



Desain Chatbot Inhibitor Korosi Berbasis Bidirectional Long Short-Term Memory Neural Network

Corrosion Inhibitor Chatbot Design Based on Bidirectional Short Term Memory Neural Network

Nibras Bahy Ardyansyah¹⁾, Nicholaus Verdhy Putranto¹⁾, Dzaki Ashari Surya Putra¹⁾, Gustina Alfa Trisnapradika¹⁾,
²⁾, Muhamad Akrom^{1),2)} *

¹⁾Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Dian Nuswantoro
Jl. Imam Bonjol No.207, Pendrikan Kidul, Kec.Semarang Tengah, Semarang, Indonesia 50131

²⁾Research Center for Materials Informatics, Fakultas Ilmu Komputer, Universitas Dian Nuswantoro
Jl. Imam Bonjol No.207, Pendrikan Kidul, Kec.Semarang Tengah, Semarang, Indonesia 50131

How to cite: N. P. Pertama and N. P. Kedua, "Petunjuk penulisan dan kirim artikel Jurnal Teknologi dan Sistem Komputer mulai penerbitan nomor 6(4) tahun 2018," *Jurnal Teknologi dan Sistem Komputer*, vol. 9, no. x, pp. xx-xx, 2021. doi: [10.14710/jtsiskom.2022.xxxxx](https://doi.org/10.14710/jtsiskom.2022.xxxxx) [Online].

Abstract – This research delves into the intricate phenomenon of corrosion, a process entailing material degradation through chemical reactions with the environment, causing consequential losses across diverse sectors. In response, corrosion inhibitors serve as a proactive measure to counteract this deleterious impact. Despite their paramount significance, public awareness regarding corrosion and inhibitors remains limited, necessitating intensified educational efforts. The primary focus of this study is the development of a Chatbot system designed to disseminate information on corrosion, inhibitors, and related topics. Employing the Machine Learning Life Cycle model, a deep learning approach, specifically the Bidirectional Long Short-Term Memory (BLSTM) architecture, is utilized to construct an optimized Chatbot model. Post-training evaluation of the BLSTM model reveals noteworthy performance metrics, including a remarkable 100% accuracy rate and a substantial 92% validation accuracy over 100 epochs. Training and validation losses are reported as 0.2292 and 0.9342, respectively. In conclusion, the BLSTM algorithm emerges as an effective tool for training and enhancing Chatbot models, ensuring commendable performance in the domain of corrosion awareness and inhibition.

Keywords – corrosion; inhibitor; Chatbot; deep learning; LSTM.

Abstrak – Penelitian ini menyelidiki fenomena korosi yang rumit, sebuah proses yang menyebabkan degradasi material melalui reaksi kimia dengan lingkungan, yang menyebabkan kerugian besar di berbagai sektor. Sebagai tanggapan, inhibitor korosi berfungsi sebagai tindakan proaktif untuk mengatasi dampak buruk ini. Meskipun sangat penting, kesadaran masyarakat mengenai korosi dan inhibitor

masih terbatas, sehingga memerlukan upaya pendidikan yang lebih intensif. Fokus utama penelitian ini adalah pengembangan sistem Chatbot yang dirancang untuk menyebarkan informasi mengenai korosi, inhibitor, dan topik terkait. Dengan menggunakan model Machine Learning Life Cycle, khususnya arsitektur BiDirectional Long Short-Term Memory (BLSTM), digunakan untuk membangun model Chatbot yang dioptimalkan. Evaluasi pasca-pelatihan model BLSTM mengungkapkan metrik kinerja yang patut diperhatikan, termasuk tingkat akurasi 100% yang luar biasa dan akurasi validasi sebesar 92% selama 100 periode. Kerugian pelatihan dan validasi dilaporkan masing-masing sebesar 0,2292 dan 0,9342. Kesimpulannya, algoritme BLSTM muncul sebagai alat yang efektif untuk melatih dan menyempurnakan model Chatbot, memastikan kinerja terpuji dalam bidang kesadaran dan penghambatan korosi.

Kata kunci – korosi; inhibitor; Chatbot; deep learning; LSTM.

I. PENDAHULUAN

Korosi merupakan suatu proses degradasi material yang terjadi melalui reaksi kimia dengan lingkungan sekitarnya, menyebabkan kerusakan struktural dan penurunan kualitas material [1], [2]. Meskipun sering terjadi dalam kehidupan sehari-hari, korosi kadang tidak disadari keberadaannya di sekitar kita. Faktor lingkungan seperti air, udara, dan zat kimia tertentu berperan dalam mempercepat proses ini [3]. Kontak antara dua logam berbeda dalam elektrolit, interaksi mikroorganisme, serta korosi yang dipicu oleh tegangan juga dapat mempercepat kerusakan material [4]. Dampak korosi merambah ke berbagai sektor, termasuk industri ekonomi, lingkungan, dan keamanan [5], [6]. Pencegahan korosi dapat dilakukan melalui metode

*) Corresponding author (Muhamad Akrom)
Email: m.akrom@dsn.dinus.ac.id

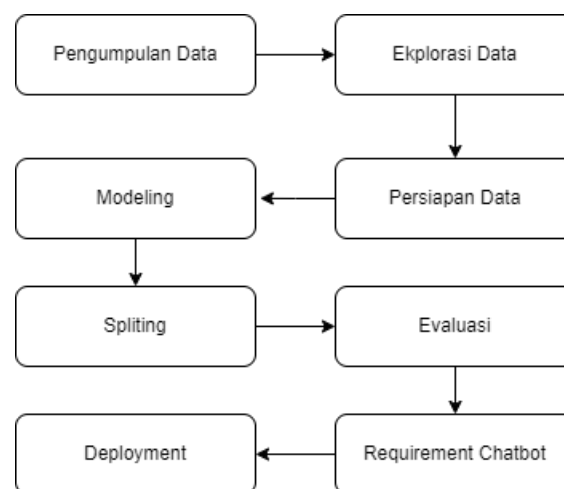
seperti pelapisan permukaan logam, perlindungan katodik, dan penggunaan inhibitor korosi [7], [8]. Penggunaan inhibitor korosi menjadi langkah penting dalam melawan dampak kerusakan yang diakibatkan oleh korosi [5], [9]. Meski demikian, pengetahuan umum tentang korosi dan inhibitor korosi masih terbatas. Upaya edukasi diperlukan untuk meningkatkan pemahaman masyarakat terhadap korosi. Metode penyampaian informasi yang efektif dan dapat diakses oleh berbagai kalangan menjadi kunci untuk meningkatkan kesadaran mengenai pencegahan korosi [10], [11].

Chatbot adalah suatu program yang dirancang untuk menjalani interaksi manusia-komputer dengan memanfaatkan sistem Artificial Intelligence (AI) seperti Natural Language Processing (NLP) [12], [13]. Chatbot mampu menerima berbagai input seperti teks dan memberikan respons berdasarkan pola yang telah diprogram sebelumnya [14]. Terdapat dua klasifikasi utama dalam Chatbot, yakni open domain dan closed domain. Chatbot open domain dapat merespons dengan tepat terhadap berbagai topik umum, sementara Chatbot closed domain hanya dapat memberikan respons terkait topik tertentu dan mungkin tidak efektif dalam merespons topik lainnya [12], [15]. Selain itu, Chatbot memiliki kemampuan untuk disesuaikan dan dilatih dalam berbagai bahasa, memenuhi kebutuhan yang beragam [16]. Dalam pengembangan Chatbot, diperlukan suatu model yang dapat menjalani proses pelatihan dan pengujian menggunakan algoritma machine learning, seperti Neural Network [17]. Jurnal ini membahas implementasi atau pengembangan dari penggunaan sistem NLP, yaitu Chatbot inhibitor korosi, dengan menggunakan pendekatan Deep Learning dan algoritma Bidirectional Long Short-Term Memory (BLSTM).

Beberapa metode yang umumnya diterapkan dalam pengembangan Chatbot melibatkan Recurrent Neural Network dan Long Short-Term Memory (RNN-LSTM) [18], Bidirectional Long Short-Term Memory (BLSTM), dan Natural Language Processing (NLP) [19]. Algoritma LSTM memungkinkan pemrosesan bahasa alami yang mempermudah pengguna berinteraksi dengan komputer [20]. Oleh karena itu, metode LSTM sering digunakan dalam pengembangan Chatbot karena mampu menerima input dan menghasilkan output berdasarkan pola yang telah dipelajari sebelumnya [21].

Penelitian ini mengungkap inovasi dengan menerapkan algoritma LSTM pada pengembangan *Chatbot*. Fokus utama *Chatbot* adalah memberikan respons terkait pertanyaan pengguna mengenai informasi dan pengetahuan seputar inhibitor korosi, serta memberikan saran efektif untuk pencegahan korosi. Proses pelatihan model LSTM dilakukan secara mandiri menggunakan platform Python dan Jupyter Notebook. Keputusan strategis dalam memilih LSTM sebagai algoritma utama dilandaskan pada hasil penelitian terdahulu yang menunjukkan tingkat akurasi klasifikasi yang lebih superior dibandingkan dengan

metode lain seperti RNN dan K-Nearest Neighbors (KNN) [22]. Dengan demikian, penelitian ini mengusulkan penggunaan LSTM sebagai fondasi yang lebih handal untuk membangun *Chatbot* yang dapat memberikan solusi inovatif terhadap pertanyaan dan tantangan seputar inhibitor korosi.



Gambar 1. Flowchart Tahapan Penelitian

II. METODE PENELITIAN

Penelitian ini dijalankan melalui serangkaian tahapan dalam Machine Learning Model Development Life Cycle (MDLC), yang menjadi panduan untuk melaksanakan penelitian [20]. MDLC adalah suatu rangkaian langkah untuk pengembangan model machine learning mulai dari tahap awal hingga tahap akhir. Proses ini mencakup langkah-langkah dalam penelitian yang dijelaskan sebagai berikut, seperti yang terlihat pada Gambar 1.

A. Pengumpulan Data

Tahap awal dalam pengembangan melibatkan pengumpulan data, yang dilakukan melalui riset dan pengumpulan data yang relevan. Data dikumpulkan secara manual, berdasarkan tinjauan literatur mengenai korosi dan inhibitor korosi dari berbagai sumber online yang terpercaya, seperti website Alodokter, HaloDoc, drugbank, idnmedis, vinmec, PubChem, drugs, alomedika, medicinka, dan sumber-sumber lainnya. Setelah data terkumpul, informasi tersebut disusun menjadi dataset untuk melatih algoritma model Chatbot, dengan format penyimpanan file JavaScript Object Notation (JSON). Dataset ini memiliki struktur sebagai berikut:

1. Tag: Berfungsi untuk mengelompokkan data teks yang serupa dan menggunakan output yang sama sebagai target untuk melatih jaringan neural.
2. Patterns: Merupakan komponen yang berisi data pola input yang diharapkan sesuai dengan inputan pengguna. Patterns digunakan sebagai predictor.
3. Responses: Bagian data yang berisi jawaban atau output yang akan dikirimkan berdasarkan index tag dan pattern yang ditentukan oleh sistem.
- 4.

B. Eksplorasi Data

Proses pada tahap ini berfungsi sebagai kualifikasi pertama sebelum melakukan pengembangan model. Data yang dikumpulkan mencakup informasi terkait pengertian korosi, pencegahan korosi, pengertian inhibitor, jenis-jenis inhibitor, mekanisme inhibitor, pengertian inhibitor korosi, pengertian obat, pengertian kadaluarsa, pengertian obat kadaluarsa, obat inhibitor korosi, dan nama-nama senyawa kimia. Data ini akan dijadikan sebagai jawaban atau respons dalam dataset. Chatbot yang dikembangkan menggunakan pendekatan single response, di mana setiap pertanyaan dari pengguna akan dijawab dengan satu respons yang singkat dan langsung sesuai dengan konteks pertanyaan yang diajukan.

C. Persiapan Data

Pada tahap ini, dilakukan pemrosesan data untuk mengubah data mentah menjadi data yang siap digunakan dalam proses pelatihan model machine learning pada tahap selanjutnya. Eksplorasi data dan persiapan data, atau yang dikenal sebagai preprocessing data, bertujuan memastikan bahwa data yang digunakan dalam pelatihan model Chatbot bersih, konsisten, dan representatif, sehingga dapat menghasilkan hasil akhir yang lebih baik [22], [23]. Tahapan yang dilakukan melibatkan transformasi data ke dalam format dataset yang sesuai, menggunakan library pandas untuk mempermudah manipulasi atau modifikasi, melakukan tokenisasi untuk mengonversi teks pada pola menjadi urutan angka, menambahkan padding terhadap urutan angka agar memiliki panjang yang sama, dan mengonversi label tag menjadi angka dengan menggunakan teknik LabelEncoder. Dengan data yang telah melewati tahapan-tahapan tersebut, data telah disiapkan secara optimal untuk digunakan dalam pelatihan model.

D. Modeling

Tahap berikutnya dalam pengembangan melibatkan proses pemodelan dengan memilih algoritma yang efektif untuk mengenali dan merespons pertanyaan dengan baik. Algoritma yang akan digunakan untuk melatih Chatbot adalah algoritma deep learning, yaitu LSTM, karena mampu mencapai tingkat akurasi yang tinggi dan memiliki kemampuan respons yang lebih baik [24], serta dapat merespons dengan cepat. Model LSTM ini akan digunakan untuk melatih Chatbot guna meningkatkan kinerjanya, menggunakan data yang telah melalui tahap pemrosesan. Struktur model LSTM yang akan digunakan mencakup Embedding layer, Bidirectional LSTM, dropout, Normalization layer, LSTM, Dense, dan proses kompilasi model menggunakan optimizer adam dan fungsi loss menggunakan categorical_crossentropy, dengan matriks evaluasi berupa accuracy.

E. Splitting

Tahap selanjutnya adalah proses pemisahan dataset yang telah melalui proses preprocessing menjadi dua bagian utama: data training dan data validation. Pembagian ini bertujuan untuk melatih model dengan sebagian data dan menguji performanya dengan menggunakan data yang belum pernah dilihat oleh model sebelumnya. Proses pelatihan dilakukan selama 100 epoch untuk meningkatkan akurasi dan kinerja model. Penggunaan fungsi train test split membagi dataset dengan proporsi 80:20, di mana 80% menjadi data training dan 20% menjadi data validation. Data training digunakan sebagai input dan output untuk model, sementara data validation digunakan untuk mengevaluasi performa model selama pelatihan [20].

F. Evaluasi

Dalam tahap evaluasi ini, hasil dari model yang telah dilatih ditampilkan melalui penggunaan matriks evaluasi. Metrik evaluasi yang digunakan mencakup akurasi, validasi akurasi, kerugian (loss), dan kerugian validasi (validation loss). Pada model ini, fungsi kerugian yang diterapkan adalah categorical cross-entropy, sementara pengoptimasi menggunakan metode adam. Selanjutnya, untuk memantau kinerja model pada setiap epoch, dilakukan visualisasi dengan menampilkan grafik akurasi pelatihan (training accuracy), akurasi validasi (validation accuracy), kerugian pelatihan (training loss), dan kerugian validasi (validation loss). Proses evaluasi ini memberikan gambaran holistik tentang sejauh mana model dapat memberikan respons yang akurat dan konsisten. Grafik akurasi dan kerugian memberikan wawasan terhadap performa model selama proses pelatihan dan validasi. Setelah proses evaluasi, model terbaik yang telah dilatih dapat dipilih untuk digunakan dalam implementasi Chatbot [25].

G. Requirement Chatbot

Pada tahap ini, model Chatbot terbaik digunakan untuk fungsi inputan pengguna yang bertujuan memberikan interaksi antara pengguna dengan model yang telah dilatih. Pengguna memasukkan pola kalimat sebagai input, dan fungsi ini akan memproses, melakukan prediksi kelas, dan menampilkan respons acak dari Chatbot berdasarkan model yang telah terlatih sebelumnya. Proses preprocessing melibatkan beberapa langkah, seperti penghapusan karakter, konversi ke huruf kecil, dan tokenisasi. Setelah mendapatkan hasil dari tokenisasi, data diubah menjadi urutan angka dan dilakukan padding jika diperlukan. Selanjutnya, model melakukan prediksi label tag, dan respons yang sesuai diambil dari dataset respons berdasarkan label tag tersebut. Hasil akhir dari interaksi ini adalah menampilkan inputan pengguna dan respons acak dari Chatbot.

H. Deployment

Pada tahap ini, model Chatbot akan di-deploy untuk dapat diakses secara online oleh pengguna. Dalam penelitian ini, proses deployment menggunakan

framework Streamlit dengan bahasa pemrograman Python. Untuk menjalankan proses development, diperlukan environment dengan versi library/scikit-learn 1.3.2, tensorflow 2.12.0, numpy 1.23., dan Streamlit 1.29.0.

III. HASIL DAN PEMBAHASAN

Pada penelitian ini menjelaskan pengembangan *Chatbot* untuk membantu meningkatkan pemahaman masyarakat tentang korosi. *Chatbot* ini dikembangkan dengan algoritma *machine learning* berdasarkan teknik *Natural Language Processing* (NLP).

Dataset yang digunakan untuk melatih algoritma model *Chatbot* merupakan dataset manual berupa file JSON. Dataset ini menyimpan beberapa komponen daya yaitu *intents*, *tag*, *pattern*, dan *response*. Struktur dataset pada Gambar 2 memungkinkan *Chatbot* untuk memahami dan memberikan respons yang tepat terhadap pertanyaan yang diajukan oleh pengguna sesuai dengan topik yang ditentukan. Dalam pengembangan model melibatkan penggunaan dataset yang telah terkumpul dalam bentuk JSON dan mengubahnya ke dalam dataframe yang terdiri dari kolom *pattern* dan *tag* sebagai kolom utama. Proses ini dilakukan dengan tujuan menyusun data agar lebih mudah diolah dan terstruktur. Data pada kolom *response* digunakan untuk memberikan jawaban yang sesuai dengan pertanyaan, berdasarkan *tag* yang dihasilkan oleh model, dan dipilih secara acak untuk meningkatkan variasi respons. Dataset ini memiliki sebanyak 948 baris dan 303 *tag*.

```

{"intents": [
  {
    "tag": "salam",
    "patterns": [
      "Halo",
      "Hi",
      "Hai",
      "Hai bot",
      "Halo bot",
      "Hi bot",
      "Ada yang bisa saya bantu?"
    ],
    "responses": [
      "Halo! Apakah ada yang bisa saya bantu?",
      "Hi! Apa yang bisa saya bantu?",
      "Hai! Apa kabar? Ada yang perlu ditanyakan?",
      "Hello! Bagaimana aku bisa membantumu?",
      "Hai! Ada yang bisa saya bantu hari ini?",
      "Hi there! Bagaimana saya bisa membantu Anda?",
      "Selamat datang! Apa yang bisa saya lakukan untuk Anda?"
    ]
  }
],
}

```

Gambar 2. Potongan Datase dalam bentuk JSON

Tokenisasi diterapkan pada teks pola *patterns*. Proses ini mengkonversi teks menjadi urutan angka, menjadikan langkah kunci dalam pemrosesan data untuk model NLP, dilanjutkan dengan proses padding, dengan menyisipkan nol sebagai awasal atau akhiran yang bertujuan untuk menyamakan panjang urutan angka pada setiap sampel data. Sehingga dimensi input selama proses pelatihan tetap seragam. Kemudian dilakukan proses *Label Encoder* diterapkan pada variabel target, yakni pada kolom data *tag*, untuk mengubah atau mengkonversi menjadi representasi angka dalam bentuk vektor biner. Hasil dari persiapan data atau preparation

pada *x* dan *y* dapat dilihat pada Gambar 3 dan Gambar 4.

```

x
array([[318,  0,  0, ...,  0,  0,  0],
       [319,  0,  0, ...,  0,  0,  0],
       [320,  0,  0, ...,  0,  0,  0],
       ...,
       [  1,  2,  3, ...,  0,  0,  0],
       [  8,  5, 317, ...,  0,  0,  0],
       [  9,  4,  7, ...,  0,  0,  0]])

```

Gambar 3. Data Preparation pada X

```

y
array([278, 278, 278, 278, 278, 278, 278, 280, 280, 280, 280, 281, 281,
       281, 281, 282, 282, 282, 282, 279, 279, 279, 275, 275, 275,
       275, 275, 158, 158, 158, 158, 158, 7, 7, 7, 7, 10, 10,
       10, 10, 166, 166, 166, 166, 13, 13, 13, 159, 159, 159, 205, 205,
       205, 205, 205, 164, 164, 164, 164, 164, 206, 206, 206, 206, 206,
       12, 12, 12, 14, 14, 14, 14, 14, 15, 15, 15, 15, 16, 16,
       16, 16, 238, 238, 238, 200, 200, 200, 18, 18, 18, 18, 17,
       17, 17, 17, 49, 49, 49, 49, 49, 287, 287, 287, 287, 11,
       11, 11, 11, 284, 284, 284, 256, 256, 256, 256, 288, 288, 288,
       288, 285, 285, 285, 285, 289, 289, 289, 289, 290, 290, 290,
       276, 276, 276, 276, 1, 1, 1, 1, 1, 2, 2, 2, 4,
       4, 4, 3, 3, 3, 5, 5, 5, 5, 0, 0, 0, 286,
       286, 286, 6, 6, 6, 6, 8, 8, 8, 8, 8, 9, 9, 277,
       277, 277, 283, 283, 283, 68, 68, 68, 21, 21, 21, 23, 23,
       23, 24, 24, 24, 47, 47, 47, 22, 22, 22, 25, 25, 25,
       26, 26, 26, 27, 27, 27, 28, 28, 28, 29, 29, 29, 30,
       30, 30, 31, 31, 31, 33, 33, 33, 32, 32, 32, 34, 34,

```

Gambar 4. Data Preparation pada Y

Algoritma deep learning Bidirectional LSTM digunakan dalam pengembangan model *Chatbot* pada data yang telah diolah untuk proses pelatihan. Dalam Gambar 5, dapat dilihat struktur arsitektur dari model yang terdiri dari beberapa lapisan. Proses dimulai dengan lapisan input sebagai lapisan pertama, yang menerima batch dari sequence dengan panjang urutan 11. Lapisan kedua adalah lapisan embedding, yang menerima input dari lapisan sebelumnya dan mengubahnya menjadi vektor dengan 100 dimensi. Lapisan ketiga merupakan lapisan Bidirectional LSTM dengan 256 unit parameter, yang dapat menghasilkan urutan nilai dalam urutan input dan dilengkapi dengan dropout untuk mencegah overfitting. Lapisan dropout selanjutnya digunakan untuk mencegah overfitting dan meningkatkan kemampuan *Chatbot*.

Lapisan normalisasi pada setiap lapisannya digunakan untuk mengoptimalkan proses pelatihan. Kemudian, lapisan LSTM kedua dilengkapi dengan dropout dan normalisasi untuk mencegah overfitting dan menormalisasi data. Setelah itu, lapisan dense layer dengan 128 unit dan fungsi aktivasi ReLU digunakan untuk memberikan non-lineritas pada model. Lapisan dropout dan normalisasi kembali diterapkan sebelum menggunakan fungsi aktivasi softmax. Model LSTM dilatih selama 100 iterasi (epoch) untuk mencapai hasil optimal. Model dikompilasi menggunakan optimizer adam dan fungsi loss categorical_crossentropy, dengan matriks evaluasi berupa accuracy.

Layer (type)	Output Shape
embedding_2 (Embedding)	(None, 11, 100)
bidirectional_2 (Bidirectional)	(None, 11, 512)
dropout_6 (Dropout)	(None, 11, 512)
layer_normalization_6 (Layer Normalization)	(None, 11, 512)
lstm_5 (LSTM)	(None, 256)
dropout_7 (Dropout)	(None, 256)
layer_normalization_7 (Layer Normalization)	(None, 256)
dense_4 (Dense)	(None, 128)
layer_normalization_8 (Layer Normalization)	(None, 128)
dropout_8 (Dropout)	(None, 128)
dense_5 (Dense)	(None, 291)

Gambar 5. Struktur Model LSTM

Proses pelatihan model ini dimulai dengan membagi dataset menjadi dua bagian utama, yaitu data pelatihan (training) dan data evaluasi (validation). Pembagian ini bertujuan untuk melatih model menggunakan sebagian data dan menguji performanya menggunakan data yang belum pernah dilihat oleh model sebelumnya. Proses pelatihan dilakukan selama 100 epoch dengan tujuan meningkatkan akurasi dan kinerja model. Dalam implementasi pelatihannya, digunakan fungsi train test split yang membagi dataset dengan proporsi 80:20, di mana 80% data menjadi bagian training dan 20% data menjadi bagian validasi. Data training digunakan sebagai input dan output untuk model, sementara data validation digunakan untuk mengevaluasi performa model selama proses pelatihan. Proses ini bertujuan agar model dapat belajar dari data training.

Berdasarkan Tabel 1 dan Gambar 6, setelah melalui 100 epoch, model LSTM berhasil mencapai akurasi tinggi sebesar 100% dan akurasi validasi sebesar 92.63% dengan nilai loss sebesar 0.2292 dan loss validasi sebesar 0.9342. Hasil ini menunjukkan bahwa model memiliki kinerja yang sangat baik pada data pelatihan dan mampu melakukan prediksi dengan baik pada data validasi.

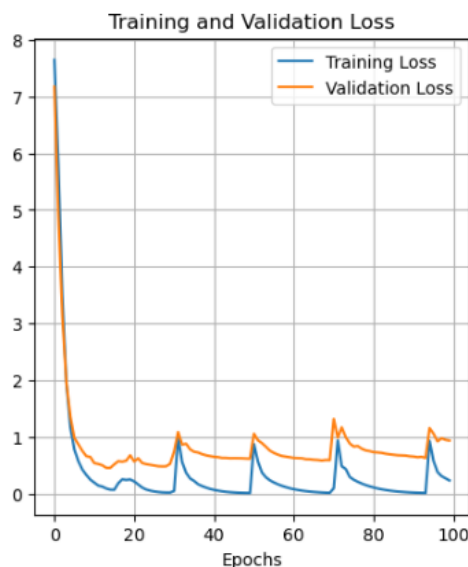


Gambar 6. Grafik Akurasi Training dan Validation

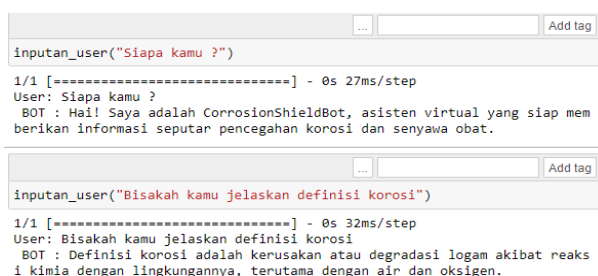
Tabel 1. Metrik Akurasi dan loss beserta validasi dari pelatihan LSTM

Epoch	Accuracy	Val_Accuracy	Loss	Val_Loss
10	0.9987	0.9368	0.2450	0.6438
20	0.9947	0.9263	0.2474	0.6736
30	1.0000	0.9263	0.0165	0.5236
40	1.0000	0.9316	0.0814	0.6596
50	1.0000	0.9316	0.0079	0.6122
60	1.0000	0.9368	0.0920	0.6388
70	1.0000	0.9368	0.0114	0.5882
80	1.0000	0.9316	0.1255	0.7475
90	1.0000	0.9316	0.0189	0.6571
100	1.0000	0.9263	0.2292	0.9342

Pada model ini, metrik evaluasi yang digunakan mencakup akurasi (accuracy), validasi akurasi (validation accuracy), loss, dan validation loss. Fungsi loss yang diterapkan pada model ini adalah sparse categorical crossentropy, dan optimizer yang digunakan adalah adam. Evaluasi performa model dilakukan dengan memvisualisasikan grafik Training Accuracy, Validation Accuracy, Training Loss, dan Validation Loss. Pada Gambar 7 dan Gambar 8, terlihat bahwa hasil dari pelatihan menunjukkan kemampuan model untuk mempelajari data training dengan baik. Namun, perlu diperhatikan bahwa validation accuracy sedikit lebih rendah dari accuracy training, yang dapat menunjukkan adanya overfitting. Hal yang serupa terjadi pada grafik loss, di mana validation loss berhasil mengurangi nilai training loss dan validation loss dengan cepat, tetapi nilai validation loss cenderung fluktuatif setelah beberapa epoch. Maka dari itu, hasil ini menunjukkan bahwa model dapat dengan baik memahami pola dalam data dan memberikan prediksi yang baik, namun perlu diperhatikan potensi overfitting pada data training.

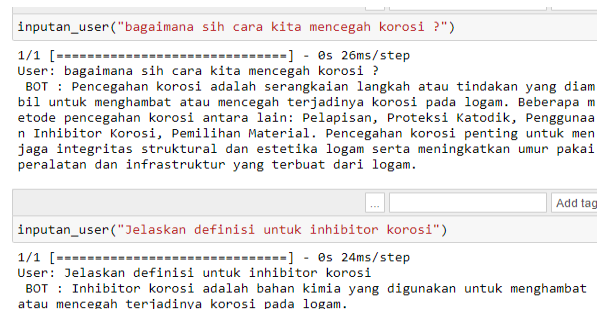


Gambar 7. Grafik loss training dan validation

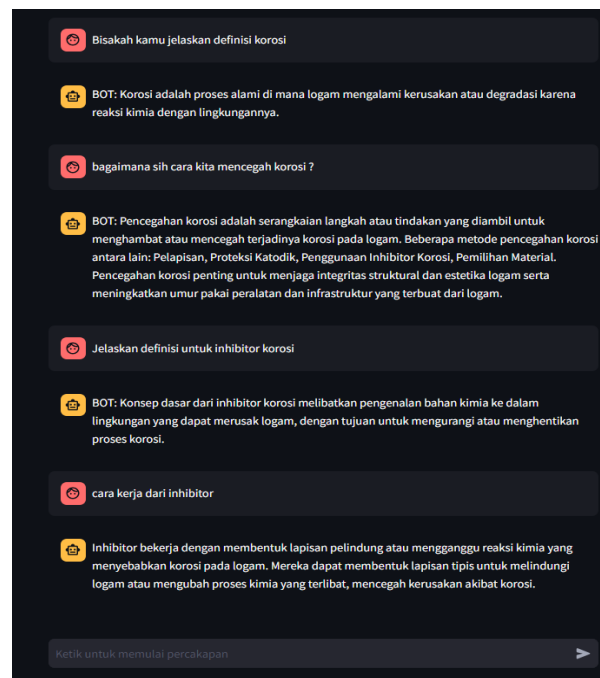


Gambar 8. Hasil menguji chatbot

Setelah melalui proses preprocessing, model memprediksi label tag berdasarkan input, dan respons yang sesuai diambil dari dataset responses yang telah di-training sebelumnya. Seperti pada Gambar 9 dan Gambar 10, beberapa hasil pengujian dengan memasukkan beberapa pertanyaan menunjukkan bahwa model Chatbot dapat memprediksi jawabannya dengan tepat. *Deployment* dilakukan agar model *Chatbot* yang dibuat dapat digunakan oleh pengguna. Model *Chatbot* pada penelitian ini dibangun kedalam sebuah aplikasi berbasis website menggunakan *framework* atau *library* yang ada pada bahasa pemrograman *python* yaitu *Streamlit* untuk merancang antarmuka pengguna. Hasil dari tahap deployment dapat dilihat di Gambar 10, yang menunjukkan antarmuka aplikasi web *Chatbot*. Pengguna dapat memasukkan pesan atau pertanyaan ke dalam kolom yang disediakan, kemudian dengan menekan tombol kirim, *Chatbot* akan memberikan jawaban yang sesuai dengan pertanyaan tersebut. Jika pengguna ingin bertanya lebih lanjut pengguna dapat memasukkan pesan baru dan mengirimkannya kepada *Chatbot* kembali dengan cara yang sama.



Gambar 9. Hasil menguji chatbot



Gambar 10. Tampilan website chatbot

IV. KESIMPULAN

Keberhasilan implementasi algoritma Long Short-Term Memory (LSTM) pada website *Chatbot* inhibitor korosi terbukti berperan dalam meningkatkan pemahaman tentang korosi. Model pelatihan *Chatbot*, yang menggunakan algoritme LSTM, telah menunjukkan kinerja yang patut dipuji, mencapai tingkat akurasi yang luar biasa sebesar 100% dan kerugian minimal sebesar 0,2292. Selain itu, keberhasilan integrasi *Chatbot* ke dalam aplikasi berbasis situs web menggunakan kerangka atau pustaka *Streamlit* telah memperluas aksesibilitasnya, memungkinkan pengguna untuk mengaksesnya baik secara online maupun lokal. Sebagai alat yang bertujuan untuk mengembangkan pemahaman tentang korosi, *Chatbot* ini mewakili solusi yang berharga. Namun demikian, penting untuk mengakui keterbatasan yang terkait dengan data yang digunakan dalam penelitian ini. Untuk meningkatkan keandalan *Chatbot* dan mendapatkan pemahaman yang lebih komprehensif tentang efektivitas dan efisiensi algoritme yang digunakan, upaya penelitian di masa depan harus memprioritaskan pengujian dan pelatihan pada



kumpulan data yang lebih besar. Pendekatan ini memastikan evolusi berkelanjutan dan penyempurnaan kemampuan *Chatbot*, berkontribusi terhadap penyebaran informasi mengenai korosi yang lebih akurat dan kuat.

DAFTAR PUSTAKA

- [1] C. A. P. Sumarjono, M. Akrom, and G. A. Trisnapradika, "Perbandingan Model Machine Learning Terbaik untuk Memprediksi Kemampuan Penghambatan Korosi oleh Senyawa Benzimidazole," *Techno.Com*, vol. 22, no. 4, Art. no. 4, Nov. 2023, doi: 10.33633/tc.v22i4.9201.
- [2] T. Sutojo, S. Rustad, M. Akrom, A. Syukur, G. F. Shidik, and H. K. Dipojono, "A machine learning approach for corrosion small datasets," *Npj Mater. Degrad.*, vol. 7, no. 1, p. 18, Mar. 2023, doi: 10.1038/s41529-023-00336-7.
- [3] A. Wahyuningsih, Y. Sunarya, and S. Aisyah, "METENAMINA SEBAGAI INHIBITOR KOROSI BAJA KARBON DALAM LINGKUNGAN SESUAI KONDISI PERTAMBANGAN MINYAK BUMI," vol. 1.
- [4] M. Akrom, "INVESTIGATION OF NATURAL EXTRACTS AS GREEN CORROSION INHIBITORS IN STEEL USING DENSITY FUNCTIONAL THEORY," *J. Teori Dan Apl. Fis.*, vol. 10, no. 1, Art. no. 1, Jan. 2022.
- [5] M. Akrom *et al.*, "Artificial Intelligence Berbasis QSPR Dalam Kajian Inhibitor Korosi," *JoMMIT J. Multi Media Dan IT*, vol. 7, no. 1, pp. 015–020, Jul. 2023, doi: 10.46961/jommit.v7i1.721.
- [6] M. Sugeng, F. M. Ismail, and J. P. Utomo, "ANALISIS PERBEDAAN LAJU KOROSI HASIL PENGUJIAN WEIGHT LOSS DAN POLARISASI PADA PIPA DENGAN PENGUJIAN KOROSI STANDAR ASTM G59 DAN ASTM G31," vol. 2, no. 1, Art. no. 1, Mar. 2022.
- [7] G. Haryono and B. Sugiarto, "Ekstrak Bahan Alam sebagai Inhibitor Korosi".
- [8] N. V. Putranto, M. Akrom, and G. A. Trinapradika, "Implementasi Fungsi Polinomial pada Algoritma Gradient Boosting Regressor: Studi Regresi pada Dataset Obat-Obatan Kadaluausa Sebagai Material Antikorosi," *J. Teknol. Dan Manaj. Inform.*, vol. 9, no. 2, Art. no. 2, Dec. 2023, doi: 10.26905/jtmi.v9i2.11192.
- [9] S. Marzorati, L. Verotta, and S. Trasatti, "Green Corrosion Inhibitors from Natural Sources and Biomass Wastes," *Molecules*, vol. 24, no. 1, p. 48, Dec. 2018, doi: 10.3390/molecules24010048.
- [10] Z. M. Hanif, "Pengembangan Aplikasi Whatsapp *Chatbot* Untuk Pelayanan Akademik Di Perguruan Tinggi," Dec. 2021, Accessed: Jan. 09, 2024. [Online]. Available: <https://dspace.uui.ac.id/handle/123456789/37445>
- [11] R. A. Yunmar and I. W. W. Wisesa, "Pengembangan Mobile based Question Answering System dengan Basis Pengetahuan Ontologi," *J. Teknol. Inf. Dan Ilmu Komput.*, vol. 7, no. 4, Art. no. 4, Aug. 2020, doi: 10.25126/jtiik.2020742255.
- [12] H. A. F. Muhyidin and L. Venica, "Pengembangan *Chatbot* untuk Meningkatkan Pengetahuan dan Kesadaran Keamanan Siber Menggunakan Long Short-Term Memory," *J. Inform. Dan Rekayasa Perangkat Lunak*, vol. 5, no. 2, pp. 152–161, Oct. 2023, doi: 10.36499/jinrpl.v5i2.8818.
- [13] E. L. Amalia and D. W. Wibowo, "Rancang Bangun *Chatbot* Untuk Meningkatkan Performa Bisnis," *J. Ilm. Teknol. Inf. Asia*, vol. 13, no. 2, p. 137, Oct. 2019, doi: 10.32815/jitika.v13i2.410.
- [14] N. A. Purwitasari and M. Soleh, "Implementasi Algoritma Artificial Neural Network Dalam Pembuatan *Chatbot* Menggunakan Pendekatan Natural Language Processing," vol. 6.
- [15] K. Nimavat and T. Champaneria, "Chatbots: An overview. Types, Architecture, Tools and Future Possibilities," Oct. 2017.
- [16] A. Elholiqi and A. Musdholifah, "Chatbot in Bahasa Indonesia using NLP to Provide Banking Information," *IJCCS Indones. J. Comput. Cybern. Syst.*, vol. 14, no. 1, Art. no. 1, Jan. 2020, doi: 10.22146/ijccs.41289.
- [17] R. Mahendra and M. Kamayani, "Menerapkan Algoritma Neural Network Pada *Chatbot* Mengenai Pariwisata Di Provinsi Bangka Belitung," *J-SAKTI J. Sains Komput. Dan Inform.*, vol. 7, no. 2, Art. no. 2, Sep. 2023, doi: 10.30645/j-sakti.v7i2.678.
- [18] P. Anki, A. Bustamam, H. S. Al-Ash, and D. Sarwinda, "Intelligent *Chatbot* Adapted from Question and Answer System Using RNN-LSTM Model," *J. Phys. Conf. Ser.*, vol. 1844, no. 1, p. 012001, Mar. 2021, doi: 10.1088/1742-6596/1844/1/012001.
- [19] V. R. Prasetyo, N. Benarkah, and V. J. Chrisintha, "Implementasi Natural Language Processing Dalam Pembuatan *Chatbot* Pada Program Information Technology Universitas Surabaya," *Teknika*, vol. 10, no. 2, Art. no. 2, Jul. 2021, doi: 10.34148/teknika.v10i2.370.
- [20] 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, Jun. 2022, doi: 10.36448/expert.v12i1.2593.
- [21] 1815061012 Hilmi Hermawan, "IMPLEMENTASI LONG SHORT-TERM MEMORY (LSTM) PADA CHATBOT INFORMASI AKADEMIK DI PROGRAM STUDI TEKNIK INFORMATIKA UNIVERSITAS LAMPUNG." Accessed: Jan. 09, 2024. [Online]. Available: <https://digilib.unila.ac.id/65316/>
- [22] A. Silvanie and R. Subekti, "APLIKASI CHATBOT UNTUK FAQ AKADEMIK DI IBK57 DENGAN LSTM DAN PENYEMATAN KATA," *JIKO J. Inform. Dan Komput.*, vol. 5, no. 1, Art. no. 1, Apr. 2022, doi: 10.33387/jiko.v5i1.3703.



- [23] K. A. Nugraha and D. Sebastian, “*Chatbot* Layanan Akademik Menggunakan K-Nearest Neighbor,” *J. Sains Dan Inform.*, vol. 7, no. 1, pp. 11–19, Mar. 2021, doi: 10.34128/jsi.v7i1.285.
- [24] Y. Denny, H. L. H. Spits Warnars, W. Budiharto, A. I. Kistijantoro, Y. Heryadi, and L. Lukas, “Lstm And Simple Rnn Comparison In The Problem Of Sequence To Sequence On Conversation Data Using Bahasa Indonesia,” Sep. 2018, pp. 51–56. doi: 10.1109/INAPR.2018.8627029.
- [25] F. Zakariya, J. Zeniarja, and S. Winarno, “Pengembangan *Chatbot* Kesehatan Mental Menggunakan Algoritma Long Short-Term Memory,” vol. 8, 2024.



©2021. This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-sa/4.0/).