Indonesian Journal on Data Science
ISSN 2987-7423 - Vol. 2, No. 2, November 2024, hlm 77-86

Sistem *Chatbot* Layanan Informasi Mahasiswa Menggunakan Algoritma *Long Short-Term Memory*

Dewi Arumsari [a,1], Kharisma [b,2,*], Ulfi Saidata Aesyi [b,3]

[a,b] Program Studi Sistem Informasi, Universitas Jenderal Achmad Yani Yogyakarta
[1]dewiarumsari38@gmail.com; [2]kharisma_anoe@gmail.com*; [3]ulfiaesyi@gmail.com

* corresponding author

ABSTRACT

In the era of globalization and rapid information flow, the demand for efficient and accurate information, especially within academic institutions, is rising. Students often face challenges in accessing educational resources and real-time information, particularly outside official working hours. Existing online information services have limitations in providing continuous access. This research focuses on developing and evaluating a student information service chatbot system at Universitas Jenderal Achmad Yani Yogyakarta (UNJAYA) using the Long Short-Term Memory (LSTM) algorithm. The primary objective is to create a system that delivers real-time, accurate, and efficient information services to students. The Machine Learning Development Cycle (MLDC) is employed in the model development process, including stages such as data collection, processing, model training, evaluation, and implementation. The system's performance is tested using a questionnaire distributed to students, with responses measured on a Likert scale. The results demonstrate a chatbot with a 97.76% accuracy rate, 98.34% precision, and 97.76% recall. The overall system evaluation yielded an average score of 3.87, categorized as good. This research concludes that the LSTM-based chatbot successfully enhances information services at the Faculty of Engineering and Information Technology, providing an innovative solution to meet student needs in real-time

ARTICLE INFO

Article history
Received: 13 November 2024
Revised: 22 November 2024
Accepted:10 December 2024

Keywords
Chatbot,
Information Service,
Long Short-Term Memory,
Machine Learning Development Cycle,
Real-time Information,
Student Services

I. Pendahuluan

Di tengah kemajuan globalisasi dan peningkatan kecepatan aliran informasi, terdapat tuntutan yang semakin meningkat untuk memperoleh informasi yang efisien dan akurat. Pemanfaatan teknologi informasi dan komunikasi menjadi fundamental dalam berbagai aspek kehidupan, terutama di sektor pendidikan, khususnya di institusi akademik[1]. Mahasiswa saat ini menghadapi tantangan yang lebih kompleks, yang mencakup kebutuhan akan aksesibilitas yang lebih tinggi terhadap sumber daya pendidikan, peningkatan interaksi dengan fakultas dan staf administrasi, serta kebutuhan mendesak akan informasi yang tersedia secara *real-time* [2]. Untuk menghadapi tantangan ini, institusi akademik perlu sebuah inovasi untuk meningkatkan layanan informasi mahasiswa dan memfasilitasi interaksi yang lebih efektif [3].

Saat ini, mahasiswa dapat mengakses berbagai layanan informasi secara *online*, seperti sistem informasi perguruan tinggi dan situs web resmi perguruan tinggi. Fakultas juga menggunakan media

sosial untuk menyebarkan informasi seperti jadwal perkuliahan, alur pendaftaran MBKM, dan jadwal ujian [4]. Mahasiswa dengan jadwal yang padat dan mengikuti berbagai kegiatan organisasi seringkali membutuhkan akses informasi akademik dan non-akademik di luar jam kerja resmi. Keterbatasan saluran komunikasi interaktif yang tersedia dapat menyebabkan mereka mengalami hambatan dalam mendapatkan jawaban atas pertanyaan mereka. Hal ini dapat berdampak pada tingkat kepuasan mahasiswa terhadap layanan informasi yang disediakan oleh kampus.

Pengembangan sistem dialog yang mampu menjawab pertanyaan pengguna secara otomatis dapat diwujudkan dengan penerapan teknik pemrosesan bahasa natural atau *Natural Language Processing* (NLP). *Chatbot* adalah program perangkat lunak yang menggunakan teks atau *text-to-speech* untuk melakukan percakapan obrolan *online* sebagai pengganti komunikasi langsung dengan manusia [5]. *Chatbot* dapat menjadi solusi yang efektif untuk mengatasi keterbatasan ini. Sebagian mahasiswa lebih menyukai penggunaan *chatbot* dalam pembelajaran, karena dapat membantu mahasiswa yang terlalu sibuk untuk mendapatkan informasi. Dengan demikian, *chatbot* tidak hanya berfungsi sebagai alat bantu untuk akses informasi tetapi juga sebagai alat untuk memfasilitasi proses pembelajaran yang lebih mandiri dan adaptif bagi mahasiswa [6].

Penelitian sebelumnya menunjukkan keberhasilan penerapan algoritma *Long Short-Term Memory* (LSTM) dalam *chatbot*. Sebagai contoh, penelitian Fahadra [7] menunjukkan bahwa LSTM mampu memberikan respons yang akurat terhadap pertanyaan tentang tugas akhir. Penelitian Wintoro et al. [8] pada *chatbot* akademik juga menyatakan bahwa model berbasis LSTM memiliki tingkat keberhasilan yang tinggi dalam memberikan respons percakapan. Selain itu, studi oleh Khaqiqi et al. [9] menunjukkan kemampuan LSTM dalam menangani bahasa gaul untuk menghasilkan respons yang komprehensif, sementara Jhaerol & Sudianto [10] mendokumentasikan skor akurasi 100% dalam *chatbot* untuk program Merdeka Belajar Kampus Merdeka.

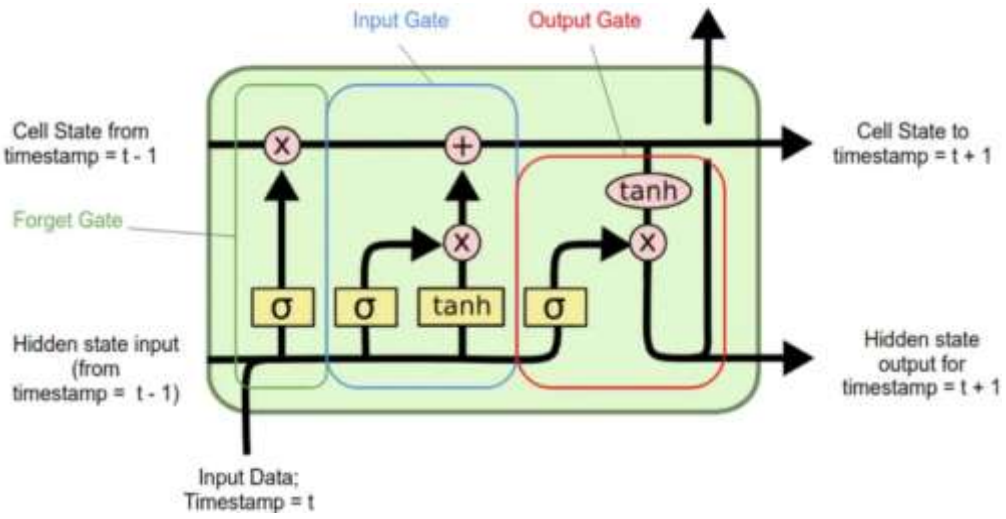
Namun, terdapat tantangan dalam memilih model yang paling tepat berdasarkan kebutuhan dan karakteristik *dataset*. Studi komparatif menunjukkan bahwa meskipun model seperti BERT unggul dalam banyak kasus, model ini cenderung mengalami *overfitting* pada *dataset* kecil, sedangkan LSTM memberikan hasil yang lebih baik dengan akurasi tinggi dan waktu pelatihan yang lebih singkat [11]. Oleh karena itu, pemilihan model harus mempertimbangkan ukuran *dataset*, tujuan penelitian, serta efisiensi dalam pengembangan.

Meskipun penelitian-penelitian sebelumnya menunjukkan keberhasilan penerapan LSTM untuk *chatbot*, penelitian ini memiliki beberapa keunikan yang membedakannya. Tidak hanya berfokus pada akurasi respons *chatbot*, penelitian ini juga mengedepankan kemampuan chatbot untuk memberikan jawaban berdasarkan konteks pertanyaan spesifik layanan mahasiswa di Fakultas Teknik dan Teknologi Informasi Universitas Jenderal Achmad Yani Yogyakarta. Data yang digunakan diambil langsung dari sumber informasi resmi fakultas, memastikan relevansi dan akurasi konten. Penelitian ini juga akan mengevaluasi efektivitas *chatbot* dalam memberikan respons cepat terhadap pertanyaan terkait administrasi akademik dan non-akademik, serta mengukur dampaknya terhadap tingkat kepuasan mahasiswa.

Natural Language Processing (NLP) adalah sains yang menggunakan algoritma dan teknologi kecerdasan buatan untuk membantu mesin mengerti dan menganalisis bahasa alami manusia [12]. Dengan kemajuan dalam NLP dan kecerdasan buatan (AI), *chatbot* telah berkembang dari sistem sederhana berbasis aturan menjadi agen percakapan canggih yang diberdayakan oleh teknologi seperti LSTM [13]. LSTM merupakan bagian dari *Recurrent Neural Network* (RNN), dirancang untuk memproses data berurutan dengan mengatasi masalah *vanishing gradient*, sehingga mampu memahami konteks percakapan dan memberikan respons yang lebih akurat [14], [15]. Universitas Jenderal Achmad Yani Yogyakarta merupakan institusi yang cocok untuk mengimplementasikan dan mengevaluasi sistem *chatbot* layanan informasi mahasiswa berbasis LSTM karena komitmennya terhadap inovasi teknologi. Diharapkan penelitian ini dapat membantu universitas mengoptimalkan layanan informasi. Dengan pendekatan berbasis *Machine Learning Development Life Cycle (MDLC)*, penelitian ini akan menyelidiki pengembangan dan evaluasi sistem *chatbot* layanan informasi mahasiswa untuk memenuhi kebutuhan layanan informasi yang berkualitas tinggi [16].

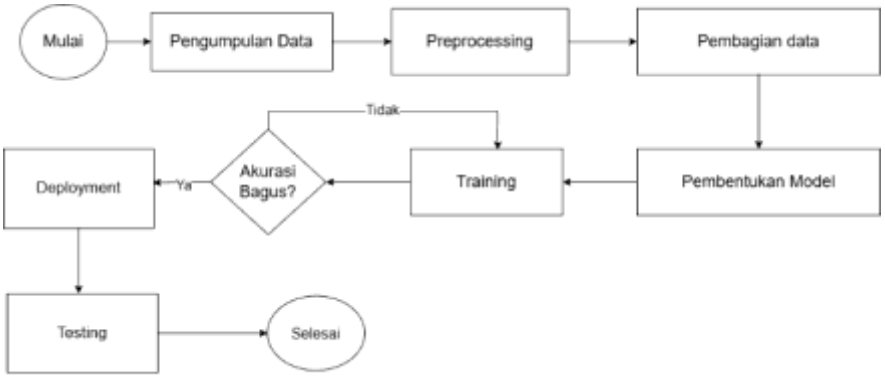
2. Metodologi Penelitian

Penelitian ini menggunakan pendekatan berbasis *Natural Language Processing (NLP)* dan algoritma *Long Short-Term Memory* (LSTM*)*. LSTM mampu mengatasi masalah *vanishing gradient* yang dihadapi oleh *Recurrent Neural Network* (RNN) tradisional, sehingga dapat mempelajari dependensi jangka panjang. Algoritma LSTM mempunyai beberapa *gate*, diantaranya *Forget Gate, Input Gate, Cell State* dan *Output Gate*. Gambar 1 merupakan arsitektur dari LSTM [17]



Gambar 1. Arsitektur LSTM [17]

LSTM *cell* menerima masukan dan menyimpannya selama periode waktu tertentu. *Input gate* menentukan seberapa banyak informasi baru yang masuk ke dalam *cell*. *Forget gate* mengatur seberapa banyak informasi yang akan disimpan dalam *cell*. *Output gate* mengendalikan seberapa banyak informasi dalam *cell* yang digunakan untuk menghitung nilai aktivasi keluaran dari unit LSTM [15]. Data yang digunakan dalam penelitian ini berkaitan dengan layanan mahasiswa di Fakultas Teknik dan Teknologi Informasi Universitas Jenderal Achmad Yani Yogyakarta. Tahapan penelitian dapat dilihat pada Gambar 2.



Gambar 2. *Flowchart* Jalan Penelitian

2.1. Pengumpulan Data

Pengumpulan data dilakukan dengan membuat data pertanyaan dan jawaban berdasarkan informasi dari fakultas terkait layanan disimpan dalam format JSON dan dikonversi menjadi *dataframe*.

2.2. Preprocessing Data

Pada tahap ini, dilakukan pembersihan data melalui *remove punctuation* dan *remove stopwords*. Kemudian, dilakukan *stemming* menggunakan Sastrawi dan *tokenization* untuk memecah teks menjadi kata-kata.

2.3. Pembagian Data

Setelah tahap *preprocessing* selesai dilakukan, data yang telah diproses kemudian dibagi menjadi dua set yang berbeda, yaitu data latih (*training data*) dan data uji (*testing data*). Pembagian ini dilakukan dengan menggunakan rasio 80:20, di mana 80% dari total data digunakan untuk melatih model, sementara 20% sisanya digunakan untuk menguji kinerja model yang telah dilatih.

2.4. Modelling

Model dibangun menggunakan algoritma *Long Short-Term Memory* (LSTM) dengan beberapa lapisan, termasuk lapisan *embedding* dan dua lapisan LSTM untuk memproses urutan input. Model ini dilatih dengan data yang telah diproses untuk dapat mengenali pola dalam pertanyaan dan memberikan jawaban yang relevan.

2.5. Training dan Evaluasi Model

Model dilatih menggunakan data latih dengan pengaturan *epoch* dan *batch size* tertentu. Selama pelatihan, digunakan *callback* seperti *early stopping* dan *model checkpoint* untuk mengoptimalkan hasil. Evaluasi dilakukan dengan *confusion matrix* untuk mengukur akurasi, presisi, dan *recall* menggunakan data uji.

2.6. Deployment

Model yang telah dilatih diimplementasikan dalam aplikasi web menggunakan Streamlit. Streamlit adalah *framework* Python yang dirancang untuk mengintegrasikan model *machine learning* dengan aplikasi web interaktif tanpa memerlukan keahlian dalam pengembangan web seperti HTML, CSS, atau JavaScript [18]. *Framework* ini memberikan kemudahan bagi pengguna untuk mengajukan pertanyaan dalam bahasa Indonesia melalui antarmuka web yang intuitif dan responsif.

2.7. Testing

Pengujian dilakukan menggunakan Skala Likert untuk mengevaluasi fungsi, tampilan, dan kualitas hasil yang diberikan oleh *chatbot*. Skala Likert dipilih karena kemampuannya mengukur persepsi pengguna dengan menggunakan serangkaian pernyataan yang dinilai berdasarkan skala ordinal. Dalam penelitian ini, skala yang digunakan adalah lima poin, yaitu 1 (Sangat Tidak Baik), 2 (Tidak Baik), 3 (Netral), 4 (Baik), dan 5 (Sangat Baik) [19].

Hasil dari kuesioner dievaluasi dengan karakteristik penilaian sebagaimana ditunjukkan dalam interval nilai, yaitu 1,00–1,08 (Sangat Tidak Baik), 1,09–2,79 (Tidak Baik), 2,80–3,69 (Netral), 3,70–4,59 (Baik), dan 4,60–5,00 (Sangat Baik) [20]. Pendekatan ini memastikan bahwa pengukuran tingkat kepuasan pengguna terhadap *chatbot* dilakukan secara objektif dan terstruktur.

3. Hasil dan Pembahasan

3.1. Pengumpulan Data

Data diperoleh dari Fakultas Teknik dan Teknologi Informasi melalui situs web https://ftti.unjaya.ac.id/. Data disusun dalam format JSON dengan total 1.437 entri. Data tersebut terdiri dari *tag* sebagai label, *patterns* sebagai pertanyaan, dan *responses* sebagai jawaban. Data tersebut kemudian dikonversi menjadi dataframe Pandas yang dapat dilihat pada Tabel 1.

Table 1. Dataset

No	*Pattern*	*Tag*
1	Halo	greeting
2	FTTI ada jurusan apa saja?	ftti
3	Berapa program studi di FTTI?	ftti
4	Lokasi FTTI dimana?	alamat

No	*Pattern*	*Tag*
5	Dimana lokasi fakultas teknik?	alamat
6	Mulai kapan UAS?	Jadwaluas
7	Ujian akhir semester kapan?	jadwal_uas
8	Kapan semester tambahan	semester_perbaikan
9	Apakah semester tambahan wajib diikuti?	semester_perbaikan
10	Gimana ngurus surat aktif kuliah?	surat_keterangan_aktif_kuliah
11	Surat aktif kuliah minta ke siapa?	surat_keterangan_aktif_kuliah
12	Bagaimana prosedur mendapatkan transkrip nilai?	Transkrip_nilai
13	Transkrip nilai hubungi siapa?	Transkrip_nilai

3.2. Prepocessing Data

Data yang telah dikumpulkan dilakukan preprocessing data. Tahapan ini meliputi *case folding, cleaning text, stopwords, stemming*, dan *tokenizing*. Pada tabel 2, ditunjukkan sampel data dari hasil *preprocessing*.

Table 2. Sampel Data Hasil *Processing*

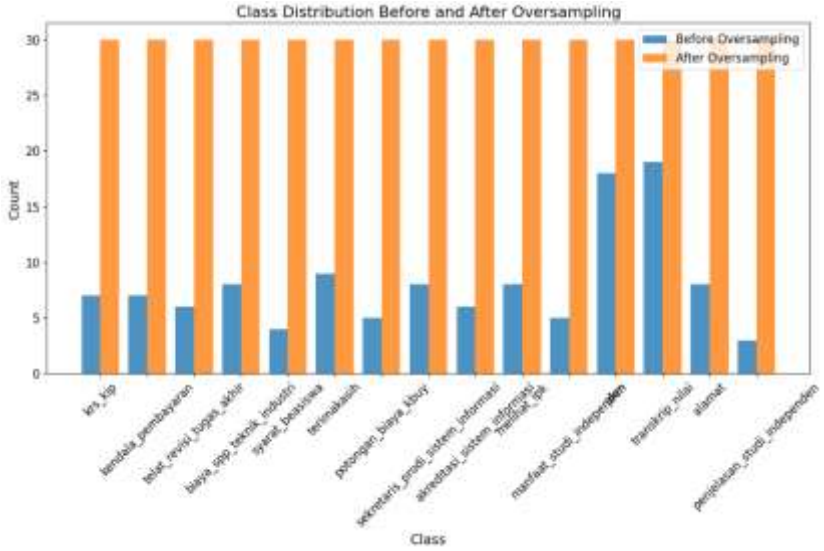
Pattern Asli	Program studi apa saja yang ada di fakultas ini?
Case Folding	program studi apa saja yang ada di fakultas ini?
Cleaning Text	program studi apa saja yang ada di fakultas ini
Stopwords	program studi fakultas
Stemming	program studi fakultas
Tokenizing	[’program’, ’studi’, ’fakultas’]

Setelah tahapan *preprocessing*, langkah selanjutnya adalah proses *embedding*, dimana setiap kata dalam teks dikonversi menjadi vektor *numerik* berdimensi tetap menggunakan model Word2Vec. Model Word2Vec dilatih dengan parameter-parameter yang tertera pada Tabel 3.

Table 3. Parameter *Embedding*

vector_size	*window*	*min_count*	*workers*
100	5	1	4

Untuk menangani masalah ketidakseimbangan kelas pada dataset dilakukan *oversampling*. Ketidakseimbangan ini terjadi dimana satu kelas memiliki jumlah sampel yang jauh lebih sedikit dibanding kelas lainnya sehingga dapat menghambat kemampuan model dalam belajar. Metode *RandomOverSampler* diterapkan untuk meningkatkan representasi kelas minoritas dengan penggandaan beberapa contoh dari kelas minoritas hingga jumlahnya seimbang dengan kelas mayoritas. Gambar 2 menunjukan perbandingan distribusi kelas sebelum dan sesudah *oversampling.*



Gambar 3. Distribusi Kelas

3.3. Pembagian Data

Proses pembagian data menjadi dua bagian yaitu data latih (80%) dan data uji (20%). Pembagian ini dilakukan secara acak untuk memastikan data uji memiliki variasi yang cukup sehingga model dapat dilatih dengan optimal dan dievaluasi dengan akurat.

3.4. Modelling

Model yang digunakan untuk mengklasifikasikan intent adalah model *Long Short-Term Memory* (LSTM). LSTM dipilih karena kemampuannya dalam menangani data berurutan dan menangkap hubungan temporal antar kata dalam teks. Model LSTM yang dikembangkan terdiri dari beberapa lapisan, dimulai dengan lapisan *input* yang menerima data *tokenized* yang telah diproses sebelumnya. Kemudian, lapisan pertama adalah *Embedding layer*, yang mengubah kata-kata menjadi representasi vektor berdimensi tetap. Matriks *embedding* yang digunakan diinisialisasi dengan menggunakan *embedding matrix* yang diperoleh dari teknik pra-pelatihan Word2Vec, yang terbukti memberikan hasil terbaik dibandingkan dengan metode pra-pelatihan lainnya berdasarkan eksperimen yang dilakukan dalam penelitian ini, yang memungkinkan model untuk menangkap konteks kata secara lebih efektif.

Setelah lapisan *embedding*, model dilanjutkan dengan dua lapisan LSTM. Lapisan pertama LSTM memiliki 256unit dan diatur untuk mengembalikan sekuens (*return_sequences=True*), yang memungkinkan informasi dari setiap langkah waktu dalam sekuens dipertahankan untuk diproses lebih lanjut di lapisan berikutnya. Lapisan LSTM kedua memiliki 128 unit dan berfungsi untuk memproses informasi yang lebih ringkas dan terfokus dari lapisan pertama.

Untuk menghindari *overfitting*, lapisan *Dropout* disisipkan setelah setiap lapisan LSTM dan *Dense*. *Dropout* membantu dalam menurunkan ketergantungan berlebihan pada *neuron-neuron* tertentu dengan cara mematikan sebagian *neuron* selama pelatihan. Nilai *dropout* yang digunakan adalah 30%.

Lapisan terakhir adalah *Dense layer*, yang berfungsi untuk memetakan hasil dari lapisan LSTM ke dalam ruang kelas yang lebih tinggi dengan fungsi aktivasi ReLU. *Output* akhir dari model ini adalah lapisan *Dense* dengan fungsi aktivasi *softmax*, yang menghasilkan probabilitas untuk setiap kelas.

Model ini dioptimalkan menggunakan algoritma Adam, dengan *learning rate* yang diatur sebesar 0.001, nilai yang dipilih berdasarkan eksperimen *trial and error* untuk mencapai kinerja terbaik. Model dilatih menggunakan *categorical crossentropy* sebagai fungsi *loss*, mengingat tugas yang dilakukan adalah klasifikasi multi-kelas. Dalam mengatasi ketidakseimbangan data yang ada, digunakan metode *oversampling* menggunakan *RandomOverSampler*. Metode ini memastikan bahwa jumlah sampel dari masing-masing kelas lebih seimbang untuk meningkatkan kinerja model, terutama untuk kelas-kelas minoritas. Dalam hal pemilihan *hyperparameter*, sejumlah parameter penting dipilih berdasarkan eksperimen awal dan literatur yang ada. Jumlah *epoch* ditetapkan sebanyak 500

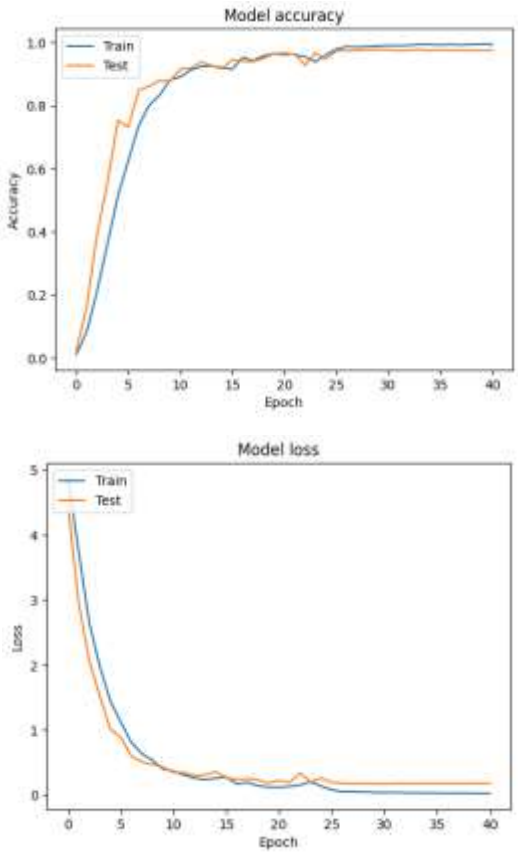
dengan pengaturan *early stopping* untuk mencegah model melatih terlalu lama jika tidak ada peningkatan dalam validasi *loss*. Penggunaan *ReduceLROnPlateau* membantu mengurangi *learning rate* jika model tidak mengalami peningkatan pada *validation loss* setelah sejumlah *epoch* tertentu.

3.5. Training dan Evaluasi Model

Proses pelatihan model dilakukan dengan menggunakan data pelatihan yang telah diproses dan di-*oversample* untuk mengatasi ketidakseimbangan kelas. Model dilatih selama 500 *epoch* dengan *batch size* 32, namun menggunakan teknik *early stopping* untuk menghentikan pelatihan jika tidak ada perbaikan pada *validation loss* setelah 10 *epoch* berturut-turut. Hal ini menghindari terjadinya *overfitting* dan memastikan bahwa model hanya berlatih sebanyak yang diperlukan.

Selama pelatihan, beberapa *callback* digunakan untuk memonitor performa model. *ModelCheckpoint* digunakan untuk menyimpan model terbaik berdasarkan *validation loss*, sehingga jika terjadi *overfitting*, model terbaik yang diperoleh selama pelatihan dapat dipertahankan. *ReduceLROnPlateau* juga diterapkan untuk mengurangi *learning rate* apabila *validasi loss* tidak menunjukkan perbaikan, yang bertujuan untuk memungkinkan model melakukan pembaruan lebih halus pada tahap akhir pelatihan.

Model mencapai *accuracy* sebesar 0,98 dan validasi *loss* sebesar 0,15, dengan validasi *accuracy* sebesar 0,96 dan *loss* sebesar 0,05. Grafik yang menggambarkan perubahan *accuracy* dan *loss* selama pelatihan, serta validasi *accuracy* dan validasi *loss* pada setiap *epoch*, dapat dilihat pada Gambar 3.



Gambar 3. Grafik Analisis Akurasi, *Val_accuracy*, *Loss dan Val_loss* Pelatihan Model LSTM

Dari grafik akurasi, terlihat bahwa garis berwarna biru menunjukkan *accuracy* pelatihan yang baik, sedangkan garis berwarna oranye menunjukkan validasi *accuracy* yang sedikit lebih rendah. Pada grafik *loss*, garis biru menunjukkan *loss* pelatihan yang rendah, sementara garis oranye menunjukkan validasi loss yang sedikit lebih tinggi. Dari grafik tersebut, dapat disimpulkan bahwa proses pelatihan model menghasilkan kinerja yang baik tanpa tanda-tanda *overfitting*.

Table 4. Evaluasi Model

Accuracy	*Precision*	*Recall*
0.977	0,983	0,977

Berdasarkan evaluasi model menggunakan *confusion matrix*, didapatkan hasil yang sangat baik dengan nilai akurasi mencapai 97.76%. Nilai akurasi yang tinggi ini menunjukan bahwa model secara keseluruhan mampu mengklasifikasikan data dengan benar hampir pada semua kasus.

Selain itu, nilai presisi sebesar 98.34% menunjukan bahwa model jarang memberikan hasil positif yang salah. Dalam konteks layanan mahasiswa, ini berarti model sangat jarang memberikan informasi yang salah ketika mengidentifikasi layanan yang tersedia. Presisi yang tinggi sangat penting untuk mengurangi jumlah kesalahan positif (*false positives*), di mana model salah mengidentifikasi suatu layanan yang sebenarnya tidak ada.

Sementara itu, *recall* sebesar 97.76% menunjukkan bahwa model mampu menemukan sebagian besar dari semua *instance* yang sebenarnya positif. Ini berarti bahwa jika ada layanan tertentu yang harus diidentifikasi oleh *chatbot*, model ini mampu mengenalinya hampir pada semua kasus. *Recall* yang tinggi penting untuk memastikan bahwa layanan – layanan yang ada tidak terlewatkan oleh model.

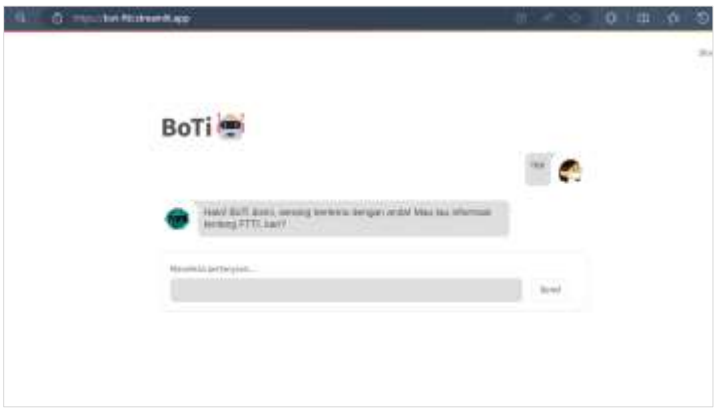
Secara keseluruhan, metrik evaluasi yang tinggi ini menandakan bahwa model LSTM yang dibangun tidak hanya akurat dalam melakukan klasifikasi tetapi juga memiliki keseimbangan yang baik antara mengidentifikasi layanan dengan benar dan menghindari kesalahan identifikasi. Hal ini menunjukkan potensi yang besar untuk diimplementasikan dalam sistem *chatbot* layanan mahasiswa yang efektif.

3.6. Deployment

Penelitian ini bertujuan untuk mempermudah penggunaan model *chatbot* melalui proses *deployment* ke dalam aplikasi berbasis web. Proses ini dilakukan menggunakan Streamlit untuk memfasilitasi akses pengguna terhadap *chatbot* layanan mahasiswa. Model LSTM yang telah dilatih sebelumnya disimpan dalam format HDF5. Selain itu, file pendukung seperti *tokenizer* dan *label encoder* juga disiapkan sebagai bagian dari implementasi.

Pada tahap implementasi, file model_LSTM.h5 yang berisi model, serta *tokenizer.pkl* dan *label_encoder.pkl* yang digunakan untuk *preprocessing* teks dan *decoding label*, dimuat ke dalam *server* aplikasi berbasis Streamlit. Aplikasi ini menyediakan antarmuka yang intuitif, di mana pengguna dapat memasukkan pertanyaan secara langsung untuk direspon oleh *chatbot*.

Jika probabilitas hasil prediksi *intent* lebih dari atau sama dengan 85%, *chatbot* akan menampilkan jawaban yang relevan dari *database* layanan mahasiswa. Sebaliknya, jika probabilitas prediksi kurang dari 85%, sistem akan mengembalikan pesan khusus yang menyatakan bahwa pertanyaan tersebut tidak dikenali. Tampilan halaman *chatbot* yang diimplementasikan pada situs web dapat dilihat pada Gambar 4.



Gambar 4. Tampilan *Chatbot* Pada Website

Gambar 4 menunjukkan contoh implementasi respons *chatbot*. *Chatbot* memberikan jawaban yang sesuai berdasarkan hasil prediksi intent dengan probabilitas tinggi. Apabila probabilitas prediksi memenuhi ambang batas yang telah ditentukan, *chatbot* akan memberikan respons yang relevan. Apabila tidak, *chatbot* akan mengembalikan pesan khusus yang menyatakan bahwa pertanyaan tersebut tidak dikenali.

3.7. Testing

Pengujian sistem *chatbot* layanan informasi dilakukan dengan metode skala Likert, melibatkan 16 responden dari mahasiswa Fakultas Teknik dan Teknologi Informasi. Responden menguji chatbot melalui serangkaian tahapan, kemudian menjawab 14 pertanyaan yang menilai aspek fungsi, tampilan, dan hasil. Skor rata-rata keseluruhan dari pengujian ini adalah 3,87, yang menunjukkan bahwa sistem *chatbot* dikategorikan baik menurut skala penilaian.

4. Kesimpulan

Berdasarkan hasil dari penelitian ini, dapat disimpulkan bahwa sistem *chatbot* layanan informasi mahasiswa menggunakan algoritma *Long Short-Term Memory* (LSTM) dapat efektif dalam memberikan informasi seputar layanan akademik. Hasil pengujian menunjukkan bahwa model LSTM yang diterapkan memiliki tingkat akurasi sebesar 97.76%, presisi 98.34%, dan *recall* 97.76%. Hal ini menegaskan bahwa model ini mampu memberikan jawaban yang relevan dan akurat terhadap pertanyaan yang diajukan oleh mahasiswa. Semakin banyak data pertanyaan yang dilatih, semakin tinggi juga tingkat akurasi dan keakuratan jawaban yang diberikan oleh *chatbot.*

5. Saran

Berdasarkan hasil penelitian, saran yang dapat digunakan untuk penelitian selanjutnya adalah sebagai berikut:

- Dataset yang digunakan sebagai bahan latih pada model perlu diperluas dengan menambahkan lebih banyak variasi pertanyaan dan jawaban yang mungkin diajukan oleh mahasiswa. Hal ini dapat mencakup informasi terbaru dan detail mengenai berbagai aspek layanan akademik.
- Mempertimbangkan penggunaan metode lain seperti BERT (*Bidirectional Encoder Representations from Transformers*) atau GPT (*Generative Pre-trained Transformer*) yang memiliki cakupan lebih luas dan kemampuan pemahaman bahasa yang lebih baik.
- Untuk menangani pertanyaan atau masalah yang tidak dapat diselesaikan oleh *chatbot*, fitur interaksi langsung dengan admin dapat ditambahkan. Hal ini akan memastikan bahwa semua kebutuhan informasi mahasiswa dapat terpenuhi secara maksimal.

REFERENSI

[1] M. N. Alfareza, “Pembangunan Chatbot Menggunakan Natural Language Processing Di Jurusan Teknik Industri Universitas Islam Indonesia,” 2020. [Online]. Available: https://api.semanticscholar.org/CorpusID:226025343

[2] N. H. Capsa, K. T. Barlian, Z. D. Amandari, and P. Rachmadi, “Analisa Kebutuhan Chatbot Di Layanan FTI Perbanas Jakarta,” in *Prosiding Seminar Nasional*, 2023, pp. 197–206.

[3] A. Lubis and I. Sumartono, “Implementasi Layanan Akademik Berbasis Chatbot untuk Meningkatkan Interaksi Mahasiswa,” *RESOLUSI: Rekayasa Teknik Informatika dan Informasi*, vol. 3, no. 5, pp. 246–252, 2023.

[4] D. G. S. Ruindungan and A. Jacobus, “Chatbot Development for an Interactive Academic Information Services using the Rasa Open Source Framework,” *Jurnal Teknik Elektro dan Komputer*, vol. 10, no. 1, pp. 61–68, 2021.

[5] B. Luo, R. Y. K. Lau, C. Li, and Y. Si, “A critical review of state-of-the-art chatbot designs and applications,” *Wiley Interdiscip Rev Data Min Knowl Discov*, vol. 12, no. 1, p. e1434, 2022.

[6] J. Yin, T.-T. Goh, B. Yang, and Y. Xiaobin, "Conversation Technology With Micro-Learning: The Impact of Chatbot-Based Learning on Students' Learning Motivation and Performance," *Journal of Educational Computing Research*, vol. 59, no. 1, pp. 154–177, Mar. 2021, doi: 10.1177/0735633120952067.

[7] L. Fahadra, "Sistem Chatbot Layanan Informasi Tugas Akhir Menggunakan Metode Long Short-Term Memory (LSTM)," *Universitas Islam Sultan Agung*, 2023.

[8] P. B. Wintoro, H. Hermawan, M. A. Muda, and Y. Mulyani, "Implementasi Long Short-Term Memory pada Chatbot Informasi Akademik Teknik Informatika Unila," *Expert J. Manaj. Sist. Inf. dan Teknol*, vol. 12, no. 1, p. 68, 2022.

[9] M. I. T. Khaqiqi, N. H. Harani, and C. Prianto, "Performance Analysis and Development of QnA Chatbot Model Using LSTM in Answering Questions," *Indonesian Journal of Computer Science*, vol. 12, no. 3, 2023.

[10] M. R. Jhaerol and S. Sudianto, "Implementation of chatbot for merdeka belajar kampus merdeka program using long short-term memory," *Jurnal Nasional Pendidikan Teknik Informatika: Janapati*, vol. 12, no. 2, 2023.

[11] A. Ezen-Can, "A Comparison of LSTM and BERT for Small Corpus," *arXiv preprint arXiv:2009.05451*, 2020.

[12] D. Khurana, A. Koli, K. Khatter, and S. Singh, "Natural language processing: state of the art, current trends and challenges," *Multimed Tools Appl*, vol. 82, no. 3, pp. 3713–3744, 2023, doi: 10.1007/s11042-022-13428-4.

[13] M. Al-Amin *et al.*, "History of generative Artificial Intelligence (AI) chatbots: past, present, and future development," *arXiv preprint arXiv:2402.05122*, 2024.

[14] A. Azni, "Implementasi Natural Language Processing Pada Sistem Chatbot Informasi Saham dengan Algoritma Long Short-Term Memory (Lstm) dan Fuzzy String Matching," *Doctoral Dissertation, Universitas Sumatera Utara*, 2021.

[15] P. W. Cahyo and U. S. Aesyi, "Perbandingan LSTM dengan Support Vector Machine dan Multinomial Na ve Bayes pada Klasifikasi Kategori Hoax," *Jurnal Transformatika*, vol. 20, no. 2, pp. 23–29, 2023.

[16] J. W. &. Sons, *Keras to Kubernetes®: The Journey of a Machine Learning Model to Production, New York: Inc*. 2019.

[17] Ryan T. J. J., "LSTMs Explained: A Complete, Technically Accurate, Conceptual Guide with Keras," Analytics Vidhya.

[18] Streamlit, "Streamlit documentation." Accessed: Mar. 21, 2024. [Online]. Available: https://docs.streamlit.io/

[19] P. B. A. A. Putra, "Pengembangan aplikasi kuesioner survey berbasis web menggunakan skala likert dan guttman," *Jurnal Sains dan Informatika p-ISSN*, vol. 2460, p. 173X, 2019.

[20] N. M. D. Parwati, I. W. Tuwi, and N. L. R. Yusmarisa, "ANALISIS SISTEM INFORMASI AKUNTANSI DALAM PENYUSUNAN LAPORAN KEUANGAN DI HOTEL XYZ DENGAN METODE PIECES," *Neraca: Jurnal Ekonomi, Manajemen dan Akuntansi*, vol. 2, no. 8, pp. 277–283, 2024.