

**PENGEMBANGAN MODEL DAN PEMBANGUNAN AI  
APLIKASI CHATBOT WAWANCARA MAGANG  
UNTUK UNIT LKM**

***AI MODEL ENHANCEMENT AND UI DEVELOPMENT  
OF AN INTERNSHIP INTERVIEW CHATBOT  
APPLICATION FOR LKM UNIT***

**Dokumen ini ditujukan untuk memenuhi persyaratan  
Mata Kuliah Tugas Akhir  
Jalur Magang Dua Semester**



Disusun oleh,  
6706220039 - FADHILAH MUHAMMAD FARHAN

**PROGRAM STUDI D3 REKAYASA PERANGKAT  
LUNAK APLIKASI  
FAKULTAS ILMU TERAPAN  
UNIVERSITAS TELKOM  
BANDUNG  
2025**

## LEMBAR PERSEMBAHAN

Dengan penuh rasa hormat, saya Fadhilah Muhammad Farhan mempersembahkan karya ini kepada:

Ayah dan ibu, terimakasih atas doa yang tiada henti dan telah menjadi sumber semangat saya. Terima kasih juga atas cinta, dukungan, dan pengorbanan tanpa batas yang telah mengiringi setiap langkah saya.

Ibu Mia Rosmiati sebagai dosen pembimbing akademik, yang telah membimbing dengan penuh kesabaran, memberikan ilmu, arahan, dan motivasi selama proses penyusunan tugas akhir ini.

Ibu Cahyana dan Ibu Inne sebagai dosen pembimbing lapangan, yang telah memberikan kesempatan, bimbingan, dan pengalaman berharga selama pelaksanaan kerja praktik.

Rekan rekan proyek dan teman teman saya, terima kasih atas kerja sama, tawa, diskusi panjang, dan semangat yang terus menguatkan hingga tugas akhir ini dapat diselesaikan.

Dan tak lupa juga saya ucapkan terima kasih banyak kepada para psikolog yang telah bersedia meluangkan waktunya dan membagikan ilmunya demi kelancaran proyek ini.

Semoga karya sederhana ini dapat menjadi awal dari kontribusi saya yang lebih besar di masa depan nanti.

# LEMBAR PENGESAHAN

## PENGEMBANGAN MODEL DAN PEMBANGUNAN AI APLIKASI CHATBOT WAWANCARA MAGANG UNTUK UNIT LKM

### *AI MODEL ENHANCEMENT AND UI DEVELOPMENT OF AN INTERNSHIP INTERVIEW CHATBOT APPLICATION FOR LKM UNIT*

Penulis

Fadhilah Muhammad Farhan

NIM 6706220039

---

Dosen Pembimbing 1

MIA ROSMIATI, S.Si., M.T

NIP 14820012

---

Tanggal Pengesahan: 31 Januari 2025 (tanggal pengumpulan revisi)

## PERNYATAAN

Dengan ini saya menyatakan bahwa:

1. Karya tulis ini adalah asli dan belum pernah diajukan untuk mendapatkan gelar akademik (Ahli Madya, Sarjana, Magister dan Doktor), baik di Fakultas Ilmu Terapan Universitas Telkom maupun di perguruan tinggi lainnya;
2. karya tulis ini murni gagasan, rumusan, dan penelitian saya sendiri, tanpa bantuan pihak lain, kecuali arahan tim pembimbing atau tim promotor atau penguji;
3. dalam karya tulis ini tidak terdapat cuplikan karya atau pendapat yang telah ditulis atau dipublikasikan orang lain, kecuali secara tertulis dengan jelas dicantumkan sebagai acuan dalam naskah dengan menyebutkan nama pengarang dan dicantumkan dalam daftar pustaka;
4. saya mengizinkan karya tulis ini dipublikasikan oleh Fakultas Ilmu Terapan Universitas Telkom, dengan tetap mencantumkan saya sebagai penulis; dan

Pernyataan ini saya buat dengan sesungguhnya dan apabila pada kemudian hari terdapat penyimpangan dan ketidakbenaran dalam pernyataan ini maka saya bersedia menerima sanksi akademik berupa pencabutan gelar yang telah diperoleh karena karya tulis ini, serta sanksi lainnya sesuai norma yang berlaku di Fakultas Ilmu Terapan Universitas Telkom.

Bandung, 28 Mei 2025 (tanggal pengumpulan revisi)

Pembuat pernyataan,

Fadhilah Muhammad Farhan

## KATA PENGANTAR

Puji dan syukur penulis panjatkan ke hadirat Tuhan Yang Maha Esa atas segala rahmat, karunia, dan kekuatan yang telah diberikan, sehingga penulis dapat menyelesaikan laporan magang ini dengan judul "*Pengembangan Model dan Pembangunan AI Aplikasi Chatbot Wawancara Magang untuk unit LKM*". Laporan ini disusun sebagai hasil dari kegiatan magang selama satu tahun yang sekaligus menjadi bagian dari proses penyelesaian tugas akhir studi.

Dalam kesempatan ini, penulis ingin menyampaikan rasa terima kasih yang sebesar-besarnya kepada semua pihak yang telah memberikan dukungan, bimbingan, dan bantuan selama proses magang hingga penyusunan laporan ini, khususnya kepada:

Keluarga tercinta, yang selalu menjadi sumber kekuatan, doa, dan semangat dalam setiap langkah yang penulis jalani.

Teman-teman dan rekan seperjuangan, yang telah banyak membantu dalam diskusi, kolaborasi, serta memberikan semangat di saat sulit.

Dosen pembimbing dan seluruh pihak di lingkungan saya yang telah memberikan arahan, bimbingan, dan kesempatan untuk terlibat langsung dalam proyek ini.

Penulis menyadari bahwa laporan ini masih jauh dari sempurna. Oleh karena itu, penulis terbuka terhadap segala bentuk kritik dan saran yang membangun.

Akhir kata, penulis berharap bahwa proyek ini dapat memberikan manfaat di masa depan, baik sebagai referensi pengembangan teknologi AI di bidang perekrutan, maupun sebagai solusi praktis dalam proses wawancara magang, khususnya di lingkungan unit LKM.

Bandung, 4 Mei 2025

**Penulis**

## ABSTRAK

Fakultas Ilmu Terapan Universitas Telkom memiliki program tahunan yaitu magang. Dalam penerapannya, ada sebuah proses yang harus dilewati mahasiswa yaitu seleksi magang. Perusahaan biasanya memanggil mahasiswa yang terpilih untuk diwawancarai. Banyak kemungkinan yang terjadi di lapangan membuat mahasiswa tidak bisa mengikuti wawancara secara langsung karena keterbatasan waktu dan tempat. Oleh karena itu diusulkanlah sebuah solusi, yaitu wawancara menggunakan chatbot. Talentern chatbot merupakan sebuah chatbot yang dirancang untuk mewawancarai, menilai, dan menyimpulkan *softskills* pengguna. Pada pembuatan prototipenya chatbot ini menggunakan *ReactJS* dan *Typescript*. Untuk integrasi ke aplikasi yang lebih besar, *Laravel* dan *PHP* digunakan untuk menyesuaikannya. Adapun *service* yang digunakan seperti openAI API dan Pinecone database untuk pembuatan logika pemrosesan jawaban. Untuk Teknik yang digunakan dalam penilaian jawaban pengguna, *word embedding* dan perhitungan *cosine similarity* digunakan. Tujuan dari penggunaan metode ini adalah, untuk menghasilkan penilaian yang lebih konsisten daripada menggunakan model AI yang besar. Berdasarkan hasil pengujian, aplikasi ini dapat meningkatkan efisiensi dalam proses wawancara magang yang dilaksanakan oleh mitra.

Kata Kunci: Chatbot, Web, React, *Laravel*, *Cosine Similarity*, *Word Embedding*

## ABSTRACT

*The Faculty of Applied Sciences at Telkom University has an annual internship program. As part of its implementation, students must undergo an internship selection process. Typically, selected students are invited for interviews by companies. However, various real-world constraints such as time and location often prevent students from attending interviews in person. To address this issue, an alternative solution is proposed; conducting interviews through a chatbot. The "Talentern Chatbot" is a chatbot designed to interview, score, and conclude users' soft skills. In the prototype development, this chatbot is built using ReactJS and Typescript. For integration into a larger application, Laravel and PHP are used for better compatibility. Services such as the OpenAI API and Pinecone database are utilized to implement the answer processing logic. The technique used to score user responses includes word embedding and cosine similarity calculations. The purpose of using this method is to produce more consistent assessments compared to relying solely on large AI models. Based on the test results, this application can improve efficiency in the internship interview process conducted by partners.*

*Keywords: Chatbot, Web, React, Laravel, Cosine Similarity, Word Embedding*

# DAFTAR ISI

<b>LEMBAR PERSEMBAHAN.....</b>	<b><i>i</i></b>
<b>LEMBAR PENGESAHAN .....</b>	<b><i>i</i></b>
<b>PERNYATAAN.....</b>	<b><i>i</i></b>
<b>KATA PENGANTAR.....</b>	<b><i>ii</i></b>
<b>ABSTRAK.....</b>	<b><i>iii</i></b>
<b>ABSTRACT.....</b>	<b><i>iv</i></b>
<b>DAFTAR ISI .....</b>	<b><i>v</i></b>
<b>DAFTAR GAMBAR.....</b>	<b><i>i</i></b>
<b>DAFTAR TABEL.....</b>	<b><i>ii</i></b>
<b>DAFTAR LAMPIRAN.....</b>	<b><i>i</i></b>
<b>BAB I PENDAHULUAN .....</b>	<b><i>1</i></b>
1.1 Latar Belakang .....	<b><i>1</i></b>
1.2 Rumusan Masalah dan Solusi.....	<b><i>3</i></b>
1.3 Tujuan .....	<b><i>3</i></b>
1.4 Batasan Masalah .....	<b><i>4</i></b>
1.5 Penjadwalan Kerja .....	<b><i>4</i></b>
<b>BAB II PROFIL ORGANISASI .....</b>	<b><i>6</i></b>
2.1 Deskripsi Organisasi.....	<b><i>6</i></b>
2.1.1 CoE Smart City .....	<b><i>6</i></b>
2.2 Struktur Organisasi dan Tata Kelola.....	<b><i>7</i></b>
2.3 Deskripsi Pekerjaan .....	<b><i>7</i></b>
<b>BAB III ANALISIS PEKERJAAN.....</b>	<b><i>9</i></b>
3.1 Tinjauan Pustaka.....	<b><i>9</i></b>
3.1.1 Finetuning model .....	<b><i>9</i></b>
3.1.2 Word embedding.....	<b><i>9</i></b>
3.1.3 Cosine similarity .....	<b><i>9</i></b>
3.2 Teknologi yang digunakan .....	<b><i>9</i></b>
3.2.1 ReactJS .....	<b><i>10</i></b>
3.2.2 NextJS .....	<b><i>10</i></b>
3.2.3 OpenAI.....	<b><i>11</i></b>
3.2.4 Laravel.....	<b><i>12</i></b>
3.2.6 Pinecone Database .....	<b><i>12</i></b>
3.3 Analisis Sistem.....	<b><i>13</i></b>
3.3.1 Gambar Sistem Saat Ini .....	<b><i>13</i></b>

3.3.2 Pengembangan Sistem .....	13
3.4 Kebutuhan Perangkat Kerja .....	20
3.4.1 Kebutuhan Perangkat Pengembangan Sistem .....	20
3.3.2 Kebutuhan Perangkat Implementasi Sistem.....	21
<b>BAB IV HASIL DAN PEMBAHASAN.....</b>	<b>23</b>
4.1 Hasil Akhir (Luaran) .....	23
4.2 Alur Penilaian Jawaban Pengguna .....	25
4.3 Perhitungan nilai .....	27
4.4 Pengujian Luaran .....	28
4.3.1 Pengujian API.....	28
4.3.1 Pengujian Usabilitas .....	29
<b>BAB V PENUTUP .....</b>	<b>34</b>
5.1 Kesimpulan.....	34
5.2 Saran.....	34
<b>DAFTAR PUSTAKA.....</b>	<b>35</b>

## DAFTAR GAMBAR

Gambar 2. 1 logo LKM.....	<b>Error! Bookmark not defined.</b>
Gambar 2. 2 Struktur organisasi CoE smart city .....	7
Gambar 2. 3 Alur pengerjaan proyek magang .....	8
Gambar 3. 1 ReactJS Dashboard .....	10
Gambar 3. 2 NextJS Dashboard .....	11
Gambar 3. 3 OpenAI Dashboard .....	11
Gambar 3. 4 Laravel Dashboard.....	12
Gambar 3. 5 Pinecone Dashboard.....	13
Gambar 3. 6 Tech Stack Prototipe .....	14
Gambar 3. 7 UI prototipe chatbot.....	15
Gambar 3. 8 Alur pemrosesan jawaban prototipe .....	16
Gambar 3. 9. UI Dashboard admin .....	17
Gambar 3. 10 UI dashboard Perusahaan.....	18
Gambar 3. 11 halaman list wawancara .....	18
Gambar 3. 12 pembagian sistem chatbot.....	20
Gambar 3. 13 VsCode versi 1.100 .....	21
Gambar 3. 14 Postman versi 11.44.0.....	21
Gambar 3. 15 Google Chrome versi 136.0.7103.93 .....	22
Gambar 4. 1 Talentern Chatbot .....	23
Gambar 4. 2 Panduan sesi wawancara .....	23
Gambar 4. 3 Halaman pra-wawancara .....	23
Gambar 4. 4 Syarat dan ketentuan wawancara .....	24
Gambar 4. 5 activity diagram taletern chatbot.....	24
Gambar 4. 6 Tech stack Talentern Chatbot .....	25
Gambar 4. 7 Alur pemrosesan jawaban pengguna .....	26
Gambar 4. 8 dataset jawaban pengguna di pinecone .....	26
Gambar 4. 9 API embedding kalimat .....	28
Gambar 4. 10 search data request.....	28
Gambar 4. 11 get-Next-question request .....	29
Gambar 4.12 Request Penyimpulan riwayat percakapan.....	29
Gambar 4. 13 Grafik kuesioner pernyataan 1 .....	30
Gambar 4. 14 Grafik kuesioner pernyataan 2 .....	30
Gambar 4. 15 Grafik kuesioner pernyataan 3 .....	30
Gambar 4. 16 Grafik kuesioner pernyataan 4 .....	30
Gambar 4. 17 Grafik kuesioner pernyataan 5 .....	31
Gambar 4. 18 Grafik kuesioner pernyataan 6 .....	31
Gambar 4. 19 Grafik kuesioner pernyataan 7 .....	31
Gambar 4. 20 Grafik kuesioner pernyataan 8 .....	31
Gambar 4. 21 Grafik kuesioner pernyataan 9 .....	32
Gambar 4. 22 Grafik kuesioner pernyataan 10.....	32
Gambar 4. 23 Index penilaian SUS.....	33

## DAFTAR TABEL

Tabel 1. 1 Tabel penerimaan magang 2024.....	1
Tabel 1. 2 Penjadwalan kerja Agustus sampai November .....	4
Tabel 1. 3 Penjadwalan kerja Desember sampai Maret .....	4
Tabel 1. 4 Penjadwalan kerja April sampai juni.....	5
Tabel 3. 1 kebutuhan perangkat keras.....	21
Tabel 3. 2 Kebutuhan perangkat lunak .....	21
Tabel 4. 1 kategori jawaban pertanyaan kriteria Manajemen Waktu .....	27
Tabel 4. 2 hasil penilaian kriteria manajemen waktu .....	27
Tabel 4. 3 Konversi SUS score .....	32

## DAFTAR LAMPIRAN

# BAB I PENDAHULUAN

## 1.1 Latar Belakang

Magang adalah salah satu program yang digunakan para siswa maupun mahasiswa untuk mendapatkan pengalaman kerja di lapangan yang sesungguhnya. Universitas Telkom khususnya di Fakultas Ilmu Terapan (FIT), Terdapat sebuah unit yang bernama LKM (Layanan Kerjasama Magang). Unit ini bertugas mengkoordinasi berjalanya magang untuk mahasiswa FIT. Terdapat 3 (tiga) fase kegiatan utama pada Magang yakni Pendaftaran dan Seleksi, Pelaksanaan Magang, dan Monitoring dan Supervisi. Pendaftaran dan seleksi memiliki sub kegiatan seperti pemberian informasi mitra kepada mahasiswa dan seleksi magang.

Selama ini proses seleksi dilaksanakan secara manual dengan Microsoft Excel yang sudah disebar di berbagai platform sosial media. Nantinya mitra akan melihat data pelamar (Mahasiswa) dari Excel yang sudah disediakan, kemudian mitra akan melakukan seleksi wawancara baik secara online maupun onsite. Durasi minimal yang digunakan adalah sekitar 20 menit dengan berbagai pertanyaan umum terkait identitas, program studi, dan minat sampai dengan pertanyaan pertanyaan spesifik seperti pengalaman, lomba, kelebihan dan kekurangan, kemampuan softskill dan lain lain[1]. Pada semester ganjil tahun 2022-2023 terlihat rata rata perbandingan antara Perusahaan dan jumlah pendaftar adalah 1:4. Artinya mitra harus mewawancarai 4 orang untuk mendapatkan 1 orang peserta yang lolos. Tabel 1.1 menunjukkan data jumlah pendaftar dan mahasiswa yg diterima.

Tabel 1. 1 Tabel penerimaan magang 2024

No	Mitra DUDI	Jumlah Pendaftar	Jumlah Diterima	Persen Diterima	Perbandingan Diterima
1	Transtrack.id (PT. Indo Trans Teknologi)	36	15	42%	1:2
2	PT. Datacomm Diangraha	2	2	100%	1:1
3	PT. BIG IO	18	8	44%	1:2
4	PT. Wibicon Karya Indonesia	6	5	83%	1:1
5	Bank Negara Indonesia (BNI)	55	36	65%	2:3
6	PT. Aero Systems Indonesia	36	13	36%	1:3

No	Mitra DUDI	Jumlah Pendaftar	Jumlah Diterima	Persen Diterima	Perbandingan Diterima
7	PT. Inovasi Daya Solusi (IDS)	21	8	38%	1:3
8	PT. Neuronworks Indonesia	32	7	22%	1:4
9	PT. Tabel Data Informatika	22	7	32%	1:3
10	PT. Alpha Beta Engineering (ABE)	27	7	26%	1:4
11	Diskominfo Pariaman	6	2	33%	1:3
12	Telkom Regional III (TREG 3)	46	13	28%	1:4
13	Telkom Regional IV (TREG 4)	43	5	12%	1:9
14	Medcom.ID	18	3	17%	1:6
15	PT. Wynettes Walden Internation	16	3	19%	1:5
16	Diskominfo Kota Bandung	17	7	41%	1:2
17	Mercure Bandung Nexa Hotel	7	4	57%	2:3
18	PT. Adi Data Utama	9	9	100%	1:1

Berdasarkan Informasi yang diperoleh dari LKM terdapat ketidakefektifan dari sistem lama yang digunakan, seperti untuk mendapatkan 1 orang, pewawancara membutuhkan minimal rata-rata 80 menit. Maka dari itu dibuatlah talentern chatbot. Pada chatbot ini, dikembangkan suatu teknologi yang bertujuan untuk mendukung proses seleksi magang di Unit LKM FIT. Biasanya, proses wawancara dilakukan secara langsung oleh pihak mitra yang menjadi penanggung jawab seleksi magang. Namun, metode wawancara langsung ini sering kali memerlukan waktu, tenaga yang cukup banyak. Chatbot ini dirancang agar dapat menanyakan berbagai pertanyaan yang umumnya muncul dalam wawancara kerja yang terkait dengan softskill yang dimiliki kandidat. Dengan demikian, secara tidak langsung chatbot ini diharapkan mampu menggantikan peran mitra dalam proses wawancara seleksi magang. Talentern chatbot mampu diakses kapan saja tanpa memerlukan jadwal khusus antara mitra dan calon peserta. Chatbot ini memungkinkan setiap pelamar menyelesaikan wawancara dalam waktu maksimal 60 menit secara bersamaan. Hal ini jauh lebih efisien dibandingkan metode wawancara satu per satu yang rata-rata memakan waktu 80 menit per pelamar.

Talentern Chatbot merupakan sebuah fitur pada aplikasi Talentern yang memungkinkan perusahaan mewawancarai kandidat magang dengan menggunakan chatbot. Chatbot ini bertugas untuk menilai softskill dari kandidat atau pelamar magang. Untuk memastikan kualitas dari chatbot ini, tentunya pemilihan data untuk dataset chatbot sangat diperhatikan. Untuk pengumpulan

data daftar pertanyaan, wawancara kepada psikolog dari Universitas Telkom pun dilakukan. Wawancara ini bertujuan untuk mendapatkan pertanyaan yang relevan dan kredibel terkait dengan softskill kandidat. Selain itu wawancara juga dilakukan di akhir proyek untuk memastikan penilaian yang dikeluarkan oleh chatbot sesuai dengan ekspektasi psikolog. Adapun metode yang digunakan untuk melakukan penilaian pada chatbot ini. NLP (*Natural Processing Language*) sederhana seperti *word embeddings* dan pencarian kesamaan makna dengan *cosine similarity* digunakan untuk mencari kesamaan makna dengan dataset yang terimpan dalam database. Selain itu chatbot ini menggunakan service seperti pre-trained model *Text-embedding-03-small* dari openAI untuk proses *embedding* datanya, dan menggunakan pinecone database untuk menyimpan *data-sample embedding*.

Pada pengembangan project ini terdapat pembagian tugas yang diberikan. pada project ini Penulis mengambil peran sebagai *Frontend Developer*. Selain menjalani tugas sebagai Frontend developer penulis juga bertugas meneliti tentang pemanfaatan dan penggunaan *Artificial Intelegent(AI)* pada proyek ini. Adapun alat atau teknologi yang digunakan pada pengembangan proyek ini seperti Bahasa *Typescript*, *framework ReactJS*, *NextJS*, dan *Tailwind CSS* untuk pembuatan prototipenya. Selain itu digunakan *Laravel* dan *PHP* untuk integrasi kedalam aplikasi yang lebih besar. Untuk penelitian AI penulis menggunakan teknologi yang disediakan oleh OpenAI, seperti *pretrained model*, dan *fine-tuning tools*. Alasan digunakanya teknologi dari OpenAI adalah untuk keefisienan penelitian. Adapaun teknologi lain yang digunakan seperti Pinecone database, untuk menyimpan *dataset* penilaian dan juga *PHP* dan *Laravel* untuk poses integrasi dari prototipe ke dalam aplikasi Talentern.

## 1.2 Rumusan Masalah dan Solusi

Berdasarkan latar belakang diatas, rumusan masalah yang akan dibahas adalah :

1. Bagaimana membuat sistem chatbot untuk wawancara seleksi magang?
2. Metode pengembangan apa yang sesuai untuk membangun UI aplikasi ini?
3. Bagaimana cara agar penilaian dari chatbot yang dihasilkan AI menjadi konsisten?

Adapun Solusi untuk rumusan masalah tersebut :

1. Membuat sistem chatbot berbasis web dengan *Laravel PHP*
2. UI chatbot dikembangkan sebagai bagian dari metode Scrum yang digunakan untuk membangun keseluruhan sistem .
3. Penggunaan *word embedding* dan *cosine similarity* untuk konsistensi output

## 1.3 Tujuan

Adapun tujuan dari pembuatan Aplikasi Talentern Chatbot:

1. Membuat aplikasi chatbot untuk wawancara magang berbasis web.
2. membuat aplikasi chatbot dengan pengerjaan yang terjadwal dan tersusun.
3. Membuat output penilian yang dikeluarkan oleh chatbot menjadi konsisten

### 1.4 Batasan Masalah

Adapun batasan masalah pada pembangunan Talentern Chatbot:

1. Aplikasi Chatbot hanya menilai dan menanyakan softskills kandidat bukan hardskill.
2. Talentern Chatbot hanya dipakai di lingkungan FIT.
3. Integrasi frontend dan backend tidak melalui API.
4. Pengerjaan hanya terfokus ke chatbot bukan keseluruhan alur aplikasi.
5. Chatbot hanya bisa menilai berdasarkan teks secara verbal.

### 1.5 Penjadwalan Kerja

Periode pengerjaan talentern chatbot dimulai dari bulan September 2024 sampai dengan April 2025. Tabel 1.2 sampai 1.4 merupakan jadwal pengerjaan dari Talentern Chatbot

Tabel 1. 2 Penjadwalan kerja Agustus sampai November

No	Deskripsi Kerja	Agustus				September				Oktober				November			
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1	Studi literatur	■	■	■	■												
2	Diskusi dan perencanaan proyek					■	■										
3	Perencanaan tools dan framework							■	■								
4	Percobaan penggunaan API Open AI dan prompting									■							
5	Pembuatan UI ChatBot										■						
6	Pembuatan UI Admin											■					
7	Pembuatan UI Perusahaan												■				
8	Pembuatan UI User													■			
9	Revisi UI Prototype														■		
10	Pembuatan logika prototipe chatbot															■	
11	Pengujian prototipe chatbot																■

Tabel 1. 3 Penjadwalan kerja Desember sampai Maret

No	Deskripsi Kerja	Desember				Januari				Februari				Maret			
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4

12	Pengumpulan dan pengolahan dataset jawaban	█																	
13	<i>Finetuning model</i> untuk chatbot dan pengujian <i>finetuned model</i>		█																
14	Pengolahan data dan percobaan penggunaan metode <i>text embedding</i> dan <i>cosine similarity</i>			█															
15	Diskusi Perencanaan Integrasi Dengan Talentern dan pembuatan sprint				█	█													
16	Pembuatan <i>Frontend</i> chatbot untuk talentern						█												
17	Pengujian FE dan revisi							█											
18	Revisi logic chatbot								█										
19	Perubahan FE chatbot									█									
20	Penambahan dan pengolahan dataset										█								
21	Pembuatan FE lanjutan(Panduan, Demo wawancara, syarat dan ketentuan)											█							
22	Pengujian aplikasi												█						
23	Revisi FE Talentern Chatbot													█					
24	Perbaikan bug sistem chatbot														█				
25	Pengumpulan dan pengolahan dataset jawaban tambahan																		█
26	Pengujian sistem dan perbaikan bug chatbot																		█

Tabel 1. 4 Penjadwalan kerja April sampai juni

No	Deskripsi Kerja	April				Mei				Juni			
		1	2	3	4	1	2	3	4	1	2	3	4
27	Testing Chatbot dan perbaikan bug	█	█	█	█	█	█	█	█	█	█	█	█

## BAB II PROFIL ORGANISASI

### 2.1 Deskripsi Organisasi

#### 2.1.1 CoE Smart City

Center of Excellence Smart City atau CoE Smart City Universitas Telkom merupakan kelompok penelitian yang berperan dalam menyelenggarakan berbagai kegiatan untuk mendukung pengembangan sistem kota cerdas dan ekosistemnya. Peran ini diwujudkan melalui berbagai upaya, seperti penelitian, publikasi hasil melalui kegiatan pengabdian kepada masyarakat, publikasi ilmiah, serta pendaftaran Hak Kekayaan Intelektual (HKI). Sebagai bentuk kontribusi nyata, CoE Smart City aktif mengembangkan produk-produk inovatif yang tidak hanya berdampak pada masyarakat luas, tetapi juga memperkuat kualitas dan kinerja akademik Universitas Telkom. Komitmen ini ditunjukkan melalui kolaborasi lintas disiplin, pemanfaatan teknologi mutakhir, dan responsif terhadap kebutuhan mitra. Dalam konteks proyek ini, Talentern Chatbot merupakan salah satu inisiatif penelitian yang dikembangkan oleh CoE Smart City sebagai tindak lanjut dari permintaan unit Layanan Kerja Sama dan Magang (LKM), yang bertindak sebagai klien. Proyek ini menjadi bagian dari upaya CoE Smart City untuk memberikan solusi digital yang aplikatif dan mendukung peningkatan layanan institusi di lingkungan kampus.

#### 2.1.2 LKM

Layanan Kerja Sama dan Magang (LKM) adalah unit di Universitas Telkom yang menyelenggarakan program magang dan kerja sama dengan mitra atau Perusahaan. Gambar 2.1 merupakan logo dari LKM.



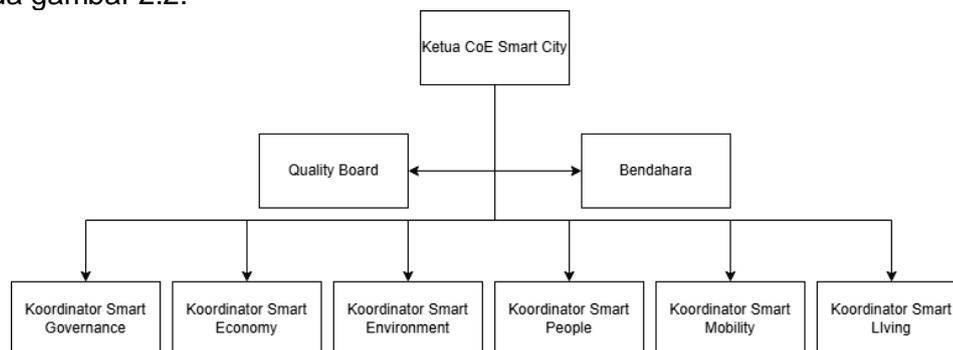
*Gambar 2. 1 Logo LKM*

LKM berlokasi di Gedung Selaru lantai 1, Fakultas Ilmu Terapan, Jl. Telekomunikasi No. 1, Bandung, Jawa Barat. Sebagai bagian dari upaya mendukung visi Fakultas Ilmu Terapan untuk menjadi fakultas vokasi unggul dalam riset terapan dan kewirausahaan, LKM menjembatani kebutuhan mahasiswa dan mitra dalam pengembangan kompetensi di bidang teknologi terapan, manajemen, dan pariwisata berbasis TI. LKM menyediakan layanan

seperti konsultasi magang (onsite dan online), pendaftaran, serta pembuatan surat pengantar. Selain itu, LKM mengatur pembimbing akademik, menyelenggarakan sosialisasi dan pembekalan magang. Selama masa magang, LKM melaksanakan monitoring dan evaluasi (MONEV) secara langsung atau daring. Setelah magang selesai, LKM mengelola pengumpulan dokumen akhir, pengecekan laporan, penilaian, dan rekap nilai di platform yang telah ditentukan. Melalui layanan ini, LKM berkomitmen memberikan pengalaman praktis terbaik bagi mahasiswa serta memperkuat kerja sama strategis dengan mitra.

## 2.2 Struktur Organisasi dan Tata Kelola

*Research Group* CoE Smart City memiliki struktur organisasi seperti yang terlihat pada gambar 2.2.



Gambar 2. 2 Struktur organisasi CoE smart city

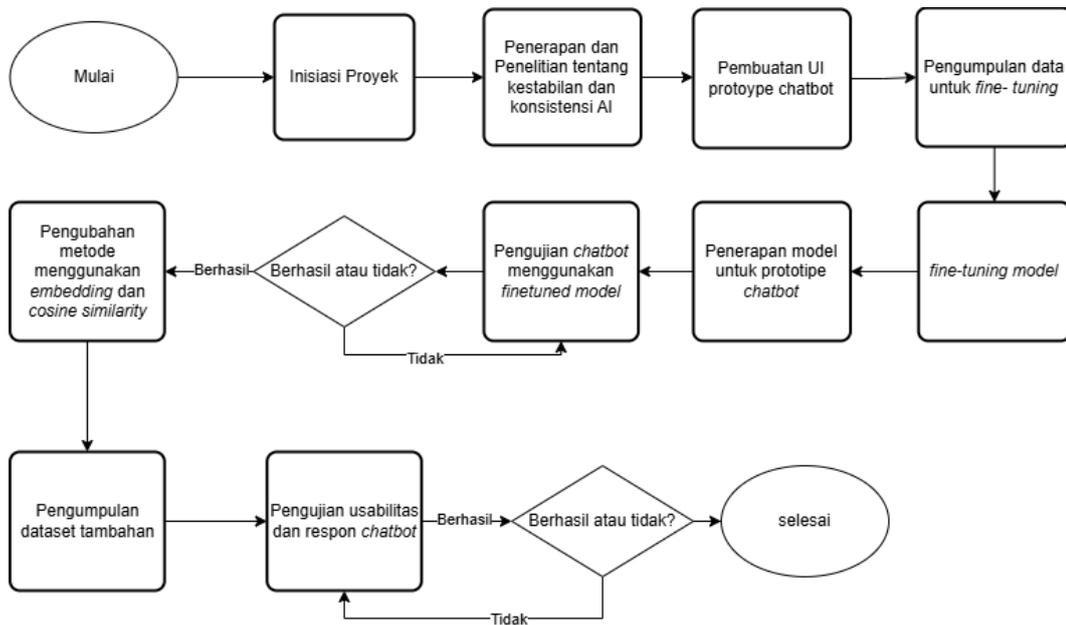
Pada struktur organisasi ini setiap divisi memiliki peranya masing masing. Ketua CoE Smart City bertanggung jawab atas koordinasi tim, pelaksanaan strategi dan pemantauan proyek. Lalu dibawahnya ada Quality board yang bertugas pada *Quality Control* untuk melakukan pengecekan kualitas produk yang dihasilkan oleh CoE Smart city. Lalu ada juga bendahara yang bertygas mengatur keuangan dari keseluruhan *Research Group* Smart City. Di bawah *Quality board* dan bendahara ada divisi divisi yang dibuat untuk melaksanakan kegiatan penelitian dan hilirisasi teknologi sesuai dengan topik topik yang dituliskan dalam Peraturan Pemerintah no.59 Tentang perkotaan Pasal 61. Koordinator Smart Governance untuk tatakelola birokrasi, Koordinator Smart Economy untuk perekonomian, Koordinator Smart Environment bertanggung jawab dalam hal yang terkait dengan lingkungan, Koordinator Smart People untuk Mengembangkan dan melakukan penelitian terkait masyarakat, Koordinator Smart Mobility untuk mobilitas, dan yang terakhir Koordinator Smart Living untuk Kehidupan Berkota. Pada proyek talentern chatbot Penulis mengambil peran dalam divisi *Smart People* yang merupakan salah satu dari divisi di CoE Smart City. Divisi ini bertugas untuk menjalankan kegiatan penelitan yang sesuai dengan PP no.59 tentang perkotaan yang meliputi keterbukaan terhadap informasi,tidak diskriminatif, tidak intoleransi; beradaptasi dengan kemajuan teknologi; disiplin dan teratur dalam melakukan kegiatan di Perkotaan; dan menerapkan budaya saling menghormati dan berbudi pekerti.

## 2.3 Deskripsi Pekerjaan

Pada divisi *Smart people*, Penulis mengambil peran pada divisi frontend developer di project ini. Penulis membangun *UI* pada bagian chatbot untuk wawancara

kandidat. selain itu penulis juga memiliki peran sampingan sebagai peneliti dimana penulis melakukan penelitian terkait kestabilan data yang dihasilkan oleh *NLP(Natural Language Processing)* dari OpenAI API, serta membuat model, dan mengumpulkan dataset penelitian.

Frontend developer bertugas membuat bagian dari sebuah website atau aplikasi yang berinteraksi langsung dengan pengguna, termasuk *layout* dan fungsionalitas[2]. Pada proyek ini selain bertugas sebagai Frontend developer juga bertanggung jawab untuk melakukan penelitian terkait model dan konstensi dari sebuah AI. AI yang digunakan pada proyek ini merupakan AI dari OpenAI dengan model *gpt-4.0 mini* dan juga *fine-tuned model*. Gambar 2.3 merupakan alur kerja yang dilakukan oleh penulis pada proyek ini



Gambar 2. 3 Alur pengerjaan proyek magang

## BAB III ANALISIS PEKERJAAN

### 3.1 Tinjauan Pustaka

Talintern chatbot membutuhkan pembelajaran lebih lanjut dalam pengembangannya. Adapun metode yang dipelajari dan digunakan dalam pengembangan talintern chatbot seperti *finetuning model*, *word embedding* dan *cosine similarity*.

#### 3.1.1 Finetuning model

*Fine tuning* adalah proses penyesuaian model yang sudah ada (*pretrained model*) agar lebih sesuai dengan kebutuhan spesifik atau konteks tertentu. *Fine tuning* dilakukan dengan menambahkan dataset baru dengan lingkup kecil pada sebuah model. Ketika *fine tuning* dilakukan pada sebuah model gpt 4.0, maka lingkup jawaban dari model tersebut yang awalnya sangat luas bisa dipersmpit sesuai dengan konteks yang diinginkan. dengan data yang cukup, model lokal melampaui kinerja model besar yang terintegrasi[3].

#### 3.1.2 Word embedding

*Word embedding* adalah representasi vektor bernilai real dari kata-kata dengan memasukkan makna semantik dan sintaksis yang diperoleh dari korpus besar yang tidak berlabel[4]. Pada proyek TalIntern Chatbot, metode ini digunakan untuk mengubah jawaban pengguna dan dataset menjadi sebuah vector numerik untuk dicari kesamaan makna kalimatnya. Model *Text-Embedding-3-small* merupakan model yang dipakai untuk melakukan *word embedding* pada Talintern Chatbot

#### 3.1.3 Cosine similarity

*Cosine similarity* adalah ukuran tingkat kesamaan antara dua vektor dan merupakan yang paling populer dalam operasi matematika yang mengambil dua vektor dan menghasilkan skalar[5]. Dalam proyek ini *cosine similarity* digunakan untuk mencari kesamaan antara jawaban pengguna saat wawancara dengan data yang ada di dalam *dataset*. Dalam penerapannya setiap data didalam dataset dan jawaban user akan diubah menjadi vektor numerik. Setelah vector didapatkan, *cosine similarity* akan digunakan untuk menghitung kesamaan / kemiripan setiap kata dalam jawaban pengguna saat wawancara dan data didalam dataset. *Cosine similarity* bisa didapat dengan membagi *dot product*(perkalian 2 vektor) vektor dengan perkalian kedua vektor. Berikut merupakan rumus *cosine similarity*

$$\frac{A \cdot B}{|A| \cdot |B|} = \frac{\sum_{i=1}^n A_i B_i}{\sqrt{\sum_{i=1}^n A_i^2} \sqrt{\sum_{i=1}^n B_i^2}} \quad (1)$$

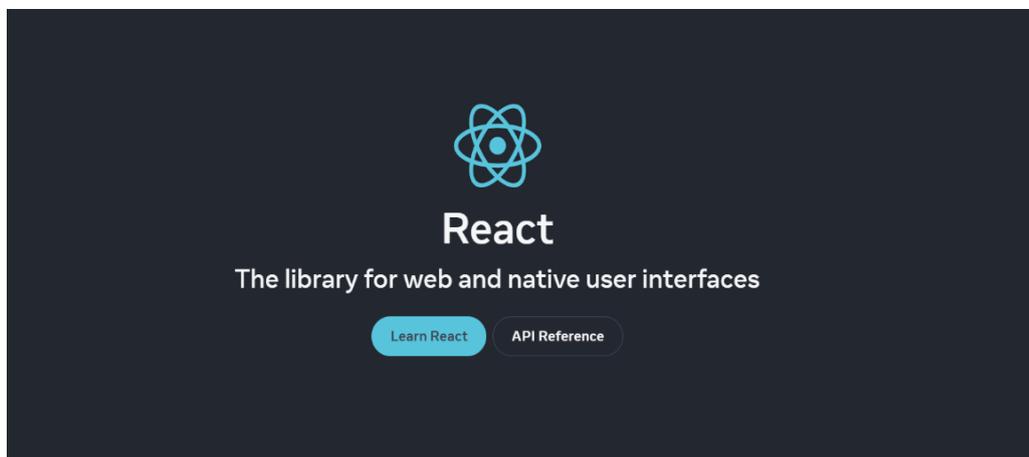
### 3.2 Teknologi yang digunakan

Talintern chatbot pada pengembangannya terbagi menjadi 2 tahap. Yang pertama merupakan pembuatan prototipe dan yang ke 2 merupakan integrasi ke aplikasi

talentern. Pada saat pengembangan prototipenya, *ReactJS*, *NextJS* dan *Typescript* digunakan untuk mengembangkan frontendnya. Selain itu untuk saat integrasi *Laravel* dan *PHP* digunakan untuk menyetarakan dengan proyek Talentern. Untuk *service* atau layanan, Talentern chatbot menggunakan pinecone database dan open AI API.

### 3.2.1 ReactJS

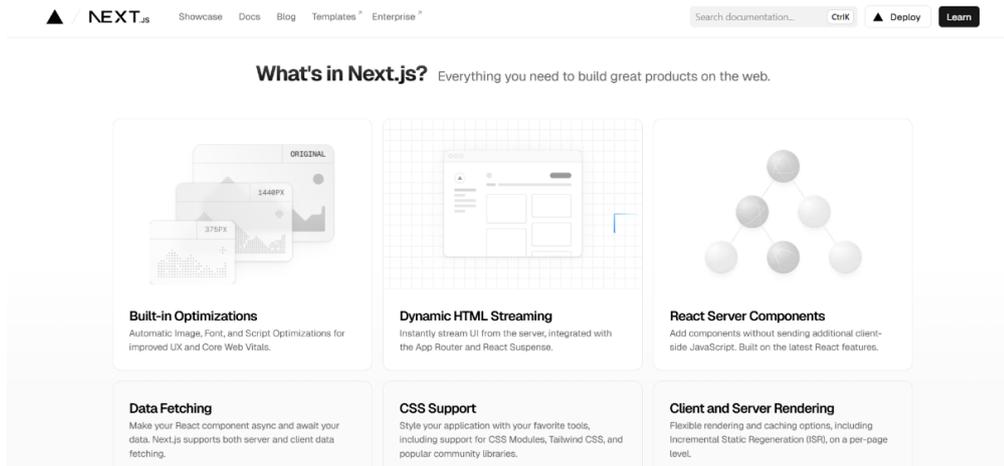
*ReactJS* adalah sebuah *open source library* yang digunakan untuk membuat UI aplikasi berbasis web[6]. Dalam pembuatan aplikasi web *ReactJS* menggunakan konsep *component-based* yang dimana setiap komponen bersifat *reusable* atau dapat digunakan berkali kali hanya dengan memanggil komponennya. Hal ini tentunya akan mempercepat proses pembuatan aplikasi web karena tidak perlu membuat kode yang berulang untuk membuat sebuah komponen. *ReactJS* digunakan penulis untuk membuat anntarmuka pada *prototype chatbot*. Gambar 3.1 merupakan tampilan dari *dashboard ReactJS*.



Gambar 3. 1 ReactJS Dashboard

### 3.2.2 NextJS

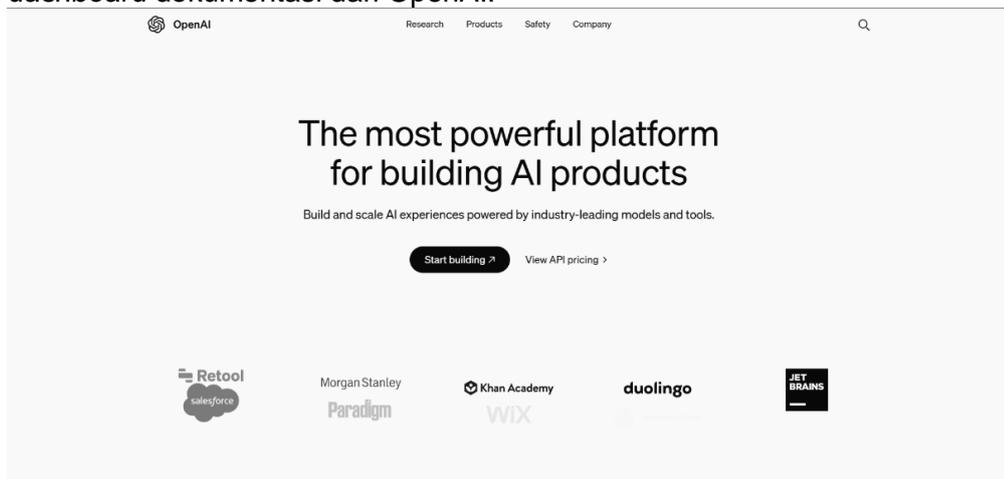
*NextJS* adalah sebuah *open source React frontend framework*. Dilihat dari dokumentasinya *NextJS* dibuat untuk mengatasi batasan masalah *server-side rendering* pada *framework React*[7]. *NextJS* dibuat diatas *React library* yang memungkinkan pengguna menggunakan *React* dengan fitur-fitur tambahan. Penggunaan *NextJS* akan membantu mengurangi beban pada sisi *client* karena dengan menggunakan *NextJS* developer dapat *generate client side document* seperti html, css dan lain lain didalam server sehingga beban untuk sisi pengguna akan berkurang. Ini juga dapat membuat dapat mempercepat performa aplikasi saat aplikasi sedang melakukan *initial loading*. Dalam proyek ini penulis menggunakan *NextJS* untuk membuat *prototype chatbot* sebelum di integrasikan ke sistem yang lebih besar. Gambar 3.2 menunjukkan tampilan *dashboard* dari *NextJS*.



Gambar 3. 2 NextJS Dashboard

### 3.2.3 OpenAI

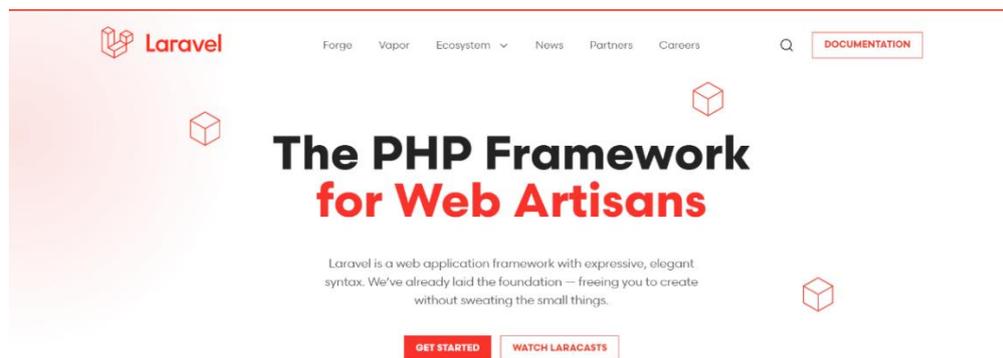
OpenAI adalah perusahaan riset nirlaba yang bergerak di bidang pengembangan dan pengarahannya kecerdasan buatan (A.I.). Lembaga ini melakukan penelitian di bidang A.I. untuk mempromosikan dan mengembangkan A.I.[8]. Di tahun 2022 OpenAI merilis model gpt 3.5 yang menggunakan dataset yang sangat besar dan dapat berinteraksi layaknya manusia dengan menggunakan system NLP (*Natural Language Processing*)[9]. Open AI menyediakan API untuk para developer aplikasi untuk menggunakannya. API yang disediakan oleh Open AI mempunyai kegunaan bermacam macam, salah satunya gpt 4.0 untuk melakukan nlp (*natural language processing*) yang biasa digunakan dalam pembuatan chatbot. Dalam project Pembuatan Chatbot Wawancara Magang Untuk LKM ini, Open AI API digunakan untuk melakukan fine-tuning model dan pembuatan sistem untuk penilaian jawaban pengguna. Gambar 3.3 merupakan tampilan *dashboard* dokumentasi dari OpenAI.



Gambar 3. 3 OpenAI Dashboard

### 3.2.4 Laravel

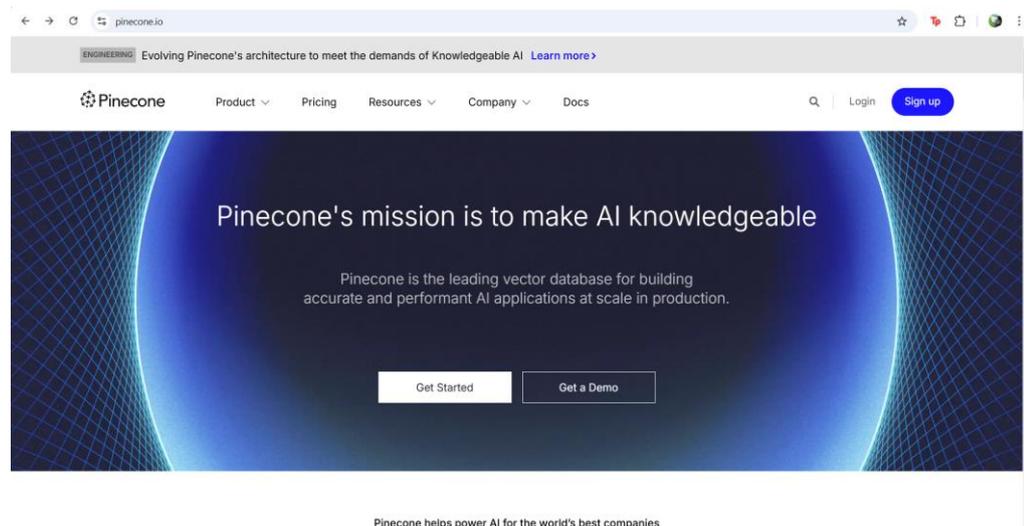
*Laravel* adalah sebuah *backend framework* untuk bahasa pemrograman *PHP* yang biasa digunakan untuk perancangan aplikasi berbasis web. *Laravel* dikembangkan oleh Taylor Otwell dan dirilis pada tahun 2011. Dirancang untuk memudahkan proses pengembangan aplikasi, *Laravel* menawarkan beragam fitur dan alat yang memungkinkan para pengembang untuk menghasilkan aplikasi web yang efisien, aman, dan mudah dikelola[10]. *Laravel* menggunakan struktur MVC (*Model, View, Controller*) dalam pembuatan aplikasi web. Dengan adanya struktur MVC pengembang lebih mudah dalam mengelola struktur projectnya. Pada proyek ini penulis menggunakan *Laravel* untuk integrasi aplikasi dengan sistem yang lebih besar. Gambar 3.4 dibawah ini menunjukkan tampilan dashboard dari web dokumentasi *Laravel*.



Gambar 3. 4 *Laravel Dashboard*

### 3.2.6 Pinecone Database

Pinecone adalah database vector, jenis database yang dirancang untuk menyimpan dan melakukan pencarian pada vektor berdimensi tinggi[11]. Dalam pemanfaatannya di proyek ini, pinecone menyimpan vektor hasil embedding dataset kedalam sebuah index(tabel). Selain itu pinecone juga akan membantu program dalam mengelompokkan jawaban pengguna mengembelikan dataset yang terdekat dengan pengguna dengan metode perhitungan *cosine similarity*. Gambar 3.5 menunjukkan *dashboard* dari pinecone database.



Pinecone helps power AI for the world's [best companies](#)  
 Gambar 3. 5 Pinecone Dashboard

### 3.3 Analisis Sistem

#### 3.3.1 Gambar Sistem Saat Ini

Fakultas Ilmu Terapan Universitas Telkom memiliki program tahunan untuk mahasiswa semester 5 dan 6, yaitu magang. Selama ini magang yang dilakukan memiliki 3 tahap utama, yaitu pendaftaran dan seleksi, *monitoring* dan pemantauan, dan yang terakhir pelaksanaan magang. Pendaftaran dan seleksi umumnya dilakukan secara manual dengan mendaftar melalui form dan wawancara manual (wawancara dengan perusahaan) secara langsung. Pada saat ini (tahun 2025), sistem pendaftaran dilakukan melalui sebuah aplikasi bernama talentern, sebuah aplikasi yang digunakan untuk mengkoordinasi magang para mahasiswa. Walaupun talentern bertugas untuk mengkoordinasi magang mahasiswa, wawancara untuk seleksi masih dilakukan secara manual.

#### 3.3.2 Pengembangan Sistem

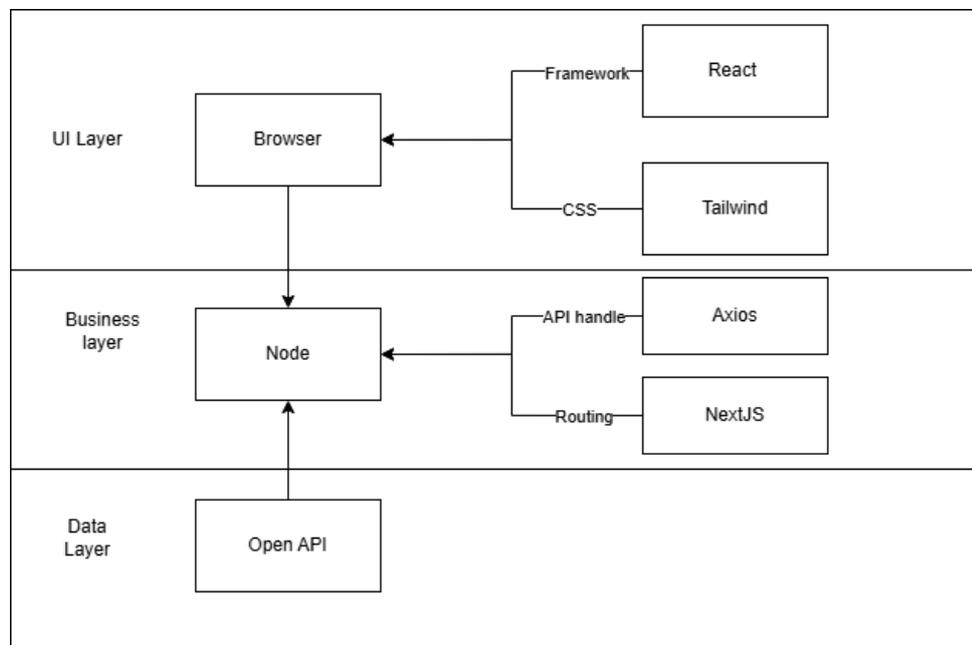
Talentern chatbot merupakan sebuah fitur pada aplikasi Talentern yang digunakan untuk mewawancarai kandidat magang terkait *softskills* yang dimilikinya. Selain menilai chatbot ini juga bertugas untuk menyimpulkan hasil wawancara kandidat magang, yang nantinya hasil nilai dan kesimpulan ini akan diinformasikan kepada mitra Perusahaan. Untuk menjaga kredibilitas aplikasi, pertanyaan yang diberikan kepada kandidat merupakan pertanyaan hasil wawancara dengan psikolog. Bukan hanya pertanyaan, tetapi dataset penilaian kandidat yang diinformasikan kepada mitra juga merupakan hasil wawancara dengan psikolog.

Pada pengembangan prototipenya talentern chatbot menggunakan bahasa *Typescript* dan *framework ReactJS*. *Framework* ini digunakan untuk mengembangkan chatbot dikarenakan *library*-nya yang sangat berguna untuk melakukan *render data* yang *realtime* dengan mudah. Adapun *tools* lain yang digunakan dalam pengembangan prototipenya seperti model dari openAI untuk percobaan *finetuning model*.

Prototipe yang telah dibuat sebenarnya merupakan sebuah aplikasi yang terpisah, dibuat untuk mengembangkan alur dan pengujian performa dari logika chatbot yang dibuat. Setelah prototipe selesai dibuat selanjutnya adalah integrasi ke dalam aplikasi web Talentern. Saat proses integrasi, Bahasa *PHP* dan *framework Laravel* digunakan. Alasan penggunaan bahasa dan *framework* ini adalah untuk menyesuaikan dengan aplikasi talentern itu sendiri.

### 3.2.1.1 Prototype Chatbot

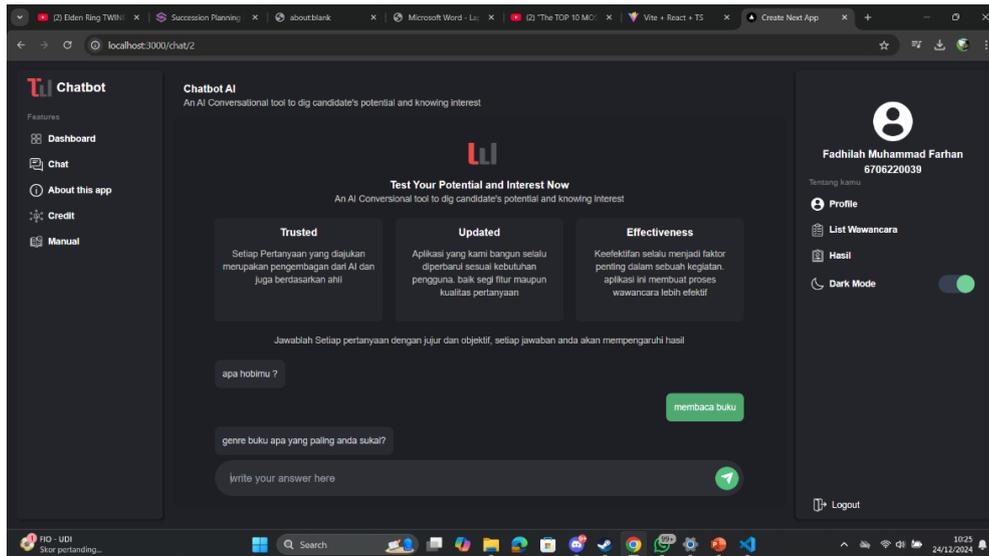
Prototipe chatbot dibuat dengan menggunakan bahasa *Typescript*, *Framework ReactJS* dan *NextJS*. Prototipe ini dibuat tanpa menggunakan database untuk penyimpanan datanya, data hanya disimpan sementara dalam sebuah *state* yang terdapat pada react. Adapun *tech stack* dalam pengembangan prototipe chatbot yang dapat dilihat pada gambar 3.6.



Gambar 3. 6 Tech Stack Prototipe

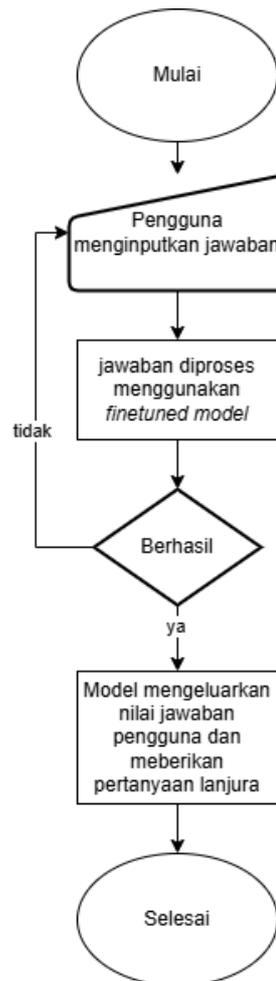
Dari gambar 3.6 terlihat ada 3 layer *tech stack* pada pengembangan prototipe chatbot. Pada layer pertama yaitu UI layer(halaman yang tampak oleh pengguna). Pada UI layer *framework React* dan *library Tailwind CSS* digunakan untuk pembuatanya. Adapun pada *Businesses layer* dimana keseluruhan sistem diatur, termasuk logika dan algoritma aplikasi, prototipe ini menggunakan Node sebagai *engine*-nya. Selain itu library dan *framework* tambahan seperti Axios untuk *handle API* dan *NextJS* untuk routing halaman juga diinstal pada prototipe ini. Yang terakhir yaitu data layer. Karena prototipe tidak dintegrasikan dengan backend, maka data yang didapat tidak disimpan kedalam database. Untuk mendapatkan data seperti penilaian, kesimpulan wawancara dan pertanyaan lanjutan, semua merupakan *AI generated* dari API tersebut.

Gambar 3.7 menunjukkan UI Layer dari prototipe chatbot yang dibuat menggunakan *React* dan *TailwindCSS*.



Gambar 3. 7 UI prototipe chatbot

Alur pemrosesan yang digunakan dalam prototipe chatbot dapat terbilang sederhana. Jawaban dan pertanyaan pengguna diproses oleh chatbot melalui *finetuned model* yang sudah dilatih sebelumnya di openAI. Lalu model akan mengembalikan nilai dan pertanyaan tambahan untuk memperdalam jawaban pengguna. Setelah semua pertanyaan selesai ditanyakan maka, chatbot akan secara otomatis menyimpulkan semua percakapan pengguna. Gambar 3.8 menunjukkan alur pemrosesan jawaban yang digunakan pada prototipe chatbot.



Gambar 3. 8 Alur pemrosesan jawaban prototipe

Prototipe chatbot yang dibuat masih menggunakan model gpt 4.0 dan *fine-tuned model* yang masih sederhana. Kedua model ini memiliki kekurangan dan kelebihan tersendiri. Model gpt-4.0 memiliki tingkat konsistensi yang rendah dan hasil jawaban yang lebih subjektif tetapi model ini dapat beradaptasi dengan baik di semua perubahan yang terjadi pada sistem maupun *prompt* yang dibuat oleh developer. Alasan tidak digunakannya model ini untuk sistem penilaian adalah ditemukan bahwa GPT-4 mengalami ketidakstabilan output, menghasilkan inkonsistensi output, termasuk munculnya referensi fiktif[12]. Di lain sisi *fine-tuned model* merupakan kebalikan dari gpt-4.0. model ini memiliki Tingkat keakuratan dan konsistensi yang lebih tinggi, tetapi tidak mampu beradaptasi dengan perubahan system dan *prompt*. Walaupun tingkat akurasi dari *fine-tuned model* tinggi model ini membutuhkan dataset yang sangat banyak dan membutuhkan waktu yang lama untuk pengumpulan datanya. Prototipe dikembangkan menggunakan model gpt-4.0 untuk kebanyakan kasus dalam proyek dan *fine-tuned model* sederhana untuk kasus yang dibutuhkan penilaian yang objektif.

Ada sebuah kelemahan yang ditemukan dalam prototipe ini, yaitu inkosistensi jawaban. Walaupun konsistensi *finetuned model* lebih tinggi, bukan berarti output

dari model itu 100% konsisten. Pada sebuah pengujian output penilaian, dilakukanlah pengiriman 10 pertanyaan dan 10 jawaban yang sama kepada prototipe chatbot. Prototipe ini menghasilkan output penilaian yang berbeda. Berikut merupakan hasil dari 10 penilaian tersebut.

Menggunakan *finetuned model*:

[9,9,9,9,9,8,9,9,9,9]

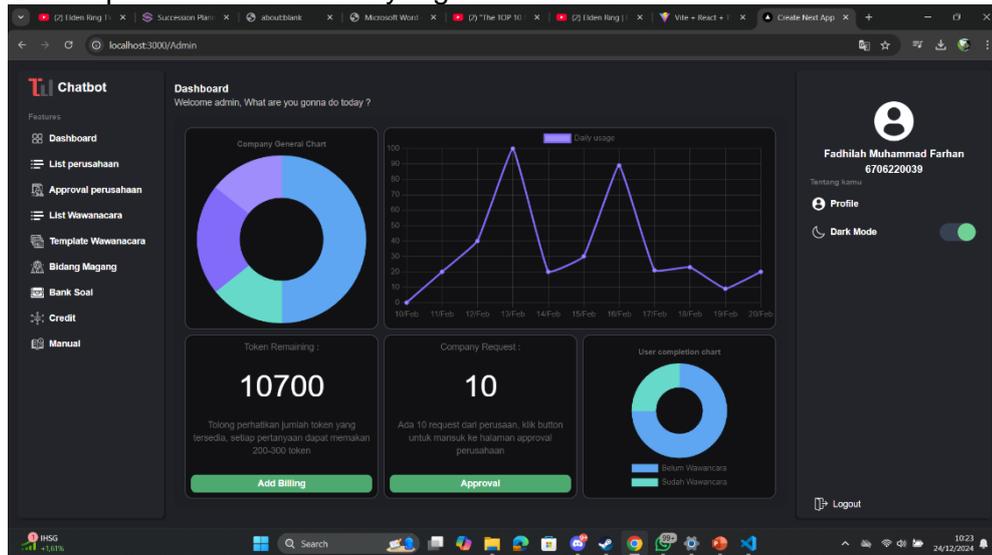
Menggunakan LLM(*Large Language Model*) gpt-4.0:

[9,9,9,9,9,8,9,9,8,8]

Dari 10 pengujian terlihat chatbot dengan menggunakan *finetuned model* menghasilkan 9 nilai 9 dan 1 nilai 8(konsistensi 90%). 3 nilai 8 dan 7 nilai 9 didapatkan dengan menggunakan LLM gpt-4.0(konsistensi 70%).

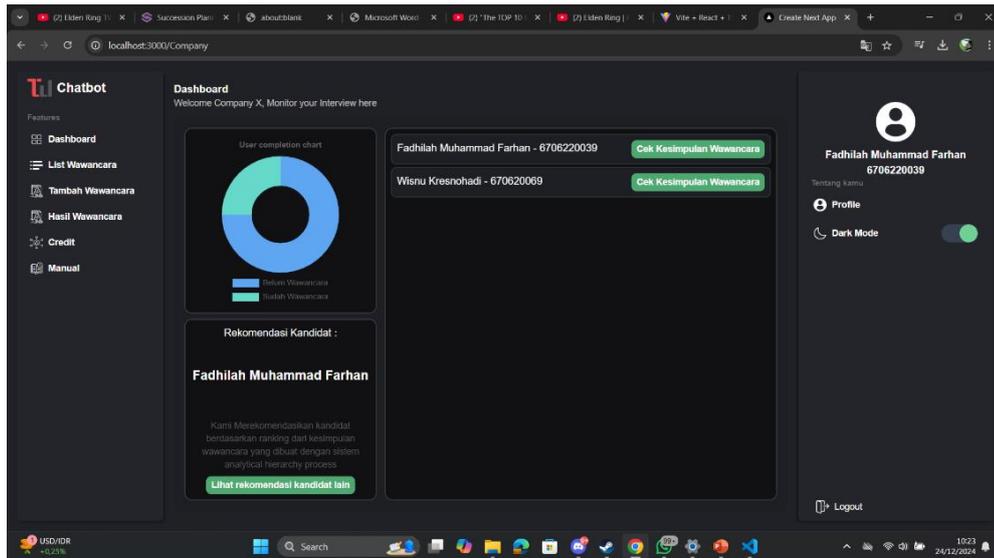
### 3.2.1.2 UI untuk Prototipe Aplikasi

UI untuk protipe dibuat untuk mempermudah uji coba dari chatbot yang telah dibuat. Dalam proyek ini user membuat UI dari semua role pengguna karena pada awalnya aplikasi chatbot ini akan berdiri secara independen sebelum akhirnya diintegrasikan dengan aplikasi yang lebih besar. Pembuatan UI ini menggunakan *ReactJS* dan juga *NextJS* untuk memudahkan proses pembangunan dan meningkatkan performa aplikasi. Gambar 3.10 sampai dengan 3.12 merupakan contoh dari UI yang telah dibuat.



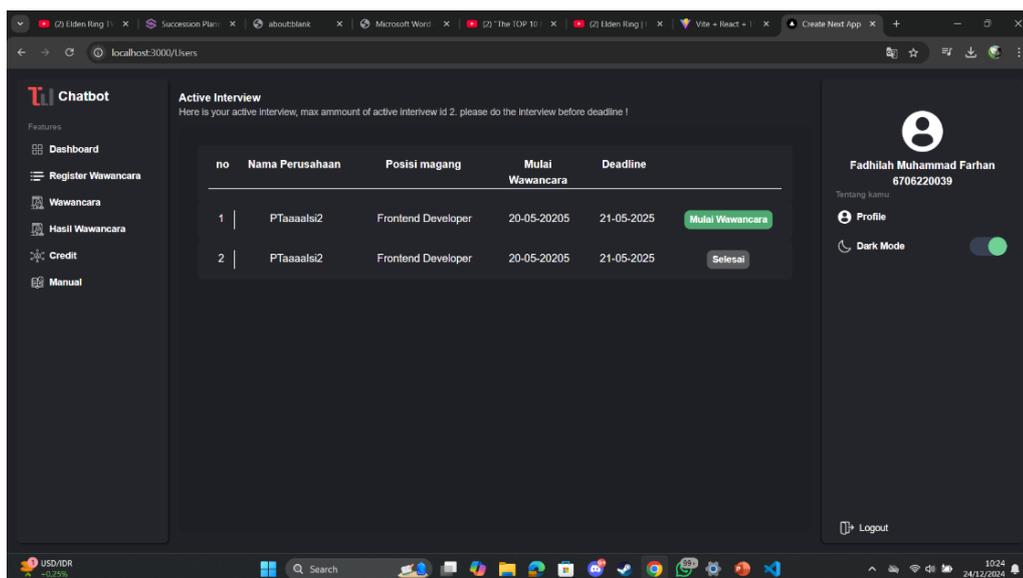
Gambar 3. 9. UI *Dashboard* admin

Gambar 3.6 menunjukkan *dashboard* dari admin. *Dashboard* ini memiliki informasi yang dibutuhkan oleh admin seperti data siswa yang sudah wawancara, penggunaan harian, sampai dengan jumlah token openAI yang masih bisa dipakai.



Gambar 3. 10 UI dashboard Perusahaan

Gambar 3.11 merupakan dashboard dari Perusahaan, yang dapat digunakan oleh mitra / Perusahaan untuk melihat data mahasiswa yang sudah wawancara dan Kesimpulan dari wawancara mahasiswa tersebut.



Gambar 3. 11 halaman list wawancara

Gambar 3.12 menjelaskan tentang halaman list wawancara yang dapat diakses oleh mahasiswa / pelamar. Pada halaman ini mahasiswa dapat memulai wawancaranya yang sudah didaftarkan sebelumnya.

### 3.2.1.3 Pengumpulan dan pengelompokan dataset

Pengumpulan dataset dilakukan dengan menggunakan 2 cara. Yang pertama adalah dengan melakukan wawancara dengan psikolog untuk mendapatkan pertanyaan dan kategori jawaban pengguna. Cara yang ke 2 adalah dengan mengumpulkan data kata kunci jawaban sebanyak banyaknya yang relevan

dengan pertanyaan wawancara, lalu mengelompokanya berdasarkan hasil wawancara dengan psikolog.

<https://docs.google.com/spreadsheets/d/1EDshTRd5Y6Wi4Gjn6umdgU0c7t1TSUyL/edit?usp=sharing&oid=116685048014166667314&rtpof=true&sd=true>

Adapun dataset pengelompokan kata kunci yang digunakan pada proyek ini dapat dilihat pada link

<https://drive.google.com/file/d/103Z4rX4y-MOWNDPaSteJ9--rE58P3IKL/view?usp=sharing>

### 3.2.1.4 Pengolahan dataset

Dataset yang dikumpulkan sebelumnya bukanlah dataset yang dapat langsung digunakan. Dataset tersebut haruslah diolah dulu. Ada beberapa langkah yang digunakan untuk mengolah data pada proyek ini diantaranya :

1. Mengubah kata kunci jawaban menjadi vector embedding.  
Perhitungan *cosine similarity* membutuhkan vektor numerik untuk membuat perhitunganya. Langkah ini dilakukan untuk mengubah semua kata kunci jawaban menjadi vektor numerik sebelum disimpan dala pinecone database

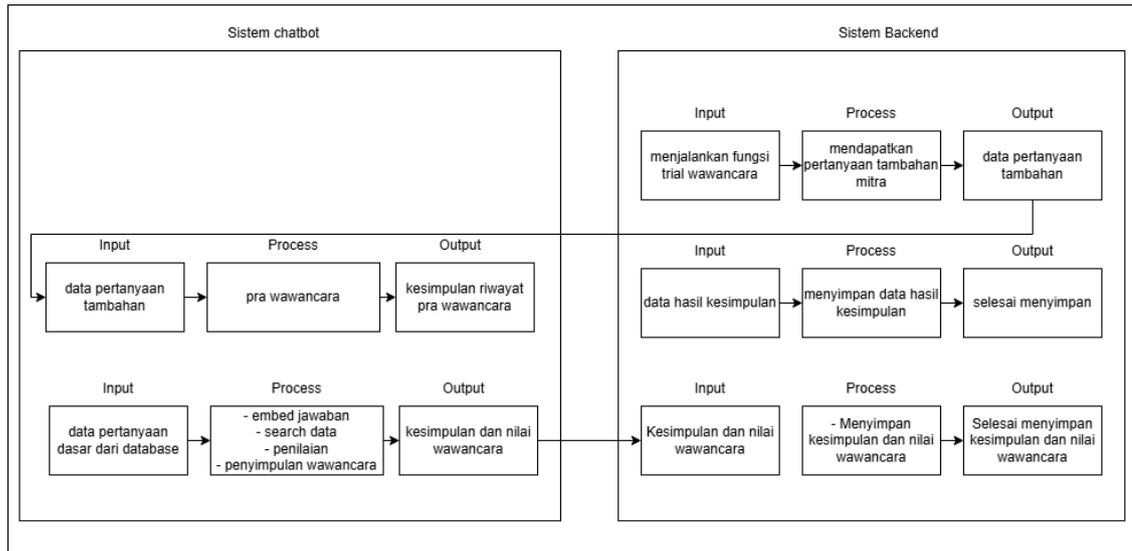
2. Mengubah struktur data menjadi JSON sesuai format pada pinecone database.

Sebelum data dimasukan kedalam database, data diolah dulu dengan memberikan metadata dan id. Dimana setiap data memiliki id dan meta data unik. Data yang sudah diolah dapat dilihat pada link berikut.  
[https://drive.google.com/file/d/1m1Whke1wfbn47EQzIXD\\_FMPWEgGOWiv1/view?usp=sharing](https://drive.google.com/file/d/1m1Whke1wfbn47EQzIXD_FMPWEgGOWiv1/view?usp=sharing)

### 3.3.1.5 Chatbot yang sudah terintegrasi

Setelah prototipe selesai dibuat, tahap selanjutnya adalah integrasi ke aplikasi Talentern dan menjadi sebuah fitur yang dinamakan Talentern Chatbot. Ada perubahan metode penilaian yang digunakan untuk model AI dari prototipe dan saat sudah di integrasi. Dalam pengembangan sistem penilaian, model prototipe chatbot hanya menggunakan model gpt-4.0 yang telah di-*finetuning* sebelumnya. Model ini memiliki sebuah kelemahan yang Dimana outputnya tidak stabil. Saat di uji coba dengan pertanyaan dan jawaban yang sama untuk 10 kali *request*, terdapat perbedaan nilai yang dikeluarkan oleh model ini. Oleh karena itu dibuatlah system penilaian menggunakan metode *word embedding* dan perhitungan *cosine similarity*. Tidak seperti penilaian dengan model besar seperti gpt-4.0 yang menggunakan NLP yang kompleks dan tidak memiliki konteks, metode sederhana seperti word embedding dan *cosine similarity* lebih baik digunakan. Secara sederhana, jawaban pengguna diubah menjadi vektor numerik menggunakan metode *word embedding*, Lalu dicari keimripan katanya berdasarkan vektor numerik tersebut dengan menggunakan perhitungan *cosine similarity* dan dicari jawaban(dari dalam dataset) yang paling mendekati vector jawaban pengguna. Setelah ditemukan datanya kemudian chatbot akan mengembalikan nilai pengguna berdasarkan dataset yang tersedia. Dengan menggunakan metode ini chatbot bisa mengembalikan nilai yang stabil karena perhitungan *cosine similarity* bersifat *absolute* dan tidak berubah-ubah.

Chatbot yang sudah terintegrasi terdiri dari 2 sistem. Gambar 3.12 menunjukkan pembagian sistem tersebut.



Gambar 3. 12 pembagian sistem chatbot

Dokumen berjudul “PENGEMBANGAN MODEL DAN PEMBANGUNAN AI APLIKASI CHATBOT WAWANCARA MAGANG UNTUK UNIT LKM” berfokus membahas sistem chatbotnya. Ada 2 hal yang dibuat dalam sistem ini, yang pertama *trial* atau pra wawancara dan yang ke 2 wawancara utama. Input dari trial wawancara merupakan data pertanyaan wawancara tambahan yang diajukan oleh perusahaan, setelah melalui proses pra-wawancara, percakapan pengguna dengan bot akan disimpulkan. Yang kedua adalah wawancara utama, input dari sistem ini adalah data pertanyaan dari database. Ada beberapa hal yang terjadi dalam proses tersebut diantaranya, perubahan jawaban pengguna menjadi vektor, pencarian data terdekat, penilaian jawaban pengguna, dan mengeluarkan kesimpulan wawancara. Output akhir dari sistem chatbot adalah kesimpulan dan hasil wawancara.

### 3.4 Kebutuhan Perangkat Kerja

Dalam pengembangan talentern chatbot beberpa perangkat lunak dan perangkat keras digunakan. Table 3.1 dan 3.2 menunjukan kebutuhan teknologi yang digunakan selama pengembangan proyek Talentern chatbot.

#### 3.3.1 Kebutuhan Perangkat Pengembangan Sistem

Tabel 3.1 menunjukan Kebutuhan perangkat keras dari proyek Talentern Chatbot.

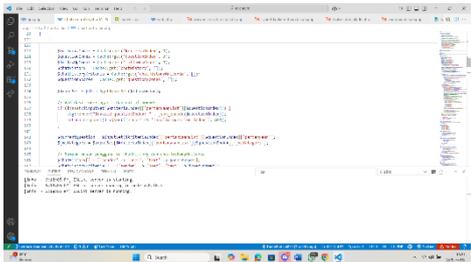
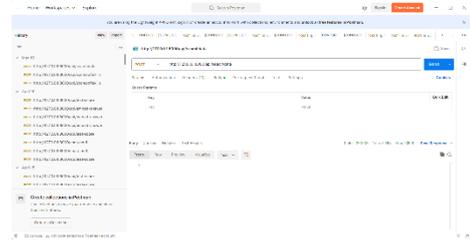
Tabel 3. 1 kebutuhan perangkat keras

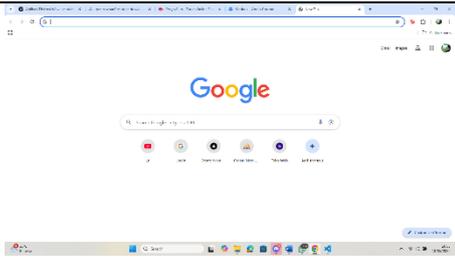
No	Perangkat keras	Penjelasan	Gambar
1	Laptop ASUS TUF F15	Processor intel core i5-12500H, Memory: DDR4 16GB, GPU: NVIDIA RTX 3050, Storage: 1TB, Sistem Operasi: Windows 11	

### 3.3.2 Kebutuhan Perangkat Implementasi Sistem

Tabel 3.2 menunjukkan Kebutuhan perangkat lunak dari proyek Talentern Chatbot.

Tabel 3. 2 Kebutuhan perangkat lunak

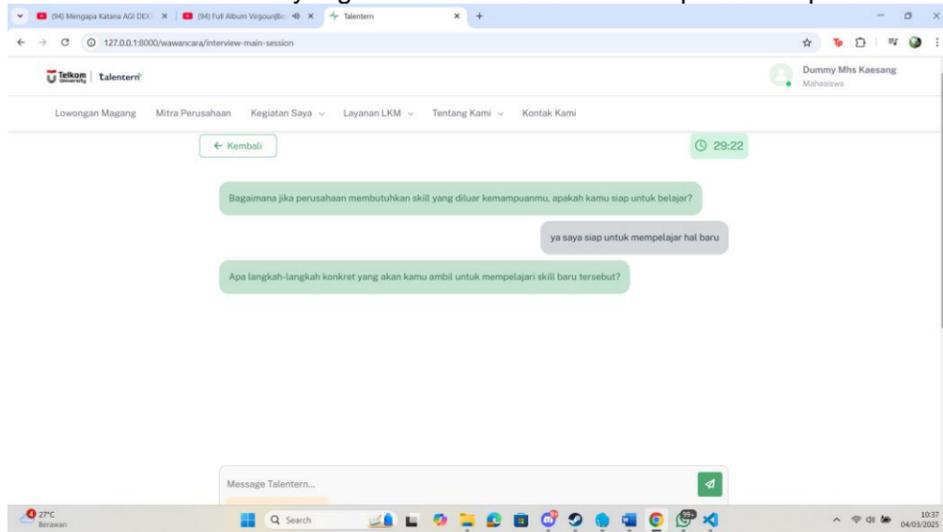
No	Perangkat Lunak	penjelasan	Gambar
1	Visual Studio Code	Visual Studio Code adalah perangkat lunak yang digunakan penulis untuk menulis seluruh kode program. Dengan API yang dapat diperluas melalui penggunaan ekstensi, VS Code menjadi IDE yang populer di kalangan developer saat ini[13]	 <p>Gambar 3. 13 VsCode versi 1.100</p>
2	Postman	Postman adalah perangkat lunak yang digunakan untuk memudahkan pengembang dalam pembuatan, distribusi, pengujian, dan dokumentasi API[14]	 <p>Gambar 3. 14 Postman versi 11.44.0</p>

3	Google Chrome	<p>Google chrome merupakan sebuah browser yang digunakan penulis untuk menguji coba prototype yang telah dibuat</p>	 <p><i>Gambar 3. 15 Google Chrome versi 136.0.7103.93</i></p>
---	---------------	---	---

## BAB IV HASIL DAN PEMBAHASAN

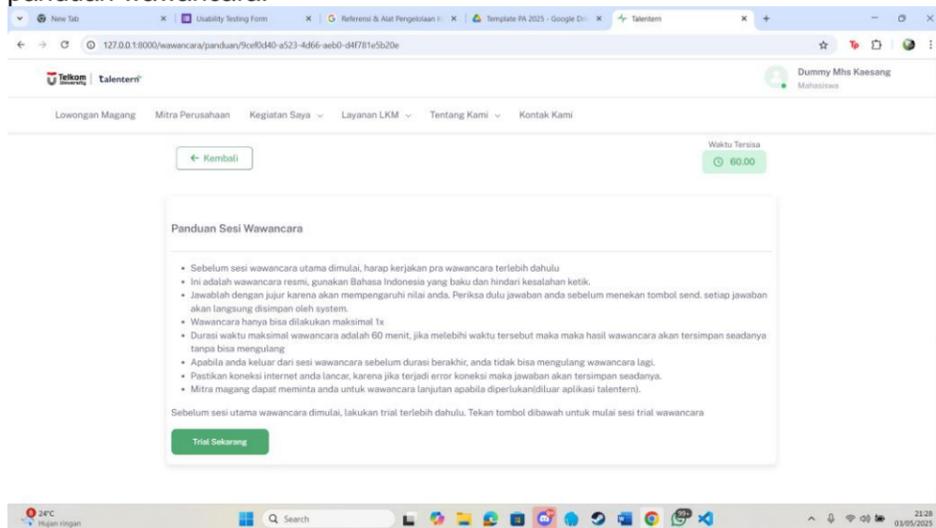
### 4.1 Hasil Akhir (Luaran)

Talentern Chatbot merupakan sebuah fitur pada aplikasi talentern. Fitur ini bertugas untuk mewawancarai kandidat magang terkait dengan softskills yang mereka miliki dengan menggunakan chatbot. Pada penggunaannya, chatbot ini bukan hanya sekedar mewawancarai saja melainkan juga menilai softskill berdasarkan jawaban pengguna dan memberikan Kesimpulan wawancara kepada mitra atau Perusahaan yang dilamar. Gambar 4.1 merupakan tampilan dari Talentern Chatbot.



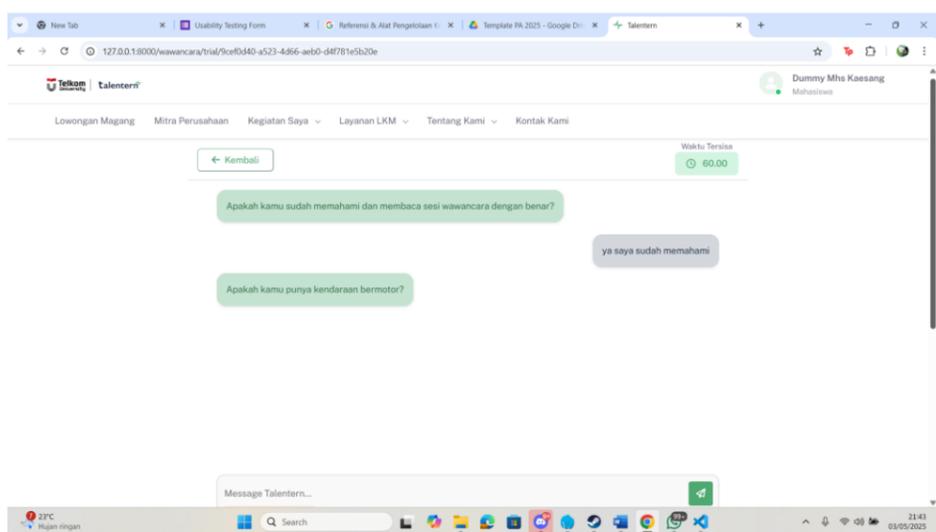
Gambar 4. 1 Talentern Chatbot

Selain dari chatbot yang merupakan halaman utama, adapun halaman penunjang lainnya yang dibutuhkan dalam alur bisnis talentern chatbot seperti halaman panduan wawancara, pra-wawancara dan halaman syarat dan ketentuan. Gambar 4.2 menunjukan halaman panduan wawancara.



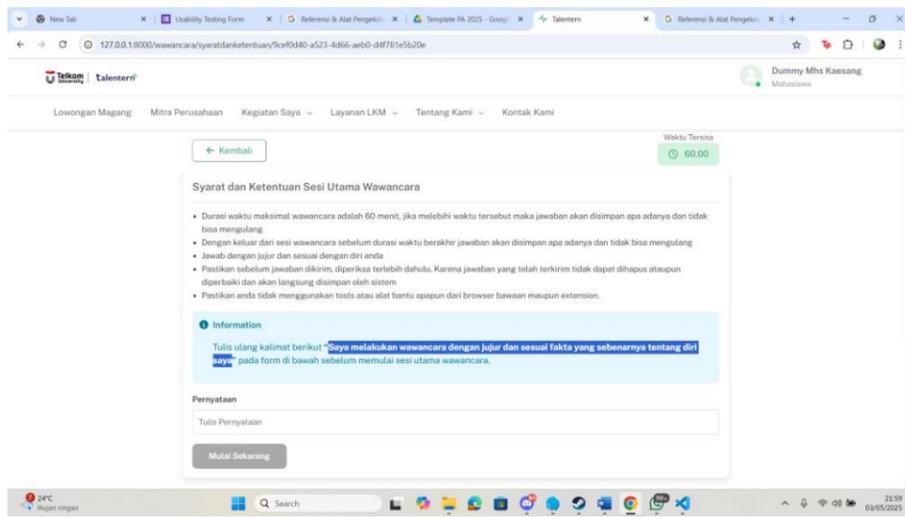
Gambar 4. 2 Panduan sesi wawancara

Pada halaman panduan wawancara disediakan tatacara dan aturan penggunaan Talentern Chatbot. Adapaun tombol "Trial sekarang" yang akan mengarahkan pengguna ke halaman pra-wawancara atau trial wawancara. Gambar 4.3 menunjukan halaman pra-wawancara



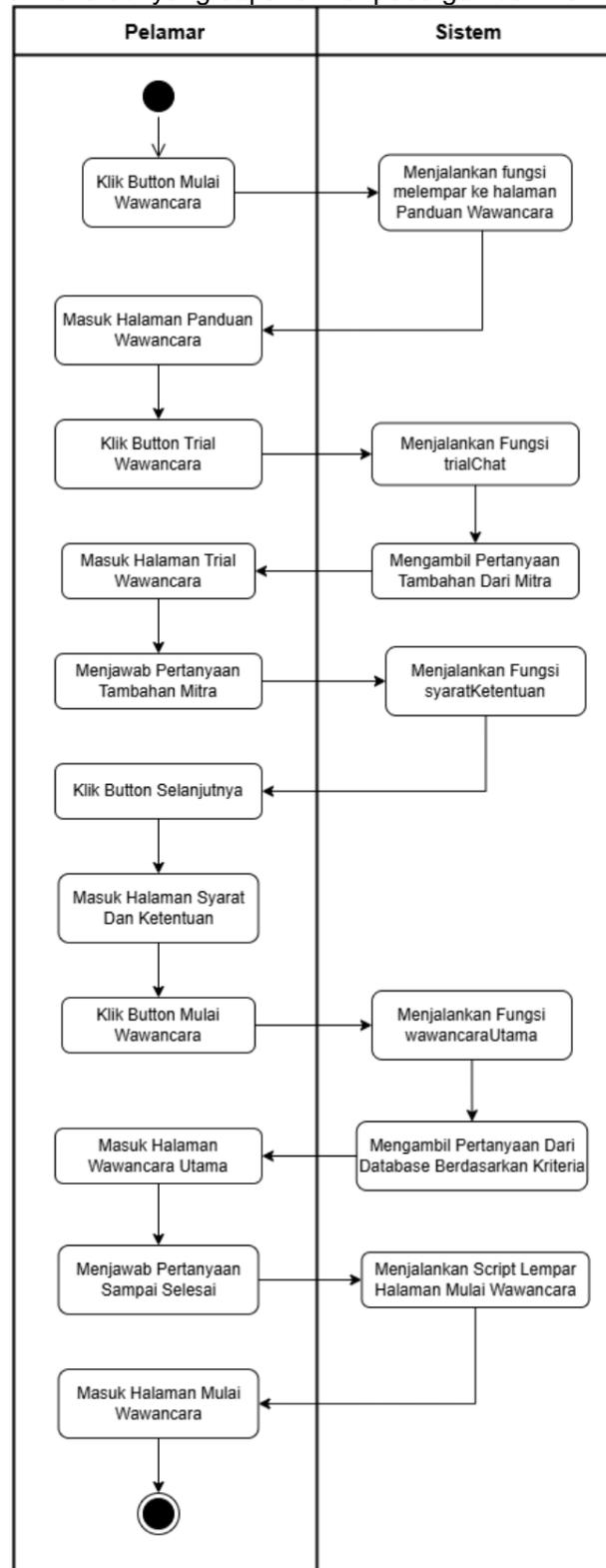
Gambar 4. 3 Halaman pra-wawancara

Setelah pengguna selesai melakukan trial atau pra-wawancara, pengguna akan diarahkan ke halaman syarat dan ketentuan. Di halaman ini pengguna akan diberitahukan syarat dan ketentuan wawancara, selain itu terdapat sebuah konfirmasi Ketika pengguna sudah menyetujui syarat dan ketentuan dengan mengetikkan kalimat "**Saya melakukan wawancara dengan jujur dan sesuai fakta yang sebenarnya tentang diri saya**". Satu saja kesalahan pengetikan pengguna tidak bisa menekan tombol mulai wawancara. Gambar 4.4 menunjukan halaman syarat dan ketentuan wawancara.



Gambar 4. 4 Syarat dan ketentuan wawancara

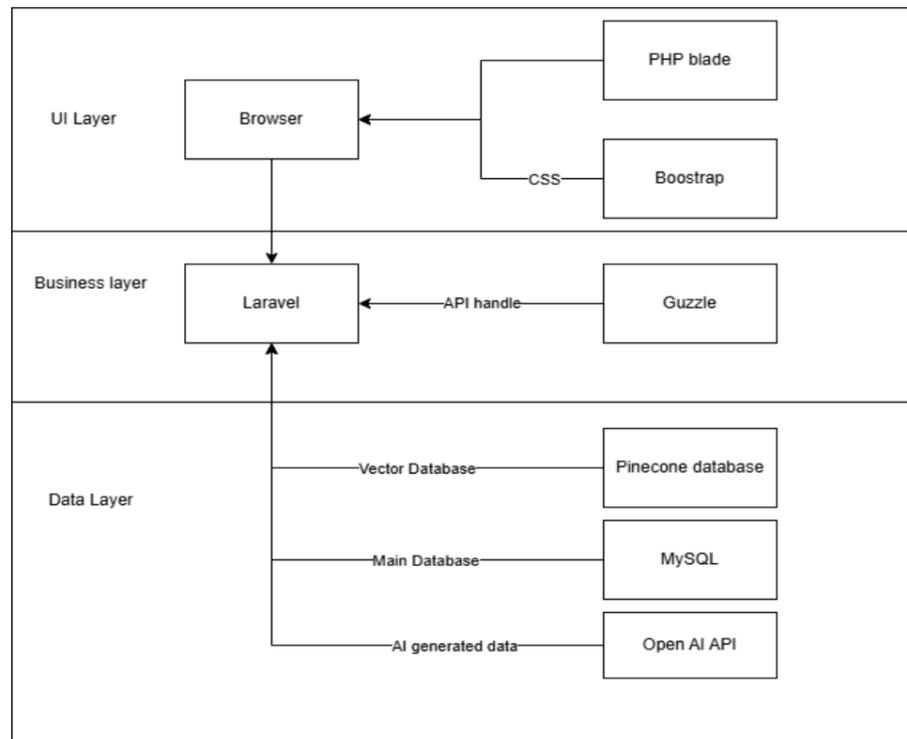
Adapun proses kesuluran dari fitur talentern chatbot yang dapat dilihat pada gambar 4.5



Gambar 4. 5 activity diagram taletern chatbot

Berdasarkan gambar 4.5, secara sederhana Talentern Chatbot memiliki 4 tahap dalam penggunaannya. Yang pertama adalah panduan, lalu pengguna akan diarahkan ke halaman pra wawancara. Setelah selesai melakukan pra wawancara pengguna akan diberitahukan syarat dan ketentuan dari wawancara utama. Tahap terakhir adalah mengerjakan wawancara sampai selesai.

Tidak seperti prototipe yang dibuat sebelumnya, Talentern Chatbot diintegrasikan ke aplikasi Talentern menggunakan *framework Laravel* dan bahasa *PHP*. Selain itu Talentern Chatbot juga sudah terkoneksi dengan database dari talentern yang menggunakan *MySQL*. Tech Stack dari talentern chatbot dapat dilihat di gambar 4.6.

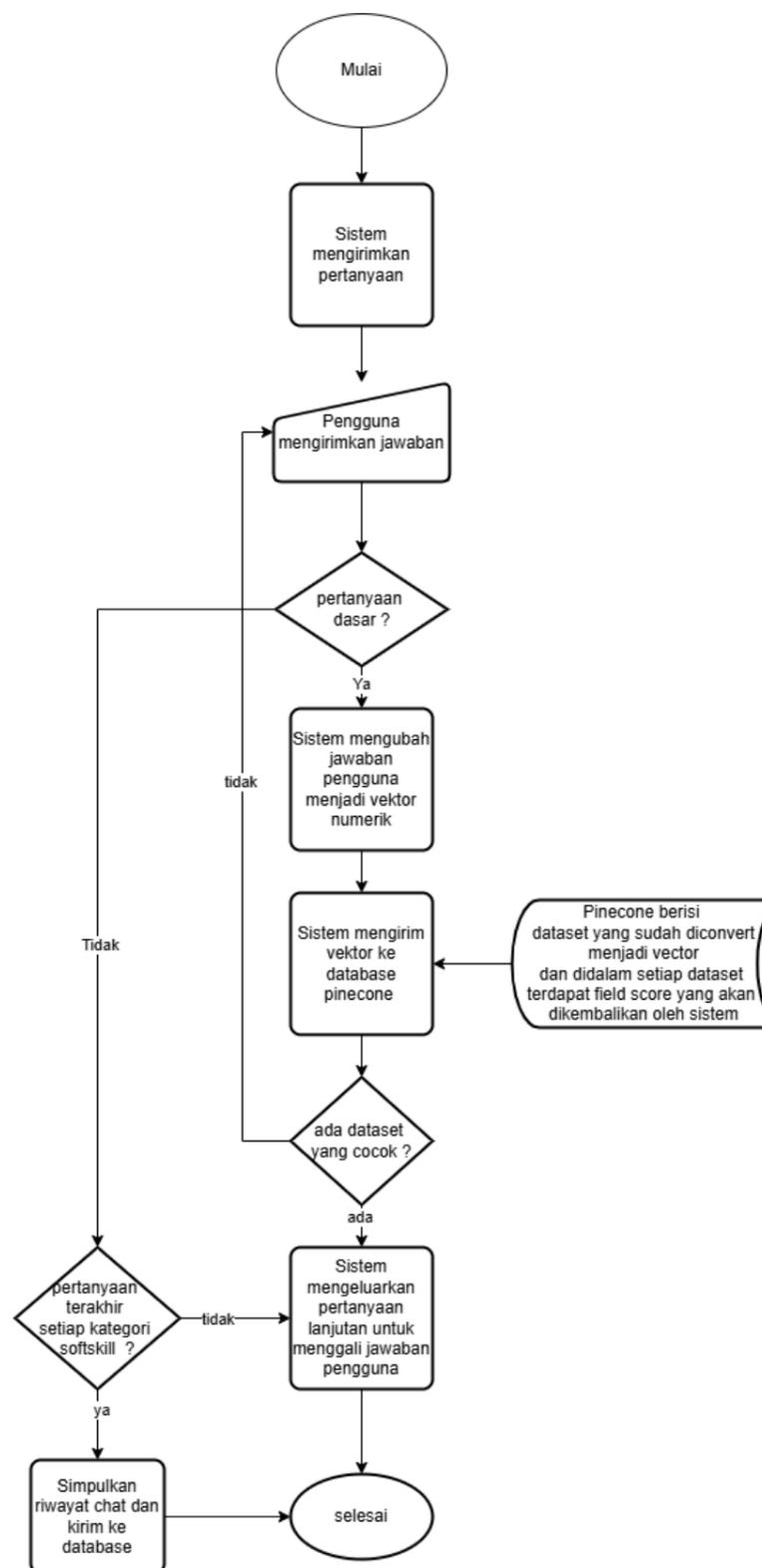


Gambar 4. 6 Tech stack Talentern Chatbot

Seperti pada prototipe yang dibuat sebelumnya, Talentern Chatbot yang sudah terintegrasi dengan aplikasi Talentern memiliki 3 layer. Walaupun demikian teknologi yang digunakan berbeda dari prototipe yang telah dibuat. Pada UI layer *PHP Blade* dan *Bootstrap* CSS digunakan. *Blade* adalah mesin *templating* yang digunakan dalam *framework Laravel* untuk menggabungkan logika *PHP* dengan *HTML*. Adapun dari business layer Talentern Chatbot menggunakan *framework Laravel* untuk logika dan mengatur keseluruhan sistemnya. Pada *business layer* juga dipasang *library* *Guzzle* untuk *API handle*-nya. Yang terakhir adalah *data layer*. Ada 3 teknologi yang digunakan pada *data layer*. Pertama ada *pinecone database* yang digunakan untuk menyimpan vektor dan melakukan pencarian data vektor. Selanjutnya ada *MySQL* yang merupakan database utama. Digunakan untuk menyimpan data data yang berkaitan dengan keseluruhan sistem seperti data perusahaan, lowongan, wawancara, dan hasil wawancara. Terakhir ada *OpenAI API*. Semua data yang merupakan *AI Generated* berasal dari sini. Contohnya mengambil pertanyaan lanjutan dan menyimpulkan riwayat wawancara.

#### 4.2 Alur Penilaian Jawaban Pengguna

Seperti yang disebutkan sebelumnya, Talentern Chatbot bukan hanya mewawancarai kandidat / pelamar magang, melainkan juga menilai *softskills* dari pengguna dan menyimpulkan riwayat chat pengguna. alur dari pemrosesan jawaban pengguna dapat dilihat pada gambar 4.5.



Gambar 4. 7 Alur pemrosesan jawaban pengguna

Dapat dilihat dari flowchart 4.5 bahwa saat halaman dibuka, chatbot akan langsung mengirimkan pertanyaan berdasarkan kriteria yang dipilih oleh mitra. Selanjutnya sistem akan mengkonfirmasi, apakah ini pertanyaan dasar (pertanyaan dari database) atau pertanyaan lanjutan (pertanyaan *Generated AI*). Jika ini merupakan pertanyaan dasar, maka sistem akan mengubah jawaban pengguna menjadi sebuah vektor numerik. Vektor numerik ini kemudian dikirimkan ke pinecone database untuk dibandingkan dan dicari dataset jawaban yang paling dekat dengan vektor jawaban pengguna menggunakan perhitungan *cosine similarity*. Berikut merupakan contoh dari dataset yang disimpan dalam database pinecone.

1	<p><b>ID:</b> 01263bd4-1efd-4ee6-bf5b-f7c91e8bf731</p> <p><b>answer:</b> "[NEU]cenderung suka memimpin"</p> <p><b>kategori:</b> 4</p> <p><b>kriteria:</b> "Kepemimpinan"</p> <p><b>question:</b> "Seberapa sering anda memimpin sebuah tim atau kelompok?"</p>
SCORE	0.9999

Gambar 4. 8 dataset jawaban pengguna di pinecone

Dari gambar 4.6 terlihat bahwa setiap index dalam tabel tersebut menyimpan beberapa field atau metadata. Ada answer yang menyimpan sample jawaban, kategori merupakan penilaian dari sample jawaban yang dikelompokkan berdasarkan wawancara dari psikolog. Lalu ada juga kriteria dari softskills dan pertanyaan yang digunakan untuk filter data pencarian jawaban. Setelah jawaban dikirim ke dalam database pinecone maka, jika hasil perhitungan *cosine similarity* nya lebih dari 0.45, maka sistem akan mengembalikan nilai (dari field kategori). Sebaliknya jika *cosine similarity*nya kurang dari 0.45, maka dia akan menanyakan kembali maksud dari jawaban pengguna untuk mengkonfirmasi ulang. Setelah melakukan penilaian sistem akan mengeluarkan pertanyaan lanjutan yang bertujuan untuk menggali jawaban pengguna lebih dalam lagi.

Untuk kasus lain yaitu jika jawaban yang diproses merupakan jawaban dari pertanyaan lanjutan, maka jawaban tersebut tidak dinilai melainkan hanya sebagai data tambahan untuk menggali lebih dalam terkait dengan jawaban sebelumnya. Setiap pertanyaan dasar memiliki 2 pertanyaan lanjutan yang digunakan untuk memperdalam jawaban pengguna. Ada risiko yang ditanggung pengguna jika pengguna menjawab pertanyaan dasar diluar konteks pertanyaan yang diajukan oleh chatbot. Nilai pengguna pada pertanyaan itu

akan dianggap 0, bahkan ketika pengguna mencoba memperbaiki konteks jawaban di pertanyaan lanjutan. Apabila dari semua pertanyaan dasar pengguna menjawab diluar konteks pertanyaan maka secara otomatis nilai pengguna dari kriteria yang ditanyakan akan menjadi 0. Untuk mengatasi ini pengguna diharuskan benar benar memahami konteks dari pertanyaan sebelum menjawab pertanyaan.

Kondisi yang terakhir jika pertanyaan tersebut merupakan pertanyaan lanjutan dari pertanyaan dasar terakhir dari sebuah kriteria, maka seluruh riwayat chat dalam kriteria tersebut akan disimpulkan dan disimpan kedalam database.

### 4.3 Perhitungan nilai

Setiap kriteria memiliki jumlah pertanyaan yang berbeda dan setiap pertanyaan dasar memiliki skala penilaian yang berbeda. Data skala penilaian didapatkan dari hasil wawancara dengan psikolog. Setiap pertanyaan dapat berskala nilai 3 – 5. Tabel 4.1 menunjukkan pengelompokan kategori jawaban dari kriteria manajemen waktu.

Tabel 4. 1 kategori jawaban pertanyaan kriteria Manajemen Waktu

Manajemen waktu	Bagaimana anda mengatur jadwal keseharian anda?	1. tidak terjadwal
		2. jarang menjadwalkan
		3. hanya yang penting
		4. terbiasa mengatur
	Jika anda memiliki tugas yang sudah tenggat waktu, Apa yang anda lakukan?	1. menyerah
		2. mencoba menyelesaikan lalu kumpulkan seadanya
		3. berusaha menyelesaikan
	Apakah anda suka menunda pekerjaan?	1. saya sering menunda dan tidak menyelesaikan tepat waktu
		2. saya suka menunda dan terkadang tidak tepat waktu
		3. saya suka menunda tapi selseai tepat waktu
		4. saya tidak pernah menunda dan tepat waktu

Dapat dilihat dari tabel 4.1 bahwa pertanyaan “Bagaimana anda mengatur jadwal keseharian anda?” memiliki 4 pengelompokan kategori jawaban, “Jika anda memiliki tugas yang sudah tenggat waktu, Apa yang anda lakukan?” 3 kategori, dan “Apakah anda suka menunda pekerjaan?” 4 kategori. Semakin tinggi angkanya maka semakin tinggi juga nilainya. Nilai akhir dari setiap kriteria didapatkan dengan mengalikan rata rata nilai jawaban dengan 100. Untuk mendapat kan rata rata dari nilai akhir maka digunakanlah rumus:

A = Nilai yang dikeluarkan chatbot  
 B = Jumlah kategori dari tiap pertanyaan  
 C = Jumlah pertanyaan

$$\frac{A_1 + A_2 + A_3 + \dots}{B_1 + B_2 + B_4 + \dots} \times 100 \quad (2)$$

Dari rumus tersebut terlihat pembilang memerlukan penyeteraan terlebih dahulu karena adanya perbedaan jumlah kategori jawaban dari setiap pertanyaan dasar. Penyeteraan dilakukan dengan operasi penjumlahan pecahan dari Nilai yang dikeluarkan chatbot dan Jumlah kategori dari tiap pertanyaan. Setelah disetarakan, hasil penjumlahan dari pecahan tersebut akan dibagi dengan jumlah pertanyaan dalam kriteria tersebut untuk mendapat rata rata. Tujuan perkalian 100 pada perhitungan ini adalah untuk mengubah nilai menjadi skala 100.

Contoh perhitungan:

Didapatkan penilaian dari chatbot dari sebuah pengujian seperti pada tabel 4.2

Tabel 4. 2 hasil penilaian kriteria manajemen waktu

Pertanyaan	Jawaban	Nilai	Jumlah Kategori
Bagaimana anda mengatur jadwal keseharian anda?	Saya terbiasa mengatur jadwal saya dengan membuat to do list harian	4	4
Apakah anda suka menunda pekerjaan?	Saya sering menunda pekerjaan, terutama pekerjaan yang saya anggap ringan	2	4
Jika anda memiliki tugas yang sudah tenggat waktu, Apa yang anda lakukan?"	Saya akan tetap mengerjakan sampai selesai	3	3

Berdasarkan rumus yang disebutkan sebelumnya, data penilaian dari tabel 4.2 dapat dihitung sebagai berikut.

$$\frac{\frac{4}{4} + \frac{2}{4} + \frac{3}{3}}{3} \times 100$$

$$= \frac{30}{3} \times 100$$

$$= 0.83 \times 100$$

$$= 83$$

Kesimpulanya, dari perhitungan tersebut, jawaban pengguna pada tabel 4.2 mendapatkan nilai 83.

#### 4.4 Pengujian Luaran

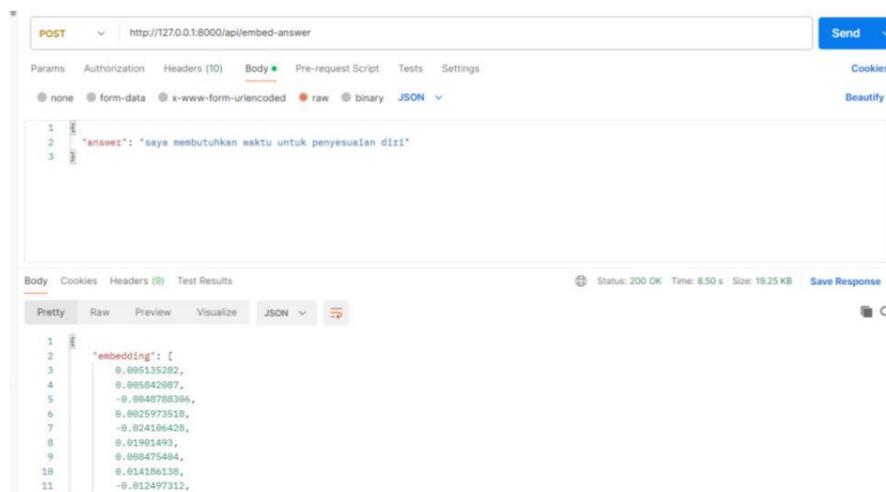
Pengujian luaran proyek dilakukan dengan 2 metode. Pengujian itu meliputi Pengujian API dan Pengujian usabilitas.

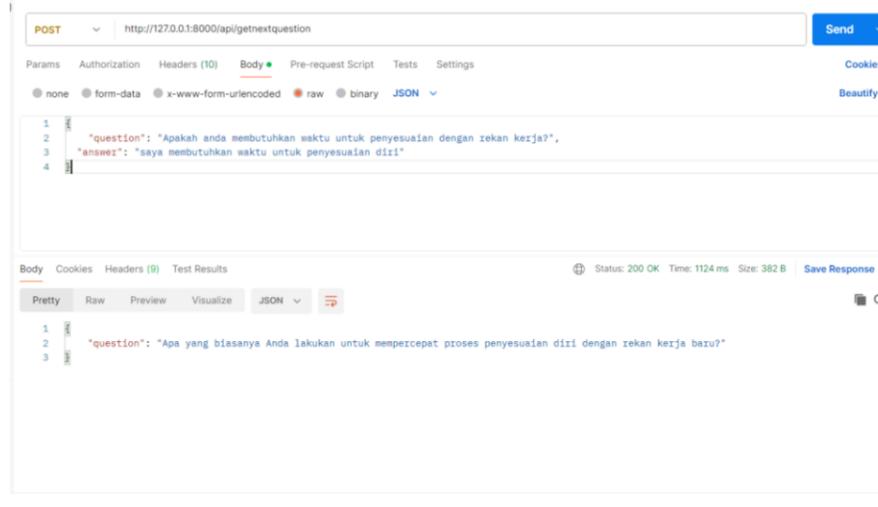
##### 4.3.1 Pengujian API

Seperti yang jelaskan pada gambar 4.7 Talentern chatbot membutuhkan 3 request ke service untuk memproses pertanyaan dasar (*embedding data, scoring* ke database pinecone, dan mengambil pertanyaan lanjutan dari OpenAI), 1 request untuk pertanyaan lanjutan (hanya mengambil pertanyaan lanjutan dari OpenAI), dan 1 request untuk menyimpulkan jawaban. Berikut merupakan hasil API request testing untuk pemrosesan jawabanya:

###### 1. Embedding data

Embedding data merupakan proses untuk mengubah kalimat (jawaban pengguna) menjadi sebuah vector numerik. Hasil dari API ini adalah sebuah vector numerik yang nantinya akan dikirimkan ke database pinecone.

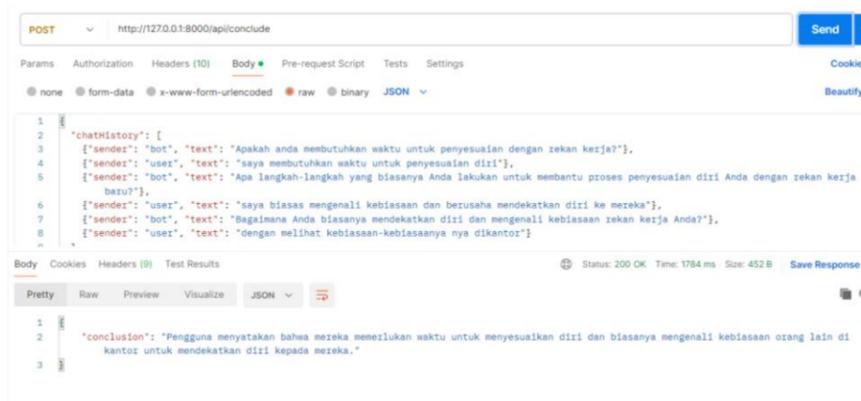




Gambar 4. 11 get-Next-question request

#### 4. Penyimpulan hasil wawancara

Pada pertanyaan terakhir disetiap kriteria, riwayat percakapan antara chatbot dan pengguna akan disimpulkan dengan service dari openAI juga. Gambar 4.10 merupakan uji coba request pengambilan kesimpulan wawancara.



Gambar 4.12 Request Penyimpulan riwayat percakapan

### 4.3.1 Pengujian Usabilitas

#### 4.3.1.1 Metode pengujian

Pengujian *usability* terhadap chatbot dilakukan dengan menggunakan metode *System Usability Scale (SUS)*. SUS sendiri berisi 10 pertanyaan, nomer ganjil untuk pertanyaan positif dan nomer genap untuk negatif[15]. Setiap pernyataan dinilai oleh partisipan menggunakan skala Likert 5 poin, mulai dari *Sangat Tidak Setuju* (1) hingga *Sangat Setuju* (5). SUS dirancang untuk mengevaluasi persepsi pengguna terhadap kemudahan penggunaan sistem secara keseluruhan, tanpa memerlukan waktu atau proses analisis yang kompleks.

Berikut adalah 10 pernyataan yang diajukan dalam kuesioner SUS:

1. Saya akan senang jika menggunakan sistem ini sering kali.
2. Saya merasa sistem ini tidak perlu didesain ulang secara menyeluruh sebelum bisa digunakan secara efisien.
3. Saya merasa sistem ini mudah digunakan.
4. Saya merasa bahwa saya membutuhkan bantuan teknis untuk dapat menggunakan sistem ini.
5. Saya merasa fitur-fitur dalam sistem ini saling terintegrasi dengan baik.
6. Saya merasa terdapat terlalu banyak inkonsistensi dalam sistem ini.
7. Saya membayangkan kebanyakan orang akan bisa belajar menggunakan sistem ini dengan sangat cepat.
8. Saya merasa sistem ini terlalu rumit untuk digunakan.
9. Saya merasa percaya diri saat menggunakan sistem ini.
10. Saya harus banyak belajar sebelum dapat menggunakan sistem ini dengan baik.

Dalam pengujian ini, chatbot diuji oleh 20 orang dengan latar belakang mahasiswa. Masing-masing partisipan diminta untuk berinteraksi langsung dengan chatbot dalam skenario penggunaan yang telah disiapkan sebelumnya, kemudian diminta untuk mengisi kuesioner SUS berdasarkan pengalaman mereka.

Data dari kuesioner kemudian dikonversi ke skor SUS sesuai pedoman perhitungan standar, yaitu dengan mengalihkan skor tiap item berdasarkan posisinya (item ganjil dan genap diperlakukan berbeda), kemudian dijumlahkan dan dikalikan untuk menghasilkan nilai dalam rentang 0 hingga 100. Semakin tinggi nilai SUS, maka semakin baik persepsi *usability* dari sistem tersebut.

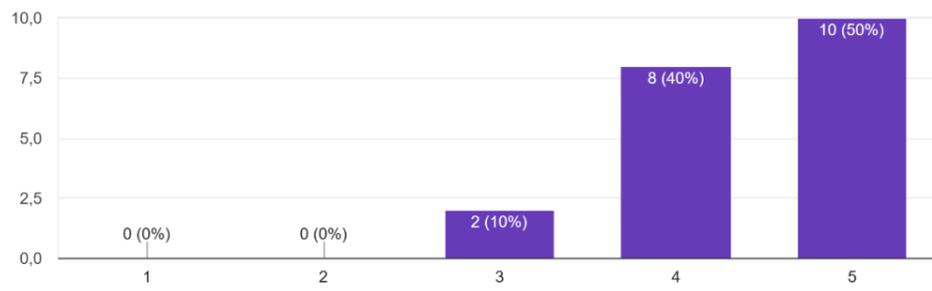
Hasil pengujian ini menjadi dasar untuk analisis lebih lanjut terkait kelebihan, kekurangan, dan potensi perbaikan terhadap sistem chatbot yang diuji.

#### 4.3.1.2 Hasil pengujian usabilitas

Pengujian dilakukan oleh 20 mahasiswa dan menghasilkan sebuah data yang direpresentasikan dalam grafik. Gambar 4.8 sampai 4.17 merupakan grafik hasil kuesioner dari setiap pernyataan yang diajukan kepada penguji

Saya berpikir akan menggunakan sistem ini lagi

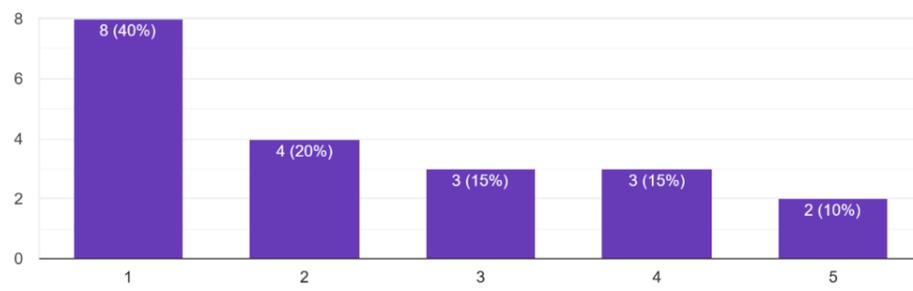
20 jawaban



Gambar 4. 13 Grafik kuesioner pernyataan 1

Saya merasa sistem ini sulit digunakan

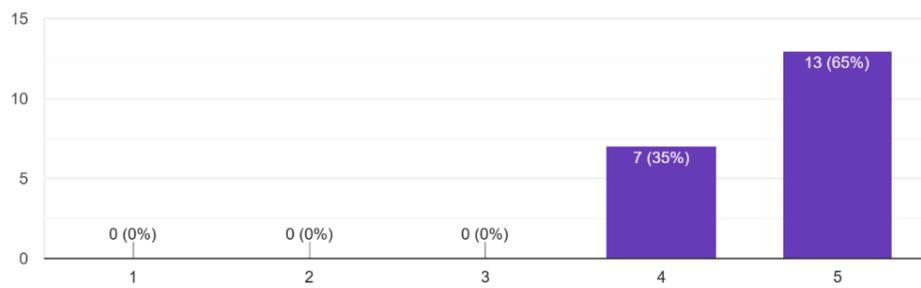
20 jawaban



Gambar 4. 14 Grafik kuesioner pernyataan 2

Saya merasa sistem ini mudah digunakan

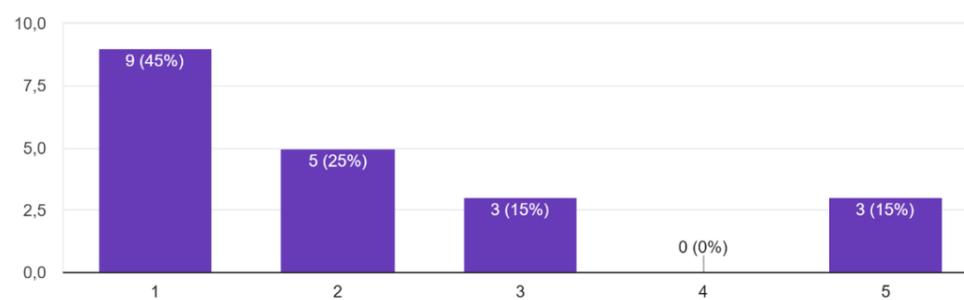
20 jawaban



Gambar 4. 15 Grafik kuesioner pernyataan 3

Saya membutuhkan bantuan dari orang lain atau teknisi dalam menggunakan sistem ini

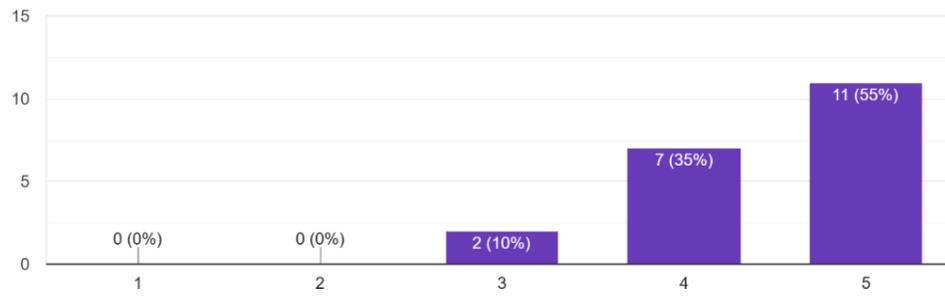
20 jawaban



Gambar 4. 16 Grafik kuesioner pernyataan 4

Saya merasa fitur-fitur sistem ini berjalan dengan semestinya

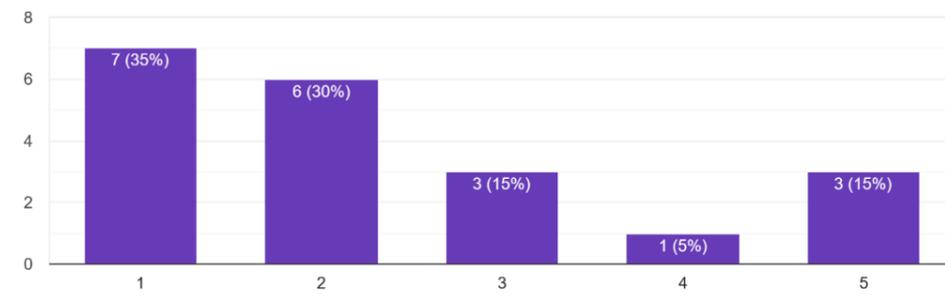
20 jawaban



Gambar 4. 17 Grafik kuesioner pernyataan 5

Saya merasa ada banyak hal yang tidak konsisten (tidak serasi) pada sistem ini.

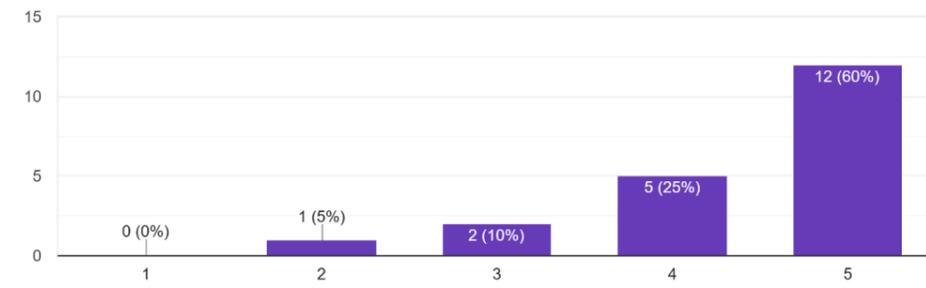
20 jawaban



Gambar 4. 18 Grafik kuesioner pernyataan 6

Saya merasa orang lain akan memahami cara menggunakan sistem ini dengan cepat.

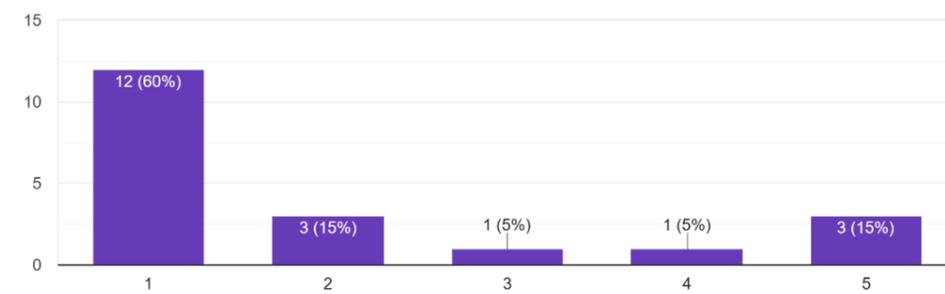
20 jawaban



Gambar 4. 19 Grafik kuesioner pernyataan 7

Saya merasa sistem ini membingungkan

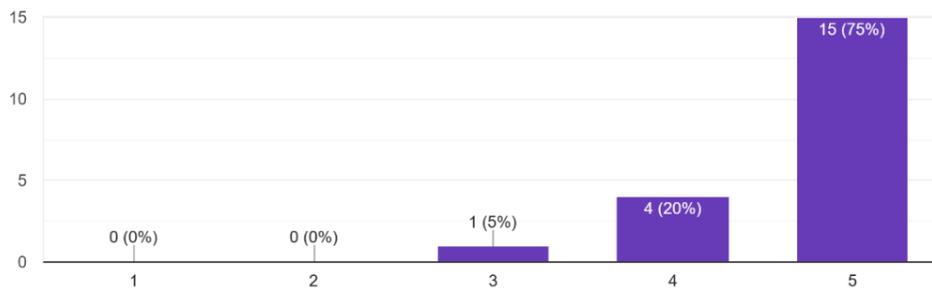
20 jawaban



Gambar 4. 20 Grafik kuesioner pernyataan 8

Saya merasa tidak ada hambatan dalam menggunakan sistem ini.

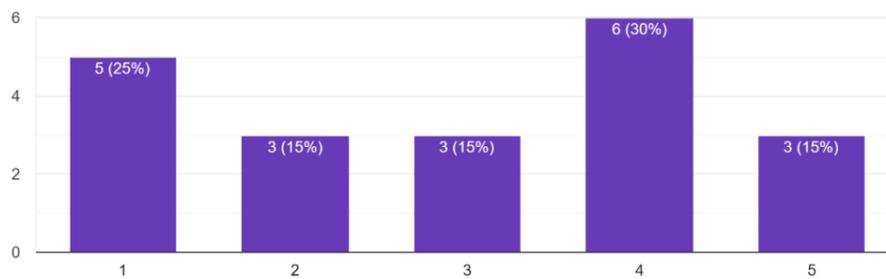
20 jawaban



Gambar 4. 21 Grafik kuesioner pernyataan 9

Saya perlu membiasakan diri terlebih dahulu sebelum menggunakan sistem ini.

20 jawaban



Gambar 4. 22 Grafik kuesioner pernyataan 10

Grafik 4.8 sampai 4.17 menunjukkan jawaban semua pengguna dari setiap pernyataan yang disediakan dalam kuesioner. Untuk lebih detailnya, hasil kuesioner dapat dilihat pada link <https://docs.google.com/spreadsheets/d/1lw5vq96DuJNv4ivY-yA6Pp7wzb-3tNFJ/edit?usp=sharing&oid=116685048014166667314&rtpof=true&sd=true>.

#### 4.3.1.3 Analisis hasil pengujian usabilitas

Pengolahan data hasil kuesioner diolah lagi menjadi *SUS* (*System Usability Scale*) score, dengan cara skor dikurang 1 untuk jawaban pernyataan positif dan 5 dikurangi skor untuk jawaban pernyataan negative. Lalu jumlahkan skor setiap pertanyaan, hasil maksimal dari tiap pengujian adalah 40. Setelah didapatkan hasil konversinya, langkah selanjutnya adalah mengubahnya menjadi skala 100 dengan mengalikannya dengan 2.5. berikut merupakan rumus perhitungan *SUS*.

A = Jawaban pernyataan ganjil

B = Jawaban pernyataan genap

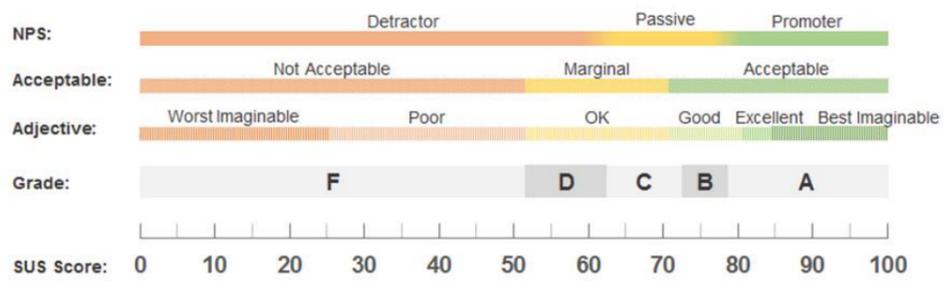
$$SUS\ Score = ((A-1)+(5-B)) \cdot 2.5$$

Tabel 4.1 merupakan tabel hasil konversi skor pengujian.

Tabel 4. 3 Konversi SUS score

Pengujian	Total Skor	SUS Score
1	36	90.0
2	30	75.0
3	31	77.5
4	28	70.0
5	16	40.0
6	40	100.0
7	40	100.0
8	27	67.5
9	21	52.5
10	30	75.0
11	23	57.5
12	39	97.5
13	20	50.0
14	23	57.5
15	37	92.5
16	37	92.5
17	36	90.0
18	39	97.5
19	40	100.0
20	23	57.5

Tabel 4.1 menunjukkan hasil konversi menjadi *SUS score*. Dapat dilihat dari tabel 4.1 dengan menjumlahkan seluruh *SUS score* dan membaginya dengan jumlah pengujian (20), didapatkan rata-rata nilai 77. Berdasarkan gambar 4.20 skor 77 mendapatkan kelas "B" dan *acceptable* atau dapat diterima



Gambar 4. 23 Index penilaian SUS

## BAB V PENUTUP

### 5.1 Kesimpulan

Talentern chatbot berhasil dikembangkan berbasis web menggunakan *Laravel* dan *PHP*. Menggunakan metode scrum, Talentern Chatbot berhasil dikembangkan dengan jadwal yang terstruktur dan rapi. Salah satu fungsi dari talentern chatbot adalah memberikan penilaian jawaban pengguna, karena itu output yang diberikan haruslah konsisten. Output penilaian yang diberikan oleh Talentern chatbot berhasil dibuat menjadi konsisten dengan menggunakan *word embedding* dan *cosine similarity*. Pada pengujian usabilitas, Talentern Chatbot mendapatkan *SUS score* 77, yang diartikan bahwa aplikasi ini termasuk *acceptable* oleh pengguna.

### 5.2 Saran

Berikut merupakan saran yang bisa diberikan setelah 9 bulan pengembangan Talentern chatbot.

1. Perubahan dataset

Pengumpulan dataset sebenarnya sudah dilakukan sebelumnya, dengan mengumpulkan 8800 sampel jawaban. Tetapi data ini terlalu banyak untuk dinilai dalam waktu beberapa bulan saja sehingga dataset menjadi kurang kredibel. Untuk saat ini pengelompokan jawaban masih berdasarkan kata kunci saja, kedepannya mungkin bisa digunakan dataset jawabanya tentunya setelah semua jawabanya dinilai oleh psikolog atau ahlinya

2. Penambahan webcam saat wawancara

Saat melakukan pengujian usabilitas ada beberapa penguji yang menyinggung dan membandingkan antara wawancara menggunakan chatbot dan secara langsung. Penguji menilai bahwa untuk menilai softskills, tidak cukup dilakukan secara verbal melainkan dengan melihat gestur dan sikap pengguna saat wawancara.

## DAFTAR PUSTAKA

[1]	M. kom Cahyana ST., “personalcommunicationPA,” 2024.
[2]	Z. Wu <i>et al.</i> , “Exploring the Trade-Offs: Unified Large Language Models vs Local Fine-Tuned Models for Highly-Specific Radiology NLI Task,” Apr. 2023, [Online]. Available: <a href="http://arxiv.org/abs/2304.09138">http://arxiv.org/abs/2304.09138</a>
[3]	Z. Wu <i>et al.</i> , “Exploring the Trade-Offs: Unified Large Language Models vs Local Fine-Tuned Models for Highly-Specific Radiology NLI Task,” Apr. 2023, [Online]. Available: <a href="http://arxiv.org/abs/2304.09138">http://arxiv.org/abs/2304.09138</a>
[4]	B. Wang, A. Wang, F. Chen, Y. Wang, and C. C. J. Kuo, “Evaluating word embedding models: Methods and experimental results,” 2019, <i>Cambridge University Press</i> . doi: 10.1017/ATSIP.2019.12
[5]	K. Park, J. S. Hong, and W. Kim, “A Methodology Combining <i>Cosine</i> Similarity with Classifier for Text Classification,” <i>Applied Artificial Intelligence</i> , vol. 34, no. 5, pp. 396–411, Apr. 2020, doi: 10.1080/08839514.2020.1723868.
[6]	P. S. Maratkar and P. Adkar, “ <i>ReactJS</i> -An Emerging Frontend Javascript Library,” 2021. [Online]. Available: <a href="https://nodejs.org/en/download/">https://nodejs.org/en/download/</a>
[7]	D. Bui and T. Mynttinen, “Degree title Bachelor of Engineering Thesis title Next.js for front-end and Compatible Backend Solutions Commissioned by XAMK Year 2023 Pages 46 pages,” 2023.
[8]	T. Nur and F. *1, “ELT FORUM 12(1) (2023) Journal of English Language Teaching Artificial intelligence (AI) technology in OpenAI ChatGPT application: A review of ChatGPT in writing English essay,” <i>Journal of English Language Teaching</i> , vol. 6, no. 1, 2023, [Online]. Available: <a href="http://journal.unnes.ac.id/sju/index.PHP/elt">http://journal.unnes.ac.id/sju/index.PHP/elt</a>
[9]	P. Limna, T. Kraiwanit, K. Jangjarat, P. Klayklung, and P. Chocksathaporn, “The use of ChatGPT in the digital era: Perspectives on chatbot implementation,” <i>Journal of Applied Learning and Teaching</i> , vol. 6, no. 1, pp. 64–74, Jan. 2023, doi: 10.37074/jalt.2023.6.1.32.
[10]	I. A. Alfarisi, A. T. Priandika, and A. S. Puspaningrum, “Penerapan <i>Framework Laravel</i> Pada Sistem Pelayanan Kesehatan (Studi Kasus: Klinik Berkah Medical Center),” <i>Jurnal Ilmiah Computer Science</i> , vol. 2, no. 1, pp. 1–9, Jul. 2023, doi: 10.58602/jics.v2i1.11.
[11]	O. Cárdenas, S. Falconí, E. Tusa, and A. Rodríguez, “Development of a ChatBot model for health telecare: Integration of LangChain, embeddings with

	OpenAI, and Pinecone using the question answering technique,” 2024. [Online]. Available: <a href="http://www.jart.icat.unam.mx">www.jart.icat.unam.mx</a>
[12]	K. Metze, R. C. Morandin-Reis, I. Lorand-Metze, and J. B. Florindo, “Bibliographic research with large language model ChatGPT-4: instability, hallucinations and sometimes alerts,” Jan. 01, 2024, <i>Universidade de Sao Paulo. Museu de Zoologia</i> . doi: 10.1016/j.clinsp.2024.100409.
[13]	im Rahmen des Studiums, P.-L. Glaser, and P.-L. Glaser Dominik Bork, “Developing Sproty-based Modeling Tools for VS Code Contribution of a Generic Development Approach and a Tool for ER Modeling Bachelor of Science Software-and Information Engineering.” [Online]. Available: <a href="http://www.tuwien.at">www.tuwien.at</a>
[14]	P. P. Kore, M. J. Lohar, M. T. Surve, and S. Jadhav, “API Testing Using Postman Tool,” <i>Int J Res Appl Sci Eng Technol</i> , vol. 10, no. 12, pp. 841–843, Dec. 2022, doi: 10.22214/ijraset.2022.48030.
[15]	Z. Sharfina and H. B. Santoso, “An Indonesian Adaptation of the System Usability Scale (SUS).”

