

PENGEMBANGAN BOT NOTULA DENGAN INTEGRASI MODEL *WHISPER* PADA *PLATFORM GOOGLE MEET*

Muhammad Ihsan Syafiul Umam^{1*}, Moh. Idris²

^{1,2} Universitas Islam Indonesia, Indonesia

*Corresponding author: 20523217@students.uii.ac.id

Abstract: Time constraints and manual error rates often lead to inaccuracies and inefficiencies in recording online meeting minutes. Many organizations face challenges in accurately and thoroughly documenting meeting discussions, especially when multiple participants are involved. Manual note-taking often cannot keep pace with the speed and volume of information presented, leading to errors and the loss of important data. This study aims to address these issues by developing an automated meeting minutes bot for Google Meet, which utilizes Puppeteer and Whisper technologies to enhance accuracy and efficiency in documentation. By providing the Google Meet URL, the bot automatically joins the meeting, records audio, and transcribes it into text. Integrating Whisper AI for transcription and additional diarization techniques, the bot can accurately identify different speakers and organize the transcription according to their contributions. Developed using a prototyping methodology for continuous improvement, the system consists of two separate services that allow for scalability. Preliminary testing shows promising results, with the bot capable of delivering high-quality transcriptions under various conditions. The use of this bot has the potential to significantly enhance meeting documentation automation, reduce human error, and improve productivity.

Keywords: Automation, Meeting Minutes, Google Meet, Puppeteer, Whisper, Diarization, Audio Transcription.

Abstrak: Keterbatasan waktu dan tingkat kesalahan manual sering kali mengakibatkan ketidakakuratan dan ketidakefisienan dalam pencatatan notula rapat secara daring. Banyak organisasi menghadapi tantangan dalam mencatat diskusi rapat secara menyeluruh dan akurat, terutama ketika melibatkan banyak peserta. Pencatatan manual sering kali tidak dapat mengimbangi kecepatan dan volume informasi yang disampaikan, mengarah pada kesalahan dan kehilangan data penting. Penelitian ini bertujuan untuk mengatasi masalah tersebut dengan mengembangkan bot notula otomatis untuk Google Meet, yang menggunakan teknologi Puppeteer dan Whisper untuk meningkatkan akurasi dan efisiensi pencatatan. Dengan memberikan URL rapat Google Meet, bot ini secara otomatis bergabung dalam rapat, merekam audio, dan mentranskripsikan menjadi teks. Mengintegrasikan Whisper AI untuk transkripsi dan teknik *diarization* tambahan, bot ini dapat dengan akurat mengidentifikasi berbagai pembicara dan mengatur transkripsi sesuai dengan kontribusi masing-masing. Dikembangkan dengan metodologi prototyping untuk perbaikan berkelanjutan, sistem ini terdiri dari dua *service* terpisah yang memungkinkan skalabilitas. Pengujian awal menunjukkan hasil yang menjanjikan, dengan bot mampu memberikan transkripsi berkualitas tinggi dalam berbagai kondisi. Penggunaan bot ini berpotensi meningkatkan otomatisasi dokumentasi rapat, mengurangi kesalahan manusia, dan meningkatkan produktivitas secara signifikan.

Kata kunci: Otomatisasi, Notula Rapat, *Google Meet*, *Puppeteer*, *Whisper*, *Diarization*, Transkripsi Audio.

PENDAHULUAN

Pada era digital yang berkembang pesat saat ini, otomatisasi menjadi salah satu kebutuhan utama dalam berbagai sektor industri. Otomatisasi tidak hanya membantu meningkatkan efisiensi operasional tetapi juga mengurangi beban kerja manual sehingga memungkinkan karyawan untuk fokus pada tugas-tugas yang lebih strategis (Wibawa, 2023). Salah satu area yang mendapat perhatian khusus adalah otomatisasi proses pencatatan notula dalam pertemuan atau rapat terutama yang dilakukan secara daring melalui platform seperti *Google Meet*.

Keterbatasan waktu dan tingkat kesalahan manual sering kali mengakibatkan ketidakakuratan dan ketidakefisienan dalam pencatatan notula rapat secara daring. Proses pencatatan manual memerlukan konsentrasi tinggi dan rentan terhadap kesalahan manusia. Hal tersebut dapat berdampak pada kualitas catatan rapat dan keputusan yang diambil berdasarkan catatan yang dibuat. Untuk itu, solusi teknologi berupa bot notula yang mampu melakukan pencatatan notula secara otomatis dengan tingkat akurasi tinggi sangat dibutuhkan (Manuel et al., 2022).

Salah satu aspek yang perlu diperhatikan dalam proses otomatisasi notula rapat secara daring yaitu transkripsi percakapan dari audio menjadi teks. Penelitian terdahulu telah menunjukkan bahwa penggunaan teknologi *diarization* dapat meningkatkan efisiensi dan akurasi dalam analisa pemrosesan teks ke audio. Misalnya, penelitian oleh Wang & Purushotam (2023) menunjukkan bahwa proses *diarization* dapat memisahkan suara pembicara yang berbeda dalam rekaman audio, sehingga menghasilkan transkripsi yang lebih akurat dan terstruktur. *Diarization* adalah teknik untuk melabeli rekaman audio atau video dengan kelas yang sesuai dengan identitas pembicara, atau singkatnya, untuk mengidentifikasi "siapa yang berbicara kapan (Wang & Purushotam, 2023). Proses ini membagi dan mengelompokkan segmen suara yang sama berdasarkan identitas pembicara, mendeteksi transisi antara pembicara dan perubahan pembicara secara otomatis.

Whisper adalah model kecerdasan buatan yang dikembangkan oleh OpenAI yang berfungsi untuk memproses dan menganalisis data audio (Amorese et al., 2023). *Whisper* dikenal dengan kemampuannya untuk menghasilkan transkripsi audio yang sangat akurat. Meskipun *Whisper* tidak memiliki fitur *diarization* bawaan, model ini dapat digabungkan dengan teknik *diarization* tambahan untuk memisahkan dan

mengidentifikasi pembicara yang berbeda dalam rekaman audio. Umumnya, teknik *diarization* mengintegrasikan beberapa teknik peningkatan kualitas suara, deteksi suara, ekstraksi fitur dan pengelompokan data (Park et al., 2022). Dengan menggunakan teknologi ini, *bot* notula dapat secara akurat mengenali dan memisahkan suara dari berbagai peserta meeting, menghasilkan notula yang lebih terstruktur dan informatif.

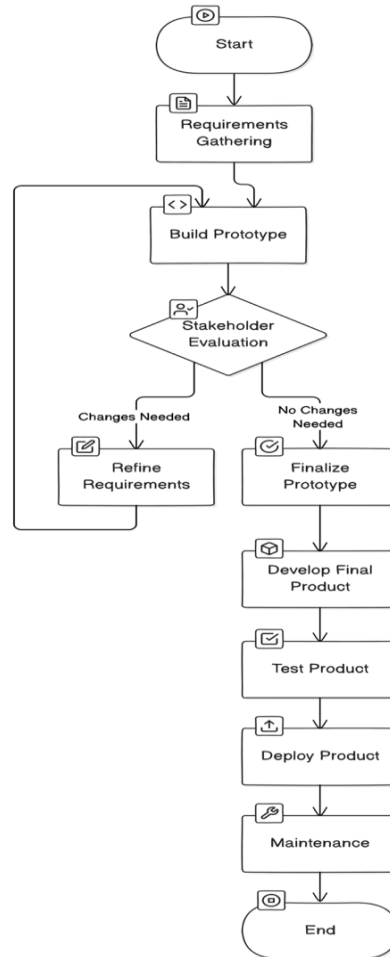
Selain itu, *Puppeteer*, sebuah *library Node.js* yang digunakan untuk mengotomatisasi interaksi dengan *browser*, memungkinkan otomatisasi dalam memasuki dan berinteraksi dengan platform *Google Meet* (Nurkholis & Yunial, 2023). *Puppeteer* dapat melakukan berbagai tugas seperti web scraping, pengambilan data, dan pengujian aplikasi web secara otomatis (srivastava, 2021). Dalam konteks pengembangan *bot* notula, *Puppeteer* digunakan untuk melakukan otomatisasi masuk ke *Google Meet* sebagai peserta dan merekam audio pertemuan. Hal ini memungkinkan bot untuk mengikuti rapat secara efisien tanpa campur tangan manual.

Bot notula dikembangkan dengan dua *service*/layanan terpisah: *Bot Notula Engine* dan Audio Processing dengan tujuan skalabilitas, artinya penulis dapat menyesuaikan skala masing-masing *service* secara independen. Misalnya, jika aplikasi pemrosesan audio membutuhkan lebih banyak sumber daya karena pemrosesan audio yang intensif, maka dapat menyesuaikan kapasitas *service* ini tanpa mempengaruhi kinerja bot *engine*. Kedua *service* ini dapat terintegrasi dengan memanfaatkan konsep *message queueing* yang ada pada teknologi *message broker*, perangkat lunak yang bertindak sebagai perantara dalam sistem komunikasi, memungkinkan aplikasi, layanan, dan sistem yang berbeda untuk berkomunikasi dengan mengirimkan dan menerima pesan. Terdapat banyak perangkat lunak yang berfungsi sebagai *message broker*. Penulis menggunakan *RabbitMQ* karena instalasi dan penggunaannya yang relatif mudah. *RabbitMQ* adalah *message broker* yang menghubungkan layanan-layanan dalam arsitektur *microservice*, membantu mengurangi beban dan waktu pengiriman dengan mengelola pesan antara *publisher* dan *consumer* menggunakan protokol AMQP (Ćatović et al., 2022).

Dengan menggabungkan *Puppeteer* dan *Whisper OpenAI* pada platform *Google Meet*, bot notula diharapkan dapat mengotomatisasi proses pencatatan rapat dengan lebih efisien dan akurat, mengurangi kesalahan manusia, dan meningkatkan produktivitas. Penelitian ini juga akan mengevaluasi efektivitas bot notula ini dalam berbagai kondisi rapat dan memberikan rekomendasi untuk pengembangan lebih lanjut.

METODE

Penelitian ini menggunakan pendekatan rekayasa perangkat lunak dengan mengadopsi metodologi *prototyping*, salah satu metode dalam *System Development Life Cycle* (SDLC). Metode *prototyping* dipilih untuk memungkinkan iterasi cepat dalam pengembangan bot notula, sehingga memfasilitasi umpan balik dan perbaikan yang



berkelanjutan dan mengedepankan kebutuhan pada pengguna (Nur et al., 2023). Diagram alir metode *prototyping* dapat ditemukan pada Gambar 1.

Gambar 1. Metode *Prototyping*

Proses pelaksanaan penelitian ini dibagi menjadi beberapa tahap utama sesuai dengan metodologi *prototyping* sebagai berikut:

1. Pengumpulan Kebutuhan

Tahap pertama dalam pengembangan bot notula ini adalah mengidentifikasi kebutuhan sistem dari *stakeholder* atau dalam kasus ini adalah *VP Engineering* dari perusahaan Widya Robotics. Tahap ini dianggap penting agar produk yang

dikembangkan sesuai dengan apa yang dibutuhkan dan mencakup rincian fitur pada produk (Fitri et al., 2019). Fokus pada tahap ini yaitu memahami fungsi utama yang diharapkan dari bot. Kebutuhan ini meliputi kemampuan bot untuk masuk ke dalam *meeting Google Meet* sebagai peserta, merekam audio selama *meeting* berlangsung, dan mentranskripsikan audio tersebut menjadi teks. Bot harus dapat beroperasi secara otomatis tanpa intervensi pengguna setelah konfigurasi awal.

Setelah kebutuhan dasar diidentifikasi, maka proses berikutnya mengumpulkan spesifikasi teknis dan non-teknis yang diperlukan untuk pengembangan bot. Spesifikasi teknis mencakup perangkat lunak dan perangkat keras yang akan digunakan dalam pengembangan dan implementasi bot. Misalnya, perangkat lunak yang digunakan termasuk *Puppeteer* untuk automasi *browser*, *Whisper* dari *OpenAI* untuk transkripsi audio, dan *Pynote.audio* untuk *diarization*. Perangkat keras yang dibutuhkan meliputi komputer dengan spesifikasi yang memadai untuk menjalankan proses otomatisasi dan pemrosesan audio. Pada tahap ini *stakeholder* juga memberikan daftar kebutuhan minimal yang harus ada pada prototipe agar mendapatkan masukan secepat mungkin dari *stakeholder*, ini juga disebut dengan istilah *minimum viable product* (Ningsih et al., 2023).

2. Pengembangan Prototipe

Pengembangan *prototype* awal dimulai setelah kebutuhan sistem dikumpulkan dan dianalisis. Prototipe ini berfungsi sebagai versi awal dari sistem yang dapat diuji dan diperbaiki berdasarkan umpan balik dari *stakeholder*. Prototipe yang dikembangkan bukan berupa tampilan pengguna, melainkan produk yang mencakup *minimum viable product* yang ditentukan oleh *stakeholder*.

3. Evaluasi dan Perbaikan Prototipe

Prototipe awal yang telah dikembangkan kemudian dievaluasi dengan melibatkan *stakeholder* untuk mendapatkan umpan balik mengenai fungsionalitas dan kinerja bot. *Stakeholder* mencoba prototipe dalam berbagai skenario meeting dan memberikan umpan balik mengenai aspek-aspek yang perlu diperbaiki atau ditingkatkan. Pada proses ini juga dilakukan peninjauan kembali terhadap kebutuhan dengan melakukan penyesuaian dan merincikan kebutuhan dalam hal yang lebih kecil sehingga dapat memastikan semua kebutuhan dapat terpenuhi dengan baik.

4. Finalisasi Pengembangan Prototipe

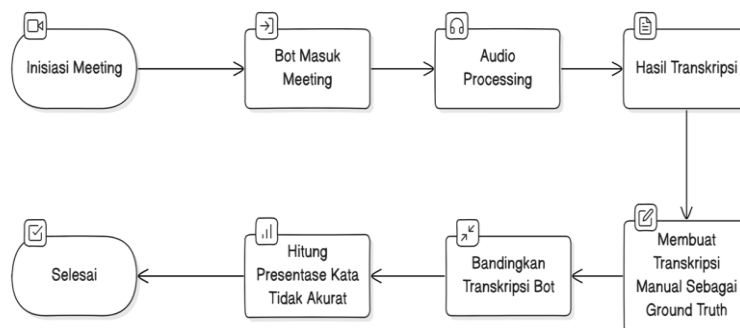
Jika tidak ada perubahan yang diperlukan, prototipe akan disempurnakan berdasarkan umpan balik yang diperoleh. Prototipe yang sudah disempurnakan ini akan menjadi dasar untuk pengembangan produk akhir. Pada tahap ini penulis melengkapi kebutuhan-kebutuhan yang belum dikerjakan pada tahap pengembangan prototipe sebelumnya.

5. Pengembangan Produk Tahap Akhir

Setelah prototipe tahap akhir telah berhasil memenuhi semua kebutuhan yang telah ditetapkan, langkah selanjutnya adalah melakukan pengembangan produk final, tahap ini mencakup meningkatkan performa, memperbaiki bug pada produk yang ditemukan pada saat pengembangan, dan peninjauan kembali terhadap struktur kode agar dapat lebih mudah dilakukan *maintenance*.

6. Pengujian

Pengujian merupakan tahap penting dalam pengembangan bot notula yang bertujuan untuk memastikan bahwa semua fungsi dan fitur yang telah dikembangkan berjalan sesuai yang diharapkan. Pengujian dilakukan untuk memvalidasi performa, keandalan, dan kesesuaian bot dalam lingkungan produksi. Dalam konteks ini, pengujian menggunakan *ground truth*, atau kebenaran dasar, diterapkan untuk memastikan keakuratan dan konsistensi hasil yang dihasilkan oleh bot. Metodenya adalah dengan membandingkan hasil transkripsi dari bot dengan transkripsi manual yang dilakukan oleh manusia (*ground truth*). Evaluasi dilakukan berdasarkan kesesuaian isi dan kesalahan pada transkripsi. Diagram alir pengujian dapat ditemukan pada Gambar 2.



Gambar 2. Alur Pengujian

7. Product Deployment

Produk yang telah memenuhi semua kebutuhan dan telah lolos dalam pengujian akan dijalankan melalui proses *deployment*, yang merupakan proses untuk

memastikan bahwa produk yang telah dikembangkan dapat diakses oleh pihak-pihak terkait untuk digunakan secara. *Deployment* meliputi instalasi semua dependensi yang diperlukan pada server yang dituju, dan memastikan semua komponen sistem bekerja dengan baik.

8. *Maintenance*

Setelah produk telah melewati tahap *deployment*, maka diasumsikan produk telah dirilis ke pengguna awal untuk melihat bagaimana pengguna berinteraksi dengan produk dan memanfaatkannya, selama proses tersebut mungkin muncul *bug* pada produk atau fitur yang perlu ditambahkan berdasarkan kebiasaan pengguna. *Maintenance* adalah tahap yang mengakomodasi hal-hal tersebut, selain itu pada tahap ini juga akan muncul masalah performa yang disebabkan lonjakan pengguna pada produk, sehingga diperlukan peningkatan performa produk baik dengan cara perbaikan pada kode program, melakukan proses *scaling* pada server baik secara vertikal maupun horizontal.

HASIL DAN PEMBAHASAN

Hasil

Hasil Rekaman Rapat *Google Meet*

Pada tahap implementasi, bot notula dirancang untuk bergabung secara otomatis ke dalam rapat *Google Meet* saat rapat dimulai. Bot ini akan merekam audio selama rapat berlangsung hingga rapat berakhir atau tidak ada lagi peserta yang tersisa di dalam rapat tersebut. Ketika rapat berakhir, proses perekaman audio dihentikan dan *file* hasil perekaman disimpan sementara pada server lokal dalam format WAV.

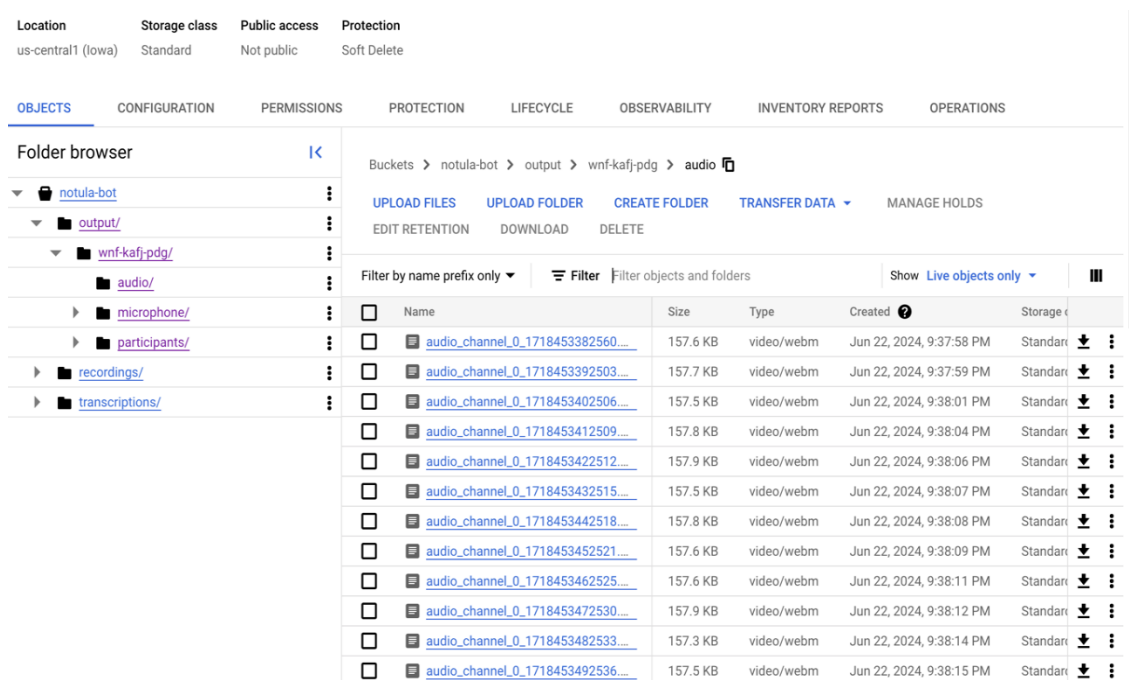
Format WAV (*Waveform Audio File Format*) adalah format audio yang sering digunakan untuk menyimpan data audio dalam bentuk digital yang tidak terkompresi (Santoso & Fakhriza, 2018). Format ini dipilih karena beberapa alasan:

1. Kualitas Audio Tinggi: WAV menyimpan data audio dalam bentuk aslinya tanpa kompresi, sehingga kualitas audio yang dihasilkan sangat tinggi. Hal ini penting untuk memastikan bahwa setiap detail percakapan terekam dengan jelas, yang akan memudahkan proses transkripsi dan analisis.
2. Kesesuaian dengan Model *Whisper*: Model *Whisper* dari *OpenAI* yang digunakan untuk transkripsi memiliki performa yang lebih baik dengan input audio berkualitas

tinggi. Penggunaan format WAV memastikan bahwa input yang diberikan kepada model adalah yang terbaik.

Langkah selanjutnya adalah mengunggah *file* audio yang telah direkam ke *Google Cloud Storage*. Pengunggahan ini bertujuan untuk mengurangi beban penyimpanan pada server lokal dan memastikan *file* audio tersimpan dengan aman dan terstruktur di cloud. Setelah proses unggah berhasil, *file* audio dihapus dari server lokal untuk mengosongkan ruang penyimpanan.

Pada *Google Cloud Storage*, *file* audio disimpan dengan penamaan acak dan unik. Penamaan ini bertujuan untuk memudahkan proses pengolahan di tahap selanjutnya, seperti transkripsi dan analisis. Selain itu, penamaan unik juga mencegah terjadinya duplikasi dan memudahkan dalam mengidentifikasi *file*. Tampilan struktur penyimpanan *file* audio pada *Google Cloud Storage* dapat ditemukan pada Gambar 3:



Gambar 3. Struktur *Google Cloud Storage*

Pada Gambar 3 terlihat bahwa setiap *file* audio disimpan dengan nama yang unik dan ukuran yang bervariasi tergantung pada durasi rapat. Semua *file* tersimpan dalam *bucket* bernama "notula-bot" yang terletak di *region us-central1* (Iowa). Penyimpanan menggunakan kelas standar yang tidak bersifat publik untuk menjaga keamanan data (*Google Cloud Docs*, n.d.).

Perekaman dan penyimpanan otomatis ini menunjukkan efisiensi dalam pengelolaan data rapat, memastikan bahwa setiap percakapan terekam dengan baik dan tersimpan secara aman untuk keperluan selanjutnya. Hal ini juga memungkinkan integrasi lebih lanjut dengan model *Whisper* untuk proses transkripsi otomatis, yang akan dibahas pada bagian selanjutnya dari jurnal ini. Dengan demikian, sistem yang dibangun telah berhasil memenuhi tujuan awal untuk merekam dan menyimpan audio rapat secara efisien, serta menyediakan data yang siap diolah untuk kebutuhan analisis lebih lanjut.

Hasil Pengujian Akurasi Transkripsi

Pengujian akurasi transkripsi dilakukan untuk mengevaluasi performa bot notula yang dikembangkan menggunakan model *Whisper* pada *platform Google Meet*. Metode pengujian yang digunakan adalah *Word Error Rate* (WER), yang merupakan metrik standar dalam mengevaluasi sistem pengenalan suara. WER menghitung kesalahan transkripsi dengan mempertimbangkan substitusi, penghapusan, dan penyisipan kata (Morris et al., 2004).

Penelitian oleh Morris et al. (2004) menjadi referensi dasar pengembangan sebuah pustaka *jiwer*, sebuah pustaka yang digunakan untuk memudahkan proses menghitung WER yang melibatkan tokenisasi teks, menghitung substitusi, penghapusan, dan penyisipan kata. Pengukuran menggunakan WER penting dilakukan untuk mengevaluasi kinerja model *whisper* dan mengetahui seberapa akurat hasil transkripsinya (Aminudin et al., 2024). Rumus yang digunakan untuk menentukan WER juga tertera dalam penelitian tersebut sebagai berikut:

$$WER = \frac{S + D + I}{N}$$

Keterangan:

- S adalah jumlah substitusi (kata yang salah ditranskripsikan).
- D adalah jumlah penghapusan (kata yang hilang dalam transkripsi).
- I adalah jumlah penyisipan (kata tambahan dalam transkripsi).
- N adalah jumlah total kata dalam teks referensi (*ground truth*).

Sedangkan untuk mengetahui nilai presentase akurasi transkripsi dapat dihitung dengan rumus berikut:

$$Presentase\ akurasi = (1 - WER) \times 100 \%$$

Selama proses pengembangan, pengujian terbatas dilakukan dengan menggunakan bot notula pada empat rapat singkat yang memiliki variasi kualitas audio, yaitu Tinggi, Sedang, dan Rendah. Berikut adalah kriteria nya:

- Kualitas Audio Tinggi: Audio jernih dengan sedikit atau tanpa gangguan latar belakang, serta volume yang konsisten.
- Kualitas Audio Sedang: Audio masih dapat dipahami, namun terdapat sedikit ketidakjelasan pada pengucapan kata atau terdapat kebisingan latar belakang yang tidak menghalangi pemahaman.
- Kualitas Audio Rendah: Audio memiliki banyak gangguan, seperti kebisingan latar belakang yang tinggi, distorsi, atau volume yang tidak konsisten, sehingga sulit dipahami.

Tabel 1. Pengujian Akurasi Bot Notula dengan Metode Ground Truth

No	Ground Truth	Transkripsi Bot	Kualitas Audio	WER	Akurasi (%)
1	<p>[00:00:000.48 - 00:00:002.40] Speaker 1: guys liburan nanti mau kemana nih? [00:00:004.19 - 00:00:005.79] Speaker 2: Gimana kalau ke Bali? Pantainya seru banget. [00:00:008.38 - 00:00:010.23] Speaker 1: Setuju sih, tapi kita harus cari tempat [00:00:010.23-00:00:010.96] Speaker 1: Penginapan yang murah [00:00:012.37 - 00:00:014.15] Speaker 2: Tenang, aku punya list tempat keren. [00:00:014.15 - 00:00:014.83] Speaker 2: Kapan mau booking?</p>	<p>[00:00:000.48 - 00:00:002.40] Speaker 1: guys liburan nanti mau kemana nih [00:00:004.19 - 00:00:005.79] Speaker 2: Gimana kalau ke Bali? Pantainya seru banget. [00:00:008.38 - 00:00:010.23] Speaker 1: Tujuh sih, tapi kita harus cari tempat [00:00:010.23-00:00:010.96] Speaker 1: Minat yang murah [00:00:012.37 - 00:00:014.15] Speaker 2: Tenang, aku punya list tempat keren. [00:00:014.15 - 00:00:014.83] Speaker 2: Kapan mau booking?</p>	Tinggi	0.06	93.75%
2	<p>[00:00:000.11 - 00:00:002.23] Speaker 1: acara kampus, tinggal seminggu lagi, siap gak? [00:00:003.67 - 00:00:005.47] Speaker 2: Siap dong, siapa yang handle dekorasinya. [00:00:006.74 - 00:00:009.52] Speaker 1: Aku boleh sih bantuin buat konsep yang keren. [00:00:012.74 - 00:00:015.54] Speaker 2: boleh banget aku bawa speaker ya biar makin asik. [00:00:016.25 - 00:00:016.73] Speaker 1: Oke, siap.</p>	<p>[00:00:000.11 - 00:00:002.23] Speaker 1: acara kampus, tinggal seminggu lagi, siap gak [00:00:003.67 - 00:00:005.47] Speaker 2: Siap dulu, siapa yang handle dekorasinya. [00:00:006.74 - 00:00:009.52] Speaker 1: Aku boleh sih bantuin buat konsep yang keren. [00:00:012.74 - 00:00:015.54] Speaker 2: boleh banget aku bawa speaker ya biar makin asik [00:00:016.25 - 00:00:016.73] Speaker 1: Oke, siap.</p>	Tinggi	0.06	93.75%
3	<p>[00:00:000.04 - 00:00:002.18]</p>	<p>[00:00:000.04 - 00:00:002.18]</p>	Sedang	0.13	87.10%

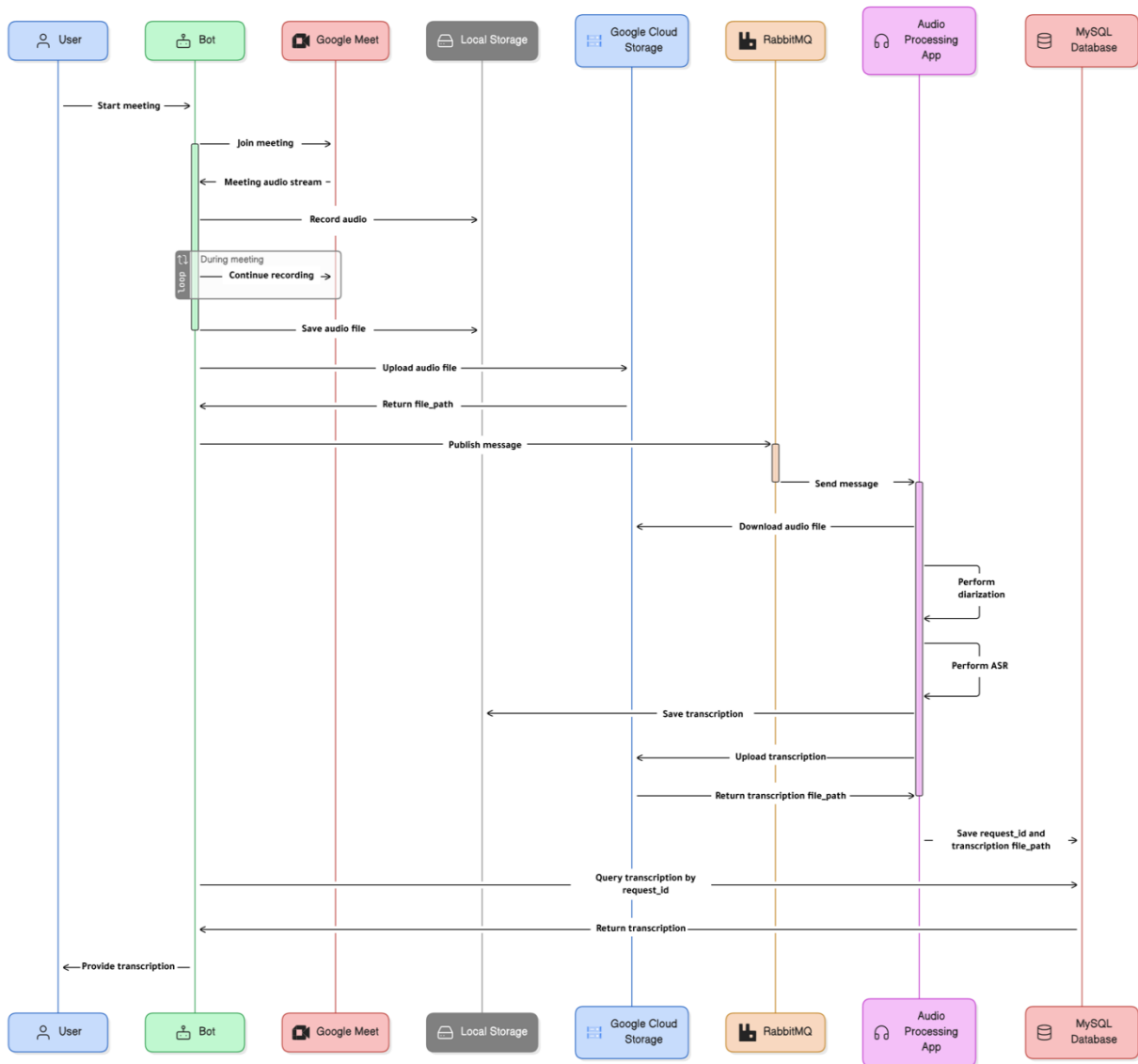
No	Ground Truth	Transkripsi Bot	Kualitas Audio	WER	Akurasi (%)
	<p>Speaker 1: Mau belajar bareng buat ujian minggu depan? [00:00:004.14 - 00:00:005.82]</p> <p>Speaker 2: boleh, tapi materinya apa dulu nih [00:00:009.08 - 00:00:011.52]</p> <p>Speaker 1: mulai dari bab terakhir banyak yang susah [00:00:012.98 - 00:00:015.94]</p> <p>Speaker 2: Setuju sih, biar gak pusing. Kita sambil makan pizza yuk. [00:00:017.59 - 00:00:017.69]</p> <p>Speaker 1: Oke.</p>	<p>Speaker 1: Mau belajar bareng buat ujian minggu depan? [00:00:004.14 - 00:00:005.82]</p> <p>Speaker 2: boleh, tapi materinya apa dulu nih [00:00:009.08 - 00:00:011.52]</p> <p>Speaker 1: mulai dari web terakhir banyak yang susah [00:00:012.98 - 00:00:015.94]</p> <p>Speaker 2: Setuju sih, biar gak pusing. Kita sambil makan pizza yuk. [00:00:017.59 - 00:00:017.69]</p> <p>Speaker 1: Terima kasih.</p>			
4	<p>[00:00:000.00 - 00:00:000.90]</p> <p>Speaker 1: jogging bareng nanti? [00:00:002.28 - 00:00:004.80]</p> <p>Speaker 2: Sabtu pagi boleh sih, gimana? Biar sekalian seger. [00:00:007.28 - 00:00:007.64]</p> <p>Speaker 1: Bisa sih. [00:00:008.31 - 00:00:009.59]</p> <p>Speaker 1: habis jogging kita sarapan bareng ya [00:00:010.97 - 00:00:013.43]</p> <p>Speaker 2: Boleh, sarapan tempat biasa ya, kopinya enak.</p>	<p>[00:00:000.00 - 00:00:000.90]</p> <p>Speaker 1: Kita jadiin baru nanti. [00:00:002.28 - 00:00:004.80]</p> <p>Speaker 2: Sabtu pagi boleh sih, gimana? Biar sekalian seger. [00:00:007.28 - 00:00:007.64]</p> <p>Speaker 1: Biasa sih. [00:00:008.31 - 00:00:009.59]</p> <p>Speaker 1: habis jogging kita sarapan bareng [00:00:010.97 - 00:00:013.43]</p> <p>Speaker 2: Boleh, sarapan tempat biasa ya, kopinya enak.</p>	Rendah	0.23	76.92%

Berdasarkan hasil penelitian pada Tabel 1, Bot notulensi menunjukkan performa yang baik pada kualitas audio tinggi, dengan tingkat kesalahan kata (WER) yang rendah dan akurasi tinggi sebesar 93.75%. Pada kualitas audio sedang, performa bot masih cukup baik dengan akurasi 87.10%. Namun, pada kualitas audio rendah, bot mengalami kesulitan dengan tingkat kesalahan yang lebih tinggi dan akurasi yang lebih rendah, yaitu 76.92%. Kualitas audio sangat mempengaruhi akurasi transkripsi bot, di mana kualitas audio yang tinggi memberikan hasil transkripsi yang lebih akurat dibandingkan dengan kualitas audio yang rendah.

Pembahasan

Arsitektur bot notula dirancang untuk memastikan alur kerja yang efisien dan terintegrasi dari perekaman hingga penyimpanan hasil transkripsi. Bot ini dirancang untuk melakukan perekaman dan transkripsi audio dari rapat *Google Meet* secara otomatis dengan dua *service* terpisah: *Bot Notula Engine* dan *Audio Processing*. Untuk

memahami arsitektur dan proses yang terjadi pada bot notula dapat dilihat pada Gambar 4.



Gambar 4. Arsitektur Bot Notula

Aplikasi bot bertugas untuk melakukan automasi masuk ke dalam rapat *Google Meet* sebagai peserta dan melakukan perekaman audio. Aplikasi ini dikembangkan menggunakan *framework Nestjs*, yang dipilih karena modularitasnya, memungkinkan pengembangan dan pemeliharaan kode yang lebih mudah dan terstruktur. Proses perekaman audio dilakukan dengan merekam audio menjadi bagian kecil berdurasi masing-masing 10 detik. Pemecahan audio ke dalam potongan kecil bertujuan untuk mengurangi beban unggahan ke *Google Cloud Storage*, serta mempermudah manajemen dan pemrosesan audio. Potongan audio dikirimkan secara bertahap ke

Google Cloud Storage. Setelah rapat selesai *file path* yang didapatkan dari hasil unggah ke *Cloud Storage* diteruskan ke aplikasi kedua untuk memproses audio melalui mengirimkan pesan yang berisi *file path audio* dan *request_id* ke *RabbitMQ*.

Service kedua bertugas untuk melakukan pemrosesan audio berdasarkan hasil rekaman *bot-notula-engine*. Servis ini dikembangkan menggunakan bahasa pemrograman *Python*, yang terkenal dengan pustaka dan dukungan kuat untuk pemrosesan audio dan *machine learning*. Berikut adalah proses kerja aplikasi pemrosesan audio:

1. Menerima *path file* audio dari *RabbitMQ* dan mengunduh semua potongan audio dari *Google Cloud Storage*.
2. Potongan audio digabungkan menjadi satu file audio utuh untuk memudahkan proses pemrosesan lebih lanjut.
3. *Diarization* dilakukan menggunakan pustaka *pyannote* audio untuk memisahkan dan mengidentifikasi pembicara yang berbeda dalam rekaman audio.
4. Proses *Automatic Speech Recognition* (ASR) dilakukan menggunakan model *Whisper* dari *OpenAI* untuk mentranskripsikan audio menjadi teks.
5. Hasil transkripsi diunggah kembali ke *Google Cloud Storage*, dan *path file* transkripsi beserta *request_id* disimpan di *MySQL Database*.

Arsitektur ini memastikan bahwa setiap komponen dalam proses perekaman dan transkripsi bekerja secara sinergis untuk menghasilkan transkripsi yang akurat dan dapat diandalkan. Proses modular dan terpisah memungkinkan pengembangan, pemeliharaan, dan peningkatan yang lebih mudah seiring dengan perkembangan kebutuhan dan teknologi.

Hasil pengujian Tabel 1 menunjukkan bahwa bot notula memiliki tingkat akurasi yang tinggi dalam mentranskripsikan percakapan pada *Google Meet*. Pengukuran akurasi menggunakan *Word Error Rate* (WER) menunjukkan hasil yang bervariasi namun pada rapat dengan audio yang tinggi dan sedang cukup akurat. Misalnya, dalam beberapa skenario percakapan, bot mampu mencapai WER 0.13, menunjukkan hanya terdapat beberapa kasus di mana kesalahan minor terjadi, seperti substitusi kata yang serupa tetapi tidak mempengaruhi pemahaman keseluruhan teks.

Secara umum, bot menunjukkan kemampuan yang baik dalam menangani percakapan informal dalam bahasa Indonesia gaul, yang sering kali memiliki struktur

dan kosakata yang berbeda dari bahasa formal. Kemampuan ini menunjukkan fleksibilitas model *Whisper* yang digunakan. Meski terdapat beberapa kesalahan minor, performa bot dalam pengujian ini menunjukkan bahwa bot notula sudah siap untuk digunakan dalam lingkungan nyata seperti di perusahaan atau institusi pendidikan untuk membantu dokumentasi rapat dan diskusi.

Pengujian ini juga mengindikasikan bahwa dengan penyesuaian dan peningkatan lebih lanjut, terutama dalam menangani variasi bahasa dan kosakata yang lebih luas, bot dapat lebih meningkatkan keakuratannya. Integrasi teknologi pendukung seperti *Natural Language Processing* (NLP) juga dapat membantu dalam menyempurnakan transkripsi dan pengenalan konteks percakapan yang lebih kompleks.

Secara keseluruhan, hasil pengujian ini memberikan gambaran bahwa bot notula yang dikembangkan memiliki performa yang memadai dan dapat diandalkan untuk mentranskripsikan percakapan di *Google Meet*. Dengan peningkatan berkelanjutan dan penyesuaian berdasarkan hasil pengujian, bot ini memiliki potensi besar untuk menjadi alat yang sangat berguna dalam mendokumentasikan rapat online secara efisien dan akurat.

SIMPULAN

Pengembangan bot notula dengan integrasi model *Whisper* pada *platform Google Meet* telah berhasil menunjukkan kemampuan untuk melakukan perekaman dan transkripsi otomatis dengan akurasi yang cukup tinggi yaitu 0.13 WER artinya hampir menyentuh akurasi 94% untuk rapat dengan kualitas audio yang tinggi. Berdasarkan hasil pengujian dan analisis, bot ini mampu mengotomatisasi seluruh proses dari bergabungnya ke rapat, merekam audio, hingga mentranskripsikan pembicaraan menjadi teks yang dapat diakses oleh pengguna. Proses ini tidak hanya meningkatkan efisiensi dalam pembuatan notula rapat tetapi juga memastikan bahwa semua percakapan terekam dan terdokumentasi dengan baik tanpa intervensi manual. Adapun substansi utama dari penelitian ini adalah:

1. **Automasi:** Bot notula mampu bergabung dalam rapat *Google Meet*, merekam audio, dan mentranskripsikannya secara otomatis, menghilangkan kebutuhan intervensi manusia dan memungkinkan pengguna untuk fokus pada konten rapat.
2. **Integrasi Modular:** Penggunaan *framework Nestjs* untuk *bot-notula-engine* dan *Python* untuk aplikasi pemrosesan audio menunjukkan bahwa pendekatan modular

dan terpisah ini efektif dalam menangani tugas yang kompleks dan memungkinkan skalabilitas serta pemeliharaan yang lebih mudah.

3. **Akurasi Tinggi:** Dengan menggunakan model *Whisper* dari *OpenAI* untuk ASR dan *pyannote audio* untuk diarization, sistem ini mampu menghasilkan transkripsi dengan akurasi yang memadai, meskipun pada variasi audio dengan kualitas rendah menunjukkan adanya ruang untuk peningkatan lebih lanjut.

Prospek pengembangan hasil penelitian ini sangat luas. Beberapa arah pengembangan di masa depan meliputi:

1. **Peningkatan Akurasi:** Penggunaan model ASR yang lebih canggih atau penggabungan beberapa model dapat meningkatkan akurasi transkripsi terutama pada audio kualitas rendah.
2. **Fitur Tambahan:** Pengembangan fitur tambahan seperti identifikasi emosi dalam suara, deteksi kata kunci penting, atau integrasi dengan kalender dan sistem manajemen proyek untuk tindakan otomatis setelah rapat.
3. **Skalabilitas:** Mengoptimalkan sistem untuk menangani banyak rapat secara bersamaan atau mengintegrasikannya dengan *platform* rapat daring lainnya selain *Google Meet*.

Prospek aplikasi penelitian ini juga menjanjikan. Bot notula ini dapat digunakan di berbagai sektor seperti pendidikan, bisnis, dan pemerintahan di mana rapat dan diskusi sering dilakukan. Implementasi bot ini dapat meningkatkan produktivitas, memastikan kepatuhan terhadap dokumentasi rapat, dan menyediakan aksesibilitas yang lebih baik bagi mereka yang membutuhkan notula rapat yang terperinci.

Dengan demikian, penelitian ini tidak hanya memenuhi tujuan awal yang dinyatakan dalam pendahuluan tetapi juga membuka peluang baru untuk pengembangan teknologi otomatisasi rapat yang lebih canggih dan serbaguna di masa depan.

DAFTAR RUJUKAN

- Aminudin, Nuryasin, I., Wicaksono, G., Chandranegara, D., Thoifah, I., Rizky, W., Ferdiansyah, D., Azzahra, K., Lathifah, F., & Aulyah, K. (2024). *Pengolahan Korpus Dataset Audio Bacaan AlQur'an Menggunakan Metode Wav2Vec 2.0*. <https://doi.org/http://dx.doi.org/10.26418/jp.v10i1>
- Amorese, T., Greco, C., Cuciniello, M., Milo, R., Sheveleva, O., & Glackin, N. (2023). *Automatic speech recognition (ASR) with Whisper: Testing Performances in Different Languages*. <http://ceur-ws.org>

- Ćatović, A., Buzadžija, N., & Lemes, S. (2022). Microservice development using RabbitMQ message broker. *Science, Engineering and Technology*, 2(1), 30–37. <https://doi.org/10.54327/set2022/v2.i1.19>
- Fitri, R., Hastuti, I., & Negeri Banjarmasin, P. (2019). *Analisis Rekayasa Kebutuhan Perangkat Lunak Menggunakan Smart Requirement Dalam Membangun Sistem Informasi Lomba Karya Tulis Ilmiah Politeknik Negeri Banjarmasin*.
- Google Cloud Docs. (n.d.). *Use public access prevention | Cloud Storage | Google Cloud*. Retrieved June 13, 2024, from <https://cloud.google.com/storage/docs/using-public-access-prevention?hl=en>
- Manuel, M., Menon, A. S., Kallivayalil, A., Isaac, S., & Professor, L. K. S. (2022). Automated Generation of Meeting Minutes Using Deep Learning Techniques. *International Journal of Computing and Digital Systems*, 12(1), 109–120. <https://doi.org/10.12785/ijcds/120110>
- Morris, A. C., Maier, V., & Green, P. (2004). From WER and RIL to MER and WIL: Improved evaluation measures for connected speech recognition. *8th International Conference on Spoken Language Processing, ICSLP 2004*, 2765–2768. <https://doi.org/10.21437/interspeech.2004-668>
- Ningsih, A. A., Rambe, R. S., Munthe, Y. N., Sialalahi, P. R., William, J., Ps, I. V., Estate, M., Percut, K., Tuan, S., & Serdang, K. D. (2023). *Pendekatan Lean Startup Pada Desain Produk Dan Teknik Minimum Viable Product Dalam Menyikapi Skeptisisme Pada Iklim Bisnis* (Vol. 2, Issue 1).
- Nur, S., Waita, R., & Asa, B. J. (2023). Rancang Bangun Sistem Informasi Desa Fudima Dengan Menggunakan Metode Prototype Di Desa Fudima. *EDUSAINTEK: Jurnal Pendidikan, Sains Dan Teknologi*, 10(3), 804–815. <https://doi.org/10.47668/edusaintek.v10i3.862>
- Nurkholis, I., & Yunial, A. H. (2023). Automation Testing Tool Dalam Pengujian Website Sistem Informasi Pembayaran Iuran Dan LKS Di MTS Al-Ittihad Menggunakan Puppeteer. *LOGIC: Jurnal Ilmu Komputer Dan Pendidikan*, 2(1), 101–108. <https://doi.org/10.4230/OASICS.ICPEC.2022.10>
- Park, T. J., Kanda, N., Dimitriadis, D., Han, K. J., Watanabe, S., & Narayanan, S. (2022). A review of speaker diarization: Recent advances with deep learning. *Computer Speech & Language*, 72, 101317. <https://doi.org/10.1016/J.CSL.2021.101317>
- Santoso, H., & Fakhriza, M. (2018). Perancangan Aplikasi Keamanan File Audio Format WAV (WAVEFORM) Menggunakan Algoritma RSA. In *ALGORITMA: Jurnal Ilmu Komputer dan Informatika*.
- srivastava, pranjali. (2021). *Node.js Puppeteer - GeeksforGeeks*. <https://www.geeksforgeeks.org/node-js-puppeteer/>
- Wang, M. Y., & Purushotam, G. N. (2023). Using Deep Learning and Augmented Reality to Improve Accessibility: Inclusive Conversations Using Diarization, Captions, and Visualization. *2023 ASEE Annual Conference & Exposition*. <https://peer.asee.org/44572>

Wibawa, S. (2023). Analisis Chatbot Otomatisasi Tugas Administratif dan Manajemen Dalam Lingkungan Digital Dengan Menggunakan Python. In *Sains Teknik Elektro* (Vol. 4, Issue 1). <http://jurnal.bsi.ac.id/index.php/insantek>