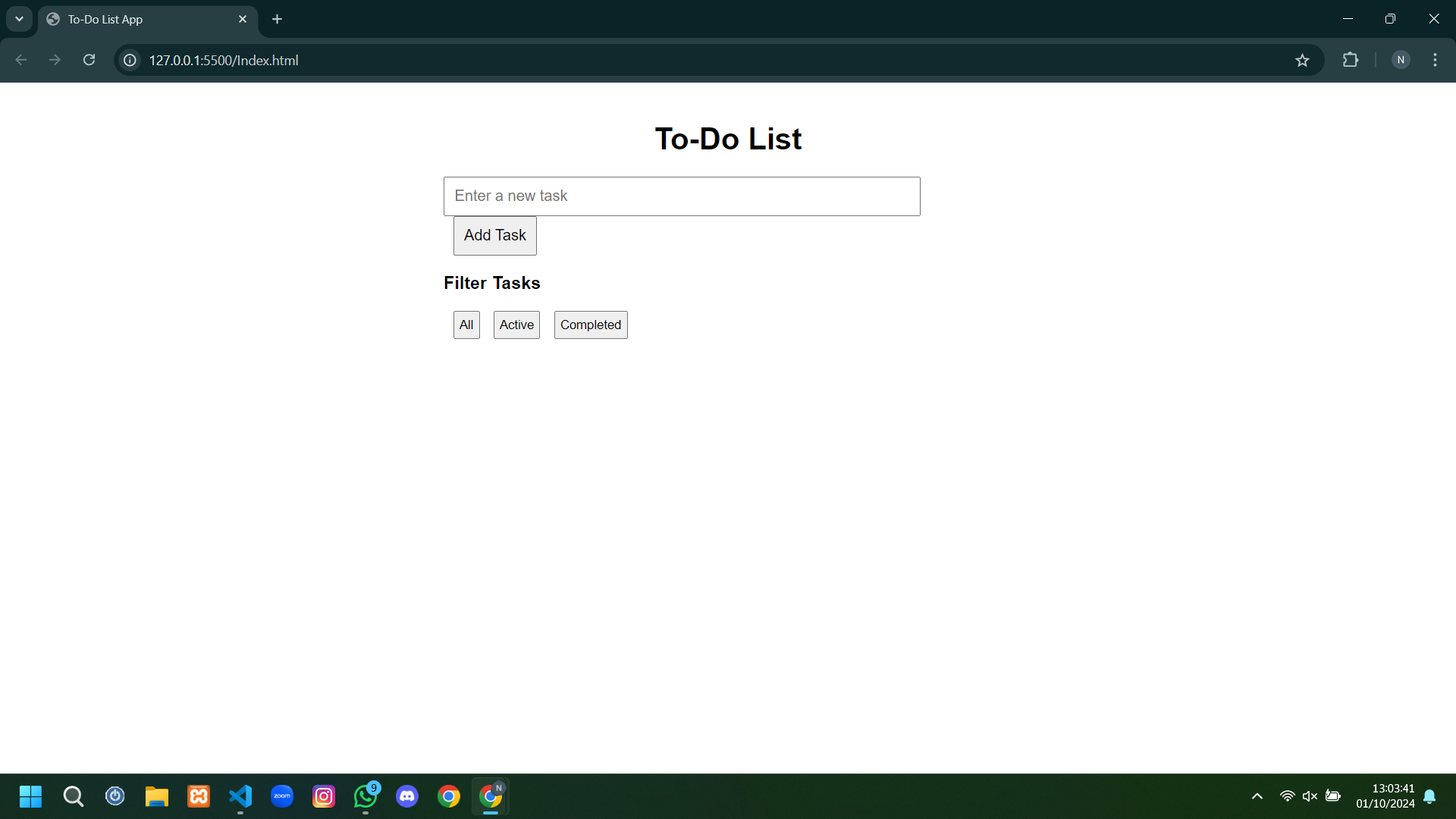
}

Setelah mengisi task dan meng-klik button, task akan menambahkan ke listnya.

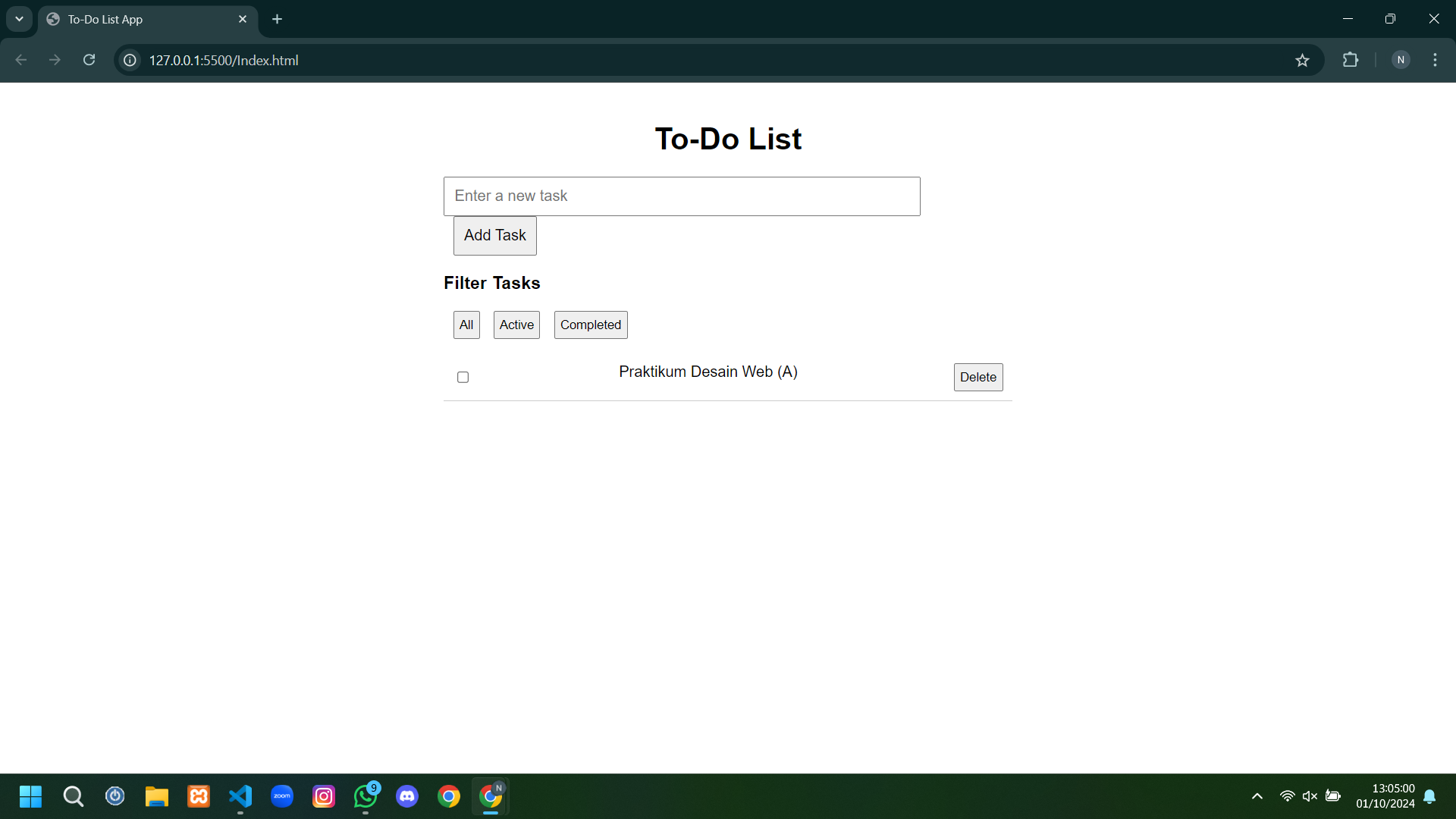
* Checkbox

Ketika meng-klik task yang di klik pada list terlihat seperti tercoret.

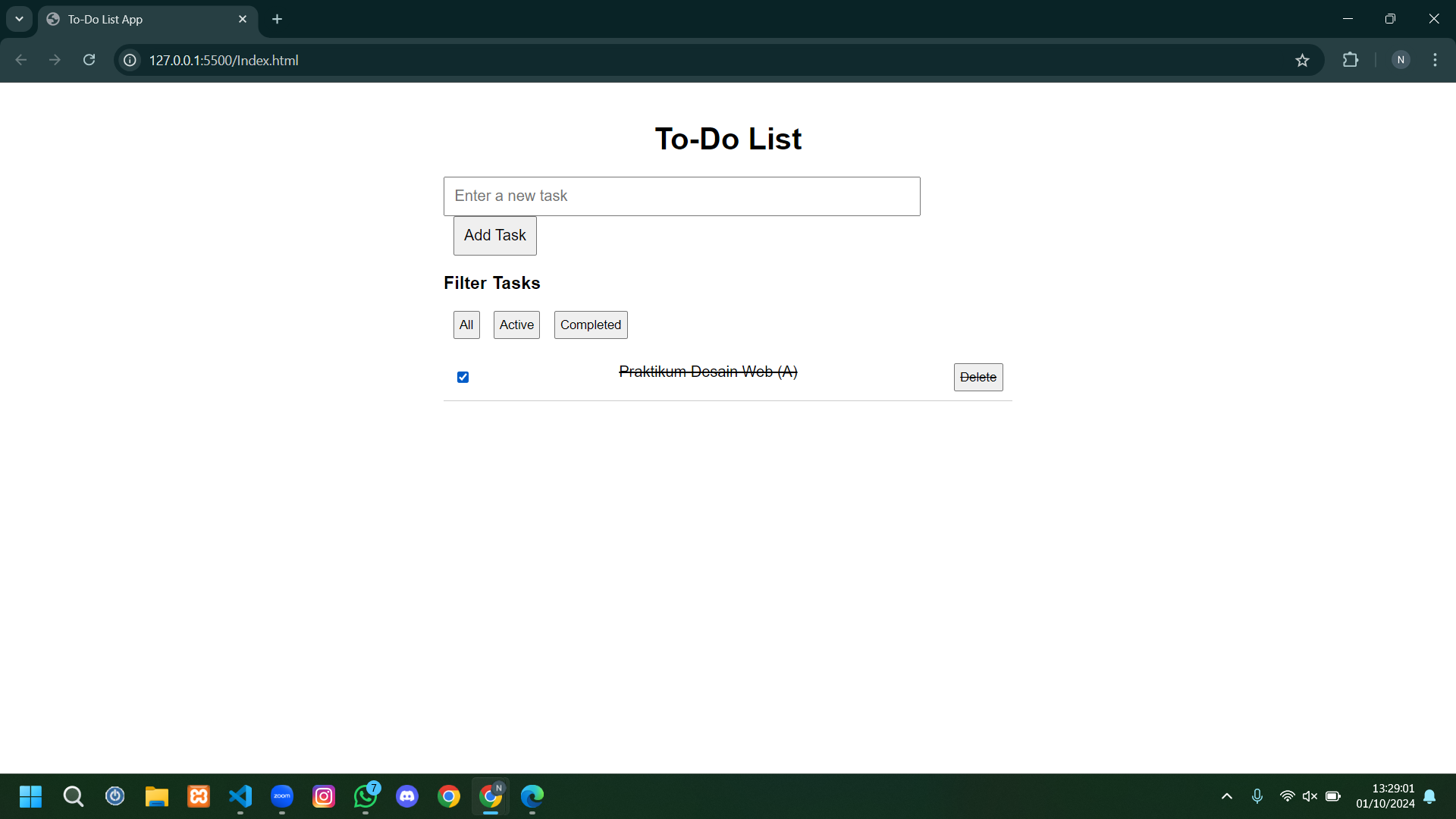
1. **Hasil Ujicoba**
2. Tampilan awal.



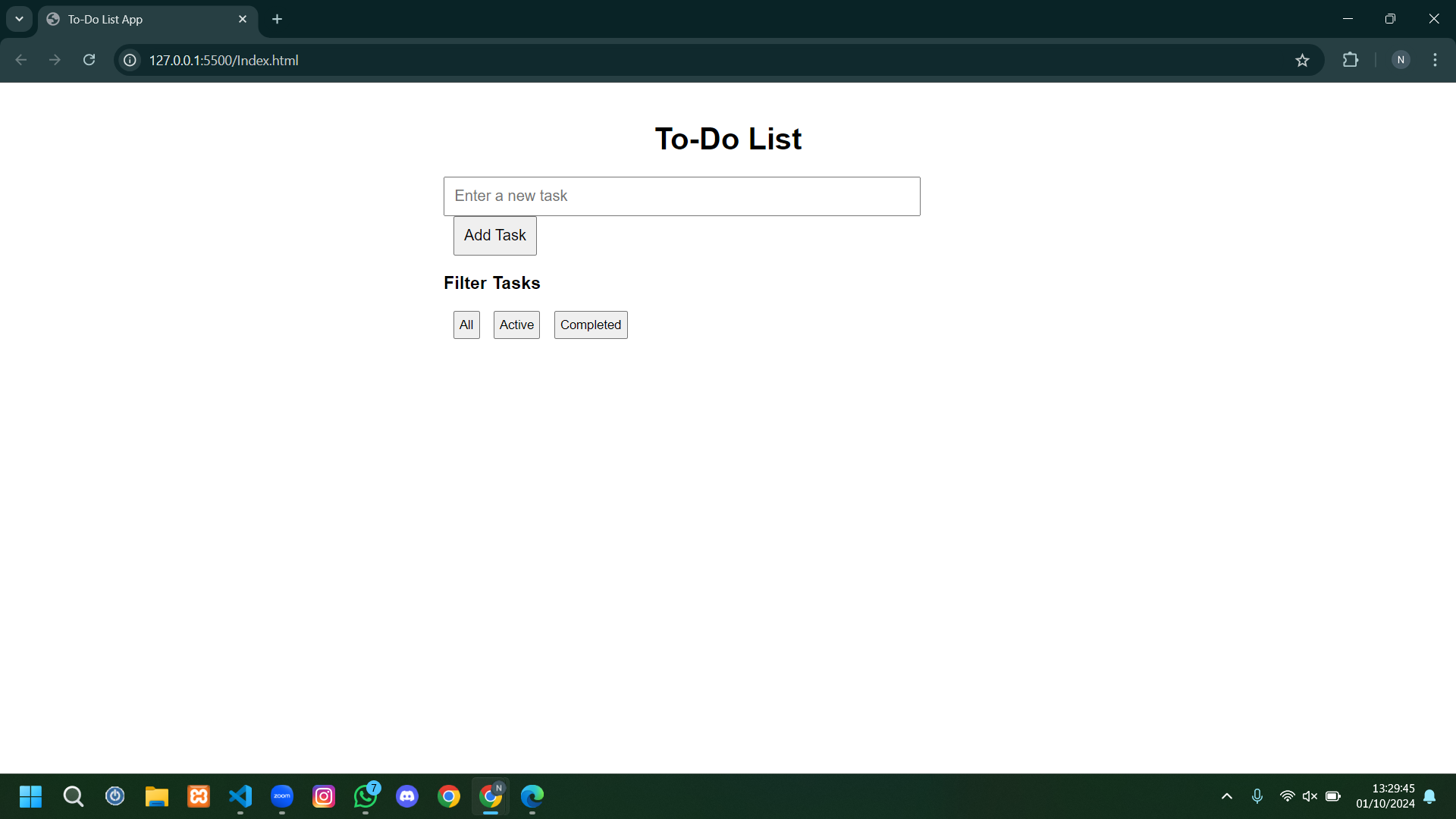
1. Mengisi task: menginput task jika ingin menambahkan tugas.



1. Menanda taskSelesai: checkbox di sebelah kiri tugas dapat digunakan untuk menandai tugas telah selesai.

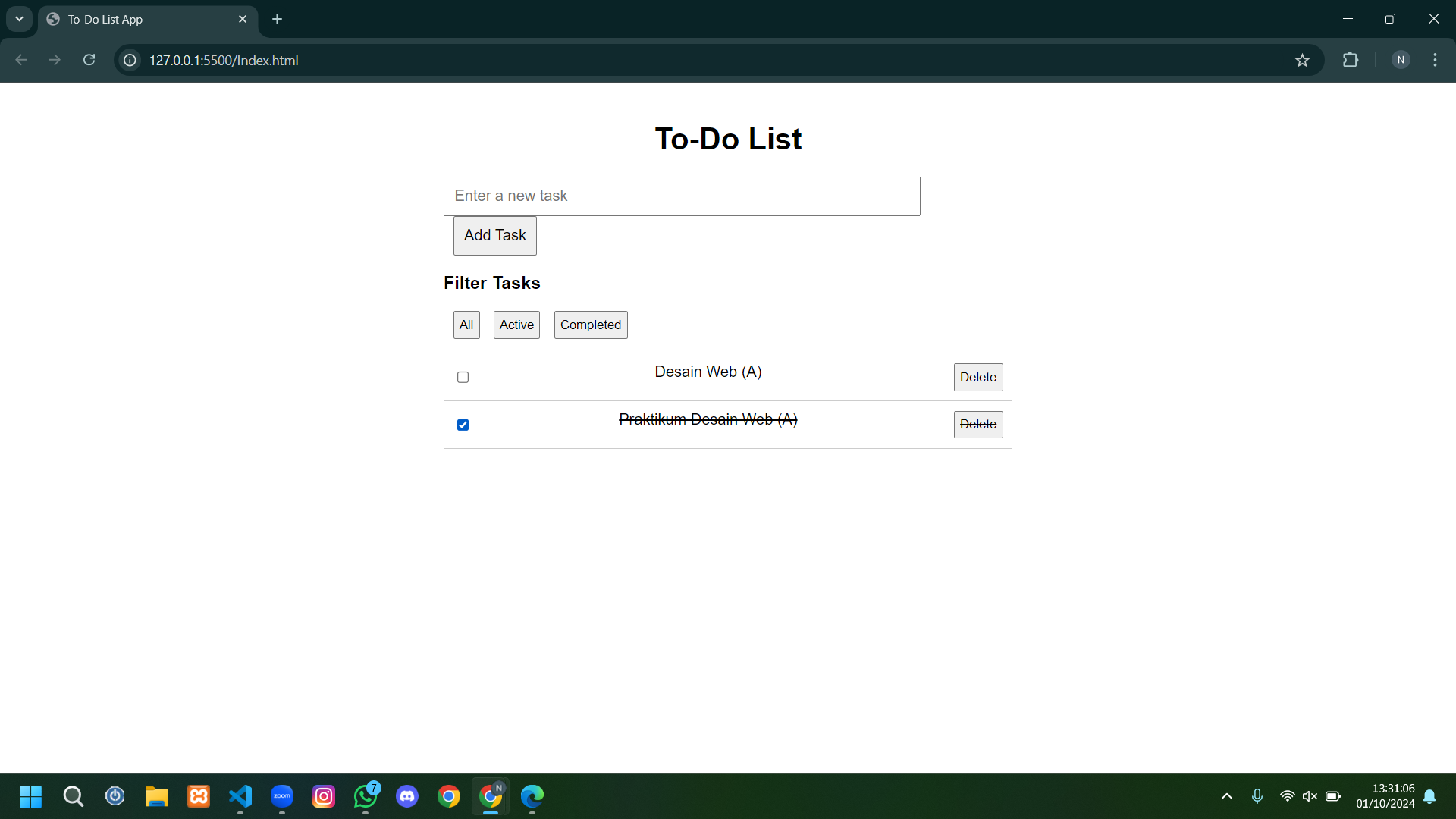


1. Menghapus task: Menekan tombol delete di sebelah kanan.

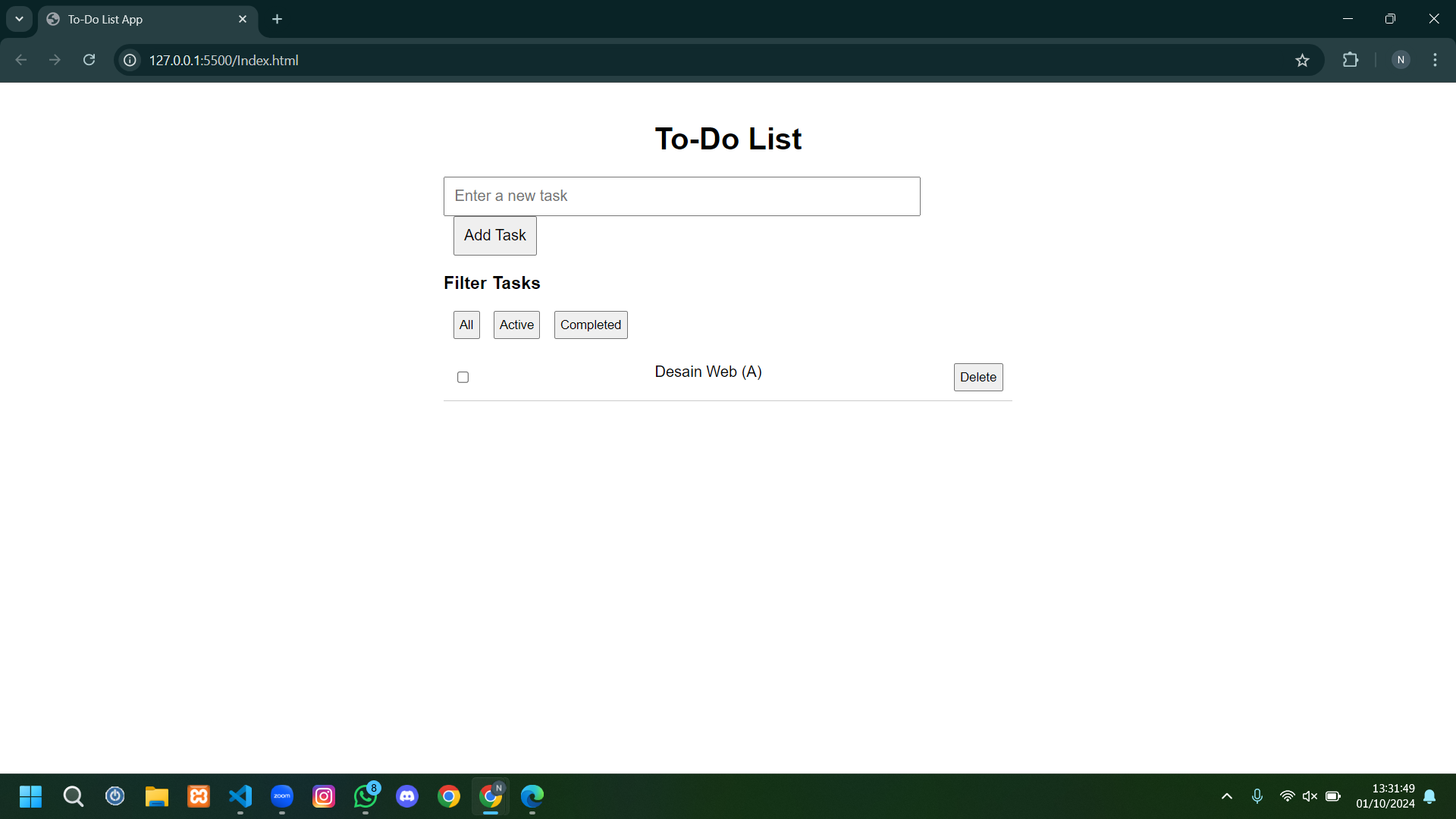


1. Filter task: memfilter tugas apakah tugas aktif, selesai, atau semua tugas.

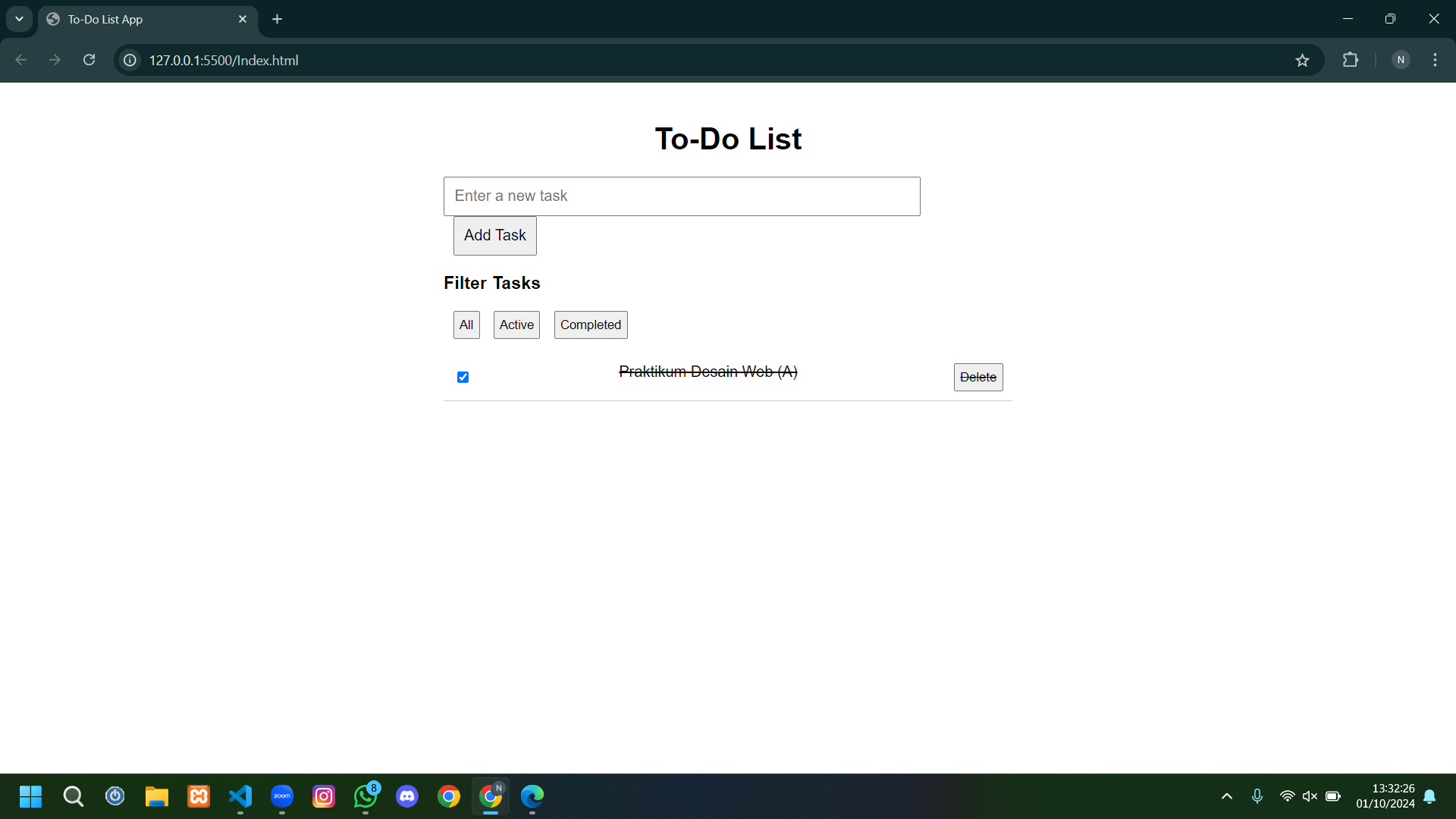
* ALL



* Active



* Completed



1. **Kesimpulan**

Aplikasi To-Do List ini berhasil dibuat dengan menerapkan dasar-dasar JavaScript seperti class, manipulasi DOM, dan event handling. Setiap fitur berjalan dengan baik, mulai dari penambahan tugas hingga filter tugas.