

# Implementasi Algoritma *Long Short-Term Memory* pada Chatbot Layanan Informasi dan Edukasi Demam Berdarah

## *Implementation of the Long Short-Term Memory Algorithm in a Chatbot for Dengue Fever Information and Education Services*

<sup>1</sup>Heru Budianto\*, <sup>2</sup>Fahmi Yusuf, <sup>3</sup>Dede Irawan, <sup>4</sup>Muhamad Akhirul Sidik, <sup>5</sup>Aiena Nurhasanah, <sup>6</sup>Sabrina Maulidya, <sup>7</sup>Silmi Nur Afifah

<sup>1,2,3,4,5,6,7</sup>Program Studi Sistem Informasi, Fakultas Ilmu Komputer, Universitas Kuningan  
<sup>1,2,3,4,5,6,7</sup>Jl. Pramuka No. 67 Purwawinangun, Kec. Kuningan, Kabupaten Kuningan, Jawa Barat  
45512, Indonesia

\*e-mail: [heru.budianto@uniku.ac.id](mailto:heru.budianto@uniku.ac.id)

(*received*: 4 February 2025, *revised*: 25 February 2025, *accepted*: 26 February 2026)

### Abstrak

Demam berdarah merupakan salah satu penyakit menular yang banyak terjadi di Indonesia. Informasi pencegahan dan penanganan demam berdarah masih bersifat parsial, terbatas jangkauan dan waktu. chatbot dikembangkan dengan menggabungkan Long Short-Term Memory untuk menjawab berbagai pertanyaan mengenai demam berdarah, termasuk gejala, pencegahan, dan cara penanganannya. Dataset yang digunakan mencakup pertanyaan dan jawaban tentang demam berdarah yang berasal dari data primer dan sekunder. Dataset diproses dan dianalisis melalui serangkaian tahapan preprocessing data untuk kemudian dilakukan pembuatan model, training dan evaluasi model. Hasil pengujian menunjukkan bahwa model yang dibuat memiliki nilai akurasi sebesar 100% pada validasi dengan fungsi loss sebesar 0,0221. Hasil tersebut menunjukkan chatbot berbasis Long Short-Term Memory dapat memberikan jawaban yang akurat dan relevan, serta efektif dalam memberikan edukasi kepada masyarakat secara efisien dan interaktif. Implementasi ini diharapkan dapat menawarkan solusi inovatif dalam meningkatkan kesadaran publik tentang pencegahan dan penanganan demam berdarah.

**Kata kunci:** demam berdarah, long short-term memory, chatbot, layanan

### Abstract

*Dengue fever is one of the most common infectious diseases in Indonesia. However, information regarding its prevention and treatment remains fragmented, with limited accessibility and availability. This study developed a chatbot integrating the Long Short-Term Memory (LSTM) algorithm to answer various questions about dengue fever, including its symptoms, prevention, and treatment. The dataset used consists of questions and answers related to dengue fever, sourced from both primary and secondary data. The data undergoes a series of preprocessing steps before being used for model development, training, and evaluation. The test results indicate that the developed model achieved an accuracy of 100% during validation with a loss function value of 0.0221. These findings demonstrate that the LSTM-based chatbot can provide accurate and relevant responses, making it an effective tool for educating the public in an interactive and efficient manner. This implementation is expected to offer an innovative solution for increasing public awareness of dengue fever prevention and management.*

**Keywords:** dengue fever, long short-term memory, chatbot, services

## 1 Pendahuluan

Demam berdarah *dengue* (DBD) tetap merupakan isu kesehatan yang signifikan di Indonesia [1]. Penyakit menular demam berdarah *Dengue* (DBD), dibawa oleh *vector* nyamuk dan merupakan

masalah kesehatan masyarakat global yang utama. Dimana cuaca ekstrem seperti curah hujan yang tinggi menimbulkan genangan air sebagai tempat nyamuk berkembang biak sehingga menimbulkan potensi kejadian luar biasa [2]. Saat ini, permukiman warga sangat rentan terhadap penyakit demam berdarah karena tingkat kesadaran masyarakat terhadap kebersihan lingkungan dan praktik hidup sehat masih rendah. Risiko masyarakat terhadap demam berdarah juga bergantung pada pengetahuan, sikap dan praktik masyarakat terhadap demam berdarah, serta pelaksanaan kegiatan rutin pengendalian *vector* berkelanjutan di masyarakat [3]. Pelaksanaan program pencegahan DBD masih terkendala cakupan yang terbatas, sosialisasi yang kurang menjangkau semua lapisan masyarakat, serta keterbatasan penyuluh dalam hal jarak dan waktu. Akibatnya, komunikasi antara masyarakat dan penyuluh menjadi sulit, menyebabkan ketidaktahuan yang berujung pada keterlambatan penanganan dan bahkan kematian pasien.

Teknologi dapat dioptimalkan sebagai media yang memberikan informasi akurat guna membantu mengurangi dampak dan penyebaran DBD. Salah satu teknologi yang mampu diaplikasikan untuk memberikan pemahaman mengenai pengetahuan praktis mengenai kesehatan dapat dilakukan dengan menggunakan chatbot [4]. Dimana dengan menggunakan chatbot masyarakat dalam mendapatkan informasi yang dibutuhkan dengan mudah dan cepat. Chatbot merupakan aplikasi atau layanan yang berinteraksi dengan pengguna melalui percakapan teks [5]. Chatbot bekerja untuk menggantikan peranan manusia dalam melayani pembicaraan melalui aplikasi pesan Chatbot telah menjadi sangat terkenal dan umum digunakan di berbagai bidang seperti layanan untuk menangani pelanggan [6] dan melakukan pelayanan terapi [7]. Chatbot, yang merupakan salah satu implementasi NLP, memiliki kemampuan untuk mengerti dan merespon pertanyaan pengguna [8] dengan cara yang menyerupai interaksi antar manusia [9] dan bersifat interaktif dengan para penggunanya [10]. Kepopuleran chatbot ini disebabkan oleh kemajuan dalam pemrosesan bahasa alami (NLP) dan pembuat chatbot. NLP memungkinkan chatbot untuk memahami bahasa manusia dengan lebih baik, sedangkan pembuat chatbot memungkinkan pengguna untuk membuat chatbot dengan cepat dan mudah.

Salah satu algoritma model pendekatan pembelajaran mendalam adalah LSTM (*Long Short-Term Memory*). *Long Short-Term Memory* (LSTM) merupakan pengembangan dari algoritma *Recurrent Neural Network* (RNN) yang dilengkapi dengan sel memori untuk menyimpan informasi dalam periode waktu yang lebih panjang. Metode ini telah terbukti efektif dalam mengelola urutan data yang panjang dan kompleks, menjadikannya sangat cocok untuk digunakan dalam pembuatan chatbot yang memerlukan pemahaman konteks percakapan yang mendalam [11]. LSTM telah menjadi metode terbaik yang banyak digunakan dalam penelitian mengenai pengembangan chatbot [12].

Penelitian ini bertujuan untuk mengimplementasikan algoritma *Long Short-Term Memory* pada chatbot dalam upaya penyediaan layanan informasi dan edukasi mengenai penyakit demam berdarah. Hasil dari penelitian ini nantinya dapat dipergunakan masyarakat untuk mengetahui informasi tentang demam berdarah sehingga dapat melakukan tindakan pencegahan dan penanganan demam berdarah yang tepat.

## 2 Tinjauan Literatur

Chatbot memiliki peran yang berguna dalam berbagai bidang, seperti pendidikan, pencarian informasi, dunia bisnis, dan *e-commerce* [13], namun penerapan chatbot masih dirasakan kurang. Penelitian mengenai penggunaan algoritma *Long Short Term Memory* (LSTM) pada chatbot banyak dilakukan oleh berbagai peneliti dalam berbagai bidang diantaranya untuk informasi akademik, layanan informasi Pendaftaran Sekolah Kejuruan, *support assistant* dan bidang lainnya.

Penelitian yang dilakukan oleh Puput dkk [12] menggunakan algoritma *Long Short-Term Memory* (LSTM) pada chatbot untuk menyediakan informasi akademik di Program Studi Teknik Informatika, Universitas Negeri Lampung dalam rangka peningkatan efektivitas layanan informasi akademik. Proses pengembangan meliputi pengumpulan data, persiapan data, pembangunan dan pelatihan model, serta pengujian model dengan menghasilkan nilai akurasi sebesar 99%.

Penelitian serupa dilakukan oleh Yudo dkk [13] memilih metode LSTM yang merupakan bagian dari *Recurrent Neural Network* (RNN) dikarenakan kemampuannya dalam menangani urutan teks dan konteks percakapan secara lebih baik dibandingkan metode tradisional. Hasil pengujian menunjukkan

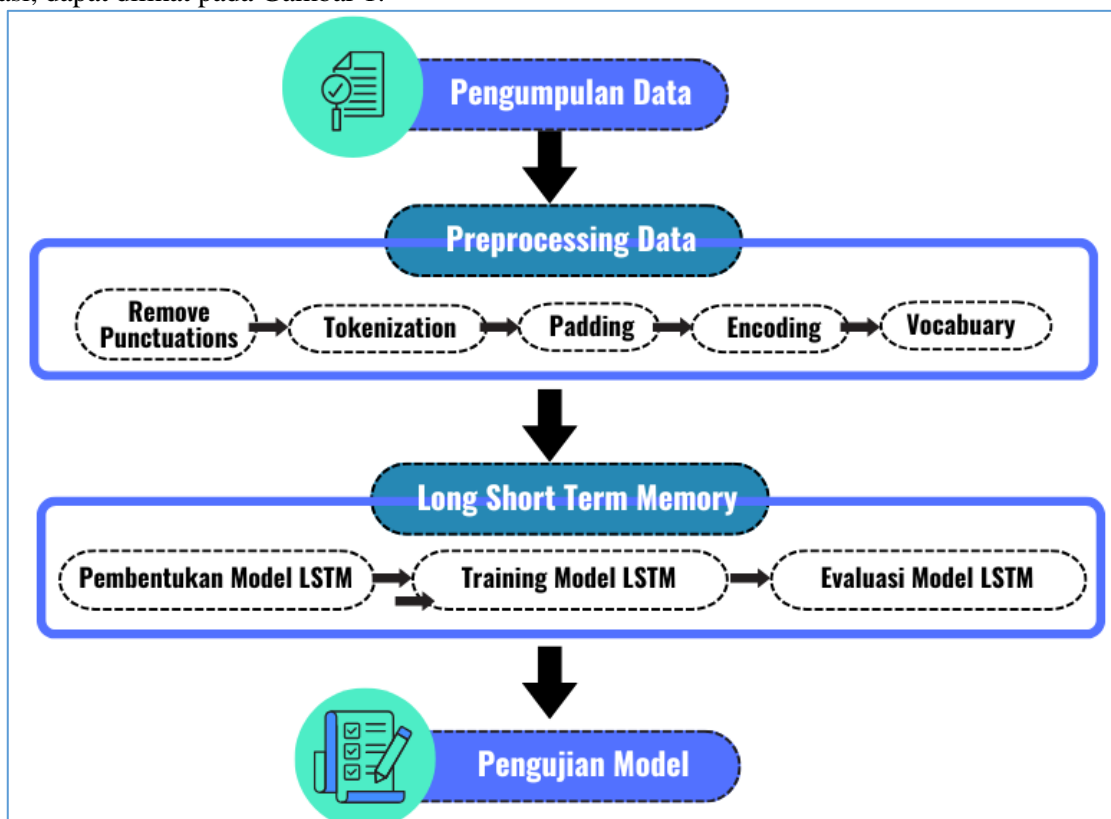
bahwa chatbot dapat memberikan solusi yang akurat terkait pertanyaan pendaftaran pelatihan dengan **Precision** sebesar 88,4%, **Akurasi** sebesar 87,6%, dan **Recall** sebesar 87,3%.

Penggunaan LSTM pada chatbot dinilai lebih baik oleh peneliti Ilham dan Egia [14] dimana hasil penelitian menunjukkan bahwa chatbot berbasis LSTM dapat memberikan respons yang lebih relevan, meningkatkan efisiensi layanan pelanggan, serta mengurangi beban kerja tenaga manusia. Pengujian menunjukkan pelatihan model menunjukkan **akurasi sebesar 99%**, **akurasi validasi 96%**, **loss sebesar 0,1%**, dan **loss validasi 0,2%** setelah 200 *epoch*, yang menegaskan efektivitas algoritma LSTM dalam pengembangan chatbot di perusahaan in

Secara keseluruhan, implementasi algoritma LSTM pada chatbot telah terbukti efektif dalam meningkatkan kualitas interaksi antara pengguna dan sistem, dengan menghasilkan respons yang lebih akurat dan kontekstual. Namun, chatbot yang dibangun sering kali belum diuji dalam skenario percakapan yang lebih kompleks atau dengan pengguna di luar domain yang telah ditentukan. Penelitian ini bertujuan untuk menunjukkan bagaimana algoritma *Long Short-Term memory* bekerja dalam chatbot.

### 3 Metode Penelitian

Metode penelitian adalah cara ilmiah yang sistematis dan cermat untuk memperoleh data guna mendeskripsikan, membuktikan, mengembangkan, serta menemukan pengetahuan atau teori dalam upaya memahami dan menyelesaikan masalah kehidupan manusia [15]. Adapun tahapan atau langkah-langkah penelitian yang dilakukan diawali dengan pengumpulan data hingga tahapan evaluasi, dapat dilihat pada Gambar 1.



Gambar 1. Tahapan penelitian

Terdapat 3 tahapan dalam perancangan chatbot seperti pada gambar 1, yaitu:

#### 3.1 Pengumpulan data

Pengumpulan data didapatkan dari data primer dan data sekunder, dimana data primer berasal dari dokumen terkait dan wawancara dengan masyarakat secara random. Data sekunder didapatkan dari brosur informasi tentang demam berdarah yang dikeluarkan oleh dinas serta pengumpulan data dari website terkait.

#### 3.2 Preprocessing data

*Pre-processing* data adalah serangkaian tahapan yang bertujuan untuk mempersiapkan data mentah agar siap digunakan dalam analisis *machine learning*. Proses ini bertujuan untuk meningkatkan kualitas data, menghilangkan ketidaktepatan, serta menyesuaikan format data agar lebih optimal untuk tahap pemrosesan selanjutnya [16].

1. *Remove Punctuation* meliputi penghapusan tanda baca dalam teks guna menyederhanakan analisis lebih lanjut. Tujuan dari penghapusan tanda baca adalah untuk mengurangi kompleksitas bacaan, standarisasi teks, serta mempermudah proses tokenisasi.
2. *Tokenization* merupakan tahap pemecahan teks menjadi unit-unit kecil. Dalam bidang NLP, token dapat berupa kata, frasa, simbol, atau elemen lain yang memiliki relevansi dalam analisis teks.
3. *Padding* adalah proses penyesuaian panjang rangkaian teks dalam suatu kesatuan naratif agar seragam untuk pemrosesan lebih lanjut.
4. *Encoding* merupakan proses mengubah data teks pada kolom data tugas menjadi data numerik dengan bahasa biner komputer yaitu 0 dan 1, bertujuan untuk mempermudah komputasi data.
5. *Vocabulary* bertujuan untuk menyusun daftar lengkap kata-kata unik yang terdapat dalam dataset. Daftar ini berfungsi sebagai referensi utama dalam proses encoding dan tokenisasi. Dengan adanya *vocabulary*, setiap kata dalam teks dapat memiliki representasi numerik yang tetap dan konsisten. [9]

### 3.3 Long Short-Term Memory

#### 1. Pembentukan model LSTM

Proses pemodelan ini mencakup pembangunan struktur jaringan LSTM yang terdiri dari berbagai *layer*, seperti *input layer*, *embedding layer*, *LSTM layer*, *dense layer*, dan *output layer*. Tahap pembentukan dan pelatihan model LSTM bertujuan untuk menciptakan sebuah model yang dapat memahami konteks percakapan dalam bahasa alami, terutama terkait dengan informasi tentang edukasi demam berdarah.

#### 2. Training model LSTM

Setelah membangun struktur jaringan LSTM yang terdiri dari *input layer*, *embedding layer*, *LSTM layer*, *dense layer*, dan *output layer*, langkah berikutnya adalah melatih model menggunakan data yang telah dipersiapkan. Proses pelatihan ini melibatkan beberapa iterasi melalui data latih (*epochs*) hingga model mencapai tingkat akurasi yang optimal.

#### 3. Evaluasi Model LSTM

Setelah proses pembentukan dan pelatihan model selesai, langkah berikutnya adalah melakukan analisis terhadap kinerja model dengan menghasilkan visualisasi grafik akurasi dan kerugian (*loss*) sebagai bahan evaluasi terhadap performa model LSTM yang telah diterapkan. Setelah proses pembentukan dan pelatihan model selesai, langkah berikutnya adalah melakukan analisis terhadap kinerja model dengan menghasilkan visualisasi grafik akurasi dan kerugian (*loss*) sebagai bahan evaluasi terhadap performa model LSTM yang telah diterapkan.

### 3.4 Pengujian Model Chatbot

Tujuan pengujian ini adalah untuk memastikan chatbot dapat memberikan respon yang tepat dan relevan terhadap pertanyaan atau pernyataan yang diajukan pengguna, sesuai dengan konteks yang telah dipelajari selama pelatihan

## 4 Hasil dan Pembahasan

Penelitian ini menggunakan algoritma Long Short-Term Memory (LSTM) pada chatbot untuk layanan informasi mengenai demam berdarah. Adapun hasil penelitian adalah sebagai berikut:

### 4.1 Pengumpulan data dan Pembuatan Dataset

Langkah pertama dalam merancang chatbot adalah mengumpulkan dan menyusun data dalam format JSON agar mudah diolah. Gambar 2 merupakan kode program dan hasil dari proses pengumpulan dan penyusunan data dalam format JSON. *Dataset* disimpan dalam file Excel sebagai referensi untuk proses berikutnya, dengan setiap entri berisi topik,

pertanyaan, dan jawaban guna mendukung *pre-processing* yang lebih efisien dan sistematis. File dataset dalam format Excel dikonversi ke JSON dengan mengganti nama kolom: "topik" menjadi "tag", "pertanyaan" menjadi "pattern", dan "jawaban" menjadi "responses". Konversi ini dilakukan agar *dataset* sesuai dengan struktur yang dibutuhkan oleh model chatbot serta mempermudah pemrosesan lebih lanjut. File hasil konversi diberi nama data.json yang kemudian dikonversi menjadi *Data Frame* yang akan dipergunakan pada tahapan selanjutnya yaitu preprocessing data.

```
# Konversi data json ke dalam dataframe
data = pd.DataFrame({"patterns":inputs, "tags":tags})
# Cetak data keseluruhan
data
```

|     | patterns  | tags                      |
|-----|---|---------------------------|
| 0   | Apa itu demam berdarah?                           | pengertian_demam_berdarah |
| 1   | Demam berdarah itu apa?                           | pengertian_demam_berdarah |
| 2   | Apa definisi penyakit demam berdarah?             | pengertian_demam_berdarah |
| 3   | Penyakit demam berdarah itu apa?                  | pengertian_demam_berdarah |
| 4   | Penyebab demam berdarah?                          | penyebab_demam_berdarah   |
| ... | ...   | ...                       |
| 116 | Bagaimana fogging membantu dalam pencegahan pe... | solusi_fogging            |
| 117 | Apakah fogging menjadi solusi yang efektif unt... | solusi_fogging            |
| 118 | Apakah fogging dapat membunuh nyamuk penyebab ... | solusi_fogging            |
| 119 | Sejauh mana fogging efektif dalam mencegah wab... | solusi_fogging            |
| 120 | Apakah fogging cukup untuk mengendalikan penye... | solusi_fogging            |

121 rows x 2 columns

Gambar 2. Hasil data frame

#### 4.2 Preprocessing Data

Tahap ini mencakup beberapa langkah penting untuk memastikan data siap digunakan dalam pembentukan model LSTM. *Preprocessing data* dilakukan di Google Colab untuk mempermudah analisis dan pengembangan. Berikut adalah langkah-langkah yang dilakukan:

##### 1. *Remove Punctuations* (penghapusan tanda baca)

Penghapusan dilakukan pada kolom 'patterns' dan 'tags' dalam *DataFrame* 'data'. Pada tahap ini, metode 'apply' digunakan untuk menerapkan fungsi lambda pada setiap elemen di kedua kolom tersebut. Fungsi lambda ini berfungsi untuk mengonversi teks menjadi huruf kecil serta menghapus tanda baca dengan memanfaatkan 'string.punctuation'. Gambar 3 merupakan kode program untuk proses *remove punctuations*.

```
# Remove Punctuations kolom 'patterns'
data['patterns'] = data['patterns'].apply(lambda wrd: ''.join([ltrs.lower() for ltrs in wrd if ltrs not in string.punctuation]))
# Remove Punctuations kolom 'tags'
data['tags'] = data['tags'].apply(lambda wrd: ''.join([ltrs.lower() for ltrs in wrd if ltrs not in string.punctuation]))
# cetak data
data
```

Gambar 3. Proses remove punctuations

##### 2. *Tokenization* (Tokenisasi)

Langkah awal dalam proses ini adalah menginisialisasi objek *Tokenizer* dari library Keras seperti yang terlihat pada Gambar 4. Proses ini berperan dalam mengkonversi teks menjadi kumpulan token. Parameter num\_words=3000 digunakan untuk membatasi

<http://sistemasi.ftik.unisi.ac.id>

pemrosesan hanya pada 3.000 kata yang paling sering muncul dalam data. Selanjutnya, *tokenizer* dilatih menggunakan metode *fit\_on\_texts* pada teks yang terdapat dalam kolom *patterns* dari variabel data. Setelah proses pelatihan selesai, metode *texts\_to\_sequences* digunakan untuk mengubah teks dalam kolom *patterns* menjadi urutan angka yang sesuai dengan indeks kata-kata dalam *tokenizer*.

```
# Tokenisasi Data
tokenizer = Tokenizer(num_words=3000)
tokenizer.fit_on_texts(data['patterns'])
train = tokenizer.texts_to_sequences(data['patterns'])
train
```

**Gambar 4. Proses tokenization**

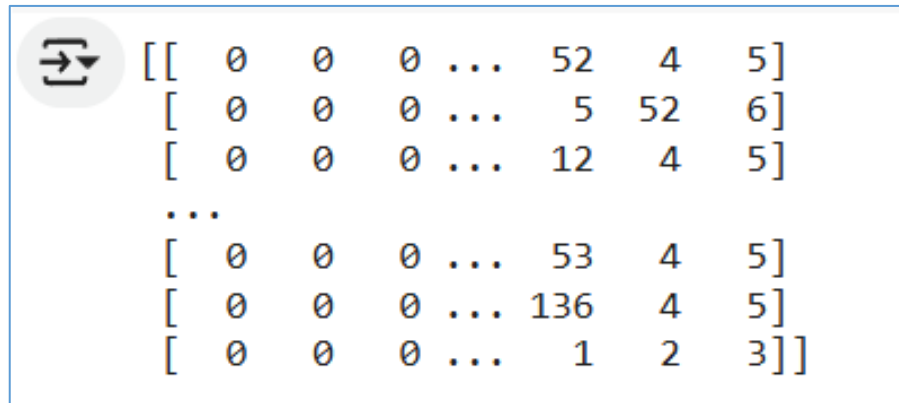
Hasil akhir dari proses tokenisasi ini disimpan dalam variabel *train*, yang berisi representasi teks dalam bentuk urutan angka. Gambar 5 memperlihatkan hasil akhir dari proses tokenisasi dimana hasilnya disimpan dalam bentuk angka.

```
[[6, 52, 4, 5],
 [4, 5, 52, 6],
 [6, 138, 12, 4, 5],
 [12, 4, 5, 52, 6],
 [53, 4, 5],
 [6, 7, 67, 12, 4, 5],
 [53, 54, 24, 12, 4, 5],
 [6, 7, 67, 54, 25, 12, 4, 5],
 [6, 44, 1, 2, 3],
 [11, 44, 1, 2, 3],
 [68, 6, 44, 1, 2, 3],
 [11, 139, 39, 1, 2, 3],
 [10, 140, 1, 2, 3, 26, 40],
 [10, 27, 1, 2, 3, 87, 45],
 [10, 27, 1, 2, 3, 16, 26, 40],
 [6, 17, 28, 88, 1, 2, 3],
 [28, 68, 6, 7, 89, 1, 2, 3, 9, 26, 40],
 [10, 28, 27, 17, 1, 2, 3, 26, 40],
 [29, 1, 2, 3, 69, 90, 18],
 [41, 91, 13, 1, 2, 3, 87, 18],
 [29, 70, 69, 141, 24, 30, 1, 2, 3],
 [8, 1, 2, 3, 18, 10, 142, 19, 143, 144],
 [91, 13, 1, 2, 3, 69, 90, 55, 71],
 [11, 145, 146, 1, 2, 3, 92, 18, 42],
 [8, 1, 2, 3, 18, 56, 147, 41, 70, 148],
 [8, 1, 2, 3, 93, 18, 42, 19, 149, 57],
 [72, 1, 2, 3, 7, 94, 18, 42, 19, 57],
```

**Gambar 5. Hasil tokenization**

### 3. *Padding*

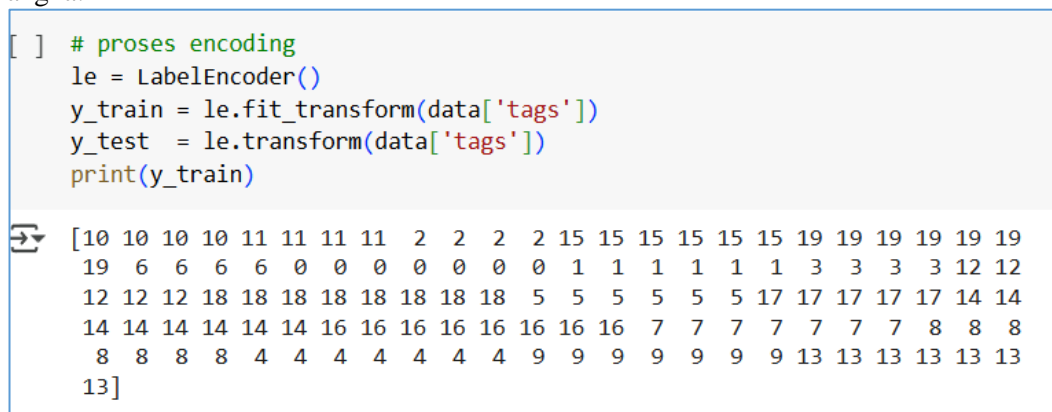
*Train* yang berupa urutan teks yang telah di tokenisasi, diubah menjadi *x\_train* dengan panjang urutan yang seragam melalui *padding*. Setelah *padding* selesai, hasilnya ditampilkan menggunakan *print(x\_train)*, yang menunjukkan semua urutan teks yang telah disesuaikan panjangnya. Hasil pada proses *padding* dapat dilihat pada Gambar 6.



Gambar 6. Hasil *padding*

#### 4. *Encoding*

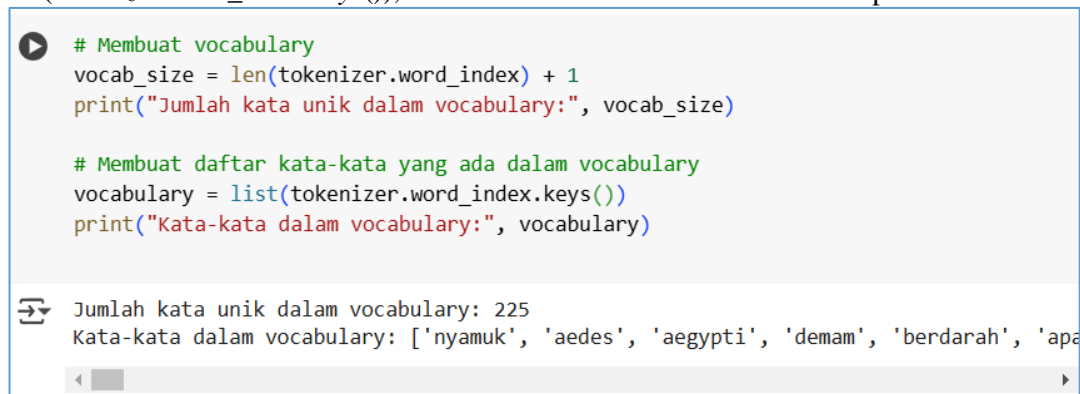
Gambar 7 merupakan proses *encoding* dan hasil *encoding* dimana proses ini diawali dengan menginisialisasi *LabelEncoder* menggunakan `le = LabelEncoder()`. Selanjutnya, label kategori dalam set pelatihan dikonversi ke bentuk numerik melalui `y_train = le.fit_transform(data['tags'])`, sedangkan data label pada set pengujian dikonversi menggunakan `y_test = le.transform(data['tags'])`. Hasil dari konversi ini kemudian ditampilkan dengan `print(y_train)`, yang menunjukkan label kategori dalam bentuk angka.



Gambar 7. Proses dan hasil *encoding*

#### 5. *Vocabulary*

Informasi mengenai jumlah kata unik ini penting untuk memahami ukuran serta kompleksitas *vocabulary* dalam proses pembuatan model. Dari hasil perhitungan, terdapat 225 kata unik dalam *vocabulary* seperti terlihat pada Gambar 8. Selanjutnya, daftar kata dalam *vocabulary* dibuat menggunakan `vocabulary = list(tokenizer.word_index.keys())`, kemudian kata-kata unik tersebut ditampilkan.

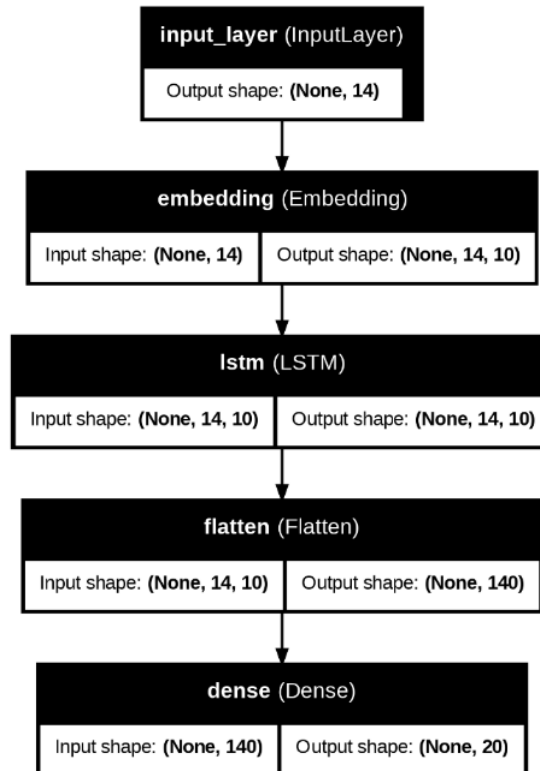


Gambar 8 Proses dan hasil *vocabulary*

### 4.3 Pemodelan Menggunakan LSTM

### 1. Pembentukan Model LSTM

Layer LSTM (*Long Short-Term Memory*) diterapkan dengan 10 unit LSTM yang mampu menyimpan informasi dalam jangka panjang, dan pengaturan *return\_sequences=True* memungkinkan urutan data diteruskan ke *layer* berikutnya. Untuk mencegah *overfitting*, diterapkan *dropout recurrent* sebesar 0.2 pada koneksi dalam LSTM.



**Gambar 9. Struktur model**

Berdasarkan Gambar 9 yang merupakan struktur model LSTM dapat disimpulkan bahwa pada *input layer*, jumlah sampel belum ditentukan, sementara panjang data adalah 14. Pada *embedding layer*, setiap kata diubah menjadi vektor *embedding* dengan panjang 14. Di LSTM layer, model menghasilkan keluaran dengan panjang urutan 14 dan dilengkapi dengan 10 fitur pada setiap urutannya. *Flatten layer* menggambarkan bahwa hasil model memiliki vektor dengan panjang 140. Terakhir, pada *dense layer*, nilai keluaran yang dihasilkan memiliki panjang 140, sesuai dengan tag yang telah dibuat

### 2. Training Model LSTM

Proses pelatihan model menggunakan metode *fit*, di mana model dilatih dengan data latih (*x\_train* dan *y\_train*) selama 300 *epoch* atau iterasi. Data validasi (*x\_test* dan *y\_test*) digunakan untuk memantau kinerja model pada data yang tidak terlibat dalam proses pelatihan.



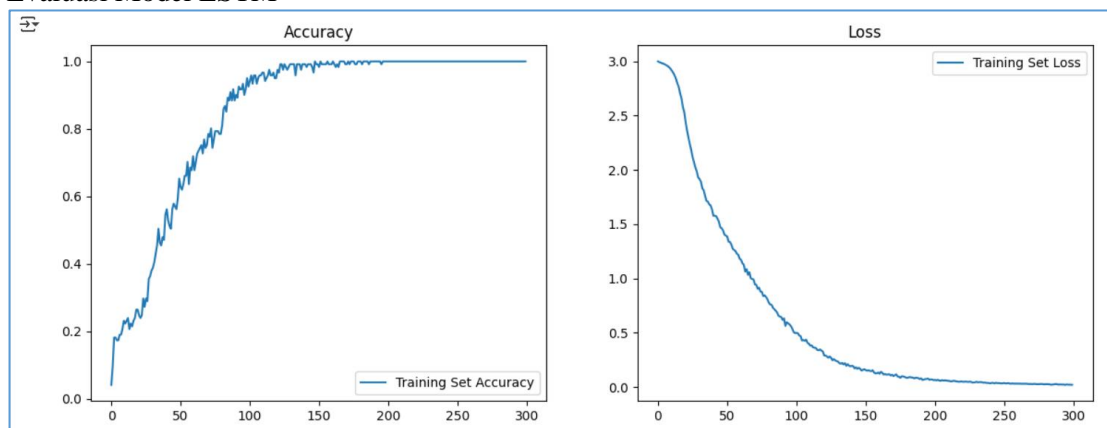
```
# Training the model (Melatih model data sampai 300 kali)
train = model.fit(x_train, y_train, epochs=300)

Epoch 272/300
4/4 ————— 0s 20ms/step - accuracy: 1.0000 - loss: 0.0276
Epoch 273/300
4/4 ————— 0s 21ms/step - accuracy: 1.0000 - loss: 0.0336
Epoch 274/300
4/4 ————— 0s 19ms/step - accuracy: 1.0000 - loss: 0.0317
Epoch 275/300
4/4 ————— 0s 20ms/step - accuracy: 1.0000 - loss: 0.0283
Epoch 276/300
4/4 ————— 0s 19ms/step - accuracy: 1.0000 - loss: 0.0323
Epoch 277/300
```

Gambar 10. Pelatihan model

Gambar 10 yang merupakan hasil pelatihan ini memperlihatkan bahwa pada *epoch* ke-300, model berhasil mencapai akurasi 100% (*accuracy*: 1.0000) pada data validasi. Fungsi loss untuk data latih tercatat sebesar 0.0221, yang mengindikasikan bahwa model dapat mengklasifikasikan data latih dengan sangat akurat.

### 3. Evaluasi Model LSTM



Gambar 11. Grafik dari pelatihan model

Pada Gambar 11 yang merupakan grafik hasil pelatihan model menunjukkan bahwa akurasi terus meningkat seiring bertambahnya jumlah *epoch*, sementara nilai kerugian (*loss*) semakin menurun. Pola ini mengindikasikan bahwa model yang dilatih berhasil mencapai kinerja yang baik, dengan kemampuan prediksi yang semakin akurat seiring proses pelatihan.

### 4.4 Pengujian model chatbot

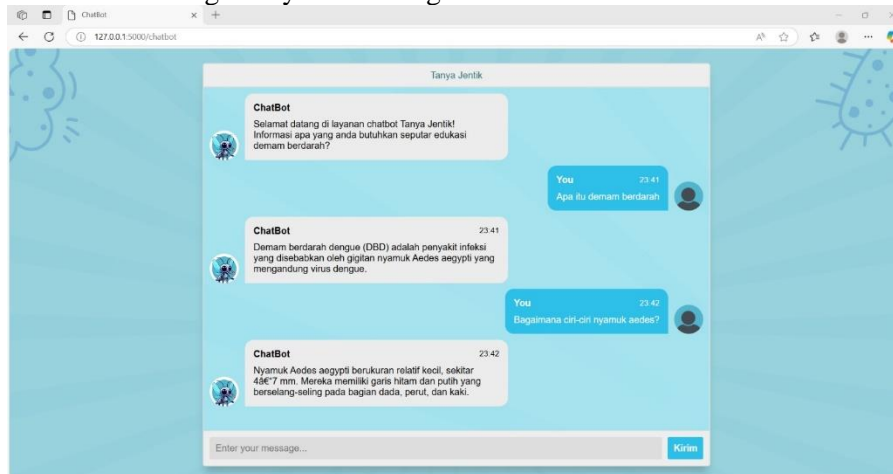
Tahap pertama dalam pengujian melibatkan integrasi pengetahuan yang telah dipelajari oleh model ke dalam sistem chatbot. Pengetahuan ini mencakup informasi tentang Edukasi Demam Berdarah, termasuk pengertian, jenis nyamuk, dan lainnya. Proses integrasi dilakukan dengan memuat data yang telah diproses dan diperoleh dari tahap pemodelan dan pelatihan. Setelah pengetahuan dimasukkan, chatbot diuji dengan berbagai input, termasuk pertanyaan terkait edukasi demam berdarah. Respon yang dihasilkan kemudian dianalisis untuk menilai kesesuaiannya dengan informasi yang telah dipelajari serta relevansinya dalam konteks percakapan.

```
You: apa itu demam berdarah
1/1 ————— 0s 83ms/step
Bot: Demam berdarah dengue (DBD) adalah penyakit infeksi yang disebabkan oleh gigitan nyamuk Aedes aegypti yang mengandung virus dengue.
You: ciri-ciri nyamuk aedes
1/1 ————— 0s 81ms/step
Bot: Nyamuk Aedes aegypti berukuran relatif kecil, sekitar 4-7 mm. Mereka memiliki garis hitam dan putih yang berselang-seling pada bagian dada, perut, dan kaki.
```

Gambar 12. Pengujian model chatbot

Berdasarkan pengujian seperti terlihat pada Gambar 12, chatbot mampu memberikan jawaban yang relevan dan informatif terhadap berbagai pertanyaan terkait Edukasi Demam Berdarah, serta dapat menghentikan percakapan saat diminta.

Tahapan terakhir setelah pengujian adalah integrasi kedalam platform yang diinginkan. Adapun chatbot yang dibangun diintegrasikan ke dalam sebuah website layanan informasi dan edukasi bagi masyarakat mengenai demam berdarah.



**Gambar 13. Tampilan chatbot**

Implementasi chatbot dapat dilihat pada gambar 10. Berdasarkan gambar 10 chatbot yang dikembangkan dengan menerapkan algoritma *Long Short-Term Memory* dapat dipergunakan sebagai layanan informasi dan edukasi mengenai demam berdarah.

## 5 Kesimpulan

Hasil pengujian model mendapatkan nilai dengan akurasi 100% (*accuracy*: 1.0000) pada data validasi serta fungsi *loss* untuk data latih tercatat sebesar 0.0221. Hal ini menunjukkan model yang dibuat dapat mengklasifikasi dengan baik dimana digunakan *epoch* sebesar 300. Model tersebut diimplementasikan setelah proses pengujian dan memperlihatkan pertanyaan yang diajukan dapat tepat dijawab oleh chatbot yang telah diintegrasikan pada web layanan informasi dan edukasi tentang demam berdarah. Untuk penelitian selanjutnya disarankan untuk melakukan *redesign* tampilan antarmuka dengan metode yang sesuai sehingga menghasilkan tampilan yang lebih baik.

## Referensi

- [1] E. Asih, V. T. Putri, N. Lusida, A. Mallongi, N. Latifah, F. Fajrini, and H. SG., "Analisis Variasi Iklim dengan Kejadian Demam Berdarah Dengue (DBD) di Kota Tangerang Selatan," *J. Kedokt. dan Kesehat.*, vol. 19, no. 1, p. 33, 2023, doi: 10.24853/jkk.19.1.33-41.
- [2] D. Nurmumpuni, B. Kurniawan, and Suharmanto, "Efektivitas Program Pemberantasan Penyakit Demam Berdarah Dengue," *J. Penelit. Perawat Prof.*, vol. 6, no. 3, pp. 1377–1386, 2024.
- [3] C. Rahmawati, B. Leny Nopitasari, A. Kusuma Wardani, B. Nurbaety, B. Lenysia Puspita Anjani, M. Permata Hati, N. Furqani, A. Rahman Wahid, I. Hendriyani, A. Pradiningsih, Y. Fitriana, D. Haryadi Ittiko, "Edukasi Mencegah Penyakit Demam Berdarah Dengue (DBD) pada Masyarakat Lingkungan Dasan Sari Ampenan," *J. Pengabd. Masy. Berkemajuan*, vol. 6, no. 4, pp. 3038–3042, 2022.
- [4] L. Indahsari, K. Kusnadi, and T. E. Putri, "Rancang Bangun LINE Chatbot Informasi dan Edukasi Kesehatan Mental menggunakan *Algoritma Jaro Winkler*," *J. Eksplora Inform.*, vol. 10, no. 2, pp. 68–79, 2021, doi: 10.30864/eksplora.v10i2.428.
- [5] M. Mustaqim, A. Gunawan, Y. B. Pratama, and I. Zaliman, "Pengembangan *Chatbot* Layanan Publik menggunakan *Machine Learning* dan *Natural Language Processing*," *J. Inf. Technol. Soc.*, vol. 1, no. 1, pp. 1–4, 2023.
- [6] L. Cui, "SuperAgent: A Customer Service Chatbot for E-commerce Websites," pp. 97–102, 2017.
- [7] J. Lee and W. Maeng, "Designing a Chatbot for Survivors of Sexual Violence: Exploratory Study for Hybrid Approach Combining Rule-based Chatbot and ML-based Chatbot," in *Asian CHI Symposium*, 2021, pp. 160–166, doi: 10.1145/3429360.3468203.
- [8] M. R. Suherlan, A. Asriyanik, and A. Pambudi, "UMMIBOT sebagai Media Layanan Informasi

- Penerimaan Mahasiswa Baru Universitas Muhammadiyah Sukabumi,” J. Inform. Terpadu, vol. 9, no. 2, pp. 82–91, 2023, doi: 10.54914/jit.v9i2.893.
- [9] Sutarni, E. Prasetyo, and L. E. Sudiati, “Evaluasi Kinerja *Chatbot* dengan Integrasi Algoritma RNN dan LSTM dalam Optimalisasi Respon Percakapan pada Sistem PMB,” SOSCIED, vol. 7, no. 2, 2024.
- [10] G. A. Pratama, N. C. Wiboeo, A. Pratama, “*Implementasi Chatbot* Konsultasi Karier berbasis *Deep Feed-Forward Neural Network* menggunakan,” J. Tek. Inform. dan Sist. Inf., vol. 11, no. 3, pp. 248–260, 2024.
- [11] S. P. Afrisia, F. M. Hana, and W. C. Wahyudin, “*Implementasi Metode Long Short Term Memory ( LSTM )* pada *Chatbot* Kesehatan Mental Mahasiswa,” Sainteks, vol. 21, no. 2, pp. 107–116, 2024, doi: 10.30595/sainteks.v21i2.23869.
- [12] P. B. Wintoro, H. Hermawan, M. A. Muda, and Y. Mulyani, “*Implementasi Long Short-Term Memory* pada *Chatbot* Informasi Akademik Teknik Informatika Unila,” Expert J. Manaj. Sist. Inf. dan Teknol., vol. 12, no. 1, p. 68, 2022, doi: 10.36448/expert.v12i1.2593.
- [13] Y. S. H. Langgeng, E. I. Setiawan, S. Imron, and J. Santoso, “*Long Short-Term Memory-based Chatbot for Vocational Registration Information Services*,” J. Appl. Data Sci., vol. 4, no. 4, pp. 414–430, 2023, doi: 10.47738/jads.v4i4.128.
- [14] I. D. Raharjo and E. R. Subhiyakto, “*Implementing Long Short Term Memory (LSTM) in Chatbots for Multi Usaha Raya*,” Adv. Sustain. SCI. Eng. Technol., vol. 6, no. 4, pp. 1–8, 2024, doi: 10.26877/asset.v6i4.934.
- [15] A. Adil, Metode Penelitian Kuantitatif dan Kualitatif: Teori dan Praktik, no. January. 2023.
- [16] F. Y. Fiddin, A. Komarudin, and M. Melina, “*Chatbot* Informasi Penerimaan Mahasiswa Baru menggunakan Metode *FastText* dan LSTM,” J. Appl. Comput. Sci. Technol., vol. 5, no. 1, pp. 33–39, 2024, doi: 10.52158/jacost.v5i1.648.