

LAPORAN UAS
PEMROGRAMAN BERBASIS PLATFORM

Dosen Pendamping :
Ir. Achmad Firman Choiri, S.Kom., M.Kom.



Disusun Oleh:

YUDA PRATAMA (222140042)

5 IT A1
PRODI INFORMATIKA
INSTITUT TEKNOLOGI DAN BISNIS WIDYAGAMA LUMAJANG
2025

KATA PENGANTAR

Puji syukur senantiasa kami panjatkan kehadirat Allah SWT yang telah melimpahkan rahmat dan karunia-Nya, sehingga saya dapat menyelesaikan laporan ini guna memenuhi tugas UAS untuk mata kuliah “Pemrograman Berbasis Platform”.

Saya menyadari bahwa dalam penulisan laporan ini tidak terlepas dari bantuan banyak pihak yang dengan tulus memberikan doa, saran dan kritik sehingga laporan ini dapat terselesaikan.

Kami menyadari bahwa laporan ini masih jauh dari sempurna dikarenakan keterbatasan pengalaman dan pengetahuan yang saya miliki. Oleh karena itu, kami mengharapkan segala bentuk saran serta masukan dan kritik yang membangun dari berbagai pihak. Akhirnya saya berharap semoga laporan ini dapat memberikan manfaat bagi perkembangan dunia pendidikan dan teknologi.

Lumajang, 26 Januari 2025

DAFTAR ISI

KATA PENGANTAR.....	2
DAFTAR ISI.....	3
BAB I.....	4
PENDAHULUAN.....	4
1.1 Latar Belakang	4
1.2 Rumusan Masalah	4
1.3 Tujuan	5
BAB II	6
PEMBAHASAN	6
2.1 Kemudahan Penggunaan Aplikasi dengan Fitur CRUD.....	6
2.2 Penggunaan Firebase Realtime Database untuk Sinkronisasi Data	6
2.3 Desain Antarmuka Pengguna yang Responsif dan Intuitif	7
2.4. Validasi Input Pengguna	8
2.5. Pengelolaan Tugas yang Terorganisir	8
2.6 Integrasi Fitur Utama dalam Aplikasi	9
BAB III.....	10
DESAIN SISTEM	10
3.1 Flowchart Aplikasi	10
3.2 Use Case Diagram.....	11
BAB IV	13
PENUTUP.....	13
4.1 Kesimpulan	13
4.2 Implikasi dan Saran.....	13
REFERENSI.....	15

BAB I

PENDAHULUAN

1.1 Latar Belakang

Dalam kehidupan modern yang serba cepat, pengelolaan tugas menjadi salah satu hal yang sangat penting untuk meningkatkan produktivitas. Banyak individu dan organisasi mencari solusi teknologi untuk membantu mereka mengelola daftar tugas (to-do list) secara efisien. Aplikasi *Flutter Task Manager* dirancang untuk menjawab kebutuhan ini, memungkinkan pengguna mencatat, mengelola, dan memonitor tugas-tugas harian mereka dengan mudah.

Aplikasi ini memanfaatkan teknologi modern seperti Flutter untuk tampilan antarmuka pengguna (*User Interface*) yang responsif menggunakan SqLite untuk penyimpanan lokal dan Firebase Realtime Database sebagai basis data cloud untuk penyimpanan tugas. Dengan integrasi ini, aplikasi dapat memberikan pengalaman pengguna yang cepat, fleksibel, sederhana dan real-time.

1.2 Rumusan Masalah

1. Bagaimana menciptakan aplikasi manajemen tugas yang mudah digunakan dengan fitur penambahan, pengeditan, penghapusan, dan tampilan daftar tugas secara real-time?
2. Bagaimana memanfaatkan Firebase Realtime Database untuk menyimpan dan mengelola data tugas secara efisien sehingga dapat diakses dan diperbarui secara sinkron oleh pengguna?
3. Bagaimana mendesain antarmuka pengguna yang responsif dan intuitif agar aplikasi dapat memberikan pengalaman pengguna yang nyaman dan menarik di berbagai perangkat?

1.3 Tujuan

1. Membantu pengguna mencatat dan mengelola daftar tugas secara terorganisir.
2. Memberikan kemampuan untuk menambahkan, mengedit, dan menghapus tugas.
3. Menyediakan antarmuka pengguna yang mudah digunakan dan dapat diakses di berbagai perangkat.

BAB II

PEMBAHASAN

2.1 Kemudahan Penggunaan Aplikasi dengan Fitur CRUD

Aplikasi *Flutter Task Manager* dirancang untuk mendukung operasi CRUD (Create, Read, Update, Delete) pada daftar tugas. Fitur-fitur ini mencakup:

- **Create:** Pengguna dapat menambahkan tugas baru melalui formulir input yang sederhana.
- **Read:** Tugas yang telah dibuat ditampilkan dalam bentuk daftar, memudahkan pengguna untuk memonitor semua tugas mereka.
- **Update:** Pengguna dapat mengedit tugas yang sudah ada untuk memperbarui detail atau informasi tugas.
- **Delete:** Pengguna dapat menghapus tugas yang tidak lagi relevan dengan sekali klik.

Proses ini dilakukan secara real-time dengan bantuan Firebase, sehingga setiap perubahan langsung tercermin pada antarmuka aplikasi.

2.2 Penggunaan Firebase Realtime Database untuk Sinkronisasi Data

Firebase Realtime Database digunakan sebagai solusi penyimpanan data berbasis cloud yang memungkinkan sinkronisasi data secara langsung di berbagai perangkat. Keunggulan Firebase meliputi:

- **Real-time Updates:** Setiap perubahan yang dilakukan oleh pengguna pada data tugas (penambahan, pengeditan, penghapusan) akan langsung diperbarui pada aplikasi tanpa perlu memuat ulang.
- **Kemudahan Akses:** Dengan Firebase, pengguna dapat mengelola tugas mereka kapan saja dan di mana saja, selama terhubung ke internet.
- **Keamanan Data:** Firebase memiliki fitur autentikasi dan aturan database yang dapat digunakan untuk menjaga data pengguna tetap aman.

Implementasi Firebase dilakukan dengan menghubungkan aplikasi ke jalur data tasks pada database. Setiap tugas direpresentasikan sebagai objek JSON yang disimpan dengan identifikasi unik.

2.3 Desain Antarmuka Pengguna yang Responsif dan Intuitif

Untuk menciptakan pengalaman pengguna yang optimal, aplikasi ini menggunakan Flutter sebagai framework pengembangan. Flutter menyediakan komponen desain yang modern dan responsif, seperti:

- **Material Design:** Digunakan untuk memberikan tampilan antarmuka yang bersih dan konsisten.
- **ListView:** Menampilkan daftar tugas secara dinamis dengan kemampuan untuk menggulir.
- **ElevatedButton:** Memberikan akses mudah ke tombol aksi seperti menambahkan tugas.

Desain aplikasi juga mempertimbangkan:

- **Teks yang Mudah Dibaca:** Ukuran font yang disesuaikan untuk memastikan teks judul dan deskripsi tugas mudah dibaca.
- **Tombol yang Mudah Diklik:** Tombol dirancang dengan ukuran yang cukup besar untuk memudahkan interaksi.

2.4. Validasi Input Pengguna

Validasi input menjadi salah satu fokus utama dalam aplikasi ini untuk memastikan data yang masuk ke dalam database adalah data yang benar.

- **Pencegahan Input Kosong:** Pengguna tidak dapat menyimpan tugas jika kolom judul atau deskripsi kosong. Jika terjadi, aplikasi akan menampilkan pesan peringatan melalui *Snackbar*.
- **Validasi Data:** Data yang diterima dari pengguna akan diperiksa sebelum dikirim ke Firebase untuk memastikan keakuratannya.

Dengan validasi yang baik, aplikasi dapat mencegah kesalahan input yang dapat menyebabkan data menjadi tidak konsisten.

2.5. Pengelolaan Tugas yang Terorganisir

Aplikasi ini bertujuan untuk meningkatkan produktivitas pengguna dengan menyediakan fitur-fitur yang mempermudah pengelolaan tugas, seperti:

- **Kemudahan Menambah dan Mengedit Tugas:** Pengguna dapat mencatat tugas baru atau memperbarui tugas lama dengan langkah-langkah sederhana.
- **Kemampuan Menghapus Tugas:** Dengan satu klik, tugas yang sudah tidak relevan dapat dihapus untuk menjaga daftar tetap terorganisir.
- **Tampilan Tugas yang Terstruktur:** Tugas ditampilkan dalam bentuk daftar yang terstruktur, dilengkapi dengan judul, deskripsi, dan opsi untuk mengedit atau menghapus.

Fitur ini memungkinkan pengguna untuk fokus pada tugas yang benar-benar penting dan relevan.

2.6 Integrasi Fitur Utama dalam Aplikasi

Semua fitur utama yang dirancang untuk aplikasi ini mendukung satu tujuan: memberikan solusi yang lengkap untuk mengelola tugas. Proses pengelolaan dimulai dari menambahkan tugas, menyimpan data ke Firebase, menampilkan daftar tugas dari data di Firebase, hingga memperbarui atau menghapus tugas sesuai kebutuhan.

BAB III

DESAIN SISTEM

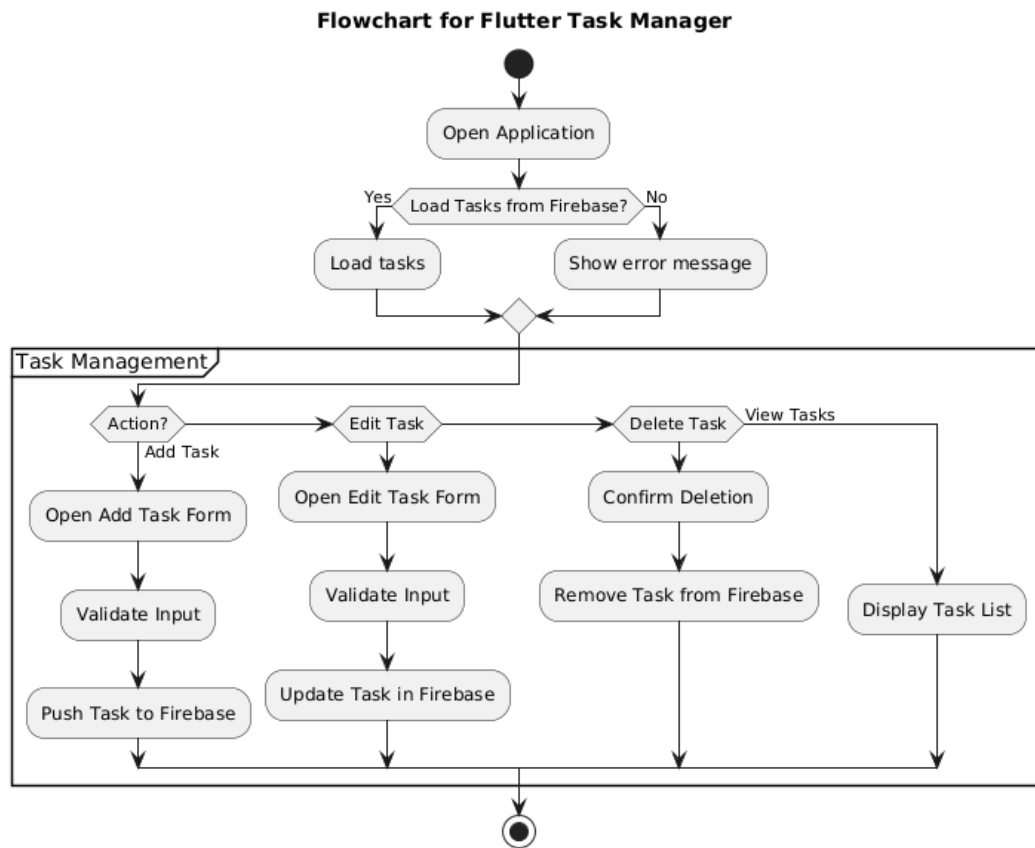
Pada bab ini, akan dijelaskan mengenai desain sistem aplikasi *Flutter Task Manager*. Desain sistem ini mencakup **flowchart** dan **use case diagram** untuk memberikan gambaran yang jelas mengenai alur kerja dan fungsi-fungsi yang terdapat dalam aplikasi.

3.1 Flowchart Aplikasi

Flowchart menggambarkan alur proses utama dalam aplikasi, mulai dari pengguna membuka aplikasi, melakukan operasi CRUD (Create, Read, Update, Delete), hingga sinkronisasi data dengan Firebase Realtime Database. Berikut adalah deskripsi alur prosesnya:

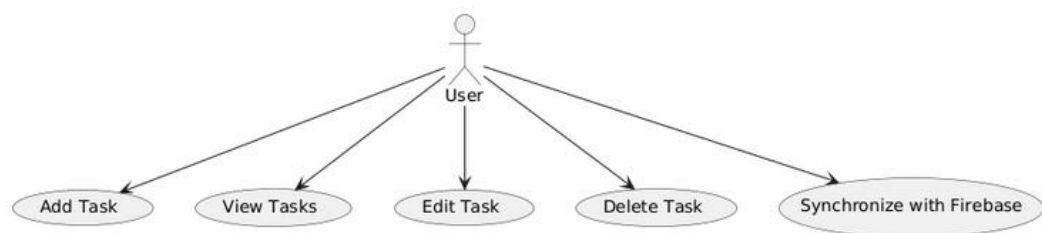
1. Sistem memuat daftar tugas dari Firebase Realtime Database.
2. Pengguna dapat memilih salah satu dari empat aksi utama:
 - Menambahkan tugas baru.
 - Melihat daftar tugas.
 - Mengedit tugas yang ada.
 - Menghapus tugas.
3. Sistem melakukan validasi input (jika menambah atau mengedit tugas).
4. Hasil dari aksi pengguna disinkronisasi dengan Firebase.
5. Tampilan aplikasi diperbarui berdasarkan data yang terbaru.

Flowchart:



Berikut adalah visualisasi flowchart aplikasi

3.2 Use Case Diagram



Use case diagram menggambarkan interaksi antara pengguna dengan fungsi-fungsi utama dalam aplikasi. Berikut adalah komponen utama dalam use case diagram:

1. **Aktor:**

- **Pengguna:** Satu-satunya aktor yang berinteraksi dengan aplikasi untuk mengelola tugas.

2. **Use Case:**

- **Menambah Tugas:** Pengguna dapat membuat tugas baru.
- **Melihat Daftar Tugas:** Pengguna dapat melihat daftar tugas yang telah disimpan.
- **Mengedit Tugas:** Pengguna dapat memperbarui detail tugas yang sudah ada.
- **Menghapus Tugas:** Pengguna dapat menghapus tugas yang tidak lagi relevan.
- **Sinkronisasi Data:** Aplikasi menyinkronkan data tugas dengan Firebase.

BAB IV

PENUTUP

4.1 Kesimpulan

Setelah melalui proses pengembangan dan implementasi, aplikasi Flutter Task Manager berhasil dirancang dengan fitur-fitur utama yang mendukung manajemen tugas secara efektif. Proyek ini menunjukkan bagaimana Flutter dan Firebase dapat digabungkan untuk menciptakan aplikasi yang fungsional dan responsif.

Aplikasi ini memungkinkan pengguna untuk menambahkan, melihat, mengedit, dan menghapus tugas dengan mudah. Proses sinkronisasi dengan Firebase memastikan data tersimpan dengan aman dan dapat diakses kapan saja.

Implementasi Firebase Realtime Database berhasil dilakukan untuk menyimpan data tugas secara real-time. Hal ini memberikan pengalaman pengguna yang lancar, terutama dalam pengelolaan data yang diperbarui secara langsung.

4.2 Implikasi dan Saran

- **Pengembangan Lebih Lanjut** Sebagai langkah pengembangan selanjutnya, aplikasi ini dapat ditingkatkan dengan fitur tambahan seperti kategori tugas, pengingat, atau sinkronisasi dengan kalender pengguna.
- **Penggunaan Teknologi Tambahan** Untuk mendukung analitik data, integrasi dengan layanan seperti Google Analytics atau Firebase Crashlytics dapat memberikan wawasan yang lebih mendalam terkait penggunaan aplikasi.
- **Keamanan Data** Meskipun Firebase menawarkan tingkat keamanan tinggi, penambahan fitur otentikasi pengguna dapat meningkatkan kepercayaan pengguna terhadap aplikasi.

Keseluruhan proyek ini telah berhasil mencapai tujuan utamanya, yaitu menyediakan alat sederhana namun efektif untuk manajemen tugas. Dengan potensi pengembangan lebih lanjut, aplikasi ini dapat menjadi solusi andal dalam membantu pengguna mengelola aktivitas harian mereka dengan lebih baik.

REFERENSI

<https://www.youtube.com/watch?v=x7PZXGc9vx0>