**IMPLEMENTASI *CHATBOT* BERBASIS *ARTIFICIAL NEURAL NETWORK* UNTUK MENINGKATKAN EFISIENSI LAYANAN INFORMASI PPDB**

**(Studi Kasus: SMK KESATRIAN PURWOKERTO)**

# HALAMAN JUDUL

**Skripsi**



Disusun oleh

**Hanan Abdul Ghani**

**21SA1035**

**PROGRAM STUDI INFRORMATIKA**

**FAKULTAS ILMU KOMPUTER**

**UNIVERSITAS AMIKOM PURWOKERTO**

**2024**

# **BAB I PENDAHULUAN**

## **Latar Belakang Masalah**

Dalam era digital saat ini, institusi pendidikan dituntut untuk memberikan layanan informasi yang cepat dan akurat, terutama terkait proses Penerimaan Peserta Didik Baru (PPDB). Informasi mengenai persyaratan administrasi, jadwal pendaftaran, biaya pendidikan, serta pengumuman penting lainnya perlu disajikan secara jelas dan mudah diakses oleh calon siswa, orang tua, dan staf sekolah (Yudahana dkk., 2023). Namun seringkali akses terhadap informasi ini masih menghadapi berbagai kendala, seperti keterbatasan waktu operasional kantor, ketersediaan staf untuk menjawab pertanyaan (Ivan dkk., 2022). Hal ini menimbulkan keterlambatan dalam penyampaian informasi yang pada akhirnya mempengaruhi kenyamanan dan efisiensi dalam proses belajar mengajar.

SMK Kesatrian Purwokerto, yang berdiri sejak tahun 1996, merupakan Sekolah Menengah Kejuruan di bawah naungan Yayasan Perguruan Islam Republik Indonesia yang berlokasi di Desa Sokanegara, Kecamatan Purwokerto, Kabupaten Banyumas. Saat ini, sekolah ini memiliki sebanyak 2.025 siswa. SMK Kesatrian Purwokerto menawarkan berbagai jurusan, antara lain Teknik Kendaraan Ringan Otomotif, Teknik dan Bisnis Sepeda Motor, Teknik Audio Video, Teknik Komputer, dan Desain Komunikasi Visual (DKV). Untuk menunjang minat dan bakat siswa, sekolah menyediakan berbagai fasilitas seperti bengkel berstandar industri, laboratorium komputer, dan lapangan olahraga. SMK Kesatrian Purwokerto juga memiliki program unggulan berupa kelas industri di setiap jurusannya. Program ini terdiri dari siswa-siswi terpilih yang diarahkan untuk mengikuti magang di perusahaan mitra sekolah. Program ini dirancang sebagai langkah awal bagi siswa yang ingin langsung bekerja setelah lulus, memberikan prospek karier yang lebih jelas dan kesiapan menghadapi dunia kerja. Sebagai sekolah yang berkomitmen untuk memberikan pendidikan berkualitas, SMK Kesatrian Purwokerto senantiasa berupaya meningkatkan layanannya. Salah satu aspek penting yang perlu diperhatikan adalah penyediaan informasi yang akurat, mudah diakses, dan *up-to-date* bagi calon siswa.

Berdasarkan wawancara dengan Bapak Agung Sulistiono, S.T., selaku staf IT dan salah satu admin di SMK Kesatrian Purwokerto, Beliau menyampaikan bahwa informasi PPDB dapat diperoleh melalui laman resmi PPDB sekolah, kegiatan promosi ke SMP di sekitar, atau dengan mengunjungi sekolah secara langsung. Meskipun sekolah telah menyediakan platform *WhatsApp* sebagai media alternatif untuk memperoleh informasi, beliau menyatakan pada platfrom *WhatsApp* ini memiliki kendala seperti calon siswa kerap mengirimkan pertanyaan di luar jam kerja. hal ini dapat memakan waktu dalam merespon pertanyaan terutama saat volume pertanyaan tinggi atau di luar jam operasional sekolah.

Permasalahan diatas terkait ketidakefisienan dalam layanan informasi telah diteliti oleh beberapa jurnal seperti penelitian oleh Nugraha & Sebastian (2021) mengembangkan *chatbot* layanan akademik menggunakan algoritma *K-Nearest Neighbor* (KNN) dengan akurasi 53,48% (K=3) dari 86 pertanyaan. Menurut Mustakim, dkk(2021) menggunakan *Artificial Neural Network* (ANN) untuk *chatbot* FAQ kuliah daring, menghasilkan akurasi 97,27% dari 110 percakapan. Menurut Hikmah, dkk (2023) mengembangkan *chatbot* layanan akademik di Telkom University dengan ANN, mencapai akurasi 100% untuk 54 pertanyaan acak dan tingkat kepuasan pengguna 93%. Dari beberapa jurnal yang dijelaskan diatas penelitian oleh Mustakim & Hayati (2021) dan Hikmah, dkk (2023) menunjukkan bahwa ANN sangat cocok untuk *chatbot* layanan akademik dengan akurasi yang sangat tinggi. Keunggulan algoritma *Artificial Neural Network* (ANN) dalam pengembangan *chatbot* didukung oleh konsep *machine learning*, yang memungkinkan sistem belajar dan berkembang dari data serta pengalaman.

*Machine learning* adalah cabang dari kecerdasan buatan (AI) yang berfokus pada pengembangan algoritma dan teknik yang memungkinkan komputer untuk belajar dari data dan pengalaman tanpa perlu diprogram secara eksplisit. Konsep dasar dari machine learning adalah bahwa sistem dapat meningkatkan kinerjanya dalam menyelesaikan tugas tertentu seiring dengan bertambahnya data dan pengalaman yang diperoleh (Ling, 2023).

Kemampuan machine learning dalam belajar dari data semakin diperkuat oleh munculnya *Artificial Neural Network* (ANN). *Artificial Neural Network* (ANN) adalah model komputasi yang dirancang untuk meniru cara kerja otak manusia dalam mengolah informasi. ANN terdiri dari kumpulan "neuron" atau elemen komputasi sederhana yang saling terhubung untuk membentuk sistem yang mampu mempelajari pola, mengklasifikasikan, dan memprediksi data. ANN menggunakan bobot koneksi antar neuron untuk menyimpan informasi yang diperoleh dari proses pembelajaran. Dengan kemampuan ini, ANN sangat berguna dalam penambangan data, terutama karena ketahanannya dalam mengelola data yang mengandung noise atau ketidakpastian (Purwono dkk., 2022).

*Chatbot* merupakan produk hasil keluaran dari *machine learning*. *Chatbot* adalah program komputer yang menyimulasikan percakapan manusia dengan pengguna akhir. *Chatbot* adalah program komputer yang mensimulasikan percakapan manusia dalam format yang alami, baik dalam bentuk teks maupun suara, dengan memanfaatkan teknik kecerdasan buatan seperti *Natural Language Processing* (NLP), pemrosesan gambar dan video, serta analisis audio(Zuraiyah dkk., 2019). Salah satu perkembangan teknologi yang mendukung kemampuan *chatbot* dalam mengolah data teks adalah *text mining*, yang memungkinkan *chatbot* menganalisis dan memahami input dari pengguna dengan lebih efektif (Nurul Puteri dkk., 2022).

Secara keseluruhan, penulis bermaksud untuk membuat *Chatbot* berbasis *machinelearning* untuk *website* PPDB SMK Kesatrian Purwokerto menggunkan Algoritma *Artificial Neural Network* (ANN) guna membatu admin dalam merespon pertanyaan-pertanyaan dari calon siswa.

## **Rumusan Masalah**

Bagaimana Pengimplementasian *chatbot* berbasis *machinelearning* pada *website* PPDB sekolah menggunakan algoritma *Artificial Neural Network* guna membantu admin PPDB dalam merespon pertanyaan ?*.*

## **Batasan Masalah**

Berdasarkan latar belakang masalah, penelitian ini dibatasi oleh hal-hal sebagai berikut :

1. *Chatbot* ini dirancang hanya untuk menjawab pertanyaan terkiat informasi PPDB sekolah atau informasi pendaftaran.
2. *Chatbot* hanya akan mendukung bahasa indonesia.
3. Pengembangan *chatbot* pada penelitian ini dilakukan dengan bahasa pemrograman Python*.*

## **Tujuan Penelitian**

Bersasarkan latar belakang dan rumusan masalah. Penelitian ini bertujuan untuk mengimplementasikan *chatbot* berbasis *machinelearning* pada *website* PPDB sekolah menggunakan algoritma *Artificial Neural Network* guna membantu admin PPDB dalam merespon pertanyaan.

## **Manfaat Penelitian**

1. Manfaat Teoritik
2. Memberikan kontribusi dalam bidang ilmu pengetahuan dan teknologi yang berbasis kecerdasan buatan (AI), khususnya dalam penerapan *Machine Learning*
3. Sebagai acuan bagi penelitian mendatang dalam bidang kecerdasan buatan (AI), khususnya dalam penerapan machine learning berbasis web.
4. Manfaat Aplikatif
5. Mempermudah admin sekolah SMK Kesatrian Purwokerto dalam memberikan layanan yang lebih efisien kepada calon siswa
6. Mambuat waktu dalam memperoleh informasi sekolah menjadi lebih efisien bagi calon siswa.

# **BAB II TINJAUAN PUSTAKA**

## **Landasan Teori**

1. Implementasi

Dalam Kamus Besar Bahasa Indonesia (KBBI), implementasi memiliki makna pelaksanaan atau penerapan. Implementasi bukan hanya sekadar sebuah aktivitas, melainkan sebuah kegiatan yang dirancang dengan baik dan dilaksanakan dengan serius sesuai pedoman atau norma tertentu untuk mencapai tujuan yang telah ditetapkan. Oleh karena itu, implementasi tidak terjadi secara terpisah, melainkan dipengaruhi oleh elemen-elemen lain yang terkait(Rosad, 2019).

Keberhasilan implementasi adalah sebuah proses yang melibatkan berbagai elemen baru dan implementasi yang berhasil bergantung pada perencanaan langkah-langkah yang tepat, khususnya terkait proses pengembangannya(Suprapto & Malik, 2019).

1. Penerimaan Peserta Didik Baru (PPDB)

Penerimaan perserta didik baru (PPDB) merupakan kegiatan wajib yang dilakukan oleh lembaga pendidikan untuk menerima peserta didik baru, baik formal maupun non formal. PPDB adalah proses seleksi calon siswa yang akan menjadi bagian dari sekolah. Kegiatan ini merupakan rutinitas tahunan yang dilakukan setiap awal tahun ajaran. Penyelenggaraan PPDB harus dilakukan sesuai dengan standar yang telah ditetapkan oleh pemerintah (Rohmah dkk., 2021).

Tujuan dari PPDB menurut Pasal 2 ayat (1) menyatakan bahwa tujuan dari PPDB adalah memastikan proses penerimaan peserta didik baru dilakukan secara objektif, transparan, akuntabel, tanpa diskriminasi, dan berkeadilan, sehingga dapat mendukung peningkatan akses terhadap layanan pendidikan .

1. *Machine Learning*

*Machine learning* (ML) adalah mesin yang dikembangkan untuk bisa belajar dengan sendirinya tanpa arahan dari penggunanya. Machine learning, atau pembelajaran mesin, adalah teknologi yang sangat bermanfaat dalam menyelesaikan berbagai masalah dan mempermudah pelaksanaan berbagai tugas(Telaumbanua dkk., 2020). Menurut (Kurniyawan, 2022) *Machine Learning* adalah ilmu yang mempelajari tentang algoritma komputer yang bisa mengenali pola-pola di dalam data, dengan tujuan untuk mengubah beragam macam data menjadi suatu tindakan yang nyata dengan sesedikit mungkin campur tangan manusia.

Dari pengertian di atas, *machine learning* (ML) dapat diartikan sebagai cabang ilmu komputer yang mengembangkan kemampuan mesin untuk belajar secara mandiri dari data tanpa arahan langsung dari manusia. Teknologi ini dirancang untuk mengenali pola-pola dalam data dan mengolahnya menjadi tindakan nyata dengan minim campur tangan manusia.

1. *Chatbot*

*Chatbot* adalah program komputer yang dirancang untuk menirukan percakapan manusia dengan pengguna, baik tertulis maupun lisan. *Chatbot* merupakan aplikasi yang menggunakan kecerdasan buatan untuk berkomunikasi secara otomatis dengan pengguna melalui antarmuka percakapan. Dengan kemampuannya dalam menjawab pertanyaan, menyampaikan informasi, dan memberikan solusi, *chatbot* dapat berfungsi sebagai asisten virtual yang efisien (Lubis & Sumartono, 2023)

*Chatbot* adalah aplikasi yang didasarkan pada ilmu Natural Language Processing (NLP) dan berbasis kecerdasan buatan (AI), yang berfungsi sebagai sumber informasi yang dapat diakses oleh pengguna aplikasi(Sujacka Retno dkk., 2023).

1. *Text Mining*

*Text mining* dalah suatu bidang khusus *data mining*. *Text mining* adalah proses ekstraksi informasi dari data berbentuk teks, dengan sumber informasi yang biasanya berasal dari dokumen. Tujuan utamanya adalah mengidentifikasi kata-kata yang mewakili isi dokumen, sehingga analisis keterkaitan antar dokumen dapat dilakukan (Purnajaya dkk., 2022).

*Text mining* mencakup berbagai teknik dan metode yang digunakan untuk menganalisis, mengklasifikasikan, dan mengekstrak informasi dari teks. Pada *text mining* terdapat tahapan t*ext Preprocessing* yang Bertujuan untuk mengonversi data teks yang awalnya tidak terstruktur menjadi data teks yang lebih terorganisasi. Data yang telah terstruktur ini kemudian dapat dimanfaatkan untuk berbagai keperluan, seperti analisis mendalam, pencarian pola, pengelompokan informasi, atau sebagai dasar dalam pengambilan keputusan berbasis data (Ramadhani dkk., 2022). Berikut tahapan dari *text prossesing*.

1. *Case Folding*

*Case folding* adalah mengubah semua huruf dalam dokumen menjadi huruf kecil, hanya huruf 'a' sampai dengan 'z' yang diterima. Karakter selain huruf dihilangkan dan dianggap delimeter.

1. Tokenisasi

Tokenisasi adalah proses memecah teks menjadi unit-unit kecil yang disebut token, seperti kata, frasa, atau simbol. Proses ini merupakan langkah awal dalam pengolahan teks untuk memudahkan analisis lebih lanjut, seperti pencarian pola atau penghitungan frekuensi kata.

1. *Stopword Removal*

Stopword removal adalah proses menghapus kata-kata umum yang tidak memiliki makna signifikan dalam analisis teks, seperti "dan," "atau," "yang," atau "itu." Tujuannya adalah untuk mengurangi kata-kata yang tidak relevan sehingga fokus analisis dapat diarahkan pada kata-kata yang lebih bermakna.

1. *Stemming*

*Stemming* adalah proses mengubah kata turunan ke bentuk dasarnya (akar kata) dengan menghapus imbuhan seperti awalan, akhiran, atau sisipan. Tujuannya adalah untuk menyederhanakan analisis teks dengan mengelompokkan kata-kata yang memiliki makna serupa. Contohnya, kata "berlari" dan "lari-lari" akan direduksi menjadi "lari."

1. *Artificial Neural Network* (ANN)

*Artificial Neural Network* merupakan model algoritma yang mencoba meniru otak manusia yang mampu memberikan stimulasi/rangsangan, melakukan proses, dan memberikan output untuk menemukan hubungan antara kumpulan data. Pemodelan berbasis *Artificial Neural Network* (ANN) adalah pendekatan yang digunakan untuk menyelesaikan masalah pengenalan pola dan teknik *data mining*. Jaringan ini terdiri dari sejumlah neuron buatan yang dikenal sebagai elemen pemrosesan (PE), unit, atau node (Iskandar & Sriharyani, 2021).

Struktur jaringan dan algoritma pelatihan memainkan peran penting dalam menentukan model-model ANN. Struktur ini berfungsi untuk menggambarkan bagaimana sinyal atau data bergerak melalui jaringan. Sementara itu, algoritma pembelajaran menjelaskan cara-cara mengubah bobot koneksi agar pasangan input-output yang diinginkan dapat tercapai (Syukri & Samsuddin, 2019).



gambar 2. 1 Arsitektur Artificial Neural Network (ANN)

pada gambar 2.1 merupakan bentuk dari arsitektur *Artificial Neural Network* (ANN), yang terdiri dari :

1. *Input Layer*

*Input layer* adalah lapisan yang terdiri dari unit-unit (neuron) yang menerima sinyal input langsung dari sumber eksternal dan mengirimkan informasi tersebut ke setiap neuron di hidden layer melalui bobot yang menghubungkan kedua lapisan tersebut.

1. *Hidden Layer*

*Hidden layer* adalah lapisan yang terdiri dari neuron-neuron tersembunyi yang terletak di antara input *layer* dan *output layer*, di mana *output*-nya tidak terlihat secara langsung. Penambahan *hidden layer* dapat memperbaiki kemampuan jaringan dalam mengenali pola.

1. *Output Layer*

*Output layer* adalah lapisan yang memiliki unit *output* yang keluarannya memiliki solusi dari algoritma *Artificial Neural Network* pada permasalahan yang di berikan.

1. Python

Python merupakan bahasa pemrograman yang biasa dipakai untuk membangun situs, software/aplikasi, dan melakukan analisis data. Python memliki kemampuan untuk membangun sofware berbasis kecerdasan buatan (*artificial intelligence*) seperti pengolahan data, *machine learning*, *deep learning*, dan *data science*. Python adalah bahasa pemrograman multi-paradigma yang menggabungkan pemrograman berorientasi objek, imperatif, dan fungsional. (Enterprise, 2019).

Dari penjelasan ditas bahasa pemrograman python memiliki Kelebihannya tersendiri antara lain :

1. Proses pengembangan perangkat lunak menjadi lebih efisien dengan pengurangan baris kode
2. Mendukung multi platform
3. Dilengkapi dengan fitur alokasi memori otomatis.
4. Python mendukung paradigma pemrograman berorientasi objek.

## **Penelitian Sebelumnya**

1. Kristian Adi Nugraha & Danny Sebastian (2021) melakukan peneliatian dengan judul “*Chatbot* Layanan Akademik Menggunakan *K-Nearest Neighbor*”. Institusi kecil hingga menengah seringkali menghadapi kendala dalam hal menjawab pertanyaan berulang. Penelitian ini berfokus pada mengembangkan *chatbot* dengan memanfaatkan algoritma *K-Nearest Neighbor* (K-NN) untuk menjawab pertanyaan terkait kegiatan akademik secara otomatis, khususnya di lingkungan Fakultas Teknologi Informasi Universitas Kristen Duta Wacana. Sistem ini dibangun menggunakan bahasa pemrograman Python dan mengadopsi teknik Natural Language Processing seperti tokenisasi, stemming, dan penghilangan kata-kata *stop*. Hasil pengujian menunjukkan akurasi tertinggi sebesar 53,48% pada nilai K=3. Kendati demikian, sistem masih menghadapi tantangan dalam mengklasifikasi pertanyaan yang memiliki struktur kata serupa atau menggunakan kata-kata tidak baku.
2. Feri Mustakin, dkk (2021) melakukan penelitian dengan judul “Algoritma *Artificial Neural Network* pada Text-based *Chatbot* *Frequently Asked Question* (FAQ) Web Kuliah Universitas Nasional”. Kurangnya literasi mahasiswa terhadap penggunaan web kuliah Universitas Nasional menyebabkan pertanyaan berulang terkait hal seperti pengumpulan tugas dan lupa kata sandi. Penelitian ini mengembangkan *chatbot* berbasis teks menggunakan algoritma *Artificial Neural Network* (ANN) dengan dataset 25 pertanyaan FAQ yang dikelompokkan menjadi 16 label. Setelah pelatihan dengan 1000 epoch dan teknik Natural Language Processing (NLP), pengujian menunjukkan akurasi tinggi sebesar 97,27%, sehingga efektif dalam menjawab pertanyaan mahasiswa secara otomatis.
3. Muhamad Sidik, dkk (2021)melakukan penelitian dengan judul “Pembuatan Aplikasi *Chatbot* Kolektor Dengan Metode *Extreme Programming* Dan Strategi *Forward Chaining*.” Penelitian ini mengatasi kendala sistem SMSCenter di PT. Indomobil Finance Indonesia, seperti ketergantungan pada kartu SIM dan biaya tinggi, dengan mengembangkan *Chatbot* Kolektor berbasis LINE menggunakan metode Extreme Programming dan strategi Forward Chaining. Dataset mencakup data kolektor dan kendaraan, seperti nomor polisi, mesin, rangka, serta informasi blacklist pelanggan. Hasil pengujian menunjukkan *chatbot* memiliki tingkat keberhasilan 95% dan waktu respon rata-rata 3,42 detik, jauh lebih cepat dari SMSCenter (24,25 detik), sehingga meningkatkan efisiensi layanan dan mengurangi biaya operasional.
4. Fahmi Yusron, dkk (2024) melakukan penelitian dengan judul “*Chatbot* Informasi Penerimaan Mahasiswa Baru Menggunakan Metode *FastText* dan LSTM” Layanan informasi penerimaan mahasiswa baru (PMB) di Fakultas Sains dan Informatika Universitas Jenderal Achmad Yani dinilai kurang efisien karena masih dilakukan secara manual, menyebabkan pengulangan jawaban atas pertanyaan serupa. Artikel ini membahas pengembangan *chatbot* berbasis metode *FastText* untuk representasi kata dan *Long Short-Term Memory* (LSTM) untuk klasifikasi teks, guna meningkatkan efisiensi dan konsistensi layanan. Dengan data dari kuesioner mahasiswa, model yang dihasilkan memiliki akurasi tinggi (89–90%) dan mampu menjawab berbagai pertanyaan terkait PMB dengan respons relevan dan informatif.
5. Mohammad Ovi Sanjaya, dkk (2023) melakukan penelitian dengan judul “*Virtual Assistant for Thesis Technical Guide Using Artificial Neural Network*” Permasalahan dalam memberikan panduan teknis skripsi secara cepat dan akurat mendorong pengembangan *chatbot* berbasis *Artificial Neural Network* (ANN). *Chatbot* ini menggunakan model ANN dengan fungsi aktivasi ReLU dan Softmax serta dioptimalkan menggunakan Stochastic Gradient Descent (SGD). Dengan basis Panduan Teknis Skripsi 2022, *chatbot* mencapai akurasi 99,49% dan skor F1 sebesar 91%. Diuji dengan confusion matrix dan diimplementasikan pada Telegram, *chatbot* ini efektif memberikan panduan teknis berbasis teks

Tabel 2. 1 Tabel Penelitian Sebelumnya

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Nama dan Tahun Penelitian** | **Judul Penelitian** | **Hasil** | **Perbandingan** | | |
| **Persamaan** | **Perbedaan** |
| 1 | Kristian Adi Nugraha dan Danny Sebastian, 2021 | *Chatbot* Layanan Akademik Menggunakan K-Nearest Neighbor | Menghasilkan *chatbot* dengan algoritma KKN untuk menjawab pertanyaan seputar kegiatan akademik mendapatkan akurasi sebesar 53,48% | 1. *Chatbot* berbasis *machine learning.* | Penelitian ini menggunakan algoritma ANN dikarenakan algoritma lebih cocok untuk dataset yang kompleks dibanding dengan algoritma KNN yang cocok pada dataset kecil yang membutuhkan interpretasi mudah. |
| 2 | Feri Mustakin, Fauziah, Nur Hayati, 2021 | Algoritma *Artificial Neural Network* pada *Text-based Chatbot Frequently Asked Question* (FAQ) Web Kuliah Universitas | Menghasilkan *chatbot* dengan algoritma ANN untuk membantu dalam menjawab pertanyaan dalam FAQ dalam bentuk GUI mendapatkan akurasi 97,27%. | Menggunakan algoritma ANN  1. Berbasis *machine laerning* | Perbedaan pada *output chatbot*  * Pada penelitian sebelumnya *chatbot* bentuk GUI * Pada penelitian ini *chatbot* berbentuk *website* |
| 3 | Muhamad Sidik, Bambang Gunawan, dan Dina Anggraini, 2021 | Pembuatan Aplikasi *Chatbot* Kolektor Dengan Metode *Extreme*Programming Dan Strategi *Forward Chaining* | Menghasilkan *chatbot* dengan algoritma *Forward Chaining* berbasis LINE sebagai media layanan informasi secara otomatis kepada dengan akurasi 95%. | 1. Penelitian sebelumnya sama-sama meneliti terkait *chatbot* | Penelitian ini menggunakan algoritma ANN dikarenakan ideal untuk *chatbot* interaktif berbasis NLP dengan fleksibilitas tinggi dari pada algoritma *Forward Chaining* berbasis logika *(rule-based)* |
| 4 | Fahmi Yusron F, Agus Komarudin, dan Melina, 2024 | *Chatbot* Informasi Penerimaan Mahasiswa Baru Menggunakan Metode *FastText* dan LSTM | Menghasilkan *chatbot*  untuk menjawab pertanyaan PMB eperti biaya, jadwal pendaftaran, beasiswa, dll. Menggunakan metode FastTex dan LSTM untuk klasifikasi teks mendapatkan akurasi 89–90% | Penelitian pengembangan *chatbot* | Pada penelitian ini menggunakan algoritma ANN  Dikarenakan lebih sederhana dibandingkan LSTM, sehingga lebih cepat dilatih dan membutuhkan lebih sedikit sumber daya komputasi |
| 5 | Mohammad Ovi Sanjaya, Saiful Bukhori, dan Muhammad ‘Ariful Furqon, 2023 | *Virtual Assistant for Thesis Technical Guide Using Artificial Neural Network* | Menghasilkan *chatbot* unturk panduan teknis tesis. Menggunakan algoritma ANN dan menggunkan metode optimasi *Stochastic Gradient Descent* (SGD*)* mendapatkan akurasi 91%*.* | Menggunakan algoritma ANN | Perbedaan pada *output chatbot*   * Penelitian sebelumnya menggunkan telegram * Penelitian ini *chatbot* berbasis *website* |

# **BAB III METODE PENELITIAN**

## **Tempat dan Waktu Penelitian**

1. Tempat Penelitian

Pada penelitian ini dilaksanakan di SMK Kesatrian Purwokerto Desa Sokanegara, Kecamatan Purwokerto, Kabupaten Banyumas, Jawa Tengah 53115.

1. Waktu Penelitian

Waktu Penelitian dilakukan selama 4 bulan mulai september 2024 hingga 14 januari 2025.

## **Metode Pengumpulan Data**

Bagian ini memuat penjelasan secara lengkap dan terinci tentang cara-cara yang digunakan dalam proses pengumpulan data untuk jenis data yang diperlukan. Misalnya melalui observasi, wawancara, eksperimen, atau kuesioner. Jika metode kuesioner digunakan, maka blangko angket kuesioner harus dilampirkan dalam laporan.

Pada penelitian ini diperlukan serangkaian kegitan untuk mendapatkan data yang di butuhkan pada penelitian. Dalam melakukan penelitian penulis menggunkan metode pengumpulan data sebagai berikut:

1. Wawancara

Menurut (Fadhallah, 2021) wawancara merupakan bentuk komunikasi antara dua atau lebih pihak yang biasanya dilakukan secara langsung. Dalam wawancara, satu pihak bertindak sebagai pewawancara (interviewer) dan pihak lainnya sebagai yang diwawancarai (interviewee) dengan tujuan tertentu, seperti memperoleh informasi atau mengumpulkan data. Pewawancara mengajukan sejumlah pertanyaan kepada yang diwawancarai untuk mendapatkan jawaban yang diperlukan.

Peneliti melakukan wawancara dengan staf IT sekaligus Admin PPDB SMK Kesatrian Purwokerto yaitu bapak Agung Sulistiono, S.T.

1. Observasi

Observasi adalah teknik pengumpulan data yang unik karena melibatkan pengamatan langsung terhadap suatu fenomena. Berbeda dengan wawancara atau kuisioner yang mengandalkan laporan subjektif, observasi memungkinkan peneliti untuk mengamati perilaku dan kejadian secara objektif. Metode ini sangat relevan untuk penelitian yang berkaitan dengan perilaku manusia, proses kerja, atau gejala alam, terutama ketika jumlah subjek penelitian relatif kecil (Sugiyono, 2018).

Penulis menggunakan teknik observasi untuk mengumpulkan data mengenai pertanyaan-pertanyaan yang sering ditanyakan oleh calon siswa mengenai informasi PPDB dan informasi sekolah untuk dijadikan *dataset* pada *chatbot*.

1. Studi Pustaka

Studi kepustakaan merupakan langkah penting dalam penelitian yang melibatkan pengkajian mendalam terhadap teori-teori relevan, mutakhir, dan asli yang berkaitan dengan objek penelitian. Teori-teori ini berperan sebagai landasan berpikir dalam menganalisis data dan menarik kesimpulan (Sugiyono, 2018).

Penulis melakukan kajian pustaka komprehensif dengan merujuk pada berbagai sumber seperti jurnal ilmiah, buku, skripsi, dan ebook untuk memperkaya landasan teori dan metodologi penelitian.

## **Alat dan Bahan Penelitian**

Dalam penelitian ini memerlukan alat dan bahan. Berikut adalah alat dan bahan yang digunakan:

1. Alat Penelitian
2. Komputer PC (Personal Computer)

Spesifikasi komputer pc yang di gunakan penulis sebagai berikut :

1. Laptop : Acer aspire 4741
2. *Processor* : Intel Core i3 i3-350M 2,26 GHz
3. RAM : 6 GB
4. *Hardisk* : 500 GB
5. Perangkat Lunak (Software)
6. Sistem Operasi Windows 10
7. Microsoft Word 2019
8. Chrome Browser
9. Visual Studio Code
10. Python
11. Bahan

Bahan penelitian ini berupa *dataset* yang berasal dari hasil wawancara dengan Bapak Agung Sulistiono, S.T. sebagai narasumber ahli, serta tanggapan responden terhadap kuesioner yang berisi pertanyaan-pertanyaan umum seputar pelaksanaan PPDB.

## **Konsep Penelitian**

1. Kerangka Berpikir



Gambar 3. 1 Alur kerangka berpikir

Penelitian ini menggunakan alur kerangka berpikir sebagai panduan untuk menyelesaikan setiap proses yang ada, sehingga dapat membantu dalam perancangan *chatbot* agar berfungsi dengan baik sesuai dengan tahapan yang telah dirancang. Berikut ini adalah penjelasan mengenai alur kerangka berpikir dalam penelitian ini:

1. Identifikasi Masalah

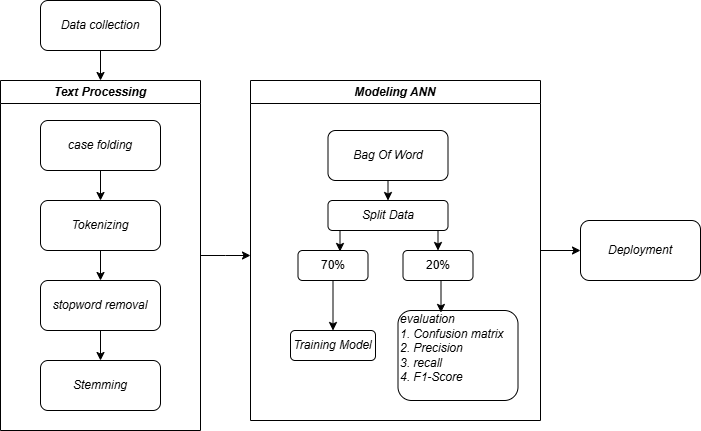
Proses identifikasi masalah adalah langkah awal dalam penelitian yang bertujuan untuk mengidentifikasi permasalahan pada objek penelitian SMK Kesatrian Purwokerto. Tahap ini memiliki peran penting karena hasilnya akan digunakan untuk merumuskan permasalahan yang ada di lokasi penelitian, yang nantinya menjadi dasar dalam merancang latar belakang penelitian tersebut.

1. Pengumpulan Data

Pada tahap ini, peneliti mengumpulkan berbagai data yang relevan dengan objek permasalahan untuk melengkapi bahan penelitian. Dalam proses pengumpulan data, peneliti menggunakan beberapa metode, antara lain wawancara, pengamatan (observasi), dan Studi pustaka.

1. Tahapan Pengembangan *Chatbot*

Dalam pengembangan *chatbot*, terdapat tahapan-tahapan yang perlu dilakukan untuk membuat *chatbot* yang akan dikembangkan. Setelah mengumpulkan data dan memilih pertanyaan-pertanyaan yang sering diajukan oleh calon siswa, data tersebut akan diolah dan dilatih.



Gambar 3. 2 Alur Proses Pengembangan Chatbot

1. *Data Collection*

*Data Collection* dikumpulkan untuk keperluan *Text Processing* dengan tujuan menyediakan bahan baku yang memadai untuk menjalankan berbagai analisis bahasa. Sumber dataset yang digunakan adalah pertanyaan-pertanyaan yang sering di ajukan tentang PPDB dan akan disimpan dalam format JSON. Dataset memiliki struktur diantaranya :

* *Intents*, Kumpulan semua data input dan output yang digunakan untuk melatih *chatbot*
* *Tag*, digunakan untuk mengelompokkan data teks yang serupa dan menjadikannya sebagai output target dalam melatih jaringan neural.
* *Patterns,* merupakan bagian yang berisi pola pertanyaan yang diinginkan pengguna.
* *Responses,* berisikan data pola *output* atau jawaban dari pertnyaan yang akan dikirimkan *chatbot* pada pengguna.

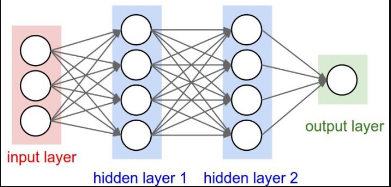
1. *Text Prossesing*

*Text processing* merupakan tahapan awal dalam pengembangan *chatbot* yang bertujuan untuk mempersiapkan data teks agar lebih mudah diproses oleh model.

* *case folding*, yaitu mengubah seluruh huruf dalam teks menjadi huruf kecil untuk menghindari perbedaan akibat kapitalisasi.
* *tokenisasi*, di mana teks dibagi menjadi potongan-potongan kecil berupa kata atau token.
* *stopword removal* dilakukan untuk menghapus kata-kata umum yang tidak memiliki makna signifikan, seperti "dan," "ke," atau "yang."
* *stemming* bertujuan mengubah kata menjadi bentuk dasar atau akarnya, sehingga kata seperti "berlari" akan dikembalikan menjadi "lari." Semua tahapan ini memastikan data teks siap digunakan untuk pembuatan model yang lebih akurat dan efisien.

1. *Modelling* ANN

Implementasi model *Artificial Neural Network* (ANN) untuk pemrosesan teks melibatkan perancangan arsitektur jaringan saraf tiruan jenis *forward neural network* (FNN) model *Multilayer perceptrons* (MLP) dengan menggunkan dua lapisan *hidden layer* dalam menghasilkan serangkaian *output* dari *input* yang diberikan.



Gambar 3. 3 *Arsitektur* Multilayer perceptrons *MLP*

Dari gambar 3.3 mejuntukan bagaimana Algoritma ANN *input layer* untuk menerima data, dua *hidden layers* untuk memproses pola kompleks. Setiap lapisan dalam jaringan saraf tiruan menggunakan fungsi aktivasi untuk menentukan keluaran neuron. Penelitian ini menggunakan fungsi ReLU, yang mengubah nilai negatif menjadi nol dan fungsi *softmax*, yang menghitung probabilitas untuk klasifikasi multi-kelas (Mustakim dkk., 2021). Berikut rumusan dari fungsi aktivitas ReLU (1) dan fungsi aktivitas softmax (2):

(1)

(2)

ReLU sederhana dan efisien, sedangkan softmax menghasilkan output berupa probabilitas antara 0 dan 1.

Model yang telah dilatih akan mampu memahami pola kompleks dalam data teks dan melakukan tugas-tugas seperti klasifikasi sentimen, terjemahan mesin, atau generasi teks.

1. *Bag of Words*

Konsep *bag of words* mereduksi teks menjadi sekumpulan kata dan menghitung jumlah kemunculan masing-masing kata dalam dokumen, menghasilkan vektor numerik yang mewakili teks tersebut (Arbizal dkk., 2024). Proses ini menginisialisasi sistem dengan data pelatihan. Setiap kata dalam data ini akan dipetakan ke sebuah vektor biner, di mana nilai 1 menunjukkan keberadaan kata dalam kamus kata, dan 0 menunjukkan ketidakhadirannya seperti yang di jelaskan pada table.

Tabel 3. 1 Proses *Bag of Words*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Kata | Bag of Words | | | | |
| Saya | Ingin | Sekali | Daftar | Sekolah |
| Saya | 1 | 0 | 0 | 0 | 0 |
| Ingin | 0 | 1 | 0 | 0 | 0 |
| Daftar | 0 | 0 | 0 | 1 | 0 |
| Sekolah | 0 | 0 | 0 | 0 | 1 |

1. *Split Data*

*Split data* atau pemisahan data adalah proses memisahkan data menjadi 70% - 20%, 70% untuk data *Training* dan 20 % untuk data evaluasi. Tujuan melakukan pemisahan data ini untuk ntuk menghindari *overfitting*

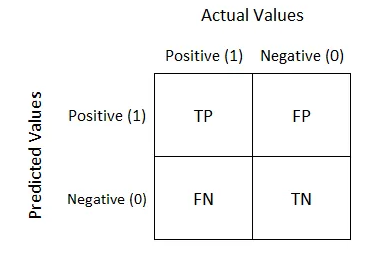
1. *Training Model*

Proses training model bertujuan untuk mengoptimalkan performa *Artificial Neural Network* (ANN) dengan mencari kombinasi bobot dan bias terbaik yang meminimalkan *error* (*loss*) dan meningkatkan tingkat akurasi. Selama training, model belajar dari data latih melalui iterasi berulang, di mana algoritma optimasi, *Stochastic Gradient Descent* (SGD) digunakan untuk menyesuaikan parameter model berdasarkan perhitungan error. Proses pelatihan data dilakukan dengan *stoping* kriteria berdasarkan jumlah epoch dan batch size.

1. Evaluasi

Dari *training* data akan di evaluasi menggunakan *confusion matrix.* Confusion matrix merupakan representasi kinerja model klasifikasi pada dataset Berhasilasi dengan nilai yang telah diketahui sebelumnya. *Confusion matrix* mencakup empat metrik evaluasi utama *recall, precision, accuracy*, dan *F1-score*. Metrik-metrik ini digunakan untuk mengukur sejauh mana model mampu memprediksi kelas pada data uji dengan benar. Selain itu, *confusion matrix* juga membantu mengidentifikasi kesalahan klasifikasi serta membandingkan performa antar model (Faurina dkk., 2023).

Confusion matrix terdiri dari empat elemen utama dalam klasifikasi, yaitu *False Positive* (FP), *True Positive* (TP), *True Negative* (TN), dan *False Negative* (FN). *True Positive* (TP) mengacu pada data positif yang berhasil terdeteksi dengan benar, sedangkan *True Negative* (TN) adalah data negatif yang diidentifikasi secara akurat. Sebaliknya, *False Positive* (FP) terjadi ketika data negatif salah diklasifikasikan sebagai positif, dan *False Negative* (FN) adalah data positif yang tidak dikenali dan dianggap sebagai negatif .



Gambar 3. 4 Confusion Matrix

*Precision* mengukur seberapa sering prediksi positif yang dibuat oleh model sebenarnya benar (Krstinic dkk., 2020) (6) .

(6)

*Recall* mengukur seberapa lengkap model dapat menemukan semua contoh positif yang sebenarnya ada dalam data (Krstinic dkk., 2020) (7).

(7)

Akurasi mengukur seberapa sering model kita membuat prediksi yang benar secara keseluruhan (Krstinic dkk., 2020) (8).

(8)

*F1-score* adalah metrik yang menggabungkan *precision* dan *recall* untuk memberikan penilaian yang lebih seimbang tentang kinerja model (Krstinic dkk., 2020) (9).

(9)

Evaluasi model *chatbot* dalam penelitian ini menggunakan akurasi, presisi, *recall*, dan *F1-score* untuk memberikan analisis kinerja model yang menyeluruh. Akurasi menilai seberapa sering prediksi benar secara keseluruhan, presisi memastikan akurasi prediksi kelas positif untuk meminimalkan *false positives*, dan *recall* mengukur kemampuan model mendeteksi semua kasus positif guna mengurangi *false negatives*. *F1-score*, sebagai rata-rata harmonis presisi dan *recall* (Faurina dkk., 2023). Evaluasi ini memastikan *chatbot* dapat memberikan informasi yang relevan, benar, dan andal kepada pengguna​.

1. *Deployment*

Proses *deployment chatbot* untuk layanan penerimaan perserta didik baru (PPDB) pada SMK Kesatrian Purwokereto dilakukan dalam bentuk *website* dengan menggunakan *framework* Flask berbasis Python. *Framework* Flask tergolong sebagai *micro-framework* karena tidak membutuhkan banyak *library* atau alat tambahan. *Framework* ini mengandalkan Werkzeug dan Jinja *Template Engine* sebagai dependensinya, serta memungkinkan pengembangan aplikasi dengan *sintaks* yang mudah dan sederhana (Larasati & Susetyo, 2024).

1. Pengujian

Pengujian dilakukan dengan pendekatan *Blackbox* *Testing*, yaitu metode pengujian aplikasi yang tidak memerlukan pemahaman mendalam tentang detail teknis aplikasi, seperti *source code* (Sasongko dkk., t.t.). Tujuan pengujian ini adalah untuk memastikan apakah *chatbot* dapat memberikan respon yang sesuai dengan pertanyaan yang diajukan oleh pengguna.

1. Penyusunan Laporan

Setelah seluruh proses penelitian tuntas, langkah berikutnya adalah menyusun laporan ilmiah. Laporan ini akan mencakup saran dan kesimpulan yang menunjukkan potensi pengembangan lebih lanjut dari penelitian ini.

# **BAB IV HASIL DAN PEMBAHASAN**

## **Identifikasi Masalah**

Dalam Proses identifikasi masalah dilakukan untuk menentukan permasalahan utama yang dihadapi oleh SMK Kesatrian Purwokerto dalam penyampaian informasi Penerimaan Peserta Didik Baru (PPDB). Berdasarkan wawancara dengan Bapak Agung Sulistiono, S.T., selaku staf IT dan admin PPDB sekolah, ditemukan beberapa kendala:

1. Keterbatasan Layanan di Luar Jam Kerja
2. Volume Pertanyaan yang Tinggi

Hasil identifikasi ini menjadi dasar untuk merancang *chatbot* yang mampu memberikan layanan informasi secara otomatis dan efisien, terutama terkait informasi PPDB seperti jadwal pendaftaran, persyaratan, biaya, dan lainnya.

## **Pengumpulan Data**

Dalam tahapan ini pengumpulan data dilakukan melalui obesrvasi dan wanwancara untuk mencari tahu bagaimana admin dalam merespon pertanyaan terkait penerimaan peserta didik baru (PPDB) pada SMK Kesatrian Purwokerto. Dari hasil wawancara dan obersavasi menyimpulkan bahwa :

1. Pertanyaan yang masuk akan di balas satu persatu sesuai pesan yang masuk terlebih dahulu.
2. Batas waktu admin menjawab hanya sampai jam kerja selesai. Apabila ada pesan yang masuk di luar jam kerja maka akan dibalas keesokan harinya.
3. Pertanyaan yang sering ditanyakan oleh calon siswa, seperti informasi jurusan yang dibuka, jadwal pendaftaran, Proses pendaftaran secara online,dan nilai rata-rata yang digunakan.

## **Pengembangan *Chatbot***

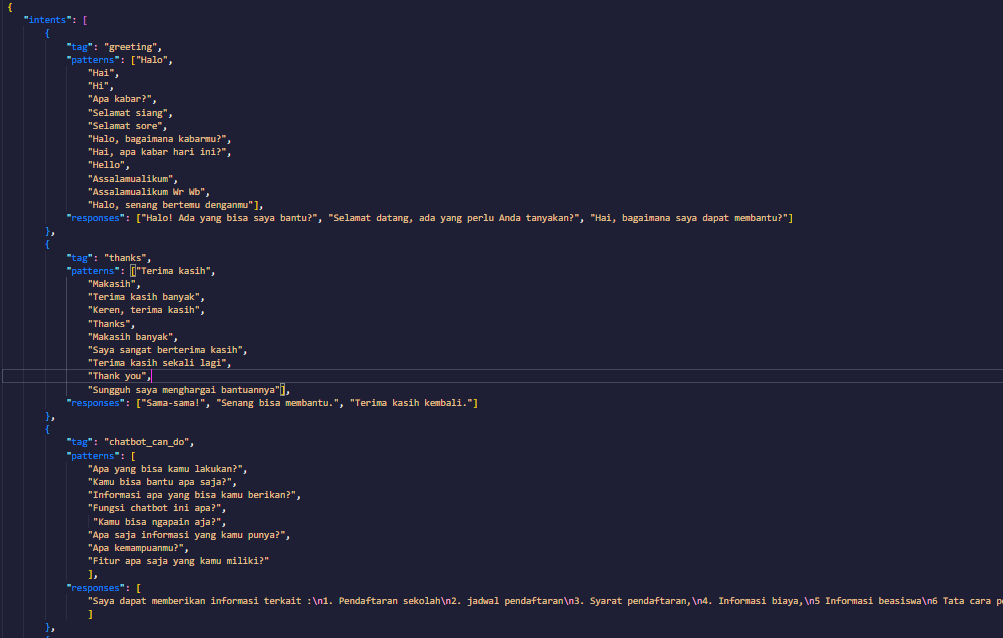
1. ***Data Collection***

Data yang sudah kita kumpulkan sebelumnya, terutama pertanyaan-pertanyaan yang sering muncul, akan diubah ke dalam format data JSON. Data ini didapatkan dari berbagai sumber, seperti brosur dan *website* PPDB SMK Kesatrian Purwokerto, serta dari hasil penelitian lainnya.

Proses selanjutnya melibatkan pembuatan dataset, dimana data dikumpulkan secara manual dan diubah menjadi format JSON. Dataset ini mencakup pertanyaan-pertanyaan yang di butuhkan untuk sistem membaca dan memproses setiap pola pertanyaan yang muncul, bersama dengan jawaban yang sesui. Dataset ini terdiri dari :

1. *Tag*, adalah kategori atau penanda untuk kumpulan pola-pola pertanyaan.
2. *Pattern*, berisi bagian pola pertanyaan yang diinginkan pengguna
3. *Respones*, yang berisi jawaban yang akan dihasilkan oleh sistem berdasarkan kombinasi dari tag dan pola yang telah ditentukan.

Pada penelitian ini, terdapat 280 dataset pola pertanyaan. Pada gambar 4.1 merupakan struktur dari dataset .



Gambar 4. 1 Struktur dataset JSO

Pada gambar 4.1 merupakan struktur dari dataset yang digunakan terdapat 3 bagian yaitu *tag*, *patterns,* dan *responses.* Sebagai contoh, tag "*greeting*" merupakan label yang mewakili maksud sapaan. Pada tag ini, terdapat beberapa pola pertanyaan pada bagian patterns, seperti"Hai", "Hi", "Apa kabar?", dan "Selamat siang". Jika *chatbot* mendeteksi salah satu pola tersebut, maka respons yang diberikan adalah "Halo! Ada yang bisa saya bantu?".

1. ***Text processing***
2. *case folding*

*Case folding* adalah teknik untuk menyeragamkan bentuk huruf dalam teks dengan mengubah semua huruf menjadi huruf kecil, mulai dari 'a' hingga 'z', dan menghilangkan karakter lainnya (Nugroho, 2019). Ini adalah langkah awal untuk *Text processing* menyamakan bentuk huruf dalam dokumen sehingga bentuk nya standar menjadi huruf kecil semua .



Gambar 4. 2 Kode Program Case Folding

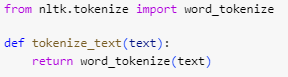
Pada gambar 4.2 merupakan *function* untuk *case folding* yang memiliki paramter (Text) yang dimana akan mengembalikan teks yang sudah di rubah menjadi huruf kecil menggunakan method .lower(). Untuk hasil *case folding* seperti pada tabel 4.1 .

Tabel 4. 1 Proses Case Folding

|  |  |
| --- | --- |
| **Sebelum *Case Folding*** | **Setelah *Case Folding*** |
| Kapan jadwal pendaftaran dibuka? | kapan jadwal pendaftaran dibuka? |
| Tanggal berapa PPDB dimulai? | tanggal berapa ppdb dimulai? |
| Di mana lokasi sekolah? | di mana lokasi sekolah? |

1. *Tokenizing*

Tokenisasi adalah proses penguraian kalimat menjadi unit-unit kata yang terpisah untuk memudahkan pengolahan dan analisis teks lebih lanjut. Pada tahapan ini seluruh kalimat pada dataset akan dipisah menjadi kata perkata.



Gambar 4. 3 Kode program Tokenizing

Pada gambar 4.3 merupakan Fungsi tokenize\_text bertujuan untuk memecah teks input menjadi kata-kata individual (token) menggunakan fungsi word\_tokenize dari *library* nltk, kemudian mengembalikan hasilnya dalam bentuk list yang berisi kata-kata tersebut, untuk kemudian diproses lebih lanjut dalam analisis teks. hasil dari *Tokenizing* pada gamabar 4.4.

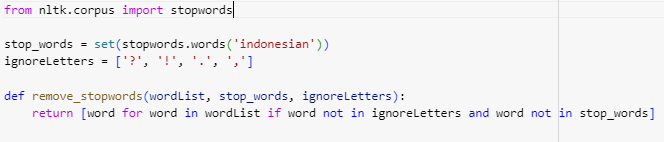


Gambar 4. 4 Hasil Tokenizing

Hasil tokenisasi pada gambar 4.2 menunjukkan proses memecah setiap kalimat menjadi unit-unit kecil yang disebut token, di mana setiap kata dan tanda baca seperti "?" dipisahkan sebagai token individual. Contohnya, kalimat "kapan jadwal pendaftaran dibuka?" diubah menjadi ['kapan', 'jadwal', 'pendaftaran', 'dibuka', '?'], dan pola serupa terlihat pada kalimat lainnya.

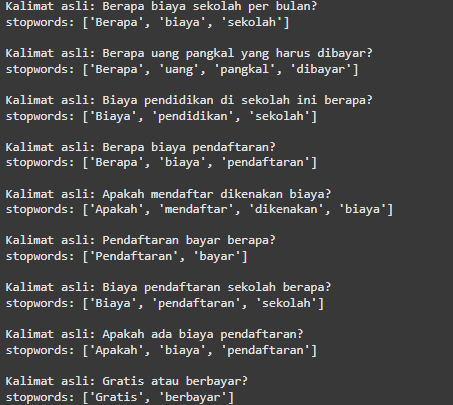
1. *Stopword Removal*

Tahapan selanjutnya melakukan *Stopword removal*  untuk menghapus kata kata yang di anggap tidak penting dan mengabaikan tanda baca seperti “ ? ”, ” ! ”, ” . ”, “ , ”. berikut kode program dari *Stopword removal* pada gambar 4.5.



Gambar 4. 5 Kode Program Stopword Removal

Pada gambar 4.5 mendefinisikan fungsi remove\_stopwords yang bertujuan untuk membersihkan suatu list kata dari *stop words* bahasa Indonesia dan tanda baca tertentu (?', '!', '.', ',') dengan memanfaatkan modul stopwords dari nltk.corpus, sehingga menghasilkan list kata yang lebih penting untuk analisis teks. Pada gambar 4.3 menunjukan hasil dari *stopword removal.*

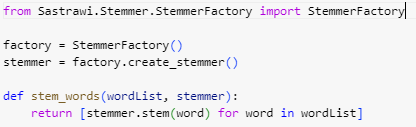


Gambar 4. 6 Hasil Stopword Removal

Hasil output pada gambar menunjukkan proses filtering yang menyeleksi kata-kata penting dari kalimat asli dengan menghapus kata-kata umum yang dianggap kurang relevan. Kalimat asli seperti "Berapa biaya sekolah per bulan?" menghasilkan kata-kata penting ['Berapa', 'biaya', 'sekolah'], di mana kata-kata seperti "per" dan "bulan" dihapus karena dianggap tidak signifikan, sementara kata kunci utama tetap dipertahankan.

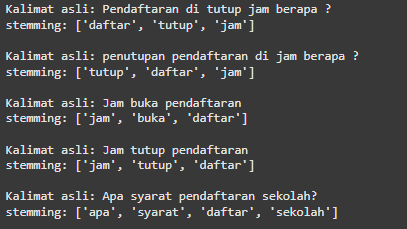
1. *Stemming*

Stemming adalah proses mengubah kata berimbuhan ke bentuk dasarnya tanpa harus menjadi akar kata (root word). Sebagai contoh, kata seperti "mendengarkan," "dengarkan," dan "didengarkan" akan diubah menjadi "dengar." (Nugroho, 2019).



Gambar 4. 7 Kode Program Stemming

Penelitian ini menggunakan *library* Sastrawi untuk melakukan stemming kata dalam bahasa Indonesia. Proses *stemming* dilakukan dengan menginstansiasi kelas StemmerFactory dan membuat objek *stemmer*. Fungsi *stem\_words* yang didefinisikan kemudian digunakan untuk melakukan *stemming* pada setiap kata dalam list kata. Hasil dari proses *stemming* seperti gambar 4.4 .



Gambar 4. 8 Hasil Stemming

1. **Modeling**
2. *Bag of Word*

Merubah hasil *Text Processing* menjadi bentuk index. Matriks yang dihasilkan dari proses ini digunakan sebagai input untuk model pembelajaran mesin atau analisis teks lebih lanjut. Hasil output dari *Bag of Word* seperti pada tabel 4.2.

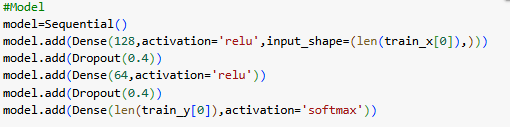
Tabel 4. 2 Hasil Bag of Word

|  |
| --- |
| Bag of Word |
| Pattern: ['formulir', 'daftar', 'mana']  Bag of Words: [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0] |

Output tersebut menunjukkan representasi *Bag of Words* untuk pola "Formulir pendaftaran ada dimana ?", yang telah diproses menjadi ['formulir', 'daftar', 'mana']. BoW adalah vektor biner yang menunjukkan keberadaan (1) atau ketidakhadiran (0) kata dalam kosakata. Pada pola ini, posisi kata 'formulir', 'daftar', dan 'mana' diisi dengan '1', sementara lainnya '0'. Representasi ini mempermudah model memahami data teks.

1. Implementasi Algoritma ANN

Dari proses *Bag of Words* merubah hasil *text processing* menjadi bentuk index agar bisa dipelajari oleh model.Model *chatbot* ini dirancang menggunakan pendekatan *Artificial Neural Network* (ANN) dengan arsitektur *Multi-Layer Perceptron* (MLP). Struktur ANN ini terdiri dari dua hidden layer Dense, masing-masing dengan 128 dan 64 neuron, menggunakan fungsi aktivasi ReLU. Untuk mengurangi risiko overfitting, diterapkan dua layer Dropout dengan tingkat dropout sebesar 0.4. Pada layer output, digunakan Dense layer dengan fungsi aktivasi softmax untuk klasifikasi multi-kelas, di mana jumlah neuron pada output layer disesuaikan dengan jumlah kelas dalam data pelatihan. Model *Artificial Neural Network* dapat dilihat pada gambar 4.9.



Gambar 4. 9 Kode Program Model ANN

Gambar 4.9 merupakan Kode program dari perancangan model *Artificial Neural Network.* Penjelasan dari kode program model diatas seperti berikut :

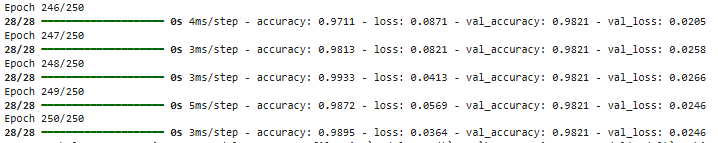
1. Inisialisasi model menggunakan Sequential API dari Keras, yang memungkinkan pemprosesan layer secara berurutan.
2. Layer pertama Lapisan Dense dengan 128 neuron ini menghubungkan seluruh fitur input ke setiap neuronnya, menggunakan fungsi aktivasi ReLU untuk memungkinkan jaringan mempelajari pola non-linear kompleks, dengan bentuk input yang sesuai dengan panjang vektor data pelatihan.
3. *Dropout layer* dengan *rate* 0.4 untuk mematikan 40% neuron selama proses pelatihan untuk menghindari *overfitting*.
4. *Hidden layer* ke 2 Lapisan Dense dengan 64 neuron berfungsi menyaring dan mengekstrak fitur-fitur penting dari data yang telah diolah sebelumnya, lalu menggunakan fungsi aktivasi ReLU untuk membantu model memahami pola-pola kompleks dalam data.
5. *Dropout layer* dengan rate 0.4 merupakan *dropout* pada layer ke 2 digunakan setelah *hidden layer 2.*
6. *Output layer Layer* dengan jumlah neuron mewakili satu kelas dalam klasifikasi. Fungsi aktivasi softmax menghasilkan probabilitas untuk setiap kelas, sehingga model dapat memprediksi kelas yang paling mungkin untuk input yang diberikan.
7. *Training model*

Setelah merancang model tahapan berikutnya adalah melakukan pelatihan pada model. Pelatihan model ini menggunakan *optimizer* **SGD (***Stochastic Gradient Descent***). Penerapan *optimizer* SGDseperti pada gambar 4.10.**



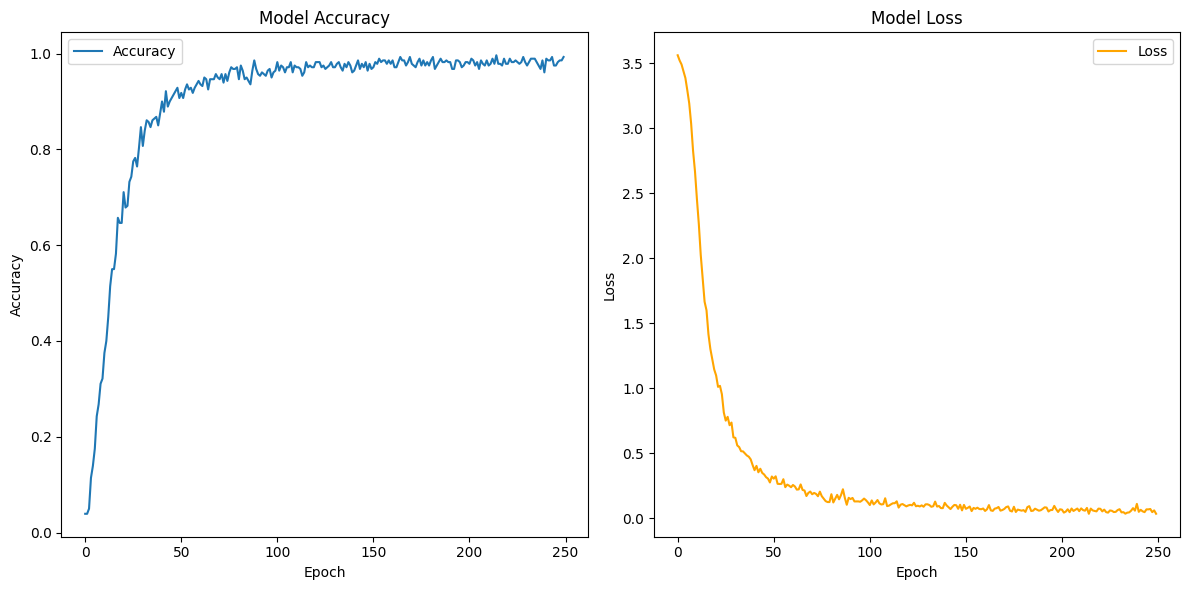
Gambar 4. 10 Kode Program Training model

Pada gambar 4.10 mendefinisikan optimizer SGD dengan learning rate 0.01 Menetapkan kecepatan pembaruan bobot dalam proses pelatihan, decay 1e-6 Mengurangi nilai learning rate secara bertahap untuk meningkatkan stabilitas pelatihan, momentum 0.9 Menambahkan momentum untuk mempercepat konvergensi dan mengurangi osilasi pada arah gradien, dan Nesterov diaktifkan, kemudian mengkompilasi model dengan fungsi kerugian *'categorical\_crossentropy'*, *optimizer* SGD, dan metrik 'accuracy'. selanjutnya model dilatih menggunakan data train\_x dan train\_y selama 300 epoch dengan batch size 10, dan menyimpan riwayat pelatihan dalam variabel history.



Gambar 4. 11 Hasil Pelatihan Model

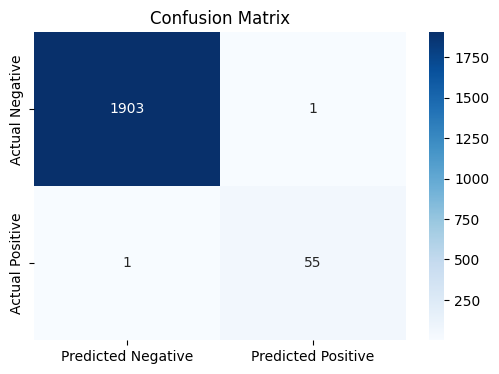
Dari gambar 4.11 menunjukan Hasil Pelatihan model dengan 250 epoch menunjukan bahwa model memiliki akurasi 98% dan *loss* yang relatif rendah. Ini menunjukkan bahwa model telah belajar dengan baik untuk mengklasifikasikan input dan memberikan respons yang tepat. Hasil proses pelatihan digambarkan dalam bentuk grafik seperti pada gambar 4.12.



Gambar 4. 12 Grafik Akurasi dan Loss

1. **Evaluasi**

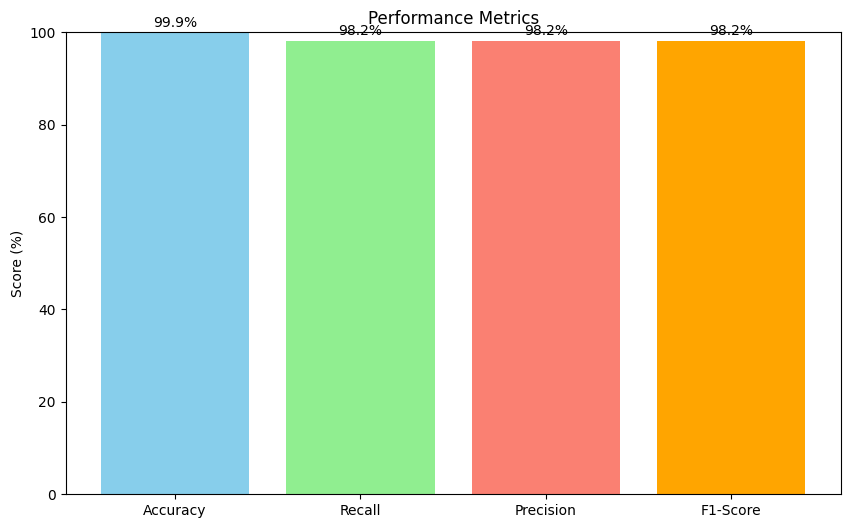
Selanjut melakukan evaluasi model menggunakan metode *confusion matriks* untuk mencari kinerja model melalui hasil akurasi, presisi, recall, dan F1-score. Untuk melakukan evaluasi data dibagi menjadi 80 – 20, 80% untuk data training dan 20% untuk data Berhasilasi atau data uji.



Gambar 4. 13 Hasil Counfusion Matriks

Pada gambar 4.13 menunjukan nilai confuison matriks dengan TP = 55, TN = 1903, FN = 1, FP = 1. selanjutnya akan dilakukan perhitungan untuk mecari nilai akurasi, recall, perisisi, dan f1-score.

Dari hasil perhitungan ditas mendapatkan hasil akurasi, *recall,* F1-score, dan presisi pada *chatbot* menggunakan model *Artificial nural network* adalah 99% untuk akurasi, 98% untuk nilai *Recal,* 98% untuk nilai presisi, dan 98% untuk nilai *F1-score.* Visualisai dari ke 4 matriks diatas dapat dilihat pada gambar 4.14.



Gambar 4. 14 Diagram Hasil Nilai Confusion Matriks

Hasil akurasi, presisi, *F1-score* dan *recall* dapat dianggap menunjukkan kualitas klasifikasi yang baik atau tidak dengan mengacu pada pedoman evaluasi hasil klasifikasi pada tabel 4.3 (Gorunescu, 2011).

Tabel 4. 3 Pedoman parameter hasil klasifikasi

|  |  |
| --- | --- |
| **Rentang** | **Hasil Klasifikasi** |
| 90 – 100% | *Excellent classification* |
| 80-90% | *Good classification* |
| 70-80% | *Fair classification* |
| 60-70% | *Poor classification* |
| 50-60% | *Failure* |

Hasil pelatihan model menunjukkan bahwa klasifikasi dataset menggunakan model Neural Network mendekati tingkat akurasi yang sempurna. Hal ini mengindikasikan bahwa model tersebut memiliki performa yang sangat baik dalam membedakan data dan mengelompokkannya ke dalam kategori atau kelas yang sesuai dengan benar.

1. ***Deployment***

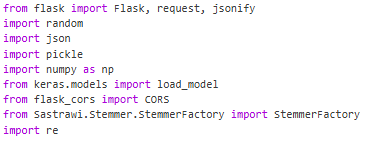
Pada tahap *deployment* ini, tujuan utamanya adalah untuk membangun antarmuka tampilan *website* PPDB dan *chatbot*. Pengembangan antarmuka ini menggunakan *Bootstrap* sebagai *frontend* dan *framework* Flask untuk membangun API respons dari *chatbot* untuk menciptakan antarmuka situs web yang menarik dan responsif.

1. Tahap pertama buat file projek dengan memasukan file seperti *chatbot*­\_model.h5, word.pkl, classes.pkl, dan intents.json yang merupkan hasil pelatihan model sebelumnya.



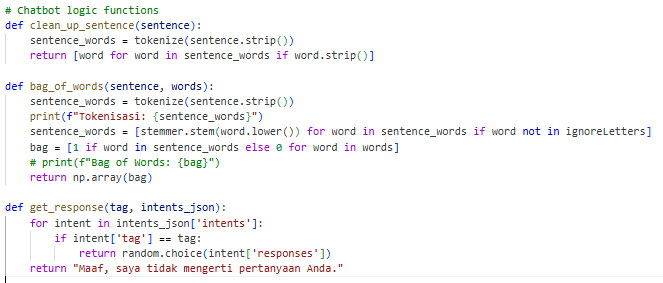
Gambar 4. 15 Struktur File

1. Pada app.py mempersiapkan *libray* yang digunakan dan membuat function *chatbot* respon dan API nya menggunakan *framework* flask



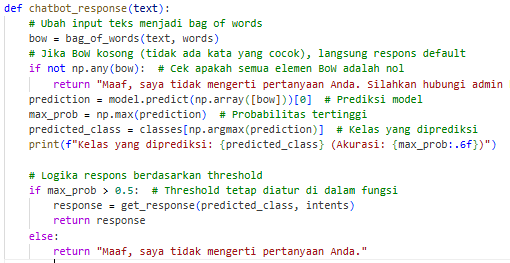
Gambar 4. 16 Libray yang digunakan

*Library* yang digunakanmencakup Flask untuk membangun API *backend*, Flask-CORS untuk mendukung *Cross-Origin Resource Sharing*, Keras untuk memuat model *machine learning* yang telah dilatih (dengan fungsi load\_model), NumPy untuk manipulasi array seperti pembentukan *bag-of-words*, Pickle untuk menyimpan dan memuat data serialisasi seperti *words*.pkl dan *classes*.pkl, serta Sastrawi untuk stemming kata-kata dalam bahasa Indonesia. Selain itu, *library* random digunakan untuk memilih respons acak, json untuk membaca dan memproses file JSON, dan re untuk manipulasi teks dengan ekspresi reguler.



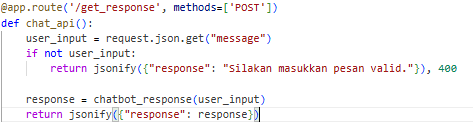
Gambar 4. 17 Sourcode Logic Chatbot Respon

Kode program di atas adalah logika *chatbot* untuk memproses input pengguna. Fungsi *clean\_up\_sentence* membersihkan kalimat dengan tokenisasi menggunakan *tokenize* dan menghilangkan kata kosong. Fungsi *bag\_of\_words* mengubah kalimat menjadi representasi numerik berbasis *Bag of Words* dengan melakukan *stemming* dan mencocokkan kata dalam daftar kata (*words*). Fungsi *get\_response* mengambil respon dari file intents\_json berdasarkan tag yang diprediksi, dan jika tag tidak ditemukan, mengembalikan pesan default. Semua fungsi ini bekerja sama untuk memproses input teks dan menghasilkan respon yang relevan dari *chatbot*.



Gambar 4. 18 Sourcode Function chatbot\_response

Fungsi *chatbot\_response* memproses input teks dengan mengubahnya menjadi *Bag of Words* (BoW) untuk prediksi kelas menggunakan model yang telah dilatih. Jika BoW kosong atau probabilitas prediksi (*max\_prob*) di bawah threshold 0.5, *chatbot* mengembalikan respon *default*. Jika probabilitas mencukupi, fungsi mengambil respon yang sesuai dengan kelas prediksi dari file intent.



Gambar 4. 19 Sourcode API Chatbot

Fungsi *chat\_api* adalah *endpoint* Flask untuk menerima input teks pengguna melalui POST *request*. Jika input kosong, fungsi mengembalikan respon *error*. Jika Berhasil, fungsi memproses input menggunakan *chatbot\_response* dan mengembalikan respon *chatbot* dalam format JSON.

1. Membuat tampilan *chatbot* menggunakan *bootstrap.*

|  |
| --- |
| <!-- *Chatbot* Container -->  <div class="chat-container card" id="chat-container" style="display: none;">  <div class="card-header bg-success text-white d-flex justify-content-between">  <h5 class="mb-0">*Chatbot*</h5>  <button id="close-*chatbot*" class="btn btn-sm btn-light">×</button>  </div>  <div class="card-body overflow-auto" id="chat-box" style="max-height: 300px;">  <!-- Chat messages will appear here -->  </div>  <div class="card-footer d-flex">  <input type="text" id="user-input" class="form-control me-2" placeholder="Ketik pesan Anda...">  <button id="send-btn" class="btn btn-success">Kirim</button>  </div>  </div> |

Gambar 4. 20 Sourcode Chatbot Container

1. Setting API *chatbot* respon pada *javascript*

|  |
| --- |
| // Fetch *chatbot* response  fetch("https://habdulghani-model-*chatbot*.hf.space/get\_response", {  method: "POST",  headers: {  "Content-Type": "application/json"  },  body: JSON.stringify({ message: userMessage })  })  .then(response => response.json())  .then(data => {  // Remove waiting message  chatBox.removeChild(waitingDiv);    const botMessageDiv = document.createElement("div");  botMessageDiv.className = "bot-message";    // Replace \n with <br> to ensure new lines are rendered correctly  let formattedResponse = data.response.replace(/\n/g, '<br>');    // Check if there is a link and make it clickable  formattedResponse = formattedResponse.replace(  /(https?:\/\/[^\s]+)/g,  '<a href="$1" target="\_blank">$1</a>'  ); |

Gambar 4. 21 Sourcode Setting API

Kode program diatas adalah untuk implementasi JavaScript untuk mengirim permintaan ke API *chatbot* menggunakan metode *fetch* dengan metode POST. Data dikirim dalam format JSON berisi pesan pengguna (*userMessage*). Setelah menerima respon dari API dalam format JSON, pesan bot akan diproses untuk menggantikan karakter *newline* (\n) dengan tag <br> agar tampilan baris baru terlihat di HTML. Selain itu, jika respon berisi tautan (URL), tautan tersebut akan diubah menjadi tautan yang dapat diklik menggunakan elemen <a> dengan atribut target="\_*blank*" untuk membukanya di tab baru. Pesan bot yang sudah diformat kemudian ditambahkan ke elemen *chat box* di halaman.

1. Hasil tampilan halaman depan *website*



Gambar 4. 22 Tampilan Awal Website

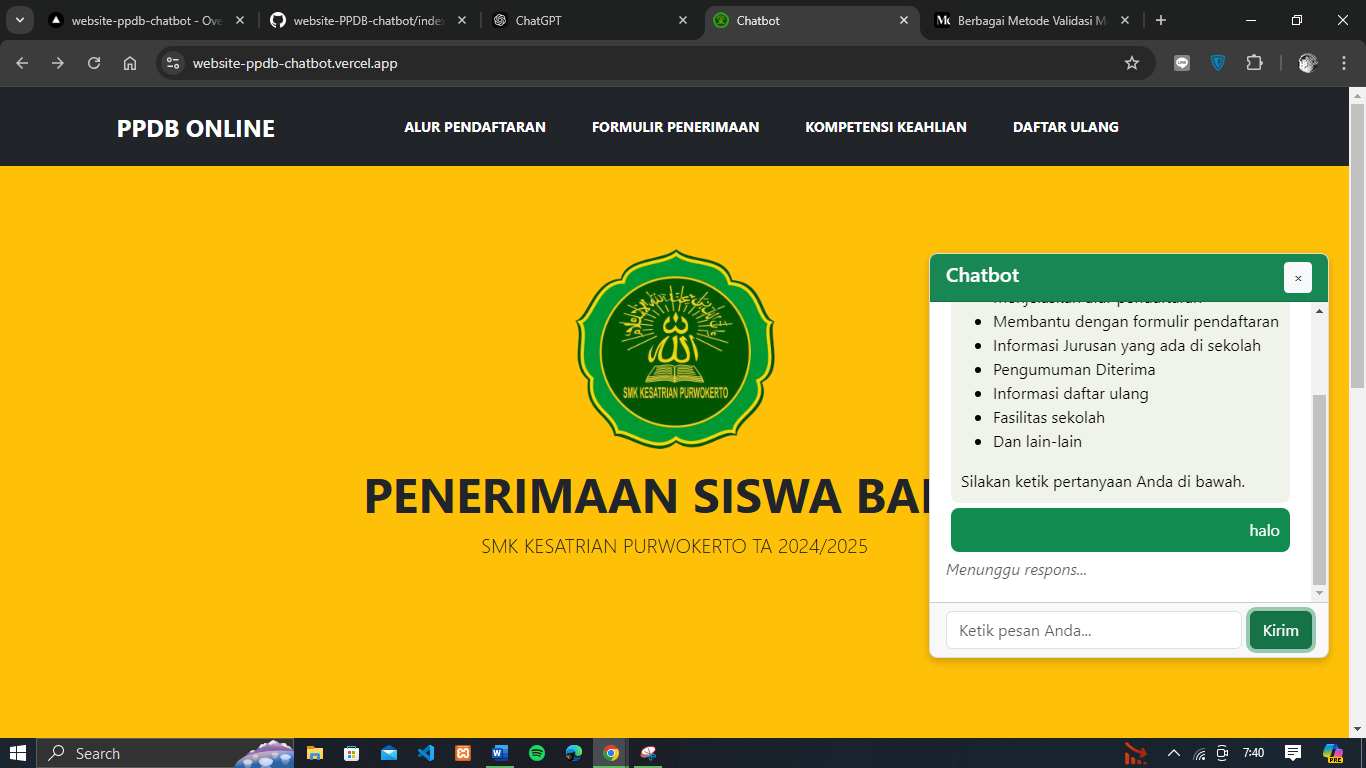
Halaman awal website ini dirancang menyerupai tampilan asli dari website PPDB resmi SMK Kesatrian Purwokerto, dengan tambahan fitur tombol chat di pojok kanan bawah. Tombol chat ini berfungsi untuk mengakses fitur chatbot yang bertujuan memfasilitasi pengujian dan interaksi dengan chatbot secara langsung.

1. Tampilan awal *chatbot*



Gambar 4. 23 Tampilan Awal Chatbot

1. Tampilan *Chatbot* saat menunggu respon



Gambar 4. 24 Tampilan Chatbot saat menunggu respon

1. Tampilan saat memberi respon



Gambar 4. 25 Tampilan Chatbot saat memberi respon

1. Tampilan error



Gambar 4. 26 Tampilan error

1. **Pengujian**

Tahapan pengujian ini bertujuan untuk memastikan bahwa aplikasi *chatbot* berfungsi sesuai dengan kebutuhan dan spesifikasi yang telah dirancang. Pengujian dilakukan dengan metode *Blackbox Testing*, di mana fokus utama adalah menguji keluaran yang dihasilkan berdasarkan masukan tertentu tanpa memeriksa kode internal aplikasi.

Skenario pengujian ini dengan menggunakan salah satu pattern pada setiap tag yang ada pada dataset. Jumlah tag pada daset sejumlah 35 tag yang bersisi konteks pertanyaan terkait PPDB Pengujian ini menggunakan 35 pertanyaan inputan sesuai tag dan 2 inputan error hasil pengujian balckbox dapat dilihat pada tabel 4.

Tabel 4. 4 Pengujian Blackbox

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Input pertanyan | Respon tag | Hasil respon yang diharapkan | Keterangan |
| 1 | Apa kabar | *greeting* | *greeting* | Berhasil |
| 2 | Terimakasih | *thanks* | *thanks* | Berhasil |
| 3 | Apa yang bisa kamu lakukan? | *chatbot\_can\_do* | *chatbot\_can\_do* | Berhasil |
| 4 | baiklah | *agree* | *agree* | Berhasil |
| 5 | Kapan jadwal pendaftaran dibuka? | jadwal\_pendaftaran | jadwal\_pendaftaran |  |
| 6 | Jam berapa pendaftaran dibuka ? | jambuka\_tutup | jambuka\_tutup | Berhasil |
| 7 | Apa syarat pendaftaran sekolah? | syarat\_pendaftaran | syarat\_pendaftaran | Berhasil |
| 8 | cara mengisi formulir pendaftaran ? | formulir\_pendaftaran | formulir\_pendaftaran | Berhasil |
| 9 | Biaya spp | spp\_dan\_uang\_masuk | spp\_dan\_uang\_masuk | Berhasil |
| 10 | Biaya pendaftaran | biaya\_pendaftaran | biaya\_pendaftaran | Berhasil |
| 11 | Jurusan yang dibuka | jurusan\_yang\_dibuka | jurusan\_yang\_dibuka | Berhasil |
| 12 | Di mana lokasi sekolah? | lokasi\_sekolah | lokasi\_sekolah | Berhasil |
| 13 | Fasilitas sekolah ? | fasilitas\_sekolah | fasilitas\_sekolah | Berhasil |
| 14 | Akreditasi sekolah apa? | akreditasi\_sekolah | akreditasi\_sekolah | Berhasil |
| 15 | Nilai rata-rata minimal berapa | nilai\_pendaftaran | nilai\_pendaftaran | Berhasil |
| 16 | Apa itu jurusan DKV ? | jurusan\_desain\_komunikasi\_visual | jurusan\_desain\_komunikasi\_visual | Berhasil |
| 17 | apa itu tkr ? | jurusan\_teknik\_kendaraan\_ringan | jurusan\_teknik\_kendaraan\_ringan | Berhasil |
| 18 | Apa itu jurusan TSM? | jurusan\_teknik\_sepeda\_motor | jurusan\_teknik\_sepeda\_motor | Berhasil |
| 19 | Apa itu TAV | jurusan\_teknik\_audio\_video | jurusan\_teknik\_audio\_video | Berhasil |
| 20 | Apa iru TKJ | jenjang\_karir\_tkj | jurusan\_teknik\_komputer\_dan\_jaringan | Tidak berhasil |
| 21 | Bagaimana cara mendaftar? | alur\_pendaftaran | alur\_pendaftaran | Berhasil |
| 22 | cara daftar ulang ? | daftar\_ulang | daftar\_ulang | Berhasil |
| 23 | Apa saja berkas yang di butuhkan untuk daftar ulang ? | berkas\_daftar\_ulang | berkas\_daftar\_ulang | Berhasil |
| 24 | Kapan pengumuman hasil seleksi? | pengumuman\_diterima | pengumuman\_diterima | berhasil |
| 25 | ukuran pas foto | ukuran\_pas\_foto | ukuran\_pas\_foto | Berhasil |
| 26 | visi misi | visi\_misi\_sekolah | visi\_misi\_sekolah | Berhasil |
| 27 | berapa jumlah siswa sekarang | jumlah\_siswa\_aktif | jumlah\_siswa\_aktif | Berhasil |
| 28 | Informasi beasiswa | beasiswa | beasiswa | Berhasil |
| 29 | Prospek kerja DKV? | jenjang\_karir\_dkv | jenjang\_karir\_dkv | Berhasil |
| 30 | Prospek kerja TAV? | jenjang\_karir\_tav | jenjang\_karir\_tav | Berhasil |
| 31 | Prospek kerja TKJ | jenjang\_karir\_tkj | jenjang\_karir\_tkj | Berhasil |
| 32 | Prospek kerja TSM? | jenjang\_karir\_tsm | jenjang\_karir\_tsm | Berhasil |
| 33 | Berapa biaya per jurusan? | biaya\_perjurusan | biaya\_perjurusan | Berhasil |
| 34 | Apa itu PPDB? | apa\_itu\_ppdb | apa\_itu\_ppdb | Berhasil |
| 35 | Prospek kerja TKR? | jenjang\_karir\_tkr | jenjang\_karir\_tkr | Berhasil |
| 36 | Kaki saya sakit | error | error | Berhasil |
| 37 | akwjadja | error | error | Berhasil |

Dari hasil pengujian *blackbox* dapat dilihat *chatbot* memberikan 36 respon sesuai prediksi dan 1 gagal. Selanjutnya menghitung akurasi dari hasil pengujian dengan menghitung jumlah prekdiksi bener dibagi jumlah keseluruhan pengejuian (Rina, 2023).

Hasil akurasi dari pengujian *blackbox* mendapatkan hasil 97%. Ini menandakan bahwa *chatbot* dapat memberikan respon yang sesaui dengan inputan pertanyaan dari pengguna.

**BAB V  
PENUTUP**

## **Kesimpulan**

Kesimpulan merupakan pernyataan singkat, jelas, dan tepat tentang apa yang diperoleh, memuat keunggulan dan kelemahan, dapat dibuktikan, serta terkait langsung dengan Rumusan Masalah dan Tujuan Penelitian. Uraian pada bagian ini harus merupakan pernyataan yang pernah dianalisis/dibahas pada bagian sebelumnya, bukan pernyataan yang sama sekali baru dan tidak pernah dibahas pada bagian sebelumnya, serta merupakan jawaban atas permasalahan yang dirumuskan. Bagian ini tidak perlu ada uraian penjelasan lagi.

## **Saran**

Saran memuat berbagai usulan atau pendapat yang sebaiknya dikaitkan oleh penelitian sejenis. Saran dibuat berdasarkan kelemahan, pengalaman, kesulitan, kesalahan, temuan baru yang belum diteliti dan berbagai kemungkinan arah penelitian selanjutnya. Saran yang disusun harus didasarkan atas hasil penelitian yang dilakukan.

# **DAFTAR PUSTAKA**

Arbizal, Y., Nurina Sari, B., & Garno. (2024). IMPLEMENTASI ALGORITMA *ARTIFICIAL NEURAL NETWORK* PADA *CHATBOT* *WEBSITE* PRODI INFORMATIKA UNSIKA. *Jurnal informasi dan Komputer*, *12*(1), 2024.

Enterprise, J. (2019). *Python untuk Programmer Pemula*. Elex media komputindo.

Fadhallah. (2021). *WAWANCARA*. UNJ PRESS.

Fahmi Yusron Fiddin, Komarudin, A., & Melina, M. (2024). *Chatbot* Informasi Penerimaan Mahasiswa Baru Menggunakan Metode FastText dan LSTM. *Journal of Applied Computer Science and Technology*, *5*(1), 33–39. https://doi.org/10.52158/jacost.v5i1.648

Faurina, R., Revanza, D., & Sopran, A. (2023). Pengembangan *Chatbot* Menggunakan Deep Feed-Forward Neural Network sebagai Pusat Layanan Informasi Akademik. *Jurnal Eksplora Informatika*, *11*(2), 120–129. https://doi.org/10.30864/eksplora.v11i2.833

Gorunescu, F. (2011). *Data Mining Concepts, Models and Techniques*. Springer Berlin Heidelberg.

Hikmah, A., Azmi, F., & Nugrahaeni, R. A. (2023). Implementasi Natural Language Processing Pada *Chatbot* Untuk Layanan Akademik. *e-Proceeding of Engineering*.

Iskandar, D., & Sriharyani, L. (2021). SOFT COMPUTING PENILAIAN KONDISI PERKERASAN JALAN BERBASIS *ARTIFICIAL NEURAL NETWORK*S. *TAPAK (Teknologi Aplikasi Konstruksi) : Jurnal Program Studi Teknik Sipil*, *10*(2), 148. https://doi.org/10.24127/tp.v10i2.1584

Ivan, G., Hadi Asnal, Muhammad Nur Cahyadi, & Zaki Mubarok G. (2022). Perancangan *Chatbot* untuk Layanan Informasi Sekolah (Studi Kasus SMK Dwi Sejahtera Pekanbaru). *SATIN - Sains dan Teknologi Informasi*, *8*(2), 198–207. https://doi.org/10.33372/stn.v8i2.880

Krstinic, D., Braović, M., Šerić, L., & Božić-Štulić, D. (2020). *Multi-label Classifier Performance Evaluation with Confusion Matrix*. 1–14. https://doi.org/10.5121/csit.2020.100801

Kurniyawan, D. (2022). *Pengenalan Machine Learning dengan Python*. Elex Media Komputindo. https://books.google.co.id/books?id=ZutsEAAAQBAJ

Larasati, S., & Susetyo, Y. (2024). Development of a Web-Based Trading Term Application Using Flask Framework at PT. XYZ. *International Journal Software Engineering and Computer Science (IJSECS)*, *4*, 367–376. https://doi.org/10.35870/ijsecs.v4i1.2339

Ling, Q. (2023). Machine learning algorithms review. *Applied and Computational Engineering*, *4*(1), 91–98. https://doi.org/10.54254/2755-2721/4/20230355

Lubis, A., & Sumartono, I. (2023). RESOLUSI : Rekayasa Teknik Informatika dan Informasi Implementasi Layanan Akademik Berbasis *Chatbot* untuk Meningkatkan Interaksi Mahasiswa. *Media Online*, *3*(5). https://doi.org/https://doi.org/10.30865/resolusi.v3i5.767

Mustakim, F., Fauziah, & Hayati, N. (2021). Algoritma *Artificial Neural Network* pada Text-based *Chatbot* Frequently Asked Question (FAQ) Web Kuliah Universitas. *Jurnal Teknologi Informasi dan Komunikasi)*, *5*(4), 2021. https://doi.org/10.35870/jti

Nugraha, K. A., & Sebastian, D. (2021). *Chatbot* Layanan Akademik Menggunakan K-Nearest Neighbor. *Jurnal Sains dan Informatika*, *7*(1), 11–19. https://doi.org/10.34128/jsi.v7i1.285

Nugroho, K. S. (2019). *Dasar Text Preprocessing dengan Python*. https://ksnugroho.medium.com/dasar-text-preprocessing-dengan-python-a4fa52608ffe.

Nurul Puteri, A., Tamrin, F., Rahman Nasir, K., Widya Anggraeni, D., & Arafah, M. (2022). Aplikasi *Chatbot* untuk Layanan Informasi Penerimaan Mahasiswa Baru. *Seminar Nasional Teknik Elektro dan Informatika (SNTEI)*.

Purnajaya, A. R., Lieputra, V., Tayanto, V., & Salim, J. G. (2022). Implementasi Text Mining untuk Mengetahui Opini Masyarakat Tentang Climate Change. *Journal of Information System and Technology*, *03*(03), 320–328.

Purwono, P., Dewi, P., Wibisono, S. K., & Dewa, B. P. (2022). Model Prediksi Otomatis Jenis Penyakit Hipertensi dengan Pemanfaatan Algoritma Machine Learning *Artificial Neural Network*. *Insect (Informatics and Security): Jurnal Teknik Informatika*, *7*(2), 82–90. https://doi.org/10.33506/insect.v7i2.1828

Ramadhani, S., Azzahra, D., & Z, T. (2022). Comparison of K-Means and K-Medoids Algorithms in Text Mining based on Davies Bouldin Index Testing for Classification of Student’s Thesis. *Digital Zone: Jurnal Teknologi Informasi dan Komunikasi*, *13*(1), 24–33. https://doi.org/10.31849/digitalzone.v13i1.9292

Rina. (2023). *Memahami Confusion Matrix: Accuracy, Precision, Recall, Specificity, dan F1-Score untuk Evaluasi Model Klasifikasi*. https://esairina.medium.com/memahami-confusion-matrix-accuracy-precision-recall-specificity-dan-f1-score-610d4f0db7cf.

Rohmah, S., Wahyudi, W., & Pamungkas, F. (2021). Pengelolaan Penerimaan Peserta Didik Baru (PPDB) Berdasarkan Sistem Zonasi di SMP Negeri 1 Mlonggo Jepara. *Jawda: Journal of Islamic Education Management*, *1*(1), 25–34. https://doi.org/10.21580/jawda.v1i1.2020.6704

Rosad, A. M. (2019). IMPLEMENTASI PENDIDIKAN KARAKTER MELALUI MANAGEMEN SEKOLAH. *Tarbawi: Jurnal Keilmuan Manajemen Pendidikan*, *5*(02), 173. https://doi.org/10.32678/tarbawi.v5i02.2074

Sanjaya, M. O., Bukhori, S., & Furqon, M. `Ariful. (2023). Virtual Assistant for Thesis Technical Guide Using *Artificial Neural Network*. *Indonesian Journal of Artificial Intelligence and Data Mining*, *6*(2), 188. https://doi.org/10.24014/ijaidm.v6i2.23473

Sasongko, B. B., Malik, F., Ardiansyah, F., Rahmawati, A. F., Dharma Adhinata, F., & Rakhmadani, D. P. (t.t.). Pengujian Blackbox Menggunakan Teknik Equivalence Partitions pada Aplikasi Petgram Mobile. Dalam *Jurnal ICTEE* (Vol. 2, Nomor 1).

Sidik, M., Gunawan, B., Anggraini, D., & Korespondensi, P. (2021). PEMBUATAN APLIKASI *CHATBOT* KOLEKTOR DENGAN METODE EXTREME PROGRAMMING DAN STRATEGI FORWARD CHAINING. *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK)*, *8*(2), 293–302. https://doi.org/10.25126/jtiik.202184298

Sugiyono. (2018). *Metode penelitian kuantitatif / Prof. Dr. Sugiyono* (Cet. 1). Alfabeta.

Sujacka Retno, Rozzi Kesuma Dinata, & Novia Hasdyna. (2023). Evaluasi model data *chatbot* dalam natural language processing menggunakan k-nearest neighbor. *Jurnal CoSciTech (Computer Science and Information Technology)*, *4*(1), 146–153. https://doi.org/10.37859/coscitech.v4i1.4690

Suprapto, & Malik, A. . A. (2019). IMPLEMENTASI KEBIJAKAN DISKRESI PADA PELAYANAN KESEHATAN BADAN PENYELENGGARA JAMINAN KESEHATAN (BPJS). *Jurnal Ilmiah Kesehatan Sandi Husada*. https://akper-sandikarsa.e-journal.id

Syukri, S., & Samsuddin, S. (2019). Pengujian Algoritma *Artificial Neural Network* (ANN) Untuk Prediksi Kecepatan Angin. *Jurnal Nasional Komputasi dan Teknologi Informasi (JNKTI)*, *2*(1), 43. https://doi.org/10.32672/jnkti.v2i1.1056

Telaumbanua, F. D., Hulu, P., Nadeak, T. Z., Lumbantong, R. R., & Dharma, A. (2020). Penggunaan Machine Learning Di Bidang Kesehatan. *Jurnal Teknologi dan Ilmu Komputer Prima (JUTIKOMP)*, *2*(2). https://doi.org/10.34012/jutikomp.v2i2.657

Yudahana, A., Riadi, I., & Elvina, A. (2023). PERANCANGAN SISTEM INFORMASI PENDAFTARAN PESERTA DIDIK BARU (PPDB) BERBASIS WEB MENGGUNAKAN METODE RAPID APLLICATION DEVELOPMENT (RAD). *Rabit : Jurnal Teknologi dan Sistem Informasi Univrab*, *8*(1), 47–58. https://doi.org/10.36341/rabit.v8i1.2977

Zuraiyah, T. A., Utami, D. K., & Herlambang, D. (2019). IMPLEMENTASI *CHATBOT* PADA PENDAFTARAN MAHASISWA BARU MENGGUNAKAN RECURRENT NEURAL NETWORK. *Jurnal Ilmiah Teknologi dan Rekayasa*, *24*(2), 91–101. https://doi.org/10.35760/tr.2019.v24i2.2388

# **LAMPIRAN**

Daftar lampiran berisi kartu bimbingan, surat keterangan dari instansi tempat penelitian, Blangko Kuesioner, instrument penelitian, listing program, peraturan-peraturan dan sebagainya yang berfungsi melengkapi laporan penelitian