# IMPLEMENTASI CHATBOT UNTUK CUSTOMER SERVICE DENGAN METODE NATURAL LANGUAGE PROCESSING (NLP) (STUDI KASUS AYODYA PALA)

SKRIPSI



OLEH:

YOGI RIZKY PANGESTU

201011400705

**PROGRAM STUDI TEKNIK INFORMATIKA**

**FAKULTAS ILMU KOMPUTER**

**UNIVERSITAS PAMULANG**

**TANGERANG SELATAN**

**2025**

**IMPLEMENTASI CHATBOT UNTUK CUSTOMER SERVICE DENGAN METODE NATURAL LANGUAGE PROCESSING (NLP)   
(STUDI KASUS AYODYA PALA)**

**SKRIPSI**

Diajukan Untuk Melengkapi Salah Satu Syarat  
Memperoleh Gelar Sarjana Komputer



OLEH:

YOGI RIZKY PANGESTU

201011400705

**PROGRAM STUDI TEKNIK INFORMATIKA**

**FAKULTAS ILMU KOMPUTER**

**UNIVERSITAS PAMULANG**

**TANGERANG SELATAN**

**2025**

# LEMBAR PERNYATAAN

Yang bertanda tangan di bawah ini:

Nama : YOGI RIZKY PANGESTU

NIM : 201011400705

Program Studi : Teknik Informatika

Fakultas : Ilmu Komputer

Jenjang Pendidikan : Strata 1

Menyatakan bahwa skripsi yang saya buat dengan judul:

IMPLEMENTASI CHATBOT UNTUK CUSTOMER SERVICE DENGAN METODE NATURAL LANGUAGE PROCESSING (NLP) (STUDI KASUS AYODYA PALA)

1. Merupakan hasil karya tulis ilmiah sendiri, bukan merupakan karya yang pernah diajukan untuk memperoleh gelar akademik oleh pihak lain, dan bukan merupakan hasil plagiat.
2. Saya ijinkan untuk dikelola oleh Universitas Pamulang sesuai dengan norma hukum dan etika yang berlaku.

Pernyataan ini saya buat dengan penuh tanggung jawab dan saya bersedia menerima konsekuensi apapun sesuai aturan yang berlaku apabila di kemudian hari pernyataan ini tidak benar.

|  |
| --- |
| Depok, .........................2025 |
|  |
|  |
| (Yogi Rizky Pangestu) |

# LEMBAR PERSETUJUAN

|  |  |  |
| --- | --- | --- |
| NIM | : | 201011400705 |
| Nama | : | YOGI RIZKY PANGESTU |
| Program Studi | : | TEKNIK INFORMATIKA |
| Fakultas | : | ILMU KOMPUTER |
| Jenjang Pendidikan | : | STRATA 1 |
| Judul Skripsi | : | IMPLEMENTASI CHATBOT UNTUK CUSTOMER SERVICE DENGAN METODE NATURAL LANGUAGE PROCESSING (NLP) (STUDI KASUS AYODYA PALA) |

Skripsi ini telah diperiksa dan disetujui oleh pembimbing untuk persyaratan sidang skripsi

Tangerang Selatan, ..............................

Pembimbing

|  |
| --- |
| Hidayatullah Al Islami, S.Kom, M.Kom. |
| NIDN: 0409028904 |

Mengetahui,  
Ketua Program Studi Teknik Informatika

|  |
| --- |
| Dr. Eng. Ahmad Musyafa, S.Kom., M.Kom. |
| NIDN: 0425018609 |

# LEMBAR PENGESAHAN

|  |  |  |
| --- | --- | --- |
| NIM | : | 201011400705 |
| Nama | : | YOGI RIZKY PANGESTU |
| Program Studi | : | TEKNIK INFORMATIKA |
| Fakultas | : | ILMU KOMPUTER |
| Jenjang Pendidikan | : | STRATA 1 |
| Judul Skripsi | : | IMPLEMENTASI CHATBOT UNTUK CUSTOMER SERVICE DENGAN METODE NATURAL LANGUAGE PROCESSING (NLP) (STUDI KASUS AYODYA PALA) |

Skripsi ini telah dipertahankan di hadapan dewan penguji ujian skripsi fakultas Ilmu Komputer, program studi Teknik Informatika dan dinyatakan LULUS.

Tangerang Selatan, ..............................

|  |  |
| --- | --- |
| Penguji I | Penguji II |
|  |  |
|  |  |
| Nama Penguji 1 | Nama Penguji 2 |
| NIDN: - | NIDN: - |

Pembimbing

|  |  |
| --- | --- |
| Hidayatullah Al Islami, S.Kom, M.Kom. | |
| NIDN: 0409028904 |

Mengetahui,  
Ketua Program Studi Teknik Informatika

|  |  |  |
| --- | --- | --- |
| |  | | --- | | Dr. Eng. Ahmad Musyafa, S.Kom., M.Kom. | | NIDN: 0425018609 | |

# *ABSTRACT*

*Keywords:*

xi+0 pages; 0 figures; 0 tables; 0 attachments

Bibliography: 0 (2020-2025)

# ABSTRAK

Kata Kunci:

xi+0 halaman; 0 gambar; 0 tabel; 0 lampiran  
Daftar acuan: 0 (2020-2025)

# KATA PENGANTAR

Puji syukur Alhamdulillah ke hadirat Allah SWT yang telah melimpahkan segala rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan skripsi ini yang berjudul **“IMPLEMENTASI CHATBOT UNTUK CUSTOMER SERVICE DENGAN METODE NATURAL LANGUAGE PROCESSING (NLP) (STUDI KASUS AYODYA PALA)”**.

Skripsi ini disusun untuk memenuhi salah satu syarat kelulusan program studi strata satu (S1) pada program studi Teknik Informatika di Universitas Pamulang.

Penulis mengucapkan terima kasih kepada:

1. **Allah SWT** atas nikmat iman, Islam, kesehatan, dan umur panjang.
2. Bapak **Dr. Pranoto, S.E., M.M.,** selaku Ketua Yayasan Sasmita Jaya.
3. Bapak **Dr. E. Nurzaman A.M., MM., M.Si.,** selaku Rektor Universitas Pamulang.
4. Bapak **Yan Mitha Djaksana, S.Kom., M.Kom.,** selaku Dekan Fakultas Teknik Universitas Pamulang.
5. Bapak **Dr. Eng. Ahmad Musyafa, S.Kom., M.Kom.,** selaku Ketua Program Studi Teknik Informatika Universitas Pamulang.
6. Bapak **Hidayatullah Al Islami, S.Kom, M.Kom.,** selaku Dosen Pembimbing.
7. Kedua **orang tua** yang selalu mendoakan dan mendukung.
8. Seluruh **dosen**, **kerabat** dan **sahabat** serta **teman-teman** **UNPAM** seperjuangan kelas **08TPLP016**.

Semoga Allah SWT membalas segala kebaikan. Penulis menyadari bahwa skripsi ini masih jauh dari sempurna. Semoga skripsi ini bermanfaat bagi pembaca.

|  |
| --- |
| Depok, 08 Agustus 2025 |
| Yogi Rizky Pangestu |

# DAFTAR ISI

[IMPLEMENTASI CHATBOT UNTUK CUSTOMER SERVICE DENGAN METODE NATURAL LANGUAGE PROCESSING (NLP) (STUDI KASUS AYODYA PALA) i](#_Toc206867622)

[LEMBAR PERNYATAAN iii](#_Toc206867623)

[LEMBAR PERSETUJUAN iv](#_Toc206867624)

[LEMBAR PENGESAHAN v](#_Toc206867625)

[*ABSTRACT* vi](#_Toc206867626)

[ABSTRAK vii](#_Toc206867627)

[KATA PENGANTAR viii](#_Toc206867628)

[DAFTAR ISI ix](#_Toc206867629)

[DAFTAR GAMBAR xii](#_Toc206867630)

[DAFTAR TABEL xiii](#_Toc206867631)

[DAFTAR LAMPIRAN xiv](#_Toc206867632)

[BAB I PENDAHULUAN 1](#_Toc206867633)

[1.1. Latar Belakang 1](#_Toc206867634)

[1.2. Identifikasi Masalah 3](#_Toc206867635)

[1.3. Rumusan Masalah 4](#_Toc206867636)

[1.4. Batasan Penelitian 4](#_Toc206867637)

[1.5. Tujuan Penelitian 4](#_Toc206867638)

[1.6. Manfaat Penelitian 5](#_Toc206867639)

[1.7. Metodologi Penelitian 5](#_Toc206867640)

[1.8. Sistematika Penulisan 7](#_Toc206867649)

[BAB II LANDASAN TEORI 8](#_Toc206867650)

[2.1. Penelitian Terkait 8](#_Toc206867652)

[2.2. Tinjauan Pustaka 12](#_Toc206867653)

[2.2.1. Kecerdasan Buatan (Artificial Intelligence) 13](#_Toc206867654)

[2.2.2. Chatbot 14](#_Toc206867655)

[2.2.3. API (*Application Programming Interface)* 16](#_Toc206867661)

[2.2.4. *Natural Language Processing* *(NLP)* 18](#_Toc206867662)

[2.2.5. *Telegram* 22](#_Toc206867663)

[2.2.6. *Bot* *Telegram – Fatherbot* 24](#_Toc206867664)

[2.3. Flowchart Diagram 25](#_Toc206867669)

[2.4. Unified Modelling Language (UML) 26](#_Toc206867670)

[2.4.1. *Use Case Diagram* 26](#_Toc206867677)

[2.4.2. Activity Diagram 28](#_Toc206867678)

[2.4.3. Sequence Diagram 30](#_Toc206867679)

[2.4.4. Class Diagram 32](#_Toc206867680)

[2.5. Pengujian Sistem 34](#_Toc206867683)

[2.5.1. Pengujian *Blackbox Testing* 34](#_Toc206867684)

[2.5.2. Pengujian *Whitebox Testing* 34](#_Toc206867685)

[2.6. Perangkat Lunak Pendukung 34](#_Toc206867687)

[2.6.1. Bahasa Pemrograman *Python* 35](#_Toc206867689)

[2.6.2. *Pycharm* 35](#_Toc206867690)

[2.6.3. *Draw.io* 36](#_Toc206867691)

[2.6.4. *Replit* 36](#_Toc206867692)

[2.6.5. *User Response* (Kuesioner) 37](#_Toc206867693)

[BAB III ANALISA DAN PERANCANGAN 38](#_Toc206867694)

[3.1. Analisa Kebutuhan 38](#_Toc206867698)

[3.2. Analisa Sistem 39](#_Toc206867701)

[3.2.1. Analisa Sistem Berjalan 39](#_Toc206867703)

[3.2.2. Analisa Sistem Usulan 40](#_Toc206867704)

[3.3. Perancangan *Unified Modelling Language (UML)* 42](#_Toc206867705)

[3.3.1. Use Case Diagram 42](#_Toc206867707)

[3.3.2. Activity Diagram 43](#_Toc206867713)

[3.3.3. Sequence Diagram 50](#_Toc206867722)

[3.3.4. *Class Diagram* 55](#_Toc206867723)

[3.4. Perancangan Antarmuka *(User Interface)* 55](#_Toc206867724)

[BAB IV IMPLEMENTASI DAN PENGUJIAN 58](#_Toc206867725)

[4.1. Spesifikasi 58](#_Toc206867727)

[4.1.1. Spesifikasi Perangkat Lunak 58](#_Toc206867730)

[4.1.2. Spesifikasi Perangkat Keras 58](#_Toc206867731)

[4.2. Implementasi Program 59](#_Toc206867732)

[4.2.1. Halaman Chatbot demo pada Perangkat Laptop 60](#_Toc206867739)

[4.2.2. Tampilan Form Button Pertanyaan Pada Chatbot 60](#_Toc206867740)

[4.2.3. Tampilan *Form* Jawaban Pada *Chatbot* 61](#_Toc206867741)

[4.3. Pengujian Sistem 61](#_Toc206867742)

[4.3.1. *Functional Testing (Black Box Testing)* 61](#_Toc206867743)

# DAFTAR GAMBAR

# DAFTAR TABEL

# DAFTAR LAMPIRAN

# BAB I PENDAHULUAN

## Latar Belakang

Era revolusi industri merupakan istilah untuk mendeskripsikan perkembangan teknologi industri yang bersifat digital. Kemajuan teknologi telah membawa perubahan besar bagi kehidupan manusia sehari-hari. Perkembangan teknologi bagian ilmu komputer yang memberi program agar mesin (komputer) dapat melakukan pekerjaan seperti dan sebaik yang dilakukan oleh manusia. Agen cerdas dapat melakukan berbagai tugas mulai dari pekerjaan tenaga kerja hingga operasi yang canggih. Kecerdasan buatan tentunya juga dikembangkan dan dimanfaatkan dalam berbagai bidang industri (Aini dkk., 2023).

Perkembangan pesat teknologi informasi yang terkomputerisasi membawa banyak sekali perubahan dalam kehidupan manusia. Salah satu contohnya teknologi Artificial Intelligence (AI). Kehadiran teknologi AI memungkinkan komputer untuk melakukan tugas atau kegiatan seperti layaknya manusia. Salah satunya adalah Chatbot. Chatbot adalah implementasi kecerdasan buatan yang mensimulasikan percakapan real time antara mesin dan manusia melalui tulisan, suara dan visual. Aplikasi Chatbot sangat bergantung pada data train yang dibuat atau masukan dari pengembang chatbot saat mengenali dan merespons. Bisa dengan menggunakan metode penandaan untuk setiap pola kalimat untuk meningkatkan akurasi sistem (Chandra dkk., 2022).

Sistem layanan informasi merupakan sarana penunjang kegiatan akademik pada suatu lembaga pendidikan yang memanfaatkan sebuah teknologi. Sistem layanan informasi siswa ini dijadikan sebagai media penyampaian informasi untuk siswa yang dapat diakses secara Online melalui telegram. Secara umum sistem layanan informasi menyediakan informasi mengenai pendaftaran siswa baru, jadwal tes dan ujian kenaikan tingkat, informasi pembayaran, penyewaan kostum dan lain – lain. Dengan adanya sistem layanan infromasi siswa ini dirasa cukup membantu dan memudahkan siswa dalam mendapakan suatu informasi mengenai kegiatan di sanggar Ayodya Pala.

Chatbot dapat diartikan juga sebagai program komputer yang diatur atau dirancang untuk dapat memahami satu atau lebih dari bahasa manusia dengan menggunakan pemrosesan bahasa alami untuk memudahkan pekerjaan manusia. Chatbot juga diartikan sebagai bot pintar, agen interaktif, asisten digital, atau entitas percakapan buatan. Dilihat dari perkembangannya, chatbot akan sangat menjanjikan dalam dunia industri terutama dalam menyediakan kebutuhan pengguna dengan cepat dan dapat menanggapi secara khusus pertanyaan pengguna. Ada empat pendekatan kecerdasan buatan (artifial intelligence/ AI), yaitu berpikir layaknya manusia (thinking humanly), berpikir secara rasional (thinking rationally), bertindak layaknya manusia (acting humanly) dan bertindak secara rasional (acting rationally). Seiring berjalannya waktu kecerdasan buatan ini akan muncul mesin yang benar-benar mampu berpikir, bertindak, mendengar, melihat, dan berbicara layaknya manusia (Lestari & Subekti, 2024).

Natural Language Processing (NLP) adalah salah satu bidang ilmu komputer yang merupakan cabang dari kecerdasan buatan, dan bahasa (linguistik) yang berkaitan dengan interaksi antara komputer dan bahasa alami manusia, seperti bahasa Indonesia atau bahasa Inggris. Tujuan utama dari studi NLP adalah membuat mesin yang mampu mengerti dan memaham makna bahasa manusia lalu memberikan respon yang sesuai. NLP adalah teknik yang memungkinkan mesin untuk memahami dan memproses bahasa manusia (Putri Oktavianita & Andreas Sutanto, 2024).

Telegram sebagai aplikasi pesan instan diklaim bisa mengatasi kelemahan yang terdapat pada Whatsapp. Telegram adalah aplikasi berbasis Cloud dan alat enkripsi. Telegram menyediakan enkripsi End‑To‑End, penghancuran pesan secara otomatis, dan infrastruktur multi-pusat data. Natural Language Processing (NLP) adalah salah satu cabang kecerdasan buatan dimana komputer dirancang untuk berkomunikasi dengan manusia menggunakan bahasa alami, seperti bahasa Indonesia model komputasi seperti ini berguna untuk memudahkan komunikasi antara manusia dan komputer dalam hal pencarian informasi (Furqan dkk., 2023).

Sistem layanan informasi siswa merupakan komponen penting dalam menjaga kepuasan pelanggan dan mempertahankan basis siswa yang loyal bagi sanggar Ayodya Pala. Namun, dengan peningkatan Volume dan harapan siswa yang semakin tinggi terhadap responsivitas dan kualitas layanan, sanggar perlu mencari solusi yang inovatif untuk mengatasi tantangan ini. Salah satu solusi yang menjanjikan adalah dengan mengimplementasikan Chatbot untuk layanan informasi menggunakan metode Natural Language Processing (NLP). Dengan memanfaatkan kemajuan teknologi dalam bidang NLP, chatbot dapat memahami dan merespons permintaan pelanggan secara alami, meningkatkan efisiensi layanan pelanggan dan mengurangi beban kerja pada tim layanan pelanggan. Studi kasus pada sanggar tari Ayodya Pala akan memberikan gambaran konkret tentang implementasi Chatbot untuk layanan pelanggan dalam konteks dunia nyata.

Penelitian ini dipilih untuk memberikan suatu solusi dengan membuat suatu layanan informasi untuk siswa serta sebagai sarana yang memudahkan siswa, apa informasi yang dicari. Berdasarkan latar belakang di atas, maka penelitian yang diambil pada penelitian ini adalah **“IMPLEMENTASI CHATBOT UNTUK CUSTOMER SERVICE DENGAN METODE NATURAL LANGUAGE PROCESSING (NLP) (STUDI KASUS AYODYA PALA)”**. Diharapkan dari pembuatan Chatbot layanan informasi ini, memberikan kemudahan dalam mengetahui informasi yang dibutuhkan bagi para siswa.

## Identifikasi Masalah

Berdasarkan latar belakang di atas, terdapat beberapa masalah yang dapat di identifikasikan, yaitu sebagai berikut:

1. Karena informasi pendaftaran siswa baru masih dilakukan dengan cara konvesional, sehingga memperlambat proses informasi kepada siswa sanggar.
2. Karena jadwal tes kenaikan tingkat rentang waktu nya terlalu jauh dengan ujian kenaikan tingkat, sehingga sering mengakibatkan para siswa lupa waktu.
3. Karena penyewaan kostum untuk ujian kenaikan tingkat masih dilakukan dengan cara konvensional, sehingga membuat para siswa tidak mendapat kostum.

## Rumusan Masalah

Berdasarkan latar belakang dan identifikasi masalah yang telah diuraikan sebelumnya, rumusan masalah dalam penelitian ini adalah:

1. Bagaimana dalam menjaga tingkat responsivitas dan kualitas layanan informasi yang diharapkan?
2. Bagaimana cara mengetahui permintaan dan keluhan para siswa yang sangat bervariasi secara signifikan?
3. Bagaimana mengatasi keadaan seperti kesalahan dalam memahami pertanyaan atau memberikan informasi yang tidak akurat?

## Batasan Penelitian

Agar dalam penulisan penelitian ini tidak melebar dan lebih terarah, maka perlu adanya batasan masalah antara lain sebagai berikut :

1. Chatbot ini hanya berjalan pada aplikasi pesan instant Telegram.
2. Chatbot ini hanya berjalan dalam kondisi online.
3. Hasil request pesan ditampilkan hanya dalam bentuk teks.

## Tujuan Penelitian

Adapun tujuan dari penelitian yang penulis lakukan adalah sebagai berikut :

1. Membuat sebuah sistem yang terintegrasi antara Chatbot dan sistem layanan informasi siswa sanggar Ayodya Pala.
2. Merancang Chatbot untuk pendaftaran siswa baru pada sistem layanan informasi siswa sanggar Ayodya Pala.
3. Membuat jadwal tes dan ujian kenaikan tingkat siswa yang terintegrasi antara Chatbot dan sistem layanan informasi pada sanggar Ayodya Pala.
4. Membuat jadwal Program yang diadakan oleh sanggar Ayodya Pala dengan Chatbot Via Telegram.

## Manfaat Penelitian

Penelitian ini diharapkan dapat memberikan kontribusi dan manfaat sebagai berikut:

1. Manfaat Untuk Penulis
2. Penulis dapat lebih memahami dan memperluas wawasan dan dapat mengembangkan ilmu yang didapat dari proses perkuliahan yang berlangsung.
3. Penelitian ini merupakan salah satu syarat untuk menyelesaikan program studi Strata 1 (S1).
4. Manfaat Untuk Pengguna
5. Penulis berharap dari penelitian ini pengguna mendapatkan interaksi dengan baik, solusi yang memadai, dan respon yang relevan akan berkontribusi pada peningkatan kepuasan pengguna terhadap penggunaan *Chatbot*.

## Metodologi Penelitian

Dalam Implementasi Chatbot dengan Telegram Menggunakan Metode Natural Language Processing (NLP), Penulis menerapkan metode penelitian sebagai berikut:

2. Metode Pengumpulan Data

Dalam memperoleh informasi yang akurat, penulis akan menggunakan metode penelitian kuantitatif yang menggunakan data berupa angka untuk menjawab hipotesis penelitian (Marinu Waruwu, 2023). Metode ini dipilih karena memungkinkan pengumpulan data yang objektif dan terukur serta lebih efisien dalam hal waktu dan sumber daya, terutama saat melakukan kuesioner yang dapat diakses secara *online*. Metode pengumpulan data yang digunakan meliputi:

1. Metode Kuesioner

Metode pengumpulan data dilakukan melalui penyediaan kuesioner yang memuat serangkaian pertanyaan atau pernyataan tertulis kepada responden untuk diisi dan dijawab.

1. Metode Studi Pustaka

Teknik Pengumpulan data dilakukan dengan cara menghimpun dan memeriksa buku-buku referensi serta sumber-sumber yang relevan dengan topik penelitian.

1. Metode Observasi

Teknik pengumpulan data dilakukan melalui observasi langsung pada Chatbot yang telah dibuat sebelumnya, serta pengamatan terkait beberapa permasalahan yang berkaitan dengan kepentingan pembuatan Chatbot sebagai layanan informasi yang akurat bagi penggunanya.

1. Metode Perancangan Sistem

Pemilihan metode Natural Language Processing NLP, karena NLP dapat membuat Chatbot berkomunikasi dengan user menggunakan bahasa sehari-hari tanpa perlu mematuhi format atau bahasa yang kaku. Dengan mengimplementasikan teknologi NLP ke dalam Telegram, dapat diciptakan sebuah Chatbot yang mampu memberikan layanan dan informasi akademik kepada mahasiswa dengan cepat dan efisien (Qalimaturrahmah & Santoso, 2024).

Data PreProcessing Pada metode Natural Language Processing, terdapat proses preprocessing, antara lain:

1. *Case Folding*

Proses di mana mengubah seluruh teks menjadi huruf besar atau huruf kecil, serta menghilangkan karakter yang tidak perlu seperti tanda baca, symbol atau angka.

1. Tokenisasi

Proses memisahkan teks menjadi kata per kata individu. Dimana suatu kata ini tidak terkait dengan kata lainnya.

1. *Stop Words*

Proses di mana kata-kata penghubung seperti “dan”, “atau” yang tidak memiliki makna penting akan dihapus.

1. *Stemming*

Proses mengubah kata-kata menjadi kata dasar dengan menghilangkan unsur sufiks atau kata imbuhan pada kata, misalnya: “mendaftar” menjadi “daftar”.

## Sistematika Penulisan

Sistematika Penulisan ini dibuat dengan tujuan memberikan gambaran mengenai isi proposal secara singkat, sehingga pembaca lebih mudah untuk memahami.

**BAB I PENDAHULUAN**

Dalam bab ini, akan diuraikan secara singkat tentang latar belakang, identifikasi masalah, rumusan masalah, batasan penelitian, tujuan penelitian, manfaat penelitian dan sistematika penulisan.

**BAB II LANDASAN TEORI**

Pada bab ini, berisi mengenai landasan teori sebagai parameter rujukan untuk terlaksana nya peneltian ini, bab ini juga mencakup studi terkait yang penulis gunakan sebagai referensi.

**BAB III ANALISA DAN PERANCANGAN**

Pada bab ini, akan membahas tentang analisis kebutuhan, Diagram Aktivitas, Diagram Alur, dan Diagram Kasus Pengguna.

**BAB IV IMPLEMENTASI DAN PENGUJIAN**

Pada bab ini, akan membahas tentang implementasi pada Chatbot, yang meliputi tahapan implementasi pengujian, pengujian menggunakan metode Blackbox, Whitebox dan hasil analisa.

**BAB V PENUTUP**

Pada bab ini, terdapat rangkuman dari sistem yang telah dibuat oleh penulis, yang mencakup semua kegiatan yang telah dilakukan, serta saran-saran untuk pengembangan di masa mendatang.

# BAB II LANDASAN TEORI



## Penelitian Terkait

Dalam menyusun skripsi ini, penulis terinspirasi dan mereferensi dari penelitian-penelitian terdahulu yang masih berkaitan dengan latar belakang masalah pada skripsi ini. Berikut ini penelitian terdahulu yang berkaitan dengan skripsi, antara lain:

**Tabel 2. 1 Penelitian Terkait**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Judul Penelitian** | **Metode** | **Hasil** | **Kesimpulan** |
| 1. | Rancang Bangun *Chatbot* Informasi Lowongan PekerjaanBerbasis *Whatsapp* dengan MetodeNLP *(Natural Language Processing).*  Peneliti:  (Eka Rosyadi dkk., 2020) | *NATURAL LANGUAGE PROCESSING* (NLP). | Para pelamar harus mendaftar terlebih dahulu sebelum mencari pekerjaan. Setelah mendaftarkan dirinya, sistem akan menampilkan berbagai lowongan pekerjaan berdasarkan kriteria dan lokasi yang diinginkan. Dan sistem tersebut secara otomatis menampilkan persyaratan-persyaratan sesuai dengan bidang yang diinginkan. | Perancangan *Chatbot* *Whatsapp* menyediakan informasi lapangan pekerjaan yang sangat tepat dan cepat. |
| 2. | Implementasi *Natural Language Processing* Dalam Pembuatan *Chatbot* Pada Program *Information Technology* Universitas Surabaya.  Peneliti:  (V. R. Prasetyo dkk., 2021) | *Natural Language Processing* (NLP). | Proses kerja *Chatbot* diawali dengan pengguna menginputkan pertanyaan pada *User Interface* yang disediakan. Sistem akan berusaha mengidentifikasi kata tanya yang terdapat pada pertanyaan yang diinputkan *User*. Pertanyaan tersebut akan mengalami tokenisasi terlebih dahulu sebelum dilakukan identifikasi kata tanya. Proses tokenisasi dilakukan dengan memanfaatkan *Library* NLTK. | *Chatbot* Program *Information Technology*, Universitas Surabaya. Akurasi yang didapatkan berdasarkan hasil dua validasi tersebut adalah lebih dari 75%, yang artinya metode NLP cukup baik untuk diterapkan pada penelitian ini. |
| 3. | Implementasi NLP Dalam Pembuatan *Chatbot Customer Service Publisher* Jurnal Studi Kasus LARISMA.  Peneliti:  (Akbar Nasution dkk., 2024) | *Natural Language Processing* (NLP). | Model hanya akan memberi tahu kita kelas yang dimilikinya, jadi kita akan mengimplementasikan beberapa fungsi yang akan mengidentifikasi kelas tersebut dan kemudian mengambil respon acak dari daftar respon. Pengujian akurasi *Chatbot* ini dilakukan oleh perancang melalui terminal di *Google Colab* dengan melihat dan menilai kesesuaian antara pertanyaan dan jawaban yang dihasilkan. | *Chatbot* yang dibangun menggunakan *Python* dan berbagai *Library* didalamnya terlihat sangat baik tanpa ada kendala, hal ini membuktikan bahwa *Python* merupakan bahasa yang sangat baik dalam pengolahan data. Pemanfaatan *Chatbot* untuk *Publisher* larisma ini diharapkan mampu mempermudah admin larisma dalam menjawab pertanyaan-pertanyaan yang muncul secara berkala yang sifatnya berulang dan sudah dilatih untuk sistem *Chatbot* yang dibangun. |
| 4. | Pengembangan *Chatbot* Pengaduan Dan *Troubleshooting* Teknologi Informasi Dengan Pendekatan NLP (Studi Kasus: Politeknik Negeri Ambon).  Peneliti:  (Usmany, 2022) | *Natural Language Processing* (NLP). | *Modeling* dilakukan dengan analisis kebutuhan *User* dalam kemudahan, kecepatan dan ketepatan dalam penanganan laporan pengaduan dan tindakan *Troubleshooting* yang sedang berjalan, kemudian desain untuk perancangan arsitektur aplikasi dan antarmuka pengguna. | Sistem *Chatbot* pengaduan dan *Troubleshooting* dapat membantu teknisi *Laboratorium* dalam memberikan respon kepada pelanggan tanpa batas waktu. |
| 5. | Rancang Bangun Penyedia Layanan Informasi Pelayanan Masyarakat Kantor Kecamatan Pandaan Menggunakan *Chatbot*.  Peneliti:  (Nur Safitri & Imron Rosadi, 2021) | *Natural Language Processing* (NLP). | Chatbot ini adalah informasi tentang pelayanan masyarakat dan juga informasi umum Kantor Kecamatan Pandaan meliputi alamat kantor, jam operasional kantor, website kantor, kontak kantor, berkas persyaratan pelayanan kantor, alur pelayanan kantor, biaya pelayanan kantor dan dokumen pengajuan kependudukan. | *Chatbot* berjalan dengan baik namun masih ada *Delay* dengan rata-rata waktu 58,54 detik sebagaimana terlihat dari pengujian *Blackbox* *Testing*. Dan untuk pengujian keefektifisan dan efesiensi *WhatsApp* *Chatbot* yang didapatkan dari uji kuesioner mendapatkan sebesar 81% dari *Staf* Kantor Kecamatan Pandaan dan 87% dari masyarakat Kecamatan Pandaan yang menunjukkan sangat setuju *WhatsApp* *Chatbot* ini berhasil menjadi penyedia layanan informasi pelayanan masyarakat yang efektif dan efisien. |

## Tinjauan Pustaka

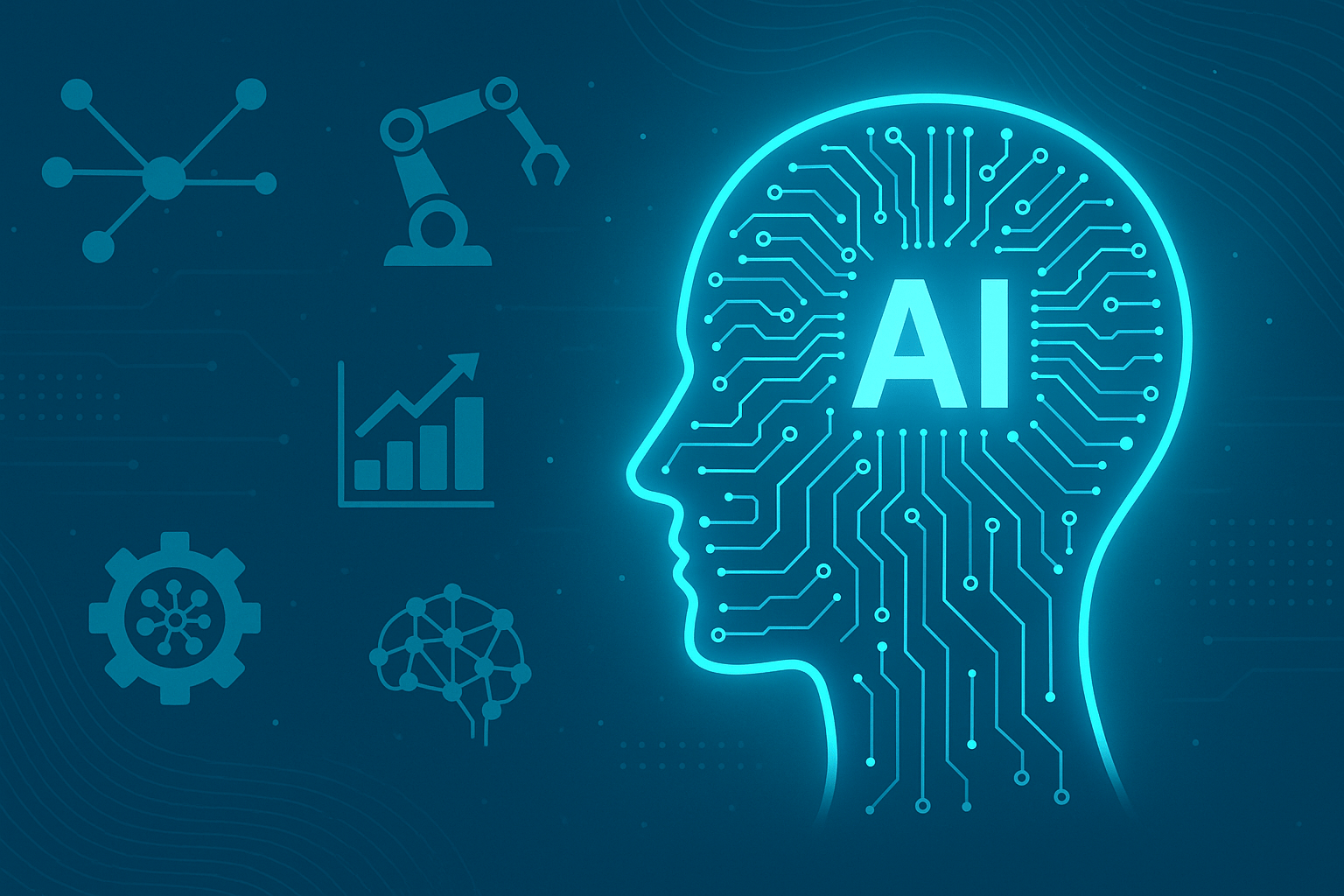
Pada bagian ini akan dibahas mengenai teori-teori dasar yang menjadi landasan dalam pengembangan sistem chatbot untuk layanan customer service di Sanggar Ayodya Pala. Teori-teori ini mencakup pemahaman mengenai teknologi chatbot, konsep dasar dan metode dalam Natural Language Processing (NLP), serta pemanfaatan bahasa pemrograman Python dan kode editor PyCharm dalam proses pengembangan. Selain itu, dijelaskan pula integrasi API Telegram sebagai media komunikasi antara chatbot dan pengguna.

Implementasi chatbot dalam penelitian ini menggunakan pendekatan berbasis NLP untuk memungkinkan sistem memahami dan merespons pesan pengguna secara alami dan kontekstual. Teknologi NLP yang digunakan berfokus pada pengolahan teks masukan pengguna, termasuk proses tokenisasi, stemming, dan klasifikasi intent, yang bertujuan untuk menciptakan alur komunikasi yang menyerupai percakapan manusia.

Bahasa pemrograman Python dipilih karena memiliki pustaka (library) yang kaya dan mendukung pengembangan chatbot berbasis NLP. Python juga memiliki integrasi yang baik dengan API Telegram, sehingga memudahkan dalam membangun sistem yang interaktif dan responsif. Untuk menunjang efisiensi dalam menulis dan menjalankan kode program, peneliti menggunakan editor PyCharm, yang memiliki fitur debugging, autocomplete, dan manajemen virtual environment yang memudahkan dalam pengelolaan proyek berbasis Python.

Penelitian ini juga memanfaatkan arsitektur bot dari Telegram yang memungkinkan pengguna berinteraksi melalui antarmuka pesan (chat interface) secara real-time. Bot Telegram memberikan fleksibilitas dalam mengatur command, menampilkan menu, dan menyimpan log percakapan pengguna. Semua komponen ini dirancang untuk mendukung peningkatan kualitas layanan informasi yang diberikan oleh Sanggar Ayodya Pala.

### Kecerdasan Buatan (Artificial Intelligence)



**Gambar 2. 1 Kecerdasan Buatan (Artificial Intelligence)**

Menurut (Padillah Akbar & Sarifudin, 2024). Kecerdasan buatan (AI), yang pada prinsipnya merupakan sebuah tiruan intelektual manusia yang diciptakan ke dalam sebuah mesin dan dibuat untuk berpikir seperti manusia dengan menggunakan algoritma pembelajaran. Dengan kata lain, AI adalah sistem komputer yang dapat melakukan pekerjaan yang biasanya membutuhkan intervensi manusia.

Menurut (Putu Ary Sri Tjahyanti dkk., 2022). Artificial Intelligence merupakan salah satu teknologi di era Revolusi Industri yang sangat berguna untuk diterapkan. Artificial Intelligence adalah kecerdasan buatan yang merupakan pemodelan dari kecerdasan manusia yang diterapkan dalam suatu mesin untuk pembuatan mesin cerdas.

Artificial Intelligence (AI) atau kecerdasan buatan merupakan suatu cabang ilmu pengetahuan dibidang komputer yang mempelajari bagaimana memberikan kemampuan atau kecerdasan seperti manusia ke dalam sebuah komputer. Pada umumnya, kecerdasan buatan di kombinasikan ke dalam komputer agar bisa membantu pekerjaan seperti yang dapat diinginkan manusia. Artificial Intelligence (AI) dapat melakukan hal-hal yang dilakukan oleh manusia, seperti menganalisis penalaran untuk menarik kesimpulan atau memutuskan atau menerjemahkan dari satu bahasa ke bahasa lain. Kecerdasan buatan mampu belajar dari pengalaman dan data yang telah dikumpulkan untuk berkomunikasi dan berbicara kata demi kata (Hikmah dkk., 2023).

Kecerdasan buatan (Artificial Intelegence, AI) yaitu suatu program yang terkomputerisasi yang membuat mesin bekerja seperti manusia contohnya seperti pengambilan keputusan, pemecahan masalah ataupun melakukan prediksi, AI sudah terjadi di berbagai sektor (publik, swasta, pemerintah, dan nirlaba) dan industri (Fathurrozi & Karimah, 2021).

### Chatbot



**Gambar 2. 2 Chatbot**

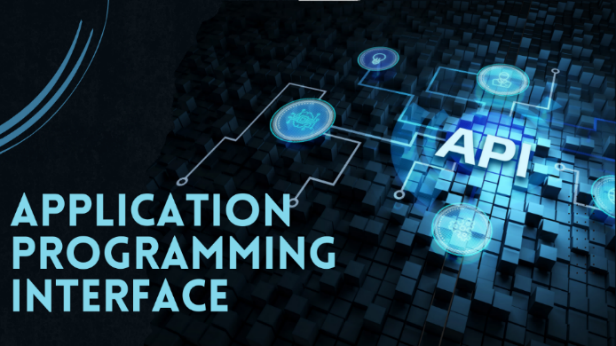
Menurut (Mursidah dkk., 2022). Chatbot adalah program berbasis Artificial Intelligence (AI) yang dapat mensimulasikan percakapan seperti manusia. Teknologi ini juga dikenal sebagai virtual assistant yang dapat memahami dan memproses permintaan pengguna serta dengan cepat memberikan jawaban yang relevan. Chatbot dapat memproses input dari pengguna, menghasilkan tanggapan, dan mengirimkannya kembali ke pengguna. Chatbot adalah aplikasi yang dapat menjadi tempat terjadinya percakapan (chatting) antara mesin (bot) dengan manusia.

Menurut (Prasojo dkk., 2024). Chatbot adalah program komputer yang dapat menyimulasikan komunikasi manusia dan dapat disertakan dalam aplikasi perpesanan seperti Facebook Messenger, WhatsApp, atau Telegram.

Menurut (Salamun, Aprialdo, dkk., 2024). Chatbot adalah program komputer yang dirancang untuk berkomunikasi dengan manusia melalui antarmuka berbasis teks atau suara. Chatbot bertujuan untuk menyediakan respons cepat dan layanan otomatis kepada pengguna, yang menjadikannya alat yang sangat berharga dalam meningkatkan efisiensi komunikasi dan memenuhi kebutuhan pengguna secara instan.



### API (*Application Programming Interface)*



**Gambar 2. 3 API (Application Programming Interface)**

API atau *Application Programming Interface* adalah sebuah antar muka yang dapat menghubungkan aplikasi satu dengan aplikasi lainnya. Jadi, API berperan sebagai perantara antar berbagai aplikasi berbeda, baik dalam satu *platform* yang sama atau lintas *platform*. Perumpamaan yang bisa digunakan untuk menjelaskan API adalah seorang pelayan di restoran. Tugas pelayan tersebut adalah menghubungkan tamu restoran dengan juru masak. Tamu cukup memesan makanan sesuai daftar menu yang ada dan pelayan memberitahukannya ke juru masak. Nantinya, pelayan akan kembali ke tamu tadi dengan masakan yang sudah siap sesuai pesanan (Athallah & Agung, 2022).

*API* adalah antarmuka yang dirancang pada sistem *server-side (backend)* untuk bertukar data dan memungkinkan interaksi antara sisi *client (frontend)* dan sisi *server (backend)*. API dapat digunakan dalam beragam bahasa pemrograman serta dapat beroperasi di berbagai jenis *server* seperti *Apache, NGINX, Tomcat,* dan sebagainya.

API terbagi menjadi 4 jenis yang digolongkan menurut hak aksesnya:

1. *Public API*

*API* Publik atau dikenal juga sebagai *API* Terbuka adalah jenis *API* yang dapat diakses secara bebas tanpa memerlukan otorisasi khusus. *API* ini kompatibel dengan berbagai sistem operasi dan cukup mudah digunakan setelah pendaftaran. Contoh dari *API* Publik adalah *Open Weather API* untuk memantau kondisi cuaca dan *News API* untuk mendapatkan berita terbaru.

1. Private API

*API* Privat adalah *API* yang dirancang khusus untuk digunakan oleh tim internal suatu organisasi dalam pengembangan aplikasi. Fungsi utamanya adalah untuk menyediakan data antara *backend* dan *frontend* pada sebuah *website*.

1. Partner API

*API* Mitra adalah jenis *API* yang bisa diakses oleh masyarakat umum, namun dengan batasan akses yang telah ditentukan. Untuk menggunakan *API* ini, pengguna harus mendaftar dan mendapatkan izin dari pemilik layanan *API*. Penggunaan *API* Mitra juga hanya diizinkan untuk aplikasi tertentu sesuai dengan syarat dan ketentuan yang berlaku. Contoh dari *API* Mitra adalah *API ChatGPT*.

1. Composite API

*API* Komposit adalah *API* yang menggabungkan data dari berbagai sumber atau *server* ke dalam satu tempat. Hal ini bertujuan untuk mempercepat proses akses data oleh pengguna karena mereka dapat mengakses berbagai jenis data dalam satu kunjungan saja (Muhammad & Paputungan, 2024).

*Application Programming Interface* secara umum adalah kumpulan fungsi yang disediakan oleh sebuah modul perangkat lunak yang dapat diakses oleh pengguna sesuai dengan aturan dan prosedur yang telah ditetapkan. Jadi, API atau *Application Programming Interface* dapat diartikan sebagai sistem yang memungkinkan beberapa aplikasi komputer berinteraksi melalui protokol dan definisi yang telah ditetapkan. API berfungsi sebagai penghubung antara dua sistem, memastikan bahwa setiap perubahan di satu sisi tidak mempengaruhi sisi lainnya. Penggunaan API dapat dibedakan berdasarkan arsitektur dan cakupan fungsionalnya.

### *Natural Language Processing* *(NLP)*



**Gambar 2. 4 Natural Language Processing (NLP)**

*Natural language processing* (NLP) merupakan pengolahan bahasa alami yang merupakan salah satu bidang kecerdsan buatan, yang dimana sistem dirancang untuk dapat berkomunikasi dengan manusia menggunakan bahasa alami (Huberta & Wijaya, 2023).

Dalam implementasi chatbot untuk layanan customer service di Sanggar Ayodya Pala dengan menggunakan metode Natural Language Processing (NLP), terdapat beberapa teknik dan metode yang umum diterapkan untuk meningkatkan kemampuan chatbot dalam memahami dan merespons percakapan pengguna. Adapun metode-metode tersebut meliputi:

1. Tokenisasi

Tokenisasi adalah proses memecah input teks dari pengguna menjadi unit-unit lebih kecil, seperti kata atau frasa. Tahap ini penting untuk mengidentifikasi struktur dan makna kalimat, serta mempermudah analisis bahasa lebih lanjut oleh sistem chatbot.

1. Part-of-Speech Tagging (POS Tagging)

Metode ini digunakan untuk menentukan kelas kata (kata benda, kata kerja, kata sifat, dan sebagainya) dari setiap kata dalam kalimat yang dikirimkan pengguna. Dengan mengetahui fungsi gramatikal setiap kata, chatbot dapat merespons dengan lebih tepat.

1. Model Bahasa Berbasis Pembelajaran Mesin

Pendekatan ini menggunakan model berbasis deep learning seperti GPT (Generative Pre-trained Transformer) untuk memahami konteks kalimat dan menghasilkan respons yang lebih alami. Model ini dilatih menggunakan data teks yang besar agar mampu meniru percakapan manusia.

1. Penerjemahan Bahasa

Chatbot juga dapat dikembangkan untuk mendukung penerjemahan otomatis dari satu bahasa ke bahasa lain, membantu pengguna dengan berbagai latar belakang bahasa.

1. Analisis Sentimen

Analisis sentimen digunakan untuk memahami emosi atau nada dari pesan yang dikirimkan oleh pengguna (positif, negatif, atau netral), yang dapat membantu chatbot menyesuaikan respons dengan konteks emosional pengguna.

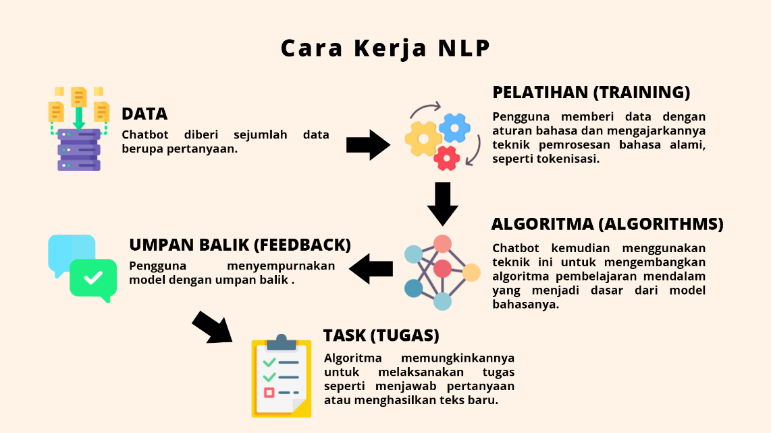
1. Named Entity Recognition (NER)

Pengenalan entitas dalam teks membantu chatbot mengenali informasi penting seperti nama orang, lokasi, tanggal, dan istilah khusus lainnya yang relevan dalam konteks layanan Ayodya Pala.

1. Klasifikasi Teks

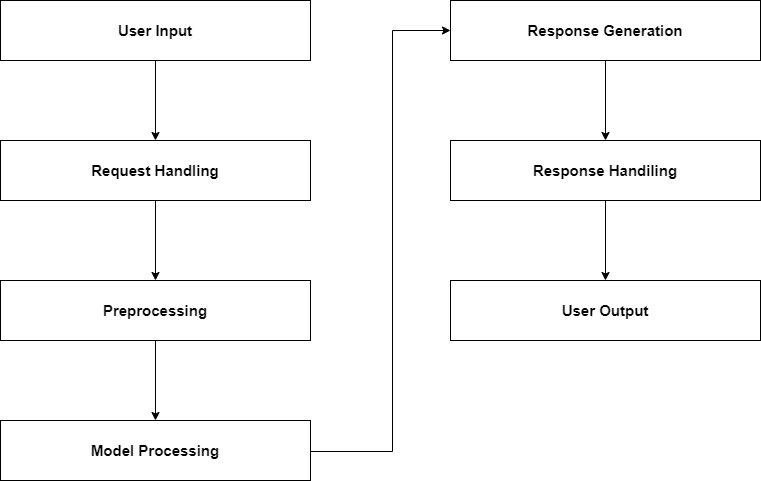
Klasifikasi teks memungkinkan chatbot mengelompokkan pesan pengguna ke dalam kategori tertentu, seperti permintaan informasi, pendaftaran, jadwal latihan, dan lainnya, sehingga respons yang diberikan lebih relevan dan terarah.

Berikut adalah cara kerja metode *Natural Language Processing (NLP)* yang umumnya diterapkan pada *chatbot*:



**Gambar 2. 5 Cara kerja metode Natural Language Processing (NLP)**

Gambar di atas menunjukan cara kerja metode *NLP* yaitupengguna memberikan sejumlah data berupa pertanyaan kepada chatbot dan mengajarkannya teknik pemrosesan bahasa alami, seperti tokenisasi. *Chatbot* tersebut kemudian menggunakan teknik-teknik ini untuk mengembangkan algoritma pembelajaran mendalam yang menjadi dasar model bahasanya. Pengguna menyempurnakan model ini dengan umpan balik penguatan, sehingga algoritma tersebut mampu melakukan tugas-tugas seperti menjawab pertanyaan atau menghasilkan teks baru.



**Gambar 2. 6 Cara kerja Natural Language Processing (NLP)**

Berikut penjelasan dari gambar 2.6 cara kerja *Natural Language Processing* *(NLP)* & *API ChatGPT*, sebagai berikut:

1. Pengguna mengirim pertanyaan atau perintah melalui antarmuka aplikasi yang menggunakan *API ChatGPT*.
2. *API* menerima masukan pengguna dan menjalankan pra-pemrosesan teks, seperti tokenisasi, normalisasi, dan penghapusan kata-kata umum *(stopwords)* jika diperlukan.
3. Teks yang telah dipra-pemroses mungkin melalui langkah-langkah ekstraksi fitur tambahan untuk memahami konteks lebih baik, meskipun dalam kasus *API ChatGPT*, sebagian besar proses ini sudah terintegrasi dalam model.
4. Teks yang telah diproses dikirim ke model *ChatGPT*. Model ini menggunakan teknik pembelajaran mendalam berbasis Transformer untuk memahami teks masukan dan konteks serta maksud dari pengguna. *ChatGPT* telah dilatih menggunakan sejumlah besar data teks untuk menghasilkan respons yang relevan.
5. Model *ChatGPT* menghasilkan respons berdasarkan analisis teks masukan. Proses tersebut mencakup pengubahan dari representasi internal model menjadi teks yang dapat dibaca oleh manusia.
6. Respons yang dihasilkan oleh model dikembalikan oleh *API* ke aplikasi yang mengirimkan permintaan. Aplikasi dapat melakukan pemrosesan lebih lanjut pada respons sebelum menampilkannya ke pengguna, seperti penyesuaian format atau integrasi dengan data lain.

Respons yang telah diproses ditampilkan kepada pengguna melalui antarmuka aplikasi. Pengguna dapat melanjutkan interaksi dengan aplikasi berdasarkan respons yang diterima.

### *Telegram*



**Gambar 2. 7 Telegram**

Teknologi komunikasi telah banyak berubah selama 20 tahun terakhir. Dari dulu yang awalnya di *handphone* hanya ada sms dan telepon sekarang sudah ada aplikasi *instant messaging* yang sudah mengakomodir keduanya bahkan bisa untuk *video call* dan masih banyak fitur lain. Saat ini telah banyak aplikasi pesan instan / *instant messaging* (IM) seperti *Whatsapp, Line, Snapchat, Facebook Messenger* dan *Telegram. Telegram* merupakan satu-satunya aplikasi pesan singkat yang menyediakan api bagi pengguna untuk dapat membuat *bot* yang bisa dimanfaatkan untuk sistem informasi (Khairunisa & Hidayat, 2023).

*Telegram* adalah suatu aplikasi pesan singkat layakannya aplikasi pesan singkat lainnya seperti *Whatsapp, Line*, dan lain lain. Aplikasi *Telegram* merupakan aplikasi *multi-platform* dalam artian aplikasi *Telegram* dapat berjalan di berbagai system operasi seperti *Windows, Android, MacOS X, Linux*. *Telegram* merupakan perusahan yang dikembangkan oleh *Telegram* *Messanger* LLP didukung oleh wirausahawan asal Rusia bernama Pavel Durov (Febriansyah & Nirmala, 2023).

### *Bot* *Telegram – Fatherbot*



**Gambar 2. 8 Bot Telegram – Fatherbot**

*Bot* pada aplikasi *chatting* *telegram* merupakan fitur *third party* yang berjalan dalam *telegram*. Pengguna dapat berinteraksi dengan *bot* dengan mengirim pesan, juga bisa membalas sesuai *request* otomatis. *Telegram* *bot* juga dapat mencari *username* pengguna yang ditarget serta menemukan kontak yang saling berkaitan (Kurnia Kasanova dkk., 2021).

Sebelum membuat program *Bot* *Telegram*, langkah yang harus dilakukan terlebih dahulu adalah membuat *Bot* *Telegram*, cara membuat *Bot* *Telegram* itu sendiri menggunakan *Botfather* yang tersedia di *Telegram* dengan menggunakan *username* *@Botfather* pada kolom pencarian untuk mendapatkan API yang akan digunakan dalam pembuatan *Chatbot* (Rizki Herfian & Rio Adriansyah, 2021).

*Telegram* *bot* adalah sebuah *bot* atau *robot* yang diprogam dengan berbagai perintah untuk menjalankan serangkaian instruksi yang diberikan oleh pengguna. *Bot* ini hanyalah sebuah akun *Telegram* yang dioperasikan oleh perangkat lunak yang memiliki fitur AI *(Artificial Intelegent)* (Sanaris & Suharjo, 2020).



## Flowchart Diagram

Flowchart adalah representasi visual dari langkah-langkah dan urutan operasi suatu program. Bagan alir (flowchart) sistem menggambarkan langkah-langkah yang terlibat dalam suatu sistem dengan menampilkan alat-alat media masukan dan keluaran serta jenis-jenis media yang digunakan untuk pemrosesan data dan penyimpanan perangkat lunak. Dengan menggunakan simbol-simbol tertentu, bagan alir menggambarkan serangkaian langkah dan menunjukkan bagaimana satu proses (instruksi) berhubungan dengan proses-proses lain dalam suatu program (Burhanuddin & Sukirman, 2024).

**Tabel 2. 2 Simbol – Simbol Flowchart Diagram**

|  |  |  |
| --- | --- | --- |
| **Simbol** | **Nama** | **Keterangan** |
|  | *Arrows* | Penghubung yang menunjukkan hubungan antara bentuk-bentuk representatif. |
|  | *Start/end* | Oval melambangkan titik awal atau akhir. |
|  | *Input/Output* | Jajar genjang melambangkan *input* atau *output* (*READ, WRITE*). |
|  | *Process* | Persegi panjang melambangkan sebuah proses (Menyatakan *assignment* *statement*). |
|  | *Decision* | Belah ketupat mengindikasikan sebuah keputusan (*True, False*). |

## Unified Modelling Language (UML)

*Unified Modeling Language* (UML) adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membanngun perangkat lunak. UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem (Setiaji dkk., 2024).



### *Use Case Diagram*

*Use Case* *Diagram* adalah teknik untuk merekam persyaratan fungsional sebuah sistem. *Use Case* mendeskripsikan interaksi tipikal antara para pengguna sistem dengan sistem itu sendiri, dengan memberi sebuah narasi tentang bagaimana sistem tersebut digunakan (Wahyu Ramdany dkk., 2024).

**Tabel 2. 3 Simbol – Simbol Use Case Diagram**

|  |  |  |
| --- | --- | --- |
| **Simbol** | **Nama** | **Deskripsi** |
|  | *Use Case* | *Use case* menggambarkan deskripsi atau skenario dari fungsionalitas yang dapat dilakukan oleh sistem dari sudut pandang pengguna. |
|  | *Actor*/Aktor | *Actor* atau Aktor adalah entitas atau elemen di luar sistem yang berinteraksi dengan sistem. |
|  | *Boundary System* | *Boundary System* adalah garis kotak yang mengelilingi kumpulan *use case* untuk menunjukkan batas sistem di mana *kumpulan use case* tersebut beroperasi. |
|  | *Association* / hubungan | *Association* adalah hubungan antara dua atau lebih *use case* dalam suatu sistem yang memiliki ketergantungan satu sama lainnya. |
|  | *Include* | *Include* menggambarkan situasi di mana fungsionalitas dari suatu *use case* selalu diikutsertakan *(included)* dalam fungsionalitas *use case* lainnya. |
|  | *Extend* | *Extend* menggambarkan situasi di mana suatu fungsionalitas tambahan dapat ditambahkan ke dalam *use case* utama berdasarkan suatu kondisi atau skenario tertentu. |
|  | *Depedency* | *Depedency* menggambarkan ketergantungan antara dua elemen. Ketergantungan ini menunjukkan bahwa satu elemen bergantung pada elemen lainnya dalam konteks tertentu. |
|  | *Generalization* | *Generalization* menggambarkan hubungan hierarki antara *use case*, di mana *use case* yang lebih umum menyediakan fungsionalitas dasar, dan *use case* yang lebih khusus mewarisi fungsionalitas tersebut serta menambahkan fungsionalitas tambahan. |
|  | *Boundary* | *Boundary* adalah representasi dalam *diagram use case* yang digunakan untuk menunjukkan batasan sistem. |

### Activity Diagram

*Activity* *Diagram* adalah pemodelan yang dilakukan pada suatu sistem dan menggambarkan aktivitas sistem berjalan. *Activity Diagram* di gunakan sebagai penjelasan aktivitas program tanpa melihat koding atau tampilan (Puturuhu, 2022).

**Tabel 2. 4 Simbol –Simbol Activity Diagram**

|  |  |  |
| --- | --- | --- |
| **Simbol** | **Nama** | **Deskripsi** |
|  | *Initial Node* | *Initial Node* merupakan titik awal dalam aliran kerja. Ini adalah langkah pertama dalam proses dan menandakan tempat di mana aliran kerja dimulai. |
|  | *Swimlane* | *Swimlane* digunakan untuk membagi aliran kerja menjadi bagian-bagian yang terkait dengan unit atau entitas tertentu. |
|  | *Activity* | *Activity* merupakan tugas yang harus dilakukan dalam aliran kerja. |
|  | *Control Flow* | *Control Flow* digunakan untuk menghubungkan *activity* dan menunjukkan urutan dalam aliran kerja. |
|  | *Decision* | *Decision* adalah titik di mana keputusan harus diambil dalam aliran kerja. |
|  | *End Node* | *End Node* adalah akhir dari aliran kerja. Setelah mencapai *End Node*, *activity* dianggap selesai. |

### Sequence Diagram

*Sequence Diagram* merupakan diagram interaksi yang menggambarkan skenario atau rangkaian langkah dari sebuah *event* untuk menghasilkan *output*. Diawali dari apa yang men-*trigger* aktivitas tersebut, proses dan perubahan yang terjadi secara internal dan output apa yang dihasilkan (Salamun, Aprialdo, dkk., 2024).

**Tabel 2. 5 Simbol – Simbol Sequence Diagram**

|  |  |  |
| --- | --- | --- |
| **Simbol** | **Nama** | **Deskripsi** |
|  | *Actor*/Aktor | Aktor adalah entitas di luar sistem yang berinteraksi dengan objek-objek dalam sistem melalui pertukaran pesan atau interaksi. |
|  | *Boundary* | *Boundary* adalah representasi antarmuka atau batasan antara sistem dan aktor. |
|  | *Control* | *Control* adalah komponen dalam sistem yang mengontrol alur eksekusi atau mengoordinasikan aktivitas sistem. |
|  | *Entity* | *Entity* adalah objek yang berisi data atau informasi dalam sistem. |
|  | *Object Lifeline* | *Object Lifeline* adalah menggambarkan seberapa lama objek atau entitas tertentu ada selama interaksi dalam sistem. *Object Lifeline*, mewakili kehidupan objek selama interaksi dan digambarkan sebagai garis vertikal yang menghubungkan objek dengan waktu. |
|  | *Activation* | *Activation* adalah kapan objek melakukan sesuatu atau menjalankan tindakan tertentu selama interaksi. |
|  | *Message* | *Message* adalah cara objek berkomunikasi satu sama lain dengan mengirim pesan yang berisi informasi tentang apa yang harus dilakukan. |
|  | *Return* | *Return* adalah kapan objek memberikan hasil atau respons setelah menerima pesan. |
|  | *Callback* | *Callback* adalah saat sistem menjalankan suatu tindakan khusus saat kondisi tertentu terpenuhi. |
|  | *Self-Call* | *Self-Call* adalah saat objek melakukan tindakan pada dirinya sendiri dengan cara memanggil metodenya sendiri. |
|  | *Frame* | *Frame* adalah menyatakan konteks dari *diagram sequence* berarti menjelaskan skenario atau alur interaksi antara aktor dan sistem dalam urutan tertentu. |

### Class Diagram

Class diagram adalah diagram yang dibuat pada tahap desain suatu perangkat lunak. Pengukuran kualitas dari desain class diagram dari software yang akan dibangun dapat mengurangi revisi-revisi yang mungkin terjadi di kemudian hari (Apriadi dkk., 2019).

**Tabel 2. 6 Simbol – Simbol Class Diagram**

|  |  |  |
| --- | --- | --- |
| **Simbol** | **Nama** | **Keterangan** |
|  | *Asosiasi* berarah */ directed association* | Hubungan di mana objek anak *(descendent)* mewarisi perilaku dan struktur data dari objek induknya *(ancestor)*. |
|  | *Class* | Sekelompok objek yang memiliki atribut dan operasi yang sama. |
|  | *Generalization*/ Pewarisan | *Generalization* digunakan untuk menunjukkan hubungan hierarki antar *class*, dengan *child class* mewarisi atribut dan metode dari *parent class*. Digambarkan sebagai garis lurus dengan segitiga putih di ujungnya, mengarah ke *parent class*. |
|  | *Aggregation*/ Agregasi | *Aggregation* digunakan untuk menunjukkan hubungan “bagian-dari” di mana satu kelas terdiri dari objek-objek *class* lain. Digambarkan sebagai garis dengan berlian putih di ujungnya. |
|  | *Composition*/ Komposisi | Mirip dengan *aggregation* tetapi hubungan ini lebih kuat; objek bagian tidak dapat berdiri sendiri tanpa objek utama. Digambarkan dengan berlian hitam di ujung garis. |
|  | *Depedency*/ Ketergantungan | *Depedency* digunakan untuk menunjukkan bahwa satu *class* bergantung pada *class* lain (menggunakan *class* lain dalam operasi). Diwakili oleh garis putus-putus dengan panah mengarah ke kelas yang menjadi tumpuan. |



## Pengujian Sistem

*Software* *testing* adalah proses identifikasi celah atau kesalahan *(bug)* dalam setiap komponen perangkat lunak dan pencatatan hasilnya. Proses ini melibatkan evaluasi menyeluruh dari setiap aspek komponen untuk memastikan bahwa perangkat lunak yang sedang dikembangkan berfungsi dengan baik. Tujuan utama dari pengujian perangkat lunak adalah untuk mendeteksi kesalahan atau cacat agar perangkat lunak dapat bekerja dengan optimal (Agustian & Yuliana, 2024).

### Pengujian *Black box Testing*

*Black box Testing* merupakan suatu pendekatan yang digunakan dalam pengujian perangkat lunak tanpa memerinci detail perangkat lunak. Dalam pengujian ini, hanya dilakukan pengecekan terhadap hasil keluaran berdasarkan nilai masukan yang diberikan, tanpa melakukan usaha untuk mengetahui kode program yang digunakan untuk menghasilkan output tersebut (M. Putri dkk., 2024).

### Pengujian *White box Testing*

White box testing merupakan metode yang menguji struktur internal perangkat lunak, rancangan dan kode program perangkat lunak terkait. White Box dapat mengungkapkan kesalahan dalam implementasi dari sebuah perangkat lunak. Penguji yang menggunakan metode white box dalam pengujian perangkat lunak harus memiliki pengetahuan atau pemahaman penuh mengenai sumber kode perangkat lunak (Bryan L Sie dkk., 2022).



## Perangkat Lunak Pendukung

Perangkat lunak yang mendukung pembuatan *chatbot* melibatkan elemen-elemen kunci atau komponen penting yang diperlukan. Beberapa di antaranya meliputi:



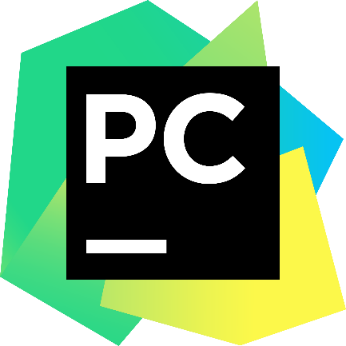
### Bahasa Pemrograman *Python*

****

**Gambar 2. 9 Bahasa Pemrograman Python**

Dengan penekanan pada keterbacaan kode sebagai filosofi desainnya, Python adalah bahasa pemrograman interpretatif yang memiliki beberapa tujuan. Python memiliki perpustakaan standar yang besar dan luas, sintaksis yang mudah dipahami, dan serangkaian karakteristik yang menjadikannya bahasa yang kuat (Prasojo dkk., 2024).

### *Pycharm*



**Gambar 2. 10 Pycharm**

*Pycharm* adalah salah satu *IDE* yang ada dalam *Python*. *Integrated Development Environment* atau *IDE* dilengkapi editor kode dan kompiler untuk menulis dan menyusun program dalam satu atau banyak bahasa pemrograman. Selain itu, *IDE* hadir dengan banyak fitur yang memfasilitasi pengembangan perangkat lunak yang komprehensif. Karena *IDE* mengalokasikan warna yang berbeda ke entitas pemrograman yang berbeda, biasanya dikenal sebagai penyorotan sintaks, itu menjadi lebih mudah diakses (Putra, 2022).

### *Draw.io*

******

**Gambar 2. 11 Draw.io**

*Draw.io* adalah sebuah perangkat lunak yang digunakan untuk simulasi jaringan. *Draw.io* merupakan perangkat lunak simulasi jaringan yang berfungsi untuk membuat versi dari suatu jaringan komputer dan melakukan simulasi terhadap jaringan tersebut. *Draw.io* menawarkan fitur simulasi, tata letak, dan kolaborasi, serta memudahkan dalam pembuatan konsep teknologi yang kompleks (Noneng Marthiawati dkk., 2024).

### *Replit*



**Gambar 2. 12 Replit**

Replit adalah *platform* pengembangan *web* berbasis *cloud* yang memungkinkan pengguna untuk membuat, menyimpan, dan menjalankan proyek-proyek pengembangan *web* langsung dari browser web mereka, tanpa perlu menginstal perangkat lunak tambahan (Melissa, 2024).

### *User Response* (Kuesioner)

Angket atau kuesioner adalah alat yang digunakan untuk mengumpulkan data melalui serangkaian pertanyaan yang telah dirancang dengan tujuan mengukur variabel penelitian (Ardiansyah dkk., 2023).

Kuesioner merupakan alat atau instrumen yang digunakan untuk menilai atau mengukur suatu peristiwa atau kejadian (Nur Amalia dkk., 2022).

# BAB III ANALISA DAN PERANCANGAN



## Analisa Kebutuhan

Penelitian ini menggunakan pendekatan rekayasa perangkat lunak karena berfokus pada perancangan dan implementasi sistem chatbot untuk layanan informasi. Tujuannya adalah menghasilkan perangkat lunak (chatbot) yang mampu memproses bahasa alami (Natural Language Processing) dan memberikan respons otomatis kepada pengguna melalui platform Telegram.

Analisa Kebutuhan pengguna dan pemangku kepentingan dalam implementasi Chatbot dengan memanfaatkan API ChatGPT melalui Telegram menggunakan Metode Natural Language Processing (NLP). Analisa kebutuhan ini menjadi landasan untuk merancang Chatbot yang responsif dan sesuai dengan harapan pengguna.

Kebutuhan yang dimaksud adalah memahami jenis pertanyaan atau permintaan yang paling sering diajukan oleh pengguna kepada Chatbot. penulis perlu memahami preferensi pengguna terkait dengan bahasa dan gaya komunikasi yang pengguna sukai. Selain itu, penulis harus mengetahui sejauh mana pengguna mengharapkan Chatbot dapat memberikan informasi yang relevan.

Selain itu, kebutuhan kemudahan penggunaan menjadi fokus utama. penulis harus memahami tingkat pengetahuan dan keterampilan pengguna dalam menggunakan teknologi dan Chatbot. Identifikasi hambatan atau kendala yang mungkin dihadapi oleh pengguna dalam berinteraksi dengan Chatbot juga menjadi bagian penting dalam analisa ini. Penulis juga harus memahami preferensi pengguna terkait dengan tampilan dan navigasi antarmuka pengguna.



## Analisa Sistem

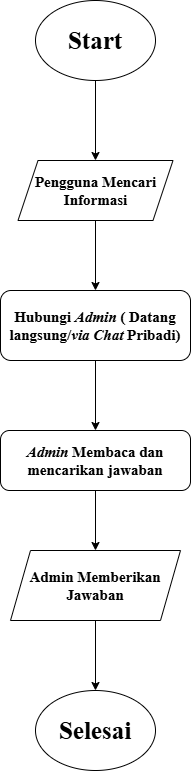


### Analisa Sistem Berjalan

Pada analisa sistem terdahulu untuk mengetahui cara kerja, kelebihan, dan kekurangan sistem. Analisis sistem terdahulu dapat menentukan apakah kebaruan sistem yang dibuat dapat menutupi kekurangan sistem sebelumnya, serta menjadi acuan untuk menemukan masalah sehingga diangkat menjadi penelitian ini.

Sistem layanan informasi yang saat ini digunakan oleh Sanggar masih bersifat manual dan belum terintegrasi dengan sistem otomatisasi. Penyampaian informasi kepada pengguna, baik pimpinan cabang maupun siswa Sanggar Ayodya Pala, masih sangat bergantung pada interaksi langsung dengan petugas atau admin. Proses komunikasi dilakukan melalui beberapa cara tradisional, antara lain pertemuan langsung secara tatap muka di ruang administrasi, pengiriman pesan pribadi melalui aplikasi WhatsApp atau Telegram tanpa menggunakan bantuan bot otomatis, serta penyampaian informasi melalui media pengumuman manual, seperti papan informasi fisik atau grup media sosial. Kondisi ini mengakibatkan penyampaian informasi menjadi kurang efisien, lambat, dan berpotensi menimbulkan miskomunikasi apabila informasi tidak diterima secara merata oleh semua pengguna.

Sistem layanan informasi yang berjalan saat ini memiliki sejumlah kendala yang berdampak pada efektivitas dan efisiensi penyampaian informasi kepada pengguna. Salah satu permasalahan utama adalah keterbatasan waktu pelayanan, di mana admin hanya dapat memberikan layanan selama jam kerja, sementara pengguna bisa membutuhkan informasi kapan saja, termasuk di luar jam operasional. Selain itu, respons yang diberikan oleh admin cenderung tidak konsisten karena tidak adanya standarisasi jawaban, informasi yang diterima pengguna bisa berbeda tergantung siapa yang membalas pertanyaan. Permasalahan lain yang kerap terjadi adalah penumpukan antrian pertanyaan. Ketika banyak pengguna menghubungi secara bersamaan, admin sering kali kewalahan dalam menangani seluruh permintaan, sehingga menimbulkan keterlambatan dalam pelayanan. Hal ini diperparah dengan penyampaian informasi yang masih dilakukan secara manual, seperti melalui papan pengumuman atau media sosial, yang menyebabkan informasi tidak selalu diterima secara tepat waktu oleh seluruh pengguna.

****

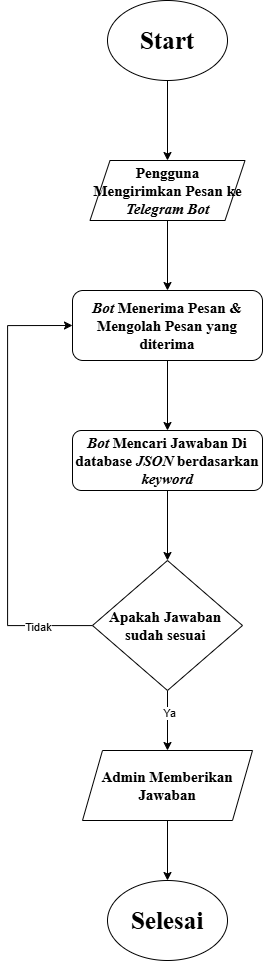
**Gambar 3. 1 Analisa Sistem Berjalan**

Berikut ini adalah penjelasan pada Gambar 3.1 Analisa Sistem Berjalan di atas:

1. Pengguna yang membutuhkan informasi harus menghubungi admin.
2. Pengguna harus datang langsung ke tempat administrasi maupun melalui pesan pribadi menggunakan aplikasi *chat* seperti *WhatsApp*.
3. *Admin* akan membaca pertanyaan yang diajukan dan mencari jawaban yang sesuai.
4. *Admin* memberikan jawaban tersebut dan disampaikan kembali kepada pengguna melalui media yang sama.

### Analisa Sistem Usulan

Analisa Sistem usulan baru yang dirancang merupakan *chatbot* layanan informasi otomatis yang diimplementasikan melalui platform *Telegram*. Sistem ini dibangun menggunakan bahasa pemrograman *Python* dengan diolah oleh aplikasi *pycharm* dan dukungan metode *Natural Language Processing* (NLP). Tujuan utama pengembangan sistem ini adalah untuk menggantikan sebagian besar proses interaksi manual antara pengguna dan *admin*, sehingga menjadi lebih responsif, cepat, dan memiliki standar jawaban yang konsisten.



**Gambar 3. 2 Analisa Sistem Usulan**

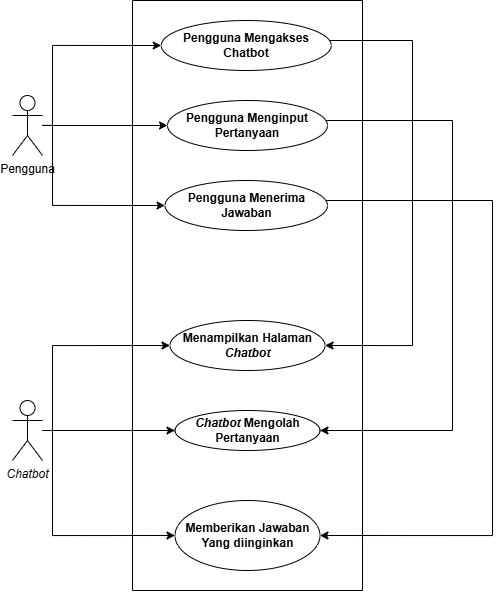
Di bawah ini merupakan penjelasan dari Analisa Sistem Usulan pada Gambar di atas:

1. Pengguna mengirimkan pertanyaan kepada sistem melalui aplikasi *Telegram*.
2. Pertanyaan tersebut diterima oleh *chatbot* (sistem) dan kemudian dianalisis menggunakan metode Natural Language Processing (NLP).
3. Sistem mengidentifikasi kata kunci dari teks pertanyaan untuk memahami maksud pengguna.
4. Sistem mencocokkan hasil analisis tersebut dengan data jawaban yang tersimpan dalam database.
5. *Chatbot* memberikan jawaban secara otomatis kepada pengguna berdasarkan kecocokan data.
6. *Admin* secara berkala melakukan pembaruan terhadap database jawaban, baik dengan menambahkan entri baru maupun mengedit data yang sudah ada.
7. Pembaruan database dilakukan melalui file *JSON* atau antarmuka pengelolaan yang telah disediakan.

## Perancangan *Unified Modelling Language (UML)*



### Use Case Diagram

****

**Gambar 3. 3 Use Case Diagram Chatbot**

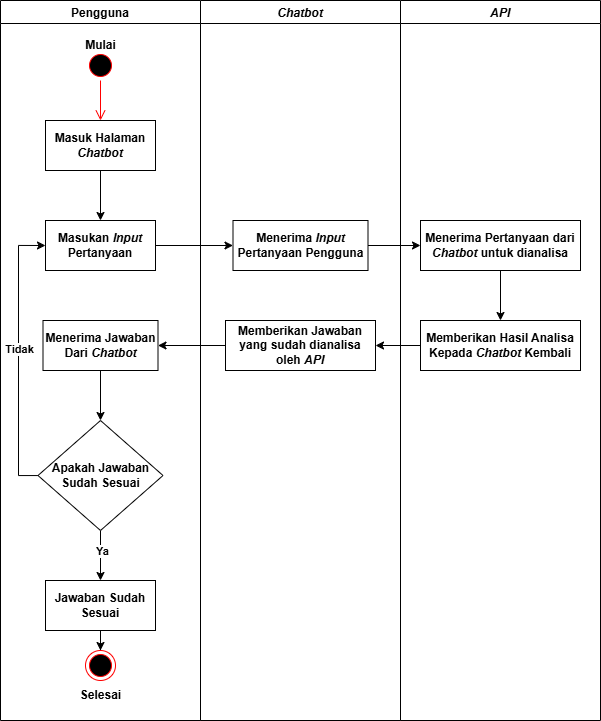
Di bawah ini merupakan penjelasan dari *Use Case Diagram Chatbot* pada Gambar di atas:

1. *User* atau pengguna merupakan aktor yang berinteraksi langsung dengan *Chatbot*. Peran *user* meliputi memasukkan pertanyaan dalam bentuk teks, menerima jawaban dari *Chatbot* dalam format teks maupun suara, serta memberikan kritik, saran, atau masukan berdasarkan pengalaman penggunaan *Chatbot*. Interaksi yang dilakukan user menjadi bagian penting dalam evaluasi kinerja sistem.
2. *Chatbot* merupakan aktor sistem yang secara otomatis merespons input dari pengguna. *Chatbot* memproses pertanyaan menggunakan teknologi Pemrosesan Bahasa Alami (NLP) untuk menghasilkan jawaban yang relevan dalam bentuk teks dan suara. Selain itu, *Chatbot* juga dapat memberikan tanggapan atau saran tambahan yang bersifat informatif, disesuaikan dengan konteks pertanyaan yang diajukan oleh pengguna.

### Activity Diagram

Diagram aktivitas *(activity diagram)* adalah sebuah diagram dalam *Unified Modeling Language* (UML) yang digunakan untuk menggambarkan aliran aktivitas atau proses dalam suatu sistem. Ketika membuat diagram aktivitas untuk *chatbot*, Anda dapat menggambarkan bagaimana *chatbot* berinteraksi dengan pengguna dan menjalankan berbagai tugas. *Activity Diagram* memperlihatkan aliran proses dari suatu aktivitas ke aktivitas lainnya dalam sebuah sistem yang dibangun. *Diagram* ini lebih menekankan penggambaran proses dan jalur aktivitas secara umum (Salamun, Aldi Aprialdo, dkk., 2024).

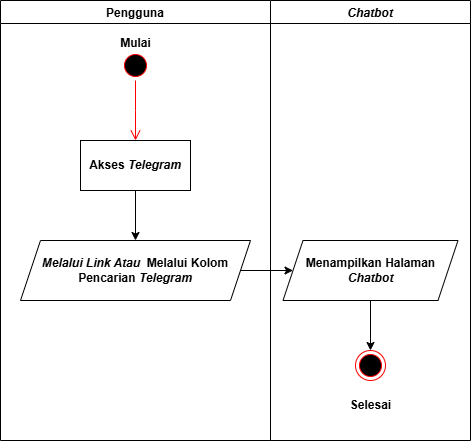
1. *Activity Diagram* *Testing* *Chatbot*



**Gambar 3. 4 Activity Diagram Testing Chatbot**

Pada gambar di atas ditampilkan *Activity Diagram* yang menjelaskan alur proses pengujian *chatbot* layanan informasi Sanggar Ayodya Pala menggunakan metode *Natural Language Processing* (NLP) melalui integrasi *API ChatGPT*. Proses ini menggambarkan interaksi antara pengguna dengan sistem *chatbot*, mulai dari *input* hingga verifikasi jawaban yang diberikan. Proses dimulai ketika pengguna membuka halaman *chatbot* yang telah diintegrasikan dalam aplikasi *Telegram*. Setelah berhasil mengakses antarmuka, pengguna kemudian memasukkan pertanyaan yang ingin diajukan kepada *chatbot*. Pertanyaan ini dapat berupa informasi umum seputar layanan sanggar, jadwal kegiatan, atau informasi pendaftaran. Selanjutnya, *chatbot* akan menerima input dari pengguna dan meneruskan pertanyaan tersebut ke *API ChatGPT* melalui permintaan *HTTP*. *API* ini kemudian menganalisis konteks dari pertanyaan menggunakan teknologi NLP dan mengembalikan hasil jawaban yang relevan. *Chatbot* kemudian menerima hasil analisis tersebut dan menampilkannya kembali kepada pengguna melalui *platform Telegram*. Langkah terakhir dalam proses ini adalah verifikasi jawaban oleh pengguna. Apabila jawaban yang diberikan oleh *chatbot* dianggap sesuai dengan kebutuhan pengguna, maka interaksi dinyatakan selesai. Namun, jika jawaban belum sesuai, pengguna diberi kebebasan untuk mengajukan pertanyaan ulang yang lebih spesifik. Hal ini bertujuan untuk memperbaiki konteks agar sistem dapat memberikan jawaban yang lebih akurat. Dengan adanya proses ini, dapat dilakukan evaluasi terhadap kemampuan *chatbot* dalam memahami serta merespons berbagai jenis pertanyaan, sehingga menjadi bagian penting dalam pengujian fungsionalitas sistem sebelum diimplementasikan secara penuh kepada pengguna akhir.

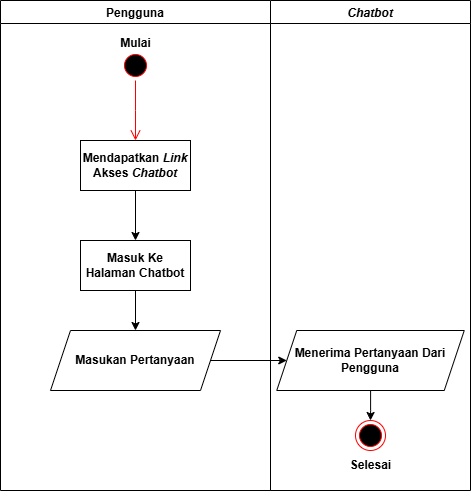
1. *Activity Diagram Link Akses*

******

**Gambar 3. 5 Activity Diagram Link Akses**

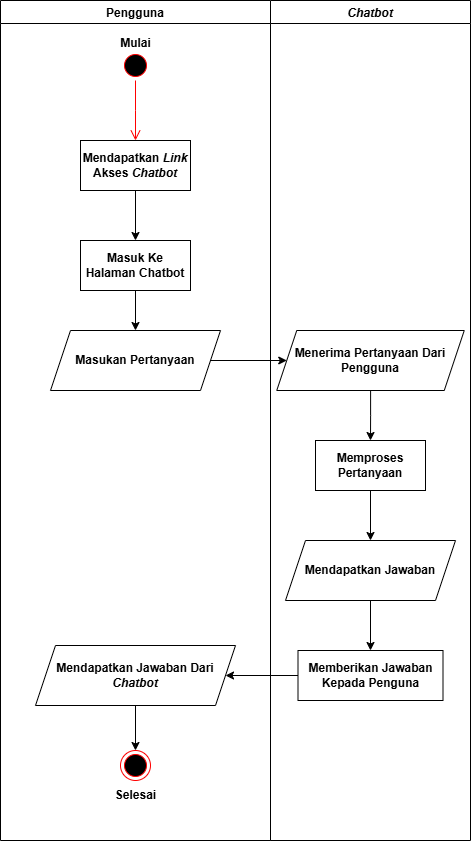
Gambar di atas menggambarkan proses awal interaksi antara pengguna dan sistem *chatbot* melalui *platform* *Telegram* sebagai bagian dari layanan informasi Sanggar Ayodya Pala. Proses dimulai saat pengguna membuka aplikasi *Telegram* (Akses *Telegram*) pada perangkat masing-masing, baik itu *smartphone* maupun komputer. Setelah berhasil mengakses aplikasi, pengguna dapat menuju *chatbot* dengan dua metode, yaitu melalui tautan langsung yang dibagikan oleh pihak pengelola, atau dengan mencari nama *chatbot* secara manual melalui kolom pencarian *Telegram*. Langkah tersebut akan mengarahkan pengguna ke tampilan awal dari sistem *chatbot*. Pada titik ini, sistem akan secara otomatis menampilkan halaman utama *chatbot*, di mana antarmuka percakapan tersedia dan siap digunakan. Dengan demikian, *chatbot* telah berada dalam kondisi aktif dan siap menerima *input* berupa pertanyaan atau permintaan informasi dari pengguna. Tahapan ini menjadi dasar dari proses komunikasi selanjutnya antara pengguna dan sistem, yang akan dilanjutkan dengan interaksi berbasis teks secara otomatis.

1. *Activity Diagram* Menerima Pertanyaan

****

**Gambar 3. 6 Activity Diagram Menerima Pertanyaan**

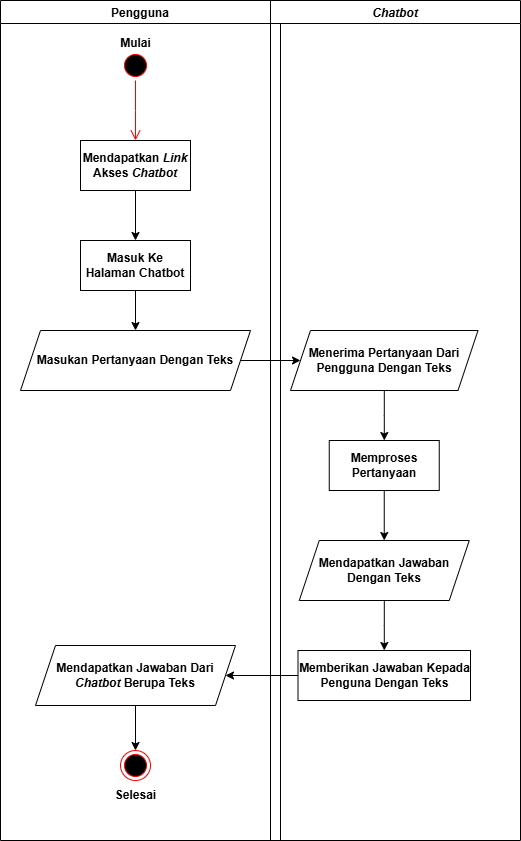
1. *Activity Diagram* Menerima Jawaban

****

**Gambar 3. 7 Activity Diagram Menerima Jawaban**

Gambar 3.7 menunjukkan alur interaksi lengkap antara pengguna dan sistem *chatbot* dalam layanan informasi Sanggar Ayodya Pala. Proses dimulai ketika pengguna memperoleh tautan akses menuju *chatbot*, baik melalui *link* yang dibagikan secara langsung maupun dengan mencari nama *chatbot* pada kolom pencarian *Telegram*. Setelah itu, pengguna masuk ke halaman utama *chatbot* yang telah disediakan. Setelah berhasil masuk, pengguna dapat langsung menginputkan pertanyaan terkait informasi yang ingin diperoleh, seperti jadwal latihan, lokasi sanggar, atau informasi pendaftaran. *Chatbot* kemudian menerima *input* tersebut dan langsung memproses pertanyaan pengguna menggunakan metode *Natural Language Processing* (NLP) guna memahami maksud dan konteks dari kalimat yang dikirimkan. Setelah proses analisis selesai, *chatbot* akan mencari jawaban yang paling relevan sesuai dengan basis data atau aturan yang telah ditentukan sebelumnya. Jawaban tersebut kemudian dikirimkan kembali kepada pengguna melalui tampilan antarmuka *chatbot*. Selanjutnya, pengguna menerima respon atau jawaban dari *chatbot*, yang menandakan bahwa satu siklus interaksi telah berhasil diselesaikan. Proses ini dapat diulang secara berkelanjutan selama pengguna masih ingin berinteraksi dengan sistem, tanpa perlu campur tangan operator manusia. Diagram ini menegaskan bahwa sistem chatbot mampu memberikan layanan informasi secara otomatis, efisien, dan responsif.

1. *Activity Diagram* Teks

****

**Gambar 3. 8 Activity Diagram Teks**

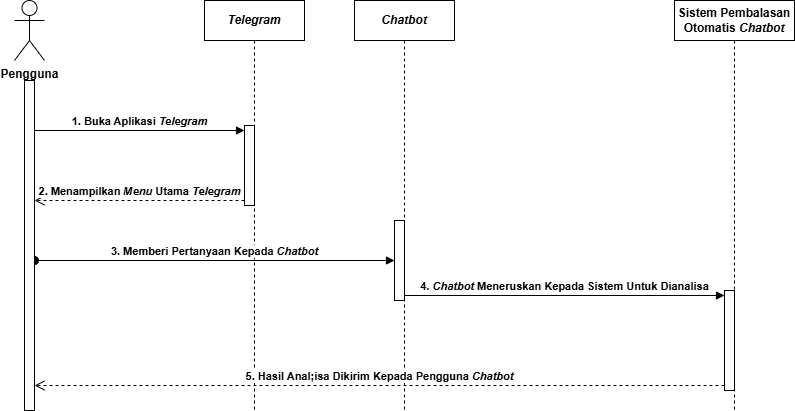
Gambar 3.8 menjelaskan alur komunikasi berbasis teks antara pengguna dan sistem *chatbot* layanan informasi Sanggar Ayodya Pala. Proses dimulai ketika pengguna memperoleh *link* akses menuju *chatbot*, baik dari tautan langsung maupun hasil pencarian di aplikasi *Telegram*. Setelah berhasil mengakses dan masuk ke halaman utama *chatbot*, pengguna kemudian mengetikkan pertanyaan dalam bentuk teks yang berkaitan dengan informasi yang dibutuhkan, seperti jadwal kegiatan, informasi pendaftaran, atau detail lokasi sanggar. *Chatbot* menerima pertanyaan tersebut sebagai input dalam bentuk teks, kemudian memproses pertanyaan tersebut menggunakan teknik pemrosesan bahasa alami (*Natural Language Processing*/NLP). Proses ini bertujuan untuk memahami konteks dan maksud dari kalimat yang dikirimkan oleh pengguna. Setelah proses analisis selesai, *chatbot* akan menghasilkan respons yang sesuai, juga dalam bentuk teks. Respons tersebut kemudian dikirimkan kembali kepada pengguna melalui antarmuka percakapan yang tersedia di *Telegram*. Akhir dari proses ini ditandai dengan pengguna menerima jawaban dari *chatbot* dalam bentuk teks, menandakan bahwa siklus interaksi telah berjalan dengan baik dan sesuai dengan tujuan layanan informasi otomatis.



### Sequence Diagram

*Sequence diagram* adalah diagram UML yang menunjukkan alur interaksi antara berbagai objek dalam sebuah sistem. Diagram ini menggambarkan urutan langkah-langkah atau pesan yang ditransmisikan di antara objek-objek, membantu kita memahami bagaimana berbagai proses dalam sistem saling berintegrasi. Diagram urutan mendukung pengenalan dan pemetaan skenario pengguna, memberikan pemahaman tentang cara sistem beroperasi, serta mengungkap kebutuhan fungsional. Dengan mengikuti alur pesan yang bergerak antar objek, kita dapat mengamati penerapan suatu fitur atau fungsi. Diagram ini sangat bermanfaat dalam desain sistem, membantu tim untuk memahami interaksi di antara komponen dan meningkatkan keselarasan pemahaman mereka. Diagram urutan juga dapat berfungsi sebagai landasan untuk implementasi kode, memastikan bahwa objek berinteraksi dengan tepat.Dengan representasi urutan interaksi yang jelas, diagram urutan memudahkan komunikasi antara tim teknis dan bisnis, serta membantu mengidentifikasi masalah potensial sejak awal. Secara keseluruhan, diagram urutan adalah alat pemodelan yang penting untuk memahami dan merekam dinamika dalam sistem (A. R. Putri dkk., 2025).

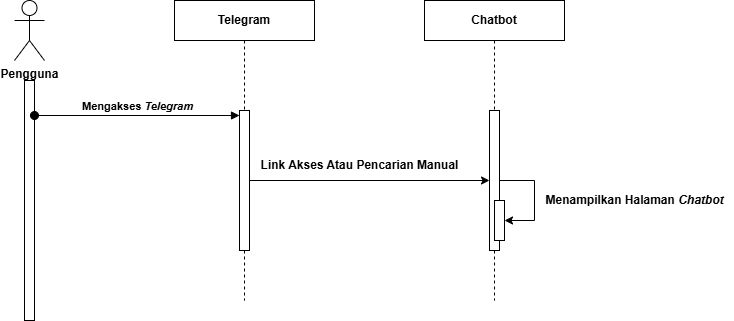
1. *Sequence Diagram Testing Chatbot*

****

**Gambar 3. 9 Sequence Diagram Testing Chatbot**

Gambar di atas menunjukkan diagram urutan interaksi antara pengguna, aplikasi *Telegram, chatbot*, dan sistem pembalasan otomatis dalam layanan informasi Sanggar Ayodya Pala. Proses dimulai ketika pengguna membuka aplikasi *Telegram*. Setelah aplikasi berhasil dijalankan, sistem *Telegram* akan secara otomatis menampilkan menu utama *Telegram* yang memungkinkan pengguna mengakses berbagai fitur, termasuk *chatbot*. Selanjutnya, pengguna melakukan pengiriman pertanyaan kepada *chatbot*, yang merupakan inti dari proses interaksi. Setelah menerima input dari pengguna, *chatbot* akan meneruskan pertanyaan tersebut ke sistem pembalasan otomatis untuk dianalisis menggunakan metode *Natural Language Processing*. Setelah proses analisis selesai dan jawaban berhasil dihasilkan oleh sistem, hasil analisis dikirimkan kembali kepada pengguna melalui *chatbot*. Diagram ini menunjukkan bahwa *chatbot* hanya berperan sebagai perantara antara pengguna dan sistem analisis, sedangkan pemrosesan pertanyaan secara cerdas dilakukan oleh sistem pembalasan otomatis yang telah ditanamkan model NLP di dalamnya.

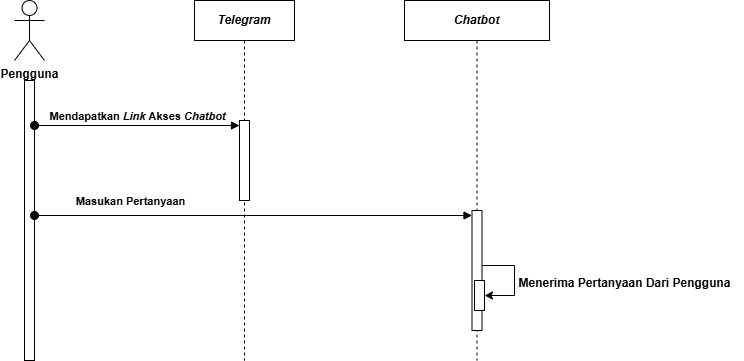
1. *Sequence Diagram Link* Akses

****

**Gambar 3. 10 Sequence Diagram Link Akses**

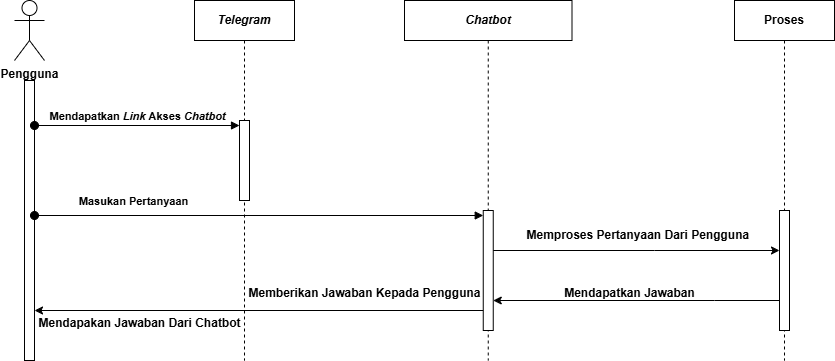
Gambar di atas menunjukkan urutan proses awal interaksi antara pengguna dan sistem *chatbot* melalui aplikasi *Telegram*. Proses dimulai ketika pengguna membuka aplikasi *Telegram* sebagai *platform* utama dalam mengakses layanan *chatbot*. Setelah *Telegram* berhasil diakses, pengguna dapat menemukan *chatbot* dengan dua cara, yaitu melalui tautan langsung (*link* akses) yang telah dibagikan sebelumnya oleh pengelola sistem, atau melalui pencarian manual di kolom pencarian *Telegram* dengan mengetikkan nama *chatbot* yang ditentukan. Permintaan akses dari pengguna kemudian diteruskan oleh *Telegram* ke sistem *Chatbot*, yang akan memproses permintaan tersebut dan menampilkan halaman utama *chatbot* kepada pengguna. Tampilan ini menandai kesiapan sistem untuk mulai menerima input berupa pertanyaan atau perintah dari pengguna, sekaligus menjadi titik awal dari proses interaksi berbasis teks antara pengguna dan *chatbot*.

1. *Sequence Diagram* Menerima Pertanyaan

****

**Gambar 3. 11 Sequence Diagram Menerima Pertanyaan**

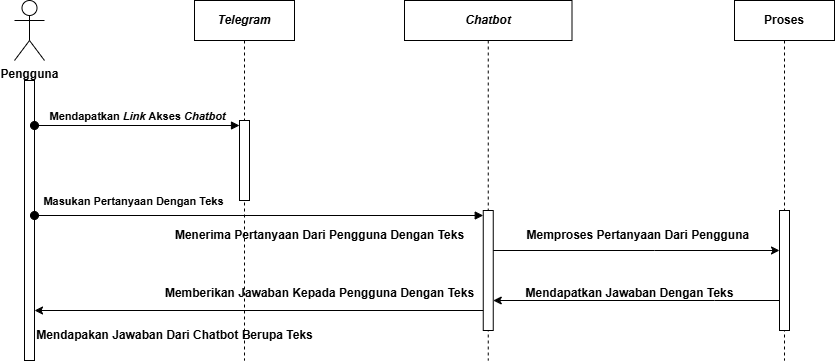
1. *Sequence Diagram* Menerima Jawaban

****

**Gambar 3. 12 Sequence Diagram Menerima Jawaban**

Gambar di atas menunjukkan urutan proses lengkap interaksi antara pengguna dan sistem *chatbot* melalui aplikasi *Telegram*, mulai dari akses awal hingga pengguna menerima jawaban dari *chatbot*. Proses diawali ketika pengguna membuka aplikasi *Telegram* sebagai *platform* utama untuk mengakses layanan *chatbot*. Pengguna dapat menemukan *chatbot* melalui dua metode, yaitu dengan menggunakan tautan langsung (*link* akses) yang telah dibagikan sebelumnya atau melalui pencarian manual di kolom pencarian *Telegram* dengan mengetikkan nama *chatbot* yang ditentukan. Setelah mendapatkan akses ke *chatbot*, pengguna dapat langsung memasukkan pertanyaan atau instruksi melalui halaman percakapan *chatbot* di *Telegram*. Pertanyaan yang dikirimkan pengguna akan diteruskan ke sistem *Chatbot*, yang bertugas menerima, memproses, dan menghasilkan jawaban. Selanjutnya, sistem *chatbot* akan meneruskan pertanyaan tersebut ke komponen Proses, yang berfungsi untuk memproses pertanyaan dan menentukan jawaban sesuai logika yang telah ditetapkan. Setelah mendapatkan jawaban dari proses internal tersebut, *chatbot* akan mengirimkan jawaban kembali kepada pengguna melalui aplikasi *Telegram*. Terakhir, pengguna akan menerima jawaban atau respon dari *chatbot*, yang dapat berupa teks, instruksi, atau informasi sesuai dengan pertanyaan yang diajukan. Proses ini merupakan siklus interaksi berbasis teks antara pengguna dan *chatbot* yang dilakukan secara *real-time* melalui aplikasi *Telegram*.

1. *Sequence Diagram* Teks

****

**Gambar 3. 13 Sequence Diagram Teks**

Gambar di atas menunjukkan diagram urutan *(sequence diagram)* yang menggambarkan proses interaksi antara pengguna dan sistem *chatbot* berbasis teks yang diakses melalui aplikasi *Telegram. Sequence diagram* ini menjelaskan tahapan komunikasi yang terjadi mulai dari pengguna mengakses layanan hingga memperoleh jawaban dalam bentuk teks dari sistem *chatbot*. Proses diawali ketika pengguna membuka aplikasi *Telegram*, yang berfungsi sebagai *platform* utama dalam mengakses layanan *chatbot*. Pengguna dapat menemukan *chatbot* dengan dua cara, yaitu dengan mengklik tautan langsung (*link* akses) yang telah dibagikan oleh pengelola sistem, atau dengan mencari nama *chatbot* secara manual melalui fitur pencarian di aplikasi *Telegram*. Setelah mendapatkan akses ke *chatbot*, pengguna dapat langsung mengajukan pertanyaan atau memberikan perintah melalui fitur percakapan di *Telegram*. Pertanyaan yang diajukan pengguna dikirim dalam bentuk teks dan diteruskan ke sistem *chatbot* melalui *platform Telegram*. Sistem *chatbot* kemudian menerima pertanyaan dari pengguna dan memprosesnya dengan berkomunikasi ke bagian subsistem Proses, yang bertugas untuk menganalisis atau menentukan jawaban atas pertanyaan yang diajukan. Proses ini dapat melibatkan logika pemrograman, basis data, atau kecerdasan buatan tergantung pada konfigurasi *chatbot* yang digunakan. Setelah jawaban berhasil diperoleh oleh subsistem proses, jawaban tersebut dikirim kembali ke sistem *chatbot* dalam bentuk teks. *Chatbot* kemudian meneruskan jawaban kepada pengguna melalui aplikasi *Telegram*. Akhir dari proses ini ditandai dengan diterimanya jawaban oleh pengguna dalam bentuk teks di halaman percakapan *Telegram*. Dengan demikian, diagram ini menjelaskan dengan jelas bahwa komunikasi antara pengguna dan *chatbot* berlangsung secara *real-time*, berbasis teks, dan melibatkan tiga komponen utama yaitu pengguna, platform *Telegram*, dan sistem *chatbot* beserta subsistem proses yang terdapat di dalamnya.

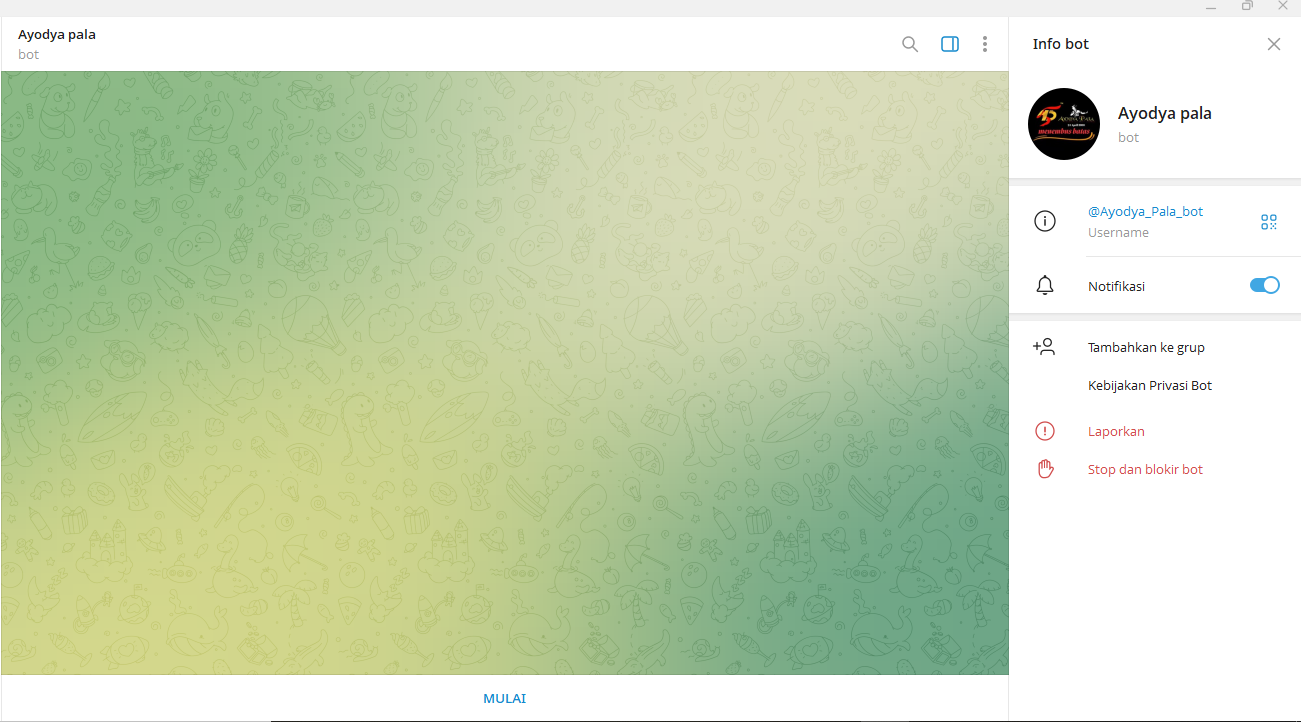
### *Class Diagram*

*Class diagram* menggambarkan hubungan antar *class* yang di dalamnya terdapat atribut dan fungsi dari suatu objek (Arianti dkk., 2022).

## Perancangan Antarmuka *(User Interface)*

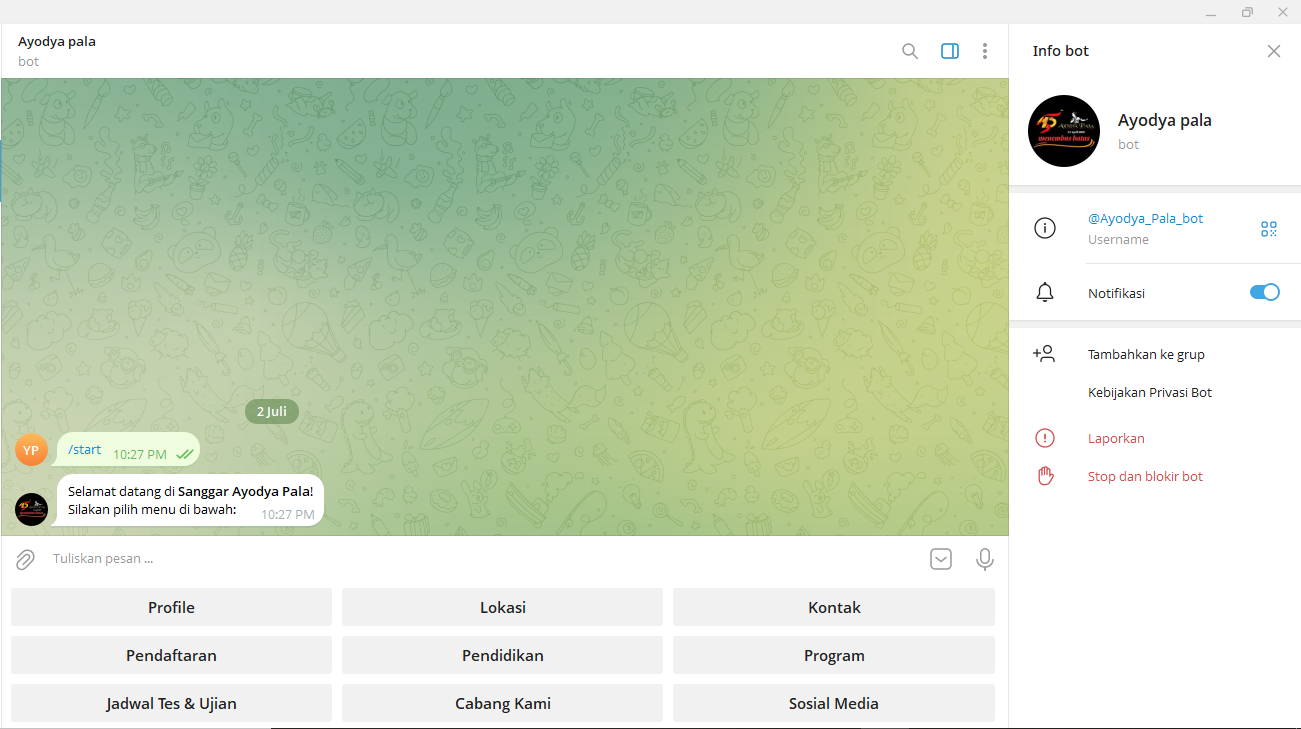
Antarmuka pengguna ialah bagian dari sebuah sistem informasi yang memerlukan interaksi pengguna guna membuat *input* serta *output*. Antarmuka pengguna menggambarkan bagaimana interaksi sistem dengan pengguna saat menggunakan sistem (interaktif) sesuai dengan kebutuhan yang sebelumnya sudah dilakukan analisis atau observasi (Ayu dkk., 2022).

1. Rancangan Halaman Awal Chatbot Pada Laptop



**Gambar 3. 14 Rancangan Halaman Awal Chatbot Pada Laptop**

1. Rancangan Halaman Awal Chatbot Saat Klik Mulai/Start Pada Laptop



**Gambar 3. 15 Rancangan Halaman Awal Chatbot Saat Klik Mulai/Start Pada Laptop**

1. Rancangan Halaman Awal Chatbot Pada Smartphone



**Gambar 3. 16 Rancangan Halaman Awal Chatbot Pada Smartphone**

1. Rancangan Halaman Awal Chatbot Saat Klik Mulai/Start Pada Smartpone



**Gambar 3. 17 Rancangan Halaman Awal Chatbot Saat Klik Mulai/Start Pada Smartpone**

Gambar – gambar di atas merupakan antarmuka *Telegram Chatbot* Layanan Informasi Ayodya Pala, tampilan tersebut adalah tampilan yang diakses melalui *laptop* atau *smartphone* pengguna dan memberikan simulasi tanya jawab pada *Chatbot*.

# BAB IV IMPLEMENTASI DAN PENGUJIAN



## Spesifikasi

Spesifikasi adalah serangkaian karakteristik atau persyaratan teknis yang menggambarkan fitur, fungsi, dan kinerja suatu sistem. Spesifikasi ini bertujuan untuk memastikan bahwa sistem tersebut memenuhi kebutuhan dan harapan mahasiswa serta standar yang ditetapkan.



### Spesifikasi Perangkat Lunak

Pengembangan perangkat lunak ini dengan menggunakan *Python, Telegram, API ChatGPT, dan Pycharm*. *Python* dipilih sebagai bahasa pemrograman utama, sementara *Telegram* sebagai pengembang perangkat lunak untuk menulis, menguji, dan menjalankan *Chatbot*. *API ChatGPT* sebagai pengembang perangkat lunak untuk berkomunikasi dengan model bahasa *GPT (Generative Pre-trained Transformer).* *Pycharm* sebagai pengembang perangkat lunak untuk menulis, menguji, dan menjalankan kode komputer. Adapun perangkat lunak yang digunakan sebagai berikut:

**Tabel 4. 1 Spesifikasi Perangkat Lunak**

|  |  |  |
| --- | --- | --- |
| **No.** | **Perangkat Lunak** | **Keterangan** |
| 1. | Sistem Operasi | *Windows 10 Pro – 64 bit* |
| 2. | *Integrated Development Environment (IDE)* | *Pycharm* |
| 3. | 3D Animasi Karakter | *Adobe Mixamo* |
| 4. | *Web Browser* | *Google Chrome* |
| 5. | Aplikasi Komunikasi | *Telegram* |

### Spesifikasi Perangkat Keras

Perangkat keras menjelaskan kebutuhan minimum perangkat keras yang diperlukan untuk mengimplementasikan program aplikasi yang dibuat. Semakin tinggi spesifikasi komputer yang digunakan untuk menjalankan aplikasi, maka akan semakin baik performa dan kecepatan respon *Chatbot* dalam berinteraksi dengan pengguna. Spesifikasi ini juga mencakup perangkat yang digunakan untuk pengembangan dan penerapan *Chatbot*. Adapun perangkat keras yang digunakan adalah sebagai berikut:

**Tabel 4. 2 Spesifikasi Perangkat Keras Laptop**

|  |  |  |
| --- | --- | --- |
| **No.** | **Perangkat Keras** | **Keterangan** |
| 1. | *Model* | *HP Laptop 14-bw0xx* |
| 2. | *Processor* | *AMD A9-9420 RADEON R5,5 COMPUTE CORES 2C+3G (2CPUs), ~3.0GHz* |
| 3. | *Video Graphics Array (VGA)* | *AMD Radeon (TM) R5 Graphics* |
| 4. | *Random Access Memory (RAM)* | *4GigaBytes* |
| 5. | Penyimpanan Internal | *MidasForce SSD 512GigaBytes* |
| 6. | Internet | *WiFi Cyberindo Aditama* |

**Tabel 4. 3 Spesifikasi Perangkat Keras Smartphone**

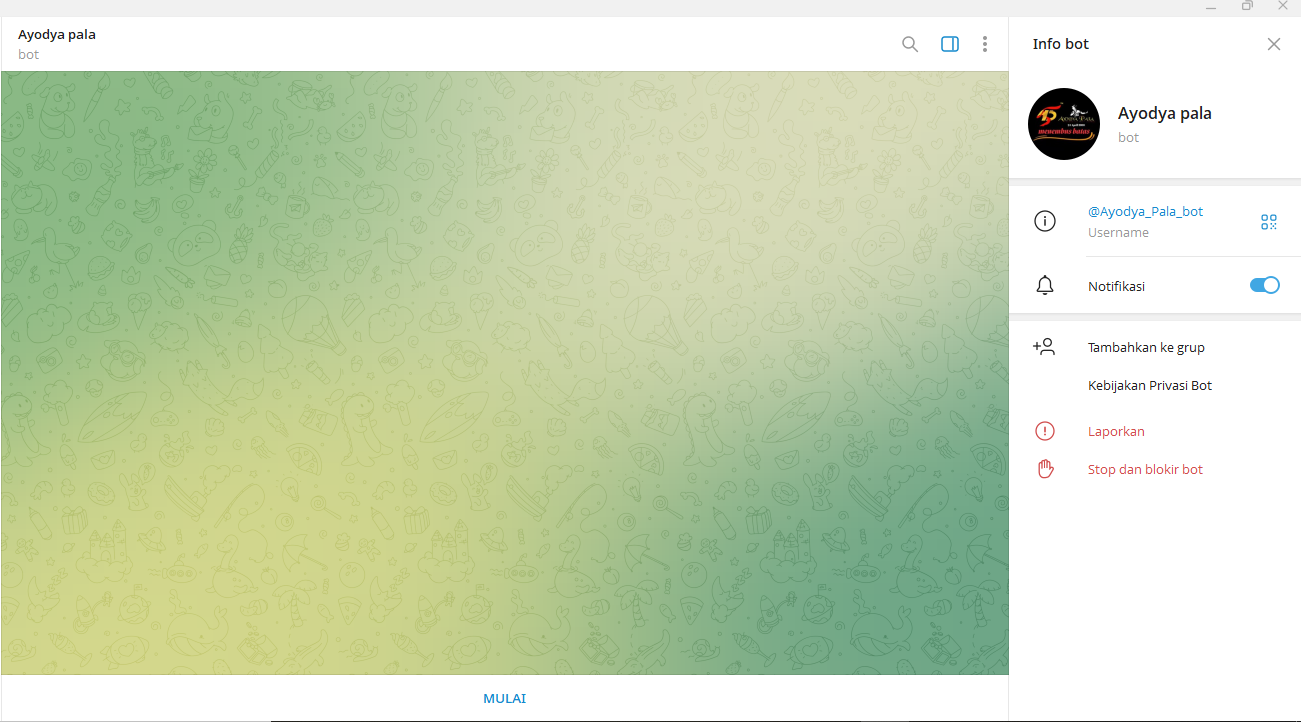
|  |  |  |
| --- | --- | --- |
| **No.** | **Perangkat Keras** | **Keterangan** |
| 1. | *Model* | *Vivo 2019* |
| 2. | *Processor* | *MediaTek Helio P35 – MT6765 Cores 8* |
| 3. | *Random Access Memory (RAM)* | *4GigaBytes* |
| 4. | Penyimpanan Internal | *64GigaBytes* |
| 5. | *Android Version* | *12 Snow Cone* |
| 6. | Sistem Operasi | *Funtouch OS\_10.5* |

## Implementasi Program

Analisis sistem merupakan suatu kegiatan yang melibatkan penemuan atau identifikasi masalah, evaluasi, pembuatan model, dan penyusunan spesifikasi sistem. Tujuan tersebut adalah untuk merancang sistem baru atau memperbaiki kelemahan yang ada dalam sistem yang sudah ada.



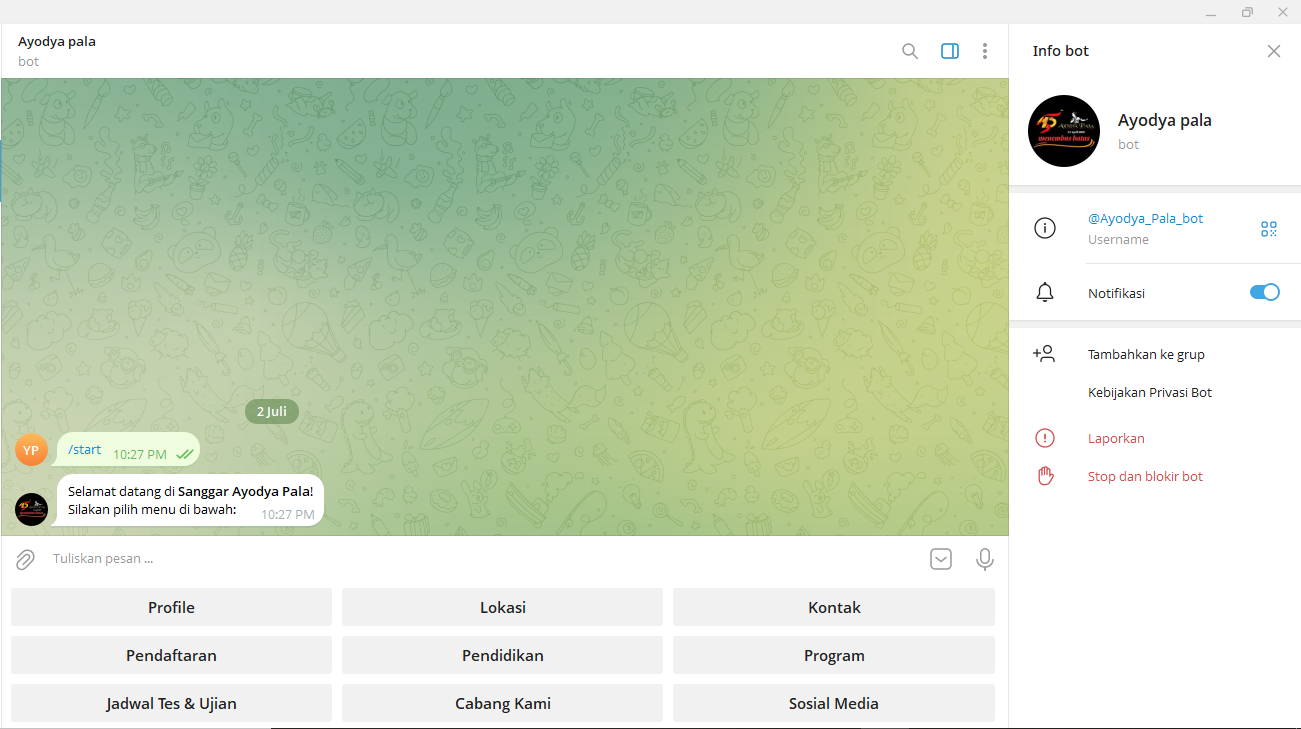
### Halaman Chatbot demo pada Perangkat Laptop



**Gambar 4. 1 Halaman Chatbot demo pada Perangkat Laptop**

Gambar 4.1 menampilkan antarmuka halaman pada *Chatbot* yang akan muncul saat pengguna mengakses *Link* tautan atau mencari di kolom pencarian yang telah disediakan. Tampilan ini memperlihatkan bagaimana pengguna dapat berinteraksi langsung dengan *Chatbot* melalui halaman tersebut.

### Tampilan Form Button Pertanyaan Pada Chatbot



**Gambar 4. 2 Tampilan Form Button Pertanyaan Pada Chatbot**

Gambar 4.2 menampilkan tampilan *form button* pertanyaan pada halaman *Chatbot*, di mana pengguna dapat mengajukan pertanyaan yang sudah disediakan di *button* kepada *Chatbot*. *Form Button* ini dirancang untuk memudahkan pengguna dalam memasukan pertanyaan secara langsung dan mendapatkan respons dari *Chatbot*.

### Tampilan *Form* Jawaban Pada *Chatbot*



**Gambar 4. 3 Tampilan Form Jawaban Pada Chatbot**

Gambar 4.3 menunjukkan tampilan *form* jawaban pada halaman *Chatbot*, di mana pengguna akan menerima jawaban yang sesuai dengan masukan pertanyaan yang diajukan kepada *Chatbot*.

## Pengujian Sistem

Pengujian sistem merupakan proses evaluasi untuk memastikan bahwa perangkat lunak yang telah dikembangkan dapat beroperasi sesuai dengan standar yang ditetapkan. Jenis pengujian sistem yang digunakan adalah *Functional Test* dan *User Acceptance Test (UAT)*.

### *Functional Testing (Black Box Testing)*

*Black Box Testing* adalah metode pengujian perangkat lunak yang dilakukan tanpa melihat atau memperhatikan struktur internal atau logika kode program. Pengujian ini berfokus pada *input* dan *output* yang diharapkan dari aplikasi tanpa mempertimbangkan bagaimana hasil tersebut dicapai. Teknik pengujian *black-box* berfokus pada domain informasi dari perangkat lunak, dengan melakukan kasus uji dengan mempartisi domain input dari suatu program dengan cara memberikan cakupan pengujian yang mendalam. Pengujian *black-box* didesain untuk mengungkap kesalahan pada persyaratan fungsional tanpa mengabaikan kerja internal dari suatu program (Kinanthi, 2023).

Berikut di bawah ini adalah data pengujian *Black Box* pada *Chatbot* yang dikembangkan oleh peneliti:

**Tabel 4. 4 Pengujian Black Box Pada Chatbot**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Fitur yang Diuji** | **Input Pengguna** | **Output yang Diharapkan** | **Status** |
| 1. | Menu Start | /start | Menampilkan Ucapan  Selamat datang di Sanggar Ayodya Pala!  Silakan pilih menu di bawah:  Menu utama dengan tombol: Profile, Lokasi, Kontak, Pendaftaran, Pendidikan, Program, Jadwal Tes & Ujian, Cabang Kami, & Sosial Media. | Valid |
| 2. | Menu Bantuan | /help | Menampilkan daftar perintah teks chatbot. | Valid |
| 3. | Menu Profil | /profile | Menampilkan informasi profil lengkap Sanggar Ayodya Pala. | Valid |
| 4. | Menu Lokasi | /lokasi | Menampilkan lokasi 2 cabang pusat dengan tombol menuju Google Maps. | Valid |
| 5. | Menu Kontak | /kontak | Menampilkan *Email*, *Whatsapp*, telepon kantor. | Valid |
| 6. | Menu Pendaftaran | /pendaftaran | Menampilkan syarat dan cara pendaftaran. | Valid |
| 7. | Menu Pendidikan | /pendidikan | Menampilkan informasi kelas & jadwal tes. | Valid |
| 8. | Menu Program | /program | Menampilkan program kegiatan sanggar. | Valid |
| 9. | Menu Jadwal Tes & Ujian | /jadwal tes, ujian, dll. | Menampilkan tanggal pelaksanaan tes ujian kenaikan tingkat dan ujian kenaikan tingkat. | Valid |
| 10. | Menu Cabang Kami | /cabang kami | Menampilkan 37 cabang aktif sesuai wilayah. | Valid |
| 11. | Menu Sosial Media | /sosial media | Menampilkan link Instagram, TikTok, dan YouTube. | Valid |
| 12. | Input tak dikenali | halo/apakah kamu pintar? | Menampilkan pesan default: “Maaf, saya belum mengerti maksud Anda.” | Valid |
| 13. | Error handler (simulasi) | Input tidak valid/sistem error | Error dicetak di terminal / log. | Valid |