



KERJA PRAKTIK - EF234603

Pengembangan Aplikasi *Artificial Intelligence* (AI) untuk Konversi Audio dan Video ke *Minutes of Meeting* (MoM) Menggunakan *Large Language Model* (LLM)

PT. ALGONACCI SOBAT NUSANTARA

Jl. Letjen S. Parman No.28, Grogol Petamburan, Kota Jakarta Barat,
Daerah Khusus Ibukota Jakarta.

Periode: 17 Februari 2025 - 17 Mei 2025

Oleh:

Rayssa Ravelia

5025211219

Pembimbing Jurusan

Bintang Nuralamsyah, S.Kom, M.Kom

Pembimbing Lapangan

Eric Julianto

DEPARTEMEN TEKNIK INFORMATIKA

Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember

Surabaya 2025



KERJA PRAKTIK - EF234603

Pengembangan Aplikasi *Artificial Intelligence* (AI) untuk Konversi Audio dan Video ke *Minutes of Meeting* (MoM) Menggunakan *Large Language Model* (LLM)

PT. ALGONACCI SOBAT NUSANTARA

Jl. Letjen S. Parman No.28, Grogol Petamburan, Kota Jakarta Barat,
Daerah Khusus Ibukota Jakarta.

Periode: 17 Februari 2025 - 17 Mei 2025

Oleh:

Rayssa Ravelia 5025211219

Pembimbing Jurusan

Bintang Nuralamsyah, S.Kom, M.Kom

Pembimbing Lapangan

Eric Julianto

DEPARTEMEN TEKNIK INFORMATIKA

Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember

Surabaya 2025

[Halaman ini sengaja dikosongkan]

DAFTAR ISI

DAFTAR ISI	iii
DAFTAR GAMBAR	vii
DAFTAR TABEL	xi
DAFTAR ALGORITMA	xiii
LEMBAR PENGESAHAN	xv
ABSTRAK	xvii
KATA PENGANTAR	xix
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Tujuan.....	2
1.3 Manfaat.....	2
1.4 Rumusan Masalah.....	3
1.5 Lokasi dan Waktu Kerja Praktik.....	3
1.6 Metodologi Kerja Praktik.....	4
1.6.1 Perumusan Masalah.....	4
1.6.2 Studi Literatur.....	4
1.6.3 Analisis dan Perancangan Sistem.....	5
1.6.4 Implementasi Sistem.....	5
1.6.5 Pengujian dan Evaluasi.....	6
1.6.6 Kesimpulan dan Saran.....	6
1.7 Sistematika Laporan.....	7
1.7.1 Bab I Pendahuluan.....	7

1.7.2	Bab II Profil Perusahaan.....	7
1.7.3	Bab III Tinjauan Pustaka	7
1.7.4	Bab IV Analisis dan Perancangan Sistem.....	7
1.7.5	Bab V Implementasi Sistem	7
1.7.6	Bab VI Pengujian dan Evaluasi	8
1.7.7	Bab VII Kesimpulan dan Saran	8
BAB 2	PROFIL PERUSAHAAN.....	9
2.1	Profil PT. Algonacci Sobat Nusantara	9
2.2	Lokasi.....	10
BAB 3	TINJAUAN PUSTAKA.....	11
3.1	Minutes of Meeting (MoM)	11
3.2	Teknologi Speech-to-Text.....	12
3.3	Whisper AI sebagai Model Transkripsi	13
3.4	<i>Large Language Model (LLM)</i> dalam Ringkasan Teks	16
3.5	<i>GPT-4</i> sebagai Model Ringkasan MoM	18
3.6	Chunking dan Tokenization dalam Pemrosesan Teks ...	20
3.7	LangChain sebagai Kerangka Kerja untuk LLM	21
3.8	LangSmith sebagai Alat Monitoring dan Evaluasi LLM	24
3.9	Pengembangan Aplikasi Berbasis FastAPI dan Streamlit	25
3.10	Pembuatan dan Ekspor Dokumen PDF	28
BAB 4	ANALISIS DAN PERANCANGAN SISTEM.....	31
4.1.	Analisis Sistem.....	31

4.1.1	Definisi Umum Aplikasi.....	31
4.1.2	Penamaan dan Desain Logo Aplikasi.....	32
4.1.3	Analisis Kebutuhan Fungsional.....	33
4.1.4	Analisis Kebutuhan Non-Fungsional.....	34
4.2.	Perancangan Sistem	36
4.1.5	Pengumpulan Data Uji (Audio/Video).....	38
4.1.6	Pembuatan Prototipe Hardcoded	39
4.1.7	Pengembangan Website MoMify	42
4.1.8	Integrasi Prototipe dengan Website MoMify.....	45
4.1.9	Pengujian dan Evaluasi.....	47
BAB 5	IMPLEMENTASI SISTEM.....	51
5.1	Implementasi Pengembangan Aplikasi.....	51
5.1.1	Struktur Halaman Aplikasi	51
5.1.2	Navigasi Antar Halaman.....	53
5.1.3	Pengunggahan File dan Validasi Format.....	62
5.2	Integrasi <i>Backend</i> dan LLM.....	64
5.2.1	Implementasi API <i>Backend</i> dengan FastAPI.....	65
5.2.2	Pemrosesan Transkripsi Menggunakan Whisper API	67
5.2.3	Chunking dan Tokenization dalam Ringkasan Teks .	69
5.2.4	Ringkasan MoM Menggunakan GPT-4.....	72
5.3	Pembuatan dan Pengelolaan Output MoM	76
5.3.1	Format dan Struktur Dokumen MoM.....	76
5.3.2	Ekspor MoM ke PDF.....	79

5.3.3	Penanganan Eror dan Log Pemrosesan.....	81
5.4	Repositori dan Kode Sumber	85
BAB 6	PENGUJIAN DAN EVALUASI.....	87
6.1	Tujuan Pengujian	87
6.2	Lingkungan Pengujian	87
6.2.1	Spesifikasi Perangkat Keras	88
6.2.2	Spesifikasi Perangkat Lunak.....	89
6.3	Kriteria Pengujian	90
6.4	Skenario Pengujian	91
6.5	Evaluasi Pengujian.....	93
6.5.1	Evaluasi Fungsionalitas	93
6.5.2	Evaluasi Kuantitatif.....	95
BAB 7	KESIMPULAN DAN SARAN	97
7.1	Kesimpulan	97
7.2	Saran	98
DAFTAR PUSTAKA		101
BIODATA PENULIS.....		105

DAFTAR GAMBAR

Gambar 2.1 Logo Perusahaan Algo Network.....	9
Gambar 3.1 Arsitektur Whisper AI: Pendekatan Sequence-to-Sequence dengan Transformer untuk Transkripsi Multibahasa dan Terjemahan Otomatis	15
Gambar 3.2 Proses chunking membagi dokumen panjang, merangkum tiap bagian dengan LLM, lalu menggabungkannya menjadi ringkasan akhir	21
Gambar 3.3 Arsitektur pipeline LangChain yang menerapkan retrieval-augmented generation (RAG).....	23
Gambar 4.1 Logo aplikasi MoMify.....	32
Gambar 4.2 Diagram alir perancangan sistem secara keseluruhan	37
Gambar 4.3 Diagram alir proses prototipe hardcoded MoMify, mulai dari ekstraksi audio hingga pembuatan Minutes of Meeting (MoM) dalam format PDF	41
Gambar 4.4 Diagram alir proses antarmuka pengguna MoMify yang mencakup unggahan file, kustomisasi, validasi, dan pembuatan output PDF	44
Gambar 4.5 Diagram alir integrasi frontend (Streamlit) dan backend (FastAPI) dalam proses unggah, transkripsi, ringkasan, dan ekspor MoM ke PDF.....	46
Gambar 4.6 Diagram alir proses pengujian dan evaluasi sistem MoMify, mulai dari pengujian transkripsi dan ringkasan hingga perbandingan hasil dengan referensi manual serta optimasi jika diperlukan.....	48
Gambar 5.1 Tampilan Landing Page MoMify yang berisi pengantar tentang Minutes of Meeting dan cara kerja aplikasi.....	54

Gambar 5.2 Tampilan expander pada MoMify yang menjelaskan konsep Minutes of Meeting (MoM), manfaatnya, dan contoh format MoM profesional.....	55
Gambar 5.3 Tampilan expander yang menjelaskan cara kerja MoMify, mulai dari pengunggahan file, transkripsi menggunakan Whisper AI, ringkasan dengan GPT-4, hingga ekspor Minutes of Meeting (MoM) dalam format PDF.	55
Gambar 5.4 Tampilan setelah pengguna mengklik "Let's Get Started", di mana mereka dapat menyesuaikan format Minutes of Meeting (MoM) dengan memilih font, warna teks tebal, dan bahasa sebelum mengunggah file audio atau video.	56
Gambar 5.5 Tampilan opsi pemilihan font pada fitur Customize Your MoM, di mana pengguna dapat memilih berbagai jenis font untuk format Minutes of Meeting (MoM) dalam file PDF.	57
Gambar 5.6 Tampilan fitur Customize Your MoM yang memungkinkan pengguna memilih warna untuk teks tebal dalam file Minutes of Meeting (MoM) menggunakan pemilih warna (color picker).....	57
Gambar 5.7 Tampilan fitur Select Notes Language yang memungkinkan pengguna memilih bahasa untuk Minutes of Meeting (MoM)	58
Gambar 5.8 Proses pemilihan file oleh pengguna untuk diunggah ke MoMify, sebelum dilakukan transkripsi dan pembuatan Minutes of Meeting (MoM).....	59
Gambar 5.9 Tampilan proses pemrosesan file setelah pengguna menekan tombol "Process File Now"	59
Gambar 5.10 Tampilan setelah seluruh proses selesai, dengan notifikasi processing complete dan tombol Download PDF untuk mengunduh hasil Minutes of Meeting (MoM)	60

Gambar 5.11 Contoh output Minutes of Meeting (MoM) dalam format PDF pada halaman 1 dari 2, menampilkan informasi seperti judul rapat, peserta, agenda, dan poin diskusi.....	61
Gambar 5.12 Contoh output Minutes of Meeting (MoM) dalam format PDF pada halaman 2 dari 2, menampilkan tindakan yang diambil, kesimpulan, dan tanggal rapat berikutnya.....	61

[Halaman ini sengaja dikosongkan]

DAFTAR TABEL

Tabel 4.1 Daftar format file audio dan video beserta jumlah sampel dan durasi yang digunakan sebagai data uji dalam pengembangan dan pengujian sistem MoMify.....	39
Tabel 4.2 Metrik evaluasi sistem MoMify beserta deskripsi dan skala penilaian	49
Tabel 5.1 Perbandingan ukuran model Whisper berdasarkan jumlah parameter, kebutuhan VRAM, dan kecepatan relatif terhadap waktu nyata.	69
Tabel 6.1 Spesifikasi Perangkat Keras yang Digunakan dalam Pengujian Aplikasi MoMify	88
Tabel 6.2 Spesifikasi Perangkat Lunak yang Digunakan dalam Pengujian Aplikasi MoMify	89
Tabel 6.3 Hasil Evaluasi Fungsionalitas MoMify	93
Tabel 6.4 Hasil Evaluasi Kuantitatif MoMify	95

[Halaman ini sengaja dikosongkan]

DAFTAR ALGORITMA

Algoritma 5.1 Struktur Halaman Aplikasi.....	52
Algoritma 5.2 Pseudocode untuk Pengunggahan File.....	62
Algoritma 5.3 Pseudocode untuk Validasi dan Ekstraksi Audio...	63
Algoritma 5.4 Pseudocode untuk menangani unggahan file	65
Algoritma 5.5 Pseudocode untuk menangani unduhan file hasil MoM.....	66
Algoritma 5.6 Pseudocode untuk transkripsi audio.....	68
Algoritma 5.7 Pseudocode Menghitung Jumlah Token	70
Algoritma 5.8 Pseudocode membagi teks menjadi chunks	70
Algoritma 5.9 Pseudocode untuk Meringkas Transkripsi ke Format MoM.....	73
Algoritma 5.10 Pseudocode untuk Pemformatan Struktur MoM dalam PDF.....	77
Algoritma 5.11 Pseudocode untuk Ekspor and Unduh MoM sebagai PDF.....	80
Algoritma 5.12 Pseudocode untuk Penanganan File Error.....	81
Algoritma 5.13 Pseudocode untuk Tahapan Log Proses.....	82
Algoritma 5.14 Pseudocode untuk Mencoba Proses	83
Algoritma 5.15 Pseudocode untuk menangani sistem eror	84

[Halaman ini sengaja dikosongkan]

LEMBAR PENGESAHAN

KERJA PRAKTIK

Pengembangan Aplikasi *Artificial Intelligence* (AI) untuk Konversi Audio dan Video ke *Minutes of Meeting* (MoM) Menggunakan *Large Language Model* (LLM)

Oleh:

Rayssa Ravelia 5025211219

Disetujui oleh Pembimbing Kerja Praktik:

1. Bintang Nuralamsyah,
S.Kom, M.Kom.
NIP. 198603122012122004



(Pembimbing Departemen)

2. Eric Julianto



(Pembimbing Lapangan)

[Halaman ini sengaja dikosongkan]

Pengembangan Aplikasi *Artificial Intelligence* (AI) untuk Konversi Audio dan Video ke *Minutes of Meeting* (MoM) Menggunakan *Large Language Model* (LLM)

Nama Mahasiswa : Rayssa Ravelia
NRP : 5025211219
Departemen : Teknik Informatika FTEIC-ITS
Pembimbing Departemen : Bintang Nuralamyah, S.Kom, M.Kom.
Pembimbing Lapangan : Eric Julianto

ABSTRAK

Pencatatan Minutes of Meeting (MoM) secara manual sering kali tidak efisien dan rentan terhadap hilangnya informasi penting. Untuk mengatasi hal tersebut, dikembangkan MoMify, sebuah aplikasi berbasis AI yang secara otomatis mengonversi rekaman audio atau video menjadi ringkasan MoM yang terstruktur. Sistem ini mengintegrasikan Whisper untuk proses transkripsi dan GPT-4 untuk penyusunan ringkasan, dengan dukungan antarmuka berbasis FastAPI dan Streamlit yang memungkinkan komunikasi frontend-backend yang lancar.

MoMify mendukung berbagai format dan durasi file serta menyediakan opsi kustomisasi dokumen seperti jenis huruf, warna, dan bahasa. Hasil evaluasi menunjukkan bahwa sistem memiliki performa optimal, dengan nilai Word Error Rate (WER) yang rendah, Readability Score yang tinggi, serta waktu pemrosesan yang efisien. Dengan kemampuan tersebut, MoMify mampu meningkatkan kualitas dan kecepatan dokumentasi rapat, baik untuk kebutuhan organisasi maupun individu.

Kata Kunci: Artificial Intelligence, Large Language Model, Minutes of Meeting, Natural Language Processing, Speech-to-Text

[Halaman ini sengaja dikosongkan]

KATA PENGANTAR

Puji syukur kehadiran Tuhan Yesus Kristus atas segala rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan laporan Kerja Praktik untuk mata kuliah Kerja Praktik dengan judul: "Pengembangan Aplikasi *Artificial Intelligence* (AI) untuk Konversi Audio dan Video ke *Minutes of Meeting* (MoM) Menggunakan *Large Language Model* (LLM)." Laporan ini disusun sebagai salah satu bentuk pertanggungjawaban akademik selama pelaksanaan Kerja Praktik yang dilakukan di PT. ALGONACCI SOBAT NUSANTARA.

Laporan ini mencakup berbagai proses yang dilakukan, mulai dari analisis kebutuhan, perancangan sistem, implementasi aplikasi, hingga pengujian. Penulis berharap laporan ini dapat memberikan gambaran yang jelas mengenai langkah-langkah pengembangan aplikasi dan bermanfaat bagi pembaca sebagai referensi di masa mendatang. Penulis juga menyadari bahwa laporan ini masih memiliki banyak kekurangan, baik dari segi isi maupun penyajiannya. Oleh karena itu, kritik dan saran yang membangun sangat diharapkan untuk perbaikan di masa mendatang.

Pada kesempatan ini, penulis ingin mengucapkan terima kasih yang sebesar-besarnya kepada pihak-pihak yang telah memberikan dukungan, baik secara langsung maupun tidak langsung, sehingga laporan ini dapat terselesaikan dengan baik. Ucapan terima kasih ini penulis sampaikan kepada:

1. Orang tua, atas doa, dukungan, dan kasih sayang yang tiada henti.
2. Bapak Bintang Nuralamsyah, S.Kom, M.Kom., selaku dosen pembimbing, atas arahan dan bimbingannya selama pelaksanaan Kerja Praktik ini.

3. Mas Eric Julianto, selaku direktur utama PT. ALGONACCI SOBAT NUSANTARA, atas kesempatan yang diberikan kepada penulis untuk belajar dan berkembang di perusahaan
4. Mas Dian Alhusari, selaku HR PT. ALGONACCI SOBAT NUSANTARA, yang telah merekrut dan memfasilitasi proses administrasi penulis dalam perusahaan
5. Seluruh anggota perusahaan PT. ALGONACCI SOBAT NUSANTARA yang telah membagikan ilmu dan pengalaman selama periode Kerja Praktik.
6. Teman-teman Teknik Informatika ITS angkatan 2025, khususnya sahabat seperjuangan selama kuliah yaitu Rr. Diajeng, Wan Sabrina, Lita, Dian, Osvaldo, dan Victor, atas dukungan, tawa, dan kebersamaan yang sangat berarti.
7. Teman-teman lainnya yang tidak dapat disebutkan satu per satu, namun turut membantu secara langsung maupun tidak langsung dalam proses ini.
8. Steve, yang telah mendukung secara emosional dan mental, memberikan, motivasi, dan pengertian selama penulis menjalani proses Kerja Praktik

Penulis menyadari bahwa laporan ini masih memiliki kekurangan, sehingga kritik dan saran yang membangun sangat penulis harapkan demi perbaikan di masa mendatang. Semoga laporan ini dapat memberikan manfaat bagi semua pihak yang membacanya.

Surabaya, 29 Mei 2025
Rayssa Ravelia

1.1 Latar Belakang

Minutes of Meeting (MoM) atau risalah rapat merupakan dokumen penting yang digunakan untuk mencatat poin-poin utama dalam suatu pertemuan, termasuk agenda, diskusi, keputusan, dan tindak lanjut yang harus dilakukan (Neville & Re, 2019). MoM membantu memastikan bahwa seluruh peserta rapat memiliki pemahaman yang sama terhadap hasil rapat dan memudahkan dalam pelacakan keputusan serta tugas yang telah disepakati. Namun, pembuatan MoM secara manual sering kali menjadi proses yang memakan waktu, rentan terhadap kelalaian manusia, serta dapat menghambat produktivitas.

Seiring dengan berkembangnya teknologi kecerdasan buatan (*Artificial Intelligence*), berbagai solusi telah dikembangkan untuk mengotomatisasi proses transkripsi dan analisis teks. Salah satu teknologi yang berperan besar dalam bidang ini adalah *Large Language Model* (LLM), yang mampu memahami, merangkum, serta mengorganisir informasi dari teks dalam berbagai bahasa. Oleh karena itu, pemanfaatan LLM dalam *pembuatan Minutes of Meeting* dapat menjadi solusi yang efektif untuk mengurangi beban kerja manual serta meningkatkan efisiensi dan akurasi dalam pencatatan hasil rapat (Anuj Pandya & Prof. Namrata Gawande, 2022).

Dalam kerja praktik ini, dikembangkan sebuah aplikasi berbasis AI yang dapat mengonversi rekaman audio dan video menjadi *Minutes of Meeting* secara otomatis. Aplikasi ini menggunakan model transkripsi suara berbasis AI untuk mengubah audio menjadi teks, kemudian memanfaatkan LLM untuk menganalisis dan merangkum informasi menjadi MoM yang terstruktur. Selain itu, aplikasi ini dilengkapi

dengan fitur kustomisasi, seperti pemilihan font, warna teks, dan bahasa untuk meningkatkan fleksibilitas dalam penggunaan. Dengan adanya aplikasi ini, diharapkan pembuatan *Minutes of Meeting* dapat menjadi lebih cepat, efisien, serta akurat, sehingga memudahkan pengguna dalam mendokumentasikan hasil rapat secara lebih profesional.

1.2 Tujuan

Tujuan Kerja Praktik ini adalah untuk mengembangkan aplikasi *MoMify*, sebuah sistem berbasis *Artificial Intelligence* (AI) yang mampu mengonversi rekaman audio dan video rapat menjadi *Minutes of Meeting* (MoM) secara otomatis. Dengan memanfaatkan teknologi *Large Language Model* (LLM), aplikasi ini dirancang untuk menghasilkan ringkasan rapat yang terstruktur dan informatif, mencakup elemen penting seperti agenda, diskusi, keputusan, dan tindak lanjut.

Selain itu, kerja praktik ini bertujuan untuk memberikan pengalaman langsung dalam mengimplementasikan teknologi AI dalam pengolahan bahasa alami (*Natural Language Processing*), serta mengintegrasikan berbagai komponen seperti transkripsi otomatis, pemrosesan teks, dan pembuatan dokumen berbasis PDF. Melalui proyek ini, diharapkan pengguna dapat menghemat waktu dalam pembuatan MoM dan meningkatkan akurasi pencatatan hasil rapat. Hasil akhir dari pengembangan ini akan memberikan manfaat bagi individu maupun organisasi dalam mengelola dokumentasi rapat secara lebih efisien dan profesional.

1.3 Manfaat

Pengembangan aplikasi *MoMify* ini diharapkan dapat membantu dalam mengotomatiskan pembuatan *Minutes of Meeting* (MoM) dari rekaman audio dan video. Dengan

memanfaatkan *Large Language Model* (LLM), aplikasi ini mampu merangkum hasil rapat secara cepat dan akurat. Selain itu, fitur kustomisasi seperti pemilihan bahasa, font, dan format dokumen memungkinkan pengguna untuk mendapatkan output yang sesuai dengan kebutuhan mereka.

1.4 Rumusan Masalah

Rumusan masalah yang menjadi fokus dalam kerja praktik ini adalah sebagai berikut:

1. Bagaimana cara mengembangkan sistem otomatisasi *Minutes of Meeting (MoM)* yang dapat mengunggah, memproses, dan meringkas transkripsi dari berbagai format audio dan video dengan tingkat akurasi yang baik?
2. Bagaimana cara melakukan transkripsi dan ringkasan *MoM* dengan struktur yang jelas serta mempertahankan konteks pembicaraan agar informasi yang disampaikan tetap akurat dan koheren?
3. Arsitektur seperti apa yang dapat digunakan dalam aplikasi *MoM* ini agar komunikasi antara *frontend* dan *backend* berjalan secara efisien serta mendukung pemrosesan data secara optimal?
4. Bagaimana cara mengevaluasi kinerja sistem dalam aspek akurasi transkripsi (*Word Error Rate*), kualitas ringkasan (*Readability Score*), serta efisiensi waktu pemrosesan untuk berbagai jenis file dan durasi yang diuji?
5. Bagaimana cara mengimplementasikan fitur ekspor ke *PDF* serta mekanisme penanganan *error handling* yang dapat memberikan notifikasi kepada pengguna saat terjadi kegagalan dalam proses transkripsi atau ringkasan?

1.5 Lokasi dan Waktu Kerja Praktik

Kerja praktik ini dilakukan secara *remote* dengan menggunakan berbagai platform *online* untuk berkomunikasi

dan berkolaborasi dengan tim secara virtual. Pelaksanaan kerja praktik dimulai pada tanggal 17 Februari 2025 hingga 17 Mei 2025, dengan jadwal kerja reguler yang ditetapkan dari setiap hari Senin hingga Jumat.

1.6 Metodologi Kerja Praktik

Metodologi yang diterapkan dalam pelaksanaan kerja praktik ini mencakup langkah-langkah terstruktur yang dirancang untuk memastikan kelancaran dan keberhasilan setiap tahapan kerja. Adapun metodologi yang digunakan dalam kerja praktik ini adalah sebagai berikut:

1.6.1 Perumusan Masalah

Tahapan ini bertujuan untuk mengidentifikasi permasalahan utama yang menjadi fokus dalam pengembangan aplikasi MoMify. Proses dimulai dengan menganalisis kebutuhan pengguna terhadap pembuatan *Minutes of Meeting* (MoM) secara otomatis, mengingat proses pencatatan manual sering kali memakan waktu dan rentan terhadap kesalahan. Permasalahan utama yang diidentifikasi meliputi bagaimana sistem dapat mengonversi rekaman audio atau video menjadi MoM secara akurat, serta bagaimana *Large Language Model* (LLM) dapat digunakan untuk menghasilkan ringkasan yang relevan dan terstruktur.

1.6.2 Studi Literatur

Studi literatur dilakukan untuk memahami teknologi yang relevan dalam pengembangan aplikasi ini. Penelitian dilakukan terhadap metode transkripsi otomatis menggunakan model Whisper AI, teknologi pemrosesan bahasa alami (*Natural Language Processing/NLP*) berbasis LLM, serta metode pembuatan laporan MoM yang umum digunakan dalam lingkungan profesional.

Selain itu, dilakukan eksplorasi terhadap teknik optimasi untuk meningkatkan akurasi transkripsi dan efisiensi sistem dalam memproses file berukuran besar.

1.6.3 Analisis dan Perancangan Sistem

Setelah memahami teknologi yang relevan, dilakukan tahap analisis dan perancangan sistem untuk menentukan arsitektur aplikasi yang akan dikembangkan. Tahapan ini mencakup perancangan alur kerja aplikasi yang dimulai dari tahap unggah file, transkripsi audio, hingga pembuatan *Minutes of Meeting* (MoM). Model Whisper AI diintegrasikan untuk melakukan transkripsi audio dan video, sementara *Large Language Model* (LLM) digunakan untuk menghasilkan ringkasan MoM dalam berbagai bahasa. Selain itu, aplikasi dilengkapi dengan fitur kustomisasi MoM, seperti pemilihan font, warna teks, dan format file output dalam bentuk PDF. Perancangan antarmuka pengguna berbasis Streamlit dilakukan untuk memastikan pengalaman pengguna yang intuitif dan mudah digunakan dalam proses pengunggahan file serta konfigurasi hasil output.

1.6.4 Implementasi Sistem

Pada tahap ini, dilakukan pengkodean dan pengintegrasian berbagai komponen sistem sesuai dengan perancangan yang telah dibuat. Proses implementasi diawali dengan pembangunan backend menggunakan FastAPI untuk menangani permintaan pengguna serta memproses file yang diunggah. Model Whisper AI kemudian diintegrasikan untuk mengubah audio menjadi teks dengan akurasi tinggi. Setelah transkripsi selesai, LLM dimanfaatkan untuk menyusun MoM yang terstruktur serta dapat dikustomisasi sesuai kebutuhan pengguna. Antarmuka pengguna dibangun dengan

Streamlit agar pengguna dapat dengan mudah mengunggah file serta mengatur preferensi untuk font, warna teks, dan format hasil MoM. Selain itu, fitur ekspor MoM dalam format PDF dikembangkan untuk memungkinkan pengguna mendapatkan dokumen yang siap digunakan dengan tampilan yang profesional dan dapat disesuaikan.

1.6.5 Pengujian dan Evaluasi

Setelah sistem berhasil diimplementasikan, dilakukan tahap pengujian untuk memastikan aplikasi berjalan dengan baik dan menghasilkan output yang sesuai dengan kebutuhan pengguna. Pengujian dilakukan dalam beberapa aspek utama. Pengujian fungsional dilakukan untuk memastikan setiap fitur, seperti transkripsi, ringkasan, dan ekspor PDF, berfungsi dengan baik sesuai spesifikasi. Selanjutnya, pengujian akurasi dilakukan untuk mengevaluasi sejauh mana hasil transkripsi dan ringkasan yang dihasilkan oleh LLM dapat menangkap informasi utama dengan baik. Selain itu, pengujian performa dilakukan untuk menilai efisiensi sistem dalam memproses file berukuran besar dan mengukur waktu yang dibutuhkan dalam setiap tahapan. Terakhir, dilakukan uji coba pengguna untuk mendapatkan umpan balik terkait kemudahan penggunaan antarmuka aplikasi serta efektivitas fitur yang disediakan dalam membantu proses pembuatan *Minutes of Meeting*.

1.6.6 Kesimpulan dan Saran

Tahap terakhir adalah menyusun kesimpulan dari hasil kerja praktik yang telah dilakukan. Kesimpulan mencakup pencapaian dalam pengembangan aplikasi, tantangan yang dihadapi, serta potensi pengembangan lebih lanjut. Selain itu, diberikan saran untuk peningkatan sistem, seperti optimalisasi model transkripsi, peningkatan

akurasi ringkasan, serta ekspansi fitur tambahan yang dapat mendukung penggunaan aplikasi dalam skala yang lebih luas.

1.7 Sistematika Laporan

1.7.1 Bab I Pendahuluan

Bab ini membahas latar belakang, tujuan, manfaat, rumusan masalah, lokasi dan waktu pelaksanaan kerja praktik, metodologi yang digunakan, serta sistematika penulisan laporan sebagai panduan dalam memahami isi dokumen secara keseluruhan.

1.7.2 Bab II Profil Perusahaan

Bab ini memberikan gambaran umum tentang organisasi tempat kerja praktik dilaksanakan, mencakup profil, lokasi, serta peran dan tanggung jawab organisasi dalam mendukung pelaksanaan kerja praktik.

1.7.3 Bab III Tinjauan Pustaka

Bab ini memuat dasar teori yang relevan dengan proyek kerja praktik, mencakup teknologi, konsep, dan metode yang digunakan dalam pengembangan sistem atau aplikasi yang menjadi fokus kerja praktik.

1.7.4 Bab IV Analisis dan Perancangan Sistem

Bab ini menjelaskan tahap analisis kebutuhan sistem yang dilakukan untuk menyelesaikan proyek kerja praktik, termasuk perancangan infrastruktur dan komponen sistem yang mendukung implementasi aplikasi.

1.7.5 Bab V Implementasi Sistem

Bab ini menguraikan tahap-tahap yang dilalui selama proses implementasi aplikasi, mulai dari realisasi hasil perancangan hingga penerapan sistem yang telah dikembangkan.

1.7.6 Bab VI Pengujian dan Evaluasi

Bab ini berisi hasil pengujian yang dilakukan terhadap aplikasi yang dikembangkan, termasuk evaluasi kinerja sistem berdasarkan hasil uji coba selama pelaksanaan kerja praktik.

1.7.7 Bab VII Kesimpulan dan Saran

Bab ini memuat kesimpulan dari seluruh tahapan kerja praktik yang telah dilaksanakan serta memberikan saran untuk pengembangan lebih lanjut agar sistem atau aplikasi yang dikembangkan dapat lebih optimal di masa mendatang.

BAB 2 PROFIL PERUSAHAAN

2.1 Profil PT. Algonacci Sobat Nusantara

Algo Network adalah sebuah *software house* yang beroperasi di bawah naungan PT. ALGONACCI SOBAT NUSANTARA, yang menyediakan berbagai layanan untuk mendukung pengembangan teknologi dan solusi bisnis.



Gambar 2.1 Logo Perusahaan Algo Network

Layanan yang ditawarkan mencakup *Data & AI/ML Solutions, Digital Marketing, Software Engineering, Management Consulting, dan Media Production*. Dalam menjalankan layanannya, Algo Network juga memiliki tiga anak perusahaan yang masing-masing fokus pada bidang spesifik:

1. *Braincore*: Laboratorium penelitian yang fokus pada teknologi seperti *AI, IoT, Blockchain, Rekayasa Jaringan, dan Keamanan Siber*.
2. *Megalogic*: *Software house* yang mengembangkan aplikasi *Frontend, Backend, dan Mobile*, serta menyediakan solusi *Quality Assurance*.
3. *BIZZAGI*: Agensi yang menangani layanan digital seperti *Digital Campaign, Ads, SEO, Content Creation, Marketplace Optimization, dan Market Research*.

2.2 Lokasi

Perusahaan ini berlokasi di Jl. Letjen S. Parman No.28, Grogol Petamburan, Kota Jakarta Barat, Daerah Khusus Ibukota Jakarta. Lokasi ini menjadi pusat operasional utama dari PT. ALGONACCI SOBAT NUSANTARA yang menaungi Algo Network.

BAB 3 TINJAUAN PUSTAKA

3.1 Minutes of Meeting (MoM)

Minutes of Meeting (MoM) adalah dokumen yang berisi ringkasan hasil rapat, mencatat poin-poin utama yang dibahas, keputusan yang diambil, serta tindak lanjut yang perlu dilakukan. MoM bertujuan untuk memastikan bahwa semua peserta rapat memiliki pemahaman yang sama mengenai hasil diskusi dan tugas yang harus dilaksanakan setelah rapat selesai (Gibson, 2022). Selain itu, MoM juga berfungsi sebagai arsip resmi yang dapat digunakan sebagai referensi di masa mendatang, terutama dalam situasi di mana diperlukan bukti mengenai keputusan atau kesepakatan tertentu. Dengan adanya MoM, organisasi dapat menghindari kesalahpahaman atau kehilangan informasi penting yang dibahas dalam rapat. Oleh karena itu, penyusunan MoM yang jelas dan sistematis sangat penting dalam lingkungan kerja yang profesional (Neville & Re, 2019).

Format dan struktur MoM umumnya disusun secara sistematis agar mudah dipahami oleh semua pihak yang terlibat. Beberapa elemen penting yang biasanya terdapat dalam MoM antara lain judul rapat, tanggal dan waktu pelaksanaan, daftar peserta, agenda rapat, poin-poin diskusi, keputusan yang diambil, serta daftar tugas atau tindakan yang harus dilakukan oleh masing-masing pihak terkait. Selain itu, beberapa organisasi juga menambahkan bagian kesimpulan serta jadwal rapat berikutnya untuk memastikan kelanjutan dari diskusi yang telah dilakukan. Penggunaan format yang konsisten dalam setiap rapat akan mempermudah pencarian informasi dan pelacakan progres dari tugas-tugas yang telah ditetapkan.

Pencatatan dan dokumentasi rapat dalam lingkungan profesional memiliki peran yang sangat penting dalam menjaga transparansi dan akuntabilitas organisasi. Dengan

adanya MoM, setiap keputusan yang diambil dalam rapat dapat dipertanggungjawabkan dan dijadikan pedoman dalam pelaksanaan pekerjaan. Selain itu, MoM juga membantu dalam mengingatkan peserta rapat mengenai tugas dan tanggung jawab mereka, sehingga mengurangi risiko kelalaian atau kesalahan dalam implementasi keputusan rapat. Dalam perusahaan atau organisasi besar, MoM juga dapat digunakan sebagai alat komunikasi yang efektif bagi anggota tim yang tidak dapat menghadiri rapat, sehingga mereka tetap mendapatkan informasi terkini mengenai perkembangan proyek atau kebijakan yang sedang dibahas (Miura et al., 2020). Oleh karena itu, pencatatan MoM yang baik dapat meningkatkan efisiensi kerja dan memastikan bahwa semua pihak memiliki pemahaman yang sama terhadap hasil rapat.

3.2 Teknologi Speech-to-Text

Teknologi *Speech-to-Text* (STT) merupakan sistem yang memungkinkan konversi otomatis dari suara manusia menjadi teks tertulis (Journal, 2024). Teknologi ini memanfaatkan algoritma pemrosesan bahasa alami (*Natural Language Processing* - NLP) dan *Machine Learning* untuk mengenali pola suara serta menerjemahkannya menjadi kata-kata yang sesuai. STT telah mengalami perkembangan pesat dalam beberapa dekade terakhir dengan meningkatnya akurasi dan kecepatan pemrosesan berkat kemajuan dalam kecerdasan buatan dan *deep learning*. Teknologi ini digunakan dalam berbagai aplikasi seperti asisten virtual, layanan transkripsi otomatis, serta sistem interaksi berbasis suara pada perangkat mobile dan *smart home* (Bera, 2021). Dengan kemampuan untuk mengubah suara menjadi teks, STT memberikan efisiensi yang lebih tinggi dalam berbagai bidang, termasuk bisnis, pendidikan, dan layanan pelanggan.

Beberapa model *speech recognition* yang umum digunakan dalam STT meliputi Hidden Markov Models

(HMM), Deep Neural Networks (DNN), dan Transformer-based models seperti OpenAI Whisper. HMM adalah pendekatan tradisional yang mengandalkan probabilitas statistik untuk mengenali urutan kata berdasarkan pola fonetik. Sementara itu, DNN dan *Recurrent Neural Networks* (RNN) meningkatkan akurasi dengan mempelajari hubungan antara suara dan kata-kata melalui data pelatihan dalam jumlah besar. Model berbasis *transformer*, seperti Whisper AI, lebih lanjut mengoptimalkan konversi suara ke teks dengan menggunakan teknik *self-attention* dan *transfer learning*, sehingga dapat mengenali berbagai aksen dan bahasa dengan tingkat akurasi yang lebih tinggi (Wang et al., 2024). Model-model ini terus diperbarui dan dikembangkan untuk mengatasi berbagai tantangan dalam pengenalan suara.

Meskipun teknologi STT semakin canggih, masih terdapat beberapa tantangan dalam konversi suara ke teks. Salah satu tantangan utama adalah keberagaman aksen, dialek, dan intonasi yang dapat memengaruhi akurasi transkripsi. Selain itu, kebisingan latar belakang dalam rekaman audio juga dapat mengganggu sistem dalam mengenali kata-kata dengan benar. Faktor lain yang mempersulit adalah homonim, yaitu kata-kata yang memiliki pengucapan serupa tetapi memiliki arti berbeda, yang dapat menyebabkan kesalahan dalam transkripsi jika tidak ada konteks yang cukup (Kharitonov et al., 2023). Untuk mengatasi tantangan ini, para peneliti terus mengembangkan model yang lebih kompleks dengan kemampuan memahami konteks kalimat serta meningkatkan toleransi terhadap kebisingan dan variasi suara.

3.3 Whisper AI sebagai Model Transkripsi

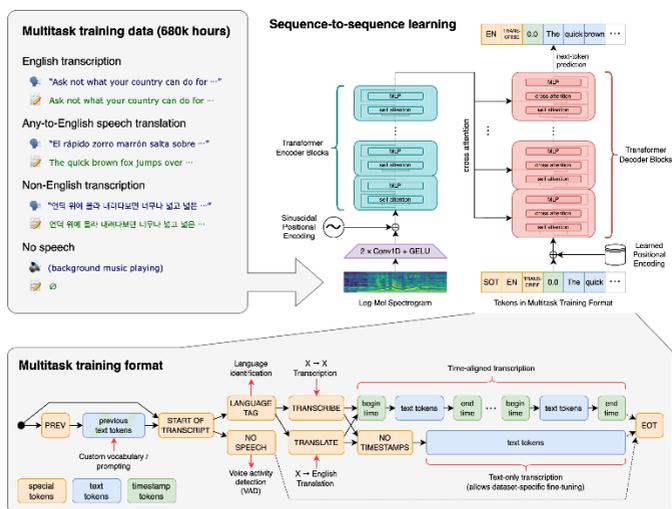
Whisper AI adalah model speech-to-text canggih yang dikembangkan oleh OpenAI untuk melakukan transkripsi otomatis dengan tingkat akurasi tinggi. Model ini

menggunakan arsitektur berbasis transformer, yang memungkinkan pemrosesan teks secara lebih kontekstual dan akurat dibandingkan metode konvensional seperti Hidden Markov Models (HMM) atau Recurrent Neural Networks (RNN)(Radford et al., 2024). Whisper AI dilatih dengan jumlah data yang sangat besar dari berbagai sumber, termasuk percakapan dalam berbagai bahasa dan aksen, sehingga memiliki kemampuan yang lebih baik dalam memahami variasi suara. Selain itu, model ini juga mampu mengenali dan menerjemahkan berbagai bahasa, menjadikannya alat yang sangat fleksibel untuk kebutuhan transkripsi otomatis di berbagai industri.

Salah satu keunggulan utama Whisper AI dibanding model transkripsi lainnya adalah kemampuannya dalam menangani variasi aksen dan kebisingan latar belakang dengan lebih baik. Model ini dilatih dengan data dari berbagai dialek dan lingkungan audio yang beragam, sehingga lebih tahan terhadap gangguan dibandingkan model speech-to-text tradisional(Rosenberg, 2024). Selain itu, Whisper AI mampu melakukan zero-shot learning, yaitu memahami dan menyesuaikan diri dengan bahasa atau aksen baru tanpa perlu pelatihan tambahan. Keunggulan lainnya adalah kemampuannya dalam melakukan segmentasi kalimat dan tanda baca secara otomatis, sehingga hasil transkripsi lebih rapi dan mudah dipahami oleh pengguna.

Pendekatan yang digunakan oleh Whisper AI dapat dilihat pada Gambar 2.1, yang menjelaskan bagaimana model ini bekerja dalam memproses audio menjadi teks. Whisper AI menggunakan pendekatan *sequence-to-sequence learning* dengan arsitektur *transformer* untuk menghasilkan transkripsi dari input audio. Model ini dilatih dengan *multitask training data* sebanyak 680 ribu jam yang mencakup berbagai jenis data, seperti transkripsi bahasa Inggris, terjemahan dari berbagai bahasa ke bahasa Inggris, transkripsi non-Inggris, dan deteksi tanpa suara (*no speech*). Data audio dikonversi

menjadi spektrogram *log-Mel*, yang kemudian diproses oleh *transformer encoder blocks* untuk mengekstrak fitur penting. Informasi ini kemudian diteruskan ke *transformer decoder blocks* untuk melakukan prediksi kata berikutnya berdasarkan token dalam format pelatihan multitasking. Dengan pendekatan ini, Whisper AI mampu menangani berbagai tugas, seperti transkripsi multibahasa, terjemahan otomatis, serta transkripsi dengan atau tanpa penanda waktu (*timestamps*), sehingga menjadikannya model yang fleksibel dan kuat dalam menangani berbagai skenario transkripsi suara ke teks.



Gambar 3.1 Arsitektur Whisper AI: Pendekatan Sequence-to-Sequence dengan Transformer untuk Transkripsi Multibahasa dan Terjemahan Otomatis

Penerapan Whisper AI dalam aplikasi otomatisasi sangat luas, terutama dalam bidang bisnis, pendidikan, dan layanan pelanggan. Dalam dunia bisnis, teknologi ini dapat digunakan untuk mencatat dan merangkum rapat secara otomatis, mengurangi kebutuhan akan pencatatan manual

yang memakan waktu. Di sektor pendidikan, Whisper AI dapat membantu dalam membuat transkrip kuliah dan seminar bagi mahasiswa, termasuk mereka yang memiliki kebutuhan khusus. Selain itu, dalam layanan pelanggan, sistem otomatis berbasis Whisper AI dapat digunakan untuk memahami dan mencatat percakapan antara pelanggan dan agen layanan, memungkinkan analisis lebih lanjut untuk meningkatkan pengalaman pelanggan. Dengan keunggulan dan fleksibilitasnya, Whisper AI menjadi salah satu model transkripsi terbaik yang dapat mendukung berbagai aplikasi berbasis suara.

3.4 *Large Language Model (LLM) dalam Ringkasan Teks*

Large Language Model (LLM) merupakan jenis model kecerdasan buatan yang dirancang untuk memahami, memproses, dan menghasilkan teks dengan tingkat kecerdasan yang menyerupai manusia (Yang et al., 2024). Model ini dilatih dengan jumlah data yang sangat besar dan menggunakan arsitektur *deep learning* berbasis Transformer, seperti yang digunakan dalam GPT-4. Dengan jumlah parameter yang besar, LLM mampu mengenali pola dalam bahasa alami, memahami konteks, dan menghasilkan teks yang koheren serta relevan. Dalam bidang Natural Language Processing (NLP), LLM telah menjadi teknologi fundamental dalam berbagai aplikasi, termasuk penerjemahan, chatbot, analisis sentimen, dan pembuatan ringkasan teks.

Dalam pemrosesan bahasa alami, LLM memiliki peran utama dalam memahami makna teks, menangkap hubungan antar kata dan kalimat, serta menghasilkan respons yang sesuai dengan konteks. Model ini bekerja dengan cara memprediksi kata atau frasa berikutnya berdasarkan input yang diberikan, sehingga dapat menghasilkan teks yang logis dan relevan. Salah satu keunggulan utama LLM adalah kemampuannya dalam memahami bahasa secara kontekstual,

sehingga dapat memberikan ringkasan yang tidak hanya berdasarkan kata-kata individu, tetapi juga mempertimbangkan makna keseluruhan dari dokumen. Dengan demikian, LLM mampu menghasilkan ringkasan yang lebih ringkas, informatif, dan tetap mempertahankan esensi dari teks asli (Chang et al., 2024).

Penerapan LLM dalam sistem otomatisasi dokumen telah memberikan dampak signifikan dalam meningkatkan efisiensi kerja, terutama dalam dunia bisnis dan akademik. Dalam konteks pembuatan *Minutes of Meeting* (MoM), LLM dapat digunakan untuk menganalisis transkrip rapat, mengidentifikasi poin-poin utama, serta menyusun ringkasan yang terstruktur (Asthana et al., 2023). Teknologi ini tidak hanya membantu dalam mempercepat proses dokumentasi, tetapi juga memastikan bahwa catatan yang dihasilkan tetap akurat dan mudah dipahami. Selain itu, LLM juga dapat digunakan dalam pembuatan laporan otomatis, penyusunan artikel berbasis data, dan penyederhanaan teks hukum atau teknis untuk mempermudah pemahaman pengguna.

Salah satu tantangan dalam penggunaan LLM untuk ringkasan teks adalah memastikan bahwa model tidak kehilangan informasi penting dalam proses penyederhanaan. Oleh karena itu, model harus dilatih dengan pendekatan yang tepat agar mampu membedakan informasi yang esensial dari yang kurang relevan. Selain itu, aspek keandalan juga menjadi perhatian utama, terutama dalam lingkungan profesional yang membutuhkan ringkasan dengan tingkat akurasi tinggi. Penggunaan teknik *fine-tuning* atau adaptasi model berdasarkan data spesifik dapat membantu meningkatkan performa LLM dalam menghasilkan ringkasan yang lebih sesuai dengan kebutuhan pengguna.

Dengan kemampuannya yang terus berkembang, LLM menjadi solusi yang semakin banyak diadopsi dalam berbagai industri, termasuk jurnalistik, hukum, pendidikan, dan layanan pelanggan. Kemampuan model ini dalam memahami

dan merangkum teks secara otomatis membuka peluang baru dalam meningkatkan produktivitas dan efisiensi kerja. Seiring dengan kemajuan penelitian di bidang NLP, diharapkan bahwa model LLM akan terus mengalami penyempurnaan, sehingga dapat memberikan hasil yang semakin presisi dan dapat diandalkan dalam berbagai konteks penggunaan.

3.5 GPT-4 sebagai Model Ringkasan MoM

Generative Pre-trained Transformer (GPT) adalah model kecerdasan buatan berbasis transformer yang dikembangkan oleh OpenAI untuk memahami dan menghasilkan teks secara alami. Model ini dilatih menggunakan teknik pembelajaran mendalam dengan jumlah data yang sangat besar agar dapat memahami berbagai bahasa serta menghasilkan teks yang koheren dan relevan. Sejak pertama kali diperkenalkan, GPT telah mengalami beberapa perkembangan, dimulai dari GPT-1 yang masih terbatas dalam pemahaman bahasa, GPT-2 yang lebih baik dalam menghasilkan teks yang natural, hingga GPT-3 yang memiliki kapasitas lebih besar dalam memahami konteks serta menghasilkan respons yang lebih kompleks. GPT-4, sebagai versi terbaru, menghadirkan peningkatan signifikan dalam akurasi, fleksibilitas, dan pemahaman bahasa, menjadikannya salah satu model terbaik dalam pemrosesan bahasa alami (*Natural Language Processing/NLP*) (Katz et al., 2024; OpenAI, 2023).

GPT-4 memiliki keunggulan dalam pemrosesan bahasa alami yang lebih baik dibandingkan pendahulunya. Model ini memiliki parameter yang lebih banyak, yang memungkinkan pemahaman konteks yang lebih mendalam dan kemampuan menangani tugas-tugas kompleks seperti terjemahan, analisis teks, serta pembuatan ringkasan. Salah satu fitur unggulan dari GPT-4 adalah kemampuannya untuk menghasilkan teks yang lebih konsisten dan sesuai dengan kebutuhan pengguna,

serta mendukung berbagai bahasa dengan akurasi tinggi. Selain itu, GPT-4 dapat memahami instruksi dengan lebih baik dan menghasilkan keluaran yang lebih relevan dengan konteks input yang diberikan. Hal ini menjadikannya pilihan ideal untuk berbagai aplikasi yang membutuhkan otomatisasi dokumen dan pemrosesan teks tingkat lanjut(Liu et al., 2023).

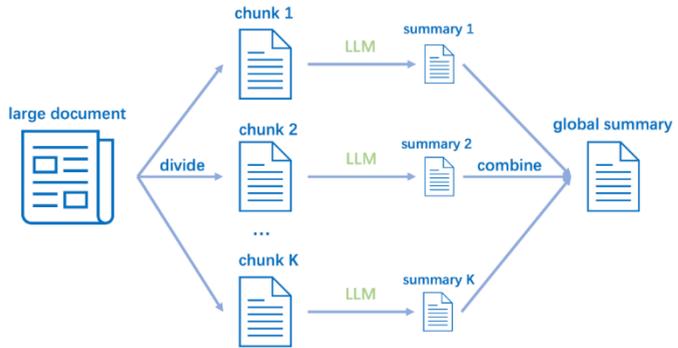
Dalam menghasilkan ringkasan yang terstruktur, GPT-4 bekerja dengan pendekatan berbasis prediksi teks secara berkelanjutan. Model ini menganalisis isi dokumen dengan membagi teks ke dalam unit-unit semantik, mengidentifikasi bagian yang paling relevan, serta menyusun ulang informasi tersebut agar tetap mudah dipahami. Salah satu metode yang digunakan adalah ekstraksi dan abstraksi informasi, di mana GPT-4 tidak hanya menyalin poin utama dari teks asli tetapi juga dapat merangkum dengan menggunakan kata-kata dan struktur yang lebih ringkas. Dengan pemrosesan yang berbasis probabilistik, model ini mampu memahami hubungan antar kalimat serta menjaga kesinambungan antara berbagai bagian ringkasan agar tetap logis dan informatif(OpenAI, 2023).

Selain itu, keunggulan utama GPT-4 dalam merangkum teks adalah fleksibilitasnya dalam menyesuaikan format keluaran sesuai dengan kebutuhan. Model ini dapat diinstruksikan untuk mengikuti struktur tertentu, seperti menggunakan poin-poin, paragraf yang lebih padat, atau format spesifik yang biasa digunakan dalam dokumentasi resmi. Dengan kemampuannya dalam memahami instruksi dalam berbagai bahasa serta menghasilkan ringkasan yang koheren dan mudah dipahami, GPT-4 menjadi salah satu solusi terbaik dalam otomatisasi ringkasan dokumen yang efisien dan akurat.

3.6 Chunking dan Tokenization dalam Pemrosesan Teks

Dalam pemrosesan teks menggunakan model kecerdasan buatan seperti *GPT-4*, terdapat batasan jumlah token yang dapat diproses dalam satu kali permintaan (*request*). Oleh karena itu, diperlukan teknik *tokenization*, yaitu proses mengubah teks menjadi unit-unit lebih kecil seperti kata atau sub-kata agar dapat dipahami oleh model. Setiap model memiliki aturan tokenisasi yang berbeda, di mana jumlah token tidak selalu sama dengan jumlah kata dalam teks. Selain *tokenization*, teknik *chunking* juga digunakan untuk membagi teks panjang menjadi beberapa bagian yang lebih kecil (*chunks*) agar dapat diproses secara efisien (Fadillah et al., 2024). Dengan adanya *chunking*, model dapat tetap bekerja dalam batasan token yang telah ditentukan tanpa mengalami *truncation* atau kehilangan sebagian informasi.

Gambar 3.2 menunjukkan bagaimana teks panjang dapat dipecah menjadi beberapa *chunk* sebelum diproses oleh model AI. Proses ini dimulai dengan membagi dokumen besar menjadi beberapa bagian lebih kecil (*chunk 1*, *chunk 2*, ..., *chunk K*) agar dapat diproses secara terpisah. Setiap *chunk* kemudian diproses secara individual oleh model *large language model (LLM)* untuk menghasilkan ringkasan terpisah (*summary 1*, *summary 2*, ..., *summary K*). Setelah semua *chunk* diringkas, hasilnya digabungkan kembali untuk membentuk ringkasan akhir yang lebih komprehensif (*global summary*). Dengan metode ini, model dapat bekerja lebih optimal tanpa kehilangan konteks yang terlalu banyak dari dokumen asli (Vikrant Singh, 2024).



Gambar 3.2 Proses chunking membagi dokumen panjang, merangkum tiap bagian dengan LLM, lalu menggabungkannya menjadi ringkasan akhir

Teknik *chunking* dan *tokenization* sangat penting dalam berbagai aplikasi *natural language processing (NLP)*, terutama dalam pemrosesan dokumen panjang. Contohnya, dalam sistem ringkasan teks, *chunking* memastikan bahwa model tetap dapat memahami isi dokumen secara keseluruhan meskipun diproses dalam bagian-bagian kecil. Selain itu, metode ini juga sering digunakan dalam analisis sentimen, penerjemahan mesin, dan ekstraksi informasi untuk meningkatkan efisiensi pemrosesan teks (Singh et al., 2024). Dengan membagi teks menjadi bagian yang lebih kecil, model dapat mempertahankan akurasi dan keterbacaan dalam hasil akhirnya. Oleh karena itu, *chunking* dan *tokenization* menjadi strategi yang penting untuk memastikan bahwa model AI dapat bekerja dengan baik dalam berbagai skenario pemrosesan teks yang kompleks.

3.7 LangChain sebagai Kerangka Kerja untuk LLM

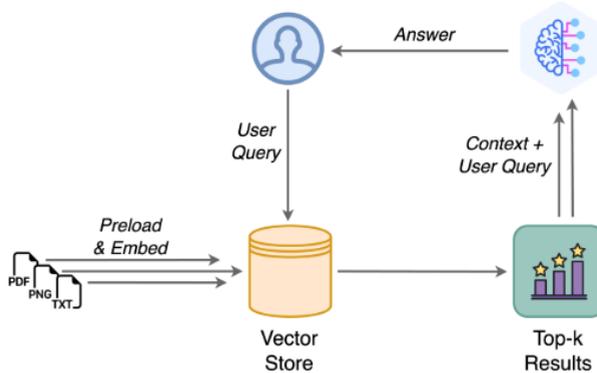
LangChain adalah kerangka kerja yang dirancang untuk mengintegrasikan *Large Language Models (LLM)* ke

dalam berbagai aplikasi dengan lebih efisien dan fleksibel. LangChain memungkinkan pengembang untuk membangun aplikasi berbasis AI yang lebih kompleks dengan memanfaatkan berbagai modul yang tersedia, seperti pemrosesan dokumen, memori percakapan, serta akses ke berbagai sumber data eksternal. Dengan kemampuannya dalam menyusun alur interaksi AI secara modular, LangChain mempermudah pengembangan aplikasi yang membutuhkan pemahaman konteks yang lebih mendalam, seperti *chatbot*, sistem tanya-jawab, dan pembuatan ringkasan otomatis (Topsakal & Akinci, 2023).

Salah satu fitur utama LangChain adalah kemampuannya untuk mengintegrasikan LLM seperti GPT-4 dengan berbagai sumber data, baik yang bersifat real-time maupun historis. Misalnya, dalam sistem yang memproses transkripsi audio ke teks, LangChain dapat mengambil hasil transkripsi, membersihkan data, dan kemudian mengirimkannya ke model AI untuk dibuatkan ringkasan yang lebih terstruktur. Selain itu, LangChain mendukung berbagai penyimpanan memori untuk mempertahankan konteks dalam percakapan multi-turn, sehingga memungkinkan model untuk menghasilkan respons yang lebih relevan berdasarkan interaksi sebelumnya.

Seperti yang ditunjukkan dalam Gambar 3.3, LangChain menggunakan pendekatan *retrieval-augmented generation (RAG)* untuk meningkatkan akurasi hasil yang dihasilkan oleh model AI. Dokumen dalam berbagai format, seperti PDF, teks, dan gambar, dimuat dan dikonversi menjadi *embedding* dalam *vector store*. Ketika pengguna mengajukan pertanyaan, sistem akan mengambil dokumen yang paling relevan berdasarkan kesamaan vektor. Dokumen-dokumen ini kemudian digabungkan dengan *query* pengguna untuk menyediakan konteks tambahan bagi model bahasa, sehingga menghasilkan jawaban yang lebih akurat dan kontekstual. Dengan pendekatan ini, LangChain mampu meningkatkan

kemampuan model AI dalam memberikan respons yang lebih faktual dengan mengandalkan sumber informasi yang telah terindeks dalam *vector store* (Mavroudis, 2024).



Gambar 3.3 *Arsitektur pipeline LangChain yang menerapkan retrieval-augmented generation (RAG)*

Selain dalam pemrosesan dokumen, LangChain juga unggul dalam pengelolaan percakapan. Dengan fitur retrieval-augmented generation (RAG), sistem dapat mengakses informasi tambahan dari sumber eksternal untuk meningkatkan akurasi respons model AI. Ini menjadikan LangChain sangat cocok untuk aplikasi yang membutuhkan konteks dinamis, seperti chatbot layanan pelanggan atau sistem otomatisasi pembuatan Minutes of Meeting (MoM).

Dengan fleksibilitas yang dimilikinya, LangChain menjadi pilihan utama bagi pengembang yang ingin membangun aplikasi berbasis LLM dengan struktur yang lebih modular dan efisien. Kemampuannya dalam mengelola alur interaksi antara pengguna, model AI, dan berbagai sumber data membuatnya semakin populer dalam berbagai bidang, mulai dari analisis teks hingga otomatisasi dokumen profesional. Dengan integrasi yang mudah dengan model

seperti GPT-4, LangChain memungkinkan pengembangan aplikasi berbasis AI yang lebih canggih dan mudah diadaptasi sesuai dengan kebutuhan spesifik pengguna.

3.8 LangSmith sebagai Alat Monitoring dan Evaluasi LLM

Dalam pengembangan aplikasi berbasis *Large Language Models (LLM)*, pemantauan dan evaluasi performa model menjadi aspek penting untuk memastikan hasil yang optimal dan akurat. Salah satu alat yang mendukung proses ini adalah *LangSmith*, sebuah platform yang dirancang untuk membantu pengembang dalam menganalisis, menguji, dan meningkatkan interaksi antara *LLM* dan aplikasi yang dikembangkan. Dengan *LangSmith*, pengembang dapat melacak bagaimana model memproses input, mengukur kualitas output, serta mengoptimalkan penggunaan sumber daya untuk meningkatkan performa sistem (Mavroudis, 2024).

Salah satu fitur utama yang ditawarkan oleh *LangSmith* adalah kemampuan untuk merekam dan meninjau setiap *prompt* yang dikirimkan ke *LLM* beserta respons yang dihasilkan. Hal ini memungkinkan pengembang untuk memahami bagaimana model merespons berbagai jenis input dan mengidentifikasi pola kesalahan yang mungkin terjadi. Selain itu, *LangSmith* juga menyediakan alat untuk melakukan uji coba berbagai versi *prompt* dan parameter model guna menemukan konfigurasi terbaik dalam menghasilkan keluaran yang lebih akurat dan relevan (Elizarov, 2024; Mavroudis, 2024).

Selain sebagai alat pemantauan, *LangSmith* juga mendukung *debugging* dan *fine-tuning* model dalam konteks aplikasi berbasis AI. Misalnya, dalam sistem otomatisasi *Minutes of Meeting (MoM)*, pengembang dapat menggunakan *LangSmith* untuk menguji seberapa baik model mampu

merangkum transkripsi audio menjadi teks yang terstruktur. Jika terdapat ketidaksesuaian dalam hasil ringkasan, pengembang dapat melakukan iterasi perbaikan dengan menyesuaikan teknik *prompt engineering* atau mengintegrasikan metode tambahan seperti *retrieval-augmented generation (RAG)* untuk meningkatkan kualitas keluaran.

Keunggulan lain dari *LangSmith* adalah kemampuannya untuk menyediakan analisis berbasis metrik yang dapat membantu pengembang mengevaluasi kinerja model dalam skala besar. Dengan data statistik yang diberikan, pengembang dapat menentukan apakah sistem perlu diperbaiki, baik dari segi kecepatan pemrosesan, kualitas ringkasan, maupun relevansi jawaban dalam skenario tertentu. Hal ini menjadikan *LangSmith* sebagai alat yang sangat berguna dalam pengembangan aplikasi berbasis *LLM* yang membutuhkan evaluasi berkala untuk memastikan performa optimal.

Dengan semakin kompleksnya implementasi *LLM* dalam berbagai aplikasi, penggunaan *LangSmith* menjadi semakin esensial untuk memastikan sistem yang dikembangkan bekerja secara efektif dan efisien. Integrasi antara *LangSmith* dan *LangChain* memungkinkan pengembang untuk membangun solusi AI yang tidak hanya canggih tetapi juga lebih dapat diandalkan dalam skenario dunia nyata.

3.9 Pengembangan Aplikasi Berbasis FastAPI dan Streamlit

Dalam pengembangan aplikasi berbasis kecerdasan buatan, pemilihan teknologi yang tepat untuk *backend* dan *frontend* sangat penting agar sistem dapat berjalan secara efisien dan responsif. FastAPI adalah salah satu framework backend berbasis Python yang banyak digunakan untuk

membangun layanan web modern, terutama yang berkaitan dengan pemrosesan data dan AI(Howard & Gugger, 2020). Framework ini dirancang dengan pendekatan asinkron yang memungkinkan eksekusi kode berjalan lebih cepat dibandingkan framework tradisional seperti Flask atau Django. FastAPI menggunakan anotasi tipe Python, yang membuatnya lebih mudah dibaca dan ditulis, serta menghasilkan dokumentasi API secara otomatis menggunakan OpenAPI dan Swagger. Dengan performa yang tinggi dan kemudahan dalam pengelolaan permintaan HTTP, FastAPI menjadi pilihan ideal untuk membangun sistem yang menangani pemrosesan teks, gambar, atau data lainnya dengan cepat(Chen, 2023).

Sementara itu, untuk antarmuka pengguna yang interaktif dan mudah digunakan, Streamlit telah menjadi pilihan populer dalam komunitas pengembang AI dan data science. Streamlit adalah framework berbasis Python yang memungkinkan pembuatan dashboard dan aplikasi web dengan sedikit kode. Dibandingkan dengan framework frontend tradisional seperti React atau Vue.js, Streamlit lebih berfokus pada kemudahan penggunaan tanpa perlu banyak konfigurasi tambahan. Dengan hanya beberapa baris kode, pengguna dapat membuat antarmuka berbasis formulir, grafik, dan elemen interaktif lainnya. Selain itu, Streamlit berjalan secara real-time, sehingga perubahan pada kode dapat langsung terlihat tanpa perlu memuat ulang aplikasi, membuatnya sangat efisien dalam pengembangan berbasis eksperimen(Chaurase et al., 2024).

Integrasi antara FastAPI dan Streamlit memungkinkan pembuatan aplikasi berbasis kecerdasan buatan yang bersifat interaktif dan responsif. Dalam arsitektur aplikasi, FastAPI bertindak sebagai backend yang menangani logika bisnis utama, seperti pemrosesan data, pengelolaan model AI, serta pengolahan file yang diunggah oleh pengguna. Di sisi lain, Streamlit berfungsi sebagai frontend yang menghubungkan

pengguna dengan sistem melalui antarmuka berbasis web. Komunikasi antara kedua komponen ini biasanya dilakukan melalui API, di mana FastAPI menyediakan endpoint untuk menerima permintaan dan mengembalikan hasil yang kemudian ditampilkan oleh Streamlit. Dengan pendekatan ini, pengembang dapat memisahkan logika backend dan frontend dengan lebih baik, sehingga aplikasi lebih modular dan mudah dikembangkan lebih lanjut (Reshini et al., 2024).

Salah satu keunggulan utama dari kombinasi FastAPI dan Streamlit adalah kemampuannya untuk menangani permintaan secara asinkron. FastAPI mendukung penggunaan `async` dan `await`, yang memungkinkan sistem untuk menangani banyak permintaan secara bersamaan tanpa harus menunggu setiap tugas selesai terlebih dahulu. Hal ini sangat berguna dalam aplikasi yang membutuhkan pemrosesan data secara intensif, seperti transkripsi audio, analisis teks, atau prediksi berbasis AI. Di sisi lain, Streamlit mempermudah interaksi pengguna dengan menyediakan elemen-elemen yang dapat dikonfigurasi dengan cepat, seperti tombol, slider, atau formulir input, tanpa perlu menulis kode HTML atau JavaScript secara manual.

Selain itu, FastAPI dan Streamlit juga mendukung integrasi dengan berbagai layanan eksternal, seperti penyimpanan cloud, database, dan model AI yang berjalan di server terpisah. FastAPI memungkinkan pengelolaan data yang lebih kompleks dengan dukungan database seperti PostgreSQL dan MongoDB, sementara Streamlit dapat digunakan untuk menampilkan hasil analisis data secara visual dengan grafik interaktif. Dengan fleksibilitas ini, kombinasi kedua framework ini tidak hanya cocok untuk pengembangan aplikasi berbasis AI, tetapi juga untuk berbagai jenis aplikasi data-driven lainnya, seperti dasbor analitik, sistem rekomendasi, dan otomasi proses bisnis berbasis data (Howard & Gugger, 2020).

Dengan performa tinggi, skalabilitas, dan kemudahan penggunaan, FastAPI dan Streamlit menjadi pilihan yang sangat efektif dalam pengembangan aplikasi modern berbasis AI. FastAPI memungkinkan backend yang cepat dan efisien dalam menangani proses berat seperti pemrosesan bahasa alami dan model AI, sementara Streamlit menyediakan antarmuka yang user-friendly bagi pengguna non-teknis. Kombinasi keduanya memungkinkan pengembang untuk membangun aplikasi yang tidak hanya fungsional, tetapi juga mudah digunakan dan dapat diakses oleh berbagai kalangan pengguna.

3.10 Pembuatan dan Ekspor Dokumen PDF

Dokumen *Portable Document Format (PDF)* adalah salah satu format file yang paling banyak digunakan untuk menyimpan dan berbagi dokumen dalam bentuk digital. Keunggulan utama format ini adalah kemampuannya untuk mempertahankan tata letak dan format dokumen di berbagai perangkat serta sistem operasi. PDF juga mendukung berbagai elemen seperti teks, gambar, tabel, dan metadata yang memudahkan pencarian serta pengelolaan dokumen (Khadim et al., 2022). Oleh karena itu, banyak sistem otomatisasi dokumen yang mengandalkan PDF untuk memastikan informasi yang disimpan memiliki tampilan profesional dan mudah dibaca.

Dalam pengembangan aplikasi berbasis *Python*, terdapat beberapa pustaka yang digunakan untuk pembuatan dan manipulasi dokumen PDF secara otomatis. Salah satu pustaka yang umum digunakan adalah FPDF, yang memungkinkan pembuatan dokumen dari awal dengan berbagai fitur seperti pengaturan margin, jenis font, warna teks, dan penambahan elemen grafis. Selain itu, ReportLab sering digunakan dalam pembuatan laporan yang lebih kompleks karena mendukung pembuatan diagram dan elemen

visual lainnya. Pustaka lain seperti *pdffit*, yang berbasis *wkhtmltopdf*, juga populer karena memungkinkan konversi halaman HTML menjadi PDF, sehingga mempermudah penggunaan template yang lebih fleksibel (Moiz, 2024).

Selain pembuatan dokumen, aspek kustomisasi dalam ekspor PDF juga menjadi pertimbangan penting. Banyak aplikasi memungkinkan pengguna untuk memilih font, warna teks, serta struktur halaman agar hasil akhir lebih sesuai dengan kebutuhan spesifik. Dalam konteks otomatisasi, fitur ini berguna untuk menghasilkan dokumen dengan tampilan yang lebih profesional dan sesuai standar industri. Selain itu, penggunaan teknologi *Optical Character Recognition (OCR)* sering diterapkan dalam sistem PDF berbasis gambar untuk mengonversi teks yang tidak dapat dicari menjadi teks yang dapat dikenali secara digital (Raj & Kos, 2022).

Dengan semakin berkembangnya kebutuhan akan dokumentasi digital, integrasi sistem pembuatan dan ekspor PDF dalam berbagai aplikasi semakin meningkat. Dengan bantuan pustaka *Python*, pengembang dapat mengotomatiskan proses pembuatan laporan, transkrip, atau ringkasan dokumen dalam format PDF yang terstruktur. Fleksibilitas dalam ekspor PDF ini memungkinkan aplikasi untuk memenuhi berbagai kebutuhan pengguna, mulai dari pelaporan bisnis hingga pencatatan akademik dan profesional.

[Halaman ini sengaja dikosongkan]

4.1. Analisis Sistem

Pada bab ini akan dijelaskan mengenai tahapan dalam mengembangkan aplikasi *MoMify*, yaitu sebuah sistem yang secara otomatis mengonversi rekaman audio dan video menjadi *Minutes of Meeting* (MoM) menggunakan teknologi *Speech-to-Text*, *Large Language Models* (LLM), serta fitur kustomisasi dan ekspor ke format PDF. Penjelasan mencakup deskripsi umum aplikasi serta analisis kebutuhan untuk memastikan bahwa sistem dapat berjalan sesuai dengan tujuan yang telah ditetapkan.

4.1.1 Definisi Umum Aplikasi

Secara umum, *MoMify* adalah aplikasi berbasis kecerdasan buatan yang dirancang untuk mempermudah pencatatan dan dokumentasi hasil rapat secara otomatis. Aplikasi ini memungkinkan pengguna untuk mengunggah rekaman audio atau video dari suatu rapat, yang kemudian akan diproses menjadi transkripsi teks menggunakan model *Speech-to-Text* (*Whisper AI*). Selanjutnya, sistem akan menggunakan model LLM (*GPT-4*) untuk menganalisis transkripsi dan menyusunnya menjadi ringkasan *Minutes of Meeting* yang terstruktur, lengkap dengan informasi seperti agenda, poin-poin penting diskusi, serta langkah tindak lanjut.

Selain itu, *MoMify* juga menyediakan fitur kustomisasi yang memungkinkan pengguna untuk memilih format keluaran MoM sesuai dengan preferensi mereka, termasuk pemilihan font, warna teks, serta bahasa yang digunakan dalam ringkasan. Hasil akhir dapat diekspor dalam format PDF yang dapat dibagikan atau digunakan sebagai dokumentasi resmi dalam suatu organisasi. Dengan fitur-fitur ini, aplikasi bertujuan untuk meningkatkan efisiensi dalam pencatatan rapat,

mengurangi kesalahan pencatatan manual, serta mempermudah akses terhadap informasi yang terdokumentasi.

4.1.2 Penamaan dan Desain Logo Aplikasi

Nama *MoMify* dipilih sebagai representasi dari fungsi utama aplikasi, yaitu mengubah rekaman rapat menjadi dokumen *Minutes of Meeting* (MoM) secara otomatis. Nama ini berasal dari singkatan MoM (*Minutes of Meeting*) dan sufiks *-ify* yang dalam bahasa Inggris berarti “mengubah menjadi” atau “membentuk”, sebagaimana pada kata *simplify*, *clarify*, dan *beautify*. Dengan menggabungkan kedua elemen ini, nama *MoMify* secara langsung merepresentasikan proses otomatisasi dalam pencatatan hasil rapat. Selain itu, nama ini terdengar modern, singkat, dan mudah diingat, sehingga sesuai untuk digunakan sebagai nama produk teknologi berbasis kecerdasan buatan. Pemilihan nama ini juga mempertimbangkan aspek *branding* agar tetap profesional dan memiliki daya tarik tersendiri bagi pengguna.



Gambar 4.1 Logo aplikasi MoMify

Logo *MoMify* dirancang dengan bentuk segitiga *rounded* mengarah ke kanan, menyerupai ikon tombol “*play*” untuk menggambarkan bahwa aplikasi menerima *input* berupa audio atau video. Di dalam logo seperti

ditunjukkan pada Gambar 4.1 terdapat elemen garis dan bulatan yang menyerupai visualisasi gelombang suara, sekaligus mencerminkan struktur data atau proses transformasi menjadi dokumen. Kombinasi elemen ini menyimbolkan alur utama aplikasi, yaitu dari rekaman suara menjadi ringkasan terstruktur. Warna utama yang digunakan adalah coral dengan kode heksadesimal #fe9481, memberikan kesan hangat, ramah, dan profesional. Sementara itu, elemen garis dan bulatan dalam logo diberi warna putih #ffffff untuk menciptakan kontras visual yang bersih dan mudah dikenali. Secara keseluruhan, desain logo ini tidak hanya berfungsi sebagai identitas visual, tetapi juga menggambarkan fungsi utama dan karakteristik aplikasi *MoMify*.

4.1.3 Analisis Kebutuhan Fungsional

Aplikasi *MoMify* dikembangkan dengan beberapa kebutuhan fungsional utama untuk memastikan sistem dapat berjalan dengan optimal. Berikut adalah kebutuhan fungsional yang telah diidentifikasi:

1. Unggah File Audio/Video

Pengguna dapat mengunggah rekaman rapat dalam berbagai format, seperti MP3, WAV, MP4, dan AVI, dengan batas ukuran tertentu.

2. Proses Transkripsi Otomatis

Sistem akan menggunakan *Whisper AI* untuk mengubah rekaman audio atau video menjadi teks yang akurat dan mendukung berbagai bahasa.

3. Analisis dan Ringkasan dengan LLM

Setelah transkripsi selesai, teks akan diproses menggunakan *GPT-4* untuk menghasilkan ringkasan *Minutes of Meeting* yang lebih terstruktur.

4. **Kustomisasi Format MoM**

Pengguna dapat menyesuaikan tampilan dokumen MoM dengan memilih font, warna teks, serta bahasa ringkasan yang diinginkan.

5. **Ekspor ke Format PDF**

Hasil akhir MoM dapat diekspor ke dalam format PDF yang telah diformat sesuai preferensi pengguna, sehingga siap untuk dibagikan atau disimpan.

6. **Tampilan Antarmuka yang Interaktif**

Sistem berbasis *Streamlit* akan menyediakan UI yang intuitif, memungkinkan pengguna untuk berinteraksi dengan fitur aplikasi secara mudah dan efisien.

7. **Notifikasi Proses**

Sistem menampilkan *progress bar* atau indikator proses selama transkripsi dan pembuatan MoM berlangsung, agar pengguna mengetahui tahapan yang sedang berjalan.

Dengan memenuhi kebutuhan fungsional ini, aplikasi diharapkan dapat memberikan solusi yang praktis dan efisien dalam pembuatan *Minutes of Meeting*, serta membantu organisasi atau individu dalam mengelola dokumentasi rapat dengan lebih baik.

4.1.4 **Analisis Kebutuhan Non-Fungsional**

Sebagai bagian dari kebutuhan non-fungsional, aplikasi *MoMify* dirancang untuk memastikan performa, skalabilitas, keamanan, dan pengalaman pengguna yang optimal. Salah satu aspek utama dari kebutuhan non-fungsional adalah desain antarmuka pengguna (*User*

Interface/UI), yang dirancang agar intuitif dan mudah digunakan. Antarmuka *MoMify* berbasis *Streamlit* menyediakan tampilan sederhana namun interaktif, yang memungkinkan pengguna untuk mengunggah file, melihat status pemrosesan, dan mengunduh hasil MoM dengan mudah.

Dari sisi performa, aplikasi harus mampu menangani berbagai ukuran file audio dan video tanpa mengalami kendala yang signifikan. Oleh karena itu, sistem memanfaatkan *Whisper AI* untuk proses transkripsi yang cepat dan akurat, serta *GPT-4* untuk menghasilkan ringkasan dalam waktu yang efisien. Selain itu, backend aplikasi dibangun menggunakan *FastAPI*, yang dikenal memiliki kecepatan tinggi dalam menangani permintaan pengguna, sehingga memastikan respons yang cepat dalam setiap tahap pemrosesan.

Keamanan data juga menjadi salah satu pertimbangan penting dalam pengembangan aplikasi ini. *MoMify* dirancang agar tidak menyimpan file yang diunggah oleh pengguna setelah proses selesai, sehingga menjaga privasi informasi yang diproses. Selain itu, konfigurasi API yang digunakan dalam model LLM dan *Whisper AI* dikelola dengan variabel lingkungan untuk memastikan bahwa kredensial tetap aman dan tidak terekspos dalam kode sumber.

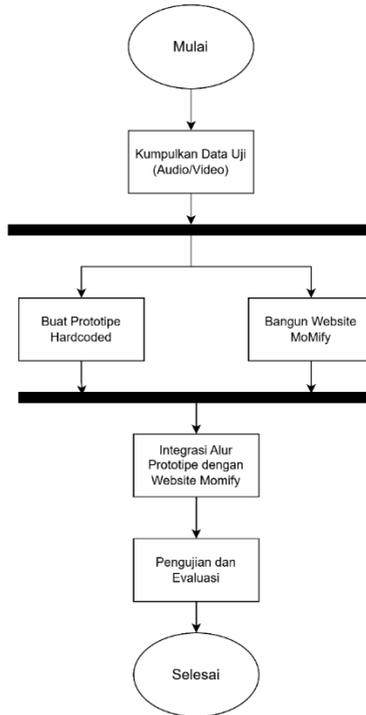
Selain faktor performa dan keamanan, skalabilitas juga menjadi aspek krusial dalam kebutuhan non-fungsional aplikasi ini. Sistem harus dapat menangani banyak pengguna secara bersamaan tanpa mengalami penurunan kinerja yang signifikan. Dengan menggunakan *FastAPI* dan integrasi layanan berbasis cloud, aplikasi dapat diperluas untuk menangani volume data yang lebih besar sesuai dengan kebutuhan pengguna.

Dengan mempertimbangkan aspek UI/UX, performa, keamanan, dan skalabilitas, aplikasi *MoMify* dirancang

untuk memberikan pengalaman pengguna yang optimal dalam pembuatan *Minutes of Meeting* secara otomatis. Hal ini memastikan bahwa aplikasi tidak hanya berfungsi dengan baik, tetapi juga dapat diandalkan dalam lingkungan profesional yang membutuhkan pencatatan rapat yang cepat dan akurat.

4.2. Perancangan Sistem

Pada tahap ini, dilakukan perancangan sistem untuk memastikan bahwa aplikasi *MoMify* dapat berjalan secara optimal dan memenuhi kebutuhan pengguna. Perancangan sistem ini mencakup alur pemrosesan dari awal hingga akhir, mulai dari unggahan file hingga pembuatan *Minutes of Meeting (MoM)* dalam bentuk PDF yang siap diunduh. Diagram alir yang menggambarkan keseluruhan proses dapat dilihat pada Gambar 4.2.



Gambar 4.2 Diagram alir perancangan sistem secara keseluruhan

Proses dimulai dengan pengumpulan data uji berupa berbagai format audio dan video yang digunakan untuk menguji akurasi transkripsi dan kompatibilitas sistem. Setelah itu, dibuat prototipe awal dengan implementasi *hardcoded* untuk memastikan bahwa model transkripsi dan ringkasan teks bekerja sebagaimana diharapkan sebelum dikembangkan lebih lanjut. Selanjutnya, dilakukan pembangunan website MoMify yang berfungsi sebagai antarmuka pengguna untuk mengunggah file dan mengatur konfigurasi. Setelah website selesai, tahap berikutnya adalah integrasi antara prototipe dengan sistem berbasis web agar seluruh alur kerja dari transkripsi hingga

pembuatan MoM dapat berjalan otomatis. Akhirnya, dilakukan pengujian dan evaluasi untuk memastikan bahwa sistem berfungsi dengan baik, mulai dari keakuratan transkripsi hingga pengalaman pengguna dalam menggunakan aplikasi. Jika ditemukan kendala atau kekurangan, sistem akan diperbaiki dan dioptimalkan sebelum digunakan secara penuh.

4.1.5 Pengumpulan Data Uji (Audio/Video)

Pada tahap ini, dilakukan proses pengumpulan data uji berupa file audio dan video yang akan digunakan untuk menguji performa sistem transkripsi dan ringkasan otomatis MoMify. Data uji dikumpulkan dalam berbagai format file, seperti MP3, WAV, MP4, M4A, WMA, AVI, dan MKV, guna memastikan sistem dapat menangani berbagai jenis input yang umum digunakan dalam dokumentasi rapat dan percakapan profesional.

Setiap jenis file memiliki jumlah sampel yang berbeda, dengan variasi durasi yang mencerminkan skenario nyata dalam penggunaan MoMify. Beberapa file berdurasi pendek sekitar 10-30 menit untuk menguji respons sistem terhadap percakapan singkat, sementara file dengan durasi lebih panjang (hingga 2 jam) digunakan untuk menguji efisiensi pemrosesan pada pertemuan yang lebih kompleks. Data ini diambil dari berbagai sumber, termasuk rekaman rapat internal, video publik, podcast, webinar, serta film dokumenter yang memiliki elemen percakapan.

Selain itu, kualitas audio/video juga bervariasi untuk menguji bagaimana MoMify menangani perbedaan bitrate dan resolusi dalam transkripsi dan ringkasan. Beberapa file audio memiliki kualitas standar 128 kbps hingga 320 kbps, sementara file video diuji dengan resolusi mulai dari 720p 30fps hingga 1080p 60fps. Dengan dataset yang beragam ini, diharapkan sistem dapat dikembangkan dan

diuji secara optimal dalam berbagai kondisi nyata yang akan dihadapi pengguna. Detail lebih lanjut mengenai format file yang digunakan serta jumlah sampel untuk setiap jenis data uji dirangkum dalam Tabel 4.1. Secara keseluruhan, total jumlah sampel yang digunakan dalam pengujian sistem adalah 30 file, mencakup berbagai jenis format yang telah disebutkan sebelumnya.

Tabel 4.1 Daftar format file audio dan video beserta jumlah sampel dan durasi yang digunakan sebagai data uji dalam pengembangan dan pengujian sistem MoMify

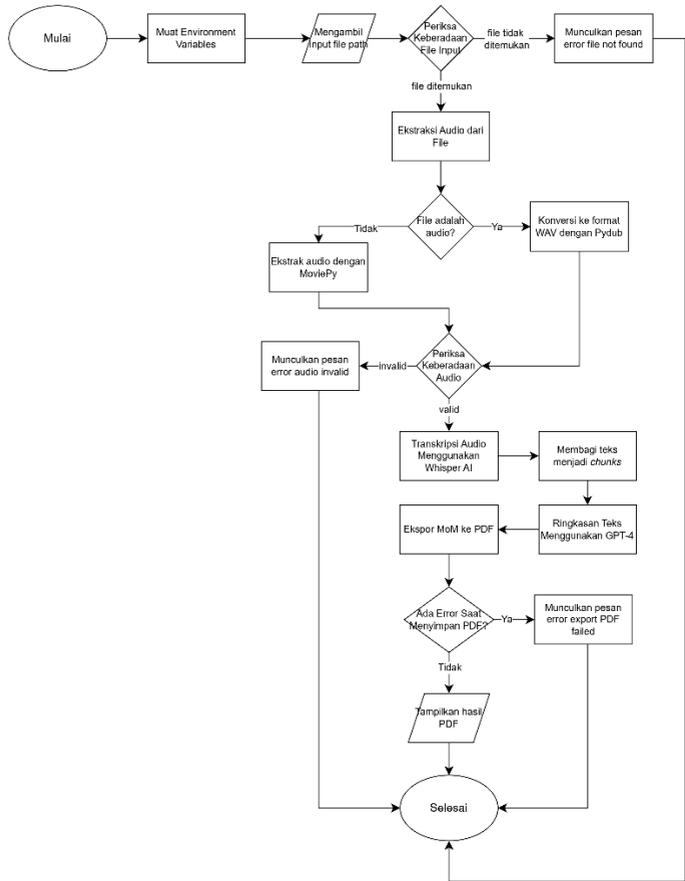
No.	Jenis File	Jumlah Sampel	Durasi (Menit)	Kualitas Audio/Video
1	mp3	5	15, 30, 45, 60, 120	128 kbps
2	wav	4	10, 25, 40, 90	320 kbps
3	mp4	5	20, 35, 50, 75, 120	1080p, 30 fps
4	m4a	5	12, 28, 55, 85, 100	256 kbps
5	wma	3	18, 33, 70	192 kbps
6	avi	4	22, 48, 95, 105	720p, 60 fps
7	mkv	4	14, 31, 67, 80	1080p, 60 fps

4.1.6 Pembuatan Prototipe Hardcoded

Sebelum membangun sistem MoMify secara penuh, dilakukan tahap pembuatan prototipe berbasis *hardcoded* untuk menguji alur utama dari proses konversi audio/video menjadi *Minutes of Meeting* (MoM). Prototipe ini bertujuan untuk memahami kebutuhan sistem secara lebih

mendalam serta mengidentifikasi potensi kendala sebelum dilakukan integrasi dengan antarmuka pengguna berbasis *web*. Pada tahap ini, implementasi dilakukan secara langsung melalui skrip Python tanpa antarmuka visual, sehingga fokus utama adalah memastikan alur pemrosesan berjalan dengan baik dari segi transkripsi, ringkasan, hingga pembuatan file PDF.

Tahapan utama dalam pembuatan prototipe ini mencakup pemrosesan file audio/video menggunakan model *Whisper AI* untuk transkripsi otomatis, diikuti dengan pemanfaatan *Large Language Model* (LLM) untuk merangkum teks yang dihasilkan. Hasil ringkasan kemudian diformat dalam struktur MoM yang telah ditentukan dan diekspor dalam bentuk PDF. Seluruh proses ini dilakukan dalam lingkungan pengujian lokal untuk memastikan bahwa setiap komponen sistem berfungsi sesuai dengan ekspektasi. Diagram alir yang menggambarkan proses ini dapat dilihat pada Gambar 4.3, di mana alur dimulai dari memuat variabel lingkungan dan membaca file input, dilanjutkan dengan ekstraksi audio, transkripsi menggunakan *Whisper AI*, peringkasan dengan *GPT-4*, hingga akhirnya menghasilkan file MoM dalam format PDF. Selain itu, diagram juga menunjukkan skenario penanganan error, seperti jika file tidak ditemukan atau terjadi kegagalan dalam menyimpan PDF.



Gambar 4.3 Diagram alir proses prototipe hardcoded MoMify, mulai dari ekstraksi audio hingga pembuatan Minutes of Meeting (MoM) dalam format PDF

Keunggulan dari pendekatan *hardcoded* ini adalah memungkinkan pengujian dan iterasi cepat tanpa perlu mengembangkan antarmuka pengguna sejak awal. Dengan demikian, pengembang dapat melakukan penyempurnaan terhadap kualitas transkripsi dan ringkasan sebelum sistem

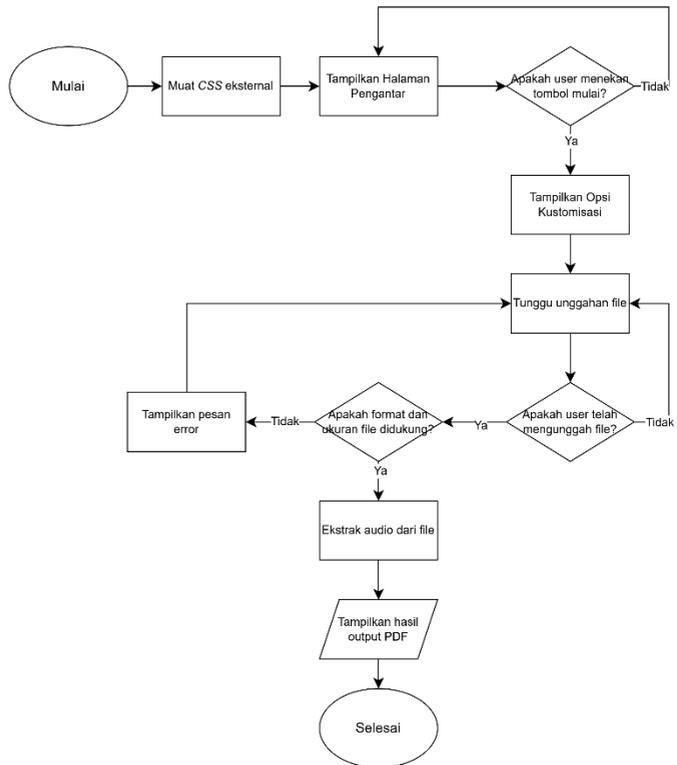
diintegrasikan ke dalam platform berbasis *web*. Setelah prototipe ini berhasil diimplementasikan dan dievaluasi, langkah selanjutnya adalah membangun antarmuka pengguna serta mengintegrasikan seluruh proses ke dalam sistem MoMify secara menyeluruh.

4.1.7 Pengembangan Website MoMify

Setelah prototipe hardcoded berhasil, langkah berikutnya adalah membangun website MoMify sebagai antarmuka pengguna berbasis *Streamlit*. Website ini memungkinkan pengguna untuk mengunggah file audio/video, mengonfigurasi pengaturan (seperti bahasa, font, dan warna), serta melihat hasil transkripsi dan ringkasan *Minutes of Meeting* (MoM) secara *real-time* sebelum mengunduhnya dalam format PDF. Dengan adanya antarmuka ini, pengguna dapat berinteraksi dengan sistem secara lebih mudah tanpa harus menjalankan skrip secara manual.

Pengembangan website ini dimulai dengan perancangan struktur halaman dan komponen UI. Pada tahap awal, halaman utama menampilkan pengantar mengenai fungsi aplikasi dan tombol untuk memulai proses. Setelah itu, pengguna dapat menyesuaikan pengaturan format MoM, termasuk pemilihan font, warna teks, serta bahasa yang digunakan dalam ringkasan. Setelah mengunggah file audio/video, sistem akan memvalidasi format dan ukuran file sebelum melanjutkan proses transkripsi dan pembuatan ringkasan. Hasil transkripsi dan ringkasan yang telah dihasilkan kemudian ditampilkan kepada pengguna dalam tampilan yang intuitif sebelum dapat diekspor menjadi PDF.

Alur utama dalam pengembangan antarmuka website MoMify dapat dilihat pada Gambar 4.4. Diagram ini menunjukkan bagaimana aplikasi *Streamlit* mengelola interaksi pengguna, dimulai dari memuat konfigurasi halaman dan gaya antarmuka menggunakan CSS eksternal, lalu menampilkan pengantar aplikasi. Pengguna kemudian diberikan opsi untuk mengunggah file dan melakukan konfigurasi format MoM. Jika pengguna tidak menekan tombol *Start*, sistem akan tetap berada di halaman utama. Setelah file diunggah, sistem akan melakukan validasi format dan ukuran file. Jika file tidak sesuai, sistem akan menampilkan pesan error dan meminta pengguna mengunggah ulang file yang sesuai. Jika validasi berhasil, sistem akan memproses file, menampilkan hasil ringkasan, dan menyediakan opsi untuk mengunduh dokumen MoM dalam format PDF.



Gambar 4.4 Diagram alir proses antarmuka pengguna MoMify yang mencakup unggahan file, kustomisasi, validasi, dan pembuatan output PDF

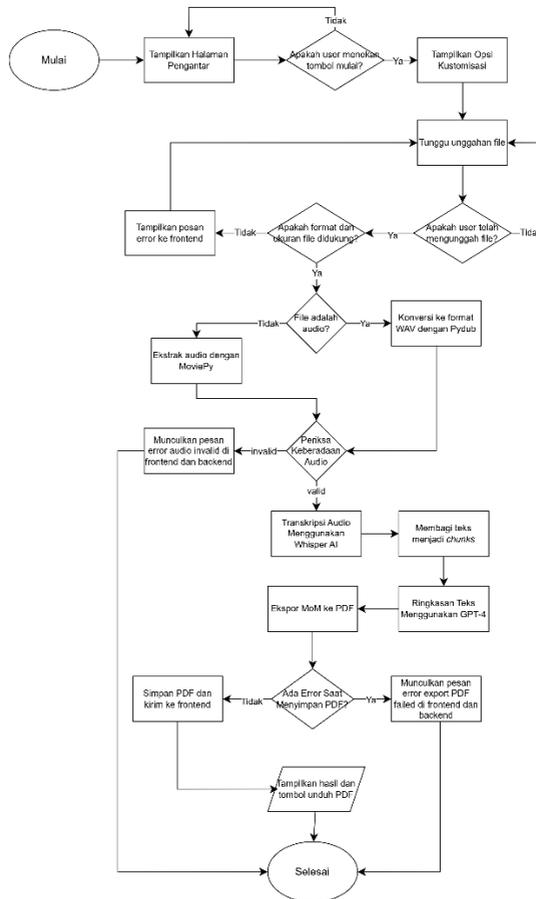
Pendekatan berbasis *Streamlit* ini memberikan fleksibilitas dalam pengembangan antarmuka pengguna karena memungkinkan pengembang untuk dengan cepat mengintegrasikan berbagai komponen UI dengan kode backend yang telah ada. Dengan adanya integrasi ini, MoMify dapat digunakan secara lebih luas oleh pengguna tanpa perlu memiliki keahlian teknis dalam menjalankan skrip Python secara langsung. Setelah antarmuka

pengguna selesai dikembangkan, tahap selanjutnya adalah melakukan integrasi penuh dengan sistem utama untuk memastikan kelancaran proses dari unggahan file hingga pembuatan dokumen MoM dalam format PDF.

4.1.8 Integrasi Prototipe dengan Website MoMify

Pada tahap ini, sistem transkripsi dan ringkasan yang sebelumnya diuji dalam bentuk prototipe *hardcoded* diintegrasikan ke dalam website MoMify untuk memungkinkan interaksi pengguna secara lebih intuitif. Integrasi ini mencakup penghubungan antara *frontend* berbasis Streamlit dan *backend* menggunakan FastAPI. FastAPI bertanggung jawab untuk menangani unggahan file, memproses transkripsi menggunakan model Whisper AI, menghasilkan ringkasan dengan GPT-4, serta mengekspor hasil akhir dalam format PDF. Sementara itu, *frontend* memberikan antarmuka yang memungkinkan pengguna mengunggah file, menyesuaikan pengaturan format MoM, serta melihat hasil transkripsi dan ringkasan secara real-time sebelum mengunduhnya.

Alur kerja sistem yang telah diintegrasikan dapat dilihat pada Gambar 4.5. Diagram ini menggambarkan bagaimana interaksi antara pengguna, *frontend*, dan *backend* berlangsung dari awal hingga akhir proses. Pengguna pertama-tama akan melihat halaman pengantar sebelum menekan tombol mulai. Setelah itu, sistem menampilkan opsi kustomisasi, seperti pemilihan font, warna teks, dan bahasa. Pengguna kemudian mengunggah file audio atau video, yang akan diperiksa format dan ukurannya oleh *backend*. Jika format tidak didukung, pesan kesalahan dikirimkan kembali ke *frontend*. Jika file valid, sistem akan mengekstrak audio (jika berupa video), kemudian melakukan transkripsi dengan Whisper AI.



Gambar 4.5 Diagram alir integrasi frontend (Streamlit) dan backend (FastAPI) dalam proses unggah, transkripsi, ringkasan, dan ekspor MoM ke PDF.

Setelah transkripsi selesai, hasilnya dikirim ke GPT-4 untuk dirangkum menjadi *Minutes of Meeting* (MoM) yang terstruktur. Proses ini dilakukan dalam beberapa tahap, termasuk pembagian teks menjadi *chunks*

untuk memastikan batas token tidak terlampaui. Setelah ringkasan selesai, sistem membuat dokumen PDF dengan format yang telah dipilih oleh pengguna. Jika terjadi kesalahan dalam penyimpanan PDF, pesan error akan ditampilkan di *frontend* dan *backend*. Jika tidak ada kendala, PDF yang berhasil dibuat akan dikirimkan ke *frontend*, di mana pengguna dapat mengunduhnya melalui tombol unduh yang muncul di halaman hasil.

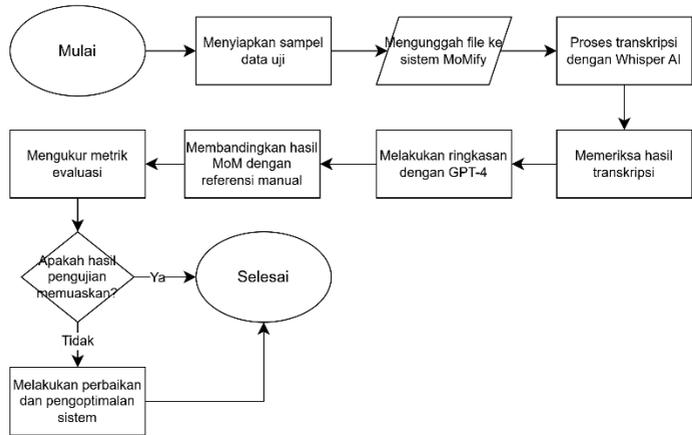
Melalui integrasi ini, seluruh proses unggahan, pemrosesan, dan pengunduhan dilakukan secara otomatis, memungkinkan pengguna untuk menggunakan MoMify tanpa perlu menjalankan skrip secara manual. Dengan optimasi pada alur kerja dan komunikasi antara frontend dan backend, sistem ini dirancang agar responsif dan efisien, memastikan pengalaman pengguna yang lebih nyaman.

4.1.9 Pengujian dan Evaluasi

Setelah proses pengembangan sistem MoMify selesai, tahap pengujian dan evaluasi dilakukan untuk memastikan bahwa sistem berfungsi sesuai dengan yang diharapkan. Pengujian ini mencakup beberapa aspek utama, yaitu keakuratan transkripsi menggunakan *Whisper AI*, efektivitas ringkasan teks yang dihasilkan oleh *GPT-4*, serta kualitas keseluruhan *Minutes of Meeting (MoM)* yang dihasilkan. Selain itu, evaluasi dilakukan dengan membandingkan hasil MoM yang dihasilkan sistem dengan referensi manual untuk mengukur tingkat kesesuaian dan ketepatan informasi.

Gambar 4.6 menunjukkan diagram alir proses pengujian dan evaluasi sistem MoMify. Proses dimulai dengan menyiapkan sampel data uji yang terdiri dari berbagai format file audio dan video dengan durasi yang beragam. Selanjutnya, file diunggah ke dalam sistem MoMify untuk diproses melalui tahapan transkripsi

menggunakan *Whisper AI*. Hasil transkripsi kemudian diperiksa untuk memastikan bahwa teks yang dihasilkan sesuai dengan isi rekaman asli. Setelah itu, teks transkripsi diringkas menggunakan *GPT-4* untuk menghasilkan MoM yang terstruktur dan informatif.



Gambar 4.6 Diagram alir proses pengujian dan evaluasi sistem MoMify, mulai dari pengujian transkripsi dan ringkasan hingga perbandingan hasil dengan referensi manual serta optimasi jika diperlukan

Untuk mengukur performa sistem secara objektif, digunakan beberapa metrik evaluasi yang dirangkum dalam Tabel 4.2. Metrik ini mencakup Word Error Rate (WER) untuk menilai tingkat kesalahan transkripsi, Brevity Penalty untuk memastikan ringkasan tidak terlalu pendek, serta Readability Score yang mengukur keterbacaan teks yang dihasilkan. Selain itu, aspek Processing Time digunakan untuk mengevaluasi kecepatan sistem dalam menghasilkan output, sementara User Satisfaction mengukur kepuasan pengguna

berdasarkan pengalaman mereka dalam menggunakan sistem.

Tabel 4.2 Metrik evaluasi sistem MoMify beserta deskripsi dan skala penilaian

No.	Metrik Evaluasi	Deskripsi	Skala
1	Word Error Rate (WER)	Mengukur persentase kesalahan dalam transkripsi dibandingkan dengan teks referensi. Semakin rendah nilai WER, semakin akurat transkripsi yang dihasilkan.	0-100% (semakin rendah semakin baik)
2	Brevity Penalty	Digunakan dalam evaluasi ringkasan untuk memastikan bahwa ringkasan tidak terlalu pendek dibandingkan teks aslinya.	0-1 (semakin mendekati 1 semakin ideal)
3	Readability Score	Menilai tingkat keterbacaan teks yang dihasilkan berdasarkan panjang kalimat, kompleksitas kata, dan struktur bahasa.	0-100 (semakin tinggi semakin mudah dibaca)

4	Processing Time	Mengukur waktu yang dibutuhkan oleh sistem untuk menyelesaikan proses transkripsi dan ringkasan dari input hingga output PDF.	Dalam detik/menit (Semakin rendah semakin cepat)
5	User Satisfaction	Evaluasi subjektif dari pengguna terkait kemudahan penggunaan, kejelasan ringkasan, dan kepuasan terhadap hasil yang dihasilkan sistem.	1 - 5 (Skala Likert)

Jika hasil pengujian dianggap belum memuaskan, sistem akan diperbaiki dan dioptimalkan sesuai dengan temuan dari evaluasi yang dilakukan. Perbaikan ini dapat mencakup penyesuaian parameter model transkripsi dan ringkasan, peningkatan kualitas pemrosesan teks, atau pengoptimalan antarmuka pengguna agar lebih responsif dan intuitif. Setelah proses perbaikan selesai, pengujian ulang dilakukan hingga sistem mencapai hasil yang diharapkan. Dengan demikian, tahap ini memastikan bahwa MoMify dapat digunakan secara optimal untuk menghasilkan MoM yang akurat, jelas, dan sesuai dengan kebutuhan pengguna.

BAB 5 IMPLEMENTASI SISTEM

Bab ini membahas implementasi sistem yang telah dirancang untuk mendukung pengembangan aplikasi MoMify. Implementasi ini mencakup beberapa bagian utama, yaitu integrasi sistem transkripsi dan ringkasan berbasis *Large Language Model (LLM)*, pengembangan antarmuka pengguna berbasis Streamlit, serta integrasi backend menggunakan FastAPI untuk memastikan aplikasi dapat berfungsi sesuai dengan kebutuhan yang telah ditentukan. Selain itu, dilakukan optimasi alur pemrosesan agar sistem dapat menangani berbagai format file dengan efisien dan menghasilkan *Minutes of Meeting (MoM)* yang akurat serta mudah dibaca.

5.1 Implementasi Pengembangan Aplikasi

Bagian ini menjelaskan implementasi pengembangan aplikasi berbasis web untuk mendukung proses pembuatan *Minutes of Meeting (MoM)* secara otomatis. Implementasi mencakup pembuatan antarmuka pengguna menggunakan *Streamlit*, yang memungkinkan pengguna mengunggah file audio atau video, mengatur preferensi tampilan, serta melihat hasil transkripsi dan ringkasan sebelum mengunduhnya dalam format PDF. Selain itu, sistem ini dirancang dengan navigasi yang intuitif agar pengguna dapat dengan mudah mengakses setiap fitur yang tersedia. Integrasi antara frontend dan backend dilakukan menggunakan *FastAPI* untuk memastikan komunikasi data berjalan secara efisien.

5.1.1 Struktur Halaman Aplikasi

Struktur halaman aplikasi dirancang untuk memberikan pengalaman pengguna yang intuitif dalam mengakses fitur utama sistem. Halaman utama aplikasi menampilkan pengantar mengenai fungsi aplikasi serta menyediakan *expander* yang berisi penjelasan tambahan

mengenai *Minutes of Meeting (MoM)* dan cara kerja sistem. Setelah pengguna memahami informasi tersebut, mereka dapat menekan tombol "*Let's Get Started*" untuk melanjutkan ke tahap berikutnya, yaitu halaman kustomisasi dan unggahan file. Pada tahap ini, pengguna diberikan opsi untuk menyesuaikan font, warna teks, dan bahasa yang digunakan dalam ringkasan sebelum mengunggah file audio atau video yang akan diproses. Setelah file diunggah, sistem akan menampilkan indikator proses hingga hasil transkripsi dan ringkasan siap untuk diunduh dalam format PDF.

Pseudocode yang menggambarkan struktur halaman aplikasi ditampilkan pada Algoritma 5.1. Algoritma ini menjelaskan bagaimana halaman utama ditampilkan, bagaimana interaksi pengguna dikendalikan menggunakan *session state*, serta bagaimana transisi dari satu tahap ke tahap berikutnya terjadi dalam antarmuka pengguna. Pseudocode ini juga mencakup pemrosesan file yang hanya dijalankan setelah pengguna memilih pengaturan dan mengunggah file yang valid. Dengan pendekatan ini, aplikasi dapat memastikan bahwa setiap tahap berjalan secara berurutan dan responsif terhadap input pengguna.

Algoritma 5.1 Struktur Halaman Aplikasi

Function:	Load_UI()
Input:	<ul style="list-style-type: none"> • <i>User click events</i> pada tombol "Let's Get Started" dan tombol unggah file • <i>User selections</i> pada opsi kustomisasi (font, warna, bahasa) • <i>Uploaded file</i> dalam format audio/video
Output:	<ul style="list-style-type: none"> • Tampilan halaman utama aplikasi dengan informasi dan expander • Tampilan halaman kustomisasi jika tombol "Let's Get Started" ditekan • Tampilan halaman unggahan file dan indikator proses

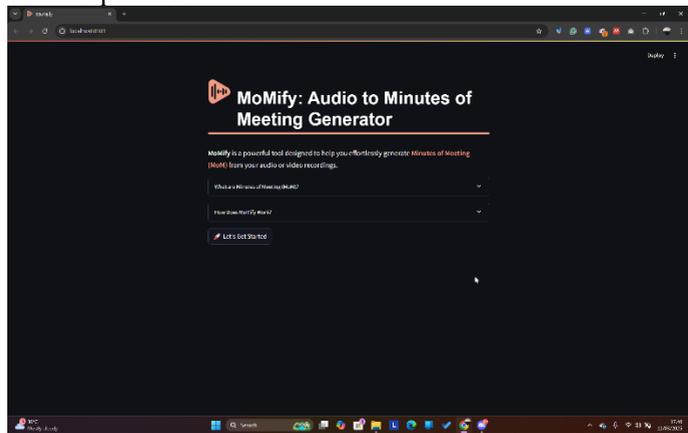
-
- Hasil MoM dalam format PDF yang dapat diunduh
-

1:	Function Load_UI()
2:	Load external CSS
3:	Set page configuration with title "MoMify" and centered layout
4:	Show introduction page with description and expander sections
5:	
6:	If "show_options" state is not initialized:
7:	Initialize "show_options" as False
8:	
9:	If user clicks "Let's Get Started" button:
10:	Set "show_options" to True
11:	
12:	If "show_options" is True:
13:	Show customization options (font, color, language)
14:	Wait for file upload
15:	
16:	If file is uploaded:
17:	Load CSS
18:	Process file with selected customization options
19:	Show progress indicator and final output
20:	Display download button for PDF file
21:	End Function

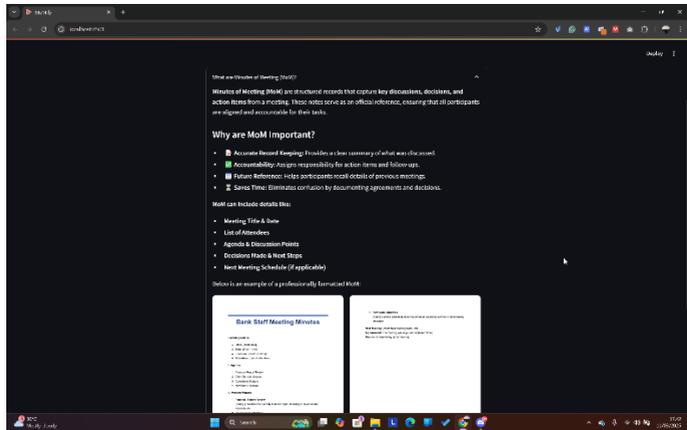
5.1.2 Navigasi Antar Halaman

Navigasi dalam aplikasi dirancang untuk memberikan pengalaman pengguna yang intuitif dan terstruktur. Ketika pengguna pertama kali membuka aplikasi, mereka akan diarahkan ke Landing Page, sebagaimana ditampilkan pada Gambar 5.1. Halaman ini berfungsi sebagai pengantar bagi pengguna untuk memahami konsep *Minutes of Meeting* (MoM) dan bagaimana aplikasi bekerja. Untuk membantu pengguna memahami lebih lanjut, tersedia dua bagian utama yang dapat diperluas (*expandable sections*), yaitu "*What are Minutes of Meeting (MoM)?*" dan "*How Does MoMify*

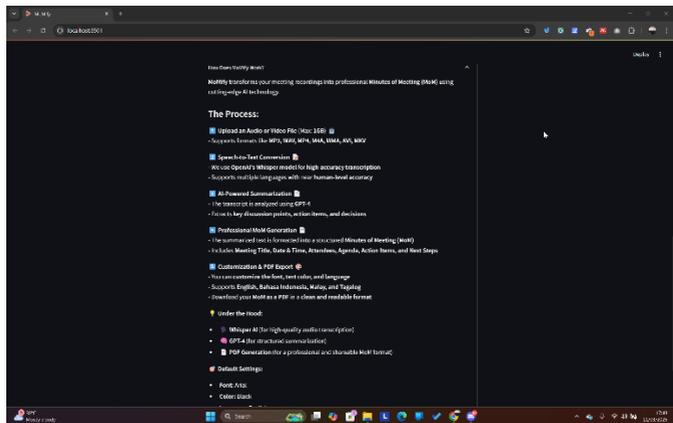
Work?''. Gambar 5.2 menunjukkan bagian pertama yang menjelaskan pentingnya MoM dan elemen-elemen yang biasanya terdapat dalam dokumen MoM. Sementara itu, Gambar 5.3 menampilkan tahapan utama dalam penggunaan aplikasi, mulai dari pengunggahan file, transkripsi dengan Whisper AI, pembuatan ringkasan menggunakan GPT-4, hingga ekspor hasil dalam format PDF. Dengan struktur navigasi ini, pengguna dapat memahami sistem sebelum mulai menggunakan fitur utama aplikasi.



Gambar 5.1 Tampilan Landing Page MoMify yang berisi pengantar tentang Minutes of Meeting dan cara kerja aplikasi



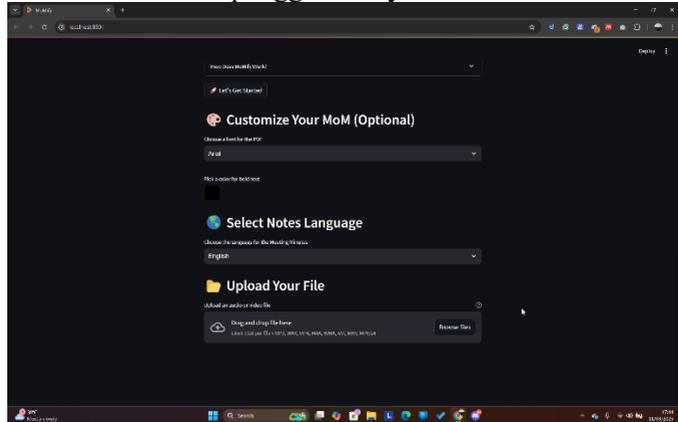
Gambar 5.2 Tampilan expander pada MoMify yang menjelaskan konsep Minutes of Meeting (MoM), manfaatnya, dan contoh format MoM profesional.



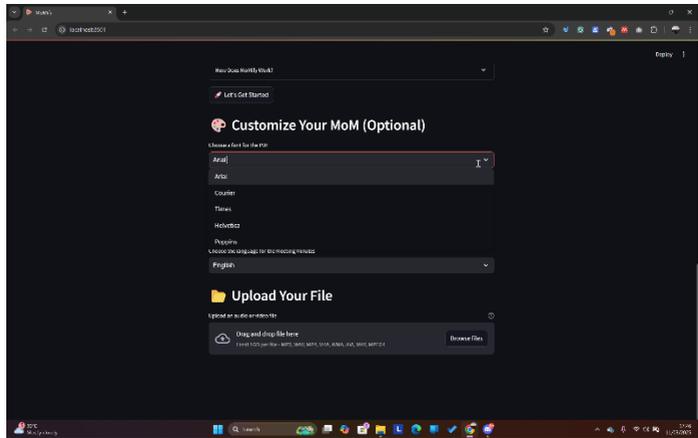
Gambar 5.3 Tampilan expander yang menjelaskan cara kerja MoMify, mulai dari pengunggahan file, transkripsi menggunakan Whisper AI, ringkasan dengan GPT-4, hingga ekspor Minutes of Meeting (MoM) dalam format PDF.

Setelah membaca penjelasan yang tersedia, pengguna dapat melanjutkan dengan menekan tombol "Let's Get Started", yang akan mengarahkan mereka ke

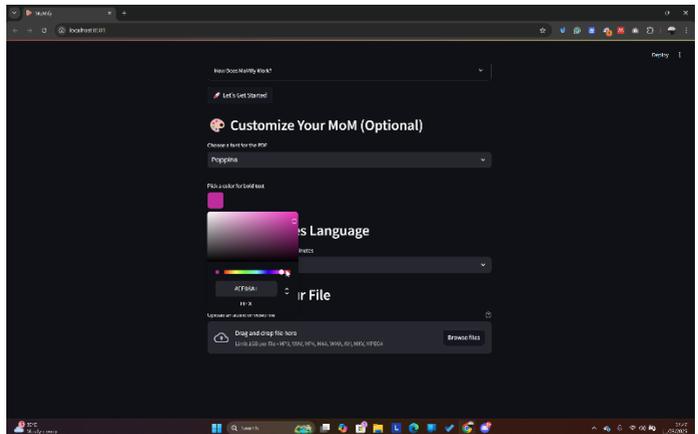
halaman kustomisasi seperti yang ditampilkan pada Gambar 5.4. Pada tahap ini, pengguna diberikan opsi untuk menyesuaikan format dokumen *Minutes of Meeting* (MoM) yang akan dihasilkan, mencakup pemilihan font, warna teks tebal, serta bahasa yang digunakan dalam hasil ringkasan. Pengguna dapat memilih dari berbagai opsi font, seperti *Arial*, *Courier*, *Times*, *Helvetica*, dan *Poppins*, sebagaimana ditampilkan pada Gambar 5.5. Selain itu, pengguna juga dapat menyesuaikan warna teks tebal menggunakan *color picker*, yang memungkinkan pemilihan warna secara fleksibel sesuai preferensi mereka, seperti yang ditunjukkan pada Gambar 5.6. MoMify juga mendukung berbagai pilihan bahasa, termasuk *English*, *Bahasa Indonesia*, *Malay*, dan *Tagalog*, sebagaimana ditampilkan pada Gambar 5.7. Dengan fitur ini, pengguna dapat menghasilkan MoM dalam bahasa yang berbeda dari bahasa asli file yang diunggah, sehingga meningkatkan fleksibilitas dalam penggunaannya.



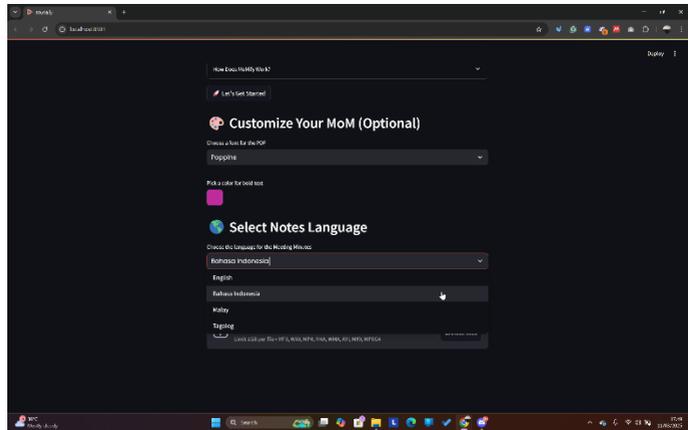
Gambar 5.4 Tampilan setelah pengguna mengklik "Let's Get Started", di mana mereka dapat menyesuaikan format *Minutes of Meeting* (MoM) dengan memilih font, warna teks tebal, dan bahasa sebelum mengunggah file audio atau video.



Gambar 5.5 Tampilan opsi pemilihan font pada fitur *Customize Your MoM*, di mana pengguna dapat memilih berbagai jenis font untuk format *Minutes of Meeting (MoM)* dalam file PDF.

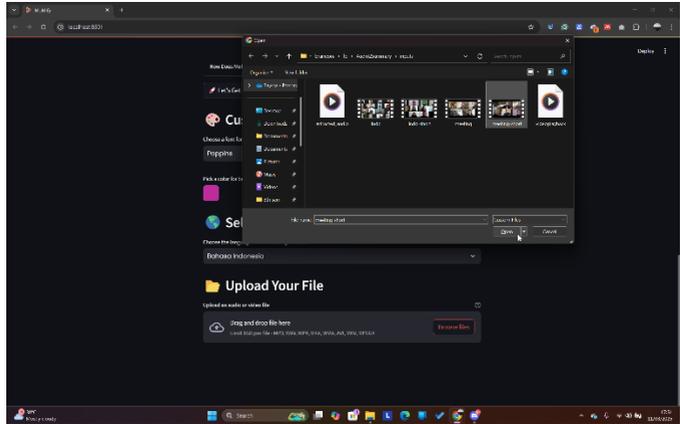


Gambar 5.6 Tampilan fitur *Customize Your MoM* yang memungkinkan pengguna memilih warna untuk teks tebal dalam file *Minutes of Meeting (MoM)* menggunakan pemilih warna (*color picker*).

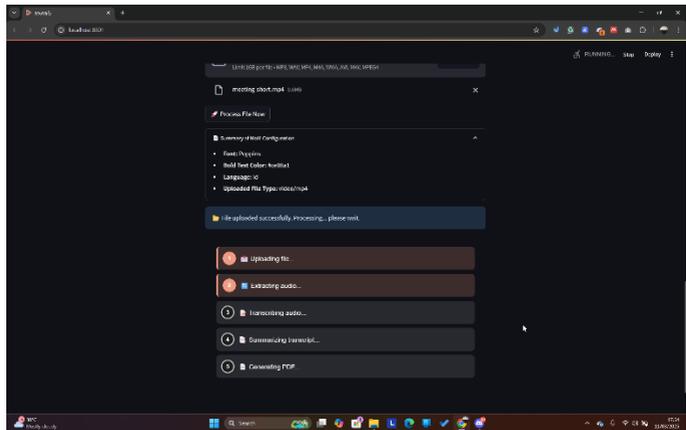


Gambar 5.7 Tampilan fitur *Select Notes Language* yang memungkinkan pengguna memilih bahasa untuk *Minutes of Meeting (MoM)*

Setelah menyelesaikan konfigurasi, pengguna akan diarahkan ke halaman unggahan file, sebagaimana ditampilkan pada Gambar 5.8. Pada halaman ini, pengguna dapat menyeret dan melepaskan (*drag & drop*) atau mencari file audio/video yang ingin mereka transkrip dan ringkas. Setelah file diunggah, sistem akan menampilkan ringkasan konfigurasi dalam *expandable section*, yang mencakup font yang dipilih, warna teks tebal, bahasa, serta format file yang diunggah, sebagaimana ditunjukkan pada Gambar 5.9. Setelah pengguna mengonfirmasi pengaturan ini, mereka dapat menekan tombol "*Process File Now*", yang akan memulai proses pemrosesan file. Selama proses berlangsung, sistem akan menampilkan notifikasi bahwa file sedang diproses dan menampilkan indikator langkah-langkah (*stepper progress*). Tahapan ini mencakup unggahan file, ekstraksi audio, transkripsi menggunakan *Whisper AI*, pembuatan ringkasan dengan *GPT-4*, hingga ekspor hasil ke format PDF.



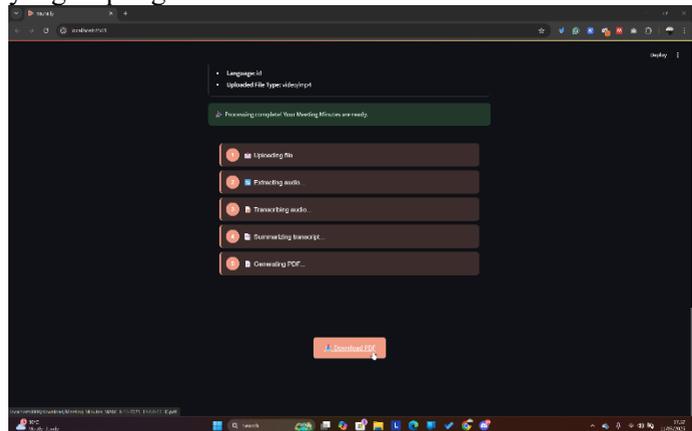
Gambar 5.8 Proses pemilihan file oleh pengguna untuk diunggah ke MoMify, sebelum dilakukan transkripsi dan pembuatan Minutes of Meeting (MoM).



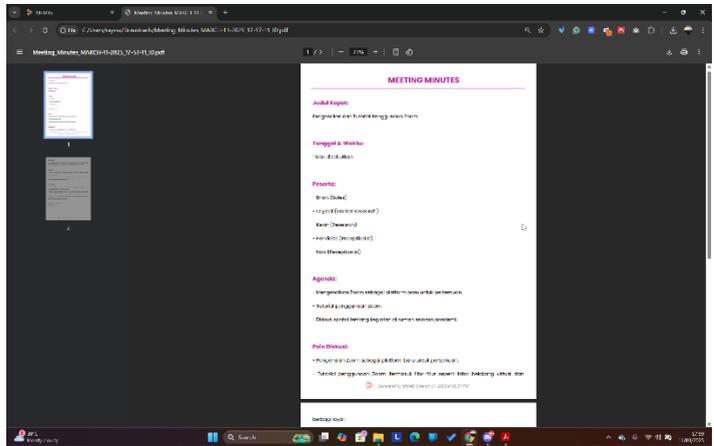
Gambar 5.9 Tampilan proses pemrosesan file setelah pengguna menekan tombol "Process File Now"

Setelah seluruh proses selesai, halaman akhir akan menampilkan notifikasi "Processing complete!" beserta tombol "Download PDF", sebagaimana ditunjukkan pada Gambar 5.10. Tombol ini memungkinkan pengguna untuk

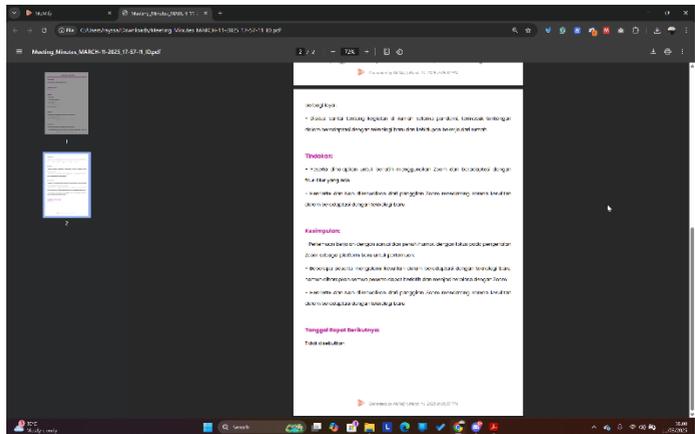
mengunduh hasil *Minutes of Meeting (MoM)* dalam format yang telah dikonfigurasi sebelumnya. Hasil unduhan berupa file PDF yang tersusun rapi sesuai dengan pilihan font, warna teks tebal, dan bahasa yang telah ditentukan sebelumnya. Contoh tampilan halaman pertama dari output PDF dapat dilihat pada Gambar 5.11, sementara halaman kedua yang berisi tindakan yang diambil, kesimpulan, serta informasi tambahan mengenai rapat ditampilkan pada Gambar 5.12. Dengan alur navigasi yang sistematis ini, pengguna dapat dengan mudah memahami langkah-langkah yang perlu dilakukan, mulai dari memahami fungsi aplikasi hingga memperoleh hasil akhir yang siap digunakan.



Gambar 5.10 Tampilan setelah seluruh proses selesai, dengan notifikasi *processing complete* dan tombol *Download PDF* untuk mengunduh hasil *Minutes of Meeting (MoM)*



Gambar 5.11 Contoh output Minutes of Meeting (MoM) dalam format PDF pada halaman 1 dari 2, menampilkan informasi seperti judul rapat, peserta, agenda, dan poin diskusi



Gambar 5.12 Contoh output Minutes of Meeting (MoM) dalam format PDF pada halaman 2 dari 2, menampilkan tindakan yang diambil, kesimpulan, dan tanggal rapat berikutnya.

5.1.3 Pengunggahan File dan Validasi Format

Pada tahap ini, pengguna diharuskan mengunggah file audio atau video yang akan dikonversi menjadi *Minutes of Meeting (MoM)*. Proses ini dikontrol oleh antarmuka unggahan file, sebagaimana dijelaskan dalam Algoritma 5.2, yang menangani tampilan serta mekanisme pemilihan file oleh pengguna. Pengguna dapat mengunggah file dengan cara menyeret dan melepaskan (*drag & drop*) atau menelusuri direktori untuk memilih file secara manual, seperti ditunjukkan pada Gambar 5.8. Sistem mendukung berbagai format *file*, termasuk *MP3*, *WAV*, *MP4*, *M4A*, *WMA*, *AVI*, dan *MKV*, dengan batas ukuran maksimal *1GB per file*.

Algoritma 5.2 Pseudocode untuk Pengunggahan File

Function:	Upload_File
Input:	<ul style="list-style-type: none">• User click events pada tombol unggah file• Uploaded file dalam format audio/video
Output:	<ul style="list-style-type: none">• Tampilan halaman unggahan dengan opsi drag & drop atau pencarian file• Validasi format file sebelum diproses• File siap diproses jika memenuhi persyaratan
1:	Function Upload_File ()
2:	Display "📁 Upload Your File" section
3:	Show file uploader:
4:	- Allowed file types: ["mp3", "wav", "mp4", "m4a", "wma", "avi", "mkv"]
5:	- Display help text: "Limit: 1GB per file"
6:	
7:	If file is uploaded:
8:	Return uploaded file
9:	Else:
10:	Return None
11:	End Function

Setelah file diunggah, sistem akan menampilkan *Summary of MoM Configuration*, yang berisi informasi

mengenai konfigurasi yang telah dipilih oleh pengguna, seperti *font* yang digunakan, warna teks tebal, bahasa, serta jenis file yang diunggah (Gambar 5.9). Jika file yang diunggah tidak sesuai dengan format yang didukung atau melebihi batas ukuran, sistem akan memberikan notifikasi kesalahan dan meminta pengguna untuk mengganti file dengan format yang sesuai. Proses validasi dan ekstraksi audio dari file yang diunggah dikelola oleh *backend*, sebagaimana diuraikan dalam Algoritma 5.3, yang bertugas memverifikasi format file, mengekstrak audio dari video, serta mengonversi audio ke dalam format *WAV* untuk memastikan kompatibilitas dengan sistem transkripsi.

Algoritma 5.3 Pseudocode untuk Validasi dan Ekstraksi Audio

Function:	Validate_And_Extract_Audio
Input:	<ul style="list-style-type: none"> • Uploaded file dalam format audio/video
Output:	<ul style="list-style-type: none"> • Validasi format file • Ekstraksi audio jika file berupa video • Konversi ke WAV jika file berupa audio
1:	Function Validate_And_Extract_Audio(FilePath)
2:	FileExtension ← GetFileExtension(FilePath)
3:	AudioPath ← UPLOAD_DIR + "/extracted_audio.wav"
4:	
5:	Print "Processing file:", FilePath, "(Type:", FileExtension, ")"
6:	
7:	If FileExtension in [".mp4", ".mov", ".avi", ".mkv", ".webm"] Then:
8:	Print "📺 Detected video file. Extracting audio..."
9:	Video ← LoadVideo(FilePath)
10:	If Video.HasAudio is False Then:
11:	RaiseError "❌ No audio found in the video file!"
12:	SaveAudio(Video.Audio, AudioPath, Format = "wav")
13:	

```

14:     Else If FileExtension in [".wav", ".mp3", ".m4a",
15:         ".wma", ".aac", ".flac", ".ogg"] Then:
16:         Print "🎵 Detected audio file. Converting to WAV..."
17:         Audio ← LoadAudio(FilePath)
18:         SaveAudio(Audio, AudioPath, Format="wav")
19:     Else:
20:         RaiseError "❌ Unsupported file format:",
21:             FileExtension
22:     Print "✅ Audio saved to", AudioPath
23:     Return AudioPath
24: End Function

```

5.2 Integrasi *Backend* dan LLM

Bagian ini menjelaskan integrasi backend dengan *Large Language Model (LLM)* untuk mendukung proses transkripsi dan pembuatan *Minutes of Meeting (MoM)* secara otomatis. Backend dikembangkan menggunakan *FastAPI*, yang berfungsi sebagai server untuk menangani permintaan dari frontend dan mengelola pemrosesan data. Sistem ini memanfaatkan *Whisper API* untuk melakukan transkripsi audio menjadi teks dengan akurasi tinggi, sehingga informasi dari file yang diunggah dapat diproses dengan lebih baik. Setelah proses transkripsi, teks yang dihasilkan melalui tahap *chunking* dan *tokenization* agar lebih optimal sebelum diproses lebih lanjut. Model *GPT-4* kemudian digunakan untuk menghasilkan ringkasan *MoM* yang terstruktur dan relevan, memastikan bahwa informasi utama dari pertemuan tetap tersampaikan dengan jelas. Integrasi ini memungkinkan pengguna untuk mendapatkan ringkasan rapat secara efisien dan dapat langsung diunduh dalam format *PDF*.

5.2.1 Implementasi API *Backend* dengan FastAPI

FastAPI digunakan sebagai *backend API* untuk menangani komunikasi antara *frontend* dan layanan pemrosesan data dalam sistem *Minutes of Meeting (MoM)*. *FastAPI* dipilih karena memiliki performa yang cepat, mendukung *asynchronous programming*, serta memudahkan pembuatan dan dokumentasi API.

Backend bertanggung jawab untuk menerima file audio atau video yang diunggah oleh pengguna, memvalidasi format serta ukuran file, dan meneruskan data ke tahap pemrosesan transkripsi dan ringkasan. Setelah file diterima, API mengelola *pipeline* pemrosesan, yang mencakup pemanggilan layanan transkripsi menggunakan *Whisper API* serta pemrosesan ringkasan dengan *GPT-4*.

Untuk menangani unggahan file, sistem menggunakan endpoint khusus yang memungkinkan pengguna mengunggah file serta mengatur preferensi format dokumen, seperti font, warna teks tebal, dan bahasa yang diinginkan. Proses ini dijelaskan dalam *pseudocode* pada Algoritma 5.4. Setelah pemrosesan selesai, API juga menyediakan endpoint untuk mengunduh hasil MoM dalam format PDF, yang dapat diakses oleh pengguna melalui antarmuka aplikasi. Mekanisme ini dirangkum dalam *pseudocode* pada Algoritma 5.5. Dengan arsitektur berbasis *FastAPI*, sistem dapat menangani permintaan secara efisien, memastikan pengguna mendapatkan hasil MoM yang cepat dan akurat.

Algoritma 5.4 Pseudocode untuk menangani unggahan file

Function:	Handle_Upload
Input:	<ul style="list-style-type: none">File: Uploaded file dalam format audio/videoFont: String, pilihan font untuk MoM

	<ul style="list-style-type: none"> • Color: String, warna teks tebal dalam format HEX • Language: String, bahasa untuk ringkasan MoM
--	--

Output:	<ul style="list-style-type: none"> • Menyimpan file yang diunggah ke direktori UPLOAD_DIR • Menampilkan konfigurasi pilihan pengguna (Font, Color, Language) • Mengembalikan respons streaming progres pemrosesan
----------------	--

```

1:  Function Handle_Upload(File, Font, Color, Language)
2:      Define API endpoint "/upload/"
3:      Save Uploaded File to UPLOAD_DIR
4:      Print "☑ Received font:", Font
5:      Print "☑ Received color:", Color
6:      Print "☑ Received language:", Language
7:
8:      Define Function Event_Stream()
9:          Set Filename as NULL
10:         For each Message in Progress_Generator(File,
11:             Font, Color, Language):
12:             Yield Message
13:             If Message starts with "FILENAME::":
14:                 Set Filename = Extract Filename from
15:                     Message
16:             If Filename is NOT NULL:
17:                 Yield "DOWNLOAD::" + Filename
18:         End Function
19:     Return Streaming Response of Event_Stream() with
20:         Media Type "text/event-stream"
21: End Function

```

Algoritma 5.5 Pseudocode untuk menangani unduhan file hasil MoM

Function:	Handle_Download
------------------	-----------------

Input:	<ul style="list-style-type: none"> • Filename: Nama file PDF yang akan diunduh
---------------	---

Output:	<ul style="list-style-type: none"> • Jika file ditemukan, mengembalikan file PDF ke pengguna
----------------	---

-
- Jika file tidak ditemukan, mengembalikan pesan error respons streaming progres pemrosesan
-

```
1: Function Handle_Download(Filename)
2: Define API endpoint "%download/{filename}"
3: Retrieve File Path from OUTPUT_DIR
4: If File does not exist:
5:     Print "✗ File not found!"
6:     Return Error Message "File not found"
7: Else:
8:     Return File Response with:
9:     - Media Type "application/pdf"
10:    - Filename as Downloaded File
11:    - Header "Content-Disposition: attachment; filename={filename}"
12: End Function
```

5.2.2 Pemrosesan Transkripsi Menggunakan Whisper API

Untuk menghasilkan *Minutes of Meeting* (MoM) yang akurat, sistem menggunakan Whisper dari OpenAI sebagai layanan transkripsi utama. Whisper dipilih karena memiliki kemampuan *speech-to-text* dengan akurasi tinggi serta mendukung berbagai bahasa, sehingga memungkinkan sistem untuk mengonversi percakapan dari file audio atau video menjadi teks dengan minim kesalahan.

Proses transkripsi diawali dengan validasi dan ekstraksi audio, terutama jika file yang diunggah berupa video. Setelah audio diekstrak, sistem menjalankan Whisper secara lokal untuk mengonversi suara menjadi teks, tanpa perlu bergantung pada API eksternal. Algoritma 5.6 menjelaskan proses ini, termasuk bagaimana sistem memuat model Whisper dan menyesuaikan penggunaan GPU atau CPU untuk meningkatkan efisiensi pemrosesan.

Algoritma 5.6 Pseudocode untuk transkripsi audio

Function:	Transcribe_Audio
------------------	------------------

Input:	<ul style="list-style-type: none">• Audio_File_Path: Lokasi file audio yang akan ditranskripsi
---------------	--

Output:	<ul style="list-style-type: none">• Transcribed_Text: Hasil transkripsi dari audio dalam format teks
----------------	--

1:	Function Transcribe_Audio(Audio_File_Path)
2:	If GPU is not available:
3:	Print "⚠ Warning: No GPU detected, running on CPU. This will be slower!"
4:	
5:	Print "Loading Whisper model..."
6:	Load Whisper model with size "medium"
7:	Assign model to GPU if available, otherwise use CPU
8:	
9:	Print "✅ Model is running on: (GPU/CPU)"
10:	Process transcription using Whisper API on Audio_File_Path
11:	Retrieve transcribed text from Whisper API response
12:	Return Transcribed_Text
13:	End Function

Dalam implementasinya, sistem menggunakan model *medium* karena memberikan keseimbangan terbaik antara akurasi dan efisiensi pemrosesan. Berdasarkan Tabel 5.1, model *medium* membutuhkan sekitar 5GB VRAM dan menawarkan kecepatan transkripsi sekitar 2x lebih lambat dibandingkan waktu nyata (*real-time*). Namun, model ini tetap lebih cepat dibandingkan model *large*, yang membutuhkan 10GB VRAM dengan kecepatan *real-time*. Pemilihan model *medium* didasarkan pada beberapa pertimbangan berikut:

- Lebih akurat dibandingkan model kecil (*tiny, base, small*).
- Lebih efisien dibandingkan model *large*, yang memiliki kebutuhan VRAM lebih tinggi.

- Mendukung transkripsi dalam berbagai bahasa, termasuk Bahasa Indonesia.

Tabel 5.1 Perbandingan ukuran model Whisper berdasarkan jumlah parameter, kebutuhan VRAM, dan kecepatan relatif terhadap waktu nyata.

Size	Parameters	Required VRAM	Relative speed
Tiny	39 M	~1 GB	~10×
Base	74 M	~1 GB	~7×
Small	244 M	~2 GB	~4×
Medium	769 M	~5 GB	~2×
Large	1550 M	~10 GB	1×
Turbo	809 M	~6 GB	~8×

Setelah transkripsi selesai, hasil teks mentah ini digunakan sebagai dasar untuk tahap pemrosesan berikutnya, yaitu pembuatan ringkasan menggunakan GPT-4. Selain itu, sistem menangani berbagai kendala teknis seperti kebisingan latar belakang dan variasi aksen, yang dapat diminimalkan melalui algoritma pemrosesan audio tambahan sebelum dikirim ke Whisper. Dengan integrasi ini, pengguna dapat memperoleh transkripsi yang cepat dan akurat, sehingga mendukung kemudahan dalam pembuatan MoM secara otomatis.

5.2.3 Chunking dan Tokenization dalam Ringkasan Teks

Dalam proses pembuatan *Minutes of Meeting* (MoM), teks hasil transkripsi sering kali memiliki panjang yang melebihi batas pemrosesan *Large Language Model* (LLM) seperti GPT-4. Oleh karena itu, diperlukan teknik *chunking* dan *tokenization* untuk membagi teks menjadi bagian-bagian yang lebih kecil agar dapat diproses secara efisien. Langkah pertama dalam proses ini adalah menghitung jumlah token dalam teks menggunakan algoritma yang diuraikan dalam Algoritma 5.7. Fungsi ini memastikan bahwa setiap bagian teks tidak melebihi batas maksimal token yang dapat diproses oleh model, sehingga menghindari pemangkasan informasi yang penting.

Setelah jumlah token dihitung, teks kemudian dibagi menjadi beberapa segmen (*chunks*) menggunakan metode yang dijelaskan dalam Algoritma 5.8, yang mempertimbangkan batas token serta struktur kalimat agar tetap mempertahankan konteks dan keterpaduan isi.

Algoritma 5.7 Pseudocode Menghitung Jumlah Token

Function:	Count_Tokens
------------------	--------------

Input:	<ul style="list-style-type: none"> Text: String teks yang akan dihitung jumlah tokennya. Model: Nama model OpenAI yang digunakan untuk tokenisasi (default: "gpt-4").
---------------	---

Output:	<ul style="list-style-type: none"> Token_Count: Jumlah token dalam teks yang diberikan.
----------------	--

1:	Function Count_Tokens(Text, Model="gpt-4")
2:	Load tokenizer for the given Model
3:	Encode the Text using the tokenizer
4:	Token_Count ← Length of encoded tokens
5:	Return Token_Count
6:	End Function

Algoritma 5.8 Pseudocode membagi teks menjadi chunks

Function:	Split_Text_Into_Chunks
------------------	------------------------

Input:	<ul style="list-style-type: none"> Text: String teks panjang yang akan dibagi menjadi beberapa bagian (<i>chunks</i>) Max_Tokens: Jumlah token maksimum per chunk (default: 7500).
---------------	--

Output:	<ul style="list-style-type: none"> Chunks: Daftar (<i>list</i>) dari teks yang telah dibagi berdasarkan batasan token.
----------------	---

1:	Function Split_Text_Into_Chunks (Text, Max_Tokens=7500)
2:	Words ← Split Text into individual words
3:	Chunks ← Empty List
4:	Current_Chunk ← Empty List
5:	Current_Token_Count ← 0
6:	
7:	For each Word in Words do:
8:	Word_Token_Count ← Count_Tokens(Word)

```

9:         If Current_Token_Count + Word_Token_Count
           > Max_Tokens then:
10:             Append Current_Chunk as a string to Chunks
11:             Current_Chunk ← Empty List
12:             Current_Token_Count ← 0
13:         Append Word to Current_Chunk
14:         Current_Token_Count ← Current_Token_Count
           + Word_Token_Count

15:
16:     If Current_Chunk is not empty then:
17:         Append Current_Chunk as a string to Chunks
18:
19:     Return Chunks
20: End Function

```

Chunking adalah teknik yang digunakan untuk membagi teks panjang menjadi beberapa bagian (*chunks*) yang lebih kecil dan tetap bermakna. Pembagian ini dilakukan berdasarkan panjang token, struktur kalimat, dan topik pembicaraan agar informasi dalam teks tetap utuh. Model GPT-4 memiliki batas token maksimal dalam satu permintaan, sehingga teks perlu dibagi menjadi segmen yang sesuai dengan batas ini. Selain itu, *chunking* juga dilakukan berdasarkan jeda alami dalam teks, seperti titik atau paragraf, untuk mencegah pemotongan informasi di tengah kalimat. Jika memungkinkan, sistem juga mengelompokkan teks berdasarkan topik diskusi dalam rapat agar ringkasan yang dihasilkan tetap memiliki kesinambungan konteks.

Setelah teks dibagi menjadi beberapa *chunks*, langkah selanjutnya adalah *tokenization*. *Tokenization* adalah proses mengubah teks menjadi unit-unit kecil yang disebut *tokens*, yang dapat berupa kata, frasa, atau bahkan karakter, tergantung pada aturan pemrosesan bahasa alami (*Natural Language Processing* atau NLP). Proses ini penting karena model GPT-4 tidak memproses teks dalam bentuk kata langsung, tetapi dalam bentuk *tokens*. Dengan melakukan tokenisasi terlebih dahulu, sistem dapat mengelola panjang teks yang dikirim ke model sehingga

pemrosesan menjadi lebih efisien. Selain itu, teknik ini memastikan bahwa tidak ada informasi penting yang terpotong akibat batas panjang input model.

Dalam implementasinya, sistem MoMify menggunakan kombinasi *chunking* berbasis panjang token dan struktur kalimat untuk memastikan bahwa transkripsi dapat diproses secara optimal. Setiap bagian teks yang telah diproses oleh Algoritma 5.8 kemudian dihitung jumlah tokennya menggunakan metode dalam Algoritma 5.7, untuk memastikan setiap *chunk* tidak melebihi batas token yang dapat diproses oleh GPT-4 untuk menghasilkan ringkasan MoM.

5.2.4 Ringkasan MoM Menggunakan GPT-4

Setelah proses transkripsi selesai dan teks telah melalui tahap *chunking* serta validasi jumlah token, sistem MoMify memasuki tahap akhir, yaitu menghasilkan *Minutes of Meeting (MoM)* menggunakan GPT-4. Ringkasan ini bertujuan untuk mengekstrak poin-poin utama dari hasil transkripsi secara ringkas dan terstruktur. Dengan pendekatan ini, pengguna dapat memahami inti diskusi rapat tanpa harus membaca keseluruhan transkripsi yang mungkin sangat panjang. Penyusunan ringkasan dilakukan dengan mempertimbangkan struktur yang telah ditentukan, seperti agenda rapat, peserta, poin diskusi utama, serta keputusan dan tindakan lanjutan. Dengan adanya struktur ini, hasil ringkasan lebih sistematis dan mudah dipahami oleh semua peserta rapat.

Proses pembuatan ringkasan dilakukan dengan mengirimkan setiap bagian teks hasil *chunking* ke GPT-4 melalui prompt yang telah dirancang secara spesifik. Algoritma 5.9 menjelaskan bagaimana sistem mengambil setiap *chunk*, mengirimkannya ke model GPT-4 dengan instruksi untuk mengidentifikasi informasi penting seperti agenda, peserta rapat, dan keputusan yang diambil. Selain itu, model juga mempertimbangkan kesinambungan

konteks antara *chunk* sebelumnya dan berikutnya agar hasil ringkasan tetap kohesif. Penggunaan *chunking* ini penting karena GPT-4 memiliki batasan dalam jumlah token yang dapat diproses dalam satu kali eksekusi. Dengan cara ini, sistem dapat menghindari kehilangan informasi akibat keterbatasan kapasitas pemrosesan model.

Algoritma 5.9 Pseudocode untuk Meringkas Transkripsi ke Format MoM

Function:	Summarize_Transcription
Input:	<ul style="list-style-type: none"> transcription → Teks hasil transkripsi rapat. language → Bahasa untuk ringkasan (<i>default: English</i>).
Output:	<ul style="list-style-type: none"> MoM_Summary → Ringkasan <i>Minutes of Meeting</i> dalam format yang terstruktur.


```

1:  Function      Summarize_Transcription(transcription,
      language)
2:      Print " ✨ Generating Minutes of Meeting..."
3:      Chunks ← Call
      Split_Text_Into_Chunks(transcription)
4:      Print " ✨ Transcript split into", Length(Chunks),
      "chunks."
5:      Initialize GPT_Model with "gpt-4"
6:      Initialize Summarized_Chunks as an empty list
7:
8:      For each Chunk in Chunks:
9:          Print " ✨ Summarizing Chunk", Index(Chunk),
          "of", Length(Chunks)
10:         Prompt ← Generate prompt for GPT-4 with
          required MoM Sections
11:         Summary ← Call GPT_Model(Prompt)
12:         Append Summary to Summarized_Chunks
13:
14:         Define MoM_Headers for supported languages
15:         Define default Not_Mentioned_Text for missing
          Date & Time and Next Meeting fields
16:
17:         Print " ✨ Merging summarized chunks into final
          MoM..."

```

- 18:** Final_Prompt ← Combine Summarized_Chunks into structured MoM format
Ensure all sections appear on a new line, with consistent formatting
MoM_Summary ← Call GPT_Model(Final_Prompt)
- 19:** Return MoM_Summary
- 20:** End Function
-

Setelah proses transkripsi selesai dan teks telah melalui tahap *chunking* serta validasi jumlah token, sistem MoMify memasuki tahap akhir, yaitu menghasilkan *Minutes of Meeting (MoM)* menggunakan GPT-4. Ringkasan ini bertujuan untuk mengekstrak poin-poin utama dari hasil transkripsi secara ringkas dan terstruktur. Dengan pendekatan ini, pengguna dapat memahami inti diskusi rapat tanpa harus membaca keseluruhan transkripsi yang mungkin sangat panjang. Penyusunan ringkasan dilakukan dengan mempertimbangkan struktur yang telah ditentukan, seperti agenda rapat, peserta, poin diskusi utama, serta keputusan dan tindakan lanjutan. Dengan adanya struktur ini, hasil ringkasan lebih sistematis dan mudah dipahami oleh semua peserta rapat.

Proses pembuatan ringkasan dilakukan dengan mengirimkan setiap bagian teks hasil *chunking* ke GPT-4 melalui prompt yang telah dirancang secara spesifik. Algoritma 5.9 menjelaskan bagaimana sistem mengambil setiap *chunk*, mengirimkannya ke model GPT-4 dengan instruksi untuk mengidentifikasi informasi penting seperti agenda, peserta rapat, dan keputusan yang diambil. Selain itu, model juga mempertimbangkan kesinambungan konteks antara *chunk* sebelumnya dan berikutnya agar hasil ringkasan tetap kohesif. Penggunaan *chunking* ini penting karena GPT-4 memiliki batasan dalam jumlah token yang dapat diproses dalam satu kali eksekusi. Dengan cara ini, sistem dapat menghindari kehilangan informasi akibat keterbatasan kapasitas pemrosesan model.

Sistem ini juga mendukung berbagai bahasa, sesuai dengan preferensi yang ditentukan pengguna saat mengunggah file. Dengan begitu, MoMify dapat menghasilkan ringkasan MoM dalam Bahasa Indonesia, Inggris, Melayu, atau Tagalog tanpa mengurangi akurasi isi ringkasan. Pemilihan bahasa ini dilakukan dengan memasukkan parameter ke dalam prompt, memastikan model menghasilkan output yang konsisten dalam bahasa yang diinginkan. Selain itu, sistem memastikan bahwa setiap istilah teknis atau spesifik yang digunakan dalam rapat tetap dipertahankan agar tidak terjadi perubahan makna dalam ringkasan. Dengan adanya fitur ini, pengguna dari berbagai latar belakang dapat memahami hasil rapat dalam bahasa yang lebih familiar bagi mereka.

Untuk meningkatkan kualitas hasil, sistem menerapkan *post-processing* setelah menerima output dari GPT-4. Langkah ini mencakup pemformatan ulang teks agar lebih terstruktur dan mudah dibaca oleh pengguna. Selain itu, sistem menyesuaikan gaya bahasa sesuai kebutuhan, baik untuk format yang lebih formal maupun lebih ringkas sesuai konteks penggunaannya. Proses validasi juga dilakukan untuk memastikan bahwa informasi penting tidak hilang selama proses peringkasan. Dengan adanya *post-processing*, hasil akhir yang diberikan kepada pengguna menjadi lebih profesional dan siap digunakan dalam laporan atau tindak lanjut rapat.

Dengan pendekatan ini, MoMify dapat menghasilkan *Minutes of Meeting* yang informatif, ringkas, dan mudah dipahami, sehingga mendukung dokumentasi rapat yang lebih efisien. Kemampuan sistem dalam mengolah transkripsi menjadi ringkasan yang terstruktur membantu pengguna untuk lebih cepat menangkap poin-poin penting dalam rapat. Selain itu, integrasi GPT-4 dengan proses *chunking* dan *tokenization* memastikan bahwa hasil yang diberikan tetap akurat dan

tidak kehilangan konteks dari percakapan asli. Kemudahan ini sangat bermanfaat bagi organisasi atau perusahaan yang sering mengadakan rapat dan membutuhkan dokumentasi yang dapat diakses dengan cepat.

5.3 Pembuatan dan Pengelolaan Output MoM

Bagian ini akan menjelaskan bagaimana *Minutes of Meeting (MoM)* yang telah diringkas dikelola dan disajikan dalam format yang mudah dibaca serta dapat diakses oleh pengguna. Proses ini mencakup pembuatan struktur dokumen MoM yang sistematis agar setiap informasi penting, seperti agenda, peserta, poin diskusi, keputusan, dan tindakan lanjutan, tersusun dengan jelas. Selain itu, sistem MoMify juga menyediakan fitur ekspor ke format PDF untuk mempermudah penyimpanan dan distribusi hasil rapat dalam bentuk dokumen yang profesional. Tidak hanya itu, untuk memastikan kelancaran proses dan meminimalkan kesalahan, sistem dilengkapi dengan mekanisme penanganan eror serta pencatatan log pemrosesan. Dengan pendekatan ini, pengguna dapat dengan mudah mendapatkan ringkasan MoM yang akurat, terstruktur, dan siap digunakan untuk keperluan dokumentasi serta tindak lanjut rapat.

5.3.1 Format dan Struktur Dokumen MoM

Supaya *Minutes of Meeting (MoM)* yang dihasilkan mudah dibaca dan digunakan, sistem *MoMify* menerapkan format dan struktur dokumen yang sistematis. Struktur ini dirancang untuk mencakup informasi utama dari sebuah rapat, sehingga pengguna dapat dengan cepat memahami poin-poin penting yang dibahas tanpa harus membaca keseluruhan transkripsi. Dokumen *MoM* terdiri dari

beberapa bagian utama, seperti judul rapat, tanggal dan waktu, daftar peserta, agenda, poin diskusi, tindakan yang harus dilakukan (*action items*), kesimpulan, serta jadwal rapat berikutnya jika tersedia. Setiap bagian diformat secara jelas dan profesional agar informasi tersaji secara rapi dan mudah diakses oleh pengguna. Pemformatan ini dijelaskan lebih lanjut dalam Algoritma 5.10, yang menguraikan langkah-langkah sistem dalam menyusun struktur dokumen *MoM* dalam format PDF.

Algoritma 5.10 Pseudocode untuk Pemformatan Struktur MoM dalam PDF

Function:	Format_MoM_PDF
Input:	<ul style="list-style-type: none"> • summary → Teks ringkasan MoM • filename → Nama file output PDF • font → Font yang dipilih pengguna • color → Warna teks dalam format HEX • language → Bahasa yang dipilih untuk MoM
Output:	<ul style="list-style-type: none"> • Dokumen MoM dalam format PDF dengan struktur dan pemformatan yang sesuai.
1:	Function Format_MoM_PDF(summary, filename, font, color, language)
2:	Create OUTPUT_DIR if it does not existPrint " ✨ Generating Minutes of Meeting..."
3:	Set formatted_date ← Current date in format "Month-Day-Year_HH-MM-SS"
4:	Define filename as "Meeting_Minutes_" + formatted_date + "_" + language + ".pdf"
5:	
6:	Convert color from HEX to RGB → (r, g, b)
7:	Create new PDF document
8:	Enable auto page break with margin of 15
9:	Add new page to PDF
10:	
11:	Register and set the chosen font:
12:	Add the selected font (Regular & Bold)
13:	Set default font size to 12 (Normal text)
14:	Set Bold font size to 14 for section headers

- 15:
 - 16: Format the "MEETING MINUTES" Title
 - 17: Set font to Bold, size 20
 - 18: Set text color to (r, g, b)
 - 19: Add "MEETING MINUTES" title centered on the page
 - 20: Add a horizontal line below the title using (r, g, b) color
 - 21: Add spacing before content
 - 22:
 - 23: Remove Markdown "bold" markers from the summary text
 - 24:
 - 25: Define section headers based on the selected language
 - 26:
 - 27: Process the summary text:
 - 28: Split the summary text into lines
 - 29: For each line:
 - 30: If the line is a header, set font Bold (14) & text color (r, g, b)
 - 31: If not a header, set font Normal (12) & text color black
 - 32: Add the formatted line to the PDF
 - 33: Save the PDF file in `OUTPUT_DIR`
 - 34: Return the filename of the exported PDF
 - 35: End Function
-

Dalam penyajian informasi, sistem menggunakan *bullet points* untuk daftar peserta, agenda, dan tugas yang harus dilakukan, guna meningkatkan keterbacaan dokumen. Sementara itu, bagian diskusi dan kesimpulan disusun dalam bentuk paragraf ringkas agar alur informasi tetap jelas dan mudah dipahami. Selain itu, pengguna dapat menyesuaikan tampilan dokumen dengan memilih font, warna teks, dan bahasa yang diinginkan, sehingga hasil akhir *MoM* dapat disesuaikan dengan preferensi individu maupun kebutuhan organisasi. Fitur ini memastikan bahwa *MoMify* dapat digunakan secara fleksibel dalam berbagai lingkungan kerja.

Supaya struktur dokumen tetap terorganisir dengan baik, sistem *MoMify* menerapkan aturan pemisahan bagian berdasarkan *header* standar. Setiap elemen informasi dipastikan memiliki tempat yang jelas dalam dokumen, sehingga tidak ada bagian penting yang terabaikan. Jika ada informasi yang tidak disebutkan dalam transkripsi atau tidak tersedia dalam hasil ringkasan, sistem akan memberikan indikator bahwa data tersebut tidak disebutkan dalam rapat. Dengan penerapan format ini, pengguna dapat dengan mudah menavigasi isi dokumen dan menemukan informasi yang relevan tanpa kesulitan.

5.3.2 Ekspor MoM ke PDF

Setelah proses transkripsi dan ringkasan selesai, sistem *MoMify* menyediakan fitur ekspor *Minutes of Meeting (MoM)* ke dalam format PDF agar dokumen dapat dengan mudah dibagikan, disimpan, dan diarsipkan. PDF dipilih sebagai format keluaran karena kompatibilitasnya yang luas, kemampuannya dalam mempertahankan tata letak dokumen, serta memastikan bahwa struktur dokumen tidak berubah saat dibuka di perangkat yang berbeda. Proses ekspor ini melibatkan beberapa tahap, termasuk pemformatan teks, penerapan struktur MoM, serta penyesuaian tampilan berdasarkan preferensi pengguna, seperti pemilihan font, warna teks, dan bahasa yang digunakan.

Tahapan ekspor ini dijelaskan dalam Algoritma 5.11, yang mengatur proses pembuatan PDF hingga penyediaan tombol unduhan di antarmuka pengguna. Sistem pertama-tama memanggil fungsi `Format_MoM_PDF` untuk menerapkan struktur dokumen, memastikan setiap bagian dari MoM tersusun dengan baik sesuai standar yang telah ditetapkan. Setelah PDF berhasil dibuat, sistem menghasilkan URL unduhan yang mengarah ke file yang telah diproses. Jika file tersedia, antarmuka pengguna akan menampilkan tombol

Download PDF, memungkinkan pengguna untuk mengunduh dokumen dengan mudah.

Algoritma 5.11 Pseudocode untuk Ekspor and Unduh MoM sebagai PDF

Function:	Export_MoM_To_PDF
------------------	-------------------

Input:	<ul style="list-style-type: none">• summary → teks MoM yang sudah diformat• font → teks font yang dipilih• color → kode warna HEX untuk headers• language → bahasa output MoM
---------------	--

Output:	<ul style="list-style-type: none">• File PDF berisi MoM dokumen yang terstruktur
----------------	--

1:	Function Export_MoM_To_PDF(summary, font, color, language)
2:	Call Format_MoM_PDF(summary, filename, font, color, language)
3:	If filename is not empty:
4:	Construct download URL → API_URL + "/download/" + filename
5:	Display a styled download button with the generated URL
6:	Else:
7:	Show an error message: " ❌ Error retrieving the file!"
8:	End Function

Proses ekspor ini juga dirancang untuk menangani berbagai skenario, termasuk pengelolaan dokumen panjang yang terdiri dari beberapa halaman. Sistem secara otomatis menyesuaikan tata letak halaman, mengatur pemisahan bagian agar tidak terpotong secara tidak wajar, serta menambahkan nomor halaman untuk mempermudah navigasi. Jika terjadi kesalahan dalam proses konversi atau jika parameter yang diberikan tidak valid, sistem akan memberikan notifikasi kepada pengguna dan meminta mereka untuk memperbaiki input sebelum mencoba kembali.

Dengan fitur ekspor ini, *MoMify* memastikan bahwa hasil *Minutes of Meeting* tidak hanya dapat diakses secara

digital tetapi juga dapat dicetak dan digunakan sebagai referensi dalam pertemuan mendatang. Implementasi sistem yang mengintegrasikan *formatting*, ekspor, dan mekanisme unduhan yang efisien memungkinkan pengguna mendapatkan dokumen MoM dalam format yang terstruktur, profesional, dan fleksibel. Hal ini menjadikan *MoMify* sebagai alat yang efektif dalam mendukung dokumentasi rapat secara otomatis

5.3.3 Penanganan Error dan Log Pemrosesan

Dalam sistem *MoMify*, penanganan error dan pencatatan log pemrosesan merupakan aspek penting untuk memastikan kelancaran dalam menghasilkan *Minutes of Meeting* (MoM). Sistem ini menerapkan berbagai mekanisme untuk menangani potensi kesalahan yang dapat terjadi selama proses unggah file, transkripsi, ringkasan, hingga ekspor ke PDF. Error yang terjadi akan ditangani secara otomatis atau melalui notifikasi kepada pengguna agar mereka dapat mengambil langkah yang diperlukan untuk memperbaiki masalah.

Salah satu mekanisme utama adalah validasi file yang diunggah, sebagaimana dijelaskan dalam Algoritma 5.12. Fungsi ini memastikan bahwa file yang diunggah sesuai dengan format yang didukung, memiliki ukuran yang tidak melebihi batas yang ditentukan, dan jika file berupa video, harus memiliki *audio track* yang valid. Jika terjadi kesalahan, sistem akan memberikan peringatan seperti “*Unsupported file format*” atau “*No audio detected*” sebelum file diproses lebih lanjut.

Algoritma 5.12 Pseudocode untuk Penanganan File Error

Function:	Handle_File_Error
Input:	• File → file audio/video file yang diunggah
Output:	• Pesan error atau konfirmasi file valid
I:	Function Handle_File_Error(file)

```

2:     If file is NULL:
3:         Return "✘ No file uploaded. Please select a file."
4:
5:     Set allowed_formats ← ["mp3", "wav", "mp4",
6:     "m4a", "wma", "avi", "mkv"]
7:     Set file_extension ← GetFileExtension(file)
8:
9:     If file_extension NOT in allowed_formats:
10:        Return "✘ Unsupported file format. Please
11:        upload a valid audio or video file."
12:
13:    If GetFileSize(file) > 1GB:
14:        Return "✘ File size exceeds limit (1GB). Please
15:        upload a smaller file."
16:
17:    If file is a video:
18:        Extract audio track
19:        If audio track is NULL:
20:            Return "✘ No audio detected in the video file."
21:            Return "☑ File is valid and ready for
22:            processing."
23:    End Function

```

Untuk memantau status setiap tahap pemrosesan, sistem menerapkan pencatatan log sebagaimana diilustrasikan dalam Algoritma 5.13. Setiap langkah yang berhasil atau gagal akan dicatat dengan *timestamp*, memungkinkan administrator atau pengguna untuk meninjau proses dan mengidentifikasi sumber kesalahan. Jika suatu tahap mengalami kegagalan, sistem dapat memberikan log error yang menunjukkan bagian mana yang bermasalah.

Algoritma 5.13 Pseudocode untuk Tahapan Log Proses

Function:	Log_Processing_Steps
------------------	----------------------

Input:	<ul style="list-style-type: none"> • Step → tahap proses saat ini • Status → gagal/berhasil
---------------	---

Output:	<ul style="list-style-type: none"> • Entri log untuk tahap proses
----------------	--

1: Function Log_Processing_Steps(step, status)

```

2:     Set timestamp ← GetCurrentTime()
3:     If status is "success":
4:         Print "[✓ " + timestamp + "] Step: " + step + " -
           Completed Successfully."
5:     Else:
6:         Print "[✗ " + timestamp + "] Step: " + step + " -
           LogError(step)
7:     End Function

```

Dalam kasus tertentu, kegagalan bisa bersifat sementara, seperti gangguan jaringan atau beban server yang tinggi. Oleh karena itu, sistem menerapkan mekanisme *retry processing*, seperti yang dijelaskan dalam Algoritma 5.14. Jika suatu proses gagal, sistem akan mencoba kembali beberapa kali sebelum menganggapnya gagal total. Misalnya, jika terjadi kegagalan saat memanggil model GPT-4 untuk membuat ringkasan, sistem dapat melakukan beberapa percobaan ulang sebelum memberikan notifikasi kesalahan kepada pengguna.

Algoritma 5.14 Pseudocode untuk Mencoba Proses

Function:	Retry_Processing
Input:	<ul style="list-style-type: none"> • Function → fungsi yang akan diulang • Max_attempts → banyaknya waktu untuk mencoba
Output:	<ul style="list-style-type: none"> • Fungsi berhasil dieksekusi atau pesan error
1:	Function Retry_Processing(function, max_attempts)
2:	Set attempt ← 0
3:	While attempt < max_attempts:
4:	Try:
5:	Call function()
6:	Return "[✓] Processing successful."
7:	Catch Exception:
8:	attempt ← attempt + 1
9:	Print "[△] Attempt " + attempt + " failed. Retrying..."

- 10: Return "✘ Maximum retry attempts reached.
 Processing failed."
11: End Function

Selain itu, sistem juga menangani berbagai jenis kesalahan lain yang lebih luas, sebagaimana diuraikan dalam Algoritma 5.15. Fungsi ini bertanggung jawab untuk memberikan respons yang sesuai terhadap berbagai jenis kesalahan sistem, seperti kegagalan transkripsi, kesalahan saat mengekspor PDF, atau kendala dalam mengakses layanan eksternal. Pesan kesalahan yang diberikan juga disertai dengan rekomendasi solusi bagi pengguna, misalnya mengunggah ulang file, memeriksa koneksi internet, atau mencoba format file yang berbeda.

Algoritma 5.15 Pseudocode untuk menangani sistem error

Function:	Handle_System_Error
------------------	---------------------

Input:	• error_message → deskripsi error
Output:	• Menampilkan pesan error dan saran solusi

```

1:     Function Handle_System_Error(error_message)
2:         Print "✘ Error encountered: " + error_message
3:         If error_message contains "file format":
4:             Suggest "Please upload a supported audio or video
               file."
5:         If error_message contains "file size":
6:             Suggest "Try compressing the file or selecting a
               smaller file."
7:         If error_message contains "no audio":
8:             Suggest "Ensure the video contains an audio track
               before uploading."
9:         If error_message contains "network":
10:             Suggest "Check your internet connection and try
               again."
11:         LogError(error_message)
12:     End Function

```

Meskipun contoh-contoh penanganan error yang dijelaskan di atas merupakan bagian penting dalam sistem MoMify, masih terdapat berbagai mekanisme lainnya

yang diterapkan di *backend*. Misalnya, sistem juga menangani kegagalan dalam tahap transkripsi jika model Whisper tidak dapat mengenali suara, atau kegagalan dalam ekspor PDF jika terjadi kesalahan format dalam ringkasan. Semua skenario ini telah diantisipasi dan dapat ditemukan dalam berbagai pseudocode sebelumnya. Dengan pendekatan ini, MoMify memastikan bahwa proses otomatisasi *Minutes of Meeting* tetap berjalan secara stabil, dapat diandalkan, dan memberikan pengalaman pengguna yang lebih baik.

5.4 Repositori dan Kode Sumber

Repositori kode sumber untuk sistem *MoMify* tersedia secara publik di [GitHub](#), memungkinkan pengembang lain untuk mempelajari, menguji, atau mengembangkan fitur lebih lanjut. Repositori ini mencakup seluruh implementasi mulai dari frontend menggunakan *Streamlit*, backend berbasis *FastAPI*, hingga pemrosesan transkripsi dan ringkasan menggunakan *Whisper* dan *GPT-4*. Struktur kode disusun secara modular untuk mempermudah kontribusi dan pemeliharaan, dengan dokumentasi yang menjelaskan setiap komponen utama dalam sistem. Selain itu, tersedia skrip tambahan untuk *deployment*, konfigurasi lingkungan, serta contoh penggunaan API untuk memudahkan integrasi dengan sistem lain. Dengan menyediakan akses terbuka ke kode sumber, MoMify diharapkan dapat terus berkembang dan berkontribusi dalam otomatisasi *Minutes of Meeting* yang lebih efisien.

[Halaman ini sengaja dikosongkan]

BAB 6 PENGUJIAN DAN EVALUASI

Bab ini menjelaskan tahap pengujian dan evaluasi terhadap sistem *MoMify*, yang dikembangkan untuk mengotomatiskan pembuatan *Minutes of Meeting* (MoM) dari rekaman audio atau video. Pengujian dilakukan untuk memastikan keakuratan transkripsi, efektivitas proses ringkasan menggunakan *GPT-4*, serta kesesuaian format keluaran dengan struktur MoM yang telah dirancang. Selain itu, evaluasi dilakukan untuk menilai performa sistem dalam berbagai skenario, termasuk pemrosesan file berdurasi panjang, penggunaan bahasa berbeda, serta respons terhadap berbagai jenis input. Hasil pengujian ini menjadi dasar untuk menyempurnakan sistem agar lebih optimal dan sesuai dengan kebutuhan pengguna.

6.1 Tujuan Pengujian

Pengujian dilakukan terhadap sistem *MoMify* untuk menguji keakuratan transkripsi, efektivitas proses ringkasan, serta kestabilan sistem dalam menangani berbagai format file audio dan video. Selain itu, pengujian bertujuan untuk memastikan bahwa arsitektur backend yang dikembangkan mampu menangani proses ekstraksi, transkripsi, dan pembuatan ringkasan secara efisien. Evaluasi juga dilakukan terhadap format keluaran *Minutes of Meeting* (MoM) dalam bentuk PDF untuk memastikan bahwa hasil yang dihasilkan sesuai dengan struktur yang telah dirancang serta dapat diakses dengan mudah oleh pengguna.

6.2 Lingkungan Pengujian

Untuk memastikan hasil pengujian, baik dari sisi fungsionalitas maupun kuantitatif, dilakukan serangkaian uji coba menggunakan perangkat dengan spesifikasi tertentu. Pengujian fungsionalitas bertujuan untuk memastikan bahwa

seluruh fitur aplikasi berjalan sesuai dengan kebutuhan pengguna, sedangkan evaluasi kuantitatif mencakup metrik seperti *Word Error Rate (WER)*, *Readability Score*, dan *Processing Time* untuk mengukur akurasi, keterbacaan, serta efisiensi waktu pemrosesan.

Pengujian dilakukan menggunakan perangkat dengan spesifikasi seperti yang dijelaskan pada Tabel 6.1 dan Tabel 6.2. Meskipun begitu, aplikasi *MoMify* bersifat fleksibel dan tetap dapat dijalankan pada perangkat keras dan lunak lain seperti sistem operasi macOS atau Linux. Namun, perlu diperhatikan bahwa beberapa konfigurasi ulang mungkin diperlukan, seperti pengaturan environment Python, dependensi library, serta penyesuaian sistem file atau path untuk mendukung kompatibilitas penuh dengan perangkat yang digunakan. Efisiensi pemrosesan juga dapat bervariasi tergantung pada performa sistem masing-masing.

6.2.1 Spesifikasi Perangkat Keras

Pengujian dilakukan menggunakan laptop dengan spesifikasi perangkat keras seperti yang ditunjukkan pada Tabel 6.1.

Tabel 6.1 Spesifikasi Perangkat Keras yang Digunakan dalam Pengujian Aplikasi MoMify

Komponen	Spesifikasi	Keterangan
Perangkat	Laptop Lenovo Legion Pro 5 16IRX9	Laptop performa tinggi untuk kebutuhan komputasi berat seperti pelatihan model NER.
Prosesor (CPU)	Intel Core i9-14900HX	Prosesor kelas atas yang mendukung pengolahan data skala besar dan komputasi intensif.
Memori (RAM)	32 GB, Dual Channel, 5600 MT/s	Memori besar untuk multitasking dan proses

		data yang berat, seperti pemrosesan NLP.
Penyimpanan	SSD 1TB	Penyimpanan cepat yang cukup untuk menyimpan dataset besar dan mendukung kecepatan akses.
Kartu Grafis (GPU)	NVIDIA GeForce RTX 4070 Laptop GPU, 8 GB	GPU dengan kemampuan akselerasi komputasi paralel untuk pelatihan model deep learning.
GPU Tambahan	Intel UHD Graphics	GPU terintegrasi untuk menangani beban grafis ringan.

6.2.2 Spesifikasi Perangkat Lunak

Aplikasi dikembangkan dan dijalankan dengan perangkat lunak pendukung seperti yang dijelaskan pada Tabel 6.2.

Tabel 6.2 Spesifikasi Perangkat Lunak yang Digunakan dalam Pengujian Aplikasi MoMify

Jenis	Nama Perangkat Lunak	Fungsi	Keterangan
Operating System (OS)	Windows 11	Sistem operasi utama untuk menjalankan semua perangkat lunak	Dipilih karena kompatibilitas tinggi dan dukungan luas untuk pengembangan
Integrated Development Environment (IDE)	PyCharm Professional	IDE utama untuk pengembangan backend dan integrasi sistem aplikasi.	Mendukung fitur lanjutan seperti smart debugging, auto-import, dan FastAPI integration.
Backend Framework	FastAPI	Menangani permintaan pengguna,	Ringan, cepat, dan cocok untuk RESTful API

		integrasi API, serta proses transkripsi dan ringkasan.	berbasis Python.
Frontend Framework	Streamlit	Menyediakan antarmuka interaktif berbasis web untuk pengguna akhir.	Mudah digunakan dan cocok untuk prototipe cepat berbasis Python.
API LLM	OpenAI GPT-4	Menghasilkan ringkasan <i>Minutes of Meeting</i> secara otomatis dari transkrip.	Model LLM canggih dengan kemampuan multi-bahasa dan konteks panjang.
API STT	OpenAI Whisper	Mengonversi audio/video ke teks secara akurat dan multibahasa.	Akurat untuk berbagai aksen dan mendukung <i>zero-shot inference</i> .

6.3 Kriteria Pengujian

Kriteria pengujian digunakan untuk menilai keberhasilan sistem *MoMify* dalam mengonversi rekaman rapat menjadi *Minutes of Meeting (MoM)* yang akurat dan terstruktur. Pengujian dilakukan untuk memastikan bahwa setiap komponen sistem berfungsi sesuai dengan tujuan pengembangannya. Kriteria yang digunakan dalam pengujian meliputi:

1. Kemampuan sistem untuk mengunggah file audio atau video dalam format yang didukung serta memastikan validasi ukuran dan jenis file berjalan dengan baik.

2. Akurasi transkripsi menggunakan Whisper API, yang diukur berdasarkan *Word Error Rate* (WER) dengan membandingkan hasil transkripsi terhadap teks referensi.
3. Efektivitas chunking dan tokenization dalam membagi teks transkripsi agar sesuai dengan batas token model GPT-4, sehingga tidak ada informasi penting yang hilang.
4. Kualitas ringkasan yang dihasilkan oleh GPT-4, dengan parameter evaluasi seperti *Brevity Penalty* dan *Readability Score* untuk menilai kesesuaian panjang dan keterbacaan teks.
5. Kecepatan pemrosesan sistem, yang diukur berdasarkan waktu yang dibutuhkan dari proses unggah file hingga hasil *MoM* dapat diunduh dalam format PDF.
6. Kemampuan sistem dalam menangani kesalahan, seperti kegagalan transkripsi, ringkasan yang tidak sesuai, atau kegagalan ekspor PDF, dengan notifikasi yang informatif kepada pengguna.

Pengujian berdasarkan kriteria ini bertujuan untuk memastikan bahwa sistem dapat menghasilkan Minutes of Meeting secara otomatis dengan akurasi tinggi, format yang jelas, serta pengalaman pengguna yang optimal.

6.4 Skenario Pengujian

Skenario pengujian dilakukan untuk memastikan bahwa sistem *MoMify* berfungsi sesuai dengan spesifikasi yang telah dirancang. Pengujian ini mencakup seluruh alur kerja sistem, mulai dari pengunggahan file, pemrosesan transkripsi, ringkasan, hingga ekspor hasil dalam format PDF. Pengujian dilakukan dengan mensimulasikan peran pengguna yang akan

menggunakan sistem untuk mengonversi rekaman rapat menjadi *Minutes of Meeting (MoM)*. Langkah-langkah skenario pengujian dilakukan sebagai berikut:

1. Pengguna membuka aplikasi MoMify dan mengakses halaman utama.
2. Pengguna mengunggah file audio atau video yang akan ditranskripsi melalui fitur *file uploader*.
3. Sistem melakukan validasi file, memastikan format dan ukuran sesuai dengan ketentuan.
4. Pengguna memilih pengaturan format dokumen, termasuk font, warna teks tebal, dan bahasa untuk MoM.
5. Pengguna menekan tombol “Process File Now” untuk memulai pemrosesan transkripsi dan ringkasan.
6. Sistem menampilkan progres pemrosesan dalam bentuk indikator bertahap, yang mencakup:
 - Pengunggahan file
 - Ekstraksi audio (jika file berbentuk video)
 - Transkripsi menggunakan Whisper API
 - Ringkasan menggunakan GPT-4
 - Pembuatan dokumen PDF
7. Setelah proses selesai, pengguna melihat ringkasan hasil MoM dan mengevaluasi apakah hasilnya sesuai dengan ekspektasi.
8. Pengguna menekan tombol “Download PDF” untuk mengunduh hasil akhir dalam format PDF.
9. Pengguna membuka file PDF yang telah diunduh dan memeriksa format serta struktur dokumen.
10. Jika terjadi kesalahan dalam pemrosesan, sistem menampilkan notifikasi error, seperti kegagalan transkripsi atau ringkasan yang tidak sesuai.

11. Pengguna dapat mengulangi proses dengan mengganti file atau menyesuaikan pengaturan jika hasil yang diperoleh belum optimal

6.5 Evaluasi Pengujian

Bagian ini membahas hasil pengujian sistem *MoMify* untuk memastikan bahwa fitur-fitur yang dikembangkan berfungsi sesuai dengan tujuan dan memenuhi standar kualitas yang diharapkan. Pengujian dilakukan di bawah supervisi langsung dari supervisor proyek, yang mengevaluasi sistem dalam dua aspek utama. Aspek pertama adalah evaluasi fungsionalitas, yang bertujuan untuk menilai apakah setiap fitur berjalan sesuai harapan. Aspek kedua adalah evaluasi kuantitatif, yang digunakan untuk mengukur kinerja sistem berdasarkan metrik tertentu. Hasil dari evaluasi ini menjadi dasar untuk perbaikan dan penyempurnaan lebih lanjut pada sistem *MoMify*.

6.5.1 Evaluasi Fungsionalitas

Evaluasi fungsionalitas dilakukan untuk memastikan bahwa setiap tahap dalam sistem, mulai dari pengunggahan file hingga ekspor ke PDF, berjalan dengan baik. Pengujian ini juga mencakup kemampuan sistem dalam menangani skenario kesalahan dan memberikan notifikasi kepada pengguna. Tabel 6.3 berikut menunjukkan hasil pengujian fungsionalitas *MoMify*.

Tabel 6.3 Hasil Evaluasi Fungsionalitas MoMify

Kriteria Pengujian	Hasil Pengujian
Sistem dapat mengunggah dan memproses file audio/video dengan format yang didukung	Terpenuhi

(MP3, WAV, MP4, M4A, WMA, AVI, MKV)	
Sistem dapat mendeteksi dan menolak file yang tidak sesuai format atau melebihi batas ukuran yang ditentukan (1GB)	Terpenuhi
Sistem dapat melakukan transkripsi dengan tingkat akurasi yang baik menggunakan model Whisper	Terpenuhi
Sistem dapat membagi teks menjadi <i>chunks</i> untuk pemrosesan yang efisien dalam GPT-4	Terpenuhi
Sistem dapat menghasilkan ringkasan yang terstruktur dengan format <i>Minutes of Meeting</i> (MoM) sesuai dengan standar yang telah ditentukan	Terpenuhi
Sistem dapat menyesuaikan format ringkasan MoM berdasarkan preferensi pengguna (font, warna teks, bahasa)	Terpenuhi
Sistem dapat mengekspor hasil MoM ke dalam format PDF tanpa gangguan atau kehilangan data	Terpenuhi
Sistem dapat memberikan notifikasi kepada pengguna jika terjadi kesalahan pada tahap pemrosesan (misalnya kegagalan transkripsi, chunking, atau ekspor)	Terpenuhi

6.5.2 Evaluasi Kuantitatif

Selain evaluasi fungsional, pengujian juga dilakukan secara kuantitatif dengan menggunakan metrik seperti Word Error Rate (WER) untuk mengukur akurasi transkripsi, Readability Score untuk menilai kemudahan membaca ringkasan, serta Processing Time untuk melihat efisiensi pemrosesan. Tabel 6.4 berikut menunjukkan hasil evaluasi kuantitatif berdasarkan metrik yang telah ditentukan.

Tabel 6.4 Hasil Evaluasi Kuantitatif MoMify

Format File	Durasi	Word Error Rate (WER) (%)	Brevity Penalty	Readability Score	Processing Time (Menit)	User Satisfaction (1-5)
MP3	15 Menit	4.5 - 6.3%	0.94 - 1.0	88 - 94	1.2 - 1.8	4.7 - 5.0
	2 Jam	5.5 - 8.0%	0.90 - 1.0	85 - 90	11 - 14	4.5 - 4.8
WAV	15 Menit	3.8 - 5.5%	0.96 - 1.0	89 - 95	1.5 - 2.2	4.8 - 5.0
	2 Jam	5.0 - 7.5%	0.92 - 1.0	86 - 91	12 - 16	4.6 - 4.9
MP4	15 Menit	6.5 - 8.8%	0.89 - 1.0	82 - 88	3.5 - 5.0	4.3 - 4.6
	2 Jam	7.5 - 10.5%	0.85 - 1.0	80 - 86	19 - 24	4.0 - 4.5
MPA	15 Menit	4.3 - 6.0%	0.93 - 1.0	87 - 93	1.6 - 2.3	4.7 - 5.0
	2 Jam	5.5 - 7.8%	0.91 - 1.0	84 - 89	10 - 14	4.5 - 4.8
WMA	15 Menit	4.5 - 6.5%	0.91 - 1.0	86 - 91	2.0 - 2.7	4.6 - 4.9
	2 Jam	5.8 - 8.2%	0.88 - 1.0	83 - 88	12 - 17	4.4 - 4.7
AVI	15 Menit	7.0 - 9.8%	0.85 - 1.0	79 - 86	4.8 - 6.5	3.9 - 4.3
	2 Jam	8.5 - 12.0%	0.80 - 1.0	77 - 83	22 - 28	3.5 - 4.0
MKV	15 Menit	7.5 - 10.2%	0.83 - 1.0	78 - 84	5.0 - 6.8	3.8 - 4.2
	2 Jam	9.0 - 13.0%	0.78 - 1.0	75 - 81	5.0 - 6.8	3.8 - 4.2

[Halaman ini sengaja dikosongkan]

BAB 7 KESIMPULAN DAN SARAN

7.1 Kesimpulan

Setelah melalui serangkaian proses mulai dari analisis kebutuhan, perancangan sistem, implementasi, hingga pengujian dan evaluasi, pengembangan sistem *MoMify* telah menghasilkan beberapa temuan utama yang mendukung tujuan dari proyek ini. Kesimpulan yang dapat diambil adalah sebagai berikut:

1. Sistem *MoMify* berhasil dikembangkan sebagai aplikasi otomatisasi pembuatan *Minutes of Meeting* (MoM), yang mampu mengunggah, memproses, dan meringkas transkripsi dari berbagai format audio dan video dengan tingkat akurasi yang baik.
2. Penggunaan *Whisper* untuk transkripsi dan GPT-4 untuk ringkasan memungkinkan sistem menghasilkan teks dengan struktur yang jelas, serta menjaga konteks pembicaraan dalam MoM yang dihasilkan.
3. Arsitektur berbasis FastAPI dan Streamlit memastikan komunikasi antara frontend dan backend berjalan dengan efisien, serta memungkinkan fleksibilitas dalam pengolahan file dan pembuatan ringkasan secara otomatis.
4. Hasil pengujian menunjukkan performa yang optimal dalam aspek akurasi transkripsi (*Word Error Rate* rendah), kualitas ringkasan (*Readability Score* tinggi), serta waktu pemrosesan yang cukup cepat untuk berbagai jenis file dan durasi yang diuji.
5. Sistem telah dilengkapi dengan fitur ekspor ke PDF dan penanganan error yang baik, sehingga memungkinkan

pengguna untuk menyimpan dan mendistribusikan hasil MoM dalam format yang lebih terstruktur serta mendapatkan notifikasi jika terjadi kesalahan selama proses.

Dengan keberhasilan implementasi ini, *MoMify* mampu mendukung otomatisasi dokumentasi rapat secara efektif, mempermudah pengguna dalam memperoleh ringkasan yang ringkas, akurat, dan siap digunakan untuk berbagai kebutuhan organisasi atau individu.

7.2 Saran

Berdasarkan hasil pengembangan dan pengujian sistem *MoMify*, terdapat beberapa saran yang dapat menjadi pertimbangan untuk pengembangan lebih lanjut guna meningkatkan fungsionalitas dan kualitas sistem, yaitu sebagai berikut:

1. Peningkatan Akurasi Transkripsi

Meskipun *Whisper* telah menunjukkan performa yang baik dalam transkripsi, penggunaan model yang lebih canggih atau kombinasi dengan teknik *noise reduction* dapat membantu meningkatkan akurasi, terutama dalam kondisi rekaman dengan latar belakang yang bising.

2. Optimasi Waktu Pemrosesan

Pengolahan file berdurasi panjang masih memerlukan waktu pemrosesan yang cukup signifikan. Implementasi teknik *parallel processing* atau optimasi pada tahap *chunking* dapat membantu mempercepat proses transkripsi dan ringkasan tanpa mengorbankan akurasi.

3. Integrasi dengan Platform Kolaborasi

Saat ini, sistem hanya menghasilkan *Minutes of Meeting (MoM)* dalam format *PDF*. Pengembangan lebih lanjut dapat mencakup integrasi dengan platform kolaborasi seperti *Google Docs*, *Microsoft OneNote*, atau sistem manajemen proyek (*Trello*, *Asana*) agar pengguna dapat langsung membagikan dan mengedit hasil ringkasan. Selain itu, integrasi dengan platform konferensi online seperti *Zoom* dan *Microsoft Teams* dapat memungkinkan sistem untuk secara otomatis merekam, mentranskripsi, dan meringkas rapat yang dilakukan secara daring, sehingga pengguna tidak perlu mengunggah file secara manual.

4. Dukungan untuk Berbagai Format Ekspor

Selain *PDF*, penambahan opsi ekspor ke format lain seperti *DOCX*, *TXT*, atau langsung ke *email* dapat memberikan fleksibilitas lebih bagi pengguna dalam menyimpan dan membagikan dokumen hasil rapat.

5. Peningkatan Fitur Kustomisasi

Saat ini, sistem telah menyediakan beberapa opsi kustomisasi, seperti pemilihan font, warna, dan bahasa. Namun, pengembangan lebih lanjut dapat mencakup fitur tambahan, seperti pengaturan format MoM yang lebih fleksibel sesuai kebutuhan pengguna, misalnya menyesuaikan struktur ringkasan atau menambahkan bagian tertentu seperti *key takeaways* atau *action plan tracking*.

6. Pembuatan Profil Pengguna dan Pengelolaan Riwayat MoM

Saat ini, sistem telah menyediakan beberapa opsi kustomisasi, seperti pemilihan font, warna, dan bahasa. Namun, pengembangan lebih lanjut dapat mencakup

fitur tambahan, seperti pengaturan format MoM yang lebih fleksibel sesuai kebutuhan pengguna, misalnya menyesuaikan struktur ringkasan atau menambahkan bagian tertentu seperti *key takeaways* atau *action plan tracking*.

Dengan saran-saran ini, diharapkan MoMify dapat terus berkembang menjadi sistem yang lebih efisien, fleksibel, dan bermanfaat bagi pengguna dalam otomatisasi dokumentasi rapat.

DAFTAR PUSTAKA

- Anuj Pandya, & Prof. Namrata Gawande. (2022). Automatic Generation of Minutes of Meetings. *International Journal of Scientific Research in Science, Engineering and Technology*, 93–99. <https://doi.org/10.32628/ijrsrset22928>
- Asthana, S., Hilleli, S., He, P., & Halfaker, A. (2023). *Summaries, Highlights, and Action items: Design, implementation and evaluation of an LLM-powered meeting recap system*. <http://arxiv.org/abs/2307.15793>
- Bera, M. (2021). Text to Speech Synthesis. In *International Journal of Innovative Science and Research Technology* (Vol. 6, Issue 2). www.ijisrt.com
- Chang, Y., Wang, X., Wang, J., Wu, Y., Yang, L., Zhu, K., Chen, H., Yi, X., Wang, C., Wang, Y., Ye, W., Zhang, Y., Chang, Y., Yu, P. S., Yang, Q., & Xie, X. (2024). A Survey on Evaluation of Large Language Models. *ACM Transactions on Intelligent Systems and Technology*, 15(3). <https://doi.org/10.1145/3641289>
- Chaurase, D., Dangra, P., & Chaube, A. (2024). EXPLORING AI-BASED TECHNIQUES FOR IMAGE PROCESSING USING STREAMLIT APPLICATION. *Gurukul International Multidisciplinary Research Journal*, 35–41. <https://doi.org/10.69758/gimrj2406i8v12p006>
- Chen, J. (2023). *Model Algorithm Research based on Python Fast API*.
- Elizarov, O. (2024). *Architecture of Applications Powered by Large Language Models*.
- Fadillah, A., Athahirah, N., & Lai, K. T. (2024). Chunking Strategy for Retrieval Augmented Generation in Regulation Documents. *11th IEEE International Conference on Consumer Electronics*

- *Taiwan, ICCE-Taiwan 2024*, 279–280.
<https://doi.org/10.1109/ICCE-Taiwan62264.2024.10674113>
- Gibson, D. R. (2022). Minutes of History: Talk and Its Written Incarnations. *Social Science History*, 46(3), 643–669.
<https://doi.org/10.1017/ssh.2022.4>
- Howard, J., & Gugger, S. (2020). Fastai: A layered api for deep learning. *Information (Switzerland)*, 11(2).
<https://doi.org/10.3390/info11020108>
- Journal, I. (2024). A Review on Speech-to-Text. *INTERANTIONAL JOURNAL OF SCIENTIFIC RESEARCH IN ENGINEERING AND MANAGEMENT*, 08(03), 1–13.
<https://doi.org/10.55041/ijrsrem29004>
- Katz, D. M., Bommarito, M. J., Gao, S., & Arredondo, P. (2024). GPT-4 passes the bar exam. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 382(2270). <https://doi.org/10.1098/rsta.2023.0254>
- Khadim, U., Munwar Iqbal, M., & Awais Azam, M. (2022). A Secure Digital Text Watermarking Algorithm for Portable Document Format (PDF). *Mehran University Research Journal of Engineering and Technology*, 41(1), 100–110.
<https://doi.org/10.22581/muet1982.2201.10>
- Kharitonov, E., Vincent, D., Borsos, Z., Marinier, R., Girgin, S., Pietquin, O., Sharifi, M., Tagliasacchi, M., & Zeghidour, N. (2023). *Speak, Read and Prompt: High-Fidelity Text-to-Speech with Minimal Supervision*. <https://doi.org/10.1162/tacl>
- Liu, Y., Han, T., Ma, S., Zhang, J., Yang, Y., Tian, J., He, H., Li, A., He, M., Liu, Z., Wu, Z., Zhao, L., Zhu, D., Li, X., Qiang, N., Shen, D., Liu, T., & Ge, B. (2023). *Summary of ChatGPT-Related Research and Perspective Towards the Future of Large Language Models*.
<https://doi.org/10.1016/j.metrad.2023.100017>

- Mavroudis, V. (2024). *LangChain*.
- Miura, H., Takegawa, Y., Terai, A., & Hirata, K. (2020). Interactive minutes generation system based on hierarchical discussion structure. *Web Intelligence*, 18(1), 85–95.
<https://doi.org/10.3233/WEB-200430>
- Moiz, M. (2024). *USING A PYTHON SCRIPT TO CAPTURE AND SUMMARIZE INSTAGRAM FEEDS AT RAUNER SPECIAL COLLECTIONS USING A PYTHON SCRIPT TO CAPTURE AND SUMMARIZE INSTAGRAM FEEDS AT RAUNER SPECIAL COLLECTIONS PUBLIC WORKFLOW*.
<https://digitalcommons.dartmouth.edu/rauner-special-collections-digital-archives-practice/4>
- Neville, K., & Re, S. (2019). Guidelines for Creating Meaningful Meeting Minutes. *Teaching and Learning in Nursing*, 14(4), 235–237. <https://doi.org/10.1016/J.TELN.2019.05.002>
- OpenAI. (2023). *GPT-4 Technical Report*.
- Radford, A., Kim, J. W., Xu, T., Brockman, G., Mcleavy, C., & Sutskever, I. (2024). *Robust Speech Recognition via Large-Scale Weak Supervision*. <https://github.com/openai/>
- Raj, R., & Kos, A. (2022). *Mixed Design of Integrated Circuits and Systems-MIXDES 2022 Mixed Design of Integrated Circuits and Systems-MIXDES 2022 A Comprehensive Study of Optical Character Recognition*.
- Reshini, S., Adarsh, N. B., Subhash, N., Kumar Reddy, S. U., Kumar Reddy, R. S., & Sai, D. H. (2024). Effective Data Analysis Assistance with AI and Streamlit Integration with LangChain. *2024 4th International Conference on Ubiquitous Computing and Intelligent Information Systems (ICUIS)*, 27–32.
<https://doi.org/10.1109/ICUIS64676.2024.10867196>
- Rosenberg, J. (2024). Automated Transcription of Interviews in Qualitative Research Using Artificial Intelligence A Simple

- Guide. *Journal of Surgery Research and Practice*, 1–6.
<https://doi.org/10.46889/jsrp.2024/5204>
- Singh, I. S., Aggarwal, R., Allahverdiyev, I., Taha, M., Akalin, A., Zhu, K., & O'Brien, S. (2024). *ChunkRAG: Novel LLM-Chunk Filtering Method for RAG Systems*.
<http://arxiv.org/abs/2410.19572>
- Topsakal, O., & Akinci, T. C. (2023). *Creating Large Language Model Applications Utilizing LangChain: A Primer on Developing LLM Apps Fast*. <http://as-proceeding.com/:Konya,Turkeyhttps://www.icaens.com/>
- Vikrant Singh. (2024, January 6). *Chunking for Enhanced LLM Applications*. Medium.
- Wang, S., Yang, C. H., Wu, J., & Zhang, C. (2024). CAN WHISPER PERFORM SPEECH-BASED IN-CONTEXT LEARNING? *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, 13421–13425.
<https://doi.org/10.1109/ICASSP48485.2024.10446502>
- Yang, J., Jin, H., Tang, R., Han, X., Feng, Q., Jiang, H., Zhong, S., Yin, B., & Hu, X. (2024). Harnessing the Power of LLMs in Practice: A Survey on ChatGPT and Beyond. *ACM Transactions on Knowledge Discovery from Data*, 18(6).
<https://doi.org/10.1145/3649506>

BIODATA PENULIS

Nama : Rayssa Ravelia
Tempat, Tanggal Lahir : Jakarta, 12 Januari 2003
Jenis Kelamin : Perempuan
Telepon : +62 83893852003
Email : 5025211219@student.its.ac.id

AKADEMIS

Kuliah : Departemen Teknik Informatika –
FTEIC , ITS
Angkatan : 2021
Semester : 8 (delapan)