

IMPLEMENTASI NATURAL LANGUAGE PROCESSING DALAM PERANCANGAN APLIKASI CHATBOT PADA FIKTI UMSU

Muhammad Izzuddin Mubarak, Mahardika Abdi

Teknologi Informasi, Universitas Muhammadiyah Sumatera Utara
Jalan Kapt. Mukhtar Basri No.3 Medan, Indonesia
mubarokizzuddin788@gmail.com

ABSTRAK

Perkembangan teknologi informasi dan komunikasi, khususnya kecerdasan buatan, telah membawa dampak signifikan dalam berbagai bidang, termasuk pendidikan. Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Muhammadiyah Sumatera Utara (FIKTI UMSU) menghadapi tantangan dalam memberikan layanan akademik dan administrasi yang cepat dan akurat kepada mahasiswa dan staf. Terbatasnya waktu operasional dan jumlah staf menyebabkan layanan informasi tidak selalu dapat diakses secara optimal. Untuk mengatasi masalah ini, penelitian ini bertujuan mengembangkan *chatbot* berbasis *Natural Language Processing* (NLP) yang dapat memberikan layanan informasi secara *real-time* dan dapat diakses kapan saja. *Chatbot* ini diharapkan dapat meningkatkan efisiensi layanan akademik dengan menjawab pertanyaan umum secara otomatis, sehingga mengurangi beban kerja staf dan memperluas akses informasi dengan layanan yang tersedia 24/7. Implementasi yang tepat, termasuk pemilihan teknologi, pengumpulan data percakapan, dan pelatihan model NLP yang akurat, menjadi kunci keberhasilan penerapan *chatbot* ini di FIKTI UMSU. Hasil pengujian *Black Box* menunjukkan performa *chatbot* mampu merespon 8 pertanyaan yang diajukan secara acak dengan benar dan mendapatkan presentase sebesar 97% yang dimana ini sudah cukup baik untuk model *chatbot* tersebut.

Kata Kunci: *Chatbot, Natural Language Processing, Layanan Akademik, FIKTI UMSU, Akses Informasi, Uji Black Box.*

1. PENDAHULUAN

Perkembangan teknologi informasi dan komunikasi telah membawa perubahan besar dalam berbagai aspek kehidupan manusia, termasuk dalam dunia pendidikan. Salah satu perkembangan signifikan adalah munculnya teknologi kecerdasan buatan seperti *chatbot* [2].

Penggunaan *chatbot* telah menjadi cara yang efektif untuk meningkatkan pengalaman pengguna dan memberikan solusi instan dalam berbagai bidang [9].

Seiring dengan meningkatnya jumlah mahasiswa dan kebutuhan akan layanan yang cepat dan akurat, FIKTI UMSU menghadapi tantangan dalam memberikan layanan yang optimal kepada seluruh mahasiswa dan staf. Layanan informasi yang sering kali membutuhkan respon cepat seperti jadwal kuliah, informasi beasiswa, prosedur administrasi, dan layanan akademik lainnya, sering kali tidak dapat diakses dengan mudah karena terbatasnya waktu operasional dan jumlah staf yang melayani dapat mengakibatkan penumpukan pekerjaan pada staf administrasi [12].

Layanan ini belum dapat dilakukan 24 jam karena keterbatasan waktu kerja. Sehingga pertanyaan yang disampaikan harus menunggu sampai petugas sudah siap untuk menjawab pertanyaan [11].

Kondisi ini memerlukan solusi yang inovatif untuk memastikan layanan yang diberikan tetap optimal tanpa menambah beban kerja berlebihan bagi staf. Untuk mengatasi tantangan tersebut, diperlukan solusi inovatif yang mampu memberikan layanan informasi secara *real-time* dan dapat diakses kapan saja. Teknologi *chatbot* berbasis *Natural Language*

Processing (NLP) merupakan salah satu solusi yang dapat menjawab kebutuhan ini [3].

Chatbot adalah program komputer yang dirancang untuk berinteraksi dengan manusia melalui percakapan, baik melalui teks atau suara [5].

Chatbot yang lebih canggih menggunakan teknologi NLP, sehingga mampu memahami konteks dan memberikan respon yang relevan [1].

Dalam konteks pendidikan, khususnya di Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Muhammadiyah Sumatera Utara (FIKTI UMSU), penerapan teknologi ini dapat memberikan kontribusi signifikan dalam meningkatkan kualitas layanan akademik dan administrasi. Implementasi *chatbot* berbasis NLP di FIKTI UMSU diharapkan dapat memberikan berbagai manfaat strategis. Dengan adanya *chatbot*, diharapkan dapat meningkatkan efisiensi layanan, karena *chatbot* dapat memberikan jawaban atas pertanyaan umum secara otomatis, sehingga mengurangi beban kerja staf administrasi dan memungkinkan mereka untuk fokus pada tugas-tugas yang lebih kompleks dan mendesak [2].

Selain itu, *chatbot* juga memperluas akses informasi dengan layanan yang tersedia 24/7 [12], sehingga mahasiswa dan dosen dapat mengakses informasi kapan saja tanpa terbatas oleh jam kerja staf. Namun, untuk merealisasikan manfaat-manfaat tersebut, diperlukan perancangan dan implementasi yang tepat, mencakup pemilihan teknologi yang sesuai, pengumpulan data percakapan yang relevan, serta pelatihan model NLP yang akurat [7].

Dalam penelitian ini, penulis bertujuan untuk mengembangkan *chatbot* berbasis NLP yang dapat

diimplementasikan di FIKTI UMSU, dengan fokus pada kemampuan *chatbot* untuk memahami dan menjawab berbagai pertanyaan akademik dan administratif.

2. TINJAUAN PUSTAKA

2.1. Penelitian Terdahulu

[7], mereka menggunakan metode *Natural Language Processing* (NLP) untuk memproses data FAQ tentang produk Jenius dari BTPN. Data FAQ tersebut diproses menggunakan NLP untuk mempersiapkan sistem agar dapat merespons pertanyaan pengguna dengan lebih baik. Proses pengolahan data meliputi tokenisasi, pengecekan kata slang, dan pemeriksaan morfologi. Dengan penerapan NLP, *chatbot* dapat memberikan informasi perbankan dengan tingkat kesesuaian jawaban sebesar 84% pada pengujian dengan 10 orang tester.

[5], mengimplementasi metode *Natural Language Processing* (NLP) dalam pembuatan *chatbot* untuk *customer service* di website Theme62.com. Sistem *chatbot* dibangun menggunakan Node.js dan library node-nlp, dengan proses *cleaning*, *normalizing*, *tokenizing*, *spell check*, *stemming*, dan *neural network*. Pengujian dilakukan dengan metode *white-box* untuk memastikan fungsionalitas sistem sesuai dengan yang diharapkan. Hasil pengujian menggunakan metode *White-box* pada aplikasi *chatbot* Theme62, diperoleh hasil sesuai dengan program yang diharapkan pada setiap pengujian. Hal ini menunjukkan bahwa metode NLP dengan pipelines yang tepat mampu menjawab pertanyaan berdasarkan basis pengetahuan yang telah diberikan sebelumnya.

[13], mengimplementasikan NLP dalam pembuatan *chatbot customer service* untuk publisher jurnal Larisma. *Chatbot* ini menggunakan

data training dari pertanyaan yang sering diajukan oleh peneliti untuk menghasilkan informasi yang akurat. Data tersebut kemudian diproses melalui metode analisis data yang melibatkan *preprocessing*, pelatihan, dan pembangunan model. Uji akurasi *chatbot* dilakukan oleh perancang melalui terminal di Google Colab dengan menilai kesesuaian antara pertanyaan dan jawaban yang dihasilkan. Hasil uji *chatbot* ini menunjukkan bahwa *chatbot* mampu memberikan respons yang efektif dengan akurasi 100%.

[2] mereka mengimplementasikan NLP dalam pengembangan *chatbot* untuk SMK YPE Nusantara Slawi memungkinkan *chatbot* memahami teks, mengenali pola bahasa, dan memberikan respons yang relevan dan kontekstual. Data pelatihan diambil dari file JSON berisi *intents* dan *patterns*, yang membantu *chatbot* merangkum informasi kompleks menjadi lebih ringkas dan mudah dipahami, serta meningkatkan interaksi dan pengalaman pengguna.

2.2. Chatbot

Chatbot adalah teknologi yang memungkinkan orang dan komputer berinteraksi melalui bahasa alami dan memberikan respons yang cepat berdasarkan apa yang dimaksudkan pengguna. Meskipun awalnya digunakan untuk meniru percakapan manusia dan menghibur pengguna, aplikasinya telah berkembang ke berbagai bidang seperti pendidikan, pencarian informasi, bisnis, dan *e-commerce*. *Chatbot* adalah salah satu contoh dari *Artificial Intelligence* (AI).

Berdasarkan penelitian yang dilakukan [9] mereka menunjukkan bahwa penggunaan *chatbot* berbasis NLP di Jurusan Teknik Informatika dan Komputer membantu mengoptimalkan penerimaan informasi. Integrasi *chatbot* ke dalam *website* memudahkan mahasiswa mengakses panduan yang akurat dan responsif, serta meningkatkan interaksi teknologi. Meskipun responnya bervariasi, mayoritas responden merasa puas dengan *chatbot* dan melihatnya sebagai alat yang meningkatkan kualitas layanan akademik.

Chatbot juga dapat meningkatkan pelayanan efektivitas pelayanan *customer service*, memberikan dukungan pelanggan yang konsisten, meningkatkan kepuasan konsumen, dan mengurangi biaya bagi perusahaan.

Berdasarkan penelitian yang dilakukan [6] menunjukkan bahwa penggunaan *chatbot* dalam *customer service* meningkatkan efisiensi, memberikan dukungan 24 jam yang konsisten, meningkatkan kepuasan konsumen, dan mengurangi biaya. *Chatbot* memberikan jawaban yang akurat tanpa dipengaruhi oleh emosi atau kelelahan, menjadikannya solusi efektif untuk menggantikan *customer service* manusia. Hasil penelitian menunjukkan bahwa *chatbot* dapat secara signifikan meningkatkan kinerja layanan pelanggan.

Teknologi *chatbot* membantu sektor pariwisata, terutama perhotelan, dengan memberikan layanan 24 jam tanpa tenaga manusia. [1] merancang *chatbot* hotel menggunakan *Natural Language Processing* (NLP) untuk memudahkan wisatawan mendapatkan informasi. NLP memungkinkan *chatbot* memahami dan merespons bahasa alami, mengenali pola dan konteks, serta memberikan respon yang akurat dan relevan, meskipun terdapat kesalahan dalam input pengguna.

2.3. Natural Language Processing

Natural Language Processing (NLP) adalah bidang kecerdasan buatan yang berfokus pada interaksi antara komputer dan bahasa manusia. Tujuannya adalah membuat mesin mampu membaca, menalar dengan bahasa manusia, dan memprosesnya secara otomatis. [4] Penelitian ini mengimplementasikan NLP dengan algoritma Cosine Similarity untuk menilai otomatis jawaban esai. Data terdiri dari 5 jawaban mahasiswa yang diolah melalui tokenisasi, lower casing, penghapusan stopwords, stemming, dan lemmatization. Hasilnya menunjukkan

tingkat kemiripan jawaban mahasiswa menggunakan metode ini.

Hiburan memiliki peran penting dalam kehidupan sehari – hari, salah satu hiburan yang sangat populer di era digital saat ini salah satunya ialah music. Music dapat membantu kita membangkitkan semangat beraktifitas atau menghilangkan penat dari lelahnya melakukan aktivitas pekerjaan.

Dalam penelitian yang dilakukan oleh [8], Penelitian ini menganalisis sentimen ulasan Spotify menggunakan metode Naive Bayes dan Logistic Regression berbasis NLP. Dataset dari Kaggle diolah melalui beberapa tahap prapemrosesan seperti ekspansi kontraksi, pengecilan huruf, penghapusan stopwords, dan lemmatisasi. Hasilnya menunjukkan bahwa Logistic Regression memiliki akurasi 79%, sedikit lebih tinggi dibandingkan Naive Bayes dengan akurasi 76%.

Pada era digital saat ini informasi merupakan salah satu kunci penting dalam aktifitas kehidupan sehari – hari, oleh sebab itu diperlukannya penyampaian informasi yang berguna bagi setiap orang.

Penelitian yang dilakukan [6], Penelitian ini menganalisis dan merancang sistem informasi menggunakan *Natural Language Processing* (NLP) untuk mahasiswa di Universitas Bhayangkara Jakarta Raya. NLP diterapkan pada chatbot untuk meningkatkan efektivitas penyampaian informasi akademik secara interaktif. Proses NLP meliputi lima langkah: *Lexical Analysis*, *Syntax Analysis*, *Semantic Analysis*, *Discourse Integration*, dan *Pragmatics Analysis*. Sistem diuji dengan *black box testing* untuk memastikan keakuratan dan kinerjanya. Hasilnya, chatbot dapat membantu staf fakultas menjawab pertanyaan mahasiswa dengan lebih cepat dan akurat.

2.4. AI Project Cycle

Dalam era digital yang semakin maju, kecerdasan buatan (AI) menjadi salah satu teknologi transformatif yang mengubah cara bisnis dan industri bekerja. Meskipun AI memiliki banyak potensi untuk meningkatkan efisiensi, mengotomatisasi proses, dan menghasilkan wawasan bermanfaat dari data yang kompleks, penting bagi organisasi untuk mengikuti pendekatan yang sistematis dalam mengembangkan dan menerapkan proyek AI. Ini adalah alasan mengapa siklus proyek AI (*AI Project Cycle*) penting.

[10] Sistem pendeteksi Covid-19 dikembangkan menggunakan siklus *AI Project Cycle* untuk mengklasifikasikan pasien terpapar Covid-19. Tahapan dalam siklus ini meliputi *Problem Scoping*, *Data Acquisition*, *Data Exploration*, *Modelling*, dan *Deployment*. Masalah klasifikasi risiko terpapar Covid-19 berdasarkan gejala diidentifikasi dan dipetakan. Data diperoleh dari Kaggle, dan *Data Exploration* dilakukan untuk memahami karakteristik dataset. Dalam tahap *Modelling*, tiga model digunakan: *Logistic Regression*, *Random Forest*

Classifier, dan *Support Vector Machine* untuk klasifikasi. Tahap *Deployment* melibatkan pembuatan website yang mudah diakses, menggunakan HTML, JS, CSS, dan Flask, serta *hosting* di Heroku. Sistem ini memiliki akurasi lebih dari 80% dengan precision dan recall yang tinggi, dan diimplementasikan secara online untuk publik.

3. METODE PENELITIAN

Pada bagian ini memuat metode yang digunakan pada pembuatan skripsi.

3.1. Pengumpulan Data

Pada penelitian ini pengumpulan data merupakan langkah yang bertujuan untuk mendapatkan informasi yang akurat dan relevan guna merancang dan mengembangkan aplikasi chatbot yang efektif. Untuk perancangan aplikasi chatbot pada FIKTI UMSU data akan diperoleh melalui dua metode yaitu wawancara dan studi literatur.

3.2. Wawancara

Metode ini melibatkan interaksi langsung antara penulis dan responden, seperti staf administrasi di FIKTI UMSU. Tujuannya adalah untuk memahami kebutuhan pengguna, mengidentifikasi masalah dalam layanan akademik dan administratif, serta mengumpulkan umpan balik untuk perbaikan prototipe chatbot. Wawancara membantu penulis menggali informasi tentang pertanyaan yang sering diajukan, kesulitan pengguna, dan preferensi interaksi dengan chatbot.

3.3. Studi Literatur

Metode ini melibatkan penelaahan sumber tertulis yang relevan, seperti buku, jurnal ilmiah, dan laporan penelitian. Tujuan studi literatur adalah untuk mengetahui perkembangan terbaru dalam teknologi chatbot dan NLP, memahami konsep dasar, serta mempelajari penerapan chatbot di institusi lain.

Kedua metode ini saling melengkapi untuk memberikan pemahaman menyeluruh tentang kebutuhan, masalah, dan solusi dalam merancang aplikasi chatbot di FIKTI UMSU, sehingga diharapkan chatbot yang dikembangkan dapat memberikan layanan yang optimal dan memenuhi harapan pengguna.

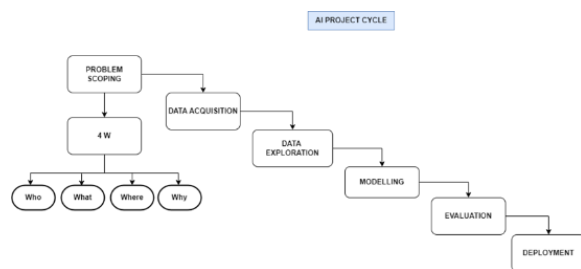
3.4. Tempat Penelitian

Penelitian ini akan dilakukan di Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Muhammadiyah Sumatera Utara (FIKTI UMSU). Kampus ini dipilih sebagai lokasi penelitian karena merupakan tempat di mana peneliti menempuh pendidikan dan memiliki akses langsung terhadap lingkungan akademik dan administratif yang akan menjadi objek studi.

3.5. Metodologi Pengembangan Sistem

Metodologi pengembangan sistem dengan menggunakan AI Project Cycle adalah pendekatan sistematis yang diterapkan untuk merancang, mengembangkan, dan mengimplementasikan sistem berbasis kecerdasan buatan [10].

AI Project Cycle terdiri dari beberapa tahap yang saling terkait, yang masing-masing memiliki peran penting dalam memastikan keberhasilan proyek AI. Berikut adalah tahapan-tahapan utama dalam AI Project Cycle:



Gambar 1. Flowchart AI Project Cycle

3.6. Problem Scoping

Tahap ini penulis mengidentifikasi secara mendalam tentang masalah atau kebutuhan spesifik yang ingin dipecahkan sehingga tujuan atau target menjadi semakin jelas dan lebih terarah serta akan lebih mudah untuk menemukan solusi [10], dengan menerapkan metode 4W (Who, What, Where, Why). Who: Siapa saja yang terlibat dalam masalah tersebut, What: Apa masalah dan faktor pendukung masalah, Where: Kondisi, Suasana atau tempat masalah yang diamati, Why: Alasan mengapa masalah tersebut perlu diselesaikan dan apa manfaatnya.

3.7. Data Acquisition

Proses *Data Acquisition* adalah langkah awal yang penting dalam pembuatan chatbot, di mana penulis mengumpulkan data untuk melatih dan mengembangkan model. Data yang dikumpulkan terkait dengan layanan akademik dan administratif, yang diperoleh melalui wawancara dengan staf administrasi di FIKTI UMSU. Semua pengumpulan data dilakukan secara manual. Setelah data terkumpul, langkah selanjutnya adalah membuat dataset dalam format file JSON yang berisi pertanyaan dan respon untuk pelatihan model.

3.8. Data Exploration

Dalam tahap eksplorasi data, penulis melakukan beberapa langkah untuk mempersiapkan data sebelum melatih model chatbot:

- Mengubah dataset JSON menjadi DataFrame: Konversi ini memudahkan penggunaan library pembelajaran mesin seperti scikit-learn dan TensorFlow.
- Menghilangkan Tanda Baca: Menghapus tanda baca dan mengubah teks menjadi huruf kecil untuk membersihkan data sebelum analisis lebih lanjut.

- Lematisasi: Mengurangi kata-kata ke bentuk dasarnya, meningkatkan akurasi analisis dengan mengurangi variasi kata yang memiliki arti sama.
- Menyortir Data Kelas Tags: Menyiapkan daftar tag untuk klasifikasi dan menghindari duplikasi yang dapat mengganggu analisis.
- Menghitung Jumlah Data Teks: Mengetahui skala dataset yang akan digunakan.
- Tokenisasi Data: Memisahkan teks menjadi unit-unit kecil (token) untuk memudahkan analisis.
- Melakukan Padding: Menambahkan nilai tertentu (biasanya 0) untuk mencapai panjang data yang seragam di semua sampel.
- Encoding Label atau Tag: Mengubah teks menjadi format numerik agar dapat diproses oleh komputer.
- Input Length dan Output Length: Menentukan jumlah unit yang digunakan dalam satu sampel data setelah tokenisasi untuk input pada algoritma LSTM.
- Menghitung Vocabulary Size: Mengidentifikasi setiap kata unik dalam pertanyaan dan jawaban untuk melatih chatbot.
- Menyimpan Model Word & Classes: Menyimpan teks yang sudah diproses untuk mempertahankan informasi model yang telah dilatih, memungkinkan penggunaan kembali model tanpa pelatihan ulang.

3.9. Modelling

Pembuatan model sangat diperlukan untuk melakukan pelatihan data. Penulis menggunakan sebuah model yang menggunakan algoritma Deep Learning dengan bahasa pemrograman Python. Proses selanjutnya dilakukan tahap pengembangan model dengan menggunakan salah satu jenis arsitektur saraf tiruan (Neural Network) yaitu LSTM (Long Short-Term Memory)

```

def msbot_bot(trainx, trainy, neuron, batch_size, epochs):
    # Input Layer
    i = tf.keras.Input(shape=(input_shape,))
    x = tf.keras.layers.Embedding(vocabulary+1,10)(i)
    # Hidden Layer
    x = tf.keras.layers.LSTM(neuron, return_sequences = True)(x)
    x = tf.keras.layers.Dropout(0.8)(x)
    # Hidden Layer
    x = tf.keras.layers.LSTM(neuron, return_sequences = True)(x)
    x = tf.keras.layers.Dropout(0.8)(x)
    # Flatten Layer
    x = tf.keras.layers.Flatten()(x)
    # Output Layer
    x = tf.keras.layers.Dense(output_length, activation="softmax")(x)
    model = tf.keras.models.Model(i,x)
    # Compile model
    model.compile(loss='sparse_categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
    # Menampilkan Parameter Model
    print(model.summary())
    history = model.fit(trainx,trainy,batch_size=batch_size,epochs=epochs,verbose=1,shuffle=False)
    return model, history
  
```

Gambar 2. Arsitektur Neural Network

Tujuan dari penggunaan arsitektur ini agar chatbot nantinya dapat memahami bahasa, konteks percakapan, dan memberikan respon yang relevan.

3.10. Evaluating

Langkah evaluasi penting untuk menilai kualitas dan kemampuan model, serta mengidentifikasi perbaikan yang perlu dilakukan. Evaluasi performa model dapat dilakukan dengan uji Black Box untuk

menilai sejauh mana model memahami konteks pertanyaan. Penulis mengajukan 20 pertanyaan secara acak kepada chatbot. Jika chatbot menjawab semua pertanyaan dengan benar, relevansi setiap jawaban adalah 1 (relevan). Nilai ini digunakan untuk menghitung Mean Average Precision (MAP), metrik evaluasi yang mengukur seberapa baik chatbot merespons pertanyaan pengguna. Dengan semua jawaban dianggap relevan, mAP dapat dihitung menggunakan rumus tertentu.

$$\text{MAP} = \frac{\text{Total AP}}{\text{Total Pertanyaan}} \times 100\% \quad (1)$$

Jika dalam kasus ini semua jawaban relevan (diasumsikan benar), maka Average Precision (AP) untuk setiap pertanyaan adalah sebagai berikut:

$$\text{AP} = \frac{\text{Total jawaban relevan yang ditemukan}}{\text{Total jawaban yang diambil}} \quad (2)$$

Karena *chatbot* mampu menjawab semua pertanyaan dengan benar, AP untuk setiap pertanyaan akan menjadi 1. Sehingga, untuk 20 pertanyaan dengan asumsi semua jawaban relevan, total AP akan menjadi $20 \times 1 = 20$. Akhirnya, nilai Mean Average Precision (MAP) akan menjadi:

$$\text{MAP} = \frac{20}{20} \times 100\% = 100\% \quad (3)$$

3.11. Deployment

Deployment adalah tahap mengintegrasikan *chatbot* ke dalam lingkungan yang sesungguhnya. Dalam proyek ini, penulis memutuskan untuk menggunakan Flask sebagai kerangka kerja web dan melakukan deployment secara lokal melalui localhost.

4. HASIL DAN PEMBAHASAN

4.1. Tahap Pelatihan

Tahap pelatihan membahas penerapan metode AI Project Cycle dengan pendekatan Natural Language Processing pada *chatbot* FIKTI UMSU seperti yang sudah disebutkan di Bab III. Sebuah model *chatbot* yang dirancang sedemikian rupa mulai dari tahap mengidentifikasi masalah sampai tahap mendeploy model *chatbot*. Berikut merupakan rincian tahap pelatihan dan tahap mendeploy model:

4.2. Problem Scoping

Chatbot adalah sebuah program yang dirancang untuk berkomunikasi dengan manusia dengan memberikan respon yang relevan terhadap pertanyaan pengguna. Dalam konteks pendidikan, khususnya di Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Muhammadiyah Sumatera Utara (FIKTI UMSU), penerapan teknologi ini dapat memberikan kontribusi signifikan dalam meningkatkan kualitas layanan akademik dan administrasi. Tujuan utama dari *chatbot* adalah untuk memberikan respons yang relevan dan berguna kepada mahasiswa, seolah-olah mereka sedang berkomunikasi dengannya.

- (What) Masalah: Banyak pertanyaan dari mahasiswa kepada staf biro administrasi yang seringkali bersifat rutin dan berulang, seperti pertanyaan mengenai prosedur akademik dan administrasi.
- Tujuan: Membangun *chatbot* berbasis AI untuk menjawab pertanyaan-pertanyaan tersebut secara otomatis, sehingga mengurangi beban kerja staf dan meningkatkan kepuasan mahasiswa
- (Why) Alasan Penting : Staf biro administrasi seringkali kewalahan dengan banyaknya pertanyaan yang sama dari mahasiswa, yang dapat mengurangi efisiensi kerja. *Chatbot* dapat mengotomatisasi proses penjawaban pertanyaan rutin, memungkinkan staf fokus pada tugas-tugas yang lebih kompleks dan penting. Ini juga meningkatkan aksesibilitas informasi bagi mahasiswa.
- (Who) Pengguna Utama: Mahasiswa yang membutuhkan informasi tentang prosedur akademik, syarat-syarat administrasi, dan pertanyaan terkait lainnya.
- (Where) Lokasi Masalah: Di lingkungan kampus, terutama di platform komunikasi digital yang digunakan oleh mahasiswa, seperti situs web kampus atau aplikasi mobile.

4.3. Data Acquisition

Data layanan akademik dan administratif dikumpulkan dari website FIKTI UMSU yang menyediakan panduan layanan akademik, serta dari wawancara dengan staf biro akademik dan administrasi melalui Google Form. Pertanyaan wawancara dikumpulkan di Google Form dan staf diminta menjawabnya. Data pola pertanyaan/percakapan diperoleh dari forum sosial, ChatGPT, dan pemikiran sendiri, semuanya dikumpulkan secara manual. Setelah data terkumpul, dataset dibuat dalam format JSON dengan 83 tag, 2.472 pola (patterns), dan 82 respon. Dataset ini berisi pertanyaan dan jawaban yang digunakan untuk melatih model *chatbot*.

4.4. Data Exploration

Memuat kesimpulan yang diperoleh dan saran-saran untuk penelitian selanjutnya (jika ada). Tuliskan Kesimpulan dan saran dalam 1 paragraf. Dalam tahap eksplorasi data, peneliti melakukan beberapa langkah untuk mempersiapkan data sebelum melatih model.

4.5. Mengubah dataset json menjadi dataframe

```
# Mengkonversi data json ke dataframe
msbot_dataset = pd.DataFrame({"patterns":inputs, "tags":tags})
```

Gambar 3. Mengubah dataset json menjadi dataframe

Dapat dilihat pada gambar 3 bahwa konversi ini memfasilitasi tahap berikutnya dalam pembelajaran mesin.

4.6. Menghilangkan Punctuasi

Membersihkan data teks dalam kolom patterns dengan menghilangkan tanda baca dan mengubah semua teks menjadi huruf kecil, sehingga mempersiapkan data tersebut untuk analisis atau proses Machine Learning lebih lanjut.

4.7. Lemmatization (Lematisasi)

```
#lemmatisasi
lemmatizer = WordNetLemmatizer()
words = [lemmatizer.lemmatize(w.lower()) for w in words if w not in ignore_words]
print(len(words), "unique lemmatized words", words)
```

Gambar 4. Lemmatization (Lematisasi)

Pada gambar 4 merupakan kode yang berfungsi mereduksi kata-kata menjadi bentuk dasarnya dengan tetap mempertimbangkan konteksnya.

4.8. Menyortir Data Kelas Tags

```
#Menyortir Data Kelas Tags
classes = sorted(list(set(classes))) # list yang berisi elemen-elemen unik yang telah diurutkan.
print(len(classes), "classes", classes) # mencetak panjang (jumlah elemen) dari variabel classes
```

Gambar 5. Menyortir data kelas tags

Pada gambar 5 merupakan proses klasifikasi untuk memastikan bahwa tidak ada duplikat yang dapat mengganggu analisis atau pelatihan model.

4.9. Mencari Jumlah Keseluruhan Data Teks

```
#Mencari Jumlah Keseluruhan Data Teks
print(len(documents), "documents")
```

Gambar 6. Mencari jumlah keseluruhan data teks

Dapat dilihat pada gambar 6 penghitungan data teks adalah untuk mengetahui skala dataset yang akan digunakan.

4.10. Tokenisasi data

```
# Tokenisasi Data
tokenizer = tf.keras.preprocessing.text.Tokenizer(num_words=2000)
tokenizer.fit_on_texts(msbot_dataset['patterns'])
train = tokenizer.texts_to_sequences(msbot_dataset['patterns'])
train
```

Gambar 7. Tokenisasi data

Pada gambar 7 merupakan proses yang memisahkan teks menjadi bagian-bagian yang lebih kecil, yang disebut token. Dalam konteks NLP, token bisa berupa kata, karakter, atau sub-kata.

4.11. Melakukan Padding

```
# melakukan padding
X_train = tf.keras.preprocessing.sequence.pad_sequences(train)
print(X_train)
```

Gambar 8. Melakukan padding

Dapat dilihat pada gambar 8 bahwa proses ini nilai tertentu (biasanya 0) ditambahkan ke data untuk mencapai panjang yang seragam di semua sampel data.

4.12. Encoding Label Atau Tag

```
# Encoding Label atau Tag
le = LabelEncoder()
Y_train = le.fit_transform(msbot_dataset['tags'])
print(Y_train)
```

Gambar 9. Melakukan padding

Dapat dilihat pada gambar 9 dimana proses ini teks diubah menjadi format yang dapat dipahami dan diproses oleh komputer, seperti mengubah kata-kata menjadi vektor numerik.

4.13. Input Length Dan Output Length

```
# Input length
input_shape = X_train.shape[1]
print("Input Shape : ", input_shape)

# Output length
output_length = le.classes_.shape[0]
print("Output length: ", output_length)
```

Gambar 10. Input length dan Output length

Dapat dilihat pada gambar 10 panjang input dan Output merujuk pada jumlah unit (seperti kata atau karakter) yang digunakan dalam satu sampel data setelah tokenisasi. Proses ini menentukan bentuk dari lapisan input yang akan diproses pada algoritma LSTM.

4.14. Menghitung Vocabulary Size

```
# Define vocabulary
vocabulary = len(tokenizer.word_index)
print("Number of unique words : ", vocabulary)
```

Gambar 11. Menghitung vocabulary size

Dapat dilihat pada gambar 11, proses ini akan mencakup setiap kata unik yang teridentifikasi dalam pertanyaan dan jawaban yang digunakan untuk melatih chatbot.

```
#simpan model pemrosesan teks tersebut dengan menggunakan format pickle.
pickle.dump(words, open('words.pkl','wb'))
pickle.dump(classes, open('classes.pkl','wb'))
```

4.15. Simpan Model

Teks yang sudah diproses selanjutnya akan disimpan untuk mempertahankan informasi model yang sudah dilatih.

4.16. Modeling

```
def msbot_bot(trainx, trainy, neuron, batch_size, epochs):
    # Input Layer
    i = tf.keras.Input(shape=(input_shape,))
    x = tf.keras.layers.Embedding(vocabulary+1,10)(i)
    # Hidden Layer
    x = tf.keras.layers.LSTM(neuron, return_sequences = True)(x)
    x = tf.keras.layers.Dropout(0.8)(x)
    # Hidden Layer
    x = tf.keras.layers.LSTM(neuron, return_sequences = True)(x)
    x = tf.keras.layers.Dropout(0.8)(x)
    # Flatten Layer
    x = tf.keras.layers.Flatten()(x)
    # Output Layer
    x = tf.keras.layers.Dense(output_length, activation="softmax")(x)
    model = tf.keras.models.Model(i,x)
    # Compile model
    model.compile(loss='sparse_categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
    # Menampilkan Parameter Model
    print(model.summary())
    history = model.fit(trainx,trainy,batch_size=batch_size,epochs=epochs,verbose=1,shuffle=False)
    return model, history
```

Gambar 12. Modeling LSTM

Pada tahap pemodelan, penggunaan arsitektur LSTM digunakan untuk membentuk, mengonfigurasi, melatih, dan mengembalikan model jaringan saraf yang sesuai dengan data yang diberikan.

Kode program diatas adalah sebuah Python function yang dirancang untuk membuat dan melatih sebuah model neural network menggunakan TensorFlow Keras. Fungsi ini dinamai msbot_bot dan mengambil beberapa parameter untuk melakukan pelatihan model.

4.17. Arsitektur Neural Network

```
# Input Layer
i = tf.keras.Input(shape=(input_shape,))
x = tf.keras.layers.Embedding(vocabulary+1,10)(i)
# Hidden Layer
x = tf.keras.layers.LSTM(neuron, return_sequences = True)(x)
x = tf.keras.layers.Dropout(0.8)(x)
# Hidden Layer
x = tf.keras.layers.LSTM(neuron, return_sequences = True)(x)
x = tf.keras.layers.Dropout(0.8)(x)
# Flatten Layer
x = tf.keras.layers.Flatten()(x)
# Output Layer
x = tf.keras.layers.Dense(output_length, activation="softmax")(x)
model = tf.keras.models.Model(i,x)
```

Gambar 13. Arsitektur Neural Network

Pada Gambar 13 merupakan arsitektur neural network menggunakan TensorFlow/Keras yang bertugas untuk pemrosesan bahasa alami.

4.18. Compile Model

```
# Compile model
model.compile(loss='sparse_categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
```

Gambar 14. Compile Model

Pada gambar 14 merupakan kodingan yang berfungsi untuk menentukan beberapa hal penting terkait dengan bagaimana model akan belajar dan dievaluasi selama proses pelatihan. Saat kode tersebut dieksekusi, model akan dikompilasi dengan fungsi kerugian `sparse_categorical_crossentropy`, optimizer 'adam', dan metrik evaluasi yang berupa 'accuracy'. Setelah kompilasi, model siap untuk dilatih (fit) dengan menggunakan data latihan yang sesuai dan kemudian dievaluasi untuk melihat seberapa baik performanya dalam memprediksi kelas-kelas dari data yang belum pernah dilihat sebelumnya.

4.19. Pelatihan Model

Menggambarkan langkah untuk melatih model dengan menggunakan data latihan (trainx sebagai input dan trainy sebagai output) yang telah dipersiapkan sebelumnya. Setelah menjalankan kode tersebut, model akan mulai melatih dirinya sendiri

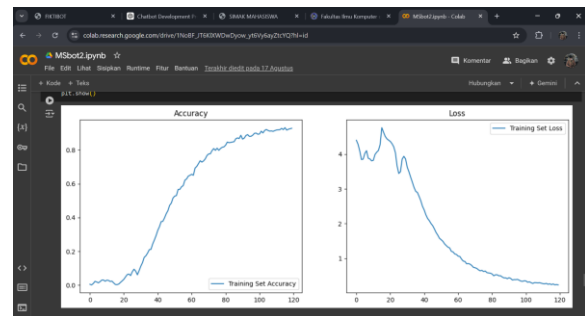
dengan data latihan yang diberikan, dan proses pelatihan akan berlangsung sejumlah epoch yang telah ditentukan, dengan batch size yang telah diatur, dan metrik evaluasi akan direkam dalam variabel history.

4.20. Output

```
return model, history
```

Gambar 15. Output

Pada Gambar 15 merupakan fungsi untuk mengembalikan model yang telah dilatih beserta riwayat pelatihannya.



Gambar 16. Grafik Akurasi LSTM

Pada gambar 16 merupakan tampilan plot akurasi dimana pada tahap ini peneliti melakukan analisa dari model yang telah dibuat dengan melihat tingkat akurasi dan loss nya dengan LSTM. Grafik menunjukkan peningkatan akurasi dan loss yang perlahan dan stabil setelah dilakukan pembersihan data dan menemukan nilai Hyperparameter yang optimal.

4.21. Evaluation

Pada tahap ini langkah evaluasi penting untuk menilai kualitas dan kemampuan model, membantu mengidentifikasi apakah model sedang belajar dengan baik, dan apakah ada langkah perbaikan yang perlu diambil. Cara untuk mengevaluasi performa model yang digunakan adalah bisa dengan menggunakan uji Blackbox untuk menilai sejauh mana model mampu memahami konteks pertanyaan. Peneliti melakukan uji coba Black Box dengan mengajukan 9 pertanyaan secara acak kepada chatbot. Data yang digunakan untuk pengujian merupakan kalimat acak yang berdasarkan pada tiap tags.

Tabel 1. Blackbox Testing

No	User Input	Chatbot Response Success/No	Tag	Status
1	Apa itu Fiktibot?	Response Success	About FIKTIBOT	1
2	gimana cara daftar kuliah di semester ini	Response Success	Pendaftaran matakuliah	1
3	apa aja syarat daftar mata kuliah	Response Success	Syarat pendaftaran mata kuliah	1
4	gimana cara bayar uang kuliah	Response Success	Prosedur Pembayaran uang kuliah	1
5	bro gimana cara dapat transkrip nilai,	Response Success	Cara mendapatkan transkrip nilai	1
6	apa aja syarat untuk ikut ujian	Response Success	Syarat ujian	1
7	gimana cara daftar kelas tambahan ,	Response Success	mendaftar kelas tambahan	1
8	syarat untuk mulai skripsi	No	Syarat skripsi	0
9	Apa aja ya syarat untuk memulai skripsi,	Response Success	Syarat skripsi	1

Dalam kasus ini, *chatbot* dapat menjawab semua 8 pertanyaan dengan benar, dan 1 pertanyaan dengan salah maka relevansi setiap jawaban adalah 1 (relevan) karena semuanya benar. Peneliti kemudian menggunakan nilai tersebut untuk melakukan perhitungan MAP. Mean Average Precision (MAP) adalah metrik evaluasi yang digunakan untuk mengukur seberapa baik *chatbot* merespons pertanyaan pengguna dengan menjawab pertanyaan-pertanyaan yang relevan dengan tepat.

Dengan hasil setiap pertanyaan dianggap relevan dan *chatbot* dapat menjawab semuanya dengan benar, MAP dapat dihitung dengan menggunakan rumus berikut:

$$\text{MAP} = \frac{\text{Total AP}}{\text{Total Pertanyaan}} \times 100\% \quad (1)$$

Jika dalam kasus ini semua jawaban relevan (diasumsikan benar), maka Average Precision (AP) untuk setiap pertanyaan adalah sebagai berikut:

$$\text{AP} = \frac{\text{Total jawaban relevan yang ditemukan}}{\text{Total jawaban yang diambil}} \quad (2)$$

Karena *chatbot* mampu menjawab 8 pertanyaan dengan benar dan hanya 1 kesalahan, AP untuk setiap pertanyaan akan menjadi 1. Sehingga, untuk 9 pertanyaan dengan asumsi semua jawaban relevan, total AP akan menjadi $8 \times 1 = 8$. Akhirnya, nilai Mean Average Precision (MAP) akan menjadi:

$$\text{MAP} = \frac{8}{82} \times 100\% = 97\% \quad (3)$$

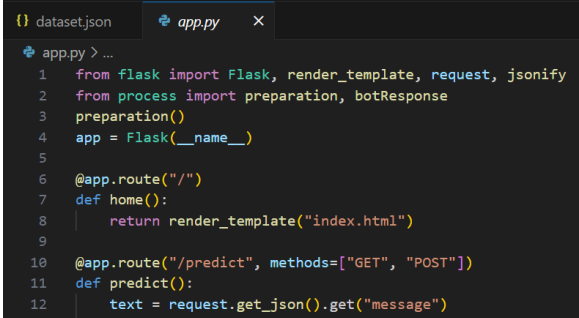
Dengan demikian, pengujian black box pada Tabel 4.1 dapat disimpulkan bahwa model *chatbot* mampu memberikan respons yang tepat terhadap 9 pertanyaan yang diajukan secara acak dengan benar dikarenakan penerapan *Natural Language Processing* (NLP) kita bisa memberikan input pertanyaan acak asalkan didalam pertanyaan tersebut terdapat kata kunci dari intent yang sudah di input. Hal ini mengindikasikan bahwa *chatbot* telah mampu memahami pertanyaan sehingga dapat memberikan respon dengan akurasi yang baik.

4.22. Deployment

Pada proses ini, peneliti memilih untuk menggunakan Flask sebagai framework web untuk menyajikan aplikasi atau situs web secara local (localhost). Proses deployment dapat dilakukan dengan tahapan berikut:

- Pembuatan folder baru bernama "Fiktibot", setelah membuat folder bernama "Fiktibot", langkah selanjutnya ada membuat subfolder yang terdiri dari folder "dataset", "model", "static", dan "templates". Di dalam subfolder "dataset", terdapat file dataset yang digunakan untuk penelitian ini. Pada subfolder "model" terdapat hasil dari proses pemodelan seperti tokenizer.pkl, le.pkl, dan fiktibot.h5. "Static" berisikan file-file pendukung seperti CSS, JavaScript, gambar (image), dan fontawesome. Sedangkan subfolder "templates" berisi file-file HTML yang digunakan untuk membuat tampilan pop-up *chatbot*.

- Membuat file Python yang terdiri dari app.py dan process.py.

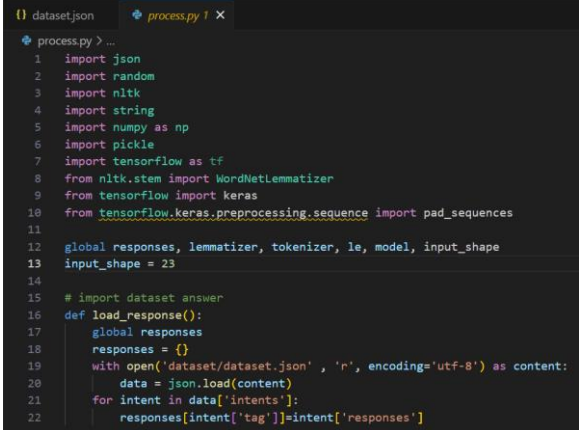


```

1 from flask import Flask, render_template, request, jsonify
2 from process import preparation, botResponse
3 preparation()
4 app = Flask(__name__)
5
6 @app.route("/")
7 def home():
8     return render_template("index.html")
9
10 @app.route("/predict", methods=["GET", "POST"])
11 def predict():
12     text = request.get_json().get("message")

```

Gambar 17. File app.py



```

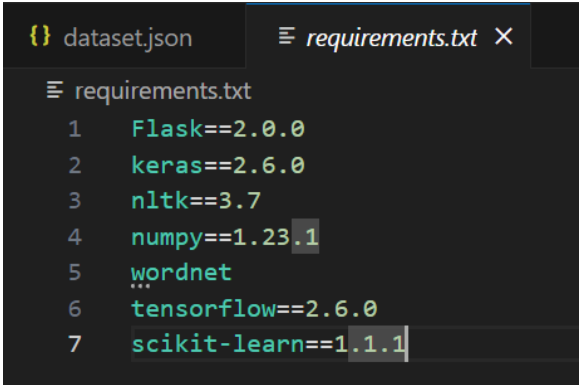
1 import json
2 import random
3 import nltk
4 import string
5 import numpy as np
6 import pickle
7 import tensorflow as tf
8 from nltk.stem import WordNetLemmatizer
9 from tensorflow import keras
10 from tensorflow.keras.preprocessing.sequence import pad_sequences
11
12 global responses, lemmatizer, tokenizer, le, model, input_shape
13 input_shape = 23
14
15 # import dataset answer
16 def load_response():
17     global responses
18     responses = {}
19     with open('dataset/dataset.json', 'r', encoding='utf-8') as content:
20         data = json.load(content)
21         for intent in data['intents']:
22             responses[intent['tag']] = intent['responses']

```

Gambar 18. File process.py

File app.py berperan sebagai file utama yang mengatur logika aplikasi Flask, sementara file process.py berisi fungsi-fungsi untuk pemrosesan data.

- Untuk menghindari eror dalam penginstalan package library, peneliti membuat file requirements.txt yang berisi daftar package library python beserta versi yang dibutuhkan proyek.



```

1 Flask==2.0.0
2 keras==2.6.0
3 nltk==3.7
4 numpy==1.23.1
5 wordnet
6 tensorflow==2.6.0
7 scikit-learn==1.1.1

```

Gambar 19. File requirements.txt

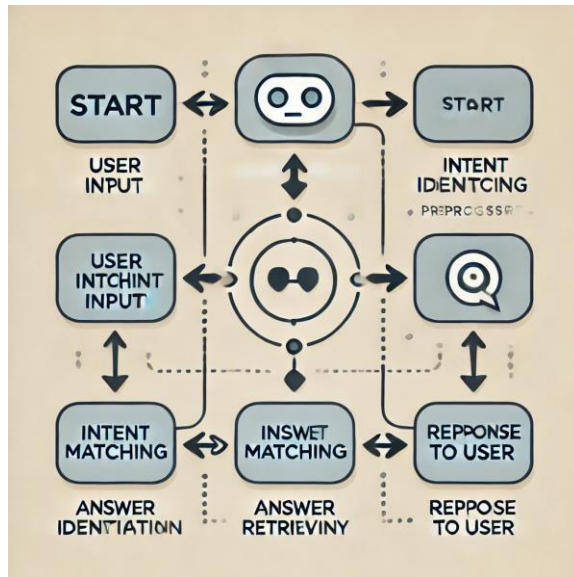
File diperlukan agar instalasi library Python sesuai dengan virtual environment tempat model dijalankan. Setelah membuat folder, subfolder, dan file, langkah berikutnya adalah membuat environment.

Setelah instalasi selesai, jalankan perintah berikut di Anaconda Prompt: ``python app.py``

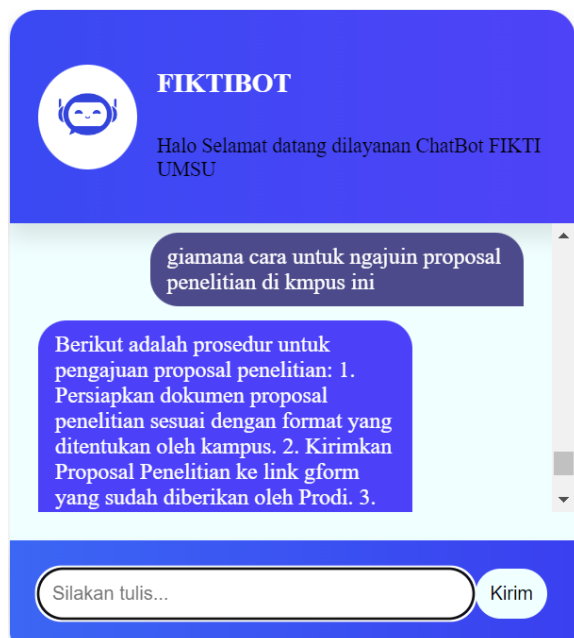
Perintah ini akan menjalankan proses deployment dan aplikasi atau website dapat diakses melalui link <http://127.0.0.1:5000/>. Jika berhasil, website yang dibuat akan muncul, menandakan proses deployment telah selesai dan website siap diakses.

4.23. Hasil dan Uji Coba

Uji coba akan dilakukan di tampilan hasil *chatbot* FIKTI UMSU yang sudah di deploy secara localhost menggunakan Flask.



Gambar 20. Flowchart sistem *chatbot*



Gambar 21. Uji Coba *Chatbot* FIKTI UMSU setelah di deploy

Pada gambar 4.25 juga menunjukkan testimoni bahwa *chatbot* mampu merespons dengan tepat bahkan ketika terdapat kesalahan pengetikan kalimat

tanya. Hal ini mengindikasikan keberhasilan peluncuran dan fungsi normal *chatbot* yang telah dikembangkan.

4.24. Kesimpulan Hasil Pengujian

Kesimpulan hasil pengujian *Chatbot* FIKTI UMSU yang sudah di deploy memiliki kelebihan dan kekurangan, berikut adalah kelebihan dan kekurangan dari aplikasi *chatbot* yang peneliti kembangkan:

Kelebihan:

- Mampu memberikan respon cepat dan akurat terkait layanan akademik dan administratif.
- Mampu memahami konteks pertanyaan meskipun terdapat kesalahan pengetikan dalam pertanyaan yang diterima.

Kekurangan:

- Chatbot* tidak dapat menolak pertanyaan yang berada di luar konteks atau pengetahuannya, sehingga dapat memberikan respon yang kurang sesuai terhadap pertanyaan di luar konteks.
- Chatbot* tidak mampu memahami bahasa asing, dialek, atau istilah khusus selain Bahasa Indonesia.

5. KESIMPULAN DAN SARAN

Adapun Kesimpulan yang dari hasil pengujian *chatbot* NLP ini ialah, *chatbot* sudah dapat menjawab pertanyaan acak secara benar berdasarkan data intents yang ada pada tiap tags dan dengan setiap kata kunci yang ada di tiap input yang diberikan oleh pengguna. Pengembangan *Chatbot* FIKTI UMSU berbasis AI dan *Natural Language Processing* (NLP) telah memberikan kontribusi signifikan dalam meningkatkan layanan akademik dan administratif, mengurangi beban kerja staf biro administrasi, serta meningkatkan kepuasan mahasiswa di Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Muhammadiyah Sumatera Utara. Proses pengembangan *chatbot* ini melibatkan AI Project Cycle dan penggunaan arsitektur Long Short-Term Memory (LSTM) yang menghasilkan akurasi tinggi dalam memahami dan menjawab pertanyaan mahasiswa. Meskipun menunjukkan kinerja yang baik, *chatbot* ini memiliki keterbatasan, seperti menangani pertanyaan di luar konteksnya. Oleh karena itu, pengembangan lebih lanjut diperlukan dengan menambahkan lebih banyak data dan pola percakapan, integrasi dengan platform lain, serta peningkatan kemampuan bahasa dan fungsionalitas. Evaluasi dan monitoring berkala juga disarankan untuk meningkatkan efektivitas layanan *chatbot* dan memberikan pengalaman pengguna yang lebih baik.

DAFTAR PUSTAKA

- [1] G. R. Kusnanda, A. A. N. H. Susila, and M. Sukarsa, "Perancangan *Chatbot* Hotel dengan Model *Natural Language Processing Chatbot* dan Button Based *Chatbot* I Gede Ryoga Kusnanda a1 , I Made Sukarsa a2 , Anak Agung

- Ngurah Hary Susila a3,” J. Ilm. Teknol. dan Komput., vol. 3, no. 1, Apr. 2022.
- [2] D. Apriliani, S. Febbi Handayani, I. Triadi Saputra, T. Informatika, and P. Harapan Bersama, “Implementasi *Natural Language Processing* (NLP) Dalam Pengembangan Aplikasi *Chatbot* Pada SMK YPE Nusantara Slawi Implementation of *Natural Language Processing* (NLP) in the Development of a *Chatbot* Application at SMK YPE Nusantara Slawi,” 2023.
- [3] F. Sudirjo, T. Purwati, Y. Unggul Budiman, and M. Manuhutu, “Analisis Dampak Strategi Pemasaran Digital dalam Meningkatkan Loyalitas Pelanggan: Perspektif Industri E-commerce,” J. Pendidik. Tambusai, vol. 7, no. 2, pp. 7524–7532, 2023.
- [4] D. O. Sihombing, “Implementasi *Natural Language Processing* (NLP) dan Algoritma Cosine Similarity dalam Penilaian Ujian Esai Otomatis,” J. Sist. Komput. dan Inform., vol. 4, no. 2, p. 396, Dec. 2022, doi: 10.30865/json.v4i2.5374.
- [5] A. F. Zahwa, R. Fiati, and A. C. Murti, “Implementasi *Chatbot* untuk *Customer service* menggunakan Metode *Natural Language Processing* (NLP) (Studi Kasus Website Theme62.com),” J I M P - J. Inform. Merdeka Pasuruan, vol. 7, no. 2, p. 82, Sep. 2023, doi: 10.51213/jimp.v7i2.542.
- [6] M. Ahmadi, “Komunikasi Simbolik: Implikasi Penggunaan *Chatbot* Sebagai Upaya Peningkatan Efektivitas Pelayanan *Customer service*,” J-KIs J. Komun. Islam, vol. 4, no. 1, pp. 101–110, Jun. 2023, doi: 10.53429/jkis.v4i1.594.
- [7] A. Elcholiqi and A. Musdholifah, “*Chatbot* in Bahasa Indonesia using NLP to Provide Banking Information,” IJCCS (Indonesian J. Comput. Cybern. Syst., vol. 14, no. 1, p. 91, Jan. 2020, doi: 10.22146/ijccs.41289.
- [8] M. A. Prasetya, M. Wulandari, and S. A. Nikmah, “Implementasi NLP(*Natural Language Processing*) Dasar pada Analisis Sentiment Review Spotify,” 2024.
- [9] A. Isma, S. Sahalik Rahman, A. Setiawan Syam, and N. Sari, “Fundamental and Applied Management Journal Analisis Penggunaan *Chatbot* Berbasis AI pada Model Hybrid di Jurusan Teknik Informatika dan Komputer,” vol. 1, no. 2, 2023, doi: 10.61220/voice.
- [10] F. Azimah, K. Rizky, and N. Wardani, “SISTEM PENDETEKSI GEJALA AWAL COVID-19 DENGAN PENGGUNAAN METODE AI PROJECT CYCLE,” 2022. [Online]. Available: www.kaggle.com.
- [11] F. Fitra Ramadhan, “*Chatbot* pada E-Commerce berbasis Android dengan Pendekatan *Natural Language Processing*,” JCSE J. Comput. Sci. an Eng., vol. 2, no. 1, pp. 27–39, 2021, doi: 10.36596/jcse.v2i1.100.
- [12] W. Kangharnando and H. Adiya, “Penerapan Teknologi *Chatbot* Pada Website Mobile E-Commerce Roti Papa,” J. Mhs. Apl. Teknol. Komput. dan Inf., vol. 4, pp. 1–7, 2022.
- [13] M. Akbar Nasution et al., “Implementasi NLP Dalam Pembuatan *Chatbot Customer service* Publisher Jurnal Studi Kasus LARISMA,” 2024.