

**Kelompok :**

**Nama : Andri Sarwono                      NIM: 23EO50008**

**Nama : Aelia Hayuningsih                NIM: 23EO50002**

# **Laporan Pengembangan Aplikasi To-Do List dengan Fitur CRUD dan Manajemen Login**

## **1. Latar Belakang**

Dalam dunia yang serba cepat dan padat aktivitas saat ini, manajemen waktu menjadi sangat penting. Aplikasi **To-Do List** hadir sebagai solusi untuk membantu pengguna dalam mengorganisasi tugas dan pekerjaan mereka. Dengan aplikasi To-Do List, pengguna dapat membuat daftar tugas yang perlu diselesaikan, mengeditnya, dan menandai tugas yang telah selesai. Selain itu, untuk memberikan pengalaman pengguna yang lebih baik, aplikasi ini juga dilengkapi dengan fitur login, di mana pengguna dapat masuk dan mengakses daftar tugas mereka secara pribadi.

Fitur-fitur utama dalam aplikasi ini mencakup **CRUD (Create, Read, Update, Delete)** untuk tugas, serta manajemen login untuk memastikan bahwa hanya pengguna terautentikasi yang dapat mengakses daftar tugas mereka.

## **2. Tujuan**

Tujuan dari pengembangan aplikasi To-Do List ini adalah untuk:

1. Menyediakan aplikasi yang memungkinkan pengguna untuk membuat, membaca, memperbarui, dan menghapus tugas (CRUD).
2. Menyediakan sistem login yang aman bagi pengguna untuk mengakses tugas mereka secara pribadi.
3. Membantu pengguna dalam mengelola tugas mereka agar lebih produktif.
4. Meningkatkan pemahaman mengenai penerapan konsep **SDLC (Software Development Life Cycle)** dalam pengembangan perangkat lunak berbasis web.

## **3. Analisis Kebutuhan**

### **3.1 User Stories**

1. **Sebagai pengguna**, saya ingin bisa **login** ke aplikasi, agar saya bisa mengakses daftar tugas saya secara pribadi.
2. **Sebagai pengguna**, saya ingin bisa **menambahkan tugas** baru, agar saya dapat mencatat apa saja yang perlu saya lakukan.

3. **Sebagai pengguna**, saya ingin bisa **melihat tugas yang ada**, agar saya bisa memantau apa yang harus dilakukan.
4. **Sebagai pengguna**, saya ingin bisa **mengedit tugas**, agar saya bisa memperbarui detail tugas jika ada perubahan.
5. **Sebagai pengguna**, saya ingin bisa **menghapus tugas**, agar saya bisa menghapus tugas yang sudah selesai atau tidak diperlukan lagi.
6. **Sebagai pengguna**, saya ingin bisa **menandai tugas sebagai selesai**, agar saya bisa mengetahui tugas yang telah diselesaikan.
7. **Sebagai pengguna**, saya ingin bisa **logout** dari aplikasi, agar saya bisa keluar dengan aman setelah menggunakan aplikasi.

### 3.2 Model SDLC

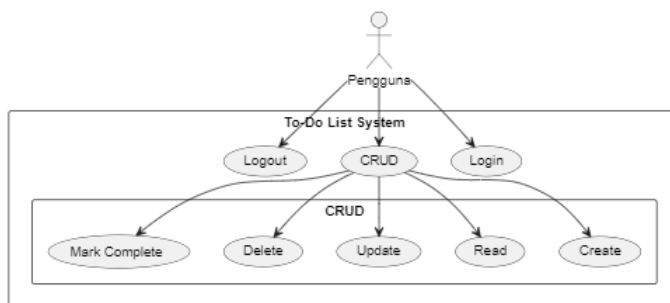
Model **Waterfall** digunakan dalam pengembangan aplikasi ini. Tahapan-tahapan yang dilalui adalah:

1. **Analisis Kebutuhan:** Mengumpulkan informasi mengenai fitur yang dibutuhkan dalam aplikasi.
2. **Desain Sistem:** Mendesain sistem yang mencakup diagram alur, struktur database, dan antarmuka pengguna.
3. **Pengembangan Sistem:** Implementasi fitur CRUD dan manajemen login.
4. **Pengujian:** Menguji setiap fitur aplikasi untuk memastikan semuanya bekerja dengan baik.
5. **Pemeliharaan:** Memperbaiki bug atau masalah yang ditemukan setelah aplikasi diluncurkan.

## 4. Desain Sistem

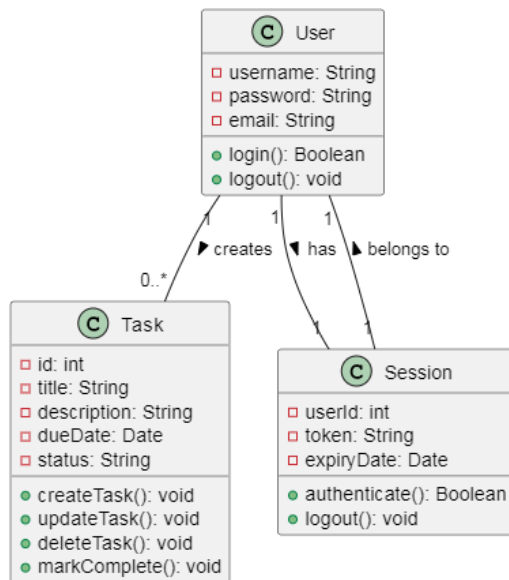
### 4.1 Use Case Diagram

Diagram ini menggambarkan interaksi antara aktor (Pengguna) dan sistem **To-Do List**, dengan fitur utama yang meliputi login, CRUD (Create, Read, Update, Delete), dan logout.



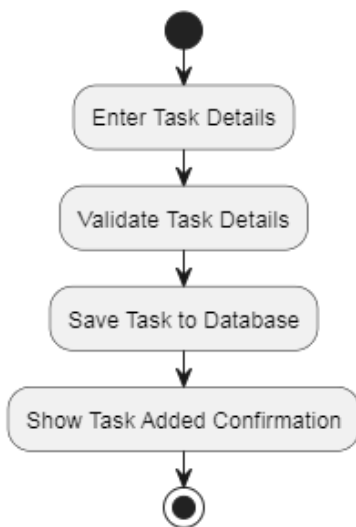
## 4.2 Class Diagram

Diagram ini menggambarkan struktur kelas dalam sistem, dengan kelas **User**, **Task**, dan **Session**, yang memiliki atribut dan metode yang terkait dengan pengelolaan pengguna, tugas, dan sesi login.



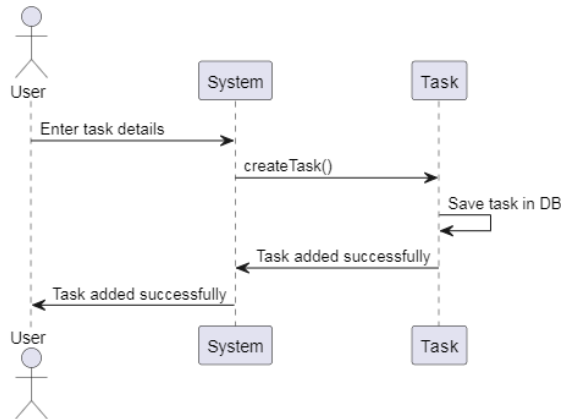
## 4.3 Activity Diagram

Diagram ini menggambarkan alur kegiatan dalam menambahkan tugas, mulai dari memasukkan detail tugas hingga konfirmasi bahwa tugas telah berhasil ditambahkan ke dalam database.



## 4.4 Sequence Diagram

Diagram ini menggambarkan urutan interaksi antara pengguna dan sistem untuk menambahkan tugas ke dalam aplikasi.



## 5. Tahap Pengembangan

Pada tahap pengembangan, sistem dibangun dengan menggunakan **HTML**, **CSS**, dan **JavaScript** untuk antarmuka pengguna, serta menggunakan **local storage** untuk menyimpan data tugas dan sesi login pengguna.

### 5.1 Penjelasan Code Program Utama Secara Ringkas

1. **HTML** digunakan untuk membangun struktur halaman web dan formulir input untuk memasukkan tugas baru.
2. **CSS** digunakan untuk mendesain tampilan antarmuka agar lebih user-friendly.
3. **JavaScript** menangani fungsionalitas dinamis seperti menambahkan tugas baru, menghapus tugas, memperbarui tugas, serta menangani sesi login menggunakan `localStorage`.

Contoh kode untuk menambahkan tugas:

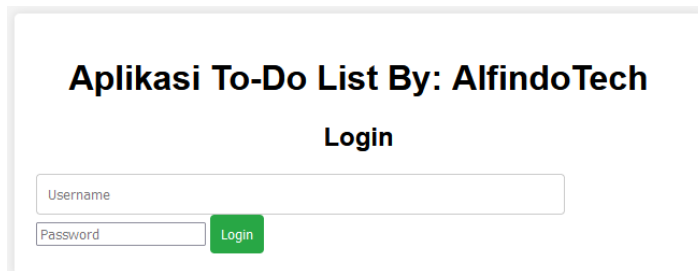
```
javascript
Salin kode
document.getElementById('addTaskBtn').addEventListener('click', function() {
  const taskTitle = document.getElementById('taskInput').value;
  if (taskTitle) {
    let tasks = JSON.parse(localStorage.getItem('tasks')) || [];
    tasks.push({ title: taskTitle, completed: false });
    localStorage.setItem('tasks', JSON.stringify(tasks));
    renderTasks();
  }
});
```

## 6. Hasil

### 6.1 Screenshot Fitur

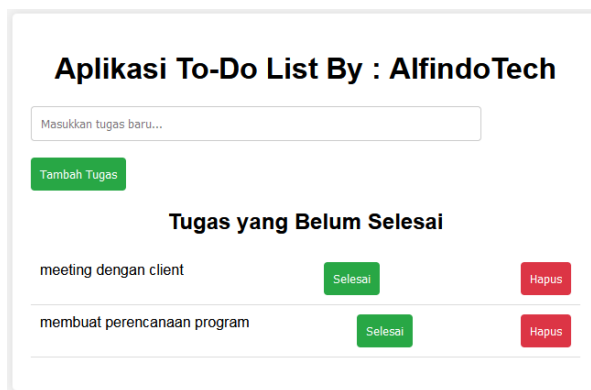
Berikut adalah screenshot beberapa fitur yang telah diimplementasikan:

1. **Login Page:** Menampilkan halaman login untuk pengguna masuk.



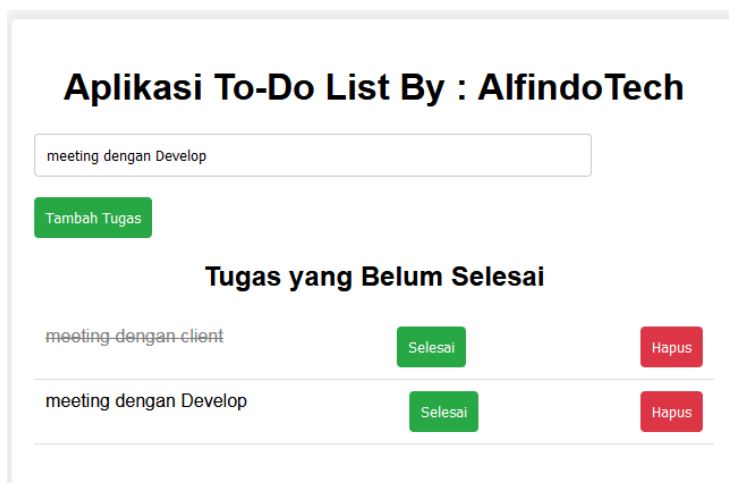
The screenshot shows a login form with the title "Aplikasi To-Do List By: AlfindoTech" and the subtitle "Login". It contains two input fields: "Username" and "Password". A green "Login" button is positioned to the right of the "Password" field.

2. **Dashboard:** Halaman utama yang menampilkan daftar tugas yang perlu diselesaikan.



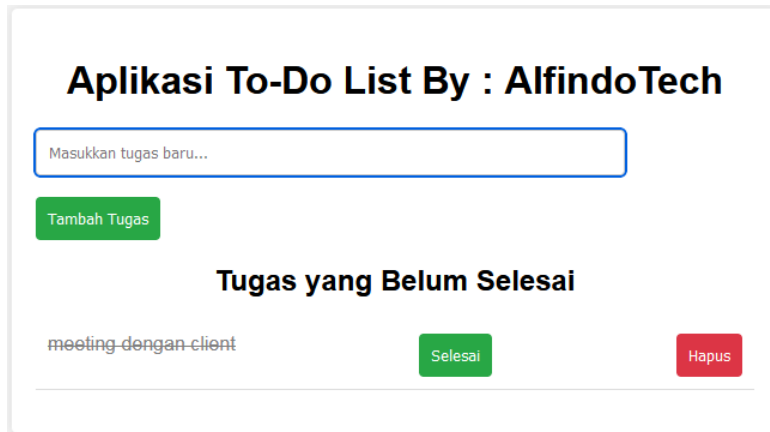
The screenshot shows the dashboard with the title "Aplikasi To-Do List By : AlfindoTech". It features a text input field labeled "Masukkan tugas baru...". Below it is a green "Tambah Tugas" button. The section "Tugas yang Belum Selesai" lists two tasks: "meeting dengan client" and "membuat perencanaan program". Each task has a green "Selesai" button and a red "Hapus" button.

3. **Add Task:** Formulir untuk menambahkan tugas baru.



The screenshot shows the "Add Task" form with the title "Aplikasi To-Do List By : AlfindoTech". It features a text input field containing "meeting dengan Develop". Below it is a green "Tambah Tugas" button. The section "Tugas yang Belum Selesai" lists two tasks: "meeting dengan client" and "meeting dengan Develop". Each task has a green "Selesai" button and a red "Hapus" button.

4. **CRUD Operations:** Menampilkan opsi untuk memperbarui, menghapus, dan menandai tugas sebagai selesai.



**Aplikasi To-Do List By : AlfindoTech**

Masukkan tugas baru...

Tambah Tugas

**Tugas yang Belum Selesai**

meeting-dengan-client

Selesai Hapus

## 6.2 Repository GitHub

Kode sumber aplikasi ini dapat diakses melalui Github Link ada di halaman jawaban

## 7. Analisis Hasil Pengembangan

Hasil pengembangan aplikasi ini menunjukkan bahwa fitur-fitur utama (CRUD dan manajemen login) dapat berjalan dengan baik. Namun, beberapa area yang dapat diperbaiki antara lain:

1. **Keamanan:** Sistem login masih menggunakan **localStorage** yang tidak aman untuk aplikasi dengan kebutuhan keamanan lebih tinggi. Perlu pengenalan **session management** dengan server.
2. **Antarmuka Pengguna:** Walaupun antarmuka sudah cukup intuitif, perbaikan lebih lanjut pada desain dan penambahan fitur seperti pengingat tugas bisa dilakukan.

## 8. Kesimpulan

Pengembangan aplikasi **To-Do List** ini berhasil menciptakan sistem yang memenuhi kebutuhan pengguna untuk mengelola tugas dengan mudah. Aplikasi ini menggunakan pendekatan **CRUD** dan memiliki sistem login dasar. Namun, masih ada beberapa area yang perlu perbaikan, terutama dalam hal keamanan dan fungsionalitas lanjutan.

## 9. Daftar Pustaka

1. PlantUML Documentation. (n.d.). <https://plantuml.com/>
2. W3Schools. (2023). HTML Tutorial. <https://www.w3schools.com/>
3. MDN Web Docs. (2023). JavaScript. <https://developer.mozilla.org/en-US/docs/Web/JavaScript>