

Laporan Praktikum JavaScript – Aplikasi To-Do List



Dosen Pengampu :

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

Disusun oleh :

Mufti Fazli (4523210067)

Matakuliah :

Pratikum Desain Web_A

Hari dan tanggal :

Selasa 1 Oktober 2024

PROGRAM STUDI TEKNIK INFORMATIKA

UNIVERSITAS PANCASILA 2024

PENDAHULUAN

1.1 Penjelasan singkat tentang tujuan dari praktikum ini.

Pratikum ini Bertujuan untuk memberikan pemahaman dan keterampilan dalam melakukan perancangan sebuah secara langsung.

1.2 Penjelasan singkat tentang aplikasi To-Do-List.

Aplikasi To do list adalah aplikasi yang di rancang untuk membantu pengguna untuk melakukan list yang harus dilakukan olehnya ,fitur utama dari aplikasi ini bisa digunakan untuk membuat daftar tugas ,jadwal sehari hari dan sebagainya.

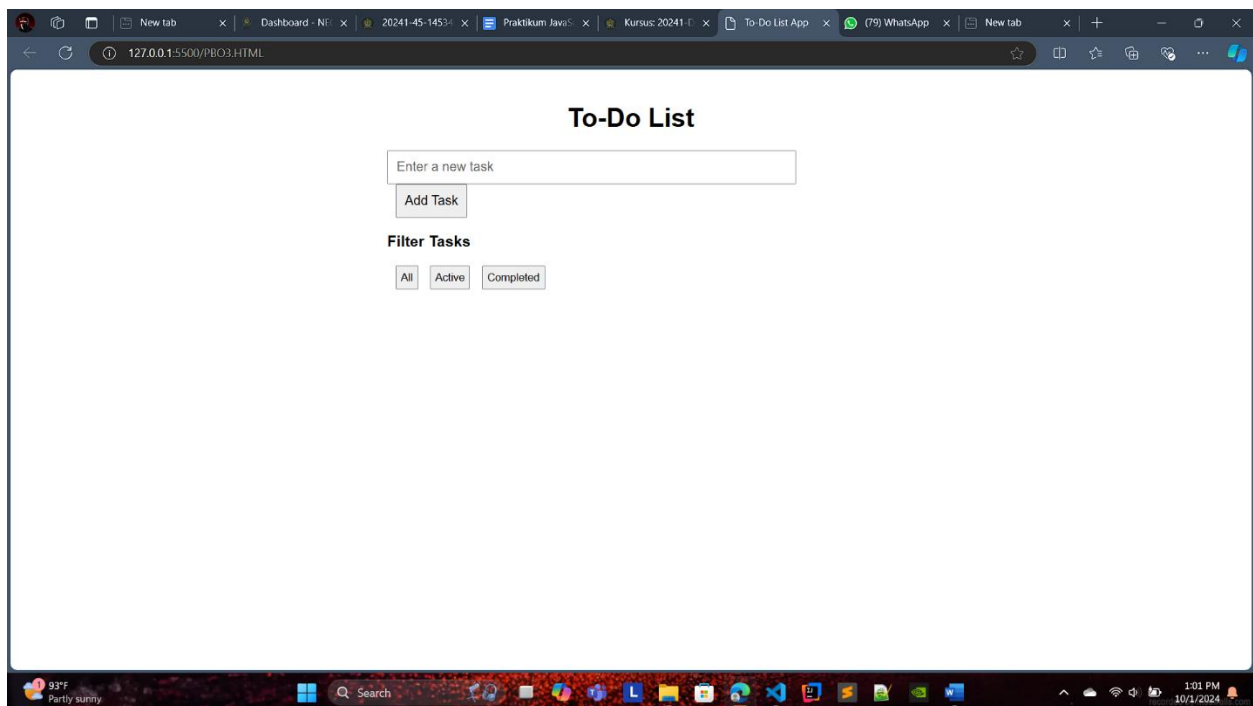
1.3 Penjelasan singkat tentang teori di praktikum ini.

LANGKAH DALAM Pengerjaan

2.1 Penjelasan tahapan dari HTML,CSS,JS dalam praktikum ini

Langkah pertama menyiapkan folder DW 3,yang kedua membuat file HTML,yang ketiga memasukan code ke dalam file HTML,yang ketiga memasukan code CSS kedalam HTML,selanjutnya membuat file Java Script dan memasukan code java ke dalam Java Script di dalam JS, membuat class task,membuat fungsi menambahkan task,membuat fungsi menambahkan task,membuat fungsi menampilkan task ,mengubah status selesai dan menghapus task,membuat filter task dan yang terakhir start/debugging.

2.2 Screenshot / hasil dari program yang telah dibuat



2.3 Codigan

STRUKTUR DAN PENJELASAN CODE

3.1 kode Java Script

```
// 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));

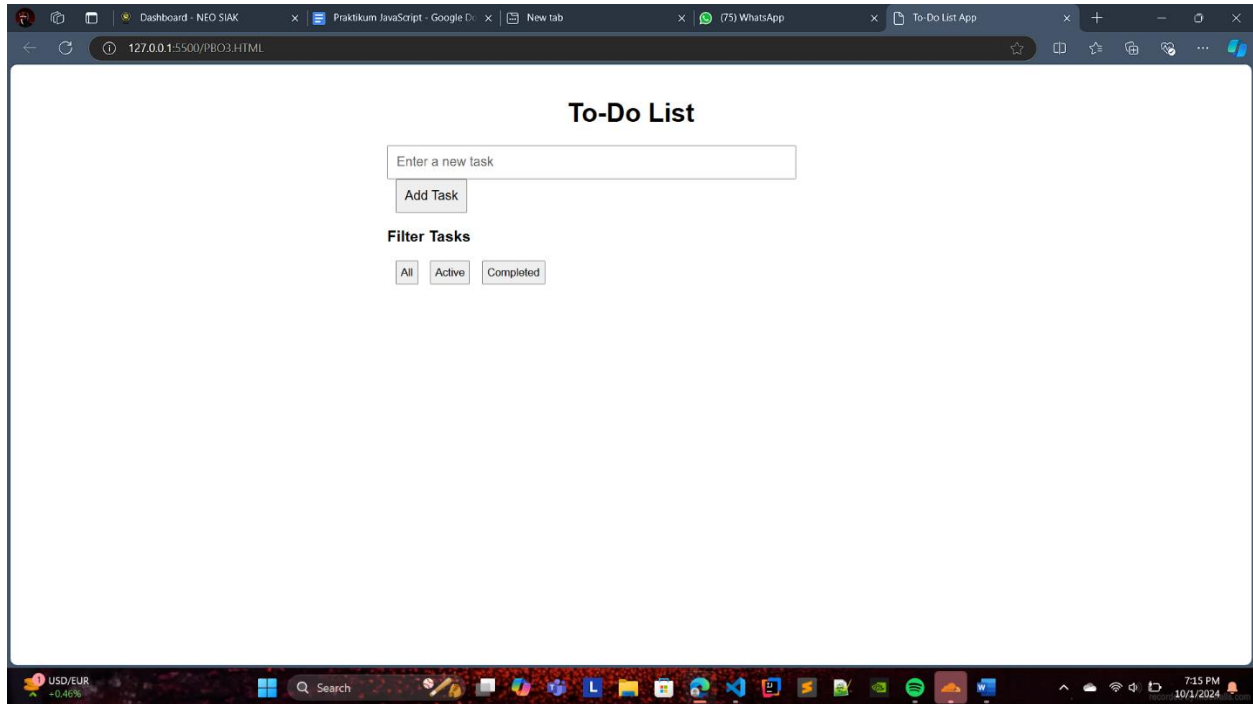
taskItem.appendChild(taskCheckbox);
taskItem.appendChild(taskText);
taskItem.appendChild(deleteButton);

taskList.appendChild(taskItem);
});
}

```

- **Class task**
Untuk mempresentasikan setiap task dengan property name dan completed
- **Fungsi addTask, toggleTask, deleteTask, dan filterTasks. :**
Fungsi add task bertanggung jawab untuk menambahkan task baru kedalam daftar tugas, fungsi toggle task bertanggung jawab untuk mengubah status completed dari sebuah task
- **Cara penggunaan manipulasi DOM untuk menampilkan tugas.**
Manipulasi DOM digunakan untuk menambah, mengubah, atau menghapus elemen HTML secara dinamis melalui JavaScript. Dalam aplikasi To-Do List, manipulasi DOM digunakan untuk menampilkan daftar tugas (tasks) yang dikelola dalam JavaScript.
- **Penjelasan event handling yang digunakan, seperti pada tombol “Add Task” dan checkbox.**
Event handling adalah proses menangani kejadian atau interaksi pengguna dengan elemen-elemen di halaman web, seperti klik tombol, perubahan input, atau tindakan lainnya.

HASIL UJI COBA



1. Dalam kolom kotak yg betruliskan (Enter A new task) itu berfungsi untuk memasukan list yang akan dilakukan
2. Add task untuk menambhakan list
3. Filter tasks, adalah filter yang bis akita pilih

KESIMPULAN

4.1

Program To-Do List App adalah aplikasi sederhana berbasis web yang dibangun menggunakan HTML, CSS, dan JavaScript. Aplikasi ini memungkinkan pengguna untuk menambahkan, menghapus, menandai selesai, serta memfilter tugas (tasks) berdasarkan statusnya (aktif, selesai, atau semua).