

Pengembangan Chatbot untuk Meningkatkan Pengetahuan dan Kesadaran Keamanan Siber Menggunakan Long Short-Term Memory

Hilya Anbiyani Fitri Muhyidin^{1*}, Liptia Venica²

¹Jurusan Pendidikan Sistem dan Teknologi Informasi, Universitas Pendidikan Indonesia

²Jurusan Mekatronika Kecerdasan Buatan, Universitas Pendidikan Indonesia

*Email: hilyaanbiyani@upi.edu

Abstrak

Cyber-crime semakin masif seiring dengan meningkatnya aktivitas online. Cyber-crime merupakan suatu tindak kejahatan yang memanfaatkan teknologi digital untuk merusak, merugikan, dan menghancurkan properti. Oleh karena itu, penting bagi pengguna internet memiliki pengetahuan keamanan siber serta dunia teknologi dan internet agar tidak menjadi korban pelaku tindak cyber-crime. Tujuan penelitian ini adalah membangun suatu sistem chatbot sebagai media informasi terpusat mengenai keamanan siber serta dunia teknologi dan internet bagi pengguna internet. Pembangunan chatbot ini diharapkan dapat mengurangi resiko dari kejahatan dunia maya dan membantu meningkatkan kewaspadaan pengguna internet terhadap cyber-crime. Penelitian ini menggunakan metode AI Project Cycle dalam pembangunan chatbot dan menerapkan algoritma model pendekatan deep learning yaitu Long Short-Term Memory (LSTM) dalam mengembangkan model untuk mendapatkan hasil akurasi yang tinggi. Hasil pelatihan model LSTM mendapatkan nilai akurasi sebesar 100% dan loss 3.09% dengan epochs sebesar 400. Dengan demikian, dapat disimpulkan bahwa algoritma LSTM sangat baik untuk melatih dan mengembangkan model chatbot.

Kata kunci: AI Project Cycle, Chatbot, Cybercrime, Flask, Long Short-Term Memory

Abstract

Cyber-crime is becoming more massive as online activities increase. Cybercrime is a criminal act that exploits digital technology to damage, harm, and destroy property. Therefore, it is crucial for internet users to have knowledge of cybersecurity and the world of technology and the internet in order to avoid falling victim to cybercrime. The aim of this study is to develop a chatbot system as a centralized information medium on cybersecurity, technology, and the internet for internet users. The development of this chatbot aims to reduce the risks of cybercrimes and help enhance internet users' awareness of cybercrime. This research employs the AI Project Cycle method in chatbot development and utilizes the Long Short-Term Memory (LSTM) deep learning model algorithm to develop a model that achieves high accuracy. The training results of the LSTM model achieved an accuracy score of 100% and a loss of 3.09% with 400 epochs. Consequently, it can be concluded that the LSTM algorithm is highly effective for training and developing a chatbot model.

Kata kunci: AI Project Cycle, Chatbot, Cybercrime, Flask, Long Short-Term Memory

PENDAHULUAN

Seiring dengan populernya teknologi informasi dan *internet*, pengguna *internet* semakin meningkat sehingga menyebabkan tingginya aktivitas *online*, ditambah lagi karena pergeseran kegiatan yang lebih banyak dilakukan secara *online* atau digital sebagai akibat dari pandemi COVID-19 (Nur Latifah et al., 2022). Hal tersebut menjadi peluang kejahatan oleh orang yang tidak bertanggung jawab untuk melakukan tindak *cyber-crime* (Rian Handoko & Tata Sutabri, 2023). *Cyber-crime* merupakan suatu tindak kejahatan yang

memanfaatkan teknologi digital atau alat komunikasi lainnya untuk menghancurkan, merugikan, dan merusak properti sehingga menimbulkan ketakutan dan kecemasan kepada orang lain (Al-Khater et al., 2020).

Menurut *Digital 2022 Global Overview Report*, dari total populasi dunia sebesar 7.19 miliar, pengguna ponsel seluler adalah 67.1%, pengguna *internet* sebanyak 4.95 miliar dengan persentase 62.5%, dan pengguna media sosial aktif sebesar 58%. Pada laporan tersebut, pengguna *internet* mengalami peningkatan sebesar 4% dari tahun 2021 (Kemp, 2022).

Tingkat resiko *cyber-crime* di Indonesia juga dapat diklasifikasikan cukup tinggi. Menurut laporan Badan Siber dan Sandi Negara Republik Indonesia, jumlah *cyber-crime* yang terjadi Januari hingga April tahun 2020 mencapai 88 juta serangan (Bagian Komunikasi Publik, 2020). Oleh karena itu, penting bagi pengguna *internet* memiliki pengetahuan keamanan siber serta dunia teknologi dan *internet* agar tidak menjadi korban pelaku tindak *cyber-crime* (Habibi & Liviani, 2020). Dalam memenuhi kebutuhan informasi atau pengetahuan tersebut, suatu sistem diperlukan sebagai media informasi terpusat mengenai keamanan siber serta dunia teknologi dan *internet* bagi pengguna *internet*. *Question System Answering* (QAS) bertujuan untuk memenuhi pengguna mencari jawaban yang cepat, tepat, dan relevan atas pertanyaan tertentu dalam bahasa alami (Yunmar & Wisesa, 2020).

Penggunaan *deep learning* telah mengalami peningkatan yang signifikan dan menjanjikan dalam beberapa tahun terakhir sehingga menarik perhatian yang besar bagi para peneliti dan praktisi di bidang *Natural Language Processing* (NLP). Salah satu tujuan khusus *deep learning* adalah untuk tugas klasifikasi teks yang mana telah mendominasi banyak bidang *Artificial Intelligence* (AI) (Gravano, 2010). Implementasi *deep learning* di bidang NLP yaitu pada pengembangan model untuk *chatbot* atau *Assistant Conversational Agent* (Dhyani & Kumar, 2019).

Chatbot merupakan salah satu contoh interaksi manusia-komputer (*Human-Computer Interaction*) yang menggunakan sistem AI dan telah tersebar luas. *Chatbot* memuat sumber jawaban untuk pertanyaan pengguna pada suatu domain tertentu yang mana program komputer akan merespon pertanyaan tersebut seperti agen cerdas ketika berinteraksi melalui teks atau suara dan mampu memahami multi bahasa menggunakan NLP (Khanna et al., 2015; Lalwani et al., 2018). *Chatbot* juga dikenal sebagai *bot* pintar, agen responsif, asisten *virtual*, atau entitas percakapan buatan (Adamopoulou & Moussiades, 2020). Berdasarkan domain pengetahuan, *chatbot* diklasifikasikan menjadi 2 jenis, yaitu *open domain* dan *closed domain*. *Open domain* dapat merespon topik yang umum dengan tepat

sedangkan *closed domain* hanya dapat merespon domain pengetahuan atau topik tertentu dan memiliki peluang gagal dalam merespon pertanyaan topik lain (Nimavat & Champaneria, 2017). Paper ini akan membahas implementasi dari NLP yaitu QAS khususnya *closed domain chatbot* menggunakan pendekatan *deep learning*.

Salah satu algoritma model pendekatan pembelajaran mendalam adalah LSTM (*Long Short-Term Memory*). Dalam konteks klasifikasi yang berhubungan dengan waktu dan data teks, LSTM merupakan pilihan yang tepat karena memiliki unit memori atau sel memori pada jaringan saraf yang mampu menyimpan informasi yang diperoleh dalam jangka panjang sehingga LSTM dapat menyelesaikan masalah *vanishing gradient* yang terjadi saat memproses data *sequential* yang panjang (Sennhauser & Berwick, 2018; Trivusi, 2022; Witanto et al., 2022). Oleh karena itu, saat ini LSTM menjadi metode yang banyak diterapkan dalam penelitian yang berkaitan dengan pengembangan *chatbot* (Wintoro et al., 2022).

Penelitian ini bertujuan menerapkan algoritma LSTM pada *chatbot* dalam menyajikan jawaban dari pertanyaan yang diajukan oleh pengguna seputar *cyber security* dan *cyber-crime* sebagai media informasi dan pengetahuan mengenai *cyber security* serta memberikan tips dan trik untuk menghindari kejahatan *cyber-crime* bagi para pengguna *internet*. Algoritma LSTM diterapkan untuk mendapatkan hasil akurasi yang tinggi seperti dalam (Silvanie & Subekti, 2022). Kajian ini menggunakan metode *AI Project Cycle* dalam pembangunan *chatbot*. Dengan demikian, pembangunan *chatbot* ini diharapkan dapat mengurangi resiko dari kejahatan dunia maya (*cyber-crime*) dan membantu meningkatkan kewaspadaan pengguna *internet* terhadap *cyber-crime* agar dapat beraktivitas dengan aman di dunia maya.

TINJAUAN PUSTAKA

Pengembangan *chatbot* menggunakan teknik *deep learning* dengan algoritma LSTM telah dilakukan oleh Astried Silvanie dan Rino Subekti (Silvanie & Subekti, 2022) yang mengembangkan *chatbot Frequently Asked Question* (FAQ) untuk memberikan respons terhadap pertanyaan mahasiswa di Institut Bisnis dan Informatika Kosgoro 1957 mengenai

masalah akademik. Algoritma pelatihan model yang digunakan yaitu LSTM yang menghasilkan akurasi sebesar 99.20% ketika 90 *epochs* dan menggunakan optimasi *adam*.

Nia Agustina.P dan Muhamad Soleh (Purwitasari & Soleh, 2022) menggunakan algoritma *Artificial Neural Network* (ANN) dalam membuat aplikasi dengan fitur *chatbot* untuk membantu masyarakat dalam dalam mengakses informasi mengenai peraturan perundang-undangan lebih mudah. *Dataset* yang digunakan untuk melatih algoritma adalah data manual berupa *JavaScript Object Notation* (JSON) yang mencapai akurasi sebesar 94.28% dan tingkat kesalahan sebesar 5.71%. Model berhasil menjawab tepat 33 percakapan dari 35 percakapan.

Selain itu, algoritma *Convolutional Neural Network* (CNN), *sequence to sequence* (*Seq2seq*), dan *convolutional sequence to sequence* (*Conv-seq2seq*) juga telah diterapkan oleh Uli Rizki (Rizki, 2019) dalam membangun *chatbot* untuk mengatasi mengatasi *multi respon rangking* pada percakapan layanan *travel* berdasarkan riwayat obrolan. Model menghasilkan akurasi sebesar 86.7% dengan menggabungkan CNN dan *sequence to sequence* (*Conv-seq2seq*), 68.20% dengan *seq2seq*, serta 84.10% dengan CNN.

Kemudian, Asep Muhidin, dkk. (Muhidin et al., 2023) menerapkan algoritma *Multinomial Naïve Bayes* dalam membuat *chatbot* layanan informasi dalam menjawab pertanyaan mahasiswa di Universitas Pelita Bangsa. *Dataset* yang digunakan untuk melatih algoritma adalah berupa pertanyaan dan jawaban yang terdiri dari 120 data dan disimpan dalam format JSON. Model pelatihan menghasilkan akurasi sebesar 98% dan akurasi tes sebesar 84%.

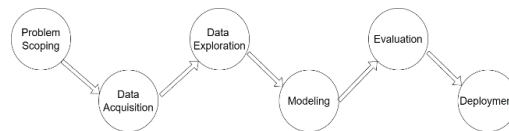
Lebih lanjut, Kristian dan Danny (Nugraha & Sebastian, 2021) menggunakan algoritma *K-Nearest Neighbor* (K-NN) dalam mengembangkan *chatbot* untuk layanan akademik di Fakultas Teknologi Informasi, Universitas Kristen Duta Wacana. Model K-NN menghasilkan akurasi sebesar 53.48% ketika nilai K=3.

Algoritma LSTM diterapkan pada pengembangan *chatbot* pada penelitian ini untuk mendapatkan hasil akurasi yang tinggi seperti dalam (Silvanie & Subekti, 2022). Namun, digunakan untuk melakukan klasifikasi jawaban dari pertanyaan yang diajukan dalam

konteks *cyber security* dan *cyber-crime*. Kajian ini menggunakan 400 *epochs* dalam pelatihan model dan menggunakan data berupa pertanyaan dan jawaban yang disimpan dalam format JSON seperti dalam (Purwitasari & Soleh, 2022; Rizki, 2019; Muhidin et al., 2023).

METODE PENELITIAN

Metode pengembangan *chatbot* yang digunakan pada penelitian melibatkan *Framework AI project cycle* terdiri dari enam tahapan, diantaranya *Problem Scoping*, *Data Acquisition*, *Data Exploration*, *Modelling*, *Evaluation*, dan *Deployment* (Siddhartha, 2021; Widodo et al., 2022). Pada Gambar 1 menunjukkan proses *AI project cycle*.



Gambar 1. *Framework AI Project Cycle*

3.1 Problem Scoping

Problem scoping pada penelitian ini membahas bagaimana menyelesaikan sebuah masalah yang diawali dengan memahami dan menganalisis masalah serta menentukan tujuan sehingga masalah dapat terselesaikan dengan baik. Metode 4Ws digunakan untuk memudahkan penyelesaian masalah pada tahap *problem scoping* dengan memetakan *who*, *what*, *where*, dan *why*. Maksudnya *who* adalah menerangkan subjek yang jelas dalam masalah, *what* yaitu masalah apa yang ditetapkan, *where* maksudnya masalah ditemukan saat apa, dan *why* merupakan alasan mengapa masalah tersebut perlu diselesaikan dan bagaimana solusinya. Berdasarkan pemetaan 4Ws, penulis melakukan analisis masalah dan menentukan solusi untuk menyelesaikan permasalahan.

3.2 Data Acquisition

Setelah menentukan lingkup masalah serta solusinya, tahapan selanjutnya adalah *data acquisition* yaitu melakukan riset dan pengumpulan data yang relevan. *Dataset* yang digunakan untuk melatih algoritma model *chatbot* merupakan *dataset* manual berupa *file JSON* (*JavaScript Object Notation*) yang dikumpulkan berdasarkan *literature review* mengenai *cyber-crime* dan keamanan dunia

maya. *Dataset* memiliki beberapa komponen penting seperti berikut:

1. *Intents*, memuat sejumlah data masukan (*input*) dan keluaran (*output*) yang dikumpulkan untuk melatih *chatbot*.
2. *Tags*, berfungsi untuk mengklasifikasikan data teks yang memiliki kesamaan dan menggunakan *output* yang serupa sebagai *target* pelatihan bagi jaringan saraf.
3. *Patterns*, merupakan komponen yang mengandung data pola *input* berdasarkan harapan dari pengguna.
4. *Responses*, mengandung data pola *output* yang akan dikirimkan oleh *chatbot* kepada pengguna.

3.3 Data Exploration

Setelah *dataset* terkumpul, tahap selanjutnya adalah *data exploration*, yaitu memahami karakteristik data dan melakukan *preprocessing* atau pembersihan data menggunakan pendekatan NLP dan *Natural Language Toolkit* (NLTK) sebelum dokumen teks diolah lebih lanjut dan masuk ke tahap *modeling* (Nugraha & Sebastian, 2021; Purwitasari & Soleh, 2022). Tujuan *preprocessing* adalah memastikan bahwa data yang digunakan dalam pelatihan model *chatbot* adalah bersih, konsisten, dan representatif sehingga dapat memberikan hasil akhir yang lebih baik (Nugraha & Sebastian, 2021). *Preprocessing* teks yang dilakukan pada *dataset chatbot* penelitian ini diantaranya:

1. *Case folding*, mengubah data *input* atau fitur menjadi huruf kecil dan menghapus karakter tanda baca. *Source code* untuk melakukan *case folding* dapat dilihat pada Gambar 2.

```
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 2. *Source Code Case Folding*

2. Lematisasi, mengubah kata ke dalam bentuk dasar menggunakan kamus kosakata dan analisis morfologi kata-kata untuk menghilangkan imbuhan (*affixes*) seperti awalan kata (*prefixes*), sisipan kata (*infixes*), akhiran kata (*suffixes*), serta awalan dan akhirankata

(*confixes*) (Silvanie & Subekti, 2022). *Source code* untuk melakukan lematisasi dapat dilihat pada Gambar 3.

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

Gambar 3. *Source Code Lematisasi*

3. Tokenisasi, memecah kalimat menjadi bagian-bagian yang disebut dengan “*token*” (Silvanie & Subekti, 2022). *Source code* untuk melakukan tokenisasi dapat dilihat pada Gambar 4.

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

Gambar 4. *Source Code Tokenisasi*

4. *Padding*, menggunakan *library pad_sequence* yang ada pada *Python* untuk menyamakan panjang dan membuat setiap urutan (*sequence*) agar memiliki ukuran yang seragam. *Source code* untuk melakukan *padding* dapat dilihat pada Gambar 5.

```
x_train = pad_sequences(train)
```

Gambar 5. *Source Code Add Padding*

5. Ekstraksi keluaran *encoding*, melakukan pengkodean yang dimana data kategorik seperti huruf atau data teks menjadi data numerik atau angka menyesuaikan dengan data label yang digunakan. Tujuan dari *encoding* ini adalah mempermudah saat proses komputasi data teks dan *modeling*. *Source code* untuk melakukan *encoding* dapat dilihat pada Gambar 6.

```
le = LabelEncoder()
y_train = le.fit_transform(data['tags'])
```

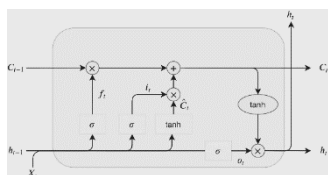
Gambar 6. *Source Code Encoding*

3.4 Modeling

Tahap *modeling* melibatkan pengembangan model yang meliputi proses

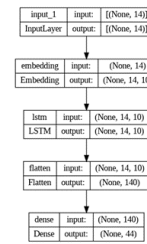
pemilihan algoritma dan pelatihan data menggunakan algoritma untuk *chatbot*. Algoritma yang akan melatih *chatbot* adalah algoritma *deep learning* yaitu LSTM karena dapat menghasilkan nilai akurasi yang tinggi dan memiliki kemampuan yang lebih baik, serta dapat merespon dalam waktu yang singkat (Prabowo et al., 2019; Silvanie & Subekti, 2022). Model tersebut akan dilatih untuk meningkatkan kinerja *chatbot* menggunakan data yang telah diproses dan dilakukan representasi data.

LSTM adalah perluasan dari *Recurrent Neural Network* (RNN) yang mana dapat menyelesaikan masalah terjadinya *vanishing gradient* atau *long dependency problem* pada RNN ketika memproses data *sequential* yang panjang (Witanto et al., 2022; Zhao et al., 2017). Algoritma LSTM melibatkan jaringan saraf dan beberapa *unit* memori terpisah yang disebut sel (Trivusi, 2022). Dalam arsitektur LSTM, sel memori terdiri dari empat komponen utama, yaitu *input gate*, koneksi berulang, *forget gate* dan *output gate*. Proses kerja pada LSTM terdapat operasi *dot product* atau *hadamard product* (*element-wise product*) dan penjumlahan serta fungsi *sigmoid* pada *forget gate* untuk melupakan (*filter*) kata-kata tertentu dan menyimpan atau mengingat informasi yang dianggap penting (Putra et al., 2021; Witanto et al., 2022). Pada Gambar 7 menunjukkan representasi sel LSTM.



Gambar 7. Blok Memori pada LSTM (Aggarwal, 2023)

Algoritma model LSTM untuk *chatbot* dilatih sebanyak 400 *epochs* dengan struktur *input layer*, satu *embedding layer*, satu LSTM *layer*, satu *flatten layer*, dan satu *output layer* dengan menggunakan fungsi aktivasi *softmax* untuk melakukan perhitungan probabilitas dan penentuan klasifikasi multi kelas dalam mengidentifikasi kelas dengan nilai probabilitas tertinggi sebagai *output* (Purwitasari & Soleh, 2022). Implementasi *chatbot* ini memiliki data label lebih dari dua kelas (44 kelas). Arsitektur model LSTM dapat dilihat pada Gambar 8.



Gambar 8. Arsitektur Model LSTM

3.5 Evaluation

Pada tahap *evaluation*, model yang telah dilatih kemudian dilakukan tes atau penilaian untuk memastikan hasil tes adalah akurat. Adapun matriks evaluasi yang digunakan pada model adalah “*accuracy*” dan *loss function* dengan *categorical_crossentropy* serta fungsi optimasi *adam*, karena model menggunakan klasifikasi multi kelas seperti penelitian (Silvanie & Subekti, 2022; Wintoro et al., 2022). Setelah mendapatkan model terbaik, selanjutnya dilakukan penyimpanan model sehingga dapat lanjut ke tahap *deployment*. *Source code* untuk melakukan evaluasi dapat dilihat pada Gambar 9.

```
# Compiling the model (kompilasi Model)
model.compile(loss="sparse_categorical_crossentropy", optimizer='adam',
metrics=['accuracy'])
```

Gambar 9. Source Code Evaluasi Model

3.6 Deployment

Model *chatbot* yang di-*deploy* dalam penelitian ini dibangun ke dalam sebuah aplikasi berbasis *web* menggunakan *framework* sebelum *chatbot* dapat digunakan oleh pengguna agar dapat diakses secara *online*. *Framework* atau *library* yang digunakan adalah *Flask* dengan bahasa pemrograman *Python*. *User interface* dibuat menggunakan *HTML*, *CSS*, dan *JavaScript*. Pembuatan *environment* diperlukan ketika pembangunan *back-end* aplikasi *chatbot*, seperti *Flask 2.2.0*, *NLTK 3.7*, *numpy 1.23.1*, *tensorflow 2.9.1*, dan *scikit-learn 1.1.1*.

HASIL DAN PEMBAHASAN

4.1 Problem Scoping

Analisis masalah dan penentuan solusi untuk menyelesaikan permasalahan berdasarkan metode 4Ws dilakukan pada tahap *problem scoping*. Penulis menganalisis permasalahan keamanan dunia maya dan menentukan solusi

untuk mencegah tindak *cyber-crime*. Solusi yang ditawarkan adalah mengembangkan aplikasi *chatbot* berbasis *website* untuk mengurangi risiko dan meningkatkan kesadaran atau kewaspadaan pengguna *internet* saat beraktivitas *online*. Tabel 1 menunjukkan *problem scoping* menggunakan 4Ws secara rinci.

Tabel 1. *Problem Scoping* 4Ws

Indikator	Keterangan	4Ws
Subjek	Pengguna <i>internet</i> .	<i>Who</i>
Mempunyai masalah	Keamanan dalam dunia maya.	<i>What</i>
Pada saat	Ruang <i>digital</i> atau dunia maya, seperti <i>website</i> dan media sosial.	<i>Where</i>
Solusi	Aplikasi <i>chatbot</i> berbasis <i>website</i> diharapkan dapat membantu dalam mengurangi risiko dan meningkatkan kewaspadaan pengguna <i>internet</i> saat beraktivitas di dunia maya.	<i>Why</i>

4.2 Data Acquisition

Dataset yang digunakan untuk melatih algoritma model *chatbot* merupakan *dataset* manual berupa *file* JSON yang menampung beberapa bagian data seperti *intents*, kelas (*tags*), *pattern*, dan *response*. *Source code dataset* dapat dilihat pada Gambar 10.

```

"intents": [
  {
    "tag": "phishing_prevented",
    "patterns": [
      "Bagaimana cara melindungi diri dari serangan phishing?",
      "Bagaimana terhindar dari serangan phishing?",
      "Bagaimana mencegah serangan phishing?"
    ],
    "responses": [
      "Serangan phishing dapat dicegah dengan tidak mengklik tautan yang mencurigakan atau mengunduh lampiran dari pengirim yang tidak dikenal, mengaktifkan otentikasi dua faktor, dan menjaga perangkat lunak Anda tetap terbaru."
    ]
  },
  .....
]

```

Gambar 10. Potongan *Dataset* dalam Bentuk JSON

4.3 Data Exploration

Tahapan selanjutnya adalah memuat *dataset* yang telah terkumpul dalam bentuk JSON dan mengubahnya ke dalam *dataframe* yang terdiri dari kolom *pattern* dan *tags*. Tujuannya adalah untuk mengolah data menjadi lebih mudah. *Data response* digunakan untuk memberikan jawaban yang sesuai dengan pertanyaan berdasarkan label (*tags*) yang dihasilkan oleh model dan dipilih secara acak. *Dataset* memiliki sebanyak 219 baris *dataset* dan 44 kelas (*tags*). Setelah melakukan memahami karakteristik data, selanjutnya adalah tahap persiapan data yang dimulai dengan *case folding* seperti menghapus puntuasi atau karakter tanda baca dan mengubah fitur menjadi huruf kecil menggunakan *library* NLTK.

Pada tahap lematisasi, data *pattern* (fitur) diubah kedalam bentuk kata dasar dengan kamus kosakata untuk dibersihkan dari imbuhanannya, baik awalan kata, sisipan kata, akhiran kata, atau serta awalan dan akhiran kata. Hasil dari proses lematisasi diketahui bahwa jumlah kata unik sebanyak 139 kata dan memiliki jumlah keseluruhan data *patterns* dan *intents* sebanyak 1441 dokumen.

Selanjutnya adalah proses ekstraksi fitur dengan memecah kalimat menjadi sebuah *token*. *Token* tersebut akan dikelompokkan sesuai dengan makna atau arti tertentu. Dalam hal ini, proses tokenisasi teks dilakukan pada data *patterns*. Pada proses ini, bagian seperti tanda baca akan dihilangkan dan jumlah kata dibatasi sebanyak 2000 kata paling umum yang akan dipertahankan dalam tokenisasi sehingga kata-kata yang muncul di luar 2000 kata akan diabaikan dan dianggap sebagai kata tidak dikenal (*out of vocabulary*). Hal ini akan membantu dalam memperhatikan ukuran vektor *input* yang digunakan dalam model agar tidak

terlalu kompleks dan mengurangi memori dalam proses tokenisasi dan pelatihan model. Setelah dilakukan tokenisasi, proses pemberian urutan (*index*) akan dilakukan pada setiap token unik dan mengubahnya menjadi sebuah *sequence* yang mewakili setiap kata pada data *patterns*.

Kemudian, setiap *sequence* dilakukan proses penambahan *padding* dengan menambahkan nol sebagai awalan atau akhiran hingga mencapai panjang maksimum *sequence* agar setiap *sequence* memiliki panjang yang sama. Selain itu, *padding* juga dapat melakukan pemotongan *sequence* agar panjangnya sesuai dengan panjang maksimum yang ditentukan. Proses ini perlu dilakukan karena *embedding layer* hanya menerima masukan yang memiliki panjang yang sama. Berdasarkan hal tersebut, *input_shape* atau panjang maksimal *input sequence* adalah 14 kata.

Label *encoder* merupakan proses ekstraksi yang dilakukan pada variabel target atau kolom data *tags* yang merupakan data teks. Proses *encode* melakukan suatu konversi atau pengkodean pada data kategorik seperti data teks menjadi numerik berupa vektor biner. Tahap terakhir dari proses *preprocessing* yaitu menyimpan hasil label yang sudah melalui ekstraksi dan menyimpan hasil tokenisasi dalam bentuk *pickle*.

4.4 Modeling

Algoritma model dilatih menggunakan *dataset* yang sudah bersih dan akan digunakan sebagai *input* untuk mengembangkan model *chatbot*. Algoritma *deep learning* yang digunakan untuk mengembangkan *chatbot* pada penelitian ini yaitu algoritma LSTM dengan rangkaian arsitektur lapisan pertama adalah *input layer*, yaitu *layer input* pertama yang menerima *batch* dari *sequence* dengan panjang 14 kata. Lapisan kedua adalah *embedding layer* yang mengambil *input* dari *layer* sebelumnya dan mengubahnya menjadi vektor dengan dimensi 10. Lapisan ketiga adalah LSTM *Layer* yang menerima *input* dari *layer embedding* dan menghasilkan *output* dengan dimensi yang sama. Lapisan keempat adalah *flatten Layer* yang mengubah *output* dari LSTM *layer* menjadi bentuk yang lebih datar dengan dimensi 140. Tujuannya adalah mengubah representasi data menjadi format yang dapat diterima oleh lapisan berikutnya. Lapisan terakhir adalah *dense layer* atau *layer fully*

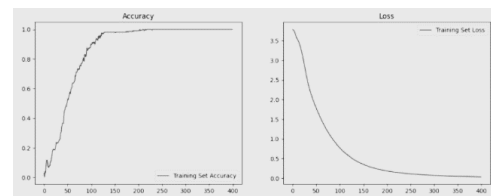
connected dengan 44 *unit neuron*. Lapisan ini menerima *input* dari *flatten layer* dan menggunakan fungsi aktivasi *softmax*. Pada penelitian ini, algoritma model LSTM dilatih dengan perulangan sebanyak 400 *epochs*.

4.5 Evaluation

Proses pelatihan algoritma model diperhatikan dan dilakukan berulang kali untuk mencapai tingkat akurasi yang tinggi dan nilai kesalahan (*loss*) yang rendah. Beberapa matriks yang digunakan untuk mengevaluasi kinerja model LSTM adalah menggunakan matriks “*accuracy*” dan *loss function* dengan *categorical_crossentropy* serta fungsi optimasi *adam*. Setelah dilatih dengan perulangan sebanyak 400 iterasi pelatihan (*epochs*), model LSTM menghasilkan akurasi sebesar 100% dengan *loss* yang kecil sebesar 3.09%. Hasil akurasi dan *loss* pada selama 400 *epochs* dapat dilihat pada Tabel 2. Sedangkan pada Gambar 11 merupakan grafik atau visualisasi plot analisis akurasi dan *loss* dari algoritma pelatihan model LSTM.

Tabel 2. Akurasi dan *Loss* Pelatihan Model LSTM Berdasarkan Jumlah *Epochs*

<i>Epochs</i>	<i>Accuracy</i> (%)	<i>Loss</i> (%)
100	88.13	69.12
200	98.63	16.93
300	100.00	6.30
400	100.00	3.09



Gambar 11. Grafik Analisis Akurasi dan *Loss* Pelatihan Model LSTM

Pada Gambar 11 terlihat bahwa model pelatihan *chatbot* dengan algoritma LSTM menghasilkan model yang baik dan tidak terjadi *overfitting* atau *underfitting*. Setelah mengetahui hasil akurasi dan *loss* pada model dengan algoritma LSTM maka selanjutnya menguji atau *testing* pada *chatbot* yang telah dilatih sebelumnya dan melihat apakah sesuai atau tidak pada saat memasukkan teks kalimat pertanyaan. Tabel 3 menunjukkan hasil pengujian pada *chatbot*.

Tabel 2. Pengujian *Chatbot*

Pertanyaan	Jawaban yang Diharapkan	Keterangan
Apa itu identitas palsu?	Penipuan <i>online</i> merupakan kejahatan seperti penipuan lelang, penjualan produk palsu, penipuan cinta (<i>romance scam</i>), atau penipuan investasi yang dilakukan melalui <i>platform online</i> .	Berhasil
Bagaimana jika ada yang meminta informasi pribadi saya melalui pesan atau <i>email</i> ?	Jangan merespons <i>email</i> atau pesan tersebut dan jangan memberikan informasi pribadi apa pun. Laporkan <i>email</i> tersebut sebagai <i>spam</i> dan hapus.	Berhasil
Bagaimana mencegah serangan <i>phishing</i> ?	Serangan <i>phishing</i> dapat dicegah dengan tidak mengklik tautan yang mencurigakan atau mengunduh lampiran dari pengirim yang tidak dikenal, mengaktifkan autentikasi dua faktor, dan menjaga perangkat lunak Anda tetap terbaru.	Berhasil

Berdasarkan hasil pengujian *chatbot* pada Tabel 3, beberapa pertanyaan dapat dijawab dengan tepat dan relevan dan telah sesuai dengan kalimat pertanyaan dan jawabannya, maka selanjutnya model *chatbot* yang telah dilatih dapat disimpan dengan format .h5 untuk *deployment* aplikasi AI *chatbot* dengan *website*.

4.6 Deployment

Deployment bertujuan membuat model *chatbot* yang telah dibuat mudah digunakan oleh pengguna. *Deployment* model *chatbot* pada penelitian ini dibangun ke dalam sebuah aplikasi berbasis *website* menggunakan *framework* atau *library* yang ada pada bahasa pemrograman *Python* yaitu *Flask* serta *HTML*,

CSS, dan *JavaScript* untuk membuat *user interface*. Halaman aplikasi *chatbot* yang terdiri dari *navigation bar* dan *chatbot*. Kemudian, *chatbot* memiliki 4 fungsi *JavaScript* yang dibuat, yaitu fungsi untuk mengirim pesan dan menampilkan pesan yang dikirim, memanggil API untuk merespon pesan yang dikirim menggunakan *AJAX*, menampilkan hasil respons, dan memanggil serta menampilkan jam dengan menit. Selanjutnya membuat desain *routing* untuk memetakan alamat URL dengan suatu fungsi, yaitu halaman *chatbot* (“/”) dan API *response chatbot* (“/get”). Setelah membuat *route*, model yang telah dilatih beserta pendukungnya kemudian dimuat atau di-load, lalu untuk menjalankan aplikasi *web* penulis menggunakan *ngrok* sebagai *server* agar dapat diakses secara *online* atau lokal. Gambar 12 menunjukkan tampilan halaman *chatbot* pada *website*.



Gambar 12. Tampilan *Chatbot* pada *Website*

SIMPULAN

Penelitian ini menunjukkan bahwa algoritma LSTM berhasil diterapkan pada *chatbot* dalam menyajikan jawaban sebagai media informasi terpusat dan pengetahuan mengenai *cyber security* serta memberikan tips dan trik untuk menghindari kejahatan *cyber-crime* bagi para pengguna *internet*. Model pelatihan *chatbot* dengan algoritma LSTM menghasilkan performa model yang baik dan tidak terjadi *overfitting* atau *underfitting* dengan akurasi sebesar 100% dan *loss* sebesar 3.09%. Kemudian, *chatbot* berhasil dibangun dalam ke dalam sebuah aplikasi berbasis *website* menggunakan *framework* atau *library* yang ada pada bahasa pemrograman *Python* yaitu *Flask* serta *HTML*, *CSS*, dan *JavaScript*. Aplikasi *web* dapat dijalankan menggunakan *ngrok*

sebagai *server* untuk diakses secara *online* atau lokal. Dalam upaya mengurangi resiko *cyber-crime*, *chatbot* ini dapat menjadi salah satu solusi dalam meningkatkan kesadaran pengguna *internet* terhadap *cyber-crime* sehingga mereka dapat beraktivitas dengan aman di dunia maya.

Namun, dikarenakan adanya keterbatasan pada data yang digunakan dalam penelitian ini, penulis menyarankan pelatihan dan pengujian algoritma dilakukan menggunakan *volume* data yang lebih besar pada penelitian selanjutnya. Hal tersebut bertujuan meningkatkan keandalan *chatbot* serta memberikan pemahaman yang lebih akurat dan kuat tentang efektivitas dan efisiensi algoritma.

DAFTAR PUSTAKA

- Adamopoulou, E., & Moussiades, L. (2020). An Overview of Chatbot Technology. In *IFIP Advances in Information and Communication Technology: Vol. 584 IFIP*. Springer International Publishing. https://doi.org/10.1007/978-3-030-49186-4_31
- Aggarwal, S. (2023). *The Ultimate Guide to Building Your Own LSTM Models*. ProjectPro. <https://www.projectpro.io/article/lstm-model/832>
- Al-Khater, W. A., Al-Maadeed, S., Ahmed, A. A., Sadiq, A. S., & Khan, M. K. (2020). Comprehensive review of cybercrime detection techniques. *IEEE Access*, 8, 137293–137311. <https://doi.org/10.1109/ACCESS.2020.3011259>
- Bagian Komunikasi Publik, B. H. dan K.-B. (2020). Rekap Serangan Siber (Januari – April 2020). In <https://Bssn.Go.Id/Rekap-Serangan-Siber-Januari-April-2020/> (pp. 1–1).
- Dhyani, M., & Kumar, R. (2019). An intelligent Chatbot using deep learning with Bidirectional RNN and attention model. *Materials Today: Proceedings*, 34, 817–824. <https://doi.org/10.1016/j.matpr.2020.05.450>
- Gravano, A. (2010). Turn-taking and affirmative cue words in task-oriented dialogue. *Dissertation Abstracts International, B: Sciences and Engineering*, 70(8), 4943. <https://doi.org/10.1162/COLI>
- Habibi, M. R., & Liviani, I. (2020). Kejahatan Teknologi Informasi (Cyber Crime) dan Penanggulangannya dalam Sistem Hukum Indonesia. *Al-Qanun: Jurnal Pemikiran Dan Pembaharuan Hukum Islam*, 23(2), 400–426. <http://jurnal.fsh.uinsby.ac.id/index.php/qanun/article/view/1132>
- Kemp, S. (2022). *Digital 2022 Global Overview Report: The Essential Guide to the World's Connected Behaviours*. DataReportal. <https://datareportal.com/reports/digital-2022-global-overview-report>
- Khanna, A., Pandey, B., Vashishta, K., Kalia, K., Pradeepkumar, B., & Das, T. (2015). A Study of Today's A.I. through Chatbots and Rediscovery of Machine Intelligence. *International Journal of U- and e-Service, Science and Technology*, 8(7), 277–284. <https://doi.org/10.14257/ijunesst.2015.8.7.28>
- Lalwani, T., Bhalotia, S., Pal, A., Bisen, S., & Rathod, V. (2018). Implementation of a Chat Bot System using AI and NLP. *International Journal of Innovative Research in Computer Science & Technology*, 6(3), 26–30. <https://doi.org/10.21276/ijircst.2018.6.3.2>
- Muhidin, A., Danny, M., & Rilvani, E. (2023). Algoritme Multinomial Naïve Bayes Pada Aplikasi Chatbot Layanan Informasi Berbasis Teks. *Progresif: Jurnal Ilmiah ...*, 71–80. <http://ojs.stmik-banjarbaru.ac.id/index.php/progresif/article/view/1113%0Ahttp://ojs.stmik-banjarbaru.ac.id/index.php/progresif/article/download/1113/640>
- Nimavat, K., & Champaneria, T. (2017). Chatbots: An Overview Types, Architecture, Tools and Future Possibilities. *International Journal of Scientific Research and Development*, 5(7), 1019–1026.
- Nugraha, K. A., & Sebastian, D. (2021). Chatbot Layanan Akademik Menggunakan K-Nearest Neighbor. *Jurnal Sains Dan Informatika*, 7(1), 11–19. <https://doi.org/10.34128/jsi.v7i1.285>
- Nur Latifah, F., Mawardi, I., & Wardhana, B. (2022). Threat of Data Theft (Phishing) Amid Trends in Fintech Users During the Covid-19 Pandemic (Study Phishing In

- Indonesia). *Perisai : Islamic Banking and Finance Journal*, 6(1), 74–86. <https://doi.org/10.21070/perisai.v6i1.1598>
- Prabowo, Y. D., Warnars, H. L. H. S., Budiharto, W., Kistijantoro, A. I., Heryadi, Y., & Lukas. (2019). Lstm and Simple Rnn Comparison in the Problem of Sequence to Sequence on Conversation Data Using Bahasa Indonesia. *1st 2018 Indonesian Association for Pattern Recognition International Conference, INAPR 2018 - Proceedings*, 51–56. <https://doi.org/10.1109/INAPR.2018.8627029>
- Purwitasari, N. A., & Soleh, M. (2022). Implementasi Algoritma Artificial Neural Network Dalam Pembuatan Chatbot Menggunakan Pendekatan Natural Language Parocessing. *Jurnal IPTEK*, 6(1), 14–21. <https://doi.org/10.31543/jii.v6i1.192>
- Putra, O. V., Musthafa, A., & Wibowo, K. P. (2021). Klasifikasi Ekspresi Teks Berbahasa Jawa Menggunakan Algoritma Long Short Term Memory. *Komputika : Jurnal Sistem Komputer*, 10(2), 137–143. <https://doi.org/10.34010/komputika.v11i1.4616>
- Rian Handoko, & Tata Sutabri. (2023). Analisa Machine Learning Dengan Algoritma Multi-Layer Perceptron Untuk Penanganan Kejahatan Phishing. *Jurnal Informatika Teknologi Dan Sains*, 5(1), 13–17. <https://doi.org/10.51401/jinteks.v5i1.2221>
- Rizki, U. (2019). Multi Respon Ranking Pada Percakapan Layanan Travel Berdasarkan Riwayat Obrolan. *Jurnal INFORMA: Jurnal Penelitian Dan Pengabdian Masyarakat*, 5(3), 73–79. <https://doi.org/10.46808/informa.v5i3.150>
- Sennhauser, L., & Berwick, R. C. (2018). Evaluating the Ability of LSTMs to Learn Context-Free Grammars. *EMNLP 2018 - 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP, Proceedings of the 1st Workshop*, 115–124. <https://doi.org/10.18653/v1/w18-5414>
- Siddhartha. (2021). *AI Project Cycle, the 5 Stages*. 7-Hidden Layers. https://7-hiddenlayers.com/ai_project_cycle/
- Silvanie, A., & Subekti, R. (2022). Aplikasi Chatbot Untuk Faq Akademik Di Ibi-K57 Dengan Lstm Dan Penyematan Kata. *JIKO (Jurnal Informatika Dan Komputer)*, 5(1), 19–27. <https://doi.org/10.33387/jiko.v5i1.3703>
- Trivusi. (2022). *Mengenal Algoritma Long Short Term Memory (LSTM)*. Trivusi. <https://www.trivusi.web.id/2022/07/algoritma-lstm.html>
- Widodo, S., Setiawan, D., Ridwan, T., & Ambari, R. (2022). Perancangan Deteksi Emosi Manusia berdasarkan Ekspresi Wajah Menggunakan Algoritma VGG16. *Syntax : Jurnal Informatika*, 11(01), 01–12. <https://doi.org/10.35706/syji.v11i01.6594>
- Wintoro, P. B., Hermawan, H., Muda, M. A., & Mulyani, Y. (2022). Implementasi Long Short-Term Memory pada Chatbot Informasi Akademik Teknik Informatika Unila. *EXPERT: Jurnal Manajemen Sistem Informasi Dan Teknologi*, 12(1), 68. <https://doi.org/10.36448/expert.v12i1.2593>
- Witanto, K. S., Sanjaya ER, N. A., Karyawati, A. E., Kadyanan, I. G. A. G. A., Suhartana, I. K. G., & Astuti, L. G. (2022). Implementasi LSTM Pada Analisis Sentimen Review Film Menggunakan Adam Dan RMSprop Optimizer. *JELIKU (Jurnal Elektronik Ilmu Komputer Udayana)*, 10(4), 351. <https://doi.org/10.24843/jlk.2022.v10.i04.p05>
- Yunmar, R. A., & Wisesa, I. W. W. (2020). Pengembangan Mobile-Based Question Answering System Mobile-Based Question Answering System Development With Ontology Based Knowledge. *Jurnal Teknologi Informasi Dan Ilmu Komputer (JTIK)*, 7(4), 693–700. <https://doi.org/10.25126/jtiik.202072255>
- Zhao, Z., Chen, W., Wu, X., Chen, P. C. Y., & Liu, J. (2017). LSTM network: A deep learning approach for Short-term traffic forecast. *IET Intelligent Transport Systems*, 11(2), 68–75. <https://doi.org/10.1049/iet-its.2016.0208>