**MAKALAH TEKNIK PEMROGRAMAN TERSTRUKTUR 1**

**“APLIKASI LIFESCAPE”**

****

**Kelompok 5:**

1. AHMAD FADHEL HAFIZHUDDIN (30123067)
2. AMANDA SABILA (30123133)
3. RAFLI ALFIAN PRIYO UTOMO (31123388)
4. RIZKY NURLIANSYAH (31123396)

**MANAJEMEN INFORMATIKA**

**DIPLOMA III TEKNOLOGI INFORMASI**

**UNIVERSITAS GUNADARMA**

**2024/2025**

**ABSTRAK**

Aplikasi Lifescape merupakan solusi digital inovatif yang dirancang untuk mendukung pengelolaan kesehatan, gaya hidup, dan produktivitas secara terpadu. Dengan pendekatan holistik, aplikasi ini mengintegrasikan berbagai fitur untuk membantu pengguna mencapai keseimbangan optimal antara aspek fisik, mental, dan emosional dalam kehidupan sehari-hari. Beberapa fitur utama Lifescape meliputi pelacakan kebiasaan sehat seperti pola tidur, aktivitas fisik, dan nutrisi; panduan personalisasi untuk rutinitas olahraga; serta alat untuk manajemen stres seperti meditasi dan teknik mindfulness.

Selain itu, Lifescape menyediakan fungsi pengelolaan waktu dan tugas melalui kalender interaktif, pengingat, dan fitur pencatatan harian yang memudahkan pengguna dalam merencanakan aktivitas mereka. Dengan dukungan teknologi analitik berbasis data, aplikasi ini memberikan rekomendasi yang disesuaikan dengan kebutuhan dan preferensi pengguna, sehingga mendorong pengembangan kebiasaan yang berkelanjutan dan peningkatan kualitas hidup.

Sebagai tambahan, Lifescape menciptakan lingkungan sosial yang interaktif dengan adanya komunitas pengguna. Fitur ini memungkinkan pengguna berbagi pengalaman, bergabung dalam tantangan bersama, dan memberikan dukungan antaranggota untuk mencapai tujuan hidup masing-masing. Antarmuka yang intuitif dan desain yang responsif memastikan aksesibilitas yang optimal, baik melalui perangkat mobile maupun desktop.

Melalui kombinasi teknologi modern, desain yang user-friendly, dan pendekatan berbasis data, Lifescape bertujuan untuk menjadi mitra yang terpercaya dalam perjalanan setiap individu menuju kehidupan yang lebih sehat, produktif, dan bermakna. Aplikasi ini tidak hanya membantu pengguna memantau dan meningkatkan kesejahteraan pribadi, tetapi juga menginspirasi mereka untuk menciptakan perubahan positif dalam kehidupan sehari-hari.

i

**KATA PENGANTAR**

Puji syukur kehadirat Tuhan Yang Maha Esa atas rahmat dan karunia-Nya, sehingga makalah ini dapat diselesaikan dengan baik. Makalah ini disusun dalam rangka memenuhi tugas serta memperluas wawasan pembaca.

Dalam penyusunan makalah ini, penulis memperoleh banyak bantuan dan dukungan dari berbagai pihak. Oleh karena itu, penulis ingin menyampaikan rasa terima kasih yang sebesar-besarnya kepada semua pihak yang telah memberikan arahan, dukungan moril maupun materil.

Penulis menyadari bahwa makalah ini masih memiliki kekurangan dan jauh dari sempurna. Oleh karena itu, kritik dan saran yang membangun sangat diharapkan untuk memperbaiki karya ini di masa mendatang.

Akhir kata, semoga makalah ini dapat bermanfaat bagi pembaca.

Jakarta, 5 Januari 2025

Penulis

ii

**DAFTAR ISI**

**ABSTRAK ....................................................................................................** i

**KATA PENGANTAR ...................................................................................** ii

**DAFTAR ISI ..................................................................................................** iii

**BAB I PENDAHULUAN .............................................................................** 1

1.1 Latar Belakang **.........................................................................................** 1

1.2 Rumusan Masalah **....................................................................................** 2

1.3 Tujuan **.......................................................................................................** 3

1.4 Manfaat **....................................................................................................** 4

**BAB II TINJAUAN PUSTAKA ..................................................................** 5

2.1 Konsep Manajemen Tugas dan Catatan **...................................................** 5

2.2 Implementasi Teknologi **...........................................................................** 8

**BAB III METODOLOGI PENGEMBANGAN ......................................** 14

3.1 Metode Pengembangan Aplikasi **............................................................** 14

3.2 Desain Sistem **.........................................................................................** 18

3.3 Alur Kerja Aplikasi **................................................................................** 24

3.4 Tools dan Framework **.............................................................................** 25

**BAB IV PEMBAHASAN ...........................................................................** 27

4.1 Pengertian Aplikasi Lifescape **................................................................** 27

4.2 Fitur Aplikasi **..........................................................................................** 28

4.3 Hasil Pengujian **......................................................................................** 31

4.4 Debugging dan Solusi **............................................................................** 37

**BAB V PENUTUP ......................................................................................** 39

5.1 Kesimpulan **............................................................................................** 39

5.2 Saran **.......................................................................................................** 40

**DAFTAR PUSTAKA ..................................................................................** 41

**BAB I**

**PENDAHULUAN**

**1.1 Latar Belakang**

Di era modern, gaya hidup manusia semakin dinamis, namun sering kali tidak seimbang. Tingginya tuntutan pekerjaan, paparan teknologi yang terus-menerus, serta kurangnya waktu untuk merawat diri sering kali menyebabkan stres, penurunan kesehatan fisik, dan gangguan kesejahteraan mental. Berdasarkan penelitian, banyak individu menghadapi tantangan dalam menjaga pola hidup sehat, seperti kurangnya olahraga, pola makan yang tidak teratur, kurang tidur, dan kesulitan mengelola waktu. Masalah ini diperburuk oleh minimnya alat yang efektif untuk memonitor kesehatan dan produktivitas secara terpadu.

Di sisi lain, kemajuan teknologi menawarkan potensi untuk menghadirkan solusi yang dapat membantu individu mengelola keseimbangan hidup mereka. Namun, banyak aplikasi yang ada saat ini hanya berfokus pada satu aspek, seperti kebugaran, meditasi, atau pengelolaan waktu, tanpa memberikan pendekatan holistik yang mencakup berbagai kebutuhan pengguna. Hal ini menciptakan kesenjangan dalam penyediaan alat yang mampu mengintegrasikan berbagai aspek kesejahteraan dalam satu platform.

Masalah ini juga diperkuat oleh fakta bahwa masyarakat sering kali kurang memiliki motivasi atau informasi yang tepat untuk memulai dan mempertahankan kebiasaan sehat. Ketiadaan komunitas pendukung atau sistem yang mampu memberikan rekomendasi berbasis data menyebabkan banyak upaya individu untuk meningkatkan kualitas hidup menjadi tidak konsisten dan tidak berkelanjutan.

Berangkat dari latar belakang ini, aplikasi Lifescape hadir sebagai solusi untuk menjawab kebutuhan tersebut. Aplikasi ini dirancang untuk mengintegrasikan

1

berbagai fitur yang mendukung gaya hidup sehat, seperti pelacakan kebiasaan, panduan kebugaran, pengelolaan waktu, serta teknik relaksasi dan mindfulness. Dengan pendekatan yang personalisasi, berbasis data, dan didukung oleh komunitas pengguna, Lifescape bertujuan untuk membantu individu tidak hanya memulai, tetapi juga mempertahankan pola hidup sehat yang seimbang di tengah tantangan kehidupan modern.

**1.2 Rumusan Masalah**

1. Bagaimana Mengintegrasikan Berbagai Aspek Manajemen Hidup dalam Satu Aplikasi?
2. Bagaimana Meningkatkan Produktivitas Pengguna Melalui Manajemen Tugas yang Efektif?
3. Bagaimana Memungkinkan Pengguna untuk Menyimpan dan Mengatur Catatan Pribadi dengan Aman?
4. Bagaimana Menyediakan Keamanan dan Privasi Data yang Dapat Diandalkan?
5. Bagaimana Membantu Pengguna Mencapai Keseimbangan Antara Kehidupan Pribadi dan Profesional?
6. Bagaimana Menyediakan Pengingat dan Notifikasi yang Fleksibel dan Tepat Waktu?
7. Bagaimana Menjaga Relevansi dan Skalabilitas Aplikasi Seiring Perkembangan Kebutuhan Pengguna?

2

**1.3 Tujuan**

1. Dengan mengembangkan aplikasi yang menggabungkan manajemen tugas dan catatan dalam satu platform yang intuitif dan mudah digunakan, sehingga memudahkan pengguna dalam mengelola berbagai aspek kehidupan mereka tanpa perlu beralih antar aplikasi.
2. Dengan menyediakan fitur To-do List yang memungkinkan pengguna untuk menambahkan, mengedit, menghapus, dan memprioritaskan tugas mereka secara efektif, serta memberikan notifikasi untuk memastikan pengguna tetap berada di jalur dan tidak melewatkan tenggat waktu.
3. Dengan mengembangkan fitur Notes yang memungkinkan pengguna untuk menyimpan informasi penting, membuat catatan harian, atau menyimpan ide dengan keamanan yang terjamin dan akses yang mudah kapan saja.
4. Dengan mengintegrasikan fitur keamanan seperti enkripsi data dan autentikasi yang kuat untuk melindungi data pribadi pengguna, memastikan bahwa informasi mereka tetap aman dan privasi terjaga.
5. Dengan membantu pengguna mengatur dan memvisualisasikan waktu mereka untuk mencapai keseimbangan antara kehidupan pribadi dan profesional, serta memastikan bahwa mereka memiliki kontrol penuh atas berbagai aspek hidup mereka.
6. Dengan memberikan pengingat dan notifikasi yang dapat disesuaikan, membantu pengguna tetap terorganisir dan tidak melewatkan tugas atau acara penting, tanpa menjadi gangguan.
7. Dengan mengembangkan aplikasi yang dapat terus diperbarui dan dikembangkan sesuai dengan umpan balik pengguna dan perkembangan teknologi, memastikan aplikasi tetap relevan dan mampu memenuhi kebutuhan pengguna di masa depan.

3

**1.4 Manfaat**

1. Manajemen Hidup yang Terintegrasi: Lifescape bisa menjadi pusat manajemen hidup yang membantu pengguna mengatur berbagai aspek kehidupan mereka seperti tugas harian, catatan penting, dan pengaturan pribadi dalam satu platform.
2. Peningkatan Produktivitas: Dengan fitur seperti To-do List atau Task Management, pengguna dapat merencanakan, memprioritaskan, dan menyelesaikan tugas lebih efisien, meningkatkan produktivitas mereka dalam kehidupan sehari-hari.
3. Catatan Pribadi yang Aman dan Terorganisir: Fitur Notes memungkinkan pengguna untuk menyimpan dan mengatur catatan pribadi, ide, atau informasi penting secara aman dan mudah diakses kapan saja.
4. Keamanan dan Privasi Data: Lifescape kemungkinan memiliki fitur Login dan Logout yang memastikan keamanan data pribadi pengguna. Pengguna bisa merasa tenang mengetahui informasi mereka terlindungi dan hanya mereka yang memiliki akses.

4

**BAB II**

**TINJAUAN PUSTAKA**

**2.1 Konsep Manajemen Tugas dan Catatan**

Konsep Manajemen Tugas dan Catatan adalah dua aspek penting dalam pengelolaan waktu dan informasi yang bertujuan untuk meningkatkan produktivitas dan organisasi pribadi. Berikut adalah penjelasan mengenai konsep tersebut:

1. **Manajemen Tugas**

Manajemen Tugas adalah proses perencanaan, pengorganisasian, pelaksanaan, dan pemantauan berbagai kegiatan atau pekerjaan untuk mencapai tujuan tertentu dengan cara yang efisien dan terorganisir. Konsep ini bertujuan untuk memastikan bahwa setiap tugas dikelola dengan baik, sesuai dengan prioritas, tenggat waktu, dan sumber daya yang tersedia. Elemen utama dalam manajemen tugas meliputi:

1. **Identifikasi Tugas:** Menentukan pekerjaan yang perlu diselesaikan.
2. **Penentuan Prioritas:** Mengurutkan tugas berdasarkan tingkat kepentingan dan urgensi.
3. **Perencanaan:** Menyusun langkah-langkah kerja yang jelas dan terstruktur.
4. **Pembagian Tanggung Jawab:** Menentukan siapa yang bertanggung jawab atas pelaksanaan tugas.
5. **Pemantauan:** Memantau perkembangan tugas secara berkala.
6. **Evaluasi:** Mengevaluasi hasil kerja untuk memastikan kesesuaian dengan tujuan yang diharapkan.

Manajemen tugas yang baik dapat meningkatkan produktivitas, mengurangi risiko kesalahan, dan membantu pencapaian hasil yang lebih optimal dalam berbagai konteks, baik pribadi maupun profesional.

5

**2. Firebase**

Firebase adalah platform pengembangan aplikasi berbasis cloud yang dikembangkan oleh Google. Firebase menyediakan berbagai layanan yang membantu pengembang dalam membangun, mengelola, dan mengembangkan aplikasi web maupun mobile dengan lebih mudah dan efisien.

Beberapa fitur utama Firebase meliputi:

1. **Realtime Database:** Database NoSQL yang memungkinkan sinkronisasi data secara real-time di seluruh perangkat.
2. **Cloud Firestore:** Database NoSQL yang lebih canggih dengan dukungan untuk query yang lebih kompleks dan skalabilitas tinggi.
3. **Authentication:** Layanan otentikasi yang mendukung metode seperti email, password, Google Sign-In, dan lainnya.
4. **Cloud Storage:** Penyimpanan file berbasis cloud untuk menyimpan gambar, video, atau file lainnya.
5. **Cloud Functions:** Fungsionalitas serverless yang memungkinkan menjalankan kode backend sebagai respons terhadap peristiwa yang dipicu oleh Firebase.
6. **Hosting:** Layanan hosting untuk aplikasi web yang cepat dan aman dengan dukungan SSL gratis.
7. **Analytics:** Alat analitik untuk melacak perilaku pengguna dalam aplikasi.

Firebase sangat cocok digunakan untuk mengembangkan aplikasi dengan kebutuhan backend yang ringan dan cepat, serta mendukung pengembangan aplikasi yang skalabel dan terintegrasi dengan ekosistem Google.

**3. Glide**

Glide adalah sebuah pustaka (library) untuk pengelolaan gambar di aplikasi Android. Glide memudahkan pengambilan, penyimpanan, dan penampilan gambar

6

dari sumber seperti URL, memori, atau penyimpanan lokal. Glide mengoptimalkan proses pengambilan gambar untuk meningkatkan kinerja aplikasi, dengan mengelola caching dan pengambilan gambar secara efisien.

**4. Photo View**

PhotoView adalah sebuah pustaka (library) yang sering digunakan dalam aplikasi Android untuk menangani tampilan gambar. Pustaka ini memungkinkan pengguna untuk melakukan gesture seperti zoom-in dan zoom-out atau memutar gambar dalam aplikasi dengan lancar. PhotoView juga memberikan kontrol lebih terhadap cara gambar ditampilkan, membuatnya lebih interaktif dan responsif.

**5. MPAndroidChart**

MPAndroidChart adalah sebuah pustaka (library) untuk pengembangan aplikasi Android yang digunakan untuk membuat berbagai jenis grafik dan diagram secara dinamis dan interaktif. Pustaka ini memungkinkan pengembang untuk menampilkan data dalam bentuk visual yang mudah dipahami oleh pengguna, seperti grafik garis, grafik batang, grafik lingkaran, dan banyak lagi.

Berikut adalah beberapa fitur utama dari MPAndroidChart:

1. **Jenis Grafik yang Didukung:**

* Grafik Line Chart (grafik garis): Untuk menunjukkan tren atau perubahan data sepanjang waktu.
* Grafik Bar Chart (grafik batang): Untuk menampilkan data kategorikal atau komparasi antar nilai.
* Grafik Pie Chart (grafik lingkaran): Untuk menunjukkan proporsi atau persentase dari total.
* Grafik Radar Chart (grafik radar): Untuk menampilkan data yang memiliki lebih dari satu variabel.

7

* Grafik Scatter Chart (grafik sebar): Untuk menunjukkan hubungan antar dua variabel data.

1. **Interaktivitas:** MPAndroidChart mendukung interaksi pengguna dengan grafik, seperti zoom, scroll, dan highlight pada data titik tertentu. Pengguna dapat berinteraksi dengan grafik untuk melihat nilai lebih mendetail.
2. **Penyesuaian Tampilan:** Pengembang dapat menyesuaikan tampilan grafik dengan berbagai opsi seperti warna, label, sumbu, teks, dan grid. Hal ini memberikan fleksibilitas penuh untuk menciptakan grafik sesuai dengan desain aplikasi yang diinginkan.
3. **Performa dan Efisiensi:** Pustaka ini dirancang untuk kinerja yang optimal, mampu menangani data dalam jumlah besar dengan lancar tanpa mengurangi pengalaman pengguna.
4. **Integrasi Mudah:** MPAndroidChart mudah diintegrasikan ke dalam aplikasi Android menggunakan Gradle, dengan dokumentasi yang lengkap dan contoh penggunaan yang tersedia.

Secara keseluruhan, MPAndroidChart adalah alat yang sangat berguna bagi pengembang Android yang ingin menyajikan data dalam bentuk visual yang menarik dan mudah dimengerti oleh pengguna.

**2.2 Implementasi Teknologi**

Implementasi teknologi dalam pengembangan aplikasi modern sangat penting untuk memastikan kinerja, keamanan, dan fungsionalitas yang optimal. Berikut penjelasan tentang bagaimana RoomDB, Firebase Authentication, dan MPAndroidChart dapat diimplementasikan dalam pengembangan aplikasi:

8

* **RoomDB**

**RoomDB** adalah salah satu teknologi utama yang sering digunakan dalam pengembangan aplikasi Android untuk mengelola database lokal. RoomDB adalah bagian dari **Android Jetpack** dan berfungsi sebagai lapisan abstraksi di atas SQLite, sehingga mempermudah pengembang dalam bekerja dengan database.

Berikut adalah penjelasan singkat tentang teknologi ini:

**1. Apa itu RoomDB?**

RoomDB adalah Object Relational Mapping (ORM) library yang dirancang untuk menyederhanakan proses pengelolaan database SQLite dalam aplikasi Android. Dengan Room, pengembang dapat menggunakan anotasi untuk mendefinisikan entitas dan operasi database, menggantikan penggunaan kueri SQL yang kompleks.

**2. Keunggulan RoomDB**

* **Kemudahan Penggunaan:** Room menyediakan API yang sederhana untuk melakukan operasi seperti CRUD (Create, Read, Update, Delete).
* **Kompatibilitas dengan LiveData:** Mendukung pengamatan data secara reaktif, memungkinkan aplikasi untuk secara otomatis memperbarui UI saat data berubah.
* **Validasi Kueri Compile-Time:** Kueri SQL yang ditulis dalam Room divalidasi selama *compile time*, sehingga mengurangi risiko kesalahan runtime.
* **Integrasi dengan Kotlin Coroutines dan Flow:** Mendukung pemrograman asinkron, yang membantu mengelola operasi database secara efisien.
* **Pengelolaan Data Lokal yang Efisien:** Memungkinkan penyimpanan data secara offline, menjadikan aplikasi tetap fungsional tanpa koneksi internet.

9

**3. Komponen Utama RoomDB**

* **Entity:** Merepresentasikan tabel dalam database. Setiap atribut di kelas Entity menjadi kolom dalam tabel.
* **DAO (Data Access Object):** Berisi metode untuk mengakses data dalam database, seperti query, insert, update, dan delete.
* **Database:** Titik akses utama untuk berinteraksi dengan database Room. Biasanya, ini adalah kelas abstrak yang memperluas RoomDatabase.

RoomDB cocok untuk aplikasi seperti Lifescape karena dapat digunakan untuk menyimpan data lokal seperti log aktivitas, kebiasaan harian, atau riwayat kesehatan pengguna, yang dapat diakses kapan saja bahkan tanpa koneksi internet.

* **Firebase Authentication**

**Firebase Authentication (Firebase Auth)** adalah layanan dari Firebase yang dirancang untuk menangani otentikasi pengguna dengan mudah dan aman. Teknologi ini mendukung berbagai metode autentikasi, seperti email dan kata sandi, nomor telepon, serta login menggunakan platform pihak ketiga seperti Google, Facebook, Twitter, dan lainnya.

Berikut adalah penjelasan singkat tentang Firebase Authentication:

**1. Apa itu Firebase Auth?**

Firebase Auth adalah layanan otentikasi berbasis cloud yang membantu pengembang mengelola proses login dan registrasi pengguna tanpa harus membangun sistem autentikasi dari awal. Firebase Auth juga memastikan keamanan data pengguna dengan menerapkan praktik keamanan modern.

**2. Fitur Utama Firebase Auth**

10

* **Otentikasi Email dan Kata Sandi:** Memungkinkan pengguna membuat akun atau masuk menggunakan email dan kata sandi.
* **Login Sosial Media:** Mendukung integrasi login dengan Google, Facebook, Twitter, GitHub, dan lainnya.
* **Otentikasi Nomor Telepon:** Menggunakan verifikasi berbasis SMS untuk otentikasi.
* **Single Sign-On (SSO):** Mendukung integrasi dengan provider seperti Microsoft dan Apple.
* **Manajemen Pengguna:** Firebase Auth menyediakan antarmuka untuk menambah, menghapus, dan memodifikasi data pengguna.
* **Integrasi yang Mudah:** Firebase Auth dapat diintegrasikan dengan layanan Firebase lainnya, seperti Firestore atau Realtime Database.

**3. Keunggulan Firebase Auth**

* **Keamanan Tinggi:** Firebase Auth menggunakan protokol keamanan modern, seperti OAuth 2.0 dan OpenID Connect, untuk melindungi data pengguna.
* **Mudah Digunakan:** API sederhana dan dokumentasi yang lengkap memudahkan pengembang mengimplementasikan otentikasi.
* **Dukungan Lintas Platform:** Firebase Auth mendukung aplikasi berbasis Android, iOS, web, dan lainnya.
* **Pemulihan Kata Sandi:** Otomatis menyediakan fitur untuk mereset kata sandi pengguna melalui email.
* **Manajemen Token Otomatis:** Firebase Auth menangani pembaruan token secara otomatis untuk menjaga pengguna tetap terotentikasi.
* **MPAndroidChart**

11

**MPAndroidChart** adalah sebuah pustaka (library) open-source yang digunakan untuk membuat berbagai jenis grafik dan diagram di aplikasi Android. Library ini sangat populer karena fleksibilitasnya, kemudahan integrasi, dan berbagai opsi kustomisasi yang ditawarkan.

**Fitur Utama MPAndroidChart**

1. **Jenis Grafik yang Didukung:**
   * Grafik Garis (Line Chart).
   * Grafik Batang (Bar Chart).
   * Grafik Lingkaran (Pie Chart).
   * Grafik Sumbu Ganda (Combined Chart).
   * Grafik Bubble.
   * Grafik Radar (Spider Chart).
2. **Interaksi Pengguna:**
   * Mendukung zoom, pinch-to-zoom, dan geser.
   * Menyediakan label dan nilai data saat pengguna menyentuh grafik (highlight).
3. **Kustomisasi:**
   * Mendukung kustomisasi warna, label, gaya sumbu, dan elemen grafik lainnya.
   * Mudah diintegrasikan dengan tema atau desain aplikasi.
4. **Animasi:**
   * Menyediakan animasi saat data grafik ditampilkan atau diperbarui.
5. **Kompatibilitas Data:**
   * Dapat menerima data dari berbagai sumber, seperti ArrayList, API, atau database.

**Keunggulan MPAndroidChart**

12

* **Kemudahan Penggunaan:** API yang sederhana membuat grafik dapat dibuat dengan mudah.
* **Kinerja Tinggi:** Dirancang untuk menangani dataset besar tanpa mengorbankan kinerja.
* **Dokumentasi Lengkap:** Dilengkapi dokumentasi dan komunitas yang aktif.
* **Open Source:** Gratis digunakan dan dapat dimodifikasi sesuai kebutuhan.

13

**BAB III**

**METODOLOGI PENGEMBANGAN**

**3.1 Metode Pengembangan Aplikasi**

Berikut adalah metode pengembangan menggunakan waterfall dan agile:

* **Waterfall**

Waterfall adalah salah satu model pengembangan perangkat lunak yang paling tradisional dan terstruktur. Model ini mengikuti pendekatan linier dan berurutan, di mana setiap fase harus diselesaikan sebelum fase berikutnya dimulai. Waterfall sering digunakan dalam proyek dengan persyaratan yang sangat jelas dan tidak mungkin berubah selama siklus pengembangan.

**Tahapan Waterfall:**

1. **Requirement Analysis (Analisis Kebutuhan)**

Semua persyaratan untuk aplikasi dikumpulkan dan didokumentasikan secara rinci. Ini mencakup kebutuhan fungsional dan non-fungsional.

Dokumen spesifikasi kebutuhan (SRS - Software Requirements Specification) dibuat dan disetujui oleh semua pemangku kepentingan.

1. **System Design (Desain Sistem)**

Berdasarkan dokumen spesifikasi kebutuhan, desain sistem dibuat. Desain ini mencakup arsitektur sistem, desain database, dan antarmuka pengguna.

Desain biasanya dipisahkan menjadi dua bagian: High-Level Design (HLD) dan Low-Level Design (LLD).

1. **Implementation (Implementasi)**

Pengembang mulai menulis kode berdasarkan desain yang telah dibuat. Setiap modul dikembangkan dan diuji secara terpisah.

Pada tahap ini, dokumentasi kode juga dibuat untuk membantu dalam pemeliharaan di masa depan.

14

1. **Integration and Testing (Integrasi dan Pengujian)**

Setelah semua modul dikembangkan, mereka diintegrasikan menjadi satu sistem.

Sistem diuji secara menyeluruh untuk memastikan bahwa semua komponen bekerja dengan baik bersama-sama dan memenuhi persyaratan yang telah ditentukan.

Jenis pengujian termasuk unit testing, integration testing, system testing, dan acceptance testing.

1. **Deployment (Penerapan)**

Aplikasi yang telah diuji diterapkan ke lingkungan produksi. Ini adalah fase di mana aplikasi dapat digunakan oleh pengguna akhir.

Dokumentasi untuk pengguna dan pelatihan (jika diperlukan) disediakan.

1. **Maintenance (Pemeliharaan)**

Setelah aplikasi diterapkan, tim pengembangan memberikan dukungan untuk pemeliharaan, perbaikan bug, dan peningkatan kecil sesuai dengan kebutuhan pengguna.

Pemeliharaan berkelanjutan dilakukan untuk memastikan aplikasi tetap relevan dan berfungsi dengan baik.

**Kelebihan Model Waterfall:**

1. **Struktur yang Jelas:** Setiap fase memiliki tujuan yang jelas dan hasil yang dapat diukur.
2. **Dokumentasi yang Lengkap:** Dokumentasi mendalam membantu dalam pemeliharaan dan pemindahan proyek ke tim baru.
3. **Mudah Dimengerti dan Diatur:** Proses yang linier dan terdefinisi dengan baik membuat model ini mudah diikuti, terutama untuk tim dengan pengalaman yang beragam.

15

**Kekurangan Model Waterfall:**

1. **Kurangnya Fleksibilitas:** Sulit untuk mengubah persyaratan setelah proyek dimulai karena setiap fase tergantung pada fase sebelumnya.
2. **Keterlambatan Umpan Balik:** Pengguna akhir hanya dapat melihat hasilnya setelah semua fase selesai, yang berarti jika ada masalah, akan diketahui pada tahap yang sangat akhir.
3. **Tidak Cocok untuk Proyek Kompleks atau Dinamis:** Proyek dengan persyaratan yang terus berubah atau kompleksitas tinggi lebih cocok menggunakan model yang lebih fleksibel seperti Agile.

**Kesimpulan**

Model Waterfall adalah pendekatan yang terstruktur dan linier dalam pengembangan perangkat lunak. Meskipun memiliki kelebihan dalam hal dokumentasi dan pengelolaan proyek yang jelas, kekurangannya dalam hal fleksibilitas membuatnya kurang ideal untuk proyek yang membutuhkan perubahan cepat atau iterasi berkelanjutan. Model ini paling cocok digunakan dalam proyek di mana semua persyaratan dapat diprediksi dan tidak berubah selama siklus pengembangan.

* **Agile**

Agile adalah model pengembangan perangkat lunak yang berfokus pada fleksibilitas, kolaborasi, dan iterasi yang berkelanjutan. Agile menekankan pada pengiriman perangkat lunak dalam siklus pendek (iterasi) yang memungkinkan tim untuk menilai dan beradaptasi terhadap perubahan dengan cepat.

**Tahapan Pengembangan dengan Agile:**

**1. Conceptualization (Konseptualisasi):**

1. Tim memulai dengan visi produk dan backlog awal yang mencakup fitur atau user stories yang diinginkan.
2. Backlog ini adalah daftar prioritas kerja yang harus diselesaikan.

**2. Iteration Planning (Perencanaan Iterasi):**

16

1. Setiap iterasi dimulai dengan perencanaan di mana tim menentukan backlog item mana yang akan diselesaikan dalam iterasi tersebut.
2. Estimasi kerja dan sumber daya yang dibutuhkan juga dilakukan.

**3. Design and Development (Desain dan Pengembangan):**

1. Pengembangan dilakukan dalam siklus pendek dan iteratif.
2. Desain dan pengembangan dapat berubah berdasarkan umpan balik yang diterima di akhir iterasi sebelumnya.

**4. Testing and Quality Assurance (Pengujian dan Penjaminan Kualitas):**

1. Pengujian dilakukan selama iterasi dan setelah setiap iterasi selesai untuk memastikan kualitas.
2. Continuous integration dan pengujian otomatis sering digunakan untuk mendukung pengujian berkelanjutan.

**5. Review and Feedback (Peninjauan dan Umpan Balik):**

1. Di akhir setiap iterasi, hasil kerja ditinjau oleh tim dan pemangku kepentingan.
2. Umpan balik digunakan untuk memperbaiki dan menyesuaikan backlog untuk iterasi berikutnya.

**6. Release (Rilis):**

* Aplikasi atau fitur dirilis setelah beberapa iterasi, memberikan nilai kepada pengguna secara berkelanjutan.

**7. Maintenance (Pemeliharaan):**

* Pemeliharaan berkelanjutan dilakukan setelah rilis, mencakup perbaikan bug dan peningkatan fitur.

**Keuntungan Agile:**

1. Fleksibilitas Tinggi: Dapat beradaptasi dengan perubahan persyaratan kapan saja.

17

1. Umpan Balik Cepat: Membantu mengidentifikasi dan mengatasi masalah lebih awal.
2. Pengiriman Lebih Cepat: Nilai dapat diberikan kepada pengguna lebih awal melalui iterasi kecil yang sering.
3. Keterlibatan Pengguna Aktif: Membantu memastikan bahwa produk akhir sesuai dengan kebutuhan pengguna.
4. Peningkatan Berkelanjutan: Setiap iterasi memungkinkan untuk perbaikan berkelanjutan pada produk.

**Kekurangan Agile:**

1. Tidak Cocok untuk Semua Proyek: Proyek dengan persyaratan yang sangat jelas dan tetap mungkin lebih baik menggunakan model seperti Waterfall.
2. Kebutuhan Manajemen yang Kuat: Agile membutuhkan manajemen yang lebih aktif untuk memantau kemajuan dan menyesuaikan prioritas.
3. Kurangnya Dokumentasi Formal: Fokus pada pengiriman cepat dapat menyebabkan kurangnya dokumentasi formal.

**Kesimpulan**

Agile adalah model pengembangan perangkat lunak yang berfokus pada fleksibilitas, kolaborasi, dan iterasi cepat. Dengan Agile, tim dapat merespon perubahan lebih cepat, mengurangi risiko, dan meningkatkan nilai produk melalui pengiriman dan umpan balik berkelanjutan. Agile sangat cocok untuk proyek-proyek yang dinamis dan memerlukan adaptasi berkelanjutan terhadap kebutuhan pengguna atau pasar.

**3.2 Desain Sistem**

app/

├── manifests/

│ └── AndroidManifest.xml

├── java/

│ └── app.lifescape/

│ ├── activity/

18

│ │ ├── AuthActivity.java

│ │ ├── BaseActivity.java

│ │ ├── MainActivity.java

│ │ ├── SplashActivity.java

│ │ └── WebViewActivity.java

│ ├── adapter/

│ │ ├── NotesAdapter.java

│ │ ├── TodoAdapter.java

│ │ └── ViewPagerAdapter.java

│ ├── database/

│ │ ├── dao/

│ │ │ ├── NotesDao.java

│ │ │ ├── TodoDao.java

│ │ │ └── UserDao.java

│ ├── entity/

│ │ ├── Notes.java

│ │ ├── Todo.java

│ │ ├── User.java

│ │ └── AppDatabase.java

│ ├── dialog/

│ │ ├── AddNoteDialog.java

│ │ ├── AddTodoDialog.java

│ │ ├── ChangePasswordDialog.java

│ │ ├── ForgotPasswordDialog.java

│ │ ├── ConfirmationDialog.java

│ │ └── NoteDetailDialog.java

│ ├── fragment/

│ │ ├── HomeFragment.java

│ │ ├── LoginFragment.java

│ │ ├── TodoFragment.java

19

│ │ ├── NotesFragment.java

│ │ ├── NotesFragment.java

│ │ ├── RegisterFragment.java

│ │ └── SettingsFragment.java

│ └── utils/

│ ├── AppUtils.java

│ ├── DiffCallback.java

│ ├── FirebaseAuthManager.java

│ ├── NativeUtils.java

│ └── NotificationReceiver.java

├── cpp/

│ ├── search/

│ │ ├── binary\_search.c

│ │ ├── binary\_search.h

│ ├── sorting/

│ │ ├── quick\_sort.c

│ │ └── quick\_sort.h

│ └── CmakeLists.txt

│

├──assets/

│ ├── about.html

│ ├── privacy.html

│ └── terms.html

**Penjelasan:**

**1. app/**

Folder utama aplikasi. Semua komponen aplikasi Android dan file sumber lainnya berada di sini.

**manifests/**

20

* **AndroidManifest.xml**: File manifest yang mendeklarasikan komponen aplikasi Android (Activity, Service, Receiver, dll.), izin yang dibutuhkan, dan metadata lainnya.

**java/**

Berisi kode sumber aplikasi dalam bahasa Java. Folder ini terstruktur sesuai dengan paket aplikasi app.lifescape.

* **activity/**: Berisi kelas-kelas Activity, yaitu antarmuka utama aplikasi.
  + **AuthActivity.java**: Mengelola proses autentikasi pengguna.
  + **BaseActivity.java**: Kelas dasar untuk Activity lain, menyediakan fungsionalitas umum.
  + **MainActivity.java**: Activity utama yang berisi navigasi atau pengelolaan fitur inti.
  + **SplashActivity.java**: Menampilkan layar pembuka sebelum aplikasi dimuat.
  + **WebViewActivity.java**: Mengelola tampilan halaman web dalam aplikasi.
* **adapter/**: Berisi adapter yang menghubungkan data dengan komponen tampilan seperti RecyclerView.
  + **NotesAdapter.java**: Adapter untuk menampilkan daftar catatan.
  + **TodoAdapter.java**: Adapter untuk daftar to-do.
  + **ViewPagerAdapter.java**: Adapter untuk ViewPager, digunakan untuk navigasi tab atau halaman.
* **database/**: Kode yang berkaitan dengan pengelolaan database lokal.
  + **dao/**: Berisi antarmuka untuk mengakses data di database.
    - **NotesDao.java**: Operasi database untuk catatan.
    - **TodoDao.java**: Operasi database untuk to-do.
    - **UserDao.java**: Operasi database untuk pengguna.
  + **entity/**: Berisi kelas model data.
    - **Notes.java**: Entitas catatan.

21

* + - **Todo.java**: Entitas to-do.
    - **User.java**: Entitas pengguna.
    - **AppDatabase.java**: Konfigurasi utama database menggunakan Room.
* **dialog/**: Berisi kelas dialog untuk berbagai interaksi pengguna.
  + **AddNoteDialog.java**: Dialog untuk menambahkan catatan.
  + **AddTodoDialog.java**: Dialog untuk menambahkan to-do.
  + **ChangePasswordDialog.java**: Dialog untuk mengubah kata sandi.
  + **ForgotPasswordDialog.java**: Dialog untuk mereset kata sandi.
  + **ConfirmationDialog.java**: Dialog untuk konfirmasi tindakan.
  + **NoteDetailDialog.java**: Menampilkan detail catatan.
* **fragment/**: Berisi kelas Fragment yang digunakan untuk membangun UI modular.
  + **HomeFragment.java**: Halaman beranda.
  + **LoginFragment.java**: Halaman login pengguna.
  + **TodoFragment.java**: Halaman untuk daftar to-do.
  + **NotesFragment.java**: Halaman untuk daftar catatan.
  + **RegisterFragment.java**: Halaman registrasi pengguna.
  + **SettingsFragment.java**: Halaman pengaturan aplikasi.
* **utils/**: Berisi utilitas atau helper untuk mendukung aplikasi.
  + **AppUtils.java**: Fungsi utilitas umum.
  + **DiffCallback.java**: Callback untuk membandingkan item di RecyclerView.
  + **FirebaseAuthManager.java**: Mengelola autentikasi menggunakan Firebase.
  + **NativeUtils.java**: Fungsi utilitas berbasis kode native (C/C++).
  + **NotificationReceiver.java**: Receiver untuk notifikasi.

**2. cpp/**

Folder kode sumber untuk fungsi yang ditulis dalam C/C++, kemungkinan

22

diintegrasikan dengan aplikasi Android menggunakan JNI (Java Native Interface).

* **search/**: Kode terkait pencarian.
  + **binary\_search.c**: Implementasi algoritma pencarian biner.
  + **binary\_search.h**: Header untuk fungsi pencarian biner.
* **sorting/**: Kode terkait pengurutan.
  + **quick\_sort.c**: Implementasi algoritma Quick Sort.
  + **quick\_sort.h**: Header untuk fungsi Quick Sort.
* **CMakeLists.txt**: Konfigurasi untuk membangun kode C/C++ menggunakan CMake.

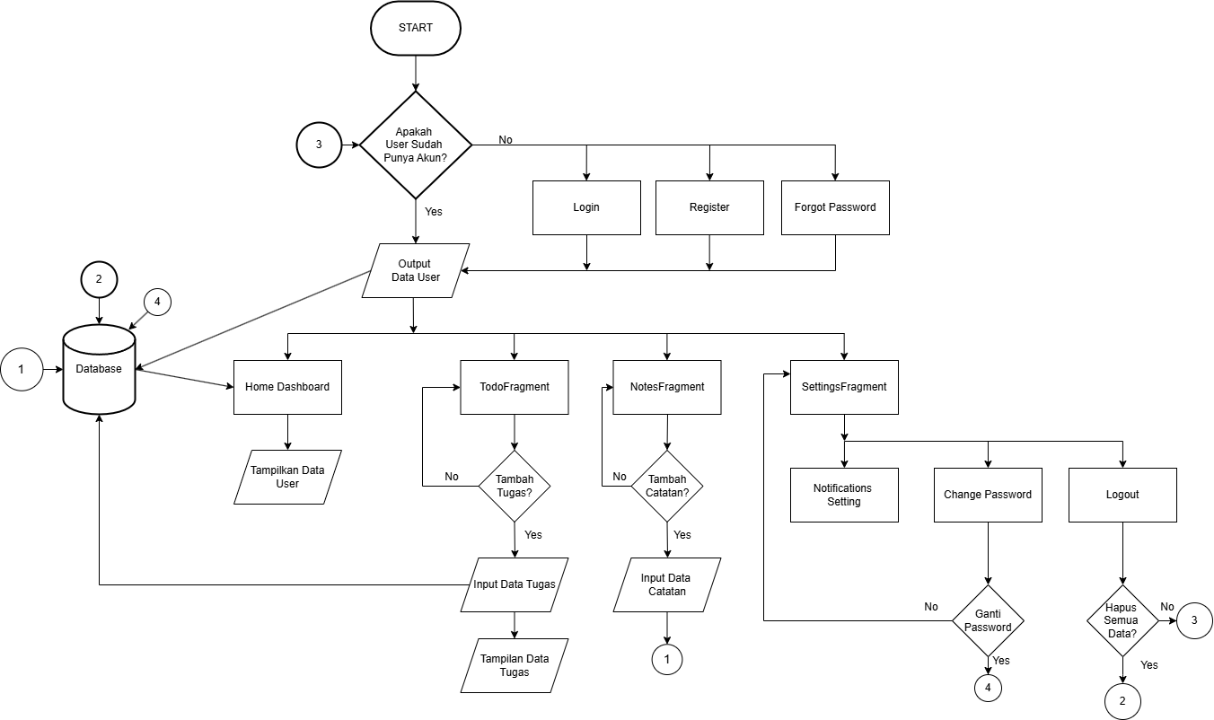
**3. assets/**

Berisi file statis seperti HTML untuk ditampilkan dalam aplikasi.

* **about.html**: Halaman tentang aplikasi.
* **privacy.html**: Halaman kebijakan privasi.
* **terms.html**: Halaman syarat dan ketentuan.

23

**3.3 Alur Kerja Aplikasi**

****

**Penjelasan:**

* **START**: Titik awal dari proses ini.
* **Apakah User Sudah Punya Akun?**: Ini adalah titik keputusan pertama di mana sistem memeriksa apakah pengguna sudah memiliki akun atau tidak.
* **Yes**: Jika pengguna sudah memiliki akun, sistem akan menampilkan **Output Data User**.
* **No**: Jika pengguna belum memiliki akun, pengguna dapat melakukan proses **Login**, **Register**, atau memilih opsi **Forgot Password** jika mereka lupa kata sandinya.
* **Output Data User**: Setelah login berhasil, sistem akan menampilkan data pengguna.
* **Home Dashboard**: Beranda utama di mana pengguna dapat melihat data mereka.

24

* **Tampilkan Data User**: Sistem akan menampilkan data pengguna yang relevan di dashboard.
* **TodoFragment**: Bagian ini memungkinkan pengguna untuk mengelola tugas mereka.
* **Tambah Tugas?**: Pengguna dapat menambahkan tugas baru.
  + **No**: Jika pengguna tidak menambahkan tugas baru, sistem akan kembali ke **TodoFragment**.
  + **Yes**: Jika pengguna menambahkan tugas baru, mereka akan memasukkan data tugas dan kemudian sistem akan **Tampilkan Data Tugas**.
* **NotesFragment**: Bagian ini memungkinkan pengguna untuk mengelola catatan mereka.
* **Tambah Catatan?**: Pengguna dapat menambahkan catatan baru.
  + **No**: Jika pengguna tidak menambahkan catatan baru, sistem akan kembali ke **NotesFragment**.
  + **Yes**: Jika pengguna menambahkan catatan baru, mereka akan memasukkan data catatan.
* **SettingsFragment**: Bagian pengaturan di mana pengguna dapat mengubah berbagai pengaturan.
* **Notifications Setting**: Pengguna dapat mengatur notifikasi.
* **Change Password**: Pengguna dapat mengubah kata sandi mereka.
  + **Ganti Password?**: Jika pengguna memilih untuk mengubah kata sandi.
    - **No**: Sistem akan kembali ke **SettingsFragment**.
    - **Yes**: Proses perubahan kata sandi berlangsung dan kemudian kembali ke langkah 4.
* **Logout**: Pengguna dapat keluar dari akun mereka.
  + **Hapus Semua Data?**: Sistem menanyakan apakah pengguna ingin menghapus semua data.

25

* + - **No**: Sistem akan kembali ke **SettingsFragment**.
    - **Yes**: Proses penghapusan data berlangsung dan kemudian kembali ke langkah 2.
* **Database**: Semua data yang dihasilkan atau diperlukan dalam proses ini disimpan di database.

**3.4 Tools dan Framework**

Pada proses pengembangan aplikasi Lifescape, berbagai tools, framework, dan library digunakan untuk mendukung pengembangan fitur-fitur yang optimal. Berikut adalah rincian dari tools dan framework yang digunakan:

**1. Bahasa Pemrograman**

* Java: Digunakan sebagai bahasa pemrograman utama untuk pengembangan aplikasi Android.
* C (Native Code/JNI): Digunakan untuk mengimplementasikan algoritma Sorting dan Searching berbasis Algoritma C yang menjadi syarat tugas.

**2. IDE (Integrated Development Environment)**

* Android Studio: IDE utama untuk pengembangan aplikasi Android, menyediakan emulator, debugging, dan pengelolaan dependensi.
* SQLite Database Browser: Digunakan untuk memeriksa struktur dan isi database Room secara manual.

**3. Database**

* Room Database: Library dari Android Jetpack untuk pengelolaan database SQLite dengan abstraksi yang lebih aman dan efisien.
* Firebase Authentication: Digunakan untuk sistem autentikasi pengguna, termasuk login, register, dan logout.

**4. Library Tambahan**

* MPAndroidChart: Library yang digunakan untuk menampilkan data dalam bentuk grafik, seperti grafik perkembangan tugas (Todo Progress).

26

* Glide: Digunakan untuk memuat dan menampilkan gambar dengan efisien (jika fitur unggah gambar di Notes diaktifkan).
* Material Design Components: Library UI/UX untuk menciptakan tampilan aplikasi yang modern dan responsif, seperti Bottom Navigation, Snackbar, dan Dialog.

**5. Dependency Management**

* Gradle: Digunakan sebagai sistem build otomatis untuk mengelola dependensi dan konfigurasi aplikasi.

27

**BAB IV**

**PEMBAHASAN**

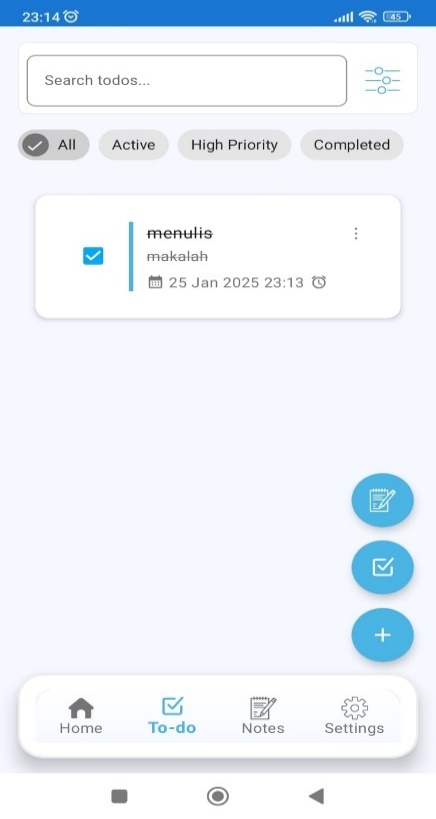
**4.1 Pengertian Aplikasi Lifescape**

Lifescape adalah istilah yang dapat digunakan untuk merujuk pada aplikasi yang dirancang untuk membantu pengguna dalam mengelola berbagai aspek kehidupan mereka dalam satu platform. Aplikasi ini biasanya mengintegrasikan beberapa fitur penting, seperti manajemen tugas dan pencatatan yang semuanya bertujuan untuk meningkatkan produktivitas dan keseimbangan hidup pengguna. Secara lebih luas, lifescape juga dapat diartikan sebagai gambaran atau peta hidup seseorang yang mencakup berbagai elemen penting seperti pekerjaan, kesehatan, hubungan, dan waktu luang. Aplikasi yang diberi nama Lifescape bertujuan untuk menjadi alat bantu dalam mengatur dan mengelola elemen-elemen tersebut secara efektif, sehingga pengguna dapat mencapai tujuan hidup mereka dengan lebih terorganisir dan efisien.

28

**4.2 Fitur Aplikasi**

**1. Todo List**

****

Fitur ini memungkinkan pengguna untuk mengatur daftar tugas mereka secara efisien.

**Fitur yang disediakan:**

* **CRUD (Create, Read, Update, Delete)**:
  + **Create**: Pengguna dapat menambahkan tugas baru dengan informasi seperti judul, deskripsi, dan tenggat waktu.
  + **Read**: Menampilkan semua tugas yang telah dibuat.
  + **Update**: Mengedit detail tugas, termasuk status penyelesaian, prioritas, atau tanggal jatuh tempo.
  + **Delete**: Menghapus tugas yang tidak diperlukan lagi.
* **Filter**:
  + Menyaring tugas berdasarkan kategori, status (selesai atau belum), tenggat waktu, atau prioritas.

29

* **Search**:
  + Mencari tugas tertentu berdasarkan kata kunci, seperti judul atau deskripsi.
* **Sort**:
  + Mengurutkan tugas berdasarkan prioritas, tenggat waktu, atau status penyelesaian.
* **Prioritas**:
  + Pengguna dapat menetapkan tingkat prioritas (rendah, sedang, tinggi) untuk setiap tugas, sehingga lebih mudah untuk memfokuskan pada yang terpenting.

**2. Notes**

****

Fitur ini membantu pengguna mencatat informasi penting dalam format fleksibel.

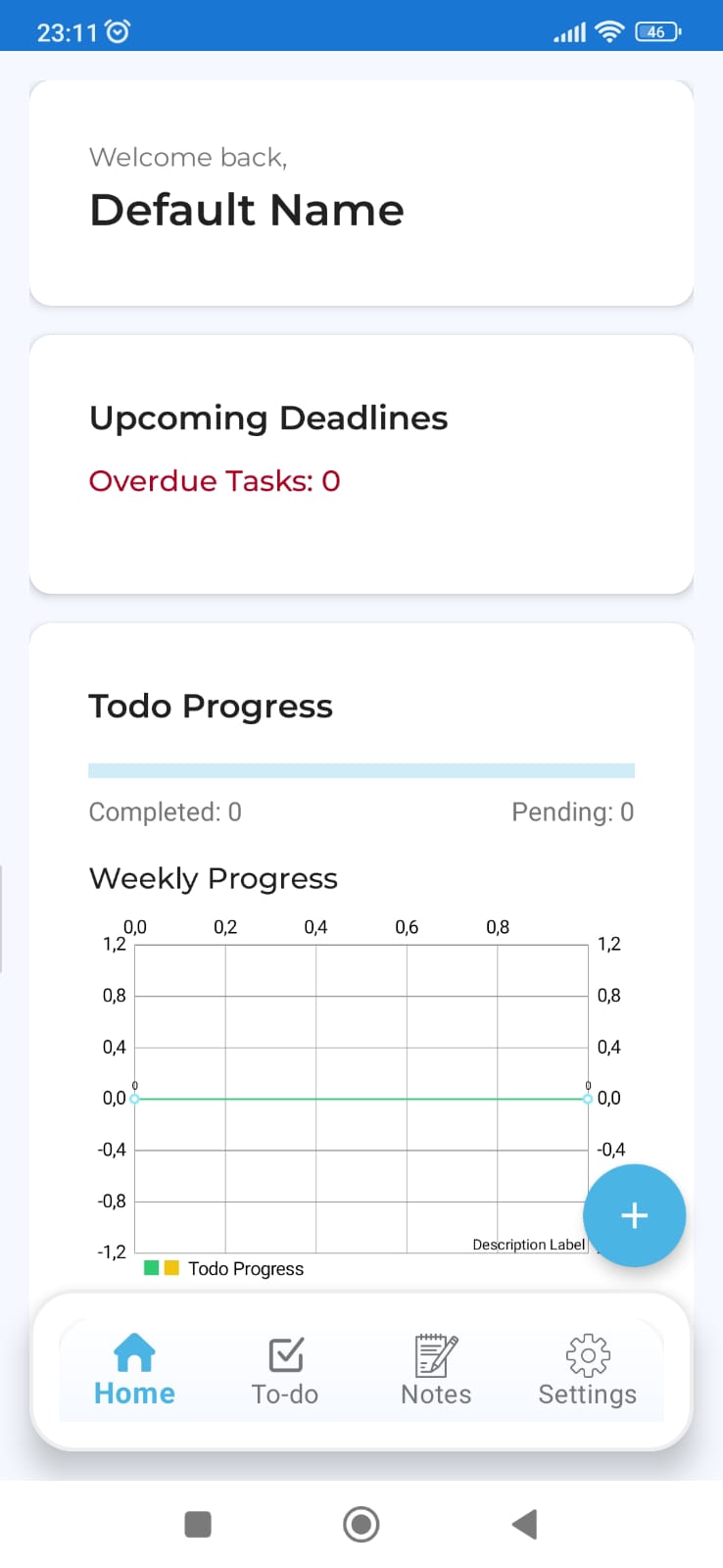
**Fitur yang disediakan:**

* **CRUD (Create, Read, Update, Delete)**:
  + **Create**: Menambahkan catatan baru dengan konten yang relevan.
  + **Read**: Membaca dan melihat daftar catatan yang telah dibuat.

30

* + **Update**: Mengedit isi catatan untuk memperbarui informasi.
  + **Delete**: Menghapus catatan yang tidak dibutuhkan.
* **Tagging**:
  + Pengguna dapat menambahkan **tag** pada catatan untuk mengelompokkan berdasarkan tema atau kategori tertentu.
* **Filter**:
  + Menyaring catatan berdasarkan tag atau kategori tertentu.
* **Search**:
  + Mencari catatan berdasarkan kata kunci yang ada di dalam judul atau isi catatan.
* **Sort**:
  + Mengurutkan catatan berdasarkan tanggal pembuatan, tanggal
  + pembaruan, atau alfabetis.

**3. Statistik Progres**

**** 31

Fitur ini memberikan gambaran visual tentang kemajuan pengguna dalam menyelesaikan tugas.

**Fitur yang disediakan:**

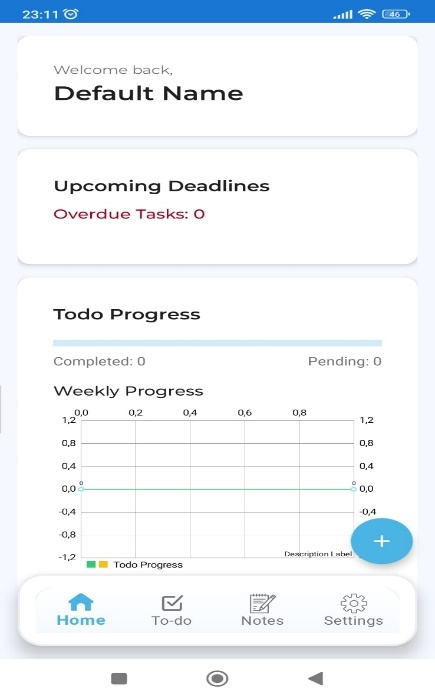
* **Grafik Penyelesaian Tugas**:
  + Menampilkan jumlah tugas yang selesai dan belum selesai dalam bentuk **diagram batang**, **pie chart**, atau **line chart**.
  + Statistik ini diperbarui secara real-time saat pengguna menandai tugas sebagai selesai atau menambah tugas baru.
* **Rincian Progres**:
  + Presentasi jumlah tugas yang selesai berdasarkan waktu (harian, mingguan, bulanan).
  + Membandingkan tingkat penyelesaian dengan target pengguna.
* **Motivasi Visual**:
  + Fitur tambahan berupa **progress bar** yang menunjukkan sejauh mana pengguna telah menyelesaikan tujuan harian atau mingguan.

**4.3 Hasil Pengujian**

Pada tahap ini, aplikasi Lifescape telah berhasil mengimplementasikan berbagai fitur utama yang mendukung manajemen tugas dan catatan pengguna. Berikut adalah hasil dan diskusi dari masing-masing fitur yang telah dikembangkan dan dioptimalkan selama proses debugging:

32

**1. Home/Dashboard: Statistik dan Grafik Todo**



**Hasil:**

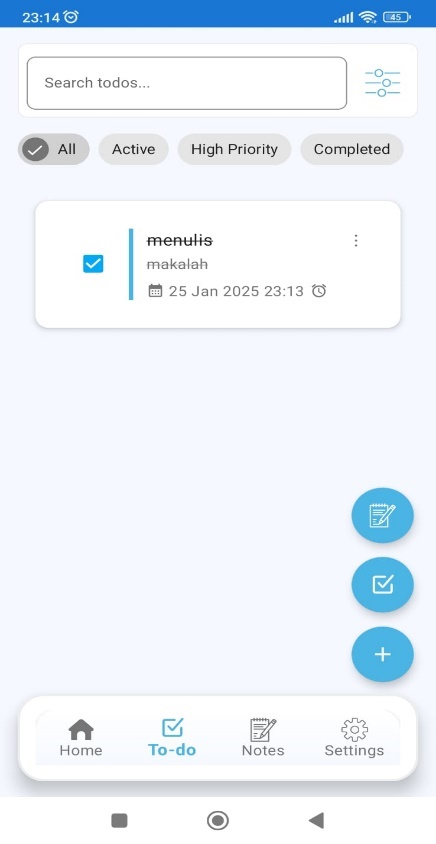
* Menampilkan statistik jumlah tugas yang Completed dan Pending dalam bentuk teks.
* Menyediakan grafik progres tugas menggunakan MPAndroidChart yang memperlihatkan perbandingan tugas yang telah selesai dan yang belum.

**Diskusi:**

* **Permasalahan:** Awalnya data statistik tidak terupdate secara real-time.

33

**2. Todo List: CRUD Tugas dengan Prioritas dan Tenggat Waktu**



**Hasil:**

* CRUD Todo (Create, Read, Update, Delete) berhasil diimplementasikan.
* Pengguna dapat mengatur prioritas (Low, Medium, High) dan deadline (otomatis 24 jam atau manual).
* Implementasi swipe gesture pada Todo:
* Swipe kanan → Tandai selesai.
* Swipe kiri → Konfirmasi hapus.
* Sorting dan Searching berbasis algoritma C (QuickSort & Binary Search) berjalan optimal.

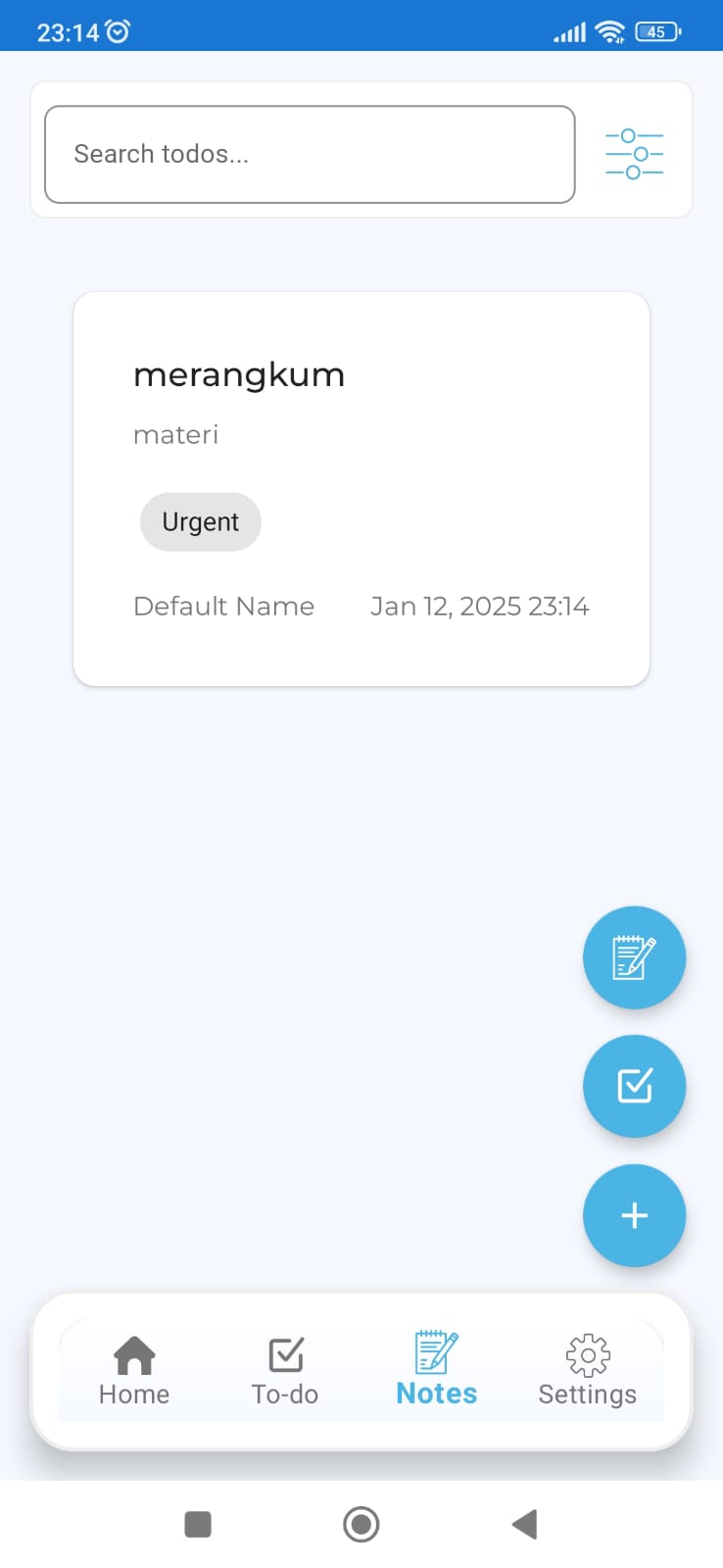
**Diskusi:**

**Permasalahan:**

* Foreign Key Constraint Error saat insert data Todo.
* Sort dan Search awalnya tidak berfungsi.

34

**3. Notes: CRUD Catatan dengan Fitur Tag, Sorting, dan Pencarian**



**Hasil:**

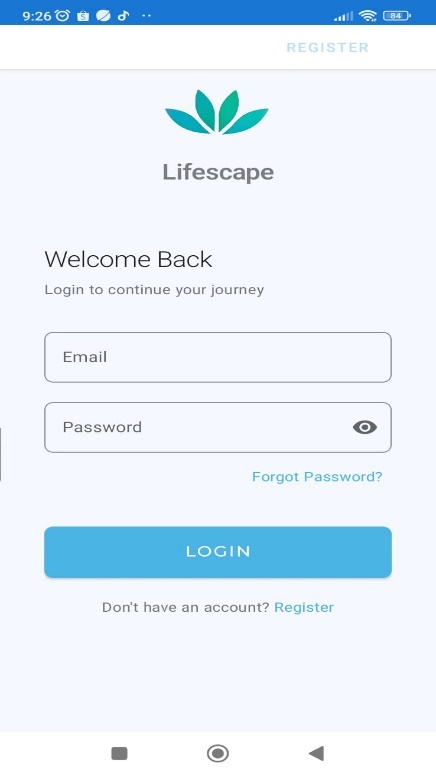
* CRUD Notes berjalan optimal.
* Penambahan dan penghapusan Tag (maksimal 3 tag per note).
* Filtering Notes berdasarkan Tag.
* Sorting berdasarkan Title, Created At, dan Updated At.
* Searching Notes menggunakan algoritma Binary Search di NativeUtils.

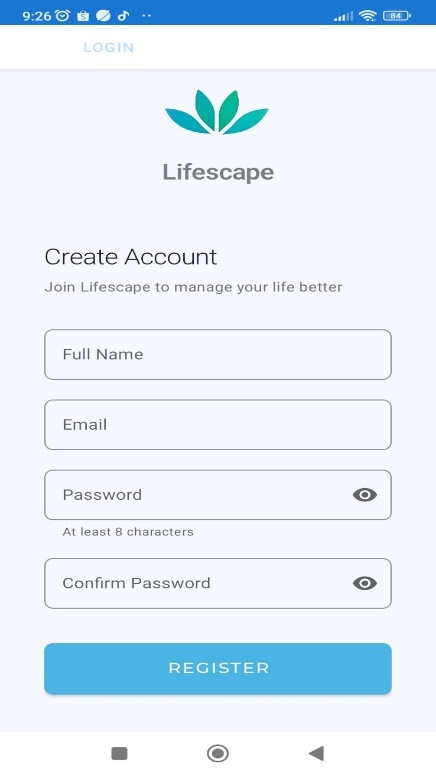
**Diskusi:**

**Permasalahan:**

* Filter tag tidak langsung menampilkan hasil.
* Fitur sort dan search awalnya tidak berfungsi.

35

**4. Integrasi Firebase Authentication dan RoomDB**



**Hasil:**

* Firebase Authentication digunakan untuk proses login, register, dan reset password.
* Data pengguna disinkronkan ke RoomDB untuk operasi offline.
* Validasi User ID diterapkan di seluruh fitur Todo dan Notes.

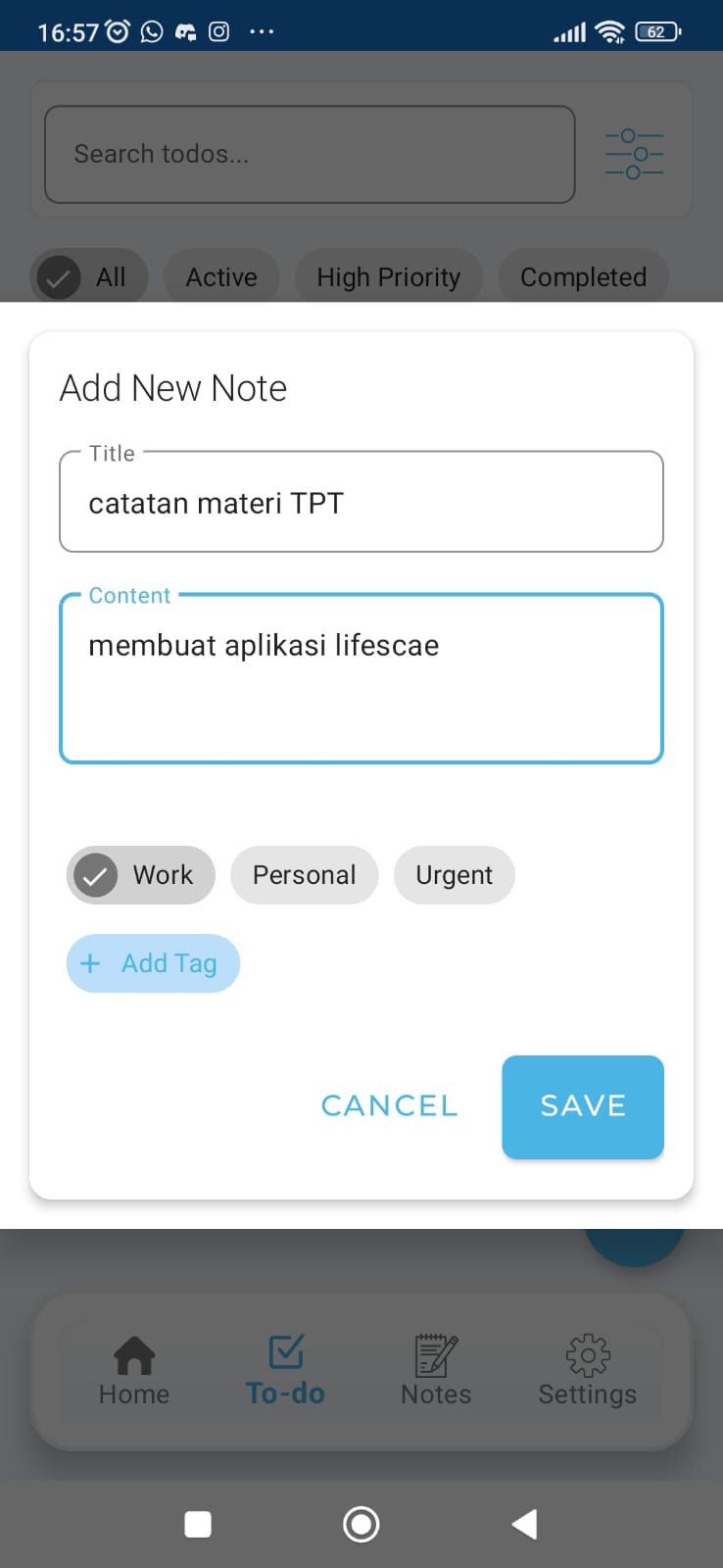
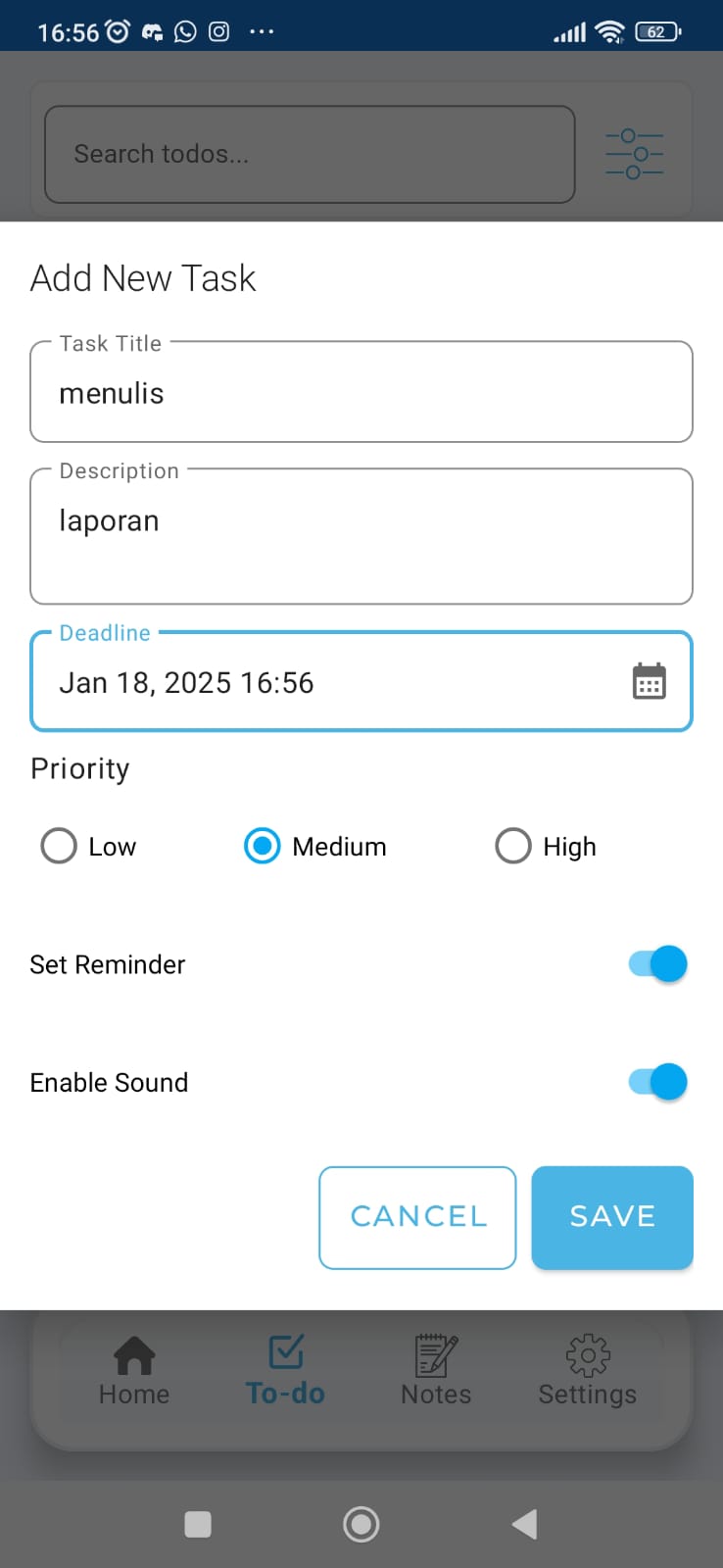
**Diskusi:**

**Permasalahan:**

* Data User tidak sinkron antara Firebase dan RoomDB.
* Saat logout, aplikasi tidak langsung mengarah ke AuthActivity.

36

**5. UI/UX dan Gesture Interaction**



**Hasil:**

* Implementasi Material Design dengan animasi Fade In/Out pada FAB.
* Dialog konfirmasi (hapus) dan form tambah Todo/Notes menggunakan desain yang responsif.
* Snackbar digunakan sebagai feedback pengguna.

**Diskusi:**

**Permasalahan:**

* Dialog konfirmasi mendem di belakang Bottom Navigation.

**Kesimpulan Hasil dan Diskusi:**

1. Fungsi CRUD Todo dan Notes berjalan sempurna setelah debugging dan optimasi query di database.

37

2. Sorting dan Searching berbasis algoritma C (QuickSort dan Binary Search) telah diimplementasikan dengan efektif.

3. Statistik Todo dan progress grafik sudah berjalan dengan baik menggunakan MPAndroidChart.

4. Sinkronisasi data pengguna antara Firebase Auth dan RoomDB sudah optimal.

5. Pengalaman pengguna (UX) ditingkatkan dengan tampilan antarmuka yang responsif dan umpan balik aksi yang jelas.

Dengan perbaikan dan pengoptimalan ini, aplikasi Lifescape telah siap digunakan untuk manajemen tugas dan catatan dengan fitur yang lengkap dan berjalan lancar.

**4.4 Debugging dan Solusi**

**1. Home/Dashboard: Statistik dan Grafik Todo**

* Solusi: Menambahkan observer di LiveData dan memisahkan query untuk Completed dan Pending.
* Pengoptimalan: Grafik diperbarui setiap kali ada perubahan pada data Todo.

**2. Todo List: CRUD Tugas dengan Prioritas dan Tenggat Waktu**

* Solusi:

1. Memastikan userId tersimpan dan divalidasi di database.
2. Memperbaiki implementasi NativeUtils untuk sorting dan searching.

* Pengoptimalan:

1. Fitur filtering Todo berdasarkan status (All, Active, Completed) berjalan dengan baik.
2. Implementasi Snackbar sebagai feedback pengguna saat aksi dilakukan.

**3. Notes: CRUD Catatan dengan Fitur Tag, Sorting, dan Pencarian**

* Solusi:

1. Perbaikan query di NotesDao dan implementasi real-time update di ChipGroup.
2. Memperbaiki logika sort dan search dengan algoritma C.

38

* Pengoptimalan:

1. Notes langsung terupdate setelah diubah tanpa perlu reload.
2. Snackbar digunakan untuk memberi notifikasi saat Notes berhasil disimpan atau dihapus.

**4. Integrasi Firebase Authentication dan RoomDB**

* Solusi:

1. Menambahkan validasi dan sinkronisasi data user pada saat login dan register.
2. Memperbaiki alur logout agar langsung menghapus data lokal dan mengarahkan ke halaman login.

* Pengoptimalan:

1. Penerapan Snackbar sebagai notifikasi saat login dan logout.
2. Penambahan validasi input di semua form login dan register.

**5. UI/UX dan Gesture Interaction**

* Solusi:

1. Mengganti Snackbar dengan Toast atau menggunakan CoordinatorLayout untuk memperbaiki tampilan.

* Pengoptimalan:

1. Swipe gesture berjalan lebih responsif setelah optimasi RecyclerView.

39

**BAB V**

**PENUTUP**

**5.1 Kesimpulan**

Berdasarkan hasil pengembangan dan pengujian aplikasi Lifescape, dapat disimpulkan bahwa:

1. **Keberhasilan Fitur Utama**

Aplikasi berhasil mengimplementasikan fitur-fitur utama seperti:

* **Todo List:** CRUD tugas dengan prioritas, tenggat waktu, dan notifikasi.
* **Notes:** Menyimpan catatan dengan fitur filter tag, sorting, dan pencarian berbasis algoritma C.
* **Statistik Todo:** Menampilkan grafik progres Todo menggunakan MPAndroidChart.
* **Firebase Authentication:** Otentikasi login dan register berjalan optimal, termasuk sinkronisasi data user ke RoomDB.

**2. Stabilitas dan Performa Aplikasi**

Optimalisasi pada penggunaan Room Database, pengelolaan LiveData, dan eksekusi background menggunakan ExecutorService membuat aplikasi berjalan lancar tanpa lag. Penggunaan Snackbar juga meningkatkan pengalaman pengguna.

**3. Keamanan dan Validasi Data**

Validasi input pada fitur login, register, dan CRUD Todo/Notes mencegah error akibat data kosong atau tidak valid. Selain itu, penerapan Foreign Key di database memastikan integritas data antara user dengan Todo dan Notes.

**4. User Interface (UI) dan User Experience (UX)**

Desain aplikasi berbasis Material Design berhasil memberikan tampilan yang modern dan responsif. Animasi dan transisi yang diimplementasikan membuat aplikasi lebih interaktif.

40

**5.2 Saran**

Untuk pengembangan lebih lanjut, beberapa saran yang dapat diterapkan adalah:

1. **Pengembangan Fitur Tambahan**

* **Export Data ke PDF/Excel:** Tambahkan fitur ekspor laporan tugas dan catatan ke format PDF atau Excel.
* **Reminder Notifikasi:** Implementasi notifikasi pengingat otomatis berdasarkan tenggat waktu di Todo.
* **Sinkronisasi Cloud:** Integrasi sinkronisasi data Todo/Notes ke Firebase Realtime Database atau Firestore untuk akses multi-device.

2. **Peningkatan UI/UX**

* **Dark Mode:** Tambahkan fitur mode gelap untuk kenyamanan pengguna.
* **Custom Theme:** Berikan opsi personalisasi tema agar lebih menarik.

3. **Optimalisasi Kinerja**

* **Database Migration:** Pastikan skema database dapat di-upgrade tanpa kehilangan data.
* **Optimasi Query:** Lakukan optimalisasi query di DAO agar lebih efisien.

4. **Keamanan Aplikasi**

* **Validasi Otentikasi Lebih Kuat:** Tambahkan verifikasi email atau OTP saat registrasi.
* **Data Encryption:** Terapkan enkripsi pada data sensitif yang tersimpan di lokal.

5. **Pengujian dan Deployment**

* **Cross-Device Testing**: Uji aplikasi di berbagai perangkat dan versi Android untuk memastikan kompatibilitas.
* **Pengoptimalan APK:** Terapkan ProGuard dan optimasi build APK agar lebih ringan.
* **Publikasi ke Play Store:** Siapkan dokumentasi dan aset aplikasi untuk rilis di Google Play Store.

41

**DAFTAR PUSTAKA**

**TodoList App :**

* <https://github.com/wizardoftrap/To_Do_List>
* <https://github.com/theodoremoreland/TodoListApp>
* <https://github.com/tanyagupta0201/ToDo-List-App>

42