

Sistem Rekomendasi Perguruan Tinggi Swasta menggunakan Metode *Long Short-Term Memory* (LSTM)

MOHAMAD ARDAN, VINY CHRISTANTI MAWARDI, TRI SUTRISNO

Program Studi Teknik Informatika, Universitas Tarumanagara
Email: mohamad.535210090@stu.untar.ac.id

Received 18 November 2024 | *Revised* 15 Maret 2025 | *Accepted* 28 April 2025

ABSTRAK

Penelitian ini bertujuan mengembangkan sistem rekomendasi berbasis Long Short-Term Memory (LSTM) untuk membantu calon mahasiswa dalam memilih perguruan tinggi swasta yang sesuai. Data dikumpulkan melalui UiPath dari situs resmi perguruan tinggi dan Pangkalan Data Pendidikan Tinggi (PDDIKTI). Sistem ini dilengkapi dengan chatbot berbasis Natural Language Processing (NLP) untuk memahami kebutuhan pengguna dan memberikan informasi sesuai dengan preferensi pengguna. Pemilihan model LSTM didasarkan pada kemampuannya dalam menangani data sekuensial dan memahami konteks, sehingga mampu menghasilkan rekomendasi yang akurat. Hasil evaluasi menunjukkan tingkat akurasi rekomendasi sebesar 90% dalam memberikan rekomendasi dan informasi yang tepat. Selain itu, tingkat kepercayaan chatbot dalam menjawab pertanyaan pengguna mencapai 100%, menunjukkan efektivitas tinggi dalam memberikan berbasis data.

Kata kunci: chatbot, rekomendasi, LSTM, web scraping, PDDIKTI.

ABSTRACT

This research aims to develop a recommendation system based on Long Short-Term Memory (LSTM) to assist prospective students in choosing a suitable private university. Data was collected through UiPath from the official websites of universities and the Higher Education Database (PDDIKTI). The system is equipped with a Natural Language Processing (NLP) based chatbot to understand user needs and provide information according to user preferences. The selection of the LSTM model is based on its ability to handle sequential data and understand context, so as to produce accurate recommendations. The evaluation results show a recommendation accuracy rate of 90% in providing the right recommendations and information. In addition, the level of confidence of the chatbot in answering user questions reached 100%, indicating high effectiveness in providing data-based.

Keywords: chatbot, recommendation, LSTM, web scraping, PDDIKTI.

1. PENDAHULUAN

Universitas adalah lembaga pendidikan tinggi yang bertanggung jawab secara signifikan untuk menumbuhkan potensi akademik dan profesional generasi muda. Sebagai konsumen utama, calon mahasiswa sering mencari universitas terbaik untuk melanjutkan pendidikan. Pelayanan yang cepat, responsif, dan informatif harus diterapkan oleh universitas untuk menarik minat mereka. Pelayanan ini mencakup kejelasan informasi, ketepatan prosedur, efisiensi waktu, dan dukungan yang memadai (**Heryati, 2023**). Pertumbuhan institusi dan reputasinya di mata publik dipengaruhi langsung oleh keberhasilan dalam menarik lebih banyak siswa. Hal ini memungkinkan institusi pendidikan tinggi untuk mengalokasikan lebih banyak sumber daya untuk meningkatkan pendidikan dan kualitasnya. Namun, perguruan tinggi swasta harus beradaptasi dengan perubahan kebutuhan siswa, kemajuan teknologi, dan dinamika pendidikan agar dapat terus berkembang. Daya saing di industri pendidikan tinggi sangat bergantung pada inovasi dalam pengajaran, diversifikasi program studi, dan peningkatan layanan mahasiswa (**NAS, 2021**).

Pada proses pemilihan perguruan tinggi, calon mahasiswa mencari informasi seperti biaya kuliah, lokasi, fasilitas, akreditasi, dan program studi yang tersedia (**Amaliya, 2019**). Faktor-faktor ini tidak hanya membantu calon siswa dalam menentukan pilihannya, tetapi juga menawarkan panduan untuk menyesuaikan pilihan dengan minat pribadi, bakat akademik, dan prospek karir di masa depan. Namun, untuk membuat keputusan ini, seringkali diperlukan pertimbangan dan penelitian yang mendalam agar siswa dapat membuat pilihan yang sesuai dengan tujuan jangka Panjang (**Ahmad R. Pratama, 2022**).

Bagi calon mahasiswa, kekurangan informasi yang cukup di situs web universitas seringkali menjadi hambatan untuk mendapatkan informasi yang relevan dan rinci. Sebagian besar situs web hanya menampilkan informasi umum seperti informasi tentang bagaimana mendaftar dan program studi (**Tjut Awaliyah Zuraiyah, 2020**). Namun, untuk membantu dalam pengambilan keputusan, calon mahasiswa juga membutuhkan rekomendasi yang didasarkan pada minat dan bakat untuk calon mahasiswa. Karena struktur organisasi yang kompleks dan pembagian informasi yang tersebar di berbagai halaman web, memperoleh informasi secara mudah dan cepat seringkali menjadi tantangan dalam organisasi besar seperti universitas (**Mohinish Daswani, 2020**).

Dibutuhkan sistem yang dapat mengolah data universitas untuk memberikan informasi dan rekomendasi dengan cepat dan mudah digunakan. Chatbot berbasis teknologi informasi dapat berinteraksi secara langsung dengan pengguna untuk mengetahui apa yang dibutuhkan dan inginkan. Teknologi ini juga dapat secara otomatis memproses percakapan dengan pengguna dan memberikan rekomendasi dan informasi yang dibutuhkan (**Antonius Randy Arjun, 2022**). Pemanfaatan UiPath dalam proses pengambilan data dari situs web sangat mendukung upaya restrukturisasi dataset. Pada umumnya, pengumpulan data secara manual dari berbagai situs perguruan tinggi membutuhkan waktu dan interaksi yang cukup banyak, namun dengan UiPath, proses ini dapat diotomatisasi. UiPath dapat mengakses berbagai situs, menavigasi halaman yang relevan, serta mengekstraksi data yang dibutuhkan dan menyimpannya dalam format yang terstruktur. Dengan demikian, penggunaan UiPath tidak hanya meningkatkan efisiensi dan ketepatan tetapi juga mempercepat proses pengambilan data yang dibutuhkan untuk mendukung sistem rekomendasi berbasis algoritma *Long short-term memory* (LSTM) (**Khang Nhut Lam, 2020**).

Dengan memanfaatkan metode *Deep Learning*, chatbot dapat menggunakan data pembelajaran untuk membuat model statistik yang memungkinkan untuk memahami dan

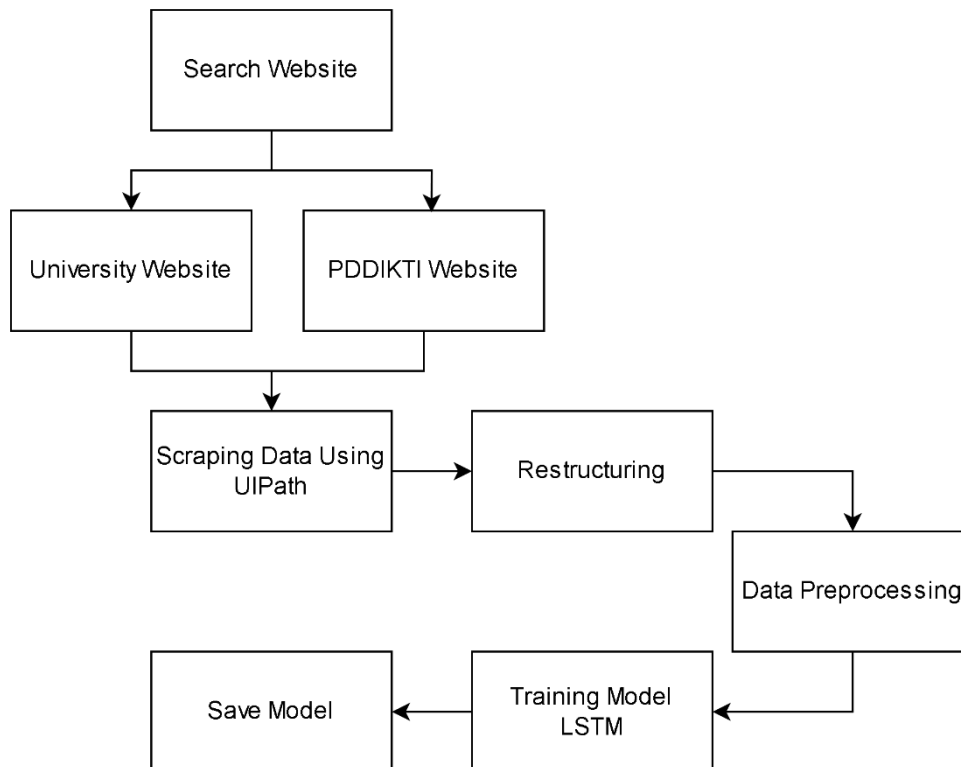
merespons percakapan pengguna. Dengan menggunakan algoritma LSTM yang terdiri dari banyak lapisan, chatbot dapat memproses data dalam bahasa alami dengan lebih baik, yang memungkinkan mereka menghasilkan respons yang lebih kontekstual dan akurat **(Tiara Eka Putri, 2024)**. Proses restrukturisasi dataset dirancang untuk mengatur ulang dan memperbaiki struktur informasi agar lebih sesuai untuk analisis berikutnya. Tujuannya adalah untuk menggabungkan informasi yang relevan dan mendukung penerapan algoritma LSTM dalam sistem rekomendasi kampus, yang dapat memberikan rekomendasi yang akurat berdasarkan preferensi calon mahasiswa. Dalam konteks chatbot, LSTM memungkinkan sistem untuk mengingat konteks percakapan, sehingga chatbot dapat memberikan respons yang lebih tepat dan sesuai dengan kebutuhan pengguna **(Andrew Ciayandi, 2020)**. Implementasi chatbot berbasis LSTM dalam sistem rekomendasi diharapkan dapat membantu calon mahasiswa yang mencari informasi tentang perguruan tinggi swasta di Jakarta. Salah satu teknik pembelajaran mesin, LSTM dapat menangani urutan data dan konteks. Chatbot dalam situasi ini dapat memberikan saran yang disesuaikan dengan preferensi siswa dan profil calon mahasiswa **(Hilya Anbiyani Fitri Muhyidini, 2023)**..

Saat ini penggunaan aplikasi chatbot sangat beragam dan mencakup berbagai sektor, seperti pada sektor pendidikan yaitu Chatbot Berbasis Whatsapp Teknik Informatika Universitas Palangkaraya dengan metode Rules Based System, Sistem berbasis aturan (rules-based) adalah metode yang menggunakan serangkaian aturan atau logika yang telah ditetapkan sebelumnya **(Ferry Saputra, 2024)**. Pada penelitian sebelumnya telah banyak mengembangkan sistem rekomendasi berbasis chatbot. Namun, metode berbasis aturan (rule-based) atau pendekatan pencocokan kata kunci sederhana adalah yang paling umum. Dengan menerapkan algoritma LSTM dalam chatbot rekomendasi kampus dan integrasi UiPath sebagai alat otomatisasi pengumpulan data universitas, penelitian ini menawarkan kebaruan. Akibatnya, penelitian ini membantu membangun sistem yang lebih efektif, akurat, dan dapat dipersonalisasi untuk calon mahasiswa yang mencari informasi tentang perguruan tinggi swasta di Jakarta.

Penelitian ini memberikan kontribusi baru dalam bidang sistem rekomendasi perguruan tinggi swasta, terutama pada penelitian ini dengan menggunakan model *Long Short-Term Memory* (LSTM) untuk menangani data sekuensial dan memahami konteks percakapan pengguna secara lebih efektif. Penggunaan LSTM ini menjadi pembeda utama dibandingkan dengan penelitian sebelumnya yang menggunakan metode berbasis aturan atau algoritma tradisional seperti FastText **(Fahmi Yusron F, 2024)**. Selain itu dengan memanfaatkan teknologi Robotic Process Automation (RPA) untuk otomatisasi pengumpulan data dari situs resmi perguruan tinggi dan Pangkalan Data Pendidikan Tinggi (PDDIKTI). Integrasi ini tidak hanya meningkatkan efisiensi proses pengumpulan data, tetapi juga memastikan dataset yang dihasilkan lebih terstruktur dan akurat dibandingkan metode manual yang banyak digunakan dalam penelitian sebelumnya. Pada penelitian ini mampu memanfaatkan data rekomendasi untuk memahami kebutuhan spesifik calon mahasiswa secara real-time. Kontribusi utama pada penelitian ini dengan menggabungkan beberapa inovasi seperti penggunaan *Long Short-Term Memory* (LSTM) dan Robotic Process Automation (RPA) yaitu untuk menciptakan sistem yang lebih efisien, personal, dan relevan dalam konteks pemilihan perguruan tinggi swasta.

2. METODE PENELITIAN

Pada penelitian ini, metode *Long Short-Term Memory* (LSTM) digunakan. LSTM digunakan untuk mengatasi keterbatasan *Recurrent Neural Network* (RNN) saat menangani data sekuensial, terutama masalah *vanishing gradient*. Mekanisme memori LSTM yang terdiri dari tiga gate yaitu, *forget gate*, *input gate* dan *output gate* memberikan kemampuan untuk memahami hubungan jangka panjang dalam data teks (**Jiaqi Meng, 2023**). Dalam prosesnya, data teks yang diterima sebagai input akan melalui tahap normalisasi untuk memastikan konsistensi dan keseragaman. Selanjutnya, *embedding* digunakan untuk mengubah data teks menjadi representasi numerik yang dapat diproses oleh model. Data ini kemudian diproses oleh model LSTM melalui lapisan *embedding* dan *dropout*, sehingga secara efektif dapat menangkap pola sekuensial dan mencegah overfitting (**Xianyun wen, 2023**). Fungsi aktivasi seperti softmax digunakan pada lapisan output untuk mengkategorikan kueri ke dalam kategori tertentu, seperti akreditasi, biaya kuliah, atau program studi. Sistem ini dibuat untuk memahami konteks percakapan secara menyeluruh dan menghasilkan respons yang akurat. Dengan demikian, sistem ini berfungsi sebagai solusi yang efektif untuk kebutuhan rekomendasi berbasis data.



Gambar 1. Skema Rancangan Proses

Pada Gambar 1 Rancangan restrukturisasi dataset dimulai dengan tahap pengumpulan data, dimana informasi dari situs web akan dikumpulkan melalui *Robotic Process Automation* (RPA) dengan teknik *web scraping* menggunakan UiPath. Proses pengambilan data diawali dengan menentukan sumber data, yaitu situs resmi Perguruan Tinggi Swasta (PTS) dan Pangkalan Data Pendidikan Tinggi (PDDikti) dari KEMEDIKBUDRISTEK, dimana sistem secara otomatis membuka situs, mengidentifikasi elemen antarmuka (UI). Pada proses ini, pengguna memasukkan tautan situs yang ingin diakses, dan sistem secara otomatis memulai pengambilan data dengan mengklik elemen UI serta memasukkan input, seperti kata kunci

atau memilih opsi tertentu dari menu. Data yang berhasil dikumpulkan kemudian diubah menjadi format tabel terstruktur menggunakan wizard, sehingga data siap untuk dianalisis lebih lanjut.

Pada tahap awal, Data yang diambil dari basis data PDDikti difilter sehingga hanya menampilkan perguruan tinggi swasta di wilayah Jakarta. Data yang dikumpulkan di halaman ini meliputi nama perguruan tinggi, lokasi, jumlah program studi, biaya kuliah, dan akreditasi. Pada tahap ini, data ini diambil dari halaman utama PDDikti, yang mencakup 76 perguruan tinggi di Jakarta, masing-masing dengan lima field. Pada tahap berikutnya, data ini akan ditambahkan dari situs web perguruan tinggi swasta di Jakarta. Pengumpulan data dari halaman detail perguruan tinggi di situs PDDikti juga dilakukan, untuk mencakup informasi tentang semua perguruan tinggi, termasuk alamat, tautan ke dalam situs web, nama program studi, dan akreditasi masing-masing program. Jumlah program yang dikumpulkan akan disesuaikan dengan ketersediaan program di setiap perguruan tinggi swasta. Sebagai contoh, data tentang 63 program studi Universitas Muhammadiyah Jakarta akan diambil secara otomatis oleh robot, bersama dengan alamat detail perguruan tinggi. Robot akan menggabungkan kedua dataset menjadi satu file setelah data dikumpulkan dengan memasukkan informasi dari tahap ini ke dalam file Excel yang sama dengan tahap pertama. Pada langkah terakhir, data dikumpulkan dari situs web perguruan tinggi swasta yang terdaftar pada PDDikti. Data yang dikumpulkan meliputi latar belakang universitas, fasilitas yang tersedia, dan beasiswa yang ditawarkan. Robot secara otomatis mengumpulkan data ini dari situs web perguruan tinggi dan mengintegrasikannya ke dalam file Excel yang sama dengan tahap awal, yang mengandung data dari PDDikti. Ini adalah tahap akhir dari proses pengumpulan data (**Sílvia Moreira, 2023**).

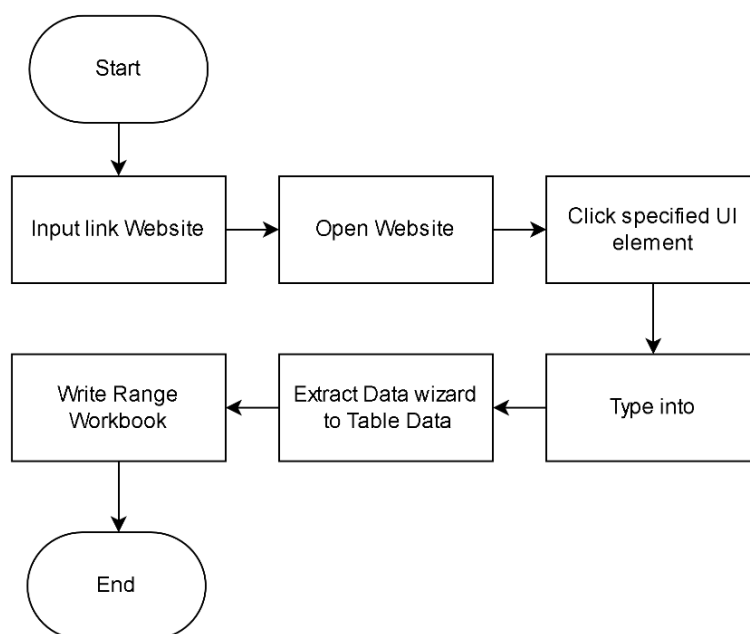
Data yang dikumpulkan kemudian diproses melalui langkah-langkah restrukturisasi, termasuk pembersihan untuk menghapus duplikat dan informasi yang tidak lengkap, serta normalisasi dan tokenisasi untuk memastikan data konsisten dan terstruktur dengan baik. Tahap ini bertujuan untuk mentransformasi dokumen teks mentah menjadi representasi yang lebih terstruktur dan siap digunakan oleh model pemrosesan data. Proses ini mencakup beberapa tahapan utama, seperti pembersihan teks untuk menghapus elemen-elemen yang tidak signifikan, normalisasi untuk menyamakan format teks, dan tokenisasi yang memecah teks menjadi unit-unit kata atau token. Selain itu, stemming dan lemmatization sering digunakan untuk mengembalikan kata-kata ke bentuk dasarnya, sehingga dapat mengurangi redundansi dalam analisis. Dengan menerapkan pemrosesan awal secara menyeluruh, teks yang dihasilkan menjadi lebih konsisten dan relevan untuk analisis lebih lanjut. Langkah ini untuk memastikan bahwa informasi penting dalam dokumen tetap terjaga, sehingga dapat mendukung model dalam memahami pola dan hubungan dalam data secara lebih akurat (**Puneet Mishra, 2020**).

Model *Long short-term memory* (LSTM) dipilih karena kemampuannya untuk menangani data sekuensial dan belajar dari pola dalam data. Model ini dilatih dengan data yang telah diproses, diuji untuk akurasi dan generalisasi, dan disesuaikan melalui tuning hyperparameter (**Federico Landi, 2021**). Pada tahap ini, sistem berupaya memahami dan mengenali pola-pola serta hubungan dalam data yang digunakan. Proses ini dilakukan dengan mengoptimalkan parameter model berdasarkan data latih yang disediakan. Model dilatih untuk mengasosiasikan input teks dengan kategori atau label yang relevan, sehingga mampu memberikan rekomendasi yang akurat dan sesuai. Tahapan pre-processing yang diterapkan pada data pelatihan mencakup pembersihan teks dan tokenisasi, yang bertujuan untuk memastikan bahwa informasi dalam data dapat dianalisis secara tepat. Prediksi dilakukan menggunakan model *Long Short-Term Memory* (LSTM), untuk menganalisis input

pengguna dan menghasilkan rekomendasi berdasarkan pola serta informasi yang telah dipelajari dari dataset. Setelah proses selesai pelatihan model selesai dan model siap untuk digunakan, lalu model akan di save dan diintegrasikan ke dalam chatbot yang dirancang untuk berinteraksi secara alami dengan pengguna.

2.1. Web Scraping

Pada tahap pengumpulan data, yaitu langkah untuk mengidentifikasi situs web perguruan tinggi swasta dan Pangkalan Data Pendidikan Tinggi (PDDikti). Proses ini melibatkan pemilihan halaman web yang memuat informasi penting mengenai universitas. Kemudian, teknik *web scraping* digunakan untuk mengotomatisasi pengambilan data dari halaman-halaman web tersebut. *Web scraping* ini mencakup penulisan skrip menggunakan alat dan pustaka seperti UiPath, yang memungkinkan program untuk menavigasi halaman web secara otomatis dan mengekstrak informasi yang dibutuhkan, Proses tahapan *web scraping* dapat dilihat pada Gambar 2.



Gambar 2. Tahapan Web Scraping Menggunakan UiPath

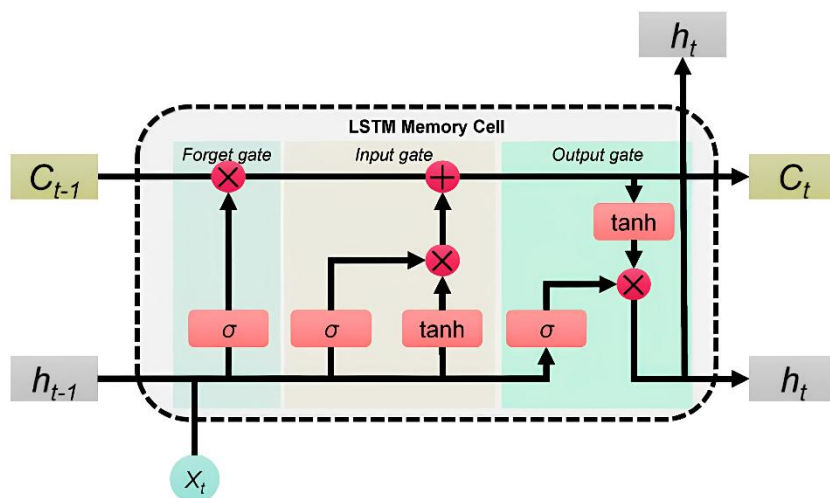
Pada tahap awal proses scraping, perangkat lunak UiPath memverifikasi bahwa semua komponen dan komponen yang dibutuhkan telah siap untuk dijalankan. Selanjutnya, sistem dimulai untuk mempersiapkan tugas-tugas scraping, termasuk mengatur parameter awal seperti lokasi penyimpanan data dan konfigurasi sumber data. Selanjutnya, pengguna memberikan input berupa URL situs web yang akan discraping. URL ini berfungsi sebagai referensi utama yang mengarahkan sistem untuk mengakses data. Input URL dimasukkan secara manual atau diatur melalui parameter otomatis untuk memungkinkan scraping berulang dari berbagai situs web secara bersamaan. Sistem secara otomatis mengakses URL situs web yang telah diberikan pada tahap sebelumnya. Untuk meluncurkan browser kompatibel, seperti Google Chrome, UiPath menggunakan aktivitas seperti "Open Browser". Sebelum melanjutkan ke langkah berikutnya, prosedur ini memastikan bahwa halaman web dimuat sepenuhnya. Pada titik ini, sistem juga memverifikasi status jaringan dan memastikan bahwa halaman web dapat diakses dengan benar (Janarthini P, 2023).

Sistem mengidentifikasi komponen antarmuka pengguna (UI) yang relevan, seperti tombol, menu dropdown, atau tautan, setelah halaman web dimuat. Element-elemen ini berinteraksi dengan sistem melalui aktivitas "Klik". Misalnya, langkah ini akan memastikan navigasi dilakukan secara otomatis hingga data yang diinginkan dapat diakses jika data target berada di dalam submenu atau memerlukan filter tertentu. Tahap ini sangat penting untuk menemukan dan menavigasi situs web yang terus berubah yang memiliki fitur interaktif yang menyembunyikan data. Pada tahap *Type Into*, sistem mengisi informasi ke komponen antarmuka pengguna tertentu, seperti kotak pencarian, formulir, atau filter. Jika pengguna ingin mencari data khusus seperti nama universitas atau program studi, sistem akan mengisi kata kunci tersebut di kotak pencarian yang relevan. Pada fitur "Data Scraping Wizard" UiPath membantu sistem mengekstrak data terstruktur, seperti tabel, daftar, atau elemen teks, dari halaman web. Pada tahap ini, sistem menemukan pola data yang konsisten, seperti kolom nama, alamat, atau akreditasi universitas. Untuk memastikan bahwa data siap untuk analisis dan pengolahan lebih lanjut, data yang diambil secara efektif disusun menjadi tabel. Setelah data diekstraksi dan disusun dalam format tabel, sistem menyimpannya ke dalam file Excel menggunakan aktivitas "Write Range". Pada tahap ini, sistem memastikan bahwa data disimpan secara terstruktur, dengan kolom yang sesuai, dan tanpa redundansi (Khan, 2020).

Pada langkah terakhir, sistem memastikan bahwa semua data yang telah diekstraksi dan disimpan dengan benar, dan browser ditutup untuk mengakhiri proses scraping. Selain itu, sistem menghasilkan laporan atau log aktivitas untuk memastikan bahwa seluruh proses berjalan sesuai rencana dan memungkinkan troubleshooting jika terjadi kesalahan. Setiap langkah ini dibuat untuk memastikan proses scraping berjalan secara otomatis, efisien, dan akurat, dan data yang diperoleh dapat digunakan untuk analisis dan pengembangan model lebih lanjut. Kombinasi aktivitas UiPath memastikan bahwa proses ini dapat disesuaikan dengan berbagai situasi pengambilan data dari situs web (Rama Krishna Debbadi, 2025).

2.2. Long Short-Term Memory (LSTM)

Long-short term Memory (LSTM) merupakan salah satu varian dari recurrent neural network (RNN) yang dirancang untuk mengatasi kelemahan RNN dalam memprediksi data berdasarkan informasi yang telah disimpan dalam jangka waktu lama. LSTM memiliki kemampuan untuk mengingat dan menghapus data lama yang sudah tidak relevan. Komponen utama struktur LSTM *forget gate*, *input gate*, *cell state*, dan *output gate* (Liu, 2022). Arsitektur pada LSTM dapat dilihat pada Gambar 3.



Gambar 3. Arsitektur LSTM

Long short-term memory (LSTM) merupakan metode yang telah terbukti berhasil dalam pembuatan chatbot. Dalam pemodelan data urutan panjang, masalah *vanishing gradient* diatasi oleh arsitektur jaringan saraf berulang LSTM (**Yetti Yuniati, 2024**). Dengan kemampuan untuk menyimpan informasi dari interaksi sebelumnya, *Long short-term memory* (LSTM) membantu chatbot menangani percakapan yang kompleks dan memberikan pengalaman pengguna yang lebih alami dan cerdas. LSTM memungkinkan chatbot untuk mengingat dan menggunakan konteks dari percakapan sebelumnya, yang memungkinkan mereka memberikan jawaban yang lebih akurat dan relevan yang didasarkan pada urutan kata dan konteks yang diperoleh dari percakapan sebelumnya. Penggunaan *Long short-term memory* (LSTM) adalah untuk mengolah data sekuensial dengan cara yang efektif dan akurat. Ini dicapai melalui penerapan metode *Long short-term memory* (LSTM). Untuk data teks sekuensial seperti informasi universitas, program studi, akreditasi, fasilitas, dan beasiswa. Dengan menggunakan metode LSTM, model akan memahami dan mengklasifikasikan sejumlah pertanyaan berdasarkan kategori yang telah ditentukan sebelumnya. Kategori-kategori ini termasuk informasi tentang program studi, biaya pendaftaran, dan biaya kuliah (**Fahmi Yusron F, 2024**). LSTM cocok untuk menangkap pola hubungan jangka panjang. Pertama, data yang telah dikumpulkan dan disusun dari situs Kementerian Pendidikan, Kebudayaan, Riset, dan Teknologi (KEMDIKBUDRISTEK) dan perguruan tinggi di Jakarta. Selanjutnya, data diolah untuk membuat formatnya siap untuk pemrosesan lanjutan. Pra-pemrosesan teks melibatkan tokenisasi, stemming, dan pengurangan huruf menjadi lebih kecil setelah data dikumpulkan. Sementara tokenisasi membagi teks menjadi kata-kata atau token, dan stemming mengubah setiap kata menjadi bentuk dasarnya untuk mendapatkan format yang terstruktur, pengubahan huruf menjadi kecil memastikan bahwa teks tetap seragam dan tidak dibedakan berdasarkan kapitalisasi.

Teknik embedding kemudian digunakan untuk mengubah data teks yang telah diproses menjadi representasi numerik. Ini mengubah kata-kata menjadi vektor angka dan memungkinkan model untuk memahami semantik setiap kata dalam kalimat. Selanjutnya, model LSTM terdiri dari beberapa lapisan utama. Lapisan embedding menerima data masukan, lapisan LSTM menangani urutan data dan mempertahankan konteks kata, dan lapisan dropout mencegah overfitting (**P Anki, 2020**). Pada lapisan output, fungsi aktivasi softmax digunakan untuk mengklasifikasikan atau memberikan probabilitas pada setiap kelas berdasarkan prediksi model. Selanjutnya, model dilatih dengan metode backpropagation dan optimisasi Adam. Dalam proses ini, model akan mengubah bobot pada setiap lapisan untuk mengurangi kesalahan prediksi. Proses pelatihan ini berlangsung selama beberapa kali hingga model mencapai tingkat akurasi terbaik (**Shruti Patil, 2020**). Setelah pelatihan, model LSTM dievaluasi menggunakan data uji untuk memastikan performanya dalam melakukan prediksi pada data baru. Ini dilakukan dengan mengukur akurasi, presisi, dan recall untuk mengetahui seberapa baik model mampu mengklasifikasikan data universitas. Setelah berhasil melalui tahap evaluasi, model dimasukkan ke dalam sistem rekomendasi kampus untuk memberikan rekomendasi yang sesuai dengan preferensi pengguna.

2.3. Integrasi Website

Setelah model melalui tahap pelatihan, langkah berikutnya adalah memasukkan model LSTM ke dalam website. Dengan menggunakan antarmuka interaktif, pengguna dapat mengaksesnya. Untuk menyelesaikan integrasi ini, backend dibangun menggunakan Flask, sebuah framework Python yang ringan namun kuat untuk pengembangan aplikasi web (**Yaganteeswarudu, 2020**). Pada bagian frontend dibangun menggunakan React.js, library JavaScript yang populer yang digunakan untuk membuat antarmuka pengguna yang responsif dan dinamis (**Archana Bhalla, 2020**).

Untuk memulai integrasi ini, API aplikasi dibuat menggunakan Flask. Model LSTM yang telah dilatih digunakan sebagai layanan backend di Flask yang menerima permintaan dari frontend, memproses data input, dan mengembalikan rekomendasi kampus yang dihasilkan oleh model. Pertama, Flask diatur untuk memuat model LSTM yang sudah dilatih, dan endpoint API dibuat untuk menerima data dari pengguna. Setiap kali pengguna mengirimkan input melalui frontend, data dikirim ke server Flask melalui permintaan HTTP. Selanjutnya, model LSTM mengolah input tersebut untuk menghasilkan rekomendasi. Setelah proses prediksi selesai, rekomendasi hasil dikirimkan kembali dalam format JSON yang dapat dipahami frontend dengan mudah (**Mochammad Fariz Syah Lazuardy, 2022**).

React.js digunakan di frontend untuk membuat antarmuka pengguna yang interaktif dan responsif yang memungkinkan pengguna mengubah preferensi kampus mereka. Untuk mengumpulkan informasi berdasarkan kriteria yang diinginkan pengguna, antarmuka ini akan menggunakan form input, seperti input teks. Antarmuka ini dapat dirancang agar mudah digunakan dan memudahkan pengguna mengisi preferensi mereka dengan menggunakan fitur React. Data akan dikirim ke backend Flask menggunakan library seperti Axios di React setelah preferensi dimasukkan. Model akan memproses input, membuat rekomendasi, dan mengembalikan hasil ke frontend setelah backend menerima permintaan (**Jalolov, 2024**).

React.js dapat mengubah hasil yang diterima dari Flask menjadi komponen UI yang menarik dan mudah dipahami untuk menampilkan hasil rekomendasi kampus dalam bentuk teks yang berisi informasi seperti nama kampus, program studi yang direkomendasikan, akreditasi, lokasi, dan beasiswa yang tersedia. Tampilan rekomendasi dapat disesuaikan secara responsif sehingga sesuai dengan berbagai perangkat, seperti desktop, dengan menggunakan CSS atau framework antarmuka pengguna seperti Bootstrap atau Material-UI (**Sairam Vakkalanka, 2020**). Hasilnya adalah sistem rekomendasi kampus yang terintegrasi dan mudah diakses melalui website. Dengan menggunakan hasil analisis model LSTM, pengguna dapat dengan cepat dan mudah mendapatkan rekomendasi kampus yang sesuai dengan preferensi mereka. Integrasi ini memungkinkan sistem rekomendasi kampus berfungsi dengan lancar, dari backend Flask hingga frontend React.js, sehingga memberikan pengalaman pengguna yang optimal dan efisien.

3. HASIL DAN PEMBAHASAN

Jumlah total 76 perguruan tinggi swasta yang terletak di wilayah Jakarta telah ditemukan melalui penggunaan data dari situs resmi perguruan tinggi dan PDDikti. Setiap universitas ini memiliki 12 kolom informasi yang memberikan detail lengkap tentang karakteristik, fasilitas, dan layanan yang ditawarkan oleh perguruan tinggi tersebut. Kolom-kolom ini mencakup berbagai elemen penting seperti nama universitas, alamat, program studi yang tersedia, akreditasi, biaya kuliah, fasilitas pendukung seperti laboratorium dan perpustakaan, serta informasi lainnya yang terkait dengan kebutuhan calon mahasiswa.

Untuk memudahkan pengolahan dan penyimpanan, data hasil restrukturisasi yang sebelumnya disimpan dalam format Excel diubah menjadi format JSON. Data ini mencakup informasi penting tentang setiap perguruan tinggi swasta yang ada di Jakarta. Informasi ini mencakup 12 elemen utama, termasuk nama perguruan tinggi, lokasi, akreditasi, biaya kuliah, fasilitas, dan program studi yang ditawarkan. Dalam JSON, setiap perguruan tinggi direpresentasikan sebagai objek, dan setiap atributnya disimpan dalam bentuk pasangan *key-value*, seperti "Alamat_Universitas", "akreditasi", dan "biaya_kuliah".

Proses untuk membangun dataset yang terstruktur dari file JSON yang berisi informasi tentang berbagai intent, pola pertanyaan, dan respons. Setiap tujuan yang ada pada data diwakili oleh pola pertanyaan (pattern) yang sering diajukan oleh pengguna, bersama dengan respons yang relevan untuk memberikan jawaban yang tepat. Dengan menggunakan Pandas DataFrame, data disusun dalam bentuk tabel. Kolom-kolom mengandung pola pertanyaan (pattern), tag (kategori dari intent), dan respons (jawaban yang sesuai dengan pola pertanyaan). Dataset ini terdiri dari 2922 total *patterns*, masing-masing menggambarkan pola pertanyaan yang berbeda yang sering diajukan, dan 750 *unique tags*, masing-masing menggambarkan kategori intent yang berbeda. Berkat pengolahan data, chatbot dapat mengenali dan mengkategorikan pertanyaan pengguna ke dalam kategori tertentu dan kemudian memberikan jawaban yang tepat. Tabel 1 menunjukkan data frame yang telah disusun.

Tabel 1. Hasil Data Frame

Patterns	Tag	Responses
hallo	greeting	Hai! Unibot di sini. Ada yang bisa saya bantu mengenai informasi kampus?
Siapa Unibot?	Unibot	Unibot adalah chatbot yang khusus membantu kamu menemukan info kampus swasta, mulai dari fasilitas, jurusan, hingga informasi beasiswa.
Unibot, yang buat kamu siapa sih?	pencipta_Unibot	Saya diciptakan oleh mahasiswa dari Universitas Tarumanagara (UNTAR).
See you	goodbye	See u! Senang bisa membantu

Pada Tabel 1 memberikan gambaran tentang struktur data yang digunakan untuk melatih chatbot, dengan kolom utama meliputi Patterns, Tag, dan Responses. Kolom Patterns berisi berbagai pola pertanyaan yang mewakili input potensial dari pengguna, seperti "hallo" untuk sapaan, "Siapa Unibot?" untuk menanyakan identitas chatbot, hingga "See you" untuk mengakhiri percakapan. Pola-pola ini menggambarkan berbagai cara pengguna berinteraksi dengan sistem. Selain itu, kolom Tag berfungsi sebagai label atau kategori yang mengkategorikan pola berdasarkan konteks atau tujuan percakapan. Misalnya, tag selamat datang digunakan untuk menyapa, Unibot digunakan untuk mengidentifikasi chatbot, dan goodbye digunakan untuk mengucapkan selamat tinggal. Kolom Responses berisi respons yang telah dirancang secara khusus untuk menjawab pola pertanyaan berdasarkan kategori yang sesuai, seperti jawaban untuk pertanyaan "Hai! Unibot di sini." Untuk sapaan, "Ada yang bisa saya bantu mengenai informasi kampus?"

Sebelum digunakan dalam pelatihan model, data dalam tabel ini kemudian diproses melalui beberapa tahap penting. Pertama, tokenisasi dilakukan, yaitu pembagian setiap pola menjadi unit kata atau token yang dapat dipahami model. Untuk menjamin bahwa proses berjalan dengan benar, label khusus diberikan pada kata-kata yang tidak dikenali dalam kamus. Selanjutnya, padding digunakan. Dalam proses ini, nilai kosong ditambahkan untuk mengubah pola-pola dengan panjang yang berbeda menjadi panjang yang sama. Untuk memenuhi kebutuhan input konsisten pada model pembelajaran mesin, langkah ini penting. Setelah itu, encoding dilakukan pada tag, yang merupakan proses mengubah kategori teks menjadi representasi numerik. Ini memungkinkan model untuk dengan mudah mengidentifikasi dan membedakan berbagai kategori intent.

Proses ini menghasilkan representasi numerik dari pola pertanyaan dan tag. Representasi numerik ini siap digunakan sebagai data pelatihan untuk model pembelajaran mesin. Pola input pengguna dapat diidentifikasi dengan data ini, dikategorikan ke dalam intent yang

sesuai, dan diberikan respons yang sesuai berdasarkan kategori tersebut. Langkah-langkah ini sangat penting untuk memastikan chatbot berfungsi dengan baik dan memenuhi kebutuhan pengguna.

Tabel 2. Hasil Output Dataset untuk Model Pelatihan

Vocabulary Size	Max Sequence Length	Number of Classes
318	12	750

Pada Tabel 2. Data yang telah diproses dan dipersiapkan untuk model ditampilkan dengan menggunakan fitur ini. Pertama, ukuran kosakata, juga dikenal sebagai *Vocabulary size*, menunjukkan berapa banyak kata yang berbeda yang ditemukan dalam dataset setelah proses tokenisasi. Ukuran ini menunjukkan kompleksitas bahasa yang dipelajari oleh model. Dalam hal ini, model akan memproses 318 kata unik. Kedua *Max sequence length* mengacu pada panjang urutan tertinggi dari token yang digunakan oleh model sebagai input. Dengan panjang urutan yang diseragamkan menjadi dua belas token, model dapat memproses setiap pola pertanyaan dengan konsistensi. Terakhir, Jumlah kelas mengacu pada jumlah kategori atau tag yang ada dalam dataset. Dalam kasus ini, dataset memiliki empat kategori atau tag khusus yang digunakan untuk mengkategorikan pola pertanyaan ke dalam kategori yang sesuai. Dengan data ini, model dapat mengidentifikasi maksud atau niat dari pertanyaan pengguna dan memberikan respons yang sesuai dengan kategori yang telah ditentukan. Secara keseluruhan, pengolahan data yang dilakukan dalam tahap ini memastikan bahwa model dapat bekerja dengan data yang sudah terstruktur, seragam, dan siap untuk digunakan dalam proses pelatihan.

Tabel 3 menunjukkan hasil pelatihan model. Tabel tersebut menunjukkan perkembangan kinerja model, Setiap epoch berlangsung selama sekitar 5 detik dan terdiri dari 313 langkah. Accuracy dan loss adalah parameter yang dievaluasi. Pada epoch pertama pelatihan, akurasi model sebesar 0.5459 dengan nilai loss sebesar 0.4412. Seiring berjalannya waktu, kinerja model menunjukkan yang lebih baik seiring berjalannya waktu. Pada epoch 100, akurasi meningkat menjadi 0.7255 meskipun nilai kehilangan sedikit naik menjadi 0.4422. Pada epoch 150, bagaimanapun, akurasi hanya meningkat tipis menjadi 0.7285, tetapi nilai loss turun secara signifikan menjadi 0.3248. Ini menunjukkan bahwa kemampuan model untuk meningkatkan kemampuan model dalam meminimalkan kesalahan prediksi.

Pada epoch 200 hingga 300, model terus menunjukkan perbaikan yang konsisten. Akurasi bertambah dari 0.7755 pada epoch 200 menjadi 0.9668 pada epoch 300, dengan penurunan signifikan pada nilai loss, yakni dari 0.4456 menjadi 0.2335. Tren ini mengindikasikan bahwa model semakin mampu memprediksi dengan lebih akurat dan lebih sedikit kesalahan. Pada tahap pelatihan lebih lanjut, yakni dari epoch 350 hingga 450, akurasi model mencapai nilai maksimal 0.9988 pada epoch 450 dengan nilai loss yang sangat rendah sebesar 0.0195. Hal ini mencerminkan bahwa model telah sepenuhnya belajar dari data pelatihan dan menghasilkan performa yang optimal. Keseluruhan hasil menunjukkan bahwa pelatihan model berlangsung secara efektif dengan peningkatan performa yang signifikan dari setiap iterasi. Namun, akurasi yang mencapai nilai maksimum 0.9988 dan loss yang mendekati nol juga dapat menjadi indikasi bahwa model mengalami overfitting, yaitu model terlalu menyesuaikan diri dengan data pelatihan dan mungkin kurang mampu melakukan generalisasi terhadap data baru. Evaluasi lebih lanjut dengan data validasi atau data uji diperlukan untuk memastikan model dapat mempertahankan performa yang baik di luar data pelatihan.

Tabel 3. Hasil Pelatihan Model

Epoch	Steps	Duration	Accuracy	Loss
50	313	5s	0.5459	0.4412
100	313	5s	0.7255	0.4722
150	313	5s	0.7285	0.3248
200	313	5s	0.7755	0.4456
250	313	5s	0.7863	0.3638
300	313	5s	0.9668	0.2335
350	313	5s	0.9780	0.2252
400	313	5s	0.9870	0.1262
450	313	5s	0.9988	0.0195

Hasil pengujian chatbot dapat dilihat pada Tabel 4. Yang menunjukkan bahwa Chatbot sangat akurat dalam menanggapi perintah pengguna, seperti yang ditunjukkan oleh hasil pengujian, yang dapat dilihat pada Tabel 4. Tingkat kepercayaan adalah bagian dari logika chatbot yang bertugas memberikan respons berdasarkan prediksi model. Dalam proses ini, chatbot akan memeriksa apakah tingkat kepercayaan prediksi model lebih tinggi dari threshold yang telah ditentukan, yang digunakan untuk memastikan bahwa hanya respons yang memiliki tingkat kepercayaan tinggi yang dikirimkan ke pengguna, yang meningkatkan akurasi dan keandalan jawaban.

Tabel 4. Hasil Pengujian Jawaban Chatbot

<i>Input</i>	<i>Response</i>	<i>Confidence</i>
hi	Hai! Selamat datang di layanan Unibot. Ada info kampus yang ingin kamu cari?	100.00%
Unibot, yang buat kamu siapa sih?	Penciptaku adalah mahasiswa dari Universitas Tarumanagara. Berkat ide-ide nya, saya bisa hadir dan membantu kamu saat ini!	100.00%
Berapa biaya kuliah di Universitas Muhammadiyah Jakarta?	Biaya per semester di Universitas Muhammadiyah Jakarta (UMJ) berkisar antara Rp275.000 hingga Rp21.600.000, tergantung program studi yang dipilih.	100.00%
Di mana alamat Universitas Bung Karno?	Jalan Kimia No. 20, Kec. Menteng, Central Jakarta City, Prov. D.K.I. Jakarta	100.00%
bye	Bye!	100.00%

Chatbot akan melihat dalam respons yang berisi pasangan tag dan daftar respons yang relevan untuk setiap tag, apakah tag prediksi yang dibuat oleh model tersedia. Jika tag ditemukan, chatbot akan mengambil daftar respons yang sesuai dengan tag tersebut. Jika tingkat kepercayaan lebih tinggi dari threshold, prosedur ini dilakukan. Selain itu, fungsi pemilihan acak digunakan untuk memilih salah satu jawaban dari daftar secara acak. Tujuan dari fungsi pemilihan acak ini adalah untuk membuat jawaban chatbot lebih bervariasi, sehingga pengguna tidak selalu menerima jawaban yang sama meskipun inputnya sama. Chatbot kemudian akan mengirimkan jawaban yang terdiri dari tiga bagian: respons yang dipilih, tingkat kepercayaan prediksi, dan tag yang relevan. Namun, jika tingkat kepercayaan chatbot tidak mencapai threshold atau jika tag prediksi tidak ditemukan dalam kamus respons, chatbot akan memberikan respons default. Dalam hal ini, respons berupa pesan yang menyatakan bahwa chatbot tidak yakin dengan jawaban atas pertanyaan yang diajukan, disertai dengan tingkat kepercayaan dan tag yang ditetapkan sebagai "unknown". Meskipun chatbot mungkin tidak selalu dapat memberikan jawaban yang tepat, jawaban default mereka dirancang untuk membuat pengalaman yang positif bagi pengguna.

Secara keseluruhan proses ini dirancang ini dirancang untuk memastikan chatbot memberikan respons relevan dan akurat jika tingkat kepercayaannya tinggi. Threshold dan pemilihan respons acak membuat chatbot lebih fleksibel dan dinamis, sementara respons default memastikan kesopanan ketika jawabannya tidak pasti. Chatbot mampu menjawab pertanyaan spesifik, seperti " Berapa biaya kuliah di Universitas Muhammadiyah Jakarta?" dengan tingkat kepercayaan 100%, serta pertanyaan berbasis konteks, seperti "Unibot, yang buat kamu siapa sih?" dengan kepercayaan yang sama. Pertanyaan terkait data, seperti biaya kuliah di Universitas Muhammadiyah Jakarta atau alamat Universitas Bung Karno, juga dijawab dengan sangat akurat dengan tingkat 100% kepercayaan. Hasil pengujian membuktikan bahwa chatbot andal dalam membantu calon mahasiswa menemukan informasi dengan cepat dan akurat.

4. KESIMPULAN

Dengan menggunakan model *Long short-term memory* (LSTM), penelitian ini berhasil mengembangkan sistem rekomendasi kampus berbasis chatbot. Hasil pengujian menunjukkan bahwa sistem rekomendasi kampus berbasis LSTM efektif memahami pola input pengguna dan memberikan rekomendasi akurat dengan tingkat kepercayaan 100%. Chatbot mampu merespons dengan cepat dan informatif, mendukung interaksi real-time untuk informasi kampus swasta di Jakarta. Penggunaan UiPath untuk scraping data dari situs universitas meningkatkan akurasi dan efisiensi proses pengumpulan data. Dengan tingkat akurasi model LSTM sebesar 99.88%, sistem ini terbukti unggul dalam menangkap konteks percakapan dan memberikan rekomendasi. Peningkatan akurasi masih dapat dilakukan melalui optimisasi model, penambahan data, atau integrasi fitur-fitur baru untuk meningkatkan pengalaman pengguna. Selain itu, respons cepat dan relevan yang dihasilkan chatbot menunjukkan potensinya dalam membantu calon mahasiswa menemukan informasi kampus dengan lebih mudah dan efisien.

UCAPAN TERIMA KASIH

Terima kasih kepada Universitas Tarumanagara yang telah menyediakan fasilitas dan sumber daya yang diperlukan untuk mendukung penelitian ini. Selain itu, penulis juga mengucapkan terima kasih kepada pihak-pihak yang telah memberikan dukungan dan kontribusi dalam proses penelitian ini. Tanpa bimbingan beliau, penelitian ini tidak akan dapat diselesaikan dengan baik.

DAFTAR RUJUKAN

- Ahmad R. Pratama, R. R. (2022). Model Klasifikasi Calon Mahasiswa Baru Untuk Sistem Rekomendasi Program Studi Sarjana Berbasis Machine Learning. *Jurnal Teknologi Informasi Dan Ilmu Komputer (Jtiik)*, 9(4), 725-734.
- Amaliya, R. (2019). Faktor-Faktor Yang Mempengaruhi Keputusan Mahasiswa Memilih Pendidikan Pada Program Studi Pendidikan Ekonomi Fe Uny. *Jurnal Pendidikan Dan Ekonomi*, 8(1).
- Andrew Ciayandi, V. C. (2020). Retrieval Based Chatbot On Tarumanagara University With Multilayer Perceptron. *Iop Conf. Series: Materials Science And Engineering*.

- Antonius Randy Arjun, Z. B. (2022). Chatbot-Based Movie Recommender System With Latent Semantic Analysis On Telegram Platform Using Dialogflow . *Journal Of Computer System And Informatics (Josy)*, 3(4), 162-170.
- Archana Bhalla, S. G. (2020). Present Day Web-Development Using Reacjs. *International Research Journal Of Engineering And Technology*, 7(05).
- Fahmi Yusron F, A. K. (2024). Chatbot Informasi Penerimaan Mahasiswa Baru Menggunakan Metode Fasttext Danlstm. *Jou Rnal Of Applied Com Puter Science And Technology (Jacost)*, 5(1), 33-39.
- Federico Landi, L. B. (2021). Working Memory Connections For Lstm. *Sciencedirect*, 144, 334-341.
- Ferry Saputra, R. M. (2024). Chatbot Berbasis Whatsapp Teknik Informatika Universitas Palangkaraya: Rules Based System. *Jurnal Teknologi Informatika Dan Komputer*, 10(1).
- Heryati, D. I. Z. (2023). Aplikasi Chatbot Untuk Penerimaan Mahasiswa Baru Universitas Indo Global Mandiri Menggunakan Deep Learning. *Journal Of Intelligent Networks And Iot Global*, 1(1).
- Hilya Anbiyani Fitri Muhyidini, L. V. (2023). Pengembangan Chatbot Untuk Meningkatkan Pengetahuan Dan Kesadaran Keamanan Siber Menggunakan Long Short-Term Memory. . *Jurnal Informatika Dan Rekayasa Perangkat Lunak*, 5(2), 152-161.
- Jalolov, T. (2024). Frontend And Backend Developer Difference Advantages. *Multidisciplinary Journal Of Science And Technology*, 4(2).
- Janarthini P, A. P. (2023). Real Estate Web Scraping Using Uipath In Rpa. . *International Journal Of Research Publication And Reviews*, 4(6).
- Jiaqi Meng, C. L. (2023). Rnn-Lstm-Based Model Predictive Control For A Corn-To-Sugar Process. *Processes*.
- Khan, S. (2020). Comparative Analysis Of Rpa Tools-Uipath, Automation Anywhere And Blueprism. *International Journal Of Computer Science And Mobile Applications*, 8(11), 1-6.
- Khang Nhut Lam, N. N. (2020). Building A Chatbot On A Closed Domain Using Rasa. 144-148.
- Laras Wiranda, M. S. (2019.). Penerapan Long Short Term Memory Pada Data Time Series Untuk Memprediksi Penjualan Produk Pt. Metiska Farma. *Jurnal Nasional Pendidikan Teknik Informatika*, 8(6).

- Liu, R. J. (2022). Forecasting The Volatility Of Specific Risk For Stocks With Lstm. *Procedia Computer Science.*, 111-114.
- Mochammad Fariz Syah Lazuardy, ,. D. (2022). Modern Front End Web Architectures With React.Js And Next.Js. *International Research Journal Of Advanced Engineering And Science*, 7(1), 132-141.
- Mohinish Daswani, K. D. (2020). Collegebot: A Conversational Ai Approach To Help Students Navigate College. *Computer Engineering Department*.
- Nas, C. (2021). Data Mining Prediksi Minat Calon Mahasiswa Memilih Perguruan Tinggi Menggunakan Algoritma C4.5. *Jurnal Manajemen Informatika (Jamika)*, 11(2).
- P Anki, A. B.-A. (2020). Intelligent Chatbot Adapted From Question And Answer System Using Rnn-Lstm Model. *International Conference On Science & Technology*.
- Puneet Mishra, A. B. (2020). New Data Preprocessing Trends Based On Ensemble Of Multiple Preprocessing Techniques. *Sciencedirect*, 132.
- Rama Krishna Debbadi, O. B. (2025). Enhancing Cognitive Automation Capabilities With Reinforcement Learning Techniques In Robotic Process Automation Using Uipath And Automation Anywhere. *International Journal Of Science And Research Archive*, 2(14), 733-752.
- Sairam Vakkalanka, R. P. (2020). A Framework For Evaluating The Quality Of Academic Websites. . *International Conference On Computational Intelligence And Informatics*.
- Shruti Patil, V. M. (2020). Lstm Based Ensemble Network To Enhance The Learning Of Long-Term Dependencies In Chatbot. *International Journal For Simulation And Multidisciplinary Design Optimization*.
- Sílvia Moreira, H. S. (2023). Process Automation Using Rpa – A Literature Review. *Procedia Computer Science*, 244-254.
- Tiara Eka Putri, G. R. (2024). Penerapan Chatbot Sebagai Alat Pembelajaran Untuk Pengembangan Pendidikan Karakter. *Indonesian Journal Of Computer Science And Engineering (Ijcse).*, 1(1).
- Tjut Awaliyah Zuraiyah, D. K. (2020). Implementasi Chatbot Pada Pendaftaran Mahasiswa Baru Menggunakan Recurrent Neural Network. *Jurnal Ilmiah Teknologi Dan Rekayasa*.
- Xianyun Wen, W. L. (2023). Time Series Prediction Based On Lstm-Attention-Lstm Model. *Ieee Access*.
- Yaganteeswarudu, A. (2020). Multi Disease Prediction Model By Using Machine Learning And Flask Api. *International Conference On Communication And Electronics Systems*.

Yetti Yuniati, F. A. (2024). Pengembangan Chatbotbatik Menggunakan Metode Long Short-Term Memory. *Digital Transformation Technology (Digitech)*, 4(2).