

Development of a Deployment System Architecture for a Flask-Based Chatbot Using an LSTM NLP Model for Customer Service Question & Answer

David Ramantya Mukti ^{1*}, Abu Salam ^{2*}

* Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Dian Nuswantoro

111202013016@mhs.dinus.ac.id ¹, abu.salam@dsn.dinus.ac.id ²

Article Info

Article history:

Received 2025-03-18

Revised 2025-04-24

Accepted 2025-05-06

Keyword:

E-commerce,
Chatbot,
Natural Language Processing (NLP),
Deep Learning (DL),
Generative Text Processing,
Customer Service.

ABSTRACT

In the past two decades, the rapid growth of e-commerce has significantly transformed global business practices. E-commerce has not only revolutionized the retail industry but also positively impacted businesses and consumer experiences. The ease of online shopping enables users to select products at more competitive prices. Amidst these changes, human-computer interactions have increasingly evolved toward natural conversations through Natural Language Processing (NLP). This study aims to develop a chatbot utilizing Long Short-Term Memory (LSTM) technology as a medium for e-commerce customer service. The dataset used for chatbot development is in JSON format and consists of 580 entries spanning 38 categories or classes. Data processing involves several preprocessing stages, including case folding, lemmatization, tokenization, and padding. The model is developed using a bidirectional LSTM and GRU architecture, followed by regularization techniques to enhance performance. Evaluation results show the model achieves 90% training accuracy and 63% validation accuracy with an F1-score of 62%. While there are indications of overfitting, the observed differences are not statistically significant, indicating the model remains capable of providing reliable responses. Additionally, the model is integrated into a Flask-based web application with an interactive interface to facilitate user access. This study demonstrates that LSTM is effective in addressing vanishing gradient problems.



This is an open access article under the [CC-BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.

I. PENDAHULUAN

Kemajuan zaman telah mempercepat perkembangan teknologi, khususnya di ranah kecerdasan buatan. Salah satu pencapaian signifikan dalam bidang ini adalah kemampuan mesin interaktif alami untuk menjadi manusia, dilakukan oleh Teknologi *Natural Language Processing* (NLP). Teknologi NLP telah digunakan dalam berbagai aplikasi, termasuk terjemahan bahasa melalui terjemahan *Google*, asisten virtual seperti *Siri*, analisis bahasa dan alat yang dapat disesuaikan seperti *Grammarly*, serta dalam pengembangan *chatbot*.

Maraknya *e-commerce* dalam dua dekade terakhir memiliki dampak besar pada masyarakat dan bagaimana cara bisnis dilakukan secara global. Selain merevolusi industri ritel, industri ini memiliki dampak positif pada bisnis dan konsumen secara personal. Memungkinkan orang untuk berbelanja online di seluruh dunia. Pengguna telah lebih bergantung pada *e-commerce* daripada sebelumnya dalam

beberapa tahun terakhir, dan platform seperti *Amazon* dengan sangat mudah menggeser perusahaan besar seperti *Walmart* [1]. Keuntungan termasuk memiliki opsi barang yang lebih beragam dengan harga terjangkau dibandingkan toko lokal, karena mereka dapat berbelanja di mana saja di dunia dan memanfaatkan nilai tukar mata uang dan perbedaan ekonomi di seluruh dunia.

Interaksi antara manusia dan komputer telah menjadi area penelitian yang menarik bagi banyak peneliti selama bertahun-tahun. Salah satu pekerjaan yang bisa dilakukan oleh mesin atau komputer seperti manusia adalah melakukan percakapan dengan manusia. Hal ini dilakukan dengan metode *Natural Language Processing* (NLP). *Natural Language Processing* (NLP) adalah cabang dari *AI* yang berkaitan dengan interaksi antara komputer atau mesin dan manusia dengan menggunakan bahasa alami [2]. NLP banyak digunakan untuk penerjemah bahasa seperti *Google Translate*, personal asisten seperti *Siri*, pemeriksa keakuratan

bahasa seperti Grammarly, ataupun *chatbot* yang telah menjadi mekanisme populer untuk melakukan komunikasi sejenis itu. Kemajuan teknologi dalam bidang Kecerdasan Buatan (Artificial Intelligence) terus mengalami peningkatan yang signifikan dan semakin sering diterapkan dalam berbagai aspek kehidupan sehari-hari, termasuk hingga saat ini[3]. Kini, Kecerdasan Buatan secara masif dimanfaatkan dalam beragam sektor untuk memberikan respons terhadap pertanyaan, teknologi ini mampu menangani percakapan yang bervariasi, Mulai dari yang rumit hingga yang mudah. Layanan bisnis adalah salah satu industri yang menggunakan teknologi kecerdasan buatan. *Chatbot* adalah contoh umum penggunaan *AI* dalam konteks bisnis[4]. *Chatbot* adalah program komputer yang membuat interaksi ini menjadi mudah dan menarik. Metode kecerdasan buatan yang ada menunjukkan kinerja yang kurang memuaskan dalam memberikan respons yang paling tepat terhadap pertanyaan pengguna[5].

Maka dari itu, melalui penelitian ini perusahaan dapat memanfaatkan *chatbot* untuk meningkatkan kualitas pelayanan serta menyediakan jawaban atas pertanyaan yang dibutuhkan oleh pelanggan atau calon pelanggan. maka dari itu, melalui penelitian ini perusahaan dapat memanfaatkan *chatbot* untuk meningkatkan kualitas pelayanan serta menyediakan jawaban atas pertanyaan yang dibutuhkan oleh pelanggan atau calon pelanggan. Dengan perkembangan teknologi *AI* (*Artifical Intellegent*), Pada saat ini, *AI* banyak digunakan dalam berbagai bidang. *Chatbot* adalah salah satu bidang yang paling sering digunakan *AI* dalam bisnis. Program yang membantu orang berbicara dengan orang lain melalui perintah suara, teks, atau bisa keduanya[6].

Metode belajar mendalam yang sesuai untuk data berurutan adalah algoritma *LSTM* (*Long Short-Term Memory*), yang digunakan dalam penelitian ini. *Deep Learning* adalah cabang dari *Machine Learning* yang algoritmanya terinspirasi oleh struktur otak manusia. Struktur-struktur ini dikenal sebagai Jaringan Saraf Tiruan (*Artificial Neural Networks/ANN*)[7]. *Deep Learning* merupakan sektor penelitian yang sedang berkembang dalam penelitian *machine learning (ML)*. Ini melibatkan berbagai lapisan tersembunyi dan jaringan saraf tiruan. *Deep Learning (DL)* sendiri adalah bagian dari *Machine Learning* yang bergantung pada serangkaian algoritma yang bertujuan untuk menunjukkan refleksi tingkat tinggi dalam data. *Deep Learning* digunakan dalam berbagai bidang untuk mencapai berbagai tingkat pemikiran, seperti ekstraksi fitur suara, teks, gambar, dan sebagainya[8].

Namun, merancang dan membangun sistem generative text processing untuk *chatbot* bukanlah tugas yang mudah. Ini memerlukan pemahaman mendalam tentang algoritma dan teknik *Machine Learning*, serta pengetahuan tentang bagaimana cara kerja *chatbot*. Oleh karena itu, penelitian ini bertujuan untuk mengisi celah ini dengan merancang dan membangun sistem pemrosesan teks generative untuk *chatbot* berbasis *Deep Learning*[9]. Penelitian ini diharapkan dapat membantu peneliti lain dalam bidang teknologi *chatbot* dan e-

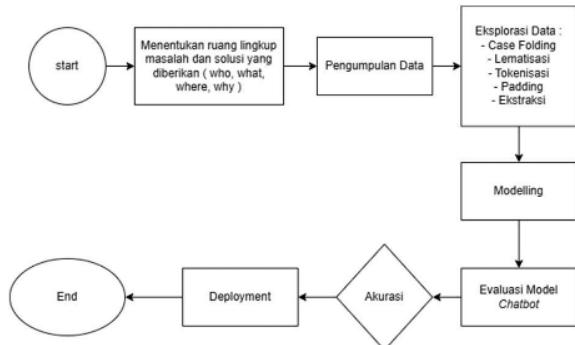
commerce[10]. Selain itu, temuan ini dapat diterapkan ke customer service bisnis sejenis yang membutuhkan layanan yang cepat dan efisien. juga bagi peneliti yang tertarik dalam bidang yang sama, serta membantu dalam pengembangan *chatbot* yang lebih canggih dan responsif dengan kombinasi metode lain oleh peneliti selanjutnya[11].

Penelitian yang dilakukan oleh R Subekti[12]. mengembangkan sebuah *chatbot* untuk Pertanyaan yang Sering Diajukan (FAQ) yang bertujuan memberikan jawaban atas pertanyaan mahasiswa terkait aspek akademik di Institut Bisnis dan Informatika Kosgoro 1957. Dalam melatih model menggunakan metode *LSTM*, dicapai akurasi sebesar 99,20% setelah 90 kali pengulangan (epoch). Long Short Term Memory (LSTM) adalah salah satu jenis Recurrent Neural Network (RNN) yang dirancang untuk mengatasi masalah vanishing gradient dan long dependency problem yang sering terjadi pada RNN[13]. LSTM merupakan inovasi dari RNN yang ditujukan untuk menangani data berurutan (temporal). [14]Dengan memanfaatkan struktur gates dan cells sebagai kontrol tambahan, LSTM mampu mengatasi kelemahan RNN dalam mengelola aliran informasi, sehingga meningkatkan performa dalam jangka waktu yang lebih lama[15]. Setiap cell LSTM dilengkapi dengan tiga gate utama: forget gate, input gate, dan output gate, yang berfungsi untuk mengatur aliran informasi di dalam cell tersebut[16]. Ada banyak metode yang dapat digunakan untuk membuat *chatbot*, seperti Recurrent Neural Network (RNN) dan Long Short-Term Memory (LSTM)[3], serta *Natural Language Processing (NLP)*. Untuk meningkatkan efisiensi dalam pemrosesan *Natural Language Processing*, pengguna dapat memanfaatkan metode *Long Short-Term Memory*, yang merupakan algoritma yang dirancang untuk mengolah data dalam format input dan menghasilkan output dalam bentuk data berurutan. Metode ini merupakan evolusi dari *Recurrent Neural Network*. Metode ini telah terbukti efektif dalam memproses urutan data yang panjang dan kompleks, menjadikannya sangat cocok untuk pengembangan *chatbot* yang memerlukan pemahaman konteks percakapan yang mendalam. Penelitian ini diharapkan dapat memberikan kontribusi penting dalam bidang teknologi *chatbot* dan e-commerce. Hasil penelitian ini diharapkan dapat digunakan sebagai panduan bagi peneliti lain dalam bidang yang sama. Mereka juga dapat diterapkan pada jenis bisnis yang membutuhkan layanan *chatbot* ini. Peneliti lain yang tertarik dalam bidang yang sama juga dapat menggunakan temuan ini untuk membantu mengembangkan *chatbot* yang lebih canggih dan responsif dengan menggabungkan berbagai teknik[11].

Untuk tujuan penelitian ini, *LSTM* (Long Short-Term Memory) dan *GRU* (Gated Recurrent Unit) dipilih karena keduanya memiliki keunggulan dalam memahami dan memproses data sekuensial, yang merupakan fitur utama dari data percakapan *chatbot*. Mekanisme *gate* seperti *forget gate*, *input gate*, dan *output gate* dimasukkan ke dalam desain *LSTM*, yang memungkinkan model untuk menyimpan informasi penting dari langkah sebelumnya secara berurutan sambil melupakan informasi yang tidak penting. Sangat

membantu dalam menangkap dependensi jangka panjang dan memastikan model berfungsi dengan baik meskipun konteks temporal data yang kompleks. *GRU*, versi yang lebih ringan dari *LSTM*, memiliki kemampuan yang sebanding dengan efisiensi komputasi yang lebih baik, yang membuatnya pilihan yang sangat baik untuk menangani data dengan sumber daya terbatas [17]. Sebaliknya, *CNN* (*Convolutional Neural Network*) lebih baik digunakan untuk analisis fitur lokal pada data seperti gambar atau teks pendek yang tidak memiliki hubungan temporal yang kuat. Namun, karena *CNN* bergantung pada operasi konvolusi untuk mengekstraksi pola spasial, *CNN* tidak cukup baik untuk mengidentifikasi hubungan antar elemen dalam urutan panjang, seperti percakapan. Akibatnya, *LSTM* dan *GRU* lebih cocok untuk jenis penelitian ini karena konteks antar-kalimat dan aliran percakapan sangat penting untuk menghasilkan respons yang relevan dan akurat.

II. METODE



Gambar 1 Flowchart Tahapan Penelitian Chatbot

Penelitian ini mengikuti siklus pengembangan model pembelajaran mesin (*MDLC*) sebagai panduan untuk setiap tahap. *MDLC* tersendiri adalah panduan lengkap yang mencakup seluruh proses pengembangan model *machine learning* dari awal hingga akhir. Tahapan utama *MDLC* meliputi pengumpulan dan analisis data eksploratif, persiapan data, pembangunan model, dan pengoperasian model[18]. Ilustrasi mengenai langkah-langkah dalam penelitian ini dapat dicermati pada Gambar 1 sebelumnya.

Pengumpulan Data

Langkah pertama yang perlu dilakukan adalah pengumpulan data. Ini dilakukan dengan mengumpulkan data dari sumber asli dan menyimpannya dalam repositori yang telah ditetapkan untuk pemrosesan dan analisis tambahan. Setelah itu, langkah selanjutnya adalah mempelajari dan menganalisis data untuk menemukan pola-pola yang dapat menghasilkan gagasan dan pengetahuan baru untuk pemodelan.

Persiapan Data

Persiapan data merupakan salah satu langkah proyek pembelajaran mesin yang paling sulit dan memakan waktu. Pada tahap ini, data mentah diproses menjadi data yang siap

digunakan untuk melatih model pembelajaran mesin. Proses ini melibatkan berbagai langkah, termasuk mentransformasi data agar sesuai untuk langkah selanjutnya, membersihkan data dari kueri yang tidak relevan, melakukan tokenisasi, dan mengindeks setiap kata yang telah ditokenisasi. Selain itu, langkah ini juga mencakup penambahan token SOS (Start of Sequence) dan EOS (End of Sequence) pada respon, serta OUT pada kata-kata yang tidak terdapat dalam kosakata. Kosakata yang telah diindeks kemudian dibalik, teks diubah menjadi urutan angka, dan padding ditambahkan pada urutan tersebut. Akhirnya, label diubah menjadi vektor biner untuk proses pelatihan selanjutnya.

Eksplorasi Data

Pelatihan model prediktif merupakan sebuah proses yang melibatkan pembelajaran dari data yang telah dikumpulkan. Oleh karena itu, penting untuk memiliki data yang memiliki kualitas tinggi sebagai prasyarat dasar dalam setiap kegiatan pengembangan model mesin. Dalam penelitian ini, Peneliti mengumpulkan data tentang pertanyaan dan jawaban yang berkaitan dengan industri layanan pelanggan e-commerce dengan membaca literatur dan mencari referensi ke jurnal-jurnal yang berkaitan dengan topik penelitian. Dataset csv yang disusun dalam format JSON akan digunakan untuk melatih model chatbot ini. Peneliti menggunakan konsep 5W + 1H dalam penyusunan dataset ini. Pemodelan chatbot berbasis *LSTM* ini mencakup berbagai jenis objek, antara lain:

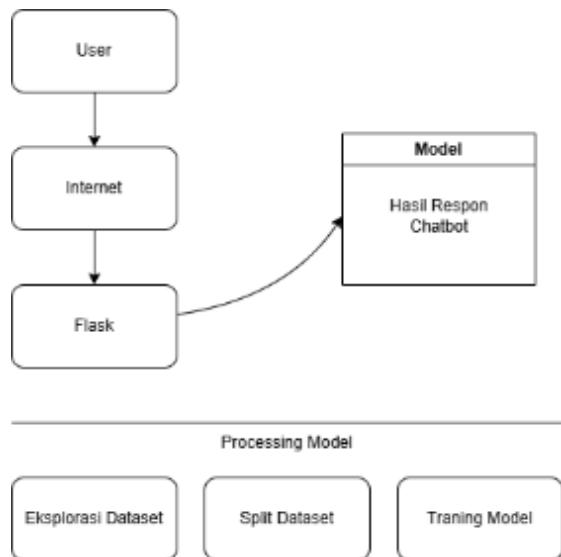
- *Intens*: Terdiri dari 3 elemen, yaitu Tags, Pattern, dan Responses.
- *Utterance*: Bagian tags memuat kategori pertanyaan dan jawaban.
- *Category*: Bagian Pattern berfokus pada pertanyaan yang akan menjadi materi pelatihan untuk model chatbot.
- *Responses*: Bagian Responses berisi tanggapan atau jawaban yang akan menjadi output dari prediksi yang dihasilkan oleh pengguna.

Pengembangan Model

Langkah berikutnya adalah membuat model. Pertama-tama, data akan dibagi menjadi dua bagian: data pelatihan dan data validasi. Sebagian dari data ini digunakan untuk mencari parameter model pembelajaran mesin terbaik dan meminimalkan kesalahan pada set data. Selanjutnya, data validasi digunakan untuk menguji kemampuan model yang telah dibangun. Untuk meningkatkan akurasi dan kinerja model, proses ini diulang beberapa kali (epoch). Pembagian data dilakukan sebesar 80% untuk pelatihan dan 20% untuk validasi. Setelah itu, langkah berikutnya adalah menentukan struktur model, menentukan metrik yang akan digunakan untuk mengukur kinerjanya, dan kemudian melatihnya. Lapisan Embedding, *LSTM*, dan Output adalah struktur model yang akan digunakan, Karena model ini menggunakan klasifikasi multikelas, fungsi kerugian yang digunakan adalah *categorical_crossentropy* dengan optimisasi menggunakan

algoritma "Adam". Model akan dievaluasi menggunakan metrik "akurasi".

Deployment



Gambar 2 Arsitektur Deployment Chatbot Flask

Sebelum model yang telah dibuat dapat diuji oleh pengguna, langkah pertama adalah melakukan deployment dan hosting model agar dapat diakses secara daring. Dalam penelitian ini, proses deployment dilakukan di sebuah situs web yang dibangun menggunakan bahasa pemrograman Python dan kerangka kerja Flask. HTML, JavaScript, dan CSS digunakan untuk membuat antarmuka yang interaktif dan ramah pengguna. Situs web ini sederhana dan terdiri dari satu halaman utama dengan gelembung. Ketika mereka mengklik gelembung tersebut, menu obrolan akan muncul. Di sana, pengguna dapat mengajukan pertanyaan tentang berbagai informasi tentang *online shopping*.

Untuk melaksanakan proses penyimpanan model dalam format .h5 tersebut, penulis memanfaatkan perpustakaan (Library) Flask. Flask adalah suatu kerangka kerja (framework) web yang sering digunakan dalam pengembangan aplikasi web dengan Python. Dalam konteks ini, Flask digunakan untuk membuat antarmuka yang memfasilitasi instalasi dan penggunaan model chatbot. Proses ini memungkinkan file model.h5 dapat diintegrasikan sebagai fitur chatbot dalam aplikasi yang dikembangkan, khususnya pada bagian pengembangan mobile website. Untuk mengakses dan menggunakan fitur chatbot tersebut, pengguna dapat berinteraksi dengan situs web melalui antarmuka API yang disediakan oleh Flask dan elemen HTML yang terintegrasi dalam mobile website tersebut. Dengan demikian, pengguna dapat dengan mudah berkomunikasi dengan chatbot melalui aplikasi web di perangkat seluler yang mereka gunakan. Proses ini memungkinkan file model.h5 dapat diintegrasikan sebagai fitur chatbot dalam aplikasi yang dikembangkan, khususnya

pada bagian pengembangan mobile website. Untuk mengakses dan menggunakan fitur chatbot tersebut, pengguna dapat berinteraksi dengan situs web melalui antarmuka API yang disediakan oleh Flask dan elemen HTML yang terintegrasi dalam mobile website tersebut. Dengan demikian, pengguna dapat dengan mudah berkomunikasi dengan chatbot melalui aplikasi web di perangkat seluler yang mereka gunakan.

Metode Pengujian Sistem

Proses pengembangan perangkat lunak menggunakan Pendekatan Siklus Pengembangan Perangkat Lunak atau *Software Development Life Cycle* (SDLC). *SDLC* mengatur dan mengelola proses pengembangan perangkat lunak dari fase perencanaan hingga implementasi dan pemeliharaan. Pemilihan *SDLC* dilakukan karena penelitian ini menitikberatkan pada aspek perancangan dan struktur arsitektur perangkat lunak chatbot yang sedang dikembangkan. Dalam konteks ini, penelitian tidak difokuskan pada hasil akurasi jawaban yang dihasilkan oleh chatbot, melainkan lebih kepada tahapan perencanaan, perancangan, pengembangan, pengujian, dan implementasi sistem chatbot di dalam lingkungan *e-commerce* [19].

Dengan menggunakan pendekatan *SDLC*, penelitian dapat memastikan bahwa tahapan-tahapan pengembangan dilakukan secara sistematis dan terstruktur[20]. Tujuannya bukan hanya untuk menghasilkan chatbot dengan akurasi yang tinggi, tetapi juga untuk memastikan keberlanjutan, kehandalan, dan efektivitas arsitektur perangkat lunak yang dibuat dalam mendukung fungsi *customer service* pada platform *e-commerce*.

III. HASIL DAN PEMBAHASAN

Penelitian ini merancang sistem chatbot berbasis kumpulan data buatan yang disimpan dalam format JSON. Dataset tersebut terstruktur melalui tiga komponen utama untuk memfasilitasi interaksi antara pengguna dan sistem. Pertama, *intent* (tujuan percakapan) yang dilabeli secara khusus guna mengkategorikan maksud percakapan pengguna. Setiap *intent* dikelompokkan berdasarkan tag unik guna mempermudah identifikasi konteks selama proses dialog. Kedua, *pattern* (pola kalimat) yang merepresentasikan variasi frasa atau pertanyaan yang mungkin diinputkan pengguna terkait suatu *intent*. Pola ini berfungsi sebagai referensi bagi sistem untuk memetakan pertanyaan pengguna ke kategori yang sesuai. Terakhir, *response* (jawaban otomatis) yang telah diprogram sebelumnya untuk memberikan balasan kontekstual sesuai dengan *intent* yang terdeteksi. Dengan struktur data terorganisir ini, chatbot mampu menciptakan dialog yang dinamis dan relevan, meningkatkan interaktivitas melalui respons yang tepat berdasarkan analisis input dari pengguna.

Gambar 3 Dataset Public TWCS (Twitter Customer Service) in JSON

Pada penelitian ini menggunakan dataset public untuk proses training pada modelnya yang bersumber dari situs Kaggle.com. Dataset yang diambil berasal dari format csv lalu ditransformasikan menjadi format JSON. Dalam Gambar 3, dataset dalam format JSON digunakan sebagai basis data untuk chatbot. Element "tag" dalam dataset ini berfungsi sebagai kategori atau label yang mengkategorikan pernyataan atau pertanyaan pengguna berdasarkan topik atau konsep tertentu. Sebagai contoh, label "selamat datang" digunakan untuk menyapa pengguna pada awal interaksi, sedangkan label "terima kasih" menggabungkan ungkapan terima kasih yang bertujuan untuk mengakhiri percakapan.

TABLE 1
DETAIL JUMLAH SAMPLE PER KATEGORI

Kategori (Tag)	Jumlah Patterns
greeting	7
SalesbotX	7
pencipta_salesbotx	6
goodbye	6
terimakasih	7
pengertian_Customer_service	6
Customer_Service	6
hours	6
returns	6
shipping	6
payment_methods	6
order_status	6
account_help	6
product_information	6
discounts	6
membership	6
contact_support	6
tracking	6
product_availability	6
refund_policy	6
technical_issues	6
order_cancellation	6
exchange_policy	6
loyalty_program	6
bulk_orders	6
gift_cards	6
account_security	6
account_creation	6
gift_wrapping	6
pre_order	6
product_warranty	6
product_inquiry	6

customer_support	6
promotions	6
product_knowledge	6
waktu_pengiriman	6
returns_and_refunds	6

Penelitian ini menggunakan dataset yang mencakup total 580 pola (patterns), yang dibagi menjadi 38 kategori (tags). Pada umumnya, setiap kategori memiliki jumlah pola yang hampir sama, berkisar antara 6 dan 7 pola. Beberapa kategori, seperti "selamat datang" dan "SalesbotX", memiliki jumlah pola yang sedikit lebih besar, masing-masing 7 pola, sementara kategori lain, seperti "pencipta_salesbotx", "refund_policy", dan "jam", memiliki 6 pola. Dataset dapat mencakup berbagai jenis pertanyaan yang terkait dengan chatbot layanan pelanggan berkat distribusi yang relatif seimbang ini. Namun, untuk memberikan representasi yang lebih luas dari berbagai pertanyaan yang mungkin muncul dalam situasi dunia nyata, penambahan data melalui teknik augmentasi teks atau pengumpulan data tambahan dapat dipertimbangkan. Ini akan meningkatkan kinerja model.

Elemen "pattern" mencakup pola atau susunan kalimat yang terkait dengan setiap tag, seperti pertanyaan dan pernyataan yang sesuai dengan ide di balik tag tersebut. Misalnya, untuk tag "salam", pola yang mungkin termasuk variasi salam seperti "Hai" atau "Halo". Elemen "response" berisi daftar jawaban yang telah ditentukan untuk setiap tag, yang akan dipilih oleh chatbot untuk menanggapi input pengguna.

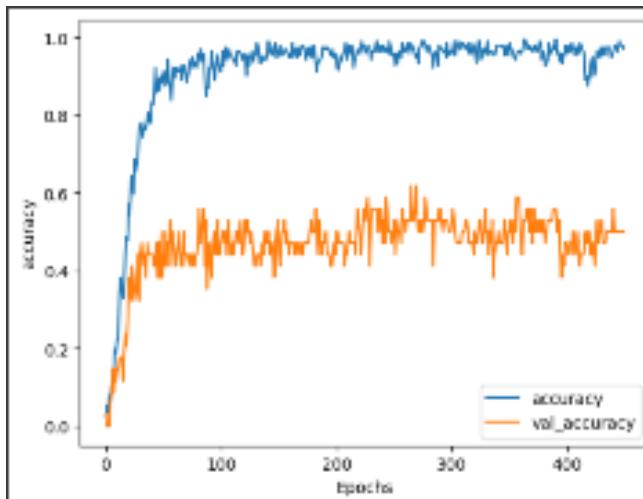
Chatbot dapat memberikan respons yang sesuai dengan ide atau topik yang telah diatur dalam dataset dengan struktur JSON yang mencakup elemen seperti tag, pola, dan tanggapan. Struktur ini memungkinkan chatbot untuk mempelajari dan memahami konteks input pengguna dan memberikan respons yang relevan. Ini meningkatkan akurasi dan kesesuaian respons terhadap kebutuhan pengguna. Setelah data diperoleh dan dipahami, langkah selanjutnya adalah mempersiapkan data. Data yang terdiri dari pertanyaan dan jawaban diubah menjadi satu tabel dengan kolom "pertanyaan" dan "jawaban" dalam format file JSON[21]. File ini kemudian akan diproses menggunakan Python. Dataset yang dihasilkan berisi 148 sampel dengan 18 informasi masing-masing. Data tersebut kemudian diubah menjadi format dataframe di Python menggunakan pustaka Pandas. Untuk mempermudah pemrosesan data pertanyaan dan jawaban, kedua data tersebut dipisahkan dan diubah menjadi daftar.

Data pertanyaan diubah menjadi daftar dan kemudian dibersihkan menggunakan pustaka NLTK. Proses pembersihan melibatkan beberapa langkah, termasuk mengubah semua huruf menjadi huruf kecil, menghapus simbol yang tidak perlu, dan menggunakan stopwords untuk menghilangkan kata-kata yang tidak penting atau tidak bermakna. Tujuan dari pembersihan data adalah untuk memastikan bahwa semua pertanyaan memiliki format yang konsisten dan bebas dari simbol yang tidak perlu, sehingga jumlah kosakata dapat dikurangi. Dengan demikian, proses

pelatihan model menjadi lebih cepat dan hasil model menjadi lebih baik.

Kosakata yang telah disusun sebelumnya merupakan tipe kamus data dengan kata-kata sebagai kunci dan angka sebagai nilai. Tujuannya adalah untuk menerjemahkan hasil prediksi keluaran yang berupa angka indeks menjadi kata-kata yang dapat dipahami. Dalam proses ini, kata-kata yang awalnya merupakan kunci dalam kamus diubah menjadi angka indeks, sedangkan nilai yang awalnya merupakan indeks diubah menjadi kata-kata [22]. Setelah selesai membuat kamus kosakata yang mencantumkan kata-kata beserta indeksnya, langkah selanjutnya adalah membuat urutan bilangan bulat. Proses ini melibatkan pengubahan urutan kata menjadi urutan bilangan bulat. Untuk melakukan konversi ini, digunakan kamus yang telah dibuat sebelumnya. Kamus ini memungkinkan kita untuk mengganti setiap kata dengan indeks yang sesuai, sehingga menghasilkan urutan bilangan bulat yang merepresentasikan urutan kata asli.

Padding adalah teknik yang digunakan untuk menyamakan panjang setiap urutan yang berbeda menjadi panjang yang seragam. Hal ini penting agar setiap input ke dalam lapisan embedding memiliki panjang yang konsisten. Dalam penelitian ini, digunakan struktur model yang terdiri dari beberapa lapisan untuk mencapai hasil yang diinginkan. Pada lapisan pertama, terdapat lapisan embedding yang berfungsi untuk merepresentasikan setiap kata menggunakan vektor berdimensi 128. Selanjutnya, lapisan kedua adalah lapisan LSTM yang memiliki 256 unit perceptron, dirancang untuk menangkap pola temporal dalam data. Akhirnya, lapisan keluaran adalah lapisan Dense Output yang terdiri dari 272 unit perceptron dan menggunakan fungsi aktivasi softmax untuk menghasilkan prediksi akhir. Struktur ini dirancang untuk mengoptimalkan kinerja model dalam menganalisis dan memproses data.



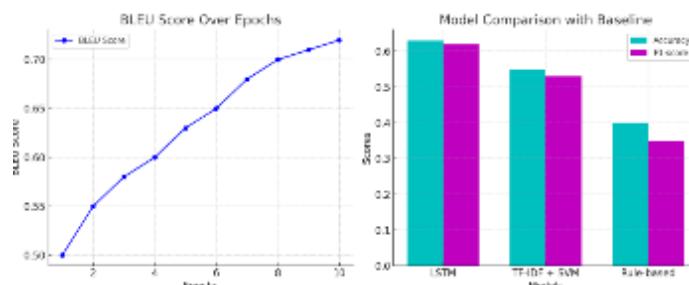
Gambar 4 Training and Validation Accuracy Graph

Penelitian ini mengembangkan model klasifikasi menggunakan arsitektur jaringan saraf yang terdiri dari beberapa lapisan, termasuk lapisan embedding, LSTM, dan

GRU. Model ini dirancang dengan menambahkan regularisasi L2 untuk mencegah overfitting, dengan parameter regularisasi diatur pada 0,02. Selain itu, tingkat dropout ditetapkan pada 0,5 untuk meningkatkan generalisasi model. Model dimulai dengan mendefinisikan lapisan input yang menerima data dengan dimensi 8. Selanjutnya, lapisan embedding digunakan untuk mengonversi kata-kata menjadi representasi vektor dengan ukuran 500, berdasarkan indeks kata yang dihasilkan oleh tokenizer. Model ini kemudian melibatkan dua lapisan Bidirectional LSTM, masing-masing dengan 64 unit, yang diikuti oleh lapisan dropout untuk mengurangi risiko overfitting. Setelah itu, lapisan Bidirectional GRU juga ditambahkan dengan jumlah unit yang sama, yaitu 64, untuk menangkap pola temporal dalam data.

Analisis Statistik Paired T-Test

Hasil pelatihan model menunjukkan akurasi pelatihan sebesar 90%, tetapi akurasi validasi hanya mencapai 63%, dengan F1-score sebesar 62%. Perbedaan yang signifikan ini mengindikasikan adanya overfitting, di mana model belajar terlalu banyak pada data pelatihan tetapi kurang mampu untuk menggeneralisasi pada data validasi. Berdasarkan hasil paired t-test, diperoleh nilai $p = 0.110$, yang lebih besar dari 0.05. Hal ini menunjukkan bahwa perbedaan antara akurasi pelatihan (mean = 0.88) dan validasi (mean = 0.86) tidak signifikan secara statistik. Oleh karena itu, model ini dianggap cukup konsisten dalam performanya pada kedua dataset.



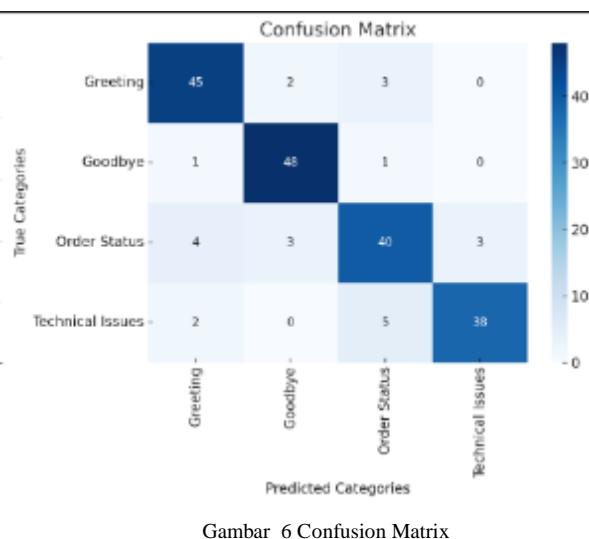
Gambar 5 BLUE Score over Epochs & Model Comparison with Baseline

Pada Gambar 5 diatas, nilai BLEU Score meningkat secara konsisten selama pelatihan, dari 0.50 pada epoch pertama hingga 0.72 pada epoch ke-10. Ini menunjukkan kemampuan model untuk menghasilkan respons yang sesuai dengan jawaban yang diharapkan.

Untuk memahami keunggulan model LSTM, dilakukan perbandingan dengan baseline seperti TF-IDF + SVM dan rule-based chatbot. Model TF-IDF + SVM menghasilkan akurasi validasi sebesar 55% dan F1-score sebesar 53%, tetapi kurang mampu menangkap hubungan kontekstual dalam percakapan. Pendekatan ini hanya mengandalkan informasi statistik dari kata-kata tanpa mempertimbangkan urutan atau hubungan temporal, sehingga performanya lebih rendah pada skenario yang memerlukan pemahaman konteks.

Sementara itu, rule-based chatbot terbatas pada pola pertanyaan yang telah ditentukan dan tidak efektif untuk

menangani variasi pertanyaan yang lebih kompleks. Dibandingkan dengan kedua baseline tersebut, model LSTM menunjukkan performa yang lebih unggul, baik dalam akurasi maupun pemahaman konteks percakapan, menjadikannya lebih cocok untuk aplikasi chatbot dalam skenario e-commerce nyata.



Gambar 6 Confusion Matrix

Matriks Confusion ini menggambarkan performa model dalam mengklasifikasikan data ke dalam empat kategori. Matriks ini menunjukkan bahwa kategori seperti 'greeting' dan 'goodbye' memiliki tingkat klasifikasi yang benar lebih tinggi dibandingkan kategori lain seperti 'technical issues', yang lebih sering salah diklasifikasikan.

Meskipun transformer seperti BERT sangat baik untuk tugas NLP, model ini membutuhkan dataset yang lebih besar, tetapi penelitian ini menggunakan dataset berskala kecil hingga menengah. Oleh karena itu, arsitektur LSTM dan GRU lebih cocok untuk situasi ini. Dibandingkan dengan transformer seperti BERT dan GPT, LSTM dan GRU lebih efisien untuk diterapkan pada dataset kecil hingga menengah, seperti dataset penelitian ini yang terdiri dari 580 entri. Karena memerlukan data besar untuk mencapai performa optimal, penggunaan transformer pada dataset kecil seringkali berlebihan[23]. arsitektur LSTM dan GRU lebih sesuai karena mampu bekerja secara efisien pada dataset yang lebih kecil sambil tetap mempertahankan kemampuan untuk menangkap konteks temporal dan hubungan antar-elemen dalam data sekuensial. Studi yang lebih baru juga menunjukkan bahwa LSTM dan GRU tetap menjadi pilihan kompetitif dalam tugas-tugas NLP, terutama dalam lingkungan dengan keterbatasan sumber daya data dan komputasi[24].



Gambar 7 Tampilan chat percakapan SalesbotX.

Setelah melewati proses lapisan-lapisan tersebut, data diratakan menggunakan lapisan Flatten sebelum diteruskan ke lapisan Dense yang memiliki 64 unit dan menggunakan fungsi aktivasi ReLU, serta dilengkapi dengan regularisasi L2.. Sebagai langkah terakhir, lapisan output terdiri dari Dense dengan jumlah kelas yang sesuai dan fungsi aktivasi softmax, yang memungkinkan model untuk melakukan klasifikasi multikelas. Struktur model yang dihasilkan dapat dilihat melalui ringkasan model yang ditampilkan dengan menggunakan fungsi `model.summary()`. Penelitian ini melibatkan klasifikasi multikelas menggunakan fungsi kehilangan categorical_crossentropy (hard) dan fungsi optimasi "Adam", serta metrik evaluasi "akurasi". Data dibagi menjadi 80% untuk pelatihan dan 20% untuk validasi.

TABEL 2
ACCURACY AND LOSS ON LSTM TRAINING BASED ON EPOCHS.

Epochs	Accuracy (%)	Val_accuracy (%)	Loss (%)	Val_loss (%)
100	87	63	0.3	4.9
200	98	51	0.17	4.2
300	99	54	0.08	3.6
450	91	54	0.48	4.8

Proses pelatihan dilakukan dengan struktur model yang telah dibuat sebelumnya selama 450 epoch. Hasil dari proses pelatihan menunjukkan tingkat akurasi sebesar 99% dan akurasi validasi sebesar 63%, dengan nilai kehilangan sebesar 1% dan kehilangan validasi sebesar 4%. Grafik yang menunjukkan hasil akurasi, kehilangan, akurasi validasi, dan kehilangan validasi pada setiap epoch dapat dilihat pada Gambar 4.

Eksperimen Mengatasi Overfitting

Untuk mengurangi overfitting, beberapa eksperimen dilakukan. Dropout diterapkan dengan tingkat sebesar 0.5 pada lapisan LSTM untuk mencegah model terlalu bergantung pada neuron tertentu selama pelatihan. Selain itu, parameter regularisasi L2 diatur pada 0.05 untuk menambahkan penalti terhadap bobot besar yang berkontribusi pada kompleksitas model. Pelatihan model juga dihentikan secara otomatis dengan teknik early stopping jika akurasi validasi tidak meningkat dalam 10 epoch berturut-turut. Selain itu, teknik augmentasi data seperti synonym replacement dan back-translation diterapkan untuk meningkatkan keragaman dan generalisasi data pelatihan.

Tahap selanjutnya adalah implementasi, yang bertujuan untuk membuat model chatbot yang telah dikembangkan dapat digunakan oleh pengguna melalui antarmuka yang mirip dengan yang digunakan selama pengujian model. Proses implementasi dilakukan menggunakan bahasa pemrograman Python dan kerangka kerja Flask, sementara antarmuka dibangun menggunakan HTML, CSS, dan JavaScript [25]. Pada situs web yang telah dibuat, ketika pengguna memasukkan pertanyaan mengenai informasi belanja online, sistem akan memberikan respons sesuai dengan yang ditampilkan pada Gambar 4. Tahap terakhir dalam proses ini adalah tahap pengujian, yang dilakukan melalui metode validasi pengguna. Dalam metode ini, pengguna diundang untuk mencoba chatbot yang telah dibangun dan memberikan penilaian mengenai apakah hasil yang diberikan oleh chatbot sesuai dengan kebutuhan mereka. Pengguna akan menerima model percobaan serta daftar topik pertanyaan dan jawaban yang diharapkan.

Dengan menggunakan sesi berbasis token, chatbot dapat menangani banyak permintaan sekaligus. ID pengguna yang dibuat secara dinamis saat pengguna mulai berinteraksi dengan chatbot dilacak. Sistem menggunakan caching dengan Redis untuk menyimpan sementara data percakapan untuk meningkatkan efisiensi. Hal ini mempercepat respons dan mengurangi beban server.

Keamanan adalah prioritas utama saat menggunakan chatbot ini. Semua input pengguna melalui proses sanitasi dan validasi untuk mencegah input berbahaya seperti SQL Injection dan prompt injection. Pustaka keamanan seperti SQLAlchemy melindungi query dengan mengelola data yang dikirim ke database atau API eksternal. Selain itu, token otentikasi yang dienkripsi membatasi akses ke API, sehingga hanya pengguna yang sah yang dapat mengakses data atau fitur tertentu.

Namun, penting untuk diingat bahwa hasil ini diperoleh dalam lingkungan yang terkontrol. Meskipun model menunjukkan kinerja yang baik dengan dataset saat ini, situasi di dunia nyata sering kali melibatkan lebih banyak variabilitas dan kompleksitas. Faktor-faktor seperti variasi input pengguna dan pertanyaan yang ambigu dapat mempengaruhi kinerja model. Oleh karena itu, meskipun model menunjukkan akurasi yang belum maksimal, pengujian lebih

lanjut dalam kondisi nyata diperlukan untuk menilai efektivitasnya secara menyeluruh.

Dengan arsitektur yang mendukung *scaling horizontal* agar dapat menampung lebih banyak pengguna. *Load balancer* mengatur lalu lintas pengguna secara merata di berbagai server. Untuk keamanan, sistem ini menggunakan *HTTPS* untuk enkripsi data *end-to-end* dan mekanisme perlindungan terhadap serangan *DDoS*.

Model LSTM mengatasi keterbatasan RNN tradisional, seperti masalah gradien yang menghilang, dengan menggunakan mekanisme pengaturan untuk mengelola ketergantungan jangka panjang, yang menghasilkan akurasi tinggi dan kehilangan yang rendah. Sementara model Transformer seperti BERT dan GPT unggul dalam menangani pola bahasa yang kompleks melalui mekanisme perhatian, yang berpotensi menawarkan kinerja yang lebih baik dalam menangkap konteks yang lebih halus, model LSTM kami tetap efektif [26]. Penelitian di masa depan dapat mengkexplorasi pendekatan hibrida yang menggabungkan teknik LSTM dan Transformer untuk memanfaatkan kekuatan masing-masing, yang berpotensi meningkatkan kinerja chatbot dalam berbagai jenis pertanyaan dan interaksi.

Untuk mencegah penyalahgunaan sistem, strategi tambahan seperti pembatasan tingkat permintaan juga diterapkan. Penelitian ini telah mengembangkan *chatbot* yang sangat memahami konteks percakapan dan dapat memberikan respons yang tepat. Namun, untuk memastikan *chatbot* dapat berfungsi dengan baik dalam lingkungan *e-commerce* yang lebih kompleks, perluasan dataset, optimisasi model, dan peningkatan infrastruktur keamanan.

IV. KESIMPULAN

Hasil pengujian model menunjukkan bahwa LSTM efektif dalam mengatasi vanishing gradient problem, yang sering menjadi kendala utama pada model seperti vanilla RNN. Hal ini ditunjukkan oleh kemampuan model untuk mempertahankan informasi konteks dalam data sekuensial, yang menghasilkan akurasi pelatihan sebesar 90% dan validasi sebesar 63%. Namun, kesimpulan ini memerlukan penguatan melalui eksperimen lebih lanjut dengan membandingkan performa LSTM terhadap model alternatif seperti vanilla RNN dan Transformer, yang juga dikenal unggul dalam tugas-tugas NLP tertentu. Perbandingan ini akan memberikan wawasan yang lebih jelas mengenai kelebihan dan keterbatasan masing-masing model dalam menangani data percakapan.

Meskipun model menunjukkan hasil yang menjanjikan, tingkat akurasi validasi dan F1-score saat ini menunjukkan bahwa model masih memerlukan perbaikan untuk dapat digunakan secara optimal dalam dunia nyata. Pengujian di lingkungan produksi yang melibatkan variabilitas input pengguna, seperti pertanyaan ambigu atau formulasi baru, juga diperlukan untuk mengevaluasi kesiapan model dalam skenario e-commerce nyata.

Dari sudut pandang bisnis, implementasi chatbot ini memiliki potensi untuk meningkatkan efisiensi layanan pelanggan dengan mengotomatisasi respons terhadap pertanyaan umum. Namun, sebelum diadopsi secara penuh, diperlukan langkah-langkah tambahan, seperti fine-tuning model pada dataset yang lebih besar dan lebih representatif, serta penerapan augmentasi data untuk meningkatkan generalisasi. Penambahan fitur seperti personalisasi respons dan integrasi dengan API e-commerce juga dapat meningkatkan nilai tambah dari chatbot ini. Dengan perbaikan yang berkelanjutan, chatbot berbasis LSTM ini dapat menjadi solusi yang andal dan efektif dalam memenuhi kebutuhan interaksi pelanggan di sektor e-commerce.

DAFTAR PUSTAKA

- [1] M. M. Khan, "Development of An e-commerce Sales Chatbot," in HONET 2020 - IEEE 17th International Conference on Smart Communities: Improving Quality of Life using ICT, IoT and AI, Institute of Electrical and Electronics Engineers Inc., Dec. 2020, pp. 173–176. doi: 10.1109/HONET50430.2020.9322667.
- [2] S. Oktriwina, "NLP: Kecerdasan Buatan yang Bantu Komputer Pahami Bahasa Manusia," 2021.
- [3] S. Patil, V. M. Mudaliar, P. Kamat, and S. Gite, "LSTM based Ensemble Network to enhance the learning of long-term dependencies in chatbot," International Journal for Simulation and Multidisciplinary Design Optimization, vol. 11, p. 25, Dec. 2020, doi: 10.1051/smd/2020019.
- [4] P. B. Wintoro, H. Hermawan, M. A. Muda, and Y. Mulyani, "Implementasi Long Short-Term Memory pada Chatbot Informasi Akademik Teknik Informatika Unila," EXPERT: Jurnal Manajemen Sistem Informasi dan Teknologi, vol. 12, no. 1, p. 68, Jun. 2022, doi: 10.36448/expert.v12i1.2593.
- [5] "Implementasi NLP Pada Chatbot Layanan Akademik Dengan Algoritma Bert Implementation Of NLP On Academic Service Chatbot With Bertalgorithm."
- [6] G. A. Santos, G. G. de Andrade, G. R. S. Silva, F. C. M. Duarte, J. P. J. Da Costa, and R. T. de Sousa, "A Conversation-Driven Approach for Chatbot Management," IEEE Access, vol. 10, pp. 8474–8486, 2022, doi: 10.1109/ACCESS.2022.3143323.
- [7] L. Sanny, A. C. Susastra, C. Roberts, and R. Yusramdaleni, "The analysis of customer satisfaction factors which influence chatbot acceptance in Indonesia," Management Science Letters, pp. 1225–1232, 2020, doi: 10.5267/j.msl.2019.11.036.
- [8] P. Rajendra Kumar and E. B. K. Manash, "Deep learning: A branch of machine learning," in Journal of Physics: Conference Series, Institute of Physics Publishing, Jun. 2019. doi: 10.1088/1742-6596/1228/1/012045.
- [9] L. Cuno Klopfenstein, S. Delpriori, and A. Ricci, "Adapting a conversational text generator for online chatbot messaging," 2019. [Online]. Available: <https://www.ibm.com/watson/services/conversation/>
- [10] M. Fachrul, "E-Commerce: Pengembangan Model Perniagaan Menggunakan Internet."
- [11] G. F. Aisyah, I. J. Putra, and S. S. Hidayat, "Open Artificial Intelligence Analysis using ChatGPT Integrated with Telegram Bot," Jurnal ELTIKOM, vol. 7, no. 1, pp. 60–66, Jun. 2023, doi: 10.31961/eltikom.v7i1.724.
- [12] A. Silvanie and R. Subekti, "Aplikasi Chatbot Untuk Faq Akademik Di Ibi-K57 Dengan LSTM Dan Penyematan Kata," JIKO (Jurnal Informatika dan Komputer), vol. 5, no. 1, pp. 19–27, Apr. 2022, doi: 10.33387/jiko.v5i1.3703.
- [13] H. & V. L. Muhyidin, "Pengembangan Chatbot untuk Meningkatkan Pengetahuan dan Kesadaran Keamanan Siber Menggunakan Long Short-Term Memory. Jurnal Informatika dan Rekayasa Perangkat Lunak.," 2023.
- [14] H. Wang, S. Chen, and L. Luo, "A diffusion algorithm based on P systems for continuous global optimization," J Comput Sci, vol. 44, p. 101112, Jul. 2020, doi: 10.1016/j.jocs.2020.101112.
- [15] P. Choudhary and S. Chauhan, "An intelligent chatbot design and implementation model using long short-term memory with recurrent neural networks and attention mechanism," Decision Analytics Journal, vol. 9, p. 100359, Dec. 2023, doi: 10.1016/j.dajour.2023.100359.
- [16] R. Luthfiansyah and B. Wasito, "Penerapan Teknik Deep Learning (Long Short Term Memory) dan Pendekatan Klasik (Regresi Linier) dalam Prediksi Pergerakan Saham BRI," Jurnal Informatika dan Bisnis, vol. 12, no. 2, pp. 42–54, Dec. 2023, doi: 10.46806/jib.v12i2.1059.
- [17] M. N. A. Putera Khan, D. R. S. Saputro, S. Sutanto, and A. Wibowo, "Sentiment Analysis With Long-Short Term Memory (Lstm) And Gated Recurrent Unit (Gru) Algorithms," BAREKENG: Jurnal Ilmu Matematika dan Terapan, vol. 17, no. 4, pp. 2235–2242, Dec. 2023, doi: 10.30598/barekengvol17iss4pp2235-2242.
- [18] R. I. Borman and Y. Purwanto, "Implementasi Multimedia Development Life Cycle pada Pengembangan Game Edukasi Pengenalan Bahaya Sampah pada Anak," Jurnal Edukasi dan Penelitian Informatika (JEPIN), vol. 5, no. 2, p. 119, Aug. 2019, doi: 10.26418/jp.v5i2.25997.
- [19] G. Yanto, S. Puspita, and Z. Efendy, "Inovasi E-Commerce Berbasis Chatbot AI Pada Deta Rancak dan Tingkulauk Kreasi Sanggar Seni Permata Hati," JISTech (Journal of Islamic Science and Technology) JISTech, vol. 9, no. 2, pp. 233–239, doi: 10.30829/jistech.v9i2.22768.
- [20] P. Sihite, A. Simorangkir, N. N. K. Sari, and V. Handrianus Pranatawijaya, "Integrasi Chatbot Custom Chatpt Dengan Chatbase Dalam Meningkatkan Pengalaman Pengguna Dan Efisiensi Layanan Dalam Website E-Commerce," JATI (Jurnal Mahasiswa Teknik Informatika), vol. 8, no. 3, pp. 3532–3536, May 2024, doi: 10.36040/jati.v8i3.9733.
- [21] A. P. J. Dwitama and S. Hidayat, "Identifikasi Ujaran Kebencian Multilabel Pada Teks Twitter Berbahasa Indonesia Menggunakan Convolution Neural Network," Jurnal Sistem Komputer dan Informatika (JSON), vol. 3, no. 2, p. 117, Dec. 2021, doi: 10.30865/json.v3i2.3610.
- [22] I. D. Raharjo and Egia Rosi Subhiyakto, "Implementing Long Short Term Memory (LSTM) in Chatbots for Multi Usaha Raya," Advance Sustainable Science Engineering and Technology, vol. 6, no. 4, p. 02404018, Oct. 2024, doi: 10.26877/asset.v6i4.934.
- [23] Khalis Sofi, Aswan Supriyadi Sunge, Sasmitoh Rahmad Riady, and Antika Zahrotul Kamalia, "Perbandingan Algoritma Linear Regression, LSTM, Dan GRU Dalam Memprediksi Harga Saham Dengan Model Time Series," Seminastika, vol. 3, no. 1, pp. 39–46, Nov. 2021, doi: 10.47002/seminastika.v3i1.275.
- [24] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov, and Q. V Le, "XLNet: Generalized Autoregressive Pretraining for Language Understanding." [Online]. Available: <https://github.com/zihangdai/xlnet>
- [25] L. Anindiyati, "Analisis dan Perancangan Aplikasi Chatbot Menggunakan Framework Rasa dan Sistem Informasi Pemeliharaan Aplikasi (Studi Kasus: Chatbot Penerimaan Mahasiswa Baru Politeknik Astra)," Jurnal Teknologi Informasi dan Ilmu Komputer, vol. 10, no. 2, pp. 291–300, Apr. 2023, doi: 10.25126/jti.20231026409.
- [26] X. Tan et al., "Biogeography of intestinal mucus-associated microbiome: Depletion of genus *Pseudomonas* is associated with depressive-like behaviors in female cynomolgus macaques," J Adv Res, May 2024, doi: 10.1016/j.jare.2024.05.013.