

**LAPORAN UAS DEEP LEARNING**

**CHATBOT LAYANAN PEMINJAMAN FASILITAS**

**UNIB**



**DISUSUN OLEH :**

- |                                       |                    |
|---------------------------------------|--------------------|
| <b>1. Elisa</b>                       | <b>(G1A021008)</b> |
| <b>2. Zhafirah Nur Shadrina Putri</b> | <b>(G1A021028)</b> |
| <b>3. Arya Bigtra Dhieva</b>          | <b>(G1A021088)</b> |

**PROGRAM STUDI INFORMATIKA**

**FAKULTAS TEKNIK**

**UNIVERSITAS BENGKULU**

**2024**

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Teknologi terus mengalami perkembangan pesat, terutama di bidang teknologi informasi. Kemajuan ini telah membawa perubahan signifikan dalam pola interaksi pada mahasiswa. Penerapan teknologi informasi menjadi keharusan bagi Universitas Bengkulu guna meningkatkan kualitas pelayanan kepada Mahasiswa. Salah satu wujud dari kemajuan teknologi informasi tersebut adalah chatbot(Wicaksana et al., 2024). Universitas Bengkulu merupakan salah satu perguruan tinggi yang perlu memanfaatkan teknologi untuk meningkatkan efektifitas layanan, seperti pembuatan chatbot untuk peminjaman fasilitas di Universitas Bengkulu. Mahasiswa seringkali terhambat dan bingung dalam mencari informasi terkait tata cara peminjaman fasilitas di Universitas Bengkulu. Oleh karena itu pengembangan chatbot untuk informasi mengenai peminjaman fasilitas di Universitas Bengkulu diperlukan agar memudahkan para mahasiswa dalam mengakses informasi mengenai peminjaman fasilitas secara mudah dan efektif.

Chatbot adalah sebuah program komputer yang diciptakan untuk meniru percakapan dengan menggunakan kecerdasan buatan (artificial intelligence). Salah satu bagian dari kecerdasan buatan adalah machine learning (ML), yang dapat memproses dan mempelajari data secara mandiri melalui proses pelatihan. Chatbot bekerja dengan memahami pesan yang diterima dari pengguna, kemudian memproses kata-kata yang diucapkan oleh pengguna, menentukan dan menjalankan perintah yang dibutuhkan oleh pengguna, dan akhirnya menyampaikan hasil dari kesimpulan kepada pengguna(Putri & Ramadhan, 2024).

Salah satu metode yang efektif dalam pengembangan chatbot adalah Long Short-Term Memory (LSTM). LSTM adalah sebuah algoritma Deep Learning yang termasuk dalam Recurrent Neural Network (RNN). Dengan LSTM, chatbot dapat

memahami konteks percakapan secara lebih mendalam dan menghasilkan tanggapan yang lebih sesuai (Yuniati & Gurning, 2024). Dengan pengembangan chatbot ini, Universitas Bengkulu diharapkan mampu meningkatkan kualitas layanan peminjaman fasilitas yang dapat memudahkan mahasiswa serta menghemat waktu dalam mencari informasi.

## **1.2 Rumusan Masalah**

1. Bagaimana mengembangkan chatbot yang dapat memberikan informasi mengenai peminjaman fasilitas di Universitas Bengkulu secara responsif dan mudah diakses?
2. Bagaimana menerapkan algoritma Long Short-Term Memory (LSTM) untuk membuat chatbot?

## **1.3 Batasan Masalah**

Dataset dikumpulkan secara kolektif dari beberapa staff administrasi Universitas Bengkulu melalui pertanyaan lisan.

## **1.4 Tujuan Penelitian**

1. Mengembangkan chatbot yang dapat memberikan informasi mengenai peminjaman fasilitas di Universitas Bengkulu secara responsif dan mudah diakses.
2. Menerapkan algoritma Long Short-Term Memory (LSTM) untuk membuat chatbot.

## **1.5 Manfaat Penelitian**

1. Membantu mahasiswa Universitas Bengkulu dalam mengakses informasi mengenai peminjaman fasilitas secara mudah dan hemat waktu.

2. Membantu Universitas Bengkulu dalam menyediakan layanan informasi mengenai peminjaman fasilitas yang akurat, dan mudah diakses.

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1 Deep Learning**

Deep Learning adalah salah satu cabang Machine Learning yang terinspirasi dari struktur dan cara kerja otak manusia, yang dikenal sebagai jaringan saraf tiruan atau Artificial Neural Network. Secara sederhana, Deep Learning merujuk pada teknik pembelajaran mesin yang memanfaatkan jaringan saraf tiruan dengan banyak lapisan neuron untuk mengidentifikasi fitur kompleks dan merepresentasikan data input secara hierarkis. Perbedaan utama antara Deep Learning dan metode pembelajaran mesin konvensional terletak pada jumlah serta kompleksitas lapisan neuronnya. Kemampuan Deep Learning untuk mempelajari pola dari data yang rumit dan abstrak membuatnya sangat unggul dalam berbagai aplikasi, seperti pengenalan gambar, pemrosesan bahasa alami, pengenalan suara, dan banyak bidang lainnya. Salah satu algoritma Deep Learning yang populer adalah Recurrent Neural Network (RNN)(Lubis et al., 2024).

#### **2.2 ChatBot**

Chatbot adalah salah satu aplikasi dari Natural Language Processing (NLP), yang merupakan cabang dari kecerdasan buatan atau Artificial Intelligence (AI). Teknologi ini memanfaatkan kecerdasan buatan dan pemrosesan bahasa alami untuk memahami teks dari pengguna serta memberikan respons otomatis sesuai dengan pertanyaan atau perintah yang diterima melalui pesan. Analisis chatbot melibatkan proses pengolahan data terkait chatbot dengan tujuan memahami cara kerja sistem, mengidentifikasi kendala dan peluang, serta memberikan rekomendasi solusi untuk meningkatkan kinerja sistem secara keseluruhan.

#### **2.3 LSTM**

LSTM adalah sebuah algoritma Deep Learning yang termasuk dalam Recurrent Neural Network (RNN). Dengan LSTM, chatbot dapat memahami konteks percakapan secara lebih mendalam dan menghasilkan tanggapan yang lebih sesuai. Model klasifikasi yang dibangun dengan LSTM mampu memahami dan mengklasifikasi beberapa pertanyaan dari pengguna berdasarkan kategorinya.

## **BAB III**

### **HASIL DAN PEMBAHASAN**

#### **3.1 Business Understanding**

Chatbot adalah perangkat lunak yang dapat berkomunikasi dengan manusia menggunakan bahasa alami. ChatBot dapat menjadi salah satu alternatif dalam membantu mahasiswa untuk mendapatkan informasi mengenai tata cara peminjaman fasilitas di Universitas Bengkulu. ChatBot merupakan teknologi yang efektif dalam membantu mahasiswa mendapatkan informasi secara akurat dan menghemat waktu. Oleh karena itu pengembangan ChatBot di Universitas Bengkulu diperlukan untuk kebutuhan layanan peminjaman fasilitas yang dapat membawa perubahan signifikan pada mahasiswa yang membutuhkan informasi secara real time dan akurat.

ChatBot menggunakan LSTM yang merupakan algoritma dari Deep Learning. LSTM atau Long Short Term Memory termasuk kedalam salah satu Recurrent Neural Network (RNN). Model klasifikasi yang dibangun dengan LSTM mampu memahami dan mengklasifikasi beberapa pertanyaan dari pengguna berdasarkan kategori yang telah ditentukan sebelumnya. Dengan pengembangan ChatBot ini, diharapkan mampu meningkatkan kualitas Universitas Bengkulu dalam layanan peminjaman fasilitas dan menghemat waktu mahasiswa dalam mendapatkan informasi yang akurat.

#### **3.2 Data Understanding**

Dataset yang digunakan pada pembuatan ChatBot “Generasi Unib” disusun dalam format JSON yang dibuat secara manual berdasarkan informasi terkait peminjaman fasilitas di Universitas Bengkulu. Data ini dikumpulkan secara kolektif

dari beberapa staff administrasi Universitas Bengkulu melalui wawancara dan pertanyaan – pertanyaan lisan.

Struktur dataset terdiri dari intents yang merupakan kumpulan kategori, yang mana setiap intents dilengkapi dengan tags yang digunakan untuk mengidentifikasi setiap intents. Pola pertanyaan atau patterns yang sebelumnya dikelompokkan dengan tags akan menggambarkan bagaimana pertanyaan yang diberikan mahasiswa, kemudian chatbot akan memberikan jawaban (responses) yang sesuai dengan pertanyaan yang diajukan.

	patterns	tags
0	assalamualaikum	greeting
1	hallo	greeting
2	hai	greeting
3	hi	greeting
4	halo	greeting
...	...	...
69	oke	goodbye
70	thankyou	goodbye
71	stop	goodbye
72	ok	goodbye
73	bye	goodbye

Gambar 3 1 Dataset

### 3.3 Data Preparation

#### 3.3.1 Menghilangkan Tanda Baca

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

Gambar 3 2 kode menghapus tanda baca

Kode diatas merupakan proses pembersihan data dengan menghapus tanda baca dalam dataframe. Setiap karakter akan diperiksa dan akan menghapus tanda baca seperti '!' (tanda seru) ',' (tanda koma) '.' (tanda titik sebagai berhenti) '?' (tanda tanya) dan tanda baca yang lain. Jika bukan tanda baca, maka karakter akan disimpan.

### 3.3.2 Lemmatization

```
lemmatizer = WordNetLemmatizer()
words = [lemmatizer.lemmatize(w.lower()) for w in words if w not in ignore_words]
words = sorted(list(set(words)))

print (len(words), "unique lemmatized words", words)
```

Gambar 3 3 kode lemmatization

*Lemmatization* merupakan proses mengubah kata kedalam bentuk dasarnya berdasarkan makna dari kata tersebut. Setelah *lemmatization*, kata akan disusun kembali menjadi sebuah teks utuh menggunakan fungsi join.

### 3.3.3 Tokenization

```
tokenizer = Tokenizer(num_words=2000)
tokenizer.fit_on_texts(data['patterns'])
train = tokenizer.texts_to_sequences(data['patterns'])

# Menampilkan hasil tokenisasi berupa daftar urutan numerik untuk setiap pola dalam dataset
train
```

Gambar 3 4 kode tokenization

Tokenization merupakan proses mengubah kata – kata menjadi angka, sehingga model dapat memprosesnya. Pada tahap ini, proses tokenisasi dilakukan dengan menggunakan TensorFlow Keras.

### 3.3.4 Padding

```
x_train = pad_sequences(train)
print(x_train) # Padding sequences
```

Gambar 3 5 kode padding

Kode ini adalah proses untuk menerapkan padding pada data yang telah di tokenisasi. Proses padding adalah proses yang akan membuat setiap kalimat pada teks memiliki panjang yang sama.

### 3.3.5 Label Encoding

```
# Membuat instance dari LabelEncoder untuk mengubah data kategori menjadi angka
le = LabelEncoder()

# Mengubah kolom 'tags' menjadi nilai numerik
y_train = le.fit_transform(data['tags'])

# Menampilkan hasil encoding output
# Output ini adalah representasi numerik dari label yang sebelumnya berupa kategori teks
print(y_train)
```

Gambar 3 6 kode encoder

Kode ini adalah proses untuk mengubah data kategori menjadi angka. Setiap kategori dalam kolom tags diberikan angka unik sehingga hasilnya adalah representasi numerik dari label yang sebelumnya adalah tags.

### 3.3.6 Menghitung Jumlah Kelas Unik

```
output_length = le.classes_.shape[0]

# Menampilkan panjang output (jumlah kelas unik)
print("output length: ", output_length)
```

Gambar 3 7 kode menghitung jumlah kelas unik

Kode ini merupakan proses untuk menghitung jumlah kelas unik yang berada pada data label setelah diakukannya proses encoding.

## 3.4 Modeling



```

# Membuat arsitektur model menggunakan TensorFlow Keras Functional API
i = Input(shape=(input_shape,))

# Layer Embedding: Mengubah indeks kata menjadi representasi vektor berdimensi 20
x = Embedding(vocabulary + 1, 20)(i)

# Layer LSTM: Menambahkan Long Short-Term Memory layer untuk mempelajari urutan kata
x = LSTM(20, return_sequences=True)(x)

# Layer Flatten: Mengubah keluaran LSTM (3D tensor) menjadi vektor 1D untuk layer Dense
x = Flatten()(x)

# Layer Dense: Menambahkan layer fully connected dengan jumlah neuron sesuai jumlah kelas
x = Dense(output_length, activation="softmax")(x)

# Membuat model dengan TensorFlow Keras Functional API
model = Model(i, x)

# Kompilasi model: Menentukan fungsi loss, optimizer, dan metrik evaluasi
model.compile(loss="sparse_categorical_crossentropy", optimizer='adam', metrics=['accuracy'])

```

Gambar 3 8 kode modeling

Model dimulai dengan layer input yang menerima data dengan panjang input sesuai input\_shape. Kemudian, layer Embedding mengubah indeks data menjadi vector berdimensi 20. Layer LSTM digunakan untuk mempelajari pola urutan data. Layer dense mengaktifkan softmax untuk menghasilkan probabilitas klasifikasi untuk setiap label. Lalu model dikompilasi menggunakan sparse\_categorical\_crossentropy, optimizer adam, dan metric accuracy.

### 3.5 Melatih Model

```

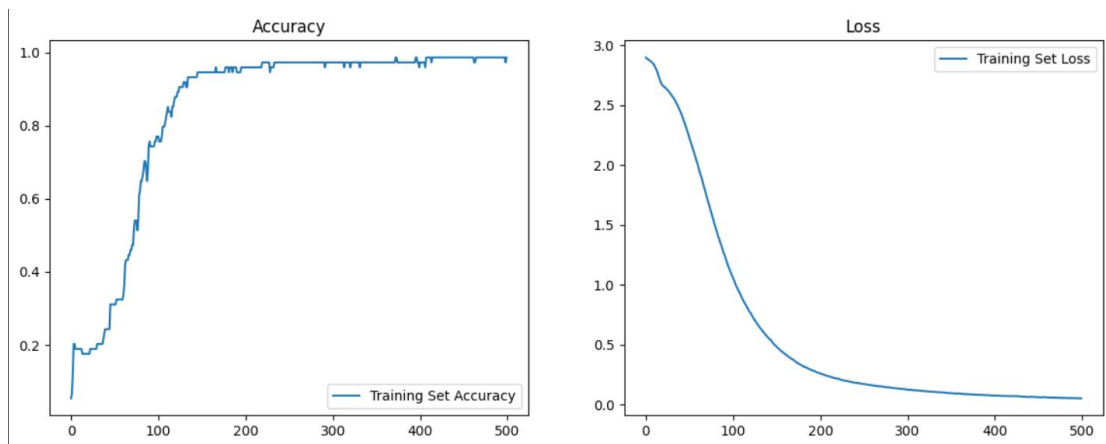
▶ train = model.fit(x_train, y_train, epochs=500)

```

Gambar 3 9 kode untuk melatih model

Kode ini merupakan proses pelatihan model selama 500 epochs menggunakan data pelatihan x\_train dan y\_train. Model mencatat metric accuracy dan loss disetiap epochs. Pelatihan ini bertujuan agar model dapat mengenali pola dalam data.

### 3.6 Evaluation



Gambar 3 10 grafik akurasi dan loss

Dari proses training didapatkan akurasi mendekati 1 yang berarti model hampir sempurna dalam memprediksi data pelatihan. Sedangkan grafik loss menunjukkan penurunan dengan sangat cepat hingga mendekati 0, yang menunjukkan bahwa model hampir tidak melakukan kesalahan pada data pelatihan.

### 3.7 Testing

```

import random
import string
import numpy as np
batas_probabilitas = 0.7
# Membuat Input Chat
while True:
    texts_p = []
    prediction_input = input(' Kamu : ')

    # Preprocessing teks
    prediction_input = [letters.lower() for letters in prediction_input if letters not in string.punctuation]
    prediction_input = ''.join(prediction_input)
    prediction_input = lemmatizer.lemmatize(prediction_input)
    texts_p.append(prediction_input)

    # Tokenisasi dan Padding
    prediction_input = tokenizer.texts_to_sequences(texts_p)
    prediction_input = np.array(prediction_input).reshape(-1)
    prediction_input = pad_sequences([prediction_input], input_shape)

    # Mendapatkan hasil keluaran pada model
    output = model.predict(prediction_input)
    output_probabilitas = round(output.max(), 2)
    output = output.argmax()

    # Memeriksa probabilitas
    if output_probabilitas < batas_probabilitas:
        print(" GenerasiUnib : Maaf, saya tidak mengerti pertanyaan anda.")
    else:
        # Menemukan respon sesuai data tag dan memainkan voice bot
        response_tag = le.inverse_transform([output])[0]
        print(" GenerasiUnib : ", random.choice(responses[response_tag]))

    time.sleep(0.08)
    print("=*60 + "\n")

    # Tambahkan respon 'goodbye' agar bot bisa berhenti
    if response_tag == "goodbye":
        break

```

```

Kamu : hai
1/1 ----- 0s 273ms/step
GenerasiUnib : Yuk, sampaikan saja kebutuhan Anda! Silakan pilih fasilitas yang ingin Anda pinjam:
1. Peminjaman Gedung
2. Peminjaman Lapangan
3. Peminjaman Barang
4. Informasi Umum

Ketik angka atau nama kategori untuk melanjutkan.
=====

Kamu : pinjam gedung
1/1 ----- 0s 21ms/step
GenerasiUnib : Berikut adalah gedung yang bisa Anda pinjam di Universitas Bengkulu:
- Gedung Serbaguna (GSG): Untuk acara besar seperti seminar atau wisuda.
- Ruang Kuliah: Untuk kegiatan belajar atau diskusi.
- Aula Fakultas: Untuk acara tingkat fakultas.

Pastikan Anda menghubungi pihak administrasi universitas atau fakultas terkait untuk proses peminjaman dan ketersediaan.
=====

Kamu : gsg
1/1 ----- 0s 22ms/step
GenerasiUnib : Untuk syarat peminjaman, silakan membawa:
1. Kartu Tanda Mahasiswa (KTM).
2. Surat Resmi Peminjaman.
3. Surat Izin Kegiatan
=====

Kamu : ok
1/1 ----- 0s 33ms/step
GenerasiUnib : Terima kasih telah berbincang dengan sahabatunib. Sampai jumpa! 😊
=====

```

*Gambar 3 11 hasil testing ChatBot*

Dari hasil diatas menunjukkan bahwa model memberikan informasi mengenai peminjaman fasilitas di Universitas Bengkulu berdasarkan input yang pengguna berikan. Chatbot dapat mengenali input sebagai sapaan maupun kalimat perpisahan dan

memberikan response yang sesuai. Secara keseluruhan, ChatBot menunjukkan kemampuan dalam memahami konteks dan memberikan response yang sesuai dengan kebutuhan mahasiswa.

### **3.8 Analisa Bagaimana Model Dapat Dikatakan Sebagai Deep Learning dan Bukan Shallow Learn**

Model yang dibuat untuk ChatBot “Generasi Unib” merupakan model Deep Learning dan bukan Shallow Learn, karena model ini menggunakan arsitektur multi lapisan yang dapat memproses data kompleks. Terdapat layer LSTM dan Embedding yang merupakan ciri khas Deep learning. Model LSTM ini juga mempunyai banyak lapisan seperti input, embedding, flatten, maupun dense. Model dalam ChatBot “Generasi Unib” dibangun menggunakan TensorFlow Keras. Sedangkan Shallow Learn adalah metode yang hanya menggunakan satu atau dua lapisan dan hanya dapat menangkap pola yang sederhana dalam data.

## **BAB IV**

### **KESIMPULAN**

#### **4.1 KESIMPULAN**

Dari pengujian model LSTM dalam membangun ChatBot layanan pinjaman fasilitas di Universitas Bengkulu dapat disimpulkan bahwa model mendapatkan akurasi yang cukup bagus yaitu dengan rata – rata akurasi sebesar 0.9893. Nilai loss pada model menunjukkan angka yang terbilang kecil yaitu sebesar 0.0448. Dengan ini, Proyek kali ini berhasil menghasilkan ChatBot yang dapat memberikan respon otomatis terhadap pertanyaan mahasiswa terkait peminjaman fasilitas di Universitas Bengkulu.

## DAFTAR PUSTAKA

- Lubis, N., Siambaton, M. Z., & Aulia, R. (2024). Implementasi Algoritma Deep Learning pada Aplikasi Speech to Text Online dengan Metode Recurrent Neural Network (RNN). *Sudo Jurnal Teknik Informatika*, 3(3), 113–126. <https://doi.org/10.56211/sudo.v3i3.583>
- Putri, T. E., & Ramadhan, G. (2024). Penerapan Chatbot sebagai Alat Pembelajaran untuk Pengembangan Pendidikan Karakter. *Indonesian Journal of Computer Science and Engineering (IJCSE)*, 01, 32.
- Wicaksana, M. P., Rahardandi, P. G., & ... (2024). Analisis Penerapan Chatbot: Survei. *Innovative: Journal Of ...*, 4, 8349–8364. <http://j-innovative.org/index.php/Innovative/article/view/13789%0Ahttps://j-innovative.org/index.php/Innovative/article/download/13789/9385>
- Yuniati, Y., & Gurning, F. A. (2024). Pengembangan Chatbot Batik Menggunakan Metode Long Short-Term Memory. 4(2), 753–759.