



CHATBOT ASSISTANT DENGAN ALGORITMA LONG SHORT-TERM MEMORY (LSTM) SEBAGAI LAYANAN TANYA JAWAB MENTEE DI INFINITE LEARNING

¹⁾ Rahmad Noor Ikhsan, ²⁾ Adi Suwondo, ³⁾ Muslim Hidayat, ⁴⁾ Dian Asmarajati, ⁵⁾ Sukowiyono

^{1,2,3,4,5)} Universitas Sains Al-Qur'an

¹⁾ Rajaikhsan2021@gmail.com

INFO ARTIKEL

Riwayat Artikel :

Diterima : 13 Juli 2026

Disetujui : 31 Juli 2026

Kata Kunci :

Studi Independen, Deep Learning, Long Short-Term Memory, Natural Language Processing, Telegram bot.

ABSTRAK

Mentor pada program Studi Independen di Infinite Learning sering menghabiskan waktu menjawab pertanyaan umum yang berulang dari mahasiswa, sehingga mengurangi efisiensi pendampingan. Penelitian ini bertujuan membuat chatbot sebagai layanan tanya jawab mentee dengan metode eksperimental. Pengembangan chatbot menggunakan Long Short-Term Memory (LSTM), arsitektur deep learning yang efektif untuk Natural Language Processing (NLP). Data berupa pertanyaan dan jawaban dibagi menjadi 3 bagian: pre-register, running program, dan end program. Model dilatih selama 39 epoch dengan nilai loss 0.0279, accuracy 99.71%, validation loss 0.0237, dan validation accuracy 99.72%. Evaluasi menggunakan confusion matrix menunjukkan accuracy 99%, precision 97%, recall 98%, dan f1-score 97%. Model kemudian diimplementasikan ke Telegram bot untuk layanan tanya jawab otomatis bagi mentee.

ARTICLE INFO

Article History :

Received : Jul 13., 2026

Accepted : Jul 31, 2026

Keywords:

Studi Independen, Deep Learning, Long Short-Term Memory, Natural Language Processing, Telegram bot

ABSTRACT

Mentors in the Studi Independen program at Infinite Learning often spend time answering repetitive general questions from students, reducing the efficiency of guidance. This study aims to develop a chatbot as a question-and-answer service for mentees using an experimental method. The chatbot was developed using Long Short-Term Memory (LSTM), a deep learning architecture effective for Natural Language Processing (NLP). The dataset of questions and answers is divided into three categories: pre-register, running program, and end program. The model was trained for 39 epochs, achieving a loss of 0.0279, accuracy of 99.71%, validation loss of 0.0237, and validation accuracy of 99.72%. Evaluation using a confusion matrix showed 99% accuracy, 97% precision, 98% recall, and 97% f1-score. The trained model was then implemented into a Telegram bot to provide an automated question-and-answer service for mentees.

1. PENDAHULUAN

Infinite Learning merupakan divisi dari PT Kinema Systrans Multimedia yang berfokus pada pelatihan kejuruan berbasis kebutuhan industri digital di kawasan Nongsa Digital Park. Salah satu program yang diselenggarakan adalah Studi Independen, yaitu program pembelajaran selama satu semester yang memberi kesempatan kepada mahasiswa untuk memperoleh kompetensi praktis langsung dari praktisi industri. Pemanfaatan kecerdasan artifisial dalam pendidikan tinggi semakin berkembang karena dapat mendukung pembelajaran, layanan akademik, serta efisiensi pengelolaan informasi bagi mahasiswa dan pengelola program (Zawacki-Richter et al., 2019; Chen et al., 2020).

Dalam pelaksanaannya, Infinite Learning menawarkan beberapa jalur pembelajaran, antara lain IBM Academy: Advance AI, IBM Academy: Hybrid Cloud & Red Hat, Android Mobile Development & UI/UX Design, Web Development & UI/UX Design, serta Game Design & Development. Kegiatan tersebut didampingi oleh Individual Mentor dan Group Challenge Mentor melalui Telegram, Discord, dan WhatsApp. Akan tetapi, mentor masih sering menerima pertanyaan administratif dan teknis dasar yang sama secara berulang, sedangkan informasi yang diperlukan mentee tersebar pada beberapa kanal. Kondisi tersebut dapat mengurangi waktu mentor untuk memberikan pendampingan yang lebih substantif. Berbagai kajian menunjukkan bahwa chatbot pendidikan dapat menyediakan jawaban segera, memperluas akses informasi, dan membantu mengurangi beban tugas rutin pendidik atau pengelola program (Okonkwo & Ade-Ibijola, 2021; Wollny et al., 2021; Labadze et al., 2023).

Chatbot merupakan program komputer yang dirancang untuk menerima masukan pengguna dan memberikan respons melalui percakapan berbasis teks atau suara dengan memanfaatkan kecerdasan artifisial dan pembelajaran mesin (Tamrakar & Wani, 2021; Puspitasari et al., 2024). Kemampuan chatbot dalam memahami pertanyaan sangat berkaitan dengan Natural Language Processing (NLP), yaitu bidang yang mempelajari representasi, pemrosesan, dan pemahaman bahasa manusia oleh komputer (Galassi et al., 2021; Young et al.,

2018). Implementasi NLP memungkinkan chatbot mengidentifikasi pola bahasa, mengenali maksud pengguna, dan memilih respons yang sesuai sehingga interaksi terasa lebih alami dan mudah digunakan (Nova et al., 2024; Smutny & Schreiberova, 2020).

Dalam konteks pendidikan, chatbot telah digunakan untuk layanan informasi, pendampingan belajar, pemberian umpan balik, dan dukungan administratif. Tinjauan sistematis menunjukkan bahwa chatbot dapat meningkatkan ketersediaan layanan dan membantu pengguna memperoleh informasi secara mandiri, meskipun kualitas respons sangat bergantung pada desain percakapan, cakupan basis pengetahuan, dan evaluasi sistem yang memadai (Pérez et al., 2020; Kuhail et al., 2023). Chatbot juga berpotensi meningkatkan pengalaman belajar dan mendukung interaksi mahasiswa dengan sistem pendidikan apabila respons yang diberikan relevan serta sesuai konteks pengguna (Kerly et al., 2007; Lee et al., 2022). Sebaliknya, data pelatihan yang terbatas dan variasi bahasa pengguna dapat menyebabkan kesalahan pengenalan maksud, sehingga reliabilitas dan akurasi respons tetap perlu diuji (Labadze et al., 2023; Wollny et al., 2021).

Salah satu pendekatan dalam pembangunan chatbot adalah klasifikasi intent menggunakan jaringan saraf untuk memetakan pertanyaan pengguna ke kelas respons tertentu. Sistem dialog berbasis pembelajaran mendalam mampu mempelajari representasi teks secara otomatis dan menangani pola bahasa yang lebih beragam dibandingkan pendekatan yang sepenuhnya bergantung pada aturan manual (Chen et al., 2017; Galassi et al., 2021). Recurrent Neural Network (RNN) banyak digunakan untuk data sekuensial karena dapat mempertahankan informasi dari urutan sebelumnya, tetapi RNN konvensional memiliki keterbatasan dalam mempelajari dependensi jangka panjang akibat masalah vanishing gradient (Rahman et al., 2024; Lasijan et al., 2023).

Long Short-Term Memory (LSTM) dikembangkan untuk mengatasi keterbatasan RNN melalui cell state serta mekanisme input gate, forget gate, dan output gate yang mengendalikan aliran informasi sepanjang sekuens (Hochreiter & Schmidhuber, 1997;

Greff et al., 2017). Arsitektur ini sesuai untuk klasifikasi teks karena mampu mempertahankan konteks kata dalam urutan kalimat dan mempelajari pola yang membedakan setiap intent (Fajar Abdillah & Kusnawi, 2023; Luo et al., 2022). Dalam layanan pendidikan, model percakapan yang diintegrasikan ke platform digital dapat menyediakan akses informasi yang lebih cepat dan fleksibel, tetapi tetap memerlukan rancangan data, respons, dan antarmuka yang sesuai dengan kebutuhan pengguna (Lin & Chang, 2020; Villegas-Ch et al., 2021).

Penelitian terdahulu telah banyak membahas chatbot untuk pembelajaran dan layanan informasi, namun masih diperlukan implementasi yang secara khusus memanfaatkan data tanya-jawab aktual dari program Studi Independen, mengelompokkan informasi berdasarkan fase pre-register, running program, dan end program, serta menyediakan layanan langsung melalui Telegram (Okonkwo & Ade-Ibijola, 2021; Kuhail et al., 2023; Pérez et al., 2020). Berdasarkan kesenjangan tersebut, penelitian ini bertujuan mengembangkan dan mengevaluasi chatbot assistant berbasis algoritma LSTM sebagai layanan tanya jawab mentee di Infinite Learning. Kontribusi penelitian terletak pada pembangunan model klasifikasi intent berbasis data institusional, evaluasi menggunakan accuracy, precision, recall, dan F1-score, serta implementasi model pada Telegram Bot untuk membantu mentee memperoleh jawaban secara cepat dan mengurangi pertanyaan berulang kepada mentor (Fajar Abdillah & Kusnawi, 2023; Chen et al., 2017).

2. METODE

Penelitian ini menggunakan pendekatan eksperimental untuk mengembangkan dan mengevaluasi chatbot berbasis *Long Short-Term Memory* (LSTM) sebagai solusi dalam menjawab pertanyaan berulang dari mentee di program Studi Independen Infinite Learning. Rancangan penelitian ini mencakup serangkaian pengujian model guna menilai efektivitas LSTM dalam memahami masukan bahasa alami dan menghasilkan respons yang sesuai. Penelitian juga mengamati pengaruh variasi konfigurasi model, jumlah data pelatihan, dan parameter pelatihan terhadap performa chatbot.

Sasaran dalam penelitian ini adalah *chatbot* berbasis teks yang mampu menjawab pertanyaan dari *mentee* secara otomatis. Data yang digunakan terdiri dari tiga kategori, yaitu *pre-register*, *running program*, dan *end program*, yang mencerminkan fase-fase dalam pelaksanaan Studi Independen. Pengumpulan data dilakukan dengan dua metode, yaitu observasi dan dokumentasi. Melalui observasi yang dilakukan bersama lead program Web & Mobile Development yang berada di Batam, diperoleh data primer berupa pertanyaan dan jawaban aktual dari proses pendampingan. Sementara itu, dokumentasi dilakukan dengan mengumpulkan data sekunder dari sumber terpercaya seperti *Android Developers Documentation* dan *W3Schools*, yang relevan dengan materi Android Development dan Website Development.

Proses analisis data dimulai dengan tahapan *text preprocessing* untuk membersihkan dan menyiapkan data sebelum dimasukkan ke dalam model LSTM. Langkah-langkah dalam tahap *preprocessing* meliputi penghapusan tanda baca (*remove punctuations*), normalisasi kata melalui *lemmatization*, serta identifikasi kelas dan pola pertanyaan berdasarkan kategori *intents*. Setelah itu dilakukan proses *tokenization*, yaitu memecah kalimat menjadi token-token kecil yang dapat dikenali oleh mesin. Tahapan selanjutnya adalah *padding*, yaitu membuat panjang setiap *sequence* menjadi seragam dengan menambahkan nilai nol jika diperlukan. Setelah urutan teks disamakan panjangnya, dilakukan proses *embedding* untuk mengubah kata-kata menjadi representasi numerik yang dapat diproses oleh model LSTM.

Model LSTM yang digunakan dirancang khusus untuk menangani data sekuensial dan mengatasi keterbatasan dari *Recurrent Neural Network* (RNN) konvensional. Model ini mampu menyimpan informasi penting dari urutan sebelumnya dan mempertahankan konteks yang relevan selama proses pelatihan. Untuk mendukung pelatihan model, digunakan perangkat keras berupa laptop dengan spesifikasi *Processor Intel Core I3* dan *RAM 12 GB* serta perangkat lunak seperti Visual Studio Code dan Google Colab untuk membangun dan melatih model.

Seluruh proses penelitian mengikuti alur yang terstruktur, dimulai dari identifikasi masalah, pengumpulan data, *preprocessing data*, perancangan dan pelatihan model, evaluasi hasil, hingga implementasi *chatbot* ke dalam platform Telegram.

Metode yang digunakan dalam penelitian ini diharapkan dapat menghasilkan *chatbot* yang mampu menjawab pertanyaan dengan akurat, serta mendukung efektivitas komunikasi antara mentor dan *mentee* di lingkungan pembelajaran Infinite Learning.

3. HASIL DAN PEMBAHASAN

Apabila pengarang lebih dari dua orang, hanya nama pengarang pertama yang dituliskan. Nama pengarang selebihnya digantikan dengan 'dkk' (dan kawan-kawan). Tulisan 'dkk' dipisahkan dari nama pengarang, yang disebutkan dengan jarak, diikuti titik, dan diakhiri dengan koma. Contohnya: membaca adalah kegiatan interaksi antara pembaca dan penulis yang kehadirannya diwakili oleh teks (Susanto dkk., 1994: 8).

3.1. Pengumpulan Data

Proses pengumpulan data dilakukan sesuai metodologi yang telah dijelaskan, yaitu melalui observasi dan dokumentasi, untuk membangun basis data pertanyaan dan jawaban chatbot. Data dikategorikan dalam tiga fase interaksi pengguna terhadap program: *pre-register*, *running program*, dan *end program*.

Metode observasi dilaksanakan melalui sesi Zoom bersama *Head Program Mobile* dan *Web Development*, Reza Kurniawan. Dalam sesi selama dua jam ini, dikumpulkan 20 pasang pertanyaan dan jawaban yang sering diajukan *mentee* selama pembelajaran berlangsung. Hasilnya disusun dalam Tabel 1 berikut:

Tabel 1. Contoh Data Tanya-Jawab Hasil Observasi

No	Pertanyaan	Jawaban
1	Apa itu Infinite Learning?	Infinite Learning adalah divisi dari PT Kinema Systrans Multimedia yang berfokus pada pelatihan vokasi untuk mendukung kebutuhan tenaga

		terampil di ekosistem Nongsa Digital Park...
2	Bisa dijelaskan mengenai Program Studi yang ada di Infinite Learning?	Studi Independen itu program belajar satu semester dengan 30% teori dan 70% praktik. Ada lima jalur: Android, Web, Game, Linux, dan AI.
3	Apa manfaat mengikuti Studi Independen di Infinite Learning?	Konversi hingga 20 SKS, kesempatan magang, pembelajaran fleksibel, dua sertifikat, dan portofolio proyek nyata.

Metode dokumentasi digunakan untuk melengkapi data dengan materi dari sumber terpercaya, yaitu dokumentasi *Android Developers* dan *W3Schools*. Diperoleh 20 pertanyaan dan jawaban yang berkaitan dengan pengembangan aplikasi *Android* dan *Web*. Contoh data tersebut disajikan pada Tabel 2.

Tabel 2. Contoh Data Tanya-Jawab Hasil Dokumentasi

No	Pertanyaan	Jawaban
1	Apa itu Kotlin?	Kotlin adalah bahasa pemrograman dari JetBrains, lebih aman dan ringkas dibanding Java, serta kompatibel dengan kode Java.
2	Apa itu HTML?	HTML adalah bahasa markup standar untuk membuat struktur halaman web.
3	Apa itu CSS?	CSS adalah stylesheet yang mendeskripsikan tampilan dan pemformatan halaman web.

Setelah pengumpulan data, dilakukan proses augmentasi untuk memperkaya variasi pertanyaan dan jawaban agar model lebih adaptif terhadap variasi input. Augmentasi menggunakan model Gemini 2.5 Pro, menghasilkan 200 variasi pertanyaan dan 3 variasi jawaban untuk setiap *intent*. Semua data dari hasil augmentasi digabung dalam satu dataset berformat JSON. Dataset ini terdiri dari tiga elemen utama: *tag*, *patterns*, dan *responses*. Contoh struktur JSON dapat dilihat sebagai berikut:

```
{
  "intents": [
    {
      "tag": "infinite_learning_info",
      "patterns": [
        "Apa itu Infinite Learning?",
        "Bisa jelaskan Infinite Learning?",
        "Saya ingin tahu tentang Infinite Learning"
      ],
      "responses": [
        "Infinite Learning adalah divisi dari PT Kinema Systrans Multimedia...",
        "Infinite Learning merupakan bagian dari PT Kinema...",
        "Infinite Learning, salah satu divisi PT Kinema Systrans Multimedia..."
      ]
    }
  ]
}
```

3.2. Pre-Processing Data

Langkah selanjutnya adalah melakukan *pre-processing* guna menyiapkan data agar siap digunakan dalam pelatihan model chatbot. Proses ini dilakukan secara bertahap seperti yang dijelaskan dalam Tabel 3.

Tabel 3. Contoh Alur Pre-Processing Data

No	Tahap	Deskripsi	Hasil
1	Teks Awal	Kalimat asli dari input pengguna	Halo! Apa itu Infinite Learning ?

2	<i>Remove Punctuation</i>	Menghapus tanda baca	Halo Apa itu Infinite Learning
3	<i>Case Folding & Lemmatization</i>	Mengubah ke huruf kecil & Mengubah kata ke bentuk dasarnya	halo apa itu infinite learning
4	<i>Tokenization</i>	Memecah kalimat menjadi token	[24, 25, 26, 27, 28]
5	<i>Padding Text</i>	Menambahkan nilai 0 di awal untuk menyamakan panjang sekuens	[0, 0, 0, 0, 0, 0, 0, 0, 24, 25, 26, 27, 28]

Pre-processing dimulai dengan menghapus tanda baca yang tidak diperlukan (*remove punctuation*) untuk mengurangi *noise*. Selanjutnya dilakukan *case folding* dan *lemmatization* untuk menyamakan bentuk kata dan menyederhanakan variasi linguistik.

Tokenisasi kemudian memecah kalimat menjadi deretan angka representatif dari kata (*token*) sesuai urutan dalam kamus. Karena panjang kalimat berbeda-beda, maka diterapkan *padding text* untuk menyeragamkan panjang input. Label dari tiap *intent* diubah menjadi angka menggunakan teknik *label encoding*.

Dataset akhir kemudian dibagi menjadi data latih dan data uji dengan rasio 70:30, menggunakan metode *stratified split* agar distribusi label tetap seimbang di kedua bagian. Semua tahapan ini bertujuan untuk memastikan data terstruktur dengan baik dan optimal untuk pelatihan model chatbot berbasis LSTM.

3.3. Perancangan Model

Model LSTM dirancang setelah tahap *pre-processing* selesai. Tujuan utama perancangan ini adalah untuk membangun arsitektur jaringan yang mampu mengenali pola sekuensial dari input teks agar dapat mengklasifikasikan *tag intent* secara akurat.

Proses ini mencakup inisialisasi *hyperparameter* dan penentuan struktur model, mulai dari jumlah unit memori, *dropout*, fungsi aktivasi, hingga regularisasi. Model dibangun secara berlapis

dengan pendekatan Long Short-Term Memory (LSTM). Berikut adalah ringkasan arsitektur model.

Model: "sequential"

Layer (type)	Output Shape	Param #
embedding (Embedding)	(None, 17, 128)	187,008
lstm (LSTM)	(None, 17, 128)	131,584
lstm_1 (LSTM)	(None, 64)	49,408
dense (Dense)	(None, 64)	4,168
dropout (Dropout)	(None, 64)	0
dense_1 (Dense)	(None, 42)	2,738

Total params: 374,896 (1.43 MB)
Trainable params: 374,896 (1.43 MB)
Non-trainable params: 0 (0.00 B)

Gambar 1. Ringkasan Arsitektur Model LSTM

Model ini menggunakan dua lapisan LSTM. LSTM pertama mempertahankan seluruh urutan sekuens, sedangkan yang kedua hanya mengambil output terakhir. Lapisan *dense* di akhir memproses hasil LSTM dengan ReLU dan regularisasi L2, sebelum masuk ke lapisan *softmax* untuk klasifikasi multi-kelas.

Adam dipilih sebagai *optimizer* karena mampu secara adaptif menyesuaikan bobot dan laju pembelajaran selama proses pelatihan, serta banyak digunakan dalam proyek deep learning karena efisiensinya dalam menangani data yang besar dan parameter yang kompleks (SYIFA & DEWI, 2022).

3.4. Pelatihan Model

Setelah merancang arsitektur model LSTM, langkah selanjutnya adalah melakukan pelatihan model untuk mengoptimalkan bobot dan bias pada setiap *neuron* agar model mampu mengenali pola dari input pertanyaan serta menghasilkan prediksi label yang akurat.

Proses pelatihan dilakukan dengan menggunakan *fungsi fit()* dari pustaka Keras, dengan parameter *X_train* dan *y_train* sebagai data latih, serta *X_test* dan *y_test* sebagai data validasi. Model dilatih selama maksimal 50 *epoch* dengan ukuran *batch* sebesar 32, yang berarti bobot akan diperbarui setiap kali 32 sampel diproses.

Untuk menghindari *overfitting*, digunakan dua mekanisme *callback*, yaitu:

1. *EarlyStopping*, yang menghentikan pelatihan jika *val_accuracy* tidak meningkat selama 10 *epoch*.

2. *ModelCheckpoint*, yang menyimpan bobot model terbaik berdasarkan nilai *val_loss* terendah.

Hasil pelatihan menunjukkan bahwa proses dihentikan otomatis oleh *EarlyStopping* pada *epoch* ke-39. Nilai *val_loss* terbaik sebesar 0.0228 dicapai pada *epoch* ke-37. Model berhasil mencapai *training accuracy* 99.71% dan *validation accuracy* 99.72%. Nilai loss akhir: *training loss* 0.0279 dan *validation loss* 0.0237. Hasil pelatihan dapat dilihat pada Gambar 2 berikut.

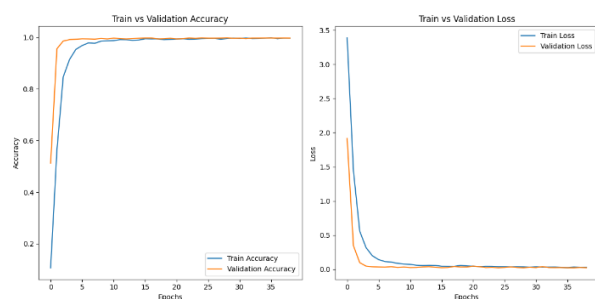
```

Epoch 34/50: 100ms/step - accuracy: 0.9951 - loss: 0.0336
Epoch 34: val_loss did not improve from 0.0224
Epoch 35/50: 100ms/step - accuracy: 0.9951 - loss: 0.0336
Epoch 35: val_loss did not improve from 0.0224
Epoch 36/50: 100ms/step - accuracy: 0.9963 - loss: 0.0287
Epoch 36: val_loss did not improve from 0.0224
Epoch 37/50: 100ms/step - accuracy: 0.9963 - loss: 0.0287
Epoch 37: val_loss did not improve from 0.0224
Epoch 38/50: 100ms/step - accuracy: 0.9968 - loss: 0.0259
Epoch 38: val_loss did not improve from 0.0224
Epoch 39/50: 100ms/step - accuracy: 0.9968 - loss: 0.0259
Epoch 39: val_loss did not improve from 0.0224
Epoch 40/50: 100ms/step - accuracy: 0.9968 - loss: 0.0259
Epoch 40: val_loss did not improve from 0.0224
Epoch 41/50: 100ms/step - accuracy: 0.9971 - loss: 0.0271
Epoch 41: val_loss did not improve from 0.0224
Epoch 42/50: 100ms/step - accuracy: 0.9971 - loss: 0.0271
Epoch 42: val_loss did not improve from 0.0224
Epoch 43/50: 100ms/step - accuracy: 0.9971 - loss: 0.0271
Epoch 43: val_loss did not improve from 0.0224
Epoch 44/50: 100ms/step - accuracy: 0.9971 - loss: 0.0271
Epoch 44: val_loss did not improve from 0.0224
Epoch 45/50: 100ms/step - accuracy: 0.9971 - loss: 0.0271
Epoch 45: val_loss did not improve from 0.0224
Epoch 46/50: 100ms/step - accuracy: 0.9971 - loss: 0.0271
Epoch 46: val_loss did not improve from 0.0224
Epoch 47/50: 100ms/step - accuracy: 0.9971 - loss: 0.0271
Epoch 47: val_loss did not improve from 0.0224
Epoch 48/50: 100ms/step - accuracy: 0.9971 - loss: 0.0271
Epoch 48: val_loss did not improve from 0.0224
Epoch 49/50: 100ms/step - accuracy: 0.9971 - loss: 0.0271
Epoch 49: val_loss did not improve from 0.0224
Epoch 50/50: 100ms/step - accuracy: 0.9971 - loss: 0.0271
Epoch 50: val_loss did not improve from 0.0224
Epoch 39: early stopping

```

Gambar 2. Hasil Pelatihan Lima Epoch Terakhir

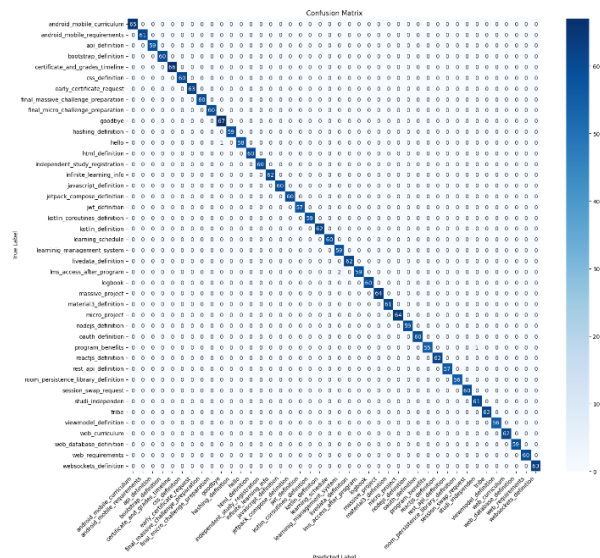
Grafik pelatihan, seperti yang ditunjukkan pada Gambar 3, memperlihatkan kurva akurasi dan *loss* yang *konvergen* cepat. Akurasi latih dan validasi stabil di atas 99% sejak *epoch* ke-10. *Loss* validasi stabil di kisaran 0.02–0.03 hingga akhir. Ini menunjukkan model mencapai performa optimal dan penghentian pelatihan oleh *EarlyStopping* efektif untuk mencegah *overfitting* dan menghemat waktu komputasi.



Gambar 3. Grafik Pelatihan Model

3.5. Evaluasi Model

Setelah model LSTM selesai dirancang dan dilatih, tahap berikutnya adalah melakukan evaluasi guna mengukur sejauh mana model mampu mengenali data baru secara akurat. Evaluasi dilakukan menggunakan *confusion matrix* dan menghasilkan metrik utama: *accuracy*, *precision*, *recall*, dan *f1-score*. Hasil evaluasi divisualisasikan pada Gambar 4.



Gambar 4. Confusion Matrix

Dari total 2.543 data uji, model berhasil mengklasifikasikan 2.536 data dengan benar. Akurasi model dihitung dengan rumus:

$$\begin{aligned} \text{Accuracy} &= \frac{\text{Jumlah prediksi benar}}{\text{Jumlah total data uji}} \\ &= \frac{2536}{2543} \approx 0,9972 \end{aligned}$$

Sebagai ilustrasi, evaluasi juga difokuskan pada label *studi_independen*. Terdapat 61 data aktual untuk label ini, dengan 60 diklasifikasikan dengan benar (*True Positive*), 1 tidak terdeteksi (*False Negative*), dan 2 data dari label lain salah diklasifikasikan sebagai *studi_independen* (*False Positive*). Maka diperoleh *precision*, *recall* dan *f1-score* dengan perhitungan sebagai berikut:

$$\begin{aligned} \text{Precision} &= \frac{TP}{TP + FP} = \frac{60}{60 + 2} = \frac{60}{62} \\ &\approx 0,9677 \\ \text{Recall} &= \frac{TP}{TP + FN} = \frac{60}{60 + 1} = \frac{60}{61} \\ &\approx 0,9836 \\ F1 &= 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \\ &= 2 \cdot \frac{0,9677 \cdot 0,9836}{0,9677 + 0,9836} \\ &= \frac{1,9027}{1,9513} \approx 0,9741 \end{aligned}$$

Hasil tersebut menunjukkan bahwa model memiliki keseimbangan yang baik antara presisi

dan sensitivitas dalam mengklasifikasikan *intent*.

Evaluasi secara menyeluruh terhadap seluruh label dalam dataset ditampilkan dalam *classification report* pada Gambar 5.

Classification Report:

	precision	recall	f1-score	support
android_mobile_curriculum	1.00	1.00	1.00	65
android_mobile_requirements	1.00	1.00	1.00	61
api_definition	1.00	1.00	1.00	59
bootstrap_definition	1.00	1.00	1.00	60
certificate_and_grades_timeline	1.00	1.00	1.00	66
css_definition	1.00	1.00	1.00	60
early_certificate_request	1.00	1.00	1.00	63
final_massive_challenge_preparation	1.00	1.00	1.00	60
final_micro_challenge_preparation	1.00	1.00	1.00	60
goodbye	0.99	1.00	0.99	67
hashing_definition	1.00	1.00	1.00	59
hello	1.00	0.98	0.99	59
html_definition	1.00	1.00	1.00	60
independent_study_registration	1.00	1.00	1.00	60
infinite_learning_info	1.00	1.00	1.00	62
javascript_definition	1.00	1.00	1.00	60
jetpack_compose_definition	1.00	1.00	1.00	60
jwt_definition	1.00	1.00	1.00	57
kotlin_coroutines_definition	1.00	1.00	1.00	59
kotlin_definition	1.00	1.00	1.00	62
learning_schedule	1.00	1.00	1.00	60
learning_management_system	0.97	1.00	0.98	59
livedata_definition	1.00	1.00	1.00	62
lms_access_after_program	1.00	0.97	0.98	61
logbook	1.00	1.00	1.00	60
massive_project	1.00	1.00	1.00	64
materials_definition	1.00	1.00	1.00	61
micro_project	1.00	1.00	1.00	64
nodejs_definition	1.00	1.00	1.00	59
oauth_definition	1.00	1.00	1.00	60
program_benefits	1.00	0.98	0.99	56
reactjs_definition	1.00	1.00	1.00	62
rest_api_definition	1.00	1.00	1.00	57
room_persistence_library_definition	1.00	1.00	1.00	56
session_swap_request	1.00	1.00	1.00	60
studi_independen	0.98	1.00	0.99	61
tribe	1.00	1.00	1.00	62
viewmodel_definition	1.00	1.00	1.00	56
web_curriculum	1.00	1.00	1.00	62
web_database_definition	1.00	1.00	1.00	59
web_requirements	1.00	1.00	1.00	60
websockets_definition	1.00	1.00	1.00	63
accuracy			1.00	2543
macro avg	1.00	1.00	1.00	2543
weighted avg	1.00	1.00	1.00	2543

Gambar 5. Classification Report

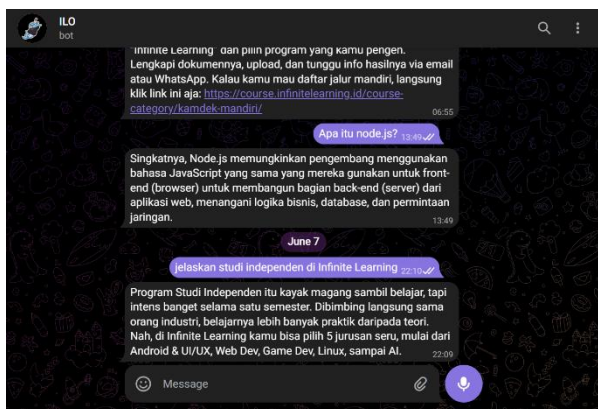
3.6. Implementasi Model ke Telegram Bot

Setelah model LSTM selesai dilatih dan dievaluasi, tahap selanjutnya adalah mengintegrasikan model ke dalam layanan *chatbot* berbasis Telegram. Proses ini dilakukan menggunakan Telegram Bot API yang memungkinkan komunikasi antara pengguna dan model melalui aplikasi Telegram.

Bot didaftarkan melalui *BotFather* untuk mendapatkan *token* API. Model LSTM yang telah disimpan sebelumnya dimuat kembali, bersama dengan *objek tokenizer* dan *label encoder* yang digunakan saat pelatihan. Setiap pesan dari pengguna akan diproses terlebih dahulu menggunakan tahapan *pre-processing* yang sama seperti saat pelatihan, seperti menghapus tanda baca, *case folding*, *lemmatization*, tokenisasi, dan *padding*.

Setelah itu, input yang telah diproses akan dikirim ke model untuk diprediksi. Hasil prediksi berupa *label intent* kemudian digunakan untuk memilih respons yang sesuai dari dataset. Respons ini dikirim kembali ke pengguna

melalui antarmuka Telegram secara real-time. Hasil *chatbot* pada Telegram bisa dilihat pada Gambar 6.



Gambar 6. ILO Bot Telegram

3.7. Pengujian *Chatbot*

Setelah *chatbot* berhasil diintegrasikan dengan platform Telegram, tahap selanjutnya adalah melakukan pengujian fungsionalitas. Tujuan pengujian ini adalah untuk menilai kemampuan *chatbot* dalam memahami pertanyaan pengguna dan memberikan jawaban yang relevan sesuai dengan intent yang dimaksud.

Pengujian dilakukan dengan memberikan berbagai pertanyaan kepada *bot* melalui Telegram, mencakup variasi struktur kalimat dan kosa kata yang berbeda namun memiliki makna yang serupa. Pertanyaan-pertanyaan tersebut mengacu pada *intent* yang telah didefinisikan sebelumnya pada dataset pelatihan. Hasil respons dari *chatbot* kemudian dievaluasi secara manual berdasarkan kesesuaian antara pertanyaan, jawaban, dan intent yang terdeteksi. Contoh pengujian *chatbot* terhadap sejumlah pertanyaan ditampilkan di bawah ini.

1. *Intent* : *program_benefits*
 Pertanyaan : Apa saja manfaat yang akan diperoleh peserta dari program Studi Independen ini?
 Jawaban : Keunggulan program mencakup pengakuan kredit hingga 20 SKS, peluang magang dan bekerja di perusahaan global, sistem belajar yang adaptif, memperoleh dua sertifikat (Studi

Independen dan Magang), serta kesempatan mengembangkan portofolio melalui proyek riil yang bisa ditampilkan.

Keterangan : Sesuai

Pertanyaan : Kalo ikut SIB di IL dapat apa aja?

Jawaban : Keuntungan yang diperoleh meliputi konversi 20 SKS, kesempatan karir di perusahaan internasional, metode pembelajaran fleksibel, dua jenis sertifikasi (Program Studi Independen dan Magang), plus pengalaman membangun portofolio profesional berbasis proyek nyata.

Keterangan : Sesuai

2. *Intent* : *jetpack_compose*

Pertanyaan : Mohon berikan definisi dari Jetpack Compose dalam pengembangan Android.

Jawaban : Jetpack Compose adalah toolkit UI modern asli Android untuk membangun antarmuka pengguna (UI) deklaratif. Ini menyederhanakan dan mempercepat pengembangan UI di Android dengan lebih sedikit kode, alat yang canggih, dan API Kotlin yang intuitif.

Keterangan : Sesuai

Pertanyaan : jetpak compse itu buat apaan sih



Jawaban : JavaScript (sering disingkat JS) adalah bahasa pemrograman tingkat tinggi, dinamis, dan serbaguna yang utamanya digunakan sebagai bagian dari halaman web untuk membuat konten interaktif, seperti animasi, validasi formulir, dan pembaruan konten secara dinamis.

Keterangan : Tidak Sesuai

3. *Intent* : fallback_intent
Pertanyaan : Jelaskan tentang penciptaan dunia

Jawaban : Maaf, ILO belum bisa memahami pertanyaan kamu. Silakan coba dengan pertanyaan lain ya

Keterangan : Sesuai

Pertanyaan : Tolong buat kan saya ringkasan tentang sejarah Indonesia.

Jawaban : Halo! Selamat datang di Infinite Learning. Ada yang bisa ILO bantu?

Keterangan : Tidak Sesuai

Hasil pengujian menunjukkan bahwa *chatbot* mampu memberikan respons yang konsisten dan sesuai terhadap berbagai bentuk pertanyaan dalam *intent* yang sama. Hal ini mengindikasikan bahwa model LSTM yang digunakan memiliki kemampuan generalisasi yang baik dalam memahami variasi bahasa alami.

4. PENUTUP

4.1. Kesimpulan

Penelitian ini berhasil mengembangkan *chatbot assistant* berbasis algoritma *Long Short-Term Memory* (LSTM) untuk mendukung layanan tanya jawab bagi *mentee* di Infinite Learning. Pengembangan dilakukan melalui

tahapan sistematis yang mencakup pengumpulan dan *pre-processing data*, perancangan arsitektur model LSTM, pelatihan model, serta integrasi ke platform Telegram. Arsitektur model yang digunakan terdiri dari lapisan embedding, dua lapisan LSTM, dan dense layer, dan di-*deploy* menggunakan Railway.app.

Model menunjukkan performa klasifikasi yang sangat baik, dengan akurasi pelatihan sebesar 99,71% dan akurasi validasi sebesar 99,72%. Evaluasi melalui *confusion matrix* serta pengujian terbatas pada sepuluh skenario pertanyaan menunjukkan bahwa *chatbot* mampu memberikan respons yang sesuai secara konsisten. Hasil ini mengindikasikan bahwa model LSTM yang dibangun memiliki kemampuan generalisasi yang baik dan layak diterapkan dalam konteks layanan informasi berbasis percakapan.

4.2. Saran

Terdapat beberapa aspek yang dapat dikembangkan lebih lanjut untuk meningkatkan fungsionalitas dan cakupan *chatbot* di masa mendatang. Salah satu arah pengembangan yang memungkinkan adalah memperluas cakupan dataset dengan memasukkan pertanyaan-pertanyaan yang lebih spesifik dan mencakup berbagai *learning path* lain di Infinite Learning, seperti *Game Development* dan *Advance AI*. Selain itu, eksplorasi terhadap arsitektur yang lebih modern seperti *Transformer* dapat menjadi alternatif untuk meningkatkan pemahaman konteks dalam percakapan yang lebih kompleks dan panjang. Penggunaan model tersebut juga berpotensi memberikan kemampuan *chatbot* dalam menangani dialog yang lebih natural. Di samping itu, integrasi fitur untuk memproses dan merespons dalam format multimedia seperti gambar atau dokumen juga menjadi peluang untuk memperluas kasus penggunaan *chatbot* secara praktis di lingkungan pembelajaran

5. DAFTAR PUSTAKA

Chen, H., Liu, X., Yin, D., & Tang, J. (2017). A survey on dialogue systems: Recent advances and new frontiers. *ACM SIGKDD Explorations Newsletter*, 19(2), 25–35.



- <https://doi.org/10.1145/3166054.3166058>
- Chen, L., Chen, P., & Lin, Z. (2020). Artificial intelligence in education: A review. *IEEE Access*, 8, 75264–75278. <https://doi.org/10.1109/ACCESS.2020.2988510>
- Fajar Abdillah, M., & Kusnawi, K. (2023). Comparative analysis of long short-term memory architecture for text classification. *ILKOM Jurnal Ilmiah*, 15(3), 455–464. <https://doi.org/10.33096/ilkom.v15i3.1906.455-464>
- Galassi, A., Lippi, M., & Torroni, P. (2021). Attention in natural language processing. *IEEE Transactions on Neural Networks and Learning Systems*, 32(10), 4291–4308. <https://doi.org/10.1109/TNNLS.2020.3019893>
- Greff, K., Srivastava, R. K., Koutník, J., Steunebrink, B. R., & Schmidhuber, J. (2017). LSTM: A search space odyssey. *IEEE Transactions on Neural Networks and Learning Systems*, 28(10), 2222–2232. <https://doi.org/10.1109/TNNLS.2016.2582924>
- Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8), 1735–1780. <https://doi.org/10.1162/neco.1997.9.8.1735>
- Kerly, A., Hall, P., & Bull, S. (2007). Bringing chatbots into education: Towards natural language negotiation of open learner models. *Knowledge-Based Systems*, 20(2), 177–185. <https://doi.org/10.1016/j.knosys.2006.11.014>
- Kuhail, M. A., Alturki, N., Alramlawi, S., & Alhejori, K. (2023). Interacting with educational chatbots: A systematic review. *Education and Information Technologies*, 28(1), 973–1018. <https://doi.org/10.1007/s10639-022-11177-3>
- Labadze, L., Grigolia, M., & Machaidze, L. (2023). Role of AI chatbots in education: Systematic literature review. *International Journal of Educational Technology in Higher Education*, 20, Article 56. <https://doi.org/10.1186/s41239-023-00426-1>
- Lasijan, T. G., Santoso, R., & Hakim, A. R. (2023). Prediksi harga emas dunia menggunakan metode long-short term memory. *Jurnal Gaussian*, 12(2), 287–295. <https://doi.org/10.14710/j.gauss.12.2.287-295>
- Lee, Y.-F., Hwang, G.-J., & Chen, P.-Y. (2022). Impacts of an AI-based chatbot on college students' after-class review, academic performance, self-efficacy, learning attitude, and motivation. *Educational Technology Research and Development*, 70(5), 1843–1865. <https://doi.org/10.1007/s11423-022-10142-8>
- Lin, M. P.-C., & Chang, D. (2020). Enhancing post-secondary writers' writing skills with a chatbot: A mixed-method classroom study. *Educational Technology & Society*, 23(1), 78–92.
- Luo, B., Lau, R. Y. K., Li, C., & Si, Y. (2022). A critical review of state-of-the-art chatbot designs and applications. *WIREs Data Mining and Knowledge Discovery*, 12(1), e1434. <https://doi.org/10.1002/widm.1434>
- Nova, S., Khotimah, N., & Wahyuningrum, M. Y. A. (2024). Pemanfaatan chatbot menggunakan natural language processing untuk pembelajaran dasar-dasar GUI Tkinter pada bahasa pemrograman Python. *Jurnal Ilmiah Teknik*, 3(1), 58–65. <https://doi.org/10.56127/juit.v3i1.1162>
- Okonkwo, C. W., & Ade-Ibijola, A. (2021). Chatbots applications in education: A systematic review. *Computers and Education: Artificial Intelligence*, 2, 100033. <https://doi.org/10.1016/j.caeai.2021.100033>
- Pérez, J. Q., Daradoumis, T., & Puig, J. M. M. (2020). Rediscovering the use of chatbots in education: A systematic literature review. *Computer Applications*



- in Engineering Education*, 28(6), 1549–1565. <https://doi.org/10.1002/cae.22326>
- Puspitasari, A., Paradhita, A. N., Tineka, Y. W., Sulistyowati, V., Noriska, N. K. S., & Haryanto. (2024). Natural language processing (NLP) technology for chatbot website. *Jurnal Penelitian Pendidikan IPA*, 10(Special Issue), 319–324. <https://doi.org/10.29303/jppipa.v10iSpecialIssue.8241>
- Rahman, R. A., Risma, P., Oktarina, Y., & Yudha, H. M. (2024). Prediksi temperatur lingkungan dengan recurrent neural network menggunakan data historis iradiasi matahari. *Journal of Applied Smart Electrical Network and Systems*, 5(1), 16–21. <https://doi.org/10.52158/jasens.v5i1.862>
- Smutny, P., & Schreiberova, P. (2020). Chatbots for learning: A review of educational chatbots for the Facebook Messenger. *Computers & Education*, 151, 103862. <https://doi.org/10.1016/j.compedu.2020.103862>
- Tamrakar, R., & Wani, N. (2021). Design and development of chatbot: A review. In *Proceedings of the International Conference on Latest Trends in Civil, Mechanical and Electrical Engineering*.
- Villegas-Ch, W., García-Ortiz, J., Mullo-Ca, K., Sánchez-Viteri, S., & Roman-Cañizares, M. (2021). Implementation of a virtual assistant for the academic management of a university with the use of artificial intelligence. *Future Internet*, 13(4), 97. <https://doi.org/10.3390/fi13040097>
- Wollny, S., Schneider, J., Di Mitri, D., Weidlich, J., Rittberger, M., & Drachsler, H. (2021). Are we there yet? A systematic literature review on chatbots in education. *Frontiers in Artificial Intelligence*, 4, 654924. <https://doi.org/10.3389/frai.2021.654924>
- Young, T., Hazarika, D., Poria, S., & Cambria, E. (2018). Recent trends in deep learning based natural language processing. *IEEE Computational Intelligence Magazine*, 13(3), 55–75. <https://doi.org/10.1109/MCI.2018.2840738>
- Zawacki-Richter, O., Marín, V. I., Bond, M., & Gouverneur, F. (2019). Systematic review of research on artificial intelligence applications in higher