

PENGEMBANGAN *VIRTUAL ASSISTANT* MENGGUNAKAN *OPENAI ASSISTANT* UNTUK LAYANAN IMIGRASI

Skripsi

Oleh

SHALAHUDDIN ABDUL AZIZ

NPM 2117051083



FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM

UNIVERSITAS LAMPUNG

BANDAR LAMPUNG

2025

PENGEMBANGAN *VIRTUAL ASSISTANT* MENGGUNAKAN *OPENAI ASSISTANT* UNTUK LAYANAN IMIGRASI

Oleh

SHALAHUDDIN ABDUL AZIZ

Skripsi

**Sebagai Salah Satu Syarat untuk Mencapai Gelar
SARJANA ILMU KOMPUTER**

Pada

Jurusan Ilmu Komputer

Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung



**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG**

2025

ABSTRAK

PENGEMBANGAN *VIRTUAL ASSISTANT* MENGGUNAKAN *OPENAI ASSISTANT* UNTUK LAYANAN IMIGRASI

Oleh

SHALAHUDDIN ABDUL AZIZ

Perkembangan teknologi di sektor *Artificial Intelligence* (AI) dan *Machine Learning* (ML) menghasilkan teknologi seperti *Large Language Model* (LLM). LLM merupakan sebuah teknologi yang memungkinkan manusia untuk berbicara dengan komputer secara natural. Teknologi ini memiliki potensi untuk menggantikan *customer service* pada banyak sektor dengan menggunakan *Virtual Assistant* berbasis LLM. Penelitian ini bertujuan untuk melakukan pengembangan *virtual assistant* yang dapat menggantikan *customer service* dan bagian *Frequently Asked Question* Direktorat Jenderal Imigrasi. Penelitian ini dilaksanakan di Jurusan Ilmu Komputer, Universitas Lampung, pada bulan September 2024 – Mei 2025. Teknologi yang digunakan adalah *OpenAI Assistant* beserta LLM *GPT 4o*, *OpenAI Whisper*, *Next.js 14*, *MongoDB*, dan *Typescript*. *Knowledge base virtual assistant* menggunakan Undang – Undang No. 6 Tahun 2011 sebagai dasar pengetahuan *virtual assistant* agar jawaban yang diberikan sesuai dengan peraturan dan Undang – Undang yang berlaku di Indonesia. Metodologi yang diterapkan dalam penelitian ini adalah Extreme Programming (XP), melalui tahapan perencanaan, desain, pengkodean, pengujian, dan pendengaran (listening).

Kata kunci: *Virtual Assistant*, *OpenAI Assistant*, *Customer Service*, Direktorat Jenderal Imigrasi, Extreme Programming.

ABSTRACT

DEVELOPMENT OF VIRTUAL ASSISTANT USING OPENAI ASSISTANT FOR IMIGRATION SERVICES

By

SHALAHUDDIN ABDUL AZIZ

The development of technology in the fields of Artificial Intelligence (AI) and Machine Learning (ML) has resulted in technologies such as Large Language Models (LLM). LLM is a technology that enables humans to communicate with computers in a natural manner. This technology has the potential to replace customer service in many sectors by utilizing Virtual Assistants based on LLM. This research aims to develop a virtual assistant that can replace customer service and address the Frequently Asked Questions of the Directorate General of Immigration. The study will be conducted at the Department of Computer Science, University of Lampung, from September 2024 to May 2025. The technologies used include OpenAI Assistant along with LLM GPT-4o, OpenAI Whisper, Next.js 14, MongoDB, and Typescript. The knowledge base for the virtual assistant will utilize Constitution No. 6 of 2011 as the foundational knowledge to ensure that the answers provided are in accordance with applicable regulations and laws in Indonesia. The methodology applied in this research is Extreme Programming (XP), which involves phases of planning, design, coding, testing, and listening.

Keywords: Virtual Assistant, OpenAI Assistant, Customer Service, Directorate General of Immigration, Extreme Programming.

Judul Skripsi : PENGEMBANGAN *VIRTUAL ASSISTANT*
MENGUNAKAN *OPENAI ASSISTANT*
UNTUK LAYANAN IMIGRASI

Nama Mahasiswa : *Shalahuddin Abdul Aziz*

Nomor Pokok Mahasiswa : 2117051083

Program Studi : S1 Ilmu Komputer

Jurusan : Ilmu Komputer

Fakultas : Matematika dan Ilmu Pengetahuan Alam



1. Komisi Pembimbing

Rico Andrian, S.Si., M.Kom

NIP. 19750627 200501 1 001

Muhagiqin, S.Kom., M.T.I.

NIP. 19930525 202203 1 009

2. Ketua Jurusan Ilmu Komputer

Dwi Sakethi, S.Si., M.Kom

NIP. 19680611 199802 1 001

MENGESAHKAN

1. Tim Penguji

Ketua : Rico Andrian, S.Si., M.Kom

Sekretaris : Muhaqiqin, S.Kom., M.T.I.

Penguji Utama : Prof. Admi Syarif, Ph.D

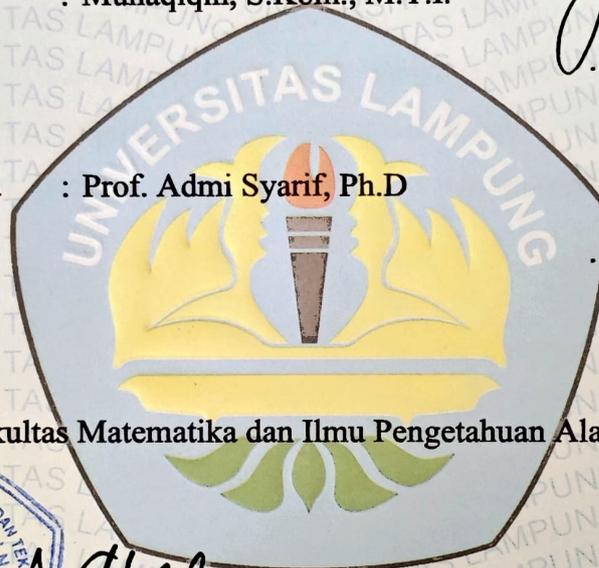
2. Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam



Dr. Eng. Heri Satria, S.Si., M.Si.

NIP 19711001 200501 1 002

Tanggal Lulus Ujian Skripsi: 3 Juni 2025



(Handwritten signatures of Rico Andrian, Muhaqiqin, and Prof. Admi Syarif)

PERNYATAAN

Saya yang bertanda tangan di bawah ini:

Nama : Shalahuddin Abdul Aziz

NPM : 2117051083

Menyatakan bahwa skripsi saya yang berjudul **“Pengembangan Virtual Assistant Menggunakan OpenAI Assistant Untuk Layanan Imigrasi”** merupakan karya saya sendiri dan bukan karya orang lain. Semua tulisan yang tertuang di skripsi ini telah mengikuti kaidah penulisan karya ilmiah Universitas Lampung. Apabila di kemudian hari terbukti skripsi saya merupakan hasil penjiplakan atau pembuatan orang lain, maka saya bersedia menerima sanksi berupa pencabutan gelar yang telah saya terima.

Bandar Lampung, 12 Juni 2025

Penulis,



Shalahuddin Abdul Aziz

NPM. 2117051083

RIWAYAT HIDUP



Penulis dilahirkan di Jakarta pada tanggal 15 Mei 2003 sebagai anak pertama dari dua bersaudara, dari Bapak Septiawan dan Ibu Melly Hayati. Pendidikan yang telah ditempuh oleh penulis diantaranya, menyelesaikan pendidikan dasar di SDIT Amalia Cibinong pada tahun 2015. Penulis menyelesaikan pendidikan menengah pertama di SMP Al-Azhar Syifa Budi Cibinong pada tahun 2018. Penulis melanjutkan pendidikan menengah atas di SMA Negeri 2 Cibinong yang selesai pada tahun 2021. Perjalanan pendidikan penulis dilanjutkan dengan mengikuti program sarjana Ilmu Komputer di Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung melalui jalur Seleksi Bersama Masuk Perguruan Tinggi Negeri (SBMPTN) pada tahun 2021.

Selama menjadi mahasiswa penulis aktif mengikuti beberapa kegiatan antara lain:

1. Asisten praktikum mata kuliah Rekayasa Perangkat Lunak dan Pemrograman Berorientasi Objek
2. Anggota Himpunan Mahasiswa Ilmu Komputer Periode 2023
3. Pada tahun 2023 mengikuti program *Bangkit by Google* sebagai *Cloud Computing Cohort* dan mendapatkan sertifikasi *Associate Cloud Engineer*
4. Pada tahun 2024 mengikuti kegiatan magang di PT. Merkle Inovasi Teknologi sebagai *Frontend Engineer*

PERSEMBAHAN

Kupersembahkan karya ini kepada:

Kepada Ayahanda dan Ibunda Tersayang

Yang senantiasa memberikan yang terbaik dan melantunkan do'a yang selalu menyertaiku. Kuucapkan pula terimakasih sebesar-besarnya karena telah mendidik dan membesarkanku dengan cara yang dipenuhi kasih sayang, dukungan, dan pengorbanan yang tak terbalaskan.

SANWACANA

Puji syukur atas segala rahmat Allah SWT yang diberikan kepada penulis sehingga dapat menyelesaikan skripsi berjudul “PENGEMBANGAN *VIRTUAL ASSISTANT* MENGGUNAKAN *OPENAI ASSISTANT* UNTUK LAYANAN IMIGRASI”. Tidak lupa shalawat serta salam senantiasa dicurahkan kepada Nabi Muhammad SAW, suri tauladan yang telah menunjukkan jalan yang benar kepada seluruh umatnya.

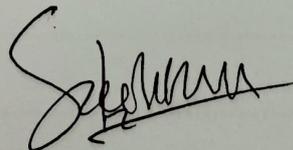
Pada kesempatan ini penulis mengucapkan terima kasih kepada semua pihak yang telah membantu dalam penyusunan skripsi ini, yaitu:

1. Allah SWT. Yang telah memberikan nikmat, rahmat, dan hidayah-Nya sehingga penulis dapat menyelesaikan skripsi ini dengan lancar hingga akhir
2. Kedua orang tua, Bapak Septiawan dan Ibu Melly Hayati yang senantiasa menyemangati, mendukung, mendoakan, dan membantu segala hal yang tidak terhitung nilainya
3. Bapak Dr. Eng. Heri Satria, S.Si., M.Si. selaku Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam
4. Bapak Dwi Sakethi S.Si., M.Kom. selaku Ketua Jurusan Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam sekaligus Dosen Pembimbing Akademik penulis
5. Ibu Yunda Heningtyas, M.Kom selaku Sekretaris Jurusan Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam
6. Bapak Tristiyanto, S.Kom, M.I.S. Ph.D selaku Ketua Prodi S1 Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam
7. Bapak Rico Andrian, S.Si., M.Kom selaku dosen pembimbing atas kesediaannya untuk memberikan bimbingan, arahan, kritik, serta sarannya selama proses penulisan skripsi

8. Bapak Muhaqiqin, S.Kom., M.T.I. selaku dosen pembimbing atas kesediaannya untuk memberikan bimbingan, arahan, kritik, serta sarannya selama proses penulisan skripsi
 9. Bapak Prof. Admi Syarif, Ph.D selaku dosen pembahas yang telah memberikan kritik, saran, serta masukan kepada penulis
 10. Bapak dan Ibu Dosen Jurusan Ilmu Komputer Universitas Lampung yang telah memberikan ilmu, pengetahuan, serta bimbingannya
 11. Ibu Ade Nora Maela selaku staf administrasi Jurusan Ilmu Komputer Universitas Lampung yang telah banyak membantu proses administrasi dalam proses penulisan skripsi
 12. Seluruh staf Jurusan Ilmu Komputer yang selalu membantu penulis mulai dari awal hingga akhir masa kuliah
 13. M. Bagas Shaka, M. Satria Surya, Rafi Ramadhan, Jhon Wilken Chrsitoper N., M. Iqbal Al-Hafidzu, Keyvin Jourdan, Nauval Hanif, Al Qolby Arief E., M. Rafi Satria, Reza Nur Ramadhan, Stefanus Ferry, S. A. Syailendra Wangsa, Zidan Pasya, M. Farrel Elmaretza, dan M. Abdi Firdaus sebagai rekan penelitian dan teman bertukar pikiran
 14. Ramadhan Ihsani Yulfa, Aryoka Natakusuma, Arrafah Raditya Rangga Putra, Setyo Galih Nugraha sebagai rekan kerja dan repo
 15. Semua pihak yang turut membantu penulis dalam penyusunan laporan skripsi
- Penulis menyadari bahwa masih banyak kekurangan dalam penulisan skripsi ini, namun semoga skripsi ini dapat membawa manfaat bagi perkembangan ilmu pengetahuan terutama bagi semua civitas Ilmu Komputer Universitas Lampung.

Bandar Lampung, 12 Juni 2025

Penulis,



Shalahuddin Abdul Aziz

NPM. 2117051083

DAFTAR ISI

	Halaman
DAFTAR ISI.....	x
DAFTAR GAMBAR	xiii
DAFTAR TABEL.....	xv
I. PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	6
1.3 Batasan Masalah	7
1.4 Tujuan.....	7
1.5 Manfaat Penelitian.....	7
II. TINJAUAN PUSTAKA.....	8
2.1 <i>Imigrasi</i>	8
2.2 <i>Virtual Assistant</i>	8
2.2.1 <i>OpenAI Assistant</i>	9
2.2.2 <i>Retrieval Augmented Generation (RAG)</i>	10
2.2.3 <i>GPT-4</i>	11
2.2.4 <i>OpenAI Whisper</i>	11
2.3 Aplikasi Web.....	11
2.3.1 <i>Next.js</i>	12

2.3.2 <i>NextAuth.js</i>	12
2.4 <i>MongoDB</i>	13
2.5 <i>Typescript</i>	13
2.6 <i>Extreme Programming</i>	14
III. METODE PENELITIAN.....	16
3.1. Waktu dan Tempat Penelitian	16
3.1.1. Waktu Penelitian	16
3.1.2. Tempat Penelitian.....	17
3.2. Bahan dan Alat Penelitian.....	17
3.2.1. Perangkat Keras	17
3.2.2. Perangkat Lunak.....	17
3.3. Metode Penelitian.....	18
3.3.1. <i>Planning</i> (Perencanaan)	18
3.3.2. <i>Design</i> (Perancangan)	21
3.3.3. <i>Coding</i> (Pengkodean).....	29
3.3.4. <i>Testing</i> (Pengujian)	31
3.3.5. <i>Listening</i> (Pendengaran).....	31
IV. HASIL DAN PEMBAHASAN.....	33
4.1 Hasil	33
4.1.1 Iterasi Pertama.....	33
4.1.2 Iterasi Kedua	43
4.1.3 Iterasi Ketiga	53
4.1.4 Iterasi Keempat	61

4.1.5 Iterasi Kelima.....	67
4.2 Pembahasan.....	73
V. SIMPULAN DAN SARAN	76
5.1. Simpulan	76
5.2. Saran.....	76
DAFTAR PUSTAKA	77

DAFTAR GAMBAR

	Halaman
Gambar 1. Alur Pengembangan <i>Extreme Programming</i>	14
Gambar 2. Tahapan Penelitian	18
Gambar 3. <i>Use Case Diagram</i>	18
Gambar 4. <i>Entity Relationship Diagram</i>	21
Gambar 5. <i>Register Activity Diagram</i>	25
Gambar 6. <i>Login Activity Diagram</i>	26
Gambar 7. <i>Chat Diagram</i>	27
Gambar 8. <i>Prototype Register Page</i>	28
Gambar 9. <i>Prototype Login Page</i>	28
Gambar 10. <i>Prototype Chat Page</i>	29
Gambar 11. <i>Architecture Diagram</i>	29
Gambar 12. <i>Entity Relationship Diagram</i>	34
Gambar 13. Prototype Halaman Register dan Login.....	34
Gambar 14. Instalasi <i>MongoDB SDK</i> dan <i>NextAuth.js</i>	35
Gambar 15. Skema Tabel <i>User</i> di <i>MongoDB</i>	35
Gambar 16. Instalasi <i>Library UI</i> dan Validasi Formulir	36
Gambar 17. Halaman Autentikasi	37
Gambar 18. Komponen <i>Chat</i>	37

Gambar 19. Integrasi <i>OpenAI Assistant</i>	38
Gambar 20. <i>Environment Variables</i>	38
Gambar 21. Percakapan dengan <i>Virtual Assistant</i>	40
Gambar 22. <i>System Prompt</i> untuk Pertanyaan	44
Gambar 23. <i>Instructions, Model, dan Tools Virtual Assistant</i>	46
Gambar 24. Tabel <i>Thread</i> dan <i>Message</i>	54
Gambar 25. Skema <i>MongoDB</i> untuk Tabel <i>Message</i>	55
Gambar 26. Skema <i>MongoDB</i> untuk Tabel <i>Thread</i>	56
Gambar 27. Potongan Kode untuk Menyimpan Pesan ke <i>Database</i>	57
Gambar 28. Fungsi <i>saveMessageToThread</i> untuk Menyimpan Pesan ke <i>Database</i>	58
Gambar 29. Desain Tombol <i>Speech-to-Text</i>	62
Gambar 30. Listening, Transcript dan Error State	62
Gambar 31. Fungsi <i>toggleListening</i> dan <i>handleSubmit</i>	63
Gambar 32. <i>useEffect Hook</i> Fitur <i>Speech-To-Text</i>	64
Gambar 33. Pengujian <i>Speech-to-Text</i> untuk Memberikan Pertanyaan.....	66
Gambar 34. Isu Tampilan di Layar Kecil.....	68
Gambar 35. <i>Prototype Sidebar</i> untuk Perangkat Layar Kecil.....	69
Gambar 36. Komponen <i>Header</i>	70
Gambar 37. Tampilan Komponen <i>Header</i>	70
Gambar 38. Diagram Hasil Survey Waktu Respon <i>Virtual Assistant</i>	73
Gambar 39. Diagram Hasil Survey Waktu Mendapatkan Jawaban dari <i>FAQ</i>	74

DAFTAR TABEL

	Halaman
Tabel 1. Jadwal Penelitian.....	16
Tabel 2. <i>User Story</i>	19
Tabel 3. <i>Acceptance Test Criteria</i>	20
Tabel 4. <i>Chat CRC</i>	22
Tabel 5. <i>OpenAI CRC</i>	22
Tabel 6. <i>Chat History CRC</i>	23
Tabel 7. <i>User CRC</i>	23
Tabel 8. <i>Authentication CRC</i>	23
Tabel 9. Tugas Iterasi Pertama	33
Tabel 10. Tabel Pertanyaan <i>Testing</i> Iterasi Pertama	39
Tabel 11. Tabel Kriteria <i>Testing</i> Iterasi Pertama.....	41
Tabel 12. Tabel Hasil Survey <i>End-User</i>	41
Tabel 13. Tabel Hasil <i>Listening</i> Iterasi Pertama	42
Tabel 14. Tugas Iterasi Kedua.....	43
Tabel 15. Pertanyaan Pengujian Iterasi Kedua.....	47
Tabel 16. Tabel Kriteria <i>Testing</i> Iterasi Kedua	51
Tabel 17. Tabel Hasil Survey <i>End-User</i> Iterasi Kedua	51
Tabel 18. Tabel Kriteria Tahap <i>Listening</i> Iterasi Kedua	52

Tabel 19. Tugas Iterasi Ketiga.....	53
Tabel 20. Tabel Kriteria <i>Testing</i> Iterasi Ketiga	59
Tabel 21. Tabel Hasil Survey <i>End-User</i> Iterasi Ketiga.....	59
Tabel 22. Tabel Kriteria Tahap <i>Listening</i> Iterasi Kedua	60
Tabel 23. Tugas Iterasi Keempat.....	61
Tabel 24. Tabel <i>Testing</i> Iterasi Keempat.....	65
Tabel 25. Tabel Hasil Survey <i>End-User</i> Iterasi Keempat	66
Tabel 26. Tabel <i>Listening</i> Iterasi Keempat.....	67
Tabel 27. Tugas Iterasi Keempat.....	67
Tabel 28. Tabel <i>Testing</i> Iterasi Kelima	71
Tabel 29. Tabel Hasil Survey <i>End-User</i> Iterasi Keempat	71
Tabel 30. Tabel <i>Listening</i> Iterasi Kelima	72
Tabel 31. Tabel Perbandingan Kecepatan Mendapatkan Jawaban	73
Tabel 32. Perbandingan <i>Virtual Assistant</i>	74

I. PENDAHULUAN

1.1 Latar Belakang

Interaksi manusia dengan teknologi semakin menjadi bagian integral dari kehidupan sehari-hari di era digital ini. Manifestasi dari kemajuan teknologi ini adalah *Virtual Assistant* (VA), yang dirancang untuk mensimulasikan interaksi manusia dalam memberikan layanan informasi dan bantuan kepada pengguna. *Virtual assistant* menggunakan *artificial intelligence* (AI) dan pembelajaran mesin (ML), untuk berinteraksi dengan pengguna dan menyediakan layanan informasi atau bantuan pelanggan. Teknologi yang sedang mengalami banyak terobosan saat ini adalah *Large Language Model* (LLM) seperti *Generative Pre-trained Transformer* (GPT) dan *embedding* dapat digunakan untuk memahami kalimat manusia. Teknologi LLM yang digunakan untuk *virtual assistant* merupakan solusi yang dapat dikembangkan untuk mengatasi masalah *customer service* (Audi Alboush, 2023). Cinnox sebagai contoh merupakan sebuah sistem yang menawarkan layanan *contact center omnichannel* yang ditenagai AI untuk mempersingkat alur *customer support* dan memberikan pengalaman yang lebih baik bagi pengguna.

Direktorat Jenderal Imigrasi (Ditjenim) adalah bagian dari Kementerian Imigrasi dan Pemasarakatan Republik Indonesia yang memiliki tugas utama merumuskan dan melaksanakan kebijakan dan standardisasi teknis di bidang Imigrasi Indonesia. Pelayanan perihal dokumen perjalanan seperti visa dan izin tinggal merupakan beberapa dokumen yang pengurusannya menjadi tanggung jawab Ditjenim. Masyarakat akan berhadapan dan berinteraksi langsung dengan petugas Ditjenim di kantor Imigrasi setempat atau mencari informasi melalui web Imigrasi mengenai alur dan syarat-syarat pengajuan dokumen.

Pentingnya *virtual assistant* dalam konteks pelayanan publik didukung oleh teori dan bukti empiris yang menunjukkan bahwa layanan otomatis dapat meningkatkan efisiensi dan konsistensi. Xu *et al.* (2020) mengamati bahwa salah satu masalah utama yang dihadapi dalam pelayanan pelanggan adalah waktu tunggu dan tingkat responsif. Pelanggan sering kali tidak menyadari waktu mereka yang terbuang saat menunggu untuk dilayani, dan mereka sangat sensitif jika harus menunggu secara *online*. Waktu tunggu *online* yang lama berkontribusi pada pengalaman layanan yang negatif (Xu *et al.* 2020). Penggunaan *virtual assistant* yang dilengkapi dengan teknologi LLM dalam skenario pelayanan dapat secara signifikan mengurangi waktu respons dengan menangani pertanyaan rutin secara otomatis, sehingga meningkatkan efisiensi dan kualitas layanan.

Layanan imigrasi Indonesia memiliki masalah dimana bagian *Frequently Asked Question* kurang lengkap untuk memecahkan masalah pengguna. Imigrasi Indonesia sangat diuntungkan dengan menerapkan *virtual assistant* karena keterbatasan sistem mereka saat ini. Bagian *FAQ* di situs web mereka sering kali tidak memadai, karena tidak selalu memberikan jawaban yang komprehensif atau mudah dinavigasi, sehingga membuat pengguna frustrasi saat mencari informasi tertentu. Mengandalkan agen manusia juga membatasi ketersediaan dukungan, karena mereka hanya dapat beroperasi selama jam kerja dan tidak dapat menangani volume pertanyaan berulang yang tinggi secara efisien. *Virtual assistant* dapat memberikan dukungan dua puluh empat jam, memastikan bahwa pengguna memiliki akses ke informasi yang akurat dan konsisten setiap saat. Hal ini akan secara signifikan mengurangi waktu tunggu dan meningkatkan pengalaman pengguna secara keseluruhan. Penggunaan *virtual assistant* dapat menawarkan dukungan multibahasa, membuat layanan lebih mudah diakses oleh beragam populasi penduduk lokal dan asing. *Virtual assistant* juga dapat diskalakan sehingga Ditjenim dapat menangani permintaan yang meningkat tanpa memerlukan staf tambahan, yang pada akhirnya membuat operasi lebih efisien dan ramah pengguna.

Kemajuan AI di bidang NLP dan LLM, menjadi solusi untuk menggantikan *customer service* yang tradisionalnya menggunakan manusia, sebagaimana disimpulkan dari akar masalah ini. *Chatbot* atau *virtual assistant* yang dilengkapi dengan *knowledge base* yang terstruktur dapat dimanfaatkan dan dilatih untuk melakukan tugas-tugas tertentu dengan baik, mirip dengan bagaimana pekerja manusia dilatih untuk melakukan tugas mereka. *Virtual assistant* yang dilatih dengan pengetahuan spesifik dapat memecahkan masalah dengan lebih cepat dan efektif dibandingkan dengan manusia yang dapat mengalami kelelahan atau melakukan kesalahan. Solusi untuk menggantikan *customer service* manusia memanfaatkan kemajuan teknologi pada bidang kecerdasan buatan (Shankar 2018).

Teknologi LLM yang ditawarkan secara publik dan dapat dilatih untuk diberikan *knowledge base* ada *OpenAI Assistant*, *Amazon Q*, dan *Llama 3.1*. *OpenAI Assistant* adalah solusi yang terjangkau dan juga memiliki fitur seperti *text-to-speech* dan *speech-to-text* yang memungkinkan pengguna untuk berkomunikasi layaknya dengan manusia. *Virtual assistant* yang ditenagai *OpenAI Assistant* menjadi pilihan yang tepat untuk diterapkan dalam konteks pelayanan publik.

Penelitian-penelitian sebelumnya telah mengeksplorasi penerapan *virtual assistant* dalam sektor kesehatan. Suárez *et al.* (2024) melakukan penelitian penerapan *ChatGPT-4* untuk membantu dokter gigi. Target utamanya adalah mengevaluasi akurasi dan keandalan respon *ChatGPT-4* terkait pertanyaan mengenai gigi dan potensinya sebagai *virtual assistant* bagi dokter gigi dalam memutuskan diagnosa. Limitasi tetap ditemukan karena data yang digunakan untuk melatih *ChatGPT-4* dan resiko informasi yang tidak tepat. Respon yang diberikan *ChatGPT-4* dievaluasi berdasarkan pedoman *Spanish Society of Oral Surgery*, dan walau evaluasi menunjukkan akurasi keseluruhan diangka 71.7%, ketidakkonsistenan dalam beberapa jawaban menunjukkan perlunya supervisi oleh manusia. *ChatGPT-4* memiliki potensi yang menjanjikan untuk menjadi alat yang dapat membantu

dokter gigi dengan supervisi yang ketat beserta validasi terhadap respon yang dihasilkan oleh ahli-ahli, terutama pada skenario dimana keselamatan pasien dipertaruhkan.

Penelitian mengenai interaksi kognitif yang dilakukan oleh Bernard & Arnold (2019) membahas dinamika interaksi manusia dan komputer/mesin. *Virtual assistant* berbeda dengan model interaksi komputer dengan manusia tradisional, karena pendekatan yang disebut interaksi kognitif. Interaksi kognitif adalah saat *virtual assistant* meniru percakapan manusia dengan memahami dan adaptasi dengan niat, tujuan, dan asumsi pengguna. Interaksi kognitif lebih dari sekadar mekanisme perintah dan respons. Interaksi kognitif mencakup penggunaan tindakan, di mana perintah tidak hanya dipandang sebagai informasi, tetapi juga sebagai tindakan kolaboratif yang mengubah konteks kognitif dan operasional. Dasar interaksi kognitif diangkat dari ilmu-ilmu manusia, khususnya filsafat pikiran dan bahasa. Gagasan-gagasan utama seperti “kesengajaan” dan “niat bersama” menjelaskan bagaimana tindakan manusia didorong oleh kondisi mental seperti keyakinan, keinginan, dan niat. Sistem interaksi kognitif harus menggabungkan elemen-elemen ini untuk memungkinkan aktivitas bersama yang mulus antara pengguna dan *virtual assistant*. Penerapan di bidang dirgantara telah menunjukkan potensi interaksi kognitif, misalnya, *virtual assistant* yang membantu dalam pemeliharaan pesawat terbang dapat melacak niat mekanik, memberikan rekomendasi yang sesuai dengan konteks, dan mempertahankan dialog yang koheren meskipun perintah dari mekanik kurang lengkap atau ambigu. Tingkat interaksi ini menunjukkan bagaimana *virtual assistant* mengubah operasi dengan memungkinkan komunikasi yang alami dan sesuai dengan konteks.

Penelitian mengenai *chatbot customer service* yang dilakukan oleh Chang *et al.* (2022) membahas penerapan *Language Understanding Intelligence Service (LUIS)* milik *Microsoft* dan *QnA Maker*. *Knowledge base* digunakan sebagai sumber pengetahuan untuk memberikan respon yang akurat. LLM digunakan untuk

memberikan respon yang manusiawi dan tidak kaku dengan tujuan memberikan rasa familiaritas bagi pengguna. *Chatbot* ini dirancang untuk merespons pertanyaan dengan intervensi manusia yang minimal, sehingga mampu memberikan bantuan dua puluh empat jam dan mengurangi biaya operasional. Implementasi pada platform *LINE* memungkinkan akses yang luas, sementara mekanisme pembelajaran aktif memungkinkan chatbot berkembang dari waktu ke waktu.

Penelitian yang dilakukan oleh Trivedi (2019) mengklaim bahwa kesuksesan *chatbot* dalam konteks *customer service* pada umumnya berkaitan erat dengan kualitas informasi, kualitas sistem, dan kualitas layanan. Zhao *et al.* (2021) menjelaskan bahwa persepsi pelanggan terhadap kualitas layanan mencakup kenyamanan, mekanisme respon, dan kepercayaan. Kenyamanan layanan mengacu pada bagaimana pengalaman pengguna terhadap prosedur di balik pemberian jawaban dan penyelesaian layanan. Proses layanan yang baik secara kenyamanan harus mudah diakses, memungkinkan pelanggan untuk mendapatkan solusi tepat dan tepat waktu. Mekanisme respon mengacu pada bagaimana pengguna dapat bertanya dengan timbal balik yang alami. Kepercayaan mengacu pada tingkat kepercayaan pelanggan terhadap sistem yang memberikan jawaban. Persepsi bahwa jawaban dari manusia lebih baik daripada sebuah mesin harus bisa dipenuhi dengan pengumpulan data dan respon yang alami.

Metode pengembangan *Extreme Programming (XP)* menawarkan alternatif yang ringan dan fleksibel terhadap metodologi pengembangan perangkat lunak tradisional, dengan menekankan nilai-nilai seperti kesederhanaan, komunikasi, dan umpan balik. *XP* memprioritaskan praktik seperti komunikasi lisan, pemrograman berpasangan, dan pengujian berkelanjutan, mendorong proses berulang di mana desain dan pengembangan terjadi secara bersamaan (Juric, 2000). Siklus rilisnya yang pendek memungkinkan umpan balik yang cepat dan kemampuan beradaptasi terhadap perubahan kebutuhan, sehingga cocok untuk proyek-proyek yang lebih kecil. Penekanan *XP* pada desain di awal dan fokusnya pada pengkodean awal dapat

menimbulkan tantangan bagi sistem yang lebih besar dan lebih kompleks, terutama yang berkaitan dengan stabilitas arsitektur jangka panjang.

Kualitas dan kepuasan layanan publik di Indonesia akan meningkat tinggi jika tantangan dalam menerapkan *virtual assistant* untuk layanan publik dapat dikalahkan, meskipun terdapat banyaknya tantangan. Penggunaan AI di sektor publik merupakan sebuah potensi, lebih spesifiknya pada peningkatan kualitas layanan. AI yang memiliki tingkat kepintaran setara dengan manusia akan memfasilitasi dan membantu petugas dalam memberikan layanan publik dan membantu pemerintah mengatur dan memenuhi kebutuhan penduduk dengan mengintegrasikan dunia nyata dan digital (Staiculescu, 2023).

Penelitian ini berfokus pada pengembangan sistem untuk mengembangkan produk *virtual assistant* yang dapat digunakan oleh pengguna layanan imigrasi untuk meningkatkan kepuasan pengguna dalam mencari informasi. Lebih spesifiknya meningkatkan kenyamanan, mekanisme respon, dan kepercayaan (Zhao *et al.*, 2021). Pengguna dapat menggunakan *virtual assistant* untuk mengakses informasi dan bantuan dengan lebih cepat dan mudah, tanpa harus mengantri atau menunggu lama. *Virtual assistant* dan diberikan pengetahuan untuk memberikan informasi dan bantuan yang akurat dan konsisten kepada semua pengguna, tanpa dipengaruhi oleh faktor-faktor seperti emosi atau kelelahan manusia. Faktor kelelahan pada pegawai dapat memengaruhi respon yang diberikan dalam *customer service* konvensional. *Virtual assistant* tidak akan mengalami hal tersebut dan akan meningkatkan kepuasan pengguna dan membangun kepuasan terhadap pelayanan publik.

1.2 Rumusan Masalah

Rumusan masalah dari penelitian ini adalah bagaimana VA berbasis AI yang dikembangkan dapat meningkatkan kualitas layanan *customer service* imigrasi.

1.3 Batasan Masalah

Batasan masalah untuk penelitian ini adalah:

1. Pengguna dibatasi sebagai pengguna layanan imigrasi Indonesia.
2. VA akan menggunakan platform web.
3. LLM yang akan digunakan adalah layanan dari *OpenAI*, tepatnya menggunakan *OpenAI Assistant*.
4. Pengembangan dilakukan menggunakan metodologi *Extreme Programming*.
5. Database yang digunakan adalah *MongoDB*.
6. *Framework* web yang digunakan adalah *Next.js 14*
7. Bahasa pemrograman yang akan digunakan adalah *Typescript*.
8. *Deployed environment* akan menggunakan *Vercel*.

1.4 Tujuan

Tujuan dari penelitian ini adalah mengembangkan sistem *virtual assistant* berbasis web menggunakan *OpenAI Assistant* untuk layanan imigrasi.

1.5 Manfaat Penelitian

Penelitian ini memiliki manfaat untuk meningkatkan kualitas layanan publik imigrasi Indonesia dengan memanfaatkan teknologi AI. *Virtual assistant* berbasis web yang dapat diakses oleh pengguna layanan imigrasi untuk mendapatkan bantuan dan panduan yang diperlukan. Peningkatan kecepatan bantuan yang dapat diberikan juga menjadi manfaat penting yang dihasilkan.

II. TINJAUAN PUSTAKA

2.1 *Imigrasi*

Direktorat Jenderal Imigrasi yang berada di bawah naungan Kementerian Imigrasi memiliki mandat untuk merumuskan dan mengimplementasikan kebijakan serta standar teknis dalam bidang keimigrasian. Lingkup tugasnya mencakup pengelolaan dokumen perjalanan, seperti visa dan izin tinggal, yang menjadi syarat bagi warga negara asing untuk memasuki dan berada di wilayah Indonesia. Inovasi teknologi seperti aplikasi M-Paspor bertujuan untuk menyederhanakan proses-proses ini dengan membuat aplikasi paspor lebih mudah diakses dan efisien. Riset yang dilakukan oleh Putu Denny Artina Mahendra *et al.* (2024) menyimpulkan bahwa keberadaan aplikasi M-Paspor membantu meningkatkan efisiensi pengajuan paspor dan layanan imigrasi lainnya. Masalah baru yang dihadapi adalah kurangnya pemahaman masyarakat terhadap prosedur penggunaan aplikasi M-Paspor dan prosedur pengajuan secara manual.

2.2 *Virtual Assistant*

Virtual assistant adalah sebuah sistem komputer yang mensimulasikan percakapan dan interaksi manusia. *Virtual assistant* dapat digunakan untuk menjawab pertanyaan, menjalankan perintah, dan menyajikan informasi untuk pengguna. *Virtual assistant* pada umumnya memanfaatkan AI, NLP, dan LLM untuk memahami konteks bahasa manusia dan memungkinkannya untuk memberi jawaban yang manusiawi. Teknologi ini menggunakan algoritma yang mampu memahami dan menanggapi perintah atau pertanyaan dari pengguna, memungkinkan interaksi yang lebih alami dan efisien (Ischen *et al.*, 2022).

Sistem *virtual assistant* seperti yang digunakan untuk membantu teknisi dirgantara dalam pemeliharaan pesawat yang diteliti oleh Bernard & Arnold (2019) memanfaatkan NLP dan LLM untuk memahami bahasa manusia dan memberikan jawaban yang manusiawi. Penelitian tentang evaluasi *ChatGPT-4* sebagai *virtual assistant* untuk dokter gigi juga dilakukan oleh Suárez *et al.* (2024) membahas peran yang dapat dimainkan oleh *ChatGPT-4* sebagai *virtual assistant* bagi dokter gigi. Tujuan utamanya adalah untuk mengevaluasi keakuratan dan keandalan respons *ChatGPT-4* terhadap pertanyaan terkait gigi, yang membahas potensinya sebagai *virtual assistant* dalam pengambilan keputusan klinis.

2.2.1 OpenAI Assistant

OpenAI adalah organisasi penelitian AI yang bertujuan untuk memastikan bahwa *Artificial General Intelligence (AGI)* bermanfaat bagi seluruh umat manusia. *OpenAI* didirikan pada bulan Desember 2015 melakukan penelitian di berbagai bidang AI, termasuk pembelajaran mesin, robotika, dan NLP (Binns, 2017). *OpenAI Assistant* memiliki kemampuan untuk melakukan *file search* yang pada dasarnya adalah *Retrieval Augmented Generation (RAG)* (Moussiades & Zografos, 2023). Fitur ini memungkinkan *virtual assistant* untuk mencari dan mendapatkan akses ke dokumen, artikel, atau berkas yang relevan dengan pertanyaan atau permintaan yang diberikan. Pengguna dapat mengajukan pertanyaan atau memberikan perintah yang berkaitan dengan konten yang ada dalam sistem penyimpanan, misalnya, pengguna dapat meminta *virtual assistant* untuk mencari dokumen tertentu berdasarkan kata kunci, judul, atau tema. *OpenAI Assistant* akan melakukan pencocokan dengan pengetahuan yang tersedia dan memberikan hasil yang sesuai.

Implementasi fitur *file search* memiliki beberapa kelebihan seperti memberikan kemudahan bagi pengguna untuk menemukan informasi tanpa perlu secara manual menjelajahi berkas yang banyak. Kelebihan ini sangat berguna dalam konteks pelayanan publik di mana pengguna sering kali dihadapkan dengan sejumlah besar

dokumen dan perlu menemukan informasi spesifik dengan cepat. Fitur *file search* ini meningkatkan produktivitas dengan memungkinkan pengguna mengakses informasi lebih efisien, sehingga mereka dapat fokus pada tugas yang lebih penting dan kemampuan untuk memahami konteks pencarian membantu *OpenAI Assistant* memberikan hasil yang lebih relevan, sehingga meningkatkan pengalaman pengguna secara keseluruhan.

2.2.2 Retrieval Augmented Generation (RAG)

Fitur *file search* menerapkan RAG yang memungkinkan LLM untuk memiliki konteks atau ilmu tambahan yang digunakan untuk menghasilkan respon yang lebih akurat. Adanya sebuah *database* terstruktur atau sebuah kumpulan data yang dapat digunakan *virtual assistant* sebagai referensi dapat meningkatkan akurasi informasi yang diberikan sebagai respon (Wibhowo & Sanjaya, 2021). Kumpulan data yang digunakan dalam konteks *virtual assistant* untuk penanganan *Borderline Personality Disorder (BPD)* adalah kalimat-kalimat distress emosional dan mekanisme penanganan yang relevan. Keberadaan data yang ada sebagai rujukan *virtual assistant* memastikan jawaban yang tervalidasi oleh ahli dan memberikan pengalaman pengguna yang jauh lebih baik.

RAG melakukan pemotongan dokumen sumber menjadi *chunks* yang kemudian disimpan dalam *vector store*. Potongan-potongan yang disimpan dalam *vector store* ini akan diberikan kepada LLM menggunakan *embedding*, yaitu sebuah representasi *vector* yang mengandung makna semantik dan hubungan di antara titik-titik data sedemikian rupa sehingga item-item serupa ditempatkan lebih berdekatan dalam ruang multidimensi. Dokumen yang sudah diberikan kepada LLM akan digunakan untuk menjawab pertanyaan dengan sumber informasi yang relevan dan sudah divalidasi.

2.2.3 GPT-4

GPT-4 adalah LLM yang dikembangkan oleh *OpenAI* sebagai bagian dari model generatifnya. *GPT-4* memanfaatkan arsitektur *transformer* yang memungkinkannya untuk memahami pertanyaan manusia dan memberikan jawaban yang manusiawi. *GPT-4* dimanfaatkan untuk menjadi *virtual assistant* yang membantu dokter gigi sebagai asisten dalam mengambil keputusan diagnosa. *GPT-4* dipilih karena kemampuan tinggi untuk memahami pertanyaan manusia dan memberikan respon yang manusiawi dengan cepat (Suárez *et al.*, 2024).

2.2.4 OpenAI Whisper

OpenAI Whisper adalah layanan *text-to-speech* yang ditawarkan oleh *OpenAI*. *OpenAI Whisper* dapat mengubah teks yang dikirim dalam *API request* menjadi suara dan mengirimnya kembali dalam bentuk *blob*. Hasil suara yang dikembalikan diputar saat *virtual assistant* memberikan jawaban dan meningkatkan pengalaman pengguna (Iannizzotto *et al.*, 2018). Pemanfaatan perintah suara meningkatkan aksesibilitas pengguna disabilitas untuk menggunakan *virtual assistant*.

2.3 Aplikasi Web

Aplikasi web adalah perangkat lunak yang dirancang untuk beroperasi pada internet yang pada dasarnya adalah sebuah *endpoint Hypertext Transfer Protocol (HTTP)* yang berjalan pada sebuah komputer pada *port* 80 atau 443. Aplikasi web berkembang pesat selama keberadaannya terutama dari *framework* dan bahasa pemrograman yang dapat digunakan untuk membangun web seperti *Next.js* dan *Typescript*. *Next.js* adalah *framework fullstack* yang dibangun diatas *React* (Thakkar, 2020). *Next.js* dirancang untuk mempermudah pembuatan aplikasi berbasis *React* dengan menyediakan fitur lengkap seperti *server-side rendering (SSR)* dan *static-site generation (SSG)*.

2.3.1 Next.js

Next.js adalah *framework* open-source yang dibangun diatas *React*, dengan tujuan memungkinkan aplikasi *React* yang di *render* di *server* (Thakkar, 2020). *Next.js* dirancang untuk mempermudah pembuatan aplikasi berbasis *React*, terutama dalam konteks pengembangan *fullstack* dengan fitur seperti *server actions*, *routing* berbasis *folder*, dan lainnya. Fitur unggulan *Next.js* adalah kemampuan untuk melakukan *rendering* sisi *server* (*server-side rendering* atau *SSR*), yang memungkinkan halaman web untuk dibangun di *server* sebelum dikirim ke klien. Ini memberikan keuntungan dalam hal optimasi *Search engine Optimization (SEO)* dan waktu muat awal yang lebih cepat, karena pengguna dapat melihat konten lebih cepat dibandingkan jika semuanya dibangun di sisi klien.

Aspek *routing* pada *Next.js* menggunakan sistem *routing* berbasis *folder*, dimana pengembang cukup membuat *file* dalam direktori *app*, dan *Next.js* otomatis menghasilkan rute untuk halaman tersebut (Ben kora & Manita, 2024). Pendekatan ini menyederhanakan pengembangan dan mengurangi *boilerplate code* yang biasanya diperlukan untuk mengonfigurasi rute secara manual. Pengambilan data pada *Next.js* dilakukan secara langsung dari *server* menggunakan *API* yang dapat dipanggil saat *render*. Kemampuan untuk *rendering* di sisi server dan klien, pengelolaan *routing* yang sederhana, dan dukungan untuk pengambilan data, *Next.js* menjadi pilihan yang sangat baik untuk mengembangkan aplikasi web modern yang responsif, cepat, dan ramah *SEO*.

2.3.2 NextAuth.js

Autentikasi email dan kata sandi, merupakan metode yang banyak digunakan untuk memverifikasi identitas pengguna yang mengakses layanan dan sistem digital. Proses ini mengharuskan pengguna untuk memberikan alamat email yang terdaftar bersama dengan kata sandi yang sesuai untuk mendapatkan akses ke akun mereka. Dasar pemikiran di balik mekanisme otentikasi ini terletak pada kesederhanaan dan kemudahan implementasinya; mekanisme ini memungkinkan pengguna untuk

membuat kredensial unik yang berfungsi sebagai bukti identitas (Ezugwu *et al.*, 2023). *Library NextAuth.js* menyediakan fungsionalitas untuk mengelola pengguna dan sesi dalam aplikasi menggunakan email dan kata sandi. *NextAuth.js* merupakan *library open source* yang memanfaatkan *JSON Web Token (JWT)* untuk menjaga sesi pengguna dengan melakukan abstraksi untuk memudahkan pengembang dalam mengatur autentikasi dalam aplikasi.

2.4 MongoDB

MongoDB adalah basis data *NoSQL* yang dikembangkan oleh *MongoDB, Inc.*, yang menggunakan struktur dokumen untuk menyimpan data, memungkinkan penyimpanan yang lebih besar dan skalabilitas cepat, terutama dalam aplikasi skala besar. Keunggulan utama *MongoDB* adalah kemampuannya untuk secara dinamis meningkatkan jumlah *cluster* sesuai kebutuhan, sehingga dapat menangani lonjakan permintaan data dengan efisien. *MongoDB* mendukung berbagai format data seperti *JSON*, *XML*, dan *CSV*, sehingga memudahkan integrasi dengan sumber data lainnya dan memberikan fleksibilitas dalam pengembangan aplikasi (Byali *et al.*, 2022). Kecepatan baca-tulis yang tinggi juga berkontribusi pada performa aplikasi yang lebih responsif. Fitur keamanan seperti autentikasi, otorisasi, enkripsi, dan pemulihan data sangat penting untuk melindungi informasi dari akses tidak sah. Kombinasi kelebihan ini dan fitur keamanan yang kuat menjadikan *MongoDB* sebagai pilihan pengembang dalam membangun aplikasi yang skalabel dan aman.

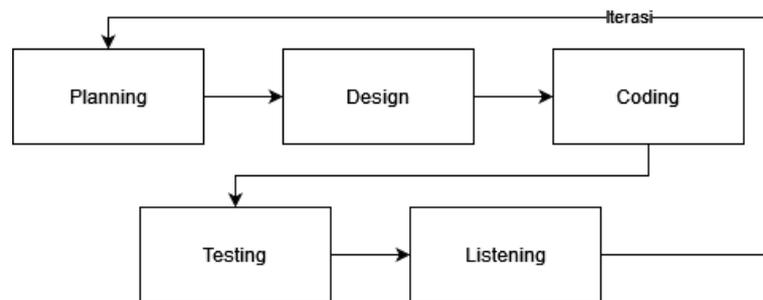
2.5 Typescript

Typescript adalah bahasa pemrograman yang dikembangkan oleh Microsoft sebagai superset dari *Javascript*, di mana semua kode *Javascript* yang valid juga merupakan kode *Typescript* yang valid (Scarsbrook *et al.*, 2023). Dengan menambahkan tipe statis, *Typescript* memungkinkan pengembang untuk mendefinisikan tipe data untuk variabel dan fungsi, sehingga membantu menangkap kesalahan lebih awal dan meningkatkan pemahaman kode. Fitur-fitur modern seperti *async/await*, *decorator*, dan *type modifiers* seperti *template literal*

types yang memberikan fleksibilitas tambahan dalam penulisan kode yang bersih dan terstruktur. Integrasi yang baik dengan *editor* seperti *Visual Studio Code* menawarkan fitur seperti *autocomplete* dan *refactoring*, yang semakin meningkatkan produktivitas pengembang. Secara keseluruhan, *Typescript* berkontribusi pada pembuatan aplikasi yang lebih *robust* dan skalabel dalam pengembangan aplikasi modern.

2.6 Extreme Programming

Extreme Programming (XP) memberikan perspektif mengenai metodologi pengembangan perangkat lunak (Juric, 2000). XP adalah pendekatan yang ringan, efisien, dan fleksibel, yang ditujukan untuk mengatasi tantangan yang dihasilkan oleh metode rekayasa perangkat lunak tradisional.



Gambar 1. Alur Pengembangan *Extreme Programming*

Extreme Programming (XP) terstruktur pada lima langkah utama sesuai Gambar 1, yaitu perencanaan, desain, pengkodean, pengujian, dan mendengarkan. Fase perencanaan dilakukan untuk menentukan ruang lingkup dan prioritas proyek, di mana pengembang terlibat dengan pelanggan untuk mengumpulkan persyaratan melalui *user story* dan memprioritaskannya berdasarkan nilai bisnis dan kelayakan. Fase perencanaan menghasilkan dua artifak, yaitu *user story* dan *use case diagram*. Fase desain menekankan kesederhanaan dan fungsionalitas; fase ini berfokus pada memasukkan hanya fitur-fitur yang diperlukan untuk memenuhi kebutuhan saat ini sambil menumbuhkan pemahaman bersama di antara anggota tim dengan menggunakan metafora yang mengilustrasikan interaksi komponen. Fase desain

menghasilkan tiga artefak, yaitu diagram *Unified Modeling Language* (UML) yang berupa *activity diagram*, ERD dan prototipe. Langkah pengkodean menekankan kolaborasi melalui pemrograman berpasangan di mana dua pengembang bekerja bersama di satu stasiun kerja, namun dalam kasus dimana hanya ada satu pengembang dapat diterapkan pendekatan seperti evaluasi kode secara berkala, pengujian, dan memiliki target untuk selalu meningkatkan kualitas kode. Tahap pengkodean menghasilkan artefak berupa *source code*. Tahap pengujian menuntut hasil pengembangan perangkat lunak berkualitas tinggi dengan mengharuskan pengujian untuk dilakukan pada semua fitur; pengujian ini berfungsi sebagai spesifikasi untuk perilaku yang diharapkan dan diintegrasikan ke dalam siklus pengujian berkelanjutan untuk menangkap masalah lebih awal. Tahap pengujian menghasilkan artefak berupa *test case* dan hasil tes. Langkah mendengarkan atau *listening* menyoroti komunikasi dan masukan dari anggota tim dan pemangku kepentingan di sepanjang siklus hidup proyek, namun dalam kasus satu pengembang dapat dilakukan diskusi dengan klien atau target pengguna mengenai hasil dari perangkat lunak yang dikembangkan. Tahap *listening* menghasilkan artefak *feedback* dari *stakeholder* dan *end-user*.

III. METODE PENELITIAN

3.1. Waktu dan Tempat Penelitian

Waktu dan tempat penelitian ini adalah sebagai berikut:

3.1.1. Waktu Penelitian

Penelitian dilaksanakan dari bulan September hingga Desember tahun ajaran 2024/2025. Waktu penelitian dijelaskan dengan Tabel 1.

Tabel 1. Jadwal Penelitian

Nama Kegiatan	2024															
	September				Oktober				November				Desember			
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
Planning	■	■														
Designing			■	■	■	■										
Coding					■	■	■	■								
Testing								■	■							
Listening										■						
Penulisan Laporan													■	■	■	■

3.1.2. Tempat Penelitian

Penelitian ini dilaksanakan di gedung Jurusan Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam (FMIPA), Kota Bandar Lampung, Lampung.

3.2. Bahan dan Alat Penelitian

Perangkat keras dan lunak yang digunakan dalam penelitian ini adalah sebagai berikut:

3.2.1. Perangkat Keras

Perangkat keras yang digunakan dalam pengembangan ini adalah laptop dengan spesifikasi:

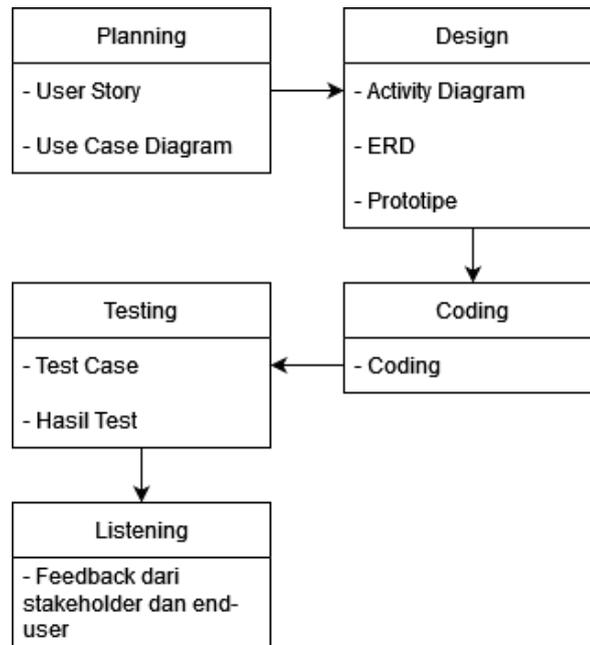
1. Model : ASUS ROG G15 G513
2. CPU : AMD Ryzen 4800H
3. RAM : 24 GB
4. OS : Windows 11

3.2.2. Perangkat Lunak

Perangkat lunak yang digunakan dalam penelitian ini adalah sebagai berikut:

1. *Visual Studio Code version 1.93*
2. *MongoDB Atlas*
3. *Next.js version 14.2.8*
4. *Typescript version 5+*
5. *OpenAI API*

3.3. Metode Penelitian

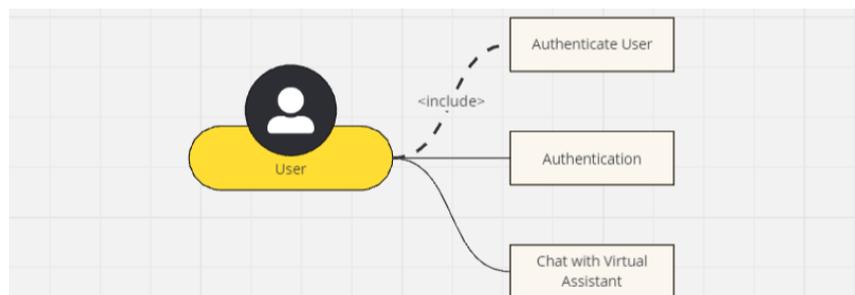


Gambar 2. Tahapan Penelitian

3.3.1. *Planning* (Perencanaan)

Tahap perencanaan melakukan perancangan *user story* yang didapatkan dari pertemuan yang dilakukan dengan perwakilan dari kantor Imigrasi.

3.3.1.1. *Use Case Diagram*



Gambar 3. *Use Case Diagram*

Gambar 3 menunjukkan *use case diagram* dengan satu aktor yang bertindak sebagai *user*. *User* dapat mendaftarkan akun lalu menggunakan akun tersebut untuk masuk ke aplikasi. *User* dapat berkomunikasi dengan *virtual assistant* setelah *login* ke dalam aplikasi dan bisa bertanya untuk mendapatkan respon sesuai *knowledge base* yang dimiliki *virtual assistant*.

3.3.1.2. User Story

User story dirancang untuk memetakan ekspektasi dari *virtual assistant* yang dikembangkan. Setiap *user story* juga memiliki *acceptance criteria* sebagai target.

Tabel 2. *User Story*

ID	<i>User Story</i>	<i>Acceptance Criteria</i>	Iterasi
VA-01	Pengguna dapat mengajukan pertanyaan umum kepada <i>virtual assistant</i> , sehingga bisa mendapatkan informasi tanpa menunggu <i>customer service</i> .	<i>Virtual assistant</i> dapat memahami pertanyaan umum seperti informasi layanan, jam buka, dan alur pengajuan.	Pertama
VA-02	Interaksi antar pengguna harus bisa dipahami dan dijawab oleh <i>virtual assistant</i> dan menjawab pertanyaan dalam bahasa alami, sehingga interaksi terasa lebih manusiawi.	<i>Virtual assistant</i> dapat membalas pertanyaan dengan gaya bahasa yang alami dan ramah. <i>Virtual assistant</i> dapat mengenali variasi pertanyaan dan merespons dengan relevansi tinggi.	Pertama
VA-03	Pengguna dapat berbicara dengan <i>virtual assistant</i>	<i>Virtual assistant</i> bisa mengidentifikasi isu-isu	Kedua

ID	User Story	Acceptance Criteria	Iterasi
	tentang masalah yang lebih kompleks, sehingga mendapatkan solusi yang tepat atau diarahkan ke manusia.	yang lebih rumit dan menawarkan pengetahuan yang mendalam.	
VA-04	Pengguna dapat melihat riwayat percakapan dengan <i>virtual assistant</i> , sehingga dapat meninjau informasi yang telah diberikan sebelumnya.	<i>Virtual assistant</i> menyimpan dan menampilkan riwayat chat dari sesi sebelumnya.	Ketiga
VA-05	Pengguna dapat menggunakan suara untuk melakukan percakapan.	Virtual assistant dapat mendengarkan pertanyaan pengguna.	Keempat

User story pada Tabel 2 didapatkan setelah melakukan diskusi antara pengembang dan klien. Analisis pasar/kompetitor juga dilakukan untuk memahami potensi aplikasi yang akan dikembangkan dan potensi partner teknikal.

3.3.1.3. Acceptance Test Criteria

Tabel 3. *Acceptance Test Criteria*

No.	Kriteria	Iterasi
1	Pengguna dapat membuat akun	Pertama
2	Pengguna dapat masuk ke akun yang telah dibuat	Pertama
3	Pengguna dapat melakukan percakapan dengan <i>virtual assistant</i>	Pertama
4	Bertanya informasi mengenai imigrasi	Kedua
5	Bertanya diluar konteks imigrasi	Kedua

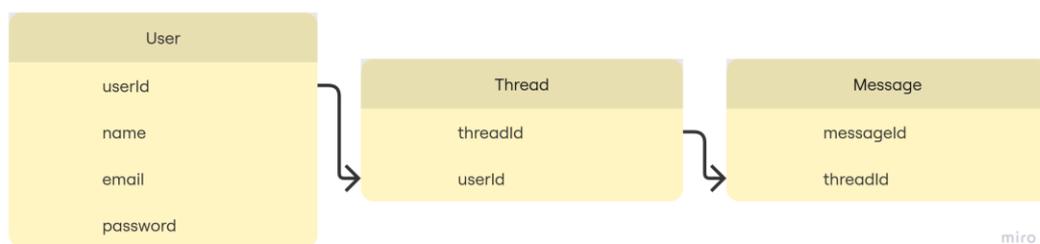
No.	Kriteria	Iterasi
6	Mengulangi pertanyaan untuk menguji konsistensi	Kedua
7	Dapat menyimpan teks dengan <i>format markdown</i> dengan sesuai	Ketiga
8	Setiap <i>message</i> tersimpan dengan <i>threadId</i> yang sesuai	Ketiga
9	Pengguna dapat melihat daftar <i>thread</i> di <i>sidebar</i>	Ketiga
10	Memastikan <i>Mic</i> dapat diakses pada <i>browser</i>	Keempat
11	Menampilkan pertanyaan secara <i>realtime</i>	Keempat
12	Koneksi ke <i>OpenAI Whisper</i>	Keempat

Tabel 3 menunjukkan *acceptance test criteria* yang diuji pada tahapan *testing*. Pengujian ini dilakukan oleh pengembang dan *end-user*.

3.3.2. Design (Perancangan)

Tahapan perancangan dilakukan perancangan *Entity Relationship Diagram*, *Use Case Diagram*, *Activity Diagram*. Berikut diagram rancangan yang telah dibuat:

3.3.2.1. Entity Relationship Diagram



Gambar 4. *Entity Relationship Diagram*

Gambar 4 menunjukkan *Entity Relationship Diagram* (ERD) yang menggambarkan hubungan pada basis data *MongoDB* yang digunakan. Tabel *user* menyimpan data

pengguna dan digunakan untuk autentikasi. Tabel *threads* digunakan menyimpan percakapan pengguna dengan *virtual assistant*. Tabel *message* digunakan untuk menyimpan pertanyaan dan respon.

3.3.2.2. Class Responsibility Collaboration (CRC) Card

Tabel 4. *Chat CRC*

<i>Chat</i>	
Menerima pertanyaan dari <i>user</i>	<i>User</i>
Mengatur pertanyaan masuk dan jawaban	<i>OpenAI</i>
Manajemen <i>thread</i> dan <i>message</i>	<i>Chat History</i>
Menyuarakan jawaban menggunakan	<i>OpenAI</i> <i>Whisper</i>

Tabel 5. *OpenAI CRC*

<i>OpenAI</i>	
Mengirim pertanyaan ke API dan	<i>Chat</i>
mengembalikan jawaban ke	<i>virtual assistant</i>
Mengirim jawaban ke	<i>Whisper</i> API <i>Chat</i>
untuk diubah menjadi suara	

Tabel 6. *Chat History CRC*

<i>Chat History</i>	
Menyimpan dan mengambil <i>message</i> <i>User</i> dan <i>thread</i> dari <i>MongoDB</i>	
Menampilkan <i>thread history</i>	<i>Chat</i>

Tabel 7. *User CRC*

<i>User</i>	
Pengguna dapat membuat akun dan masuk ke akun	<i>Authentication</i>
Pengguna dapat bertanya kepada <i>virtual assistant</i>	<i>Chat</i>
Pengguna dapat melihat <i>chat history</i>	<i>Chat History</i>
Mengatur kepemilikan <i>thread</i> dan <i>message</i>	<i>Authentication</i>

Tabel 8. *Authentication CRC*

<i>Authentication</i>	
Mengatur sesi <i>user</i>	<i>User</i>
Mengatur akses <i>user</i> ke <i>thread</i> dan <i>User message</i>	<i>User</i>

Class-Responsibility-Collaborator atau disingkat CRC merupakan metodologi desain yang digunakan dalam *Object-Oriented Programming* (OOP) untuk menggambarkan konsep komponen dalam sistem dan interaksinya. *CRC card* merupakan pendekatan yang direkomendasikan oleh ahli untuk digunakan dalam metode *XP*. Setiap *card* merepresentasikan sebuah *class* yang menggambarkan tanggung jawab (*responsibility*) pada kolom kiri yang merupakan aksi yang

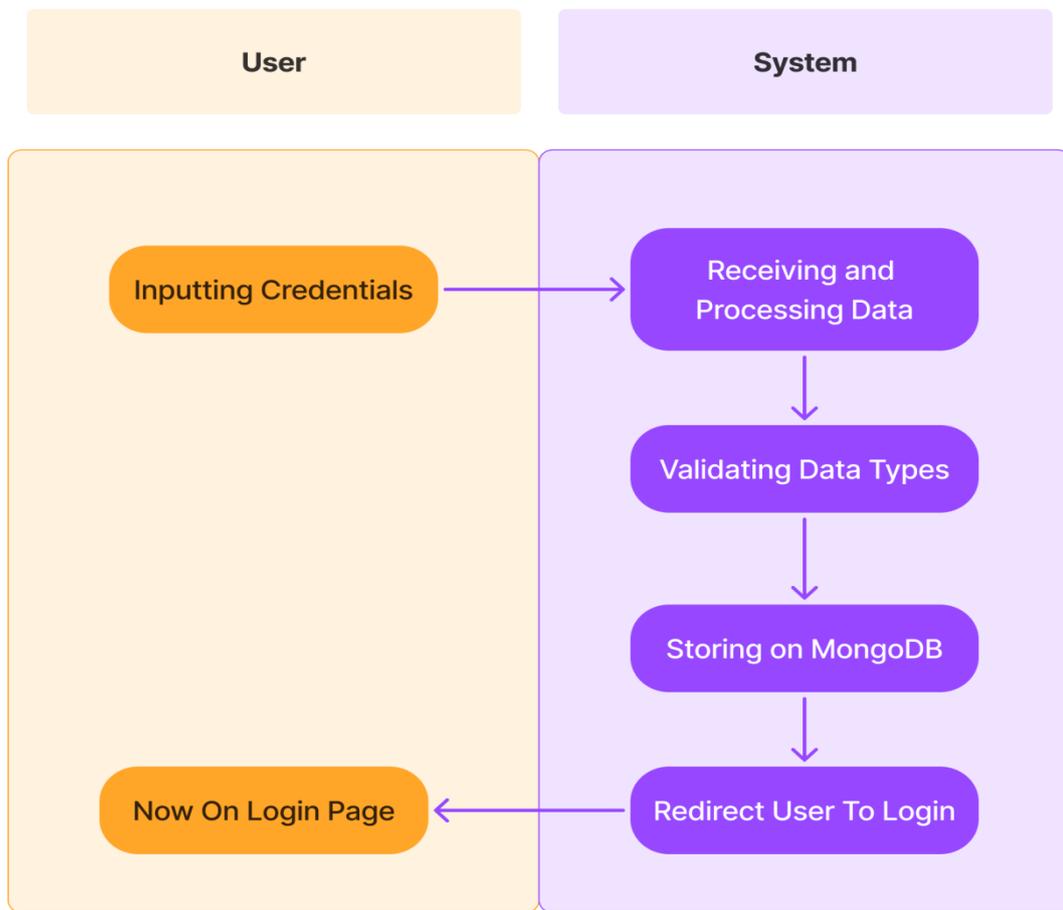
dilakukan dan kolaborator (*collaborator*) pada kolom kanan yang merupakan *class* lainnya yang terhubung untuk melakukan aksi tersebut.

CRC *virtual assistant* terdapat lima *class* yaitu, *Chat*, *OpenAI*, *Chat History*, *User*, dan *Authentication*. Setiap *class* didefinisikan dengan beberapa tanggung jawab dan kolaborator yang berhubungan untuk menyelesaikan aksi tersebut. *Class Chat* memiliki kolaborasi dengan *OpenAI* untuk melakukan percakapan dan penyuaran jawaban, *User* untuk menerima pertanyaan dari pengguna, dan *Chat History* untuk mendapatkan riwayat percakapan. *Class OpenAI* memiliki kolaborasi dengan *Chat* untuk menerima pertanyaan, mengirimnya ke API, mengembalikan jawaban, dan mengubah teks jawaban menjadi suara. *Class Chat History* memiliki kolaborasi dengan *User* untuk menyimpan dan mengambil riwayat percakapan ke *MongoDB* dan *Chat* untuk menampilkan riwayat percakapan. *Class User* memiliki kolaborasi dengan *Authetication* untuk membuat akun baru, masuk ke akun, dan mengatur kepemilikan riwayat percakapan, *Chat* untuk bertanya kepada *virtual assistant*, dan *Chat History* untuk menampilkan riwayat percakapan. *Class Authentication* memiliki kolaborasi dengan *User* untuk mengatur sesi pengguna dan akses ke riwayat percakapan.

3.3.2.3. Activity Diagram

Activity Diagram adalah jenis diagram dalam *Unified Modeling Language* (UML) yang digunakan untuk menggambarkan alur kerja atau proses tertentu dalam sistem. Diagram ini biasanya berfokus pada langkah-langkah yang terlibat dalam proses, serta interaksi di antara berbagai aktor atau komponen yang berpartisipasi.

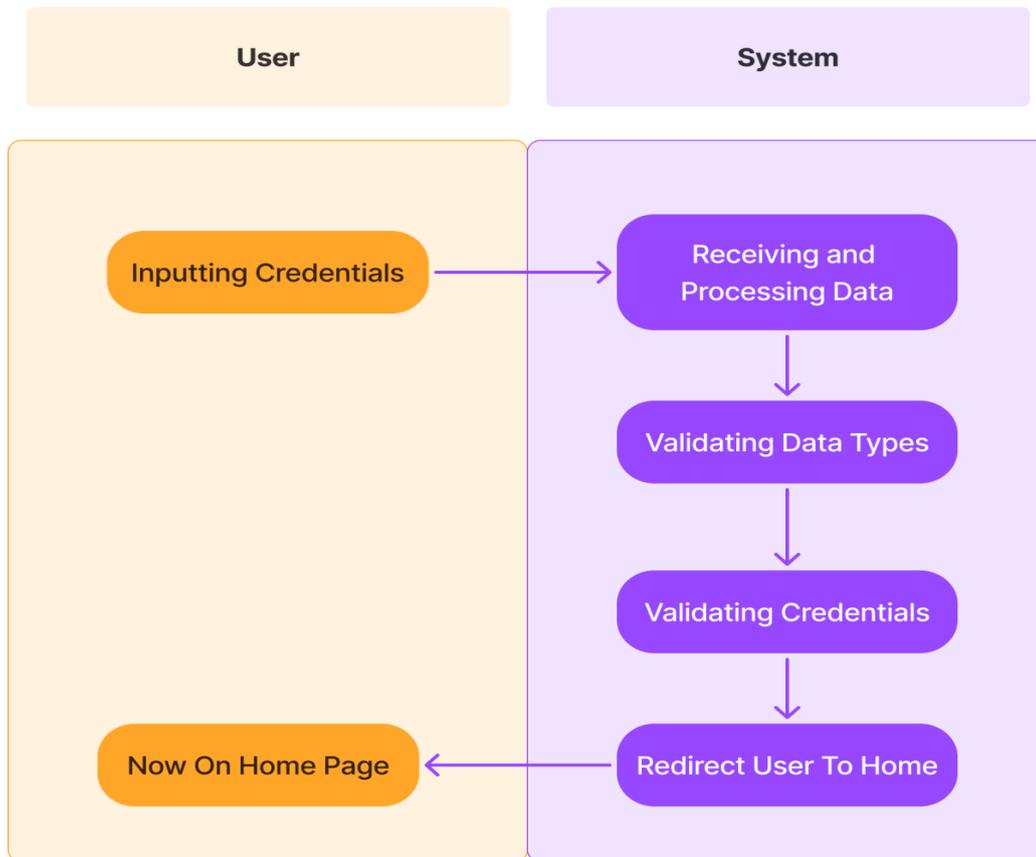
Register Diagram



Gambar 5. Register Activity Diagram

Gambar 5 menunjukkan *activity diagram* untuk proses registrasi yang terdiri dari dua *swimlane*, yaitu *User* dan *System*, yang menunjukkan peran dan aktivitas masing-masing. *Activity* pertama dimulai dengan pengguna mengisi data pendaftaran yang diperlukan, seperti nama, email, dan *password*. Data yang diisi pengguna akan dikirimkan ke API yang ditangani oleh sistem. Data yang diterima API akan diproses untuk melakukan validasi untuk memastikan bahwa informasi yang diberikan oleh pengguna adalah benar dan memenuhi kriteria yang telah ditentukan. *System* akan mengirimkan data yang sudah divalidasi ke *MongoDB SDK* yang melanjutkan dengan menyimpan informasi pengguna ke database *MongoDB*. *User* akan diarahkan ke halaman *login* setelah proses registrasi berhasil.

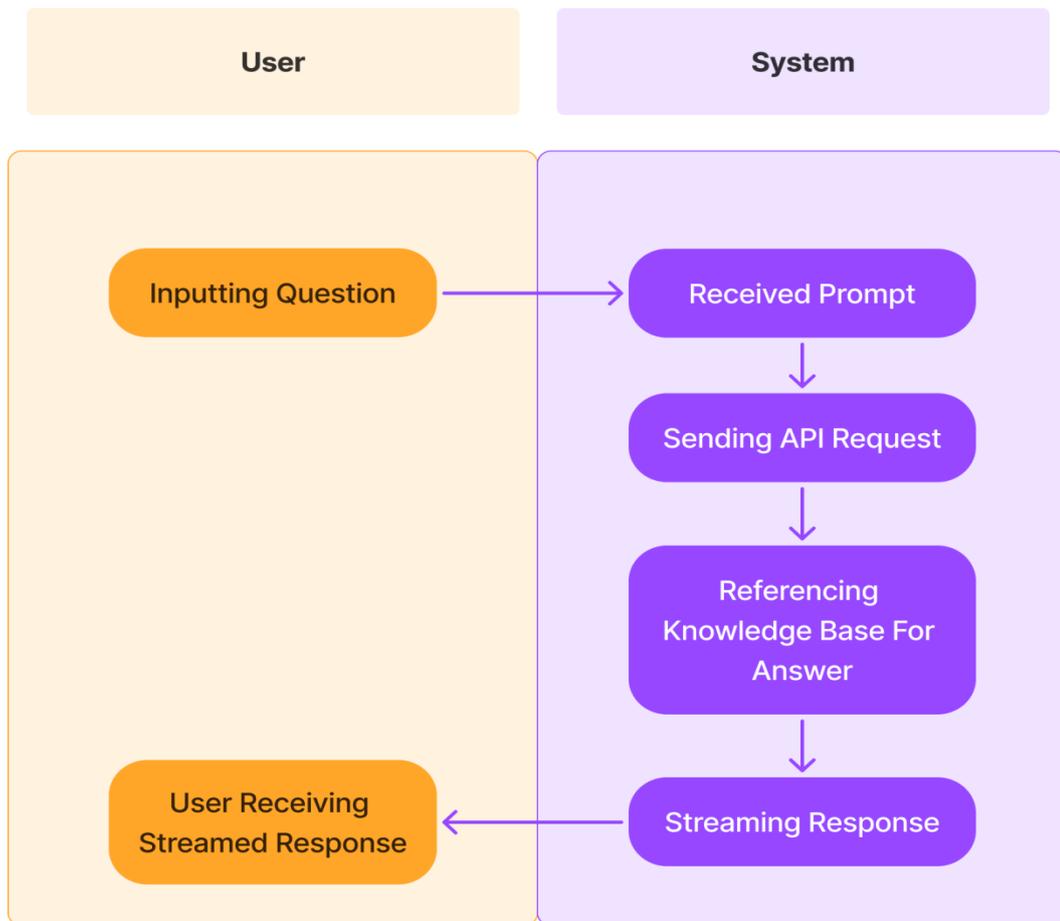
Login Diagram



Gambar 6. *Login Activity Diagram*

Gambar 6 menunjukkan *activity diagram* proses *login* yang terdiri dari dua *swimlane*, yaitu *User* dan *System*. *Activity* dimulai ketika pengguna memasukkan *email* dan *password* halaman login. Pengguna mengirimkan informasi login tersebut ke API yang ditangani oleh sistem. API akan melakukan validasi terhadap *email* dan *password* yang telah dimasukkan. Proses validasi ini melibatkan pencocokan informasi yang diberikan dengan data yang tersimpan di database *MongoDB*, yang apabila kredensial berhasil divalidasi, sistem akan mengarahkan pengguna ke halaman utama, namun jika informasi yang dimasukkan tidak valid, API akan mengirimkan pesan kesalahan kepada pengguna, meminta mereka untuk memeriksa kembali *email* dan *password* yang dimasukkan.

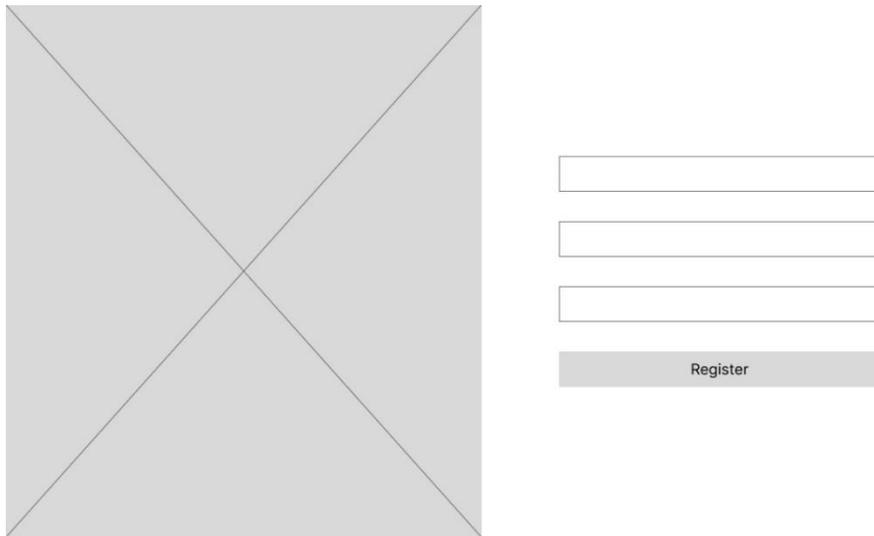
Chat Diagram



Gambar 7. Chat Diagram

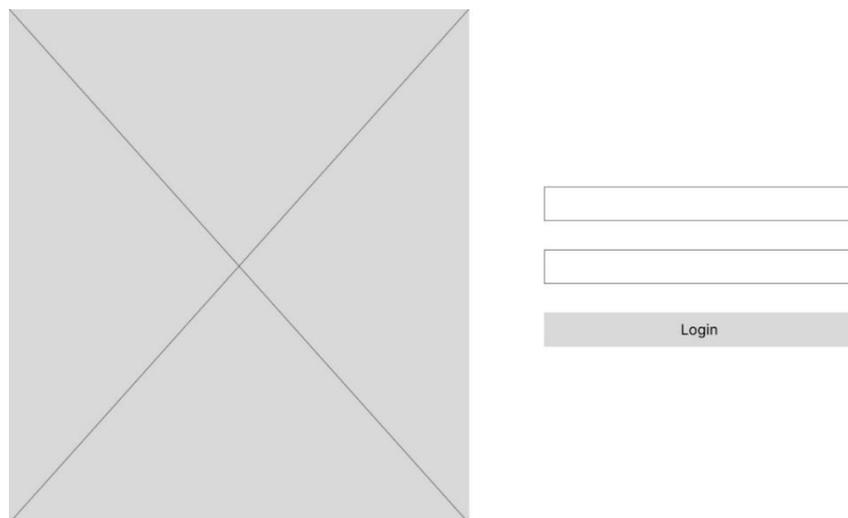
Gambar 7 menunjukkan *activity diagram* proses *chat* yang terdiri dari dua *swimlane*, yaitu *User* dan *System*. *Activity* dimulai ketika pengguna memberikan pertanyaan melalui teks ataupun suara. Pertanyaan yang diberikan akan dikirimkan dan di proses oleh API *OpenAI Assistant* akan merancang jawaban sesuai dengan *knowledge base* yang dimiliki dan mengembalikan jawaban dengan menggunakan LLM (*GPT 4 Omni Mini*). Jawaban akan dikirim ke API *OpenAI Whisper* untuk dilakukan *text-to-speech*. Pengguna akan mendapatkan jawaban dari *virtual assistant* dalam bentuk teks dan suara.

3.3.2.4. *Prototype*

The image shows a prototype for a register page. On the left, there is a gray square with a large 'X' drawn across it from corner to corner. To the right of this square are three horizontal white input fields stacked vertically. Below these fields is a gray rectangular button with the word 'Register' written in white text in the center.

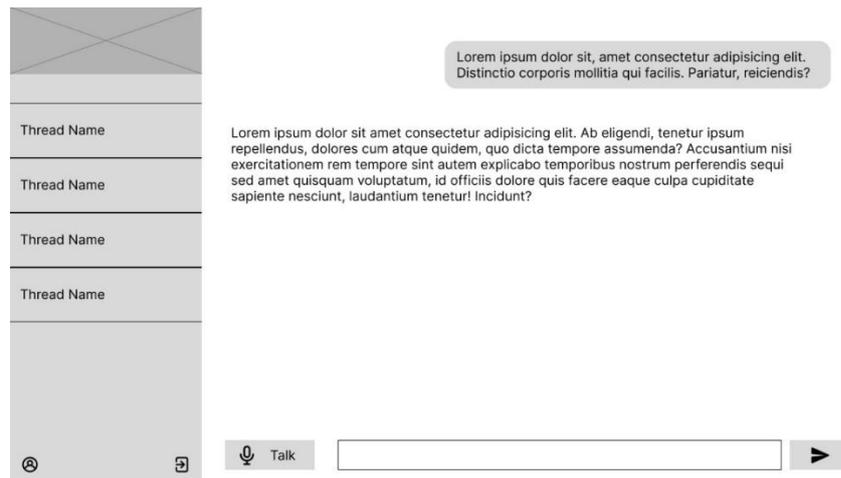
Gambar 8. *Prototype Register Page*

Gambar 8. merupakan *prototype* halaman *register* yang digunakan pengguna untuk membuat akun untuk mengakses *virtual assistant*.

The image shows a prototype for a login page. On the left, there is a gray square with a large 'X' drawn across it from corner to corner. To the right of this square are two horizontal white input fields stacked vertically. Below these fields is a gray rectangular button with the word 'Login' written in white text in the center.

Gambar 9. *Prototype Login Page*

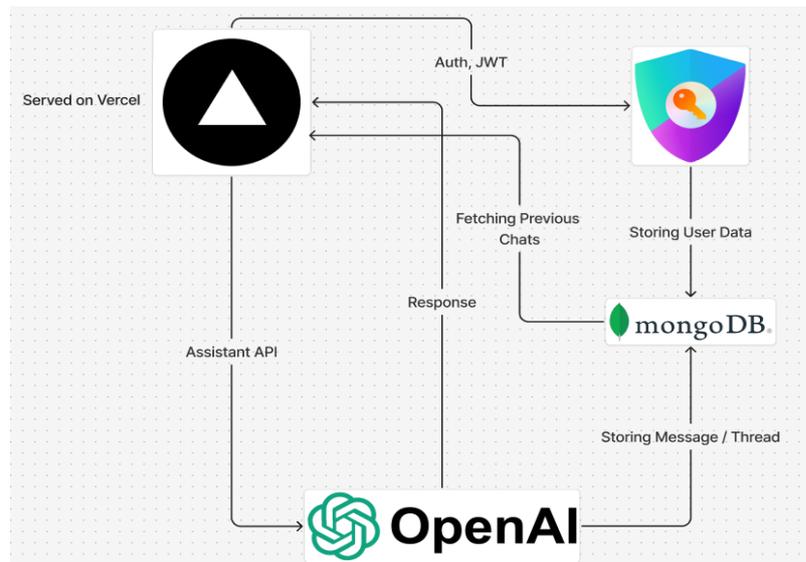
Gambar 9. merupakan *prototype* halaman *login* untuk menggunakan akun yang telah dibuat untuk mengakses *virtual assistant*.



Gambar 10. *Prototype Chat Page*

Gambar 10 merupakan *low-fidelity prototype* yang akan digunakan sebagai acuan dalam merancang tampilan *virtual assistant*. *Prototype* dirancang untuk menggambarkan rencana tampilan halaman-halaman yang akan dikembangkan.

3.3.3. Coding (Pengkodean)



Gambar 11. *Architecture Diagram*

Gambar 11 menunjukkan *architecture diagram* yang menggambarkan alur dan interaksi *antara* berbagai komponen dalam aplikasi *chatbot* AI yang di-*deploy* di *Vercel*. Alur ini akan dijadikan panduan pada tahapan pengkodean.

3.3.3.1. Vercel (Platform Deployment)

Aplikasi di-hosting di *Vercel* yang merupakan sebuah platform untuk *deployment* secara *serverless*.

3.3.3.2. Autentikasi dan JWT (JSON Web Tokens)

Autentikasi ditangani menggunakan penyedia autentikasi pihak ketiga (misalnya, *NextAuth.js*) yang menghasilkan *JWT* setelah pengguna berhasil *login*. *Token* ini kemudian digunakan untuk komunikasi aman antara klien dan *server*, memastikan bahwa permintaan dibuat oleh pengguna yang terautentikasi.

NextAuth.js autentikasi bertanggung jawab untuk mengelola sesi pengguna, mengeluarkan token, dan menangani alur autentikasi. Penyedia ini juga berinteraksi dengan database (*MongoDB*) untuk menyimpan dan mengelola data spesifik pengguna.

3.3.3.3. MongoDB (Database untuk Menyimpan Data Pengguna dan Thread Percakapan)

MongoDB digunakan untuk menyimpan data pengguna, termasuk riwayat percakapan dan informasi *thread*. Ketika pengguna berinteraksi dengan *chatbot*, pertanyaan dan respon disimpan di *MongoDB*. Ini memungkinkan pengguna untuk meninjau kembali percakapan sebelumnya dan menjaga.

3.3.3.4. OpenAI (Assistant API)

Fungsi inti *chatbot* bergantung pada API *OpenAI* yang memproses pertanyaan pengguna dan menghasilkan respons. API ini dipanggil dari *back-end* yang di-hosting di *Vercel*. *Input* pengguna dikirim ke API *OpenAI*, dan respons diambil dan dikembalikan ke pengirim.

3.3.3.5. Alur Kerja Aplikasi

Proses interaksi pengguna dengan sistem dimulai ketika pengguna mengirim pesan melalui antarmuka *front-end*. Pesan tersebut kemudian dikirim ke *back-end* di mana autentikasi dilakukan menggunakan *NextAuth.js* untuk memastikan identitas pengguna. Sistem melanjutkan dengan memeriksa database *MongoDB* untuk menemukan percakapan atau *thread* sebelumnya yang relevan sebagai konteks tambahan. Input yang diberikan oleh pengguna, bersama dengan konteks yang telah diidentifikasi, kemudian dikirim ke *API OpenAI* untuk menghasilkan respons yang sesuai. Respons yang dihasilkan oleh *OpenAI* selanjutnya dikembalikan ke *back-end*. Pesan asli beserta respons dari AI disimpan dalam *MongoDB* untuk referensi di masa mendatang. Respons akhir dikirim kembali kepada pengguna melalui antarmuka *front-end* yang dikelola oleh *Vercel*.

3.3.4. Testing (Pengujian)

Pengujian dilakukan untuk memenuhi *acceptance criteria* seperti yang dapat dilihat di Tabel 3. Metode yang digunakan adalah *Blackbox Testing*, karena pengujian ini tidak melibatkan kode atau cara kerja internal aplikasi. Umpan balik dari pengguna akan digunakan untuk menentukan apa yang harus diubah dan disesuaikan untuk memenuhi *acceptance criteria*. *Refactor* dan revisi kode akan dilakukan untuk memenuhi *acceptance criteria* selama masa iterasi. Setelah pengujian *acceptance criteria* terpenuhi akan lanjut ke iterasi selanjutnya.

3.3.5. Listening (Pendengaran)

Tahap *listening* dilakukan untuk mengumpulkan umpan balik dan masukan dari pengguna dan *stakeholder* untuk meningkatkan kualitas layanan yang disediakan. Pengguna akan berinteraksi dengan virtual assistant dan data percakapan akan dianalisis, mencakup pertanyaan yang diajukan dan tingkat kepuasan terhadap respons sistem. Umpan balik ini diperoleh melalui survei atau formulir, yang memberikan wawasan mengenai pengalaman pengguna dan efektivitas *virtual*

assistant. Informasi tersebut digunakan untuk memperbarui *knowledge base OpenAI Assistant* agar tetap relevan dengan informasi terkini mengenai layanan imigrasi, merujuk pada sumber resmi dari Direktorat Jenderal Imigrasi. Proses ini bersifat berkelanjutan; setiap iterasi akan diuji kembali dengan melibatkan pengguna dan *stakeholder* dalam pengujian beta sebelum rilis final, sehingga perbaikan dapat dilakukan secara konsisten.

V. SIMPULAN DAN SARAN

5.1. Simpulan

Penelitian ini berhasil mengembangkan *virtual assistant* yang dapat secara efektif meningkatkan kualitas layanan *customer service* Imigrasi. Data yang didapatkan menunjukkan bahwa *virtual assistant* jauh lebih efektif dan efisien daripada *FAQ* dalam menyediakan informasi mengenai paspor, pengajuan visa, dan lain-lain. *Virtual assistant* memungkinkan pengguna untuk mendapatkan informasi dalam waktu yang lebih cepat dibandingkan dengan mencari *FAQ*.

Simpulan dari penelitian yang telah dilakukan adalah sebagai berikut:

1. Penelitian ini berhasil mengembangkan virtual assistant untuk Imigrasi dengan menggunakan teknologi *OpenAI Assistant*, *Next.js* dan *MongoDB*
2. Fitur seperti *chat history*, *speech-to-text*, dan *text-to-speech* berhasil diimplementasikan, meningkatkan aksesibilitas dan pengalaman pengguna
3. *Virtual assistant* yang dikembangkan dapat menjawab pertanyaan mengenai Imigrasi berdasarkan *knowledge base* dengan sangat baik
4. Pengujian oleh *end-user* menunjukkan tingkat kepuasan yang tinggi terhadap fungsionalitas, kecepatan respon, dan jawaban yang dihasilkan *virtual assistant*

5.2. Saran

Saran yang dapat diberikan untuk melanjutkan penelitian ini adalah sebagai berikut:

1. Penggunaan LLM *open-source* yang di *deploy on-premise/private* untuk menjaga data agar tidak keluar dari Imigrasi
2. Menggunakan algoritma *speech-to-text* yang lebih ringan dan *open-source*
3. Meneruskan pengembangan dengan menerapkan *RAG* secara manual dengan *tools* seperti *LangChain* atau *Haystack* untuk meningkatkan fleksibilitas dan penambahan fitur

DAFTAR PUSTAKA

- Audi Albtoush, E. al. (2023). ChatGPT: Revolutionizing User Interactions with Advanced Natural Language Processing. *International Journal on Recent and Innovation Trends in Computing and Communication*, 11(9), 3354–3360. <https://doi.org/10.17762/ijritcc.v11i9.9541>
- Ben kora, H. H., & Manita, M. S. (2024). Modern Front-End Web Architecture Using React.js and Next.js. *University of Zawia Journal of Engineering Sciences and Technology*, 2(1), 1–13. <https://doi.org/10.26629/uzjest.2024.01>
- Bernard, D., & Arnold, A. (2019). Cognitive interaction with virtual assistants: From philosophical foundations to illustrative examples in aeronautics. *Computers in Industry*, 107, 33–49. <https://doi.org/10.1016/j.compind.2019.01.010>
- Binns, R. (2017). Fairness in Machine Learning: Lessons from Political Philosophy. *Proceedings of Machine Learning Research*, 81, 149–159. <https://arxiv.org/abs/1712.03586v3>
- Byali, R., Jyothi, Ms., & Shekadar, M. C. (2022). "Evaluation of NoSQL Database MongoDB with Respect to JSON Format Data Representation ". *International Journal of Research Publication and Reviews*, 867–871. <https://doi.org/10.55248/gengpi.2022.3.9.24>
- Chang, C.-C., Cheng, W.-S., & Hsiao, S. (2022). Customer Service Chatbot Enhanced with Conversational Language Understanding and Knowledge Base. *2022 IEEE 4th Eurasia Conference on IOT, Communication and Engineering (ECICE)*, 231–234. <https://doi.org/10.1109/ECICE55674.2022.10042940>
- Ezugwu, A., Ukwandu, E., Ugwu, C., Ezema, M., Olebara, C., Ndunagu, J., Ofusori, L., & Ome, U. (2023). Password-based authentication and the experiences of

end users. *Scientific African*, 21, e01743.
<https://doi.org/10.1016/J.SCIAF.2023.E01743>

Iannizzotto, G., Bello, L. Lo, Nucita, A., & Grasso, G. M. (2018). A vision and speech enabled, customizable, virtual assistant for smart environments. *Proceedings - 2018 11th International Conference on Human System Interaction, HSI 2018*, 50–56. <https://doi.org/10.1109/HSI.2018.8431232>

Ischen, C., Araujo, T. B., Voorveld, H. A. M., Van Noort, G., & Smit, E. G. (2022). Is voice really persuasive? The influence of modality in virtual assistant interactions and two alternative explanations. *Internet Research*, 32(7), 402–425. <https://doi.org/10.1108/INTR-03-2022-0160>

Juric, R. (2000). Extreme programming and its development practices. *ITI 2000. Proceedings of the 22nd International Conference on Information Technology Interfaces (Cat. No.00EX411)*, 97–104.

Moussiades, L., & Zografos, G. (2023). *OpenAi's GPT4 as coding assistant*. <https://arxiv.org/abs/2309.12732v1>

Putu Denny Artina Mahendra, Komang Adi Sastra Wijaya, & I Ketut Winaya. (2024). Optimalisasi Layanan M-Paspor Dari Sudut Pandang Responsiveness dan Reliability di Kantor Imigrasi Denpasar. *Sawala : Jurnal Administrasi Negara*, 12(1), 229–239. <https://doi.org/10.30656/SAWALA.V12I1.8133>

Scarsbrook, J., Utting, M., & Ko, R. (2023). *TypeScript's Evolution: An Analysis of Feature Adoption Over Time*. <https://doi.org/10.48550/arXiv.2303.09802>

Staiculescu, O. (2023). From Bureaucracy to Digital Efficiency: AI Adoption in Romania's Public Sector. *Academic Journal of Law and Governance*, 11.1-11.2, 15–20. <https://doi.org/10.56177/AJLG.11.1.11.2.2023.ART.2>

Suárez, A., Jiménez, J., Llorente de Pedro, M., Andreu-Vázquez, C., Díaz-Flores García, V., Gómez Sánchez, M., & Freire, Y. (2024). Beyond the Scalpel: Assessing ChatGPT's potential as an auxiliary intelligent virtual assistant in oral surgery. *Computational and Structural Biotechnology Journal*, 24, 46–52. <https://doi.org/10.1016/j.csbj.2023.11.058>

- Thakkar, M. (2020). Building React Apps with Server-Side Rendering: Use React, Redux, and Next to Build Full Server-Side Rendering Applications. *Building React Apps with Server-Side Rendering: Use React, Redux, and Next to Build Full Server-Side Rendering Applications*, 1–192. <https://doi.org/10.1007/978-1-4842-5869-9>
- Trivedi, J. (2019). Examining the Customer Experience of Using Banking Chatbots and Its Impact on Brand Love: The Moderating Role of Perceived Risk. *Journal of Internet Commerce*, 18(1), 91–111. <https://doi.org/10.1080/15332861.2019.1567188>
- Wibhowo, C., & Sanjaya, R. (2021). Virtual Assistant to Suicide Prevention in Individuals with Borderline Personality Disorder. *2021 International Conference on Computer & Information Sciences (ICCOINS)*, 234–237. <https://doi.org/10.1109/ICCOINS49721.2021.9497160>
- Xu, Y., Shieh, C.-H., van Esch, P., & Ling, I.-L. (2020). AI Customer Service: Task Complexity, Problem-Solving Ability, and Usage Intention. *Australasian Marketing Journal*, 28(4), 189–199. <https://doi.org/10.1016/j.ausmj.2020.03.005>
- Zhao, Y., Zhang, T., Liu, Y., Zhu, Y., & Gao, Y. (2021). Research on the Influence Mechanism of Artificial Intelligence(AI) Customer Service on User Satisfaction with Online Shopping. *2021 2nd International Conference on Computer Science and Management Technology (ICCSMT)*, 253–260. <https://doi.org/10.1109/ICCSMT54525.2021.00056>