

**SISTEM *CHATBOT* DENGAN *LARGE LANGUAGE*
MODELS DAN *VECTOR DATABASE* UNTUK LAYANAN
INFORMASI DINAS KEPENDUDUKAN DAN
PENCATATAN SIPIL KOTA SEMARANG**

SKRIPSI



**Diajukan oleh :
EROS ALFEDO HERMANTO
NIM : 2108096006**

**PROGRAM STUDI TEKNOLOGI INFORMASI
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI WALISONGO
SEMARANG**

2025

PERNYATAAN KEASLIAN NASKAH

Yang bertanda tangan di bawah ini :

Nama : Eros Alfredo Hermanto

NIM : 2108096006

Jurusan : Teknologi Informasi

Menyatakan bahwa skripsi berjudul :

**SISTEM *CHATBOT* DENGAN *LARGE LANGUAGE MODELS* DAN
VECTOR DATABASE UNTUK LAYANAN INFORMASI DINAS
KEPENDUDUKAN DAN PENCATATAN SIPIL KOTA
SEMARANG**

Secara keseluruhan adalah hasil penelitian/karya sendiri, kecuali bagian tertentu yang dirujuk dari sumbernya

Semarang, 12 Januari 2025

Pembuat pernyataan

Eros Alfredo Hermanto

2108906006

LEMBAR PENGESAHAN



KEMENTERIAN AGAMA REPUBLIK INDONESIA
UNIVERSITAS ISLAM NEGERI WALISONGO SEMARANG
FAKULTAS SAINS DAN TEKNOLOGI

Alamat: Jl. Prof. Dr. Hamka Km.1 Semarang
E-mail: fst@walisongo.ac.id Web: <http://fst.walisongo.ac.id>

LEMBAR PENGESAHAN

Naskah Skripsi Berikut :

Judul : Sistem *Chatbot* dengan *Large Language Models* dan *Vector Database* untuk
Layanan Informasi Dinas Kependudukan dan Pencatatan Sipil Kota Semarang
Penulis : Eros Alfredo Hermanto
NIM : 2108096006
Jurusan : Teknologi Informasi

Telah diujikan dalam sidang skripsi oleh Dewan Penguji Fakultas Sains dan Teknologi
UIN Walisongo Semarang dan dapat diterima sebagai salah satu syarat memperoleh gelar
sarjana dalam bidang ilmu teknologi informasi

Semarang , 23 April 2025

DEWAN PENGUJI

Ketua Sidang

Heri Mustofa, M.Kom.

NIP. 198703172019031007

Sekretaris Sidang

Nur Cahyo Hendro Wibowo, S.T., M.Kom

NIP. 197312222006041001

Penguji Utama I

Adzhar Arwani Mufidh, M.Kom

NIP. 199107032019031006

Penguji Utama II

Mokhamad Ikil Mustofa, M.Kom

NIP. 198808072019031010

Pembimbing I

Nur Cahyo Hendro Wibowo S.T., M.Kom

NIP. 197312222006041001

Pembimbing II

Dr. Wenty Dwi Purniarti S.Pd., M.Kom.

NIP. 197706322006042005

NOTA PEMBIMBING

Yth. Ketua Program Studi Teknologi Informasi
Fakultas Sains dan Teknologi UIN Walisongo Semarang

Assalamu'alaikum Warahmatullah Wabarakatuh,

Dengan hormat, saya ingin menginformasikan bahwa saya telah melakukan bimbingan, arahan, dan koreksi terhadap naskah skripsi dengan :

Judul : Sistem *Chatbot* dengan *Large Language Models* dan
Vector Database untuk Layanan Informasi Dinas
Kependudukan dan Pencatatan Sipil Kota Semarang

Nama : Eros Alfredo Hermanto

Nim : 2108096006

Jurusan : Teknologi Informasi

Saya memandang bahwa naskah skripsi tersebut sudah dapat diajukan kepada Program Studi Teknologi Informasi dan Fakultas Sains dan Teknologi untuk ujian dalam ujian munaqasah di UIN Walisongo.

Wassalamualaikum Warahmatullahi Wabarakatuh.

Semarang, 12 Januari 2025
Pembimbing I

Nur Cahyo Hendro Wibowo S.T., M.Kom

NIP. 197312222006041001

NOTA PEMBIMBING

Yth. Ketua Program Studi Teknologi Informasi
Fakultas Sains dan Teknologi UIN Walisongo Semarang

Assalamu'alaikum Warahmatullah Wabarakatuh,

Dengan hormat, saya ingin menginformasikan bahwa saya telah melakukan bimbingan, arahan, dan koreksi terhadap naskah skripsi dengan :

Judul : Sistem *Chatbot* dengan *Large Language Models* dan
Vector Database untuk Layanan Informasi Dinas
Kependudukan dan Pencatatan Sipil Kota Semarang

Nama : Eros Alfredo Hermanto

Nim : 2108096006

Jurusan : Teknologi Informasi

Saya memandang bahwa naskah skripsi tersebut sudah dapat diajukan kepada Program Studi Teknologi Informasi dan Fakultas Sains dan Teknologi untuk ujian dalam ujian munaqasah di UIN Walisongo.

Wassalamualaikum Warahmatullahi Wabarakatuh.

Semarang, 12 Januari 2025
Pembimbing II

Dr. Wenty Dwi Yuniarti S.Pd., M.Kom.

NIP. 197706222006042005

LEMBAR PERSEMBAHAN

Dengan penuh rasa syukur dan terima kasih, saya persembahkan skripsi ini kepada :

1. Orang Tua Tercinta

Bagi Ayah dan Ibu yang selalu memberikan cinta, doa, dan dukungan tanpa henti. Terima kasih atas segala pengorbanan dan bimbingan yang tiada duanya dalam kehidupan saya. Semoga karya ini dapat menjadi bukti rasa terima kasih dan dedikasi saya kepada kalian.

2. Dosen Pembimbing

Kepada Bapak Cahyo dan Ibu Wenty, yang dengan sabar memberikan bimbingan, arahan, dan pengetahuan yang sangat berharga selama proses penyusunan skripsi ini. Terima kasih atas segala bantuan dan dukungannya dalam mewujudkan penelitian ini.

3. Teman-teman Mahasiswa

Untuk teman-teman seangkatan yang selalu mendukung dan memberikan semangat di setiap langkah saya. Terima kasih atas persahabatan serta kebersamaan yang selalu membangkitkan motivasi.

MOTTO

“Kamu hanya perlu lebih baik dari hari kemarin, bukan orang lain.”

SISTEM *CHATBOT* DENGAN *LARGE LANGUAGE MODELS* DAN *VECTOR DATABASE* UNTUK LAYANAN INFORMASI DINAS KEPENDUDUKAN DAN PENCATATAN SIPIL KOTA SEMARANG

Oleh :
Eros Alfredo Hermanto
2108096006

ABSTRAK

Penelitian ini membahas pengembangan sistem *chatbot* berbasis *Large Language Models (LLM)* dan *vector database* untuk layanan informasi di Dinas Kependudukan dan Pencatatan Sipil Kota Semarang. Tujuan penelitian adalah merancang *chatbot* yang responsif, akurat, dan tersedia 24/7 guna meningkatkan aksesibilitas dan efisiensi layanan publik. Sistem ini mengintegrasikan *Langchain*, *GPT-4o Mini*, dan *Pinecone*. *Langchain* digunakan untuk logika percakapan, *GPT-4o Mini* memproses bahasa alami, dan *Pinecone* mempercepat pencarian data berbasis *vector*. *FastAPI* berfungsi sebagai *backend*, *Twilio* menghubungkan *chatbot* ke *WhatsApp*, dan *Redis* menyimpan riwayat percakapan untuk meningkatkan responsivitas. Metode pengembangan menggunakan pendekatan *waterfall*, mulai dari analisis kebutuhan, desain sistem, implementasi, hingga pengujian dengan metode *Black Box Testing* dan *User Acceptance Testing (UAT)*. Hasil pengujian menunjukkan sistem mampu menjawab 85% pertanyaan pengguna secara akurat (17 dari 20 skenario). Evaluasi UAT dengan 23 responden menghasilkan skor kelayakan 87% (kategori Sangat Layak), dengan aspek fungsionalitas (88%), kegunaan (88%), efisiensi (86%), dan keandalan (86%). *Chatbot* ini berhasil memberikan informasi kependudukan dan pencatatan sipil secara akurat serta mengurangi beban kerja pegawai. Sistem telah dipublikasikan secara online dan dapat diakses 24/7. Evaluasi menunjukkan beberapa kendala, seperti pemahaman terhadap pertanyaan spesifik dan kecepatan respons yang bervariasi. Kendala tersebut diatasi dengan memperluas dataset dan

mengoptimalkan sistem. Kesimpulannya, *chatbot* ini memberikan solusi inovatif dan efisien untuk layanan informasi pemerintahan, meningkatkan transparansi, dan mempermudah masyarakat dalam mengakses informasi kependudukan.

Kata Kunci : *Chatbot, Large Language Models (LLM), vector database, Langchain, GPT-4o Mini, Pinecone, FastAPI, Twilio, Redis, waterfall, Black Box Testing, User Acceptance Testing (UAT).*

.

**CHATBOT SYSTEM WITH *LARGE LANGUAGE MODELS*
AND *VECTOR DATABASE* FOR INFORMATION
SERVICES OF THE POPULATION AND CIVIL
REGISTRATION OFFICE OF SEMARANG CITY**

By :
Eros Alfredo Hermanto
2108096006

ABSTRACT

This study discusses the development of a *chatbot* system based on *Large Language Models* (LLMs) and a *vector database* to support information services at the Department of Population and Civil Registration in Semarang City. The aim is to design a responsive, accurate, and always-available *chatbot* to improve public service accessibility and efficiency. The system integrates Langchain for conversational logic, GPT-4o Mini for natural language processing, and Pinecone for fast vector-based data retrieval. FastAPI functions as the backend, Twilio connects the *chatbot* to WhatsApp, and Redis stores conversation history to enhance responsiveness. The development follows a waterfall model, including requirements analysis, system design, implementation, and testing using Black Box Testing and User Acceptance Testing (UAT). Test results showed 85% accuracy (17 out of 20 scenarios), and UAT involving 23 respondents yielded an 87% feasibility score (Highly Feasible), with functionality, usability, efficiency, and reliability scoring above 85%. The *chatbot* effectively provides population and civil registration information while reducing staff workload. It is available online 24/7. Some challenges, such as understanding specific queries and inconsistent response times, were addressed by expanding the dataset and optimizing the system. Overall, the *chatbot* offers an innovative, efficient solution for government information services, improving transparency and public access to essential information.

Keywords: *Chatbot, Large Language Models (LLM), vector database, Langchain, GPT-4o Mini, Pinecone, FastAPI, Twilio, Redis, waterfall, Black Box Testing, User Acceptance Testing (UAT).*

KATA PENGANTAR

Puji syukur penulis panjatkan kepada Allah SWT, Tuhan Yang Maha Esa, atas segala rahmat, hidayah, dan karunia-Nya yang tiada terhingga. Dengan izin-Nya, penulis dapat menyelesaikan skripsi ini yang berjudul "*Sistem Chatbot dengan Large Language Models dan Vector Database untuk Layanan Informasi Dinas Kependudukan dan Pencatatan Sipil Kota Semarang*." Skripsi ini disusun sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer di Universitas Islam Negeri Walisongo Semarang.

Proses penyusunan skripsi ini tidak terlepas dari bantuan, dukungan, dan bimbingan yang sangat berarti dari berbagai pihak. Oleh karena itu, penulis ingin menyampaikan rasa terima kasih yang sebesar-besarnya kepada:

1. Bapak/Ibu Dosen Pembimbing, yang telah memberikan bimbingan, masukan, dan arahan yang sangat berharga dalam setiap tahap penyusunan skripsi ini. Tanpa bimbingan mereka, penulis tidak dapat menyelesaikan skripsi ini dengan baik.
2. Seluruh Dosen di Program Studi Teknik Informatika, yang telah memberikan ilmu, pengetahuan, dan pengalaman yang sangat berguna selama masa perkuliahan. Semangat dan dedikasi mereka dalam mengajar telah memberikan

inspirasi bagi penulis untuk terus berjuang dalam mencapai tujuan akademis ini.

3. Pihak Dinas Kependudukan dan Pencatatan Sipil Kota Semarang, yang telah memberikan akses, data, dan informasi yang sangat penting untuk kelancaran penelitian ini. Kerjasama yang terjalin sangat membantu penulis dalam menggali informasi yang relevan untuk penelitian ini.
4. Kedua orang tua tercinta, yang telah memberikan dukungan, doa, dan kasih sayang tanpa batas. Tanpa mereka, penulis tidak akan pernah sampai pada titik ini. Pengorbanan mereka adalah sumber kekuatan dan motivasi terbesar dalam perjalanan hidup penulis.
5. Teman-teman seperjuangan, yang selalu memberikan dukungan moral, semangat, dan motivasi. Kehadiran mereka memberikan warna dan kekuatan dalam setiap tantangan yang dihadapi selama proses penyusunan skripsi ini.
6. Keluarga besar yang senantiasa memberikan dorongan dan semangat baik secara moral maupun materil. Mereka selalu berada di samping penulis, memberikan dukungan tanpa henti dalam setiap langkah perjalanan ini.
7. Rekan-rekan sejawat, yang turut membantu dalam berbagai hal teknis dan memberi kontribusi berharga dalam menyelesaikan skripsi ini. Bantuan dan kerjasama

mereka sangat memudahkan penulis dalam menyelesaikan penelitian ini.

Penulis menyadari bahwa skripsi ini masih jauh dari kesempurnaan. Oleh karena itu, kritik dan saran yang membangun dari berbagai pihak sangat diharapkan demi perbaikan dan pengembangan lebih lanjut. Semoga skripsi ini dapat memberikan manfaat bagi dunia pendidikan dan pengembangan teknologi informasi, khususnya dalam bidang pelayanan publik. Penulis berharap bahwa penelitian ini dapat memberikan kontribusi positif dalam memajukan sistem pelayanan informasi di Dinas Kependudukan dan Pencatatan Sipil Kota Semarang serta memotivasi pengembangan teknologi *chatbot* di sektor publik.

Semoga segala upaya yang dilakukan selama proses penyusunan skripsi ini mendapatkan berkah dan memberikan manfaat bagi kemajuan ilmu pengetahuan dan teknologi.

Semarang, 12 Januari 2025

Penulis

DAFTAR ISI

PERNYATAAN KEASLIAN NASKAH.....	ii
LEMBAR PENGESAHAN.....	iii
NOTA PEMBIMBING.....	iv
NOTA PEMBIMBING.....	v
LEMBAR PERSEMBAHAN	vi
MOTTO.....	vii
ABSTRAK.....	viii
ABSTRACT	x
KATA PENGANTAR.....	xii
DAFTAR ISI.....	xv
DAFTAR GAMBAR	xviii
DAFTAR TABEL.....	xix
DAFTAR LAMPIRAN	xx
BAB I.....	1
A. Latar Belakang.....	1
B. Rumusan Masalah	6
C. Tujuan dan Kegunaan Penelitian	7
D. Batasan Penelitian	7
E. Manfaat Penelitian.....	8
F. Sistematika Penelitian	10
BAB II.....	11
A. Kajian Teori.....	11

B. Kajian Penelitian yang Relevan.....	17
BAB III	24
A. Metode Pengumpulan Data.....	24
B. Metode Pengembangan Aplikasi.....	25
1. Analisis Kebutuhan	26
2. Desain Sistem.....	28
3. Implementasi.....	34
4. Pengujian.....	36
5. Maintenance	42
BAB IV	43
A. Sumber Data.....	43
B. Hasil Implementasi.....	44
C. Akurasi <i>Chatbot</i>	74
D. <i>UAT</i>	88
E. Maintenance	97
BAB V	99
A. Kesimpulan	99
B. Saran	100
DAFTAR PUSTAKA	102
LAMPIRAN	106
Lampiran 1 : Lembar persetujuan pembimbing proposal	106
Lampiran 2 : Lembar pengesahan ujian proposal.....	107
Lampiran 3 : Statistik penggunaan API <i>OpenAI</i>	108
Lampiran 4 : <i>RedisInsight</i>	109

Lampiran 5 : Penyimpanan Riwayat Chat <i>Redis</i>	110
Lampiran 6 : Status Indeks <i>Vector</i> pada <i>Pinecone</i>	111
Lampiran 7 : Status Hosting Aplikasi pada Railway	112
Lampiran 8 : Hasil Kuisoner <i>UAT</i>	113
Lampiran 9 : Twilio.....	114
Lampiran 10 : Data Primer.....	115
Lampiran 11 : Data Sekunder.....	116
DAFTAR RIWAYAT HIDUP	117

DAFTAR GAMBAR

Gambar	Judul	Halaman
Gambar 3.1	Metode Waterfall	26
Gambar 3.2	Arsitektur Sistem	29
Gambar 3.3	Use Case Diagram	31
Gambar 3.4	Flowchart Alur Interaksi Pengguna	33
Gambar 3.5	Implementasi	35
Gambar 4.1	Profile <i>Chatbot</i>	70
Gambar 4.2	Percakapan Bot	70

DAFTAR TABEL

Tabel	Judul	Halaman
Tabel 2.1	Kajian Penelitian yang relevan	17
Tabel 3.1	Penilaian Mean Option Score	39
Tabel 3.2	Kriteria Penilaian Skor	40
Tabel 3.3	Daftar Pertanyaan <i>UAT</i>	41
Tabel 4.1	Implementasi Teknologi dalam sistem <i>chatbot</i>	45
Tabel 4.2	Skenario Pengujian	75
Tabel 4.3	Hasil survei penerimaan sistem	89
Tabel 4.4	Analisis <i>UAT</i> P1	91
Tabel 4.5	Analisis <i>UAT</i> P2	91
Tabel 4.6	Analisis <i>UAT</i> P3	92
Tabel 4.7	Analisis <i>UAT</i> P4	92
Tabel 4.8	Analisis <i>UAT</i> P5	93
Tabel 4.9	Analisis <i>UAT</i> P6	93
Tabel 4.10	Analisis <i>UAT</i> P7	94
Tabel 4.11	Analisis <i>UAT</i> P8	95
Tabel 4.12	Analisis <i>UAT</i> P9	95
Tabel 4.13	Hasil Keseluruhan perhitungan <i>UAT</i>	96

DAFTAR LAMPIRAN

Lampiran	Judul	Halaman
Lampiran 1	Lembar Persetujuan Pembimbing Proposal	107
Lampiran 2	Lembar Pengesahan Ujian Proposal	108
Lampiran 3	Statistik Penggunaan API OpenAI	109
Lampiran 4	RedisInsight	110
Lampiran 5	Penyimpanan Riwayat Percakapan <i>Redis</i>	111
Lampiran 6	Status Indeks Vector pada Pinecone	112
Lampiran 7	Status Hosting Aplikasi pada Railway	113
Lampiran 8	Hasil Kuisioner UAT	114
Lampiran 9	Twilio	115
Lampiran 10	Data Primer	116
Lampiran 11	Data Sekunder	117

BAB I

PENDAHULUAN

A. Latar Belakang

Kemajuan teknologi yang pesat di era modern ini dapat memberikan efek yang sangat signifikan dalam berbagai aspek kehidupan masyarakat. Salah satu dampak positif yang paling terlihat adalah kemajuan dalam penyediaan layanan informasi yang lebih efisien. Teknologi informasi kini telah merambah ke berbagai bidang, termasuk di dalamnya adalah sektor jasa pelayanan informasi pemerintahan, yang sangat penting untuk meningkatkan transparansi dan aksesibilitas bagi publik. Saat ini, banyak sekali inovasi baru yang terus dikembangkan dengan tujuan untuk memberikan layanan informasi yang tidak hanya lebih mudah diakses, tetapi juga lebih cepat dan lebih akurat. Salah satu contohnya yang menonjol adalah aplikasi layanan informasi berbasis *chatbot* yang diterapkan dalam sistem pemerintahan, yang memungkinkan masyarakat untuk mendapatkan informasi yang mereka butuhkan dengan waktu respon yang singkat dan tanpa harus melalui prosedur yang rumit (Muhammad Hanif Muslim, 2023).

Namun, meskipun berbagai kemajuan ini telah dirasakan, masih terdapat tantangan dalam penyampaian

informasi yang cepat dan tepat sasaran, terutama di lingkungan pemerintahan. Salah satu alasan utama perlunya dikembangkan sistem *chatbot* di Dinas Kependudukan dan Pencatatan Sipil Kota Semarang adalah untuk menjawab kebutuhan masyarakat yang semakin tinggi akan informasi yang akurat, tersedia 24/7, dan mudah diakses. Saat ini, laman web sistem informasi di Dinas Kependudukan dan Pencatatan Sipil Kota Semarang masih menghadapi beberapa keterbatasan, seperti proses pencarian informasi yang seringkali memakan waktu karena struktur data yang kurang terorganisir, serta keterbatasan dalam interaksi langsung dengan masyarakat yang membutuhkan bantuan spesifik.

Dengan adanya *chatbot*, pelayanan informasi dapat menjadi jauh lebih efisien. *Chatbot* mampu menjawab pertanyaan yang sering diajukan secara otomatis, memberikan panduan mengenai prosedur administrasi, serta membantu mengurangi beban kerja pegawai di front office. *Chatbot* yang memanfaatkan teknologi *Large Language Models (LLM)* seperti *GPT* mampu memahami pertanyaan dengan bahasa natural, memberikan jawaban yang lebih relevan, dan meningkatkan pengalaman pengguna secara keseluruhan. Keunggulan inilah yang membuat *chatbot* menjadi salah

satu solusi penting dalam menjembatani kesenjangan antara pemerintah dan masyarakat.

Dinas Kependudukan dan Pencatatan Sipil Kota Semarang terus berupaya memperbaiki layanan publiknya. Namun, dengan semakin banyaknya jumlah penduduk dan kompleksitas kebutuhan informasi, sistem informasi yang ada sering kali belum mampu memenuhi ekspektasi masyarakat dalam hal kecepatan dan kemudahan akses. Oleh karena itu, integrasi *chatbot* berbasis *LLM* dengan dukungan *vector database* diharapkan dapat menjadi terobosan yang signifikan untuk meningkatkan interaksi antara pemerintah dan masyarakat, serta mempercepat akses informasi yang dibutuhkan oleh warga Kota Semarang.

Dalam konteks pengembangan sistem *chatbot* yang diusulkan, penting untuk memperhatikan bagaimana teknologi modern dapat digunakan secara optimal dalam pelayanan publik, sebagaimana dijelaskan dalam firman Allah SWT dalam Al-Qur'an, Surah Al-Hujurat ayat 13:

يَا أَيُّهَا النَّاسُ إِنَّا خَلَقْنَاكُمْ مِنْ ذَكَرٍ وَأُنْثَىٰ وَجَعَلْنَاكُمْ شُعُوبًا وَقَبَائِلَ لِتَعَارَفُوا إِنَّ أَكْرَمَكُمْ عِنْدَ اللَّهِ أَتْقَاهُ إِنَّ اللَّهَ عَلِيمٌ خَبِيرٌ

“Wahai manusia, sesungguhnya Kami telah menciptakan kamu dari seorang laki-laki dan perempuan. Kemudian, Kami menjadikan kamu berbangsa-bangsa dan bersuku-

suku agar kamu saling mengenal. Sesungguhnya yang paling mulia di antara kamu di sisi Allah adalah orang yang paling bertakwa. Sesungguhnya Allah Maha Mengetahui lagi Mahateliti.” (QS. Al-Hujurat [49]:13)

Ayat ini mengajarkan tentang pentingnya hubungan dan interaksi antar manusia, yang dalam konteks modern, termasuk interaksi antara pemerintah dan masyarakat. Teknologi yang dikembangkan seperti *chatbot* dapat membantu menciptakan interaksi yang lebih baik dan saling memahami melalui penyediaan informasi yang lebih mudah diakses dan responsif.

Dalam upaya mengembangkan *chatbot* yang mampu menjawab pertanyaan seputar , beberapa teknologi utama yang diterapkan adalah *Langchain*, *Large Language Models (LLM)* seperti *GPT-4o-mini*, serta *vector database* seperti *Pinecone*. *Langchain* adalah *framework* yang membantu pengintegrasian berbagai komponen sistem berbasis *LLM*, memungkinkan *chatbot* untuk memanfaatkan teknologi kecerdasan buatan dengan lebih fleksibel dan modular. Dengan *Langchain*, pengembang dapat dengan mudah merancang alur logika percakapan yang kompleks dan menghubungkannya dengan berbagai sumber data.

Large Language Models (LLM), termasuk *GPT-4o-mini*, merupakan model kecerdasan buatan yang dilatih untuk memahami dan menghasilkan teks dalam bahasa

alami. *GPT-4o-mini*, varian lebih ringan dari *GPT-4o*, dipilih karena kemampuannya untuk menangani konteks yang lebih besar dengan penggunaan sumber daya yang lebih efisien. Model ini ideal untuk sistem *chatbot* yang harus merespons beragam pertanyaan pengguna dengan cepat dan tepat, tanpa mengorbankan kualitas jawaban. Selain itu, model ini juga lebih hemat biaya dalam penggunaan *API*, sehingga cocok untuk proyek pemerintah yang skalanya luas namun tetap harus memperhatikan efisiensi.

Sistem ini juga menggunakan *vector database* seperti *Pinecone*. *Vector database* adalah teknologi penyimpanan data yang dioptimalkan untuk pencarian informasi berbasis format *vector* yang lebih baik dalam menangkap makna semantik dari teks. Ketika *chatbot* beroperasi, pertanyaan pengguna akan diproses menjadi representasi *vector*, yang kemudian dicocokkan dengan data yang telah disimpan di *Pinecone*. *Pinecone* dipilih karena kemampuannya yang unggul dalam melakukan pencarian dan pengambilan informasi yang cepat dari sejumlah besar data teks, serta skalabilitasnya yang mendukung implementasi dalam skala besar seperti Pemerintahan Kota Semarang.

Berdasarkan penjelasan yang telah disampaikan sebelumnya, dapat disimpulkan bahwa penerapan

teknologi *chatbot* berbasis *Large Language Models* dan *vector database* di Dinas Kependudukan dan Pencatatan Sipil Kota Semarang sangat relevan dan penting untuk meningkatkan efisiensi serta kualitas layanan informasi kepada masyarakat. Dengan dukungan teknologi seperti *Langchain*, *GPT-4o-mini*, dan *Pinecone*, sistem ini diharapkan mampu menjawab tantangan dalam memberikan layanan yang cepat, akurat, dan responsif. Oleh karena itu, penelitian ini bertujuan untuk merancang dan membangun sistem *chatbot* yang dapat memperbaiki interaksi antara pemerintah dan masyarakat, serta mendukung peningkatan transparansi dan aksesibilitas informasi di lingkungan pemerintahan.

B. Rumusan Masalah

1. Bagaimana merancang dan membangun sistem *chatbot* berbasis *LLM* dan *vector database* pada Dinas Kependudukan dan Pencatatan Sipil Kota Semarang?
2. Pengujian kinerja pada sistem *chatbot WhatsApp* berbasis *Large Language Model* untuk layanan informasi di Dinas Kependudukan dan Pencatatan Sipil Kota Semarang?

C. Tujuan dan Kegunaan Penelitian

Berdasarkan rumusan masalah dan latar belakang masalah yang disampaikan, tujuan penelitian ini adalah sebagai berikut :

1. Merancang dan membangun sistem *chatbot* berbasis *Large Language Models (LLM)* dan *vector database* pada layanan informasi di Dinas Kependudukan dan Pencatatan Sipil Kota Semarang
2. Menguji kinerja sistem *chatbot* dalam memberikan informasi kepada masyarakat Kota Semarang mengenai layanan kependudukan dan pencatatan sipil.

D. Batasan Penelitian

Untuk memastikan bahwa pembuatan aplikasi ini tetap terfokus pada tema yang akan diangkat, masalah yang akan dibahas akan dibatasi pada poin-poin berikut:

1. Penelitian ini berfokus pada pengembangan sistem *chatbot* untuk Dinas Kependudukan dan Pencatatan Sipil Kota Semarang, tidak mencakup instansi pemerintah lainnya di Kota Semarang.
2. Sistem *chatbot* yang dikembangkan terbatas pada platform *WhatsApp* sebagai antarmuka komunikasi dengan pengguna.
3. Informasi yang disediakan oleh *chatbot* terbatas pada layanan kependudukan dan pencatatan sipil yang

menjadi wewenang Dinas Kependudukan dan Pencatatan Sipil Kota Semarang.

4. Pengembangan *chatbot* menggunakan teknologi *Large Language Model GPT-4o-mini*, *framework Langchain*, dan *vector database Pinecone*.
5. Penelitian ini tidak mencakup aspek keamanan data pribadi dan hanya berfokus pada kemampuan *chatbot* dalam memberikan informasi umum terkait layanan kependudukan dan pencatatan sipil.

E. Manfaat Penelitian

1. Manfaat Teoritis

Dari sudut pandang teoritis, penelitian ini memberikan kontribusi penting pada pengembangan ilmu pengetahuan di bidang kecerdasan buatan, khususnya dalam konteks penerapan *Large Language Models (LLM)* dan *vector database* untuk sistem *chatbot* di sektor pemerintahan. Dengan mengintegrasikan teknologi *Langchain*, *GPT-4o-mini*, dan *Pinecone* dalam konteks pelayanan publik, khususnya di bidang kependudukan dan pencatatan sipil, penelitian ini memperkaya literatur akademis yang ada. Hal ini membuka jalan bagi penelitian lebih lanjut mengenai optimalisasi interaksi antara pemerintah dan masyarakat melalui teknologi

kecerdasan *buatan*, serta memperdalam pemahaman tentang efektivitas penggunaan *chatbot* berbasis *LLM* dalam meningkatkan kualitas layanan informasi pemerintah.

2. Manfaat Praktis

Penelitian ini memberikan manfaat praktis bagi berbagai pemangku kepentingan. Bagi Dinas Kependudukan dan Pencatatan Sipil Kota Semarang, sistem *chatbot* dapat meningkatkan efisiensi pelayanan dengan mengurangi beban kerja petugas dan menyediakan akses informasi 24/7 secara otomatis. Masyarakat juga mendapat kemudahan dalam mengakses informasi layanan kependudukan dengan cepat dan jelas. Bagi Dinas Kependudukan dan Pencatatan Sipil Kota Semarang, ini mendukung program *smart city* serta meningkatkan citra pemerintah yang responsif. Selain itu, penelitian ini memberikan wawasan berharga bagi pengembang teknologi dan menjadi referensi bagi penelitian lanjutan di bidang layanan publik berbasis kecerdasan *buatan*.

F. Sistematika Penelitian

Sistematika penelitian yang diterapkan dalam penelitian ini adalah sebagai berikut :

BAB I PENDAHULUAN

Bab ini berisi pendahuluan yang mencakup latar belakang, identifikasi masalah, rumusan masalah, tujuan penelitian, manfaat penelitian, serta sistematika penelitian.

BAB II KAJIAN TEORI DAN LANDASAN PUSTAKA

Bab ini menyajikan tinjauan terhadap penelitian-penelitian terdahulu serta penjelasan mengenai landasan teori yang relevan dengan penelitian ini.

BAB METODOLOGI PENELITIAN

Bab ini menjelaskan proses penelitian dan metode yang digunakan, termasuk diagram blok dan perencanaan penelitian.

BAB IV HASIL DAN PEMBAHASAN

Bab ini memaparkan hasil penelitian yang diperoleh dari Akurasi Jawaban serta analisis *UAT*

BAB V KESIMPULAN DAN SARAN

Bab ini menyajikan kesimpulan yang diambil dari hasil penelitian serta beberapa rekomendasi dari penulis untuk pengembangan di masa depan.

BAB II

TINJAUAN PUSTAKA

A. Kajian Teori

1. *Chatbot*

Chatbot merupakan sistem perangkat lunak yang dirancang untuk mensimulasikan percakapan manusia, biasanya melalui teks atau suara, dengan tujuan memberikan layanan atau informasi secara otomatis (Guntoro et al., 2020). Dalam konteks layanan publik, *chatbot* telah terbukti sangat efektif dalam meningkatkan aksesibilitas dan kecepatan layanan. Pemerintah dapat memanfaatkan *chatbot* untuk menyediakan informasi yang sering ditanyakan masyarakat secara *real-time*, mengurangi beban kerja manual pegawai, dan mempercepat proses layanan administratif. Misalnya, *chatbot* dapat digunakan untuk memandu masyarakat dalam mengurus dokumen seperti KTP, akta kelahiran, dan layanan kependudukan lainnya, tanpa perlu interaksi langsung dengan petugas.

Di pemerintahan, penggunaan *chatbot* memungkinkan akses informasi 24/7, yang menjadi solusi bagi keterbatasan jam kerja pegawai. Selain itu, *chatbot* juga membantu mengurangi antrian dan

keluhan masyarakat yang biasanya terkait dengan lambatnya respon dari layanan manual. Oleh karena itu, penerapan *chatbot* dalam layanan publik menjadi sebuah inovasi yang dapat meningkatkan efisiensi operasional pemerintah dan memberikan pengalaman yang lebih baik bagi masyarakat.

2. *Large Language Models (LLM)*

Large Language Models (LLM) adalah jenis kecerdasan buatan yang dilatih menggunakan dataset yang sangat besar untuk memahami, menganalisis, dan menghasilkan teks dalam bahasa alami (Alberts et al., 2023). Model ini tidak hanya dapat mengenali pola dalam teks tetapi juga mampu memberikan jawaban atau respon yang koheren dan relevan berdasarkan konteks percakapan. *LLM* bekerja dengan memprediksi kata-kata berikutnya dalam sebuah kalimat atau frasa berdasarkan konteks sebelumnya, sehingga memungkinkan percakapan yang lebih alami antara manusia dan sistem komputer.

Salah satu implementasi *LLM* yang paling menonjol adalah *GPT (Generative Pre-trained Transformer)*, model bahasa yang dirancang untuk mengolah teks dengan sangat presisi. Dalam proyek ini, *GPT-4o-mini* digunakan sebagai inti dari sistem

chatbot. *GPT-4o-mini* adalah versi yang lebih ringan dan hemat sumber daya dari *GPT-4*, tetapi tetap memiliki kemampuan yang kuat untuk memahami pertanyaan pengguna dan memberikan jawaban yang relevan (Yin et al., 2024). Model ini cocok untuk *chatbot* layanan publik karena dapat menangani berbagai macam pertanyaan dengan konteks yang kompleks, seperti prosedur administrasi atau layanan kependudukan.

3. *Langchain Framework*

Langchain adalah *framework* yang dirancang untuk memudahkan pengembangan aplikasi berbasis *LLM*. *Framework* ini memungkinkan pengembang untuk menghubungkan berbagai komponen dalam sistem *LLM*, seperti logika percakapan, pengambilan data dari berbagai sumber, dan integrasi dengan teknologi lainnya (Herwanza et al., 2024). Dalam sistem *chatbot* Dinas Kependudukan dan Pencatatan Sipil Kota Semarang, *Langchain* digunakan untuk merancang alur percakapan yang fleksibel dan modular, yang mampu menangani berbagai macam skenario interaksi dengan pengguna.

Keunggulan *Langchain* adalah kemampuannya untuk mengelola logika percakapan yang kompleks

dan mengintegrasikannya dengan model *LLM* dan basis data yang ada. Ini sangat penting dalam pengembangan *chatbot* layanan publik yang harus mampu menjawab pertanyaan yang sangat bervariasi, mulai dari prosedur administratif hingga pertanyaan spesifik mengenai layanan kependudukan. Dengan menggunakan *Langchain*, *chatbot* dapat menawarkan pengalaman interaksi yang lebih responsif dan efisien, sekaligus memungkinkan pengembang untuk dengan mudah memperbarui atau mengembangkan fitur baru sesuai dengan kebutuhan.

4. *Vector Database*

Vector database adalah teknologi yang digunakan untuk menyimpan dan mengelola data dalam bentuk representasi *vector*, yang lebih efektif dalam menangkap makna semantik dari teks (Taipalus, 2024). Dalam sistem *chatbot* ini, *Pinecone* dipilih sebagai *vector database* karena kemampuannya dalam melakukan pencarian informasi berbasis *vector* secara cepat dan efisien. Ketika pengguna mengajukan pertanyaan, pertanyaan tersebut diubah menjadi representasi *vector* dan dicocokkan dengan data yang telah disimpan dalam bentuk *vector* di *Pinecone*.

Keunggulan *vector database* dibandingkan basis data tradisional terletak pada kemampuannya untuk memahami makna dari teks, bukan hanya berdasarkan pencocokan kata-kata. Ini memungkinkan *chatbot* untuk memberikan jawaban yang lebih relevan meskipun pengguna mengajukan pertanyaan dengan variasi bahasa yang berbeda. *Pinecone* dipilih karena skalabilitasnya yang mendukung implementasi dalam skala besar, seperti sistem informasi pemerintah, dan kemampuannya untuk menangani jumlah data yang besar dengan tetap menjaga performa yang tinggi.

5. FastAPI

FastAPI adalah *framework* web yang cepat dan modern untuk membangun *API* (Application Programming Interface) dengan *Python*. *FastAPI* dirancang dengan performa tinggi dan kemudahan penggunaan, sehingga sangat ideal untuk mengembangkan sistem yang membutuhkan respons cepat, seperti *chatbot*. Salah satu keunggulan *FastAPI* adalah kemampuannya untuk menangani *request asynchronous*, yang memungkinkan *chatbot* untuk memproses banyak permintaan pengguna secara bersamaan tanpa mengorbankan performa (Azhari, 2022).

Dalam konteks pengembangan *chatbot* untuk Pemerintahan Kota Semarang, *FastAPI* digunakan untuk mengelola *backend* dari sistem *chatbot*. *Backend* ini akan menangani logika bisnis, menghubungkan *chatbot* dengan *LLM (GPT-4o-mini)*, serta mengatur proses komunikasi antara berbagai komponen, seperti *vector database Pinecone* dan platform pengiriman pesan *WhatsApp*.

6. *WhatsApp*

WhatsApp adalah salah satu aplikasi perpesanan yang paling populer dan memiliki jangkauan luas, sehingga menjadi platform yang tepat untuk berinteraksi dengan masyarakat secara langsung. Dalam proyek ini, *WhatsApp* digunakan sebagai antarmuka komunikasi utama antara pengguna (masyarakat) dan *chatbot* (Hong et al., 2021). Dengan menggunakan *API* dari *Twilio*, sistem *chatbot* dapat mengirim dan menerima pesan melalui *WhatsApp*. *Twilio* menyediakan layanan *API* yang handal dan terukur, memungkinkan *chatbot* untuk merespon pertanyaan pengguna secara *real-time* di platform *WhatsApp*.

Twilio memudahkan integrasi antara sistem *chatbot* dan *WhatsApp*, memungkinkan *bot* untuk

menangani pesan masuk, mengirimkan respons otomatis, serta berinteraksi dengan pengguna dengan cara yang natural melalui percakapan berbasis teks (Parlika et al., 2020). Penggunaan *WhatsApp* sebagai platform utama memastikan bahwa masyarakat dapat dengan mudah mengakses layanan *chatbot* tanpa perlu mengunduh aplikasi tambahan atau melakukan pendaftaran yang rumit.

B. Kajian Penelitian yang Relevan

Pada tabel 2.1, akan dibahas hasil-hasil penelitian terdahulu yang relevan dengan topik penelitian ini, yaitu penerapan *Large Language Models (LLMs)*, *chatbot* berbasis *LLM*, dan penggunaan basis data *vector* dalam konteks layanan informasi. Penelitian-penelitian ini memberikan wawasan penting mengenai perkembangan teknologi terkini serta tantangan yang dihadapi dalam pengembangan sistem serupa, khususnya yang terkait dengan layanan publik dan sektor pemerintahan.

Tabel 2.1 Kajian Penelitian yang relevan

No .	Nama Penulis	Tahun	Judul Penelitian	Hasil Penelitian
1	Aji Bayu Permadi, Nazruddin Safaat H,	2024	<i>Implementasi Question Answering System Tafsir</i>	Hasil penelitian menunjukkan bahwa mereka berhasil

	Lestari Handayani, dan Yusra		<i>Al-Azhar Menggunakan Langchain dan Large Language Model Berbasis Chatbot Telegram</i>	mengimplementasi kan sistem question answering untuk tafsir Al-Azhar menggunakan <i>Langchain</i> dan <i>Large Language Model</i> berbasis <i>chatbot</i> Telegram. Pengujian sistem menggunakan <i>User Acceptance Testing (UAT)</i> menghasilkan skor akurasi 83.71%, sedangkan pengujian dengan <i>framework</i> DeepEval menghasilkan skor hallucination 41%, contextual precision 90%, dan contextual relevancy 79%. Meskipun sistem berhasil menjawab sebagian besar pertanyaan, masih ada beberapa pertanyaan yang tidak bisa dijawab meski konteksnya ada dalam tafsir Al-Azhar
2	Yuhao Dan, Zhikai Lei, dan Yiyang Gu	2023	EduChat: A Large-Scale Language Model-based Chatbot System for	Hasil penelitian menunjukkan bahwa EduChat, sebuah sistem <i>chatbot</i> berbasis model bahasa

			Intelligent Education	berskala besar untuk pendidikan cerdas, berhasil dikembangkan dengan kemampuan seperti jawaban pertanyaan terbuka berbasis pencarian, penilaian esai, pengajaran metode Socrates, dan dukungan emosional. EduChat menunjukkan performa yang baik pada <i>benchmark C-Eval</i>, terutama setelah diintegrasikan dengan kemampuan pencarian. Sistem ini juga dilengkapi antarmuka demo yang ramah pengguna dan tersedia sebagai proyek open-source untuk mendorong penelitian dan aplikasi model bahasa berskala besar dalam pendidikan cerdas.
3	Narendra Nadh Vema	2024	Role of <i>Vector Databases</i> in <i>Large Language Models (LLMs)</i>	Hasil penelitian menunjukkan bahwa penggunaan basis data <i>vector</i> meningkatkan efisiensi pencarian kesamaan dan

				analisis semantik dalam <i>Large Language Models (LLMs)</i>, yang mempercepat respons dan memperbaiki pengolahan data dalam model AI. Meskipun demikian, penelitian ini juga menemukan bahwa terdapat tantangan signifikan, seperti bottleneck kinerja dalam operasi <i>Input/Output (I/O)</i>, kompleksitas algoritma, dan kendala skalabilitas. Tantangan-tantangan ini dapat memengaruhi kemampuan <i>LLM</i> dalam menangani data besar. Oleh karena itu, penelitian ini merekomendasikan inovasi teknologi untuk mengatasi hambatan-hambatan tersebut, guna meningkatkan kinerja dan kepercayaan pengguna terhadap model AI berbasis basis data <i>vector</i>
--	--	--	--	---

4	Ibnu Pujiono, Irfan Murtadho Agtyaputra, dan Yova Ruldeviyan.	2024	IMPLEMENTING RETRIEVAL - AUGMENTED GENERATION AND VECTOR DATABASES FOR CHATBOTS IN PUBLIC SERVICES AGENCIES CONTEXT	Hasil penelitian menunjukkan bahwa model GPT-4 memiliki performa terbaik dalam menjawab pertanyaan terkait manajemen keuangan badan layanan umum berdasarkan Peraturan Menteri Keuangan Nomor 129 tahun 2020, dengan skor cosine similarity rata-rata 0,404. Penelitian ini menggunakan metode retrieval-augmented generation (RAG) dan <i>database vector</i> untuk mengembangkan <i>chatbot</i> yang dapat membantu penasihat keuangan dalam mengelola badan layanan umum. Meski demikian, masih terdapat beberapa tantangan seperti inkonsistensi antar model, biaya yang cukup mahal, dan kesiapan infrastruktur di lingkungan internal organisasi.
---	---	------	---	--

Berdasarkan Tabel 2.1, penelitian sebelumnya telah memberikan kontribusi penting mengenai pengembangan dan penerapan sistem berbasis *Large Language Models (LLMs)* dan *chatbot* dalam berbagai konteks, seperti pendidikan, layanan publik, dan tafsir agama. Permadi et al. (2024) berhasil mengimplementasikan sistem question answering untuk Tafsir Al-Azhar menggunakan *Langchain* dan *LLM* berbasis *chatbot* Telegram, dengan tingkat akurasi 83,71% dan beberapa tantangan terkait relevansi konteks. Di sisi lain, Dan et al. (2023) mengembangkan EduChat, sebuah *chatbot* berbasis *LLM* untuk pendidikan cerdas, yang mampu menangani penilaian esai, pengajaran metode Socrates, dan dukungan emosional, dengan performa tinggi pada *benchmark C-Eval*.

Penelitian Vema (2024) menyoroti peran penting basis data *vector* dalam meningkatkan efisiensi pencarian semantik di *LLMs*, meskipun ada tantangan pada *I/O* dan skalabilitas yang perlu diatasi. Terakhir, Pujiono et al. (2024) menerapkan metode *retrieval-augmented generation (RAG)* dan basis data *vector* dalam *chatbot* yang dirancang untuk badan layanan umum, menunjukkan bahwa model *GPT-4* memberikan

performa terbaik dalam manajemen keuangan, meskipun terdapat kendala biaya dan kesiapan infrastruktur.

Oleh karena itu, penelitian ini berfokus pada penerapan *LLM* dan *vector database* untuk meningkatkan interaksi serta layanan informasi pada sektor publik, khususnya di Dinas Kependudukan dan Pencatatan Sipil Kota Semarang. Dengan memanfaatkan teknologi seperti *LLM* dan *Pinecone* untuk pencarian berbasis konteks serta *FastAPI* untuk integrasi layanan, penelitian ini bertujuan untuk mengembangkan *chatbot* yang mampu memberikan layanan informasi yang responsif dan relevan kepada masyarakat, khususnya terkait data kependudukan dan pencatatan sipil. Evaluasi melalui *User Acceptance Testing (UAT)* dan akurasi jawaban diharapkan memberikan gambaran tentang efektivitas dan akurasi sistem ini, sekaligus mengidentifikasi area yang perlu diperbaiki.

BAB III

METODOLOGI PENELITIAN

A. Metode Pengumpulan Data

Metode pengumpulan data dalam penelitian ini dilakukan melalui dua pendekatan, yaitu data primer dan data sekunder. Kedua jenis data ini digunakan untuk memberikan gambaran yang komprehensif tentang kebutuhan layanan informasi di Dinas Kependudukan dan Pencatatan Sipil Kota Semarang serta untuk mendukung pengembangan sistem *chatbot* yang relevan

1. Data Primer

Data primer diperoleh melalui observasi langsung di kantor Dinas Kependudukan dan Pencatatan Sipil Kota Semarang serta wawancara dengan petugas yang bertanggung jawab dalam memberikan layanan informasi kepada masyarakat. Observasi dilakukan untuk memahami alur pelayanan dan jenis informasi yang sering diminta oleh masyarakat, sedangkan wawancara dengan petugas bertujuan untuk mengidentifikasi kendala-kendala dalam proses pelayanan, serta jenis pertanyaan yang paling sering diajukan oleh masyarakat. Data ini membantu dalam merancang fitur *chatbot* yang dapat menangani

permintaan informasi secara efisien dan sesuai dengan kebutuhan warga.

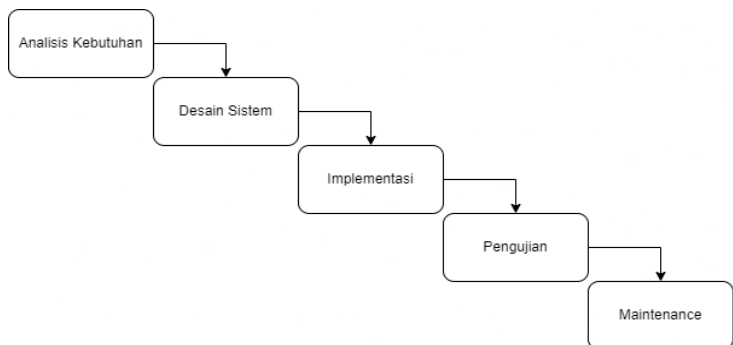
2. Data Sekunder

Data sekunder dikumpulkan dari sumber-sumber yang sudah ada, termasuk website resmi Dinas Kependudukan dan Pencatatan Sipil Kota Semarang. Data ini mencakup informasi tentang prosedur pelayanan yang tersedia secara daring, seperti pembuatan KTP, Kartu Keluarga, dan Akta Kelahiran, serta informasi kontak dan petunjuk layanan yang dapat diakses oleh masyarakat. Data sekunder ini berguna untuk memahami bagaimana layanan digital saat ini disediakan dan bagaimana *chatbot* dapat diintegrasikan dengan sistem informasi yang sudah ada untuk meningkatkan kualitas layanan.

B. Metode Pengembangan Aplikasi

Pengembangan sistem *chatbot* yang dirancang khusus untuk Dinas Kependudukan dan Pencatatan Sipil Kota Semarang ini dilakukan dengan menggunakan pendekatan yang dikenal sebagai *Waterfall*. Metode *Waterfall* sendiri merupakan sebuah model pengembangan perangkat lunak yang memiliki struktur yang jelas dan berurutan, di mana setiap tahap dalam proses pengembangan harus diselesaikan dengan tuntas sebelum tim pengembang dapat

melanjutkan ke tahap berikutnya(Kusumawaty & Kurniawan, 2024). Pendekatan ini sangat sesuai diterapkan pada proyek-proyek di mana persyaratan dan kebutuhan sistem dapat diidentifikasi dengan jelas dan mendetail sejak awal, sehingga memungkinkan perencanaan yang lebih baik dan pengelolaan risiko yang lebih efektif. Dalam konteks ini, berikut adalah tahapan-tahapan detail yang menjadi bagian dari model *Waterfall* yang akan diikuti dalam pengembangan sistem *chatbot* tersebut:



Gambar 3. 1 Metode Waterfall

1. Analisis Kebutuhan

a. Identifikasi Pengguna dan Stackholder

1. Pengguna utama sistem ini adalah masyarakat umum yang membutuhkan informasi mengenai layanan administrasi kependudukan, seperti pembuatan KTP,

akta kelahiran, kartu keluarga, dan sebagainya.

2. Pihak lain yang terlibat adalah pegawai di Dinas Kependudukan dan Pencatatan Sipil yang akan memantau dan menggunakan sistem ini.

b. Kebutuhan Sistem

1. Sistem harus mampu menerima dan merespons pertanyaan umum dari masyarakat dengan menggunakan *chatbot* berbasis *Large Language Model (LLM)*.
2. Sistem perlu mendukung integrasi dengan aplikasi *WhatsApp* melalui *Twilio API*, memungkinkan pengguna berinteraksi secara langsung melalui aplikasi ini.
3. Sistem harus menyediakan layanan *real-time* dengan respons yang relevan dan cepat, menggunakan penyimpanan *vector database (Pinecone)* untuk mempermudah pencarian data berbasis teks.
4. Sistem harus dapat menangani berbagai jenis pertanyaan yang berhubungan dengan layanan kependudukan dan pencatatan sipil serta menyediakan jawaban yang sesuai berdasarkan data yang disediakan.

5. Keamanan data juga menjadi prioritas utama, terutama karena sistem ini menangani data pribadi pengguna.

c. Lingkungan Pengembangan

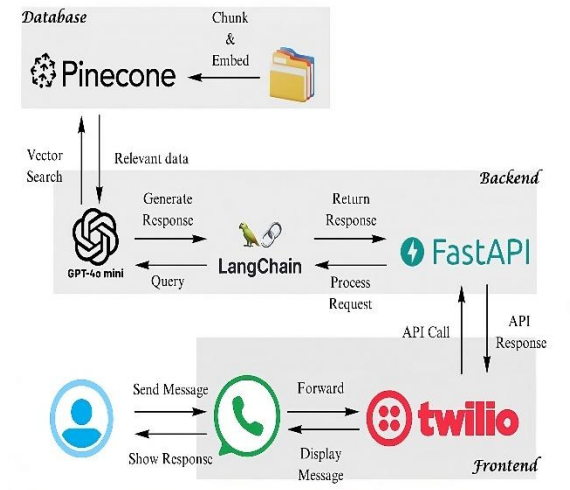
Teknologi utama yang digunakan adalah *Langchain* untuk mengelola aliran percakapan, *GPT-4o-mini* sebagai language model, *Pinecone* untuk penyimpanan embedding, dan *FastAPI* untuk mengembangkan *API* sistem. Sistem akan di-deploy di platform Railway untuk mendukung skalabilitas

2. Desain Sistem

Setelah semua kebutuhan sistem telah terdefinisi dengan jelas dan dipahami secara mendalam oleh seluruh tim pengembang, tahap selanjutnya yang harus dilakukan dalam proses pengembangan sistem adalah tahap desain sistem (system design). Pada tahap desain ini, berbagai elemen penting dan kritis akan dirancang secara rinci dan sistematis. Ini mencakup perancangan komponen-komponen sistem yang akan berfungsi secara harmonis, pengembangan arsitektur perangkat lunak yang efisien dan scalable, serta penyusunan struktur data yang tepat dan optimal, yang semuanya

bertujuan untuk mendukung dan meningkatkan kinerja keseluruhan sistem yang akan dibangun.

a. Arsitektur



Gambar 3. 2 Arsitektur Sistem

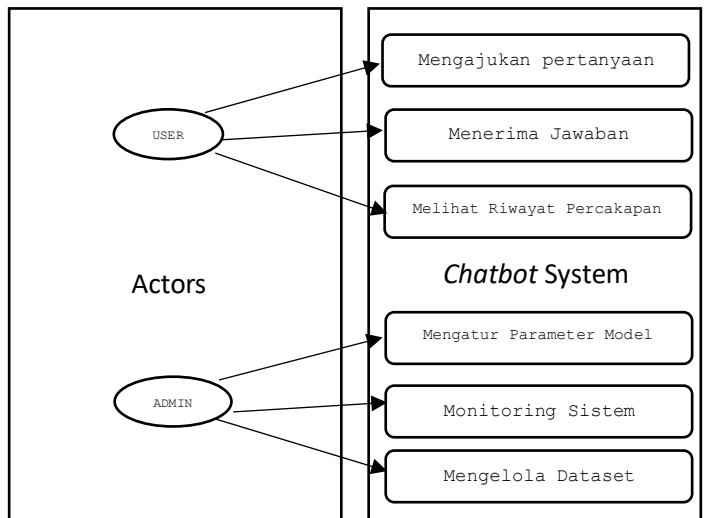
Arsitektur sistem untuk *chatbot* ini dirancang untuk meningkatkan interaksi dan layanan informasi di Dinas Kependudukan dan Pencatatan Sipil Kota Semarang. Sistem ini terdiri dari beberapa komponen utama yang saling terintegrasi, yaitu:

1. Frontend: Pengguna mengirim pesan melalui aplikasi *WhatsApp*, yang dikelola oleh *Twilio*. Pesan ini diteruskan ke *backend* untuk diproses lebih lanjut.

2. *Backend*: Menggunakan *FastAPI* sebagai *framework* untuk menangani permintaan dari frontend. Setelah menerima pesan, *FastAPI* akan mengirimkan query ke *LangChain* untuk menghasilkan respons yang relevan. *LangChain* berfungsi sebagai penghubung antara model *GPT-4o-mini* dan *database vector*.
3. *Database*: *Pinecone* digunakan sebagai penyimpanan data *vector* yang memungkinkan pencarian informasi yang relevan secara efisien. Data yang ada di dalam *Pinecone* akan di-chunk dan di-embed untuk memudahkan akses dan pencarian.
4. *Proses Respons*: Setelah data yang relevan ditemukan, *LangChain* akan menggenerasi respons menggunakan *GPT-4o-mini*. Respons ini kemudian dikembalikan ke *FastAPI*, yang selanjutnya mengirimkan kembali pesan tersebut ke pengguna melalui *Twilio*.

b. Use Case Diagram

Use Case Diagram memodelkan interaksi antara pengguna dan sistem *chatbot*, menggambarkan aktor serta fungsi utama yang menyediakan layanan informasi dari Dinas Kependudukan dan Pencatatan Sipil Kota Semarang.



Gambar 3. 3 Use Case

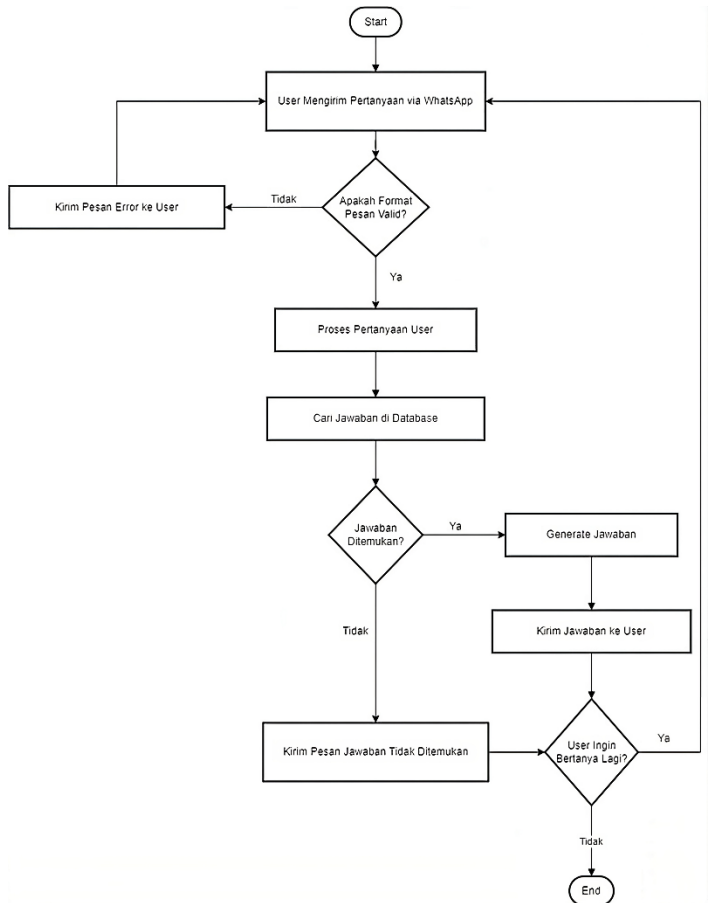
Use Case Diagram menggambarkan interaksi antara aktor dan sistem. Dalam diagram ini, terdapat dua aktor utama, yaitu User dan Admin. User memiliki kemampuan untuk mengajukan pertanyaan, menerima jawaban, dan melihat riwayat percakapan dengan sistem

chatbot. Di sisi lain, Admin memiliki peran penting dalam mengatur parameter model, memonitoring sistem, dan mengelola dataset yang digunakan. Diagram ini memberikan gambaran yang jelas mengenai fungsi-fungsi utama yang ada dalam sistem, sehingga memudahkan pengembangan dan implementasi sistem *chatbot*.

c. Flowchart Alur Interaksi Pengguna

Flowchart merupakan representasi diagram yang menggambarkan urutan operasi yang dilakukan untuk mendapatkan solusi masalah (Yuniarti, 2019).

Dalam konteks pengembangan sistem *chatbot* ini, flowchart berfungsi sebagai alat yang sangat berguna untuk memvisualisasikan alur interaksi antara pengguna dan sistem secara keseluruhan. Proses ini mencakup berbagai tahapan, mulai dari pengiriman pertanyaan oleh pengguna, melalui pemrosesan informasi oleh sistem, hingga penyajian jawaban yang relevan dan tepat sasaran kepada pengguna. Dengan demikian, flowchart membantu dalam memahami dan mengorganisir setiap langkah yang terlibat dalam interaksi tersebut.



Gambar 3. 4 Flowchart Alur Interaksi Pengguna

Berdasarkan gambar 3.3 tersebut menggambarkan alur interaksi pengguna dengan sistem *chatbot* yang diimplementasikan melalui platform *WhatsApp*. Proses dimulai ketika pengguna mengirimkan pertanyaan. Sistem

kemudian memeriksa apakah format pesan yang diterima valid. Jika tidak valid, sistem akan mengirimkan pesan kesalahan kepada pengguna.

Apabila format pesan valid, pertanyaan pengguna akan diproses, dan sistem akan mencari jawaban dalam *database*. Jika jawaban ditemukan, sistem akan menghasilkan jawaban dan mengirimkannya kembali ke pengguna. Namun, jika jawaban tidak ditemukan, sistem akan mengirimkan pesan yang menyatakan bahwa jawaban tidak dapat ditemukan.

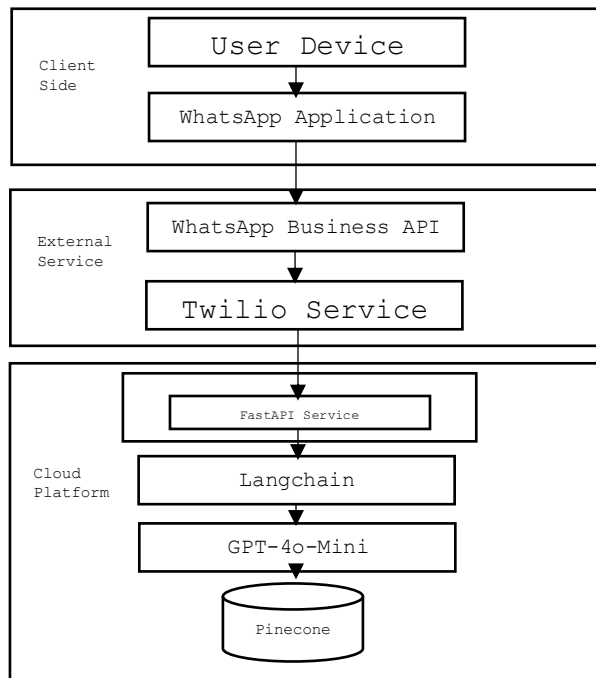
Selain itu, sistem juga memberikan opsi kepada pengguna untuk mengajukan pertanyaan lain. Proses ini diulang hingga pengguna selesai berinteraksi dengan sistem, yang ditandai dengan akhir sesi interaksi.

3. Implementasi

Sistem *chatbot* ini dirancang untuk memudahkan masyarakat dalam mengakses informasi terkait layanan Dinas Kependudukan dan Pencatatan Sipil Kota Semarang. Sistem ini menggunakan *Large Language Models* (*GPT-4o-mini*) yang diintegrasikan dengan basis data *vector* (*Pinecone*) untuk memahami

dan menjawab pertanyaan dari pengguna secara relevan dan akurat.

Pengguna dapat mengakses sistem ini melalui aplikasi *WhatsApp*, yang merupakan salah satu platform pesan yang mudah dijangkau oleh masyarakat luas. Dengan bantuan *WhatsApp Business API*, sistem dapat menangani permintaan pengguna secara *real-time* dan mengirimkan respons yang sesuai. Layanan *Twilio* digunakan sebagai antarmuka untuk menghubungkan pesan dari *WhatsApp* ke sistem yang ada di sisi *server (cloud)*.



Gambar 3.5 Implementasi

berdasarkan desain yang telah ditentukan. Pengguna mengakses aplikasi *WhatsApp* melalui perangkat mereka, yang kemudian terhubung dengan *API Bisnis WhatsApp*. Antarmuka ini berkomunikasi dengan layanan *Twilio* untuk mengelola permintaan. Di sisi cloud, layanan *FastAPI* berfungsi sebagai lapisan *API* yang memproses data menggunakan *Langchain* dan mengajukan pertanyaan kepada model *GPT-4o-mini*. Hasilnya kemudian dicari dalam *Pinecone Vector DB* untuk memberikan respons yang relevan kepada pengguna.

4. Pengujian

a. Akurasi *Chatbot*

Pengujian akurasi *chatbot* bertujuan untuk mengevaluasi apakah sistem *chatbot* memiliki tingkat akurasi yang cukup untuk digunakan. Pengujian ini dilakukan dengan memastikan apakah jawaban yang diberikan oleh *chatbot* sesuai dengan ekspektasi pengguna saat mereka mengajukan pertanyaan. Perhitungan dalam pengujian ini mengikuti rumus berikut. (Hikmah et al., 2023)

$$\text{Nilai Akurasi} = \frac{\text{Jumlah Jawaban Benar}}{\text{Jumlah Pertanyaan}}$$

Pada pengujian ini, setiap pertanyaan yang diajukan oleh pengguna akan dievaluasi apakah *chatbot* memberikan jawaban yang benar atau salah. Jika jawaban *chatbot* sesuai dengan ekspektasi atau informasi yang diinginkan, maka dianggap benar; sebaliknya, jika tidak sesuai, dianggap salah. Dengan cara ini, pengujian ini bertujuan untuk mengukur sejauh mana *chatbot* dapat memberikan informasi yang akurat dan relevan kepada pengguna.

b. *User Acceptance Test*

User Acceptance Testing (UAT) adalah tahap penting dalam proses pengembangan perangkat lunak yang bertujuan untuk memastikan bahwa sistem yang dibangun dapat memenuhi kebutuhan dan harapan pengguna akhir (Suabdinegara et al., 2021). *UAT* memberikan kesempatan bagi pengguna untuk menilai apakah sistem yang telah dikembangkan berfungsi dengan baik dan sesuai dengan kebutuhan mereka. Dalam penelitian ini, fokus *UAT* diarahkan pada pengujian *chatbot* yang digunakan oleh Dinas Kependudukan dan Pencatatan Sipil Kota Semarang. Menurut Nielsen Norman Group

(2000), pengujian usability pada aplikasi atau sistem baru tidak memerlukan jumlah responden yang besar. Dengan melibatkan 5 responden. Pendekatan ini sering digunakan dalam pengembangan perangkat lunak karena efektif untuk mengukur kemudahan penggunaan dan memvalidasi fungsi sistem tanpa memerlukan sampel yang terlalu besar.

Pengguna akan diminta untuk memberikan penilaian terhadap setiap pertanyaan menggunakan skala *Mean Option Score* (MOS). Metode ini adalah pendekatan kuantitatif yang populer dalam pengujian perangkat lunak, karena memungkinkan pengukuran tingkat kepuasan pengguna secara objektif. Melalui proses ini, diharapkan dapat diidentifikasi area yang perlu diperbaiki serta aspek-aspek yang sudah berfungsi dengan baik.

Mean Option Score (MOS) adalah metode evaluasi yang digunakan untuk mengukur kepuasan pengguna dalam menggunakan *chatbot*. Setiap pengguna diminta memberikan penilaian pada skala 1 hingga 5, di mana setiap angka mewakili tingkat persetujuan terhadap pernyataan

yang diberikan. Interpretasi skala ini adalah sebagai berikut:

Tabel 3.1 Penilaian Mean Option Score

MOS	Skor	Interpretasi
TT	1	Tidak Tahu
TS	2	Tidak Setuju
KS	3	Kurang Setuju
S	4	Setuju
SS	5	Sangat Setuju

Berdasarkan tabel 3.1, setelah pengguna memberikan penilaian, nilai rata-rata dari semua responden untuk setiap pertanyaan akan dihitung. Rata-rata ini berfungsi sebagai dasar untuk mengevaluasi kinerja *chatbot*, memberikan gambaran tentang seberapa baik *chatbot* memenuhi harapan pengguna.

Untuk mengukur tingkat penerimaan pengguna terhadap *chatbot*, digunakan rumus persentase kelayakan yang sederhana namun efektif(Pasaribu et al., 2019):

$$\begin{aligned}
 & \text{Persentase Kelayakan} \\
 &= \left(\frac{\text{Total Skor yang diperoleh}}{\text{Total Skor Maksimal}} \right) \times 100\%
 \end{aligned}$$

Rumus tersebut memungkinkan perhitungan seberapa baik *chatbot* diterima oleh pengguna berdasarkan skor yang diperoleh dari penilaian mereka. Interpretasi dari hasil pengukuran ini dapat dilihat pada tabel 3.2 :

Tabel 3.2 Kriteria Penilaian Skor

Persentase	Kriteria
20,00% - 36,00%	Sangat Tidak Layak
36,01% - 52,00%	Tidak Layak
52,01% - 68,00%	Cukup
68,01% - 84,00%	Layak
84,01% - 100%	Sangat Layak

Berdasarkan tabel 3.2, dengan menggunakan kriteria ini, dapat diidentifikasi apakah *chatbot* sudah memenuhi standar yang diharapkan oleh pengguna atau perlu dilakukan perbaikan lebih lanjut.

Berikut adalah daftar pertanyaan yang akan diajukan kepada pengguna, diorganisir berdasarkan aspek-aspek yang diujikan. Setiap pertanyaan dirancang untuk menggali umpan balik yang spesifik dari pengguna mengenai berbagai dimensi pengalaman mereka saat menggunakan

chatbot. Pada Tabel 3.3, ditunjukkan daftar pertanyaan yang akan diajukan kepada pengguna dalam bentuk kuisioner. Pertanyaan-pertanyaan ini diperoleh dari penelitian yang dilakukan oleh (Saputra, 2023).

Tabel 3.3 Daftar Pertanyaan UAT

Kode	Pertanyaan
Aspek <i>Functionality</i> (Fungsionalitas)	
P1	Apakah <i>Chatbot</i> memberikan informasi yang mudah dipahami sesuai dengan pertanyaan yang diajukan. ?
P2	Apakah <i>Chatbot</i> mampu memahami berbagai variasi pertanyaan dengan topik yang sama. ?
P3	Apakah <i>Chatbot</i> mampu memberikan respons yang relevan dengan konteks pertanyaan. ?
Aspek <i>Usability</i> (Kegunaan)	
P4	Apakah <i>chatbot</i> ini dapat membantu anda dalam mencari informasi mengenai layanan Dinas Kependudukan dan Pencatatan Sipil Kota Semarang ?
P5	Apakah <i>Chatbot</i> memberikan respons dengan format yang mudah dibaca. ?
Aspek <i>Efficiency</i> (Efisiensi)	

P6	Apakah anda puas dengan kecepatan respon yang diberikan oleh <i>chatbot</i> ini?
P7	Apakah menggunakan <i>chatbot</i> lebih menghemat waktu untuk mencari informasi layanan dibandingkan dengan mengunjungi situs web dan bertanya kepada admin?
Aspek <i>Reliability</i> (Keandalan)	
P8	Apakah <i>Chatbot</i> dapat diakses setiap saat ?
P9	Apakah informasi yang diberikan <i>chatbot</i> ini akurat?

5. Maintenance

Tahap setelah pengujian adalah pemeliharaan (maintenance), yang bertujuan memastikan sistem berfungsi dengan baik dan dapat diperbarui sesuai kebutuhan pengguna. Kegiatan dalam tahap ini meliputi perbaikan bug yang muncul setelah aplikasi digunakan secara luas, pembaruan sistem untuk meningkatkan akurasi dan relevansi, serta optimasi kinerja berdasarkan analisis performa. Selain itu, pengelolaan skalabilitas juga penting, di mana penambahan sumber daya di *platform* seperti *Railway* memungkinkan aplikasi menangani lebih banyak permintaan pengguna tanpa mengurangi performa.

BAB IV

HASIL DAN PEMBAHASAN

A. Sumber Data

Dalam penelitian ini, data primer diperoleh melalui wawancara dan observasi di Dinas Kependudukan dan Pencatatan Sipil Kota Semarang. Wawancara dilakukan dengan SOFWAN, S.E., M.Si., selaku Subkoordinator Identitas Penduduk pada tanggal 25 November 2024, di mana narasumber mengungkapkan bahwa masyarakat sering mengalami kesulitan dalam memperoleh informasi terkait layanan kependudukan, seperti pendaftaran akta dan KTP. Tantangan yang dihadapi antara lain adalah waktu tunggu dan antrean yang panjang, sehingga muncul harapan akan adanya solusi yang lebih efisien, seperti penggunaan *chatbot* yang dapat memberikan informasi secara akurat dan mudah diakses kapan saja.

Melalui observasi, terlihat bahwa proses pelayanan masih bergantung pada interaksi langsung, yang memperpanjang antrean. Observasi ini menunjukkan bahwa belum terdapat sistem otomatis yang memungkinkan masyarakat memperoleh informasi secara mandiri. Selain itu, data sekunder diperoleh dari situs resmi Dinas Kependudukan dan Pencatatan Sipil Kota

Semarang, yang mencakup berbagai prosedur pengajuan dokumen dan layanan daring yang tersedia. Informasi ini meliputi prosedur pengajuan KTP, Kartu Keluarga, dan Akta Kelahiran, serta layanan daring seperti pengajuan dokumen secara online dan akses informasi status pengajuan. Data sekunder ini juga memberikan informasi kontak dan lokasi kantor dinas, yang berguna dalam memahami struktur serta layanan yang tersedia.

Data primer dan data sekunder yang digunakan dalam penelitian ini dapat dilihat pada lampiran 10 untuk data primer dan lampiran 11 untuk data sekunder. Selain itu, informasi lebih lanjut dapat diakses melalui tautan resmi Dinas Kependudukan dan Pencatatan Sipil Kota Semarang: <https://dispendukcapil.semarangkota.go.id>.

B. Hasil Implementasi

Pada bagian ini, akan dijelaskan secara rinci mengenai implementasi sistem *chatbot WhatsApp* yang telah dikembangkan sebagai bagian dari penelitian ini. Sistem ini dirancang untuk menerima dan memproses pesan yang dikirimkan oleh pengguna melalui platform *WhatsApp*, kemudian mengolah pertanyaan tersebut dengan memanfaatkan berbagai layanan eksternal yang relevan, seperti pengolahan bahasa alami, basis data *vector*, dan layanan *API* pihak ketiga. Seluruh proses ini bertujuan untuk menghasilkan respons yang kontekstual dan sesuai

dengan kebutuhan pengguna. Penjelasan ini mencakup uraian mendalam mengenai setiap komponen yang membangun sistem ini, termasuk implementasi teknologi dan fungsi utama dalam kode. Penjelasan ini juga menguraikan bagaimana setiap bagian dari sistem saling terintegrasi untuk mencapai tujuan utama, yaitu memberikan layanan informasi yang optimal dan efisien kepada pengguna

1. Implementasi Teknologi

Dalam implementasi sistem ini, berbagai teknologi diterapkan untuk memastikan performa *chatbot* dapat berjalan dengan optimal. Berikut adalah penjelasan mengenai spesifikasi dan alat teknologi yang berkontribusi dalam mendukung kinerja sistem secara keseluruhan:

Tabel 4.1 Implementasi Teknologi dalam Sistem Chatbot

Teknologi	Spesifikasi	Deskripsi
<i>GPT-4o-Mini</i>	• <i>200.000 Token Per Minute</i> • <i>500 Request Per Minute</i> • <i>10.000 Request Per Day</i>	Model bahasa besar yang digunakan untuk menghasilkan jawaban atau melakukan analisis berbasis teks.

	• <i>2.000.000 Token Per Day</i>	
Voyage-3	• <i>Context Length (tokens) : 32000</i> • <i>Embedding Dimension : 1024</i> • <i>8.000.000 Tokens Per Minute</i> • <i>2000 Request per Minute</i>	Model untuk menghasilkan representasi embedding dari teks guna mendukung pencarian atau analitik.
Railway	• <i>8 GB RAM / 8 vCPU per layanan</i> • <i>Wilayah: US-West</i>	Platform deployment cloud dengan konfigurasi sederhana dan dukungan pengembang untuk layanan <i>backend</i> .
Redis	• <i>Max Commands Per Second : 1000</i> • <i>Max Concurrent Connections : 1000</i> • <i>Max Data Size : 10 GB</i>	Penyimpanan data in-memory yang cepat dan digunakan untuk caching, pub/sub, dan <i>database real-time</i> .

	• <i>Monthly Bandwith Limit : 200 GB</i>	
<i>Pinecone</i>	• <i>Storage : 2GB</i> • <i>Write Units : 2M/mo.include</i> • <i>Read Units : 1M/mo.include</i> • <i>Indexes Per Project : 5</i> • <i>Namespaces Per Indeks : 100</i>	Layanan untuk manajemen indeks dan pencarian berbasis <i>vector</i> dalam aplikasi berbasis AI.

Analisis ini menunjukkan bahwa spesifikasi alat dan teknologi yang digunakan saat ini sudah memadai untuk memenuhi kebutuhan sistem. Namun, peningkatan lebih lanjut dapat dilakukan guna mengakomodasi potensi peningkatan skala pengguna di masa mendatang.

2. Integrasi Pinecone *Vector Database*

Pada bagian ini, dibahas secara rinci proses integrasi *Pinecone Vector Database* sebagai salah satu komponen utama dalam sistem *chatbot*. *Pinecone* digunakan untuk mengelola dan mencari *vector* dengan efisien, memungkinkan sistem memberikan

respons yang relevan berdasarkan pencocokan semantik. Integrasi ini dirancang untuk memastikan pengolahan data yang cepat dan akurat, terutama dalam menangani basis data yang besar dan kompleks.

a. Membaca Dokumen PDF

Pada tahap awal dari proses ini, kami melakukan pengolahan terhadap file PDF yang berisi data penting dari Disdukcapil Kota Semarang. File PDF tersebut diolah dengan menggunakan sebuah program khusus yang telah dirancang secara khusus untuk membaca dan mengekstrak isi teks dari setiap halaman dalam file PDF. Proses pengolahan ini dilakukan melalui sebuah fungsi yang bernama `extract_text_from_pdf`. Fungsi ini bertugas untuk mengambil seluruh teks yang terdapat di dalam file PDF dan kemudian menyimpannya dalam bentuk daftar (list). Dalam daftar tersebut, setiap elemen mewakili teks dari masing-masing halaman dalam file PDF. Untuk memberikan gambaran yang lebih jelas, berikut adalah cuplikan kode yang digunakan untuk mendukung proses pembacaan dan ekstraksi teks dari file PDF::

```

import pdfplumber
def extract_text_from_pdf(pdf_path):
    with pdfplumber.open(pdf_path) as pdf:
        text_data = [page.extract_text() for
page in pdf.pages]
    return text_data
pdf_path = "Dataset Disdukcapil Kota
Semarang.pdf"
text_data = extract_text_from_pdf(pdf_path)

```

b. Memecah Dokumen Menjadi Potongan-potongan

Setelah seluruh teks dari file PDF berhasil diekstrak dengan sukses, langkah selanjutnya yang harus dilakukan adalah memecah teks tersebut menjadi bagian-bagian kecil yang disebut chunk. Proses pemecahan ini dilakukan dengan tujuan untuk mempermudah pengolahan serta analisis oleh sistem yang akan menggunakan teks tersebut. Setiap potongan teks yang dihasilkan memiliki panjang maksimal 800 karakter, dan di antara potongan-potongan tersebut terdapat tumpang tindih (overlap) sepanjang 100 karakter. Tumpang tindih ini sangat penting karena bertujuan untuk memastikan bahwa informasi yang saling berdekatan tetap terhubung dan tidak terputus antar potongan teks yang telah dibagi. Dengan cara ini, dapat menjaga konteks serta

makna dari informasi yang ada di dalam teks. Pemecahan teks ini dilakukan secara sistematis menggunakan fungsi yang disebut `chunk_text_with_overlap`, seperti yang ditunjukkan pada cuplikan kode berikut:

```
def chunk_text_with_overlap(text,
    chunk_size=800, overlap=100):
    chunks = []
    start = 0
    text_length = len(text)
    while start < text_length:
        end = min(start + chunk_size,
            text_length)
        chunks.append(text[start:end])
        start += chunk_size - overlap
    return chunks

chunks = []
for page_text in text_data:
    if page_text:

        chunks.extend(chunk_text_with_overlap(page_text,
            chunk_size=900, overlap=100))
```

c. Inisialisasi Pinecone



Gambar 4.3 Detail Konfigurasi Pinecone

Setelah membagi dokumen menjadi potongan-potongan kecil, langkah selanjutnya adalah menyiapkan alat bernama *Pinecone*. *Pinecone* adalah tempat penyimpanan data yang membantu memudahkan pencarian dan pengolahan informasi. Dari gambar di atas, peneliti bisa lihat bahwa *Pinecone* menggunakan cara tertentu untuk menentukan kesamaan antara data dengan ukuran 1024 dan berada di alamat tertentu di internet. *Pinecone* ini disimpan di layanan cloud AWS di Amerika Serikat dengan cara yang tidak memerlukan pengelolaan *server* secara langsung. Menggunakan model khusus untuk mengubah teks menjadi bentuk yang lebih mudah diproses. Dengan menyiapkan *Pinecone* dengan pengaturan ini, peneliti dapat menyimpan dan mengelola potongan-potongan dokumen dengan lebih efisien, sehingga pencarian dan pengolahan data menjadi lebih mudah.

d. Menyimpan Dokumen Ke Pinecone

Representasi numerik yang dihasilkan dari teks, yang berbentuk *embedding*, disimpan ke dalam sebuah database yang berbasis vector dan dikenal dengan nama *Pinecone*. *Pinecone*

dirancang secara khusus dengan teknologi canggih untuk menangani penyimpanan dan pencarian data yang berbasis *vector*, sehingga dapat mengelola data dalam jumlah besar dengan efisien. Setiap *embedding* yang dihasilkan dari potongan teks tidak hanya disimpan, tetapi juga dilengkapi dengan metadata yang relevan, yaitu teks aslinya, yang memungkinkan pelacakan dan pengambilan informasi dengan mudah dan akurat. Proses ini memastikan bahwa data tetap terorganisir dan dapat diakses dengan cepat saat dibutuhkan. Selain itu, data ini kemudian diunggah ke dalam *Pinecone* menggunakan *namespace* tertentu, yang dalam konteks ini dinamakan pencatatan-sipil, untuk memfasilitasi pengelolaan dan identifikasi data secara lebih efektif. Berikut adalah hasil integrasinya:

1	ID	VALUES
	chunk-270	0.0130732199, -0.0295321643, -0.0306873936, -0.0528436117, 0.0286354255, -0.0240796134, -...
SCORE 0.0615	METADATA	
	text:	"3. Petugas menyerahkan tanda bukti pendaftaran dan/pengambilan kutipan akta kelahiran yang sudah ditandatangani penetapan

Gambar 4.4 Simpan Dokumen ke Pinecone

3. Pengaturan Awal dan Impor Modul

Pertama-tama, peneliti memulai proses pengembangan dengan mengimpor berbagai modul yang diperlukan untuk menjalankan sistem *chatbot* secara efektif. Modul-modul ini mencakup berbagai fungsi standar *Python* yang esensial, serta library yang dirancang khusus untuk manajemen file dan logging, yang sangat penting untuk mencatat aktivitas sistem dan memantau kinerjanya. Selain itu, peneliti juga mengimpor library yang diperlukan untuk berkomunikasi dengan layanan eksternal, seperti Redis untuk penyimpanan data, Twilio untuk pengiriman pesan, dan layanan lainnya yang mendukung fungsionalitas *chatbot*. Dengan melakukan impor modul-modul ini, peneliti memastikan bahwa semua komponen yang diperlukan untuk membangun sistem dapat berfungsi dengan baik. Berikut adalah potongan kode yang mengimpor modul-modul tersebut:

```
import os
import sys
import logging
from typing import List, Optional
import redis
from dotenv import load_dotenv
from fastapi import FastAPI, Request
```

```

from fastapi.responses import PlainTextResponse
from twilio.twiml.messaging_response import
MessagingResponse
from voyageai import Client as VoyageAIClient
from langchain.community.vectorstores import
RetrievalQA
from langchain_openai import ChatOpenAI
from pinecone import Pinecone

```

Modul *os* dan *sys* digunakan untuk mengakses variabel lingkungan dan melakukan operasi sistem, sementara *logging* digunakan untuk mencatat aktivitas sistem ke berbagai tujuan seperti konsol dan file log. *FastAPI* digunakan untuk membuat *API* yang akan menerima dan merespons pesan dari *WhatsApp*, sementara *twilio.rest.Client* dan *twilio.twiml.messaging_response.MessagingResponse* digunakan untuk berkomunikasi dengan layanan *Twilio*. Library *voyageai*, *langchain_community.vectorstores*, *langchain.chains*, dan *langchain_openai* digunakan untuk berbagai fungsi seperti membuat embedding teks, mengatur *vector store*, dan memproses pertanyaan dengan model bahasa *OpenAI*.

4. Pengaturan Lingkungan dan Logging

Setelah mengimpor modul-modul yang diperlukan, peneliti memuat variabel lingkungan dari file *.env* menggunakan fungsi *load_dotenv()*. File *.env* ini berisi kredensial dan konfigurasi yang diperlukan untuk berkomunikasi dengan layanan eksternal seperti *Redis*, *Twilio*, dan lainnya. Kemudian, peneliti mengatur konfigurasi logging untuk mencatat aktivitas sistem ke konsol dan file *log*. Ini membantu peneliti memantau dan menyelesaikan masalah yang mungkin terjadi dengan lebih efektif.

Potongan kode untuk mengatur logging :

```
# Memuat variabel lingkungan
load_dotenv()
# Konfigurasi logging yang lebih komprehensif
logging.basicConfig(
    level=logging.INFO,
    format="%(asctime)s - %(name)s - %(levelname)s
- %(message)s",
    handlers=[
        logging.StreamHandler(sys.stdout),
        logging.FileHandler('chatbot.log')
    ]
)
logger = logging.getLogger(__name__)
```

Dalam konfigurasi *logging*, peneliti menetapkan tingkat logging menjadi *INFO*, yang berarti semua

pesan dengan tingkat *logging INFO* dan yang lebih tinggi akan dicatat. Format pesan *logging* mencakup waktu, nama logger, tingkat logging, dan pesan logging. Pesan logging kemudian dicatat ke dua tempat: konsol (*StreamHandler*) dan file *chatbot.log* (*FileHandler*). Ini memungkinkan peneliti untuk melacak aktivitas sistem secara *real-time* melalui konsol dan juga memiliki catatan permanen di *file log*.

5. Validasi Variabel Lingkungan

Metode _validate_env_vars() berfungsi sebagai mekanisme pemeriksaan untuk memastikan bahwa seluruh variabel lingkungan (*environment variables*) yang dibutuhkan oleh sistem telah disetel dengan benar dan lengkap sebelum sistem dijalankan. Pemeriksaan ini sangat krusial karena variabel-variabel lingkungan tersebut menyimpan informasi sensitif dan penting seperti kredensial, token akses, serta konfigurasi lainnya yang dibutuhkan agar sistem dapat terhubung dan beroperasi dengan layanan eksternal, seperti *Redis*, *Twilio*, dan berbagai layanan pihak ketiga lainnya. Dengan adanya validasi ini, sistem dapat meminimalisir risiko error atau kegagalan komunikasi akibat variabel yang hilang atau salah konfigurasi.

Berikut adalah potongan kode yang digunakan untuk melakukan validasi terhadap variabel lingkungan tersebut:

```
:
def _validate_env_vars(self):
    required_vars = [
        "UPSTASH_REDIS_HOST",
        "UPSTASH_REDIS_PORT",
        "UPSTASH_REDIS_PASSWORD",
        "TWILIO_ACCOUNT_SID",
        "TWILIO_AUTH_TOKEN",
        "TWILIO_PHONE_NUMBER",
        "VOYAGE_API_KEY",
        "PINECONE_API_KEY",
        "OPENAI_API_KEY"
    ]
    missing_vars = [var for var in required_vars
if not os.getenv(var)]
    if missing_vars:
        logger.error(f"Missing environment
variables: {missing_vars}")
        raise EnvironmentError(f"Missing required
environment variables: {missing_vars}")
```

Dalam metode ini, peneliti mendefinisikan daftar *required_vars* yang berisi nama-nama variabel lingkungan yang diperlukan. Kemudian, peneliti menggunakan *list comprehension* untuk mengecek apakah setiap variabel dalam daftar tersebut telah disetel. Jika ada variabel yang hilang, sistem akan

mencatat kesalahan dan menghentikan proses dengan menimbulkan *EnvironmentError*.

6. Inisialisasi Klien Eksternal

Metode `_init_external_clients()` memiliki tanggung jawab penting dalam mengatur dan menginisialisasi koneksi ke berbagai layanan eksternal yang diperlukan oleh sistem. Layanan-layanan ini mencakup, namun tidak terbatas pada, *Redis* untuk penyimpanan riwayat percakapan, *Twilio* untuk pengiriman dan penerimaan pesan melalui platform *WhatsApp*, serta *Voyage AI* untuk menghasilkan representasi vektor dari teks yang diproses.

Koneksi ke layanan-layanan ini sangat krusial karena memungkinkan *chatbot* menjalankan berbagai fungsi utama dengan baik. Tanpa inisialisasi yang benar, sistem tidak akan bisa berinteraksi dengan layanan eksternal secara optimal, yang dapat mengganggu performa dan responsivitas *chatbot* dalam memberikan pengalaman yang baik bagi pengguna.

Berikut adalah potongan kode yang digunakan untuk proses inisialisasi klien eksternal:

```

def _init_external_clients(self):
    try:
        self.redis_db = redis.StrictRedis(
            host=os.getenv('UPSTASH_REDIS_HOST'),

port=int(os.getenv('UPSTASH_REDIS_PORT')),

password=os.getenv('UPSTASH_REDIS_PASSWORD'),
            decode_responses=True
        )
    except Exception as e:
        logger.error(f'Redis connection error:
{e}')
        raise
    self.twilio_client = Client(
        os.getenv('TWILIO_ACCOUNT_SID'),
        os.getenv('TWILIO_AUTH_TOKEN')
    )
    self.twilio_number =
os.getenv('TWILIO_PHONE_NUMBER')
    self.voyage_client = VoyageAClient(
        api_key=os.getenv('VOYAGE_API_KEY')
    )

```

Dalam metode ini, peneliti mencoba untuk menginisialisasi koneksi ke *Redis* menggunakan kredensial yang disimpan dalam variabel lingkungan. Jika terjadi kesalahan saat menghubungkan ke *Redis*, sistem akan mencatat kesalahan dan menghentikan proses. Kemudian, peneliti menginisialisasi klien *Twilio* dengan menggunakan *Account SID* dan *Auth Token* yang disimpan dalam variabel lingkungan.

Twilio digunakan untuk mengirim dan menerima pesan *WhatsApp*. Terakhir, peneliti menginisialisasi klien *Voyage AI* dengan menggunakan *API Key* yang disimpan dalam variabel lingkungan. *Voyage AI* digunakan untuk membuat representasi *vector* dari teks.

7. Setup *Vector Store*

Metode `_setup_vector_store()` berfungsi untuk mengatur *vector store* dengan memanfaatkan *Pinecone*, serta menyiapkan *retriever* yang akan digunakan untuk mengambil data. *Vector store* ini memainkan peran penting dalam menyimpan dan mengambil data yang berbasis *vector*, sementara *retriever* berfungsi untuk mencari dan menemukan data yang paling relevan berdasarkan pertanyaan yang diajukan oleh pengguna. Selain itu, dalam proses ini, peneliti juga melakukan pengaturan terhadap model bahasa OpenAI, yang bertujuan untuk memproses pertanyaan yang masuk dan memberikan jawaban yang sesuai.

Berikut adalah potongan kode yang digunakan untuk setup *vector store*:

```

def _setup_vector_store(self):
    def voyage_embed(text):
        try:
            response = self.voyage_client.embed(
                texts=[text],
                model='voyage-3'
            )
            return response.embeddings[0]
        except Exception as e:
            logger.error(f'Embedding error: {e}')
            raise

    try:
        pc =
Pinecone(api_key=os.getenv('PINECONE_API_KEY'))
        self.vector_store =
PineconeVectorStore.from_existing_index(
            index_name='diskusipall_semarang',
            embedding=voyage_embed,
            namespace='pencatan-spil'
        )
        self.retriever =
self.vector_store.as_retriever(
            search_kwargs={'k': 10}
        )

        self.llm = ChatOpenAI(
            model_name='gpt-3.5-turbo',
            temperature=0.5
        )
        self.qa_chain =
RetrievalQA.from_chain_type(
            llm=self.llm,
            chain_type='stuff',
            retriever=self.retriever
        )

```

```

except Exception as e:
    logger.error(f'Vector store setup error:
{e}')
    raise

```

Dalam metode ini, peneliti terlebih dahulu mendefinisikan fungsi *voyage_embed(text)* yang digunakan untuk membuat representasi *vector* dari teks menggunakan layanan *Voyage AI*. Fungsi ini menerima teks sebagai input dan mengembalikan *vector* yang direpresentasikan oleh teks tersebut. Jika terjadi kesalahan saat membuat *vector*, sistem akan mencatat kesalahan dan menghentikan proses.

Kemudian, peneliti menginisialisasi koneksi ke *Pinecone* menggunakan *API Key* yang disimpan dalam variabel lingkungan. *Pinecone* digunakan sebagai *vector store* untuk menyimpan dan mengambil data berbasis *vector*. Setelah menginisialisasi koneksi ke *Pinecone*, peneliti mengatur *vector store* dengan menggunakan indeks yang sudah ada bernama "disdukcapii-semarang" dan fungsi embedding yang telah didefinisikan sebelumnya. *Namespace* "pencatatan-sipil" digunakan untuk membedakan data yang disimpan dalam *vector store*.

Selanjutnya, peneliti mengatur *retriever* dari *vector store* yang telah diinisialisasi. *Retriever* ini digunakan untuk mencari data yang paling relevan

berdasarkan pertanyaan pengguna. Parameter `search_kwargs={"k": 10}` menunjukkan bahwa *retriever* akan mengembalikan 10 hasil pencarian teratas.

Kemudian, peneliti menginisialisasi model bahasa *OpenAI* dengan menggunakan nama model "*gpt-4o-mini*" dan parameter `temperature=0.5`. Parameter `temperature` mengontrol keacakan respons yang dihasilkan oleh model. Nilai 0.5 menunjukkan bahwa respons akan memiliki keseimbangan antara keakuratan dan keacakan.

Terakhir, peneliti mengatur *QA chain* menggunakan model bahasa yang telah diinisialisasi dan *retriever* yang telah diatur sebelumnya. *QA chain* ini digunakan untuk memproses pertanyaan pengguna dan memberikan jawaban yang akurat dan relevan.

8. Menyimpan Riwayat Percakapan

Metode `save_conversation()` digunakan untuk menyimpan riwayat percakapan di *Redis*. Riwayat percakapan ini digunakan untuk memberikan konteks saat memproses pertanyaan pengguna.

Potongan kode untuk menyimpan riwayat percakapan:

```

def save_conversation(self, user_id: str,
message: str, role: str = "user"):
    """Menyimpan pesan percakapan di Redis."""
    try:
        self.redis_db.rpush(
            f"conversation:{user_id}",
            f"{role}: {message}"
        )

    self.redis_db.ltrim(f"conversation:{user_id}", -
3, -1)
    except Exception as e:
        logger.error(f'Error saving conversation:
{e}')

```

Dalam metode ini, peneliti mencoba untuk menyimpan pesan ke *Redis* dengan menggunakan metode *rpush()*. Pesan disimpan dengan key yang berformat *conversation:user_id*, di mana *user_id* adalah identifikasi unik pengguna. Pesan disimpan dengan format *role: message*, di mana *role* bisa berupa "user" atau "assistant". Setelah pesan disimpan, peneliti membatasi jumlah pesan yang disimpan di *Redis* dengan menggunakan metode *ltrim()*, sehingga hanya 3 pesan terakhir yang disimpan.

9. Memproses Webhook

Metode yang dinamakan `process_webhook()` memiliki fungsi utama untuk memproses pesan-pesan yang diterima dari platform komunikasi populer, *WhatsApp*. Setelah menerima pesan tersebut, sistem dengan cepat melakukan pemrosesan lebih lanjut untuk mencari dan menentukan jawaban yang paling relevan berdasarkan informasi yang ada. Proses pencarian jawaban ini tidak hanya bergantung pada pesan yang baru diterima, tetapi juga mempertimbangkan riwayat percakapan yang telah terjadi sebelumnya serta data yang tersedia dalam sistem. Dengan pendekatan ini, metode `process_webhook()` berperan sangat penting dalam memastikan bahwa respon yang diberikan kepada pengguna benar-benar sesuai dengan konteks dan kebutuhan mereka. Hal ini memungkinkan interaksi yang lebih alami dan relevan dalam setiap sesi percakapan. Di bawah ini merupakan potongan kode yang menunjukkan cara untuk memproses webhook secara efektif :

```

async def process_webhook(self, request:
Request):
    try:
        data = await request.form()
        user_message = data.get('Body',
('').strip()
        from_number = data.get('From')
        if not user_message or not from_number:
            logger.warning('Invalid input:
message or phone number.')
            return
        self._create_twilio_response('Pesan tidak
valid.')
        logger.info(f'Menerima pesan dari
{from_number}: {user_message}')
        conversation_history =
self.get_conversation_history(from_number)
        conversation_context =
"\n".join(conversation_history) + f"\nUser:
{user_message}"
        try:
            response = await
self.qa_chain.run(conversation_context)
        except Exception as e:
            logger.error(f'QA Chain error: {e}')
            response = "Maaf, saya mengalami
kesulitan menjawab pertanyaan Anda saat ini."
            if not response:
                response = "Maaf, saya tidak memiliki
informasi yang relevan untuk pertanyaan Anda."
            self.save_conversation(from_number,
user_message, role='user')
            self.save_conversation(from_number,
response, role='assistant')

```

```

        return
    self._create_twilio_response(response)
    except Exception as e:
        logger.error(f'Webhook processing error:
        {e}', exc_info=True)
        return
    self._create_twilio_response('Terjadi kesalahan
    internal.')

```

Dalam metode ini, peneliti mencoba untuk memproses pesan yang diterima dari *WhatsApp*. Pertama-tama, peneliti mengambil data pesan dari *form Twilio* menggunakan *request.form()*. Data pesan kemudian diekstraksi dengan menggunakan *data.get('Body')* untuk pesan dan *data.get('From')* untuk nomor pengirim. Jika pesan atau nomor pengirim kosong, sistem akan mencatat peringatan dan mengirimkan respons "Pesan tidak valid".

Jika pesan dan nomor pengirim valid, sistem akan mencatat pesan yang diterima dan mengambil riwayat percakapan dari *Redis* menggunakan metode *get_conversation_history(from_number)*. Riwayat percakapan kemudian digabungkan dengan pesan baru untuk membentuk konteks percakapan yang akan diproses.

Kemudian, peneliti menjalankan *QA chain* dengan menggunakan konteks percakapan yang telah dibentuk. *QA chain* ini akan mencari jawaban yang

paling relevan berdasarkan konteks percakapan dan data yang tersedia. Jika terjadi kesalahan saat menjalankan *QA chain*, sistem akan mencatat kesalahan dan mengirimkan respons "Maaf, saya mengalami kesulitan memproses pertanyaan Anda saat ini".

Jika *QA chain* tidak menghasilkan jawaban, sistem akan mengirimkan respons "Maaf, saya tidak memiliki informasi yang relevan untuk pertanyaan Anda". Jawaban yang dihasilkan kemudian dipotong menjadi beberapa bagian jika panjangnya melebihi batas maksimum menggunakan metode *split_message(response)*.

Setelah itu, pesan pengguna dan jawaban dari *chatbot* disimpan ke *Redis* menggunakan metode *save_conversation()*. Akhirnya, sistem mengirimkan respons ke *WhatsApp* menggunakan metode *_create_twilio_response(messages)*.

10. Fungsi Utama

Fungsi *main()* digunakan untuk menjalankan aplikasi *chatbot*. Fungsi ini menginisialisasi objek *WhatsAppChatbot* dan mengembalikan aplikasi *FastAPI* yang siap untuk menerima pesan dari *WhatsApp*.

Potongan kode untuk fungsi `main()`:

```
def main():
    try:
        chatbot = WhatsAppChatbot()
        return chatbot.app
    except Exception as e:
        logger.error(f'Aplikasi gagal
diinisialisasi: {e}')
        sys.exit(1)
app = main()
if __name__ == "__main__":
    import uvicorn
    uvicorn.run('whatsapp_bot:app',
host="0.0.0.0", port=8080, reload=True)
```

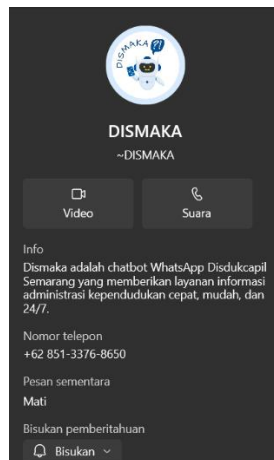
Dalam fungsi `main()`, peneliti mencoba untuk menginisialisasi objek *WhatsAppChatbot*. Jika inisialisasi berhasil, fungsi akan mengembalikan aplikasi *FastAPI* yang telah diatur oleh objek *WhatsAppChatbot*. Jika terjadi kesalahan saat menginisialisasi objek *WhatsAppChatbot*, sistem akan mencatat kesalahan dan menghentikan proses dengan menggunakan `sys.exit(1)`.

Setelah fungsi `main()` dijalankan, objek `app` yang dihasilkan akan digunakan untuk menjalankan *server FastAPI*. *Server FastAPI* kemudian dijalankan menggunakan `uvicorn` dengan konfigurasi `host="0.0.0.0"`, `port=8000`, dan `reload=True`. Konfigurasi `host="0.0.0.0"` menunjukkan bahwa

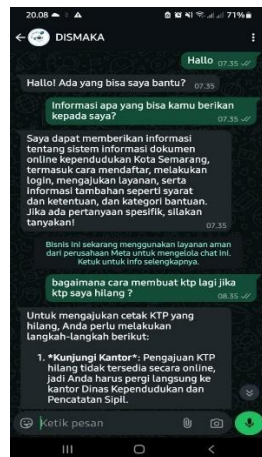
server akan menerima permintaan dari alamat IP apa pun, *port=8000* menunjukkan bahwa *server* akan berjalan pada *port 8000*, dan *reload=True* menunjukkan bahwa *server* akan otomatis merefresh ketika ada perubahan pada kode.

11. Tampilan *Chatbot WhatsApp*

Pada bagian ini, akan dijelaskan tampilan antarmuka pengguna dari *chatbot* yang diakses melalui platform *WhatsApp*. Tampilan ini dirancang untuk memberikan kemudahan interaksi antara pengguna dan sistem *chatbot*, sekaligus memastikan pesan yang disampaikan dapat diproses secara efektif.



Gambar 4.1 Profile Chatbot



Gambar 4.2 Percakapan Bot

Pada gambar 4.1 memperlihatkan profil *chatbot* DISMAKA, yang merupakan layanan berbasis *WhatsApp* dari Dinas Kependudukan dan Pencatatan Sipil (Disdukcapil) Kota Semarang. *Chatbot* ini dirancang untuk memberikan kemudahan bagi masyarakat dalam mengakses informasi terkait administrasi kependudukan, seperti layanan pembuatan dokumen, persyaratan administrasi, dan panduan lainnya. Layanan ini tersedia 24 jam sehari, sehingga pengguna dapat mengaksesnya kapan saja dengan mudah.

Sementara itu, Gambar 4.2 menampilkan simulasi percakapan antara pengguna dan *chatbot* DISMAKA. Dalam percakapan tersebut, pengguna bertanya mengenai prosedur pengurusan KTP yang hilang. *Chatbot* memberikan respon berupa panduan langkah-langkah yang harus dilakukan, seperti mengunjungi kantor Disdukcapil secara langsung karena layanan tersebut tidak dapat dilakukan secara online. Hal ini menunjukkan bahwa *chatbot* DISMAKA mampu memberikan informasi yang relevan dan membantu pengguna dengan panduan yang jelas dan terstruktur

12. Dashboard (Monitoring)

Dashboard monitoring merupakan fitur yang digunakan untuk memantau kinerja dan status sistem secara *real-time*. Dengan adanya dashboard ini, pengelola sistem dapat memastikan seluruh komponen berjalan sesuai dengan fungsinya. Berikut adalah penjelasan masing-masing alat yang digunakan dalam monitoring:

a. *RedisInsight* (Monitoring cache)

RedisInsight digunakan untuk memantau performa cache yang berfungsi menyimpan data sementara untuk mempercepat akses informasi. Pada *RedisInsight*, statistik seperti penggunaan *Bandwidth*, jumlah key yang tersimpan dapat dimonitor. Detail tampilan *RedisInsight* dapat dilihat pada Lampiran 5 dan Lampiran 6.

b. *Twilio* (Log Pesan *WhatsApp*)

Twilio menyediakan *log* untuk mencatat aktivitas pengiriman dan penerimaan pesan *WhatsApp*. *Log* ini mencakup waktu pengiriman, status pesan (terkirim, gagal, atau diterima), dan nomor tujuan. Visualisasi dari *log* pesan ini tersedia pada Lampiran 11.

c. *Railway* (Status hosting aplikasi)

Railway digunakan untuk memantau status *hosting* aplikasi secara *real-time*. Dashboard ini menyediakan informasi mengenai uptime aplikasi, penggunaan *resource* seperti *memory* dan *cpu*, serta *log error* jika terjadi gangguan. Fitur *monitoring* ini membantu pengelola untuk segera mendeteksi dan menangani permasalahan yang muncul, sehingga aplikasi dapat berjalan secara optimal. Visualisasi tampilan dashboard *Railway* dapat dilihat pada Lampiran 8.

d. *OpenAI* (Statistik penggunaan API *OpenAI*)

Monitoring penggunaan *API OpenAI* dilakukan melalui dashboard *OpenAI* yang menyediakan statistik lengkap. Statistik ini mencakup jumlah permintaan yang dikirim, rata-rata waktu respons, dan biaya yang dihasilkan. Data ini sangat berguna untuk mengevaluasi efisiensi penggunaan *API* dalam sistem *chatbot*. Tampilan dashboard *OpenAI* yang menunjukkan statistik penggunaan dapat dilihat pada Lampiran 4.

e. *Pinecone (Status Indeks Vector)*

Pinecone merupakan layanan yang digunakan untuk memantau status indeks *vector* dalam sistem. Indeks *vector* ini merupakan inti dari pencarian berbasis *vector* yang digunakan dalam *chatbot*. Dashboard *Pinecone* menampilkan informasi seperti performa pencarian, serta tingkat ketersediaan layanan. Tampilan visual dari dashboard *Pinecone* tersedia pada Lampiran 7.

C. Akurasi *Chatbot*

Data untuk pengujian akurasi *chatbot* diambil dari riwayat chat yang disimpan di *Redis*. Langkah selanjutnya adalah memberi label pada data dengan nilai "*true*" atau "*false*" untuk mengidentifikasi apakah pertanyaan yang diajukan pengguna dapat dijawab dengan benar oleh *chatbot* atau tidak.

1. Skenario Pengujian

Berikut adalah data pengujian yang digunakan untuk memastikan bahwa respons *chatbot* dapat diidentifikasi dengan benar berdasarkan label "*true*" atau "*false*" sesuai dengan kemampuannya dalam menjawab pertanyaan pengguna:

Tabel 4.2 Skenario Pengujian

No	Pertanyaan	Jawaban	Label
1	Apa saja yang bisa dilakukan di sidenok	SiDenok, Anda dapat melakukan berbagai layanan terkait dokumen kependudukan, antara lain: 1. Pendaftaran online untuk Kartu Keluarga (KK). 2. Mengajukan perubahan data pada Kartu Keluarga. 3. Mengajukan permohonan pembuatan KTP. 4. Mengajukan perpanjangan Kartu Identitas Anak (KIA).	True

		5. Mengajukan permohonan akta kelahiran dan pendaftaran untuk anak baru lahir. 6. Mengajukan pindah domisili. 7. Mengunggah bukti dukung yang diperlukan untuk permohonan. Untuk informasi lebih lanjut, Anda bisa mengunjungi website resmi Disdukcapil Kota Semarang.	
2	Layanan apa yang diberikan oleh disdukcapil?	DISMAKA memberikan layanan informasi terkait administrasi kependudukan, termasuk panduan prosedur untuk pengurusan dokumen seperti Kartu Tanda Penduduk (KTP), Kartu	True

		Keluarga (KK), Akta Kelahiran, Akta Kematian, Surat Pindah Domisili, dan Perubahan Data Kependudukan. Selain itu, DISMAKA juga siap menjawab pertanyaan umum seputar layanan kependudukan.	
3	Siapa sih yang bikin DISMAKA?	DISMAKA diciptakan oleh Eros Alfredo Hermanto, seorang mahasiswa semester akhir dari UIN Walisongo Semarang. Dia memiliki dedikasi tinggi untuk menciptakan solusi digital berbasis kecerdasan buatan (AI) dan menghadirkan DISMAKA sebagai kontribusi nyata dalam mendukung transformasi digital di lingkungan pemerintahan.	True

4	Bagaimana cara mengurus pencatatan pernikahan	Saya tidak tahu.	False
5	Cara membuat ktp di kota semarang	Untuk membuat KTP di Kota Semarang, Anda harus melakukan pengajuan secara langsung di kantor Dinas Kependudukan dan Pencatatan Sipil (Disdukcapil) yang berlokasi di Jl. Kanguru Raya No. 3 Semarang. Pastikan untuk membawa dokumen yang diperlukan, seperti fotokopi Kartu Keluarga (KK) dan akta kelahiran. Pengajuan KTP baru dilayani setiap hari kerja, yaitu Senin hingga Juma	True
6	jam buka mulai jam berapa?	Jam buka pelayanan adalah sebagai berikut:	True

		- Senin – Kamis: jam 08.00 – 14.00 WIB - Jumat: jam 08.00 – 11.00 WIB Pelayanan daring tersedia 24 jam.	
7	Apa aja layanan dismaka	Saya tidak tahu	False
8	Dimana lokasi kantor dukcapil semarang	Kantor Dinas Kependudukan dan Pencatatan Sipil Kota Semarang berlokasi di Jl. Kanguru Raya No. 3, Semarang, Jawa Tengah, dengan kode pos 50161.	True
9	Buat WNA (Orang Asing), syarat bikin KTP baru apa bedanya?	Syarat untuk WNA (Warga Negara Asing) dalam pembuatan KTP baru meliputi:	True

		1. Izin kawin dari Kedutaan. 2. Fotokopi Akta Kelahiran yang diterjemahkan oleh Lembaga Penerjemah Resmi. 3. Paspor dan visa. 4. Dokumen imigrasi. 5. Rekomendasi dari Ditjen Protokol Konsuler Kementerian Luar Negeri jika negara asal tidak memiliki perwakilan di Indonesia. Jika Anda membutuhkan informasi lebih lanjut, silakan tanyakan!	
--	--	---	--

10	Urus KK itu bayar atau gratis?	Urus KK itu gratis, tidak dipungut biaya.	True
11	Kalo Akta Kelahiran hilang, cara mengurusnya gimana?	Untuk mengurus akta kelahiran yang hilang, Anda perlu mengajukan permohonan pembuatan akta kelahiran baru. Persyaratan yang biasanya diperlukan meliputi surat keterangan lahir dari rumah sakit atau bidan, kartu keluarga (KK) dan KTP orang tua, serta surat nikah orang tua halaman lengkap. Anda dapat mengajukan permohonan secara online melalui aplikasi si d'nok atau langsung ke kantor TPDK kecamatan domisili.	True
12	Dokumen apa aja yang dibutuhkan buat mengurus	Untuk mengurus akta kematian WNI, Anda perlu	True

	Akta Kematian WNI?	memenuhi persyaratan berikut: 1. Surat kematian (visum) dari dokter atau petugas kesehatan (asli) atau DPP-5 dari kelurahan bagi yang meninggal di rumah. 2. Surat keterangan kematian dari lurah (asli). 3. KTP elektronik dan KK almarhum. 4. Kutipan akta perkawinan atau akta nikah almarhum (jika sudah menikah) dengan	
--	--------------------------	--	--

		menunjukkan dokumen asli. 5. Bukti atau ketetapan ganti nama (jika ada) dengan menunjukkan dokumen asli. 6. Surat pernyataan hubungan keluarga dengan almarhum bermaterai Rp10.000. 7. Nama dan identitas dua orang saksi (usia 21 tahun ke atas). 8. Untuk PNS/pensiunan: lampirkan SK/KARIP. 9. Surat kuasa bermaterai Rp10.000 dan fotokopi	
--	--	--	--

		KTP penerima kuasa (jika diwakilkan).	
13	Bisa gak legalisir Akta Kelahiran di Disdukcapil Semarang kalo KTP kita bukan Semarang?	Tidak bisa. Legalisasi akta kelahiran di Disdukcapil Kota Semarang hanya dapat dilakukan jika KTP dan KK Anda sudah berdomisili di Kota Semarang.	True
14	berapa jumlah penduduk semarang	Saya tidak tahu.	False
15	Berapa lama waktu pengurusan Akta Kematian?	Waktu penyelesaian pengurusan Akta Kematian adalah 3 hari kerja setelah berkas dinyatakan lengkap oleh petugas.	True
16	Proses ngurus SKP kalo dateng langsung gimana?	Proses pengurusan SKP (Surat Keterangan Pindah) secara tatap muka	True

		adalah sebagai berikut: 1. Pemohon mendaftar melalui aplikasi SiDenok dan mengunggah dokumen persyaratan. 2. Petugas pelayanan memeriksa berkas: a. Jika berkas tidak lengkap, berkas dikembalikan untuk dilengkapi, dan permohonan ditolak. b. Jika berkas lengkap, pemohon diberikan tanda bukti	
--	--	--	--

		pengambilan, dan permohonan diproses lebih lanjut untuk penerbitan SKP. 3. SKP asli diserahkan kepada pemohon, dan berkas formulir serta dokumen diarsipkan. Jangka waktu penyelesaian untuk pelayanan tatap muka adalah 3 hari kerja sejak berkas dinyatakan lengkap oleh petugas. Biaya layanan adalah gratis.	
17	Berapa lama proses mengurus surat pindah (SKP)?	Proses pengurusan Surat Keterangan Pindah (SKP) adalah 3 hari kerja sejak berkas dinyatakan	True

		lengkap oleh petugas pelayanan.	
18	Bisa gak sih mengurus KTP orang tuaku yang domisilinya di luar kota (misal Jakarta) tapi lagi tinggal di Semarang?	Pembuatan KTP baru untuk warga dengan domisili Jakarta yang tinggal di Semarang tetap bisa dilakukan dengan datang ke kantor dinas di Jl. Kanguru Raya No. 3 Semarang.	True
19	Bisa gak minta nomor KK-ku yang hilang ke Disdukcapil?	Permintaan informasi nomor KK yang hilang tidak dapat diberikan oleh Disdukcapil.	True
20	Tingkat pendidikan apa yang paling banyak di Kota Semarang?	Berdasarkan statistik pendidikan penduduk Kota Semarang, tingkat pendidikan yang paling banyak adalah SD (Sekolah Dasar) dengan total 249.000 jiwa. ²	True

Berdasarkan data yang telah dilabeli, diperoleh perhitungan tingkat akurasi *chatbot* sebesar 85% melalui metode sebagai berikut.

$$\text{Nilai Akurasi} = \frac{17}{20} \times 100\% = 85\%$$

D. *UAT*

Evaluasi lanjutan dalam studi ini fokus pada Pengujian Penerimaan Pengguna (*User Acceptance Test*), yang bertujuan untuk mengidentifikasi tingkat kelayakan sistem yang dikembangkan dari sudut pandang pengguna akhir. Metode pengumpulan data dilaksanakan menggunakan instrumen survei berbasis skala Likert, yang telah dirinci pada Bab 3 Metodologi Penelitian, dan disebarluaskan melalui platform google form. Kuesioner yang mencakup delapan item pertanyaan dalam empat kategori utama aspek pengujian tersebut selanjutnya dianalisis untuk mengevaluasi tingkat penerimaan sistem. Responden diminta memberikan tanggapan yang akan mengungkap apakah sistem yang dihasilkan memenuhi kebutuhan dan harapan pengguna. Pembaca dapat merujuk kembali pada Bab 3 untuk melihat detail spesifik daftar pertanyaan yang digunakan dalam pengujian.

Tabel 4.3 Hasil Survei Penerimaan Sistem

Kode	Frekuensi Jawaban				
	SS (5)	S (4)	KS (3)	TS (2)	TT (1)
P1	12	10	1	0	0
P2	10	12	0	1	0
P3	11	11	1	0	0
P4	14	5	2	2	0
P5	13	8	2	0	0
P6	9	10	3	1	0
P7	12	10	1	0	0
P8	12	11	0	0	0
P9	8	12	2	0	1
Total	101	89	12	4	1

Berdasarkan analisis data pada tabel 4.3, perhitungan dilakukan dengan mengalikan setiap jawaban responden dengan bobot yang telah ditetapkan. Proses perhitungan mencakup penilaian dari berbagai kategori jawaban dengan pembobotan sebagai berikut:

- Jumlah skor untuk responden yang memilih SS =
 $101 \times 5 = 505$
- Jumlah skor untuk responden yang memilih S =
 $89 \times 4 = 356$
- Jumlah skor untuk responden yang memilih KS =
 $12 \times 3 = 36$

- Jumlah skor untuk responden yang memilih TS =
 $4 \times 2 = 8$
- Jumlah skor untuk responden yang memilih TT =
 $1 \times 1 = 1$

Setelah melakukan perhitungan, diperoleh total skor dari responden sebesar 906. Untuk menentukan rentang penilaian, dilakukan perhitungan nilai tertinggi dan terendah:

Nilai Tertinggi $\Rightarrow 23 \times 9 \times 5 \Rightarrow 1035$

Nilai Terendah $\Rightarrow 23 \times 9 \times 1 \Rightarrow 207$

Nilai maksimum yang diperoleh adalah 1035. Untuk menghitung persentase skor aktual, digunakan rumus berikut:

$$\text{skor aktual (\%)} = \frac{906}{1035} \times 100\% = 87.54\%$$

Hasil persentase ini menunjukkan bahwa skor aktual sebesar 87.54% tergolong dalam kategori "sangat baik" berdasarkan kriteria penilaian *UAT*.

Selain itu, untuk memastikan hasil pengujian lebih optimal, dilakukan analisis terhadap setiap item dari 9 soal dengan menghitung jumlah skor ideal. Perhitungan dilakukan sebagai berikut:

$$\begin{aligned} \text{Jumlah skor ideal} &= 5 \times \text{jumlah responden} \\ &= 5 \times 23 = 115 \end{aligned}$$

Rata-rata skor dalam bentuk persentase dihitung menggunakan rumus:

$$\text{Rata - rata skor (\%)} = \frac{\text{Jumlah skor}}{\text{Jumlah skor ideal}} \times 100\%$$

Chatbot memberikan informasi yang mudah dipahami sesuai dengan pertanyaan yang diajukan. ?

Tabel 4.4 Analisis UAT P1

Kategori Jawaban	Hasil		Total
	Jumlah	Skor	
Sangat Setuju	12	12 x 5 = 60	103
Setuju	10	10 x 4 = 40	
Kurang Setuju	1	1 x 3 = 3	
Tidak Setuju	0	0 x 2 = 0	
Tidak Tahu	0	0 x 1 = 0	
Persentase	$\frac{103}{115} \times 100\% = 89.5\%$		

Chatbot mampu memahami berbagai variasi pertanyaan dengan topik yang sama. ?

Tabel 4.5 Analisis UAT P2

Kategori Jawaban	Hasil		Total
	Jumlah	Skor	
Sangat Setuju	10	10 x 5 = 50	100
Setuju	12	12 x 4 = 48	

Kurang Setuju	0	$0 \times 3 = 0$	
Tidak Setuju	1	$1 \times 2 = 2$	
Tidak Tahu	0	$0 \times 1 = 0$	
Persentase	$\frac{100}{115} \times 100\% = 86.9\%$		

Chatbot mampu memberikan respons yang relevan dengan konteks pertanyaan. ?

Tabel 4.6 Analisis UAT P3

Kategori Jawaban	Hasil		Total
	Jumlah	Skor	
Sangat Setuju	11	11 x 5 = 55	102
Setuju	11	11 x 4 = 44	
Kurang Setuju	1	1 x 3 = 3	
Tidak Setuju	0	0 x 2 = 0	
Tidak Tahu	0	0 x 1 = 0	
Persentase	$\frac{102}{115} \times 100\% = 88.7\%$		

Chatbot ini dapat membantu anda dalam mencari informasi mengenai layanan Dinas Kependudukan dan Pencatatan Sipil Kota Semarang ?

Tabel 4.7 Analisis UAT P4

Kategori Jawaban	Hasil		Total
	Jumlah	Skor	

Sangat Setuju	14	$14 \times 5 = 70$	100
Setuju	5	$5 \times 4 = 20$	
Kurang Setuju	2	$2 \times 3 = 6$	
Tidak Setuju	2	$2 \times 2 = 4$	
Tidak Tahu	0	$0 \times 1 = 0$	
Persentase	$\frac{100}{115} \times 100\% = 86.9\%$		

Chatbot memberikan respons dengan format yang mudah dibaca. ?

Tabel 4.8 Analisis UAT P5

Kategori Jawaban	Hasil		Total
	Jumlah	Skor	
Sangat Setuju	13	13 x 5 = 65	103
Setuju	8	8 x 4 = 32	
Kurang Setuju	2	2 x 3 = 6	
Tidak Setuju	0	0 x 2 = 0	
Tidak Tahu	0	0 x 1 = 0	
Persentase	$\frac{103}{115} \times 100\% = 89.5\%$		

Kecepatan respon yang diberikan oleh chatbot ini?

Tabel 4.9 Analisis UAT P6

Kategori Jawaban	Hasil		Total
	Jumlah	Skor	

Sangat Setuju	9	$9 \times 5 = 45$	96
Setuju	10	$10 \times 4 = 40$	
Kurang Setuju	3	$3 \times 3 = 9$	
Tidak Setuju	1	$1 \times 2 = 2$	
Tidak Tahu	0	$0 \times 1 = 0$	
Persentase	$\frac{96}{115} \times 100\% = 83.4\%$		

Chatbot lebih menghemat waktu untuk mencari informasi layanan dibandingkan dengan mengunjungi situs web dan bertanya kepada admin?

Tabel 4.10 Analisis UAT P7

Kategori Jawaban	Hasil		Total
	Jumlah	Skor	
Sangat Setuju	12	12 x 5 = 60	103
Setuju	10	10 x 4 = 40	
Kurang Setuju	1	1 x 3 = 3	
Tidak Setuju	0	0 x 2 = 0	
Tidak Tahu	0	0 x 1 = 0	
Persentase	$\frac{103}{115} \times 100\% = 89.5\%$		

Chatbot dapat diakses setiap saat ?

Tabel 4.11 Analisis UAT P8

Kategori Jawaban	Hasil		Total
	Jumlah	Skor	
Sangat Setuju	12	$12 \times 5 = 60$	104
Setuju	11	$11 \times 4 = 44$	
Kurang Setuju	0	$0 \times 3 = 0$	
Tidak Setuju	0	$0 \times 2 = 0$	
Tidak Tahu	0	$0 \times 1 = 0$	
Persentase	$\frac{104}{115} \times 100\% = 90.4\%$		

Informasi yang diberikan chatbot ini akurat?

Tabel 4.12 Analisis UAT P9

Kategori Jawaban	Hasil		Total
	Jumlah	Skor	
Sangat Setuju	8	$8 \times 5 = 40$	95
Setuju	12	$12 \times 4 = 48$	
Kurang Setuju	2	$2 \times 3 = 6$	
Tidak Setuju	0	$0 \times 2 = 0$	
Tidak Tahu	1	$1 \times 1 = 1$	
Persentase	$\frac{95}{115} \times 100\% = 82.6\%$		

Tabel 4.13 Hasil Keseluruhan perhitungan UAT

Kode	Frekuensi Jawaban					Persentase
	SS	S	KS	TS	TT	
Aspek <i>Functionality</i> (Fungsionalitas)						88%
P1	12	10	1	0	0	89.5%
P2	10	12	0	1	0	86.9%
P3	11	11	1	0	0	88.7%
Aspek <i>Usability</i> (Kegunaan)						88%
P4	14	5	2	2	0	86.9%
P5	13	8	2	0	0	89.5%
Aspek <i>Efficiency</i> (Efisiensi)						86%
P6	9	10	3	1	0	83.4%
P7	12	10	1	0	0	89.5%
Aspek <i>Reliability</i> (Keandalan)						86%
P8	12	11	0	0	0	90.4%
P9	8	12	2	0	1	82.6%
Rata – rata persentase						87%

Berdasarkan tabel 4.12, diperoleh hasil sebagai berikut: aspek fungsionalitas sebesar 88%, kegunaan 88%, efisiensi 86%, dan keandalan 86%, dengan rata-rata persentase keseluruhan sebesar 87%. Rata-rata tersebut berada dalam rentang 84,01% - 100%, yang dikategorikan sebagai "sangat layak." Dengan demikian, *chatbot* ini dinilai sangat layak untuk digunakan.

E. Maintenance

Tahap maintenance menjadi langkah penting untuk memastikan aplikasi berfungsi dengan baik dan terus berkembang sesuai kebutuhan pengguna. Proses ini mencakup pemantauan kinerja sistem, pengumpulan umpan balik pengguna melalui kuesioner, serta identifikasi dan perbaikan kendala yang muncul selama penggunaan.

1. Evaluasi Hasil Penggunaan Publik

Setelah sistem *chatbot* dipublikasikan dan diuji oleh pengguna melalui survei dan kuesioner, ditemukan beberapa kendala yang dihadapi pengguna. Evaluasi ini dilakukan untuk mengidentifikasi masalah dan menentukan tindakan perbaikan yang diperlukan. Berikut adalah ringkasan kendala yang dilaporkan pengguna:

a. Kendala Pemahaman Pertanyaan

Chatbot kesulitan memahami beberapa pertanyaan yang bersifat spesifik atau di luar cakupan data yang disediakan.

b. Kecepatan Respon yang Lambat

Pengguna melaporkan bahwa respons *chatbot* terkadang lambat, terutama saat jumlah pengguna meningkat.

c. Keterbatasan Informasi

Chatbot belum mencakup seluruh informasi yang dibutuhkan, seperti status pengajuan dan lokasi layanan.

d. Interaktivitas dan Fitur Tambahan

Pengguna mengusulkan penambahan fitur interaktif seperti gambar, tautan, dan dukungan multibahasa.

e. Batasan Pesan *Twilio*

Ditemukan batasan jumlah pesan dari *Twilio* yang membatasi jumlah permintaan pengguna.

2. Perbaikan yang Dilakukan saat ini

a. Penambahan Informasi dan Fitur Baru

1. Memperluas dataset dengan menambahkan lebih banyak contoh pertanyaan yang relevan.
2. Menyediakan tautan langsung ke halaman fitur layanan di situs resmi Disdukcapil.

b. Penghapusan Batasan Pesan *Twilio*

Menghubungi tim dukungan *Twilio* untuk menghapus batasan pesan dengan memberikan justifikasi penggunaan sebagai layanan publik.

BAB V

KESIMPULAN DAN SARAN

A. Kesimpulan

Berdasarkan hasil penelitian dan pengembangan sistem *chatbot* menggunakan *Large Language Model (LLM)* dan *vector database* untuk layanan informasi Dinas Kependudukan dan Pencatatan Sipil Kota Semarang, dapat disimpulkan sebagai berikut:

1. Implementasi Teknologi: Sistem *chatbot* berbasis *Large Language Models (LLM)* dan *vector database* berhasil dikembangkan untuk Dinas Kependudukan dan Pencatatan Sipil Kota Semarang. Integrasi teknologi seperti GPT-40-mini, Pinecone, Langchain, dan FastAPI terbukti efektif dalam meningkatkan responsivitas dan akurasi layanan informasi.
2. Akurasi Sistem: Pengujian akurasi menunjukkan bahwa *chatbot* mampu menjawab 85% pertanyaan pengguna dengan benar. Hal ini membuktikan kemampuan sistem dalam menyediakan informasi yang relevan terkait layanan kependudukan..
3. Penerimaan Pengguna: Hasil User Acceptance Test (UAT) mencatat skor kelayakan sebesar 87% (kategori "Sangat Layak"). Aspek fungsionalitas,

kegunaan, efisiensi, dan keandalan sistem mendapat respons positif dari pengguna, terutama dalam kemudahan akses dan kecepatan respons..

4. Dampak Praktis: *Chatbot* ini mengurangi beban kerja petugas, menyediakan layanan 24/7, serta mendukung program smart city Pemerintah Kota Semarang melalui peningkatan transparansi dan aksesibilitas informasi.

B. Saran

Berdasarkan kesimpulan yang diperoleh, ada beberapa saran yang dapat diajukan untuk pengembangan sistem *chatbot* ini di masa depan:

1. Peningkatan Pemahaman terhadap Pertanyaan Spesifik: Untuk meningkatkan akurasi respons, disarankan untuk memperluas dataset dengan lebih banyak contoh pertanyaan yang spesifik. Penambahan data yang lebih variatif akan membantu *chatbot* dalam memahami pertanyaan yang lebih kompleks dan memberikan jawaban yang lebih tepat.
2. Optimasi Kecepatan Respons: Dalam rangka meningkatkan pengalaman pengguna, sebaiknya dilakukan optimasi pada sistem *backend* dan *database*. Penggunaan auto-scaling pada platform hosting, seperti Railway, dapat membantu

menangani lonjakan jumlah pengguna dan mempercepat waktu respons.

3. Peningkatan Dukungan Fitur dan Interaktivitas: Pengguna mengusulkan agar *chatbot* dilengkapi dengan fitur tambahan, seperti kemampuan untuk mengirim gambar atau tautan, serta dukungan untuk bahasa selain Indonesia. Penambahan fitur ini akan meningkatkan interaktivitas dan memperluas cakupan layanan bagi pengguna dengan latar belakang bahasa yang berbeda.
4. Pengelolaan Skalabilitas: Untuk memastikan sistem dapat menangani peningkatan jumlah pengguna secara efisien, perlu adanya pengelolaan skalabilitas yang lebih baik, termasuk pengoptimalan kapasitas *server* dan infrastruktur cloud.
5. Penyempurnaan Layanan Melalui *Feedback* Pengguna: Secara berkelanjutan, pengumpulan *feedback* pengguna melalui survei dan analisis log percakapan dapat digunakan untuk menyempurnakan dan memperbaharui sistem. Hal ini akan memastikan *chatbot* tetap relevan dan mampu memenuhi kebutuhan informasi masyarakat.

DAFTAR PUSTAKA

- Alberts, I. L., Mercolli, L., Pyka, T., Prenosil, G., Shi, K., Rominger, A., & Afshar-Oromieh, A. (2023). *Large language models (LLM) and ChatGPT: what will the impact on nuclear medicine be? European Journal of Nuclear Medicine and Molecular Imaging*, 50(6), 1549–1552.
<https://doi.org/10.1007/s00259-023-06172-w>
- Azhari, R. Y. (2022). Web Service Framework: Flask dan FastAPI. *Technology and Informatics Insight Journal*, 1(1), 80–87.
<https://jurnal.universitaspurabangsa.ac.id/index.php/tiij>
- Guntoro, G., Loneli Costaner, & Lisnawita, L. (2020). Aplikasi *Chatbot* untuk Layanan Informasi dan Akademik Kampus Berbasis Artificial Intelligence Markup Language (AIML). *Digital Zone: Jurnal Teknologi Informasi Dan Komunikasi*, 11(2), 291–300.
<https://doi.org/10.31849/digitalzone.v11i2.5049>
- Herwanza, N. A. M., Harahap, N. S., Yanto, F., & Insani, F. (2024). Penerapan Langchain Retriever dengan Model Chat Openai dalam Pengembangan Sistem *Chatbot* Hadis Berbasis Telegram. *JTIM: Jurnal Teknologi Informasi Dan Multimedia*, 6(1), 70–83.
<https://doi.org/10.35746/jtim.v6i1.514>
- Hikmah, A., Elektro, F. T., Telkom, U., Azmi, F., Elektro, F. T., Telkom, U., Nugrahaeni, R. A., Elektro, F. T., Telkom, U., Akademik, L., & Pendahuluan, I. (2023).

- 3120600010_Penelitian Terkait 2. 10(1), 371–382.
- Hong, T. E. O. K. U. O., Johan, M., & Khiri, A. (2021). *FCSIT WhatsApp Chatbot. 4*, 41–51.
- Kusuma Dewi, E. H., Pratama, I. S., Putera, A. S., & Carudin, C. (2022). Black Box Testing pada Aplikasi Pencatatan Peminjaman Buku Menggunakan Boundary Value Analysis. *STRING (Satuan Tulisan Riset Dan Inovasi Teknologi)*, 6(3), 315. <https://doi.org/10.30998/string.v6i3.11958>
- Kusumawaty, I., & Kurniawan, T. B. (2024). *Development and Evaluation of an AI-Based Chatbot for Preventing Social Media Addiction : A Waterfall Model Approach. 1*(1), 1–13.
- Muhammad Hanif Muslim, G. (2023). Survei Chatbot Layanan Informasi Sekolah Metode Artificial Intelligence Markup Language (Aiml). *Seminar Nasional Teknologi Informasi & Ilmu Komputer*, 2(1), 265–277.
- Parlika, R., Pradika, S. I., Hakim, A. M., & Manab, K. R. N. (2020). Bot Whatsapp Sebagai Pemberi Data Statistik COVID-19 Menggunakan PHP, Flask, Dan MySQL. *Jurnal Informatika Dan Sistem Informasi (JIFoSI)*, 1(2 SE-Articles), 282–293. <http://jifosi.upnjatim.ac.id/index.php/jifosi/article/view/101>
- Pasaribu, A. F. O., Darwis, D., Irawan, A., & Surahman, A. (2019). Sistem Informasi Geografis Untuk Pencarian Lokasi Bengkel Mobil Di Wilayah Kota Bandar Lampung. *Jurnal Tekno Kompak*, 13(2), 1. <https://doi.org/10.33365/jtk.v13i2.323>
- Taipalus, T. (2024). *Vector database* management systems:

- Fundamental concepts, use-cases, and current challenges. *Cognitive Systems Research*, 85(January), 101216. <https://doi.org/10.1016/j.cogsys.2024.101216>
- Yin, H., Aryani, A., & Nambiar, N. (2024). *Evaluating the Performance of Large Language Models for SDG Mapping (Technical Report)*. 1–6. <http://arxiv.org/abs/2408.02201>
- Permadi, A. B., & Handayani, L. (2024). IMPLEMENTASI QUESTION ANSWERING SYSTEM TAFSIR AL-AZHAR MENGGUNAKAN LANGCHAIN DAN LARGE LANGUAGE MODEL BERBASIS *CHATBOT* TELEGRAM. *Jurnal Teknoif Teknik Informatika Institut Teknologi Padang*, 12(1), 62-69.
- Dan, Y., Lei, Z., Gu, Y., Li, Y., Yin, J. P., Lin, J., ... & Qiu, X. (2023). EduChat: A large-scale language model-based *chatbot* system for intelligent education. *arXiv*, 2308.02773 v1.
- Vema, N. N. ROLE OF VECTOR DATABASES IN *LARGE LANGUAGE MODELS* (LLMS).
- PUJIONO, Ibnu, et al. IMPLEMENTING RETRIEVALAUGMENTED GENERATION AND VECTOR DATABASES FOR *CHATBOTS* IN PUBLIC SERVICES AGENCIES CONTEXT. *JITK (Jurnal Ilmu Pengetahuan dan Teknologi Komputer)*, 2024, 10.1: 216-223.
- Saputra, I. R. (2023). Implementasi natural language processing dalam pembuatan telegram *chatbot* di Program Studi

Teknologi Informasi UIN Walisongo Semarang [Skripsi,
Universitas Islam Negeri Walisongo Semarang].

Yuniarti, W. D. (2019). Dasar-dasar pemrograman dengan Python.
Deepublish.

Nielsen Norman Group. (2000). Why you only need to test with 5
users. Nielsen Norman Group. Retrieved December 28, 2024,
from [https://www.nngroup.com/articles/why-you-only-need-
to-test-with-5-users/](https://www.nngroup.com/articles/why-you-only-need-to-test-with-5-users/)

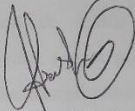
LAMPIRAN

Lampiran 1 : Lembar persetujuan pembimbing proposal

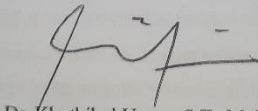
PERSETUJUAN PEMBIMBING

Proposal Skripsi ini telah disetujui oleh pembimbing untuk dilaksanakan Seminar Proposal

Disetujui pada
Hari : Selasa
Tanggal : 26 November 2024

Pembimbing 1,	Pembimbing 2,
	
Nur Cahyo Hendro Wibowo S.T., M.Kom NIP. 197312222006041001	Dr. Wenty Dwi Yuniarti S.Pd., M.Kom. NIP. 197706222006042005

Mengetahui
Ketua Jurusan Teknologi Informasi


Dr. Khothibul Umam S.T., M. Kom
NIP. 197908272011011007

iii

Lampiran 2 : Lembar pengesahan ujian proposal

 **KEMENTERIAN AGAMA REPUBLIK INDONESIA**
UNIVERSITAS ISLAM NEGERI WALISONGO SEMARANG
FAKULTAS SAINS DAN TEKNOLOGI
Alamat: Jl. Prof. Dr. Hamka Km.1 Semarang
E-mail: fst@walisongo.ac.id Web: [Http://fst.walisongo.ac.id](http://fst.walisongo.ac.id)

PENGESAHAN UJIAN KOMPREHENSIF

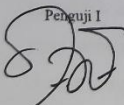
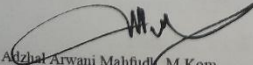
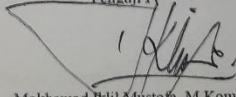
Naskah Proposal Skripsi Berikut :

Judul : Sistem Chatbot dengan Large Language Models dan Vector Database untuk Layanan Informasi Dinas Kependudukan dan Pencatatan Sipil Kota Semarang
Penulis : Eros Alfredo Hermanto
Nim : 2108096006
Jurusan : Teknologi Informasi

Telah diujikan dalam sidang komprehensif oleh Dewan Penguji Fakultas Sains dan Teknologi UIN Walisongo Semarang pada Selasa, 17 Desember 2024

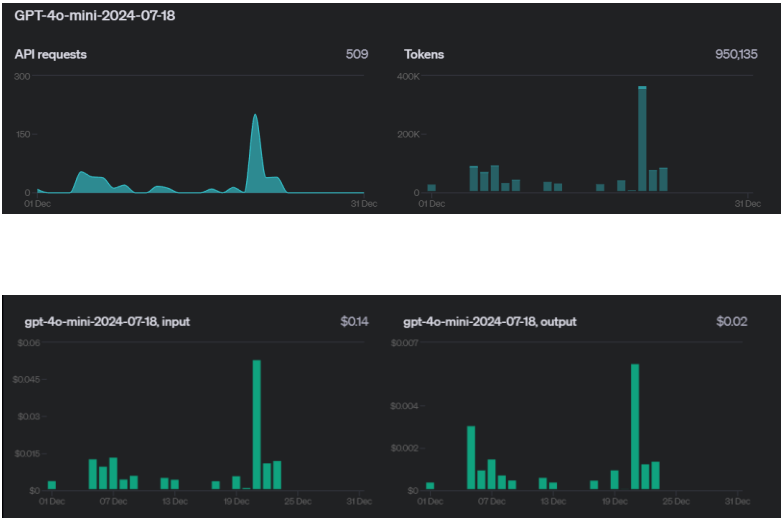
Semarang , 19 Desember 2024

Dewan Penguji

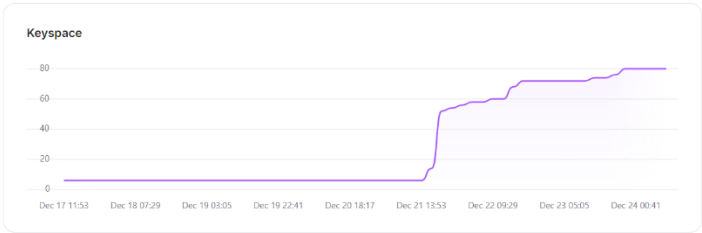
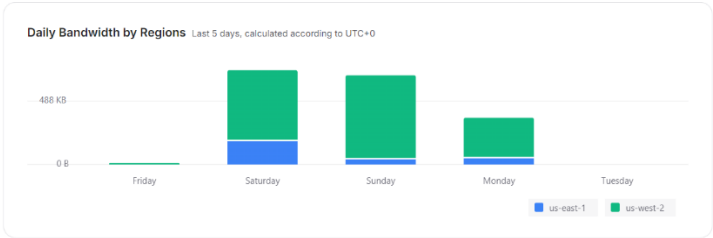
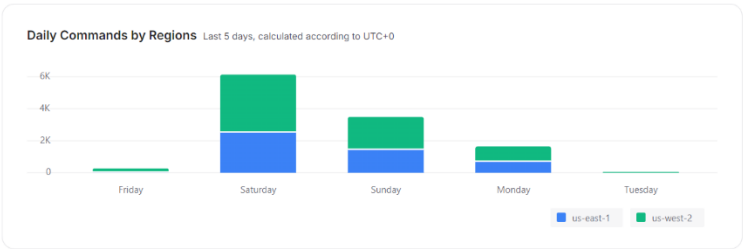
Penguji I  Hery Mustofa, M.Kom. NIP. 198703172019031007	Penguji II  Nur Cahyo Hendro Wibowo, S.T.,M.Kom NIP. 197312222006041001
Penguji III  Akhlat Arwani Mahfud, M.Kom. NIP. 199107032019031006	Penguji IV  Mokhamad Heli Mustofa, M.Kom NIP. 198808072019031010

iv

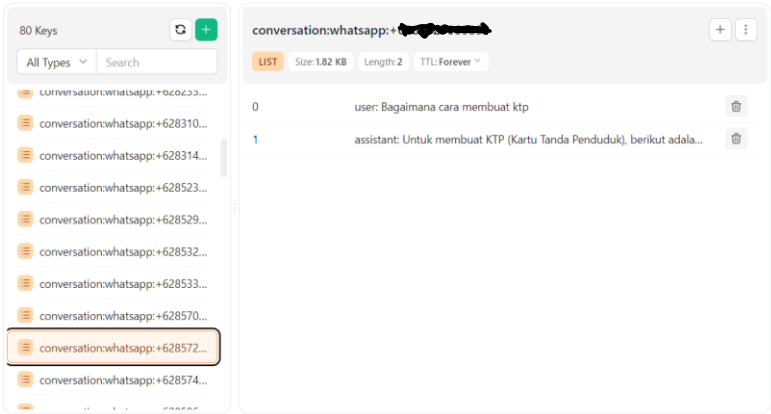
Lampiran 3 : Statistik penggunaan API *OpenAI*



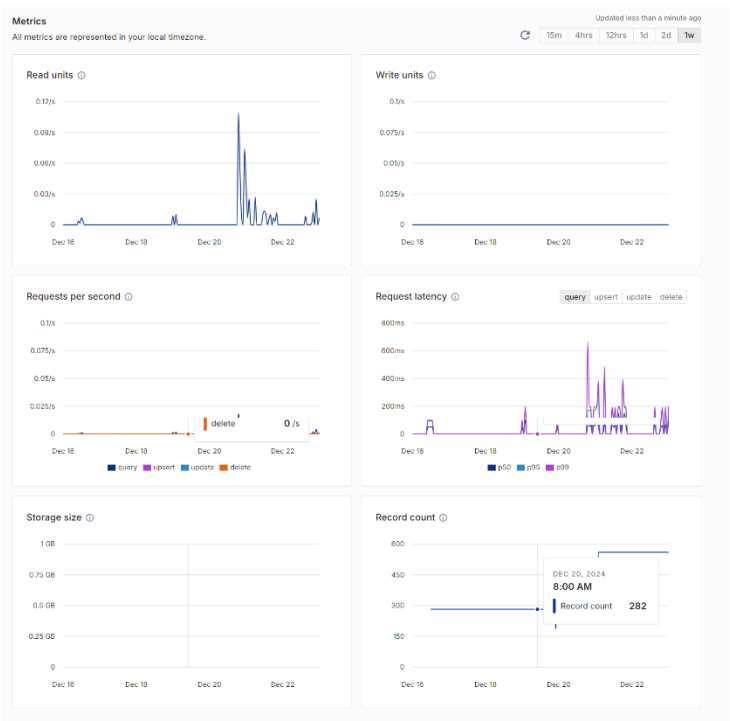
Lampiran 4 : RedisInsight



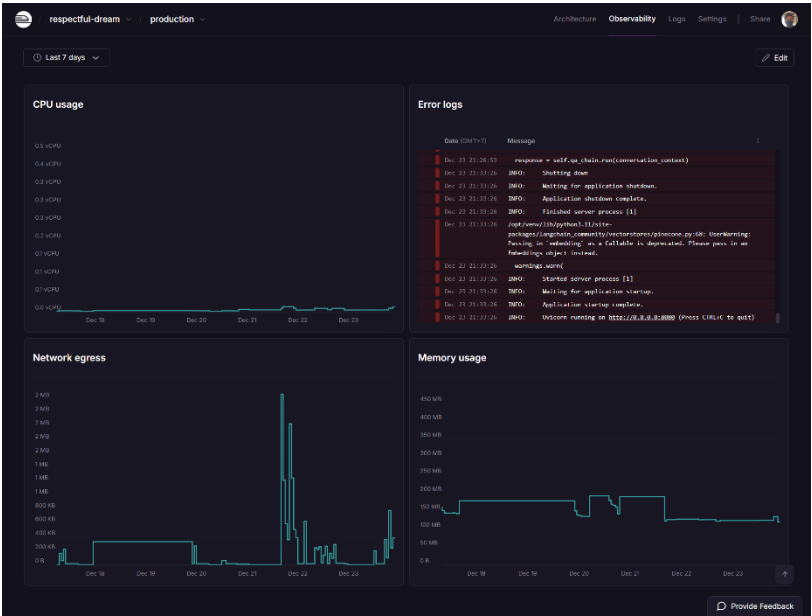
Lampiran 5 : Penyimpanan Riwayat Chat *Redis*



Lampiran 6 : Status Indeks *Vector* pada *Pinecone*



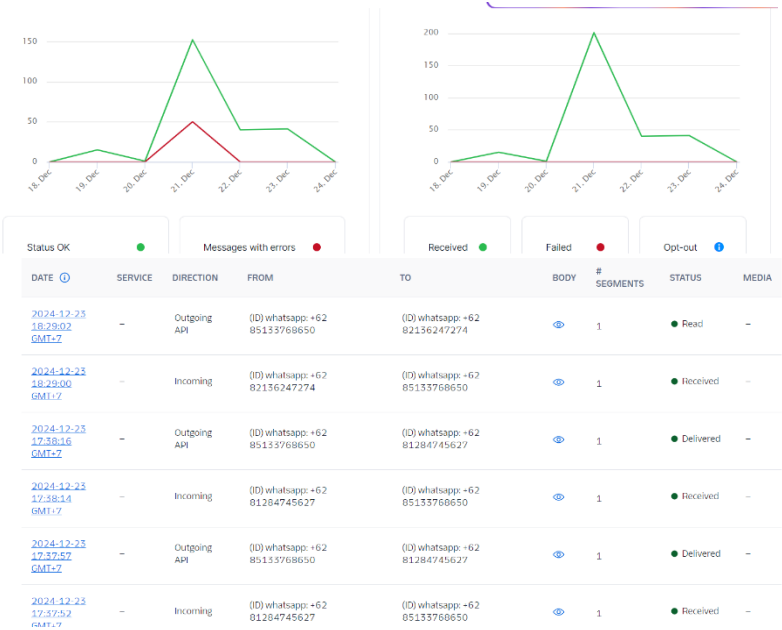
Lampiran 7 : Status Hosting Aplikasi pada Railway



Lampiran 8 : Hasil Kuisoner *UAT*

A	B	C	D	E	F	G
Form_Responden						
Timestamp	Nama	Email	Apakah chatbot memberikan informasi yang	Apakah chatbot mampu memahami berbagai	Apakah chatbot mampu memberikan respon	Apakah chatbot ba
12/21/2024 16:31:31	ina	2108096031@student.e	Setuju	Setuju	Setuju	Hangat Setuju
12/21/2024 17:29:01	Nabila	2108096011@student.e	Sangat Setuju	Sangat Setuju	Sangat Setuju	Sangat Setuju
12/21/2024 21:04:14	Lusiana Maulida	lusianamaulida@gmail	Setuju	Setuju	Sangat Setuju	Tidak Setuju
12/21/2024 21:11:07	Nanda Nurul Aulia Trian	nandanet92@gmail.com	Sangat Setuju	Sangat Setuju	Sangat Setuju	Sangat Setuju
12/21/2024 21:20:45	Naufal	naufalmarq003@gmail.e	Sangat Setuju	Setuju	Setuju	Sangat Setuju
12/22/2024 3:42:10	Nikmah Rina Wulandari	nikmahrinawulandari@g	Setuju	Setuju	Setuju	Sangat Setuju
12/22/2024 9:08:41	Muhammad Barkly	kuliahulu060@gmail.co	Setuju	Setuju	Setuju	Setuju
12/22/2024 10:03:19	Abdur Rohman Zidane P	abdurrohmanzidane92@g	Setuju	Setuju	Setuju	Kurang Setuju
12/22/2024 10:31:26	Elin Eka	elinekaturnahmi.eting@r	Kurang Setuju	Tidak Setuju	Kurang Setuju	Tidak Setuju
12/22/2024 11:13:40	Puja armedia	Pujiammedia9@gmail.i	Sangat Setuju	Sangat Setuju	Sangat Setuju	Sangat Setuju
12/22/2024 13:55:44	Nirina Nur Sa'diah	2108096006@student.e	Sangat Setuju	Sangat Setuju	Setuju	Sangat Setuju
12/22/2024 17:01:57	Tika	Tablaserem@gmail.com	Setuju	Setuju	Setuju	Sangat Setuju

Lampiran 9 : Twilio



Lampiran 10 : Data Primer

• Bagaimana prosedur a ◦ Apa ◦ Bag ◦ pros • Apakah su sini? Jika t ◦ Apa ◦ Jika	
3. Pertanyaan ui	
• Apakah An ◦ Apak ◦ infor ◦ Apak • Apa kendal ◦ Apak ◦ Apak • Apakah info ◦ Dari i ◦ Apak • Jika ada cha menggunakan ◦ Apak ◦ infor ◦ Fitur ; ◦ dokur • Bagaimana j keseluruhan? ◦ Apak ◦ Apa y	
	1. Pertanyaan untuk Observasi
	• Bagaimana alur pelayanan administrasi kependudukan di sini? ◦ Apakah masyarakat perlu mengambil nomor antrian terlebih dahulu? ◦ Berapa langkah yang harus dilalui oleh masyarakat untuk menyelesaikan satu layanan? • Dokumen apa saja yang biasanya diminta dari masyarakat dalam setiap jenis layanan (seperti KTP, KK, Akta Kelahiran)? ◦ Apakah dokumen pendukung seperti surat pengantar diperlukan? ◦ Apakah ada perbedaan persyaratan untuk kasus tertentu, seperti kehilangan KTP? • Apa saja kesulitan yang sering dihadapi masyarakat dalam memahami alur layanan? ◦ Apakah ada keluhan terkait informasi yang tidak jelas? ◦ Bagaimana petugas menangani masyarakat yang kurang memahami prosedur? • Berapa lama waktu yang dibutuhkan untuk setiap jenis layanan dari awal hingga selesai? ◦ Apakah durasi layanan bervariasi tergantung jenis dokumen yang diurus? ◦ Apakah ada layanan yang memakan waktu lebih lama dari biasanya, dan mengapa? • Bagaimana sistem antrian atau sistem informasi yang digunakan di kantor ini? ◦ Apakah sistem antrian manual, berbasis aplikasi, atau kombinasi keduanya? ◦ Bagaimana petugas memastikan tidak ada kesalahan dalam pengelolaan antrian?
	2. Pertanyaan untuk Wawancara Petugas
	• Apa jenis pertanyaan yang paling sering ditanyakan masyarakat saat mengurus dokumen kependudukan? ◦ Apakah pertanyaan seputar persyaratan dokumen lebih dominan dibandingkan pertanyaan lainnya? ✓ ◦ Apakah ada masyarakat yang meminta informasi terkait layanan lain, seperti legalisasi dokumen? ✗ • Apa kendala utama yang dihadapi petugas saat memberikan informasi kepada masyarakat? ◦ Apakah ada kendala terkait perbedaan bahasa atau istilah? ✗ ◦ Bagaimana petugas menghadapi masyarakat yang datang tanpa membawa dokumen lengkap? • Apakah ada proses yang menurut Bapak/Ibu bisa diotomatisasi untuk mempercepat layanan? Jika ya, apa saja? ◦ Apakah proses pendaftaran awal bisa dilakukan secara online? ◦ Apakah ada peluang untuk mempermudah verifikasi dokumen dengan sistem elektronik?

Lampiran 11 : Data Sekunder

**DINAS KEPENDUDUKAN DAN
PENCATATAN SIPIL
KOTA SEMARANG**

 **semarang**
simpatik dan inovatif

BERANDA • PROFIL • BERITA DAN GALERI • STATISTIK • INFORMASI • PENDAFTARAN ONLINE • WHISTLE BLOWING SYSTEM • LAPORAN PENGENDALIAN GRATIFIKASI • PPID •

LOKASI UNIT LAYANAN ADMINISTRASI •

**DINAS KEPENDUDUKAN DAN
PENCATATAN SIPIL
KOTA SEMARANG**

**GERAKAN INDONESIA SADAR
ADMINISTRASI KEPENDUDUKAN
#GISA**

 (024) 6712563  J. Kardiata Raya No. 3
Pekunden, Kecamatan Pekunden, Kota Semarang



**dinas kependudukan dan pencatatan sipil
kota semarang**

Informasi umum :

- nama dinas : dinas kependudukan dan pencatatan sipil kota semarang
- kepala dinas : drs. yudi hardianto wibisono
- alamat dinas : jl. kardiata raya no. 3 semarang - 50161 - Jawa tengah
- link website diidukapil kota semarang : <https://idukapil-cipil.semarangkota.go.id/>
- link website si d'week : <https://idweek.semarangkota.go.id/>
- whatsapp : 089675300299
- fax : 024 6787203
- jam buka kantor / pelayanan kantor :
 - senin hingga kamis : 08.15 - 15.00 wib (dengan istirahat 11.30 - 12.30 wib)
 - jumat : 08.00 - 13.00 wib (tanpa istirahat)
 - sabtu dan minggu : tutup

Visi :

"terwujudnya kota semarang yang semakin hebat berlandaskan Pancasila dalam tingkat nilai yang ber-bhineka tunggal ika"

Misi :

1. meningkatkan kualitas dan kapasitas sumber daya manusia yang unggul dan produktif untuk mencapai kesejahteraan dan keadilan sosial.
2. meningkatkan prestasi ekonomi lokal yang berdaya saing dan stimulasi pembangunan industri, berdasarkan riset dan inovasi dengan prinsip demokrasi ekonomi Pancasila.
3. menjamin pemerataan masyarakat dalam menjalankan ibadah, pemenuhan hak dasar, dan

DAFTAR RIWAYAT HIDUP

a. Identitas Diri

Nama : Eros Alfredo Hermanto
Tempat, Tanggal Lahir : Semarang, 10 Agustus 2002
Alamat : Jl. Tegalkangkung No. 131,
Rt/Rw 05/02, Kelurahan
Kedungmundu, Kecamatan
Tembalang, Kota Semarang
Hp : 0899-0084-293
Email : erosalfeo1@gmail.com

b. Riwayat Pendidikan

1. MI Taufiqiyah
2. SMP N 29 Semarang
3. SMA N 2 Mranggen