

Implementasi Chatbot Natural Language Processing untuk Sistem Pendaftaran Siswa: Studi Kasus di SMP Al Washliyah 9 Medan

Implementation of Natural Language Processing Chatbot for Student Registration System: A Case Study at SMP Al Washliyah 9 Medan

Aidil Bahri Panggabean¹, Lahmudin Sipahutar²

Informatika Universitas Potensi Utama, K.L. Yos Sudarso KM 6,5 No. 3A Tj. Mulia, Medan, 20241 – 2-244, Indonesia

aidilbahri2003@gmail.com¹, mudinsipa@gmail.com²

ABSTRAK

Dalam era digital, efisiensi pendaftaran siswa menjadi krusial, terutama di SMP Al Washliyah 9 Medan. Penelitian ini mengembangkan chatbot berbasis Natural Language Processing (NLP) untuk mempermudah proses pendaftaran. Chatbot dirancang untuk memberikan informasi akurat dan memandu calon siswa serta orang tua dari pengisian data hingga pengunggahan dokumen. Menggunakan Python dan Flask, pengujian menunjukkan chatbot mampu menjawab pertanyaan dengan akurasi tinggi dan memproses pendaftaran secara otomatis dengan cepat. Implementasi ini meningkatkan efisiensi pendaftaran, mengurangi beban kerja staf, dan memberikan kemudahan bagi pengguna. Chatbot ini diharapkan menjadi solusi efektif untuk pendaftaran siswa di sekolah-sekolah lain.

Kata Kunci : *Natural Language Processing, Chatbot, Pendaftaran Siswa, Otomatisasi, SMP Al Washliyah 9 Medan*

ABSTRACT

In the digital era, the efficiency of student registration is crucial, especially at SMP Al Washliyah 9 Medan. This research develops a Natural Language Processing (NLP)-based chatbot to simplify the registration process. The chatbot is designed to provide accurate information and guide prospective students and parents from data entry to document upload. Using Python and Flask, testing showed the chatbot was able to answer questions with high accuracy and process enrolments automatically on the fly. This implementation improves registration efficiency, reduces staff workload, and provides convenience for users. This chatbot is expected to be an effective solution for student enrolment in other schools.

Keywords : *Natural Language Processing, Chatbot, Student Registration, Automation, SMP Al Washliyah 9 Medan*

Info Artikel :

Disubmit: 11 Agustus 2024

Direview: 06 September 2024

Diterima : 15 Oktober 2024

Copyright © 2024 - Journal UPU. All rights reserved.

1. PENDAHULUAN

Dalam era digital saat ini, teknologi kecerdasan buatan (Artificial Intelligence) semakin berperan penting dalam meningkatkan efisiensi dan kualitas layanan di berbagai sektor, termasuk pendidikan. Salah satu penerapan teknologi ini adalah chatbot berbasis Natural Language Processing (NLP), yang memungkinkan interaksi otomatis antara sistem dan pengguna melalui bahasa alami. Chatbot telah terbukti efektif dalam merespons permintaan pengguna dengan cepat dan akurat, menjadikannya alat yang sangat berguna dalam berbagai aplikasi, seperti layanan pelanggan, perbankan, dan pendidikan[1].

Implementasi chatbot dalam sektor pendidikan, khususnya untuk proses pendaftaran siswa baru, telah menunjukkan dampak positif. Penelitian sebelumnya mengindikasikan bahwa chatbot dapat mengurangi

beban kerja administratif dengan menjawab pertanyaan umum secara otomatis, menyediakan informasi, dan bahkan membimbing pengguna melalui proses pendaftaran[2]. Sebagai contoh, sistem pendaftaran berbasis chatbot mampu merespons pertanyaan pendaftar dengan tingkat akurasi hingga 88% dan mengurangi waktu tunggu secara signifikan, menjadikannya solusi efektif bagi lembaga yang menerima volume pendaftaran tinggi[3].

Dengan memanfaatkan NLP, chatbot dapat meningkatkan pemahaman terhadap maksud pengguna dan menyesuaikan jawabannya secara lebih efektif. Hal ini memungkinkan proses pendaftaran berjalan lebih efisien tanpa tergantung pada batasan waktu operasional, yang dapat menjadi keunggulan besar bagi lembaga pendidikan. Sebagai contoh, penelitian menunjukkan bahwa chatbot dapat mengotomatiskan proses pendaftaran, memungkinkan staf untuk fokus pada tugas yang lebih membutuhkan interaksi manusia, seperti memberikan konsultasi atau pengawasan. Di samping itu, penggunaan chatbot berpotensi besar dalam mendukung transformasi digital di dunia pendidikan, yang tidak hanya mengoptimalkan alur kerja tetapi juga memberikan kemudahan bagi pendaftar dengan menyediakan informasi yang cepat dan akurat kapan saja diperlukan. Teknologi ini diharapkan menjadi solusi berkelanjutan dalam memenuhi kebutuhan informasi pendaftar dan meningkatkan pengalaman pengguna secara keseluruhan[4], [5].

2. METODE

Proyek ini adalah chatbot berbasis web yang dikembangkan menggunakan Python dan pustaka-pustaka yang relevan. Framework Flask digunakan sebagai basis aplikasi web, sementara alat-alat untuk Pemrosesan Bahasa Alami disediakan oleh NLTK (Natural Language Tool Kit). Model deep learning dibangun menggunakan Keras, dan data serta model disimpan dalam format JSON dan file biner agar mudah diakses dan disesuaikan dengan kebutuhan pendaftaran siswa. Chatbot ini menyediakan langkah-langkah pendaftaran secara otomatis, mulai dari pengumpulan data hingga pengunggahan berkas pendaftaran dalam format JPG. Selain itu, chatbot juga memungkinkan pengguna mengunduh data pendaftaran dalam format PDF.

A. Persyaratan Perangkat Keras dan Perangkat Lunak

- ✓ PC dengan RAM minimal 4GB dan Penyimpanan minimal 2GB
- ✓ Windows 10 atau lebih baru / macOS 10.15 atau lebih baru
- ✓ Visual Studio Code editor
- ✓ Python versi 3.9 direkomendasikan sesuai dengan penulis gunakan
- ✓ Framework Flask, Nltk, Keras, Numpy dan Fpdf
- ✓ Browser Google Chrome / Mozilla Firefox / Microsoft Edge.

B. NLTK - Natural Language Tool Kit

Ini adalah pustaka Python gratis dan open-source yang banyak digunakan dalam NLP (Natural Language Processing)[6], [7]. Ini menyediakan berbagai metode, fungsi, dan model prebuilt yang dapat digunakan untuk melakukan berbagai operasi analisis teks dan bahasa. Ini banyak digunakan dalam membangun semua jenis model pembelajaran mesin analisis teks, bahasa, dan sentimen[8]. NLTK digunakan dalam proyek ini untuk melakukan tokenizing, stemming, dan pembuatan kantong kata[9]. Operasi-operasi ini dilakukan dengan bantuan paket NLTK khusus yang disebut "punkt"[7].

C. Json

JSON (JavaScript Object Notation) digunakan sebagai format untuk menyimpan dan mengelola data intensi (maksud) pengguna. File JSON berfungsi sebagai database ringan yang menyimpan pasangan pertanyaan dan jawaban serta pola-pola percakapan yang mungkin terjadi selama interaksi dengan chatbot[10], [11]. Dalam file data.json, setiap entri mengandung kategori intensi tertentu, seperti "pendaftaran" atau "informasi umum," yang dikaitkan dengan berbagai pola kata dan respons yang sesuai. Format JSON ini mudah dibaca dan digunakan oleh manusia maupun mesin, sehingga sangat ideal untuk chatbot yang membutuhkan akses cepat ke berbagai data untuk memberikan respons

interaktif dan real-time[12], [13]. JSON memungkinkan chatbot untuk dengan mudah memuat dan memproses data intensi[14], sehingga meningkatkan kemampuan bot dalam memberikan respons yang relevan dan tepat berdasarkan input pengguna.

D. Dataset

Dataset ini dirancang untuk melatih model Natural Language Processing (NLP), sehingga chatbot dapat mengenali berbagai intensi (intents) dan respons yang tepat sesuai input pengguna. Setiap entri dalam dataset mencakup pola kata yang mewakili cara berbeda pengguna mengajukan pertanyaan atau permintaan, serta respons yang sudah ditentukan sebelumnya untuk dijawab oleh chatbot[15]. Dataset ini memungkinkan model untuk belajar dari pola bahasa umum, sehingga dapat mengidentifikasi dan memahami berbagai variasi kalimat dari pengguna. Dengan demikian, dataset tersebut berkontribusi signifikan dalam meningkatkan akurasi chatbot dalam memahami maksud pengguna, memberikan respons yang tepat, dan memastikan interaksi yang lebih alami dan efisien.

```
model > {} dataset > {} intents > {} 0 > {} patterns
1
2
3
4   "intents": [
5     {
6       "tag": "greeting",
7       "patterns": [
8         "halo",
9         "hi",
10        "hello",
11        "selamat pagi",
12        "selamat siang",
13        "selamat sore",
14        "selamat malam"
15      ],
16      "responses": [
17        "Halo! Ada yang bisa saya bantu?",
18        "Hi! Bagaimana saya bisa membantu Anda hari ini?",
19        "Hello! Ada yang ingin ditanyakan?"
20      ],
21      "context": {
22        ""
23      }
24    },
25    {
26      "tag": "pendaftaran_siswa",
27      "patterns": [
28        "bagaimana cara mendaftar di SMP SWASTA AL WASHLIYAH 9 MEDAN?",
29        "prosedur pendaftaran siswa baru",
30        "cara mendaftar di sekolah ini",
31        "pendaftaran siswa baru",
32        "bagaimana cara masuk ke SMP SWASTA AL WASHLIYAH 9 MEDAN?"
33      ]
34    }
35  ]
```

Gambar 1 JSON Dataset Pendaftaran

Dataset berisi kumpulan intensi (intents) yang ditulis dalam format JSON, seperti contoh di atas. Setiap entri dalam dataset memiliki tag, pola (patterns), respons (responses), dan konteks (context).

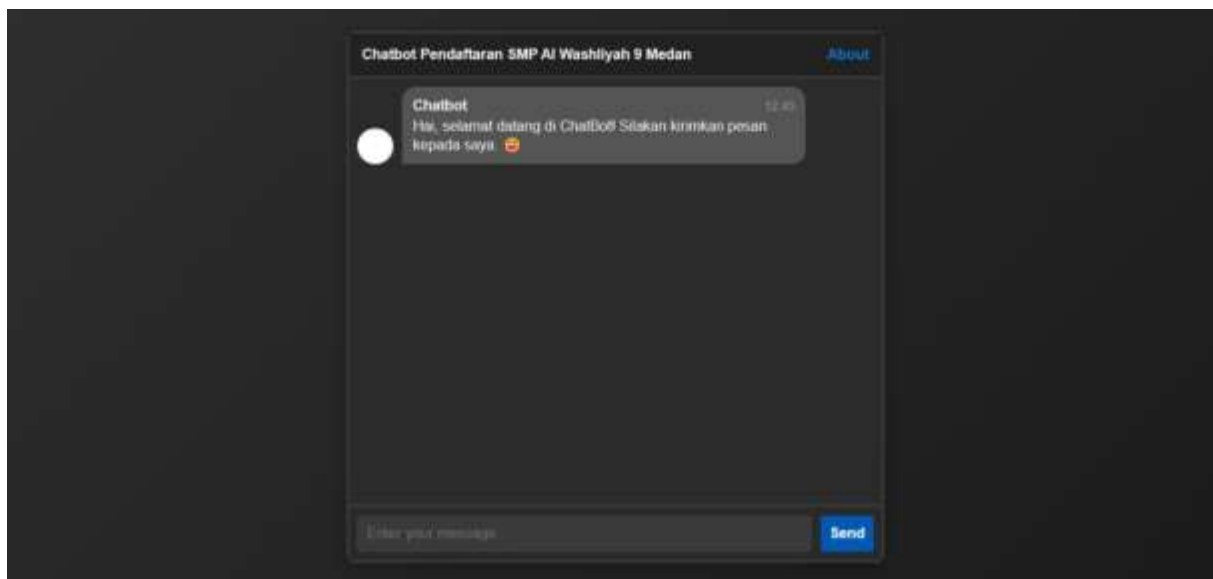
1. Tag: Setiap intent diberi nama tag unik, seperti "greeting," "pendaftaran_siswa," atau "dokumen_pendaftaran." Tag ini mengidentifikasi jenis pertanyaan atau pernyataan yang diajukan pengguna.
2. Patterns: Bagian ini berisi variasi kalimat atau frasa yang mungkin diucapkan pengguna. Sebagai contoh, pada intent "greeting," chatbot mengenali sapaan pengguna seperti "halo," "hi," dan "selamat pagi."
3. Responses: Setiap intent juga dilengkapi respons yang disiapkan chatbot untuk dijawab. Misalnya, ketika pengguna menanyakan dokumen yang diperlukan, chatbot akan memberikan daftar dokumen yang perlu dipersiapkan dalam bentuk kalimat yang informatif.
4. Context: Bagian ini dapat digunakan untuk menyimpan konteks percakapan tertentu agar chatbot dapat menyesuaikan respons sesuai konteks, walaupun dalam contoh ini masih kosong.

E. Flask

Flask adalah framework web minimalis untuk Python yang digunakan untuk membangun aplikasi web, terutama API dan aplikasi berbasis server[16]. Dikenal karena kesederhanaan dan fleksibilitasnya, Flask memungkinkan pengembang membuat aplikasi web tanpa perlu berurusan dengan banyak konfigurasi atau struktur yang kompleks. Flask digunakan untuk mengembangkan antarmuka chatbot berbasis web yang memungkinkan pengguna mengakses chatbot melalui browser[17]. Flask berperan sebagai server yang menerima permintaan dari klien (pengguna), memproses data (seperti input pengguna yang dikirim ke chatbot), dan kemudian mengirimkan respons kembali ke klien[18]. Dengan Flask, kita dapat dengan mudah mengatur routing atau alur navigasi aplikasi[19], membuat endpoint untuk menangani permintaan tertentu[20], dan menghubungkan berbagai komponen seperti model NLP, modul JSON, dan penyimpanan file yang dibutuhkan untuk fungsi chatbot[21]. Flask mendukung berbagai pustaka eksternal, termasuk nltk untuk pemrosesan bahasa[22], dan memungkinkan integrasi dengan basis data serta file statis[23]. Struktur yang sederhana dan modularitas Flask membuatnya menjadi pilihan populer bagi pengembang yang membutuhkan solusi cepat, namun tetap dapat dikembangkan menjadi sistem yang lebih kompleks sesuai kebutuhan.

F. HTML

HTML, atau HyperText Markup Language, merupakan standar dasar dalam pengembangan halaman web, yang memungkinkan penyusunan struktur dan tampilan informasi pada halaman web[24]. Dalam proyek ini, HTML digunakan untuk menyusun elemen-elemen tampilan antarmuka web, seperti halaman utama (index.html) dan halaman informasi (about.html). HTML berperan penting dalam memformat konten agar mudah dibaca dan digunakan oleh pengguna.



Gambar 2 Tata letak chatbot

Pada tampilan chatbot di atas, HTML berperan penting dalam menyusun respons chatbot secara terstruktur dan rapi untuk pengguna. Dengan HTML, chatbot dapat menampilkan berbagai elemen seperti pesan teks, formulir input, serta tautan untuk mengunduh data, sehingga memudahkan pengguna dalam berinteraksi. HTML juga memungkinkan pengaturan tata letak dan gaya tampilan, sehingga komunikasi antara chatbot dan pengguna menjadi lebih jelas, mudah dipahami, dan nyaman digunakan.

A. Pra-processing

Proses pembuatan chatbot dimulai dengan persiapan data dari *corpus* yang berisi berbagai pola dan respons. Pertama, pustaka-pustaka yang diperlukan, seperti nltk dan keras, diimpor untuk memproses

bahasa alami serta membangun model *neural network*. Untuk membersihkan data teks, objek *WordNetLemmatizer* dari pustaka NLTK digunakan untuk mengubah kata-kata menjadi bentuk dasar (*lemmatization*), yang membantu mengenali kata-kata berbeda dengan makna serupa. Dengan cara ini, input pengguna dapat lebih mudah dipahami oleh sistem meskipun terdapat variasi bentuk kata. Selanjutnya, data chatbot yang disimpan dalam file JSON diakses dan diproses. Untuk setiap pola kalimat di dalam data, sistem melakukan tokenisasi, yaitu memecah kalimat menjadi kata-kata individu, dan menyimpan kata-kata ini dalam variabel untuk membangun kosa kata chatbot. Setiap pasangan pola dan tag tertentu disimpan dalam struktur *documents*, yang akan menjadi dasar untuk melatih model *neural network*. Kelas respons juga diidentifikasi dan disimpan, agar model dapat mengenali dan mengklasifikasikan konteks kalimat dengan tepat berdasarkan kategori tertentu.

Proses berikutnya adalah menormalisasi kata-kata yang ada, dengan melewati kata-kata seperti tanda baca yang tidak relevan, dan melakukan *lemmatization* untuk mengurangi redundansi dalam kosa kata. Semua kata kemudian disortir dalam urutan abjad untuk menciptakan daftar unik yang lebih efisien. Demikian pula, daftar kelas diurutkan untuk memudahkan proses pelatihan. Sebagai persiapan untuk pelatihan model, objek Python ini disimpan dalam bentuk *pickle* untuk menghindari pengolahan data berulang kali. Ini memungkinkan chatbot untuk lebih cepat mengakses informasi kosa kata dan kelas saat prediksi dilakukan.

B. Pelatihan dan Modelling

Setelah menyelesaikan tahap pra-pemrosesan data, proses berikutnya adalah menyiapkan data pelatihan yang akan digunakan untuk melatih model jaringan saraf. Data pelatihan terdiri dari fitur dan label yang merepresentasikan pola kalimat dan kelas terkait masing-masing. Dalam konteks ini, fitur diperoleh melalui representasi *bag of words*, di mana setiap pola kalimat diubah menjadi array biner yang mencerminkan keberadaan atau ketidakhadiran kata-kata tertentu. Untuk menghasilkan *bag of words* ini, setiap pola kalimat yang terdiri dari token yang dihasilkan selama fase pra-pemrosesan akan dianalisis. Pertama, setiap kata dalam pola tersebut akan di-lemmatize menggunakan objek *WordNetLemmatizer*, yang mengubah kata menjadi bentuk dasarnya, sehingga variasi kata yang serupa dapat dikelompokkan bersama. Proses ini penting untuk mengurangi kompleksitas data dan memastikan bahwa kata-kata yang berhubungan satu sama lain, seperti "bermain" dan "bermainlah," diperlakukan sama.

Setelah melakukan *lemmatization*, kita akan membangun array *bag of words* untuk setiap pola kalimat. Dalam proses ini, sebuah loop akan memeriksa setiap kata unik dalam daftar *words*. Jika kata tersebut muncul dalam pola kalimat saat ini, kita menambahkan nilai 1 ke dalam array *bag*, sedangkan jika tidak, kita menambahkan nilai 0. Selanjutnya, untuk menghasilkan label yang sesuai, kita membuat array biner yang menunjukkan kelas yang terkait dengan pola kalimat. Dalam array ini, hanya satu nilai yang akan menjadi 1, sesuai dengan indeks kelas dari pola kalimat, sementara sisanya adalah 0. Ini menciptakan representasi yang jelas tentang mana kelas yang relevan untuk setiap pola kalimat.

Setelah seluruh dataset diolah menjadi format *bag of words* dan label biner, kita mengacak data pelatihan untuk menghilangkan bias yang mungkin timbul dari urutan data. Kemudian, fitur dan label dipisahkan dan dikonversi ke dalam format *numpy arrays*, yang memfasilitasi efisiensi dalam pengolahan dan pelatihan model. Dengan data pelatihan yang siap, langkah selanjutnya adalah membangun model jaringan saraf. Model ini menggunakan arsitektur bertingkat yang terdiri dari beberapa lapisan: lapisan input yang menerima vektor fitur dari *bag of words*, satu lapisan tersembunyi dengan 128 neuron dan fungsi aktivasi *ReLU* untuk menangkap hubungan non-linear dalam data, dan lapisan keluaran yang menggunakan fungsi aktivasi *Softmax* untuk menghasilkan probabilitas bagi setiap kelas, yang memungkinkan pemodelan klasifikasi multi-kelas. Untuk mengurangi risiko *overfitting*, teknik *dropout* diterapkan setelah lapisan tersembunyi, mengacak neuron yang aktif selama pelatihan.

Model kemudian dikompilasi menggunakan algoritma *Stochastic Gradient Descent* (SGD), yang dipilih karena kemampuannya untuk memberikan hasil yang baik dalam pelatihan model semacam ini. Pengaturan *learning rate*, *decay*, *momentum*, dan penggunaan *Nesterov accelerated gradient* dioptimalkan untuk mempercepat konvergensi dan meningkatkan akurasi. Model dilatih selama 200

epoch dengan ukuran *batch* 5, dan hasil pelatihan disimpan untuk digunakan di masa depan. Proses pelatihan ini menghasilkan model yang siap digunakan untuk memprediksi respons berdasarkan input pola kalimat yang diberikan, memanfaatkan pengetahuan yang diperoleh dari data pelatihan untuk memberikan jawaban yang relevan dan kontekstual. Dengan demikian, model siap untuk digunakan dalam aplikasi chatbot yang akan memberikan interaksi yang lebih alami dan responsif kepada pengguna.

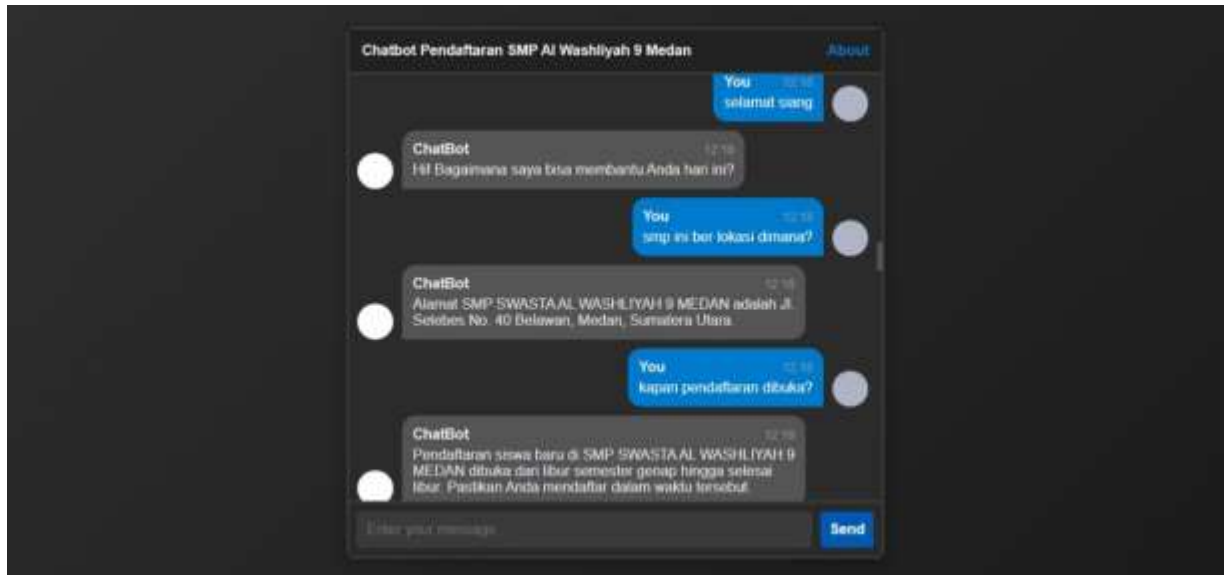
C. Graphical User Interface

Tahap ini adalah pembangunan antarmuka pengguna (User Interface) dengan HTML, CSS, dan JavaScript. Di tahap ini, antarmuka chatbot dirancang agar pengguna dapat berinteraksi langsung dengan chatbot melalui halaman web. Struktur HTML mendefinisikan elemen-elemen dasar halaman, seperti bagian utama percakapan, formulir pengiriman pesan, serta header yang menampilkan judul chatbot dan tautan *About*. Untuk desain visual, elemen-elemen ini dihubungkan dengan *stylesheet* eksternal melalui file CSS. JavaScript memainkan peran penting dalam menangani fungsionalitas percakapan interaktif dan fitur unggah data. Misalnya, ketika pengguna mengirim pesan melalui formulir input, fungsi *botResponse* dipanggil untuk mengirimkan teks ke chatbot dan menerima balasan yang ditampilkan di area percakapan. Jika chatbot meminta unggahan berkas, bagian unggahan akan muncul, dan berkas dapat dikirim ke server melalui *AJAX request*. Selain itu, jika pengguna mengetik `/unduh_data`, fungsi *handleUnduhData* akan mengaktifkan tautan untuk mengunduh data pengguna. Melalui kombinasi HTML, CSS, dan JavaScript, antarmuka ini memungkinkan pengalaman percakapan yang dinamis antara pengguna dan chatbot, yang mendukung interaksi percakapan berbasis teks, unggahan berkas, dan unduhan data.

3. HASIL DAN PEMBAHASAN

Pembuatan chatbot ini bertujuan untuk mempermudah proses pendaftaran di SMP Al Washliyah 9 Medan. Oleh karena itu, penulis merancang chatbot ini tidak hanya untuk memberikan informasi terkait pendaftaran, tetapi juga untuk memfasilitasi proses pendaftaran secara langsung. Dengan adanya chatbot ini, calon siswa dapat melakukan seluruh tahapan pendaftaran dengan mudah dan efisien, mulai dari pengisian data diri hingga pengunggahan berkas yang diperlukan.

Dalam ilustrasi di bawah ini, pengguna memulai interaksi dengan chatbot melalui sapaan. Setelah menyapa, pengguna melanjutkan dengan mengajukan pertanyaan terkait pendaftaran di SMP Al Washliyah 9 Medan. Chatbot merespons dengan ramah, memberikan informasi yang relevan dan jelas mengenai proses pendaftaran di sekolah tersebut, sehingga membantu pengguna mendapatkan pemahaman yang lebih baik tentang langkah-langkah pendaftaran yang diperlukan.



Gambar 3 Interaksi pengguna dan chatbot

Jika pengguna mengajukan pertanyaan di luar konteks pendaftaran SMP Alwashliyah 9 Medan dan tidak sesuai dengan informasi yang tercantum dalam dataset JSON, maka chatbot akan merespons dengan pesan: "Maaf, saya tidak mengerti pertanyaan Anda. Silakan hubungi administrasi sekolah untuk informasi lebih lanjut." Seperti yang ditampilkan pada gambar berikut.



Gambar 4 Respon chatbot ketika pertanyaan di luar konteks

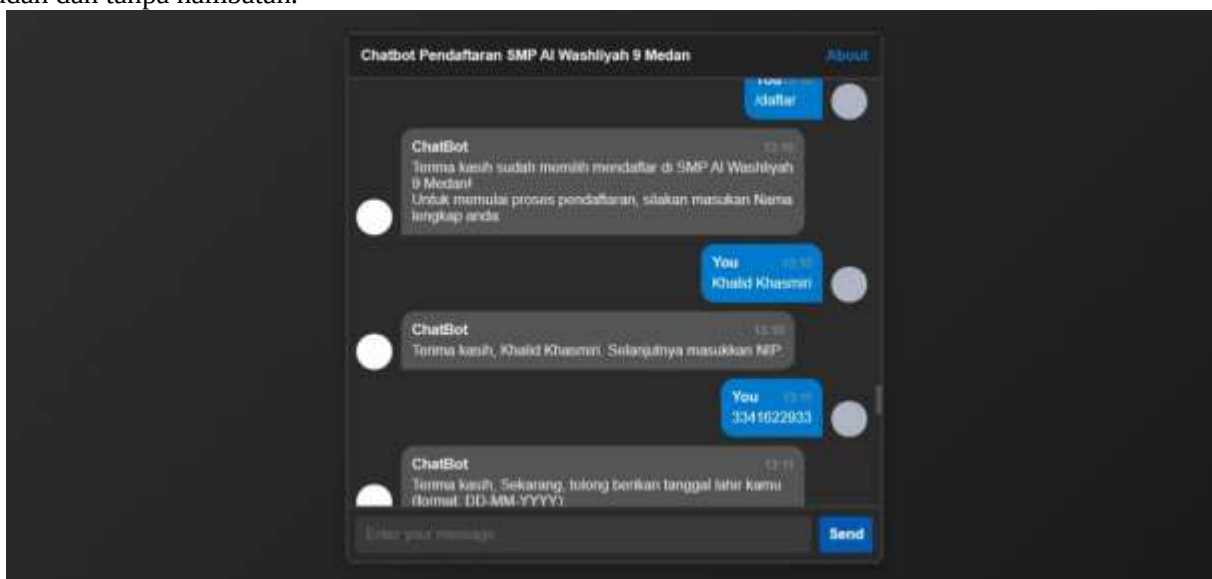
Jika pengguna atau calon siswa belum mengetahui data dan berkas yang dibutuhkan untuk melakukan pendaftaran, mereka dapat memasukkan perintah **/cara_daftar**. Setelah perintah ini dikirim, chatbot akan memberikan respons berupa daftar lengkap informasi dan dokumen yang diperlukan untuk pendaftaran. Informasi yang harus disiapkan meliputi nama lengkap, NIP (Nomor Induk Pribadi), tanggal lahir, alamat, dan nomor telepon. Sedangkan untuk berkas dokumen, pengguna perlu menyiapkan salinan akta kelahiran, kartu keluarga (KK), dan ijazah SD. Dengan memasukkan perintah ini, pengguna akan mendapatkan panduan yang jelas mengenai semua persyaratan

pendaftaran di SMP Al Washliyah 9 Medan, sehingga mereka dapat menyiapkan data dan dokumen secara lengkap dan sesuai sebelum memulai proses pendaftaran.



Gambar 5 Respon chatbot ketika ditanya cara mendaftar

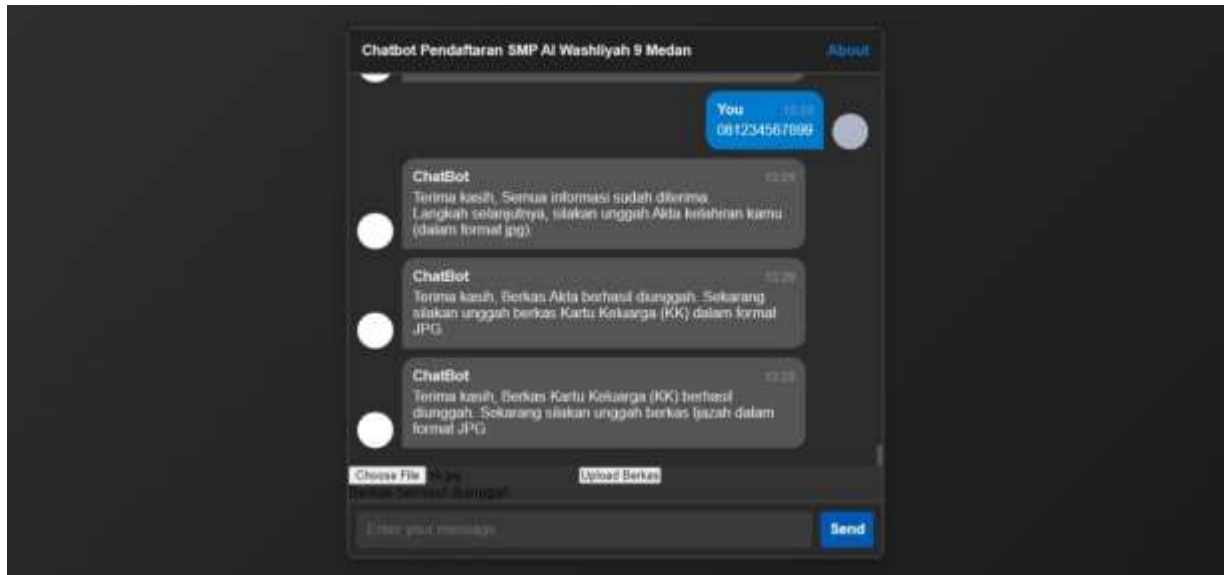
Untuk memulai proses pendaftaran, pengguna perlu memasukkan perintah **/daftar** pada chatbot. Setelah perintah ini dikirimkan, chatbot akan memandu pengguna dengan meminta beberapa informasi yang diperlukan sebagai persyaratan pendaftaran. Informasi yang dibutuhkan mencakup nama lengkap, NIP (Nomor Induk Pribadi), tanggal lahir, alamat lengkap, serta nomor telepon yang dapat dihubungi. Dengan mengirimkan data-data ini, pengguna akan memenuhi syarat administratif yang diperlukan untuk melanjutkan proses pendaftaran di SMP Al Washliyah 9 Medan. Chatbot memastikan bahwa informasi ini terverifikasi dengan baik, sehingga pengguna dapat menyelesaikan pendaftaran dengan mudah dan tanpa hambatan.



Gambar 6 Melakukan pendaftaran

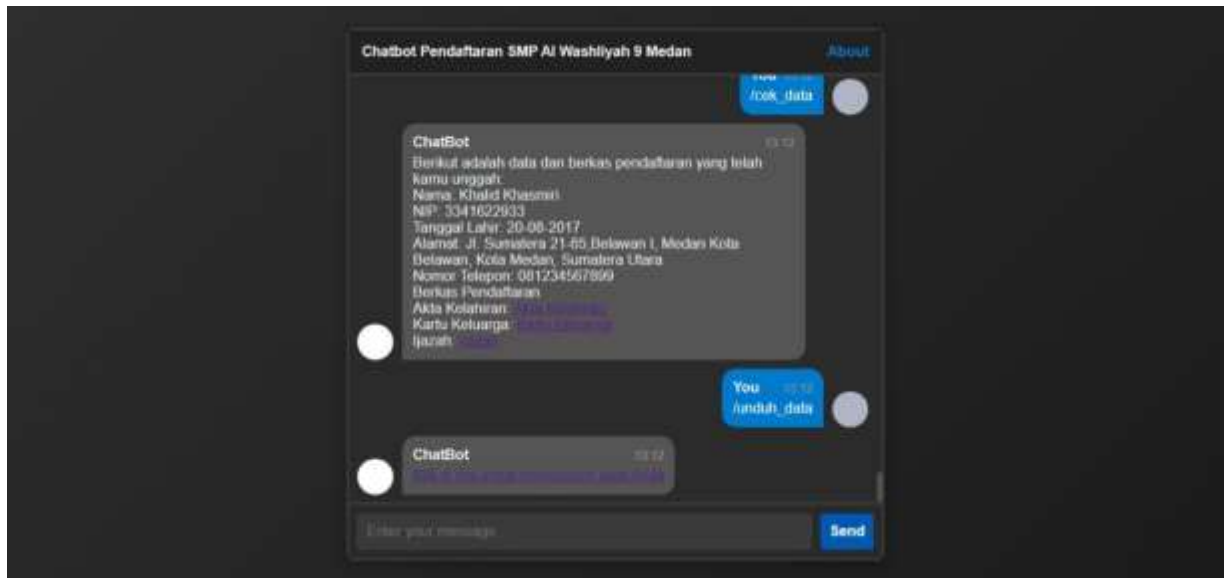
Setelah semua data formulir pendaftaran terisi, pengguna akan diminta untuk melengkapi proses pendaftaran dengan mengunggah beberapa berkas penting. Berkas yang harus diunggah meliputi salinan

akta kelahiran, kartu keluarga (KK), dan ijazah SD, semuanya dalam format JPG. Dokumen-dokumen ini merupakan persyaratan wajib untuk memverifikasi identitas dan kelayakan pendaftaran pengguna di SMP Al Washliyah 9 Medan. Dengan mengunggah berkas-berkas ini, pengguna memastikan bahwa semua data pendukung sudah lengkap, sehingga proses pendaftaran dapat berjalan lancar hingga tahap verifikasi akhir.



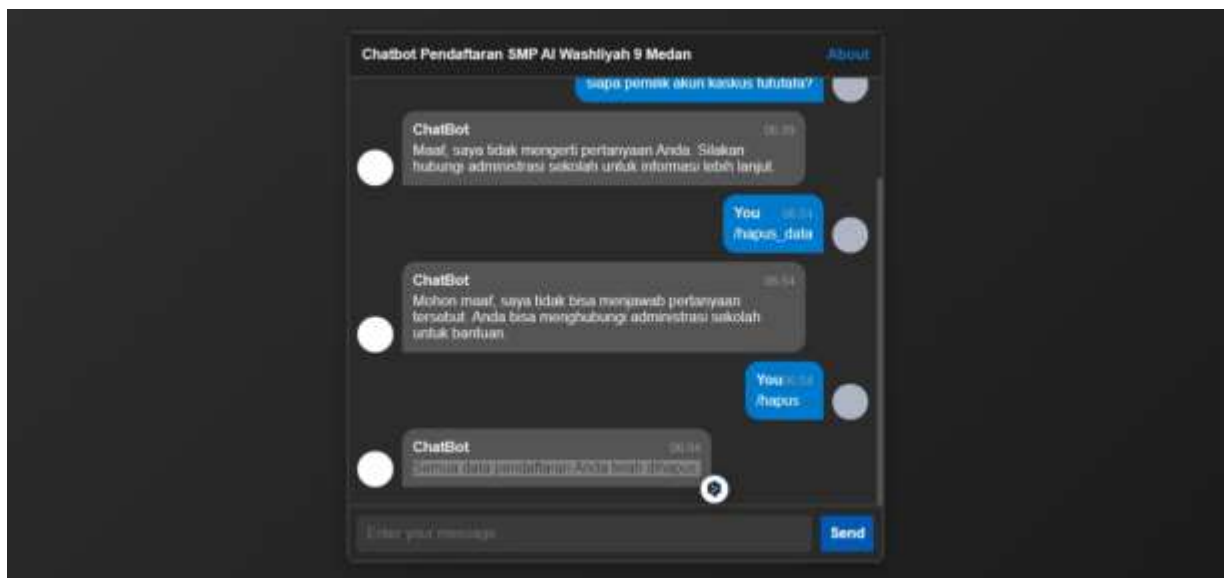
Gambar 7 Upload berkas pendaftaran

Agar pengguna dapat memastikan bahwa data yang telah dikirimkan benar, mereka dapat menggunakan perintah **/cek_data**. Dengan perintah ini, chatbot akan menampilkan kembali seluruh data yang telah diunggah oleh pengguna untuk diperiksa. Untuk memverifikasi berkas-berkas yang sudah dikirim, pengguna dapat meninjau setiap berkas dengan mengklik nama berkas yang diinginkan. Berkas tersebut akan ditampilkan dengan format nama **namaPendaftar_NIP_namaBerkas.jpg**, sehingga mudah dikenali dan diperiksa ulang. Langkah ini membantu pengguna memastikan bahwa semua data dan dokumen telah diunggah dengan benar sebelum proses pendaftaran berlanjut, mengurangi risiko kesalahan atau kekurangan informasi dalam pengajuan pendaftaran.



Gambar 8 Cek dan unduh data terdaftar

Apabila pengguna merasa bahwa data yang ditampilkan oleh chatbot tidak sesuai atau terdapat kesalahan dalam informasi pendaftaran yang tersimpan, pengguna dapat menggunakan perintah “/hapus” untuk menghapus data pendaftaran yang sudah tersimpan. Setelah data dihapus, pengguna dapat melakukan proses pendaftaran ulang dengan data yang benar. Hal ini memastikan bahwa setiap informasi pendaftaran yang tersimpan adalah akurat dan sesuai dengan kebutuhan pengguna.



Gambar 9 Hapus data pendaftaran

Pengguna juga memiliki opsi untuk menyimpan data pendaftaran dalam format PDF dengan menggunakan perintah **/unduh_data**. Setelah perintah ini dimasukkan, chatbot akan mengumpulkan seluruh informasi dan berkas yang telah dikirim, lalu menyusunnya dalam satu file PDF. Nama berkas PDF yang diunduh akan mengikuti format **namaPendaftar_NIP_namaBerkas.pdf**, yang memudahkan pengguna untuk menyimpan dan mengorganisir dokumen pendaftaran mereka untuk keperluan dokumentasi pribadi atau referensi di masa mendatang. Fitur ini memastikan bahwa pengguna dapat dengan mudah menyimpan salinan lengkap pendaftaran mereka dalam bentuk digital yang terstruktur, sehingga mempermudah akses dan pengelolaan informasi pendaftaran SMP Al Washliyah 9 Medan.



Gambar 10 Hasil unduhan data pendaftar dalam format pdf

4. KESIMPULAN

Kesimpulan dari penelitian ini menunjukkan bahwa pembuatan chatbot pendaftaran berbasis Natural Language Processing telah berhasil memenuhi tujuan untuk mempermudah proses pendaftaran siswa di SMP Al Washliyah 9 Medan. Chatbot yang dikembangkan mampu memberikan informasi pendaftaran, memandu calon siswa dalam melengkapi data, dan mengunggah berkas yang diperlukan secara efisien. Dengan adanya chatbot ini, diharapkan komunikasi antara calon siswa dan pihak sekolah menjadi lebih praktis dan responsif, sehingga proses pendaftaran dapat dilakukan dengan lebih cepat dan mengurangi beban administrasi pada pihak sekolah. Keberhasilan implementasi ini juga membuktikan bahwa teknologi chatbot berbasis NLP memiliki potensi besar untuk diintegrasikan dalam proses pendidikan dan manajemen sekolah.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada Bapak Lahmudin Sipahutar, selaku dosen pembimbing, yang telah memberikan arahan, bimbingan, serta waktu dan perhatian penuh selama penyusunan karya tulis ini. Berkat saran yang Bapak berikan, penulis dapat menyelesaikan penelitian ini dengan baik dan mencapai hasil yang memuaskan.

Ucapan terima kasih juga penulis sampaikan kepada Bapak Abdul Karim Nasution, selaku Kepala Sekolah di SMP Swasta Al Washliyah 9 Medan, yang telah memberikan dukungan penuh, izin, serta fasilitas yang sangat membantu dalam proses penelitian ini. Bantuan dan perhatian yang diberikan oleh pihak sekolah sangat berarti dalam mendukung kelancaran dan kesuksesan penelitian ini.

Selain itu, rasa terima kasih yang mendalam penulis tujuikan kepada teman-teman di lokasi riset yang sama, yang telah memberikan dukungan, berbagi pengalaman, serta saling membantu dalam menghadapi berbagai tantangan selama penelitian. Kebersamaan dan kerja sama yang terjalin dengan baik sangat berperan penting dalam menyelesaikan karya tulis ini.

Semoga segala bantuan dan dukungan yang telah diberikan mendapatkan balasan terbaik, dan karya tulis ini dapat memberikan manfaat bagi perkembangan ilmu dan penelitian di masa mendatang.

REFERENSI

- [1] G. Battineni, G. G. Sagaro, N. Chinatalapudi, and F. Amenta, "Applications of Machine Learning Predictive Models in the Chronic Disease Diagnosis," *J Pers Med*, vol. 10, no. 2, p. 21, 2020, doi: 10.3390/jpm10020021.
- [2] M. Chung, E. Ko, H. Joung, and S. J. Kim, "Chatbot E-Service and Customer Satisfaction Regarding Luxury Brands," *J Bus Res*, vol. 117, pp. 587–595, 2020, doi: 10.1016/j.jbusres.2020.06.001.
- [3] T. A. Zuraiyah, D. K. Utami, and D. Herlambang, "Implementasi Chatbot Pada Pendaftaran Mahasiswa Baru Menggunakan Recurrent Neural Network," *Jurnal Ilmiah Teknologi dan Rekayasa*, vol. 24, no. 2, pp. 91–101, 2020.
- [4] Peter Koblyakov, "AI Chatbot For Higher Education: Top, Benefits, Use Cases. Raccoon Gang," RACCOON GANG.
- [5] D. Ramandanis and S. Xinogalos, "Investigating the Support Provided by Chatbots to Educational Institutions and Their Students: A Systematic Literature Review," *Multimodal Technologies and Interaction*, vol. 7, no. 11, p. 103, 2023, doi: 10.3390/mti7110103.
- [6] F. Millstein, *Natural Language Processing with Python: Natural Language Processing Using NLTK*. Frank Millstein, 2020.
- [7] J. Perkins, *Python 3 Text Processing with NLTK 3 Cookbook*. Packt Publishing Ltd, 2014.
- [8] D. Yogish, T. N. Manjunath, and R. S. Hegadi, "Review on Natural Language Processing Trends and Techniques Using NLTK," in *Recent Trends in Image Processing and Pattern Recognition: Second International Conference, RTIP2R 2018, Solapur, India, December 21–22, 2018, Revised Selected Papers, Part III*, Springer, 2019, pp. 589–606.
- [9] N. Hardeniya, J. Perkins, D. Chopra, N. Joshi, and I. Mathur, *Natural Language Processing: Python and NLTK*. Packt Publishing Ltd, 2016.
- [10] N. Schäfer and S. Michel, "JODA: A vertically scalable, lightweight JSON processor for big data transformations," in *2020 IEEE 36th International Conference on Data Engineering (ICDE)*, 2020, pp. 1726–1729.
- [11] J. M. S. Heldal and K. Hermansen, "Chatbot Generation for Open Data Accessibility," The University of Bergen, 2023.
- [12] P. Bourhis, J. L. Reutter, F. Suárez, and D. Vrgoč, "JSON: Data Model, Query Languages and Schema Specification," in *Proceedings of the 36th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems*, 2017, pp. 123–135.
- [13] R. Kar and R. Haldar, "Applying Chatbots to the Internet of Things: Opportunities and Architectural Elements," *arXiv preprint arXiv:1611.03799*, 2016.
- [14] Y. Li, N. R. Katsipoulakis, B. Chandramouli, J. Goldstein, and D. Kossmann, "Mison: A Fast JSON Parser for Data Analytics," *Proceedings of the VLDB Endowment*, vol. 10, no. 10, pp. 1118–1129, 2017.

-
- [15] M. Aleedy, H. Shaiba, and M. Bezbradica, "Generating and Analyzing Chatbot Responses Using Natural Language Processing," *International Journal of Advanced Computer Science and Applications*, vol. 10, no. 9, 2019.
 - [16] O. Adedeji, *Full-Stack Flask and React: Learn, code, and deploy powerful web applications with Flask 2 and React 18*. Packt Publishing Ltd, 2023.
 - [17] S. V Prajwal, G. Mamatha, P. Ravi, D. Manoj, and S. K. Joisa, "Universal Semantic Web Assistant Based on Sequence to Sequence Model and Natural Language Understanding," in *2019 9th International Conference on Advances in Computing and Communication (ICACC)*, IEEE, 2019, pp. 110–115.
 - [18] M. Grinberg, *Flask Web Development*. O'Reilly Media, Inc., 2018.
 - [19] D. Gaspar and J. Stouffer, *Mastering Flask Web Development: Build Enterprise-grade, Scalable Python Web Applications*. Packt Publishing Ltd, 2018.
 - [20] N. Loubser and N. Loubser, "Creating a RESTful API: Flask," *Software Engineering for Absolute Beginners: Your Guide to Creating Software Products*, pp. 193–233, 2021.
 - [21] K. Relan and K. Relan, "Beginning with Flask," *Building REST APIs with Flask: Create Python Web Services with MySQL*, pp. 1–26, 2019.
 - [22] A. L. S. Saabith, T. Vinothraj, and M. Fareez, "Popular Python Libraries and Their Application Domains," *International Journal of Advance Engineering and Research Development*, vol. 7, no. 11, 2020.
 - [23] S. Aggarwal, *Flask Framework Cookbook: Enhance Your Flask Skills with Advanced Techniques and Build Dynamic, Responsive Web Applications*. Packt Publishing Ltd, 2023.
 - [24] R. Nixon, *Learning PHP, MySQL, JavaScript, CSS & HTML5: A step-by-step guide to creating dynamic websites*. "O'Reilly Media, Inc.," 2014.