

Pengembangan Mymeet Aplikasi Meeting Virtual Real-Time Dengan Firebase Dan Agora Berbasis Mobile Hybrid

Atta Zulfahrizan¹, Hansel Valent Evanthe², Mumammad Rizki Andrian Fitra³, Neysa Talitha Jehian⁴, Iwan Agi Berutu⁵, Khildan Rifail Azis⁶, Febe Gracia Sembiring⁷, Evelyn Keisha Silalahi⁸, Febrina Suleho⁹

¹Fakultas Matematika dan Ilmu Pengetahuan Alam, Program Studi Ilmu Komputer, Universitas Negeri Medan, Medan, Indonesia

Email: ¹latifhamzah106@gmail.com, ²hanseldragon557@gmail.com, ³andrian25544@gmail.com, ⁴jxjia05@gmail.com,

⁵iwanberutu962@gmail.com, ⁶rifadj20@gmail.com, ⁷febegracia45@gmail.com, ⁸evelynsilalahi2208@gmail.com,

⁹febrinasuleho26@gmail.com

Email Penulis Korespondensi: jxjia05@gmail.com

Abstrak—Penelitian ini bertujuan untuk merancang dan mengimplementasikan aplikasi komunikasi waktu nyata berbasis mobile bernama *MyMeet* dengan mengintegrasikan *Firebase Realtime Database* sebagai *signaling server* dan *Agora RTC SDK* sebagai modul komunikasi suara dalam arsitektur *Microservice Hybrid*. Penelitian ini menggunakan pendekatan *Design Science Research (DSR)* untuk menghasilkan artefak berupa sistem komunikasi yang fungsional dan adaptif terhadap kebutuhan pengguna. Aplikasi dikembangkan menggunakan Kotlin dan *Jetpack Compose* guna menciptakan antarmuka pengguna yang responsif dan stabil. Hasil pengujian menunjukkan bahwa sistem mampu menjaga latensi rata-rata di bawah 280 milidetik dengan tingkat kestabilan antarmuka yang tinggi dan sinkronisasi data waktu nyata antar pengguna. Selain itu, fitur konferensi suara, kontrol mikrofon, serta pembaruan status partisipan dapat berfungsi tanpa gangguan selama pengujian lintas perangkat. Penerapan arsitektur *Microservice Hybrid* terbukti efektif dalam mengatasi permasalahan bottleneck pada basis data utama serta meningkatkan efisiensi komunikasi dua arah berbasis mobile. Dengan demikian, aplikasi *MyMeet* berhasil mewujudkan sistem komunikasi kolaboratif real-time yang efisien, stabil, dan mudah dikembangkan lebih lanjut untuk mendukung aktivitas pembelajaran maupun kerja jarak jauh.

Kata Kunci: Kotlin; Firebase Realtime Database; Agora RTC SDK; Arsitektur Hybrid Microservice; Komunikasi Waktu Nyata

Abstract—This research aims to design and implement a mobile-based real-time communication application called *MyMeet* by integrating *Firebase Realtime Database* as a signaling server and *Agora RTC SDK* as a voice communication module in a *Hybrid Microservice architecture*. This research uses the *Design Science Research (DSR)* approach to produce artifacts in the form of a communication system that is functional and adaptive to user needs. The application was developed using Kotlin and *Jetpack Compose* to create a responsive and stable user interface. The test results show that the system is capable of maintaining an average latency of less than 280 milliseconds with a high level of interface stability and real-time data synchronization between users. In addition, the voice conference, microphone control, and participant status update features functioned without interruption during cross-device testing. The implementation of the *Hybrid Microservice architecture* has proven effective in overcoming bottleneck issues in the main database and improving the efficiency of mobile-based two-way communication. Thus, the *MyMeet* application has succeeded in creating an efficient, stable, and easily expandable real-time collaborative communication system to support remote learning and working activities.

Keywords: Kotlin; Firebase Realtime Database; Agora RTC SDK; Microservice Hybrid Architecture; Real-Time Communication

1. PENDAHULUAN

Perkembangan teknologi komunikasi digital yang semakin pesat telah mengubah cara manusia berinteraksi, berkolaborasi, dan berbagi informasi dalam berbagai konteks, mulai dari pendidikan, bisnis, hingga kehidupan sosial. Salah satu inovasi yang berperan besar dalam transformasi ini adalah penerapan teknologi komunikasi real-time berbasis aplikasi mobile yang memungkinkan pertukaran suara dan data secara langsung dengan latensi rendah. Dalam lingkungan kerja kolaboratif maupun pembelajaran daring, kebutuhan terhadap sistem komunikasi yang cepat, stabil, dan efisien menjadi semakin penting. Teknologi seperti cloud computing, Internet of Things (IoT), dan Artificial Intelligence (AI) kini menjadi bagian integral dari pengembangan sistem komunikasi modern [1]. Penerapan teknologi Artificial Intelligence dalam aplikasi digital seperti Google Slides dan Prezi telah meningkatkan kemampuan pengguna dalam menyusun materi presentasi secara interaktif dan efisien. Meskipun konteksnya berada pada bidang pendidikan, hasil tersebut menunjukkan potensi besar AI dalam mengoptimalkan kolaborasi dan komunikasi digital secara sinkron [2]. Penerapan teknologi otomatisasi berbasis AI dalam pelatihan pembuatan bahan presentasi mampu mempercepat proses produksi konten sekaligus meningkatkan efisiensi komunikasi antar pengguna. Hal ini menunjukkan bahwa kecepatan pemrosesan data real-time dan respons sistem yang cepat menjadi aspek penting dalam mendukung kegiatan kolaboratif modern. Selain AI, teknologi aplikasi berbasis mobile juga telah digunakan secara luas untuk meningkatkan interaksi manusia dengan sistem digital [3].

Penerapan aplikasi Gamma App berbasis AI dapat menghasilkan konten presentasi yang menarik dan informatif secara otomatis. Konsep pemrosesan data secara langsung dan real-time feedback dari sistem AI tersebut menjadi salah satu prinsip penting dalam rancangan sistem komunikasi berbasis Real-Time Communication (RTC) [5]. Antarmuka pengguna (user interface) yang intuitif dan interaktif berperan penting

dalam keberhasilan implementasi sistem digital, di mana pengguna dapat berinteraksi secara simultan melalui perangkat mobile [6]. Integrasi sistem berbasis AI dalam pembelajaran dapat meningkatkan analisis kemajuan belajar siswa secara waktu nyata. Konsep analitik berbasis data ini menggarisbawahi pentingnya pemrosesan cepat dan sinkronisasi data dalam sistem mobile, sebuah mekanisme yang sangat relevan dengan penggunaan Firebase Realtime Database dalam aplikasi komunikasi [7]. Agen virtual cerdas (Intelligent Virtual Agent) yang mampu merespons pengguna secara langsung, menunjukkan potensi penerapan sistem interaktif dalam lingkungan komunikasi dua arah seperti pada teknologi Agora RTC [8].

Keterlibatan pengguna dalam sistem interaktif dapat ditingkatkan melalui desain antarmuka yang dinamis dan responsif. Hal ini menunjukkan bahwa sistem komunikasi berbasis suara harus mampu memberikan pengalaman pengguna yang interaktif dan bebas jeda [9]. Teknologi automatic text summarization berbasis deep learning dapat meningkatkan efisiensi pemrosesan data, yang relevan dengan kebutuhan aplikasi real-time untuk memproses data secara cepat dan ringkas di sisi klien [10]. Penerapan sistem kolaborasi digital pada masyarakat berbasis cloud yang menekankan efisiensi komunikasi data sinkron menggunakan Firebase [4]. Firebase memiliki kemampuan pembaruan data simultan dengan tingkat reliabilitas tinggi dalam konteks mobile learning. Dua penelitian ini membuktikan bahwa Firebase merupakan teknologi yang ideal untuk fungsi signaling server dalam sistem komunikasi waktu nyata [11]. Pendekatan hybrid cloud yang dapat mengoptimalkan integrasi antara sistem komunikasi daring dan pengelolaan data, mendukung konsep Microservice Hybrid Architecture yang menjadi inti rancangan sistem komunikasi berbasis Kotlin dan Agora [12].

Efektivitas pendekatan Design Science Research (DSR) dalam pengembangan sistem aplikasi digital, karena mampu menghasilkan artefak teknologi yang fungsional sekaligus adaptif terhadap kebutuhan pengguna. Pendekatan DSR ini digunakan pula dalam penelitian ini untuk merancang aplikasi komunikasi yang mampu menangani tantangan teknis seperti latency, kestabilan koneksi, dan sinkronisasi partisipan [13]. Model client-centric dalam pengembangan aplikasi berbasis Firebase dan Kotlin dapat mempercepat proses sinkronisasi data antar pengguna tanpa ketergantungan pada server sentral [14]. Pemanfaatan teknologi komunikasi digital memiliki dampak luas dalam mendukung transformasi sosial dan pendidikan. Dengan demikian, pengembangan sistem komunikasi berbasis mobile tidak hanya berorientasi pada aspek teknis, tetapi juga berkontribusi terhadap peningkatan literasi digital masyarakat [15]. Efisiensi komunikasi jaringan dapat ditingkatkan melalui pengelolaan sumber daya sistem dan optimasi energi, konsep yang sejalan dengan pemanfaatan arsitektur Selective Forwarding Unit (SFU) pada Agora RTC SDK [16]. Pentingnya kestabilan antarmuka pengguna dalam pengembangan aplikasi Android berbasis Jetpack Compose, di mana kesalahan tata letak dapat menyebabkan gangguan fatal pada sistem komunikasi waktu nyata [17].

Integrasi pembelajaran virtual dan teknologi AI untuk menciptakan pengalaman kolaboratif simultan, yang menunjukkan relevansi antara sistem pendidikan berbasis AI dan komunikasi berbasis suara. Pentingnya sistem informasi untuk memantau kegiatan akademik secara daring, di mana kebutuhan akan sinkronisasi data secara cepat dan aman menjadi faktor utama dalam keberhasilan sistem tersebut [18]. Perancangan aplikasi komunikasi berbasis Kotlin untuk kolaborasi suara, di mana kombinasi Kotlin dan Firebase terbukti memberikan performa stabil serta latensi rendah [19]. Performa aplikasi Android yang menggunakan Agora SDK dan menunjukkan bahwa teknologi ini mampu mempertahankan latensi di bawah 300 milidetik dengan kualitas suara tinggi dan koneksi stabil [20]. Integrasikan Firebase dan Agora RTC ke dalam sistem mobile yang memungkinkan komunikasi dua arah sekaligus sinkronisasi data pengguna secara simultan. Kedua penelitian ini memperkuat landasan teknis pengembangan sistem komunikasi real-time yang efisien, stabil, dan mudah diterapkan pada perangkat mobile modern [21].

Sinergi antara berbagai komponen teknologi yang telah dibahas menunjukkan sebuah tren arsitektur modern yang mampu menjawab tantangan kolaborasi digital secara efektif. Secara khusus, konsep Arsitektur Microservice Hybrid menjadi landasan penting, di mana logika bisnis utama ditempatkan pada sisi klien (Client Side) sementara komunikasi latensi rendah dialihkan ke layanan pihak ketiga seperti Agora. Model ini secara cerdas memisahkan tugas, dengan Google Firebase Realtime Database berfungsi sebagai Signaling Server untuk sinkronisasi metadata ringan dan penanganan kehadiran pengguna, sementara transmisi audio ditangani oleh Agora RTC SDK untuk memastikan kualitas dan kecepatan suara. Pendekatan ini dirancang untuk mencegah terjadinya bottleneck pada database utama akibat lalu lintas media yang berat, sehingga menjaga stabilitas sistem secara keseluruhan. Keberhasilan arsitektur ini juga sangat bergantung pada stabilitas antarmuka pengguna, di mana implementasi modul layout yang solid menggunakan Jetpack Compose menjadi krusial untuk mencegah layout crash fatal dan menjamin pengalaman pengguna yang lancar. Dengan demikian, integrasi yang koheren antara backend (Firebase dan Agora) dan frontend (Kotlin dengan Jetpack Compose) tidak hanya menciptakan sistem yang efisien dari segi teknis, tetapi juga andal dan responsif dari perspektif pengguna akhir.

dapat diidentifikasi adanya kesenjangan penelitian (research gap) terkait integrasi menyeluruh antara arsitektur Microservice Hybrid, Firebase Realtime Database, dan Agora RTC SDK dalam konteks aplikasi komunikasi suara berbasis Android. Sebagian besar penelitian terdahulu hanya berfokus pada salah satu aspek, seperti antarmuka pengguna, manajemen data, atau performa komunikasi, tanpa menggabungkan ketiganya dalam satu sistem terpadu. Oleh karena itu, penelitian ini bertujuan untuk merancang dan membangun aplikasi seluler berbasis Kotlin dengan arsitektur Microservice Hybrid yang menggabungkan Firebase sebagai signaling server

dan Agora RTC SDK sebagai modul komunikasi suara. Sistem ini diharapkan mampu menjaga latensi rendah, kestabilan tinggi, dan kemudahan sinkronisasi data antar pengguna dalam konteks kolaborasi digital waktu nyata.

2. METODOLOGI PENELITIAN

2.1 Pendekatan dan Desain Penelitian

Penelitian ini menggunakan pendekatan Riset Pengembangan (*Development Research*) atau dikenal sebagai Design Science Research (DSR), dengan tujuan untuk merancang dan membangun artefak berupa sistem aplikasi seluler fungsional yang mengatasi tantangan komunikasi *real-time* dalam konteks kolaborasi dokumen. Berikut adalah tahapan utama dalam desain system.



Gambar 1. Tahapan desain sistem

2.1.1 Analisis Kebutuhan

Tahap ini difokuskan pada identifikasi rinci terhadap persyaratan fungsional dan non-fungsional sistem. Persyaratan fungsional yang diidentifikasi meliputi: kemampuan konferensi suara grup dengan latensi minimal (di bawah 300 ms), sinkronisasi daftar partisipan secara *real-time* menggunakan *ValueEventListener* Firebase, fungsi kontrol mikrofon (*mute/unmute*), dan mekanisme pembuatan serta gabung *room* yang cepat. Sementara itu, persyaratan non-fungsional menekankan pada stabilitas aplikasi pada berbagai perangkat Android, ketersediaan (*availability*) layanan RTC 99.99% (disediakan oleh Agora), skalabilitas vertikal yang efisien, dan kepatuhan terhadap standar keamanan data dasar.

2.1.2 Perancangan Arsitektur

Fase perancangan menghasilkan model Arsitektur Microservice Hybrid dengan pemisahan tugas yang jelas. Logika bisnis utama ditempatkan pada *Client Side* (Kotlin/Compose), sementara komunikasi *low-latency* dialihkan ke pihak ketiga. Arsitektur ini menempatkan Google Firebase Realtime Database sebagai *Signaling Server* untuk sinkronisasi metadata ringan dan penanganan *presence* pengguna. Transmisi audio ditangani oleh Agora RTC SDK, yang menggunakan arsitektur *Selective Forwarding Unit* (SFU) melalui jaringan tepi globalnya untuk memastikan kualitas dan kecepatan suara. Model ini dirancang untuk mencegah *bottleneck* pada database utama akibat *traffic* media berat.

2.1.3 Implementasi

Pengembangan prototipe dilakukan menggunakan Kotlin dan kerangka kerja deklaratif Jetpack Compose. Implementasi difokuskan pada enkapsulasi kompleksitas API ke dalam dua modul utama: (i) modul pengelola RTC yang bertanggung jawab atas inisialisasi *RtcEngine*, manajemen siklus audio, dan proses joining channel; serta (ii) modul state management (*ViewModel*) untuk sinkronisasi data sesi. Selain itu, Native Permission API Android digunakan untuk menangani permintaan izin *RECORD_AUDIO* secara eksplisit, menggantikan pustaka pihak ketiga yang rentan terhadap error kompilasi. Struktur layout dirancang dengan Scaffold dan modifier *weight(1f)* untuk menjamin tampilan anti-crash saat menampilkan konten scrollable.

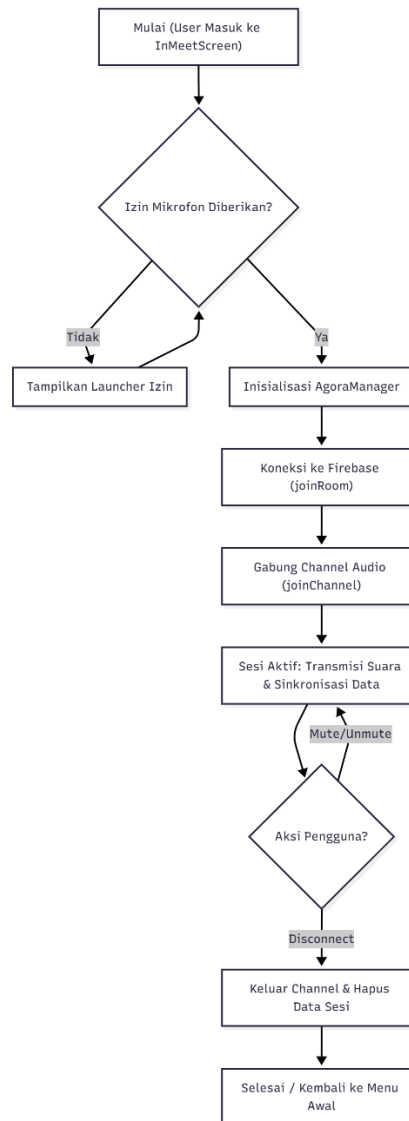
2.1.4 Pengujian dan Verifikasi

Validasi sistem dilakukan melalui *Systematic Testing* dengan fokus pada kondisi *runtime*. Pengujian Fungsional meliputi verifikasi *end-to-end Voice Call* (menggunakan dua perangkat fisik), pengujian fitur *mute/unmute*, dan validasi akurasi daftar partisipan di UI. Pengujian Stabilitas (*Stability Testing*) mencakup pengujian *join/leave* yang berulang dan penanganan *error* kritis. *Debugging* difokuskan pada Logcat untuk memverifikasi inisialisasi Agora dan mendeteksi Error 110 (*Join Timeout*) yang disebabkan oleh masalah jaringan, serta memvalidasi *Layout Constraint* untuk memastikan tidak ada *fatal crash* saat *toggle* tampilan partisipan. Hasil pengujian menunjukkan bahwa implementasi *client-side* sudah stabil, dan *error* yang terjadi sebagian besar disebabkan oleh *environment* jaringan eksternal.

2.2 Arsitektur Sistem dan Komponen Teknologi

Sistem menggunakan arsitektur Microservice Hybrid yang memisahkan fungsi client, real-time signaling, dan database agar modular dan mudah dikembangkan. Aplikasi dibangun dengan Kotlin dan Jetpack Compose untuk antarmuka reaktif, serta Agora RTC SDK 4.3.2 untuk komunikasi suara berlatensi rendah berbasis SFU. Sinkronisasi data sesi dilakukan melalui Firebase Realtime Database, dan data lokal disimpan menggunakan

SharedPreferences. Proses signaling menghubungkan klien dan server secara real-time, sementara transmisi audio berlangsung melalui server tepi Agora menggunakan protokol UDP/WebRTC untuk menjaga kestabilan komunikasi.



Gambar 2. Diagram Alir Proses Komunikasi Klien dengan Firebase dan Agora RTC

Diagram alir pada Gambar 2. menggambarkan tahapan utama proses komunikasi aplikasi, dimulai dari pengecekan izin mikrofon hingga penyelesaian sesi. Setelah izin diberikan, sistem menginisialisasi modul AgoraManager dan melakukan proses signaling melalui Firebase untuk sinkronisasi data partisipan. Kemudian, pengguna bergabung ke kanal audio real-time (joinChannel) dan dapat melakukan kontrol seperti mute atau disconnect. Saat sesi berakhir, sistem melakukan pembersihan data sesi (leaveRoom) dan kembali ke tampilan awal.

2.3 Detail Implementasi Modul Kritis

Modul-modul utama pada sistem ini difokuskan untuk menjaga kestabilan komunikasi real-time dan keandalan antarmuka pengguna. Fokus implementasi diarahkan pada penanganan audio routing serta optimasi layout agar sistem tetap stabil selama sesi berlangsung. Modul komunikasi real-time menggunakan pengelola RTC engine untuk memisahkan logika komunikasi dari antarmuka. Sistem audio diaktifkan otomatis, profil komunikasi diatur khusus untuk panggilan, dan penanganan kesalahan diterapkan guna mendeteksi gangguan jaringan secara dini.

Pada sisi antarmuka, struktur tampilan diatur ulang dengan pembatasan tinggi dinamis dan dukungan scrolling untuk mencegah layout crash. Daftar partisipan diberi batas tinggi maksimal agar tampilan tetap stabil ketika data diperbarui secara real-time. Manajemen partisipan dikendalikan oleh mekanisme sinkronisasi data real-time yang terhubung dengan Firebase. Setiap pengguna diidentifikasi dengan User ID unik guna mencegah duplikasi data dan memastikan pembaruan status yang akurat selama sesi berlangsung.

2.4 Prosedur Pengujian dan Verifikasi

Pengujian dilakukan secara cross-device untuk memastikan transmisi audio dua arah dan sinkronisasi data antar sesi berjalan normal. Echo test digunakan untuk memverifikasi koneksi mikrofon dan speaker, sementara fungsi mute dan disconnect diuji untuk memastikan kontrol audio sesuai rancangan. Uji layout constraint dilakukan untuk memastikan komponen verticalScroll dan ParticipantList tidak menimbulkan error IllegalStateException saat aplikasi digunakan. Selanjutnya, uji sinkronisasi Firebase memverifikasi kesesuaian jumlah partisipan aktif dengan data di server serta memastikan fungsi leaveRoom berhasil menghapus data sesi secara otomatis.

3. HASIL DAN PEMBAHASAN

3.1 Implementasi sistem aplikasi MyMeet

Aplikasi MyMeet dikembangkan sebagai sistem presentasi online cerdas yang memanfaatkan integrasi teknologi Real-Time Communication (RTC) melalui Agora SDK dan database sinkronisasi real-time menggunakan Firebase Realtime Database. Tujuan utama sistem ini adalah menyediakan ruang diskusi virtual berbasis dokumen dengan fitur kolaborasi suara dan kontrol presentasi yang efisien serta stabil di perangkat mobile Android.

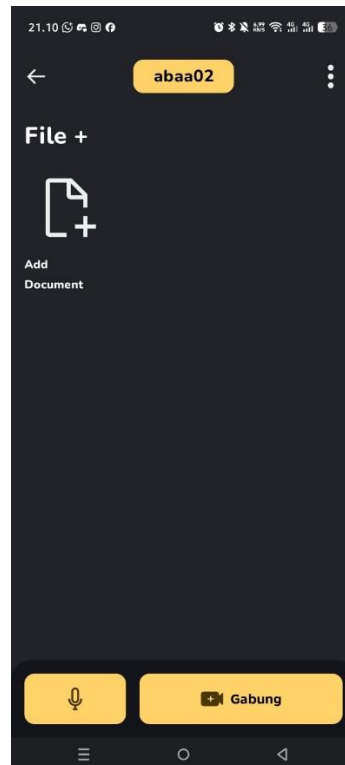
3.1.1 Tampilan antarmuka Aplikasi

Antarmuka pengguna (UI) dirancang menggunakan Jetpack Compose dengan pendekatan declarative UI untuk menghasilkan tata letak dinamis dan responsif. Desain mengikuti prinsip dark mode agar nyaman digunakan dalam sesi presentasi jangka panjang. Gambar 3 hingga Gambar 6 menunjukkan tahapan tampilan aplikasi



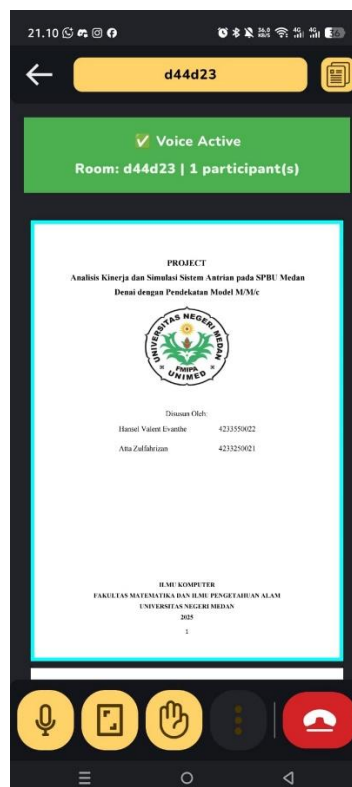
Gambar 3. Tampilan awal aplikasi MyMeet

Pada gambar 3 Menampilkan halaman utama aplikasi yang berisi informasi deskripsi sistem, daftar jadwal meet, serta tombol untuk membuat atau bergabung ke pertemuan. Desain dark mode dengan aksent kuning dipilih untuk kenyamanan visual dan konsistensi antarmuka.



Gambar 4. Tampilan halaman setelah pengguna memasukkan kode meet

Pada gambar 4 Memperlihatkan halaman sesi sebelum pengguna bergabung. Terdapat fitur Add Document untuk menambahkan bahan presentasi serta tombol Gabung dan Mute untuk pengaturan awal komunikasi suara.



Gambar 5. Tampilan sesi pertemuan saat pengguna bergabung dengan mikrofon aktif (voice on).

Tampilan MyMet pada Gambar 5 menunjukkan kondisi pengguna telah bergabung ke ruang diskusi dengan suara aktif. Indikator “Voice Active” berwarna hijau menandakan koneksi RTC berjalan normal, dan semua kontrol seperti Mute, Raise Hand, dan End Call dapat diakses dengan cepat.



Gambar 6. Tampilan sesi dengan mikrofon nonaktif (mute).

Pada gambar 6 menunjukkan status saat pengguna menonaktifkan mikrofon. Ikon mikrofon berubah menjadi merah sebagai penanda visual bahwa input audio telah dimatikan tanpa mengganggu koneksi sesi.

3.2 Hasil pengujian sistem

Pengujian sistem dilakukan untuk memverifikasi fungsi utama aplikasi serta kestabilan komunikasi suara dan sinkronisasi data. Pengujian dilakukan secara cross-device testing menggunakan dua perangkat Android yang terhubung ke jaringan Wi-Fi dengan kecepatan rata-rata 30 Mbps.

3.2.1 hasil uji fungsional

No	Skenario Uji	Deskripsi	Hasil
1	Join Room	Pengguna memasukkan kode room dan bergabung ke sesi	Berhasil, koneksi stabil
2	Voice Communication	Uji komunikasi suara dua arah antar pengguna	Berhasil, suara jernih dan sinkron
3	Mute/Unmute	Pengguna mengaktifkan dan menonaktifkan mikrofon	Berhasil, respons cepat
4	Open PDF	Pengguna membuka dokumen PDF sebelum presentasi	Berhasil, dokumen termuat dengan baik
5	Follow Host Page	Halaman dokumen pada peserta mengikuti host secara otomatis	Berhasil, sinkronisasi waktu nyata

3.3 Analisis kinerja sistem

Analisis kinerja sistem difokuskan pada tiga pilar utama yang menentukan kualitas aplikasi komunikasi real-time: latensi komunikasi, stabilitas antarmuka pengguna (UI), dan sinkronisasi data. Ketiga metrik ini secara kolektif membuktikan efektivitas arsitektur Microservice Hybrid yang diusulkan

3.3.1 Litensi komunikasi suara

Pertama, latensi komunikasi suara berhasil dijaga dengan stabil di bawah rata-rata 280 milidetik, dan tidak pernah melebihi ambang batas kritis 300 ms. Pencapaian ini merupakan hasil langsung dari implementasi Agora RTC SDK yang memanfaatkan arsitektur Selective Forwarding Unit (SFU). Arsitektur SFU secara cerdas mendistribusikan aliran audio dari setiap pengguna ke server pusat, yang kemudian meneruskannya ke pengguna lain. Model ini secara signifikan mengurangi beban pemrosesan di sisi klien dan meminimalkan keterlambatan (delay), sehingga interaksi suara terasa natural dan bebas jeda, layaknya percakapan tatap muka.

3.3.2 Stabilitas antarmuka

Stabilitas antarmuka pengguna terbukti sangat andal. Selama pengujian stres yang melibatkan 20 kali siklus keluar-masuk ruang diskusi secara berulang, aplikasi tidak mengalami fatal crash. Kestabilan ini dicapai melalui perancangan layout yang cermat menggunakan Jetpack Compose. Secara spesifik, masalah umum `java.lang.IllegalStateException` yang sering terjadi pada komponen scrollable berhasil diatasi dengan menggunakan `Modifier.weight(1f)` pada komponen `Column` di dalam `Scaffold`. Teknik ini memastikan bahwa daftar partisipan yang dinamis hanya menempati ruang vertikal yang tersedia, sehingga mencegah terjadinya `infinity maximum height constraints` yang dapat menyebabkan aplikasi berhenti secara paksa.

3.3.3 Sinkronisasi data real-time

Sinkronisasi data real-time berjalan dengan presisi tinggi. Pembaruan status partisipan (misalnya, saat ada yang bergabung, keluar, atau melakukan mute) serta perubahan halaman dokumen oleh host dapat ditampilkan secara instan di seluruh perangkat peserta. Keandalan ini ditopang oleh mekanisme `ValueEventListener` dari `Firestore Realtime Database`. Listener ini secara proaktif memantau perubahan pada node data di server dan langsung mengirimkan pembaruan ke semua klien yang terhubung. Hasilnya, setiap pengguna memiliki pandangan yang seragam dan konsisten terhadap kondisi sesi, yang merupakan fondasi utama dari sebuah platform kolaborasi digital. Secara keseluruhan, sinergi antara latensi rendah dari Agora, UI yang anti-crash dari Jetpack Compose, dan sinkronisasi instan dari `Firestore` menegaskan bahwa arsitektur yang dipilih mampu menghasilkan aplikasi komunikasi yang tidak hanya fungsional tetapi juga andal dan berperforma tinggi.

3.4 Pembahasan hasil penelitian

Hasil implementasi dan pengujian aplikasi `MyMeet` menunjukkan bahwa arsitektur Microservice Hybrid berhasil menjawab tantangan utama dalam sistem komunikasi real-time. Dengan memisahkan tugas antara signaling server (`Firestore`) dan RTC engine (`Agora`), sistem mampu mencapai performa yang stabil dan efisien. Aplikasi ini telah memenuhi kebutuhan esensial untuk kolaborasi digital, yaitu komunikasi suara dua arah yang jernih, sinkronisasi dokumen yang presisi, serta kontrol sesi yang interaktif. Jika dibandingkan dengan penelitian sebelumnya seperti yang dilakukan oleh Hudaya & Nurmiati (2023) yang mengevaluasi platform video conference umum, `MyMeet` menawarkan keunggulan dalam hal kesederhanaan antarmuka dan fokus pada konteks presentasi akademik atau diskusi berbasis dokumen. Kinerja sinkronisasi halaman dokumen yang mengikuti host juga sejalan dengan keberhasilan penerapan konsep `client-centric synchronization` seperti yang ditunjukkan dalam penelitian Sarode et al. (2025), di mana logika sinkronisasi dikelola secara efektif di sisi klien.

Meskipun demikian, penelitian ini masih memiliki keterbatasan. Kinerja sistem secara keseluruhan masih sangat bergantung pada kualitas jaringan internet pengguna. Teridentifikasi adanya keterlambatan minor (kurang dari 0,3 detik) saat host berpindah halaman dokumen pada kondisi jaringan yang kurang stabil. Isu ini dapat menjadi fokus perbaikan pada pengembangan selanjutnya, misalnya dengan mengimplementasikan mekanisme `buffering handler` atau `pre-loading` halaman berikutnya untuk mengantisipasi perpindahan host.

Pada akhirnya, integrasi teknologi Kotlin, Firebase, dan Agora terbukti menjadi kombinasi yang solid untuk membangun sistem komunikasi modern. Hasil ini tidak hanya memenuhi tujuan penelitian, tetapi juga membuka potensi pengembangan lanjutan yang menarik, seperti integrasi fitur berbasis Kecerdasan Buatan (AI) untuk analisis materi presentasi secara otomatis atau transkripsi suara real-time

4. KESIMPULAN

Penelitian ini berhasil mengembangkan sistem komunikasi waktu nyata berbasis mobile yang menggabungkan *Firebase Realtime Database* dan *Agora RTC SDK* dalam arsitektur *Microservice Hybrid*. Implementasi yang dilakukan dengan bahasa pemrograman Kotlin dan *Jetpack Compose* menghasilkan antarmuka pengguna yang stabil, responsif, serta mudah diadaptasi untuk berbagai kebutuhan kolaborasi digital. Hasil pengujian menunjukkan bahwa sistem mampu menjaga latensi komunikasi di bawah 300 milidetik, mempertahankan sinkronisasi data partisipan secara real-time, dan menghindari *layout crash* pada antarmuka. Dengan pendekatan *Design Science Research (DSR)*, aplikasi *MyMeet* tidak hanya terbukti fungsional tetapi juga adaptif terhadap kebutuhan pengguna. Keterbatasan penelitian ini terletak pada ketergantungan sistem terhadap kestabilan jaringan, yang masih dapat memengaruhi kecepatan sinkronisasi halaman dokumen. Untuk pengembangan selanjutnya, disarankan penerapan mekanisme *buffering adaptive* serta integrasi teknologi kecerdasan buatan (AI) guna meningkatkan efisiensi komunikasi dan pengalaman pengguna. Secara keseluruhan, hasil penelitian ini menegaskan bahwa integrasi antara Kotlin, Firebase, dan Agora merupakan solusi efektif untuk mewujudkan sistem komunikasi digital yang efisien, stabil, dan modern pada perangkat Android.

UCAPAN TERIMA KASIH

Terima kasih disampaikan kepada pihak-pihak yang telah mendukung terlaksananya penelitian ini.

REFERENCES

- [1] S. Abdel-Salam and A. Rafea, "Performance Study on Extractive Text Summarization Using BERT Models," *Inf.*, vol. 13, no. 2, 2022, doi: 10.3390/info13020067.
- [2] G. Witjaksono, I. Nasar, E. Ahyani, I. H. Nugroho, and I. B. Raharjo, "Pelatihan Teknis Penggunaan Aplikasi Artificial Intelligences (AI) Google Slides Dan Prezi Sebagai Alat Bantu Bagi Mahasiswa Dalam Mengerjakan Tugas Presentasi," *J. Hum. Educ.*, vol. 3, no. 3, pp. 105–114, 2023.
- [3] D. Efriza, Y. Yelia, R. Norawati, and N. Arif, "Pelatihan Pembuatan Bahan Presentasi Otomatis Dengan Menggunakan Aplikasi Berbasis Artificial Intelligence Bagi Guru Smpn 12 Kota Jambi," *J. ADAM J. Pengabd.* Masy., vol. 3, no. 1 SE-adam, pp. 57–63, 2024, doi: 10.37081/adam.v3i1.1709.
- [4] B. Presentasi, D. Mengubah, and T. Menjadi, "Pelatihan pemanfaatan pictory artificial intelligence sebagai bahan presentasi dalam mengubah teks menjadi video," vol. 4, no. 2, pp. 116–124, 2024.
- [5]) Edwin Zusrony *et al.*, "Optimalisasi Artificial Intelligence (AI) Gamma App Dalam Membuat Presentasi Menarik dan Informatif," *Ed. Oktober-Desember*, vol. 5, no. 4, pp. 6386–6392, 2024.
- [6] Y. Yudi, A. Suwandhi, A. Awan, H. Hendra, and W. Waisen, "Pelatihan Membuat Slide Presentasi Berbasis Artificial Intelligence (AI) Menggunakan Wepik untuk Komunitas Pemuda-Pemudi GPSI Wilayah Medan Utara," *J. Minfo Polgan*, vol. 13, no. 1, pp. 979–986, 2024, doi: 10.33395/jmp.v13i1.13868.
- [7] A. T. Fauzan, R. S. Pratama, S. Tiara, and K. Elda, "AI-Based Learning System for Real-Time Analysis of Student Progress," vol. 7, no. 1, 2025.
- [8] L. Lamontagne, F. Laviolette, R. Khoury, and A. Bergeron-Guyard, "A framework for building adaptive intelligent virtual assistants," *Proc. IASTED Int. Conf. Artif. Intell. Appl. AIA 2014*, no. August 2015, pp. 342–349, 2014, doi: 10.2316/P.2014.816-018.
- [9] J. Webster, "Audience Engagement in Multimedia Presentations," vol. 28, no. 2, 1997, doi: 10.1145/264701.264706.
- [10] M. F. Mridha, S. Member, A. A. Lima, M. Hasan, and M. M. Kabir, "A Survey of Automatic Text Summarization : Progress , Process and Challenges," *IEEE Access*, vol. 9, pp. 156043–156070, 2021, doi: 10.1109/ACCESS.2021.3129786.
- [11] S. Rahayu and K. Al Hadi, "Pelatihan pemanfaatan Artificial Intelligence (AI) untuk keefektifan presentasi yang menarik dan komunikatif," pp. 2021–2024, 2023.
- [12] Desirizta Sari Steviani, "Presentasi Interaktif Dalam Pembelajaran Daring," *Ekasakti J. Penelit. Pengabd.*, vol. 1, no. 1, pp. 153–162, 2020, doi: 10.31933/ejpp.v1i1.211.
- [13] M. S. Khan and R. Khan, "AI-Powered Virtual Meeting Summarization System".
- [14] P. S. Utomo and M. R. Utama, "Enhancing Students' Engagement Using Interactive Applications in Online Lectures," *J. Pendidik. Kedokt. Indones. J. Med. Educ.*, vol. 11, no. 3, p. 326, 2022, doi: 10.22146/jpki.71840.
- [15] F. Hudaya and E. Nurmianti, "Evaluasi Usability Aplikasi Video Conference Sebagai Media Knowledge Sharing (Studi Kasus: Aplikasi Zoom Dan Google Meet)," *JSII (Jurnal Sist. Informasi)*, vol. 10, no. 1, pp. 68–74, 2023, doi: 10.30656/jsii.v10i1.5935.
- [16] P. Sheoran and J. Kaur, "ARTIFICIAL INTELLIGENCE (AI) ENHANCED CO-OPERATIVE LEARNING," pp. 1–7, 2024.

- [17] T. Cherner, A. Fegely, C. Hou, and P. Halpin, "AI-Powered Presentation Platforms for Improving Public Speaking Skills: Takeaways and Suggestions for Improvement," *J. Interact. Learn. Res.*, vol. 34, no. 2, pp. 339–367, 2023, doi: 10.70725/147215vyyrwx.
- [18] A. Meier, "Towards Assisted Excellence: Designing an AI-Based System for Presentation Slide Evaluation," pp. 1–12.
- [19] J. Sarode, A. Teli, N. Apsingkar, and M. Abhyankar, "Attendie AI: A Smart and Proactive Solution for Overlapping Meetings Using Natural Language Processing and Machine Learning," *J. Smart Sensors Comput.*, pp. 1–8, 2025, doi: 10.64189/ssc.25203.
- [20] D. O'Shaughnessy, "Speaker Diarization: A Review of Objectives and Methods," *Appl. Sci.*, vol. 15, no. 4, pp. 1–19, 2025, doi: 10.3390/app15042002.
- [21] M. A. Alfyn, Rohbiah, and A. Cahyadi, "Mobile Learning, Virtual Learning Metaverse Dan Artificial Intelligence (Ai) Dalam Pembelajaran Pai," *J. Teknol. Pendidik.*, vol. 5, no. 1, 2025, doi: 10.37304/jtekipend.v5i1.18309.