Pengembangan Fitur *Speech Recognition* pada Aplikasi Notulensi Menggunakan Whisper AI



Disusun Oleh:

N a m a : Raihan Rifaldi
NIM : 20523030

PROGRAM STUDI INFORMATIKA – PROGRAM SARJANA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA
2025

HALAMAN PENGESAHAN DOSEN PEMBIMBING

Pengembangan Fitur *Speech Recognition* pada Aplikasi Notulensi Menggunakan Whisper AI

TUGAS AKHIR JALUR MAGANG

Disusun Oleh:

N a m a : Raihan Rifaldi

NIM : 20523030

Yogyakarta, 22 Mei 2025

Pembimbing,



(Andhik Budi Cahyono, S.T., M.T.)

HALAMAN PENGESAHAN DOSEN PENGUJI

Pengembangan Fitur *Speech Recognition* pada Aplikasi Notulensi Menggunakan Whisper AI

TUGAS AKHIR JALUR MAGANG

Telah dipertahankan di depan sidang penguji sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer dari Program Studi Informatika – Program Sarjana di Fakultas Teknologi Industri Universitas Islam Indonesia

Yogyakarta, 22 Mei 2025

Tim Penguji

Andhik Budi Cahyono, S.T., M.T. _____

Anggota 1

Dr. Ahmad Luthfi, S.Kom., M.Kom. _____

Anggota 2

Dhomas Hatta Fudholi, S.T., M.Eng., Ph.D. _____

Mengetahui,

Ketua Program Studi Informatika – Program Sarjana

Fakultas Teknologi Industri

Universitas Islam Indonesia

(Dhomas Hatta Fudholi, S.T., M.Eng., Ph.D.)

HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan di bawah ini:

Nama	:	Raihan Rifaldi
NIM	:	20523030

Tugas akhir dengan judul:

Pengembangan Fitur *Speech Recognition* pada Aplikasi Notulensi Menggunakan Whisper AI

Menyatakan bahwa seluruh komponen dan isi dalam tugas akhir ini adalah hasil karya saya sendiri. Apabila di kemudian hari terbukti ada beberapa bagian dari karya ini adalah bukan hasil karya sendiri, tugas akhir yang diajukan sebagai hasil karya sendiri ini siap ditarik kembali dan siap menanggung risiko dan konsekuensi apapun.

Demikian surat pernyataan ini dibuat, semoga dapat dipergunakan sebagaimana mestinya.

Yogyakarta, 22 Mei 2025



(Raihan Rifaldi)

HALAMAN PERSEMBAHAN

Puji syukur kepada Allah SWT atas rahmat, hidayah dan petunjuk-Nya sehingga saya dapat menyelesaikan tugas akhir ini. Tugas akhir ini yang berjudul "Pengembangan Fitur *Speech Recognition* pada Aplikasi Notulensi Menggunakan Whisper AI" sebagai salah satu syarat untuk lulus dari perkuliahan. Laporan ini jauh dari kata sempurna, akan tetapi penulis sangat bangga telah mencapai titik ini. Semoga laporan ini dapat membantu orang sebagai bahan pengembangan serta menjadikannya sebagai referensi ataupun panduan.

Skripsi atau tugas akhir ini saya persembahkan untuk:

- Ayah dan ibu, terimakasih atas dukungan, doa nasehat serta kasih sayang yang telah diberikan tanpa henti sampai saat ini.
- Keluarga tersayang, yang merupakan pendukung setia, tempat cerita serta dalam menyelesaikan tugas akhir ini.
- Delia, yang telah membantu dalam membantu mendukung dan menemani dalam pengerjaan tugas akhir ini
- Kepada semua teman-teman, saudara yang tidak bisa saya sebutkan satu per satu, yang telah berjuang bersama sejauh ini.

HALAMAN MOTO

"Seungguhnya Allah tidak akan mengubah keadaan suatu kaum, sebelum mereka mengubah keadaan diri mereka sendiri." (QS Ar -Rad 11)

"Di mana pun engkau berada selalulah menjadi yang terbaik dan berikan yang terbaik dari yang bisa kita berikan." – BJ Habibie

“Sometimes we have to go deep inside ourselves to solve our problems” – Patrick Star

KATA PENGANTAR

Alhamdulillah segala puji dituturkan kepada Allah SWT atas berkah dan rahmat yang telah diberikan beserta izin-Nya sehingga proses magang, kuliah dan penulisan laporan tengah ini dapat berjalan dengan baik.

Laporan magang ini merupakan bentuk pertanggungjawaban atas kegiatan magang yang telah dilaksanakan oleh penulis selama hampir 6 (Enam) bulan di PT Telekomunikasi Selular. kelancaran kegiatan magang tidak terlepas dari bantuan berbagai pihak yang telah mendukung proses ini baik secara langsung maupun tidak langsung. Oleh karena, itu penulis ingin mengucapkan terimakasih kepada segenap pihak yang telah membantu:

1. Allah SWT yang telah memberikan kehidupan dan berkah nikmat yang tak terhingga.
2. Kedua orang tua yang telah memberikan dukungan dan doa untuk kesuksesan penulis
3. Bapak Dr. Raden Teduh Dirgahayu, S.T., M.Sc. selaku Ketua Jurusan Informatika Fakultas Teknologi Industri Universitas Islam Indonesia.
4. Bapak Dhomas Hatta Fudholi, S.T., M.Eng., Ph.D. selaku Ketua Program Studi Informatika-Program Sarjana Fakultas Teknologi Industri Universitas Islam Indonesia.
5. Bapak Andhik Budi Cahyono, S.T., M.T selaku Dosen Pembimbing Magang.
6. Bapak Syaiful H Hasibuan, Ibu Ina R Nasihati, Defri Mulyadi, Hanny D Asmoro sebagai mentor IT Development yang selalu memberi masukan dan ilmu dalam membangun project ini.
7. Seluruh Karyawan IT dan Staff Telkomsel yang selalu menjadi tempat bertanya dan memberikan support selama melaksanakan magang.

Atas bantuan dari berbagai pihak tersebut, penulis dapat menyelesaikan laporan ini dengan baik. Dengan disusunnya laporan ini diharapkan dapat memberikan suatu gambaran terhadap kegiatan magang yang berlangsung di PT Telekomunikasi Selular. Namun, penulis sangat menyadari jika terdapat kekurangan atau kesalahan pada laporan ini karena keterbatasan pengalaman yang penulis miliki.

Oleh karena itu, penulis sangat mengharapkan kritik dan saran yang bersifat membangun dari para pembaca. Akhir kata, penulis berharap kepada Allah SWT semoga apa yang telah ia berikan dengan keikhlasannya akan menjadi pahala dan amal serta mendapat kemudahan dari Allah SWT.

Yogyakarta, 22 Mei 2025



(Raihan Rifaldi)

SARI

Rangkuman rapat perusahaan sangat membantu memastikan bahwa semua peserta mendapatkan hasil rapat sesegera mungkin setelah rapat selesai. Namun, proses penulisan rangkuman masih dilakukan secara manual sehingga menyebabkan keterlambatan ketika mendistribusikan hasil ke peserta. Untuk mengatasi masalah tersebut, dikembangkan SPERCO, sebuah aplikasi berbasis *speech recognition* menggunakan Whisper Model dari OpenAI. SPERCO dirancang untuk mencatat rangkuman secara otomatis dengan fitur *Speaker Diarization* yang dapat mengidentifikasi berbagai pembicara. Teknologi ini diimplementasi pada Hugging face yang sangat membantu dalam menyediakan PTM (*Pre-Trained Models*). Hasil dari proyek ini dapat mengurangi kesalahan informasi penulisan rangkuman dari rapat perusahaan, sehingga dapat membantu perusahaan unutk evaluasi kedepannya. Proyek ini dikerjakan sebagai bagian dari tugas magang di PT Telekomunikasi Selular.

Kata kunci: *Speech Recognition*, Whisper AI, *Speaker Diarization*, Notulensi.

GLOSARIUM

Hugging Face	Menyediakan platform untuk berbagi model bahasa yang dapat digunakan dalam aplikasi berbasis AI.
NLP	Digunakan untuk membuat *chatbot* yang dapat merespons pertanyaan pelanggan dengan cara yang alami.
Speaker Diarization	Proses mengidentifikasi dan memisahkan pembicara yang berbeda dalam rekaman audio, memungkinkan sistem untuk mengetahui "siapa yang berbicara kapan".
Speech Recognition	Teknologi yang memungkinkan komputer untuk mengenali dan menginterpretasikan ucapan manusia menjadi teks atau perintah yang dapat dipahami
Visual Studio Code	Pengembang untuk membuat dan menguji aplikasi berbasis Windows.
Website	Sekumpulan halaman web yang terhubung dan dapat diakses melalui internet.
Whisper AI	Model kecerdasan buatan yang dikembangkan oleh OpenAI untuk pengenalan suara.

DAFTAR ISI

HALAMAN PENGESAHAN DOSEN PEMBIMBING ii
HALAMAN PENGESAHAN DOSEN PENGUJI iii
HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR iv
HALAMAN PERSEMBAHAN v
HALAMAN MOTO vi
KATA PENGANTAR vii
SARI ix
GLOSARIUM x
DAFTAR ISI xi
DAFTAR TABEL xiii
DAFTAR GAMBAR xiv
BAB I PENDAHULUAN 1
1.1 Latar Belakang 1
1.2 Tujuan 2
1.3 Manfaat 2
1.4 Sistematika Penulisan 3
BAB II LANDASAN TEORI 4
2.1 Speech Recognition 4
2.2 Hugging Face 6
2.3 Natural language processing 7
2.4 Whisper AI 8
2.5 Tinjauan Pustaka 9
BAB III PELAKSANAAN MAGANG 12
3.1 Aktivitas Magang 12
3.1.1 Pembelajaran Mandiri 14
3.1.2 Fixing Bug Aplikasi SPERCO 15
3.1.3 Persiapan *Tools pendukung* Pembuatan Aplikasi 22
3.2 Implementasi dan Pengujian Aplikasi 29
3.2.1 Analisis Aktivitas melalui *Log Container* 34
3.2.2 Hasil Pengujian Akurasi Transkripsi 34

3.3 Pembuatan Dokumentasi & Penutupan Proyek 40
BAB IV REFLEKSI PELAKSANAAN MAGANG 41
4.1 Relevansi Akademik 41
4.2 Pembelajaran Magang 42
4.2.1 Teknis 44
4.2.2 Non Teknis 45
BAB V PENUTUP 46
5.1 Kesimpulan 46
5.2 Saran 46
DAFTAR PUSTAKA 48
LAMPIRAN 50

DAFTAR TABEL

Tabel 3. 1 Aktivitas Magang 12
Tabel 3. 2 *Whisper Models details* 23
Tabel 3. 3 Pengujian akurasi menggunakan Metode *Ground Truth* 37

DAFTAR GAMBAR

Gambar 2. 1 Proses basis ASR 4
Gambar 2. 2 Contoh penggunaan Pyannote Audio 5
Gambar 2. 3 Alur kerja dari Hugging Face 6
Gambar 2. 4 Alur kerja dari Whisper 8
Gambar 3. 1 Pembelajaran secara mandiri 14
Gambar 3. 2 Pelatihan di Datacamp Academy 15
Gambar 3. 3 Dokumentasi *error* dan *solving* pada Notion 16
Gambar 3. 4 Platform Notion untuk laporan 17
Gambar 3. 5 Rapat Rutin Proyek 18
Gambar 3. 6 Metode Scrum 19
Gambar 3. 7 Webinar bulanan 21
Gambar 3. 8 Kutipan kode menggunakan *Whisper Model* 23
Gambar 3. 9 Tampilan SPERCO di Hugging Face 24
Gambar 3. 10 Halaman pembuatan *Workspace* di Hugging Face 26
Gambar 3. 11 *Remote* Hugging Face dari Visual Studio Code 27
Gambar 3. 12 Tabel *research* pendukung SPERCO 28
Gambar 3. 13 Implementasi SPERCO di Hugging Face 30
Gambar 3. 14 *Video podcast* Youtube 31
Gambar 3. 15 *Whisper Models* 31
Gambar 3. 16 Pemilihan jumlah pembicara 32
Gambar 3. 17 Data dialog pembicara 33
Gambar 3. 18 *Log Container* dari SPERCO 34

BAB I
PENDAHULUAN

1.1 Latar Belakang

Teknologi pengenalan suara (*speech recognition*) telah berkembang pesat dan memberikan dampak signifikan pada berbagai bidang, termasuk untuk perusahaan. Salah satu penerapan teknologi pengenalan suara adalah pembuatan notula rapat. Proses pencatatan manual yang dilakukan selama rapat sering kali memakan waktu lama dan rentan terhadap kesalahan manusia. Hal ini dapat menyebabkan ketidaksesuaian antara hasil rapat dan laporan yang dihasilkan, yang pada akhirnya dapat berdampak negatif terhadap pengambilan keputusan (Rogers et al., 2019).

PT Telekomunikasi Selular (Telkomsel), sebagai salah satu perusahaan telekomunikasi terbesar di Indonesia, secara rutin mengadakan rapat dalam jumlah besar setiap harinya. Dalam proses bisnisnya, pencatatan berita acara rapat dilakukan secara manual oleh notulis yang harus mendengarkan, mencatat, kemudian mengetik ulang isi percakapan rapat. Metode ini menimbulkan tantangan berupa beban kerja yang tinggi, waktu yang lama untuk menghasilkan laporan, serta potensi kesalahan pencatatan akibat *human error*. Sehingga, Telkomsel memerlukan solusi teknologi yang mampu mengotomatisasi proses pencatatan notula secara efektif. Untuk menjawab tantangan tersebut, Telkomsel mengembangkan aplikasi berbasis teknologi pengenalan suara bernama SPERCO (*Speech Recognition*). SPERCO dirancang khusus untuk membantu notulis dalam merekam dan mentranskripsikan percakapan rapat secara otomatis. Dengan SPERCO, notulis memiliki kemudahan dalam mengedit transkripsi yang dihasilkan, seperti menambahkan atau menghapus kata, menyesuaikan hasil pencatatan dengan konteks rapat.

Tujuan pengembangan SPERCO yang lain diantaranya: meningkatkan efisiensi pencatatan notulensi, mengurangi kesalahan transkripsi serta mempermudah proses penyusunan laporan rapat. Selain itu, dengan meningkatnya adopsi teknologi *speech recognition* di berbagai sektor, pengembangan SPERCO diharapkan dapat memberikan kontribusi signifikan dalam meningkatkan produktivitas karyawan dan mengurangi waktu yang dihabiskan dalam proses pembuatan notula.

Namun demikian, implementasi teknologi pengenalan suara menghadapi beberapa tantangan, di antaranya kualitas suara yang bervariasi, perbedaan aksen pembicara, serta gangguan dari kebisingan latar belakang. Untuk mengatasi tantangan tersebut, Telkomsel

memerlukan teknologi pengenalan suara yang canggih dan mampu memberikan hasil transkripsi yang akurat meskipun dalam kondisi audio yang kurang ideal.

Whisper AI, teknologi yang dikembangkan dengan kemampuan tinggi dalam mengenali suara, hadir sebagai solusi potensial. Whisper AI mampu mengelola berbagai bahasa, intonasi, dan aksen dengan tingkat akurasi tinggi, menjadikannya relevan untuk diterapkan pada proyek SPERCO. Integrasi teknologi Whisper AI ke dalam SPERCO bertujuan agar aplikasi mampu memproses audio secara langsung dengan presisi tinggi, mencatat percakapan atau diskusi secara otomatis, serta memastikan tidak ada detail penting yang terlewatkan. Dengan fitur unggulan yang dimiliki Whisper AI, diharapkan dapat secara signifikan mengurangi kendala pencatatan manual yang selama ini dihadapi oleh Telkomsel, serta mendukung proses bisnis internal perusahaan secara lebih efektif dan efisien.

1.2 Tujuan

Tujuan yang ingin dicapai dalam ruang lingkup ini yaitu, mengimplementasikan teknologi Whisper AI pada aplikasi SPERCO untuk membantu mentranskripsikan audio ke teks dengan tingkat akurasi tinggi. Serta melakukan pengujian teknologi *Speech Recognition* dan fitur *Speaker Diarization* pada *Whisper Model* pada Huggingface dengan melakukan *testing* di Video Podcast Youtube serta menggunakan pengujian menggunakan metode *Ground Truth* untuk mengetahui sejauh mana teknologi dapat menjangkau informasi yang diberikan dari pembicara.

1.3 Manfaat

Manfaat yang dapat diperoleh dari ruang lingkup laporan tugas akhir ini yaitu:

a. Hasil Pengujian: Untuk mengetahui akurasi kata dari teknologi SPERCO menggunakan metode *Ground Truth*.
b. Kenaikan Produktivitas: SPERCO digunakan sebagai aplikasi internal Telkomsel yang dapat membantu notulis membuat rangkuman rapat sehingga evaluasi perusahaan dapat dilakukan secepatnya.
c. *Speaker Diarization*: Mengetahui hasil kualitas dari fitur *Speaker Diarization* yang dapat mengidentifikasi setiap pembicara dalam suatu diskusi, sehingga dapat menghasilkan transkripsi yang lebih tepat dan meminimalisir potensi kesalahan.

1.4 Sistematika Penulisan

Susunan sistematika penulisan dalam laporan ini sebagai berikut:

a. BAB I : Pendahuluan

Bab ini menjelaskan tentang pembahasan secara umum yang meliputi latar belakang, ruang lingkup magang, tujuan dan manfaat serta sistematika penulisan.

b. BAB II : Landasan Teori dan Tinjauan Pusaka

Bab ini berisi teori – teori pembahasan dari proyek sebelumnya yang menjadi landasan dan juga pendukung atas penugasan yang dikerjakan selama mengikuti kegiatan magang.

c. BAB III : Pelaksanaan Magang

Bab ini berisi uraian dari tugas dan aktivitas yang dilakukan ketika pelaksanaan magang.

d. BAB IV : Refleksi Pelaksanaan Magang

Bab ini berisi penjelasan mengenai relevansi akademik dan pembelajaran yang didapatkan selama melaksanakan kegiatan magang.

e. BAB V: Penutup

Bab ini berisi kesimpulan dan saran dari pembahasan yang telah disampaikan pada bab-bab sebelumnya.

BAB II
LANDASAN TEORI

2.1 Speech Recognition

Dalam beberapa tahun belakangan, teknologi pengenalan ucapan otomatis atau *Automatic Speech Recognition* (ASR) telah mengalami perkembangan pesat yang dipicu oleh penerapan teknik pembelajaran mendalam. Di sisi lain, tersedianya kumpulan data suara yang dapat diakses secara luas juga menjadi faktor yang mendorong percepatan inovasi di bidang ini. Sistem ASR dirancang untuk menerima masukan berupa suara atau rekaman audio, lalu mengubahnya menjadi teks dalam bahasa yang sesuai. Idealnya, sistem ini mampu memahami dan mengenali ujaran secara akurat serta memanfaatkan hasil pengenalan tersebut untuk menjalankan berbagai fungsi lanjutan.

Pengembangan teknologi ASR menghadapi sejumlah hambatan. Salah satu tantangan utama adalah keberagaman suara manusia, termasuk aksen dan pola bicara yang kompleks, terutama pada pengguna bilingual atau multibahasa. Selain itu, keterbatasan sumber daya untuk melatih model ASR menjadi tantangan lain, karena hanya segelintir bahasa dari ribuan bahasa di dunia yang memperoleh dukungan data yang memadai. Berbagai penelitian telah mengulas isu-isu ini, seperti jumlah kosakata yang dapat dikenali dan kualitas sinyal audio. Studi ini bertujuan untuk mengevaluasi metode ekstraksi fitur, teknik klasifikasi terkini serta penggunaan perangkat daring dan basis data yang relevan, demi meningkatkan akurasi dan efektivitas model bahasa (Malik et al., 2021). Untuk memahami cara kerja dasar sistem ASR, berikut ini ditampilkan bagan alur proses yang umumnya digunakan

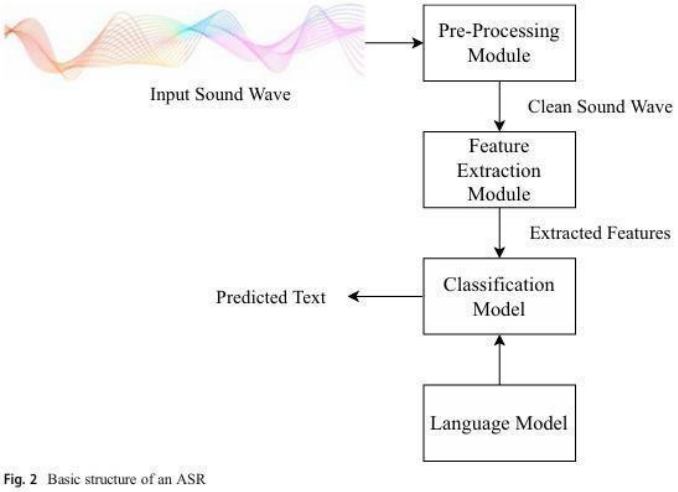


Fig. 2 Basic structure of an ASR

Gambar 2. 1 Proses basis ASR

Sumber: (Malik et al., 2021)

Gambar 2. 1 Proses basis ASR menggambarkan proses utama dalam sistem ASR, yang dimulai dari masukan berupa gelombang suara (*input sound wave*). Gelombang ini pertama-tama diproses melalui modul pra-pemrosesan (*pre-processing module*) untuk membersihkan sinyal dari gangguan atau *noise*. Setelah itu, *feature extraction* module berperan untuk mengambil ciri-ciri penting dari suara yang telah dibersihkan. Ciri-ciri ini kemudian dimasukkan ke dalam classification model, yang menghasilkan teks prediksi berdasarkan pola-pola suara yang dikenali. Hasil prediksi ini disempurnakan oleh language model, yang memastikan bahwa teks akhir sesuai dengan struktur bahasa yang benar dan konteks yang tepat.

2.1.1 Speaker Diarization

Speaker Diarization adalah teknik yang mengenali ciri-ciri suara masing-masing individu, pemisahan sinyal audio asli serta pengelompokan berdasarkan karakteristik khas setiap pembicara. *Speaker Diarization* tidak hanya mampu membedakan antara berbagai individu tanpa memerlukan data awal tentang mereka, tetapi juga dapat diterapkan dalam berbagai situasi nyata, seperti siaran berita, diskusi konferensi, interaksi di media sosial dan dalam konteks persidangan. Selain itu, *diarization* berkontribusi dalam meningkatkan ketepatan pengenalan suara dalam lingkungan yang melibatkan banyak pembicara (Wang & Purushotam, 2023).

Pada proyek ini penulis menggunakan pustaka untuk pada fitur *Speaker Diarization* adalah *pyannote.audio* yang menggunakan Python sebagai bahasa pemograman yang berbasis *framework* PyTorch pada proyek ini. Berikut kode penggunaan Pyannote pada Gambar 2. 2 Contoh penggunaan Pyannote Audio.

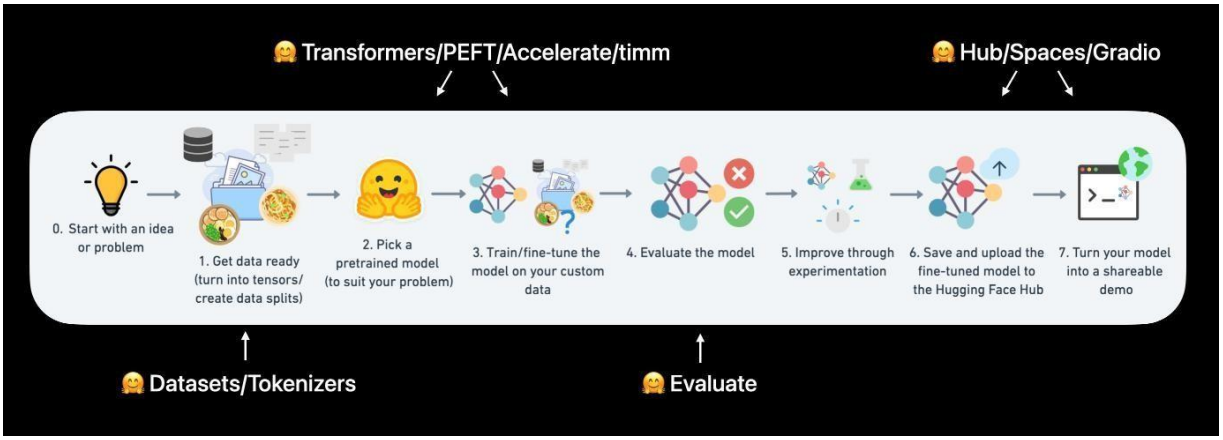
```
import torch
# embedding dari suara pembicara untuk verifikasi pembicara
import pyannote.audio
from pyannote.audio.pipelines.speaker_verification import
PretrainedSpeakerEmbedding
# memuat file suara
from pyannote.audio import Audio
from pyannote.core import Segment
```

Gambar 2. 2 Contoh penggunaan Pyannote Audio

Gambar 2. 2 Contoh penggunaan Pyannote Audiomemperlihatkan potongan kode Python yang digunakan untuk melakukan embedding suara pembicara sebagai bagian dari proses verifikasi pembicara. Pustaka *pyannote.audio* memungkinkan pemuatan file audio, segmentasi suara menggunakan Segment, serta penggunaan *PretrainedSpeakerEmbedding* untuk mengekstrak representasi vektor dari suara. Modul-modul tersebut berperan penting dalam proses *diarization* karena membantu sistem mengenali dan memisahkan bagian ucapan berdasarkan identitas suara masing-masing pembicara.

2.2 Hugging Face

Hugging Face merupakan platform yang menyediakan PTM (*Pre-Trained Models*) untuk NLP (*Natural Language Processing*) dan pemrosesan teks. Platform ini memungkinkan peneliti dan pengembang untuk dengan mudah mengakses dan berbagi model-model AI (*Artificial Intelligence*), khususnya dalam bidang NLP. Hugging Face menjadi SPR (*Software Package Registry*) terkemuka untuk PTM, memfasilitasi penggunaan kembali model *open-source* dalam pengembangan sistem AI (Jones et al., 2024). Untuk mempermudah pemahaman mengenai proses kerja di dalam ekosistem Hugging Face, Gambar 2. 3 berikut menyajikan alur kerja terintegrasi mulai dari ide hingga implementasi model dalam bentuk demo interaktif.



Gambar 2. 3 Alur kerja dari Hugging Face

Sumber: learnhuggingface.com

Dari Gambar 2. 3 Alur kerja dari Hugging Face memiliki Face *ecosystem*, proses ini dimulai dengan merumuskan ide atau permasalahan yang ingin dipecahkan. Langkah berikutnya adalah menyiapkan data, termasuk mengonversinya ke dalam format tensor dan membagi dataset menjadi bagian pelatihan, validasi serta pengujian menggunakan alat seperti Datasets dan Tokenizers. Setelah data siap, model pra-latih yang sesuai dipilih dari Hugging Face Hub untuk diadaptasikan dengan kebutuhan spesifik. Model tersebut kemudian dilatih atau disesuaikan

lebih lanjut (*fine-tuning*) menggunakan data khusus dengan memanfaatkan alat seperti *Transformers*, PEFT, *Accelerate* atau timm. Setelah proses pelatihan selesai, model diuji performanya menggunakan data validasi untuk memastikan kualitasnya, dengan bantuan alat seperti Evaluate. Jika hasilnya belum optimal, eksperimen dilakukan untuk meningkatkan kinerjanya, misalnya dengan menyesuaikan parameter atau memperluas dataset. Model yang telah disempurnakan dapat diunggah ke Hugging Face Hub agar tersedia bagi pengguna lain. Sebagai langkah akhir, model tersebut dapat dikemas menjadi demo interaktif yang mudah dibagikan menggunakan alat seperti *Gradio* atau *Spaces*. Alur ini memberikan panduan terstruktur untuk membangun solusi berbasis kecerdasan buatan secara efektif.

Platform ini menyediakan berbagai solusi untuk proyek dan implementasi model AI, termasuk *Neuro Linguistic Programming* (NLP). Salah satu layanan intinya adalah *Transformers*, sebuah pustaka *open-source* yang menyediakan antarmuka untuk bekerja dengan berbagai model pembelajaran mesin canggih, seperti GPT, BERT dan lainnya. Selain itu, Hugging Face menawarkan *datasets*, yaitu pustaka yang memfasilitasi akses ke berbagai kumpulan data yang dirancang untuk pelatihan model AI, mempermudah pengolahan data dalam skala besar dan kompatibel dengan model AI populer. Spaces adalah platform yang memungkinkan pengguna meng-*host* aplikasi AI berbasis model *machine learning* secara interaktif, baik untuk prototipe maupun demonstrasi. Teknologi yang digunakan di dalam ekosistem ini melibatkan bahasa pemrograman Python, pustaka *deep learning* seperti PyTorch dan TensorFlow serta *framework* web untuk *Spaces* seperti Gradio dan Streamlit. Di samping itu, layanan *cloud* Hugging Face memungkinkan integrasi yang lancar dengan platform *cloud* lainnya untuk pengolahan model dan data skala besar, mendukung berbagai kebutuhan pengembangan AI dari *prototyping* hingga produksi.

2.3 Natural language processing

Natural Language Processing (NLP) merupakan tekonolgi yang berperan di bidang Kesehatan yang bertujuan untuk menggabungkan teknologi NLP dan *Speech Recognition* serta meningkatkan produktivitas radiolog dengan mempercepat proses pelaporan serta memfasilitasi analisis data yang lebih baik untuk proyek dan peningkatan kualitas pelayanan kesehatan (Jorg et al., 2023).

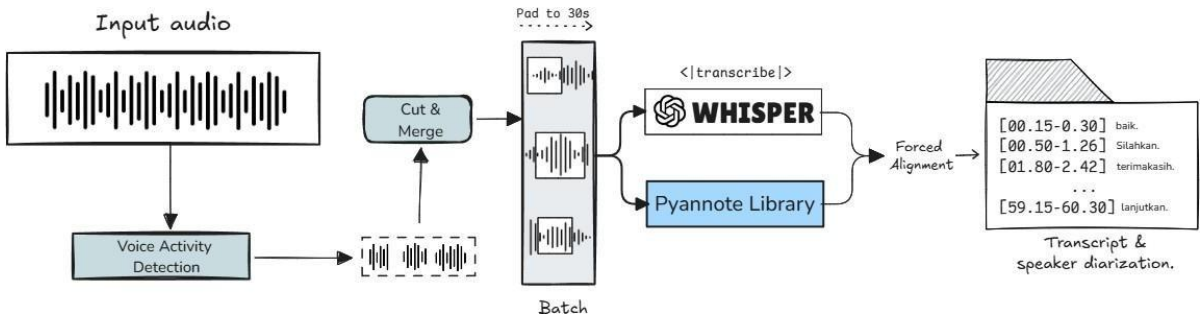
Dalam proses *summarization*, NLP berperan penting dengan cara:

1. Mengekstrak informasi kunci dari teks hasil *Speech Recognition.*
2. Menstrukturisasi data yang diekstrak ke dalam format yang mudah dibaca.

3. Menghasilkan ringkasan singkat namun komprehensif dari temuan radiologi.

2.4 Whisper AI

Whisper merupakan salah satu model *Speech Recognition* yang powerful yang dikembangkan oleh OpenAI. Untuk mengadaptasi Whisper ke ASR *Workflow* dari Hugging Face *ecosystem* pada ucapan *multi-speaker* yang tumpang tindih, diusulkan metode *prompting* yang efisien. Metode ini menggabungkan teknik *prompt tuning*, *deep prompting* dan reparameterisasi untuk meningkatkan performa (Ma et al., 2023). Representasi visual dari alur kerja yang menggabungkan Whisper dan pustaka Pyannote untuk proses transkripsi serta speaker *diarization* secara terstruktur disajikan pada Gambar 2. 4 Alur kerja dari Whisper berikut.



Gambar 2. 4 Alur kerja dari Whisper

Alur kerja yang ditunjukkan pada Gambar 2. 4 Alur kerja dari Whisper yaitu menggambarkan proses transkripsi audio menjadi teks dengan *speaker diarization*. Proses dimulai dengan *input* audio berupa gelombang suara yang masuk ke dalam sistem. Audio ini kemudian diproses melalui metode *Voice Activity Detection* (VAD) untuk mendeteksi bagian-bagian yang mengandung aktivitas suara. Setelah dideteksi, audio melewati proses *Cut* & *Merge* yang berfungsi untuk memotong dan menggabungkan bagian-bagian audio yang relevan.

Hasil dari proses *Cut* & *Merge* kemudian dibagi menjadi *batch* dengan *padding* hingga 30 detik. *Batch* audio ini selanjutnya diproses secara *parallel*. Pertama, model Whisper yang bertugas untuk melakukan transkripsi audio menjadi teks. Kedua, Pyannote *Library* yang berfungsi untuk melakukan diarisasi pembicara atau mengidentifikasi siapa yang berbicara pada setiap segmen audio.

Output akhir dari proses ini adalah transkrip teks yang dilengkapi dengan timestamp dan informasi pembicara. Format output menampilkan waktu mulai dan selesai untuk setiap kata

yang diucapkan, misalnya [00.15-0.30] untuk kata pertama, [00.50-1.26] untuk kata kedua dan seterusnya. Hasil ini memberikan transkrip yang terstruktur dengan penanda waktu yang akurat dan identifikasi pembicara yang jelas.

Evaluasi sistem dilakukan dengan memanfaatkan data dari acara diskusi dan program ulasan politik di Taiwan yang melibatkan 46 pembicara dari latar belakang yang beragam. Hasil pengujian menunjukkan tingkat kesalahan diarization kata (WDER) yang mengesankan, yaitu 2,68% untuk situasi dengan dua pembicara dan 11,65% untuk tiga pembicara, dengan rata-rata WDER sebesar 6,96%. Kinerja ini setara dengan model baseline non-real-time, menunjukkan kemampuan sistem untuk beradaptasi terhadap berbagai pola interaksi percakapan yang kompleks (Lyu et al., 2024).

Salah satu pustaka untuk membantu proyek ini pada fitur *speaker diarization* adalah *pyannote.audio*. Pustaka ini menggunakan bahasa Python yang berbasis *framework* PyTorch. Dengan menggabungkan teknologi ini, dapat membantu dalam proses untuk mengenali ucapan secara akurat, menentukan kapan *speaker* berbicara, sehingga dapat mengetahui siapa yang mengatakan apa dan kapan ia mengatakannya. Teknologi ini dapat dimanfaatkan untuk tranksrip pertemuan dengan banyak orang, menganalisis pembicara dan membantu dalam pembuatan notulensi sederhana di Perusahaan.

2.5 Tinjauan Pustaka

Tugas akhir ini membahas terkait aplikasi notulensi berbasis pengenalan suara atau *Speech Recognition* yang dikembangkan dengan Whisper AI. Teknologi ini membantu notulis dalam merangkum hasil dari rapat Telkomsel ataupun perusahaan lainnya secara otomatis melalui suara sehingga dapat mengetahui sejauh mana teknologi dapat menjangkau informasi yang diberikan dari pembicara.

Penelitian oleh (Matsuura et al., 2023) berjudul "*Leveraging Large Text Corpora for End-to-End Speech Summarization*" bertujuan untuk mengotomatisasi proses *summarization* dari hasil percakapan audio menjadi teks ringkas. Permasalahan utama yang diangkat dalam penelitian ini adalah keterbatasan kemampuan notulis dalam menangkap dan merangkum informasi penting dari percakapan panjang secara manual. Metode yang digunakan adalah pendekatan *end-to-end* berbasis *deep learning* yang memanfaatkan korpus teks berukuran besar. Model *Automatic Speech Recognition* (ASR) yang digunakan termasuk Whisper, yang kemudian diintegrasikan dengan pipeline summarization untuk menghasilkan ringkasan otomatis. Hasil penelitian ini menunjukkan bahwa pendekatan tersebut mampu menghasilkan

ringkasan teks yang padat dan relevan serta mempercepat proses pencatatan. Namun penelitian ini belum membahas secara detail kemampuan *diarization*, fitur untuk mengidentifikasi siapa yang berbicara dalam segmen audio tertentu. Hal ini menjadi fokus penting dalam penelitian yang sedang dilakukan karena *diarization* menjadi aspek krusial dalam konteks pencatatan hasil rapat dalam banyak dialog.

Penelitian oleh (Radford et al., 2022) yang berjudul "*Robust Speech Recognition via Large-Scale Weak Supervision*" mengangkat tantangan utama dalam pengenalan suara, yakni menurunnya akurasi dalam kondisi lingkungan audio yang penuh gangguan atau *noise*. Penelitian ini mengembangkan model Whisper melalui pelatihan berskala besar dengan metode *weak supervision* yang memanfaatkan jutaan jam data suara tanpa pelabelan eksplisit. Model Whisper yang dihasilkan memiliki keunggulan dalam mengenali berbagai bahasa dan aksen dengan performa tinggi, bahkan dalam kondisi audio yang tidak ideal. Hasil penelitian menunjukkan bahwa Whisper dapat mempertahankan akurasi transkripsi yang tinggi pada berbagai kondisi dan mampu mendukung penggunaan di berbagai domain termasuk layanan pelanggan, kesehatan hingga media. Meski begitu, penelitian ini tidak membahas integrasi *diarization* dan implementasi model dalam skenario aplikasi berbasis web seperti Hugging Face yang menjadi poin utama dalam penulisan ini.

Penelitian lain oleh (Lyu et al., 2024) berjudul "*Real-time Multilingual Speech Recognition and Speaker Diarization System based on Whisper Segmentation" mengembangkan sistem real-time* yang mampu mengenali ucapan banyak bahasa serta melakukan *speaker diarzation* secara bersamaan. Penelitian ini menggunakan model Whisper untuk transkripsi dan pustaka Pyannote untuk *diarization*. Fokus utama dari penelitian ini adalah menyempurnakan proses segmentasi ucapan serta meningkatkan akurasi *diarization* dalam situasi banyak pembicara seperti siaran diskusi atau debat politik. Evaluasi dilakukan menggunakan skenario percakapan nyata dengan hasil *Word Diarization Error Rate* (WDER) rata-rata sebesar 6,96%, menunjukkan performa tinggi untuk aplikasi *real-time*. Kelemahan dari penelitian ini adalah belum adanya pembahasan khusus mengenai aplikasi sistem ini dalam bentuk perangkat lunak siap pakai yang bisa digunakan langsung oleh perusahaan seperti yang dilakukan dalam pengembangan aplikasi SPERCO.

Berdasarkan studi diatas dapat dilihat bahwa pengembangan teknologi pengenalan suara telah menjangkau berbagai aspek mulai dari *summarization, robustness* dalam kondisi *noise* hingga *speaker diarization*. Namun belum ada studi yang secara utuh mengintegrasikan seluruh

komponen tersebut ke dalam sebuah sistem praktis yang dapat digunakan dalam lingkungan korporasi untuk membantu pencatatan rapat secara otomatis.

Penulisan ini bertujuan mengisi kekosongan tersebut dengan mengembangkan aplikasi SPERCO yang memanfaatkan Whisper AI untuk transkripsi dan *Speaker diarization*. Aplikasi ini dibangun pada platform Hugging Face untuk memastikan aksesibilitas dan skalabilitas sistem. Evaluasi dilakukan menggunakan metode *Ground Truth* dan metrik *Word Error Rate* (WER) untuk mengukur akurasi hasil transkripsi.

Dengan pendekatan yang menyeluruh, penulisan ini tidak hanya melengkapi kekurangan dari studi-studi terdahulu tetapi juga memberikan kontribusi nyata melalui implementasi sistem berbasis web yang siap digunakan dalam lingkungan kerja. Penelitian ini menawarkan solusi praktis, efisien dan akurat untuk pencatatan hasil rapat serta dapat dikembangkan lebih lanjut untuk mendukung fitur-fitur NLP lanjutan seperti *summarization* otomatis dan klasifikasi isi percakapan.

BAB III
PELAKSANAAN MAGANG

3.1 Aktivitas Magang

Telkomsel memberikan kesempatan penulis untuk merasakan kegiatan magang di perusahaan ini, banyak hal baru yang ditemukan oleh penulis selama mengikuti program kegiatan magang. Program magang ini dapat menjadi sebuah proses belajar kepada para pemagang agar dapat menerapkan ilmu dan skill yang dimiliki untuk diterapkan ke dunia kerja secara langsung. Selain itu, penulis juga mendapatkan relasi dan serta ilmu baru yang akan menjadi batu loncatan di masa yang akan datang.

Untuk memberikan gambaran yang terstruktur mengenai seluruh aktivitas yang dilakukan selama periode magang di Telkomsel, berikut disajikan tabel yang merinci rangkaian kegiatan secara kronologis. Adapun rincian lengkap aktivitas tersebut dapat dilihat dalam Tabel 3.1 berikut ini.

Tabel 3. 1 Aktivitas Magang

No	Aktivitas	Sep		Okt					Nov					Des		
		4	5	1	2	3	4	5	1	2	3	4	5	1	2	3
1	Pengenalan Lingkungan															
2	Pengenalan Proyek SPERCO															
3	Studi Literatur Teknologi Speech Recognition & Whisper AI															
4	Pembelajaran Mandiri															
5	Fixing bug Aplikasi SPERCO															
6	Melakukan Research Pendukung SPERCO															
7	Merancang fitur Speaker Diarization															
8	Implementasi & Pengujian SPERCO di Hugging Face															
9	Evaluasi Akurasi Transkripsi															
10	Pembuatan Dokumentasi Proyek															

Berdasarkan rangkaian aktivitas yang telah diuraikan pada tabel di atas, dapat dilihat bahwa program magang dilakukan secara bertahap, mulai dari pengenalan lingkungan hingga tahap dokumentasi proyek. Adapun program magang yang diikuti penulis untuk memenuhi

syarat penjaluran yaitu mengerjakan proyek perusahaan bersama tim dan divisi IT perusahaan, lebih tepatnya pada departemen IT *Operation* serta mendapat bimbingan intensif dari beberapa mentor yang berpengalaman di Departemen IT Operation yang turut membantu penulis dalam menyelesaikan setiap tahapan proyek tersebut.

Pada Minggu pertama pelaksanaan magang, penulis mulai diperkenalkan dengan proyek utama bernama SPERCO yaitu sebuah aplikasi berbasis *speech recognition* dan *speaker diarization* yang dikembangkan untuk mendukung kebutuhan internal perusahaan Telkomsel. Proyek ini sebelumnya telah dirintis oleh pemagang terdahulu dan kini dilanjutkan oleh penulis bersama tim dari divisi IT Operation. Dalam proses pengenalan tersebut, penulis mendapatkan pemahaman awal mengenai tujuan proyek, teknologi yang digunakan, serta ruang lingkup pengembangannya. Namun, dalam tahapan awal ini juga ditemukan sejumlah kendala teknis. Aplikasi SPERCO masih belum sepenuhnya sempurna dan stabil. Terdapat beberapa *bug* dan *error* sistem seperti ketidaksesuaian dalam output transkripsi. Selain itu, dokumentasi teknis yang terbatas dari pengembangan sebelumnya menjadi tantangan tersendiri bagi penulis dalam memahami alur kerja proyek secara menyeluruh.

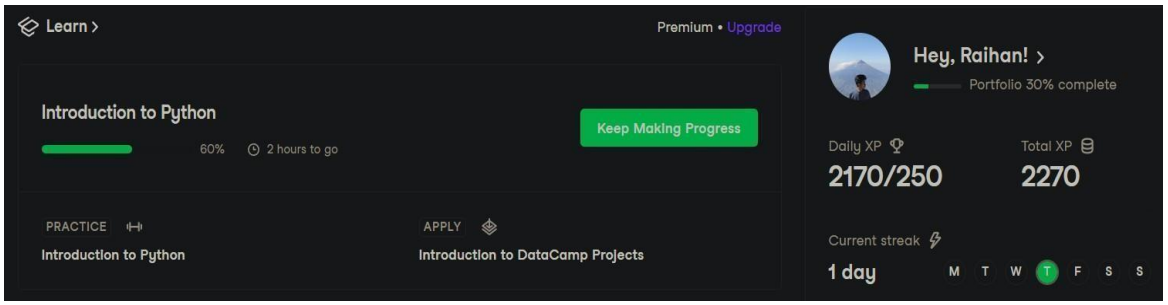
Setelah diperkenalkan dengan proyek SPERCO, Minggu berikutnya penulis mulai masuk ke tahap studi literatur, terutama tentang teknologi yang akan digunakan dalam aplikasi ini. Fokus utama ada pada *Speech Recognition* dan model Whisper AI dari OpenAI. Karena sebelumnya belum pernah mendalami bidang ini, penulis merasa penting untuk memahami bagaimana teknologi ini bekerja agar lebih mudah saat proses implementasi nanti.

Whisper sendiri merupakan salah satu model *speech-to-text* yang cukup populer karena kemampuannya mengenali berbagai bahasa dan tetap akurat meskipun kondisi audio tidak terlalu ideal. Dari hasil pembacaan dokumentasi dan beberapa jurnal, penulis mulai mengenal berbagai jenis model Whisper mulai dari versi kecil seperti *tiny* sampai yang paling kompleks seperti *large-v2*. Masing-masing punya kelebihan dan kekurangan tergantung kebutuhan proyek.

Untuk lebih memahami kebutuhan proyek secara teknis, penulis juga melakukan pembelajaran secara mandiri. Tujuannya tentu agar bisa lebih siap saat mulai mengembangkan fitur-fitur dalam aplikasi SPERCO. Salah satu langkah awal yang diambil adalah mengikuti pelatihan online melalui DataCamp Academy. Platform ini dipilih karena menyediakan materi yang cukup lengkap, terutama untuk pemula yang ingin memperdalam bahasa pemrograman Python dan dasar-dasar data science.

3.1.1 Pembelajaran Mandiri

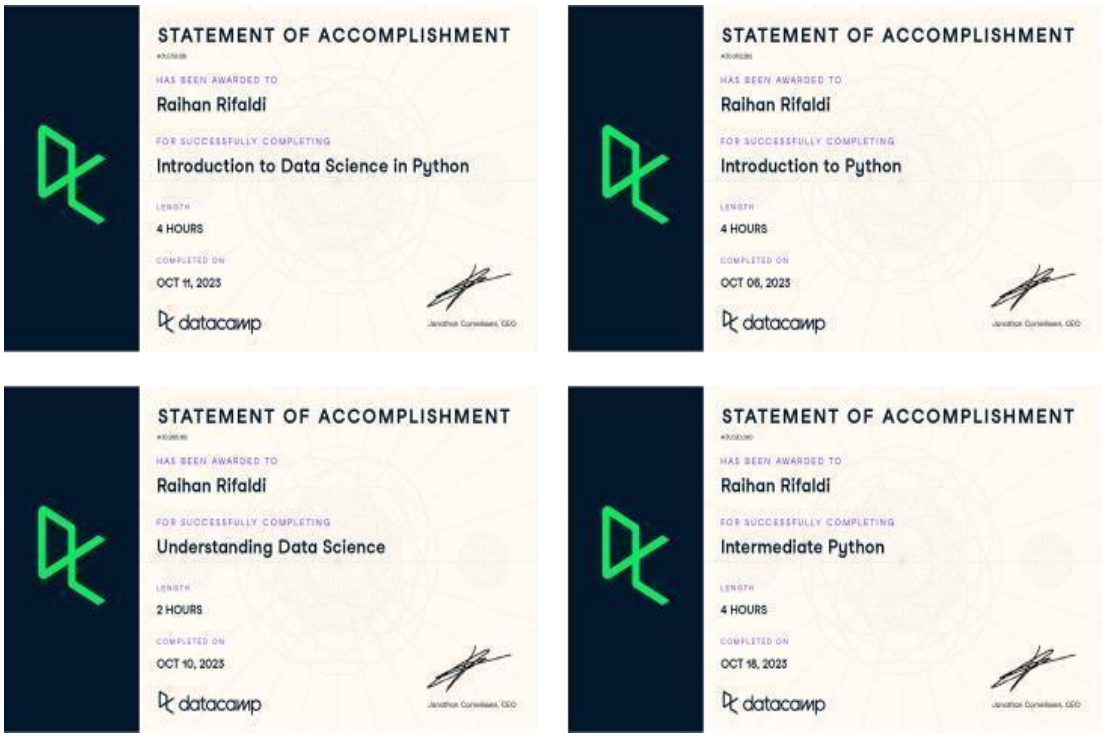
Untuk lebih memahami kebutuhan proyek secara teknis, penulis juga melakukan pembelajaran secara mandiri. Tujuannya tentu agar bisa lebih siap saat mulai mengembangkan fitur-fitur dalam aplikasi SPERCO. Kebutuhan dari proyek ini seperti mempelajari bahasa pemograman Python dari tahap dasar hingga menengah. Penulis dapat melihat progres pembelajaran mandiri pada Gambar 3. 1 Pembelajaran secara mandiri.



Gambar 3. 1 Pembelajaran secara mandiri

Tampilan *dashboard* pembelajaran pada gambar diatas menunjukkan progres belajar penulis selama mengikuti pelatihan online di DataCamp Academy. *Dashboard* ini memberikan ringkasan pencapaian seperti persentase penyelesaian modul, durasi belajar yang tersisa dan serta statistik harian dan mingguan terkait aktivitas pembelajaran

Dalam proses pelatihan, penulis memulai pembelajaran secara mandiri dengan mendalami materi *Introduction to Python*, yang mencakup dasar-dasar Python, manipulasi *lists*, penggunaan fungsi, pengelolaan *packages*, hingga pengenalan awal pada *NumPy*. Setelah memahami fondasi tersebut, penulis melanjutkan pembelajaran mandiri ke tahap berikutnya, yaitu eksplorasi *Understanding and Introduction to Data Science in Python*.



Gambar 3. 2 Pelatihan di Datacamp Academy

Pada tahap ini, penulis mempelajari langkah awal dalam Python untuk analisis data, teknik visualisasi data menggunakan Matplotlib dan berbagai jenis diagram untuk menyajikan informasi. Tahap berikutnya adalah *Intermediate Python*, di mana penulis mendalami materi lanjutan, seperti penggunaan Matplotlib secara lebih kompleks, pengelolaan data dengan *dictionaries* dan *pandas*, logika pemrograman, *control flow* serta teknik penyaringan data. Proses belajar ini ditutup dengan penerapan keterampilan melalui *case study* yang berjudul *Hacker Statistics*, memberikan penulis pemahaman mendalam dan pengalaman praktis dalam analisis data. Ketika penulis berhasil menyelesaikan pelatihan di Datacamp Academy akan mendapatkan sertifikat yang dapat dilihat pada Gambar 3. 2 Pelatihan di Datacamp Academy.

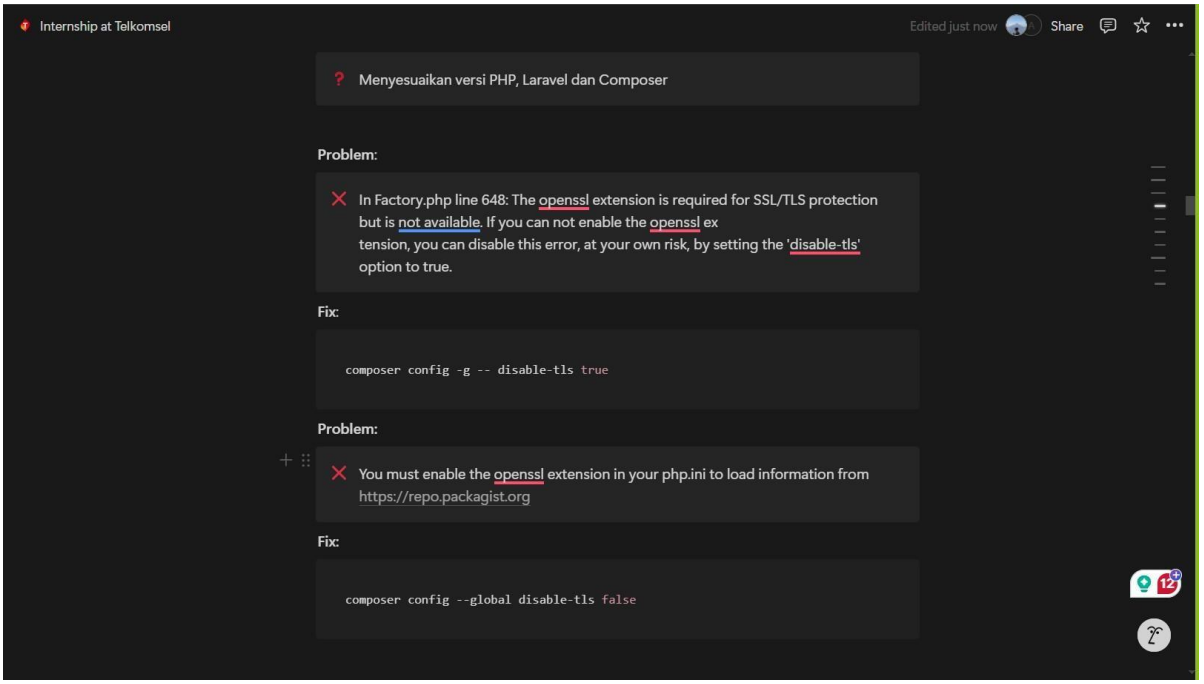
3.1.2 Fixing Bug Aplikasi SPERCO

Setelah menyelesaikan pembelajaran mandiri, penulis pun masuk ke tahapan selanjutnya, yaitu *fixing bug*. Tahapan ini mulai dikerjakan pada minggu berikutnya dan menjadi langkah penting sebelum benar-benar mengembangkan fitur baru di aplikasi SPERCO. Tujuannya untuk menghindari berbagai kendala teknis yang pernah muncul saat dikerjakan oleh pemagang sebelumnya bisa dihindari sedini mungkin.

Bug yang ditemukan cukup beragam, mulai dari error pada proses transkripsi otomatis, kendala dalam pemilihan model yang sesuai, hingga tampilan antarmuka yang masih belum

optimal. Proses debugging ini tidak selalu mudah, tapi sangat membantu penulis dalam memahami struktur kode dan alur kerja aplikasi secara menyeluruh.

Selama proses *fixing bug* ini penulis tidak bekerja sendirian. Ada bantuan dari mentor dan tim pengembang, yang siap memberikan arahan atau membantu ketika penulis mengalami kesulitan. Melalui diskusi dan sesi *troubleshooting* bareng tim, Penulis jadi lebih terbiasa dalam menangani masalah teknis secara sistematis. Selain itu, banyak pembelajaran baru yang didapat, terutama dalam hal identifikasi *error* dan cara menyelesaikannya dengan pendekatan yang efisien. Aktivitas pemagang tertulis pada *logbook* yang digunakan di Notion yang dapat dimanfaatkan sebagai dokumentasi *error* dan *solving* pada proyek yang dikerjakan seperti Gambar 3. 3 Dokumentasi *error* dan *solving* pada Notion.



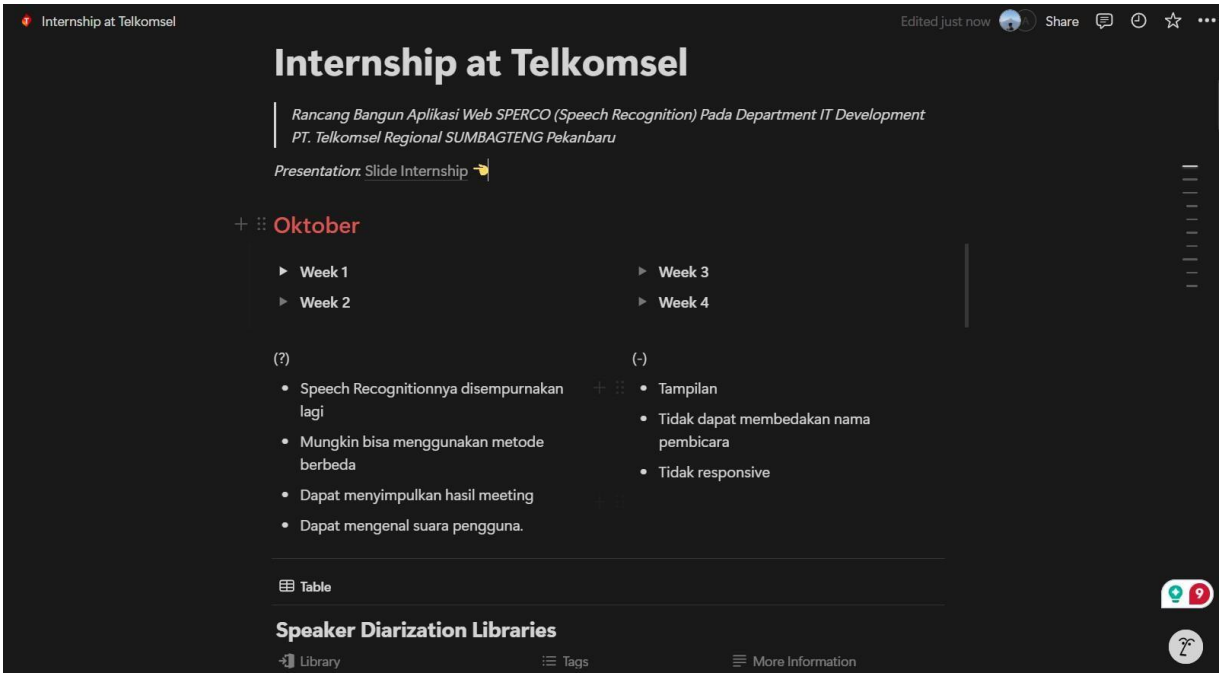
Gambar 3. 3 Dokumentasi *error* dan *solving* pada Notion

Gambar diatas merupakan salah satu *fixing bug* yang dilakukan penulis. Ada kendala teknis yang berkaitan dengan ekstensi *openssl* dan konfigurasi *Composer*. Masalah pertama muncul ketika *Composer* memunculkan *error* bahwa ekstensi openssl tidak tersedia, padahal ekstensi ini diperlukan untuk perlindungan SSL/TLS dalam komunikasi dengan repositori seperti *Packagist*. Sebagai solusi sementara, penulis menggunakan perintah seperti Gambar 3. 3 Dokumentasi *error* dan *solving* pada Notion untuk menonaktifkan TLS secara global demi melanjutkan instalasi dependensi.

Masalah kedua berkaitan dengan keharusan mengaktifkan kembali ekstensi *openssl* agar *Composer* dapat mengambil informasi dari repositori secara aman. Oleh karena itu, penulis

mengubah kembali konfigurasi TLS *Composer* dan memastikan ekstensi *openssl* aktif. Proses ini menunjukkan pentingnya kesesuaian antara versi dan konfigurasi PHP dengan ekosistem *Laravel* dan *Composer* serta pemahaman teknis dalam menangani *error* konfigurasi sistem untuk memastikan pengembangan aplikasi tetap berjalan lancar dan aman.

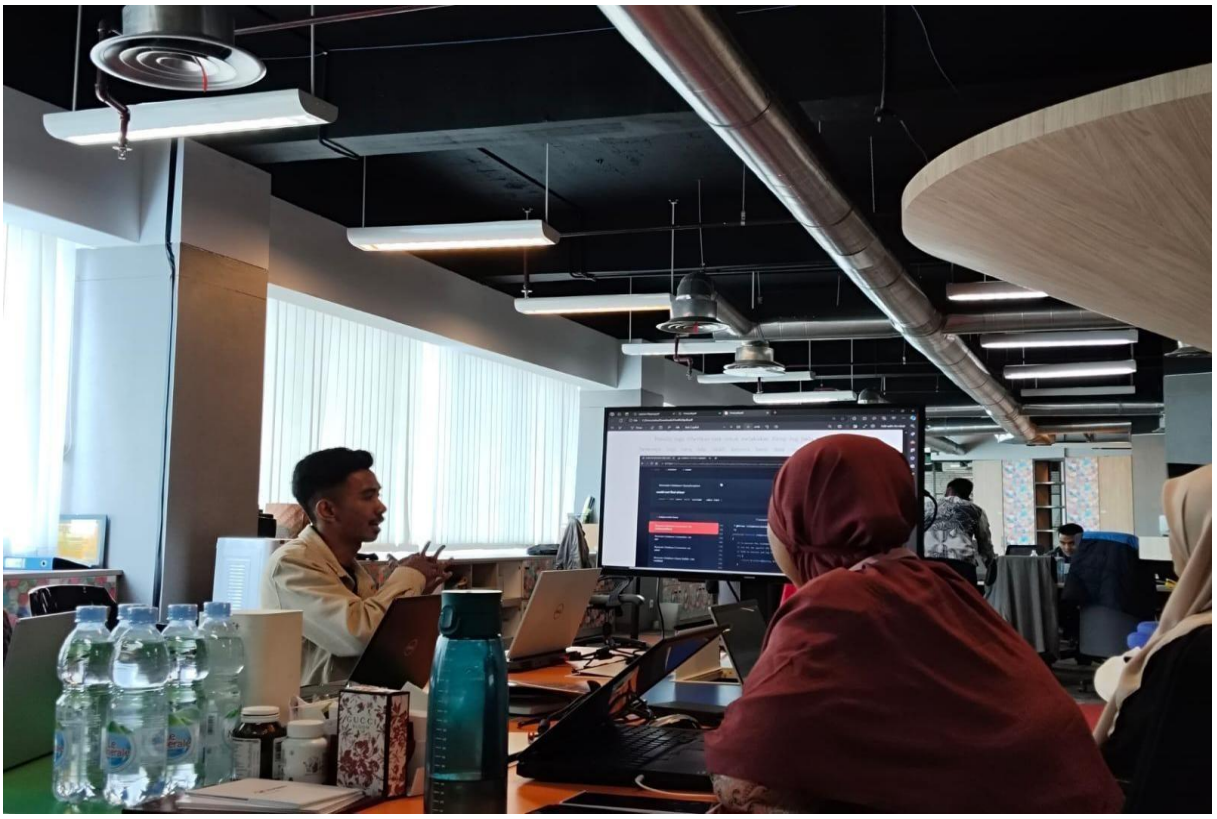
Selain *Fixing bug*, penulis juga secara rutin melakukan pencatatan *logbook* kegiatan magang yang diakses melalui Notion. *Logbook* ini berfungsi sebagai media dokumentasi tugas-tugas yang telah dan sedang dikerjakan, termasuk catatan progres, kendala yang dihadapi serta rencana tindak lanjut seperti pada Gambar 3. 4 Platform Notion untuk laporan.



Gambar 3. 4 Platform Notion untuk laporan

Pengisian *logbook* dilakukan setiap minggu sebagai bentuk tanggung jawab dan transparansi terhadap supervisor magang. Data yang tersimpan dalam *logbook* juga menjadi dasar evaluasi kinerja serta perkembangan pembelajaran penulis selama masa magang.

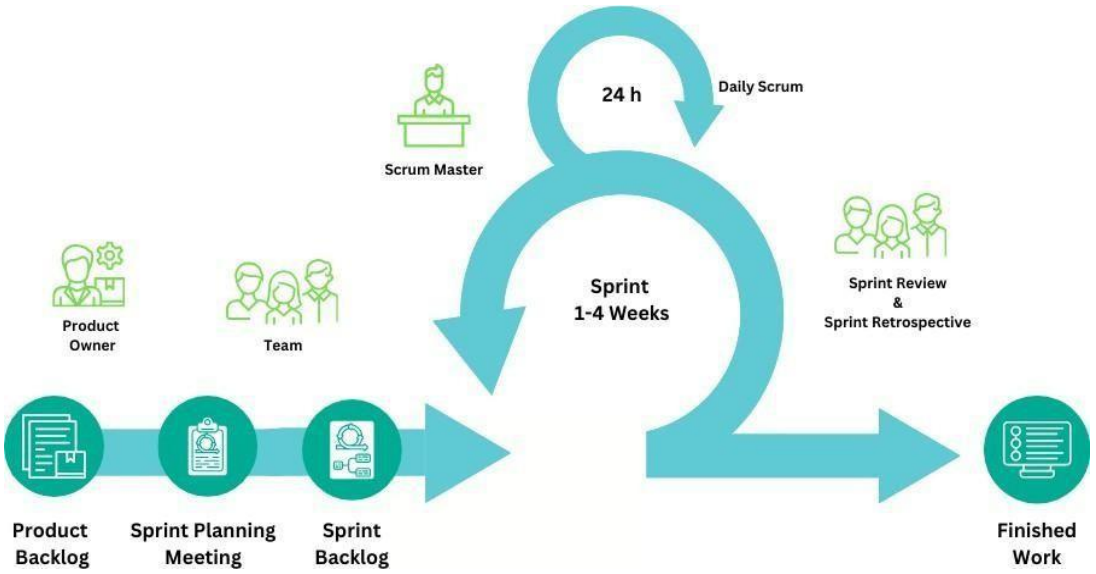
Selain itu menulis laporan pada notion, Penulis juga melakukan rapat rutin proyek yang diadakan rutin setiap minggunya bersama tim, mentor dan supervisor seperti pada Gambar 3. 5 Rapat Rutin Proyek.



Gambar 3. 5 Rapat Rutin Proyek

Pada sesi ini penulis juga menyampaikan logbook magang agar dapat diberi masukkan ataupun hal lainnya. Tujuan dari rapat rutin proyek ini untuk mengetahui tugas dan progres dari proyek yang sedang dikerjakan yang harapannya jika terjadi kendala yang tidak dapat diselesaikan akan dibantu dalam mencari solusi untuk menyelesaikan masalah terkait. Dalam peran sebagai pemagang, penulis bertanggung jawab menyelesaikan isu atau tugas yang diberikan mentor, berkontribusi dalam pengembangan aplikasi bersama tim serta mengelola *logbook* dan laporan mingguan. Ada aplikasi internal Telkomsel yang digunakan pada pada setiap region yang berguna untuk menulis laporan serta garis waktu dari pengerjaan setiap proyek, namun aplikasi bersifat internal dan tidak boleh untuk diperlihatkan. Sebagai gantinya *logbook* dan laporan menggunakan aplikasi Notion.

Untuk mendukung pengelolaan proyek yang efektif dan terstruktur selama masa magang, penulis bersama tim menerapkan pendekatan manajemen proyek berbasis *Agile* yaitu metode Scrum. Metode ini dipilih karena memiliki kerangka kerja yang adaptif dan berorientasi pada kolaborasi tim secara berkelanjutan. Dalam. Pendekatan ini memungkinkan evaluasi progres kerja secara berkala, identifikasi kendala lebih dini serta peningkatan kualitas hasil kerja secara bertahap seperti pada Gambar 3. 6 Metode Scrum.



Gambar 3. 6 Metode Scrum

Sumber : (Muay et al., 2024)

Scrum adalah sebuah kerangka kerja yang sering diterapkan dalam metode *Agile* untuk mengelola proyek secara bertahap dengan melibatkan kolaborasi aktif antar anggota tim. Proses ini dimulai dengan pembuatan *product backlog*, yang dilakukan oleh product owner dan scrum master dengan berdiskusi dengan stackholder terkait yaitu daftar kebutuhan atau fitur yang perlu diwujudkan dalam pengembangan. Sebagian dari *backlog* ini dipilih untuk masuk ke dalam *sprint backlog*, yang akan menjadi fokus pengerjaan selama satu periode *sprint* seperti Tabel 3. 2.

Tabel 3. 2 Sprint Planing

Sprint	**Fitur**	**Durasi**	**Prioritas**
Sprint 1	Pengenalan proyek, studi literatur Whisper & *Speech Recognition*	7 hari	Tinggi
Sprint 2	Pembelajaran mandiri, *fixing bug* awal aplikasi SPERCO	14 hari	Tinggi, Menengah
Sprint 3	Perancangan fitur *diarization* & integrasi dengan Hugging Face	21 hari	Tinggi
Sprint 4	Implementasi fitur transkripsi dan evaluasi akurasi transkripsi	21 hari	Tinggi
Sprint 5	Penyusunan dokumentasi teknis & pengujian akhir	10 hari	Menengah

Tabel 3.2 menyajikan pembagian *sprint planning* yang dirancang untuk memetakan tahapan kerja dalam pengembangan aplikasi SPERCO selama periode magang. Setiap sprint disusun dengan memperhatikan kompleksitas fitur yang akan dikerjakan, kebutuhan teknis proyek serta alokasi waktu yang tersedia. Pembagian ini bertujuan agar seluruh proses

pengembangan berlangsung terstruktur, terukur, dan selaras dengan prinsip kerja Agile berbasis Scrum.

Namun pada praktek di lapangan, ada perubahan prioritas dalam pengembangan aplikasi. Dalam setiap *sprint*, tim yang terdiri dari *product owner*, *scrum master* dan *development team* bekerja bersama untuk menghasilkan Increment, yakni bagian hasil kerja yang sudah siap diuji atau digunakan. Sepanjang *Sprint*, diadakan pertemuan harian singkat atau *daily* scrum seperti Gambar 3. 5 Rapat Rutin Proyek yang terjadwal setiap Jumat Pagi pukul 09.00 WIB hingga selesai. Rapat ini bertujuan untuk memantau kemajuan dan menyelesaikan kendala yang ada. Setelah *sprint* selesai, masuk pada tahapan *sprint review* dan *Sprint Retrospective*

Sprint Review

Pada tahap Sprint Review, kegiatan evaluasi dilakukan oleh tim pengembang bersama Product Owner dan Supervisor dari pihak Telkomsel. Waktu pelaksanaan review umumnya dilakukan di akhir Minggu, menyesuaikan dengan jadwal kerja yang lebih fleksibel sehingga memungkinkan evaluasi dilakukan secara lebih menyeluruh. Setiap fitur diuji secara langsung untuk memastikan bahwa hasil pengembangan sesuai dengan spesifikasi kebutuhan proyek. Selain proses peninjauan, dilakukan pula pengujian fungsional secara unit terhadap fitur-fitur utama seperti transkripsi otomatis, identifikasi pembicara (*speaker diarization*) serta antarmuka pengguna. Pengujian ini dilakukan pada platform Hugging Face, tempat di mana aplikasi SPERCO dikembangkan untuk keperluan demonstrasi dan validasi. Adapun hasil dari *Sprint Review* dan pengujian akhir yang dilakukan oleh supervisor proyek di Telkomsel adalah sebagai berikut:

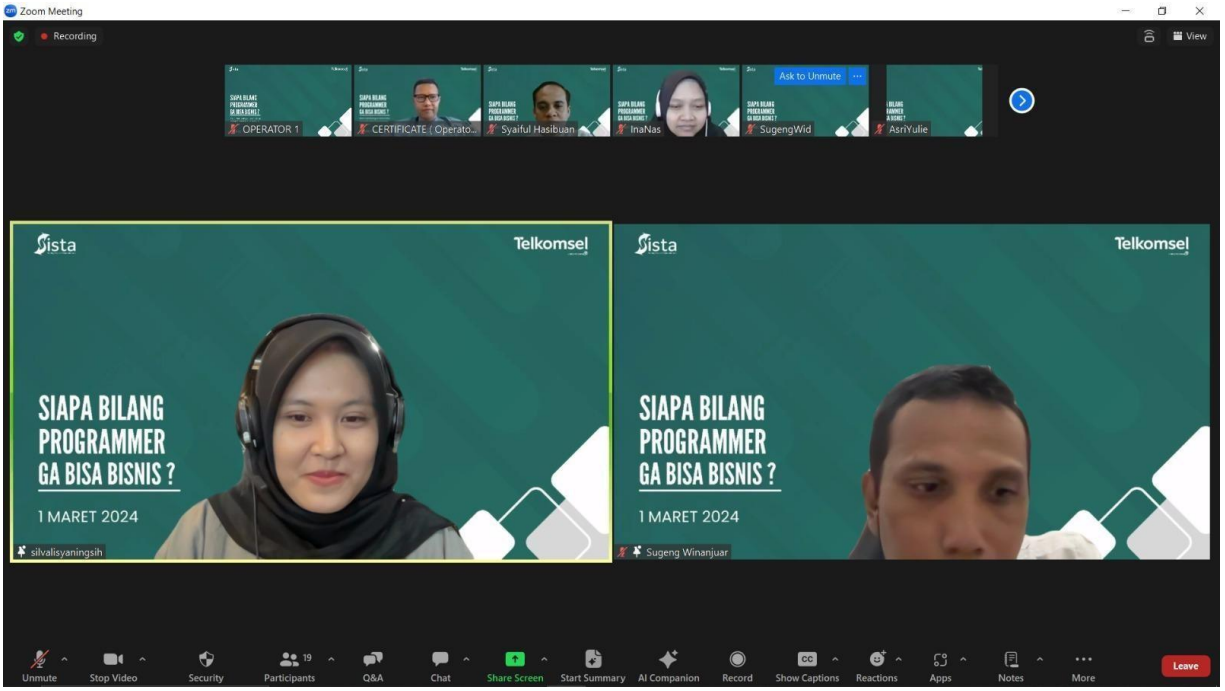
a. Seluruh fitur utama, termasuk transkripsi dan diarization, telah berfungsi sesuai harapan.
b. Aplikasi dapat diakses melalui platform Hugging Face dan memberikan hasil transkripsi yang akurat serta responsif terhadap input pengguna.

Sprint Retrospective

Setelah setiap *Sprint Review*, dilakukan sesi *Sprint Retrospective* bersama mentor, supervisor dan penulis sebagai bagian dari tim pengembang aplikasi SPERCO. Tahap ini bertujuan untuk mengevaluasi proses pelaksanaan *sprint* secara menyeluruh, mengidentifikasi strategi yang telah berjalan baik serta mengulas kendala yang muncul selama pengerjaan. Diskusi dilakukan

secara terbuka melalui pertemuan rutin dan dokumentasi aktivitas, kendala teknis serta rencana tindak lanjut. Secara umum, seluruh fitur yang direncanakan seperti transkripsi otomatis, pemilihan model Whisper hingga integrasi dengan Hugging Face berhasil dikembangkan dengan baik. Namun demikian, ketika di lapangan masih tetap ada kendala yang tidak terduga muncul selama proses pengembangan.

Selama masa magang berlangsung, setiap bulannya Telkomsel menyelenggarakan sesi pengembangan diri yang dilakukan secara daring melalui platform Zoom. Pada Gambar 3. 7 Webinar bulanan kegiatan tersebut dikemas dalam bentuk webinar yang terbuka untuk seluruh peserta magang dengan topik yang beragam dan relevan dengan kebutuhan pengembangan karier masa depan.



Gambar 3. 7 Webinar bulanan

Dari sesi tersebut, penulis mendapatkan sudut pandang baru bahwa kemampuan teknis sebagai programmer dapat menjadi modal awal untuk berwirausaha, terlebih di era digital yang semakin terintegrasi dengan teknologi. Para narasumber memberikan inspirasi melalui pengalaman pribadi mereka yang sukses membangun usaha sambil tetap aktif dalam dunia IT. Kegiatan semacam ini sangat relevan dan membuka peluang besar bagi peserta magang dab karyawan untuk mempelajari banyak hal diluar bidang mereka. Telkomsel membentuk ekosistem pembelajaran yang menyeluruh yang tidak hanya teknikal tetapi juga mindset dan soft skill yang menunjang karier jangka panjang.

3.1.3 Persiapan *Tools pendukung* Pembuatan Aplikasi

1. Model Whisper OpenAI

Berdasarkan studi proyek, pengembangan sistem pengenalan suara multibahasa *real-time* dan *diarization* pembicara canggih memanfaatkan model Whisper dari OpenAI. Model ini secara khusus dirancang untuk mengatasi tantangan *Automatic Speech Recognition* (ASR) dan *Speaker Diarization* (SD) di lingkungan dinamis dengan banyak pembicara, dengan fokus pada pemrosesan akurasi ucapan.

Dalam proyek ini, model Whisper dipilih sebagai komponen utama karena beberapa pertimbangan penting dari data jurnal. Whisper cukup popular dalam pengembangan *speech recognition* yang digunakan sebagai sampel untuk evaluasi kinerja. Setelah dilakukan pengujian menyeluruh, Whisper menunjukkan performa yang paling memuaskan di antara model-model yang diuji. Lebih lanjut, ketika diimplementasikan dalam konteks proyek ini, Whisper konsisten memberikan hasil yang sangat baik. Berdasarkan kombinasi faktor-faktor tersebut, Whisper menjadi pilihan yang paling tepat untuk memenuhi tujuan dan kebutuhan spesifik proyek menggunakan *Whisper Model* seperti gambar dibawah.

```
from faster_whisper import WhisperModel

#daftar model dalam ukuran, akurasi, dan bahasa
whisper_models = ["tiny", "base", "small", "medium", "large-v1",
"large-v2"]
source_languages = {
    "id": "Indonesian",
    "en": "English",
  "jw": "Javanese",
    "su": "Sundanese", }

#Mengonfigurasi pipeline untuk memproses data audio
MODEL_NAME = "cahya/whisper-medium-id"
lang = "id"
device = 0 if torch.cuda.is_available() else "cpu"
pipe = pipeline(
    task="automatic-speech-recognition",
    model=MODEL_NAME,
```

```
    chunk_length_s=30,
device=device, )
```

Gambar 3. 8 Kutipan kode menggunakan *Whisper Model*

Pada Gambar 3. 8 Kutipan kode menggunakan *Whisper Model*terdapat kode yang digunakan sebagai perintah untuk mengimpor Whisper Model dari *Faster_Whisper* yang digunakan untuk menjalankan model Whisper untuk transkripsi audio menjadi teks. Terdapat juga model Whisper yang tersedia, mulai dari model kecil (*Tiny*) hingga paling besar (*large-v2*) yang dapat dilihat pada tabel di bawah ini. Setiap model memiliki karakteristik berbeda seperti jumlah parameter, kebutuhan VRAM serta kecepatan proses transkripsi. Pemilihan model ini dapat disesuaikan dengan kebutuhan proyek, baik dari segi akurasi hasil transkripsi maupun efisiensi waktu pemrosesan berdasarkan spesifikasi perangkat yang digunakan.

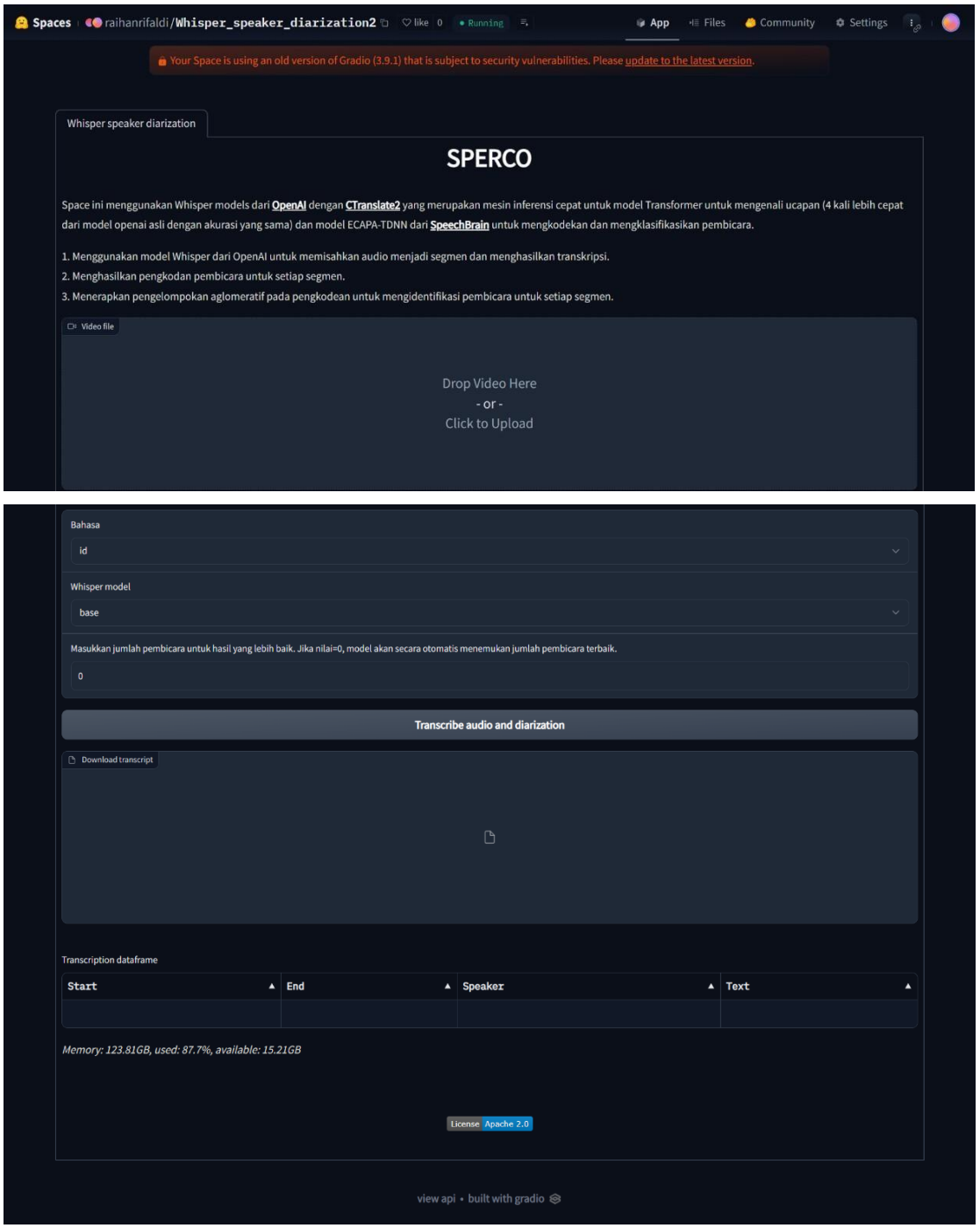
Tabel 3. 3 *Whisper Models details*

Multilingual model	**Parameters**	**Required VRAM**	**Relative Speed**
Tiny	39 M	~1 GB	~32x
Base	74 M	~1 GB	~16x
Small	244 M	~2 GB	~6x
medium	769 M	~5 GB	~2x
Large	1550 M	~10 GB	~1x

Pada Tabel 3. 3 menunjukkan bahwa Whisper menggunakan pendekatan pembelajaran *multi-task* untuk menyempurnakan dan meningkatkan parameter modelnya yang mencakup tugas seperti transkripsi, mengubah ucapan ke dalam bahasa Inggris, mendeteksi aktivitas suara dan mengidentifikasi bahasa. Keunggulannya terletak pada kemampuan luar biasa dalam menangani ASR multibahasa dan terjemahan dengan tingkat keakuratan yang tinggi

2. Hugging face

Proyek ini dikembangkan menggunakan platform Hugging Face, sebuah layanan yang mendukung pengembangan dan penyebaran aplikasi berbasis kecerdasan buatan dengan berbagai fitur dan kemudahan yang dapat digunakan dalam proyek sehingga dapat membantu penulis dalam melakukan pengembangan yang dapat dilihat pada gambar dibawah ini.



Gambar 3. 9 Tampilan SPERCO di Hugging Face

Pada Gambar 3.9 SPERCO dapat dijalankan di Hugging Face. Dikarenakan proyek ini merupakan aplikasi yang akan nantinya digunakan pada kebuthan internal perusahaan, pengujian serta data yang boleh digunakan pada penulisan ini hanya tertera pada gambar di atas. Space Hugging face pada proyek ini digunakan pada layanan gratis yang memiliki spesifikasi terdiri dari CPU *basic* dengan 2 vCPU dan RAM 16 GB, SDK Gradio dan *Storage* 50 GB (*ephemeral*).

Namun, ada beberapa tantangan yang perlu diperhatikan dan solusi yang dapat diterapkan untuk mengoptimalkan penggunaan Hugging Face:

1. Penyimpanan dan Pengelolaan File:

- Tantangan: Storage 50 GB (*ephemeral*) yang bersifat sementara, data atau file yang disimpan di dalamnya akan hilang jika Space tersebut dimulai ulang (*restart*) atau dihentikan (*stopped*).
- Solusi: Menyimpan data secara lokal di pengguna atau menghubungkan Visual Studio Code pada Hugging Face, bertujuan jika sewaktu-waktu data terhapus akan dapat digunakan kembali dengan cara mengunggah file menggunakan terminal *command* pada Visual Studio Code.

2. Transkripsi Video Audio:

1. Tantangan: Proses transkripsi video audio akan lama jika video tersebut memiliki jumlah suara yang lebih dari 2 pembicara.
2. Solusi: Ketika mengunggah file dapat memasukkan jumlah pembicara secara manual untuk membantu mempercepat proses pengelolaan audio, jika tidak ditulis, model akan secara otomatis menemukan jumlah pembicara terbaik dan membutuhkan waktu yang cukup lama.

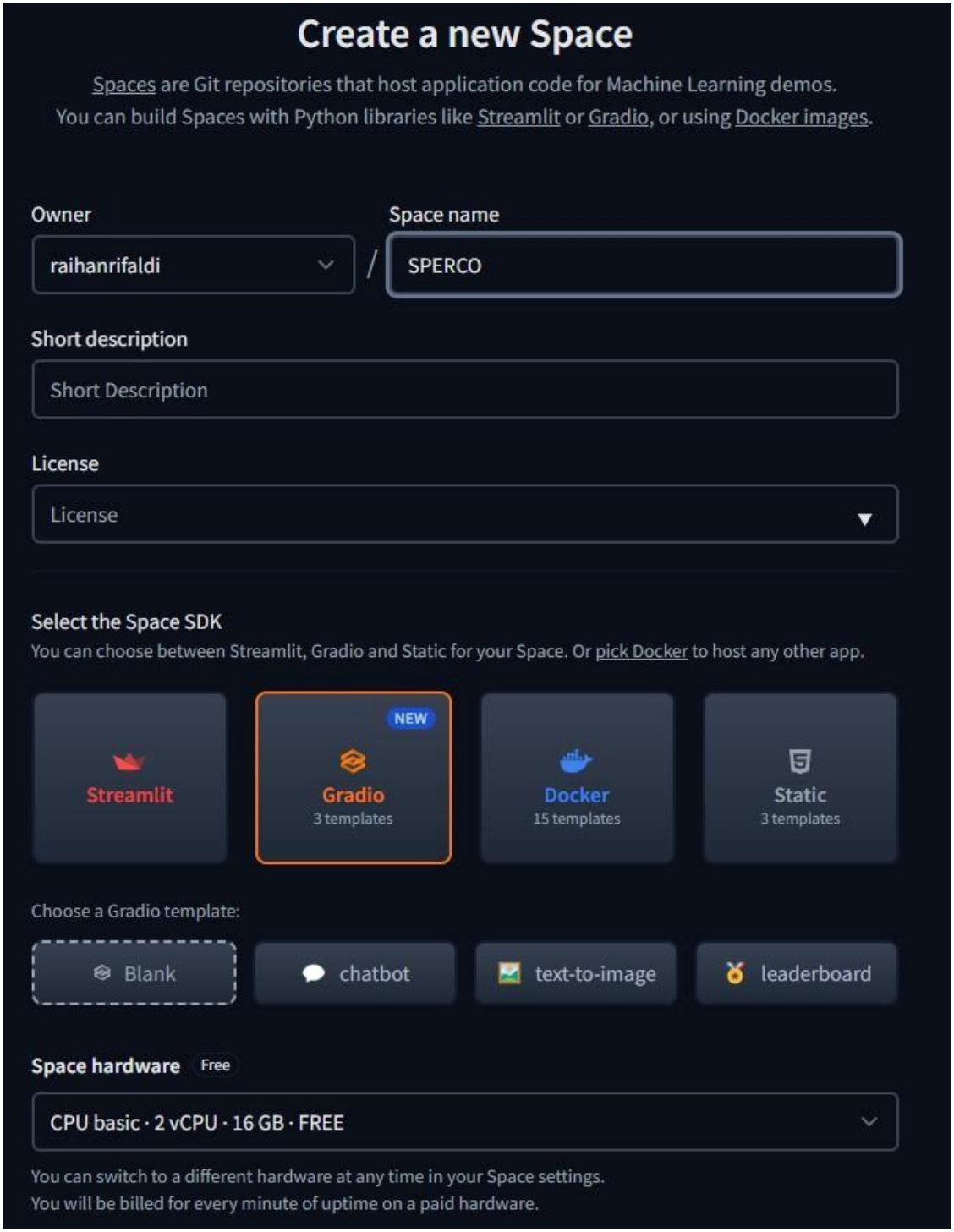
3. Kualitas Transkrip

- Tantangan: Proses transkripsi video audio akan membutuhkan waktu yang sangat lama bahkan tidak terdeteksi jika memiliki tingkat *noise* audio yang cukup tinggi.
- Solusi: Memilih *Whisper Model* sebelum melakukan input transkrip audio untuk mengefisiensikan waktu dalam mengelola audio.

Pada pengembangan di Hugging Face, pustaka memberikan kemudahan dalam membangun antarmuka pengguna interaktif untuk aplikasi *machine learning* dan AI yang disebut Gradio. Gradio sering digunakan untuk membuat antarmuka pengguna yang interaktif untuk model-model NLP, pengenalan gambar, pengolahan suara dan lainnya. Antarmuka ini memungkinkan pengguna untuk mengunggah data, memberikan input dan menerima hasil prediksi atau keluaran model secara *real-time* melalui antarmuka berbasis *browser*.

Selain itu, Gradio memiliki kemampuan integrasi langsung dengan model yang di *hosting* pada Hugging Face. Pengguna dapat dengan cepat memasukkan model yang tersedia di Hugging Face dan membuat antarmuka Gradio untuk mendemonstrasikan fungsionalitas model tersebut. Misalnya, sebuah model dari Whisper dapat dihubungkan dengan Gradio untuk memungkinkan pengguna mengunggah file audio dan menerima transkripsi teks dalam waktu

singkat. Berikut tampilan sebelum memulai *workspace* yang ada di Hugging Face seperti tertera Gambar 3.10.



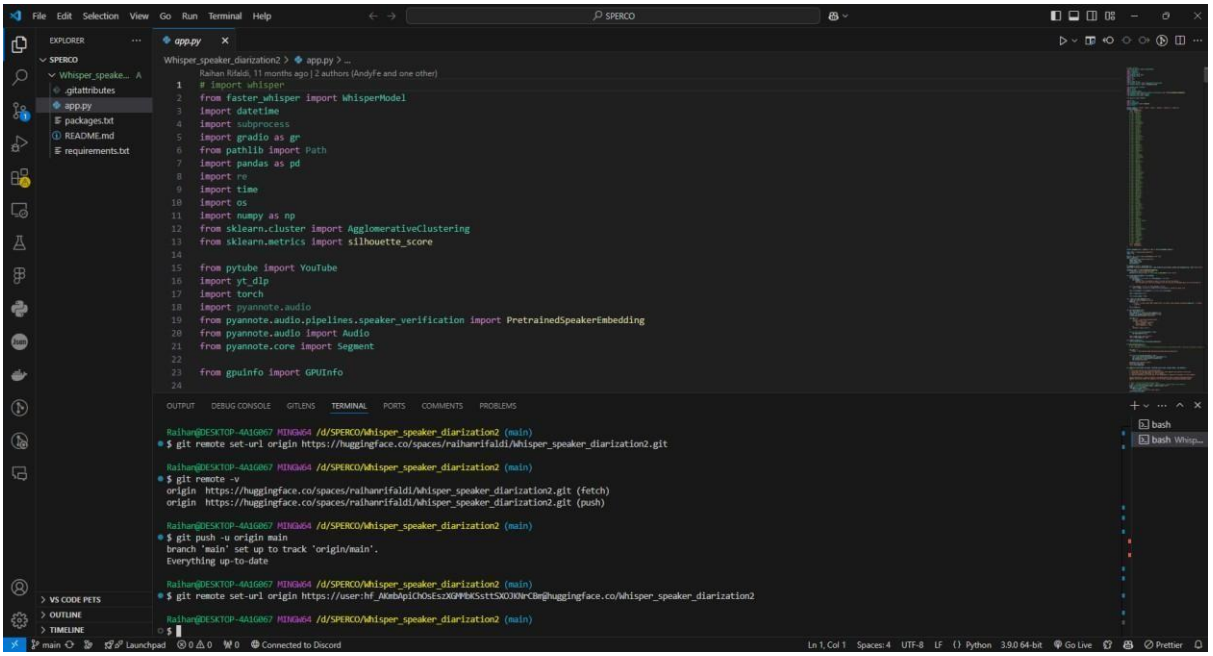
Gambar 3. 10 Halaman pembuatan *Workspace* di Hugging Face

Gambar di bawah ini menunjukkan tampilan antarmuka saat membuat ruang kerja (*space*) baru di Hugging Face. Pada tahap ini, pengguna dapat menentukan nama proyek, deskripsi singkat, lisensi serta memilih jenis SDK yang akan digunakan seperti Gradio,

Streamlit, Docker atau Static. Dalam proyek SPERCO, Gradio dipilih karena kemampuannya dalam membangun antarmuka pengguna yang interaktif dan mudah diintegrasikan dengan model yang dihosting. Selain itu, pengguna juga dapat menentukan spesifikasi perangkat keras yang akan digunakan, termasuk jumlah CPU dan RAM. Langkah ini menjadi awal penting dalam proses penyebaran aplikasi berbasis AI di Hugging Face. Tampilan lengkapnya dapat dilihat pada Gambar 3. 10 Halaman pembuatan *Workspace* di Hugging Face.

3. Visual Studio Code

Visual Studio Code ini dirancang untuk mempermudah proses pemrograman kode dengan menyediakan integrasi langsung ke platform Hugging Face. Fitur ini memungkinkan pengguna untuk melakukan *git clone* dan memperbarui kode secara berkala tanpa hambatan. Dengan adanya integrasi ini seperti seperti pada Gambar 3. 11 *Remote* Hugging Face dari Visual Studio Code *developer* dapat mengakses dan memperbarui repositori mereka dengan mudah, mengurangi kemungkinan terjadinya kesalahan yang disebabkan oleh ketidaksesuaian versi kode melalui Visual Studio yang sudah terhubung dengan Hugging Face.



Gambar 3. 11 *Remote* Hugging Face dari Visual Studio Code

Git pull dan *push* kode juga dapat dilakukan seperti yang tertera di terminal pada Gambar 3. 11 *Remote* Hugging Face dari Visual Studio Codeyang memudahkan penulis melakukan pemeliharaan secara berkala. Data Selain itu, proses ini juga sangat efektif dalam meningkatkan efisiensi waktu, karena meminimalkan kebutuhan untuk melakukan peninjauan ulang program secara manual. Pengguna dapat lebih fokus pada pengembangan dan perbaikan

kode, sementara *editor* secara otomatis menjaga agar kode tetap *up-to-date* dengan perubahan yang terjadi. Hal ini sangat membantu dalam menjaga kelancaran alur kerja dan memastikan kualitas kode yang lebih baik, seperti yang dapat dilihat pada Gambar 3.11.

4. Melakukan *Research* pendukung SPERCO

Pada tahapan ini *research* disajikan dalam bentuk tabel yang menguraikan informasi seperti Nama, Status Cara Implementasi, Deskripsi dan Kelemahan. Masing-masing teknologi dikategorikan berdasarkan fungsinya, misalnya sebagai, *Toolkit* atau *Model*. Dalam konteks ini, Whisper Open AI menjadi salah satu teknologi utama yang telah dirampungkan dan dipilih untuk mendukung SPERCO dikarenakan data tabel pada Gambar 3. 12 Tabel *research* pendukung SPERCO yang telah dilakukan pencarian informasi lebih lanjut.

SPERCO

Name	Tags	More Information	Category
Diart	Done	- Akurasi tidak konsisten pada audio dengan multiple speakers - Performa menurun pada audio dengan noise tinggi	Library
Riva by Nvidia	Done	Kinerja kurang optimal pada CPU-only, membutuhkan perangkat keras NVIDIA untuk performa terbaik.	Library
SpeechBrain	Done	- Kesulitan dalam mendeteksi perubahan pembicara (speaker change) yang akurat2 - Performa kurang baik saat menangani overlapping speech (ketika beberapa orang berbicara bersamaan) - Kurangnya dokumentasi terkait kasus tertentu	Library
Assemblyai (Premium)	Premium	Hanya tersedia dalam versi premium	API
Kaldi	Done	- Performa kurang optimal untuk aksen bahasa tertentu - Tidak tersedia sebagai sistem ASR siap pakai (out-of-the-box)	Library
OpenVINO	Done	- Kesulitan dalam implementasi real-time processing - Dokumentasi yang masing sangat kurang	Toolkit
Watson by IBM	Premium	Hanya tersedia dalam versi premium	Library
Whisper OPENAI	Done	- Akurasi yang sangat baik dari pada library yang lain - Kemampuan multi-bahasa yang superior - Model pre-trained yang berkualitas tinggi	Model
faster_whisper	Support Whisper	library utama yang digunakan untuk menjalankan model Whisper dengan optimasi performa, biasanya lebih cepat dibandingkan implementasi asli	Library
torch	Support Whisper	bagai kerangka kerja pembelajaran mendalam. `faster_whisper` juga memanfaatkan PyTorch untuk inferensi model	Library

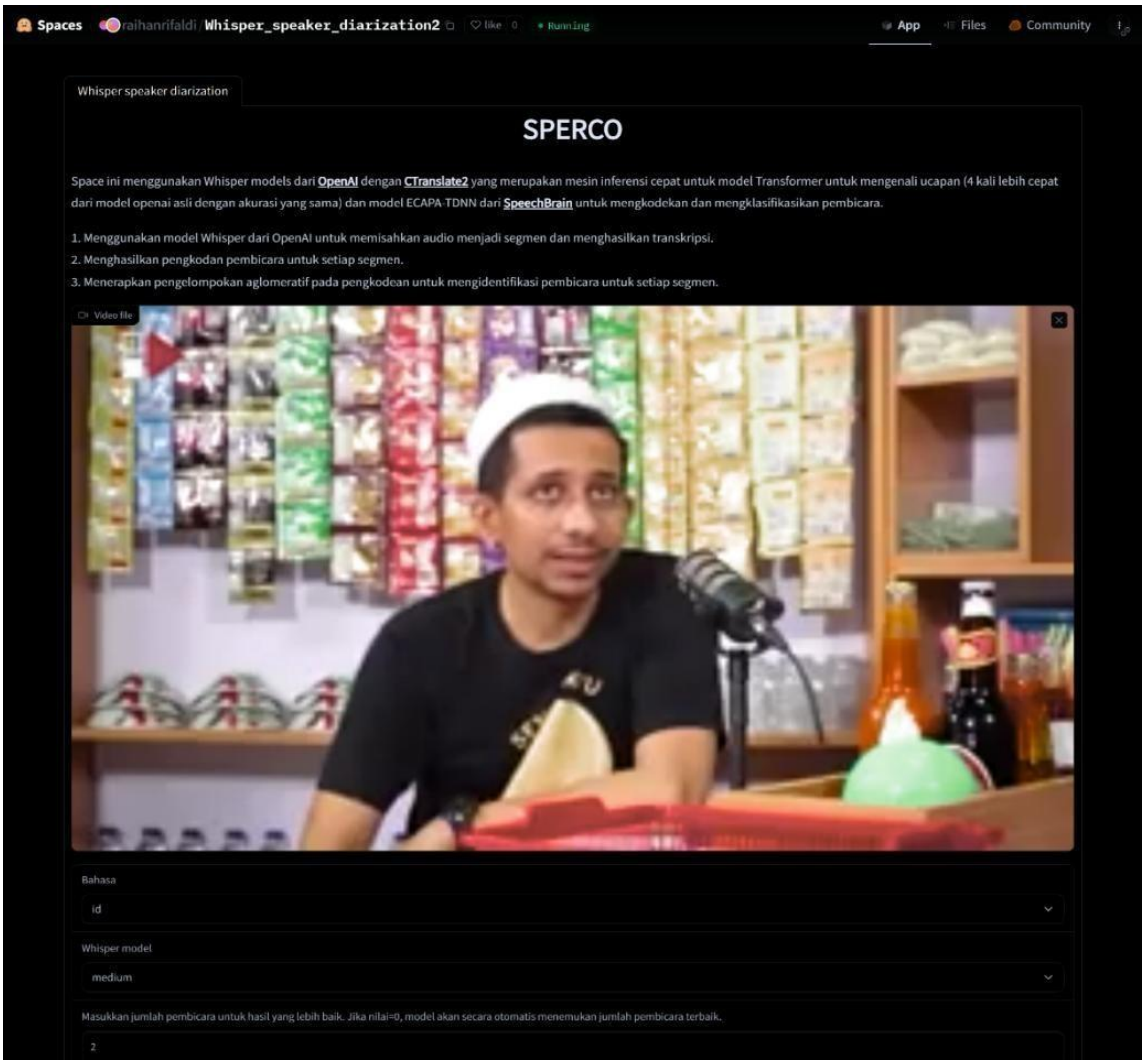
Gambar 3. 12 Tabel *research* pendukung SPERCO

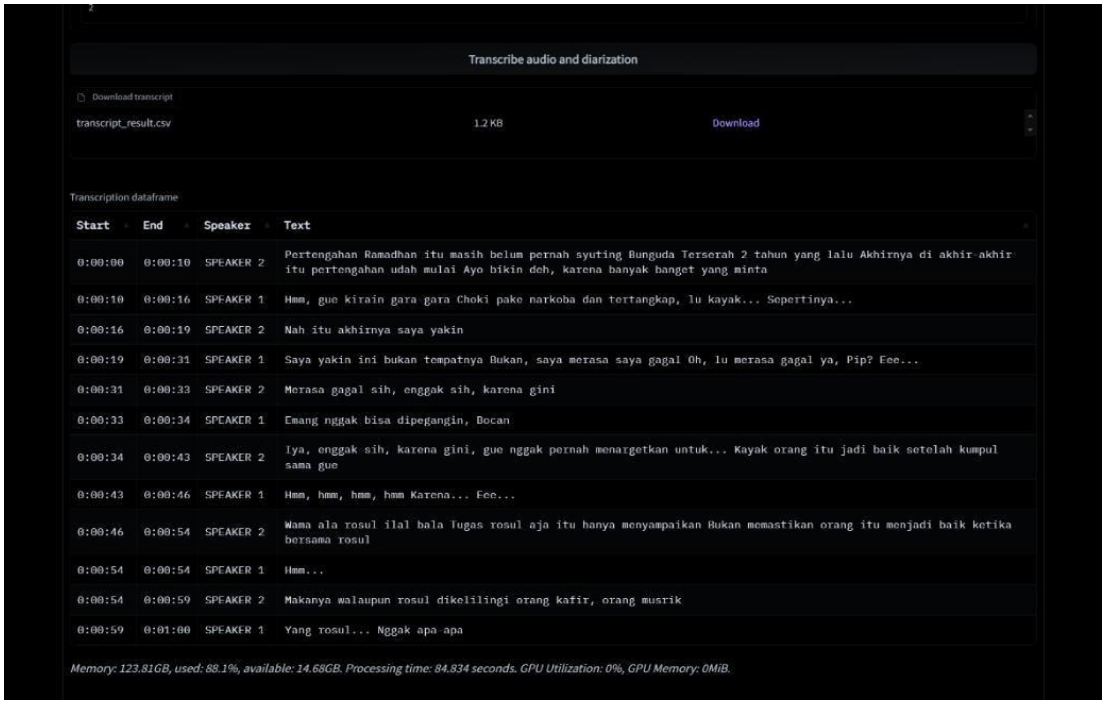
Pada tabel Gambar 3. 12 Tabel *research* pendukung SPERCO menjukkan Whisper sebagai teknologi ini memiliki sejumlah keunggulan signifikan, seperti tingkat presisi yang sangat baik dibandingkan opsi lainnya, dukungan untuk banyak bahasa, kualitas model pra-latih yang mumpuni dan banyaknya sampel dokumentasi ketika melakukan uji coba. Selain itu, pustaka tambahan seperti *faster_whisper* dapat digunakan untuk mengoptimalkan kinerja Whisper, terutama dalam hal kecepatan pengolahan data. Penggunaan Whisper sangat relevan dengan tema skripsi ini yang berfokus pada pengolahan audio, seperti sistem transkripsi otomatis, pengenalan bahasa secara *real-time,* atau aplikasi yang tetap efektif di lingkungan

dengan gangguan suara. Jika dibandingkan dengan pustaka lain. Hal ini menjadi Whisper sebagai model utama yang diguanakan dalam pengembangan SPERCO.

3.2 Implementasi dan Pengujian Aplikasi

Implementasi dan pengujian aplikasi merupakan tahap krusial dalam siklus pengembangan yang memastikan bahwa aplikasi yang dibuat dapat berjalan sesuai dengan rancangan dan memenuhi kebutuhan pengguna. Pada tahap ini, kode program diimplementasikan pada hugging face seperti pada Gambar 3. 13 Implementasi SPERCO di Hugging Face



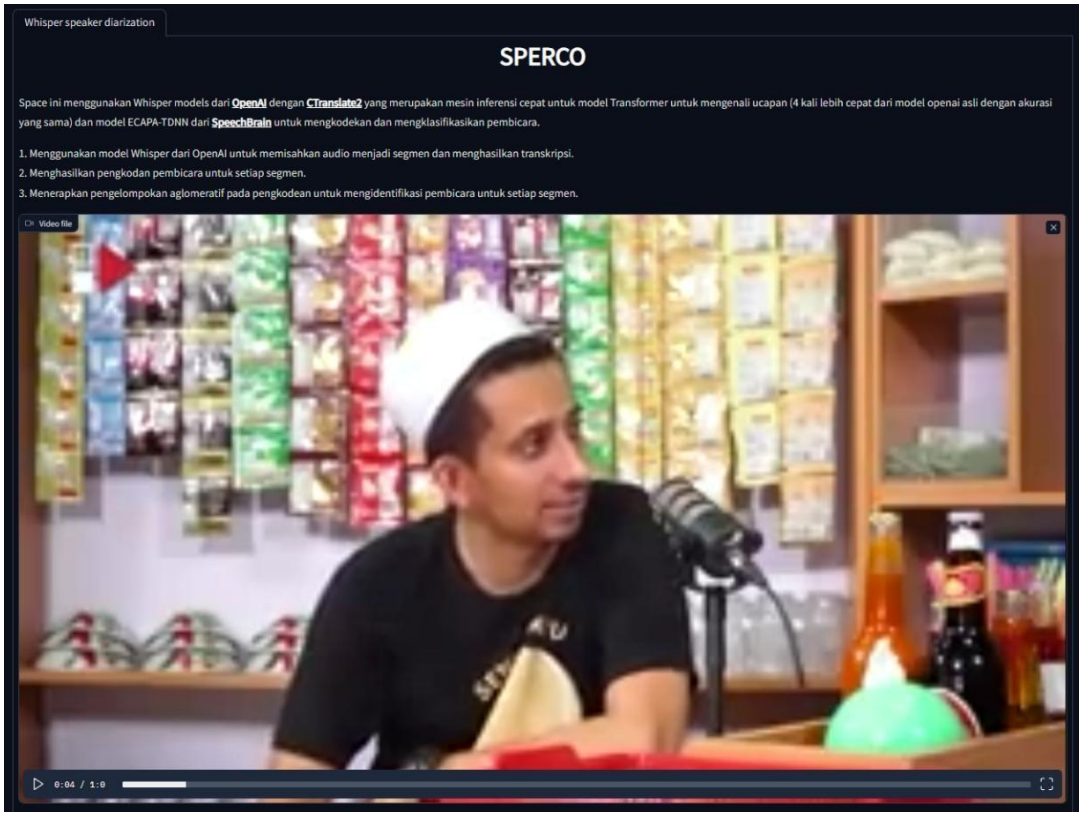


Gambar 3. 13 Implementasi SPERCO di Hugging Face

Gambar 3.13 memperlihatkan tampilan antarmuka aplikasi SPERCO yang telah berhasil diimplementasikan dan dijalankan melalui platform Hugging Face. Pada gambar tersebut, ditampilkan proses transkripsi otomatis dari sebuah cuplikan video yang dimuat ke dalam aplikasi, lengkap dengan hasil teks transkripsi dan identifikasi pembicara secara real-time. Antarmuka ini dirancang agar pengguna dapat langsung mengunggah video/audio, memilih model dan bahasa yang diinginkan, serta mengakses hasil transkripsi yang disajikan dalam format tabel yang mudah dibaca.

Ketika program telah berhasil di *deploy* dan di *push* melalui Visual Studio Code. Selanjutnya masuk pada tahap penggunaanya. Pada sampel kali ini penulis menggunakan video *podcast* berjudul berjudul “PWK - Ternyata Cerita Masa Kecil Habib Ja'far Kocak Banget”

dengan mengambil salah satu cuplikasi berdurasi satu menit dengan jumlah dua pembicara yang dapat dilihat pada gambar dibawah ini.



Gambar 3. 14 *Video podcast* Youtube

Dapat dilihat cuplikan pada Gambar 3. 14 *Video podcast* Youtubedari Podcast ini, berfokus pada dialog pembicaraan dan kualitas audio yang digunakan cukup baik sehingga dapat digunakan sebagai sampel dalam penggunaan pada aplikasi SPERCO. Langkah selanjutnya yaitu memilih bahasa yang akan digunakan. Pada proyek ini tersedia dalam 4 (Empat) bahasa yaitu Indonesia, Inggris, Jawa dan Sunda. Ketika telah memilih bahasa yang diinginkan, selanjutnya memilih model Whisper yang akan digunakan seperti yang tertera pada Gambar 3.15 *Whisper Models.*

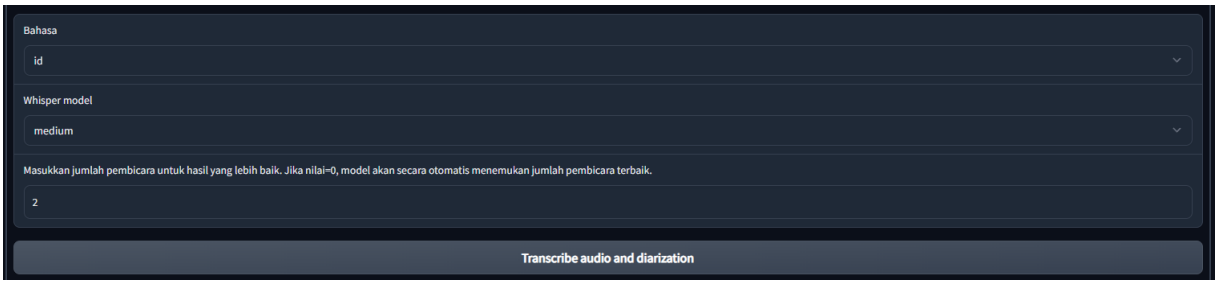


Gambar 3. 15 *Whisper Models*

Pemilihan *Whisper models* seperti gambar diatas berperan penting ketika akan digunakan. Pemilihan ini bertujuan untuk mementukan keseimbangan antara akurasi, kecepatan dan

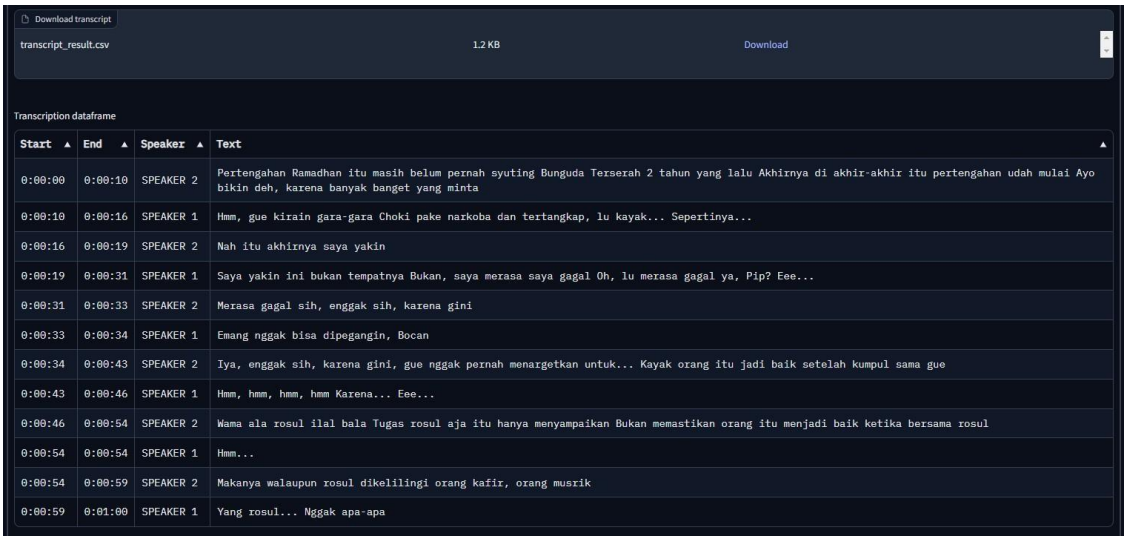
kebutuhan perangkat. Untuk aplikasi yang membutuhkan presisi tinggi, model besar seperti *large* atau *medium* ideal. Namun, untuk efisiensi dan kecepatan, model yang lebih kecil seperti *tiny* atau *base* lebih relevan. Performa Whisper sangat bervariasi bergantung pada bahasanya. Penjelasan lebih lengkap dapat dilihat pada Tabel 3. 3. Penyesuaian ini harus dilakukan berdasarkan kebutuhan dari file yang diunggah di SPERCO.

Setelah memilih model Whisper, langkah berikutnya adalah menentukan jumlah pembicara dalam video yang akan ditranskripsi. Proses ini bertujuan untuk menghasilkan transkripsi yang lebih akurat, sebagaimana diperlihatkan pada Gambar 3.16. Jika pengguna tidak mengetahui jumlah pembicara dalam video yang diunggah, angka 0 dapat dimasukkan, sehingga SPERCO secara otomatis akan mengidentifikasi jumlah pembicara yang paling sesuai.



Gambar 3. 16 Pemilihan jumlah pembicara

Setelah semua kolom yang diperlukan terisi, pengguna dapat menekan tombol "*Transcribe audio and diarization*". Hasil transkripsi akan ditampilkan pada halaman SPERCO. Jika dibutuhkan dalam format lain, hasil tersebut juga tersedia untuk diunduh dalam bentuk file *CSV*. Pada halaman ini, pengguna dapat melihat data transkripsi dalam bentuk dataframe, yang mencakup waktu awal dan akhir dialog, identitas pembicara serta isi dialog yang terekam, sebagaimana ditunjukkan pada Gambar 3.17 tentang data dialog pembicara


Download transcript

transcript_result.csv 1.2 KB Download

Transcription dataframe

Start	End	Speaker	Text
0:00:00	0:00:10	SPEAKER 2	Pertengahan Ramadhan itu masih belum pernah syuting Bunguda Terserah 2 tahun yang lalu Akhirnya di akhir-akhir itu pertengahan udah mulai Ayo bikin deh, karena banyak banget yang minta
0:00:10	0:00:16	SPEAKER 1	Hmm, gue kirain gara-gara Choki pake narkoba dan tertangkap, lu kayak... Sepertinya...
0:00:16	0:00:19	SPEAKER 2	Nah itu akhirnya saya yakin
0:00:19	0:00:31	SPEAKER 1	Saya yakin ini bukan tempatnya Bukan, saya merasa saya gagal Oh, lu merasa gagal ya, Pip? Eee...
0:00:31	0:00:33	SPEAKER 2	Merasa gagal sih, enggak sih, karena gini
0:00:33	0:00:34	SPEAKER 1	Emang nggak bisa dipegangin, Bocan
0:00:34	0:00:43	SPEAKER 2	Iya, enggak sih, karena gini, gue nggak pernah menargetkan untuk... Kayak orang itu jadi baik setelah kumpul sama gue
0:00:43	0:00:46	SPEAKER 1	Hmm, hmm, hmm, hmm Karena... Eee...
0:00:46	0:00:54	SPEAKER 2	Wama ala rosul ilal bala Tugas rosul aja itu hanya menyampaikan Bukan memastikan orang itu menjadi baik ketika bersama rosul
0:00:54	0:00:54	SPEAKER 1	Hmm...
0:00:54	0:00:59	SPEAKER 2	Makanya walaupun rosul dikelilingi orang kafir, orang musrik
0:00:59	0:01:00	SPEAKER 1	Yang rosul... Nggak apa-apa

Gambar 3. 17 Data dialog Pembicara

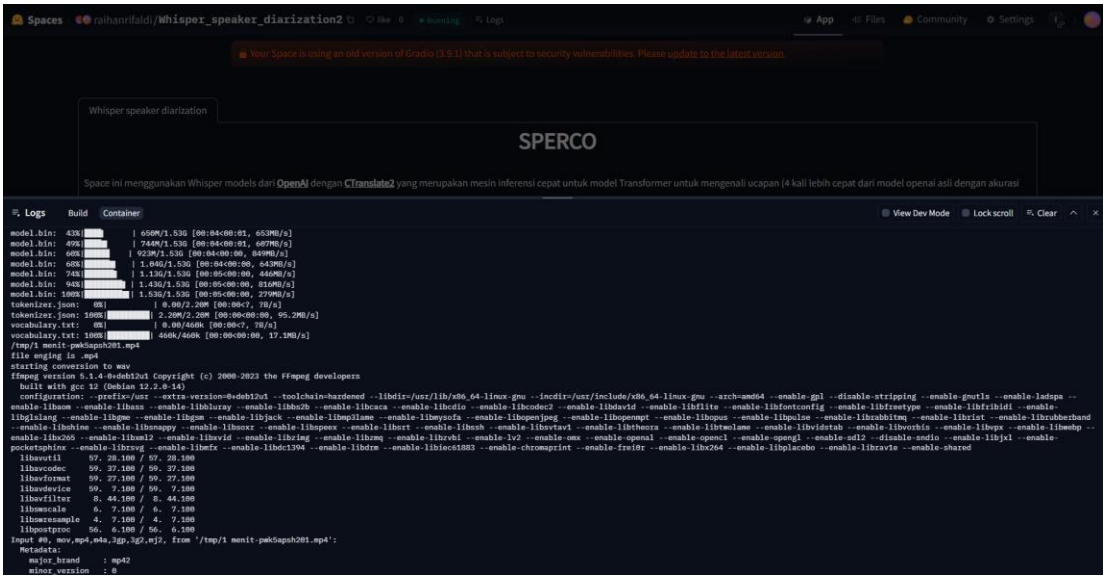
Melalui tampilan data transkripsi pada Gambar 3. 17 Data dialog Pembicara, pengguna dapat dengan mudah menelusuri hasil percakapan yang telah diuraikan berdasarkan waktu, pembicara serta isi percakapan secara runtut. Fitur ini sangat membantu dalam proses analisis percakapan, terutama dalam konteks pengolahan data audio untuk keperluan internal perusahaan. Informasi yang ditampilkan juga dapat dimanfaatkan untuk berbagai kebutuhan lanjutan, seperti pelabelan data, pembuatan ringkasan isi percakapan atau pelatihan model lanjutan. Keakuratan identifikasi waktu serta pembicara yang terekam menjadikan SPERCO sebagai alat bantu yang andal dalam proses transkripsi otomatis berbasis AI.

3.2.1 Analisis Aktivitas melalui *Log Container*

Pada Gambar 3.18 tersebut menunjukkan tampilan *Log Container* dari SPERCO yang di-*host* dari Spaces Hugging Face. *Log* ini mencatat aktivitas yang dilakukan oleh aplikasi atau *model* seperti pemuatan file, proses *decoding* dan penggunaan sumber daya. *Log* juga mencatat konfigurasi teknis seperti pustaka yang digunakan, pengaturan *model* dan informasi terkait.

Gambar 3. 18 *Log Container* dari SPERCO

Pada Gambar 3. 18 *Log Container* dari SPERCOmenunjukan bahwa *Log container* memiliki beberapa kelebihan yang sangat berguna dalam pengelolaan aplikasi. Pertama, pemantauan *real-time* memungkinkan akses langsung ke aktivitas aplikasi yang sedang berjalan, memberikan informasi terkini mengenai operasionalnya. Kedua, *log* dapat mendeteksi *error* dengan cepat, sehingga developer bisa langsung melihat kesalahan beserta penyebabnya tanpa harus membuka kode sumber. Ketiga, *log* juga memudahkan dalam reproduksi masalah, karena memungkinkan pelacakan kembali untuk mengetahui kapan dan di mana kesalahan tersebut terjadi. Terakhir, *log container* berperan dalam analisis kinerja



aplikasi, dengan memberikan wawasan tentang kebutuhan sumber daya yang diperlukan, sehingga aplikasi bisa dioptimalkan lebih efisien.

3.2.2 Hasil Pengujian Akurasi Transkripsi

Pengujian tingkat akurasi dilakukan menggunakan pendekatan *Word Error Rate* (WER). Metode WER merupakan alat ukur yang digunakan untuk menilai kualitas hasil transkripsi yang dihasilkan oleh sistem *Automatic Speech Recognition* (ASR). Dalam berbagai penerapan,

sering kali diperlukan estimasi WER berdasarkan pasangan data berupa rekaman suara dan hasil transkripsinya.

Uji coba adaptasi berbasis bahasa dengan berbagai dialek China menunjukkan bahwa penerapan pendekatan *speech-based in-context learning* (SICL) pada sistem pengenalan suara otomatis (ASR) untuk identifikasi kata secara individual berhasil mengurangi tingkat kesalahan transkrip (WER). Hasil ini divalidasi melalui pengujian pada tugas penyesuaian pembicara dan pengenalan ucapan berkesinambungan, di mana keduanya menunjukkan penurunan relatif WER yang cukup signifikan (S. Wang et al., 2023). Penelitan ini membatu pengujian akurasi transkripsi pada proyek ini.

Penelitian sebelumnya tentang estimasi WER berfokus pada membangun model yang dilatih dengan mempertimbangkan sistem ASR tertentu salah satunya Whisper. Rumus untuk WER sebagai berikut:

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

Keterangan:

• Substitusi (S): Jumlah kata yang salah ditranskripsi.

• Penyisipan (I): Jumlah kata tambahan yang tidak ada dalam referensi.

• Penghapusan (D): Jumlah kata yang hilang dari referensi.

WER = 0 menunjukkan transkripsi sempurna, sedangkan nilai lebih tinggi menunjukkan tingkat kesalahan yang lebih besar. Untuk mengetahui nilai presentasi akurasi transkripsi dapat dihitung dengan rumus berikut:

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

Proses pengujian ini dilakukan menggunakan model Whisper tipe *Medium* dengan sampel video podcast berjudul “PWK - Ternyata Cerita Masa Kecil Habib Ja'far Kocak Banget,” menggunakan cuplikan berdurasi 1 menit. Setelah itu, hasil transkripsi dibandingkan dengan transkripsi manual (*ground truth*) untuk melihat tingkat kesesuaiannya. Perbandingan dilakukan dengan menghitung *Word Error Rate* (WER) yaitu jumlah kata yang salah, hilang, atau berlebih. Nilai WER kemudian dikonversi menjadi persentase akurasi. Pengujian

mencakup tiga kategori kualitas audio yaitu Sangat Baik, Baik dan Kurang Baik dengan rincian sebagai berikut:

- Sangat Baik: Audio sangat jernih, tidak ada *noise*, suara pembicara jelas dan tidak tumpang tindih. Memiliki WER < 0.10 dan akurasi ≥ 90%.
- Baik: Audio cukup jelas, mungkin ada sedikit *noise*, tapi masih mudah dimengerti. Memiliki WER antara 0.10–0.20 dan akurasi 80–89%
- Kurang Baik: Terdapat gangguan seperti *noise,* suara lingkungan atau *overlapping* antar pembicara. Memiliki WER > 0.20 dan akurasi < 80%.

Melalui pengelompokan ini, evaluasi sistem transkripsi menjadi lebih komprehensif, karena tidak hanya menilai hasil akhir, tetapi juga mempertimbangkan faktor-faktor lainnya yang memengaruhi performa model. Pendekatan ini juga mencerminkan kondisi nyata yang kemungkinan akan dihadapi oleh aplikasi SPERCO saat digunakan dalam berbagai skenario percakapan, baik dalam lingkungan yang terkendali maupun yang penuh gangguan suara. Oleh karena itu, hasil dari pengujian WER ini menjadi referensi penting dalam menentukan kesiapan sistem untuk digunakan secara operasional.

Berdasarkan hasil pengujian yang telah dilakukan terhadap model Whisper menggunakan metode *Ground Truth*, diperoleh data numerik yang menggambarkan performa sistem dalam mengenali dan mentranskripsi percakapan audio. Untuk memberikan gambaran yang lebih rinci dan terukur, hasil pengujian ini disusun dalam bentuk tabel. Tabel tersebut menyajikan perbandingan antara transkrip referensi (*ground truth*) dan hasil transkripsi dari SPERCO yang dilengkapi dengan nilai WER (*Word Error Rate*), akurasi transkripsi dalam bentuk persentase serta klasifikasi kualitas audio yang diuji. Rincian lengkap dari pengujian ini dapat dilihat pada Tabel 3. 4 sebagai berikut

Tabel 3. 4 Pengujian akurasi menggunakan Metode *Ground Truth*

No	Ground Truth	Transkrip SPERCO	WER	Akurasi(%)	Kualitas Audio
1	[0:00:00-0:00:10] **Pembicara 2**: Pertengahan Ramadhan itu masih belum pernah syuting pemuda tersesat 2 tahun yang lalu Akhirnya di akhir-akhir itu pertengahan udah mulai Ayo bikin deh, karena banyak banget yang minta	[0:00:00-0:00:10] **Pembicara 2**: Pertengahan Ramadhan itu masih belum pernah syuting Bunguda Terserah 2 tahun yang lalu Akhirnya di akhir-akhir itu pertengahan udah mulai Ayo bikin deh, karena banyak banget yang minta	0,07	93%	Sangat Baik
2	[:00:10-0:00:16] **Pembicara 1**: Hmm, gue kirain gara-gara Choki pake narkoba dan tertangkap, lu kek... Sepertinya...	[:00:10-0:00:16] **Pembicara 1**: Hmm, gue kirain gara-gara Choki pake narkoba dan tertangkap, lu kayak... Sepertinya...	0.14	86%	Baik
3	[0:00:16-0:00:19] **Pembicara 2**: Hehe.. nah itu akhirnya saya yakin.	[0:00:16-0:00:19] **Pembicara 2**: Nah itu akhirnya saya yakin.	0,16	84%	Baik

4	[0:00:19-0:00:31] **Pembicara 2** : Saya yakin ini bukan tempatnya. Bukan, saya merasa ah saya gagal. **Pembicara 1**: Oh, lu, lu merasa gagal ya, Bib? Eee...	[0:00:19-0:00:31] **Pembicara 1**: Saya yakin ini bukan tempatnya. Bukan, saya merasa saya gagal. Oh, lu merasa gagal ya, Pip? Eee...	0,26	74%	Kurang Baik
5	[0:00:31-0:00:33] **Pembicara 2**: Eee merasa gagal sih, enggak sih, karena gini.	[0:00:31-0:00:33] **Pembicara 2**: Merasa gagal sih, enggak sih, karena gini.	0,12	87,5%	Baik
6	[0:00:33-0:00:34] **Pembicara 1**: Emang nggak bisa dipegangin, Bocah	[0:00:33-0:00:34] **Pembicara 1**: Emang nggak bisa dipegangin, Bocan.	0,8	80%	Baik
7	[0:00:34-0:00:43] **Pembicara 2**: Iya, enggak sih, karena gini, gue nggak pernah menargetkan untuk... Kayak orang itu jadi baik setelah kumpul sama	[0:00:34-0:00:43] **Pembicara 2**: Iya, enggak sih, karena gini, gue nggak pernah menargetkan untuk... Kayak orang itu jadi baik setelah kumpul sama gue.	0	100%	Sangat Baik

	gue				
8	[0:00:43-0:00:46] **Pembicara 2**: Hmm, hmm, hmm, hmm. Karena... Eee...	[0:00:43-0:00:46] **Pembicara 1**: Hmm, hmm, hmm, hmm. Karena... Eee...	0	100%	Sangat Baik
9	[0:00:46-0:00:54] **Pembicara 2**: Wama ala rosul ilal bala. Tugas rasul aja itu hanya menyampaikan. Bukan memastikan orang itu menjadi baik ketika bersama rosul.	[0:00:46-0:00:54] **Pembicara 2**: Wama ala rosul ilal bala. Tugas rosul aja itu hanya menyampaikan. Bukan memastikan orang itu menjadi baik ketika bersama rosul.	0,05	95%	Sangat Baik
10	[0:00:54-0:00:54] **Pembicara 1**: Hmm...	[0:00:54-0:00:54] **Pembicara 1**: Hmm...	0	100%	Sangat Baik
11	[0:00:54-0:00:59] **Pembicara 2**: Makanya walaupun rasul dikelilingi orang kafir, orang musyrik	[0:00:54-0:00:59] **Pembicara 2**: Makanya walaupun rosul dikelilingi orang kafir, orang musrik.	0,2	80%	Baik
12	[0:00:59-0:01:00] **Pembicara 2**: Ya rasul... Nggak apa-apa.	[0:00:59-0:01:00] **Pembicara 1**: Yang rosul... Nggak apa-apa.	0,2	80%	Baik

Berdasarkan pengujian pada Tabel 3. 4 menggunakan *Word Error Rate* (WER) dan metode *Ground Truth*, disimpulkan bahwa kualitas audio secara signifikan menentukan tingkat akurasi transkripsi otomatis. Berdasarkan pengujian yang telah dilakukan, audio dengan kategori "Sangat Baik" mendapatkan rekaman jelas tanpa gangguan suara atau tumpang tindih pembicara mampu mencapai akurasi hingga optimal dengan nilai WER di bawah 0,10. Sementara itu, kategori "Baik", meskipun terdapat sedikit *noise* atau ketidaksempurnaan audio, masih mampu mempertahankan akurasi tinggi di kisaran 80–89%, dengan WER antara 0,10 hingga 0,20. Adapun kategori "Kurang Baik", yang dicirikan oleh *noise* lingkungan yang tinggi atau tumpang tindih pembicaraan, menghasilkan akurasi yang relatif rendah di bawah 80% dengan nilai WER lebih besar dari 0,2.

Temuan ini sejalan dengan penelitian dalam jurnal TED-LIUM versi 3 yang menekankan bahwa kualitas dan jumlah data audio sangat berpengaruh pada performa sistem transkripsi otomatis. Penelitian tersebut secara khusus menunjukkan bahwa peningkatan jumlah data pelatihan audio yang berkualitas tinggi mampu secara signifikan, serta meningkatkan kemampuan adaptasi sistem terhadap beragam karakteristik suara dan pembicara

3.3 Pembuatan Dokumentasi & Penutupan Proyek

Proses pembuatan dokumentasi proyek menjadi bagian penting dalam pengembangan SPERCO, khususnya karena aplikasi ini akan terus dikembangkan dan digunakan secara internal di Telkomsel. Dokumentasi ini dibuat sebagai panduan teknis yang dapat memudahkan pemahaman dan penerusan pekerjaan bagi tim pengembang berikutnya. Penulis memulai tahap dokumentasi dengan mengumpulkan informasi lengkap mulai dari proses awal hingga hasil akhir implementasi aplikasi. Informasi ini mencakup deskripsi sistem, tujuan pengembangan, teknologi yang digunakan seperti Whisper AI dan platform Hugging Face serta detail fitur utama yang telah berhasil dikembangkan. Selain itu, dokumentasi teknis mencakup tata cara penggunaan aplikasi, panduan instalasi dan konfigurasi sistem serta *troubleshooting* untuk kendala umum yang mungkin muncul. Setiap tahapan pengembangan, terutama proses *debugging* dan fitur.

Penulis telah menyerahkan laporan berupa *soft file* dan *hard file* kepada Telkomsel sebagai dokumentasi untuk mempermudah pengelolaan. Proyek ini akan terus berlanjut, dan aplikasi dikembangkan sesuai kebutuhan perusahaan. Penulis telah menyelesaikan versi awal proyek. Dokumentasi ini memudahkan pengembangan lanjutan serta membantu pemeliharaan agar SPERCO tetap andal dalam operasional internal Perusahaan.

BAB IV
REFLEKSI PELAKSANAAN MAGANG

4.1 Relevansi Akademik

Dalam pelaksanaan magang, terutama dalam proyek Pengembangan Fitur Speech Recognition pada aplikasi notulensi menggunakan Whisper AI, penulis melihat adanya kesinambungan yang erat dengan pembelajaran yang telah diperoleh di bangku perkuliahan, khususnya pada tiga aspek utama: teknologi yang digunakan, manajemen proyek serta metode pengembangan perangkat lunak.

Teknologi yang diterapkan dalam proyek ini berakar pada konsep-konsep yang diajarkan dalam mata kuliah seperti *Deep Learning*, yang mencakup dasar-dasar *Machine Learning*, kecerdasan buatan (AI) serta kegiatan praktis lainnya yang relevan. Namun, jika dibandingkan, pendekatan pembelajaran di perkuliahan lebih berfokus pada pengenalan dasar teknologi, analisis data dan prinsip kerjanya. Pengetahuan dasar tersebut membantu bagi penulis untuk memulai proyek ini. Dalam konteks proyek ini, penerapan teknologi menjadi lebih cepat dipelajari dan diadaptasikan oleh penulis dikarenakan dasar dan fondasi yang telah didapatkan ketika masa perkuliahan. Terkadang, teori yang dipelajari di bangku kuliah dan kenyataan di lapangan dapat sangat berbeda, menghadirkan tantangan tersendiri dalam proses pengerjaan proyek. Namun, dengan pendekatan yang tepat dan kemampuan untuk beradaptasi, semua perbedaan tersebut dapat diatasi dan disesuaikan dengan kebutuhan proyek yang sedang berlangsung.

Teknologi pengenalan suara dalam proyek ini menggunakan *Speech Recognition* dengan fitur *Speaker Diarization* yang diterapkan melalui *pyannote.audio* dan model Whisper. Teknologi ini mampu mengidentifikasi siapa yang berbicara dan kapan sehingga membantu meningkatkan akurasi transkripsi dengan memisahkan suara dari berbagai pembicara. Walaupun *pyannote.audio* memiliki performa yang andal dalam melakukan *diarization*, terdapat sejumlah tantangan yang perlu diatasi, seperti integrasi dengan sistem yang sudah ada dan penyesuaian model Whisper agar sesuai dengan situasi tertentu, misalnya rapat perusahaan dengan beberapa karyawan. Proses ini memerlukan pengujian dan modifikasi berulang yang kemudian diterapkan menggunakan platform Hugging Face.

Pada mata kuliah Manajemen Diri, penulis mempelajari pentingnya pengelolaan waktu dan komunikasi yang efektif sebagai dasar dalam menyelesaikan tugas serta bekerja sama dalam tim. Kedua keterampilan ini menjadi sebuah tantangan dalam memastikan

keberlangsungan proyek sekaligus mencapai target secara efisien. Ketika melaksanakan magang, penerapan pengelolaan waktu dan komunikasi yang baik sangat diperlukan, terutama dalam koordinasi dengan tim pengembang dan pihak terkait dalam proyek.

Metode pengembangan yang diterapkan dalam proyek ini adalah Scrum. Meskipun metode ini telah diperkenalkan di Perkuliahan, implementasinya dalam proyek SPERCO memberikan pemahaman yang lebih dapat diaplikasikan. Penulis mengalami secara langsung proses pembentukan *backlog*, pelaksanaan *sprint* serta review hasil proyek. Dalam praktiknya, Scrum memberikan fleksibilitas dalam menyusun prioritas dan pembagian tugas namun juga membutuhkan keterampilan manajemen waktu dan komunikasi yang baik agar proyek tetap berjalan sesuai target. Namun kenyataan di Lapangan sering kali menghadirkan tantangan berupa tekanan waktu, hal tidak terduka dan dinamika tim yang dapat memengaruhi efektivitas implementasi keterampilan tersebut.

4.2 Pembelajaran Magang

Magang menjadi salah satu bagian penting dalam perjalanan pendidikan seorang mahasiswa, yang berfungsi untuk mengasah kemampuan yang diperlukan di dunia kerja. Program ini tidak hanya berfungsi sebagai wadah pembelajaran, tetapi juga sebagai langkah persiapan menyeluruh untuk menghadapi berbagai tantangan karier di masa mendatang. Dalam kegiatan ini, mahasiswa diberi peluang untuk menerapkan teori-teori yang telah dipelajari di ruang kuliah ke dalam situasi nyata, sehingga mampu memperkuat pemahaman sekaligus meningkatkan keterampilan yang bersifat praktis.

a. Penguasaan Teknologi *Speech Recognition* dan Platform AI

Sebelum memasuki masa magang, penulis telah memperoleh pemahaman dasar mengenai pengolahan data dan audio dari mata kuliah yang telah ditempuh. Namun, dalam pelaksanaannya di proyek SPERCO, tantangan yang dihadapi jauh lebih kompleks. Penulis dipercaya untuk menangani pengembangan backend sistem notulensi otomatis, khususnya dalam menerapkan teknologi Whisper AI untuk transkripsi dan *pyannote.audio* untuk fitur pemisahan pembicara (*speaker diarization*).

Kendala utama yang dihadapi ialah kualitas audio yang bervariasi seperti mengandung noise tinggi, tumpang tindih suara antar pembicara dan mengatur jumlah pembicara dalam model agar transkripsi tetap akurat. Selain itu, terdapat keterbatasan dokumentasi terutama untuk modul pyannote.audio serta keterbatasan sumber daya pada versi gratis Hugging Face.

Untuk mengatasi hal tersebut, penulis melakukan pendekatan iteratif dalam menguji berbagai konfigurasi model Whisper (seperti ukuran *base, small* hingga *medium*) dan membandingkan hasilnya menggunakan metrik *Word Error Rate* (WER). Selain itu, penulis menggali referensi teknis dari komunitas *open-source* dan eksperimen mandiri untuk menyesuaikan model pengolahan suara dengan kebutuhan sistem. Tantangan minimnya dokumentasi disiasati dengan menelusuri diskusi GitHub, forum komunitas dan membaca kode sumber pustaka secara langsung.

Penulis juga memanfaatkan platform Hugging Face untuk menyusun model dan mengintegrasikannya dengan antarmuka pengguna berbasis Framework Gradio. Visual Studio Code digunakan bersama kontrol versi *git* untuk memastikan pengembangan dilakukan secara terstruktur. Meskipun banyak kendala teknis yang dihadapi, proses ini justru memperdalam pemahaman penulis mengenai SPERCO dan kemampuan untuk berpikir kritis dalam menyusun solusi.

b. Menerapkan Scrum dalam Pengelolaan Proyek

Dalam proyek SPERCO, tim pengembang menerapkan metodologi Scrum untuk mengelola pekerjaan secara bertahap dan terstruktur. Penulis terlibat langsung dalam penyusunan *product backlog* dan *sprint backlog*, rapat rutin proyek untuk memantau progres, menyampaikan kendala dan berdiskusi teknis bersama pembimbing.

Salah satu kendala ialah ketika terjadi perubahan prioritas fitur di tengah sprint yang sedang berjalan. Hal ini menuntut penulis untuk cepat beradaptasi terhadap perubahan rencana kerja dan mengatur ulang waktu pengerjaan agar beban kerja tidak menumpuk. Selain itu, ketika beberapa anggota tim memiliki perbedaan ritme kerja atau keterbatasan waktu, koordinasi menjadi kurang optimal dan dapat memengaruhi kelancaran *sprint* secara keseluruhan.

Untuk menyikapi hal ini, penulis berinisiatif untuk komunikasi dengan anggota dan melakukan penyesuaian ritme team. Penulis juga melihat perubahan prioritas dan membuat *timeline* mandiri agar tetap fokus pada target yang telah disepakati bersama. Pendekatan Scrum memberikan pelajran bahwa fleksibilitas dalam manajemen proyek tidak berarti tanpa arah, justru di balik kerangka kerja yang adaptif tersebut, diperlukan kedisiplinan dalam komunikasi, pencatatan dan penyesuaian strategi. Pengalaman ini menjadi bekal penting dalam memahami dinamika kerja tim di lingkungan pengembangan perangkat lunak yang nyata walaupun banyak perubahan rencana yang tidak terduga ketika di lapangan.

4.2.1 Teknis

Dalam pengembangan SPERCO di Telkomsel, keterampilan teknis yang diperoleh mencakup berbagai aspek teknologi yang kompleks. SPERCO dikembangkan dengan teknologi pengenal suara atau biasa disebut *Speech Recognition* dengan fitur *Speaker Diarizaition* yang diimplementasi pada Hugging Face dengan tujuan kemudahan akses dan pengembangan. Pada aplikasi SPERCO yang melibatkan pemrosesan *Speech Recognition*, Model Whisper AI dan Pustaka yang digunakan membutuh sumber daya yang cukup besar oleh karena itu dapat disesuaikan dengan layanan gratis dari Hugging Face.

Sebelumnya, proyek ini dikembangkan pada *localhost* yang membutuhkan waktu yang cukup lama dan perangkat keras yang memadai. Terdapat kendala dalam implementasi proyek dalam *localhost* seperti kesulitan dalam melakukan konfigurasi dan sedikitnya dokumentasi penunjang akan perihal tersebut. Proyek ini diambil dari *framework* yang tersedia di Hugging Face lalu disesuaikan dengan kebutuhan aplikasi notulensi yang telah dikembangkan.

Selain itu, implementasi Whisper pada aplikasi notulensi ini merupakan teknologi yang baru sehingga menjadi sebuah tantangan dikarenakan masih sedikitnya dokumentasi dan penerapan teknologi pada beberapa proyek serupa terutama di Hugging face serta teknologi yang masih belum sempura ketika penulis melakukan pengembangan proyek ini. Model Whisper digunakan untuk melakukan transkripsi audio menjadi teks, yang meningkatkan efisiensi dan akurasi pencatatan rapat serta mengurangi *human error*.

Selanjutnya adalah penggunaan Pustaka *Pyannote.audio* pada fitur *speaker diarization* dengan tujuan memisahkan suara pembicara yang berbeda dalam rekaman audio. Teknik ini memungkinkan identifikasi dan pemisahan segmen-segmen yang diucapkan oleh pembicara yang berbeda, sehingga meningkatkan akurasi transkripsi dan analisis lebih lanjut. *Pyannote.audio* dibantu dengan *framework* PyTorch yang dapat membantu dalam proses menentukan kapan *speaker* berbicara, sehingga dapat mengetahui siapa yang mengatakan apa dan kapan ia mengatakannya. Output akhir dari proses ini adalah transkrip teks yang dilengkapi dengan *timestamp* dan informasi pembicara. Format output menampilkan waktu mulai dan selesai untuk setiap kata yang diucapkan. Hasil ini memberikan transkrip yang terstruktur dengan penanda waktu yang akurat dan identifikasi pembicara yang jelas.

4.2.2 Non Teknis

Selain meningkatkan kemampuan teknis, program magang di Telkomsel juga memberikan kontribusi besar terhadap pengembangan keterampilan non-teknis yang esensial di dunia kerja. Selama menjalani magang, terdapat berbagai kesempatan untuk mengasah kemampuan berkomunikasi secara efektif dengan Karyawan, tim dan supervisor. Aktivitas seperti mengidentifikasi kebutuhan proyek, menyampaikan laporan kemajuan, hingga mengelola umpan balik menjadi bagian penting dalam proses ini.

Kemampuan mengatur waktu juga mengalami peningkatan, mengingat pentingnya menetapkan prioritas dan menyelesaikan tugas sesuai dengan batas waktu yang telah ditentukan. Selain itu, kerja sama tim menjadi aspek yang sangat ditekankan, di mana kolaborasi dalam tim pengembangan perangkat lunak, penyelesaian masalah bersama dan berbagi pengetahuan menjadi pengalaman berharga.

Keterampilan dalam menganalisis dan memecahkan masalah juga memainkan peran penting selama pengembangan proyek ini. Kemampuan ini terasah melalui identifikasi kendala yang muncul serta upaya menemukan solusi yang paling efektif. Di samping itu, kesempatan untuk memimpin sub-proyek kecil atau mengelola tugas tertentu dalam tim pengembangan memberikan pengalaman dalam memperkuat *leadership skill*.

Secara keseluruhan, pengalaman magang di Telkomsel tidak hanya mengasah kemampuan teknis dan non-teknis yang relevan dengan bidang informatika, tetapi juga memperkaya pemahaman tentang dinamika kerja di industri teknologi, memberikan bekal yang bernilai untuk memasuki dunia profesional.

BAB V
PENUTUP

5.1 Kesimpulan

Pengembangan aplikasi notulensi berbasis *Speech Recognition* (SPERCO) menggunakan Whisper AI dari Open AI telah selesai dikerjakan dan berhasil mencapai tujuan utama yaitu dapat digunakan sebagai alat bantu notulis dalam merangkum rapat secara otomatis berbasis suara yang digunakan pada tim internal di Telkomsel. Fitur utama dari SPECO yaitu *Speaker Diarization* yang berfungsi mengenali pembicara, pemisahan sinyal audio serta pengelompokan berdasarkan suara khas setiap pembicara. Teknologi ini mempermudah pembuatan catatan rapat tanpa campur tangan manual dan mendokumentasikan seluruh pembicaraan secara menyeluruh.

Aplikasi ini memiliki fitur utama *Speaker Diarization* untuk mengidentifikasi pembicara dan menghasilkan transkripsi yang lebih tepat serta meningkatkan produktivitas dengan mempermudah pembuatan catatan rapat tanpa campur tangan manual. Hasil pengujian akurasi menggunakan metode *Ground Truth* dan *Word Error Rate* (WER) menunjukkan bahwa SPERCO memberikan hasil transkripsi yang presisi dan cukup baik seperti yang tertera pada Tabel 3.2. Teknologi ini mempermudah pembuatan catatan rapat tanpa campur tangan manual dan mendokumentasikan seluruh pembicaraan secara menyeluruh sehingga evaluasi perusahaan dapat dilakukan secepatnya. Selain itu, aplikasi ini juga memfasilitasi pemantauan operasional melalui *Log container*, yang membantu mendeteksi error dan melacak masalah yang terjadi pada aplikasi.

5.2 Saran

Proyek pengembangan SPERCO sebagai sistem notulensi berbasis pengenalan suara telah menunjukkan potensi besar dalam membantu proses pencatatan rapat secara otomatis dan efisien. Meskipun sistem ini telah mencapai tahap implementasi awal dengan hasil uji yang cukup menjanjikan, masih terdapat ruang yang luas untuk peningkatan dan pengembangan lebih lanjut. Oleh karena itu, diperlukan strategi lanjutan agar aplikasi ini dapat mencapai performa optimal dan memberikan manfaat yang lebih luas bagi pengguna, terutama dalam konteks operasional perusahaan yang kompleks dan dinamis. Untuk pengembangan lebih lanjut ada beberapa hal yang dapat dipertimbangkan adalah:

1. Melakukan riset dan eksplorasi terhadap teknologi: terutama dalam hal peningkatan akurasi pada kondisi audio dengan kualitas rendah. Penggunaan model yang lebih baik serta pemanfaatan data latih yang mencakup lingkungan audio yang beragam akan membantu membuat model lebih adaptif terhadap berbagai variasi suara dan kondisi nyata.

2. Pemanfaatan teknologi *Noise Reduction*: SPERCO juga dapat ditingkatkan dengan integrasi teknologi pengurangan kebisingan (*noise reduction*) seperti *spectral gating* atau *adaptive filtering* guna meminimalisasi suara latar belakang yang mengganggu proses pengenalan suara. Selain itu, pengujian aplikasi perlu diperluas ke dalam berbagai skenario rapat, termasuk rapat formal, diskusi terbuka, dan lingkungan dengan kondisi akustik yang menantang, seperti ruangan bergema atau ramai.

3. Penting juga untuk mengumpulkan umpan balik: Data pengguna akhir terkait pengalaman penggunaan, akurasi transkripsi dan kegunaan sistem sangat penting untuk melakukan pemantauan pola penggunaan dari SPERCO untuk memperoleh kebutuhan dan ekspektasi pengguna yang lebih spesifik. Informasi ini akan sangat berguna untuk pengambilan keputusan dalam iterasi pengembangan berikutnya.

4. Dokumentasi Pengembangan: Aspek lain yang juga tidak kalah penting adalah dokumentasi proses pengembangan. Setiap tahapan, mulai dari rancangan sistem, struktur kode, hingga konfigurasi teknis yang dicatat secara rinci dan sistematis. Dokumentasi yang lengkap dan jelas akan memudahkan pengembang berikutnya dalam memahami alur kerja sistem yang sudah dibangun. Hal ini sangat berguna apabila terjadi pergantian anggota tim sehingga pengetahuan tidak hilang dan proses pengembangan dapat berlanjut tanpa hambatan.

Penulis juga ingin menekankan pentingnya pengembangan berkelanjutan terhadap proyek ini setelah masa magang berakhir. SPERCO merupakan Langkah awal yang untuk mendukung efisiensi operasional perusahaan. Oleh karena itu, proyek ini sangat disarankan untuk terus dikembangkan oleh tim internal Telkomsel maupun oleh pemagang selanjutnya. Dukungan terhadap dokumentasi teknis dan kolaborasi tim menjadi kunci utama agar transisi pengembangan berjalan lancar dan dapat menghasilkan sistem yang semakin matang di masa depan.

DAFTAR PUSTAKA

Hernandez, F., Nguyen, V., Ghannay, S., Tomashenko, N., & Estève, Y. (n.d.). TED-LIUM 3: Twice as Much Data and Corpus Repartition for Experiments on Speaker Adaptation. https://www.ubiqus.comhttps://lium.univ-lemans.fr/

Jones, J., Jiang, W., Synovic, N., Thiruvathukal, G., & Davis, J. (2024). What do we know about Hugging Face? A systematic literature review and quantitative validation of qualitative claims. *Proceedings of the 18th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*, 13–24. https://doi.org/10.1145/3674805.3686665

Jorg, T., Kämpgen, B., Feiler, D., Müller, L., Düber, C., Mildenberger, P., & Jungmann, F. (2023). Efficient structured reporting in radiology using an intelligent dialogue system based on speech recognition and natural language processing. *Insights into Imaging*, *14*(1). https://doi.org/10.1186/s13244-023-01392-y

Lyu, K. M., Lyu, R. yuan, & Chang, H. T. (2024). Real-time multilingual speech recognition and speaker diarization system based on Whisper segmentation. *PeerJ Computer Science*, *10*. https://doi.org/10.7717/peerj-cs.1973

Ma, H., Peng, Z., Shao, M., Li, J., & Liu, J. (2023). *Extending Whisper with prompt tuning to target-speaker ASR*. http://arxiv.org/abs/2312.08079

Malik, M., Malik, M. K., Mehmood, K., & Makhdoom, I. (2021). Automatic speech recognition: a survey. *Multimedia Tools and Applications*, *80*(6), 9411–9457. https://doi.org/10.1007/s11042-020-10073-7

Matsuura, K., Ashihara, T., Moriya, T., Tanaka, T., Ogawa, A., Delcroix, M., & Masumura, R. (2023). *Leveraging Large Text Corpora for End-to-End Speech Summarization*. http://arxiv.org/abs/2303.00978

Radford, A., Kim, J. W., Xu, T., Brockman, G., McLeavey, C., & Sutskever, I. (2022). *Robust Speech Recognition via Large-Scale Weak Supervision*. http://arxiv.org/abs/2212.04356

Wang, M., Li, Y., Guo, J., Qiao, X., Li, Z., Shang, H., Wei, D., Tao, S., Zhang, M., & Yang, H. (2023). WhiSLU: End-to-End Spoken Language Understanding with Whisper. *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, *2023-August*, 770–774. https://doi.org/10.21437/Interspeech.2023-1505

Wang, M. Y., & Purushotam, G. N. (n.d.). *Using Deep Learning and Augmented Reality to Improve Accessibility: Inclusive Conversations Using Diarization, Captions, and Visualization*.

Wang, S., Yang, C.-H. H., Wu, J., & Zhang, C. (2023). *Can Whisper perform speech-based in-context learning?* http://arxiv.org/abs/2309.07081

LAMPIRAN