

**IMPLEMENTASI *NATURAL LANGUAGE PROCESSING*
DALAM PERANCANGAN APLIKASI *CHATBOT*
PADA FIKTI UMSU**

SKRIPSI

DISUSUN OLEH

MUHAMMAD IZZUDDIN MUBAROK

NPM. 2009020142



**PROGRAM STUDI TEKNOLOGI INFORMASI
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
UNIVERSITAS MUHAMMADIYAH SUMATERA UTARA**

MEDAN

2024

LEMBARAN PENGESAHAN

Judul Skripsi : IMPLEMENTASI NATURAL LANGUAGE PROCESSING
DALAM PERANCANGAN APLIKASI CHATBOT PADA
FIKTI UMSU

Nama Mahasiswa : MUHAMMAD IZZUDDIN MUBAROK

NPM : 2009020142

Program Studi : TEKNOLOGI INFORMASI

Menyetujui Komisi Pembimbing



(Mahardika Abdi Prawira S.Kom., M.Kom)

NIDN. 0117088902

Ketua Program Studi



(Fatma Sari Hutagalung, S.Kom, M.Kom)

NIDN. 0117019301

Dekan



(Dr. Al-Khowarizmi, S.Kom., M.Kom.)

NIDN. 0127099201

PERNYATAAN ORISINILITAS

IMPLEMENTASI NATURAL LANGUAGE PROCESSING DALAM PERANCANGAN APLIKASI CHATBOT PADA FIKTI UMSU

SKRIPSI

Saya menyatakan bahwa karya tulis ini adalah hasil karya sendiri, kecuali beberapa kutipan dan ringkasan yang masing-masing disebutkan sumbernya.

Medan, Juli 2024

Yang membuat pernyataan



MUHAMMAD IZZUDDIN MUBAROK

NPM. 2009020142

PERNYATAAN PERSETUJUAN PUBLIKASI KARYA ILMIAH UNTUK KEPENTINGAN AKADEMIS

Sebagai sivitas akademika Universitas Muhammadiyah Sumatera Utara, saya bertanda tangan dibawah ini:

Nama : Muhammad Izzuddin Mubarak
NPM : 2009020142
Program Studi : Teknologi Informasi
Karya Ilmiah : Skripsi

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Muhammadiyah Sumatera Utara Hak Bedas Royalti Non-Eksekutif (*Non-Exclusive Royalty free Right*) atas penelitian skripsi saya yang berjudul:

IMPLEMENTASI NATURAL LANGUAGE PROCESSING DALAM PERANCANGAN APLIKASI CHATBOT PADA FIKTI UMSU

Beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Non-Eksekutif ini, Universitas Muhammadiyah Sumatera Utara berhak menyimpan, mengalih media, memformat, mengelola dalam bentuk database, merawat dan mempublikasikan Skripsi saya ini tanpa meminta izin dari saya selama tetap mencantumkan nama saya sebagai penulis dan sebagai pemegang dan atau sebagai pemilik hak cipta.

Demikian pernyataan ini dibuat dengan sebenarnya.

Medan, Juli 2024

Yang membuat Pernyataan



MUHAMMAD IZZUDDIN MUBAROK

NPM. 2009020142

RIWAYAT HIDUP

DATA PRIBADI

Nama Lengkap : Muhammad Izzuddin Mubarak
Tempat dan Tanggal Lahir : Binjai, 10 Oktober 2003
Alamat Rumah : Binjai
Telepon/Faks/HP : 081265573897
E-mail : mubarokizzuddin788@gmail.com
Instansi Tempat Kerja : -
Alamat Kantor : -

DATA PENDIDIKAN

SD : SD HARAPAN II BINJAI TIMUR
TAMAT: 2014
SMP : SMP NEGERI 4 BINJAI TIMUR
TAMAT: 2017
SMA : MADRASAH ALIYAH NEGERI KOTA BINJAI
TAMAT: 2020

KATA PENGANTAR



Puji syukur atas kehadiran Allah SWT, berkat limpahan rahmat, hidayah dan karunianya, penulis bisa menyelesaikan skripsi dengan judul “**IMPLEMENTASI NATURAL LANGUAGE PROCESSING DALAM PERANCANGAN APLIKASI CHATBOT PADA FIKTI UMSU**”. Skripsi ini adalah salah satu dari beberapa persyaratan untuk menyelesaikan pendidikan dan memperoleh gelar sarjana pada program studi S1 Teknologi Informasi di Universitas Muhammadiyah Sumatera Utara.

Penyusunan skripsi ini tidak lepas dari bimbingan, bantuan, arahan dan dukungan dari berbagai pihak terkait. Oleh karena itu pada kesempatan ini penulis menyampaikan terimakasih kepada:

1. Bapak Prof. Dr. Agussani, M.AP., Rektor Universitas Muhammadiyah Sumatera Utara (UMSU)
2. Bapak Dr. Al-Khowarizmi, S.Kom., M.Kom, Dekan Fakultas IlmuKomputer dan Teknologi Informasi (FIKTI) UMSU.
3. Ibu Fatma Sari Hutagalung, S.Kom., M.Kom, Ketua Program Studi Teknologi Informasi
4. Bapak Mhd. Basri, S.Si, M.Kom, Sekretaris Program Studi Teknologi Informasi
5. Pembimbing saya Bapak Mahardika Abdi Prawira, S.Kom,M.Kom
6. Ayahanda Eka Saptono Serta Ibunda Dini Lamsihar Munthe atas doa dan kasih sayangnya yang tulus dan tak terhingga kepada penulis, terima kasih selalu menjadi *support sytem* terbaik sepanjang perjalanan hidup penulis yang selalu memberikan kasih sayang tak terhingga dengan penuh cinta, tulus dan selalu memberikan motivasi, semangat serta do'a yang terus mengalir dan tak pernah henti di setiap langkah hingga penulis mampu menyelesaikan studinya sampai sarjana.
7. Kakak saya yang telah memberikan motivasi dan perhatiannya untuk terus

bersemangat menyelesaikannya.

8. Seseorang yang bernama Fatimah Al Mujahhidah selaku seseorang yang tak kalah penting kehadirannya. Yang selalu mendukung, menghibur, mendengarkan keluh kesah, serta memberikan tenaga dan waktunya kepada penulis hingga saat ini.
9. Kepada teman-teman seperjuangan saya terkhusus teman sekelas saya yang memberikan banyak bantuan, menemani dalam penelitian, memberikan motivasi dan telah menemani perjuangan saya dalam menyelesaikan skripsi ini.
10. Semua Pihak yang terlibat baik langsung maupun tidak langsung dalam pengerjaan Skripsi ini yang tidak penulis sebutkan satu persatu diucapkan terima kasih.

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.

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

ABSTRACT

The advancement of information and communication technology has significantly impacted various aspects of human life, including education. A notable development is the emergence of artificial intelligence technologies such as chatbots. The use of chatbots has proven effective in enhancing user experience and providing instant solutions across various domains. At FIKTI UMSU, the increasing number of students and the need for prompt and accurate information services present challenges in delivering optimal services to both students and staff. Common inquiries related to class schedules, scholarships, administrative procedures, and other academic services often face delays due to limited operational hours and staff availability, leading to a backlog of work. This situation necessitates an innovative solution to ensure optimal service delivery without excessively burdening the administrative staff. To address these challenges, a real-time information service solution that is accessible at any time is required. A chatbot based on Natural Language Processing (NLP) technology represents a viable solution to meet this need. Chatbots are computer programs designed to interact with humans through conversation, either text or voice. Advanced chatbots utilize NLP technology to understand context and provide relevant responses. Implementing an NLP-based chatbot at FIKTI UMSU is expected to significantly improve academic and administrative service quality. The chatbot can automatically handle common queries, reducing the workload on administrative staff and allowing them to focus on more complex tasks. Additionally, the chatbot will provide 24/7 access to information, enabling students and faculty to obtain information anytime without being restricted by office hours. However, to realize these benefits, proper design and implementation are essential, including selecting appropriate technologies, gathering relevant conversation data, and training an accurate NLP model. This study aims to develop an NLP-based chatbot for FIKTI UMSU, focusing on its ability to understand and respond to a range of academic and administrative queries. The Black Box testing results show that the chatbot is able to correctly respond to 8 randomly posed questions.

Keywords: Chatbot, Natural Language Processing (NLP), Academic Services, Black Box Testing

DAFTAR ISI

LEMBARAN PENGESAHAN	i
PERNYATAAN ORISINILITAS	ii
PERNYATAAN PERSETUJUAN PUBLIKASI KARYA ILMIAH UNTUK KEPENTINGAN AKADEMIS	iii
RIWAYAT HIDUP	iv
KATA PENGANTAR	v
ABSTRAK	vii
DAFTAR ISI	9
DAFTAR GAMBAR	11
DAFTAR TABEL	12
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Batasan Masalah.....	3
1.4 Tujuan Penelitian.....	4
1.5 Manfaat Penelitian.....	4
BAB II LANDASAN TEORI	6
2.1 Penelitian Terdahulu.....	6
2.2 <i>Chatbot</i>	8
2.3 <i>Natural Language Processing</i>	10
2.4 <i>AI Project Cycle</i>	11
BAB III METODOLOGI PENELITIAN	13
3.1 Pengumpulan Data	13
3.2 Tempat Penelitian.....	14
3.3 Metodologi Pengembangan Sistem	14
3.3.1 Problem Scoping.....	15
3.3.2 Data Acquisition	15

3.3.3 <i>Data Exploration</i>	16
3.3.4 <i>Modeling</i>	19
3.3.5 <i>Evaluating</i>	19
3.3.6 <i>Deployment</i>	21
BAB IV HASIL DAN PEMBAHASAN	22
4.1 Tahap Pelatihan	22
4.1.1 Problem Scoping	22
4.1.2 Data Acquisition	23
4.1.3 Data Exploration	25
4.1.4 Modeling	28
4.1.5 Evaluation	33
4.1.6 Deployment	37
4.2 Hasil dan Uji Coba	42
4.3 Kesimpulan Hasil Pengujian	44
BAB V KESIMPULAN DAN SARAN	45
5.1 Kesimpulan	45
5.2 Saran	46
DAFTAR PUSTAKA	48

DAFTAR GAMBAR

Gambar 3. 1 AI Project Cycle	15
Gambar 3. 2 Tampilan Dataset	16
Gambar 3. 3 Arsitektur Neural Network.....	19
Gambar 4. 1 File json	24
Gambar 4. 2 Dataset json tag,patterns & response.....	24
Gambar 4. 3 Mengubah dataset json menjadi dataframe	25
Gambar 4. 4 Menghilangkan Tanda Baca	25
Gambar 4. 5 Lemmatization (Lematisasi).....	26
Gambar 4. 6 Menyortir data kelas tags	26
Gambar 4. 7 Mencari jumlah keseluruhan data teks	26
Gambar 4. 8 Tokenisasi data	26
Gambar 4. 9 Melakukan padding	27
Gambar 4. 10 Melakukan padding	27
Gambar 4. 11 Input length dan Output length.....	28
Gambar 4. 12 Menghitung vocabulary size	28
Gambar 4. 13 Simpan Model	28
Gambar 4. 14 Modeling LSTM.....	29
Gambar 4. 15 Fungsi msbot	29
Gambar 4. 16 Arsitektur Neural Network.....	30
Gambar 4. 17 Compile Model.....	31
Gambar 4. 18 Pelatihan Model.....	31
Gambar 4. 19 Output.....	32
Gambar 4. 20 Grafik Akurasi LSTM	32
Gambar 4. 21 Folder Fiktibot.....	38
Gambar 4. 22 File app.py	39
Gambar 4. 23 File process.py bagian (1)	39
Gambar 4. 24 File process.py bagian (2)	40
Gambar 4. 25 File requirements.txt.....	40
Gambar 4. 26 Tampilan hasil Chatbot FIKTI UMSU.....	42
Gambar 4. 27 Uji Coba Chatbot FIKTI UMSU setelah di deploy.....	43

DAFTAR TABEL

Tabel 4. 1 Uji Blackbox	33
-------------------------------	----

BAB 1

PENDAHULUAN

1.1 Latar Belakang

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 (Apriliani et al., 2023). Penggunaan chatbot telah menjadi cara yang efektif untuk meningkatkan pengalaman pengguna dan memberikan solusi instan dalam berbagai bidang (Rosyadi et al., 2020).

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 (Guntoro, *et al* 2020). Layanan ini belum dapat dilakukan 24 jam karena keterbatasan waktu kerja. Sehingga pertanyaan yang disampaikan harus menunggu sampai petugas sudah siap untuk menjawab pertanyaan (Ramadhan, 2021). 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 (Adiya & Kangharnando, 2022). Chatbot adalah program komputer yang dirancang untuk berinteraksi dengan manusia melalui percakapan, baik melalui teks atau suara (Zahwa et al., 2023). Chatbot yang lebih canggih menggunakan teknologi NLP, sehingga mampu memahami konteks dan memberikan respon yang relevan (Kusnanda et al., 2022).

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 (Apriliani et al., 2023). Selain itu, chatbot juga memperluas akses informasi dengan layanan yang tersedia 24/7 (Adiya & Kangharnando, 2022), 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 (Elcholiqi & Musdholifah, 2020). 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.

1.2 Rumusan Masalah

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

1. Bagaimana proses merancang *chatbot* dengan menggunakan metode NLP?
2. Bagaimana kemampuan chatbot dalam memahami dan merespon berbagai pertanyaan akademik dan administratif yang diajukan?

1.3 Batasan Masalah

Berdasarkan rumusan masalah diatas, maka terdapat beberapa batasan masalah pada penelitian ini yang berfungsi agar penelitian ini tidak terpecah. Adapun batasan-batasan masalah pada penelitian ini yaitu:

1. Perancangan *chatbot* hanya memuat informasi tentang pertanyaan akademik dan administratif
2. Metodologi pengembangan sistem menggunakan metode *AI Project Cycle*.
3. Penulis membatasi dataset sebanyak 82 tag.
4. *Chatbot* tidak dapat menolak pertanyaan yang berada di luar konteks atau pengetahuannya, sehingga dapat memberikan respon yang kurang sesuai terhadap pertanyaan di luar konteks.
5. *Chatbot* tidak mampu memahami bahasa asing, dialek, atau istilah khusus selain Bahasa Indonesia.
6. *Chatbot* tersebut sudah bisa di deploy tetapi masih dalam mode offline atau standalone desktop menggunakan flask.

1.4 Tujuan Penelitian

Adapun tujuan penelitian ini adalah sebagai berikut:

1. Mengembangkan dan mengimplementasikan aplikasi chatbot berbasis Natural Language Processing (NLP) di Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Muhammadiyah Sumatera Utara (FIKTI UMSU).
3. Menilai kemampuan chatbot dalam memahami dan merespons berbagai pertanyaan akademik dan administratif.
4. Memastikan chatbot mampu memberikan respon yang relevan dan akurat melalui proses pelatihan model NLP yang tepat.
5. Mengurangi beban kerja staf administrasi dan meningkatkan efisiensi layanan informasi di FIKTI UMSU.

1.5 Manfaat Penelitian

Berikut ini merupakan manfaat penelitian ini bagi Institusi, Keilmuan, penulis, dan masyarakat:

Bagi Penulis

1. Pengembangan Keahlian Teknis: Penulis akan mendapatkan pengalaman langsung dalam merancang dan mengimplementasikan *chatbot* menggunakan teknologi NLP, yang akan meningkatkan keahlian teknis dan pemahaman tentang AI dan pemrograman.
2. Pemecahan Masalah Nyata: Penulis dapat mengaplikasikan teori dan konsep yang telah dipelajari untuk memecahkan masalah nyata, meningkatkan kemampuan analitis dan pemecahan masalah.

Bagi Keilmuan

1. Kontribusi pada Penelitian AI dan NLP: Penelitian ini akan menambah literatur dan pengetahuan dalam bidang AI, khususnya dalam penerapan NLP untuk pengembangan *chatbot*.
2. Pengembangan Metodologi: Penelitian ini dapat menjadi referensi bagi penelitian lain yang ingin mengembangkan metode serupa dalam pembuatan dan implementasi *chatbot*.

Bagi Institusi

1. Peningkatan Reputasi Akademik: Penelitian ini dapat meningkatkan reputasi institusi pendidikan yang terkait dengan penulis, melalui publikasi hasil penelitian dan kontribusi ilmiah yang dihasilkan.
2. Pengurangan Beban Kerja Staf: Chatbot dapat mengurangi beban kerja staf administrasi dengan menjawab pertanyaan umum secara otomatis, sehingga staf dapat fokus pada tugas-tugas yang lebih kompleks dan strategis.

Bagi Masyarakat

Kemudahan Akses Informasi: Penelitian ini memungkinkan masyarakat, terutama mahasiswa dan staf akademik, untuk mendapatkan informasi akademik dan administratif kapan saja, tanpa terbatas oleh jam operasional staf.

BAB II

LANDASAN TEORI

2.1 Penelitian Terdahulu

Elcholiqi & Musdholifah (2020), 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.

Zahwa *et al.* (2023), 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.

Nasution *et al.* (2024), 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%.

Apriliani *et al.* (2023) mereka mengimplementasikan NLP dalam membuat *chatbot* untuk memberikan layanan informasi seputar sekolah di SMK YPE Nusantara Slawi. Metode NLP digunakan karena dapat mengolah dan menganalisis data teks secara komprehensif, mengenali pola linguistik, dan memahami konteks yang kompleks. Data yang diperlukan untuk pelatihan model diperoleh dari file JSON yang berisi *intents* dan *patterns* yang terkait dengan setiap *intent*, kemudian diolah dalam format yang cocok untuk pelatihan model. NLP memungkinkan *chatbot* untuk mengurai informasi yang lebih kaya dan terstruktur dari dataset, serta memberikan respon yang lebih kontekstual dan responsif kepada pengguna. Hasil dari penerapan *Natural Language Processing* (NLP) dalam pengembangan *chatbot* adalah kemampuan *chatbot* untuk merangkum informasi kompleks menjadi format yang lebih ringkas dan mudah dimengerti, serta memberikan layanan interaktif yang lebih canggih dan berarti kepada pengguna. *Chatbot* dapat mengenali maksud pengguna, mengekstrak informasi penting dari dataset, dan memberikan jawaban yang relevan dan kontekstual, meningkatkan aksesibilitas dan pengalaman pengguna dalam mendapatkan informasi dari dataset yang kaya dan beragam.

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 Isma *et al.* (2023) mereka menganalisis dampak positif penggunaan *chatbot* dalam mengoptimalkan proses penerimaan informasi di lingkungan pendidikan khususnya di Jurusan Teknik Informatika dan Komputer.. Dengan mengintegrasikan *chatbot* ke dalam website, mahasiswa dapat dengan mudah mengakses panduan yang akurat dan responsif, serta memfasilitasi proses penerimaan informasi dengan lebih baik. Selain itu, integrasi *chatbot* ke dalam website juga dapat memberikan kemudahan bagi mahasiswa dalam berinteraksi dengan teknologi *chatbot* untuk mendapatkan informasi yang mereka butuhkan. implementasi *chatbot* dilakukan dengan menggunakan pendekatan *Natural Language Processing* (NLP) untuk memberikan panduan yang akurat kepada mahasiswa serta memfasilitasi proses penerimaan informasi dengan lebih baik. Meskipun terdapat variasi dalam respon, hasil survei menunjukkan bahwa sebagian besar responden merasa puas dengan penggunaan *chatbot* dan melihatnya sebagai alat yang dapat 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 Ahmadi (2023) *customer service* perlu digantikan dengan *chatbot* karena *chatbot* dapat bekerja selama 24 jam, memberikan dukungan konsumen yang konsisten, dan memberikan jawaban yang konsisten dan akurat atas pertanyaan pelanggan tanpa dipengaruhi oleh emosi atau kelelahan seperti manusia. Metode penelitian yang digunakan adalah kualitatif deskriptif dengan mengambil data dari perusahaan yang telah menerapkan *chatbot*. Hasil penelitian menunjukkan bahwa penggunaan *chatbot* dapat meningkatkan efisiensi, menyediakan dukungan pelanggan yang konsisten, meningkatkan kepuasan konsumen, dan mengurangi biaya bagi perusahaan. *Chatbot* dapat membantu dalam memberikan layanan kepada konsumen dan meningkatkan kinerja *customer service* secara efektif.

Teknologi *chatbot* dapat membantu berkembangnya sektor pariwisata mengingat setiap tahunnya banyak tempat wisata baru dibuka. Salah satunya khusus perhotelan, kemampuan yang dimiliki *chatbot* dapat selalu aktif tanpa diperlukannya tenaga manusia. Kusnanda *et al.* (2022) membuat *chatbot* hotel untuk memudahkan wisatawan mendapatkan informasi seputar hotel. Proses perancangan *chatbot* menggunakan *Natural Language Processing* (NLP). Teknologi NLP memungkinkan *chatbot* untuk memahami dan merespons input pengguna dalam bentuk bahasa alami. Dengan NLP, *chatbot* dapat mengenali pola kalimat, entitas, dan konteks dari input pengguna, sehingga dapat memberikan respon yang lebih akurat dan relevan. Hal ini memungkinkan *chatbot* untuk tetap responsif dan efektif dalam berinteraksi dengan pengguna, bahkan ketika terdapat kesalahan dalam penelitian atau struktur kalimat.

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. Sihombing (2022) mengimplementasikan NLP dengan menggunakan algoritma *Cosine Similarity* untuk melakukan penilaian otomatis ujian esai. Data yang digunakan dalam penelitian ini adalah 5 data jawaban mahasiswa diambil dari data jawaban mahasiswa yang telah mengerjakan soal esai. Data jawaban mahasiswa diolah melalui *tokenization*, *lower casing*, *stopwords removal*, *stemming*, dan *lemmatization*. Hasil penilaian akhir menunjukkan tingkat kemiripan jawaban menggunakan NLP dan *Cosine Similarity* untuk setiap mahasiswa.

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 Prasetya *et al.* (2024), mereka melakukan analisis sentimen terhadap ulasan *Spotify* menggunakan metode *Naive Bayes* dan *Logistic Regression* berbasis NLP. Dataset di ambil dari *Kaggle* yang berisi ulasan pengguna aplikasi *Spotify*, dataset diolah melalui beberapa tahap prapemrosesan, termasuk ekspansi kontraksi, pengecilan huruf, penghapusan stopwords, dan lemmatisasi. Tahap ini penting untuk membersihkan data dan memastikan kualitas data yang digunakan dalam analisis. Setelah diterapkan NLP pada analisis sentimen review *Spotify* menggunakan algoritma *Naive Bayes* dan *Logistic Regression*, hasilnya

menunjukkan bahwa *Logistic Regression* memiliki akurasi sebesar 79%, sedangkan *Naive Bayes* memiliki akurasi sebesar 76%. *Logistic Regression* memiliki akurasi yang sedikit lebih tinggi daripada *Naive Bayes* dalam analisis sentimen ulasan *Spotify*.

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 Nurfiyah & Ramadhani (2024), mereka melakukan analisis dan perancangan sistem informasi menggunakan metode *Natural Language Processing* (NLP) untuk media informasi mahasiswa di Universitas Bhayangkara Jakarta raya. Implementasi NLP diterapkan pada chatbot yang bertujuan untuk meningkatkan efektivitas penyampaian informasi akademik kepada mahasiswa melalui media interaktif. NLP memungkinkan komputer untuk memahami dan merespons instruksi bahasa yang diinputkan oleh pengguna melalui lima langkah analisis, yaitu *Lexical Analysis*, *Syntax Analysis*, *Semantic Analysis*, *Discourse Integration*, dan *Pragmatics Analysis*. Sistem ini diuji menggunakan *black box testing* untuk memastikan keakuratan dan kinerja sistem, hasil dari sistem ini menggunakan *chatbot* yang dapat membantu staff fakultas dalam 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 transformasional 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. Azimah & Wardani (2022) membuat sebuah sistem pendeteksi Covid-19 menggunakan siklus *AI Project Cycle* untuk mengklasifikasikan pasien yang terpapar Covid-19 atau tidak. Tahapan pelaksanaan dari siklus *AI Project Cycle* meliputi *Problem Scoping*, *Data Acquisition*, *Data Exploration*, *Modelling*, dan *Deployment*. Mereka mengidentifikasi dan memetakan batasan masalah terkait klasifikasi risiko terpapar Covid-19 berdasarkan gejala yang muncul. Selanjutnya, tahap *Data Acquisition* dilakukan dengan mengumpulkan dataset dari situs resmi *Kaggle* untuk digunakan dalam penelitian. Proses *Data Exploration* dilakukan untuk memahami karakteristik dataset dan pola data yang ada. Kemudian, dalam tahap *Modelling*, menggunakan tiga model, yaitu *Logistic Regression*, *Random Forest Classifier*, dan *Support Vector Machine*, untuk mengklasifikasi pasien yang terpapar Covid-19 atau tidak. Tahap *Deployment* dilakukan dengan mengimplementasikan model AI dalam bentuk *website* agar dapat diakses dengan mudah oleh pengguna, melalui proses desain UI/UX, pengembangan *website* menggunakan HTML, JS, dan CSS, pembuatan *back-end* dan integrasi menggunakan Flask, serta *hosting website* ke Heroku. Hasil dari penerapan metode *AI Project Cycle* sistem memiliki tingkat akurasi lebih dari 80% dengan tingkat *precision* dan *recall* yang tinggi. Untuk digunakan secara publik, sistem klasifikasi COVID-19 ini juga diimplementasikan secara *online* dengan Heroku.

BAB III

METODOLOGI PENELITIAN

Pada Bab ini akan membahas tentang pengumpulan data, tempat penelitian, dan metodologi pengembangan sistem perancangan *chatbot* pada FIKTI UMSU.

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.

1. Wawancara: Wawancara adalah metode pengumpulan data yang melibatkan interaksi langsung antara penulis dan responden. Pada penelitian ini, wawancara akan dilakukan dengan pihak terkait di FIKTI UMSU, seperti staf administrasi. Tujuan wawancara ini adalah untuk memahami kebutuhan pengguna, mengidentifikasi permasalahan yang ada dalam layanan akademik dan administratif, serta mengumpulkan feedback untuk perbaikan prototipe chatbot. Melalui wawancara, penulis dapat menggali informasi mengenai jenis pertanyaan yang sering diajukan, kesulitan yang sering dihadapi pengguna, serta preferensi dalam berinteraksi dengan sistem chatbot. Selain itu, wawancara juga berguna untuk mendapatkan masukan yang membantu dalam melakukan penyesuaian agar aplikasi chatbot dapat berfungsi dengan optimal.

2. Studi Literatur: Studi literatur adalah metode pengumpulan data dengan cara menelaah berbagai sumber tertulis yang relevan dengan topik penelitian. Sumber-sumber ini bisa berupa buku, jurnal ilmiah, artikel, laporan penelitian sebelumnya, dan dokumen resmi lainnya. Studi literatur bertujuan untuk mengetahui perkembangan terbaru dalam teknologi chatbot dan NLP, memahami konsep dasar dan teori yang berkaitan dengan NLP dan chatbot, serta mempelajari kasus-kasus penerapan chatbot di institusi lain.

Kedua metode pengumpulan data ini saling melengkapi dan memberikan pemahaman yang komprehensif mengenai kebutuhan, permasalahan, dan solusi yang tepat dalam perancangan aplikasi chatbot di FIKTI UMSU. Dengan demikian, diharapkan chatbot yang dikembangkan dapat memberikan layanan yang optimal dan memenuhi harapan pengguna.

3.2 Tempat Penelitian

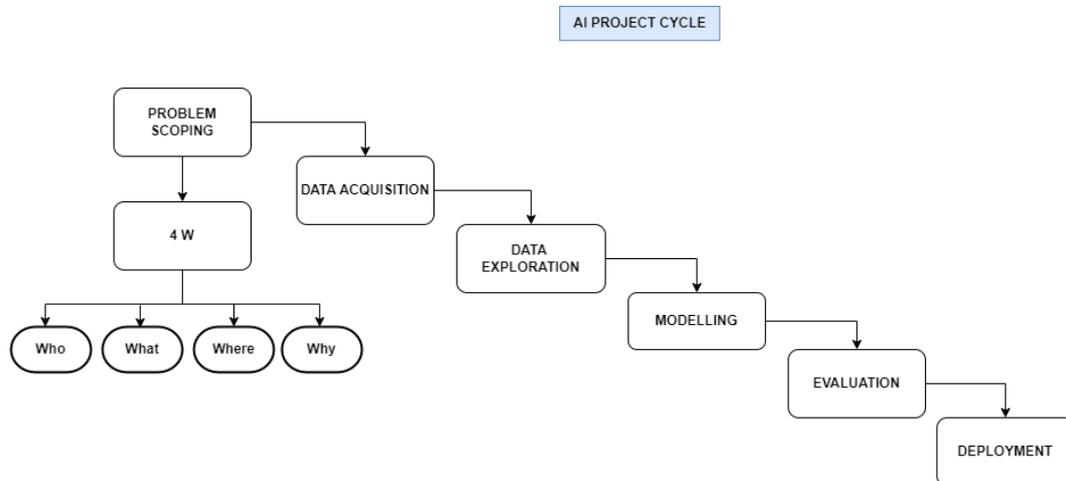
Penelitian ini akan dilakukan di Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Muhammadiyah Sumatera Utara (FIKTI UMSU). Kampus ini dipilih sebagai lokasi penelitian karena merupakan tempat di mana peneliti menempuh pendidikan dan memiliki akses langsung terhadap lingkungan akademik dan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

3.3 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 (Azimah & Wardani,

2022). 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 3. 1 AI Project Cycle

3.3.1 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 (Azimah & Wardani, 2022), 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.

3.3.2 Data Acquisition

Proses *data acquisition* merupakan langkah awal yang penting dalam pembuatan *chatbot* dimana penulis mengumpulkan data yang akan digunakan untuk melatih dan mengembangkan model *chatbot*. Data yang dikumpulkan adalah data mengenai layanan akademik dan administratif. Sumber pengumpulan data layanan

akademik dan administratif diambil dari hasil wawancara kepada pihak terkait di FIKTI UMSU yaitu staf administrasi. Semua pengumpulan data dilakukan secara manual. Setelah data diperoleh, selanjutnya akan dilakukan proses pembuatan dataset dengan format *file json*. Dataset tersebut nantinya akan berisi pertanyaan dan respon yang akan digunakan dalam pelatihan model.

```

1  {"intents": [
2    {
3      "tag": "greeting",
4      "patterns": [
10         "oi",
11         "yo",
12         "yow",
13         "ehem",
14         "Apa kabar",
15         "halo",
16         "hai",
17         "hai kak",
18         "woi",
19         "wee",
20         "hello",
21         "hi",
22         "hy",
23         "hy admin",
24         "hey",
25         "hei",
26         "pagi",
27         "siang",
28         "sore",
29         "malam",
30         "pnten",
31         "hallo",
32         "Assalamualaikum",
33         "Permisi",
34         "Mici",
35         "Ada orang?"
36       ],
37       "responses": [
38         "Halo FIKTIBOT di sini, ada yang bisa saya bantu?"

```

Gambar 3. 2 Tampilan Dataset

Gambar 3.2 merupakan bentuk dari contoh dataset json yang terdiri dari *tag*, *patterns*, dan *response*. *Tag* mengacu pada label yang digunakan untuk mengidentifikasi maksud tertentu dalam dataset. *Patterns* adalah daftar contoh pesan atau pertanyaan yang terkait dengan label. Sedangkan *responses* (respon) adalah balasan atau tanggapan yang sesuai dengan label tertentu.

3.3.3 Data Exploration

Dalam tahap eksplorasi data atau pemrosesan data, penulis melakukan serangkaian langkah untuk mempersiapkan data sebelum melatih model.

a. Mengubah dataset json menjadi dataframe

Library pembelajaran mesin, seperti *scikit-learn*, *TensorFlow*, dan *Keras*, bekerja dengan baik dengan data dalam format *DataFrame*, sehingga konversi ini memfasilitasi tahap berikutnya dalam pembelajaran mesin.

b. Menghilangkan Punctuasi

kode tersebut 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.

c. *Lemmatization* (Lematisasi)

Selama proses ini, kata-kata direduksi menjadi bentuk dasarnya dengan tetap mempertimbangkan konteksnya. Proses ini bertujuan untuk meningkatkan akurasi analisis dengan mengurangi variasi kata-kata yang sebenarnya memiliki arti yang sama.

d. Menyortir data kelas *tags*.

Fungsi dari proses ini adalah untuk menyiapkan daftar tag atau label yang akan digunakan dalam proses klasifikasi, serta untuk memastikan bahwa tidak ada duplikat yang dapat mengganggu analisis atau pelatihan model.

e. Mencari jumlah keseluruhan data teks

Tujuan dari penghitungan data teks adalah untuk mengetahui skala dataset yang akan digunakan.

f. Tokenisasi data:

Pada proses ini, teks akan dipisah menjadi bagian-bagian yang lebih kecil, yang disebut token. Dalam konteks NLP, token bisa berupa kata, karakter, atau sub-kata. Tujuan dari tokenisasi adalah untuk memudahkan analisis lebih lanjut dengan

mengkonversi teks panjang menjadi unit yang lebih kecil dan dapat diolah oleh algoritma pembelajaran mesin atau NLP.

g. Melakukan *padding*:

Pada proses ini nilai tertentu (biasanya 0) ditambahkan ke data untuk mencapai panjang yang seragam di semua sampel data. Proses ini dilakukan agar semua input data memiliki panjang yang sama, yang diperlukan oleh banyak model pembelajaran mesin, terutama yang berbasis jaringan saraf tiruan, untuk menjaga konsistensi dimensi input.

h. *Encoding label* atau *tag*:

Pada proses ini teks diubah menjadi format yang dapat dipahami dan diproses oleh komputer, seperti mengubah kata-kata menjadi vektor numerik. *Encoding* memungkinkan komputer untuk melakukan operasi matematika pada teks, yang penting dalam tugas-tugas seperti klasifikasi teks dan pemrosesan bahasa alami.

i. *Input length* dan *Output length*

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.

j. Menghitung *vocabulary size*

Dalam konteks *chatbot*, proses ini akan mencakup setiap kata unik yang teridentifikasi dalam pertanyaan dan jawaban yang digunakan untuk melatih *chatbot*.

k. Menyimpan model *word & Classes*

Teks yang sudah diproses selanjutnya akan disimpan untuk mempertahankan informasi model yang sudah dilatih. Menyimpan model memungkinkan penggunaan kembali model yang sama tanpa perlu melatih ulang dari awal.

3.3.4 Modeling

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*) seperti yang terlihat pada gambar 3.2.

```
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 3. 3 Arsitektur *Neural Network*

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

3.3.5 Evaluating

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. Penulis melakukan uji coba *Black Box* dengan mengajukan 20 pertanyaan secara acak kepada *chatbot*. Dalam kasus ini, jika chatbot dapat menjawab semua 20 pertanyaan dengan benar, maka relevansi setiap jawaban adalah 1 (relevan) karena semuanya benar. Penulis 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:

$$\text{MAP} = \frac{\text{Total AP}}{\text{Total Pertanyaan}} \times 100\% \dots \dots \dots (3.3.5)$$

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

$$\text{AP} = \frac{\text{Total jawaban relevan yang ditemukan}}{\text{Total jawaban yang diambil}} \dots \dots \dots (3.3.5)$$

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:

$$\text{MAP} = \frac{20}{20} \times 100\% = 100\% \dots \dots \dots (3.3.5)$$

3.3.6 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*.

BAB IV 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 :

4.1.1 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.

4.1.2 Data Acquisition

Sumber pengumpulan data layanan akademik dan administratif diambil langsung dari website FIKTI UMSU yang menyediakan tentang panduan-panduan layanan akademik dan juga pengumpulan data dilakukan dengan wawancara kepada staff biro akademik dan administrasi pertanyaan untuk wawancara di data di google form, lalu staff diminta untuk menjawab pertanyaan yang ada di google form tersebut. Sumber pengumpulan data berupa pola pertanyaan/percakapan diperoleh dari forum obrolan sosial, ChatGPT, dan hasil pemikiran sendiri. Semua pengumpulan data dilakukan dengan secara manual. Setelah data diperoleh, selanjutnya akan dilakukan proses pembuatan dataset dengan format file json dengan jumlah 83 tag 2,472 patterns dan 82 response yang nantinya akan berisi

pertanyaan dan respon yang akan digunakan dalam pelatihan model. Bentuk dari dataset json terdiri dari tag, patterns, dan response.

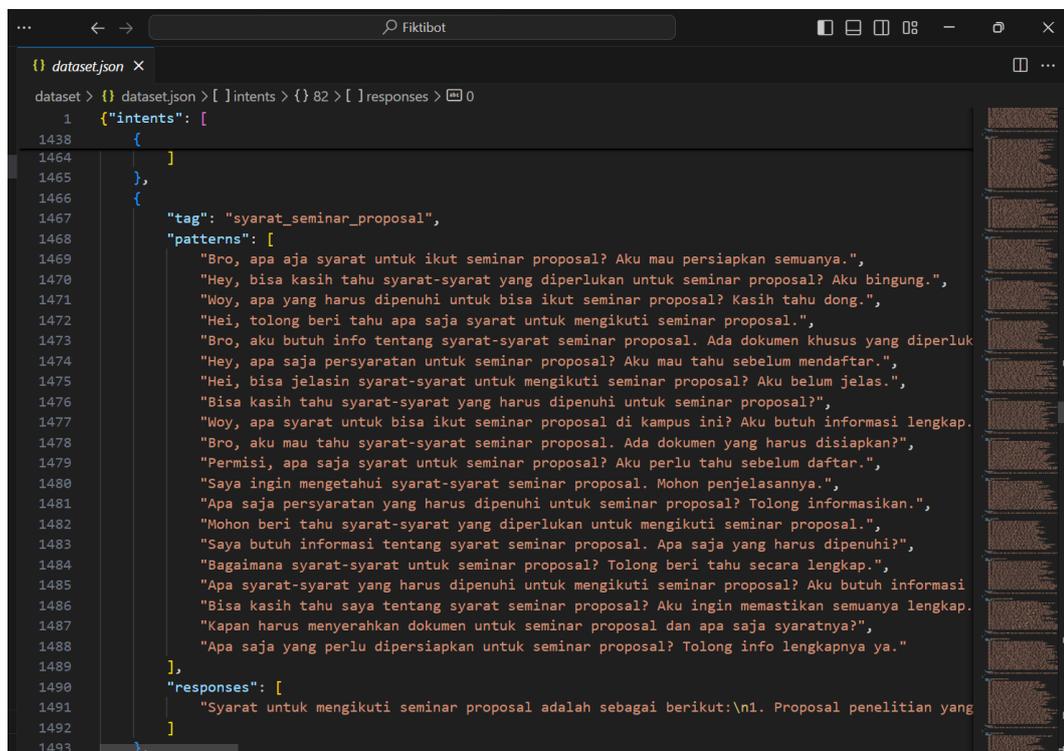
```

1  {"intents": [
2    {
3      "tag": "salam",
4      "patterns": [
5        "p",
6        "ping",
7        "permisi",
8        "we",
9        "what's up",

```

Gambar 4. 1 File json

Seperti yang terlihat pada gambar 4.1, pembuatan dataset json dimulai dengan membuat perintah “intents” terlebih dahulu. Intents memuat informasi tentang berbagai tujuan atau niat (intents) dari percakapan yang ingin dipelajari oleh chatbot.



```

dataset > {} dataset.json > [ ] intents > {} 82 > [ ] responses > 0
1  {"intents": [
1438  {
1464  }
1465  },
1466  {
1467    "tag": "syarat_seminar_proposal",
1468    "patterns": [
1469      "Bro, apa aja syarat untuk ikut seminar proposal? Aku mau menyiapkan semuanya.",
1470      "Hey, bisa kasih tahu syarat-syarat yang diperlukan untuk seminar proposal? Aku bingung.",
1471      "Woy, apa yang harus dipenuhi untuk bisa ikut seminar proposal? Kasih tahu dong.",
1472      "Hei, tolong beri tahu apa saja syarat untuk mengikuti seminar proposal.",
1473      "Bro, aku butuh info tentang syarat-syarat seminar proposal. Ada dokumen khusus yang diperluk",
1474      "Hey, apa saja persyaratan untuk seminar proposal? Aku mau tahu sebelum mendaftar.",
1475      "Hei, bisa jelasin syarat-syarat untuk mengikuti seminar proposal? Aku belum jelas.",
1476      "Bisa kasih tahu syarat-syarat yang harus dipenuhi untuk seminar proposal?",
1477      "Woy, apa syarat untuk bisa ikut seminar proposal di kampus ini? Aku butuh informasi lengkap.",
1478      "Bro, aku mau tahu syarat-syarat seminar proposal. Ada dokumen yang harus disiapkan?",
1479      "Permisi, apa saja syarat untuk seminar proposal? Aku perlu tahu sebelum daftar.",
1480      "Saya ingin mengetahui syarat-syarat seminar proposal. Mohon penjelasannya.",
1481      "Apa saja persyaratan yang harus dipenuhi untuk seminar proposal? Tolong informasikan.",
1482      "Mohon beri tahu syarat-syarat yang diperlukan untuk mengikuti seminar proposal.",
1483      "Saya butuh informasi tentang syarat seminar proposal. Apa saja yang harus dipenuhi?",
1484      "Bagaimana syarat-syarat untuk seminar proposal? Tolong beri tahu secara lengkap.",
1485      "Apa syarat-syarat yang harus dipenuhi untuk mengikuti seminar proposal? Aku butuh informasi",
1486      "Bisa kasih tahu saya tentang syarat seminar proposal? Aku ingin memastikan semuanya lengkap.",
1487      "Kapan harus menyerahkan dokumen untuk seminar proposal dan apa saja syaratnya?",
1488      "Apa saja yang perlu dipersiapkan untuk seminar proposal? Tolong info lengkapnya ya."
1489    ],
1490    "responses": [
1491      "Syarat untuk mengikuti seminar proposal adalah sebagai berikut:\n1. Proposal penelitian yang
1492    ]
1493  },

```

Gambar 4. 2 Dataset json tag.patterns & response

Pada gambar 4.2 yaitu setelah intents dibuat, langkah selanjutnya adalah membuat "tag", "patterns", dan "response". Proses ini diulang dengan berbagai konteks untuk meningkatkan variasi dan ukuran dataset JSON. Hal ini bertujuan untuk memperkaya variasi percakapan yang dapat dipelajari oleh chatbot sehingga dapat merespons dengan lebih luas dan tepat dalam berbagai situasi.

4.1.3 Data Exploration

Dalam tahap eksplorasi data, peneliti melakukan beberapa langkah untuk mempersiapkan data sebelum melatih model.

a. Mengubah dataset json menjadi dataframe

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

Gambar 4. 3 Mengubah dataset json menjadi dataframe

Dapat dilihat pada gambar 4.3 bahwa konversi ini memfasilitasi tahap berikutnya dalam pembelajaran mesin.

b. Menghilangkan Punctuasi

```
#Menghilangkan Punctuasi
msbot_dataset['patterns'] = msbot_dataset['patterns'].apply(lambda wrd:[ltrs.lower() for ltrs in wrd if ltrs not in string.punctuation])
msbot_dataset['patterns'] = msbot_dataset['patterns'].apply(lambda wrd: ''.join(wrd))
```

Gambar 4. 4 Menghilangkan Punctuasi

Dapat dilihat pada gambar 4.4 dimana kode tersebut 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.

c. Lemmatization (Lematisasi)

```
#Lematisasi
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. 5 *Lemmatization* (Lematisasi)

Pada gambar 4.5 merupakan kode yang berfungsi mereduksi kata-kata menjadi bentuk dasarnya dengan tetap mempertimbangkan konteksnya.

d. 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 4. 6 Menyortir data kelas tags

Pada gambar 4.6 merupakan proses klasifikasi untuk memastikan bahwa tidak ada duplikat yang dapat mengganggu analisis atau pelatihan model.

e. Mencari jumlah keseluruhan data teks

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

Gambar 4. 7 Mencari jumlah keseluruhan data teks

Dapat dilihat pada gambar 4.7 penghitungan data teks adalah untuk mengetahui skala dataset yang akan digunakan.

f. 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 4. 8 Tokenisasi data

Pada gambar 4.8 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.

g. Melakukan padding

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

Gambar 4. 9 Melakukan padding

Dapat dilihat pada gambar 4.9 bahwa proses ini nilai tertentu (biasanya 0) ditambahkan ke data untuk mencapai panjang yang seragam di semua sampel data.

h. Encoding label atau tag

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

Gambar 4. 10 Melakukan padding

Dapat dilihat pada gambar 8.10 dimana proses ini teks diubah menjadi format yang dapat dipahami dan diproses oleh komputer, seperti mengubah kata-kata menjadi vektor numerik.

i. 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 4. 11 *Input length dan Output length*

Dapat dilihat pada gambar 4.11 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.

j. Menghitung vocabulary size

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

Gambar 4. 12 Menghitung vocabulary size

Dapat dilihat pada gambar 4.12, proses ini akan mencakup setiap kata unik yang teridentifikasi dalam pertanyaan dan jawaban yang digunakan untuk melatih *chatbot*.

k. Simpan Model

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

Gambar 4. 13 Simpan Model

Dapat dilihat pada gambar 4.13 teks yang sudah diproses selanjutnya akan disimpan untuk mempertahankan informasi model yang sudah dilatih.

4.1.4 Modeling

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

```

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 4. 14 Modeling LSTM

Kode program yang Anda berikan 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. Berikut adalah penjelasan lebih lanjut beberapa bagian penting dari dari Modeling LSTM diatas:

a. Fungsi `msbot`

```

def msbot_bot(trainx, trainy, neuron, batch_size, epochs):

```

Gambar 4. 15 Fungsi `msbot`

Pada Gambar 4.15 merupakan fungsi awal untuk mendefinisikan parameter-parameter berikutnya. Berikut ini penjelasan lebih rinci mengenai kode tersebut:

- 1) `trainx`: Data input untuk pelatihan model.
- 2) `trainy`: Label atau target yang sesuai dengan data input.
- 3) `neuron`: Jumlah neuron yang akan digunakan dalam layer LSTM.

- 4) `batch_size`: Jumlah sampel data yang digunakan untuk satu iterasi dalam pelatihan.
- 5) `epochs`: Jumlah epochs (iterasi) untuk melatih model.

b. 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 4. 16 Arsitektur Neural Network

Pada Gambar 4.16 merupakan arsitektur neural network menggunakan TensorFlow/Keras yang bertugas untuk pemrosesan bahasa alami. Berikut ini penjelasan lebih rinci mengenai kode tersebut:

- 1) `tf.keras.Input`: Mendefinisikan layer input dengan bentuk `input_shape`.
- 2) `tf.keras.layers.Embedding`: Layer embedding untuk mengubah input menjadi representasi vektor.
- 3) `tf.keras.layers.LSTM`: Layer LSTM (Long Short-Term Memory) untuk memproses urutan data dalam bentuk sequence.
- 4) `tf.keras.layers.Dropout`: Layer Dropout untuk mengurangi overfitting dengan mengabaikan sebagian unit selama pelatihan.
- 5) `tf.keras.layers.Flatten`: Mengubah output menjadi satu dimensi.
- 6) `tf.keras.layers.Dense`: Layer dense (fully connected) dengan aktivasi softmax sebagai output layer.

c. Compile Model

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

Gambar 4. 17 Compile Model

Pada gambar 4.17 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. Berikut ini penjelasan lebih rinci mengenai kode tersebut:

- 1) `loss='sparse_categorical_crossentropy'`: Fungsi loss untuk pelatihan model.
- 2) `optimizer='adam'`: Algoritma optimisasi yang digunakan untuk menyesuaikan bobot model.
- 3) `metrics=['accuracy']`: Metrik evaluasi yang digunakan untuk mengukur performa model.

d. Pelatihan Model

```
history = model.fit(trainx,trainy,batch_size=batch_size,epochs=epochs,verbose=1,shuffle=False)
```

Gambar 4. 18 Pelatihan Model

Kode pada gambar 4.18 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.

Berikut ini penjelasan lebih rinci mengenai kode tersebut:

- 1) `model.fit`: Melatih model dengan data yang diberikan dengan jumlah batch (`batch_size`) dan epochs (`epochs`) yang telah ditentukan.

e. Output

```
return model, history
```

Gambar 4. 19 Output

Pada Gambar 4.19 merupakan fungsi untuk mengembalikan model yang telah dilatih beserta riwayat pelatihannya.



Gambar 4. 20 Grafik Akurasi LSTM

Pada gambar 4.20 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.

4.1.5 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 4. 1 Uji Blackbox

No	User Input	Chatbot Resnponse	Tag	Status
1	Apa itu Fiktibot?	Perkenalkan nama saya FIKTIBOT. FIKTIBOT adalah sebutan untuk nama asisten chatbot dari FIKTI UMSU yang akan menjadi teman konsultasi untuk menjawab seputar layanan akademik ataupun layanan administratif	About FIKTIBOT	1
2	gimana cara daftar kuliah di semester ini	Untuk mendaftar mata kuliah, kamu perlu masuk ke portal akademik UMSU, pilih menu pendaftaran mata kuliah, dan ikuti langkah-langkah yang	Pendaftaran matakuliah	1

		ada di sana. Pastikan semua data sudah benar sebelum kamu submit.		
3	apa aja syarat daftar mata kuliah di semester ini	Syarat pendaftaran mata kuliah di semester ini meliputi: sudah menyelesaikan seluruh prasyarat mata kuliah sebelumnya, mengisi formulir pendaftaran mata kuliah, dan membayar biaya SKS sesuai ketentuan. Pastikan semua syarat terpenuhi sebelum mendaftar.	Syarat pendaftaran mata kuliah	1
4	gimana cara bayar uang kuliah di semester ini aku butuh info nya	Berikut adalah prosedur pembayaran uang kuliah: 1. Login ke akun mahasiswa Kamu di portal UMSU. 2. Pilih menu pembayaran dan pilih semester yang akan dibayar biasanya ada 4 tahap pembayaran. 3. Pilih metode pembayaran yang diinginkan (transfer bank, kartu kredit, dll.). 4. Ikuti instruksi untuk menyelesaikan pembayaran. 5. Simpan bukti pembayaran dan laporkan ke bagian keuangan jika diperlukan.	Prosedur Pembayaran uang kuliah	1
5	bro gimana cara dapat transkrip nilai, kasih tau dong	Untuk mendapatkan transkrip nilai, ikuti langkah-langkah berikut: 1. Login ke portal mahasiswa menggunakan akun Kamu. 2. Akses menu Transkrip Nilai . 3. Lalu Klik Transkrip Nilai dan Download maka	Cara mendapatkan transkrip nilai	1

		Transkrip Nilai sudah tersimpan di penyimpanan perangkat kamu.		
6	apa aja syarat untuk ikut ujian di semester ini	<p>Untuk mengikuti ujian semester ini, Kamu perlu memenuhi syarat-syarat berikut:</p> <ol style="list-style-type: none"> 1. Menyelesaikan semua tugas dan kuis yang terkait dengan mata kuliah. 2. Memiliki kehadiran yang memenuhi syarat (minimal 75% hadir). 3. Membayar uang semester jika diperlukan. 4. Mengikuti semua instruksi yang diberikan oleh dosen atau pihak kampus. <p>Pastikan semua syarat dipenuhi untuk dapat mengikuti ujian tanpa kendala.</p>	Syarat ujian	1
7	gimana cara daftar kelas tambahan , aku butuh infonya	<p>Berikut adalah langkah-langkah untuk mendaftar kelas tambahan:</p> <ol style="list-style-type: none"> 1. Login ke portal akademik UMSU mahasiswa. 2. Pilih menu 'KRS Online' dari daftar opsi yang tersedia. 3. Cari kelas tambahan yang ingin diikuti dan klik 'Tambahkan'. 4. Verifikasi data dan submit KRS Online. 5. Pastikan sebelum melakukan pengisian kelas tambahan, lakukan pembayaran sesuai dengan instruksi yang diberikan. <p>Pastikan Kamu mengikuti semua</p>	Cara mendaftar kelas tambahan	1

		langkah ini untuk memastikan pendaftaran Kamu berhasil.		
8	apa aja ya syarat untuk mulai skripsi	Sama-sama kak, senang bisa membantu	Syarat skripsi	0
9	apa aja ya syarat untuk memulai skripsi, mohon panduannya	Berikut adalah syarat untuk memulai skripsi: 1. Telah menyelesaikan minimal 120 SKS. 2. Terdaftar dan aktif sebagai mahasiswa pada semester berjalan. 3. Telah lulus mata kuliah metode penelitian. 4. Telah menyelesaikan kewajiban administrasi keuangan.	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:

$$\text{MAP} = \frac{\text{Total AP}}{\text{Total Pertanyaan}} \times 100\% \dots \dots \dots (4.1.5)$$

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

$$AP = \frac{\text{Total jawaban relevan yang ditemukan}}{\text{Total jawaban yang diambil}} \dots\dots(4.1.5)$$

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:

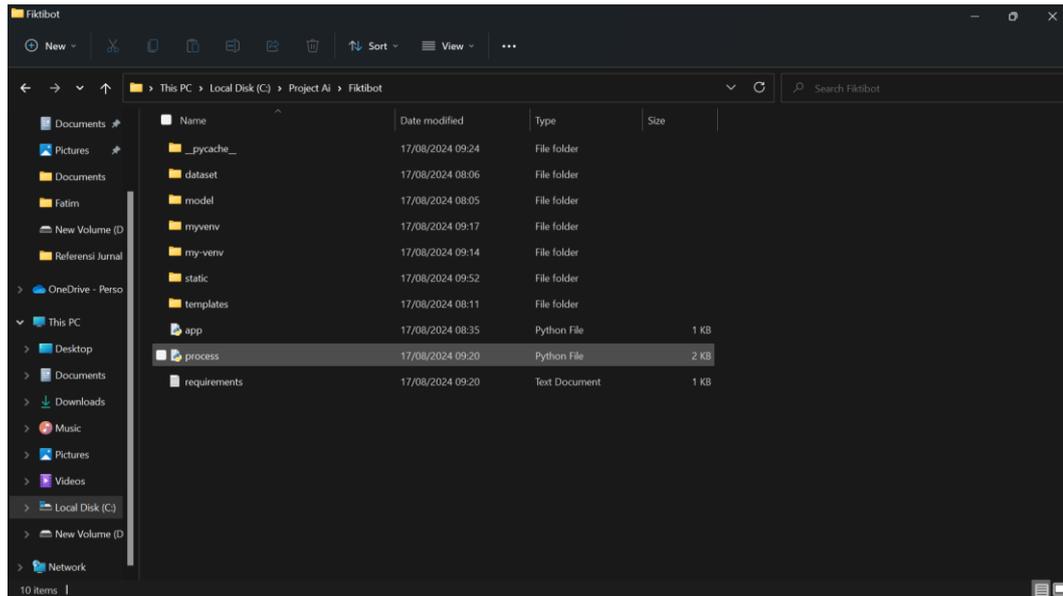
$$MAP = \frac{8}{82} \times 100\% = 97\% \dots\dots(4.1.5)$$

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. Hal ini mengindikasikan bahwa chatbot telah mampu memahami pertanyaan sehingga dapat memberikan respon dengan akurasi yang baik.

4.1.6 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:

- a. Pembuatan folder baru bernama “Fiktibot”



Gambar 4. 21 Folder Fiktibot

b. 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*.

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

```

dataset.json  app.py  X
app.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")
13     response = botResponse(text)
14     message = {"answer": response}
15     return jsonify(message)
16
17 if __name__ == "__main__":
18     app.run(debug=True)

```

Gambar 4. 22 File app.py

```

dataset.json  process.py 1  X
process.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']
23
24 # import model dan download nltk file
25 def preparation():
26     load_response()
27     global lemmatizer, tokenizer, le, model
28     tokenizer = pickle.load(open('model/tokenizers.pkl', 'rb'))
29     le = pickle.load(open('model/le.pkl', 'rb'))
30     model = keras.models.load_model('model/Fiktibot.h5')
31     lemmatizer = WordNetLemmatizer()
32     nltk.download('punkt', quiet=True)

```

Gambar 4. 23 File process.py bagian (1)

```

dataset.json  process.py 1 x
process.py > ...
34     nltk.download('omw-1.4', quiet=True)
35
36     # hapus tanda baca
37     def remove_punctuation(text):
38         texts_p = []
39         text = [letters.lower() for letters in text if letters not in string.punctuation]
40         text = ''.join(text)
41         texts_p.append(text)
42         return texts_p
43
44     # mengubah text menjadi vector
45     def vectorization(texts_p):
46         vector = tokenizer.texts_to_sequences(texts_p)
47         vector = np.array(vector).reshape(-1)
48         vector = pad_sequences([vector], input_shape)
49         return vector
50
51     # klasifikasi pertanyaan user
52     def predict(vector):
53         output = model.predict(vector)
54         output = output.argmax()
55         response_tag = le.inverse_transform([output])[0]
56         return response_tag
57
58     # menghasilkan jawaban berdasarkan pertanyaan user
59     def botResponse(text):
60         texts_p = remove_punctuation(text)
61         vector = vectorization(texts_p)
62         response_tag = predict(vector)
63         answer = random.choice(responses[response_tag])
64         return answer
65

```

Gambar 4. 24 File process.py bagian (2)

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

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

```

dataset.json  requirements.txt x
requirements.txt
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 4. 25 File requirements.txt

File tersebut sangat dibutuhkan agar saat penginstalan library python bisa sesuai dengan virtual environment dimana model tersebut dijalankan.

Setelah pembuatan folder, subfolder, dan file selesai, langkah selanjutnya dalam proses deployment adalah membuat environment. Dalam program ini, environment dibuat menggunakan aplikasi Anaconda, dengan nama environment "FiktiBot". Setelah environment dibuat, buka terminal Anaconda, yaitu Anaconda Prompt.

Ketikkan perintah berikut di terminal Anaconda Prompt:

```
``conda activate Fiktibot
```

```
``
```

Perintah tersebut akan mengaktifkan environment "Fiktibot" yang telah dibuat sebelumnya.

Selanjutnya, ketik perintah berikut di terminal Anaconda Prompt:

```
``pip install -r requirements.txt
```

```
``
```

Perintah tersebut akan menginstall semua requirement atau library yang dibutuhkan oleh aplikasi.

Setelah proses instalasi selesai, ketik perintah berikut di terminal Anaconda Prompt:

```
``python app.py
```

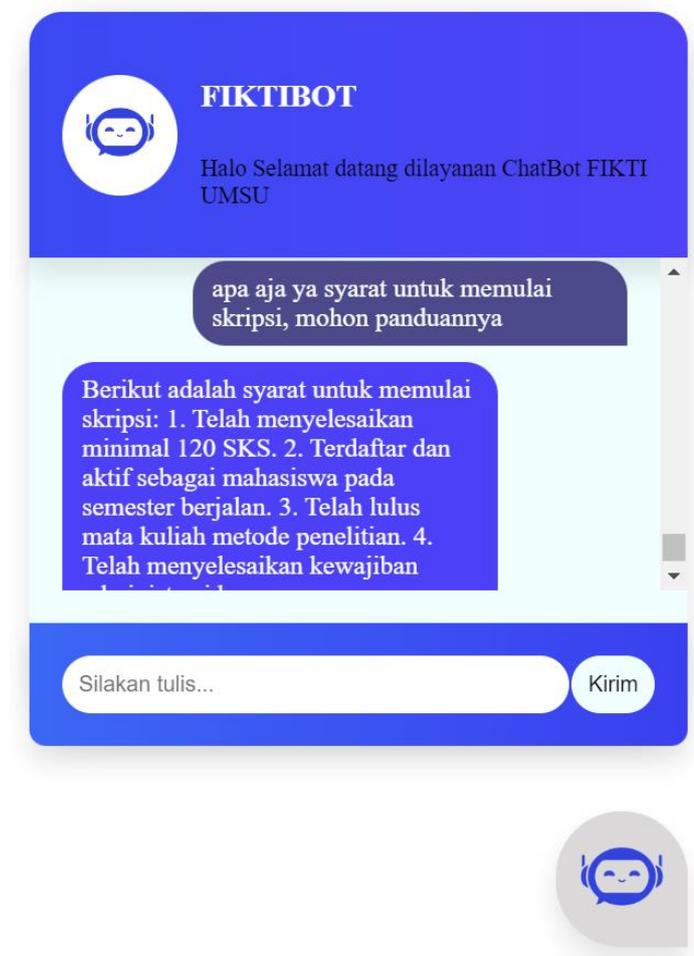
```
``
```

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

maka akan muncul tampilan website yang telah dibuat. Dengan demikian, proses deployment aplikasi atau website telah selesai dan dapat diakses melalui link yang disebutkan.

4.2 Hasil dan Uji Coba

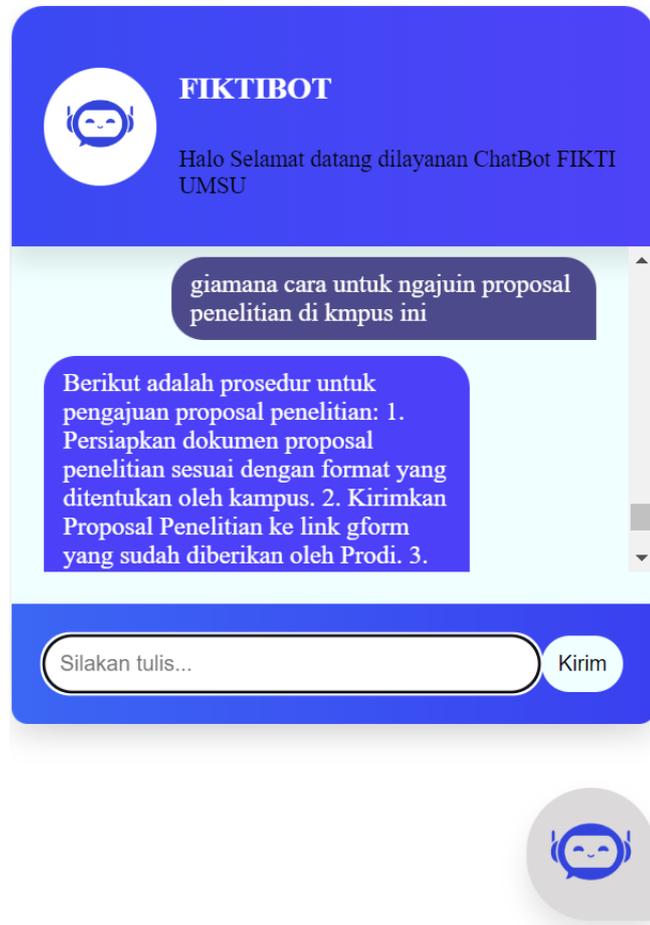
Uji coba akan dilakukan di tampilan hasil *chatbot* FIKTI UMSU yang sudah di deploy secara localhost menggunakan Flask.



Gambar 4. 26 Tampilan hasil *Chatbot* FIKTI UMSU

Pada Gambar 4.26 merupakan implementasi aplikasi web menggunakan Flask, yang merupakan framework web Python. Secara keseluruhan, kode ini adalah kerangka dasar untuk aplikasi web Flask yang memiliki beberapa halaman dan juga

menyediakan layanan chatbot yang merespons permintaan pengguna melalui JSON.



Gambar 4. 27 Uji Coba *Chatbot* FIKTI UMSU setelah di deploy

Pada gambar 4.27 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.

4.3 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:

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

Kekurangan:

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

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan pembahasan dari bab-bab sebelumnya yang telah dilakukan maka dapat diambil beberapa kesimpulan sebagai berikut:

- a. Pentingnya Pengembangan Chatbot untuk Peningkatan Layanan Akademik dan Administratif Chatbot FIKTI UMSU dikembangkan untuk merespons kebutuhan mahasiswa dalam mendapatkan informasi terkait layanan akademik dan administratif. Penggunaan teknologi chatbot berbasis AI di lingkungan pendidikan, khususnya di Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Muhammadiyah Sumatera Utara (FIKTI UMSU), terbukti dapat mengurangi beban kerja staf biro administrasi serta meningkatkan kepuasan mahasiswa.
- b. Implementasi AI Project Cycle dan Natural Language Processing (NLP) pada Pengembangan Chatbot, proses pengembangan chatbot ini mengikuti metode AI Project Cycle dengan pendekatan Natural Language Processing (NLP). Dimulai dari tahap Problem Scoping, Data Acquisition, Data Exploration, hingga Modeling. Data layanan akademik dan administratif yang diperoleh melalui website FIKTI UMSU dan wawancara dengan staf biro administrasi kemudian diolah untuk melatih model AI yang digunakan dalam chatbot.
- c. Keberhasilan Pemodelan dengan Arsitektur LSTM pemodelan menggunakan arsitektur Long Short-Term Memory (LSTM) berhasil menghasilkan chatbot yang mampu memberikan respon dengan akurasi tinggi. Evaluasi melalui uji Blackbox menunjukkan bahwa chatbot mampu memahami dan menjawab

pertanyaan mahasiswa dengan tingkat relevansi yang tinggi, mencapai Mean Average Precision (MAP) sebesar 97%.

- d. Keberhasilan Deployment Chatbot menggunakan Flask Proses deployment chatbot dilakukan menggunakan Flask sebagai framework web untuk menyajikan aplikasi secara lokal. Proses ini berhasil dilakukan dengan mengaktifkan environment yang telah dibuat, menginstal library yang diperlukan, dan menjalankan aplikasi melalui link lokal. Hasil uji coba menunjukkan bahwa chatbot berfungsi dengan baik dan dapat merespons pertanyaan mahasiswa meskipun terdapat kesalahan pengetikan.
- e. Kelebihan dan Kekurangan Chatbot FIKTI UMSU, seperti kemampuan memberikan respon yang cepat dan akurat serta memahami konteks pertanyaan meskipun terdapat kesalahan pengetikan. Namun, chatbot ini juga memiliki beberapa keterbatasan, seperti ketidakmampuan untuk menolak atau menangani pertanyaan yang berada di luar konteks atau pengetahuannya.

5.2 Saran

Pengembangan lebih lanjut pada aplikasi *chatbot* FIKTI UMSU menggunakan metode *AI Project Cycle* dan penerapan *Natural Language Processing* (NLP) ini, maka dapat diberikan beberapa saran sebagai berikut:

1. Peningkatan Pengetahuan dan Kepahaman: Menambahkan lebih banyak data, pola percakapan, dan respons yang dapat dikelola oleh chatbot untuk meningkatkan pengetahuan dan pemahaman konteks.
2. Penambahan Fungsionalitas: Mengembangkan atau menambahkan fungsionalitas chatbot agar dapat melakukan lebih banyak tugas atau

memberikan layanan yang lebih kompleks. Ini bisa termasuk integrasi dengan sistem backend atau basis data layanan yang lebih luas. Selain itu, integrasi dengan platform e-learning dapat membantu mahasiswa dalam mengakses materi kuliah dan tugas secara langsung melalui chatbot.

3. Integrasi dengan Platform Lain: Untuk menjangkau lebih banyak pengguna, disarankan agar chatbot ini dikembangkan untuk dapat diintegrasikan dengan berbagai platform media sosial yang umum digunakan oleh mahasiswa, seperti WhatsApp, Telegram, dan Facebook Messenger. Hal ini akan memudahkan mahasiswa dalam mengakses informasi kapan saja dan di mana saja.
4. Pengembangan Kemampuan Bahasa: Memungkinkan chatbot untuk memahami dan merespons dalam bahasa asing atau dialek tertentu, atau bahkan menggunakan model multibahasa.
5. Mengatasi Kekurangan: Mengidentifikasi dan mengatasi kekurangan chatbot, seperti kemampuan untuk menolak pertanyaan di luar konteks, agar pengalaman pengguna menjadi lebih baik.
6. Monitoring dan Evaluasi Berkala : Disarankan untuk melakukan monitoring dan evaluasi berkala terhadap kinerja chatbot guna memastikan bahwa chatbot selalu memberikan layanan terbaik kepada pengguna. Pemantauan ini dapat mencakup analisis terhadap pola interaksi, tingkat keberhasilan dalam menjawab pertanyaan, serta identifikasi area yang memerlukan perbaikan.

DAFTAR PUSTAKA

- Ahmadi, M. (2023). Komunikasi Simbolik: Implikasi Penggunaan Chatbot Sebagai Upaya Peningkatan Efektivitas Pelayanan Customer Service. *J-KIs: Jurnal Komunikasi Islam*, 4(1), 101–110.
- Akbar Nasution, M., Fitri, A., Sabila Rizwine, K., Styphen Silaban, V., Khoirani, F., Iskandar Ps, W. v, Baru, K., & Percut Sei Tuan, K. (2024). *Implementasi NLP Dalam Pembuatan Chatbot Customer Service Publisher Jurnal Studi Kasus LARISMA*. 1(1), 13–17.
- Apriliani, D., Febbi Handayani, S., Triadi Saputra, I., Informatika, T., & Harapan Bersama, P. (2023). *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* (Vol. 22, Issue 4).
- Azimah, F., Rizky, K., & Wardani, N. (2022). *SISTEM PENDETEKSI GEJALA AWAL COVID-19 DENGAN PENGGUNAAN METODE AI PROJECT CYCLE*.
- Elcholiqi, A., & Musdholifah, A. (2020). Chatbot in Bahasa Indonesia using NLP to Provide Banking Information. *IJCCS (Indonesian Journal of Computing and Cybernetics Systems)*, 14(1), 91.
- Fitra Ramadhan, F. (2021). Chatbot pada E-Commerce berbasis Android dengan Pendekatan Natural Language Processing. *JCSE Journal of Computer Science an Engineering*, 2(1), 27–39.
- Guntoro, G., Loneli Costaner, & Lisnawita, L. (2020). Aplikasi Chatbot untuk Layanan Informasi dan Akademik Kampus Berbasis Artificial Intelligence Markup Language (AIML). *Digital Zone: Jurnal Teknologi Informasi Dan Komunikasi*, 11(2), 291–300.
- Isma, A., Sahalik Rahman, S., Setiawan Syam, A., & Sari, N. (2023). *Fundamental and Applied Management Journal Analisis Penggunaan Chatbot Berbasis AI pada Model Hybrid di Jurusan Teknik Informatika dan Komputer*. 1(2).
- Kangharnando, W., & Adiya, H. (2022). Penerapan Teknologi Chatbot Pada Website Mobile E-Commerce Roti Papa. *Jurnal Mahasiswa Aplikasi Teknologi Komputer Dan Informasi*, 4, 1–7.
- Kusnanda, G. Ryoga., Susila, A. A. N. H., & Sukarsa, M. (2022). Perancangan Chatbot Hotel dengan Model Natural Language Processing Chatbot dan Button Based. *Jurnal Ilmiah Teknologi Dan Komputer*, 3(1).
- Prasetya, M. A., Wulandari, M., & Nikmah, S. A. (2024). Implementasi NLP(Natural Language Processing) Dasar pada Analisis Sentiment Review Spotify. In *Universitas Nusantara PGRI Kediri* (Vol. 3).

- R. Bangun, C. Informasi, and L. Pekerjaan, "Rancang Bangun Chatbot Informasi Lowongan Pekerjaan Berbasis Whatsapp dengan Metode NLP (Natural Language Processing)," vol. 5, no. 1, pp. 619–626, 2020.
- Sihombing, D. O. (2022). Implementasi Natural Language Processing (NLP) dan Algoritma Cosine Similarity dalam Penilaian Ujian Esai Otomatis. *Jurnal Sistem Komputer Dan Informatika (JSON)*, 4(2), 396.
- Zahwa, A. F., Fiati, R., & Murti, A. C. (2023). Implementasi Chatbot untuk Customer Service menggunakan Metode Natural Language Processing (NLP) (Studi Kasus Website Theme62.com). *J I M P - Jurnal Informatika Merdeka Pasuruan*, 7(2), 82.

KODE PROGRAM

a. file app.py

```
from flask import Flask, render_template, request, jsonify
from process import preparation, botResponse
preparation()
app = Flask(__name__)

@app.route("/")
def home():
    return render_template("index.html")

@app.route("/predict", methods=["GET", "POST"])
def predict():
    text = request.get_json().get("message")
    response = botResponse(text)
    message = {"answer": response}
    return jsonify(message)

if __name__ == "__main__":
    app.run(debug=True)
```

b. file process.py

```
import json
import random
import nltk
import string
import numpy as np
import pickle
import tensorflow as tf

from nltk.stem import WordNetLemmatizer
from tensorflow import keras
from tensorflow.keras.preprocessing.sequence import pad_sequences

global responses, lemmatizer, tokenizer, le, model, input_shape
input_shape = 23
```

```

# import dataset answer
def load_response():
    global responses
    responses = {}
    with open('dataset/dataset.json' , 'r', encoding='utf-8') as content:
        data = json.load(content)
    for intent in data['intents']:
        responses[intent['tag']] = intent['responses']

# import model dan download nltk file
def preparation():
    load_response()
    global lemmatizer, tokenizer, le, model
    tokenizer = pickle.load(open('model/tokenizers.pkl', 'rb'))
    le = pickle.load(open('model/le.pkl', 'rb'))
    model = keras.models.load_model('model/Fiktibot.h5')
    lemmatizer = WordNetLemmatizer()
    nltk.download('punkt', quiet=True)
    nltk.download('wordnet', quiet=True)
    nltk.download('omw-1.4', quiet=True)

# hapus tanda baca
def remove_punctuation(text):
    texts_p = []
    text = [letters.lower() for letters in text if letters not in string.punctuation]
    text = ".join(text)
    texts_p.append(text)
    return texts_p

# mengubah text menjadi vector
def vectorization(texts_p):
    vector = tokenizer.texts_to_sequences(texts_p)

```

```

vector = np.array(vector).reshape(-1)
vector = pad_sequences([vector], input_shape)
return vector

# klasifikasi pertanyaan user
def predict(vector):
    output = model.predict(vector)
    output = output.argmax()
    response_tag = le.inverse_transform([output])[0]
    return response_tag

# menghasilkan jawaban berdasarkan pertanyaan user
def botResponse(text):
    texts_p = remove_punctuation(text)
    vector = vectorization(texts_p)
    response_tag = predict(vector)
    answer = random.choice(responses[response_tag])
    return answer

```

c. File index.html pop-up chatbot

```

<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="utf-8">
    <meta content="width=device-width, initial-scale=1.0" name="viewport">

    <title>FIKTIBOT</title>
    <meta content="" name="description">
    <meta content="" name="keywords">

    <link href="static/style/chatbot.css" rel="stylesheet">

```

```

</head>
<body>

<!-- Chatbot Mulai-->

<div class="container">
  <div class="chatbox">
    <div class="chatbox__support">
      <div class="chatbox__header">
        <div class="chatbox__image--header">
          
        </div>
        <div class="chatbox__content--header">
          <h4 class="chatbox__heading--header">FIKTIBOT</h4>
          <p class="chatbox__description--header">Halo Selamat datang
dilayanan ChatBot FIKTI UMSU</p>
        </div>
      </div>
    </div>
    <div class="chatbox__messages">
      <div></div>
    </div>
    <div class="chatbox__footer">
      <input type="text" placeholder="Silakan tulis...">
      <button class="chatbox__send--footer send__button">Kirim</button>
    </div>
  </div>
  <div class="chatbox__button">
    <button></button>
  </div>
</div>

```

```

</div>

<!-- Chatbot Selesai -->
<script src="static/js/app.js"></script>
</body>

</html>

```

d. File chatbot.css

```

/* CHATBOX
===== */
.chatbox {
    position: fixed;
    bottom: 10px;
    right: 10px;
    z-index: 1;
}

/* CONTENT IS CLOSE */
.chatbox__support {
    display: flex ;
    flex-direction: column;
    background: rgb(240, 255, 255);
    height: 450px;
    width: 400px;
    /* z-index: -1 */
    opacity: 0;
    box-shadow: 0px 0px 15px rgba(0, 0, 0, 0.1);
    border-top-left-radius: 40px;
    border-top-right-radius: 40px;
    border-bottom-left-radius: 40px;
    border-bottom-right-radius: 40px;
}

```

```
    transition: all .5s ease-in-out;
}
```

```
/* CONTENT ISOPEN */
.chatbox--active {
    transform: translateY(-40px);
    z-index: 123456;
    opacity: 1;
}
```

```
/* BUTTON */
.chatbox__button {
    text-align: right;
}
```

```
.send__button {
    padding: 6px;
    /* background: transparent; */
    background-color: #4727f7;
    border: none;
    outline: none;
    cursor: pointer;
}
```

```
/* MESSAGES */
.chatbox__messages {
    margin-top: auto;
    display: static;
    overflow-y:auto;
```

```
    flex-direction: column-reverse;
}

.messages__item {
    max-width: 60.6%;
    width: fit-content;
}

.messages__item--operator {
    margin-left: auto;
}

.messages__item--visitor {
    margin-right: auto;
}

/* FOOTER */
.chatbox__footer {
    position: static;
    bottom: 0;
}

/* HEADER */
.chatbox__header {
    background: linear-gradient(to right, #3a4af3, #4f43f7);
    display: flex;
    flex-direction: row;
    align-items: center;
    justify-content: center;
    padding: 15px 20px;
    border-top-left-radius: 20px;
    border-top-right-radius: 20px;
}
```

```
    box-shadow: 0px 10px 15px rgba(0, 0, 0, 0.1);
}
```

```
.chatbox__image--header {
    margin-right: 14px;
    padding: 10px;
    line-height: 100px 100px;
    border-radius: 100%;
    background-color: white;
}
```

```
.chatbox__heading--header {
    font-size: 1.2rem;
    color: rgb(255, 255, 255);
}
```

```
.chatbox__description--header {
    font-size: .9rem;
    color: rgb(0, 0, 0);
}
```

```
/* Messages */
```

```
.chatbox__messages {
    padding: 0 20px;
}
```

```
.messages__item {
    margin-top: 10px;
    background: #6b9e9a;
    color: white;
    padding: 8px 12px;
```

```
    max-width: 70%;
}

.messages__item--visitor,
.messages__item--typing {
    border-top-left-radius: 20px;
    border-top-right-radius: 20px;
    border-bottom-right-radius: 20px;
    background: #4c40f9;;
    color: white;
}

.messages__item--operator {
    border-top-left-radius: 20px;
    border-top-right-radius: 20px;
    border-bottom-left-radius: 20px;
    background: #4d4a8b;;
    color: white;
}

/* FOOTER */
.chatbox__footer {
    display: flex;
    flex-direction: row;
    align-items: center;
    justify-content: space-between;
    padding: 20px 20px;
    background: linear-gradient(to right, #3c67f4, #3a40f0);
    box-shadow: 0px 10px 15px rgba(0, 0, 0, 0.1);
    border-bottom-right-radius: 10px;
    border-bottom-left-radius: 10px;
    margin-top: 20px;
```

```
}

.chatbox__footer input {
  width: 80%;
  border: none;
  padding: 10px 10px;
  border-radius: 30px;
  text-align: left;
}

.chatbox__send--footer {
  color: rgb(0, 0, 0);
  background-color: azure;
  border: none;
  padding: 10px 10px;
  border-radius: 30px;
}

.chatbox__button button,
.chatbox__button button:focus,
.chatbox__button button:visited {
  padding: 10px 10px;
  background: rgb(219, 217, 217);
  border: none;
  outline: none;
  border-top-left-radius: 50px;
  border-top-right-radius: 50px;
  border-bottom-left-radius: 50px;
  box-shadow: 0px 10px 15px rgba(0, 0, 0, 0.1);
  cursor: pointer;
}
```

```
.chatbox__button button:hover{
  background: #1b08ef;
}
```

d. File app.js

```
class Chatbox {
  constructor() {
    this.args = {
      openButton: document.querySelector('.chatbox__button'),
      chatBox: document.querySelector('.chatbox__support'),
      sendButton: document.querySelector('.send__button')
    }

    this.state = false;
    this.messages = [];
  }

  display() {
    const { openButton, chatBox, sendButton } = this.args;

    openButton.addEventListener('click', () => this.toggleState(chatBox))

    sendButton.addEventListener('click', () => this.onSendButton(chatBox))

    const node = chatBox.querySelector('input');
    node.addEventListener("keyup", ({key}) => {
      if (key === "Enter") {
        this.onSendButton(chatBox)
      }
    })
  }
}
```

```
toggleState(chatbox) {
  this.state = !this.state;

  // show or hides the box
  if(this.state) {
    chatbox.classList.add('chatbox--active')
  } else {
    chatbox.classList.remove('chatbox--active')
  }
}

onSendButton(chatbox) {
  var textField = chatbox.querySelector('input');
  let text1 = textField.value
  if (text1 === "") {
    return;
  }

  let msg1 = { name: "User", message: text1 }
  this.messages=[];
  this.messages.push(msg1);

  fetch('/predict', {
    method: 'POST',
    body: JSON.stringify({ message: text1 }),
    mode: 'cors',
    headers: {
      'Content-Type': 'application/json'
    },
  })
  .then(r => r.json())
```

```

.then(r => {
  let msg2 = { name: "SuskaBot", message: r.answer };
  // this.messages=[];
  this.messages.push(msg2);
  this.updateChatText(chatbox)
  textField.value = ""

}).catch((error) => {
  console.error('Error:', error);
  this.updateChatText(chatbox)
  textField.value = ""
});
}

updateChatText(chatbox) {
  var html = "";
  //console.log(this.messages)
  this.messages.slice().forEach(function(item, index) {
    // console.log(item.name)
    if (item.name === "SuskaBot")
    {
      html += '<div class="messages__item messages__item--visitor">' +
item.message + '</div>'
    }
    else
    {
      html += '<div class="messages__item messages__item--operator">' +
item.message + '</div>'
    }
  });

  const chatmessage = chatbox.querySelector('.chatbox__messages');

```

```
//chatmessage.innerHTML = html;  
chatmessage.insertAdjacentHTML('beforeend', html);
```

```
    }  
}
```

```
const chatbox = new Chatbox();  
chatbox.display();
```

LIST DATASET FIKTIBOT

List Dataset FIKTIBOT

1. Tag: Pendaftaran_Mata_Kuliah
2. Tag: Syarat_Pendaftaran
3. Tag: Jadwal_Pendaftaran
4. Tag: Prosedur_Pembayaran
5. Tag: Biaya_Uang_Kuliah
6. Tag: Prosedur_Cuti_Akademik
7. Tag: Batas_Waktu_Cuti_Akademik
8. Tag: Cara_Mendapatkan_Transkrip
9. Tag: Biaya_Transkrip_Nilai
10. Tag: Jadwal_Ujian
11. Tag: Syarat_Ujian
12. Tag: Cara_Mendaftar_Untuk_Kelas_Tambahan
13. Tag: Melihat_Hasil_Ujian
14. Tag: Kesalahan_Nilai
15. Tag: Pendaftaran_Wisuda
16. Tag: Syarat_Wisuda
17. Tag: Perpanjangan_Waktu_Studi
18. Tag: Dosen_Pembimbing_Skripsi
19. Tag: Syarat_Skripsi
20. Tag: Template_Laporan_Skripsi
21. Tag: Pengajuan_Proposal_Penelitian
22. Tag: Batas_Waktu_Proposal_Penelitian
23. Tag: Prosedur_Sidang_Skripsi
24. Tag: Jadwal_Sidang_Skripsi
25. Tag: Jadwal_Kuliah_Semester_Depan
26. Tag: Syarat_Mata_Kuliah_Pilihan
27. Tag: Pengurusan_KP
28. Tag: Jadwal_KP
29. Tag: Perpindahan_Kelas
30. Tag: Komplain_Nilai
31. Tag: Izin_Penelitian
32. Tag: Pindah_Kampus
33. Tag: Syarat_Pindah_Kampus
34. Tag: Informasi_Seminar
35. Tag: Syarat_Seminar
36. Tag: Pengajuan_Pendaftaran_KKN
37. Tag: Pengurusan_Surat_Izin_KKN
38. Tag: Syarat_KKN
39. Tag: Syarat_Konversi_Nilai
40. Tag: Syarat_Mengikuti_Kegiatan_MBKM
41. Tag: Mendapatkan_SK_Pembimbing

42. Tag: Tidak_Melaksanakan_Ujian
43. Tag: Telat_Mengisi_KRS
44. Tag: Metode_Pembayaran_Yang_Tersedia_Di_Kampus
45. Tag: Cara_Mengurus_Kartu_Tanda_Mahasiswa_Yang_Hilang
46. Tag: Batas_Waktu_Pembayaran_Uang_Kuliah
47. Tag: KRS_Dan_Bagaimana_Cara_Mengisinya
48. Tag: Cara_Memilih_Mata_Kuliah_Pilihan
49. Tag: Periode_Pengisian_KRS_Dibuka
50. Tag: Syarat_Kelulusan_Dari_Program_Studi
51. Tag: Mengulang_Mata_Kuliah
52. Tag:
Cara_Mengurus_Rekomendasi_Dari_Program_Studi_Untuk_Beasiswa
53. Tag: Cara_Mengurus_Surat_Pindah_Kuliah_Ke_Kampus_Lain
54. Tag: Cara_Mengakses_Materi_Kuliah_Online
55. Tag: Masalah_Dengan_Absensi
56. Tag: Langkah-Langkah_Untuk_Mengajukan_Topik_Skripsi
57. Tag: Mengajukan_Permohonan_Perubahan_Topik_Skripsi
58. Tag: Masalah_Dengan_Nilai_Skripsi
59. Tag: Permohonan_Perubahan_Dosen_Pembimbing_Akademik
60. Tag: Masalah_Dengan_Nilai_Ujian_Akhir_Semester
61. Tag: Masalah_Dengan_Nilai_Ujian_Tengah_Semester
62. Tag: Memperbarui_Informasi_Kontak_Di_Sistem_Akademik
63. Tag: Akses_Ke_Platform_E-Learning_Kampus
64. Tag: Tidak_Bisa_Hadir_Dalam_Ujian_Akhir_Semester
65. Tag: Syarat_Untuk_Registrasi_Ulang_Setiap_Semester
66. Tag: Lupa_Password_Akun_Mahasiswa
67. Tag: Prosedur_Untuk_Mengambil_Cuti_Semester
68. Tag: Mengurus_Legalisasi_Ijazah
69. Tag: Mengajukan_Permohonan_Perubahan_Jurusan
70. Tag: Mengajukan_Permohonan_Cuti_Akademik_Untuk_Satu_Semester
71. Tag: Langkah-Langkah_Dalam_Penyusunan_Skripsi
72. Tag: Waktu_Pendaftaran_Sidang_Skripsi
73. Tag: Langkah-Langkah_Untuk_Menyusun_Proposal_Skripsi
74. Tag: Mengalami_Kesulitan_Dalam_Mata_Kuliah_Tertentu
75. Tag: Mengajukan_Permohonan_Perubahan_Jadwal_Kuliah
76. Tag: Nilai_Ujian_Tidak_Sesuai
77. Tag: Mengetahui_Jadwal_Bimbingan_Skripsi_Dengan_Dosen
78. Tag: Topik_Kerja_Praktek
79. Tag: Tahap_Pengajuan_Kerja_Praktek
80. Tag: About_FIKTIBOT
81. Tag: Pencipta_FIKTIBOT
82. Tag: Salam