
Software Requirements Specification

untuk mobile app

Mindly

Versi 1.0

Oleh :

Susan Jong (231401014)

Parulian Dwi Reslia (231401032)

Charissa Haya Qanita (231401113)

Program Studi Ilmu Komputer

Universitas Sumatera Utara

30 Oktober 2025

Daftar Isi

1. Pendahuluan.....	4
1.1 Tujuan.....	4
1.2 Lingkup Masalah.....	4
1.3 Batasan Produk.....	5
1.4 Audien yang Ditujukan dan Pembaca yang Disarankan.....	6
1.5 Definisi, Akronim, dan Singkatan.....	7
1.6 Metodologi Pengembangan.....	9
1.7 Referensi.....	9
2. Deskripsi Keseluruhan.....	9
2.1 Perspektif Produk.....	9
2.2 Fungsi Produk.....	10
2.3 Karakteristik Pengguna.....	12
2.4 Lingkungan Operasi.....	15
3. Fitur-Fitur Sistem.....	15
3.1 Fitur Login dengan Google.....	15
3.2 Fitur Login Manual.....	16
3.3 Fitur Register Akun Manual.....	17
3.4 Fitur Forgot Password / Reset Password.....	18
3.5 Fitur Dashboard Harian.....	18
3.6 Fitur Notifikasi.....	19
3.7 Fitur Mark All Read Notification.....	20
3.8 Fitur Profile dan Pengaturan Akun.....	21
3.9 Fitur Edit Account Information.....	22
3.10 Fitur Notification Preference Settings.....	22
3.11 Fitur Help & FAQ.....	23
3.12 Fitur About Application.....	24
3.13 Fitur Reset Password (dari Profile).....	24
3.14 Fitur Delete Account.....	25
3.15 Fitur Melihat Semua Catatan (All Notes).....	26
3.16 Fitur Filter Notes by Category.....	26
3.17 Fitur Filter Notes by Favorite.....	27
3.18 Fitur Search Notes by Keyword.....	27
3.19 Fitur Menambah Catatan Baru.....	28
3.20 Fitur Mengedit Catatan.....	29
3.21 Fitur Menghapus Catatan.....	29
3.22 Fitur Menambahkan Catatan ke Favorite.....	30
3.23 Fitur Long Press Notes untuk Quick Actions.....	30
3.24 Fitur Menambah Kategori Notes.....	31
3.25 Fitur Mengedit Nama Kategori Notes.....	31

3.26 Fitur Menghapus Kategori Notes.....	32
3.27 Fitur Menambahkan Kategori ke Favorite.....	33
3.28 Fitur Multi-Select & Batch Actions (Aksi Massal).....	33
3.29 Fitur Melihat All To-Do List.....	34
3.30 Fitur Melihat Urgent To-Do List.....	35
3.31 Fitur Melihat Overdue To-Do List.....	36
3.32 Fitur Menambah Task Baru.....	37
3.33 Fitur Mengedit Task.....	38
3.34 Fitur Menghapus Task.....	38
3.35 Fitur Menandai Task sebagai Completed.....	39
3.36 Fitur Menambah Kategori To-Do List.....	39
3.37 Fitur Melihat Progress Kategori To-Do List.....	40
3.38 Fitur Melihat Uncategorized Task.....	40
3.39 Fitur Mengatur Deadline Task.....	41
3.40 Fitur Mengatur Deskripsi Task.....	42
3.41 Fitur Melihat Calendar (Day/Month/Year View).....	42
3.42 Fitur Menambah Event di Calendar.....	43
3.43 Fitur Mengedit Event di Calendar.....	44
3.44 Fitur Menghapus Event di Calendar.....	45
3.45 Fitur Mengatur Reminder untuk Event.....	46
3.46 Fitur Mengatur Event Repeat/Recurring.....	47
3.47 Fitur Mengkategorikan Event.....	48
3.48 Fitur Search Event by Keyword.....	48
3.49 Fitur Log Out.....	49
4. System Models & Diagram.....	50
4.1 Use Case Diagram.....	50
Use Case Utama:.....	51
4.2 Activity Diagram.....	52
4.3 Data Flow Diagram.....	56
5. Antarmuka Pengguna (UI).....	61
5.1 Antarmuka Perangkat Keras.....	62
5.2 Antarmuka Perangkat Lunak.....	62
5.3 Antarmuka Komunikasi.....	63
6. Kebutuhan Non Fungsional.....	63
6.2 Kebutuhan Kinerja.....	65
6.3 Kebutuhan Keamanan.....	66
6.3.1 Diagram Alur Keamanan.....	68
6.4 Requirement Keselamatan.....	69
6.4.1 Pencegahan Error.....	69
6.6.2 Pemulihan.....	69
6.5 Requirement Lain.....	70
6.5.1 Kompabilitas.....	70

6.5.2 Pemeliharaan (Maintenance).....	71
Additional Notes.....	71

1. Pendahuluan

1.1 Tujuan

Dokumen *Software Requirements Specification* (SRS) ini disusun sebagai landasan penting dalam proses pengembangan aplikasi Mindy. Tujuan utama dari dokumen ini adalah untuk memberikan gambaran yang jelas dan rinci mengenai kebutuhan perangkat lunak, baik dari segi fungsional maupun non-fungsional, guna memastikan bahwa aplikasi yang dikembangkan memenuhi harapan semua pihak terkait. Aplikasi ini dirancang untuk mendukung produktivitas pengguna dengan mengintegrasikan berbagai fitur manajemen kegiatan harian, catatan, dan kalender dalam satu platform terpadu. Selain itu, Dokumen ini menjadi pedoman utama bagi tim pengembang, penguji, dan pihak terkait lainnya dalam memahami dan menerjemahkan kebutuhan pengguna menjadi desain yang sesuai lingkup sistem, kapabilitas, serta spesifikasi teknis yang harus diimplementasikan selama proses pengembangan aplikasi.

Dokumen ini ditunjukkan untuk mengurangi potensi kesalahan atau ketidaksesuaian selama proses pengembang dengan menyediakan spesifikasi yang jelas, terstruktur, dan terukur. Dengan adanya dokumen ini, pengembang dan pemangku kepentingan dapat memantau proses pengembangan perangkat lunak untuk memastikan bahwa aplikasi yang dihasilkan dapat digunakan secara optimal oleh pengguna, terutama orang tua, tenaga kesehatan, dan pemerintah. Lebih jauh lagi, dokumen ini akan menjadi dasar untuk evaluasi dan verifikasi perangkat lunak pada tahap akhir guna memastikan bahwa produk akhir mendukung tujuan utama, yaitu memantau dan mencegah stunting secara efektif melalui teknologi berbasis data.

1.2 Lingkup Masalah

Aplikasi Mindly dirancang untuk membantu pengguna mengelola aktivitas, catatan, dan jadwal mereka dalam satu platform terpadu. Masalah utama yang diatasi Mindly adalah fragmentasi aktivitas produktivitas, di mana pengguna sering kali harus berpindah antar aplikasi seperti *to-do list*, kalender, dan catatan untuk menyimpan catatan atau ide dari user. Kebiasaan berpindah aplikasi ini menyebabkan hilangnya fokus, duplikasi informasi, dan rendahnya efisiensi dalam pengelolaan waktu maupun tugas.

Situasi seperti ini dapat menimbulkan beberapa masalah, seperti yang berikut ini:

- a. Hasil pengguna yang terpisah dapat menciptakan kesenjangan dan dapat membuat

- pengguna kesulitan untuk mengaitkan catatan dengan tugas dan jadwal.
- b. Hal ini juga dapat menimbulkan gangguan alur kerja, di mana pengguna diperlukan untuk berpindah-pindah antara aplikasi yang mengakibatkan penurunan fokus dan produktivitas.
 - c. Permasalahan lain yang dapat diakibatkan dari kurangnya sistem terintegrasi pengelola informasi adalah duplikasi informasi yang berpotensi menyimpan hal yang sama di banyak tempat dan informasi yang hilang atau tidak sinkron.
 - d. Distraksi yang dihasilkan dari kesenjangan suatu aplikasi dapat mengakibatkan kesulitan pengguna di dalam hal mengingat kontekstual yang disertakan dalam mengingat yang diatur.

Dengan demikian, Mindly tidak hanya bertujuan untuk menyatukan fungsi-fungsi produktivitas menjadi satu aplikasi saja, tetapi menghadirkan solusi yang mengintegrasikan tiga fungsi utama, *to-do list*, *calendar*, dan *notes*, dalam satu antarmuka yang intuitif, adaptif, dan kontekstual untuk mengoptimalkan efisiensi waktu, mengurangi beban kognitif pengguna, dan meningkatkan konsistensi dalam manajemen aktivitas sehari-hari.

1.3 Batasan Produk

Aplikasi Mindly dirancang untuk membantu pengguna dalam pengelolaan kegiatan, catatan, dan jadwal secara terpadu, meskipun pada awal pengembangannya, ada beberapa batasan yang perlu dicamkan, antara lain:

1. Input Data Secara Manual

Untuk daftar tugas, catatan dan jadwal, pengguna masih harus input secara manual. Untuk versi awal aplikasi, pengembang belum menyentuh beberapa fitur otomatisasi yang lebih advanced, seperti mendeteksi kalender luar, input suara, dan diintegrasikan dengan berbasis Artificial Intelligence dibuat rekomendasi pintar.

2. Dependensi pada Data User

Aktivitas pengelolaan dan rekomendasi aktivitas yang bersifat konteks dan pengelolaan aktivitas bersifat pengelolaan dan rekomendasi pengelolaan aktivitas bersifat pengelolaan dan rekomendasi aktivitas dan pengguna tidak sinkron, atau tugas.

3. Sinkronisasi yang Terbatas

Pada awal pengembangan Mindly, fitur sinkronisasi dan penyimpanan otomatis di *cloud computing* dan lintas perangkat belum dikembangkan.

4. Terbatasnya pada Fitur Kolaborasi

Mindly pada awal pengembangan berorientasi penggunaan pada individu. Fitur kolaborasi yang memungkinkan pengguna, tim, teman, atau keluarga untuk melakukan sharing catatan, tugas, atau kalender secara paralel belum di buat.

5. Internet Access

Beberapa fitur tambahan seperti sinkronisasi atau pencadangan otomatis data, yang tersedia belum dibuat untuk sementara waktu.

1.4 Audien yang Ditujukan dan Pembaca yang Disarankan

Dokumen ini dirancang untuk dibaca dan digunakan oleh berbagai pihak yang terlibat langsung maupun tidak langsung dalam pengembangan dan implementasi aplikasi Mindly. Berikut adalah audien yang dituju:

a. Tim Pengembang Perangkat Lunak

Untuk membantu tim pengembang perangkat lunak memahami kebutuhan dan implementasi fitur aplikasi, dokumen ini berfungsi sebagai panduan utama. Semua spesifikasi teknis yang dibahas dalam dokumen ini dirancang untuk membantu pengembang menerjemahkan kebutuhan pengguna ke dalam fungsi aplikasi yang optimal.

b. Manajer proyek

Manajer proyek menggunakan dokumen ini sebagai pedoman untuk mengelola proses pengembangan perangkat lunak agar kesesuaian dengan tujuan dan ruang lingkup yang ditetapkan. Ini juga membantu merencanakan tahapan pengembangan, menetapkan prioritas fitur, dan memantau kemajuan proyek sesuai jadwal dan anggaran.

c. Penguji Sistem Perangkat Lunak

Dokumen ini berfungsi sebagai referensi utama dalam desain skenario pengujian perangkat lunak untuk memastikan bahwa semua fitur aplikasi berjalan sesuai spesifikasi dan memenuhi kebutuhan pengguna.

d. Desainer UI/UX

Desainer antarmuka pengguna menggunakan dokumen ini untuk menciptakan desain aplikasi yang ramah pengguna, menarik secara visual, dan fungsional sesuai kebutuhan pengguna.

e. Perwakilan Pengguna

Ini adalah bagian untuk membantu pengguna dan calon pengguna memahami fitur utama Mindly. Ini termasuk bagaimana aplikasi membantu mengelola catatan, jadwal, dan aktivitas sehari-hari di ekosistem digital berbasis aplikasi *mobile*.

f. Penulis Dokumentasi

Panduan pengguna, dokumentasi teknis, dan materi bantuan aplikasi yang jelas, sesuai dengan alur dan fungsi aplikasi yang dijelaskan di dalam spesifikasi, didasarkan pada dokumen ini.

g. Dosen Pengampu Mata Kuliah

Selain itu, dokumen ini ditujukan untuk guru atau pihak penilai yang bertanggung jawab untuk menilai kesesuaian rancangan sistem dengan tujuan pembelajaran. Struktur desain, fungsi, dan fitur aplikasi Mindly telah dijelaskan secara sistematis dan sesuai dengan prinsip rekayasa perangkat lunak untuk menilai seberapa besar kapasitas aplikasi memenuhi kebutuhan sistem.

1.5 Definisi, Akronim, dan Singkatan

Untuk memastikan bahwa semua pihak yang terlibat dalam pengembangan dan implementasi sistem memiliki pemahaman yang sama, bagian ini menjelaskan istilah, akronim, dan singkatan yang digunakan dalam dokumen Persyaratan Persyaratan Software (SRS) untuk aplikasi Mindly.

1. Definisi

- Mindly

Aplikasi mobile Mindly memungkinkan pengguna mengelola catatan, daftar tugas (to-do list), dan jadwal harian dalam satu platform terpadu dengan antarmuka yang mudah digunakan dan mudah digunakan.

- Catatan (Notes)

Fitur aplikasi yang memungkinkan pengguna menulis, menyimpan, dan mengorganisasi konsep, informasi, atau pengingat penting secara digital.

- Daftar Tugas (To-Do-List)

Fitur yang digunakan untuk mencatat, mengatur, dan memantau tugas atau aktivitas yang harus diselesaikan oleh pengguna sesuai prioritas dan tenggat waktu.

- **Event Calender**
Kalender adalah komponen aplikasi yang membantu pengguna menjadwalkan kegiatan. Aplikasi ini memiliki catatan dan daftar tugas serta agenda harian, mingguan, dan bulanan.
- **Pengguna Aplikasi (User)**
Orang yang menggunakan aplikasi Mindly untuk menyimpan catatan, mengatur jadwal, dan mengatur aktivitas pribadi mereka.
- **Antarmuka Pengguna (UI)**
Komponen aplikasi yang berinteraksi langsung dengan pengguna dan terdiri dari komponen visual seperti tombol, ikon, dan tata letak.
- **Pengalaman Pengguna (User Experience/UX)**
Bagaimana pengguna melihat dan puas dengan kemudahan, kenyamanan, dan efektivitas penggunaan aplikasi.
- **Sinkronisasi (Synchronization)**
Proses menyatukan data antara perangkat lokal pengguna dan penyimpanan berbasis awan untuk memastikan bahwa data konsisten di seluruh perangkat.
- **Kolaborasi (Collaboration)**
Fitur yang memungkinkan satu pengguna untuk ke pengguna lainnya dapat melakukan berbagi untuk mengakses catatan dari pemilik catatan yang dibuat kemudian dibagikan ke orang lain dan pengguna lain dapat melakukan menambahkan, modifikasi, dan lain-lain pada catatan yang dibuat oleh pemilik catatan tersebut.

2. Akronim dan Singkatan

- **SRS:** *Software Requirements Specification*
- **UI:** *User Interface*
- **UX:** *User Experience*
- **API:** *Application Programming Interface*
- **SRS-MINDLY.K-xxxx:** Kode yang digunakan untuk merepresentasikan kebutuhan pada Mindly, dengan MINDLY merupakan kode perangkat lunak, MINDLY.K adalah kode fase, dan xxxx adalah digit/nomor kebutuhan.
- **KK-xxx:** Kode yang digunakan untuk merepresentasikan kebutuhan

keamanan pada sistem aplikasi Mindly, dimana xxx adalah ID/nomor kebutuhan.

1.6 Metodologi Pengembangan

Untuk memastikan pengembangan aplikasi Mindly berjalan secara terstruktur dan sistematis, proyek ini menggunakan model Waterfall sebagai alur pengembangannya. Metode Waterfall (Model Air Terjun) adalah pendekatan SDLC yang bersifat linear dan sekuensial. Dalam model ini, setiap fase pengembangan harus diselesaikan sepenuhnya sebelum beralih ke fase berikutnya. Alur prosesnya mengalir satu arah dari analisis kebutuhan hingga *maintenance*.

Alasan dari dipilihnya metode Waterfall untuk pengembangan aplikasi Mindly adalah karena model ini menyediakan struktur jelas, yang memudahkan pengelolaan, pelacakan progres, dan memastikan bahwa setiap fase terdokumentasi dengan baik sebelum tim melanjutkan ke pekerjaan berikutnya.

1.7 Referensi

- K. E. Wiegers, “Software Requirements Specification IEEE,” 1999.
- Materi perkuliahan Analisis dan Desain Perangkat Lunak.
- Materi perkuliahan Pemrograman Mobile

2. Deskripsi Keseluruhan

2.1 Perspektif Produk

Mindly merupakan aplikasi produktivitas terpadu berbasis android yang dirancang untuk membantu pengguna dalam mengatur kegiatan, mencatat ide, dan memantau progres harian secara efisien. Alih-alih bergantung pada beberapa aplikasi terpisah seperti *Google Keep*, *Todoist*, atau *Google Calendar*, Mindly mengintegrasikan seluruh fungsi tersebut dalam satu ekosistem intuitif.

Masalah utama yang ingin diatasi:

a. Fragmentasi Aplikasi Produktivitas

Pengguna sering berpindah antar aplikasi hanya untuk mengakses catatan, daftar tugas, dan jadwal, mengurangi fokus dan konsistensi.

b. **Kurangnya Kesadaran Progres Harian**

Sulit untuk memantau sejauh mana target harian atau mingguan telah tercapai tanpa tampilan ringkasan terpusat.

c. **Kehilangan Konteks antar Aktivitas**

Catatan, agenda, dan tugas sering kali berdiri sendiri tanpa keterkaitan yang jelas, menyebabkan hilangnya konektivitas informasi.

Mindly menawarkan pendekatan berbeda melalui integritas penuh:

- a. **Dashboard Dinamis:** Menampilkan progres harian (jumlah tugas selesai/total tugas), pengingat tenggat terdekat, *upcoming events*, dan *daily notes update*.
- b. **Notes Management:** Sistem pencatatan dengan kategori, favorit, pencarian cepat, serta pengelompokan berdasarkan folder atau topik.
- c. **To-Do List Management:** Fitur pembuatan tugas harian, pengelompokan berdasarkan folder, tampilan *progress bar*, serta klasifikasi tugas *urgent*, *overdue*, dan *uncategorized*.
- d. **Calendar Integration:** Penjadwalan acara dengan opsi pengingat, *repeat event*, dan pengkategorian kegiatan.
- e. **Profile & Notification Center:** Menyediakan pengaturan akun, preferensi notifikasi, *help center*, serta laporan mingguan dan *achievement recap*.

Dengan kombinasi fitur tersebut, Mindly tidak hanya membantu pengguna mencatat dan menjadwalkan aktivitas, tetapi juga menumbuhkan kesadaran produktivitas melalui pemantauan progres yang jelas dan antarmuka yang ramah pengguna.

2.2 Fungsi Produk

No	ID Fungsi	Fungsi Perangkat Lunak
1	SRS-MINDLY-0001	Fungsi Log In dengan akun Google atau Manual
2	SRS-MINDLY-0002	Fungsi Register Account Manual
3	SRS-MINDLY-0003	Fungsi Reset/Forgot Password
4	SRS-MINDLY-0004	Fungsi Dashboard Harian (Daily Progress, Today's Event, Today's Notes)

5	SRS-MINDLY-0005	Fungsi Notifikasi (Daily Goal, Upcoming Deadline, Weekly Report, Overdue Task)
6	SRS-MINDLY-0006	Fungsi Mark All Read Notification
7	SRS-MINDLY-0007	Fungsi Profil dan Pengaturan Akun
8	SRS-MINDLY-0008	Fungsi Edit Account Information
9	SRS-MINDLY-0009	Fungsi Notification Preference Settings
10	SRS-MINDLY-0010	Fungsi Help & FAQ
11	SRS-MINDLY-0011	Fungsi About Application
12	SRS-MINDLY-0012	Fungsi Reset Password (dari Profile)
13	SRS-MINDLY-0013	Fungsi Delete Account
14	SRS-MINDLY-0014	Fungsi Melihat Semua Catatan (All Notes)
15	SRS-MINDLY-0015	Fungsi Filter Notes by Category
16	SRS-MINDLY-0016	Fungsi Filter Notes by Favorite
17	SRS-MINDLY-0017	Fungsi Search Notes by Keyword
18	SRS-MINDLY-0018	Fungsi Menambah Catatan Baru
19	SRS-MINDLY-0019	Fungsi Mengedit Catatan
20	SRS-MINDLY-0020	Fungsi Menghapus Catatan
21	SRS-MINDLY-0021	Fungsi Menambahkan Catatan ke Favorite
22	SRS-MINDLY-0022	Fungsi Memindahkan Catatan ke Kategori Lain (Move To)
23	SRS-MINDLY-0023	Fungsi Long Press Notes untuk Quick Actions
24	SRS-MINDLY-0024	Fungsi Melihat Daftar Kategori Notes
25	SRS-MINDLY-0025	Fungsi Menambah Kategori Notes
26	SRS-MINDLY-0026	Fungsi Mengedit Nama Kategori Notes
27	SRS-MINDLY-0027	Fungsi Menghapus Kategori Notes
28	SRS-MINDLY-0028	Fungsi Menambahkan Kategori ke Favorite
29	SRS-MINDLY-0029	Fungsi Melihat Notes dalam Kategori Tertentu
30	SRS-MINDLY-0030	Fungsi Melihat All To-Do List

31	SRS-MINDLY-0031	Fungsi Melihat Urgent To-Do List
32	SRS-MINDLY-0032	Fungsi Melihat Overdue To-Do List
33	SRS-MINDLY-0033	Fungsi Menambah Task Baru
34	SRS-MINDLY-0034	Fungsi Mengedit Task
35	SRS-MINDLY-0035	Fungsi Menghapus Task
36	SRS-MINDLY-0036	Fungsi Menandai Task sebagai Completed
37	SRS-MINDLY-0037	Fungsi Menambah Kategori To-Do List
38	SRS-MINDLY-0038	Fungsi Melihat Progress Kategori To-Do List
39	SRS-MINDLY-0039	Fungsi Melihat Uncategorized Task
40	SRS-MINDLY-0040	Fungsi Mengatur Deadline Task
41	SRS-MINDLY-0041	Fungsi Mengatur Deskripsi Task
42	SRS-MINDLY-0042	Fungsi Melihat Calendar (Day/Month/Year View)
43	SRS-MINDLY-0043	Fungsi Menambah Event di Calendar
44	SRS-MINDLY-0044	Fungsi Mengedit Event di Calendar
45	SRS-MINDLY-0045	Fungsi Menghapus Event di Calendar
46	SRS-MINDLY-0046	Fungsi Mengatur Reminder untuk Event
47	SRS-MINDLY-0047	Fungsi Mengatur Event Repeat/Recurring
48	SRS-MINDLY-0048	Fungsi Mengkategorikan Event
49	SRS-MINDLY-0049	Fungsi Search Event by Keyword
50	SRS-MINDLY-0050	Fungsi Log Out

2.3 Karakteristik Pengguna

Pengguna	Proses yang Dilakukan	Kode Perangkat Lunak
User (Pengguna Aplikasi)	Melakukan Log In dengan Google	SKPL-MINDLY-0001
	Melakukan Log In Manual	SKPL-MINDLY-0002
	Membuat Akun Baru (Register)	SKPL-MINDLY-0003

	Reset Password	SKPL-MINDLY-0004
	Melihat Dashboard Harian	SKPL-MINDLY-0005
	Melihat dan Membaca Notifikasi	SKPL-MINDLY-0006
	Mark All Read pada Notifikasi	SKPL-MINDLY-0007
	Melihat dan Mengedit Profil	SKPL-MINDLY-0008
	Mengatur Preference Notifikasi	SKPL-MINDLY-0009
	Mengakses Help & FAQ	SKPL-MINDLY-0010
	Melihat About Application	SKPL-MINDLY-0011
	Reset Password dari Profile	SKPL-MINDLY-0012
	Menghapus Akun (Delete Account)	SKPL-MINDLY-0013
	Melihat Semua Catatan	SKPL-MINDLY-0014
	Filter Catatan berdasarkan Kategori	SKPL-MINDLY-0015
	Filter Catatan Favorite	SKPL-MINDLY-0016
	Search Catatan by Keyword	SKPL-MINDLY-0017
	Menambah Catatan Baru	SKPL-MINDLY-0018
	Mengedit Catatan	SKPL-MINDLY-0019
	Menghapus Catatan	SKPL-MINDLY-0020
	Menambahkan/Menghapus Favorite pada Catatan	SKPL-MINDLY-0021
	Memindahkan Catatan ke Kategori Lain	SKPL-MINDLY-0022
	Long Press Catatan untuk Quick Action	SKPL-MINDLY-0023
	Melihat Daftar Kategori	SKPL-MINDLY-0024
	Menambah Kategori Baru	SKPL-MINDLY-0025
	Mengedit Nama Kategori	SKPL-MINDLY-0026
	Menghapus Kategori	SKPL-MINDLY-0027
	Menambahkan Kategori ke Favorite	SKPL-MINDLY-0028

	Melihat Notes dalam Kategori Tertentu	SKPL-MINDLY-0029
	Melihat All To-Do List	SKPL-MINDLY-0030
	Melihat Urgent To-Do List	SKPL-MINDLY-0031
	Melihat Overdue To-Do List	SKPL-MINDLY-0032
	Menambah Task Baru	SKPL-MINDLY-0033
	Mengedit Task	SKPL-MINDLY-0034
	Menghapus Task	SKPL-MINDLY-0035
	Mencentang Task sebagai Completed	SKPL-MINDLY-0036
	Menambah Kategori/Folder To-Do	SKPL-MINDLY-0037
	Melihat Progress Kategori	SKPL-MINDLY-0038
	Melihat Uncategorized Task	SKPL-MINDLY-0039
	Mengatur Deadline Task	SKPL-MINDLY-0040
	Mengatur Deskripsi Task	SKPL-MINDLY-0041
	Melihat Calendar (Day/Month/Year)	SKPL-MINDLY-0042
	Menambah Event Baru	SKPL-MINDLY-0043
	Mengedit Event	SKPL-MINDLY-0044
	Menghapus Event	SKPL-MINDLY-0045
	Mengkategorikan Event	SKPL-MINDLY-0046
	Mengatur Reminder Event	SKPL-MINDLY-0047
	Mengatur Event Repeat	SKPL-MINDLY-0048
	Search Event by Keyword	SKPL-MINDLY-0049
	Melakukan Log Out	SKPL-MINDLY-0050

2.4 Lingkungan Operasi

a. Sistem Operasi

1. Development Environment

- Windows 10/11

2. Target Platform

- Android: Minimum API Level 21 (Android 5.0 Lollipop) hingga API Level 34 (Android 14)
- iOS: Minimum iOS 12.0 atau lebih baru (jika dikembangkan untuk iOS)

b. Backend & Server-Side

1. **Operating System:** Android

2. **Platform:** Google Firebase

3. **Database Management System (DBMS):** Google Cloud Firestore, untuk menyimpan data pengguna, daftar tugas (*todo*), dan kategori dalam bentuk dokumen dan koleksi (*collections*). Mendukung sinkronisasi data secara *real-time* ke aplikasi klien.

4. **Authentication & Authorization Service:** Firebase Authentication.

5. **API:** Firebase SDK (FlutterFire). Mindly tidak menggunakan REST API tradisional yang di-hosting sendiri. Sebaliknya, aplikasi berkomunikasi langsung dengan layanan Firebase menggunakan SDK.

c. Koneksi dan Perangkat

Koneksi Internet: Diperlukan koneksi internet yang cepat untuk mengakses aplikasi Mindly.

3. Fitur-Fitur Sistem

3.1 Fitur Login dengan Google

3.1.1 Deskripsi dan Prioritas

Fitur ini memungkinkan pengguna untuk masuk ke aplikasi menggunakan akun Google mereka melalui OAuth 2.0. Prioritas TINGGI karena menjadi metode autentikasi utama yang paling cepat dan aman.

3.1.2 Stimulus / Urutan Respon

Tindakan Pengguna	Respon Sistem
-------------------	---------------

User menekan tombol "Sign in with Google"	Sistem membuka Google Sign-In dialog
User memilih akun Google	Sistem memverifikasi token OAuth dari Google
Google mengirim token autentikasi	Sistem membuat/mengambil data user dari Firebase Authentication, menyimpan session, dan mengarahkan ke Dashboard

3.1.3 Functional Requirements

- Sistem harus menggunakan Firebase Authentication dengan Google Sign-In
- Sistem harus menyimpan user profile (name, email, photo) dari google username atau dari perubahan yang dilakukan oleh user di *fieldtext*
- Jika login gagal, tampilkan pesan error yang jelas
- Sistem harus membuat user document baru di Firestore jika first-time login

3.2 Fitur Login Manual

3.2.1 Deskripsi dan Prioritas

Fitur ini memungkinkan pengguna login menggunakan email dan password yang telah didaftarkan. Prioritas TINGGI sebagai alternatif login tanpa akun Google.

3.2.2 Stimulus/Urutan Respon

Tindakan Pengguna	Respon Sistem
User memasukkan email dan password	Sistem memvalidasi format email dan password (minimal 8 karakter)
User menekan tombol "Login"	Sistem memverifikasi kredensial dengan Firebase Authentication
Jika valid	Sistem menyimpan session dan mengarahkan ke Dashboard
Jika tidak valid	Sistem menampilkan pesan "Email atau password salah"

3.2.3 Functional Requirements

- Email harus dalam format valid (mengandung @)

- Password minimal 8 karakter
- Sistem harus menampilkan loading indicator saat proses autentikasi

3.3 Fitur Register Akun Manual

3.3.1 Deskripsi dan Prioritas

Fitur untuk membuat akun baru menggunakan email dan password. Prioritas TINGGI untuk user yang tidak memiliki atau tidak ingin menggunakan akun Google.

3.3.2 Stimulus/Urutan Respon

Tindakan Pengguna	Respon Sistem
User mengisi form: Email, Password, Confirm Password	Sistem validasi real-time (format email, kekuatan password, password match)
User menekan "Create Account"	Sistem memeriksa apakah email sudah terdaftar
Email belum terdaftar	Daftarkan akunnya terlebih dahulu di akun di Firebase Auth, membuat user document di Firestore, mengirim verification email
Email sudah terdaftar	Sistem menampilkan "Email sudah terdaftar. Silakan login atau gunakan email lain"
User menyetujui Terms & Privacy Policy	Sistem mencatat consent timestamp

3.3.3 Functional Requirements

- Email harus unik (tidak boleh duplikat)
- Password minimal 8 karakter, harus mengandung: huruf besar, huruf kecil, angka, dan simbol
- Password dan Confirm Password harus identik
- Sistem harus mengirim email verifikasi setelah registrasi
- Wajib centang checkbox "Agree to Terms and Privacy Policy"

3.4 Fitur Forgot Password / Reset Password

3.4.1 Deskripsi dan Prioritas

Fitur untuk mereset password yang lupa. Prioritas **TINGGI** karena critical untuk user recovery.

3.4.2 Stimulus/Urutan Respon

Tindakan Pengguna	Respon Sistem
User klik "Forgot Password?" di halaman login	Sistem menampilkan form input email
User memasukkan email dan klik "Send Reset Link"	Sistem memeriksa apakah email terdaftar
Email terdaftar	Firebase mengirim email reset password, sistem menampilkan "Reset link telah dikirim ke [email]"
Email tidak terdaftar	Sistem tetap menampilkan pesan sukses (security measure untuk tidak expose registered email)
User klik link di email	Sistem membuka halaman reset password
User memasukkan password baru	Sistem memvalidasi dan mengupdate password

3.4.3 Functional Requirements

- Link reset password hanya bisa digunakan 1 kali
- Password baru harus memenuhi kriteria huruf kecil dan besar ada serta minimal 8 karakter
- Sistem harus logout dari semua devices setelah password berhasil direset

3.5 Fitur Dashboard Harian

3.5.1 Deskripsi dan Prioritas

Dashboard sebagai halaman utama setelah login yang menampilkan ringkasan aktivitas harian. Prioritas **TINGGI** karena menjadi pusat informasi utama user.

3.5.2 Stimulus/Urutan Respon

Tindakan Pengguna	Respon Sistem
User membuka aplikasi setelah login	Sistem mengambil data dari Firestore dan menampilkan: - Daily Progress (% tasks completed hari ini) - Needs Attention (tasks mendekati deadline) - Today's Events (events hari ini) - Today's Notes (notes yang dibuat hari ini) - Greeting message dengan nama user
User pull-to-refresh	Sistem refresh semua data dari server

3.5.3 Functional Requirements

- Data harus real-time sync dengan Firestore
- Daily Progress dihitung dari jumlah completed tasks / total tasks hari ini
- Needs Attention menampilkan maksimal 2 tasks dengan deadline terdekat dengan tanggal saat ini
- Today's Events diurutkan berdasarkan waktu mulai (*ascending*)
- Today's Notes menampilkan maksimal 3 notes terbaru hari ini
- Greeting disesuaikan dengan waktu: "Good Morning/Afternoon/Evening, [Name]"

3.6 Fitur Notifikasi

3.6.1 Deskripsi dan Prioritas

Sistem notifikasi untuk mengingatkan user tentang berbagai event penting. Prioritas **TINGGI** karena core functionality untuk productivity app.

3.6.2 Stimulus/Urutan Respon

Tindakan Pengguna	Respon Sistem
-------------------	---------------

User klik icon notifikasi di dashboard	Sistem menampilkan list notifikasi, diurutkan dari terbaru dan terdapat tanda jika pesan belum dibaca
User tap notifikasi tertentu	Sistem akan memberikan informasi atau reminder akan diberitahu di halaman tersebut

3.6.3 Functional Requirements

- Notifikasi harus tersimpan di Firestore untuk cross-device sync
- Support in-app notification dan system notification
- Badge count pada icon notifikasi menampilkan jumlah unread

3.7 Fitur Mark All Read Notification

3.7.1 Deskripsi dan Prioritas

Fitur untuk menandai semua notifikasi sebagai sudah dibaca sekaligus. Prioritas **SEDANG** untuk user convenience.

3.7.2 Stimulus/Urutan Respon

Tindakan Pengguna	Respon Sistem
User klik "Mark All as Read"	Sistem menampilkan konfirmasi "Mark all notifications as read?"
User konfirmasi "Yes"	Nanti kotak berwarna itu dengan opacity 20% akan menjadi backgorund putih dan bulatan biru akan hilang
User tidak tap notifikasinya	Sistem akan menganggap itu notifikasi belum dibaca oleh user maka nanti akan terdaapt bulatan warna biru
User menekan button "delete all"	Seluruh notifikasi akan terhapus di firebase dan di page notificationnya

3.7.3 Functional Requirements

- Batch update di Firestore untuk performa optimal

- Animation fade untuk notifikasi yang sudah read
- Badge count update secara real-time
- Tidak menghapus notifikasi, hanya mengubah status read

3.8 Fitur Profile dan Pengaturan Akun

3.8.1 Deskripsi dan Prioritas

Halaman untuk melihat dan mengelola informasi profil user. Prioritas **SEDANG** karena tidak sering diakses namun penting untuk personalisasi.

3.8.2 Stimulus/Urutan Respon

Tindakan Pengguna	Respon Sistem
User tap icon profile di dashboard	Sistem menampilkan halaman Account Profile dengan: - Profile picture - Nama - Email - Bio profile - Opsi Edit Info
User scroll ke bawah	Sistem menampilkan sections: - PREFERENCE (Notifications) - SUPPORT (Help & FAQ, About) - SECURITY (Reset Password) - DANGER ZONE (Logout, Delete Account)

3.8.3 Functional Requirements

- Profile picture diambil dari Google (jika login via Google) atau default avatar
- Email tidak bisa diubah setelah registrasi
- Setiap section memiliki icon dan navigation arrow
- Tampilan harus responsive dan scrollable

3.9 Fitur Edit Account Information

3.9.1 Deskripsi dan Prioritas

Fitur untuk mengedit informasi profil user. Prioritas **SEDANG** untuk personalisasi akun.

3.9.2 Stimulus/Urutan Respon

Tindakan Pengguna	Respon Sistem
User tap edit icon di profile	Sistem menampilkan form edit dengan field:- Profile Picture (upload/change) - Full Name (editable) - Email (read-only, greyed out)
User upload foto baru	Sistem compress image (max 2MB), upload ke Firebase Storage, update photoURL di user document
User ubah nama	Sistem validasi (min 2 karakter, max 50 karakter)
User tap "Save Changes"	Sistem update user document di Firestore, tampilkan snackbar "Profile updated successfully"

3.9.3 Functional Requirements

- Email tidak bisa diubah (read-only field)
- Nama wajib diisi, min 2 karakter, max 50 karakter
- Support foto dari camera atau gallery
- Image auto-crop ke 1:1 ratio (square)
- Maksimal ukuran foto 2MB, auto-compress jika lebih
- Format foto: JPG, PNG
- Cancel button untuk membatalkan perubahan

3.10 Fitur Notification Preference Settings

3.10.1 Deskripsi dan Prioritas

Pengaturan preferensi notifikasi untuk mengontrol jenis dan waktu notifikasi. Prioritas **SEDANG** untuk user experience customization.

3.10.2 Stimulus/Urutan Respon

Tindakan Pengguna	Respon Sistem
User tap "Notifications" di profile settings	Sistem menampilkan toggle switches untuk: - Daily Goal Achievement - Upcoming Deadlines (24h before) - Weekly Report (every Sunday) - Next Agenda (1h before event) - Overdue Tasks
User toggle ON/OFF preference tertentu	Sistem langsung save ke Firestore user preferences

3.10.3 Functional Requirements

- Perubahan tersimpan otomatis (auto-save) tanpa tombol save
- Toggle harus smooth dengan animation
- Snackbar confirmation setiap perubahan: "Preference saved"

3.11 Fitur Help & FAQ

3.11.1 Deskripsi dan Prioritas

Halaman bantuan berisi pertanyaan umum dan panduan penggunaan. Prioritas **RENDAH** namun penting untuk onboarding user baru.

3.11.2 Stimulus/Urutan Respon

Tindakan Pengguna	Respon Sistem
User tap "Help & FAQ"	Sistem menampilkan list FAQ
User tap kategori tertentu	Sistem expand accordion untuk menampilkan Q&A
User tidak menemukan jawaban	Sistem menampilkan "Contact Support" button

3.11.3 Functional Requirements

- FAQ disimpan sebagai static content atau fetched dari Firestore
- Accordion-style untuk menghemat space
- Search functionality untuk cari keyword di FAQ
- Deep link ke specific FAQ dari notifikasi help

- Contact Support button membuka email client dengan template

3.12 Fitur About Application

3.12.1 Deskripsi dan Prioritas

Halaman informasi tentang aplikasi. Prioritas **RENDAH** untuk transparansi dan legal compliance.

3.12.2 Stimulus/Urutan Respon

Tindakan Pengguna	Respon Sistem
User tap "About"	Sistem menampilkan info mengenai Mindly

3.12.3 Functional Requirements

- App version harus auto-fetch dari package info

3.13 Fitur Reset Password (dari Profile)

3.13.1 Deskripsi dan Prioritas

Fitur untuk mengubah password dari halaman profile (bukan forgot password). Prioritas **SEDANG** untuk security management.

3.13.2 Stimulus/Urutan Respon

Tindakan Pengguna	Respon Sistem
User tap "Reset Password" di profile	Sistem menampilkan form
User mengisi form dan tap "Update Password"	Sistem verifikasi current password dengan Firebase Auth
Current password benar	Sistem validasi new password (min 8 char, strong), update password, tampilkan "Password updated successfully", auto logout
Current password salah	Sistem tampilkan error "Current password is incorrect"

3.13.3 Functional Requirements

- Current password wajib diisi dan diverifikasi

- New password harus berbeda dari current password
- Password strength indicator (weak/medium/strong)
- Confirm password harus match dengan new password
- Setelah berhasil update, auto logout dan redirect ke login page
- Send email notification tentang password change

3.14 Fitur Delete Account

3.14.1 Deskripsi dan Prioritas

Fitur untuk menghapus akun permanen. Prioritas **RENDAH** namun wajib untuk GDPR compliance.

3.14.2 Stimulus/Urutan Respon

Tindakan Pengguna	Respon Sistem
User tap "Delete Account"	Sistem menampilkan warning dialog
User tap "Cancel"	Sistem menutup dialog
User tap "Delete"	Sistem meminta re-authentication (password untuk manual login atau Google re-sign)
User berhasil re-authenticate	Sistem menghapus akun dan re-direct ke splash screen dengan message "Account Deleted"

3.14.3 Functional Requirements

- Wajib re-authentication sebelum delete (security measure)
- Data deletion harus complete (cascade delete all user data)
- Proses deletion bisa memakan waktu, tampilkan progress indicator
- Tidak ada grace period (langsung permanent delete)
- Optional: kirim email konfirmasi deletion
- Log deletion action untuk audit (tanpa personal data)

3.15 Fitur Melihat Semua Catatan (All Notes)

3.15.1 Deskripsi dan Prioritas

Halaman utama yang menampilkan ringkasan semua catatan pengguna dalam bentuk *grid* atau *list*. **Prioritas: TINGGI**

3.15.2 Stimulus/Urutan Respon

Tindakan Pengguna	Respon Sistem
User tap "Notes" di bottom navigation	Sistem menampilkan All Notes, Categories, Favorite, Search Bar, Floating Action Button, Grid view toggle, Notes card dengan preview
User scroll	Sistem lazy load notes berikutnya (pagination 20 items)
User tap note card	Sistem navigasi ke detail/edit note

3.15.3 Functional Requirements

- **Sorting:** Catatan diurutkan berdasarkan updatedAt secara *descending* (terbaru di atas) sesuai *query* di `NoteService.getNotesStream()`.
- **Card Content:** Menampilkan judul (tebal), pratinjau isi (maksimal 6 baris, format *plain text* dari JSON Quill), dan tanggal (`formattedDate`).
- **Background Color:** Warna kartu diambil dari field color di database (integer Hex).
- **Empty State:** Jika kosong, tampilkan ikon dan teks "Tap + to create your first note".

3.16 Fitur Filter Notes by Category

3.16.1 Deskripsi dan Prioritas

Melihat notes berdasarkan kategori tertentu. Prioritas **TINGGI** untuk organisasi notes.

3.16.2 Stimulus/Urutan Respon

Tindakan Pengguna	Respon Sistem
User tap "Categories" tab	Sistem menampilkan list categories
User tap kategori tertentu	Sistem filter dan menampilkan hanya notes dari kategori tersebut
User tap "All Categories"	Sistem kembali menampilkan semua notes

3.16.3 Functional Requirements

- **Real-time Data:** Menggunakan StreamBuilder ganda (Categories & Notes) untuk memastikan jumlah catatan per kategori selalu akurat.
- **Visual:** Menampilkan nama kategori, jumlah catatan, dan ikon favorit.
- **Selection:** Mendukung seleksi kategori untuk aksi massal (Hapus/Favorit).

3.17 Fitur Filter Notes by Favorite

3.17.1 Deskripsi dan Prioritas

Menampilkan hanya notes yang difavoritkan. Prioritas **SEDANG** untuk quick access notes penting.

3.17.2 Stimulus/Urutan Respon

Tindakan Pengguna	Respon Sistem
User tap "Favorite" tab	Sistem query Firestore dengan filter <code>isFavorite == true</code> , menampilkan notes dan categories yang difavorite
Tidak ada favorite	Sistem tampilkan empty state: "No favorites yet. Long press any note to add to favorites"

3.17.3 Functional Requirements

- Menampilkan notes dan categories yang favorited
- Sections terpisah: "Favorite Notes" dan "Favorite Categories"
- Real-time update saat note/category difavorite atau unfavorite
- Sort by last modified

3.18 Fitur Search Notes by Keyword

3.18.1 Deskripsi dan Prioritas

Mencari notes berdasarkan keyword di title atau content. Prioritas **TINGGI** untuk produktivitas.

3.18.2 Stimulus/Urutan Respon

Tindakan Pengguna	Respon Sistem
User tap search bar	User tap search bar Sistem focus search bar, tampilkan keyboard
User mengetik keyword	Sistem search real-time (debounce

	300ms) di title dan content notes
Hasil ditemukan	Sistem highlight keyword, tampilkan matching notes
Tidak ada hasil	Sistem tampilkan "No results found for '[keyword]'"
User tap "X" atau back	Sistem clear search, kembali ke All Notes

3.18.3 Functional Requirements

- Case-insensitive search
- Search di title dan content body
- Highlight matching keyword di hasil

3.19 Fitur Menambah Catatan Baru

3.19.1 Deskripsi dan Prioritas

Membuat note baru. Prioritas **TINGGI** sebagai fungsi core.

3.19.2 Stimulus/Urutan Respon

Tindakan Pengguna	Respon Sistem
User tap FAB "+" > "Add Note"	Sistem membuka halaman editor note
User mengetik title dan content	Sistem auto-save draft setiap 3 detik ke local storage
User select category	Sistem tampilkan dialog dengan list categories, save selection
User tap favorite icon	Sistem toggle favorite status (heart icon berubah warna)
User tap "Save" atau back	Sistem save note ke Firestore dengan timestamp, tampilkan snackbar "Note saved", kembali ke All Notes

3.19.3 Functional Requirements

- **Auto-Save:** Mencegah kehilangan data dengan menyimpan otomatis saat user mengetik.
- **Default Data:** isFavorite default false, categoryId default string kosong, warna acak dari palet NoteService.

3.20 Fitur Mengedit Catatan

3.20.1 Deskripsi dan Prioritas

Mengubah note yang sudah ada. Prioritas **SEDANG**.

3.20.2 Stimulus/Urutan Respon

Tindakan Pengguna	Respon Sistem
User tap note card dari All Notes	Sistem membuka editor dengan data note yang sudah ada
User edit title/content	Sistem enable auto-save draft

3.20.3 Functional Requirements

- **Data Integrity:** Tidak boleh membuat dokumen baru (duplikat), harus *update* dokumen *existing*.
- **Format Preservation:** Struktur format teks (bold, list, dll) harus dipertahankan saat dimuat ulang.

3.21 Fitur Menghapus Catatan

3.21.1 Deskripsi dan Prioritas

Menghapus satu atau banyak catatan. **Prioritas: SEDANG**

3.21.2 Stimulus/Urutan Respon

Tindakan Pengguna	Respon Sistem
User long-press note card > "Delete" ATAU tap delete icon di editor	Sistem tampilkan dialog: "Delete this note?" dengan tombol Cancel dan Delete
User tap "Delete"	Sistem delete note dari Firestore, tampilkan snackbar "Note deleted" dengan UNDO button (5 detik)

3.21.3 Functional Requirements

- **Batch Operation:** Menggunakan WriteBatch Firestore untuk efisiensi dan konsistensi.
- **UI Update:** Mode seleksi otomatis tertutup (`_exitNoteSelectionMode`) setelah penghapusan berhasil.

3.22 Fitur Menambahkan Catatan ke Favorite

3.22.1 Deskripsi dan Prioritas

Toggle favorite status note. Prioritas **SEDANG**.

3.22.2 Stimulus/Urutan Respon

Tindakan Pengguna	Respon Sistem
User tap heart icon di note card ATAU di editor	Sistem toggle isFavorite field di Firestore (true ↔ false), update UI icon (filled heart vs outline)
Favorited	Snackbar "Added to favorites"
Unfavorited	Snackbar "Removed from favorites"

3.22.3 Functional Requirements

- Instant feedback: icon berubah langsung tanpa delay
- Animation: heart icon berubah warna
- Favorited notes accessible via "Favorite" tab

3.23 Fitur Long Press Notes untuk Quick Actions

3.23.1 Deskripsi dan Prioritas

Context menu untuk aksi cepat pada note. Prioritas **SEDANG** untuk UX *efficiency*.

3.23.2 Stimulus/Urutan Respon

Tindakan Pengguna	Respon Sistem
User long-press note card (hold 500ms)	Sistem tampilkan popup menu dengan options: - Favorite/Unfavorite - Move to Category - Delete
User tap salah satu action	Sistem execute action yang dipilih

3.23.3 Functional Requirements

- Haptic feedback saat long-press detected
- Menu muncul di posisi finger/touch point
- Dismiss menu jika tap outside

3.24 Fitur Menambah Kategori Notes

3.24.1 Deskripsi dan Prioritas

Membuat kategori baru. Prioritas **TINGGI**.

3.24.2 Stimulus/Urutan Respon

Tindakan Pengguna	Respon Sistem
User tap FAB "+" > "Add Category" ATAU "New Category"	Sistem tampilkan dialog dengan: - Input field: Category Name - Color picker (predefined colors) - Cancel dan OK button
User input nama dan pilih warna	Sistem validasi nama (min 2 char, max 30 char)
User tap "OK"	Sistem check duplikasi nama, jika unique maka create category di Firestore dengan auto-generated ID, snackbar "Category created"

3.24.3 Functional Requirements

- Nama kategori wajib diisi, min 2 max 30 karakter
- Nama tidak boleh duplikat (case-insensitive check)
- Default color jika user tidak pilih
- Auto-trim whitespace di nama

3.25 Fitur Mengedit Nama Kategori Notes

3.25.1 Deskripsi dan Prioritas

Mengubah nama atau warna kategori. Prioritas **SEDANG**.

3.25.2 Stimulus/Urutan Respon

Tindakan Pengguna	Respon Sistem
User tap edit icon di category item	Sistem tampilkan dialog edit (sama seperti add, tapi pre-filled dengan data existing)
User ubah nama dan/atau warna	Sistem validasi
User tap "Save"	Sistem update category di Firestore, snackbar "Category updated"

3.25.3 Functional Requirements

- Validasi sama dengan Add Category
- Check duplikasi nama (exclude nama sendiri)

- Update real-time di semua notes yang menggunakan category ini

3.26 Fitur Menghapus Kategori Notes

3.26.1 Deskripsi dan Prioritas

Menghapus kategori. Prioritas **SEDANG** dengan handling khusus untuk notes yang menggunakan kategori tersebut.

3.26.2 Stimulus/Urutan Respon

Tindakan Pengguna	Respon Sistem
User long-press category > "Delete" ATAU swipe-to-delete	Sistem tampilkan dialog konfirmasi: "Delete this category? [X] notes will be moved to Uncategorized"
User tap "Cancel"	Sistem menutup dialog
User tap "Delete"	Sistem: 1. Update semua notes dengan categoryId ini menjadi null (uncategorized) 2. Delete category dari Firestore 3. Snackbar "Category deleted. [X] notes moved to Uncategorized" dengan UNDO (5 detik)
User tap "Undo"	Sistem restore category dan reassign notes, snackbar "Category restored"

3.26.3 Functional Requirements

- Cannot delete jika category sedang digunakan KECUALI notes di-move ke uncategorized
- Show jumlah notes yang akan terpengaruh di confirmation dialog
- Soft delete dengan 5 detik grace period (undo functionality)
- Batch operation untuk update multiple notes efficiently
- Transaction safety untuk ensure data consistency

3.27 Fitur Menambahkan Kategori ke Favorite

3.27.1 Deskripsi dan Prioritas

Toggle favorite status category. Prioritas **SEDANG**.

3.27.2 Stimulus/Urutan Respon

Tindakan Pengguna	Respon Sistem
User tap heart icon pada category item	Sistem toggle isFavorite field di category document, update UI icon (filled vs outline heart)
Favorited	Snackbar "Category added to favorites", category pindah ke top of list
Unfavorited	Snackbar "Category removed from favorites", category kembali ke alphabetical order

3.27.3 Functional Requirements

- Instant UI update dengan animation
- Favorited categories appear di Favorite tab
- Favorited categories sorted to top di Categories list
- Animation: category item smoothly moves to new position

3.28 Fitur Multi-Select & Batch Actions (Aksi Massal)

3.28.1 Deskripsi dan Prioritas

Memungkinkan pengguna memilih banyak catatan sekaligus untuk dihapus, dipindahkan, atau difavoritkan. Prioritas **TINGGI** (berdasarkan SelectionActionBar dan `_isSelectionMode` di kode).

3.28.2 Stimulus/Urutan Respon

Tindakan Pengguna	Respon Sistem
Pengguna melakukan <i>long-press</i> pada salah satu kartu catatan.	- Masuk ke <code>_isSelectionMode</code>. - Bottom Navigation Bar digantikan oleh SelectionActionBar (Move, Favorite, Delete). - Tiap kartu catatan menampilkan indikator

	checkbox atau highlight.
Tap pada catatan lain untuk menambah seleksi.	Memasukkan catatan lain ke <code>isSelected</code>
Tap tombol "Move to".	Tampilkan <code>MoveToCategoriesDialog</code> (Bottom Sheet).
Pilih kategori tujuan.	Jalankan <code>_noteService.moveNotesBatch</code> , update <code>categoryId</code> untuk semua dokumen terpilih di Firestore, tutup mode seleksi, tampilkan <code>SnackBar</code> .

3.28.3 Functional Requirements

- Sistem harus mendukung `Set<String> selectedNoteIds` untuk melacak item.
- **Batch Delete:** Menggunakan `firestore.batch()` untuk menghapus banyak dokumen sekaligus demi efisiensi.
- **Batch Favorite:** Mengecek apakah semua item terpilih sudah favorit. Jika ya, *unfavorite* semua. Jika tidak, *favorite* semua.
- Tombol "Select All" pada `AppBar` untuk memilih semua catatan yang sedang tampil (filtered list).

3.29 Fitur Melihat All To-Do List

3.29.1 Deskripsi dan Prioritas

Halaman utama to-do list management. Prioritas **TINGGI** sebagai core productivity feature.

3.29.2 Stimulus/Urutan Respon

Tindakan Pengguna	Respon Sistem
User tap "To-Do" di bottom navigation	Sistem menampilkan: - Top tabs: "All", "Urgent", "Overdue" - Month summary card: "You have X things to do this month" dengan breakdown (All: X, Urgent: X, Overdue: X)

	- Mini calendar untuk current month (highlight dates dengan tasks) - FAB "Add Task" - Section: "Uncategorized Tasks" - Expandable folders untuk categorized tasks dengan progress bar
User tap checkbox pada task	Sistem toggle isCompleted , strikethrough text, update progress bar kategori

3.29.3 Functional Requirements

- Default view: All tab
- Tasks diurutkan: Incomplete (by deadline) > Completed (by completion date)
- Completed tasks dengan strikethrough styling
- Progress calculation per category: $(\text{completed} / \text{total}) \times 100\%$
- Uncategorized tasks ditampilkan di atas categories
- Smooth animation untuk expand/collapse category folders
- Real-time progress bar update
- Empty state jika tidak ada tasks

3.30 Fitur Melihat Urgent To-Do List

3.30.1 Deskripsi dan Prioritas

Filter untuk tasks yang urgent (deadline yang ≤ 12 jam). Prioritas **TINGGI** untuk time management.

3.30.2 Stimulus/Urutan Respon

Tindakan Pengguna	Respon Sistem
User tap "Urgent" tab	Sistem query tasks dengan filter: - <code>return !task.isCompleted && !diff.isNegative & diff.inHours <= 12;</code> - Tampilkan dalam list dengan:

	1. Task name
- Category badge
- Deadline countdown (e.g., "Due in 5 hours", "Due tomorrow") 2. Orange indicator untuk urgency level
Empty urgent	Sistem tampilkan: "No urgent tasks. You're doing great!"

3.30.3 Functional Requirements

- Urgent definition: deadline dalam 12 jam ke depan tidak boleh lewat dari deadline
- Sorted by deadline (ascending): paling dekat di atas
- Visual indicator: orange jika <12h
- Exclude completed tasks
- Auto-refresh saat app resume (update countdown)

3.31 Fitur Melihat Overdue To-Do List

3.31.1 Deskripsi dan Prioritas

Filter untuk tasks yang melewati deadline. Prioritas **TINGGI** untuk accountability.

3.31.2 Stimulus/Urutan Respon

Tindakan Pengguna	Respon Sistem
User tap "Overdue" tab	Sistem query tasks dengan filter: - <code>deadline < today AND isCompleted == false</code> - Tampilkan dalam list dengan: 1. Task name 2. Category badge 3. Overdue duration (e.g., "1 day overdue", "3 days overdue") 4. Red warning indicator
Empty overdue	Sistem tampilkan: "No overdue tasks."

	Keep up the good work! ✨
--	--------------------------

3.31.3 Functional Requirements

- Overdue definition: deadline sudah lewat dan task belum completed
- Sorted by overdue duration (descending): paling lama di atas
- Red visual indicator untuk semua overdue items
- Show overdue badge count di tab header

3.32 Fitur Menambah Task Baru

3.32.1 Deskripsi dan Prioritas

Membuat task baru. Prioritas **TINGGI** sebagai *core functionality*.

3.32.2 Stimulus/Urutan Respon

Tindakan Pengguna	Respon Sistem
User tap FAB "Add Task"	Sistem tampilkan dialog/bottom sheet dengan form: - Task name (required) - Description (optional) - Deadline: Day, Month, Year dropdowns - TIME (optional) - Category dropdown - Cancel dan Save button
User isi nama task (min) dan deadline	Sistem validate inputs
User tap "Save"	Sistem create task document di Firestore dengan: - userId - title - description - deadline (timestamp) - categoryId (nullable) - isCompleted: false - createdAt - Snackbar "Task created",

	auto-navigate ke All tab
--	--------------------------

3.32.3 Functional Requirements

- Task name required (min 3 char, max 100 char)
- Description optional (max 500 char)
- Deadline wajib diisi (default: tomorrow)
- Date picker untuk deadline selection (By default set 09.00)
- Category optional (default: uncategorized)
- Auto-calculate urgency status based on deadline
- Schedule notification reminder based on deadline

3.33 Fitur Mengedit Task

3.33.1 Deskripsi dan Prioritas

Mengubah task existing. Prioritas **TINGGI**.

3.33.2 Stimulus/Urutan Respon

Tindakan Pengguna	Respon Sistem
User tap task item	Sistem tampilkan detail task dengan mode read (bukan edit langsung)
User tap edit icon	Sistem switch ke edit mode, enable form editing
User ubah data dan tap "Save"	Sistem update Firestore, update updatedAt , snackbar "Task updated"

3.33.3 Functional Requirements

- Same validation dengan Add Task
- Show completion status (can toggle here)
- User hanya bisa mengedit Title dan Description
- Concurrent edit handling (jika multi-device sync)

3.34 Fitur Menghapus Task

3.34.1 Deskripsi dan Prioritas

Menghapus task. Prioritas **KECIL**.

3.34.2 Stimulus/Urutan Respon

Tindakan Pengguna	Respon Sistem
User tap delete button di task detail ATAU long-press > delete	Sistem tampilkan konfirmasi: "Delete this task?"
User tap "Delete"	Sistem delete task dari Firestore, update category progress bar, snackbar "Task deleted" dengan UNDO (5 detik)
User tap "Undo"	Sistem restore task

3.34.3 Functional Requirements

- Soft delete dengan 5 detik grace period
- Update category progress setelah delete
- Cancel scheduled notifications untuk task ini
- Batch delete option untuk multiple tasks (select mode)

3.35 Fitur Menandai Task sebagai Completed

3.35.1 Deskripsi dan Prioritas

Toggle completion status. Prioritas **TINGGI** sebagai main interaction.

3.35.2 Stimulus/Urutan Respon

Tindakan Pengguna	Respon Sistem
User tap checkbox di task item	Sistem toggle isCompleted
Task completed	Sistem trigger

3.35.3 Functional Requirements

- Instant UI feedback dengan animation
- Completed tasks pindah ke bottom of list (within same category)
- Update category progress bar real-time
- Daily goal achievement check dan notification

3.36 Fitur Menambah Kategori To-Do List

3.36.1 Deskripsi dan Prioritas

Membuat folder/kategori untuk grouping tasks. Prioritas **TINGGI**.

3.36.2 Stimulus/Urutan Respon

Tindakan Pengguna	Respon Sistem
User tap "Add new category" button di bottom list	Sistem tampilkan dialog "New Category"
User input nama kategori	Sistem validate (min 2, max 30 char, no duplicate)
User tap "Save"	Sistem create category di Firestore dengan type: "todos", snackbar "Category created"

3.36.3 Functional Requirements

- Category name required, unique (case-insensitive)
- Same category can be used untuk Notes dan To-Dos (shared categories) - ATAU separate categories by type
- New category immediately available di task creation/edit dropdown

3.37 Fitur Melihat Progress Kategori To-Do List

3.37.1 Deskripsi dan Prioritas

Visual progress indicator per kategori. Prioritas **TINGGI** untuk motivation & tracking.

3.37.2 Stimulus/Urutan Respon

Tindakan Pengguna	Respon Sistem
User melihat All To-Do list	Sistem display progress bar untuk each category folder
User tap category folder	Sistem expand untuk show tasks di kategori tersebut

3.37.3 Functional Requirements

- Progress calculation: $(\text{completed tasks} / \text{total tasks}) \times 100\%$
- Real-time update saat task di-check/uncheck
- Show 0% jika category baru tanpa tasks

3.38 Fitur Melihat Uncategorized Task

3.38.1 Deskripsi dan Prioritas

Section khusus untuk tasks tanpa kategori. Prioritas **SEDANG** untuk better

organization.

3.38.2 Stimulus/Urutan Respon

Tindakan Pengguna	Respon Sistem
User membuka All To-Do list	Sistem query tasks dengan <code>categoryId == null</code> , tampilkan di section "Uncategorized Tasks" di atas category folders
Empty uncategorized	Section hidden atau collapsed

3.38.3 Functional Requirements

- Uncategorized tasks always di top of list
- No progress bar untuk uncategorized (karena bukan "real" category)
- User bisa move uncategorized tasks ke category via edit

3.39 Fitur Mengatur Deadline Task

3.39.1 Deskripsi dan Prioritas

Set atau ubah deadline task. Prioritas **TINGGI** karena critical untuk time management.

3.39.2 Stimulus/Urutan Respon

Tindakan Pengguna	Respon Sistem
User tap deadline field saat add/edit task	Sistem tampilkan date picker
User select date	Sistem validate date (tidak boleh di masa lalu untuk new task)
User save task	Sistem calculate urgency status, schedule reminder notification

3.39.3 Functional Requirements

- Date picker intuitif (material design date picker atau custom)
- Cannot select past date untuk new task (hanya warn untuk edit task)
- Time optional (default: end of day 23:59)
- Show relative date di task item: "Today", "Tomorrow", "in 3 days", "May 15"
- Auto-reschedule notification jika deadline diubah

3.40 Fitur Mengatur Deskripsi Task

3.40.1 Deskripsi dan Prioritas

Menambahkan detail/notes untuk task. Prioritas **SEDANG** untuk context.

3.40.2 Stimulus/Urutan Respon

Tindakan Pengguna	Respon Sistem
User mengetik deskripsi	Auto-resize textarea sesuai content
User save task	Sistem simpan description ke Firestore
User tap "Description" field saat add/edit task	Sistem expand textarea, show keyboard

3.40.3 Functional Requirements

- Description optional, max 500 characters
- Multiline textarea dengan auto-expand
- Character counter: "120 / 500"
- Show description preview di task detail (collapsed, tap to expand)
- Support basic formatting: line breaks preserved

3.41 Fitur Melihat Calendar (Day/Month/Year View)

3.41.1 Deskripsi dan Prioritas

Memungkinkan pengguna melihat jadwal dalam berbagai perspektif waktu. Fitur ini menggunakan StreamBuilder untuk data real-time. Prioritas: **TINGGI**.

3.41.2 Stimulus/Urutan Respon

Tindakan Pengguna	Respon Sistem
User menekan menu "Calendar" di <i>bottom navigation</i> .	Sistem memuat CalendarMainPage dengan tampilan default (terakhir dipilih). Mengambil data kategori dan event dari Firestore.
User menekan ikon <i>toggle view</i> di <i>AppBar</i> .	Sistem mengubah mode tampilan secara siklus: Daily (Harian), Monthly (Bulanan), Yearly

	(Tahunan).
Mode Monthly: User menekan tanggal tertentu.	Sistem mengubah tampilan ke Daily View yang berfokus pada tanggal tersebut dan memuat <i>list</i> jadwal di bawahnya.
Mode Daily: User memilih tanggal di <i>Mini Calendar</i> horizontal.	Sistem memperbarui daftar <i>ScheduleCardWidget</i> di bagian bawah sesuai tanggal yang dipilih.
Mode Yearly: User menekan bulan tertentu.	Sistem masuk ke Monthly View pada bulan yang dipilih.

3.41.3 Functional Requirements

- **View Modes:**
 - **Daily:** Menampilkan *MiniCalendarWidget* (strip satu minggu) dan daftar detail event.
 - **Monthly:** Menampilkan grid kalender (*MonthlyViewWidget*) dengan indikator titik (*dots*) berwarna sesuai kategori event.
 - **Yearly:** Menampilkan ringkasan 12 bulan.
- **Data Fetching:** Menggunakan Stream dari Firestore (*getEventsForMonth*, *getEventsForDate*) untuk memastikan perubahan data langsung terlihat tanpa *refresh* manual.
- **Visual Cue:** Event pada kalender bulanan ditandai dengan titik warna yang sesuai dengan *hexColor* kategori.

3.42 Fitur Menambah Event di Calendar

3.42.1 Deskripsi dan Prioritas

Memungkinkan pengguna menjadwalkan acara baru, baik acara tunggal maupun acara berulang (*recurring*), lengkap dengan kategori, lokasi, catatan, dan pengingat.

Prioritas: **TINGGI**.

3.42.2 Stimulus/Urutan Respon

Tindakan Pengguna	Respon Sistem
-------------------	---------------

User tap FAB "+" atau tap date lalu "Add Event"	Sistem tampilkan form "Add New Event"
User isi form dan tap "Create Event"	Sistem validate

3.42.3 Functional Requirements

- **Validasi Waktu:** Sistem harus menolak penyimpanan jika $endTime < startTime$.
- **Default Category:** Jika user tidak memilih kategori, sistem harus menggunakan kategori pertama yang tersedia atau kategori *default* (warna biru).
- **Recurring Logic:**
 - Jika user memilih *Repeat*, sistem harus menggenerasi banyak dokumen event sekaligus (Batch Operation) berdasarkan pola (Harian: 30 hari, Mingguan: 12 minggu, Bulanan: 12 bulan).
 - Setiap event turunan harus memiliki `parentEventId` yang merujuk pada event pertama.
- **Notification Scheduling:** Sistem harus menghitung waktu notifikasi ($eventTime - reminderDuration$) dan menjadwalkannya di sistem operasi (Android/iOS).

3.43 Fitur Mengedit Event di Calendar

3.43.1 Deskripsi dan Prioritas

Memungkinkan pengguna mengubah detail acara yang sudah ada. Fitur ini memiliki kompleksitas tinggi karena harus menangani logika pemecahan rantai acara berulang (recurring event logic). Prioritas: **TINGGI**

3.43.2 Stimulus/Urutan Respon

Tindakan Pengguna	Respon Sistem
User tap event di calendar atau di list	Sistem tampilkan "Event Detail" sheet
User tap edit icon	Sistem open "Edit Event" form (same dengan Add Event, pre-filled)
User ubah data dan tap "Save	istem update Firestore, re-schedule

Changes"	notifications jika reminder/time berubah, snackbar "Event updated"
Recurring event	Sistem tanya: "Edit this event" atau "Edit all future events"

3.43.3 Functional Requirements

- **Data Pre-filling:** Form edit harus otomatis terisi dengan data *existing* (Judul, Catatan, Kategori, Waktu, dll).
- **Recurring Update Logic (Sesuai Kode EventService):**
 - **Mode Single:** Event diubah menjadi repeat: 'Does not repeat' dan dipisahkan dari *series*.
 - **Mode Following:** Menggunakan *query* where('startTime', '>', currentStartTime) untuk menghapus event masa depan lama, lalu *batch write* seri baru.
 - **Mode All:** Menggunakan *query* where('parentEventId', '==', seriesId) untuk update massal.
- **Notification Rescheduling:**
 - Saat event diupdate, notifikasi lama (berdasarkan notificationId lama) harus dibatalkan (cancelNotification).
 - Notifikasi baru dijadwalkan ulang sesuai waktu/pengingat baru.
- **Category Consistency:** Jika kategori diubah, warna kartu event pada kalender harus berubah secara *real-time* (via Stream).

3.44 Fitur Menghapus Event di Calendar

3.44.1 Deskripsi dan Prioritas

Menghapus event dari database. Logika penghapusan bergantung pada apakah event tersebut tunggal atau berulang. **Prioritas: SEDANG**

3.44.2 Stimulus/Urutan Respon

Tindakan Pengguna	Respon Sistem
User tap delete icon di Event Detail	Sistem tampilkan konfirmasi: "Delete this event?"
Single event	User tap "Delete" > Sistem delete dari Firestore, cancel notifications,

	snackbar "Event deleted" dengan UNDO (5s)
Recurring event	Sistem tampilkan options: - "Delete this event only" - "Delete this and all future events" - "Delete all events in series"

3.44.3 Functional Requirements

- **Batch Operations:** Penghapusan massal (untuk opsi *Following* dan *All*) harus menggunakan `firestore.batch()` untuk menjamin konsistensi data.
- **Notification Cancellation:** Saat event dihapus, sistem wajib memanggil `_notificationHelper.cancelNotification(id)` menggunakan ID unik event tersebut.
- **Parent Handling:** Jika event induk (*parent*) dihapus dalam mode *Following*, sistem harus menangani pemutusan rantai event masa lalu atau penghapusan total sesuai logika bisnis.

3.45 Fitur Mengatur Reminder untuk Event

3.45.1 Deskripsi dan Prioritas

Notification scheduling untuk events. Prioritas **TINGGI** untuk timely reminders.

3.45.2 Stimulus/Urutan Respon

Tindakan Pengguna	Respon Sistem
User tap "Reminder" field saat create/edit event	Sistem tampilkan dropdown/bottom sheet dengan options
User select reminder time	Sistem save selection, calculate notification trigger time
Event saved dengan reminder	Sistem schedule local notification atau FCM notification pada waktu reminder
Multiple reminders	User bisa add multiple reminders (e.g., 1 day before + 1 hour before)

3.45.3 Functional Requirements

- **Local Notifications:** Menggunakan `flutter_local_notifications` dengan

kalkulasi zona waktu (timezone) yang tepat.

- **Unique ID:** ID notifikasi dihasilkan dari eventId.hashCode agar bisa di-update atau dibatalkan spesifik per event.
- **Background Check:** Service OverdueCheckerService berjalan secara periodik untuk memvalidasi apakah ada reminder yang terlewat (berdasarkan data di koleksi event_reminders).

3.46 Fitur Mengatur Event Repeat/Recurring

3.46.1 Deskripsi dan Prioritas

Recurring events untuk regular activities. Prioritas **TINGGI** untuk repeated schedules.

3.46.2 Stimulus/Urutan Respon

Tindakan Pengguna	Respon Sistem
User tap "Repeat" field saat create/edit event	Sistem tampilkan options: - Does not repeat (default)
- Daily - Weekly (specify day of week) - Monthly (same date each month) - Yearly (same date annually)
User save recurring event	

3.46.3 Functional Requirements

- **Hardcoded Limits:** Sesuai kode, sistem membatasi jumlah pengulangan otomatis untuk performa (misal: Harian = 30 instansi, Mingguan = 12 instansi, Tahunan = 5 instansi).
- **Instance Independence:** Setiap hasil pengulangan adalah dokumen terpisah di Firestore, sehingga user bisa mengedit satu instansi (misal: mengubah jam rapat minggu depan) tanpa merusak jadwal minggu lainnya, kecuali user memilih "Update All".

3.47 Fitur Mengkategorikan Event

3.47.1 Deskripsi dan Prioritas

Mengelompokkan event dengan warna visual untuk memudahkan identifikasi.

Prioritas **TINGGI**.

3.47.2 Stimulus/Urutan Respon

Tindakan Pengguna	Respon Sistem
User tap "Select Category" saat create/edit event	Sistem tampilkan bottom sheet dengan: - List existing categories (dengan color badges) - Radio button selection - "+ Add new" option di bottom
User select category	Sistem assign categoryId ke event, event akan tampil dengan category color di calendar
User tap "+ Add new"	Sistem tampilkan "New Category" dialog (reuse dari Notes/Todos category creation)
User save event	Event akan color-coded di calendar sesuai category

3.47.3 Functional Requirements

- **Auto Color Generation:** Sistem menggunakan algoritma hash pada ID dokumen untuk menentukan warna dari palet warna yang telah ditentukan (`_categoryColors`).
- **Relasi Data:** Event hanya menyimpan categoryId. Warna dan nama diambil secara *join* (client-side mapping) menggunakan data dari service kategori.

3.48 Fitur Search Event by Keyword

3.48.1 Deskripsi dan Prioritas

Search functionality untuk find events. Prioritas **SEDANG** untuk large event databases.

3.48.2 Stimulus/Urutan Respon

Tindakan Pengguna	Respon Sistem
User tap search icon di calendar page	Sistem tampilkan search bar di top, focus input, show keyboard
User mengetik keyword	Sistem search real-time (debounce 300ms)
Hasil ditemukan	Sistem tampilkan matching events dalam list view
Tidak ada hasil	Sistem tampilkan "No events found for '[keyword]'" dengan suggestion: "Try different keywords"

3.48.3 Functional Requirements

- **Scope Pencarian:** Pencarian dilakukan pada *Title* dan *Description* secara *case-insensitive*.
- **Real-time Feedback:** Hasil pencarian diperbarui setiap kali user mengetik (menggunakan listener pada controller teks).
- **Visual:** Hasil pencarian tetap menampilkan warna kategori yang sesuai.

3.49 Fitur Log Out

3.49.1 Deskripsi dan Prioritas

Keluar dari aplikasi dengan aman. Prioritas **TINGGI** untuk security.

3.49.2 Stimulus/Urutan Respon

Tindakan Pengguna	Respon Sistem
User tap "Logout" di profile page	Sistem tampilkan konfirmasi dialog: "Are you sure you want to logout?"
User tap "Cancel"	Sistem close dialog, tetap logged in
User tap "Logout"	

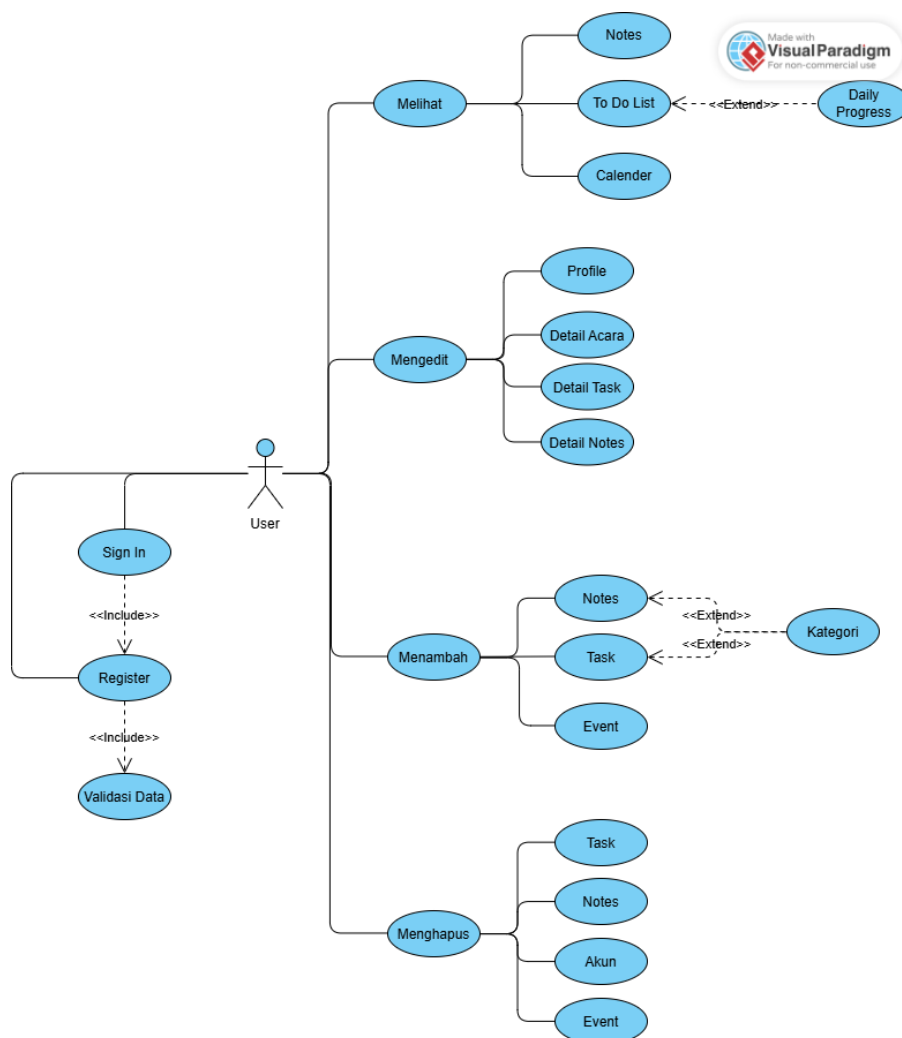
3.49.3 Functional Requirements

- Confirmation dialog untuk prevent accidental logout
- Session termination: Firebase Auth sign out
- Local data handling options:
 - Keep cached data untuk faster re-login (default)
 - Clear all local data (untuk shared devices)

- Cancel all scheduled local notifications
- Firestore listeners detached
- Analytics log: logout event
- Re-authentication required untuk login again
- Multi-device: logout hanya affect current device (session per device)

4. System Models & Diagram

4.1 Use Case Diagram



Use Case Diagram untuk aplikasi Mindly App menggambarkan interaksi antara pengguna dengan fitur-fitur utama yang tersedia dalam aplikasi. Aktor utama dalam aplikasi ini adalah Pengguna (*User*) yang menggunakan aplikasi untuk mencatat,

menjadwalkan, dan memantau aktivitas pribadi maupun pekerjaan harian.

Use Case Utama:

1. Registrasi Akun (Register Account)

Pengguna membuat akun baru untuk menyimpan data aktivitas dan catatan secara personal.

2. Login

Pengguna masuk ke aplikasi menggunakan akun terdaftar untuk mengakses seluruh fitur utama.

3. Membuat dan Mengelola Catatan (Add & Manage Notes)

- Pengguna dapat menulis, mengedit, atau menghapus catatan harian.
- Catatan digunakan untuk menyimpan ide, rencana, atau catatan kegiatan penting.

4. Membuat dan Mengelola To-Do List (Add & Manage To-Do List)

- Pengguna dapat menambahkan daftar tugas, menetapkan prioritas, dan menandai tugas yang telah selesai.
- Setiap tugas dapat dikategorikan berdasarkan tanggal dan status penyelesaian.

5. Menambahkan Acara ke Kalender (Add Calendar Event)

- Pengguna dapat menjadwalkan acara atau kegiatan penting dengan tanggal, waktu, dan deskripsi tertentu.
- Fitur ini juga terhubung dengan tampilan *Today's Event* di halaman Home.

6. Melihat Progress Harian (View Daily Progress)

- Sistem menampilkan *progress bar* yang menunjukkan persentase penyelesaian tugas dari To-Do List.
- Pengguna dapat memantau sejauh mana produktivitas harian mereka.

7. Menerima Notifikasi dan Peningat (Receive Notifications)

- Sistem mengirimkan pengingat otomatis untuk tugas yang mendekati deadline, acara kalender, atau pencapaian harian.

8. Sinkronisasi Data (Data Synchronization)

- Semua data catatan, kalender, dan tugas diperbarui secara *real-time* di halaman Home.

4.2 Activity Diagram

1. Login/Register

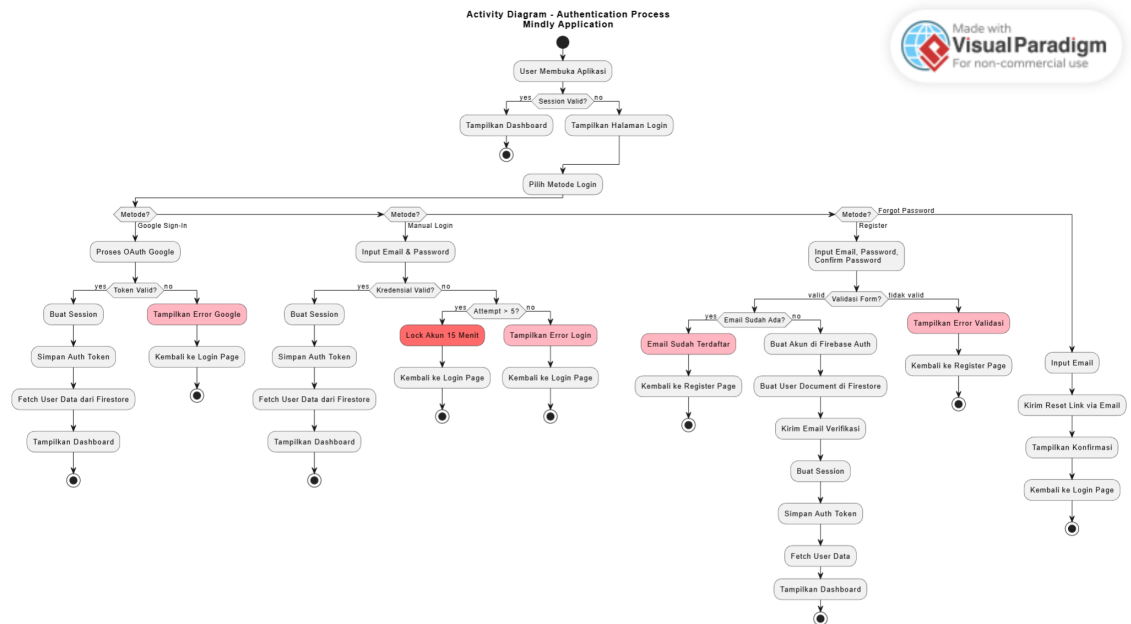


Diagram ini menggambarkan alur lengkap proses autentikasi pengguna:

- Start:** User membuka aplikasi
- Session Check:** Sistem mengecek apakah ada session aktif
- Login Methods:** User dapat memilih:
 - Google Sign-In:** OAuth 2.0 authentication
 - Manual Login:** Email & password
 - Register:** Membuat akun baru
 - Forgot Password:** Reset password via email
- Validation Process:** Setiap metode memiliki validasi tersendiri
- Security Features:**
 - Lock account setelah 5x gagal login
 - Email verification untuk registrasi
- Session Creation:** Setelah berhasil, sistem membuat session dan fetch user data
- End:** User diarahkan ke Dashboard

2. Dashboard Management

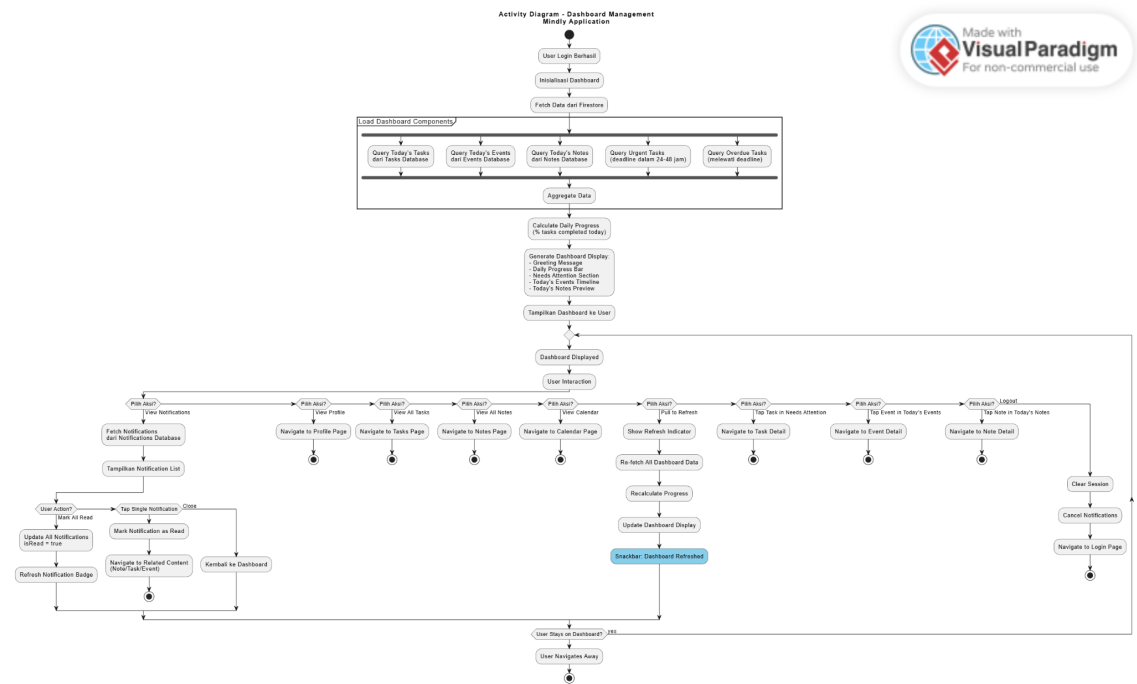


Diagram ini menggambarkan seluruh proses dan fitur utama yang terjadi setelah user berhasil login dan masuk ke halaman Dashboard aplikasi *Mindly*. Tujuannya adalah menampilkan status harian, data penting dari Firestore, serta menyediakan berbagai aksi cepat dan navigasi utama bagi pengguna.

1. Initialization & Data Loading

- **Inisialisasi Dashboard:** Dimulai setelah user login berhasil.
- **Data Fetching:** Sistem mengambil data dari Firestore, termasuk *tasks*, *events*, *notes*, dan *user information*.
- **Parallel Loading:** Menggunakan *fork-join* untuk memuat secara paralel:
 - a. Tasks hari ini
 - b. Events hari ini
 - c. Notes hari ini
 - d. Urgent tasks (deadline dalam 24–48 jam)

- **Category Management:** CRUD operations untuk categories
- **Quick Actions:** Long press untuk akses cepat
- c. **Validations:** Setiap input divalidasi sebelum disimpan
- d. **Favorites:** Toggle favorite untuk notes dan categories
- e. **Move To:** Pindahkan notes antar categories

4. To-Do-List Management



Diagram ini menggambarkan komprehensif manajemen To-Do List:

- **Summary Display:** Menampilkan summary bulanan, mini calendar, dan progress per category
- **Task Operations:**
 - **Add Task:** Membuat task dengan deadline, description, dan category
 - **Check/Uncheck:** Toggle completion dengan animation dan confetti untuk daily goal
 - **Edit/Delete:** Modifikasi atau hapus task dengan soft delete
- **Filters:**
 - **Urgent:** Tasks dengan deadline dalam 48 jam
 - **Overdue:** Tasks yang melewati deadline
- **Progress Tracking:** Real-time progress bar per category
- **Notifications:** Schedule reminder berdasarkan deadline
- **Daily Goal:** Deteksi achievement dan kirim congratulatory notification

5. Calender & Event Management

Diagram ini menggambarkan manajemen calendar dan events secara lengkap:

- a. **Calendar Display:** Menampilkan month view dengan event

indicators

b. **Event Creation:**

- Input comprehensive details (name, note, date, time, location)
- All-day event toggle
- Multiple reminders (max 5)
- Recurring events dengan custom rules

c. **View Modes:** Day (timeline 24h), Month (grid), Year (overview)

d. **Event Operations:**

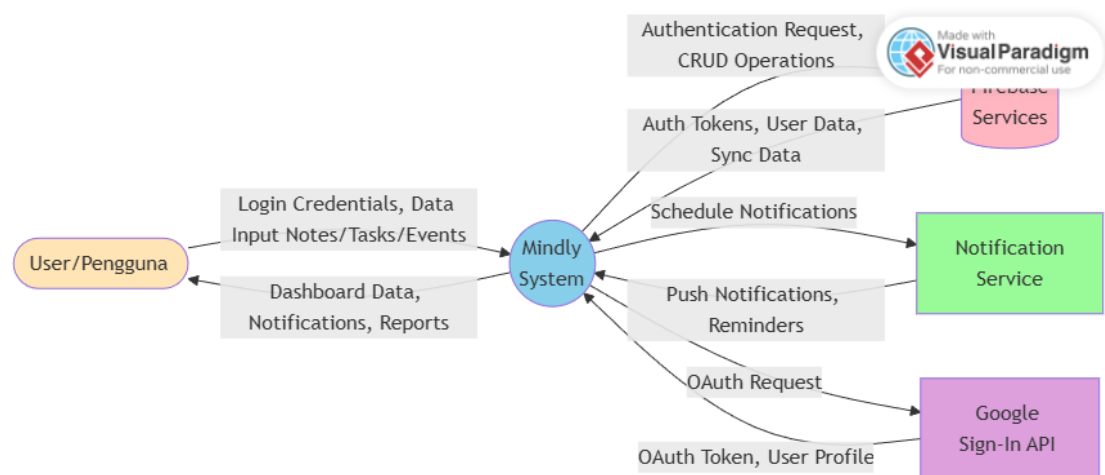
- **Edit:** Dengan opsi granular untuk recurring events
- **Delete:** Soft delete dengan undo, granular untuk recurring

e. **Search:** Real-time search di name, description, dan location

f. **Recurring Events:** Full support dengan edit/delete per instance atau series

4.3 Data Flow Diagram

a. **Level 0**

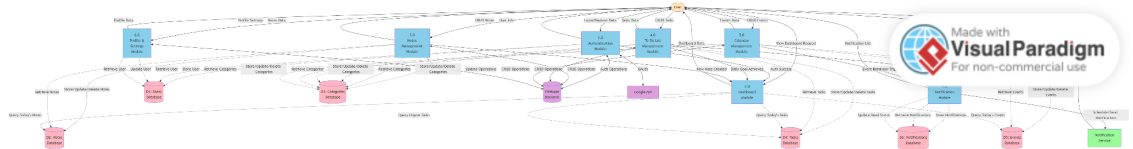


Context Diagram menggambarkan sistem Mindly sebagai satu entitas tunggal dan interaksinya dengan entitas eksternal:

- **User (Pengguna):** Memberikan input dan menerima output dari sistem
- **Firebase Services:** Backend untuk authentication, firestore, reset password email, dan email verification
- **Notification Service:** Sistem notifikasi lokal dan push notifikasi ke page notifikasi

- **Google Sign-In API:** OAuth authentication untuk login dengan Google

b. Level 1



DFD Level 1 memecah sistem Mindly menjadi 7 modul utama dengan 6 data store:

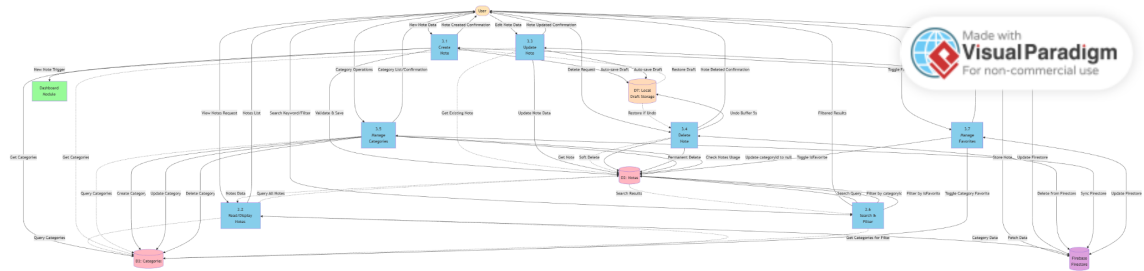
Modul-Modul:

- **Authentication Module (1.0):** Login, register, reset password
- **Dashboard Module (2.0):** Menampilkan summary dan progres harian
- **Notes Management Module (3.0):** CRUD notes dan categories
- **To-Do List Management Module (4.0):** CRUD tasks dan progress tracking
- **Calendar Management Module (5.0):** CRUD events dan scheduling
- **Profile & Settings Module (6.0):** Manage user profile dan preferences
- **Notification Module (7.0):** Handle semua notifikasi

Data Stores:

- D1: Users Database
- D2: Notes Database
- D3: Categories Database
- D4: Tasks Database
- D5: Events Database
- D6: Notifications Database

c. Level 2 - Notes Management

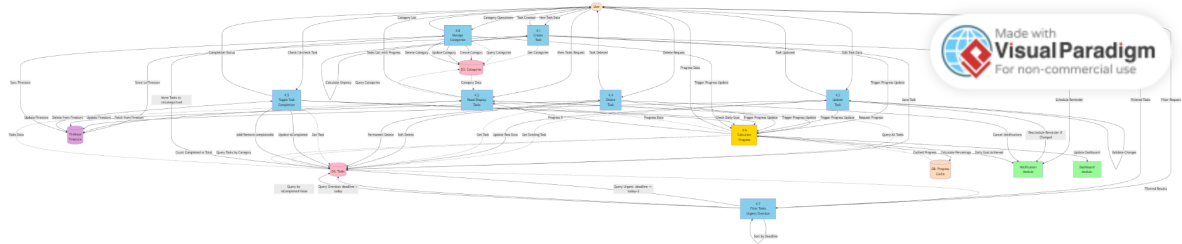


Detail dari Notes Management Module dipecah menjadi 7 sub-proses:

- **Create Note (3.1):**
 - Auto-save draft ke local storage setiap 3 detik
 - Validasi dan simpan ke Firestore
 - Trigger update dashboard
- **Read/Display Notes (3.2):**
 - Query notes dari database
 - Fetch categories untuk display
 - Real-time sync dengan Firestore
- **Update Note (3.3):**
 - Get existing note data
 - Auto-save draft during edit
 - Update ke Firestore
- **Delete Note (3.4):**
 - Soft delete dengan 5 detik undo buffer
 - Permanent delete setelah timeout
 - Sync dengan Firestore
- **Manage Categories (3.5):**
 - CRUD operations untuk categories
 - Check notes usage sebelum delete
 - Move notes ke uncategorized saat category deleted
- **Search & Filter (3.6):**
 - Real-time search di title dan content
 - Filter by category atau favorite
 - Debounce untuk optimize queries
- **Manage Favorites (3.7):**

- Toggle favorite status untuk notes dan categories
- Instant update ke Firestore

d. Level 2 - To Do List Management

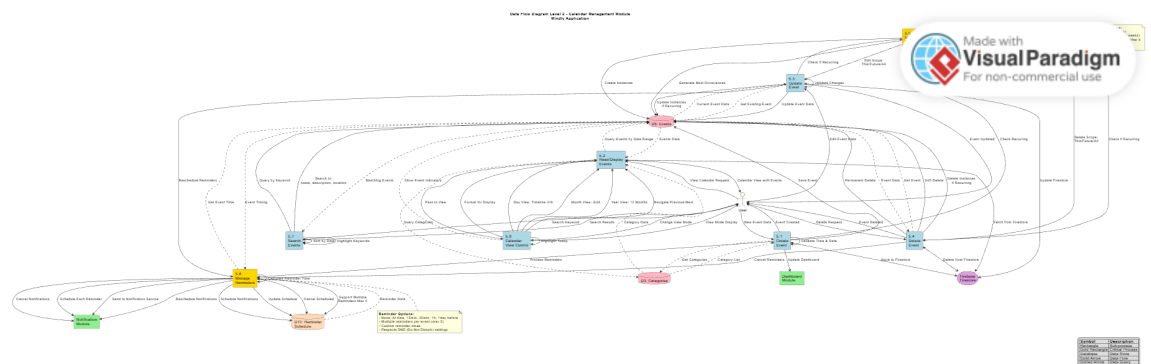


Detail dari To-Do List Management Module dengan 8 sub-proses:

- **Create Task (4.1):**
 1. Validasi deadline (tidak boleh masa lalu)
 2. Calculate urgency status
 3. Schedule notification reminder
 4. Trigger progress update
- **Read/Display Tasks (4.2):**
 1. Query tasks dengan categories
 2. Request progress calculation
 3. Real-time sync
- **Update Task (4.3):**
 1. Validate changes
 2. Reschedule reminder jika deadline berubah
 3. Update progress
- **Delete Task (4.4):**
 1. Soft delete dengan undo
 2. Cancel scheduled notifications
 3. Update progress
- **Toggle Completion (4.5):**
 1. Update completion status dan timestamp
 2. Check daily goal achievement
 3. Trigger notification jika goal tercapai
 4. Update dashboard

- **Calculate Progress (4.6):**
 1. Hitung percentage completed/total per category
 2. Cache untuk performance
 3. Check daily goal
- **Filter Tasks (4.7):**
 1. Filter urgent ($\text{deadline} \leq \text{today} + 2$)
 2. Filter overdue ($\text{deadline} < \text{today}$)
 3. Sort by deadline
- **Manage Categories (4.8):**
 1. CRUD categories
 2. Move tasks ke uncategorized saat delete

e. Level 2 - Calendar Management



Detail dari Calendar Management Module dengan 8 sub-proses:

- **Create Event (5.1):**
 1. Validasi date dan time
 2. Check recurring pattern
 3. Generate recurring instances
 4. Process reminders (multiple)
 5. Update dashboard
- **Read/Display Events (5.2):**
 1. Query events by date range
 2. Fetch dengan categories
 3. Pass ke calendar view untuk formatting

- **Update Event (5.3):**
 1. Check recurring status
 2. Edit scope: This/Future/All
 3. Update instances jika recurring
 4. Reschedule reminders
- **Delete Event (5.4):**
 1. Check recurring status
 2. Delete scope: This/Future/All
 3. Cancel all reminders
 4. Soft delete dengan undo
- **Manage Recurrence (5.5):**
 1. Validate recurrence pattern
 2. Generate instances (max 100)
 3. Handle edge cases (Feb 31)
 4. iCalendar RRULE compatible
- **Manage Reminders (5.6):**
 1. Support multiple reminders (max 5)
 2. Calculate reminder time
 3. Schedule notifications
 4. Reschedule on changes
- **Search Events (5.7):**
 1. Search di name, description, location
 2. Sort by date
 3. Highlight keywords
- **Calendar View Control (5.8):**
 1. Day view (timeline 24h)
 2. Month view (grid)
 3. Year view (12 months)
 4. Navigation control

5. Antarmuka Pengguna (UI)

Aplikasi Mindly App menggunakan antarmuka berbasis mobile dengan tampilan yang intuitif, interaktif, dan responsif untuk memudahkan pengguna dalam mencatat aktivitas harian, mengelola to-do list, serta menjadwalkan acara. Antarmuka dirancang menggunakan prinsip user-friendly design dan minimalist UI, dengan navigasi melalui touchscreen gestures seperti tap, swipe, dan long-press. Aplikasi mendukung sistem operasi Android, serta dapat menyesuaikan tampilan (*responsive*) pada berbagai ukuran layar perangkat.

Fitur utama yang ditampilkan di antarmuka pengguna meliputi:

- a. **Home Page:** Menampilkan ringkasan aktivitas harian seperti *Needs Attention*, *Today's Event*, dan *Today's Notes*.
- b. **Notes Page:** Untuk menambah, mengedit, atau menghapus catatan pribadi.
- c. **To-Do List Page:** Menampilkan daftar tugas, status penyelesaian, dan progress bar.
- d. **Calendar Page:** Untuk menambahkan dan melihat jadwal kegiatan.
- e. **Notifications Page:** Menampilkan pengingat tugas, acara, dan capaian harian.
- f. **Profile Page:** Menampilkan informasi akun pengguna dan pengaturan preferensi aplikasi.

5.1 Antarmuka Perangkat Keras

Aplikasi ini berjalan di atas perangkat mobile yang saling terhubung oleh jaringan internet. Dimana file aplikasi ditempatkan pada webhosting yang dioperasikan oleh administrator. Namun, agar performa optimal, perangkat disarankan memiliki spesifikasi minimum sebagai berikut:

- a. **RAM:** Minimal 4 GB
- b. **Penyimpanan internal:** Minimal 50 MB ruang kosong
- c. **Prosesor:** Quad-core 1.5 GHz atau lebih tinggi
- d. **Sistem Operasi:** Android, minimum API Level 21 (Android 5.0 Lollipop) hingga API Level 34 (Android 14)

5.2 Antarmuka Perangkat Lunak

Aplikasi Mindly App dikembangkan menggunakan teknologi cross-platform dengan fokus pada efisiensi dan integrasi layanan cloud. Spesifikasi perangkat lunak utama yang digunakan adalah sebagai berikut:

Komponen	Teknologi yang Digunakan	Fungsi
Bahasa Pemrograman	Dart	Bahasa utama pengembangan aplikasi
Framework	Flutter	Framework untuk membangun aplikasi Android & iOS
Database	Firebase Cloud Firestore	Penyimpanan dan sinkronisasi data secara real-time
Backend Service	Firebase Authentication, Firebase Cloud Messaging	Autentikasi pengguna dan pengiriman notifikasi
Storage (opsional)	Firebase Storage	Penyimpanan file pengguna (misalnya foto profil atau lampiran catatan)
API Format	REST API (JSON-based)	Komunikasi data dengan layanan cloud
Push Notification Service	Firebase Cloud Messaging (FCM)	Pengingat tugas, event, dan pencapaian harian

5.3 Antarmuka Komunikasi

Komunikasi dalam Mindly App dilakukan melalui layanan Firebase Cloud Platform dengan koneksi internet yang aman menggunakan protokol HTTPS dan WebSocket channel untuk data *real-time*. Semua komunikasi antara aplikasi dan server Firebase dienkripsi menggunakan SSL/TLS untuk memastikan keamanan data pengguna. Mindly App sepenuhnya bergantung pada infrastruktur cloud Firebase, sehingga tidak memerlukan pengaturan jaringan lokal atau server fisik.

6. Kebutuhan Non Fungsional

6.1 Kebutuhan Operasional

ID	Parameter	Kebutuhan
NF – 001	Availability	Aplikasi Mindly App harus memiliki tingkat ketersediaan minimal 95% selama waktu operasional dengan koneksi internet stabil. Karena sistem menggunakan Firebase Cloud Firestore (Spark Plan) yang di-hosting oleh Google, ketersediaan server mengikuti ketentuan layanan gratis Firebase.
NF – 002	Reliability	Sistem harus mampu menjaga keandalan penyimpanan data seperti catatan, daftar tugas, dan jadwal pengguna. Data disimpan secara real-time di Firebase Cloud Firestore, namun penyimpanan bergantung pada batasan kapasitas dan kuota layanan gratis Firebase.
NF – 003	Usability	Aplikasi harus memiliki antarmuka yang intuitif dan mudah dipahami oleh pengguna baru. Navigasi antar menu seperti <i>Notes</i> , <i>To-Do List</i> , <i>Calendar</i> , dan <i>Progress Bar</i> harus dapat diakses dalam maksimal dua langkah dari halaman utama. Desain antarmuka menggunakan prinsip Material Design agar tampil konsisten dan mudah dipelajari.
NF – 004	Perfomance	Aplikasi harus dapat memuat data utama dalam waktu kurang dari 4 detik pada koneksi internet normal (≥ 3 Mbps). Karena Firestore menggunakan sinkronisasi cloud, keterlambatan data kecil (< 1 detik) masih dapat ditoleransi pada saat penambahan atau penghapusan data.
NF – 005	Security	Komunikasi antara aplikasi dan server Firebase harus dilakukan menggunakan protokol HTTPS (SSL/TLS). Autentikasi pengguna dilakukan melalui Firebase Authentication (email dan password). Akses data Firestore dibatasi oleh Firebase Security Rules untuk mencegah akses tidak sah ke data pengguna lain.
NF – 006	Maintainability	Aplikasi dikembangkan menggunakan framework Flutter dengan struktur kode modular dan terdokumentasi dengan baik. Pengembang dapat dengan mudah melakukan pembaruan atau menambah fitur baru tanpa mempengaruhi fungsionalitas utama aplikasi.
NF – 007	Portability	Aplikasi harus dapat berjalan dengan baik pada perangkat Android (minimal

		versi 8.0/Oreo) dan iOS (minimal versi 13). Fungsionalitas dan tampilan utama harus tetap konsisten di berbagai ukuran layar smartphone dan tablet.
NF – 008	Scalability	Sistem dapat menangani peningkatan jumlah data pengguna dalam batas kuota Firestore versi gratis (maksimum 1 GiB penyimpanan dan 50.000 pembacaan per hari). Jika melebihi batas tersebut, sistem akan memberikan peringatan kepada pengguna untuk menghapus data lama atau melakukan upgrade ke plan berbayar terhadap trafik dan penyimpanan data secara otomatis.
NF – 009	Backup & Recover	Data pengguna akan otomatis tersimpan di Firebase Cloud Firestore dan tetap tersedia selama akun pengguna aktif. Jika pengguna login kembali di perangkat lain, semua data akan disinkronkan kembali dari cloud. Tidak disediakan fitur backup eksternal di luar Firestore karena keterbatasan versi gratis.

6.2 Kebutuhan Kinerja

Untuk aplikasi Mindly berjalan dengan cepat, efektif, dan responsif di berbagai perangkat mobile, bagian ini menjelaskan apa yang diperlukan. Aplikasi ini memiliki kapasitas dan waktu respons yang terbatas karena menggunakan basis data *real-time* Firebase Cloud Firestore sebagai layanan gratis.

Berikut adalah spesifikasi requirements kinerja yang dibutuhkan dalam aplikasi Mindly yaitu:

ID	Kebutuhan Kinerja	Deskripsi
PR-001	Kapasitas Transaksi	Sistem harus mampu memproses minimal 10 transaksi data per menit, mencakup penambahan, penghapusan, dan pembaruan catatan, to-do list, serta event kalender. Firebase Firestore akan menangani sinkronisasi data secara otomatis antar pengguna.
PR-002	Kapasitas Penyimpanan	Database Firestore pada plan gratis (Spark) mampu menampung hingga 1 GiB data total. Hal ini setara dengan kurang lebih 5.000–10.000 catatan teks ringan per pengguna. Pengguna akan diberikan peringatan jika kapasitas hampir penuh.

PR-003	Jumlah Pengguna Aktif	Sistem dirancang untuk mendukung minimal 5 pengguna aktif secara bersamaan tanpa penurunan performa yang signifikan. Karena Firestore menggunakan arsitektur real-time cloud, performa bergantung pada kestabilan koneksi pengguna.
PR-004	Sinkronisasi Data	Perubahan data pada perangkat pengguna harus tersinkronisasi ke Firestore dalam waktu maksimal 1 menit setelah koneksi internet tersedia kembali
PR-005	Waktu Respon Dasar	Waktu respon maksimal untuk operasi umum seperti menambah catatan, memperbarui to-do list, atau menandai tugas selesai adalah ≤ 10 detik.
PR-006	Waktu Respon Pencarian	Proses pencarian catatan atau event di dalam aplikasi harus selesai dalam waktu ≤ 20 detik untuk dataset hingga 1.000 item.
PR-007	Waktu Pemrosesan Kalender	Penambahan atau pembaruan event kalender harus ditampilkan di tampilan utama (<i>Home</i>) dalam waktu ≤ 8 detik setelah disimpan.
PR-008	Sinkronisasi Realtime	Perubahan data antar perangkat (misalnya dari ponsel ke tablet) harus terlihat dalam waktu ≤ 15 detik setelah penyimpanan di Firestore.
PR-009	Responsivitas UI	Aplikasi harus memberikan umpan balik visual seperti <i>loading indicators</i> , <i>snackbar</i> , atau <i>toast message</i> setiap kali pengguna menambah, memperbarui, atau menghapus data, agar pengguna mengetahui bahwa sistem sedang memproses permintaan user.

6.3 Kebutuhan Keamanan

Bagian ini memberikan penjelasan tentang persyaratan keamanan aplikasi Mindly App untuk menjaga kerahasiaan, integritas, dan ketersediaan data pengguna. Sistem keamanan dirancang berdasarkan prinsip CIA Triad (Kerahasiaan, Integritas, dan Ketersediaan), serta mengikuti standar praktik keamanan aplikasi mobile dan Peraturan Keamanan Firebase.

Berikut requirement keamanan yang dibutuhkan untuk memberikan keamanan dalam penggunaan aplikasi yang telah dirancang:

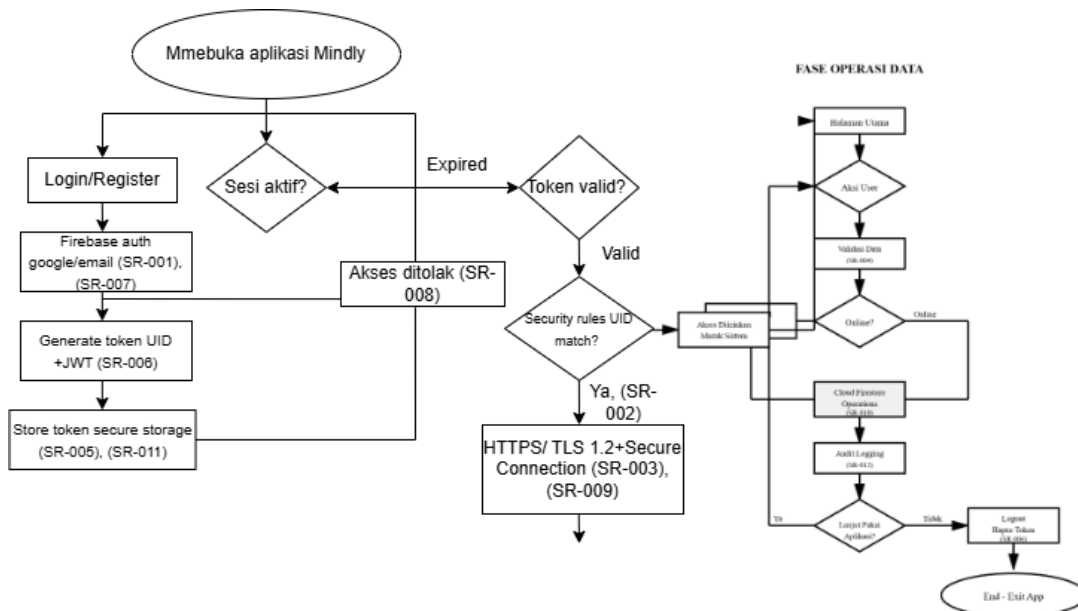
ID	Parameter	Deskripsi Kebutuhan Keamanan
KK-001	Autentikasi Pengguna	Aplikasi harus menggunakan Firebase Authentication jika Sign up menggunakan google untuk memverifikasi identitas pengguna. Setiap pengguna wajib memiliki akun unik untuk mengakses catatan, daftar tugas, dan jadwal pribadi.
KK-002	Otorisasi Akses	Akses data di Cloud Firestore harus dibatasi menggunakan Firebase Security Rules, di mana pengguna hanya dapat membaca, menulis, atau memperbarui data miliknya sendiri berdasarkan <i>User UID</i> yang dihasilkan oleh Firebase Authentication.
SR-003	Kerahasiaan Data	Seluruh pertukaran data antara aplikasi dan server Firebase harus dienkripsi menggunakan protokol HTTPS dengan SSL/TLS 1.2 atau lebih tinggi, untuk mencegah penyadapan atau <i>man-in-the-middle attack</i> .
KK-004	Integritas Data	Aplikasi harus memastikan bahwa data yang dikirim dan diterima dari Firebase tidak diubah tanpa izin. Setiap operasi <i>write</i> , <i>update</i> , dan <i>delete</i> hanya dapat dilakukan oleh pengguna yang sudah terautentikasi dan sesuai dengan <i>security rules</i> Firestore.
SR-005	Keamanan Penyimpanan Lokal	Data yang tersimpan sementara di perangkat (cache offline Firestore) harus dilindungi oleh sistem penyimpanan internal aplikasi Flutter, yang tidak dapat diakses langsung oleh pengguna lain atau aplikasi eksternal tanpa izin.
KK-006	Pengelolaan Sesi	Sesi login pengguna dikelola oleh Firebase Authentication. Token autentikasi bersifat sementara dan otomatis diperbarui untuk menjaga keamanan akses. Aplikasi harus menghapus token saat pengguna melakukan <i>logout</i> .
KK-007	Proteksi Password	Password pengguna tidak disimpan di perangkat ataupun server aplikasi. Firebase Authentication mengenkripsi semua password menggunakan bcrypt hashing dan menyimpannya secara aman di server Firebase.

KK-008	Perlindungan dari Akses Tidak Sah	Sistem harus menolak setiap permintaan akses ke Firestore yang tidak disertai dengan token autentikasi yang valid. Pengguna yang belum login tidak dapat melihat, menambah, atau mengubah data apa pun.
KK-009	Keamanan Jaringan	Semua komunikasi antar aplikasi dan server dilakukan melalui koneksi internet yang aman (HTTPS). Aplikasi tidak boleh mengirimkan data sensitif melalui protokol tidak terenkripsi seperti HTTP biasa.
KK-010	Keamanan Backup & Pemulihan	Data pengguna secara otomatis disimpan di cloud Firebase. Jika perangkat pengguna hilang atau aplikasi dihapus, data akan dipulihkan saat pengguna login kembali. Backup dilakukan secara otomatis oleh Firebase, dan hanya dapat diakses melalui akun terautentikasi.
KK-011	Perlindungan terhadap Akses Fisik	Aplikasi tidak menyimpan informasi sensitif (seperti password atau token autentikasi) dalam bentuk teks terbuka di penyimpanan perangkat. Semua data penting disimpan di area yang dilindungi oleh sistem keamanan perangkat.
KK-012	Audit dan Logging	Firebase menyediakan log aktivitas untuk setiap operasi penting seperti login, logout, dan akses database. Log ini digunakan untuk mendeteksi aktivitas mencurigakan atau pelanggaran keamanan, serta mendukung proses debugging dan audit sistem.

6.3.1 Diagram Alur Keamanan

Diagram alur keamanan ini menjelaskan langkah-langkah yang digunakan dalam aplikasi Mindly App untuk autentikasi pengguna, otorisasi akses, dan perlindungan data. Setiap langkah dirancang untuk memastikan bahwa hanya pengguna yang diakui dapat mengakses, mengubah, dan menyimpan data di Firebase Cloud Firestore.

Berikut terdapat gambar alur keamanan sistem didalam aplikasi ini:



6.4 Requirement Keselamatan

6.4.1 Pencegahan Error

Untuk menjaga stabilitas dan mencegah kesalahan pengguna, aplikasi Mindly App menerapkan mekanisme validasi dan perlindungan sebagai berikut:

No	Parameter	Deskripsi
1	Validasi Input Angka	Setiap kolom numerik (seperti progress persentase atau jumlah tugas selesai) divalidasi agar tidak menerima angka negatif atau melebihi batas maksimum (100%).
2	Konfirmasi Penghapusan Data	Sebelum pengguna menghapus catatan, daftar tugas, atau acara di kalender, sistem akan menampilkan pop-up konfirmasi berisi pesan “Apakah Anda yakin ingin menghapus data ini?” untuk mencegah kehilangan data tidak sengaja.
3	Validasi Format Email/Username	Selama proses registrasi dan login, sistem memverifikasi format email agar sesuai dengan standar (example@domain.com) sebelum dikirim ke Firebase Authentication. Username atau nama pengguna juga tidak boleh

		mengandung karakter ilegal.
--	--	-----------------------------

6.6.2 Pemulihan

Untuk menjaga keandalan dan keamanan data pengguna, Mindly App menerapkan beberapa lapisan pemulihan data sebagai berikut:

No	Fitur	Deskripsi
1	Ekspor Data Manual	Pengguna dapat mengekspor seluruh data catatan, to-do list, dan kalender ke format Excel (.xlsx) atau PDF untuk keperluan pencadangan pribadi.
2	Panduan Pemulihan Data	Aplikasi menyediakan panduan recovery sederhana di menu bantuan, menjelaskan cara memulihkan data dari hasil ekspor manual jika terjadi kehilangan lokal.
3	Backup Otomatis Berkala	Sistem secara otomatis melakukan backup mingguan ke Firebase Cloud Firestore, dengan enkripsi data untuk melindungi privasi pengguna. Backup ini disimpan di server cloud yang aman milik Google.

6.5 Requirement Lain

6.5.1 Kompabilitas

Mindly App dirancang agar dapat berjalan dengan baik di berbagai perangkat dan sistem operasi umum yang digunakan pengguna modern.

No	Komponen	Rincian
1	Platform Utama	Aplikasi berbasis mobile cross-platform menggunakan Flutter SDK, dapat dijalankan di perangkat Android dan iOS.

2	Browser Pendukung (untuk versi web opsional)	Chrome, Firefox, dan Edge versi terbaru (jika pengguna mengakses dari web).
3	Sistem Operasi Minimum	Android 8.0 (Oreo) dan iOS 13 atau lebih baru.
4	Koneksi Internet	Diperlukan untuk sinkronisasi ke Firebase. Aplikasi tetap dapat berjalan offline untuk mencatat data sementara.

6.5.2 Pemeliharaan (*Maintenance*)

Agar mudah dipelihara dan diperbarui di masa depan, sistem dirancang dengan dokumentasi dan struktur kode yang jelas.

No	Aspek	Deskripsi
1	Dokumentasi Sistem	Developer menyediakan dokumentasi teknis singkat dalam format PDF yang menjelaskan arsitektur aplikasi, konfigurasi Firebase, serta prosedur build dan deploy.
2	Pemeliharaan Kode	Struktur kode Flutter mengikuti standar <i>Model-View-Controller</i> (MVC) sederhana untuk memudahkan modifikasi.
3	Skalabilitas	Karena menggunakan Firebase Spark Plan (gratis), sistem tidak ditujukan untuk skalabilitas tinggi. Namun, arsitektur tetap dapat di-upgrade ke Blaze Plan bila dibutuhkan di masa depan.
4	Pembaruan Berkala	Pengembang dapat memperbarui fitur dan memperbaiki bug melalui update aplikasi

		secara periodik melalui Google Play Store dan App Store.
--	--	--

Additional Notes

Terdapat fitur tambahan yang sebelumnya tidak tertulis pada dokumen SRS, seperti yang disertakan pada tabel dibawah ini.

Nama Fitur	Rencana Awal	Hasil Akhir
Upload Foto Profile	Foto profile di set oleh pengembang secara seragam (<i>default</i>), dengan alasan keterbatasan pengetahuan mengenai akses media pada hp	User dapat mengatur foto profil pada bagian Edit Account Information dengan media gambar dari galeri pribadinya
Session Login	Ingin dibuat session login jadi kalo berhasil login dapat masuk ke aplikasi tanpa harus login akun lagi	Tidak diterapkan karena keterbatasan waktu dan sempat mencoba tetapi error
User bisa login tanpa melakukan verifikasi	User bisa masuk dengan gmail random walaupun tidak terdaftar di gmail	Diterapkan verifikasi email untuk memastikan apabila forgot password nanti bisa dilakukan dengan email akun sesuai yang telah terdaftar
Badge Notifikasi	Badge number notification pada icon notifikasi (lonceng) untuk memudahkan user dalam mengecek jumlahnya	Alternatif lain, yaitu didalam screen notifikasi ada jumlah pesan yang belum terbaca
Backup data dan memasukkan file ke notes	Penerapan backup data untuk hapus akun dan bisa mendukung fitur insert pdf dan file-file lainnya	Keterbatasan waktu dan cenderung lebih rumit karena harus menggunakan firebase storage sedangkan storage di firebase berbayar sehingga terbatas.