

Laporan Praktikum JavaScript

Aplikasi To-Do List

Untuk memenuhi Tugas Praktikum Desain Web



Disusun Oleh :

NAMA : AIDA FITRIA

NPM : 4523210007

Tanggal Praktikum : 01/10/2024

Dosen:

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

PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS TEKNIK

- **Pendahuluan**

Tujuan praktikum ini adalah untuk memahami konsep dasar dari pembuatan aplikasi berbasis web yang interaktif dengan menggunakan html,css dan javascript. Dan praktikum ini berfokus pada pengembangan to-do list yang berfungsi sebagai menambahkan,menghapus serta memberikan status selesai.

Aplikasi to-do list yang dibuat untuk mengelola daftar list. Fitur utama terdapat tugas baru/list baru,penandaan tugas selesai,dan tugas sudah selesai.

Penjelasan html dari praktikum ini digunakan untuk struktur data seperti elemen input dan Css digunakan untuk menata tampilan aplikasi, sedangkan javascript digunakan untuk menangani logika aplikasi seperti penambahan tugas ke list/daftar,perubahan status,dan penghapusan tugas.

• Langkah Pengerjaan

1. Membuat struktur code html yang terdapat input untuk menambahkan daftar yang ditampilkan dengan elemen ul

```
<h1>List Checkin Hotel</h1>
<input type="text" id="task-input" placeholder="Nomor Kamar">
<button id="add-task-btn">Setuju</button>
```

2. Membuat struktur code css yang terdapat input untuk menambahkan desain yang ditampilkan agar antarmuka mudah digunakan

```
1 body {
2     background-image: url('aga.jpeg');
3     background-size: cover;
4     font-family: Arial, sans-serif;
5     margin: 0 auto;
6     padding: 20px;
7     max-width: 600px;
8 }
9 h1 {
10     text-align: center;
11     color: #aliceblue;
12 }
13 #task-input {
14     width: 80%;
15     padding: 10px;
16     font-size: 16px;
17 }
18 #add-task-btn {
19     padding: 10px;
20     font-size: 16px;
21 }
22 ul {
23     list-style-type: none;
24     padding: 0;
25 }
26 li {
27     color: #f0eae6;
28     display: flex;
29     justify-content: space-between;
30     padding: 10px;
31     border-bottom: 1px solid #f0eae6;
32 }
```

```
17 }
18 #add-task-btn {
19     padding: 10px;
20     font-size: 16px;
21 }
22 ul {
23     list-style-type: none;
24     padding: 0;
25 }
26 li {
27     color: #f0eae6;
28     display: flex;
29     justify-content: space-between;
30     padding: 10px;
31     border-bottom: 1px solid #f0eae6;
32 }
33 li.completed {
34     text-decoration: line-through;
35 }
36 button {
37     margin-left: 10px;
38     padding: 5px;
39 }
40 h3 {
41     color: #aliceblue;
42 }
```

3. Membuat class task di javascript

```
1  // app.js
2  class Task {
3      constructor(name) {
4          this.name = name;
5          this.completed = false;
6      }
7
8      toggleCompleted() {
9          this.completed = !this.completed;
10     }
11 }
12 let tasks = [];
```

4. Menambahkan function

```
12 let tasks = [];
13
14 function addTask() {
15     const taskInput = document.getElementById('task-input');
16     const taskName = taskInput.value.trim();
17
18     if (taskName === "") {
19         alert("Task cannot be empty!");
20         return;
21     }
22
23     const task = new Task(taskName);
24     tasks.push(task);
25     displayTasks();
26     taskInput.value = ''; // Kosongkan input setelah menambahkan
27 }
```

5. Menambahkan daftar di dom

```
30 function displayTasks() {
31     const taskList = document.getElementById('task-list');
32     taskList.innerHTML = '';
33
34     tasks.forEach((task, index) => {
35         const taskItem = document.createElement('li');
36         taskItem.className = task.completed ? 'completed' : '';
37
38         const taskText = document.createElement('span');
39         taskText.textContent = task.name;
40
41         const taskCheckbox = document.createElement('input');
42         taskCheckbox.type = 'checkbox';
43         taskCheckbox.checked = task.completed;
44         taskCheckbox.addEventListener('click', () => toggleTask(index));
45
46         const deleteButton = document.createElement('button');
47         deleteButton.textContent = 'Delete';
48         deleteButton.addEventListener('click', () => deleteTask(index));
49
50         taskItem.appendChild(taskCheckbox);
51         taskItem.appendChild(taskText);
52         taskItem.appendChild(deleteButton);
53
54         taskList.appendChild(taskItem);
55     });
56 }
57 function toggleTask(index) {
58     tasks[index].toggleCompleted();
59     displayTasks();
60 }
```

6. Menghapus daftar

```
57 function toggleTask(index) {
58     tasks[index].toggleCompleted();
59     displayTasks();
60 }
61
62 function deleteTask(index) {
63     tasks.splice(index, 1);
64     displayTasks();
65 }
```

7. Menambahkan fitur filter tugas seperti filterTask

```
66 function filterTasks(filter) {
67     let filteredTasks = tasks;
68
69     if (filter === 'active') {
70         filteredTasks = tasks.filter(task => !task.completed);
71     } else if (filter === 'completed') {
72         filteredTasks = tasks.filter(task => task.completed);
73     }
74
75     const taskList = document.getElementById('task-list');
76     taskList.innerHTML = '';
77
78     filteredTasks.forEach((task, index) => {
79         const taskItem = document.createElement('li');
80         taskItem.className = task.completed ? 'completed' : '';
81
82         const taskText = document.createElement('span');
83         taskText.textContent = task.name;
84
85         const taskCheckbox = document.createElement('input');
86         taskCheckbox.type = 'checkbox';
87         taskCheckbox.checked = task.completed;
88         taskCheckbox.addEventListener('click', () => toggleTask(index));
89
90         const deleteButton = document.createElement('button');
91         deleteButton.textContent = 'Delete';
92         deleteButton.addEventListener('click', () => deleteTask(index));
93
94         taskItem.appendChild(taskCheckbox);
95         taskItem.appendChild(taskText);
96         taskItem.appendChild(deleteButton);
97         taskList.appendChild(taskItem);
98     });
99 }
100 }
```

- Struktur dan Penjelasan Kode

1. Class task dan fungsinya =>

Task digunakan merepresentasikan tugas dan memiliki 2 atribut yaitu name dan completed.

```
class Task {  
  constructor(name) {  
    this.name = name;  
    this.completed = false;  
  }  
  
  toggleCompleted() {  
    this.completed = !this.completed;  
  }  
}  
  
let tasks = [];
```

2. Fungsi add task =>

Untuk menambah tugas baru kedalam daftar

```
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  
}
```

3. Fungsi toggle task =>

Menerima indeks dalam array task

```
function toggleTask(index) {  
  tasks[index].toggleCompleted();  
  displayTasks();  
}
```

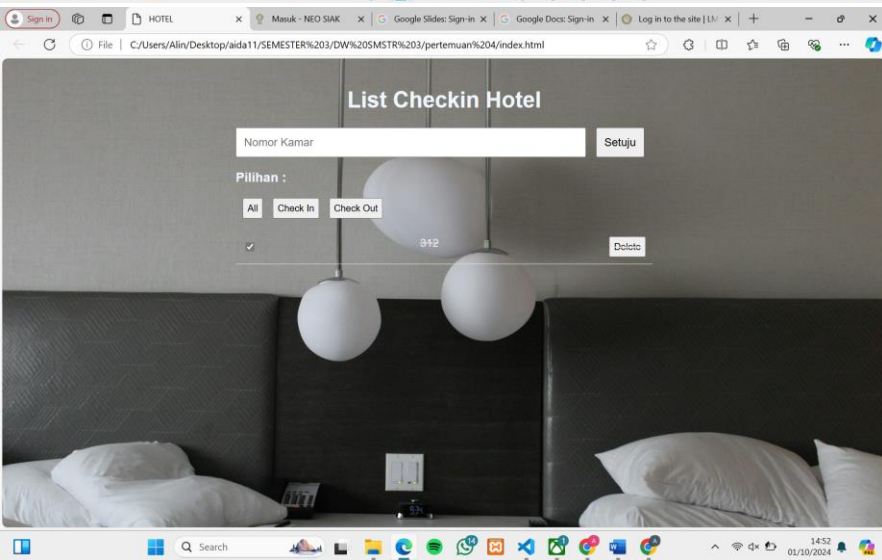
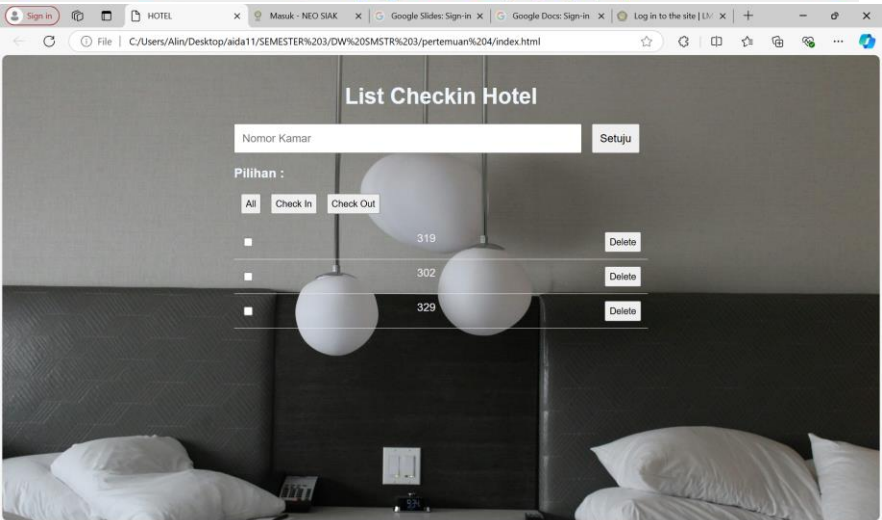
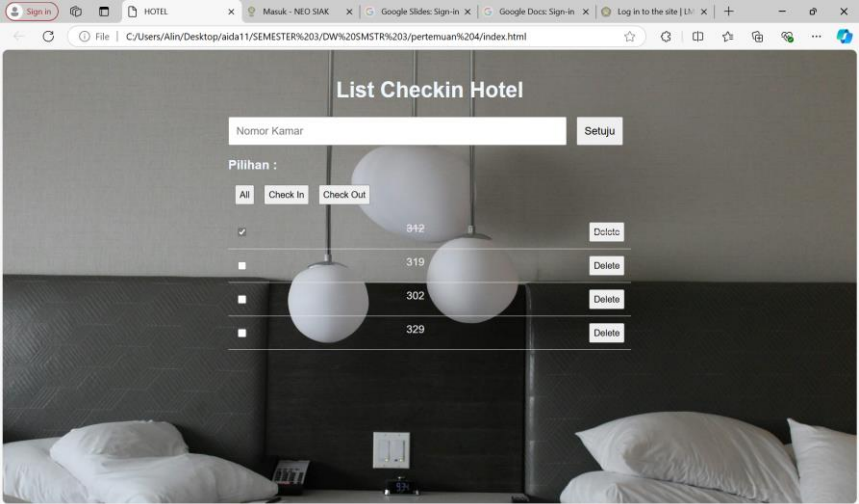
4. Fungsi filtertask

```
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 = '';
```

5. manipulasi dom

```
# style.css > h3  
1  body {  
2  
3  }  
4  h1 {  
5    text-align: center;  
6    color: #aliceblue;  
7  }  
8  #task-input {  
9    width: 80%;  
10   padding: 10px;  
11   font-size: 16px;  
12 }  
13 #add-task-btn {  
14   padding: 10px;  
15   font-size: 16px;  
16 }  
17 ul {  
18   list-style-type: none;  
19   padding: 0;  
20 }  
21 li {  
22   color: #f0eaeae;  
23   display: flex;  
24   justify-content: space-between;  
25   padding: 10px;  
26   border-bottom: 1px solid #f0eaeae;  
27 }  
28 li.completed {  
29   text-decoration: line-through;  
30 }  
31 button {  
32   margin-left: 10px;  
33   padding: 5px;  
34 }  
35 h3
```

• Hasil uji coba



•Kesimpulan

Aplikasi to-do list yang kita buat merupakan contoh dari penggunaan html,css dan javascript untuk membangun web yang interaktif, aplikasi to-do list ini adalah contoh implementasi javascript yang efisien untuk menciptakan aplikasi berbasis web dengan fitur yang mudah dioperasikan.