

LAPORAN
PEMROGRAMAN WEB TEORI



DANIS WARA WARDANA

5220411216

PROGRAM STUDI INFORMATIKA S1
FAKULTAS SAINS & TEKNOLOGI
UNIVERSITAS TEKNOLOGI YOGYAKARTA
2023

1. Penjelasan dari codingan script.js

Program di vscode adalah implementasi sederhana dari aplikasi manajemen tugas (todos) menggunakan pola arsitektur Model-View-Controller (MVC) dalam konteks JavaScript.

Berikut adalah penjelasan kegunaan dari setiap baris kode:

Model (Class: Model)

`constructor()`: Membuat instance dari Model. Menginisialisasi `this.todos` dengan data dari local storage atau array kosong jika tidak ada data.

`bindTodoListChanged(callback)`: Memungkinkan pengikat (callback) untuk mendaftar dan mendengarkan perubahan pada daftar todos.

`_commit(todos)`: Menyimpan perubahan ke local storage dan memanggil callback yang terdaftar untuk memberi tahu bahwa daftar todos telah berubah.

`addTodo(todoText)`: Menambahkan tugas baru ke daftar todos, kemudian menyimpan perubahan.

`editTodo(id, updatedText)`: Mengedit teks tugas dengan ID yang diberikan, kemudian menyimpan perubahan.

`deleteTodo(id)`: Menghapus tugas dengan ID yang diberikan, kemudian menyimpan perubahan.

`toggleTodo(id)`: Mengganti status tugas (selesai/belum selesai) dengan ID yang diberikan, kemudian menyimpan perubahan.

View (Class: View)

`constructor()`: Membuat instance dari View. Menginisialisasi elemen-elemen HTML yang akan digunakan untuk menampilkan tugas.

`_initLocalListeners()`: Mendengarkan input pengguna pada elemen-elemen lokal (seperti tombol hapus dan checkbox) untuk mengelola sementara teks tugas.

`bindAddTodo(handler)`: Menyambungkan handler untuk menangani penambahan tugas saat formulir disubmit.

`bindDeleteTodo(handler)`: Menyambungkan handler untuk menangani penghapusan tugas saat tombol hapus ditekan.

`bindEditTodo(handler)`: Menyambungkan handler untuk menangani pengeditan tugas saat teks tugas berubah (focus out).

`bindToggleTodo(handler)`: Menyambungkan handler untuk menangani perubahan status tugas (selesai/belum selesai) saat checkbox diubah.

`displayTodos(todos)`: Menampilkan daftar tugas ke dalam elemen HTML, dengan menghapus elemen-elemen sebelumnya dan membuat elemen-elemen baru.

Controller (Class: Controller)

`constructor(model, view)`: Membuat instance dari Controller. Menghubungkan Model dan View, serta menyambungkan handler untuk setiap perubahan yang mungkin terjadi pada aplikasi.

`onTodoListChanged(todos)`: Handler yang dipanggil ketika daftar todos berubah, untuk menampilkan perubahan di View.

`handleAddTodo(todoText)`: Handler untuk menangani penambahan tugas yang dipicu oleh pengguna.

`handleEditTodo(id, todoText)`: Handler untuk menangani pengeditan tugas yang dipicu oleh pengguna.

`handleDeleteTodo(id)`: Handler untuk menangani penghapusan tugas yang dipicu oleh pengguna.

`handleToggleTodo(id)`: Handler untuk menangani perubahan status tugas yang dipicu oleh pengguna.

Inisialisasi Aplikasi

`const app = new Controller(new Model(), new View())`: Membuat instance dari Controller dengan menginisialisasi Model dan View. Hal ini memulai aplikasi dan menampilkan daftar todos awal.

Top of Form

2. Penjelasan dari `script.min.js`

Baris kode program di atas merupakan implementasi aplikasi To-Do List menggunakan pola arsitektur Model-View-Controller (MVC) dalam JavaScript. Berikut adalah penjelasan kegunaan dari setiap baris kode:

Model

`"use strict";`: Mengaktifkan mode strict untuk memberikan kontrol yang lebih ketat terhadap kesalahan dan menghindari beberapa kesalahan umum dalam JavaScript.

`Function _defineProperty(obj, key, value)`: Fungsi utilitas untuk mendefinisikan properti pada objek, khususnya digunakan untuk mendefinisikan properti pada objek dengan cara yang memastikan properti tersebut dapat diiterasi.

`Function _instanceof(left, right)`: Fungsi utilitas untuk menentukan apakah objek di sebelah kiri (left) merupakan instance dari kelas atau fungsi tertentu yang ditentukan oleh parameter di sebelah kanan (right).

`Function _classCallCheck(instance, Constructor)`: Fungsi utilitas yang memeriksa apakah instance dari suatu kelas telah dibuat menggunakan operator `new`. Jika tidak, maka akan melempar `TypeError`.

`Function _defineProperties(target, props)`: Fungsi utilitas untuk mendefinisikan satu atau lebih properti pada objek target.

`Class Model`: Representasi dari model dalam pola MVC. Mengelola data aplikasi, seperti menambah, mengedit, menghapus, dan mengganti status tugas.

View

`"use strict";`: Mode strict yang diaktifkan pada bagian View juga, memberikan kontrol yang lebih ketat terhadap kesalahan.

`Class View`: Representasi visual dari model dalam pola MVC. Menangani elemen-elemen HTML untuk menampilkan dan mengelola tugas-tugas.

Controller

"use strict"; Mode strict yang diaktifkan pada bagian Controller juga, memberikan kontrol yang lebih ketat terhadap kesalahan.

Class Controller: Menghubungkan input pengguna (dari View) dan output visual (dari Model). Menyambungkan metode dari Model dan View serta menginisialisasi aplikasi dengan menampilkan tugas-tugas awal.

Instance Creation

var app = new Controller(new Model(), new View());: Membuat instance dari Controller, Model, dan View, sehingga memulai eksekusi aplikasi To-Do List dengan menghubungkan komponen-komponennya.

Top of Form

3. Penjelasan dari index.html

`<!DOCTYPE html>`: Deklarasi tipe dokumen HTML. Memberi tahu browser bahwa dokumen ini adalah dokumen HTML5.

`<html lang="en">`: Menandakan awal dari elemen HTML. Atribut lang digunakan untuk menentukan bahasa dokumen, dalam hal ini, bahasa Inggris (English).

`<head>`: Elemen yang berisi meta-informasi tentang dokumen seperti karakter set, viewport, dan informasi lainnya yang tidak ditampilkan pada halaman web itu sendiri.

`<meta charset="utf-8">`: Menentukan karakter set dokumen sebagai UTF-8, yang mendukung karakter internasional.

`<meta name="viewport" content="width=device-width, initial-scale=1.0">`: Memberikan instruksi kepada browser untuk menyesuaikan tampilan halaman web sesuai dengan lebar perangkat dan menetapkan skala awal ke 1.0.

`<meta http-equiv="X-UA-Compatible" content="ie=edge">`: Menentukan mode kompatibilitas Internet Explorer untuk menggunakan rendering terbaru.

`<title>To Do List App</title>`: Menetapkan judul halaman web yang akan ditampilkan di tab browser.

`<link rel="stylesheet" href="style.css">`: Menghubungkan dokumen HTML dengan file stylesheet eksternal (style.css), yang digunakan untuk memperindah tata letak dan gaya visual halaman.

`<body>`: Elemen yang berisi konten halaman web yang akan ditampilkan kepada pengguna.

`<div id="root"></div>`: Elemen div dengan id "root" yang akan digunakan sebagai tempat untuk me-render isi dari aplikasi To-Do List. Hal ini sesuai dengan penggunaan #root pada file JavaScript untuk menyematkan elemen-elemen HTML yang dibuat dinamis oleh aplikasi.

`<script src="script.min.js"></script>`: Menghubungkan dokumen HTML dengan file script JavaScript eksternal (script.min.js), yang berisi logika aplikasi To-Do List.

`</body>`: Menandakan akhir dari elemen body.

`</html>`: Menandakan akhir dari elemen HTML.

