

Implementasi Long Short-Term Memory pada Chatbot Informasi Akademik Teknik Informatika Unila

Puput Budi Wintoro, Hilmi Hermawan*, Mona Arif Muda, Yessi Mulyani

Teknik Informatika, Fakultas Teknik, Universitas Lampung
Bandar Lampung, Indonesia

budi.wintoro@eng.unila.ac.id, hilmihermawan07@gmail.com*, mona.batubara@eng.unila.ac.id, yessi.mulyani@eng.unila.ac.id

Abstract – The times have made technology develop rapidly and rapidly, one of which is artificial intelligence. According to an Accenture survey, it can help organizations with up to 30% operational costs and users enjoy having 24/7 instant access to the answers they order. However, there are many organizations and companies that have not used it to facilitate business, one of which is the Informatics Engineering Study Program, University of Lampung (PSTI Unila). The purpose of this research is to be able to apply the Long Short-Term Memory (LSTM) algorithm to the academic information at PSTI Unila and demonstrate the accuracy of the algorithm. Broadly speaking, this research consists of 4 stages: data collection and exploration, data preparation, building model and training, and testing the model. The model is tested using the user validation method which gets an accuracy of 99%. With these accurate results, this model is quite feasible to be used as a place for students to ask questions about PSTI Unila academic information.

Keywords: Chatbot; Academic Information; Long Short-Term Memory; User Validation.

Abstrak – Perkembangan zaman membuat teknologi berkembang semakin cepat dan pesat, salah satunya yaitu kecerdasan buatan. Menurut survei Accenture, *chatbot* dapat membantu organisasi dengan memotong biaya operasional hingga 30% dan pengguna merasa senang memiliki akses instan 24/7 untuk jawaban yang mereka butuhkan. Namun masih banyak organisasi dan perusahaan yang belum menggunakan *chatbot* untuk mempermudah bisnis, salah satunya yaitu Program Studi Teknik Informatika Universitas Lampung (PSTI Unila). Tujuan penelitian ini yaitu mampu menerapkan algoritma *Long Short-Term Memory (LSTM)* pada *chatbot* informasi akademik di PSTI Unila dan menunjukkan keakuratan dari algoritma tersebut. Secara garis besar, penelitian ini terdiri dari 4 tahap: pengumpulan dan eksplorasi data, persiapan data, pembangunan dan pelatihan model, serta pengujian model. Model diuji dengan metode *user validation* yang mendapatkan akurasi sebesar 99%. Dengan hasil akurasi tersebut, maka model ini cukup layak untuk digunakan sebagai tempat mahasiswa-mahasiswi untuk bertanya seputar informasi akademik PSTI Unila.

Kata Kunci: Chatbot; Informasi Akademik; Long Short-Term Memory; User Validation.

1. PENDAHULUAN

Perkembangan zaman membuat teknologi berkembang semakin cepat dan pesat, salah satunya yaitu kecerdasan buatan. Salah satu pekerjaan yang bisa dilakukan oleh mesin atau komputer seperti manusia adalah melakukan percakapan dengan manusia. Hal ini dilakukan dengan metode *Natural Language Processing (NLP)*. NLP banyak digunakan untuk penerjemah bahasa seperti Google Translate, personal asisten seperti Siri, pemeriksa keakuratan bahasa seperti Grammarly, ataupun *chatbot*.

Chatbot merupakan program atau aplikasi yang dirancang agar dapat melakukan obrolan *online* dengan manusia. *Chatbot* menerima input melalui teks, ucapan-ke-teks, atau pemilihan opsi dan memberikan tanggapan dari serangkaian opsi yang telah ditentukan. Hal ini berbeda dari percakapan biasa, yang memungkinkan pengguna mengobrol dengan perwakilan langsung dengan cara yang

sama. Meskipun dirancang untuk mensimulasikan percakapan manusia, sebenarnya tidak ada manusia di dalam percakapan tersebut [1]. *Chatbot* dapat menghemat waktu dan mengurangi biaya tertentu. Menurut temuan dari survei Accenture, *chatbot* dapat membantu organisasi dengan memotong biaya operasional hingga 30% dan pengguna merasa senang memiliki akses instan 24/7 untuk jawaban yang mereka butuhkan [2].

Seiring waktu, penelitian mengenai *chatbot* terlihat semakin bertambah. Seperti penelitian yang memanfaatkan *chatbot* menggunakan algoritma Bidirectional *Long Short-Term Memory* untuk media interaktif yang bisa tanya-jawab mengenai tema ayat, tafsir, dan juga asbabun nuzul dari sebuah ayat Al-Quran [3]. Selain itu, penelitian yang memanfaatkan untuk peningkatan komunikasi antara warga dan pemerintah yang telah menjadi masalah sejak lama dalam sektor publik [4].



Metode yang digunakan dalam pembuatan *chatbot* juga mulai beragam. Beberapa metode yang banyak digunakan antara lain *Recurrent Neural Network* dan *Long Short-Term Memory (RNN-LSTM)* [5], *Bidirectional Long Short-Term Memory* [4], dan *Natural Language Processing* [6]. Metode *Long Short-Term Memory* memungkinkan dalam melakukan pemrosesan bahasa alami dengan tujuan untuk memudahkan *user* untuk melakukan komunikasi dengan komputer menggunakan bahasa alami. Metode *Long Short-Term Memory* cukup banyak digunakan dalam penelitian dalam pembangunan *chatbot* karena LSTM merupakan algoritma yang dapat menerima input dan mengeluarkan *output* berupa data *sequences* yang merupakan pengembangan dari metode *Recurrent Neural Network (RNN)*.

Di Indonesia, penerapan *chatbot* masing terbilang kurang. Menurut survey yang dilakukan Pancake terhadap UMKM di Indonesia, UMKM yang sudah menerapkan *chatbot* hanya 15% [7]. Dapat disimpulkan bahwa masih banyak organisasi dan perusahaan yang belum menggunakan untuk mempermudah bisnis, salah satunya yaitu Program Studi Teknik Informatika Universitas Lampung. Program Studi Teknik Informatika Universitas Lampung (PSTI UNILA) masih menjalankan kegiatan secara konvensional dalam hal informasi akademik. Mahasiswa terbiasa untuk bertanya secara langsung kepada admin di Gedung Jurusan mengenai pertanyaan seputar akademik. Ketika mahasiswa meminta surat-surat keperluan akademik, mahasiswa biasanya memberikan nomor whatsapp kepada admin, kemudian admin memberikan surat tersebut melalui Whatsapp. Kegiatan yang masih konvensional tersebut tentu membutuhkan waktu yang tidak sedikit, apalagi jika kebutuhannya mendesak. Hal ini tentu sedikit merepotkan apalagi ketika pandemi COVID-19 yang mengharuskan mahasiswa belajar di rumah. Mahasiswa memerlukan jawaban yang cepat dan responsif yang mudah didapatkan dimana saja mahasiswa berada dan kapan saja. Untuk mengatasi masalah tersebut maka diperlukan *chatbot* yang dapat menghasilkan informasi jawaban yang dibutuhkan dengan tingkat akurasi yang baik. Penelitian ini menggunakan algoritma LSTM yang merupakan metode *Deep Learning* yang cocok untuk data *sequences*. Penelitian ini diharapkan mampu memberikan solusi dari beberapa permasalahan tersebut. Penelitian ini diharapkan menghasilkan *chatbot* yang dapat memudahkan pengguna khususnya mahasiswa dan mahasiswi PSTI Unila untuk mendapatkan informasi - informasi akademik.

2. DASAR TEORI

Saat ini ada berbagai penelitian yang membahas *chatbot* dengan metode pendekatan *Natural Language Processing (NLP)*, diantaranya yaitu penelitian yang membahas mengenai *chatbot* untuk pelayanan online. Penelitian ini bertujuan untuk membantu bagian akademik untuk memberikan jawaban dari pertanyaan - pertanyaan yang diajukan oleh mahasiswa. Metode yang digunakan berasal dari kuesioner dari mahasiswa - mahasiswi di Universitas Muhammadiyah Sumatera

Barat. Kemudian diolah dengan *Natural Language Processing*. Hasil pengujian penelitian ini mampu memberikan respon pertanyaan yang diajukan oleh mahasiswa [8].

Kedua yaitu jurnal yang membahas mengenai *chatbot* untuk pelayanan dalam *Coffee Shop*. Penelitian ini yaitu membangun *chatbot* yang mampu melayani pelanggan *Coffee Shop* untuk memberikan informasi diharapkan pelanggan bisa melakukan transaksi pembelian secara otomatis. *Chatbot* pada penelitian ini memanfaatkan sebuah *tools Dialog flow* milik Google [9]. Dasar Teori akan memuat hasil - hasil penelitian sebelumnya dan/atau literatur - literatur maupun referensi yang akan digunakan sebagai dasar analisis penelitian ini.

Neural Network terdiri dari beberapa lapisan unit komputasi yang terhubung yang disebut *neuron*. Jaringan menerima sinyal input dan menghitung *output* melalui rangkaian operasi perhitungan matriks dan transformasi non-linear. Setiap neural network terdiri atas satu input dan satu *output layer*, dan satu atau beberapa *hidden layer*, dimana setiap lapisan terdiri atas beberapa *neuron*. Hubungan di antara *neuron* dari *layer* yang berbeda membawa beban. Pelatihan jaringan mengacu pada penyetelan bobot sedemikian rupa sehingga keluaran jaringan cocok dengan variabel target seakurat mungkin. Pelatihan jaringan saraf melalui penyesuaian bobot koneksi setara dengan tugas memperkirakan model statistik melalui minimalisasi risiko empiris [10].

Deep learning adalah salah satu *machine learning* yang terdiri atas banyak *layer* atau lapisan yang membentuk sebuah tumpukan. Lapisan itu adalah sebuah algoritma ataupun metode yang bertugas melakukan klasifikasi perintah yang diinput pada *layer* input hingga dapat menghasilkan sebuah output. Metode *Deep Learning* yang sedang berkembang di antaranya yaitu *Recurrent Neural Network (RNN)*, *Long Short-Term Memory (LSTM)*, dan *Convolutional Neural Network (CNN)* [11].

Chatbot adalah program atau aplikasi yang dirancang untuk melakukan obrolan online dengan manusia. *Chatbot* menerima input melalui teks, ucapan-ke-teks, atau pemilihan opsi dan memberikan tanggapan dari serangkaian opsi yang telah ditentukan. Ini berbeda dari percakapan biasa, yang memungkinkan pengguna mengobrol dengan perwakilan langsung dengan cara yang sama. Meskipun *chatbot* dirancang untuk mensimulasikan percakapan manusia, sebenarnya tidak ada manusia di dalam percakapan tersebut. *Chatbot* dianggap sebagai bentuk kecerdasan buatan (AI) karena mereka "belajar" dengan menjadi lebih akurat saat mereka memperoleh lebih banyak data [1].

Word Embedding merupakan representasi kata yang dibuat dari teks mentah atau dokumen berdasarkan konteks kalimat. Representasi untuk setiap kata tersebut berupa vektor yang berisi bilangan riil. Salah satu tujuan penggunaan representasi vektor kata yaitu untuk menghitung kemiripan kata dari setiap nilai kata [12].

Case folding yaitu tahapan yang bertujuan untuk mengubah seluruh huruf atau kata yang berada di dalam dataset menjadi huruf kecil dan hanya huruf "a" hingga "z" yang akan diterima. *Case folding* juga akan melakukan

proses menghilangkan karakter lain selain huruf “a” hingga “z” seperti tanda titik, tanda tanya, koma, titik dua, dan sebagainya.

Tokenizing adalah tahapan memotong setiap kata dalam suatu kalimat dalam dataset yang menggunakan spasi sebagai delimiter sehingga dapat menghasilkan token berupa sebuah kata.

Filtering adalah menyaring kata-kata yang tidak penting atau mungkin tidak mempunyai makna yang disebut *stopword*. *Stopword* berisi kata-kata umum yang sering muncul dalam sebuah dokumen dalam jumlah yang cukup banyak tetapi tidak memiliki kaitan dengan suatu tema tertentu.

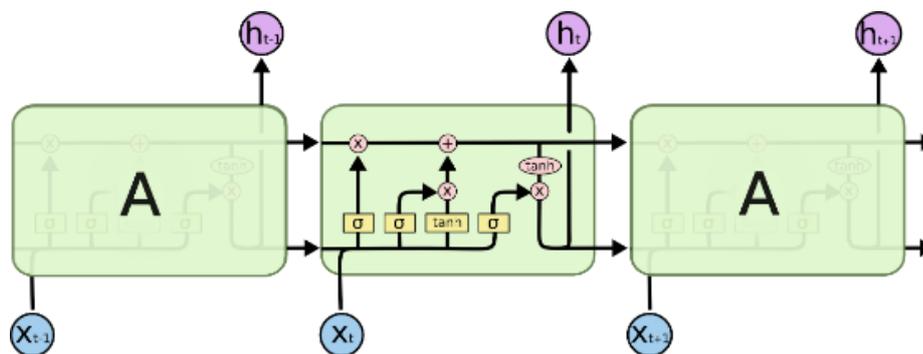
One Hot Encoding merupakan proses mengubah data kategorikal dalam hal ini yaitu teks menjadi angka biner. Algoritma *machine learning* ataupun *deep learning* tidak dapat bekerja secara langsung dengan data kategorikal. Data kategorikal harus dikonversi menjadi angka biner terlebih dahulu. Hal ini berlaku pada penelitian ini karena penelitian ini bekerja dengan jenis klasifikasi urutan teks menggunakan algoritma *deep learning Long Short-Term Memory (LSTM)* [13].

Recurrent Neural Network atau biasa disebut dengan RNN adalah metode Jaringan syaraf tiruan yang berfungsi untuk melakukan proses data *sequential* atau berkesinambungan. Umumnya, dalam membuat suatu

keputusan, manusia akan mempertimbangkan informasi dari sebelumnya atau informasi pada masa lalu. Seperti halnya manusia, informasi dari masa lalu disimpan dengan melakukan perulangan pada arsitekturnya yang akan menjadikan informasi dari masa sebelumnya tetap disimpan. Dalam aturan rantai RNN dapat menyebabkan masalah yang terkait erat dengan menghilangnya gradien. Masalah ini disebut ledakan gradien dan itu terjadi saat mengalikan bobot secara rekursif matriks dengan beberapa entri di atas satu di jalur mundur pelatihan jaringan [14].

Long Short-Term Memory (LSTM) adalah salah satu jenis dari arsitektur RNN atau *Recurrent Neural Network* yang berguna untuk menghindari masalah dependensi jangka panjang. Dalam satu neuron, jaringan pada LSTM mempunyai empat lapisan yang saling berinteraksi, yaitu satu lapisan tanh dan tiga lapisan sigmoid. Fungsi aktivasi sigmoid adalah fungsi yang mengembalikan nilai dalam rentang nol dan satu. Sedangkan fungsi aktivasi tanh mengembalikan nilai dalam rentang negatif satu hingga satu [15].

Komponen utama dari struktur LSTM adalah *forget gate*, *input gate*, *cell state*, dan *output gate*. Misalkan x adalah input sequence dan h menjadi hidden state yang dihasilkan dari LSTM layer, maka hubungan antara x dan h yaitu berdasarkan persamaan sebagai berikut.



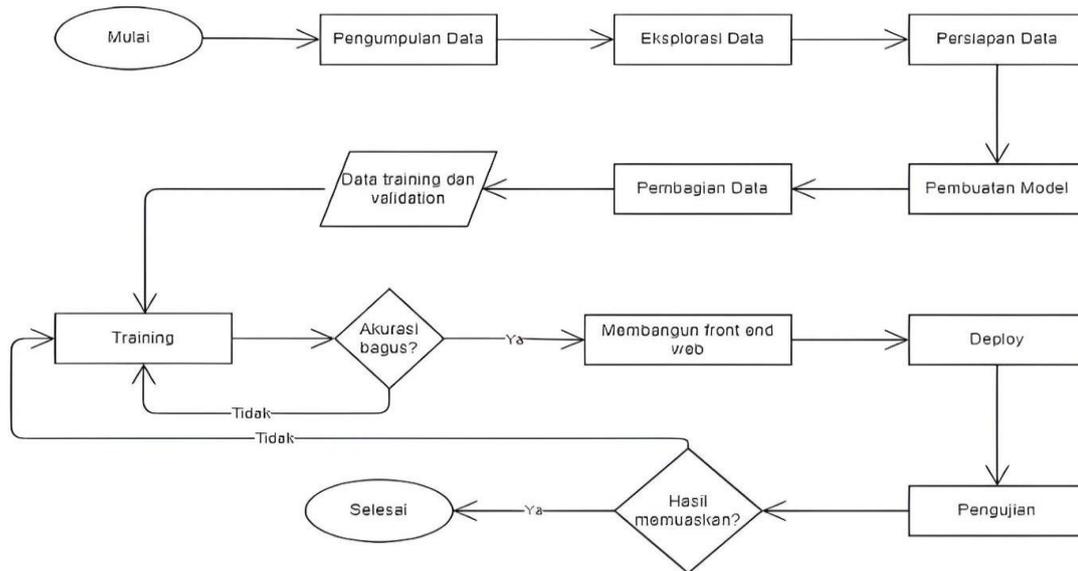
Gambar 1. Jaringan LSTM [15]

Berdasarkan gambar 1, kotak berwarna kuning merupakan *neural network layer* yaitu jaringan saraf yang dipelajari. Anak panah tanpa cabang atau garis penggabungan adalah vektor transfer yaitu garis yang membawa seluruh vektor, dari keluaran satu neuron ke masukan sel lainnya. Lingkaran berwarna merah muda adalah *pointwise operation* yaitu operasi yang dilakukan seperti penambahan vektor. Garis yang mengalami penggabungan adalah concatenate yaitu proses penggabungan yang menandakan rangkuman. Garis bercabang adalah *Copy* yaitu proses penyalinan nilai yang kemudian dibawa ke lokasi yang berbeda. Notasi x_t menyatakan masukan untuk setiap unit pada *time step* t dan nilainya berupa vektor. Notasi h_t menyatakan *hidden state* yang dihasilkan oleh setiap unit pada *time step* t dan nilainya berupa vektor. Notasi C_t menyatakan *cell state* yang dihasilkan oleh setiap unit pada *time step* t dan nilainya berupa vektor [15].

3. METODOLOGI

Penelitian ini menggunakan *Machine Learning Model Development Life Cycle (MDLC)* sebagai panduan dalam mengerjakan proses penelitian. Singkatnya, MDLC yaitu panduan dalam pengembangan model machine learning dari awal hingga selesai. Fase utama dalam MDLC antara lain yaitu *acquire & exploratory data analysis*, *data preparation*, *model creation*, dan *model operation*. Garis besar tahapan penelitian adalah sebagai berikut, Gambar 2.

Tahap pertama adalah tahap pengumpulan data. Tahap ini yaitu tahap pengumpulan data dari sumber aslinya ke dalam repositori data tujuan yang bertujuan untuk melakukan pemrosesan dan analisa data yang ekstensif. Data yang telah dikumpulkan kemudian dieksplorasi dengan cara memahami dan menganalisis data sehingga dapat menemukan pola untuk menghasilkan ide-ide dan pengetahuan yang akan berguna untuk pemodelan.



Gambar 2. Flowchart Tahapan Penelitian

Langkah ini sebagai kualifikasi pertama sebelum melakukan pendalaman dalam mengembangkan model. Proses pada fase ini diantaranya yaitu pengumpulan data, dan memahami jenis data. Data informasi yang berisi syarat-syarat, alur, prosedur dan surat-surat seputar informasi akademik didapat dari Admin Program Studi Teknik Informatika Universitas Lampung. Data ini akan dijadikan sebagai jawaban dalam dataset. Pertanyaan dibuat sendiri berdasarkan jawaban dengan masukan dari kuesioner yang diberikan kepada mahasiswa-mahasiswi PSTI Universitas Lampung.

Persiapan data dapat dikatakan sebagai salah satu langkah tersulit dan terlama dalam proyek machine learning. Tahap ini yaitu tahap melakukan pemrosesan data sehingga data yang mentah diubah menjadi data yang siap digunakan dalam proses training dalam model machine learning. Tahapan yang dilakukan antara lain yaitu transformasi data yang sesuai untuk proses selanjutnya, membersihkan data pertanyaan, tokenisasi dan memberikan indeks pada setiap kata yang telah di tokenisasi, memberikan SOS, EOS pada jawaban dan OUT pada kosakata yang tidak ada dalam vocab, inversi kosakata hasil indexing, mengubah sequence teks menjadi sequence integer, memberikan padding pada sequence, dan mengubah label menjadi vektor biner.

Tahap selanjutnya yaitu pembangunan model. Pertama, data dibagi menjadi data training dan validation. Pada tahap ini, sebagian dari data digunakan untuk menemukan parameter model terbaik dalam machine learning yang membantu meminimalkan kesalahan untuk kumpulan data yang diberikan. Data *validation* digunakan untuk menguji kemampuan model. Kedua langkah ini diulang beberapa epoch untuk meningkatkan akurasi dan kinerja model. Pembagian data training dan validation yaitu 20% *data validation* dan 80% *data training*. Selanjutnya yaitu menentukan metrik yang akan digunakan untuk mengukur kinerja model, menentukan struktur model, kemudian training. Struktur model yang akan digunakan yaitu *Embedding layer*, LSTM, dan *Output layer*. Karena

model ini menggunakan multiclass, maka *loss function* menggunakan *categorical_crossentropy* dengan fungsi optimasi "Adam". Model ini akan menggunakan matriks evaluasi yaitu "*accuracy*".

Sebelum model yang dibuat diuji kepada pengguna, model harus di *deploy* dan hosting terlebih dahulu agar dapat diakses secara online oleh pengguna. Pada penelitian ini, model di *deploy* ke dalam *website* menggunakan bahasa pemrograman python dan *framework flask*. Untuk antarmuka pengguna, digunakan HTML, Javascript, dan CSS. Web yang dibangun hanya terdiri dari satu halaman yang berisi *bubble*. Ketika *bubble* tersebut di klik, maka akan tampil menu *chatting*. Pada menu ini pengguna dapat bertanya mengenai informasi akademik.

Tahap selanjutnya yaitu Pengujian. Pengujian dilakukan dengan metode user validation. User validation merupakan metode validasi dimana user akan mencoba *chatbot* yang telah dibangun secara langsung yang kemudian user dapat langsung menilai apakah hasil yang diberikan oleh *chatbot* sesuai dengan kebutuhan user. User akan diberikan link uji coba model serta daftar topik pertanyaan dan jawaban yang seharusnya. Pengujian diberikan kepada 20 user yaitu mahasiswa dan mahasiswi Program Studi Teknik Informatika Universitas Lampung dari angkatan 2018, 2019, 2020, dan 2021 dengan masing - masing user mendapat 5 topik pertanyaan dan jawaban serta user dapat memasukkan pertanyaan di luar dari 5 topik pertanyaan tersebut. Topik didapat dari dataset dengan cara diacak menggunakan bahasa pemrograman python. Program Python ini menghasilkan 20 file data uji yang tentunya pertanyaan yang diberikan kepada user tidak semuanya sama. Pengguna akan melakukan uji coba model dengan memasukkan pertanyaan sesuai dengan topik yang diberikan.

Hasil dari pengujian yaitu akurasi. Akurasi dihitung dengan cara melakukan pembagian jumlah jawaban benar dengan jumlah pertanyaan lalu dikalikan dengan 100%. Hasil pengujian merupakan hasil akhir apakah model

bekerja secara baik dan layak untuk digunakan. Jika masih banyak penyimpangan, latih ulang model dengan data dan *hyperparameter* yang diperbarui.

4. HASIL DAN PEMBAHASAN

Data yang didapat terdapat 29 buah informasi yang akan dijadikan jawaban. Data ini merupakan data yang sering dicari dan ditanyakan mahasiswa kepada admin. Data didapat dari admin PSTI Unila yang dikirim berupa file doc atau docx. File yang didapat adalah file surat penghapusan mata kuliah & sunting mata kuliah di kartu hasil studi, berita acara seminar kp, surat rekomendasi msib, daftar persyaratan ujian komprehensif, surat kp ke wadek 1, surat izin survey penelitian mahasiswa, surat pengantar aktif kuliah, daftar nilai dan berita acara seminar komprehensif, persyaratan seminar proposal, lembar pengesahan, daftar nilai dan berita acara seminar hasil, surat perbaikan mata kuliah dan nilai, daftar nilai dan berita acara seminar proposal, surat pengantar perpanjangan cuti akademik, surat rekomendasi pembimbing KP, persyaratan seminar komprehensif, daftar persyaratan seminar proposal, persyaratan seminar hasil, daftar persyaratan bebas administrasi, surat keterlambatan spp, surat berkelakuan baik, permohonan tanda tangan transkrip, alur wisuda, syarat syarat bebas administrasi, surat bebas ukt, contoh surat peminjaman, dan kartu kendali bimbingan.

Setelah didapat data informasi akademik, selanjutnya membuat data pertanyaan berdasarkan data jawaban yang didapat. Pertanyaan dibuat sendiri berdasarkan jawaban dengan masukan dari kuesioner yang diberikan kepada mahasiswa-mahasiswi Program Studi Teknik Informatika Universitas Lampung.

Data yang didapat terdiri dari beberapa jenis yaitu data berupa syarat-syarat, alur, prosedur, surat, gambar dan pdf. Syarat-syarat, alur, dan prosedur dikategorikan sebagai jenis non surat dan file berupa surat, gambar dan pdf dikategorikan sebagai file jenis surat. Informasi kategori non surat dijadikan sebuah kalimat, sementara informasi yang berupa masuk kategori surat dijadikan sebuah link yang di *upload* di Google Drive.

Setelah mendapatkan data dan melakukan pemahaman terhadap data, selanjutnya adalah tahap persiapan data. Data yang dijadikan sebagai jawaban dan data pertanyaan selanjutnya diubah menjadi satu tabel yang terdiri dari kolom question dan answer dalam file csv yang kemudian akan diproses dengan python. Jumlah sampel pada dataset sebanyak 449 dengan informasi sebanyak 29 data. Data yang telah diubah menjadi format csv kemudian diubah menjadi format dataframe dalam python menggunakan library pandas. Untuk dapat mengolah data pertanyaan dan jawaban agar lebih mudah, maka jawaban dan pertanyaan dipisah dan diubah menjadi sebuah list. Berikut merupakan *source code* mengubah data csv menjadi dataframe serta mengubah dataframe menjadi sebuah *list question* dan *answer*.

Data pertanyaan yang telah diubah menjadi list kemudian dibersihkan. Pembersihan data dilakukan dengan menggunakan library nltk. Pembersihan yang

dilakukan diantaranya yaitu pertanyaan dijadikan menjadi huruf kecil, simbol - simbol yang tidak penting dihapus, kemudian menggunakan *stopwords* untuk membuang kata-kata yang kurang penting atau dianggap tidak memiliki makna. *Cleaning* data digunakan agar semua pertanyaan berformat sama dan tidak memiliki simbol untuk menurunkan jumlah *vocab* agar proses training model lebih cepat dan hasil model dapat lebih baik.

Pada tahap tokenisasi, dari setiap *sequence* dilakukan pemisahan setiap kata, rangkaian kata, angka, dan rangkaian angka yang memiliki makna tertentu. Setelah dilakukan tokenisasi, setiap token unik akan diberikan index untuk masuk ke dalam vocabulary atau kosakata. Jumlah token unik yang telah diberi index yang kemudian akan dimasukkan ke dalam *vocab* berjumlah 376. Berikut merupakan *source code* untuk memberikan indeks pada token.

SOS adalah singkatan dari *Start of Sequence*, EOS adalah singkatan dari *End of Sequence*, OUT adalah singkatan dari *Out of Vocabulary*. SOS diberikan pada setiap awal jawaban atau label sedangkan EOS diberikan pada setiap akhir jawaban atau label. Hal ini untuk menandakan awal dan akhir dari sebuah jawaban. OUT digunakan untuk menandai kata yang tidak terdapat dalam vocabulary. SOS, OUT, dan EOS kemudian dimasukkan dalam *vocabulary*.

Vocab yang telah dibuat sebelumnya bertipe data dictionary dimana key dictionary berupa kata dan *value dictionary* berupa angka *index*. Hal ini dilakukan untuk menerjemahkan hasil prediksi output yang berupa index kedalam sebuah kata. *Key dictionary* yang awalnya kata diubah menjadi angka *index* dan *value dictionary* yang awalnya *index* diubah menjadi kata.

Setelah membuat *dictionary* kosakata yang berisi kata dan *index*, selanjutnya adalah membuat *sequence integer*. Proses ini yaitu mengubah *sequence* yang awalnya berupa kata menjadi sebuah *integer*. Proses pengubahan ini memanfaatkan *dictionary* yang telah dibuat sebelumnya.

Padding adalah proses menjadikan sebuah *sequence* yang awalnya memiliki panjang yang berbeda menjadi *sequence* yang memiliki panjang yang sama. *Padding* perlu dilakukan karena masukan di *embedding layer* harus memiliki panjang yang sama. Pada penelitian ini, panjang terpanjang dari *sequence* adalah 261. Dengan demikian, *max_length* atau panjang maksimal *sequence* adalah 261 kata.

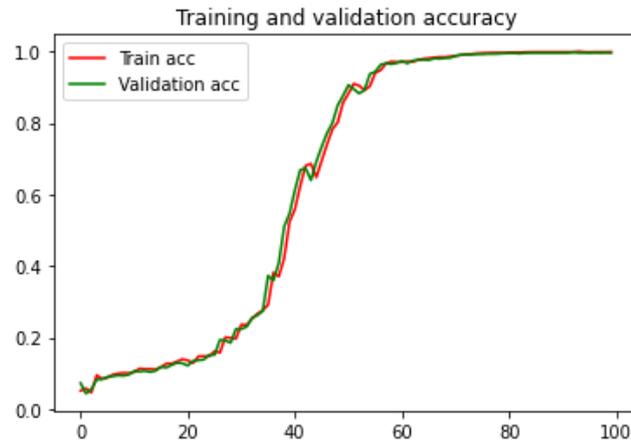
One Hot Encoding adalah representasi dari variabel label sebagai vektor biner. Setiap nilai integer direpresentasikan sebagai vektor biner yang semuanya bernilai nol kecuali indeks integer yang ditandai dengan 1. Dalam penelitian ini, *One Hot Encoding* data ini yaitu 3 dimensi berukuran 449, 261, 376. Artinya, dimensi pertama yaitu 449 yang menunjukkan bahwa terdapat 449 sampel data. Dimensi kedua yaitu 261 yang menunjukkan bahwa setiap sampel data memiliki 261 kata. Dimensi ketiga yaitu 376 yang menunjukkan bahwa setiap kata dalam dimensi kedua memiliki 376 kemungkinan kosakata.

Struktur model yang dipakai pada penelitian ini yaitu lapisan pertama: *embedding layer* yang menggunakan vektor sepanjang 128 yang digunakan untuk merepresentasikan

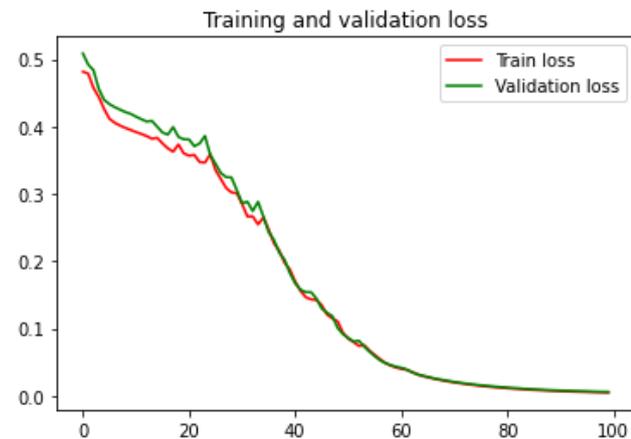
setiap kata. Lapisan kedua yaitu LSTM *layer* dengan 256-unit perceptron. Lapisan terakhir yaitu Dense *Output layer* yang terdiri dari 272-unit perceptron dan menggunakan fungsi softmax.

Penelitian ini merupakan klasifikasi dengan *multiclass classification* yang menggunakan *loss function* yaitu *categorical_crossentropy* (keras) dengan menggunakan fungsi optimasi “Adam” dengan metrik evaluasi yaitu

“accuracy”. Pembagian data pada dilakukan dengan 20% *data validation* dan 80% data training. Kemudian dilakukan *training data* dengan menggunakan struktur model yang telah dibuat sebelumnya dengan 100 *epochs*. Hasil dari *training* yaitu 99,88% *accuracy* dan 99,61% *validation accuracy* serta 0,51% *loss* dan 0,61% *validation loss*. Grafik hasil *accuracy* dan *loss* serta validasi *accuracy* dan validasi *loss* pada setiap epoch dapat dilihat pada Gambar 3, Gambar 4.



Gambar 3. Grafik Akurasi Training dan validation

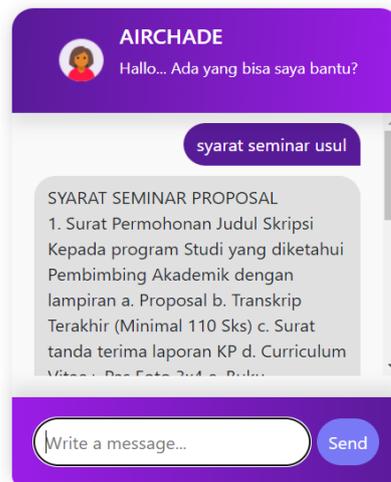


Gambar 4. Grafik Loss Training dan Validation

Selanjutnya yaitu tahap *Deployment*, *Deployment* disini bertujuan agar model *chatbot* yang dibuat dapat digunakan oleh user dengan tampilan layaknya *chatbot* yang digunakan untuk pengujian model. *Deployment* dilakukan dengan menggunakan bahasa pemrograman Python dan *flask framework*, serta halaman antarmuka yang dibangun menggunakan HTML, CSS, dan Javascript, yang kemudian dilakukan *hosting* dengan menggunakan heroku. Pada website akan menampilkan *bubble* yang jika kita memasukkan pertanyaan seputar informasi akademik, maka sistem akan menjawab seperti pada Gambar 5.

Tahapan terakhir yaitu tahap pengujian. Metode yang digunakan yaitu *user validation*. Metode ini merupakan metode validasi dimana *user* akan mencoba *chatbot* yang

telah dibangun secara langsung yang kemudian user dapat langsung menilai apakah hasil yang diberikan oleh *chatbot* sesuai dengan kebutuhan *user*. *User* akan diberikan link uji coba model serta daftar topik pertanyaan dan jawaban yang seharusnya. Pengujian diberikan kepada 20 user yaitu mahasiswa/mahasiswi Program Studi Teknik Informatika Universitas Lampung (PSTI Unila) dari angkatan 2018, 2019, 2020, dan 2021 dengan masing-masing *user* mendapat 5 topik pertanyaan dan jawaban serta *user* dapat memasukkan pertanyaan di luar dari 5 topik pertanyaan tersebut. Topik pertanyaan didapat dari dataset dengan cara diacak menggunakan bahasa pemrograman Python.

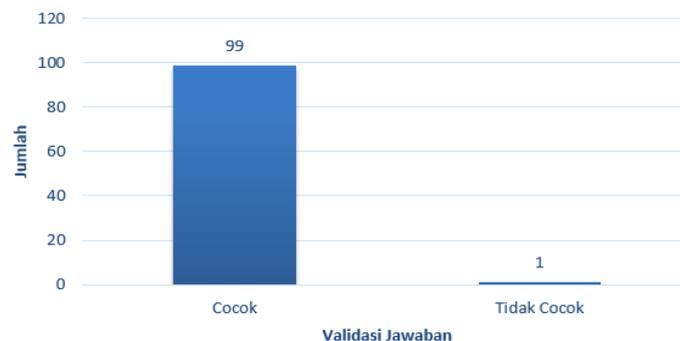


Gambar 5. Halaman Percakapan setelah Bertanya

User akan diberikan file yang telah dibuat beserta panduan pengujiannya. User akan membuka web *chatbot* yang telah di *deploy*, kemudian *user* memasukkan pertanyaan yang telah diberikan. Jawaban yang diberikan pada file akan dicocokkan dengan respon jawaban dari *chatbot*. Jika cocok, maka *user* dapat memasukkan kata “cocok” pada kolom cocok atau tidak cocok. Jika tidak cocok, maka *user* dapat memasukkan kata “tidak cocok” pada kolom cocok atau tidak cocok.

Dengan menggunakan 5 data pada masing - masing 20 *user* yang menghasilkan total 100 data uji, dapat dilihat pada Gambar 6 didapatkan hasil jawaban benar sebanyak 99 dan salah sebanyak 1 pertanyaan. Akurasi didapatkan dengan perhitungan berikut:

$$Akurasi = \frac{99}{100} \times 100\% = 99\%$$



Gambar 6. Grafik Hasil Pengujian

Masing-masing *user* memasukkan pertanyaan di luar dari 5 topik pertanyaan yang diberikan. Total pertanyaan bebas yang dimasukkan *user* yaitu sebanyak 20 pertanyaan. Sebanyak 12 pertanyaan dari 20 pertanyaan sudah ada dalam database dan berhasil memberikan jawaban dengan benar. Sebanyak 8 pertanyaan dari 20 pertanyaan tidak ada dalam dataset. Pertanyaan tersebut dapat dijadikan untuk masukkan dalam penelitian selanjutnya agar dapat memenuhi kebutuhan pengguna.

5. KESIMPULAN

Berdasarkan hasil penelitian ini, dapat diperoleh kesimpulan bahwa algoritma *Long-Short Term Memory (LSTM)* berhasil diimplementasikan pada *chatbot* informasi akademik di Program Studi Teknik Informatika Universitas Lampung dengan hasil training yang cukup

baik dan model dapat berjalan dengan cukup baik pada tahap pengujian. Dengan demikian, model ini cukup layak untuk digunakan sebagai tempat mahasiswa-mahasiswi untuk bertanya seputar informasi akademik PSTI Unila.

Berdasarkan pengujian yang dilakukan, masih ada beberapa informasi yang dibutuhkan oleh mahasiswa namun belum ada dalam dataset, oleh karena itu dataset informasi akademik perlu ditambahkan lagi seperti *form* pengajuan kuliah praktik, tata cara pengajuan berhenti studi, surat bebas perpustakaan, informasi lainnya agar *chatbot* dapat lebih memenuhi kebutuhan mahasiswa-mahasiswi PSTI Unila. Penelitian ini dapat dikembangkan dengan melakukan *deployment* ke dalam website PSTI Unila atau aplikasi Android untuk dapat diakses dengan mudah oleh mahasiswa-mahasiswi PSTI Unila. Dapat ditambahkan fitur chat dengan admin mengenai masalah yang tidak bisa dipecahkan oleh *chatbot*.

Daftar Pustaka

- [1] R. Stauffer, "Understanding & Implementing *Chatbots* at Your Agency," 20 April 2021. [Online]. Available: <https://blog.hawksoft.com/chatbots>. [Diakses 22 February 2022].
- [2] Accenture, "*Chatbots* are Here to Stay," 2018. [Online]. Available: https://www.accenture.com/_acnmedia/pdf-77/accenture-research-conversational-ai-platforms.pdf. [Diakses 22 February 2022].
- [3] B. Kurniawan, "*Chatbot* Eksplorasi Ayat Al-Quran Menggunakan Algoritma Bidirectional Long Short-Term Memory (Bi-LSTM) dan Fuzzy String Matching," Universitas Sumatera Utara, Medan, 2021.
- [4] I. M. Untari, "*Chatbots* and Government Communications in Covid-19 Pandemic," Jurnal Komunikasi Indonesia, vol. IX, 2020.
- [5] A. B. H. S. A.-A. D. S. P Anki, "Intelligent *Chatbot* Adapted from Question-and-Answer System Using RNN-LSTM Model," Journal of Physics: Conference Series, 2021.
- [6] V. R. Prasetyo, N. Benarkah dan V. J. Chrisintha, "Implementasi Natural Language Processing Dalam Pembuatan Chatbot Pada Program Information Technology Universitas Surabaya," TEKNIKA, 2021.
- [7] H. A. Amalia, "Berita Satu," 1 Oktober 2021. [Online]. Available: <https://www.beritasatu.com/archive/835281/survei-pancake-baru-15-pelaku-umkm-indonesia-pakai-chatbot>. [Diakses 23 Mei 2022].
- [8] Mulyono, "Identifikasi *Chatbot* dalam Meningkatkan Pelayanan Online," Jurnal Informatika Ekonomi Bisnis, 2021.
- [9] A. Y. Chandra, D. Kurniawan dan R. Musa, "Perancangan *Chatbot* Menggunakan Dialogflow Natural Language," Jurnal Media Informatika Budidarma, 2020.
- [10] W. K. & L. S. Härdle, Applied multivariate statistical analysis (4th ed.), Cham: Springer, 2015.
- [11] G. A. Royani Darma Nurfiti, "Implementasi Deep Learning Berbasis Tensorflow untuk Pengenalan Sidik Jari," Emitor: Jurnal Teknik Elektro, vol. 18, 2018.
- [12] A. Leeuwenberg, M. Vela, J. Dehdari dan J. Genabith, "A Minimally Supervised Approach for Synonym Extraction with Word Embeddings. The Prague Bulletin of Mathematical Linguistics," pp. 111-142, 2016.
- [13] D. P. R. R. F. M. I. S. B. A. Winda Kurnia Sari, "Klasifikasi Teks Multilabel pada Artikel Berita Menggunakan Long ShortTerm Memory dengan Word2Vec," Jurnal RESTI, vol. 4, 2020.
- [14] I. B. Y. & C. A. Goodfellow, "Deep learning," Cambridge: MIT Press., 2016.
- [15] C. Olah, "Understanding LSTM Networks," 27 Agustus 2015. [Online]. Available: <http://colah.github.io/posts/2015-08-Understanding-LSTMs/>. [Diakses 22 February 2022].