EXPLORER
Journal of Computer Science and Information Technology
E-ISSN 2774-4647
Website: https://journal.fkpt.org/index.php/Explorer DOI: https://doi.org/10.47065/explorer.v6i2.3013

Design Science Research dalam Perancangan Sistem Otomasi Notulen Rapat Berbasis Large Language Model dengan Mekanisme Human-in-the-Loop

Yoshida Sary[1*], Al-Khowarizmi [2], Martiano [3], Firahmi Rizky[4]
[1*,4] Sistem Informasi, Universitas Muhammadiyah Sumatera Utara, Medan, Indonesia
[2]Teknologi Informasi, Universitas Muhammadiyah Sumatera Utara, Medan, Indonesia
[3] Sains Data, Universitas Muhammadiyah Sumatera Utara, Medan, Indonesia
[1*]yoshidasary@umsu.ac.id, [2]alkhowarizmi@umsu.ac.id, [3]martiano@umsu.ac.id, [4]firahmirizky@umsu.ac.id
[*)] yoshidasary@umsu.ac.id

Abstrak-Dokumentasi hasil rapat sering tidak dilakukan secara konsisten, menyebabkan organisasi kehilangan jejak proses pengambilan keputusan dan lemahnya akuntabilitas tindak lanjut. Penelitian terdahulu mengenai otomasi peringkasan rapat berbasis *large language model* umumnya berorientasi pada kualitas teknis ringkasan, tanpa mempertimbangkan verifikasi manusia maupun representasi proses pengambilan keputusan dalam notulen. Penelitian ini bertujuan merancang, mengimplementasikan, dan mengevaluasi arsitektur sistem otomasi notulen rapat yang menyisipkan dua titik verifikasi manusia, pada tahap transkripsi dan peringkasan, menggunakan pendekatan Design Science Research. Arsitektur dirancang berbasis *client-side* integration yang mengintegrasikan layanan *speech-to-text* dan large language model secara langsung dari sisi klien tanpa infrastruktur server tambahan, melalui tujuh iterasi perancangan dan penyempurnaan instruksi peringkasan dari struktur keluaran tetap menjadi format Markdown yang mempertahankan jejak usulan dan Keputusan. Demonstrasi pada audio rapat nyata berdurasi sekitar sembilan puluh menit menunjukkan mekanisme pemotongan audio otomatis berhasil mengatasi keterbatasan ukuran berkas, dengan peningkatan ukuran berkas 44% hanya berdampak pada peningkatan waktu pemrosesan 29%. Penelitian ini berkontribusi berupa model arsitektur notulen rapat yang dapat diadaptasi organisasi dengan kapasitas teknologi informasi terbatas, sekaligus menegaskan pentingnya verifikasi manusia dalam pemanfaatan large language model untuk dokumen yang menjadi bukti proses pengambilan keputusan organisasi.

Kata Kunci: Notulen Rapat, *Large Language Model*, Human-in-the-Loop, Design Science Research, *Client-Side Integration*

Abstract-Meeting documentation is often inconsistently produced, causing organizations to lose track of decision-making processes and weakening follow-up accountability. Prior research on large language model-based meeting summarization automation has generally focused on the technical quality of summaries, without considering human verification or the representation of decision-making processes in the resulting minutes. This research aims to design, implement, and evaluate a meeting minutes automation system architecture incorporating two human verification points, at the transcription and summarization stages, using a Design Science Research approach. The architecture is based on client-side integration, directly integrating speech-to-text and large language model services from the client side without additional server infrastructure, developed through seven design iterations and refinement of summarization instructions from a fixed output structure to a Markdown format preserving the trace of proposals and decisions. Demonstration on real meeting audio lasting approximately ninety minutes showed that automatic audio segmentation successfully addressed file size limitations, with a 44% increase in file size resulting in only a 29% increase in processing time. This research contributes a meeting minutes architecture model adaptable by organizations with limited information technology capacity, while affirming the importance of human verification in using large language models for documents that serve as evidence of organizational decision-making.

Keywords: Meeting Minutes, Large Language Model, Human-in-the-Loop, Design Science Research, Client-Side Integration

1. PENDAHULUAN

Rapat merupakan instrumen koordinasi yang fundamental dalam organisasi, baik pemerintah, swasta, maupun pendidikan tinggi, karena menjadi forum pengambilan keputusan yang berdampak pada arah kebijakan dan operasional organisasi. Rapat di tempat kerja yang sudah lazim dilakukan mendukung koordinasi, kolaborasi, pengambilan keputusan, distribusi informasi, dan interaksi kolegial di antara tim dan organisasi, dan dipandang sebagai elemen krusial bagi keberhasilan pemimpin dan organisasi secara lebih luas [1], [2].

EXPLORER
Journal of Computer Science and Information Technology
E-ISSN 2774-4647
Website: https://journal.fkpt.org/index.php/Explorer **DOI:** https://doi.org/10.47065/explorer.v6i2.3013

Dokumentasi hasil rapat dalam bentuk notulen juga dipakai sebagai rujukan hukum, historis, dan operasional ketika keputusan lama perlu ditelusuri, dipertanyakan, atau dievaluasi ulang. [3]. Namun, dalam praktiknya, masih banyak rapat yang tidak terdokumentasi sama sekali, bukan sekadar terdokumentasi dengan kualitas yang rendah. Observasi awal pada studi kasus penelitian ini menunjukkan bahwa ketiadaan notulis pada sebagian besar rapat menyebabkan tidak adanya catatan keputusan maupun tindak lanjut yang dapat diakses kembali oleh peserta rapat. Pihak eksternal pun cenderung hanya menerima dokumentasi pada rapat yang dianggap rutin dan penting, sementara rapat dengan agenda baru, yang justru sering membutuhkan tindak lanjut paling jelas, lebih sering tidak terdokumentasi. Kondisi ini berakibat pada hilangnya jejak keputusan, lemahnya akuntabilitas tindak lanjut, dan kesulitan organisasi dalam melakukan audit atau evaluasi terhadap proses pengambilan keputusan yang telah berlangsung.

Perkembangan *Large Language Model* (LLM) membuka peluang otomasi pencatatan rapat melalui kemampuannya memahami bahasa natural dan menghasilkan ringkasan yang koheren. Pemanfaatan korpus teks yang luas memungkinkan LLM menangkap pola linguistik kompleks, hubungan semantik, dan konteks secara mendalam, sehingga mampu menghasilkan ringkasan berkualitas tinggi yang menyaingi ringkasan buatan manusia [4]. Sejumlah penelitian terdahulu telah mengeksplorasi pemanfaatan LLM untuk peringkasan rapat. Penelitian oleh Asthana dan rekan-rekannya merancang dua rancangan tampilan recap rapat berbasis LLM, yaitu *highlights* dan *minutes* hierarkis terstruktur, kemudian mengevaluasi efektivitas keduanya melalui wawancara semi-terstruktur dengan tujuh pengguna dalam konteks rapat kerja mereka sendiri. Temuan penelitian tersebut lebih menitikberatkan pada pengalaman pengguna (UX) terhadap hasil recap dibanding kedalaman arsitektur teknis sistem [5]. Pada sisi algoritmik, penelitian lain mengusulkan kerangka peringkasan rapat yang sadar-agenda secara dinamis. Berbeda dari pendekatan pada umumnya yang menghasilkan ringkasan setelah rapat selesai, kerangka tersebut dirancang untuk menghasilkan ringkasan secara inkremental dan koheren secara *real-time* selama rapat masih berlangsung, dengan memanfaatkan agenda yang dapat terus diperbarui sebagai pemandu fokus ringkasan seiring pergeseran topik dalam dialog multi-peserta yang panjang [6]. Penelitian sejenis lainnya berfokus pada rangkaian teknik *natural language processing*, mulai dari speaker diarization, ekstraksi kata kunci, sentiment analysis, hingga abstractive summarization, untuk menghasilkan notulen [7].

Pada sisi penerapan praktis di lingkungan organisasi, penelitian mengenai abstractive summarization rapat menunjukkan bahwa model berbasis encoder-decoder maupun LLM dapat menghasilkan ringkasan terstruktur yang mengaitkan topik diskusi dengan pembicara apabila disebutkan secara eksplisit dalam transkrip, namun model dengan kapasitas lebih kecil cenderung menghasilkan keluaran yang kurang koheren untuk transkrip dengan volume besar [8]. Temuan serupa pada penerapan industri menunjukkan bahwa pemilihan LLM untuk peringkasan rapat secara nyata melibatkan *trade-off* antara kualitas keluaran, biaya operasional, dan kebutuhan privasi data [9], [10].

Ketiga kelompok penelitian tersebut menunjukkan kemajuan signifikan pada sisi pemrosesan bahasa alami dan pengalaman pengguna, namun memiliki kesamaan orientasi. Penelitian dipusatkan pada bagaimana sistem menghasilkan ringkasan yang baik secara teknis, bukan pada bagaimana sistem tersebut diarsitektur agar dapat diintegrasikan ke dalam alur kerja organisasi secara menyeluruh, termasuk mekanisme verifikasi manusia sebelum dan sesudah proses peringkasan otomatis. Padahal, keberhasilan adopsi suatu sistem otomasi tidak hanya ditentukan oleh akurasi algoritmanya, tetapi juga oleh bagaimana sistem tersebut dirancang agar tetap melibatkan pengawasan dan keterlibatan manusia pada berbagai tahap proses, khususnya untuk aplikasi yang berdampak langsung pada pengambilan [11]. Keluaran model generatif, meski tampak koheren dan masuk akal, tidak menjamin akurasi faktual, sehingga keluaran tersebut perlu diperlakukan sebagai informasi yang harus diverifikasi, bukan teks final yang dapat langsung dipercaya [11] . Kebutuhan akan verifikasi ini menjadi semakin relevan ketika sistem bergantung pada layanan *speech-to-text* yang belum memiliki *benchmark* akurasi resmi untuk Bahasa Indonesia. Studi pada bahasa daerah Indonesia yang memiliki karakteristik linguistik serumpun menunjukkan bahwa model Whisper dapat mengalami tingkat kesalahan kata yang signifikan tanpa penyesuaian khusus, dan performanya dapat menurun tajam pada kondisi akustik yang kurang ideal seperti rekaman dengan derau latar [12], [13].

Pada sisi arsitektur sistem, penelitian terdahulu menunjukkan beragam pendekatan integrasi LLM ke dalam alur kerja otomatis, mulai dari penggunaan platform orkestrasi *workflow* hingga pemanggilan API model bahasa secara langsung dari sisi klien. Pendekatan berbasis *platform* orkestrasi memberikan keunggulan pada kemampuan visualisasi alur proses dan dukungan konektor bawaan untuk berbagai layanan eksternal, namun memerlukan infrastruktur tambahan berupa server yang harus disediakan dan dikelola secara berkelanjutan [14]. Sebaliknya, pendekatan pemanggilan API langsung dari sisi klien (*client-side integration*) menawarkan kesederhanaan implementasi tanpa kebutuhan infrastruktur server tambahan, sebuah karakteristik yang relevan bagi organisasi dengan kapasitas pengelolaan teknologi informasi yang terbatas, sekalipun dengan trade-off berupa keterbatasan dalam orkestrasi proses yang kompleks dan ketergantungan pada keterbukaan sesi peramban (*browser*) selama proses berlangsung [15].

EXPLORER
Journal of Computer Science and Information Technology
E-ISSN 2774-4647
Website: https://journal.fkpt.org/index.php/Explorer DOI: https://doi.org/10.47065/explorer.v6i2.3013

Berdasarkan uraian tersebut, penelitian ini diposisikan untuk mengisi kesenjangan antara penelitian peringkasan rapat berbasis LLM yang berorientasi teknis-algoritmik dan kebutuhan praktis organisasi akan sistem yang sederhana, dapat diimplementasikan tanpa infrastruktur server tambahan, namun tetap mempertahankan kontrol manusia pada titik-titik kritis proses. Penelitian ini mengadopsi pendekatan Design Science Research (DSR) [16] untuk merancang, mendemonstrasikan, dan mengevaluasi arsitektur sistem otomasi notulen rapat yang mengintegrasikan layanan *speech-to-text* dan *large language model* secara langsung dari sisi klien, dengan menyisipkan dua titik verifikasi manusia: verifikasi transkripsi sebelum peringkasan, dan verifikasi hasil notulen sebelum publikasi. Mengacu pada kerangka kontribusi pengetahuan Design Science Research, penelitian ini diposisikan sebagai kontribusi improvement, yaitu solusi baru untuk masalah yang telah dikenal dengan memanfaatkan kombinasi pendekatan arsitektural yang belum banyak diterapkan pada konteks ini, mengingat domain masalah telah cukup dipahami namun solusi yang tersedia masih memiliki keterbatasan dalam memenuhi kebutuhan praktis organisasi, khususnya pada konteks Bahasa Indonesia yang belum memiliki benchmark akurasi resmi untuk teknologi *speech-to-text* yang digunakan [17]. Penelitian ini bertujuan untuk menganalisis kebutuhan fungsional dan non-fungsional sistem otomasi notulen rapat berdasarkan temuan lapangan pada studi kasus, merancang dan mengimplementasikan arsitektur sistem berbasis *client-side integration* yang menyisipkan titik verifikasi manusia pada proses transkripsi dan peringkasan serta mempertahankan jejak proses pengambilan keputusan dalam keluaran notulen, serta mendemonstrasikan dan mengevaluasi rancangan tersebut melalui pengujian langsung pada data rapat nyata, termasuk identifikasi keterbatasan teknis yang ditemukan selama proses demonstrasi.

2. METODOLOGI PENELITIAN

2.1 Tahapan Penelitian

Penelitian ini menggunakan pendekatan Design Science Research (DSR) mengacu pada metodologi yang dikembangkan oleh Peffers dan rekan-rekannya, yang terdiri atas enam tahapan: identifikasi masalah dan motivasi, penetapan tujuan solusi, perancangan dan pengembangan artefak, demonstrasi, evaluasi, dan komunikasi hasil [16]. Keenam tahapan tersebut dipilih karena sesuai dengan karakteristik penelitian ini yang berorientasi pada perancangan artefak sistem dan bersifat iteratif. Rancangan awal mengalami beberapa kali penyesuaian berdasarkan kendala teknis dan masukan yang muncul selama proses pengembangan, sebuah karakteristik yang melekat pada pendekatan DSR yang memandang perancangan dan evaluasi sebagai proses yang saling terkait dan dapat diulang sebelum mencapai artefak yang dianggap memadai.

2.2 Analisis Kebutuhan Pemangku Kepentingan

Analisis kebutuhan dilakukan melalui diskusi dengan perwakilan tiga kelompok pemangku kepentingan organisasi yang terlibat dalam proses rapat: staf/pegawai sebagai peserta rapat reguler, pengambil keputusan (pimpinan unit kerja), serta pihak eksternal sebagai peserta tambahan pada rapat tertentu. Hasil analisis kebutuhan dirangkum pada Tabel 1.

Tabel 1. Kebutuhan Fungsional dan Non-Fungsional per Kelompok Pemangku Kepentingan

Kelompok	Temuan Utama	Implikasi Non-Fungsional
Staf/pegawai	Tidak ada pencatatan sama sekali pada banyak rapat	Sistem harus menjamin notulen selalu dihasilkan untuk setiap rapat yang direkam
Pengambil keputusan	Kesulitan menindaklanjuti keputusan karena tidak ada catatan; menginginkan validasi sebelum distribusi	Diperlukan titik verifikasi manusia sebelum publikasi
Pihak eksternal	Notulen hanya dikirim untuk rapat rutin yang dianggap penting	Sistem berpotensi mengurangi bias selektif dokumentasi

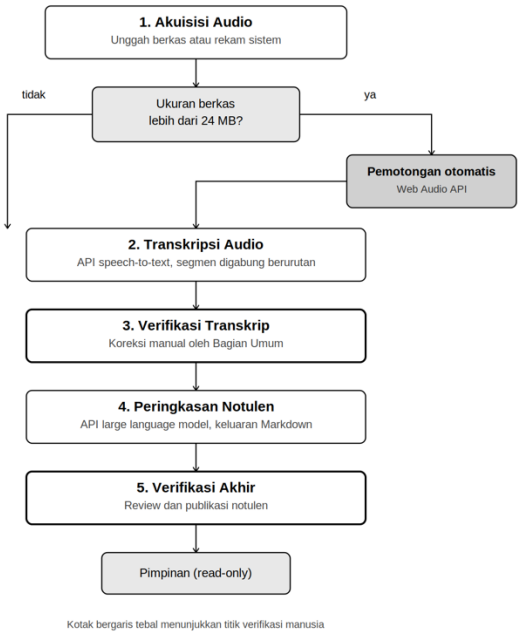
Berdasarkan kebutuhan tersebut, ditetapkan tiga kebutuhan non-fungsional utama yang menjadi dasar perancangan arsitektur. Pertama, ketersediaan mekanisme otomatis yang menjamin notulen tetap dihasilkan untuk setiap rapat yang direkam, tanpa bergantung pada ketersediaan notulis manusia. Kedua, keberadaan titik verifikasi manusia sebelum informasi didistribusikan secara resmi. Ketiga, konsistensi proses dokumentasi yang tidak membedakan jenis rapat, untuk mengurangi kecenderungan dokumentasi selektif yang ditemukan pada praktik sebelumnya.

EXPLORER
Journal of Computer Science and Information Technology
E-ISSN 2774-4647
Website: https://journal.fkpt.org/index.php/Explorer **DOI:** https://doi.org/10.47065/explorer.v6i2.3013

2.3 Perancangan Arsitektur Sistem

Berdasarkan kebutuhan yang teridentifikasi, dirancang arsitektur sistem berbasis client-side integration, di mana seluruh logika orkestrasi proses dijalankan pada sisi klien (peramban) tanpa memerlukan server perantara. Arsitektur ini terdiri atas lima tahapan pemrosesan, yaitu akuisisi audio rapat, transkripsi audio menjadi teks melalui pemanggilan API *speech-to-text*, verifikasi dan koreksi transkrip oleh penanggung jawab rapat, peringkasan teks menjadi notulen terstruktur melalui pemanggilan API large language model, serta verifikasi akhir dan publikasi notulen.

Pemilihan pendekatan client-side integration didasarkan pada evaluasi terhadap kendala praktis yang ditemui selama proses pengembangan, dibanding pendekatan berbasis platform orkestrasi *workflow* seperti n8n yang dipertimbangkan dan diimplementasikan pada tahap awal perancangan. Implementasi awal menggunakan n8n sebagai *workflow* orchestration engine yang menghubungkan layanan *speech-to-text* dan *large language model* melalui serangkaian node fungsional . Namun, proses demonstrasi pada tahap ini mengungkap kendala signifikan terkait kompleksitas konfigurasi binary data, kebutuhan infrastruktur server tambahan, serta inkonsistensi konfigurasi pada antarmuka node [14]. Berdasarkan evaluasi terhadap akumulasi kendala tersebut, rancangan arsitektur dialihkan ke pendekatan *client-side integration*, yang menawarkan pengurangan overhead operasional dan portabilitas yang lebih baik karena seluruh pemrosesan dapat dijalankan secara lokal tanpa memerlukan infrastruktur backend [15]. Gambaran umum arsitektur sistem yang dirancang ditunjukkan pada Gambar 1, yang menggambarkan lima tahap pemrosesan beserta titik percabangan untuk penanganan berkas audio berukuran besar.

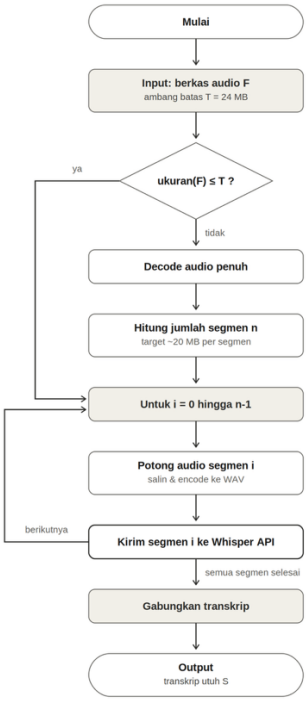


Gambar 1. Arsitektur Sistem

2.3.1 Akuisisi Audio

Sistem mendukung dua mekanisme akuisisi audio rapat. Mekanisme pertama adalah pengunggahan berkas hasil perekaman melalui aplikasi pihak ketiga seperti OBS Studio, Zoom, atau Microsoft Teams, yang tidak memiliki ketergantungan terhadap keterbukaan sesi peramban selama rapat berlangsung. Mekanisme kedua adalah perekaman langsung audio sistem melalui antarmuka peramban menggunakan Screen Capture API, yang memungkinkan akuisisi audio tanpa memerlukan aplikasi perekam tambahan. Izin penggunaan mekanisme ini tidak dapat dipertahankan untuk sesi berikutnya dan harus diminta kembali setiap kali, sebuah keputusan desain keamanan yang ditetapkan secara sengaja pada tingkat spesifikasi dan berlaku konsisten di seluruh peramban modern, termasuk ketika diusulkan perubahan agar izin tersebut dapat dipertahankan selama sesi pengguna berlangsung [18], [19]. Karakteristik ini berimplikasi pada keterbatasan praktis yang bersifat permanen, bukan sekadar keterbatasan teknis sementara, yaitu sesi perekaman akan berhenti apabila jendela peramban ditutup atau dimuat ulang sebelum rapat selesai. Kedua mekanisme disediakan secara bersamaan untuk mengakomodasi variasi kondisi penggunaan, mengingat mekanisme kedua memiliki keterbatasan struktural tersebut pada rapat berdurasi panjang.

EXPLORER
Journal of Computer Science and Information Technology
E-ISSN 2774-4647
Website: https://journal.fkpt.org/index.php/Explorer **DOI:** https://doi.org/10.47065/explorer.v6i2.3013



Gambar 2. Diagram alur algoritma pemotongan audio

2.3.2 Transkripsi Audio

Layanan *speech-to-text* yang digunakan dalam penelitian ini menetapkan batas maksimum ukuran berkas audio sebesar 25 megabyte per permintaan [20]. Mengingat durasi rapat yang menjadi objek demonstrasi penelitian ini berkisar satu hingga dua jam, berkas audio yang dihasilkan dalam format tanpa kompresi tinggi dapat dengan mudah melampaui batas tersebut. Untuk mengantisipasi hal ini dengan margin keamanan, sistem menetapkan ambang batas internal sebesar 24 megabyte. Berkas yang melebihi ambang batas tersebut secara otomatis dipecah menjadi beberapa segmen berdasarkan durasi menggunakan Web Audio API pada sisi klien. Setiap segmen dikirimkan ke layanan transkripsi secara berurutan, bukan secara paralel, untuk menghindari pelampauan batas permintaan API secara bersamaan. Hasil transkripsi setiap segmen disimpan sementara, dan setelah seluruh segmen selesai diproses, teks dari setiap segmen digabungkan secara berurutan oleh logika pada sisi klien menjadi satu transkrip utuh sebelum ditampilkan kepada penanggung jawab rapat untuk diverifikasi.

2.3.3 Verifikasi Transkrip

Transkrip hasil tahap sebelumnya ditampilkan kepada penanggung jawab rapat (Bagian Umum) untuk diperiksa dan dikoreksi sebelum diteruskan ke tahap peringkasan. Penyisipan tahap ini didasarkan pada pertimbangan bahwa layanan *speech-to-text* yang digunakan belum memiliki benchmark akurasi resmi untuk Bahasa Indonesia. Studi pada bahasa daerah Indonesia yang memiliki karakteristik linguistik serumpun menunjukkan bahwa model Whisper tanpa penyesuaian khusus (*zero-shot*) dapat menghasilkan tingkat kesalahan yang sangat tinggi, bahkan pada kondisi audio bersih, dan performanya menurun lebih tajam lagi pada kondisi akustik dengan derau latar belakang [13]. Studi sejenis lainnya melaporkan tingkat kesalahan kata sebesar 89,40 persen pada model dasar Whisper, yang menurun menjadi 13,77% setelah dilakukan penyesuaian khusus (*fine-tuning*) untuk bahasa daerah tertentu, menegaskan kesenjangan akurasi yang besar antara penggunaan model secara langsung dan model yang telah disesuaikan [21]. Berdasarkan kondisi tersebut, kesalahan transkripsi berpotensi terbawa dan diperkuat pada tahap peringkasan berikutnya apabila tidak dikoreksi terlebih dahulu. Hal ini sejalan dengan prinsip Human-in-the-Loop yang memperlakukan keluaran model sebagai informasi yang perlu diverifikasi, bukan hasil akhir yang dapat langsung dipercaya [11].

2.3.4 Peringkasan Notulen

Transkrip yang telah dikoreksi diteruskan ke large language model melalui instruksi (*prompt*) terstruktur. Instruksi tersebut meminta model untuk menyusun notulen berdasarkan topik yang teridentifikasi dari transkrip secara mandiri, dengan setiap topik mencakup pihak yang mengajukan usulan atau pertanyaan apabila disebutkan secara eksplisit dalam transkrip, pertimbangan atau opsi yang dibahas, serta bagaimana keputusan terkait topik tersebut dicapai. Penyusunan notulen dengan pendekatan ini ditujukan agar dokumen yang dihasilkan tidak hanya memuat kesimpulan akhir, tetapi juga jejak proses yang melatarbelakangi tercapainya kesimpulan tersebut, sejalan dengan

fungsi notulen sebagai catatan formal yang merepresentasikan proses pengambilan keputusan, bukan sekadar ringkasan hasil [22]. Instruksi tetap menekankan penyajian yang ringkas dan netral, tanpa merekonstruksi dialog secara verbatim, konsisten dengan praktik notulen formal yang menghindari pencatatan perdebatan secara rinci [22]. Data tabular disajikan dalam format tabel apabila relevan, dan bagian keputusan serta tindak lanjut disusun secara terpisah pada bagian akhir dokumen. Keluaran model diminta dalam format *Markdown* agar dapat direpresentasikan secara terstruktur baik pada antarmuka web maupun dokumen hasil ekspor.

2.3.5 Verifikasi Akhir dan Publikasi

Notulen hasil peringkasan ditampilkan kepada Bagian Umum dalam bentuk terformat, yaitu hasil konversi *Markdown* ke HTML, untuk diperiksa kembali dengan opsi pengeditan langsung pada teks sumber apabila diperlukan koreksi. Rancangan awal mempertimbangkan mekanisme persetujuan terpisah oleh pimpinan melalui pesan instan sebelum distribusi. Namun, hasil iterasi rancangan menetapkan bahwa verifikasi dilakukan oleh penanggung jawab rapat yang memiliki konteks langsung terhadap pelaksanaan rapat, sementara pimpinan mengakses notulen final secara hanya-baca (*read-only*) setelah publikasi

2.4 Implementasi

Implementasi dilakukan menggunakan dua layanan application programming interface (API) dari penyedia yang sama untuk menyederhanakan pengelolaan kredensial, yaitu layanan *speech-to-text* (Whisper) untuk tahap transkripsi dan layanan *large language model* (GPT-4o-mini) untuk tahap peringkasan. Pemilihan penyedia API tunggal didasarkan pada pertimbangan kemudahan implementasi dan pengelolaan biaya operasional, setelah pertimbangan awal terhadap penyedia lain ditemukan memiliki kendala teknis terkait dukungan input audio pada antarmuka pemanggilan API standar.

Pemilihan model large language model dengan kapasitas relatif kecil (GPT-4o-mini), dibanding model dengan kapasitas lebih besar) mempertimbangkan efisiensi biaya pada tugas peringkasan teks berstruktur. Penelitian sejenis menemukan bahwa model berkapasitas kecil pada umumnya belum mampu mengungguli model berkapasitas besar dalam kondisi *zero-shot* untuk tugas peringkasan rapat, dan keunggulan model kecil baru terlihat signifikan setelah dilakukan penyesuaian khusus (*fine-tuning*) untuk tugas tersebut. Mengingat penelitian ini tidak melakukan *fine-tuning* khusus terhadap model yang digunakan, pemilihan GPT-4o-mini merupakan keputusan yang mengutamakan efisiensi biaya dan kemudahan implementasi, dengan kesadaran akan kemungkinan *trade-off* pada kualitas keluaran dibandingkan model berkapasitas lebih besar [23], [9].

Hasil notulen dalam format *Markdown* direpresentasikan pada antarmuka web menggunakan pustaka *rendering Markdown* ke HTML. Notulen tersebut juga dapat diekspor menjadi dokumen PDF melalui mekanisme penguraian (parsing) elemen *Markdown* secara manual, meliputi heading, tabel, dan senarai (*list*), ke dalam elemen dokumen PDF asli, sehingga teks pada dokumen hasil ekspor tetap dapat ditelusuri dan disalin.

2.5 Skenario Demonstrasi dan Evaluasi

Demonstrasi dilakukan menggunakan rekaman audio rapat nyata dengan durasi sekitar satu setengah jam, yang menghasilkan berkas audio berukuran sekitar 120,73 *megabyte*, untuk menguji mekanisme pemotongan audio otomatis pada skenario rapat berdurasi panjang. Evaluasi dilakukan melalui dua pendekatan. Pendekatan pertama adalah evaluasi teknis terhadap keberhasilan setiap tahap pemrosesan. Pendekatan kedua adalah evaluasi formatif yang bersifat iteratif, di mana hasil notulen yang dihasilkan sistem diperiksa terhadap kriteria kelengkapan proses pengambilan keputusan dan dibandingkan secara kualitatif dengan hasil peringkasan menggunakan *large language model* populer pada transkrip yang sama. Hasil evaluasi pada setiap iterasi digunakan untuk menyempurnakan instruksi (prompt) yang diberikan kepada model, sejalan dengan karakteristik DSR yang memandang perancangan dan evaluasi sebagai proses yang saling memengaruhi. Evaluasi melalui tinjauan oleh pemangku kepentingan eksternal (*expert review*) terhadap hasil akhir belum dilakukan pada tahap penelitian ini dan diidentifikasi sebagai agenda lanjutan, sebagaimana dibahas pada bagian Keterbatasan.

3. HASIL DAN PEMBAHASAN

Berdasarkan metode penelitian yang telah diuraikan, penelitian ini merancang dan mengimplementasikan arsitektur sistem otomasi notulen rapat berbasis client-side integration melalui tujuh iterasi perancangan, menguji arsitektur tersebut pada data audio rapat nyata berdurasi sekitar satu setengah jam, serta mengevaluasi kualitas notulen yang dihasilkan melalui perbandingan dengan *large language model* populer dan penyempurnaan instruksi (*prompt*) secara bertahap. Hasil yang diperoleh menunjukkan bahwa arsitektur final yang menggabungkan layanan *speech-to-text* dan *large language model* dengan dua titik verifikasi manusia mampu menghasilkan notulen yang merepresentasikan proses pengambilan keputusan rapat, sekaligus mengungkap sejumlah keterbatasan teknis yang menjadi dasar penyempurnaan rancangan pada tahap selanjutnya.

EXPLORER
Journal of Computer Science and Information Technology
E-ISSN 2774-4647
Website: https://journal.fkpt.org/index.php/Explorer DOI: https://doi.org/10.47065/explorer.v6i2.3013

3.1 Proses Iteratif Perancangan Arsitektur

Sesuai dengan karakteristik Design Science Research yang memandang perancangan dan evaluasi sebagai proses yang saling memengaruhi [16], tahap perancangan dan pengembangan pada penelitian ini melalui dua jenis iterasi yang saling terkait. Iterasi pertama berkaitan dengan keputusan arsitektural mendasar, mencakup tujuh tahap perubahan rancangan sebelum mencapai arsitektur final. Iterasi kedua berkaitan dengan penyempurnaan kualitas keluaran sistem, khususnya pada tahap peringkasan notulen, yang berlangsung melalui tujuh kali penyempurnaan instruksi kepada model, dimulai dari format keluaran JSON terstruktur dengan field tetap hingga format Markdown bebas-struktur yang mempertahankan jejak proses pengambilan keputusan. Bagian ini menguraikan iterasi pertama, yaitu perubahan keputusan arsitektural.

Setiap iterasi arsitektural dipicu oleh kendala teknis konkret yang ditemukan selama proses demonstrasi awal, dan menjadi dasar pengambilan keputusan desain pada iterasi berikutnya. Ringkasan proses iteratif tersebut disajikan pada Tabel 2.

Tabel 2. Proses Iteratif Perancangan Arsitektur Sistem

Iterasi	Pendekatan yang Dicoba	Kendala yang Ditemukan	Keputusan Selanjutnya
1	Gemini API untuk transkripsi dan peringkasan dalam satu model	Kunci API mengalami kendala autentikasi pada level akun yang bersifat sistemik, dilaporkan oleh sejumlah pengguna lain	Beralih ke penyedia API lain
2	Claude API untuk transkripsi dan peringkasan	Messages API tidak mendukung modalitas input audio secara native sesuai dokumentasi resmi	Memisahkan tahap transkripsi (speech-to-text) dan peringkasan menjadi dua layanan terpisah
3	Whisper (OpenAI) untuk transkripsi, Claude (Anthropic) untuk peringkasan	Kompleksitas pengelolaan dua kredensial API dan peningkatan biaya operasional tanpa keuntungan teknis signifikan	Konsolidasi ke satu penyedia API
4	n8n sebagai workflow orchestration engine untuk menghubungkan layanan speech-to-text dan large language model	Kompleksitas konfigurasi binary data, kebutuhan infrastruktur server tambahan, inkonsistensi mode ekspresi pada antarmuka node	Beralih ke pendekatan client-side integration
5	Arsitektur client-side diuji dengan audio rapat berdurasi panjang (sekitar satu setengah jam)	Berkas audio melebihi batas maksimum API transkripsi sebesar 25 megabyte	Implementasi mekanisme pemotongan audio otomatis berbasis durasi menggunakan Web Audio API
6	Keluaran large language model dalam format JSON terstruktur dengan field tetap (ringkasan, keputusan, action items)	Struktur tetap tidak mengakomodasi topik yang bervariasi antar rapat, dan notulen hanya memuat kesimpulan tanpa jejak proses pengambilan keputusan	Mengubah format keluaran dari JSON tetap menjadi Markdown bebas-struktur, dengan model yang sama
7	Persetujuan notulen oleh pimpinan melalui pesan instan (WhatsApp Business API) sebelum distribusi	Membutuhkan komponen backend untuk menerima respons pesan, bertentangan dengan keputusan arsitektural client-side pada iterasi keempat	Reposisi mekanisme verifikasi ke penanggung jawab rapat (Bagian Umum); pimpinan mengakses notulen secara hanya-baca

Iterasi pertama dan kedua menunjukkan bahwa pemilihan large language model untuk memproses audio rapat tidak dapat dilakukan tanpa mempertimbangkan dukungan teknis penyedia API terhadap modalitas input audio secara eksplisit. Percobaan awal menggunakan Gemini API terhambat oleh kendala pada level penyediaan kunci API yang ditemukan bersifat sistemik, bukan kesalahan konfigurasi pada sisi peneliti, sebagaimana dilaporkan oleh sejumlah pengguna lain pada forum *developer* resmi penyedia layanan tersebut [24]. Percobaan berikutnya menggunakan Claude API mengungkap keterbatasan yang berbeda. Dokumentasi resmi menyatakan secara eksplisit bahwa *Messages* API hanya mendukung modalitas teks dan gambar, sehingga audio yang disertakan dalam permintaan akan diabaikan oleh sistem. Temuan ini menegaskan pentingnya verifikasi terhadap

EXPLORER
Journal of Computer Science and Information Technology
E-ISSN 2774-4647
Website: https://journal.fkpt.org/index.php/Explorer **DOI:** https://doi.org/10.47065/explorer.v6i2.3013

dokumentasi resmi penyedia API sebelum tahap implementasi, alih-alih mengandalkan asumsi kesetaraan kapabilitas antar penyedia layanan *large language model*.

Iterasi ketiga dan keempat berkaitan dengan pertimbangan arsitektural pada tingkat orkestrasi proses. Penggunaan dua penyedia API berbeda pada iterasi ketiga, yaitu Whisper dari OpenAI dan Claude dari Anthropic, meningkatkan kompleksitas pengelolaan kredensial dan biaya operasional tanpa keuntungan teknis yang signifikan, sehingga diputuskan konsolidasi ke satu penyedia. Pada iterasi keempat, implementasi awal menggunakan n8n sebagai workflow orchestration engine [14] dipilih karena kemampuan visualisasi alur proses dan dukungan node bawaan untuk berbagai layanan eksternal. Implementasi ini mengungkap tiga kendala signifikan, yaitu kompleksitas konfigurasi binary data audio yang memerlukan penyesuaian berulang pada antarmuka node, kebutuhan infrastruktur kontainerisasi yang menambah beban operasional, serta inkonsistensi antara mode ekspresi (*expression mode*) yang diharapkan node HTTP Request dengan cara data dikirimkan, yang berulang kali menghasilkan kegagalan validasi JSON pada permintaan API. Kendala-kendala tersebut konsisten dengan laporan pengguna n8n lain yang mendokumentasikan kesulitan serupa dalam mengintegrasikan layanan perpesanan eksternal ke dalam *workflow node-based* [25] . Berdasarkan evaluasi terhadap akumulasi kendala ini, rancangan arsitektur dialihkan ke pendekatan *client-side integration*, yang menawarkan pengurangan *overhead* operasional dan portabilitas yang lebih baik karena seluruh pemrosesan dapat dijalankan secara lokal tanpa memerlukan infrastruktur *backend* [15].

Iterasi kelima muncul setelah arsitektur *client-side* diuji dengan data audio rapat berdurasi panjang, sekitar satu setengah jam, yang menghasilkan berkas berukuran melebihi batas maksimum API transkripsi sebesar 25 megabyte [20]. Kendala ini bersifat spesifik-domain, tidak muncul pada pengujian awal dengan audio durasi singkat, dan menunjukkan pentingnya pengujian dengan data yang representatif terhadap kondisi penggunaan nyata, bukan hanya data uji berdurasi pendek. Solusi yang diimplementasikan adalah mekanisme pemotongan audio otomatis berbasis durasi menggunakan Web Audio API pada sisi klien, dengan ambang batas internal sebesar 25 megabyte sebagai margin keamanan, tanpa memerlukan pemrosesan di luar peramban. Pendekatan pemotongan berdasarkan durasi yang diimplementasikan pada iterasi ini termasuk dalam kategori fixed-length chunking, yaitu strategi yang membagi aliran audio menjadi segmen seragam berdasarkan durasi waktu dan menjamin cakupan penuh seluruh rekaman, meskipun tidak mempertimbangkan batas semantik seperti jeda antar pembicara atau transisi topik [26]. Pendekatan serupa yang umum diterapkan pada sistem transkripsi audio panjang juga menghadapi tantangan yang sama, di mana pemrosesan segmen secara independen berpotensi memperkenalkan inkonsistensi pada titik sambungan antar segmen akibat hilangnya konteks lintas segmen [27]. Keterbatasan ini menjadi salah satu dasar pertimbangan penelitian lanjutan.

Iterasi keenam berkaitan dengan struktur keluaran sistem. Rancangan awal yang meminta large language model menghasilkan keluaran dalam format JSON terstruktur dengan field tetap (ringkasan, keputusan, action items) menghasilkan dua keterbatasan. Pertama, struktur tetap tersebut tidak mengakomodasi pembagian topik yang bervariasi antar rapat, karena setiap rapat memiliki agenda dan jumlah topik yang berbeda. Kedua, dan lebih signifikan, notulen yang dihasilkan cenderung hanya memuat kesimpulan akhir tanpa menampilkan proses yang melatarbelakanginya, seperti pihak yang mengajukan usulan dan pertimbangan yang dibahas sebelum suatu keputusan tercapai, sehingga dokumen yang dihasilkan lebih menyerupai ringkasan eksekutif dibanding notulen yang berfungsi sebagai bukti proses pengambilan keputusan [22]. Berdasarkan kedua keterbatasan tersebut, model large language model yang digunakan tetap sama (GPT-4o-mini), namun instruksi (*prompt*) dan format keluaran yang diminta diubah dari struktur JSON tetap menjadi format *Markdown* bebas-struktur, sehingga model memiliki keleluasaan menyusun topik secara dinamis sekaligus diarahkan untuk menyertakan jejak proses pengambilan keputusan pada setiap topik. Proses penyempurnaan instruksi ini berlangsung secara iterative.

Iterasi ketujuh menyangkut mekanisme verifikasi sebelum publikasi notulen. Rancangan awal mempertimbangkan persetujuan oleh pimpinan melalui pesan instan (WhatsApp Business API) sebagai bentuk kontrol akuntabilitas, sejalan dengan temuan analisis kebutuhan yang menunjukkan keinginan pimpinan untuk memvalidasi notulen sebelum distribusi. Namun, implementasi mekanisme tersebut memerlukan komponen backend untuk menerima dan memproses respons pesan, yang bertentangan dengan keputusan arsitektural pada iterasi keempat untuk menghindari kebutuhan infrastruktur server. Pertimbangan lebih lanjut terhadap alur kerja organisasi menghasilkan reposisi mekanisme verifikasi. Validasi dipindahkan ke penanggung jawab rapat (Bagian Umum) yang memiliki konteks langsung terhadap pelaksanaan rapat, sementara pimpinan mengakses notulen final secara hanya-baca setelah publikasi. Reposisi ini tetap memenuhi kebutuhan akuntabilitas yang teridentifikasi pada analisis kebutuhan, namun melalui aktor pelaksana yang berbeda dari rancangan awal.

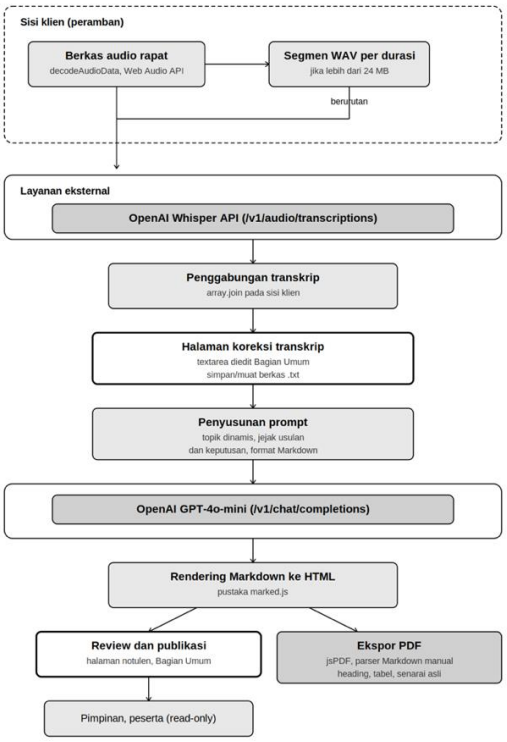
Secara keseluruhan, proses iteratif arsitektural ini menunjukkan bahwa keputusan arsitektur final pada penelitian ini, yaitu *client-side integration* dengan dua titik verifikasi manusia, bukan merupakan pilihan yang ditetapkan sejak awal, melainkan hasil eliminasi sistematis terhadap pendekatan-pendekatan yang terbukti tidak optimal melalui pengujian langsung. Iterasi penyempurnaan kualitas keluaran, yang berlangsung secara berkelanjutan pada tahap peringkasan notulen, dibahas secara terpisah sebagai bagian dari evaluasi formatif penelitian ini.

EXPLORER
Journal of Computer Science and Information Technology
E-ISSN 2774-4647
Website: https://journal.fkpt.org/index.php/Explorer **DOI:** https://doi.org/10.47065/explorer.v6i2.3013

3.2 Hasil Implementasi Arsitektur

Seluruh komponen sistem, termasuk logika pemotongan audio, pemanggilan API, dan penguraian (*parsing*) *Markdown* ke elemen PDF, diimplementasikan sebagai berkas *HyperText Markup Language* (HTML) tunggal yang dijalankan langsung pada peramban tanpa proses instalasi maupun konfigurasi server. Pendekatan ini memungkinkan sistem dijalankan pada komputer mana pun yang memiliki peramban modern, tanpa memerlukan keahlian teknis untuk instalasi atau pemeliharaan infrastruktur.

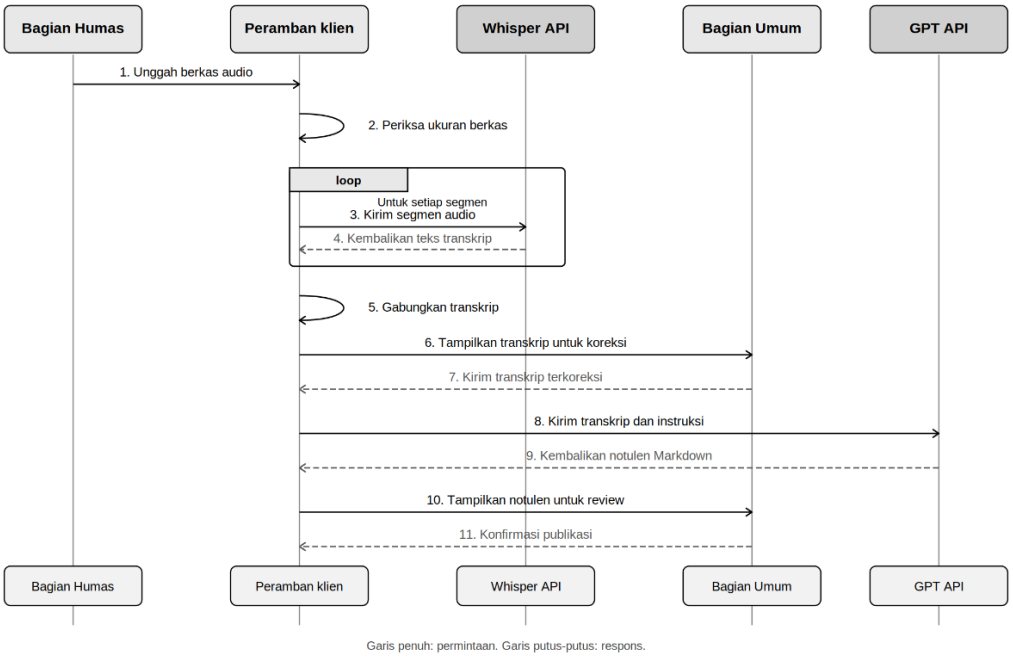
Implementasi teknis dari arsitektur berbasis client-side integration yang terdiri atas lima tahap pemrosesan ditunjukkan pada Gambar 3, yang memetakan setiap tahap pemrosesan ke layanan API dan pustaka perangkat lunak spesifik yang digunakan.



Gambar 3. Diagram Implementasi Teknis

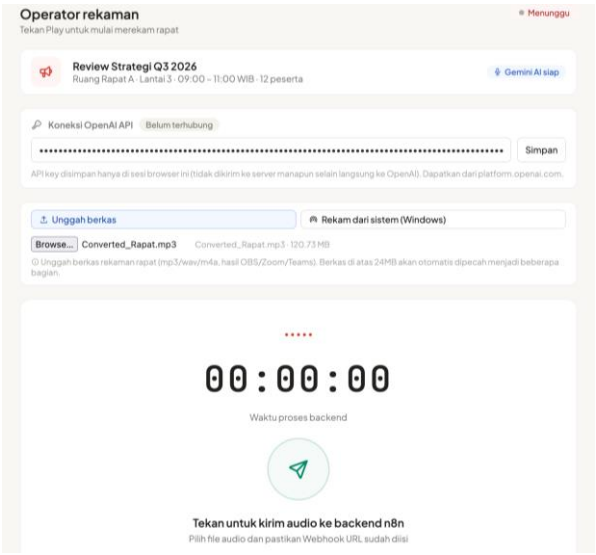
Antarmuka sistem dirancang dengan empat peran pengguna yang masing-masing memiliki akses berbeda terhadap fungsi sistem, yaitu Bagian Humas sebagai operator perekaman dan pengunggahan audio, Bagian Umum sebagai penanggung jawab koreksi transkrip, peringkasan, dan publikasi notulen, Pimpinan sebagai pengakses notulen secara hanya-baca tanpa kewenangan persetujuan, dan Pembaca Umum (*Viewer*) sebagai pengakses notulen yang telah dipublikasikan. Pembagian peran ini merupakan hasil akhir dari reposisi mekanisme verifikasi yang semula dirancang dilakukan oleh pimpinan melalui pesan instan, namun dialihkan kepada penanggung jawab rapat karena implementasi respons pesan memerlukan komponen backend yang bertentangan dengan keputusan arsitektur client-side.

EXPLORER
Journal of Computer Science and Information Technology
E-ISSN 2774-4647
Website: https://journal.fkpt.org/index.php/Explorer **DOI:** https://doi.org/10.47065/explorer.v6i2.3013



Gambar 4. Squence Diagram Interaksi Sistem

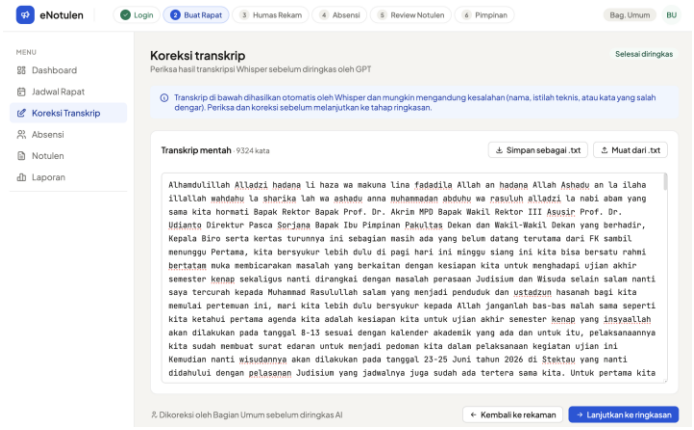
Urutan interaksi antar komponen sistem secara runtime, mencakup Bagian Humas, peramban klien, layanan Whisper API, Bagian Umum, dan layanan GPT API, divisualisasikan dalam bentuk sequence diagram pada Gambar 4. Diagram tersebut menunjukkan sebelas langkah interaksi, dimulai dari pengunggahan berkas audio oleh Bagian Humas hingga konfirmasi publikasi oleh Bagian Umum, termasuk perulangan pengiriman segmen audio ke layanan Whisper API secara berurutan untuk mengatasi batasan ukuran berkas 25 megabyte per permintaan API.



Gambar 5. Halaman Operator Rekaman

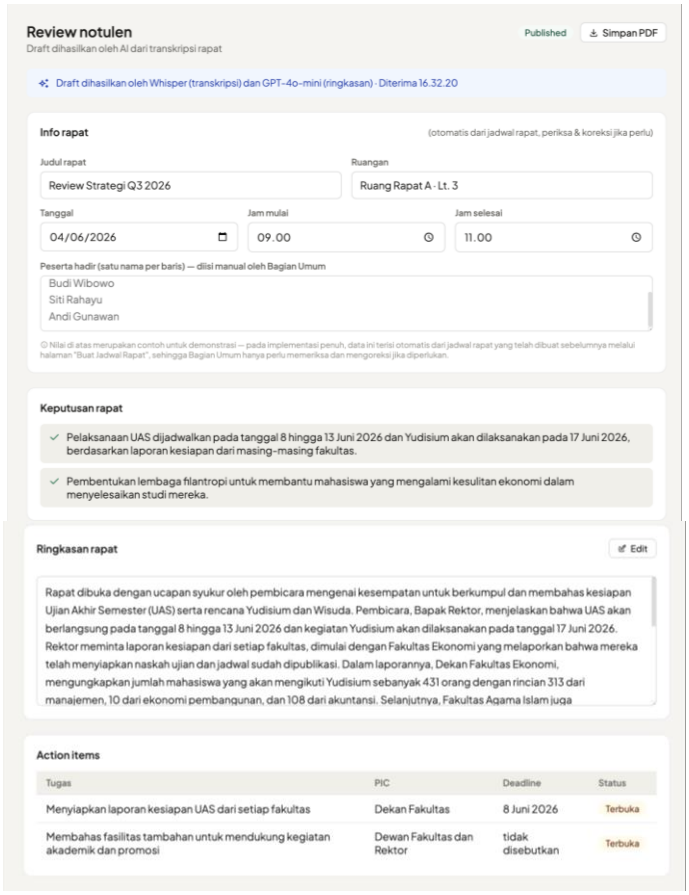
Gambar 5 menunjukkan antarmuka Operator Perekaman, yang menyediakan dua mekanisme akuisisi audio yaitu pengunggahan berkas hasil perekaman dari aplikasi pihak ketiga dan perekaman langsung audio sistem melalui antarmuka peramban menggunakan Screen Capture API.

EXPLORER
Journal of Computer Science and Information Technology
E-ISSN 2774-4647
Website: https://journal.fkpt.org/index.php/Explorer **DOI:** https://doi.org/10.47065/explorer.v6i2.3013



Gambar 6. Halaman Koreksi Transkrip

Gambar 6 menunjukkan antarmuka Koreksi Transkrip, tempat Bagian Umum memeriksa dan mengoreksi hasil transkripsi sebelum diteruskan ke tahap peringkasan, dilengkapi dengan fitur untuk menyimpan dan memuat ulang transkrip dalam format berkas teks guna mendukung pengujian berulang tanpa perlu mengulang proses transkripsi audio.



Gambar 7. Halaman Review Notulen

Gambar 7 menunjukkan antarmuka *Review* Notulen, yang menampilkan hasil peringkasan dalam bentuk Markdown yang telah dirender menjadi tampilan terformat, beserta bidang metadata rapat (judul, tanggal, waktu, dan ruangan) yang pada implementasi saat ini diisi dengan nilai contoh untuk keperluan demonstrasi, mewakili kondisi apabila data tersebut telah tersedia dari proses penjadwalan rapat sebelumnya. Bidang peserta hadir diisi secara manual oleh Bagian Umum setelah rapat selesai.

Keempat peran pengguna diimplementasikan menggunakan mekanisme pemilihan peran pada halaman masuk, tanpa autentikasi kredensial sesungguhnya, mengingat sistem ini berada pada tahap demonstrasi penelitian dan belum diimplementasikan pada lingkungan produksi penuh. Integrasi otomatis dengan modul penjadwalan rapat dan sistem presensi organisasi, yang pada implementasi saat ini masih bersifat simulasi, diidentifikasi sebagai bagian dari agenda pengembangan lanjutan yang dibahas lebih rinci pada bagian Kesimpulan.

3.3. Hasil Demonstrasi

Demonstrasi dilakukan menggunakan dua berkas audio rapat nyata dengan ukuran berbeda untuk menguji konsistensi mekanisme pemotongan audio otomatis pada skenario rapat berdurasi panjang. Berkas pertama berukuran 120,73 *megabyte* membutuhkan waktu pemrosesan transkripsi selama 23 menit 4 detik hingga selesai, sementara berkas kedua berukuran 83,82 *megabyte* membutuhkan waktu 17 menit 52 detik. Kedua proses berhasil menyelesaikan tahap transkripsi tanpa kegagalan permintaan pada layanan Whisper API, dengan hasil transkripsi setiap segmen berhasil digabungkan menjadi satu transkrip utuh dan ditampilkan pada antarmuka koreksi transkrip.

Pemeriksaan terhadap transkrip yang dihasilkan menemukan adanya kesalahan penulisan pada sejumlah kata, sebagaimana umum terjadi pada keluaran sistem *speech-to-text* untuk Bahasa Indonesia yang belum memiliki *benchmark* akurasi resmi, konsisten dengan studi pada bahasa daerah Indonesia yang melaporkan tingkat kesalahan kata hingga 89,40 persen pada model Whisper tanpa penyesuaian khusus.. Tahap verifikasi dan koreksi manual oleh Bagian Umum berfungsi sebagaimana dirancang, memungkinkan kesalahan tersebut diperbaiki sebelum transkrip diteruskan ke tahap peringkasan.

Hasil peringkasan awal, yang menggunakan struktur keluaran JSON terstruktur, menghasilkan notulen yang dinilai terlalu ringkas, dengan poin-poin pertanyaan dan usulan dari peserta rapat tidak tertangkap dalam keluaran. Berdasarkan temuan tersebut, format keluaran diubah dari JSON terstruktur dengan field tetap menjadi Markdown bebas-struktur, dengan instruksi yang secara eksplisit meminta model menyertakan pihak pengusul, pertimbangan yang dibahas, dan dasar tercapainya keputusan pada setiap topik. Setelah instruksi (*prompt*) disempurnakan untuk menghasilkan keluaran *Markdown* dengan jejak proses pengambilan keputusan, hasil peringkasan menunjukkan perbaikan, dengan notulen yang memuat pembagian topik yang lebih sesuai dengan isi rapat serta keterkaitan antara usulan dan keputusan yang lebih eksplisit dibandingkan hasil sebelumnya. Dokumen notulen yang dihasilkan, baik dalam tampilan web maupun hasil ekspor PDF, berhasil menampilkan struktur heading, tabel, dan senarai sesuai dengan format *Markdown* yang dihasilkan model.

3.4 Hasil Evaluasi

Evaluasi terhadap kualitas keluaran sistem dilakukan secara formatif dan iterative. Pada tahap awal, struktur keluaran berbasis JSON dengan *field* tetap dibandingkan secara kualitatif dengan hasil peringkasan transkrip yang sama menggunakan *large language model* populer. Perbandingan ini menunjukkan bahwa keluaran sistem secara konsisten lebih ringkas dan kurang menampilkan jejak proses pengambilan keputusan dibandingkan hasil pembanding, yang menjadi dasar bagi perubahan instruksi

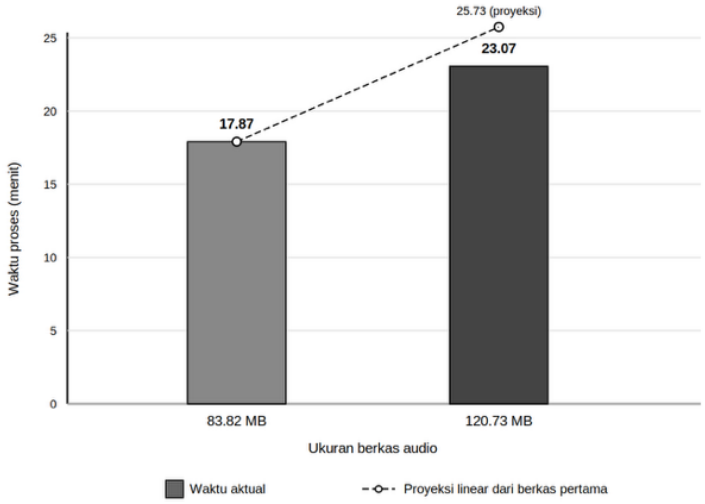
kepada model dari format JSON terstruktur dengan field tetap menjadi format Markdown bebas-struktur yang secara eksplisit meminta pencantuman pihak pengusul, pertimbangan yang dibahas, dan dasar tercapainya keputusan pada setiap topik. Hasil perbandingan ini bersifat kualitatif berdasarkan pengamatan langsung terhadap kedua keluaran, tanpa pengukuran metrik kuantitatif seperti panjang ringkasan atau jumlah entitas yang teridentifikasi, dan diidentifikasi sebagai keterbatasan evaluasi yang dibahas lebih lanjut pada bagian Kesimpulan.

Setelah instruksi disempurnakan untuk menghasilkan keluaran *Markdown* dengan jejak proses keputusan, hasil peringkasan menunjukkan perbaikan kualitatif berupa pembagian topik yang lebih sesuai dengan isi rapat, narasi yang mempertahankan keterkaitan antara usulan dan keputusan, serta keterkaitan eksplisit antara pernyataan dan pembicara apabila disebutkan dalam transkrip. Perbaikan ini konsisten dengan tujuan perancangan yang menetapkan notulen bukan sekadar memuat kesimpulan akhir, melainkan juga jejak proses yang melatarbelakangi tercapainya kesimpulan tersebut, mencakup pihak yang mengajukan usulan, pertimbangan yang dibahas, dan dasar pengambilan keputusan

Evaluasi teknis terhadap waktu pemrosesan pada dua berkas audio rapat nyata berukuran berbeda, yaitu berkas pertama berukuran 83,82 megabyte dan berkas kedua berukuran 120,73 megabyte, menunjukkan bahwa peningkatan ukuran berkas sebesar 44% (dari 83,82 menjadi 120,73 *megabyte*) hanya berdampak pada peningkatan waktu pemrosesan sebesar 29% (dari 17 menit 52 detik menjadi 23 menit 4 detik). Temuan ini mengindikasikan bahwa waktu pemrosesan tidak meningkat secara proporsional linear terhadap ukuran berkas, kemungkinan disebabkan oleh komponen waktu tetap pada setiap permintaan API yang proporsinya mengecil relatif terhadap total waktu pemrosesan pada berkas yang lebih besar. Observasi ini bersifat awal dan terbatas pada dua titik data, sehingga belum dapat digeneralisasi sebagai pola yang konsisten tanpa pengujian pada sampel yang lebih banyak dan bervariasi.

EXPLORER
Journal of Computer Science and Information Technology
E-ISSN 2774-4647
Website: https://journal.fkpt.org/index.php/Explorer **DOI:** https://doi.org/10.47065/explorer.v6i2.3013

Terkait pemilihan model large language model berkapasitas kecil (GPT-4o-mini) yang didasarkan pada pertimbangan efisiensi biaya dan kemudahan implementasi, hasil demonstrasi menunjukkan bahwa setelah instruksi disempurnakan, model tersebut mampu menghasilkan notulen dengan kualitas yang dinilai memadai untuk kebutuhan penelitian ini. Namun, evaluasi ini tidak mencakup perbandingan langsung dan terukur antara GPT-4o-mini dengan model berkapasitas lebih besar pada transkrip yang identik, sehingga *trade-off* antara efisiensi biaya model berkapasitas kecil dan potensi kualitas keluaran yang lebih baik dari model berkapasitas lebih besar tidak dapat dipastikan secara kuantitatif melalui penelitian ini.
Evaluasi melalui tinjauan oleh pemangku kepentingan eksternal (*expert review*) terhadap hasil akhir, yang diidentifikasi sebagai agenda lanjutan karena penilaian kualitas notulen pada penelitian ini masih bersifat formatif berdasarkan pengamatan peneliti belum dilakukan pada tahap penelitian ini.



Gambar 8. Grafik waktu proses audio

3.5 Perbandingan dengan Penelitian Terdahulu

Penelitian terdahulu mengenai peringkasan rapat berbasis large language model menunjukkan orientasi yang secara umum berbeda dari penelitian ini. Penelitian oleh Asthana dan rekan-rekannya berfokus pada evaluasi pengalaman pengguna terhadap dua bentuk tampilan recap, tanpa membahas arsitektur sistem secara mendalam maupun mekanisme verifikasi terhadap keluaran model sebelum ditampilkan kepada pengguna. Studi lain mengusulkan pendekatan peringkasan secara real-time selama rapat berlangsung [6]. sebuah orientasi yang berbeda dari penelitian ini yang secara sengaja dirancang untuk beroperasi pasca-rapat, mengingat keberadaan tahap verifikasi manusia pada transkrip dan hasil peringkasan tidak memungkinkan pemrosesan secara *real-time* tanpa intervensi pengguna. Penelitian oleh Jagadesh dan rekan-rekannya menitikberatkan pada rangkaian teknik pemrosesan bahasa alami untuk menghasilkan notulen dari rekaman audio, namun tidak menyertakan mekanisme verifikasi manusia maupun pembahasan mengenai bagaimana notulen yang dihasilkan dapat merepresentasikan proses pengambilan keputusan, sebagaimana menjadi fokus pada penelitian ini.
Perbedaan paling mendasar antara penelitian ini dan ketiga penelitian tersebut terletak pada penyisipan dua titik verifikasi manusia secara eksplisit pada arsitektur sistem, yaitu verifikasi transkrip sebelum peringkasan dan verifikasi hasil notulen sebelum publikasi. Pendekatan ini sejalan dengan prinsip Human-in-the-Loop yang menekankan keterlibatan manusia pada tahap kritis pemrosesan oleh model kecerdasan buatan, sebuah karakteristik yang belum menjadi fokus eksplisit pada ketiga penelitian pembanding tersebut [11]. Selain itu, penelitian ini juga menempatkan notulen sebagai dokumen yang merepresentasikan proses pengambilan keputusan, bukan sekadar ringkasan hasil akhir, sebuah orientasi yang juga belum menjadi fokus pada penelitian-penelitian terdahulu yang lebih menitikberatkan pada kualitas ringkasan secara teknis.
Perbedaan orientasi ini juga mencerminkan perspektif yang lebih luas dari penelitian integrasi AI dalam alur kerja organisasi, yang menunjukkan bahwa hambatan adopsi sistem otomasi sering kali bukan semata-mata teknis, melainkan berkaitan dengan resistensi psikologis dan kekhawatiran akan berkurangnya peran manusia dalam proses kerja [28]. Pendekatan verifikasi manusia yang diterapkan pada penelitian ini secara langsung mengakomodasi kekhawatiran tersebut dengan memposisikan manusia sebagai validator aktif dalam alur otomasi, bukan sebagai pihak yang digantikan oleh sistem.

EXPLORER
Journal of Computer Science and Information Technology
E-ISSN 2774-4647
Website: https://journal.fkpt.org/index.php/Explorer DOI: https://doi.org/10.47065/explorer.v6i2.3013

Tabel 3. Perbandingan Dimensi Penelitian dengan Penelitian Terdahulu

Dimensi perbandingan	Penelitian ini	Asthana dkk.	Liu, Lu, Xu	Jagadesh dkk.
Verifikasi manusia eksplisit	Ya	Tidak	Tidak	Tidak
Notulen sebagai jejak proses keputusan	Ya	Sebagian	Tidak	Tidak
Arsitektur tanpa infrastruktur server	Ya	Tidak dibahas	Tidak dibahas	Tidak dibahas
Operasi pasca-rapat (bukan real-time)	Ya	Ya	Tidak	Ya

Pada sisi arsitektur sistem, penelitian ini juga berbeda dari pendekatan yang umum digunakan pada penelitian penerapan industri, yang cenderung mengasumsikan ketersediaan infrastruktur backend untuk orkestrasi proses. Penelitian ini menunjukkan bahwa arsitektur client-side integration yang menjalankan seluruh logika orkestrasi pada sisi peramban tanpa infrastruktur server tambahan dapat menjadi alternatif yang relevan bagi organisasi dengan kapasitas pengelolaan teknologi informasi terbatas, sekalipun keputusan arsitektural tersebut dicapai melalui eliminasi sistematis terhadap pendekatan berbasis platform orkestrasi n8n yang terbukti tidak optimal akibat kompleksitas konfigurasi dan kebutuhan infrastruktur kontainerisasi.

4. KESIMPULAN

Penelitian ini berhasil merancang, mengimplementasikan, dan mendemonstrasikan arsitektur sistem otomasi notulen rapat berbasis client-side integration dengan dua titik verifikasi manusia pada tahap transkripsi dan peringkasan, melalui tujuh iterasi arsitektural yang mencerminkan karakteristik Design Science Research sebagai pembelajaran melalui artefak nyata. Arsitektur yang dihasilkan berkontribusi sebagai improvement [17] berupa model yang dapat diadaptasi organisasi dengan kapasitas teknologi informasi terbatas dan dijalankan tanpa instalasi server. Keterbatasan penelitian ini mencakup belum dilakukannya expert review, pengukuran kuantitatif yang masih terbatas pada dua titik data, belum adanya perbandingan langsung antara model berkapasitas kecil dan besar, serta sistem yang belum mendukung pemisahan pembicara dan integrasi otomatis dengan modul penjadwalan dan presensi, yang seluruhnya menjadi agenda penelitian lanjutan.

REFERENSI

[1] W. Standaert, S. Muylle, and A. Basu, “Business meetings in a postpandemic world: When and how to meet virtually,” *Bus. Horiz.*, vol. 65, no. 3, pp. 267–275,, doi: 10.1016/j.bushor.2021.02.047.
[2] N. Lehmann-Willenbrock, S. G. Rogelberg, J. A. Allen, and J. E. Kello, “The critical importance of meetings to leader and organizational success: Evidence-based insights and implications for key stakeholders,” *Organ. Dyn.*, vol. 47, no. 1, pp. 32–36,, doi: 10.1016/j.orgdyn.2017.07.005.
[3] A. S. H. Sheikhi, E. Iosif, O. Basov, and I. Dionysiou, “SmartMinutes—A Blockchain-Based Framework for Automated, Reliable, and Transparent Meeting Minutes Management,” *Big Data Cogn. Comput.*, vol. 6, no. 4, 2022, doi: 10.3390/bdcc6040133.
[4] H. Zhang, P. S. Yu, and J. Zhang, “A Systematic Survey of Text Summarization: From Statistical Methods to Large Language Models,” *ACM Comput. Surv*, vol. 57, no. 11, Art. 277, doi: 10.1145/3731445.
[5] S. Asthana, S. Hilleli, P. N. He, and A. Halfaker, “Summaries, Highlights, and Action Items: Design, Implementation and Evaluation of an LLM-powered Meeting Recap System,” *Proc. ACM Hum.-Comput. Interact*, vol. 9, no. 2, Art. CSCW176.
[6] X. Liu, J. Lu, and A. Xu, “Dynamic agenda-aware real-time meeting summarization with large language models,” *J. King Saud Univ. Comput. Inf. Sci*, vol. 37, p. 275.
[7] D. Sanapala, K. Choudhary, and S. Shetty, *Multi-Person Speech Curator for Minutes of Meetings along with Meeting Summarization and Language Translation*, vol. 1, no. 1. Association for Computing Machinery, 2023. doi: 10.1145/3633598.3633600.
[8] V. Rennard, G. Shang, J. Hunter, and M. Vazirgiannis, “Abstractive meeting summarization: A survey,” *Trans. Assoc. Comput. Linguist.*, vol. 11, pp. 861–884, 2023, doi: 10.1162/tacl_a_00578.
[9] M. T. R. Laskar, X.-Y. Fu, C. Chen, and S. B. TN, “Building Real-World Meeting Summarization Systems using Large Language Models: A Practical Perspective,” *dalam Proc*, pp. 343–352.
[10] X.-Y. Fu, M. T. R. Laskar, E. Khasanova, C. Chen, and S. B. TN, “Tiny Titans: Can Smaller Large

Language Models Punch Above Their Weight in the Real World for Meeting Summarization?," in *dalam Proc. 2024 Conf. North American Chapter Assoc. Comput. Linguistics: Human Lang. Technol*, Ind. Track), Mexico City, pp. 387–394.

[11] K. Lazaros, A. G. Vrahatis, and S. Kotsiantis, "Human-in-the-Loop Artificial Intelligence: A Systematic Review of Concepts, Methods, and Applications," *Entropy*, vol. 28, no. 4, Art. 377, doi: 10.3390/e28040377.

[12] A. R. A. Zahra, "Javanese and Sundanese speech recognition using Whisper," *Comput. Sci. Inf. Technol*, vol. 6, no. 3, pp. 253–261,, doi: 10.11591/csit.v6i3.p253-261.

[13] S. Z. Pranida, M. C. Airlangga, R. A. Genadi, and S. Shehata, "ASR Under Noise: Exploring Robustness for Sundanese and Javanese."

[14] GmbH, "n8n — Workflow Automation," *Juni, Diakses*. [Online]. Available: https://n8n.io.

[15] A. Pawar, Y. Meng, L. Tang, and Z. Xi, "Improving Clinical Data Accessibility Through Automated FHIR Data Transformation Tools."

[16] K. Peffers, T. Tuunanen, M. A. Rothenberger, and S. Chatterjee, "A Design Science Research Methodology for Information Systems Research," *J. Manag. Inf. Syst*, vol. 24, no. 3, pp. 45–77,.

[17] S. G. A. R. Hevner, "Positioning and Presenting Design Science Research for Maximum Impact," *MIS Q.*, vol. 37, no. 2, pp. 337–355,.

[18] M. D. Network, "MediaDevices: getDisplayMedia() method," *Web APIs Doc.*, [Online]. Available: https://developer.mozilla.org/en-US/docs/Web/API/MediaDevices/getDisplayMedia.

[19] W. W. R.-T. C. W. Group, "Issue #144: allow getDisplayMedia 'granted' permission to persist during the user session," *mediacapture-screen-share GitHub Repos.*, [Online]. Available: https://github.com/w3c/mediacapture-screen-share/issues/144.

[20] OpenAI, "Audio API FAQ," *OpenAI Help Cent.*, [Online]. Available: https://help.openai.com/en/articles/7031512-whisper-api-faq.

[21] R. S. A. P. A. Amrullah, "Analysis of Whisper Automatic Speech Recognition Performance on Low Resource Language," *J. Pilar Nusa Mandiri*, vol. 20, no. 1, pp. 1–8,, doi: 10.33480/pilar.v20i1.4633.

[22] K. L. LLP, "Board minutes under the microscope: Evidentiary value, privilege pitfalls and post-crisis reconstruction," *Kennedys Law Insights*, [Online]. Available: https://www.kennedyslaw.com/en/thought-leadership/article/2026/board-minutes-under-the-microscope-evidentiary-value-privilege-pitfalls-and-post-crisis-reconstruction/.

[23] J. Jiménez, "Automating IETF Insights generation with AI," in *dalam Proc. Internet Res. Task Force (IRTF) RASPG*, Dublin, Irlandia.

[24] "Account restricted to AQ. prefix keys instead of standard AIza format," *Google AI Developer Forum*.

[25] "Issue with WhatsApp Business API Webhook and n8n (Not Receiving Messages," *n8n Community Forum*, [Online]. Available: https://community.n8n.io/t/issue-with-whatsapp-business-api-webhook-and-n8n-not-receiving-messages/89184.

[26] S. B. R., M. C. R., and S. B. F, "Chunking Strategies for Multimodal AI Systems," 2025. doi: https://arxiv.org/abs/2512.00185.

[27] W.-T. Lee, K. Kamahori, and B. Kasikci, "MURMUR: An Efficient Inference System for Long-Form ASR."

[28] S. Faria, "AI Integration in Organisational Workflows: A Case Study on Job Reconfiguration, Efficiency, and Workforce Adaptation," *Information*, vol. 16, no. 9, Art. no. 764, doi: 10.3390/info16090764.