

APLIKASI MOBILE NOTEBOOK BERBASIS AI

Disusun guna memenuhi tugas mata kuliah
Pemrograman Perangkat Bergerak

Dosen Pengampu:
Arif Rifai Dwiyanto, ST., MTI



Oleh:
Kelompok 1

Dendi Yogia Pratama	NPM. 202310715051
Raihan Ariq Muzakki	NPM. 202310715297
Fazri Abdurahman	NPM. 202310715082

KELAS F5A5

**UNIVERSITAS BHAYANGKARA JAKARTA RAYA
FAKULTAS ILMU KOMPUTER
PROGRAM STUDI INFORMATIKA
BEKASI
2025**

KATA PENGANTAR

Puji syukur kami panjatkan kepada Tuhan Yang Maha Esa, karena atas limpahan rahmatnya penyusun dapat menyelesaikan laporan ini tepat waktu tanpa ada halangan yang berarti dan sesuai dengan harapan.

Ucapan terima kasih kami sampaikan kepada bapak Arif Rifai Dwiyanto, ST., MTI sebagai dosen pengampu mata kuliah Pemrograman Perangkat Bergerak yang telah membantu memberikan arahan dan pemahaman dalam penyusunan laporan ini.

Kami menyadari bahwa dalam penyusunan laporan ini masih banyak kekurangan karena keterbatasan kami. Maka dari itu penyusun sangat mengharapkan kritik dan saran untuk menyempurnakan laporan ini. Semoga apa yang ditulis dapat bermanfaat bagi semua pihak yang membutuhkan.

Bekasi, 25 Desember 2025

Kelompok 1

DAFTAR ISI

KATA PENGANTAR.....	ii
DAFTAR ISI.....	iii
DAFTAR TABEL	iv
DAFTAR LAMPIRAN	iv
BAB I PENDAHULUAN.....	6
1.1 Latar Belakang.....	6
1.2 Tujuan Aplikasi.....	7
1.3 Target Pengguna	8
1.4 Daftar Fitur Utama.....	8
BAB II ANALISIS DAN PERANCANGAN SISTEM	10
3.1 Analisis Pasar	10
3.1.1 Tren dan Kebutuhan Produktivitas dan AI (2025).....	10
3.1.2 Tren Pasar Note-taking Apps.....	10
3.1.3 Implikasi Tren Terhadap NotesAI	11
3.2 Analisis Kebutuhan Pengguna.....	11
3.2.1 Target Pengguna	11
3.2.2 Kebutuhan Fungsional	11
3.3 Fitur Utama Aplikasi.....	12
3.3.1 Note Management System	12
3.3.2 Category Organization	12
3.3.3 Archive & Trash.....	13
3.3.4 Rich Text Editor (WYSIWYG)	13
3.3.5 AI Assistant.....	13
3.4 Tampilan Aplikasi (Wireframe)	15
3.5 Teknologi dan Alat yang Digunakan.....	16
3.6 Arsitektur Aplikasi	17
3.7 Resource yang Digunakan.....	17
3.8 Pengujian Fungsional	17
BAB III PANDUAN PENGGUNA.....	21
3.1 Cara Install/Akses.....	21

3.1.1	Prasyarat Perangkat.....	21
3.1.2	Cara Menjalankan Aplikasi.....	21
3.1.3	Izin Aplikasi (Permissions).....	21
3.2	Mockup + Userflow (Alur Penggunaan)	21
3.2.1	Alur Utama: Membuat Kategori dan Catatan	21
3.2.2	Alur Editor: Menulis Catatan dengan Rich Text (WYSIWYG)	22
3.2.3	Alur Prioritas: Pin Catatan dan Pin Kategori	23
3.2.4	Alur Keamanan Data.....	24
3.2.5	Alur Pencarian: Search Realtime + Filter Kategori	25
3.2.6	Alur AI Assistant: Chat Berbasis Konteks Catatan.....	26
3.2.7	Alur Upload Dokumen: PDF/TXT/DOCX → Ringkasan AI	27
3.3	Batasan.....	29
BAB VI	PENUTUP	30
4.1	Kesimpulan	30
4.2	Saran Pengembangan Selanjutnya	30
4.2.1	Peningkatan Arsitektur dan Skalabilitas Data.....	30
4.2.2	Keamanan dan Privasi (High Priority).....	31
4.2.3	Peningkatan UX dan Fitur Produktivitas	32
4.2.4	Peningkatan Kualitas melalui Pengujian dan Otomasi	32
DAFTAR PUSTAKA		33

DAFTAR TABEL

Tabel 1 Test Cases Summary.....	19
---------------------------------	----

DAFTAR LAMPIRAN

Gambar 1 Wireframe Aplikasi.....	15
Gambar 2 TC-01 &TC-02.....	20
Gambar 3 TC-03	20
Gambar 4 TC-04	20
Gambar 5 TC-05	20
Gambar 6 TC-06	20
Gambar 7 Logo Aplikasi.....	22
Gambar 8 Main Screen	22

Gambar 9 Category Dialog	22
Gambar 10 Notes Screen	22
Gambar 11 Notes Dialog	22
Gambar 12 Notes Screen 2	23
Gambar 13 Editable Note Screen.....	23
Gambar 14 Kategori Sebelum di Pin	24
Gambar 15 Kategori Sesudah di Pin.....	24
Gambar 16 Catatan Sebelum di Pin	24
Gambar 17 Catatan Sesudah di Pin.....	24
Gambar 18 Hapus Kategori	25
Gambar 19 Setelah Kategori Terhapus	25
Gambar 20 Drawer Menu	25
Gambar 21 Halaman Sampah	25
Gambar 22 Setelah di Pulihkan.....	25
Gambar 23 Catatan di Kategori Kuliah.....	26
Gambar 24 Catatan di Kategori Magang	26
Gambar 25 Pencarian Catatan dengan Keyword Cocok.....	26
Gambar 26 Pencarian Catatan dengan Keyword Asal	26
Gambar 27 Catatan Machine Learning	27
Gambar 28 AI Helper Screen.....	27
Gambar 29 AI Response 1	27
Gambar 30 AI Response 2	27
Gambar 31 History Chat	27
Gambar 32 File PDF yang di Upload.....	28
Gambar 33 Proses File Parser	28
Gambar 34 Proses Summary.....	28
Gambar 35 Hasil Summary.....	28
Gambar 36 History Chat Summary.....	28

BAB I

PENDAHULUAN

1.1 Latar Belakang

Perkembangan penggunaan perangkat mobile menjadikan aktivitas pencatatan (notes) sebagai bagian penting dari produktivitas harian, baik untuk kebutuhan akademik maupun profesional. Namun, dalam praktiknya, catatan sering tersebar dalam banyak entri tanpa struktur yang jelas, sehingga menimbulkan beberapa masalah utama:

1. Catatan tidak terorganisir (bercampur antara catatan penting dan tidak penting).
2. Catatan sulit dicari kembali ketika jumlahnya meningkat.
3. Catatan kurang interaktif, terutama untuk kebutuhan merangkum dokumen atau mengekstrak poin penting secara cepat.

Pada tren 2025, kebutuhan aplikasi produktivitas semakin mengarah pada integrasi kecerdasan buatan (AI) untuk mempercepat pekerjaan yang repetitif, seperti pencarian informasi, peringkasan, dan penyusunan insight. Hal ini didukung oleh temuan Eurobarometer (Februari 2025) bahwa 70% responden percaya AI meningkatkan produktivitas di tempat kerja. European Union Selain itu, laporan McKinsey tentang “State of AI” menekankan bahwa penggunaan AI semakin meluas, termasuk agentic AI, walaupun banyak organisasi masih berupaya mengubah implementasi AI dari sekadar “pilot” menjadi dampak yang konsisten.

Di sisi lain, pasar aplikasi note-taking juga menunjukkan pertumbuhan dan mendorong inovasi fitur, termasuk penguatan pengalaman pengguna dan integrasi AI. Laporan pasar note-taking apps dari penyedia riset industri menggambarkan adanya tren pertumbuhan dan kompetisi fitur yang semakin intensif, sehingga aplikasi catatan perlu memiliki diferensiasi yang jelas.

Berdasarkan kondisi tersebut, kelompok kami mengembangkan NotesAI, yaitu aplikasi pencatatan di Android yang mengutamakan pengelolaan catatan terstruktur dan dukungan AI untuk membantu pengguna mengurangi waktu pencarian serta memperoleh ringkasan/insight dengan lebih cepat.

1.2 Tujuan Aplikasi

Aplikasi NotesAI bertujuan untuk menyelesaikan masalah pengelolaan catatan sehari-hari yang sering tidak terorganisir, sulit dicari, dan kurang interaktif, melalui dua pendekatan utama:

1. Peningkatan manajemen catatan secara terstruktur
 - Menyediakan sistem kategori visual, pin prioritas, arsip, serta trash dengan soft delete.
 - Menyediakan pencarian dan filter agar catatan cepat ditemukan.
2. Peningkatan produktivitas melalui integrasi AI
 - Menyediakan AI Assistant berbasis Gemini 2.5 Flash untuk respons yang kontekstual dan efisien.
 - Dokumentasi resmi menyebut Gemini 2.5 Flash sebagai model unggul “price-performance”, cocok untuk low-latency dan high-volume tasks yang membutuhkan “thinking”.

Dengan demikian, NotesAI diharapkan dapat:

- Mengurangi waktu yang dibutuhkan untuk menemukan informasi penting pada catatan.
- Meningkatkan kualitas pemanfaatan catatan melalui ringkasan dan analisis kontekstual.
- Mengurangi risiko kehilangan catatan melalui mekanisme arsip dan soft delete.

1.3 Target Pengguna

Target utama NotesAI adalah pengguna Android usia 18–40 tahun dengan kebutuhan produktivitas harian, meliputi:

1. Mahasiswa
 - Membuat catatan kuliah, rangkuman materi, serta menyimpan ringkasan dokumen (PDF/DOCX/TXT).
2. Profesional kantor
 - Mencatat hasil rapat, daftar tugas, dan informasi kerja yang perlu diprioritaskan (pin) serta cepat dicari.
3. Pengguna umum
 - Menyimpan ide, catatan pribadi, daftar aktivitas, atau informasi penting yang perlu tersusun rapi.

Aplikasi ini menekankan penyimpanan lokal untuk catatan (local-first) serta integrasi AI untuk analisis/summary ketika dibutuhkan.

1.4 Daftar Fitur Utama

1. Note Management System (Primary)
 - Create, Read, Update, Delete (CRUD) catatan
 - Kategori untuk organisasi catatan
 - Pin notes untuk akses cepat
 - Archive & Trash dengan **soft delete**
 - Full-screen rich text editor (WYSIWYG)
 - Search & filter realtime
2. AI Assistant (Primary)
 - Chat dengan AI menggunakan **Gemini 2.5 Flash**
 - Respons kontekstual berbasis catatan pengguna
 - Penyimpanan chat history secara persisten
 - Upload dokumen (PDF/TXT/DOCX) dan auto-summary
 - Dukungan Markdown pada respons AI
 - Copy output plain text maupun formatted text

3. Category Organization (Core)

- Manajemen kategori dengan preset gradien warna
- Pin kategori favorit
- Sorting cerdas: pinned → timestamp terbaru
- Statistik dan filtering berbasis kategori
- Layout grid responsif

4. Rich Text Editor (Core)

- WYSIWYG: bold/italic/underline, heading (H1–H3), bullet list
- Toolbar draggable dengan indikator status aktif
- Undo/redo
- Kompatibel import/export Markdown
- Autosave lifecycle-aware (dengan debounce)

5. Data Persistence (Essential)

- Local storage menggunakan **DataStore Preferences**
- Autosave dengan debounce (500ms)
- Penyimpanan preferensi tema (dark/light)
- Penyimpanan chat history dengan serialisasi
- Reactive loading berbasis Flow
- Error handling pada operasi I/O

BAB II

ANALISIS DAN PERANCANGAN SISTEM

3.1 Analisis Pasar

3.1.1 Tren dan Kebutuhan Produktivitas dan AI (2025)

1. AI semakin diasosiasikan dengan peningkatan produktivitas kerja.

Survei Eurobarometer “Artificial Intelligence and the future of work” melaporkan bahwa **70% responden percaya AI meningkatkan produktivitas** di tempat kerja. Ini relevan dengan positioning NotesAI, AI dipakai untuk meringkas/menganalisis konteks catatan agar pengguna menghemat waktu.

2. Adopsi AI meluas, tetapi tantangan utama adalah membawa AI dari “sekadar fitur” menjadi dampak yang terukur.

Di platform informasi internet McKinsey “The state of AI in 2025: Agents, innovation, and transformation” menekankan adopsi AI makin luas, termasuk meningkatnya agentic AI, namun banyak organisasi masih berjuang mentransformasikan pilot menjadi impact yang konsisten. Ini bisa Anda jadikan landasan, NotesAI menargetkan use case yang jelas dan terukur (pencarian cepat, ringkasan dokumen, insight dari catatan).

3. Model AI “Flash” diposisikan untuk beban kerja high-volume dan low-latency.

Dokumentasi resmi menyebut **Gemini 2.5 Flash** sebagai model unggulan “price-performance” yang cocok untuk low-latency dan high volume tasks yang membutuhkan reasoning (“thinking”). Ini cocok dengan kebutuhan aplikasi mobile dengan respons cepat dan biaya efisien.

3.1.2 Tren Pasar Note-taking Apps

Pasar aplikasi pencatatan dilaporkan terus bertumbuh, didorong kebutuhan organisasi informasi serta integrasi fitur pintar/AI. Sebagai referensi industri, laporan pasar note-taking apps 2025 menyajikan segmentasi pengguna (komersial vs privat) dan proyeksi pertumbuhan.

3.1.3 Implikasi Tren Terhadap NotesAI

Berdasarkan tren di atas, NotesAI menargetkan gap utama:

- Catatan banyak tetapi sulit ditemukan → search & filter real-time + kategori + pin.
- Catatan penting bercampur dengan catatan biasa → pin notes dan pin kategori serta sorting prioritas.
- Pengguna butuh ringkasan/insight cepat dari catatan/dokumen → AI Assistant (Gemini 2.5 Flash) + upload dokumen + auto-summary.
- Risiko salah hapus → Trash + soft delete (restore sebelum hapus permanen).

3.2 Analisis Kebutuhan Pengguna

3.2.1 Target Pengguna

1. **Mahasiswa (18–25):** mencatat materi, tugas, ide, ringkasan dokumen kuliah (PDF/DOCX).
2. **Pekerja kantor (22–40):** meeting notes, to-do, catatan kerja dengan kebutuhan prioritas dan pencarian cepat.
3. **Pengguna umum (18–40):** catatan pribadi, daftar belanja, rencana harian, jurnal ringkas.

3.2.2 Kebutuhan Fungsional

- Kinerja: **respons UI cepat (Compose + state management); pencarian efisien untuk data lokal.**
- Reliabilitas: **penyimpanan lokal dan pemulihan data (restore dari trash).**
- Privasi: **data catatan disimpan local, namun saat AI digunakan, konten yang relevan dikirim ke layanan AI.**
- Kompatibilitas: **Android minimal API 24 (berdasarkan konfigurasi proyek).**

3.3 Fitur Utama Aplikasi

3.3.1 Note Management System

Tujuan: membuat, membaca, mengubah, menghapus catatan serta mengelolanya per kategori.

Alur utama:

1. Pengguna memilih kategori → membuat note.
2. Note tampil di daftar (grid) dan bisa dibuka ke tampilan full-screen editor.
3. Note dapat dipin untuk prioritas, dipindah ke arsip, atau masuk trash (soft delete).

Struktur data inti (model):

- Note: id, categoryId, title, description (preview), content (konten penuh), timestamp, isPinned, isArchived, isDeleted.
- Sistem ini memungkinkan: daftar note aktif (bukan archived, bukan deleted), daftar archived, dan daftar trash.

Aturan sorting yang diimplementasikan:

- Daftar note disorting: pinned terlebih dahulu, lalu timestamp terbaru.

Edge case yang perlu dicatat di laporan:

- Pencarian saat kategori dipilih (filter + search query).
- Note kosong/hasil pencarian kosong → ditangani dengan Empty State.

3.3.2 Category Organization

Tujuan: pengelompokan visual dengan prioritas kategori favorit.

Struktur data inti:

- Category: id, name, gradientStart, gradientEnd, timestamp, isDeleted, isPinned.

Aturan sorting kategori (diimplementasikan):

- pinned categories → terbaru (timestamp descending) pada daftar kategori aktif.

Edge case:

- Kategori dihapus masuk trash (soft delete) dan bisa direstore/hapus permanen.

3.3.3 Archive & Trash

Tujuan: mencegah kehilangan data akibat salah hapus.

Perilaku:

- Archive: note ditandai isArchived = true (tidak muncul di daftar utama).
- Trash: note/kategori ditandai isDeleted = true.
 - Di Trash tersedia Restore (set isDeleted = false) dan Delete Permanent (menghapus permanen dari list/penyimpanan).

Catatan penting untuk laporan:

- Soft delete adalah praktik umum untuk usability; Anda dapat tekankan bahwa sistem ini mengurangi risiko kehilangan informasi penting.

3.3.4 Rich Text Editor (WYSIWYG)

Tujuan: pengalaman mengetik dan format teks tanpa markdown mentah.

Implementasi ringkas:

- Menggunakan pendekatan AnnotatedString dan state editor khusus (RichEditorState) untuk mempertahankan span (bold, italic, underline, undo, redo) saat pengguna mengedit.
- Toolbar format bersifat draggable dan menampilkan state aktif.

3.3.5 AI Assistant

Tujuan: analisis dan respons AI berbasis konteks catatan pengguna dan dokumen.

Implementasi ringkas (yang tampak di kode):

- Model dipanggil dengan modelName = "gemini-2.5-flash" menggunakan Google AI client library. Model ini dideskripsikan sebagai unggul “price-performance” untuk low-latency dan high-volume.
- Mendukung penyimpanan chat history secara persisten.

Upload dokumen & parsing:

- File picker menggunakan Activity Result API (GetContent).
- Parsing file:
 - PDF: memakai PdfBox-Android (wajib init resource loader dan ekstraksi teks).
 - TXT: buffered reader.

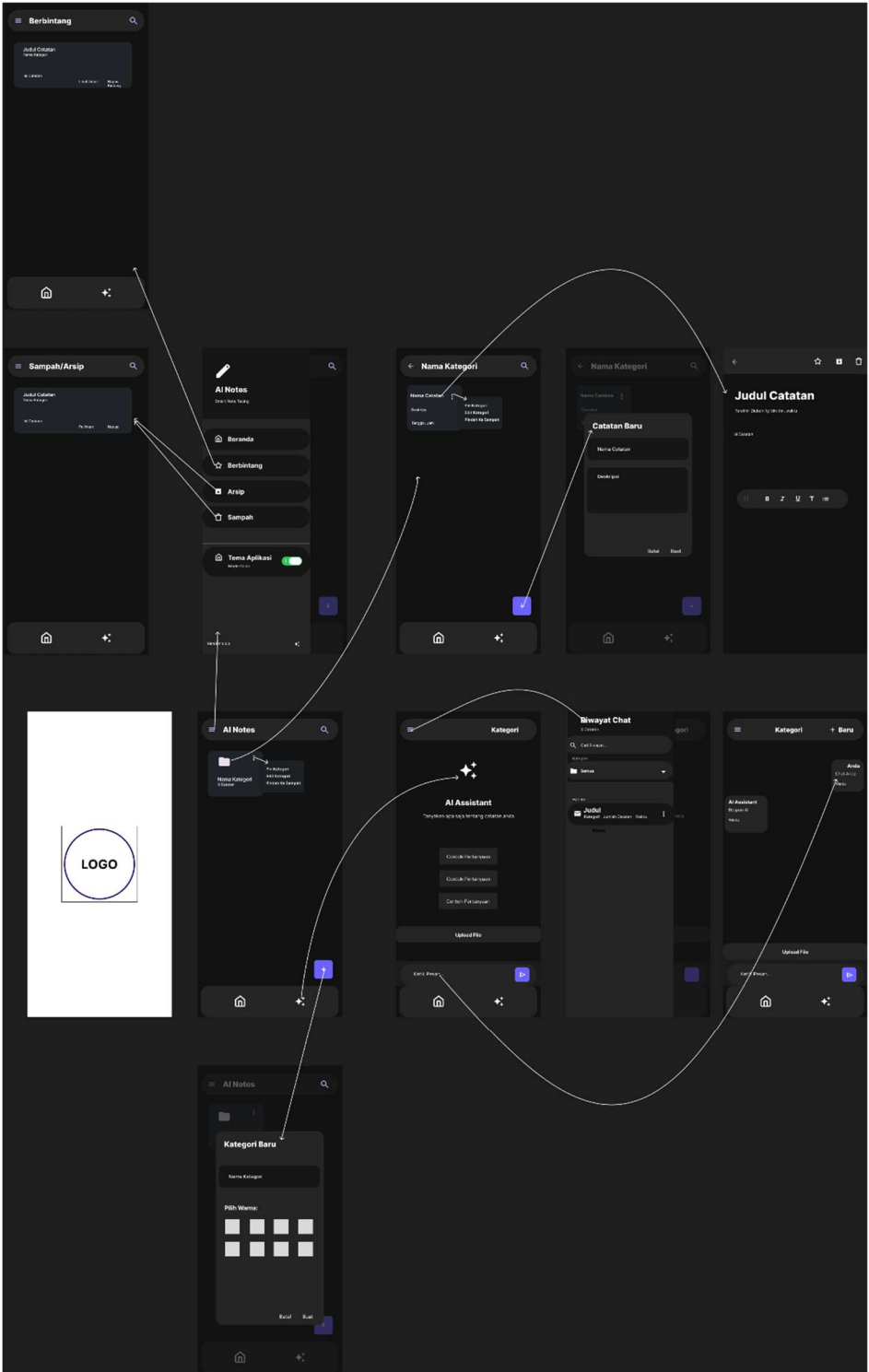
- DOCX: ekstraksi word/document.xml dari ZIP dan parsing text tags (regex).

- Format yang tidak didukung menghasilkan error message.

Edge case penting untuk dicantumkan:

- File kosong atau gagal dibaca → error.
- Koneksi internet / limit API saat memanggil AI.
- Privasi: konten yang dikirim ke AI perlu disclosure.

3.4 Tampilan Aplikasi (Wireframe)



Gambar 1 Wireframe Aplikasi

3.5 Teknologi dan Alat yang Digunakan

1. Bahasa & UI

Kotlin + Jetpack Compose (Material 3).

2. Arsitektur & State

- Single-activity Compose UI (state dikelola di Composable/MainActivity), dengan pemisahan package data dan presentation.
- Untuk rujukan standar arsitektur, Android merekomendasikan minimal UI layer dan Data layer (opsional Domain).

3. Penyimpanan local

Preferences DataStore untuk categories, notes, chat history, dan theme (diserialisasi JSON).

4. Serialisasi

Kotlinx Serialization JSON untuk menyimpan struktur data ke string JSON.

5. AI

- Integrasi Gemini via Gemini API / model Gemini 2.5 Flash.
- Claude AI untuk Pengembangan Aplikasi

6. File Handling

- Activity Result API (GetContent) untuk pemilihan file.
- PDFBox-Android untuk parsing PDF.
- DocumentFile dependency tersedia untuk akses dokumen berbasis URI (rekomendasi Jetpack dan API referensi).

7. Version Control

Selama pengembangan NotesAI, tim menggunakan sistem version control berbasis **Git** yang di-host pada **git.lab.ubharajaya.ac.id** untuk kolaborasi 3 anggota. Git digunakan untuk mengelola perubahan kode secara terstruktur melalui mekanisme **commit**, **branching** (pemisahan pekerjaan per fitur), **push**, dan **merge** (penggabungan kontribusi anggota), sehingga memudahkan koordinasi, pelacakan riwayat perubahan, serta meminimalkan konflik saat pengembangan paralel.

3.6 Arsitektur Aplikasi

Data Layer

- DataStoreManager: penyimpanan lokal untuk notes, categories, chat history, theme.
- FileParser: parsing PDF/TXT/DOCX menjadi teks.
- model: Note, Category, ChatHistory, ChatMessage + serializable wrapper.

Presentation Layer

- MainActivity + Composable screens (MainScreen, ArchiveScreen, TrashScreen, AIHelperScreen, dll).
- Komponen UI reusable: Card, Drawer, BottomBar, EmptyState.

3.7 Resource yang Digunakan

Dependensi Utama:

- Compose Material 3 untuk UI.
- Preferences DataStore untuk storage lokal.
- Kotlinx Serialization JSON untuk encode/decode data.
- PdfBox-Android untuk ekstraksi teks PDF.
- Activity Result GetContent untuk file picker
- Gemini 2.5 Flash untuk AI responses.

3.8 Pengujian Fungsional

Pengujian fungsional dilakukan secara manual dengan metode black-box berbasis test case, didukung oleh Automated Unit Tests, memverifikasi kesesuaian expected vs actual result pada fitur inti (notes, kategori, pin, trash/restore, search, AI context, dan upload-summary) serta didukung bukti screenshot dan status pengujian. Unit testing adalah pengujian otomatis terhadap unit terkecil dari kode (functions, methods) untuk memvalidasi logic internal aplikasi.

ID	Fitur	Skenario	Langkah Uji	Expected Result	Actual Result	Status
TC-01	Create Note & Category + Autosave	Membuat note dan category baru dengan autosave debounce 500ms	1. Buat category baru; 2. Buat note baru; 3. Edit note; 4. Tunggu 500ms; 5. Verify autosave	- Category tersimpan dengan benar; - Note tersimpan dengan benar; - Autosave berjalan setelah 500ms debounce; - Data ter-update;	- ☒ Category tersimpan; - ☒ Note tersimpan ; - ☒ Multiple categories ; - ☒ Autosave works	☒ PASSED (8 tests)
TC-02	Pin Note	Pin note muncul di urutan teratas	1. Buat 3 notes dengan timestamp berbeda; 2. Pin note terlama; 3. Verify urutan	- Pinned note muncul di posisi teratas; - Unpinned notes diurutkan berdasarkan timestamp; - Multiple pinned notes diurutkan by timestamp	- ☒ Pinned note first; - ☒ Multiple pins sorted; - ☒ Unpin works; - ☒ Category pin	☒ PASSED (included in TC-01)
TC-03	Soft Delete & Restore	Soft delete note/category dan restore dari trash	1. Delete note/category (soft delete); 2. Verify item masuk trash; 3. Restore item dari trash; 4. Verify item kembali aktif; 5. Test permanent delete	- Item ditandai `isDeleted=true`; - Item muncul di trash screen; - Restore mengembalikan `isDeleted=false`; - Data tetap preserved; - Permanent delete menghapus sepenuhnya	- ☒ Soft delete note; - ☒ Restore note; - ☒ Soft delete category; - ☒ Restore category; - ☒ Filter deleted; - ☒ Permanent delete; - ☒ Data preserved	☒ PASSED (11 tests)
TC-04	Search Realtime	Search realtime menemukan keyword di notes dan categories	1. Buat multiple notes dengan content berbeda; 2. Input search query; 3. Verify hasil realtime; 4. Test case-insensitive; 5. Test partial match; 6. Test filter by category	- Search menemukan notes by title; - Search menemukan notes by content; - Case-insensitive search works; - Partial match ditemukan; - Empty query return all; - Exclude deleted & archived notes	- ☒ Search by title; - ☒ Search by content; - ☒ Case insensitive; - ☒ Partial match; - ☒ Empty query; - ☒ Exclude deleted; - ☒ Search category; - ☒ Realtime update	☒ PASSED (14 tests)
TC-05	AI Chat with Context	AI chat menjawab dengan konteks note (gunakan 1-2 contoh note)	1. Buat 1-2 sample notes; 2. Open AI chat; 3. Send query tentang notes; 4. Verify AI dapat akses context; 5. Save chat history; 6. Load chat history	- Context mencakup semua notes; - Context filtered by category; - Exclude archived notes; - Chat history tersimpan; - Chat history bisa di-load;	- ☒ Build context; - ☒ Filter by category; - ☒ Exclude archived; - ☒ Save history; - ☒ Load history; - ☒ Sort histories; - ☒ Update history;	☒ PASSED (14 tests)

				- Custom title bisa di-set	- ☒ Delete history; - ☒ Custom title	
TC-06	Upload PDF → Summary	Upload PDF dan summary tersimpan/terbaca	1. Upload file (PDF/TXT/DOCX); 2. Verify file parsed; 3. Generate AI summary; 4. Save summary to chat history; 5. Verify summary readable; 6. Test error handling	- File di-parse dengan benar; - Word count calculated; - File type identified; - Summary generated & saved; - Summary tersimpan di chat history; - Metadata preserved; - Error handling gracefully	- ☒ Word count; - ☒ File type; - ☒ File size format; - ☒ Save summary; - ☒ Multiple uploads; - ☒ Summary readable; - ☒ Error handling; - ☒ Structured format; - ☒ Metadata preserved	☒ PASSED (12 tests)

Tabel 1 Test Cases Summary

Test Results	Duration	samsung SM-A127F
✓ DataStoreManagerTest	1 s	8/8
✓ testPinCategory_shouldPersist	756 ms	✓
✓ testUnpinNote_shouldMoveToNormalPosition	61 ms	✓
✓ testCreateNote_shouldSaveSuccessfully	38 ms	✓
✓ testAutoSave_shouldUpdateExistingNote	58 ms	✓
✓ testCreateMultipleCategories_shouldSaveInCorre	33 ms	✓
✓ testPinNote_shouldAppearFirst	136 ms	✓
✓ testCreateCategory_shouldSaveSuccessfully	38 ms	✓
✓ testMultiplePinnedNotes_shouldSortByTimestamp	49 ms	✓

Gambar 2 TC-01 &TC-02

Test Results	Duration	samsung SM-A127F
✓ TrashFunctionalityTest	1 s	11/11
✓ testRestoreNoteFromTrash_shouldUnmarkDelete	438 ms	✓
✓ testFilterDeletedNotes_shouldOnlyShowDeleted	66 ms	✓
✓ testSearchInTrash_shouldFindDeletedItems	41 ms	✓
✓ testSoftDeleteNote_shouldMarkAsDeleted	48 ms	✓
✓ testSoftDeleteCategory_shouldMarkAsDeleted	22 ms	✓
✓ testRestoreMultipleNotes_shouldRestoreAll	68 ms	✓
✓ testPermanentDeleteNote_shouldRemoveComple	219 ms	✓
✓ testDeletedNotePreservesAllData_shouldKeepCc	42 ms	✓
✓ testFilterDeletedCategories_shouldOnlyShowDel	32 ms	✓
✓ testRestoreCategoryFromTrash_shouldUnmarkDe	40 ms	✓
✓ testPermanentDeleteCategory_shouldRemoveCo	38 ms	✓

Gambar 3 TC-03

Test Results	Duration	samsung SM-A127F
✓ SearchFunctionalityTest	879 ms	14/14
✓ testSearchNoteByTitle_shouldFindMatches	429 ms	✓
✓ testSearchMultipleKeywords_shouldFindAll	25 ms	✓
✓ testSearchCategoryPartialMatch_shouldFind	39 ms	✓
✓ testSearchExcludesDeletedNotes_shouldNotFind	66 ms	✓
✓ testSearchCaseInsensitive_shouldFindMatches	10 ms	✓
✓ testSearchRealTime_shouldUpdateImmediately	11 ms	✓
✓ testSearchEmptyQuery_shouldReturnAllNotes	61 ms	✓
✓ testSearchNoteByContent_shouldFindMatches	39 ms	✓
✓ testSearchPartialMatch_shouldFindResults	40 ms	✓
✓ testSearchNoMatches_shouldReturnEmpty	38 ms	✓
✓ testSearchWithSpecialCharacters_shouldHandle	40 ms	✓
✓ testSearchCategory_shouldFindByName	9 ms	✓
✓ testSearchFilteredByCategory_shouldOnlySearch	35 ms	✓
✓ testSearchExcludesArchivedNotes_shouldNotFin	39 ms	✓

Gambar 4 TC-04

Test Results	Duration	samsung SM-A127F
✓ AIChatFunctionalityTest	1 s	14/14
✓ testUpdateChatTitle_shouldUpdate	486 ms	✓
✓ testEmptyNotesContext_shouldHandleGracefully	53 ms	✓
✓ testLoadChatHistory_shouldRestoreMessages	48 ms	✓
✓ testUpdateChatHistory_shouldUpdateExisting	58 ms	✓
✓ testChatWithContext_shouldBuildCorrectPrompt	35 ms	✓
✓ testSaveChatHistory_shouldPersist	66 ms	✓
✓ testBuildNotesContext_shouldExcludeArchivedN	60 ms	✓
✓ testChatPreview_shouldTruncateLongMessages	28 ms	✓
✓ testMultipleChatHistories_shouldSortByTimestar	461 ms	✓
✓ testChatMessageConversion_shouldPreserveDat	30 ms	✓
✓ testCustomChatTitle_shouldPersist	60 ms	✓
✓ testDeleteChatHistory_shouldMarkAsDeleted	63 ms	✓
✓ testBuildNotesContext_shouldFilterByCategory	56 ms	✓
✓ testBuildNotesContext_shouldIncludeAllNotes	25 ms	✓

Gambar 5 TC-05

Test Results	Duration	samsung SM-A127F
✓ FileUploadFunctionalityTest	1 s	12/12
✓ testSearchInSummaries_shouldFindKeywords	635 ms	✓
✓ testFileParseResult_shouldIdentifyFileType	21 ms	✓
✓ testFileParseResult_shouldCalculateWordCount	39 ms	✓
✓ testSummaryContent_shouldBeReadable	37 ms	✓
✓ testSaveSummaryToChatHistory_shouldPersist	48 ms	✓
✓ testMultipleFileUploads_shouldTrackAll	31 ms	✓
✓ testLongSummary_shouldTruncatePreview	200 ms	✓
✓ testFileParseError_shouldContainMessage	56 ms	✓
✓ testFileMetadata_shouldBePreserved	66 ms	✓
✓ testFormatFileSize_shouldFormatCorrectly	32 ms	✓
✓ testFileUploadError_shouldHandleGracefully	48 ms	✓
✓ testPDFSummaryFormat_shouldBeStructured	29 ms	✓

Gambar 6 TC-06

BAB III

PANDUAN PENGGUNA

3.1 Cara Install/Akses

3.1.1 Prasyarat Perangkat

1. Perangkat Android minimal Android 7.0 (API 24, dan minSdk=24).
2. Koneksi internet dibutuhkan **untuk fitur AI Assistant**, karena request dikirim ke layanan model Gemini.

3.1.2 Cara Menjalankan Aplikasi

Langkah ini digunakan untuk dosen/penguji yang menjalankan dari source code:

1. Buka proyek **NotesAI** di **Android Studio**.
2. Lakukan **Gradle sync** hingga dependensi terunduh.
3. Jalankan aplikasi melalui menu **Run** ke emulator atau perangkat fisik. Prosedur “build and run” di Android Studio dijelaskan pada dokumentasi resmi Android Developers.
4. Jika menggunakan perangkat fisik, aktifkan mode developer dan debugging USB sesuai panduan Android Developers.

3.1.3 Izin Aplikasi (Permissions)

Pada implementasi NotesAI, aplikasi mendeklarasikan:

1. **INTERNET**: untuk komunikasi AI.
2. **Akses media/berkas**:
untuk kebutuhan pemilihan dokumen saat upload (PDF/TXT/DOCX). Pada sisi UI, pemilihan file menggunakan mekanisme pemilih konten Android (ActivityResultContracts.GetContent) yang mengembalikan URI file.

3.2 Mockup + Userflow (Alur Penggunaan)

3.2.1 Alur Utama: Membuat Kategori dan Catatan

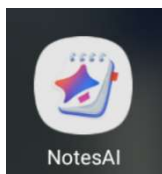
Tujuan: membuat struktur catatan terorganisir per kategori.

Langkah:

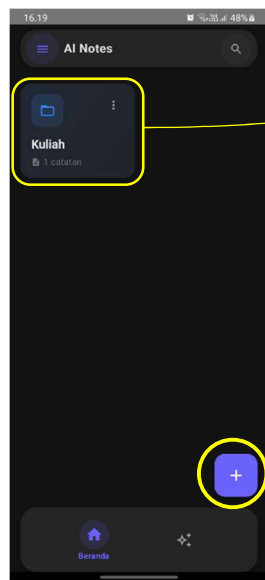
1. Buka Aplikasi → Main Screen.
2. Buat kategori baru (pilih nama dan preset warna/gradien).
3. Pilih kategori tersebut → buat catatan baru.
4. Catatan tampil di daftar kategori dan bisa dibuka ke editor.

Hasil yang diharapkan:

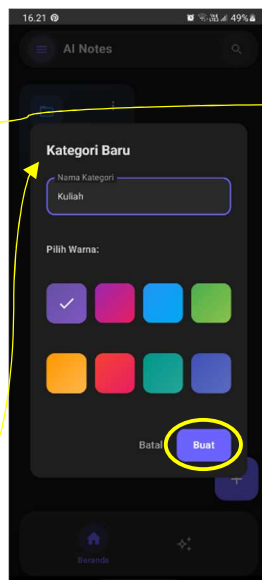
- Kategori tersimpan, catatan tersimpan, dan daftar menampilkan catatan terbaru pada kategori terpilih.



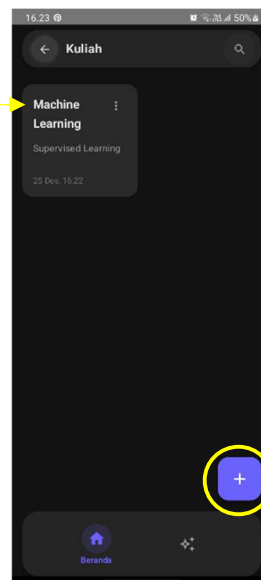
Gambar 7
Logo Aplikasi



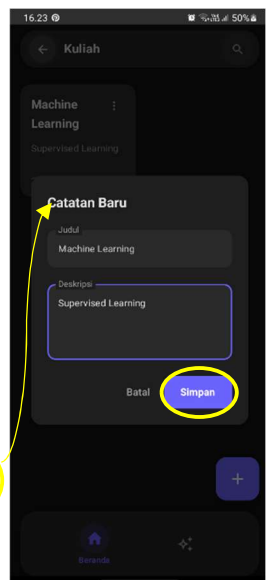
Gambar 8
Main Screen



Gambar 9
Category Dialog



Gambar 10 Notes
Screen



Gambar 11 Notes
Dialog

3.2.2 Alur Editor: Menulis Catatan dengan Rich Text (WYSIWYG)

Tujuan: menulis catatan dengan format tanpa markdown mentah.

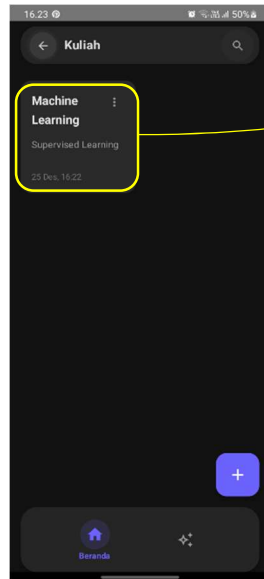
Langkah:

1. Dari daftar catatan, tap salah satu catatan untuk membuka Full-screen Editor.
2. Gunakan toolbar format (mis. bold/italic/underline/undo/redo) sesuai kebutuhan.
3. Keluar dari editor (tombol back/navigasi).

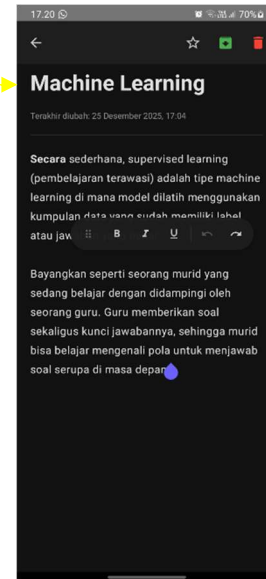
4. Aplikasi menyimpan perubahan ketika lifecycle aplikasi berpindah ke background (mis. pengguna keluar aplikasi), sehingga risiko kehilangan perubahan berkurang.

Hasil yang diharapkan:

- Isi catatan tersimpan dan format tampilan tetap sesuai yang ditulis.



Gambar 12 Notes Screen 2



Gambar 13 Editable Note Screen

3.2.3 Alur Prioritas: Pin Catatan dan Pin Kategori

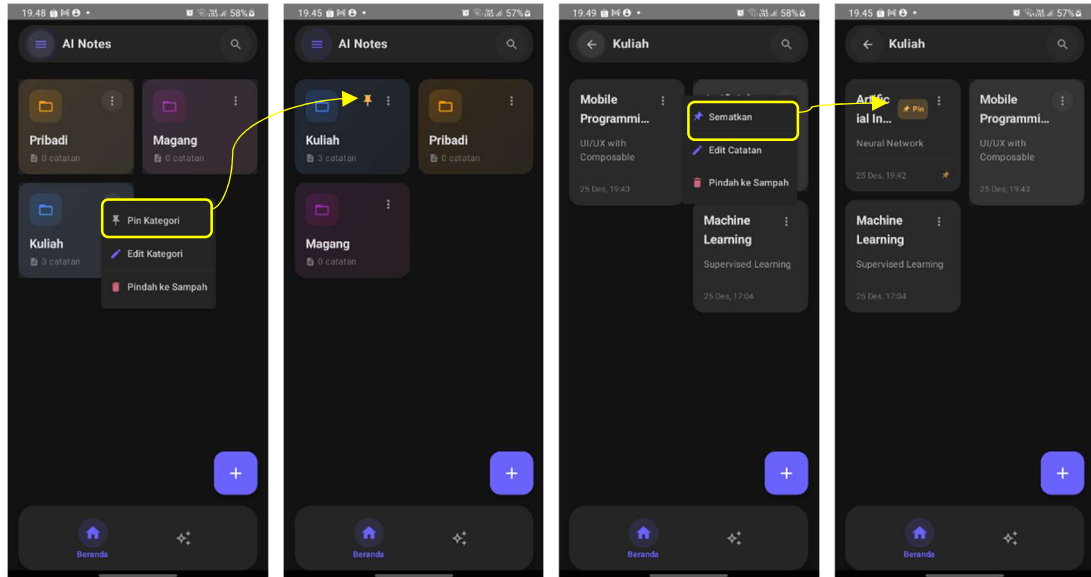
Tujuan: memudahkan akses cepat ke item penting.

Langkah:

1. Pada daftar catatan, aktifkan Pin untuk catatan yang prioritas.
2. (Opsional) Pin kategori favorit agar tampil lebih atas pada daftar kategori.
3. Verifikasi urutan tampilan: pinned muncul lebih dahulu, lalu item lain mengikuti urutan terbaru.

Hasil yang diharapkan:

- Catatan/kategori yang dipin tampil di posisi atas sehingga mudah ditemukan.



Gambar 14
Kategori Sebelum di Pin

Gambar 15
Kategori Sesudah di Pin

Gambar 16
Catatan Sebelum di Pin

Gambar 17
Catatan Sesudah di Pin

3.2.4 Alur Keamanan Data

Tujuan: mengurangi risiko kehilangan data akibat salah hapus.

A. Archive (pengarsipan)

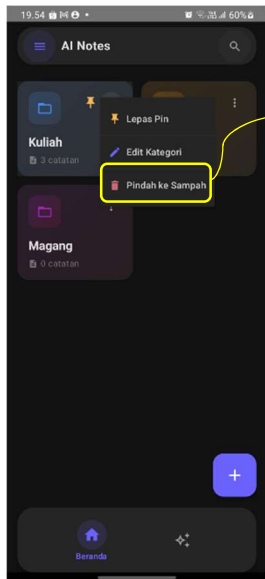
1. Pilih catatan → Archive.
2. Catatan pindah ke Archive Screen.
3. Restore/unarchive bila ingin mengembalikan ke daftar utama.

B. Trash (soft delete)

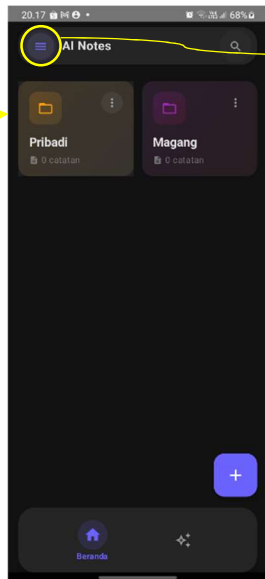
1. Hapus catatan/kategori → item masuk Trash Screen (bukan hilang permanen).
2. Pada Trash Screen, pilih:
 - Restore untuk mengembalikan, atau
 - Permanent Delete untuk menghapus permanen.

Hasil yang diharapkan:

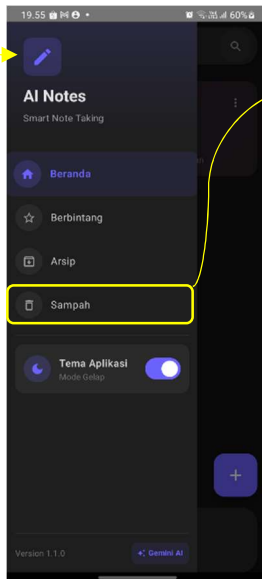
- Item yang dihapus bisa dipulihkan selama belum dihapus permanen.



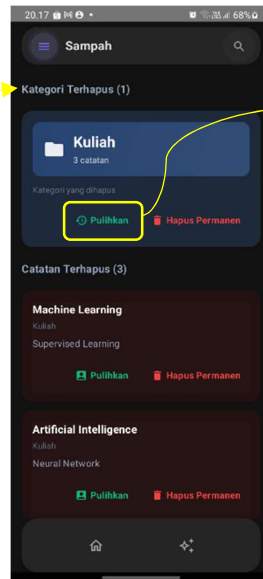
Gambar 18
Hapus Kategori



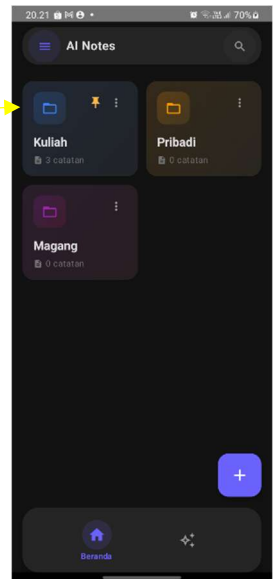
Gambar 19
Setelah Kategori Terhapus



Gambar 20
Drawer Menu



Gambar 21
Halaman Sampah



Gambar 22
Setelah di Pulihkan

Catatan: Fungsi pengarsipan hanya berlaku pada Catatan saja dan implementasinya sama dengan menghapus dan memulihkan kategori/catatan

3.2.5 Alur Pencarian: Search Realtime + Filter Kategori

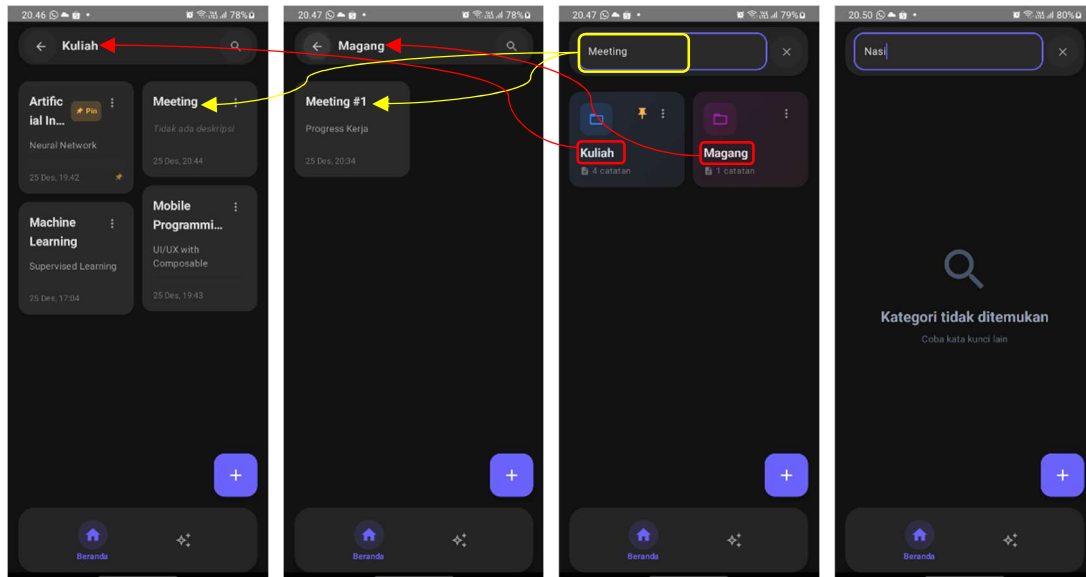
Tujuan: menemukan catatan lebih cepat berdasarkan judul/deskripsi.

Langkah:

1. Pada Main Screen, ketik keyword pada kolom search.
2. (Opsional) Pilih kategori untuk mempersempit konteks pencarian.
3. Verifikasi hasil ter-update saat query berubah.

Hasil yang diharapkan:

- Catatan yang sesuai keyword tampil, dan filter kategori pada catatan bekerja.



Gambar 23
Catatan di Kategori
Kuliah

Gambar 24
Catatan di Kategori
Magang

Gambar 25
Pencarian Catatan
dengan Keyword Cocok

Gambar 26
Pencarian Catatan
dengan Keyword Asal

3.2.6 Alur AI Assistant: Chat Berbasis Konteks Catatan

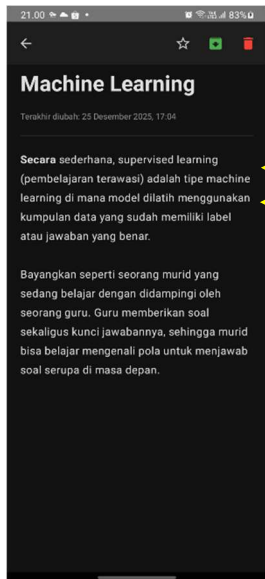
Tujuan: mendapatkan ringkasan/insight/penjelasan dari catatan.

Langkah:

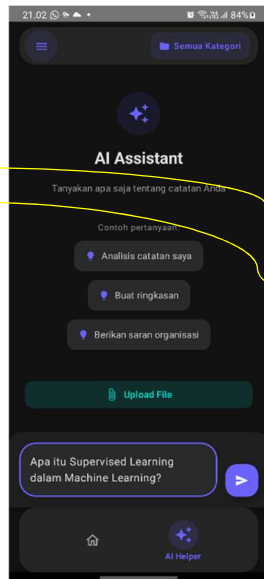
1. Buka AI Screen.
2. Pilih konteks:
3. Semua Kategori atau kategori tertentu (membatasi konteks).
4. Ketik pertanyaan yang terkait catatan, lalu kirim.
5. Lihat jawaban AI; riwayat chat tersimpan dan dapat dibuka kembali.

Catatan:

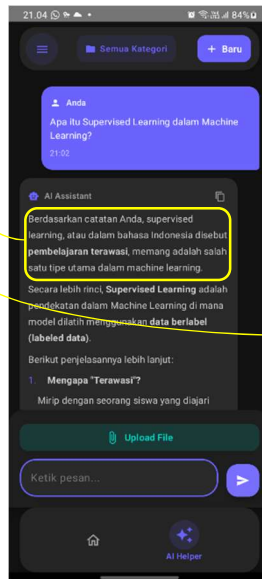
- NotesAI menggunakan model Gemini 2.5 Flash, yang pada dokumentasi resmi dijelaskan sebagai model “price-performance” untuk low-latency/high-volume tasks dan aplikasi menggunakan ‘free tier’ sehingga terbatas.



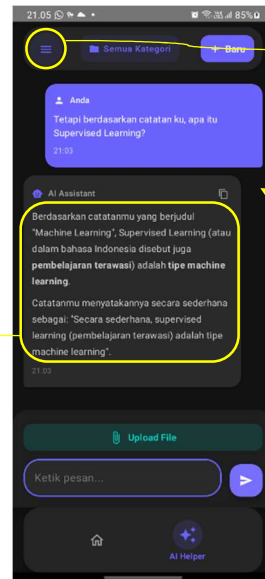
Gambar 27
Catatan Machine Leaning



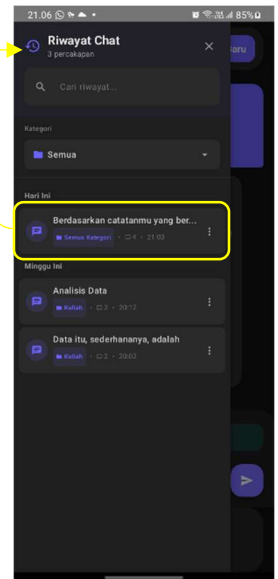
Gambar 28
AI Helper Screen



Gambar 29
AI Response 1



Gambar 30
AI Response 2



Gambar 31
History Chat

3.2.7 Alur Upload Dokumen: PDF/TXT/DOCX → Ringkasan AI

Tujuan: merangkum dokumen menjadi informasi singkat dan dapat disimpan di chat.

Langkah:

1. Di AI Screen, pilih Upload File.
2. Pilih file PDF/TXT/DOCX dari penyimpanan perangkat.
3. Sistem melakukan parsing dan menampilkan metadata (nama file, tipe file, word count).
4. Gunakan tombol "Upload File" otomatis sistem akan menghasilkan summary.
5. Summary tersimpan di chat history.

Hasil yang diharapkan:

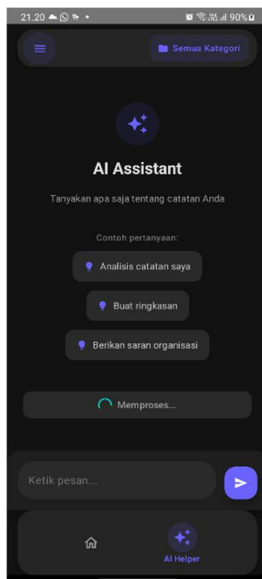
- File berhasil dibaca, ringkasan terbentuk, dan riwayat tersimpan.
- **Fitur AI membutuhkan koneksi internet** dan ketersediaan layanan model AI. (Saat offline, fitur catatan lokal tetap dapat digunakan.)
- **Format dokumen yang didukung untuk upload** dibatasi pada PDF,

TXT, DOCX (sesuai implementasi parser).

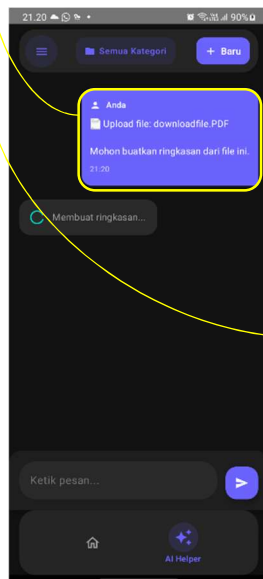
- **Privasi dan data sharing:** catatan disimpan lokal; namun saat pengguna memakai AI, konten yang relevan dikirim sebagai input ke layanan Gemini. Rekomendasi laporan: cantumkan disclosure/consent pada bagian AI.
- **Skala data besar:** jika jumlah catatan sangat banyak, performa pencarian/filter dapat menurun bila belum ada optimasi indexing khusus (ini wajar untuk versi awal).
- **Lingkup penyimpanan:** penyimpanan menggunakan mekanisme data lokal (DataStore). Android merekomendasikan DataStore untuk penyimpanan key-value/objek ringan pada layer data.



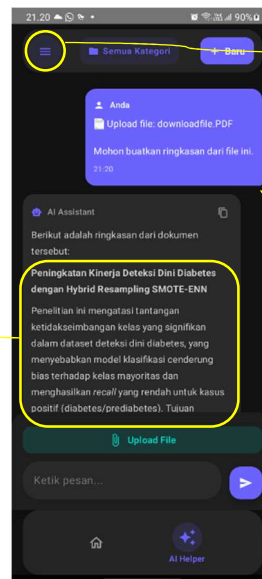
Gambar 32 File PDF yang di Upload



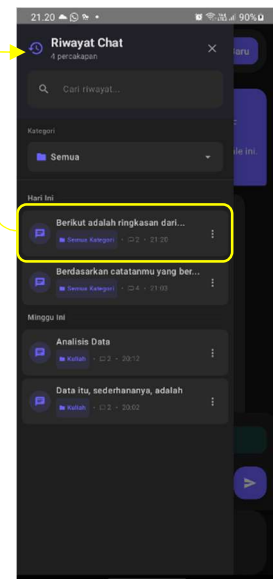
*Gambar 33
Proses File Parser*



*Gambar 34
Proses Summary*



*Gambar 35
Hasil Summary*



*Gambar 36
History Chat Summary*

3.3 Batasan

1. **Fitur AI membutuhkan koneksi internet dan ketersediaan layanan model AI terbatas karena aplikasi menggunakan ‘free tier’.** (Saat offline, fitur catatan lokal tetap dapat digunakan.)
2. Format dokumen yang didukung untuk upload **dibatasi pada** PDF, TXT, DOCX (sesuai implementasi parser).
3. **Privasi dan data sharing:** catatan disimpan lokal; namun saat pengguna memakai AI, konten yang relevan dikirim sebagai input ke layanan Gemini. Rekomendasi laporan: cantumkan disclosure/consent pada bagian AI.
4. **Skala data besar:** jika jumlah catatan sangat banyak, performa pencarian/filter dapat menurun bila belum ada optimasi indexing khusus (ini wajar untuk versi awal).
5. **Lingkup penyimpanan:** penyimpanan menggunakan mekanisme data lokal (DataStore). Android merekomendasikan DataStore untuk penyimpanan key-value/objek ringan pada layer data.

BAB VI PENUTUP

4.1 Kesimpulan

Berdasarkan proses analisis, perancangan, implementasi, dan pengujian fungsional pada aplikasi **NotesAI**, dapat disimpulkan bahwa:

1. **Tujuan utama aplikasi sesuai dengan fitur**, yaitu membantu pengguna mengelola catatan harian agar lebih terorganisir dan mudah ditemukan melalui:
 - Sistem **kategori visual**, **pin** untuk prioritas, serta **archive** dan **trash** dengan **soft delete**.
 - Fitur **pencarian real-time** dan filtering untuk mempercepat retrieval catatan.
2. **Integrasi AI berfungsi sebagai akselerator produktivitas**, karena pengguna dapat memperoleh ringkasan atau respons kontekstual berdasarkan catatan maupun dokumen yang diunggah (PDF/TXT/DOCX), sehingga proses ekstraksi informasi menjadi lebih cepat dibanding pencarian manual.
3. **Reliabilitas fitur inti tervalidasi melalui pengujian fungsional**, dibuktikan dengan test case yang mencakup pembuatan catatan/kategori dan autosave debounce, pin & sorting, soft delete & restore, search real-time, AI chat berbasis konteks, serta upload dokumen dan penyimpanan ringkasan dengan status pengujian **PASSED** sesuai bukti yang telah Anda lampirkan.

4.2 Saran Pengembangan Selanjutnya

Untuk meningkatkan kualitas aplikasi dari sisi skalabilitas, keamanan, dan pengalaman pengguna, berikut saran pengembangan yang realistis dan relevan dengan standar praktik Android modern.

4.2.1 Peningkatan Arsitektur dan Skalabilitas Data

1. **Migrasi penyimpanan catatan dari Preferences DataStore ke database (Room) untuk data besar**

DataStore (Preferences) ideal untuk preferensi ringan, sedangkan catatan yang bertambah banyak akan lebih tepat dikelola dengan database yang mendukung query, indexing, dan performa retrieval yang stabil.

Rujukan: Android Developers merekomendasikan pemisahan layer (UI–Data) dan penggunaan penyimpanan yang tepat untuk skala data. (Android App Architecture / Data Layer guidance).

2. **Implementasi MVVM (ViewModel) per screen + Repository pattern**

Saat fitur bertambah, ViewModel membantu pemisahan logika dari UI (Composable) sehingga maintainability meningkat, testing lebih mudah, dan state management lebih rapi.

Rujukan: Android Developers (Guide to app architecture).

3. **Optimasi pencarian dengan indexing**

Untuk memastikan search tetap cepat pada jumlah catatan besar, dapat ditambahkan mekanisme indexing (mis. inverted index sederhana) atau full-text search jika menggunakan database.

4.2.2 Keamanan dan Privasi (High Priority)

1. **Hilangkan hardcoded API key dan gunakan mekanisme secret yang aman**

Kunci API sebaiknya tidak disimpan di source code. Gunakan local.properties, BuildConfig, atau secret manager agar aman, terutama bila repo dipublikasikan.

Rujukan praktik: OWASP Mobile Application Security (prinsip pengelolaan secret dan keamanan konfigurasi).

2. **Tambahkan disclosure/consent yang jelas untuk fitur AI**

Karena AI memerlukan pengiriman konten tertentu ke layanan eksternal, perlu:

- Consent sebelum penggunaan pertama,
- Penjelasan data apa yang dikirim,
- Opsi mematikan AI atau membatasi konteks (kategori tertentu).

Rujukan praktik: prinsip privacy-by-design dan transparansi pemrosesan data (umum di pedoman privasi aplikasi).

3. **Enkripsi data lokal (opsional tetapi kuat untuk nilai laporan)**

Jika catatan memuat informasi sensitif, pertimbangkan enkripsi penyimpanan lokal menggunakan Jetpack Security (EncryptedSharedPreferences/EncryptedFile) atau pendekatan enkripsi di database. Rujukan: AndroidX Security Crypto (praktik enkripsi lokal pada Android).

4.2.3 Peningkatan UX dan Fitur Produktivitas

1. **Sinkronisasi dan backup multi-perangkat (opsional, fase lanjut)**

Dukungan backup terenkripsi atau sinkronisasi akun akan meningkatkan keberlanjutan data pengguna.

2. **Reminder, checklist, dan template catatan**

Tambahkan jenis catatan khusus (to-do checklist), template meeting notes, atau template kuliah untuk mempercepat input.

3. **Widget home-screen dan quick capture**

Shortcut tambah catatan cepat atau widget akan meningkatkan utilisasi untuk pengguna harian.

4. **Aksesibilitas dan internationalization (i18n)**

Perbaiki kontras, ukuran teks dinamis, dukungan screen reader, serta multi-bahasa akan memperluas cakupan pengguna.

4.2.4 Peningkatan Kualitas melalui Pengujian dan Otomasi

5. **Tambahkan unit test dan UI/instrumentation test**

Saat ini Anda sudah kuat di pengujian fungsional manual. Tahap berikutnya adalah otomatisasi test untuk fitur kritis: parsing dokumen, sorting pinned, restore trash, dan state AI chat.

6. **CI pipeline (GitHub Actions) untuk build & test**

Mencegah regresi dan memastikan kualitas rilis stabil

DAFTAR PUSTAKA

- ActivityResultContracts.GetContent* | *API reference* | *Android Developers*. (n.d.). Retrieved December 25, 2025, from <https://developer.android.com/reference/androidx/activity/result/contract/ActivityResultContracts.GetContent?>
- Artificial Intelligence and the future of work - February 2025 - - Eurobarometer survey*. (n.d.). Retrieved December 25, 2025, from <https://europa.eu/eurobarometer/surveys/detail/3222?>
- Desain Material 3 di Compose* | *Jetpack Compose* | *Android Developers*. (n.d.). Retrieved December 25, 2025, from <https://developer.android.com/develop/ui/compose/designsystems/material3?>
- Documentfile* | *Jetpack* | *Android Developers*. (n.d.). Retrieved December 25, 2025, from <https://developer.android.com/jetpack/androidx/releases/documentfile?>
- Gemini 2.5 Flash* | *Generative AI on Vertex AI* | *Google Cloud Documentation*. (n.d.). Retrieved December 25, 2025, from <https://docs.cloud.google.com/vertex-ai/generative-ai/docs/models/gemini/2-5-flash?>
- Menangani Preferences DataStore*. (n.d.). Retrieved December 25, 2025, from <https://developer.android.com/codelabs/android-preferences-datastore?>
- Note Taking App Market Report 2025 - Research and Markets*. (n.d.). Retrieved December 25, 2025, from <https://www.researchandmarkets.com/reports/5790688/note-taking-app-market-report?>
- Note Taking App Market Report 2025 Analysis And Forecast 2034*. (n.d.). Retrieved December 25, 2025, from <https://www.thebusinessresearchcompany.com/report/note-taking-app-global-market-report?>
- Panduan arsitektur aplikasi* | *App architecture* | *Android Developers*. (n.d.). Retrieved December 25, 2025, from <https://developer.android.com/topic/architecture?>
- Serialization* | *Kotlin Documentation*. (n.d.). Retrieved December 25, 2025, from <https://kotlinlang.org/docs/serialization.html?>
- The State of AI: Global Survey 2025* | *McKinsey*. (n.d.). Retrieved December 25, 2025, from <https://www.mckinsey.com/capabilities/quantumblack/our-insights/the-state-of-ai?>

TomRoush/PdfBox-Android: The Apache PdfBox project ported to work on Android. (n.d.).

Retrieved December 25, 2025, from <https://github.com/TomRoush/PdfBox-Android?>