BAB I
PENDAHULUAN

1.1 Latar Belakang Masalah

Perkembangan teknologi informasi dan komunikasi, khususnya pada perangkat telepon pintar (*smartphone*), telah mengalami lonjakan yang sangat eksponensial. *Smartphone* modern yang dilengkapi dengan prosesor berkinerja tinggi, kapasitas memori yang besar, dan konektivitas internet pita lebar telah bertransformasi dari sekadar alat komunikasi suara dua arah menjadi asisten komputasi genggam yang sangat vital. Fenomena ini memicu lahirnya ribuan aplikasi berbasis produktivitas dan pemantau kebiasaan (*habit tracker*) di toko aplikasi.

Sebagian besar aplikasi *habit tracker* berupaya memfasilitasi pengguna untuk mencatat aktivitas harian. Metode pencatatan ini secara historis terbukti dapat meningkatkan konsistensi seseorang dalam mencapai target (Clear, 2018). Namun demikian, sebuah permasalahan laten yang kerap dihadapi oleh pengembang adalah tingginya angka penghentian penggunaan aplikasi (*churn rate*). Menurut kajian empiris, salah satu penyebab utama dari rendahnya retensi pengguna pada aplikasi pengingat tradisional adalah antarmuka yang pasif dan kaku, yang pada akhirnya memicu fenomena *notification fatigue* atau kelelahan notifikasi di mana pengguna mulai mengabaikan peringatan dari gawai mereka (Stawarz et al., 2015).

Untuk memecahkan masalah ini, teori perubahan perilaku dapat diterapkan. *Fogg Behavior Model* (Model Perilaku Fogg) menyatakan bahwa pembentukan perilaku (*Behavior*) dipengaruhi oleh konvergensi tiga elemen utama: Motivasi (*Motivation*), Kemampuan (*Ability*), dan Pemicu (*Prompt/Trigger*), yang dirumuskan sebagai *B=MAP* (Fogg, 2009). James Clear (2018) juga menggemakan hal serupa melalui prinsip "*Make It Obvious*" (Buatlah Terlihat Jelas). Menyimpan aplikasi habit

tracker di dalam laci aplikasi (menu utama) menyembunyikannya dari pandangan pengguna. Hal ini bertentangan dengan prinsip '*Make It Obvious*' dari James Clear, yang menekankan bahwa agar sebuah kebiasaan dapat terbentuk, pemicunya harus terlihat jelas dan mudah diakses oleh pengguna. Oleh karena itu, diperlukan pergeseran desain antarmuka menjadi *Desktop Pet* atau *Floating Widget,* entitas virtual yang secara persisten melayang di layar pengguna, terlepas dari aplikasi apa pun yang sedang dieksekusi. Keberadaan visual ini berfungsi sebagai pemicu yang kuat.

Di luar antarmuka visual, aspek motivasi membutuhkan pendekatan kognitif tingkat tinggi. Sistem notifikasi biasa tidak mampu menyuplai kebutuhan emosional manusia. Di sinilah *Large Language Models* (*LLM*) memainkan peran kunci. Model bahasa raksasa yang dilatih untuk mematuhi instruksi kustom (*instruction tuning*) mampu menghasilkan teks responsif yang sangat menyerupai dialog emosional manusia (Ouyang et al., 2022). Untuk memicu motivasi unik, penelitian ini menanamkan persona "*Tsundere*" pada asisten virtual tersebut dengan gaya komunikasi yang memarahi pengguna secara sarkastis saat malas, namun peduli terhadap perbaikan diri penggunanya. Interaktivitas ini juga diwujudkan dalam modalitas auditorial menggunakan layanan *Text-To-Speech* (*TTS*) modern (*Voice Vox*) guna memberikan sensasi entitas yang hidup.

Namun, rekayasa arsitektur ini menghadirkan tantangan teknis. *LLM* pada dasarnya bersifat amnesia dan tidak memiliki memori bawaan tentang interaksi masa lalu. Untuk mengatasi keterbatasan ini, pendekatan *Retrieval-Augmented Generation* (*RAG*) secara lokal diimplementasikan. Sistem menggunakan teknik *Object-Relational Mapping* (*ORM*) berbasis *Room Database* di *Android* untuk mengamankan persistensi memori konteks (Sarker, 2019). Sejarah percakapan dicatat dan disuntikkan secara dinamis ke dalam *payload LLM* setiap kali pesan baru dikirimkan.

Pada rancangan awal, peneliti bermaksud menggunakan *Computer Vision* (*MediaPipe Pose Landmarker*) dan aset 3D secara terus-menerus di

latar belakang. Akan tetapi, kajian performa membuktikan bahwa memaksakan proses komputasi latar belakang terus-menerus pada sistem operasi *Android* modern sangat rentan terhadap terminasi paksa oleh *OS* (*process kill*) serta memicu manajemen suhu yang buruk (*thermal throttling*) (Hasan & Hossain, 2020). Akibatnya, iterasi pengembangan memfokuskan sumber daya komputasi pada interaksi berbasis bahasa alami dengan memori terintegrasi dalam wadah *Floating Widget* yang ringan. Berdasarkan paparan di atas, berikut ini adalah penelitian dengan judul: "**Rancang Bangun *Virtual Assistant* Berbasis *Android* Menggunakan *LLM (Large Language Model)* Sebagai Pengingat dan Penasihat Aktivitas**".

1.2 Rumusan Masalah

1) Bagaimana mengintegrasikan arsitektur *Large Language Model* (*API OpenRouter*) dan *Text-To-Speech* (*VoiceVox*) ke dalam ekosistem *Android* secara asinkron tanpa memicu pembekuan antarmuka (*Main Thread Blocking*)?
2) Bagaimana mengimplementasikan *System Alert Window* dan *Foreground Service* di *Android* untuk menciptakan *Floating Widget* (*Desktop Pet*) yang persistent?
3) Bagaimana merancang manajemen memori luring menggunakan *Room Database* agar *state* percakapan historis dapat dikonversikan sebagai injeksi memori (mekanisme *RAG*) pada pemanggilan *LLM*?

1.3 Tujuan Penelitian

1. Membangun sistem asisten virtual di *Android* yang berhasil menjembatani output *LLM* dengan sintesis suara menjadi interaksi percakapan dua arah secara *real-time*.
2. Memproduksi fitur *Desktop Pet* berwujud *Floating Widget* 2D/2.5D yang mematuhi batasan *lifecycle Foreground Service* yang stabil.
3. Mewujudkan arsitektur basis data relasional *SQLite* (*Room*) yang terbukti mampu merekonstruksi aliran dialog masa lalu (*Message History*).

1.4 Batasan Masalah

A. ***Peniadaan Modul Computer Vision (No Camera Analysis):*** Berdasarkan limitasi komputasi seluler dan protokol privasi keamanan terbaru dari sistem operasi *Android* yang sangat ketat terhadap proses latar belakang (Hasan & Hossain, 2020), penelitian ini tidak menggunakan analisis postur (*MediaPipe*) demi menghindari pengurasan baterai.

B. **Penggunaan Aset Visual 2D/2.5D:** Menghindari fenomena penurunan laju *rendering* (*UI lagging*), antarmuka dikendalikan oleh *Jetpack Compose*, dan tidak merender berkas 3D (.glb) via *engine* eksternal.

C. **Ketergantungan *Cloud*:** Inferensi *LLM* tidak dilakukan secara *On-Device*. *Tokenisasi* 100% didelegasikan kepada *Open Router*, sedangkan sintesis suara pada *Voice Vox*.

D. **Target *Platform*:** *Android OS Minimum SDK* (*API Level* 26) hingga *Target SDK* terkini (*API Level* 34).

1.5 Manfaat Penelitian

1.5.1 Manfaat Teoritis

1. Memperkaya literatur *Human-Computer Interaction* (*HCI*) mengenai pergeseran paradigma *Habit Tracker*, di mana konsep B=MAP milik Fogg (2009) diuji secara ekstrem melalui *Floating Window* dan *Tsundere Personality*.
2. Menjadi rujukan teknis manajemen arsitektur pemanggilan *API* ganda asinkron pada kerangka kerja antarmuka deklaratif.

1.5.2 Manfaat Praktis

1. Menyediakan aplikasi pemantau aktivitas yang proaktif dan menghibur dibandingkan dengan sekadar mencentang kolom daftar tugas.
2. Memberikan wawasan empiris mengenai *Engineering Trade-off* dalam komputasi seluler.

1.6 Metodologi Penelitian

Penelitian ini mengadopsi kerangka metodologi rekayasa perangkat lunak dengan pendekatan empiris. Langkah-langkah sistematis yang

ditempuh dalam menyusun dan merealisasikan asisten virtual ini adalah sebagai berikut:

1. Studi Literatur: Melakukan komparasi kritis terhadap jurnal-jurnal terkini mengenai teori perubahan perilaku (*Fogg Behavior Model*) dan arsitektur *Large Language Models* (*LLM*), serta batasan perizinan *OS Android* (seperti *Foreground Service* dan *Doze Mode*).
2. Analisis Kebutuhan Sistem: Mendefinisikan kebutuhan fungsional (kemampuan memicu *API* awan dan menyimpan riwayat di *database SQLite*) dan kebutuhan non-fungsional (batasan penggunaan *RAM* dan pelestarian baterai gawai tanpa mengorbankan kelancaran antarmuka 60 *frame per second*).
3. Perancangan Arsitektur (Desain Sistem): Memetakan relasi logika menggunakan *Unified Modeling Language* (*UML*) seperti *Use Case, Activity Diagram,* dan *Sequence Diagram*, serta menyusun cetak biru *Entity Relationship* untuk modul *Retrieval-Augmented Generation* (*RAG*) lokal.
4. Implementasi Sistem (*Coding*): Menulis kode logika aplikasi menggunakan bahasa *Kotlin*, mendirikan antarmuka secara deklaratif menggunakan *Jetpack Compose*, dan membangun jembatan *HTTP RESTful* menggunakan pustaka *Retrofit*.
5. Pengujian Kinerja (*Testing & Evaluation*): Melakukan pengujian fungsional lapisan antarmuka (*Black-Box Testing*) serta pengujian komparatif kecerdasan buatan (*A/B Testing*) untuk memvalidasi latensi dan tingkat akurasi persona yang dihasilkan oleh *LLM*.
6. Penyusunan Laporan: Pendokumentasian seluruh artefak desain, kendala rekayasa (*engineering trade-offs*), dan kesimpulan evaluasi ke dalam struktur pelaporan ilmiah.

1.7 Sistematika Penulisan

Untuk memberikan panduan struktur yang komprehensif, penulisan skripsi ini disusun secara runut dalam lima bab utama sebagai berikut:

BAB I PENDAHULUAN

Menguraikan latar belakang isu rendahnya retensi pengguna pada aplikasi pengingat tradisional, rumusan masalah, batasan teknis penelitian, tujuan yang hendak dicapai, manfaat eksperimen, metodologi penelitian, dan kerangka sistematika penulisan.

BAB II TINJAUAN PUSTAKA

Menyajikan tinjauan terhadap kajian ilmiah terdahulu dan landasan teori fundamental yang mencakup psikologi pembentukan kebiasaan, mekanika *Large Language Model*, paradigma *Declarative UI Jetpack Compose*, hingga manajemen basis data luring (*SQLite/Room*).

BAB III METODOLOGI PENELITIAN

Mendeskripsikan secara konkret desain sistem dan instrumen penelitian, meliputi spesifikasi perangkat uji, pemodelan interaksi *UML*, arsitektur basis data, serta rancangan skenario evaluasi *Black-Box* dan uji komparasi komputasional *LLM*.

BAB IV HASIL DAN PEMBAHASAN

Membedah realisasi antarmuka dan *Floating Widget* di atas *OS Android*, menganalisis struktur memori *RAG*, meresolusi galat, serta menginterpretasi secara empiris hasil pengujian kecepatan dan akurasi logika kecerdasan buatan.

BAB V PENUTUP

Menarik kesimpulan akhir berdasarkan hasil uji komputasi serta merumuskan saran bagi pengembangan arsitektur dan kapabilitas *assistant virtual* di masa mendatang.