

**IMPLEMENTASI *LONG SHORT-TERM MEMORY* (LSTM) PADA
CHATBOT INFORMASI AKADEMIK DI PROGRAM STUDI TEKNIK
INFORMATIKA UNIVERSITAS LAMPUNG**

(Skripsi)

Oleh

HILMI HERMAWAN



**FAKULTAS TEKNIK
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG
2022**

ABSTRAK

IMPLEMENTASI *LONG SHORT-TERM MEMORY* (LSTM) PADA CHATBOT INFORMASI AKADEMIK DI PROGRAM STUDI TEKNIK INFORMATIKA UNIVERSITAS LAMPUNG

Oleh

HILMI HERMAWAN

Teknologi kecerdasan buatan saat ini berkembang dengan sangat pesat. Salah satu aplikasi kecerdasan buatan yang dapat membantu user dalam mencari informasi adalah *chatbot*. 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 dari penelitian ini adalah mampu menerapkan algoritma *Long Short-Term Memory* (LSTM) pada *chatbot* informasi akademik di PSTI Unila dan menunjukkan keakuratan dari algoritma tersebut. Penelitian ini terdiri dari 4 tahap: pengumpulan dan eksplorasi data, persiapan data, pembangunan dan pelatihan model, serta pengujian model. Model telah 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*.

ABSTRACT

IMPLEMENTASI LONG SHORT-TERM MEMORY (LSTM) PADA CHATBOT INFORMASI AKADEMIK DI PROGRAM STUDI TEKNIK INFORMATIKA UNIVERSITAS LAMPUNG

By

HILMI HERMAWAN

Artificial intelligence technology is currently developing very rapidly. One of the artificial intelligence applications that can assist users in finding information is a chatbot. According to an Accenture survey, chatbots can help organizations cut operational costs by up to 30% and users enjoy having 24/7 instant access to the answers they need. However, there are still many organizations and companies that have not used chatbots 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 chatbot at PSTI Unila and demonstrate the accuracy of the algorithm. This research consists of 4 stages: data collection and exploration, data preparation, model development and training, and model testing. The model has been tested using the user validation method which has an accuracy of 99%. With these accuracy 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.

**IMPLEMENTASI *LONG SHORT-TERM MEMORY* (LSTM) PADA
CHATBOT INFORMASI AKADEMIK DI PROGRAM STUDI TEKNIK
INFORMATIKA UNIVERSITAS LAMPUNG**

Oleh

HILMI HERMAWAN

Skripsi

**Sebagai Salah Satu Syarat Untuk Mencapai Gelar
SARJANA TEKNIK**

Pada

**Program Studi Teknik Informatika
Jurusan Teknik Elektro
Fakultas Teknik Universitas Lampung**



**FAKULTAS TEKNIK
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG
2022**

Judul

: IMPLEMENTASI *LONG SHORT-TERM
MEMORY (LSTM)* PADA *CHATBOT
INFORMASI AKADEMIK* DI PROGRAM
STUDI TEKNIK INFORMATIKA
UNIVERSITAS LAMPUNG

Nama Mahasiswa

: **Hilmi Hermawan**

Nomor Pokok Mahasiswa

: **1815061012**

Program Studi

: **Teknik Informatika**

Jurusan

: **Teknik Elektro**

Fakultas

: **Teknik**



1. Komisi Pembimbing

Mona Arif Muda, S.T., M.T.
NIP. 197111122000031002

Puput Budi Wintoro, S. Kom, M.T.I
NIP. 198410312019031004

2. Mengetahui

Ketua Jurusan
Teknik Elektro

Ketua Program Studi
Teknik Informatika

A handwritten signature in black ink, appearing to be 'Heflinawati'.

Heflinawati, S.T., M.T.
NIP. 197103141999032001

A handwritten signature in black ink, appearing to be 'Mona Arif Muda'.

Mona Arif Muda, S.T., M.T.
NIP. 197111122000031002

MENGESAHKAN

1. Tim Penguji

Ketua : **Mona Arif Muda, S.T., M.T.**

Sekretaris : **Puput Budi Wintoro, S. Kom, M.T.I**

Penguji : **Yessi Mulyani, S.T., M.T.**

2. Dekan Fakultas Teknik



Dr. Eng. Helmy Fitriawan, S.T., M.Sc.
NIP. 197509282001121002

Tanggal Lulus Ujian Skripsi: 22 Juli 2022

SURAT PERNYATAAN

Saya yang bertandatangan dibawah ini, menyatakan bahwa skripsi saya yang berjudul "IMPLEMENTASI *LONG SHORT-TERM MEMORY* (LSTM) PADA *CHATBOT* INFORMASI AKADEMIK DI PROGRAM STUDI TEKNIK INFORMATIKA UNIVERSITAS LAMPUNG" dengan ini menyatakan bahwa skripsi saya dibuat oleh saya sendiri. Semua hasil yang tertuang dalam skripsi ini telah mengikuti kaidah penulisan karya ilmiah Universitas Lampung. Apabila di kemudian hari terbukti bahwa skripsi ini merupakan salinan atau dibuat oleh orang lain, maka saya bersedia menerima sanksi sesuai dengan ketentuan hukum atau akademik yang berlaku.

Bandar Lampung, 9 Agustus 2022

Pembuat Pernyataan,



Hilmi Hermawan

NPM 1815061012

RIWAYAT HIDUP



Penulis dilahirkan di Serang, Banten pada tanggal 22 Juli 1999, sebagai anak pertama dari tiga bersaudara, Bapak Hendri Budiman dan Ibu Enok Ita. Penulis menyelesaikan pendidikan Sekolah Dasar diselesaikan di SDN 1 Kadubeureum, Serang pada tahun 2011, Sekolah Menengah Pertama di MTSN Model Padarincang, Serang pada tahun 2014, dan Sekolah Menengah Atas (SMA) di SMAN 1 Ciomas, Serang pada tahun 2017. Tahun 2018, penulis terdaftar sebagai mahasiswa Program Studi Teknik Informatika Universitas Lampung melalui jalur SBMPTN. Selama menjadi mahasiswa, penulis pernah menjadi asisten Laboratorium Teknik Komputer dan aktif di Organisasi Himpunan Mahasiswa Teknik Elektro (HIMATRO) Unila sebagai anggota Departemen Pendidikan dan Pengembangan Diri serta di Unit Kegiatan Mahasiswa (UKM) Sains dan Teknologi Unila sebagai kepala Departemen Dana dan Usaha. Pada tahun 2021, penulis mendapat kesempatan untuk mengikuti Bangkit Academy dengan bidang Machine Learning. Pada tahun yang sama, penulis mengikuti Studi Independen di Dicoding dengan bidang Pengembang Machine Learning dan Front End. Pada tahun 2022, penulis mendapatkan kesempatan untuk magang di PT. Media Kreasi Abadi yang bertempat di Balikpapan, Kalimantan Timur sebagai Data Scientist.

PERSEMBAHAN



Skripsi ini kupersembahkan untuk orang yang menjadi motivasi terbesar dalam segala kegiatan dalam hidupku, abah dan mama. Karya kecil ini kupersembahkan kepada mu sebagai tanda bakti, hormat, dan rasa terima kasih atas segala dukungan selama hidupku.

Mama, terimakasih telah melahirkanku, membesarkanku, dan mendidikku dengan penuh ketulusan dan kasih sayang. Terimakasih atas semua doa dan nasihat yang selalu menjagaku dalam perjalanan menuju kesuksesan.

Abah, terimakasih atas perjuangan tanpa lelah mengasihiku yang tiada terhingga, memberikanku kehidupan yang layak, memberikan dukungan atas segala keinginanku, dan mendidikku tanpa rasa lelah.

Tulisan ini adalah bukti bahwa kalian telah berhasil menjadi mama dan abah yang mampu mendidik anakmu dengan baik.

Serta, almamater yang selalu saya banggakan,
UNIVERSITAS LAMPUNG.

SANWACANA

Puji syukur penulis ucapkan kehadiran Tuhan Yang Maha Esa, karena atas rahmat dan hidayah-Nya skripsi ini dapat diselesaikan.

Skripsi dengan judul “*Implementasi Long Short-Term Memory (LSTM) Pada Chatbot Informasi Akademik Di Program Studi Teknik Informatika Universitas Lampung*” adalah salah satu syarat untuk memperoleh gelar Sarjana Teknik di Universitas Lampung.

Dalam kesempatan ini penulis mengucapkan terima kasih kepada:

1. Dr. Eng. Helmy Fitriawan, S.T., M.Sc. selaku Dekan Fakultas Teknik Universitas Lampung.
2. Ibu Herlinawati, S.T.,M.T. selaku Ketua Jurusan Teknik Elektro Universitas Lampung.
3. Bapak Mona Arif Muda, S.T., M.T. selaku Ketua Program Studi Teknik Informatika Jurusan Teknik Elektro Universitas Lampung sekaligus Pembimbing Utama yang telah bersedia meluangkan waktu untuk memberikan arahan dan bimbingan hingga skripsi ini selesai.
4. Bapak Puput Budi Wintoro, S. Kom, M.T.I selaku Pembimbing Kedua yang telah bersedia meluangkan waktu untuk memberikan arahan dan bimbingan hingga skripsi ini selesai.
5. Bapak Wahyu Eko Sulistiono, S.T., M.Sc, selaku Pembimbing Akademik yang telah bersedia meluangkan waktu untuk memberikan pengarahan dan bimbingan selama menempuh pendidikan Program Studi Teknik Informatika Universitas Lampung.
6. Seluruh Dosen Program Studi Teknik Informatika yang telah membagikan banyak ilmunya kepada penulis.
7. Mbak Rika yang telah membantu penulis dalam segala urusan administrasi serta membantu dalam pencarian data untuk penelitian ini.

8. Adikku, Putri Mastuffah dan Fauji Alfarizi, serta seluruh keluarga besar yang selalu mendukungku serta telah menjadi keluarga yang mampu membuatku tumbuh dan berkembang dengan baik hingga sekarang.
9. Ahmad Arbain, Indria Agustina, Sawiti Fina Kartika, Uji Khaidah Resti, dan Rizki Ramadan yang telah banyak membantu selama proses perkuliahan di Program Studi Teknik Informatika dan telah menjadi teman, sahabat, sekaligus keluarga yang paling berharga selama di Lampung.
10. Seluruh teman teman Teknik Elektro dan Teknik Informatika Angkatan 2018 atas dukungan yang telah diberikan selama menempuh studi di Program Studi Teknik Informatika Universitas Lampung.
11. Anida Susanti, Yulan Yunia, Azizah, Ardianti, Siti Aisyah, Intan Purnamasari, Ahmad Rizky Wahyuda, Munir, dan Sofian yang telah menjadi teman dan sahabat yang selalu memberikan dukungan dan semangat.

Semoga Allah SWT membalas segala bentuk kebaikan hati dan jasa yang telah kalian berikan kepada saya. Saya menyadari meskipun skripsi ini sudah disusun dengan sebaik mungkin, skripsi ini masih terdapat kekurangan dan masih jauh dari kata sempurna, namun saya sangat berharap melalui skripsi ini akan memberikan manfaat bagi siapapun yang membacanya dan bagi penulis dalam mengembangkan dan mengamalkan ilmu pengetahuan yang telah ditempuh selama ini.

Bandar Lampung, 9 Agustus 2022

Penulis,

Hilmi Hermawan

NPM 1815061012

DAFTAR ISI

	Halaman
DAFTAR ISI.....	iv
DAFTAR GAMBAR.....	vi
DAFTAR TABEL	vii
BAB I.....	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah	3
1.3. Tujuan Penelitian.....	3
1.4. Manfaat Penelitian.....	4
1.5. Batasan Masalah.....	4
1.6. Sistematika Penulisan Skripsi	4
BAB II	7
2.1. Penelitian Terkait	7
2.2. Kecerdasan Buatan	8
2.3. Neural Network	9
2.4. Deep Learning	10
2.5. Chatbot	11
2.6. Word Embedding	13
2.7. Reccurent Neural Network.....	13
2.8. Long Short-Term Memory	15
2.9. Dense Layer.....	25
2.10. Machine Learning Development Life Cycle	25
2.11. Text Mining	27
2.12. Indexing word.....	28
2.13. Padding	28
2.14. One Hot Encoding	29
2.15. Optimizers.....	30

2.16.	User Validation	30
BAB III		31
3.1.	Waktu dan Tempat Penelitian	31
3.2.	Alat dan Bahan	31
3.3.	Tahapan Penelitian	32
3.3.1.	Pengumpulan dan Eksplorasi Data	33
3.3.2.	Persiapan Data	34
3.3.3.	Pembangunan dan Pelatihan Model.....	34
3.3.4.	Deployment.....	35
3.3.5.	Pengujian Model.....	35
BAB IV		36
4.1.	Pengumpulan dan Eksplorasi Data	36
4.1.1.	Pengumpulan Data	36
4.1.2.	Eksplorasi Data	37
4.2.	Persiapan Data	43
4.2.1.	Transformasi Data.....	43
4.2.2.	Cleaning Data	45
4.2.3.	Tokenisasi dan Indexing Word Vocabulary	47
4.2.4.	Memberikan SOS, OUT, dan EOS	48
4.2.5.	Inversi Vocab.....	49
4.2.6.	Sequence Integer.....	50
4.2.7.	Padding	51
4.2.8.	One Hot Encoding	52
4.3.	Pembangunan dan Pelatihan Model	53
4.4.	Deployment	59
4.5.	Pengujian Model.....	61
BAB V		64
5.1.	Kesimpulan.....	64
5.2.	Saran	64
DAFTAR PUSTAKA		66

DAFTAR GAMBAR

Gambar 2.1 Struktur RNN [22].....	14
Gambar 2.2 Layer dalam RNN	15
Gambar 2.3 Jaringan LSTM.....	16
Gambar 2.4 Forget Gate <i>LSTM</i>	17
Gambar 2.5 Input Gate <i>LSTM</i>	17
Gambar 2.6 Cell State <i>LSTM</i>	18
Gambar 2.7 Output Gate <i>LSTM</i>	18
Gambar 2.8 Alur MDLC [26]	25
Gambar 3.1 Flowchart Tahapan Penelitian.....	33
Gambar 4.1 Dataframe dataset	45
Gambar 4.2 <i>Source code</i> transformasi data.....	45
Gambar 4.3 <i>Source code data cleaning</i>	46
Gambar 4.4 Perbandingan Data sebelum dibersihkan dan setelah dibersihkan.....	46
Gambar 4.5 <i>Source code</i> tokenisasi dan pemberian indeks	47
Gambar 4.6 <i>Source code</i> pemberian SOS, OUT, dan EOS	48
Gambar 4.7 Contoh Pemberian SOS dan EOS	49
Gambar 4.8 <i>Source code</i> inversi <i>vocab</i>	49
Gambar 4.9 Perbandingan <i>vocab</i> sebelum dan sesudah di inversi.....	50
Gambar 4.10 <i>Source code</i> sequence integer	50
Gambar 4.11 <i>Source code</i> memberikan padding	51
Gambar 4.12 Sequence sebelum diberikan padding	52
Gambar 4.13 Sequence setelah diberikan padding	52
Gambar 4.14 <i>Source code One Hot Encoding</i>	52
Gambar 4.15 Contoh <i>One Hot Encoding</i>	53
Gambar 4.16 <i>Source code</i> model	56
Gambar 4.17 Struktur Model	56
Gambar 4.18 Grafik Akurasi Training dan validation	59
Gambar 4.19 Grafik <i>Loss Training</i> dan <i>Validation</i>	59
Gambar 4.20 Halaman Chatbot.....	60
Gambar 4.21 Halaman Percakapan	60
Gambar 4.22 Halaman percakapan setelah bertanya	61
Gambar 4.23 <i>Source code</i> membuat data uji	62
Gambar 4.24 Contoh Data Uji	62
Gambar 4.25 Grafik Hasil Pengujian	63

DAFTAR TABEL

Tabel 3.1 Alat yang digunakan dalam penelitian.....	31
Tabel 3.2 Bahan yang digunakan dalam penelitian	32
Tabel 4.1 Nama dan Jenis File	37
Tabel 4.2 Jumlah informasi setiap pertanyaan.....	43
Tabel 4.3 Contoh Kosakata	47
Tabel 4.4 Contoh Konversi Teks menjadi Sequence Integer	51
Tabel 4.5 Tune Hyperparameter Learning rate 0,001	54
Tabel 4.6 Tune Hyperparameter Learning rate 0,1	55

BAB I

PENDAHULUAN

1.1. Latar Belakang

Seiring dengan perkembangan zaman, teknologi berkembang semakin cepat dan pesat, salah satunya yaitu kecerdasan buatan. Menurut Simon, 1987, kecerdasan buatan merupakan kecerdasan yang ditampilkan oleh suatu entitas buatan. Kecerdasan buatan adalah kawasan penelitian, instruksi, dan aplikasi yang berkaitan dengan komputer yang dalam pandangan manusia hal yang dilakukan komputer tersebut adalah cerdas [1]. 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). *Natural Language Processing* (NLP) adalah cabang dari AI yang berkaitan dengan interaksi antara komputer atau mesin dan manusia dengan menggunakan bahasa alami [2]. NLP banyak digunakan untuk penerjemah bahasa seperti Google Translate, personal asisten seperti Siri, pemeriksa kekurangan bahasa seperti Grammarly, ataupun chatbot.

Chatbot adalah program atau aplikasi yang dirancang untuk melakukan percakapan 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 chatbot dirancang untuk mensimulasikan percakapan manusia, sebenarnya tidak ada manusia di dalam percakapan tersebut [3]. 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 [4].

Seiring waktu, penelitian mengenai chatbot terlihat semakin bertambah. Seperti penelitian yang dilakukan oleh Vincentius Riandaru Prasetyo, Njoto Benarkah, dan Vioni Jannet Chrisintha pada tahun 2021 yang membahas mengenai Implementasi NLP dalam membuat chatbot di Program Information Technology Universitas Surabaya. Penelitian ini bertujuan untuk membuat aplikasi chatbot yang mampu membantu pengguna memperoleh informasi berkaitan dengan Program Information Technology pada Jurusan Teknik Informatika Universitas Surabaya. Chatbot ini berfokus pada pertanyaan dalam bahasa Inggris saja. Penelitian ini menggunakan pendekatan NLP dalam memproses pertanyaan dan memberikan jawabannya [5]. Selain itu, penelitian yang dilakukan oleh Androutsopoulou pada tahun 2019 yang memanfaatkan chatbot untuk peningkatan komunikasi antara warga dan pemerintah yang telah menjadi masalah sejak lama dalam sektor publik [6].

Metode yang digunakan dalam pembuatan chatbot juga mulai beragam. Beberapa metode yang digunakan antara lain *Recurrent Neural Network* dan *Long Short-Term Memory* (RNN-LSTM) [7], dan *Natural Language Processing* [8]. Metode *Long Short-Term Memory* memungkinkan pemrosesan bahasa alami yang akan memudahkan pengguna untuk melakukan komunikasi dengan komputer dalam bahasa alami. Metode *Long Short-Term Memory* cukup banyak digunakan dalam penelitian dalam pembangunan chatbot karena LSTM merupakan algoritma yang dapat menerima inputan dan mengeluarkan output berupa data sequences yang merupakan pengembangan dari metode *Recurrent Neural Network* (RNN).

Di Indonesia, penerapan chatbot masih terbilang kurang. Menurut survey yang dilakukan Pancake terhadap UMKM di Indonesia, UMKM yang sudah menerapkan chatbot hanya 15% [9]. Dapat disimpulkan bahwa masih banyak organisasi dan perusahaan yang belum menggunakan chatbot 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 bisa didapatkan dimana saja 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.

1.2. Rumusan Masalah

Rumusan masalah dalam penelitian ini adalah “Bagaimana membuat chatbot informasi akademik menggunakan algoritma *Long Short-Term Memory* (LSTM) di Program Studi Teknik Informatika Universitas Lampung”.

1.3. Tujuan Penelitian

Tujuan dari penelitian ini adalah mampu menerapkan algoritma *Long Short-Term Memory* (LSTM) pada chatbot informasi akademik di Program Studi Teknik Informatika Universitas Lampung dan menunjukkan keakuratan dari algoritma tersebut.

1.4. Manfaat Penelitian

Dengan adanya penelitian ini diharapkan dapat memberikan manfaat sebagai berikut:

1. Bagi Program Studi Teknik Informatika Universitas Lampung diharapkan dapat menerapkan chatbot untuk meningkatkan kemudahan mahasiswa/i dalam mendapatkan informasi terkait layanan akademik.
2. Dapat mengetahui penerapan *Long Short- Term Memory* (LSTM) pada chatbot.
3. Dapat mengetahui keakuratan algoritma *Long Short-Term Memory* (LSTM) pada chatbot informasi akademik di PSTI Universitas Lampung.
4. Penelitian ini diharapkan dapat menjadi referensi untuk perkembangan penelitian selanjutnya.

1.5. Batasan Masalah

Ruang lingkup permasalahan dalam penelitian ini akan dibatasi sebagai berikut:

1. Bahasa yang digunakan pada chatbot adalah Bahasa Indonesia.
2. Topik percakapan chatbot dibatasi seputar informasi layanan akademik PSTI Universitas Lampung.
3. Bahasa pemrograman yang digunakan adalah Python dengan framework Tensorflow dan Keras.

1.6. Sistematika Penulisan Skripsi

Sistematika penulisan yang digunakan pada skripsi ini adalah sebagai berikut:

BAB I : PENDAHULUAN

Pada bab ini menguraikan secara umum mengenai latar belakang penelitian, rumusan masalah, tujuan penelitian, manfaat penelitian, batasan penelitian serta sistematika penulisan penelitian Implementasi Long Short-Term Memory (LSTM) Pada Chatbot Informasi Akademik Di Program Studi Teknik Informatika Universitas Lampung.

BAB II : TINJAUAN PUSTAKA

Pada bab ini membahas mengenai teori-teori dasar yang digunakan sebagai sumber untuk mendukung penelitian mengenai seperti Kecerdasan Buatan, Neural Network, Deep Learning, Chatbot, Word Embedding, Recurrent Neural Network, Algoritma *Long Short-Term Memory* (LSTM), Dense Layer, Machine Learning Development Life Cycle, Text Mining, Indexing Word, Padding, One Hot Encoding, Optimizers, dan User Validation.

BAB III : METODOLOGI PENELITIAN

Pada bab ini membahas mengenai metodologi penelitian yang digunakan dalam Implementasi Long Short-Term Memory (LSTM) Pada Chatbot Informasi Akademik Di Program Studi Teknik Informatika Universitas Lampung.

BAB IV : HASIL DAN PEMBAHASAN

Pada bab ini membahas mengenai hasil serta pembahasan yang diperoleh dalam penelitian Implementasi Long Short-Term Memory (LSTM) Pada Chatbot Informasi Akademik Di Program Studi Teknik Informatika Universitas Lampung.

BAB V : KESIMPULAN DAN SARAN

Pada bab ini berisi tentang kesimpulan dan saran dari hasil penelitian yang telah dilakukan untuk penelitian lebih lanjut

yang berkaitan dengan masalah dalam penelitian ini di masa mendatang.

DAFTAR PUSTAKA

LAMPIRAN

BAB II

TINJAUAN PUSTAKA

2.1. Penelitian Terkait

Saat ini ada berbagai penelitian yang membahas chatbot dengan metode pendekatan Natural Language Processing (NLP), diantaranya yaitu penelitian oleh Elisabet Nila dan Irawan Afrianto pada tahun 2015 yang membahas mengenai aplikasi chatbot untuk informasi objek wisata Kota Bandung. Penelitian ini bertujuan untuk mempermudah pengunjung objek wisata Kota Bandung untuk menemukan informasi yang mereka butuhkan. Penelitian ini menggunakan metode waterfall dalam melakukan teknik analisis data dalam pembuatan perangkat lunak, diantaranya yaitu *system engineering, analysis, design, coding, implementation and testing*, dan *maintanance*. Hasil penelitian ini yaitu pengunjung setuju bahwa aplikasi chatbot dapat memudahkan pengunjung dalam mencari informasi sekitar wisata Kota Bandung.

Yang kedua yaitu jurnal oleh Mulyono pada tahun 2021 yang membahas mengenai chatbot untuk pelayanan online. Penelitian ini bertujuan untuk membantu bagian akademik dalam menjawab pertanyaan yang diajukan oleh mahasiswa. Metode yang digunakan bersumber dari kuesioner dari mahasiswa di Universitas Muhammadiyah Sumatera Barat. Kemudian diolah dengan Natural Language Processing. Hasil pengujian telah mampu merespon pertanyaan yang diajukan oleh mahasiswa [10].

Yang ketiga yaitu jurnal oleh Albert Yakobu Chandra, Didik Kurniawan, dan Rahmat Musa pada tahun 2020 yang membahas mengenai chatbot untuk pelayanan dalam *Coffee Shop*. Penelitian ini akan membangun chatbot yang mampu melayani

pelanggan *Cofee Shop* dalam memberikan informasi diharapkan pelanggan mampu melakukan transaksi secara otomatis. Chatbot pada penelitian ini dibangun dengan memanfaatkan tools Dialogflow dari Google [11].

Yang keempat yaitu jurnal oleh Vincentius Riandaru Prasetyo, Njoto Benarkah, dan Vioni Jannet Chrisintha pada tahun 2021 yang membahas mengenai Implementasi NLP dalam membuat chatbot di Program Information Technology Universitas Surabaya. Penelitian ini bertujuan untuk membuat aplikasi chatbot yang mampu membantu pengguna memperoleh informasi berkaitan dengan Program Information Technology pada Jurusan Teknik Informatika Universitas Surabaya. Chatbot ini berfokus pada pertanyaan dalam bahasa Inggris saja. Penelitian ini menggunakan pendekatan NLP dalam memproses pertanyaan dan memberikan jawabannya.

Yang kelima yaitu jurnal oleh Laras Nurhayatunnufus, Muhammad Poncut Ridha, dan Hariandi Maulid, S.T., M.Sc. pada tahun 2020 yang membahas aplikasi chatbot untuk informasi dalam memilih laptop menggunakan metode NLP. Penelitian ini bertujuan untuk membantu pengguna dalam mencari laptop sesuai dengan kebutuhan pengguna. Hasil pengujian menunjukkan bahwa pengguna sangat setuju chatbot dapat membantu dalam mencari laptop sesuai dengan kebutuhan [12].

2.2. Kecerdasan Buatan

Menurut John McCarthy, 1956, kecerdasan buatan merupakan cara untuk membuat model yang dapat melakukan proses berpikir seperti manusia dan mendesain mesin sehingga mesin tersebut dapat melakukan perilaku seperti manusia [13]. Menurut Simon, 1987, kecerdasan buatan merupakan kecerdasan yang ditampilkan oleh suatu entitas buatan. Kecerdasan dibuat dan dimasukkan ke dalam suatu mesin atau komputer agar mampu melakukan pekerjaan atau kegiatan yang dapat dilakukan oleh manusia. Dengan demikian, kecerdasan buatan adalah suatu kecerdasan yang diterapkan dalam mesin atau komputer sehingga mesin atau komputer tersebut dapat berperilaku layaknya manusia, seperti melakukan percakapan dengan

manusia, bermain *game* dengan manusia, menerjemahkan bahasa, memberikan rekomendasi, mendeteksi penyakit tertentu, dan masih banyak lagi.

Menurut Kaplan, 2010, kelebihan dari kecerdasan buatan, yaitu sebagai berikut.

1. Kecerdasan buatan lebih bersifat permanen karena tidak berubah selama program tidak berubah.
2. Kecerdasan buatan mudah dibagi karena pengetahuan pada suatu komputer tetap dapat digunakan walau berpindah komputer.
3. Kecerdasan buatan lebih murah karena lebih konsisten dalam mengerjakan aktivitas dalam jangka waktu yang lama.
4. Kecerdasan buatan dapat didokumentasikan karena dapat dilacak setiap aktivitas dari sistem [14].

2.3. Neural Network

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 dari satu input dan satu output layer, dan satu atau beberapa hidden layer, dimana setiap lapisan terdiri dari beberapa neuron. Hubungan 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 [15].

Layer *Feed Neural Network* terdiri dari neuron yang terhubung tanpa jalan pintas atau *loop* umpan balik. Saat memproses data sekuensial, aktivasi unit *hidden layer* pada waktu t dapat dilihat pada persamaan berikut.

$$h_t = g_h W_h^T x_t b_h$$

Dimana fungsi aktivasi hidden layer g_h adalah fungsi dari matriks bobot tersembunyi W_h , sebuah vektor masukan x_t , dan bias b_h . Hal ini menghasilkan fungsi aktivasi tersembunyi h_t . Sebuah prediksi adalah hasil dari penerapan fungsi aktivasi output yang sesuai dengan pembobotan jumlah aktivasi yang diterima di layer output. Oleh karena itu, prediksinya hanya fungsi dari input, bobot, dan bias.

$$h_t = g_h W_h^T x_t b_h$$

Bobot dan bias menentukan mapping dari x_t untuk y_t dan akan dipelajari selama pelatihan. Feed Neural Network memperlakukan urutan input sebagai data waktu invarian dan dengan demikian menjadi agonistik fitur inheren deret waktu. Pada suatu titik waktu tertentu, t , sebuah FNN mengambil input x_t memetakannya ke h_t aktivasi tersembunyi. Hal ini berulang pada saat pengamatan berikutnya, $t+1$, sedangkan kedua pemetaan tidak saling berhubungan [16].

2.4. Deep Learning

Deep learning adalah bagian dari machine learning yang terdiri dari banyak layer atau lapisan yang membentuk tumpukan. Lapisan tersebut yaitu sebuah algoritma ataupun metode yang melakukan klasifikasi perintah yang diinput hingga menghasilkan sebuah output. Metode *Deep Learning* yang sedang berkembang di antaranya yaitu RNN (*Reccurent Neural Network*), LSTM (*Long Short-Term Memory*), dan CNN (*Convolutional Neural Network*).

Bidang *deep learning* saat ini telah banyak berkembang sehingga lebih mudah digunakan karena banyaknya *library* dan *Application Program Interface* (API). *Library* yang digunakan yaitu Tensorflow yang merupakan antarmuka untuk melakukan implementasi algoritma pembelajaran mesin atau *machine learning* dan untuk melakukan eksekusi perintah dengan menggunakan informasi atau data yang dimiliki tentang suatu objek. Tensorflow mempunyai fitur untuk menjalankan training model menggunakan CPU (*Central Processing Unit*) dan training model GPU (*Graphic Processing Unit*) [17].

Deep learning merupakan salah satu kecerdasan buatan yang dapat meniru proses dari kerja otak manusia. Teknologi deep learning sangat efektif dalam melakukan pengolahan data mentah dan menciptakan pola yang digunakan untuk pengambilan keputusan. Deep learning adalah bagian dari machine learning yang mempunyai jaringannya sendiri. Deep learning memiliki kemampuan untuk mengenali pola dan informasi tanpa pengawasan dari data yang tidak memiliki label.

Jenis – jenis algoritma deep learning yaitu *convolutional neural networks* (CNN), *Long short term memory network* (LSTM), *reccurent neural network* (RNN), dan *self organizing maps* (SOM). CNN terdiri dari beberapa lapisan dan biasa digunakan untuk melakukan pemrosesan gambar dan deteksi suatu objek Contoh penerapan.CNN diantaranya yaitu untuk mengidentifikasi citra pada satelit, memprediksi *time series* memproses citra medis, dan mendeteksi suatu anomali. SOM adalah algoritma deep learning yang mampu membuat visualisasi data secara mandiri. Kemampuan SOM berfungsi dalam mengurangi dimensi pada data melalui jaringan saraf tiruan yang mampu bekerja secara otomatis. Visualisasi data ini bertujuan biasanya untuk permasalahan yang secara umum cukup sulit diselesaikan oleh manusia.

Contoh deep learning yang paling terkenal adalah Netflix dan Youtube. Deep learning yang digunakan dalam kedua platform tersebut adalah sistem rekomendasi video. Platform tersebut memanfaatkan minat dan *behavior* pengguna untuk menemukan video yang sesuai dengan kebutuhan pengguna.

Manfaat teknologi deep learning diantaranya yaitu untuk memaksimalkan kinerja data yang tidak terstruktur, menghilangkan kebutuhan teknologi rekayasa fitur, memberikan hasil yang lebih baik dan berkualitas, dapat mengurangi biaya dalam operasional, serta dapat membantu dalam teknik manipulasi data yang efektif [18].

2.5. Chatbot

Chatbot adalah program atau aplikasi yang dirancang untuk melakukan percakapan 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. Ada dua jenis utama chatbot: *rule-based* dan *contextual*.

Chatbot rule-based (juga disebut skrip) adalah bentuk chatbot yang paling sederhana, berfungsi sebagai pohon keputusan dasar dengan pernyataan if/then (jika orang tersebut mengatakan/memilih x, maka respons dengan y). Chatbot paling dasar dalam kategori ini bergantung sepenuhnya pada menu atau tombol, daripada mengizinkan pengguna untuk mengetik permintaan mereka. Pengguna memilih opsi, dan bot menjawab dengan tindakan atau respons yang telah ditentukan sebelumnya, tergantung pada pilihannya. Ini berfungsi seperti IVR telepon, di mana Anda memilih nomor untuk memberi sinyal apa yang ingin Anda lakukan dan diberikan opsi lebih lanjut sampai Anda diarahkan ke informasi atau orang yang benar.

Chatbot kontekstual jauh lebih canggih daripada rule-based. Bot ini menggunakan Natural Language Processing (NLP) untuk memahami ucapan atau teks pengguna secara akurat. Disebut kontekstual karena dapat memahami konteks percakapan yang sebenarnya dan maksud di baliknya daripada hanya memindai teks untuk kata kunci, mirip dengan cara asisten virtual seperti Siri atau Alexa berfungsi. Misalnya, jika Anda bertanya "Jam berapa sekarang di London?" itu bisa merespon dengan waktu yang tepat, dan jika Anda mengikutinya dengan bertanya, "Bagaimana dengan Paris?" itu dapat menyimpulkan bahwa Anda masih bertanya tentang waktu dan merespons dengan benar, meskipun waktu tidak termasuk dalam permintaan kedua.

2.6. Word Embedding

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 [19].

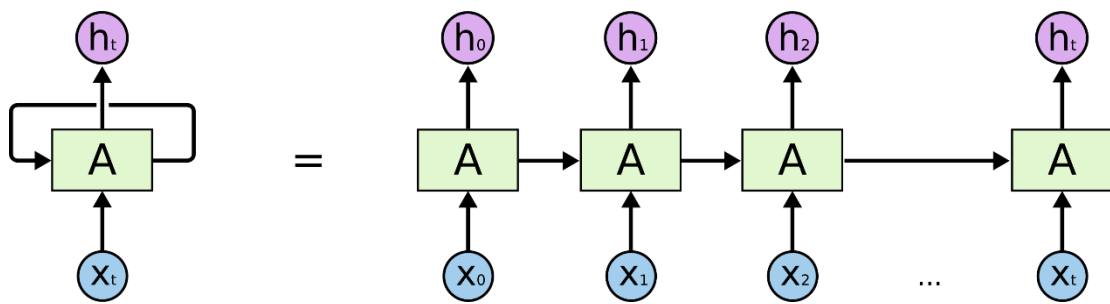
Berdasarkan Mkolov dkk, 2013a, arsitektur model untuk menentukan nilai vektor dari suatu kata ada dua, yaitu *continous bag-of-words model* (CBoW) dan *skip gram* (SG) [20]. Dari nilai vektor dapat dihitung kemiripan kata maupun kemiripan antara pasangan kata. Dua model vektor kata tersebut dapat dibangun dengan *feedforward neural network* sederhana dan dapat dilihat berdasarkan konteksnya. Perbedaannya, CBoW dioptimalkan untuk melakukan prediksi kata dari kata – kata di sekitar atau konteksnya, sedangkan SG lebih dioptimalkan untuk melakukan prediksi konteks dari kata sekarang dan melakukan prediksi kata-kata disekitarnya.

2.7. Reccurent Neural Network

Reccurent Neural Network atau biasa disebut dengan RNN adalah metode Jaringan saraf 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 pemrosesan data sekuensial, RNN memasukkan loop umpan balik yaitu memberi output neuron sebelumnya sebagai input dari neuron setelahnya. Informasi dari *time step* sebelumnya disimpan sebagai hidden state dan dapat dijadikan input untuk layer selanjutnya. Jika dilihat dari urutan dari *time step* ke *time step*, RNN melakukan salinan jaringan sarap yang mirip rantai, masing-masing *time step* meneruskan informasi ke time step penerusnya. Dalam RNN, *training*

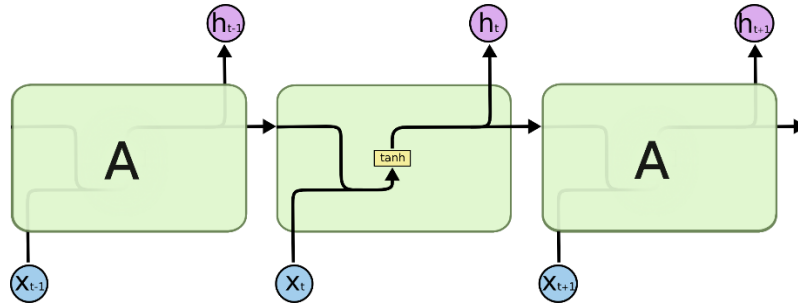
gradient descent disebut dengan backpropagation melalui waktu, sebagai turunan dari error tidak hanya backpropagation melalui jaringan itu sendiri tapi juga kembali melalui waktu berulang [21]. Struktur pada RNN dapat dilihat pada gambar sebagai berikut.



Gambar 2.1 Struktur RNN [22]

Tujuan dari training adalah menghitung turunan dari fungsi aktivasi dengan menggunakan aturan rantai sehingga gradien akan semakin mengecil. Dalam masalah RNN, terjadi masalah penghilangan gradien, diperkuat oleh pemrosesan data sekuensial dari jaringan. Terkadang, gradien menghilang tidak hanya melintasi layer tetapi juga melintasi time step. Akibatnya, RNN menghadapi kesulitan dalam menghadapi kesulitan dalam pemodelan dependensi jangka panjang.

Perhatikan bahwa RNN dari aturan rantai dalam pelatihan jaringan saraf juga 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. Semua jaringan saraf berulang memiliki bentuk rantai modul berulang jaringan saraf. Dalam RNN standar, modul berulang ini akan memiliki struktur yang sangat sederhana, seperti lapisan tanh tunggal seperti terlihat pada gambar 2.2.



Gambar 2.2 Layer dalam RNN

2.8. Long Short-Term Memory

Long short-term memory atau yang disingkat LSTM telah digunakan secara ekstensif dalam *Natural Language Processing* (NLP). LSTM diperkenalkan pertama kali pada 1997 oleh Hochreiter and Schmidhuber yang kemudian menjadi populer dalam beberapa tahun belakangan karena meningkatnya kecepatan perkembangan kecepatan *hardware* untuk kecepatan deep learning [23].

Long Short-Term Memory (LSTM) merupakan salah satu jenis dari arsitektur RNN atau *Reccurent 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.

LSTM dapat mendeteksi data yang akan disimpan dan data yang tidak digunakan untuk dipangkas, karena LSTM memiliki 4 layer neuron yang biasa disebut gates untuk mengatur memori pada setiap neuron [24]. 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.

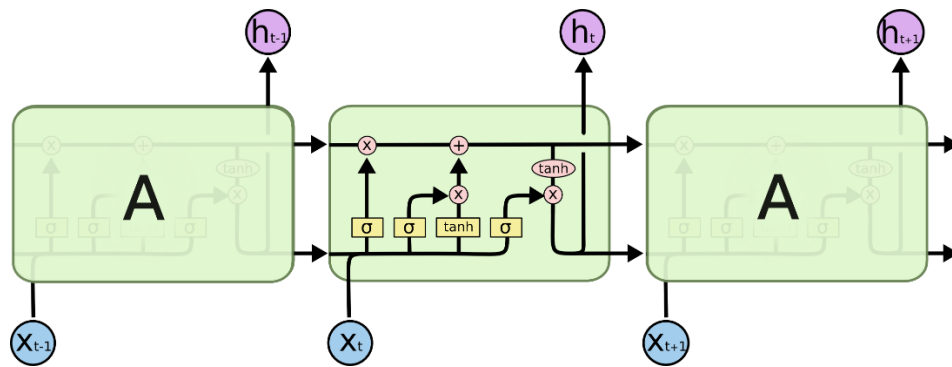
$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (2.1)$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (2.2)$$

$$C_t = f_t C_t + i_t \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \quad (2.3)$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (2.4)$$

$$h_t = o_t \tanh(C_t) \quad (2.5)$$

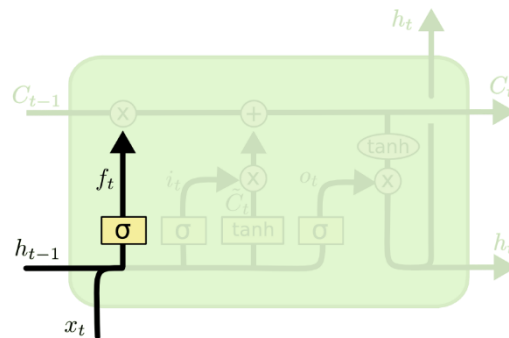


Gambar 2.3 Jaringan LSTM

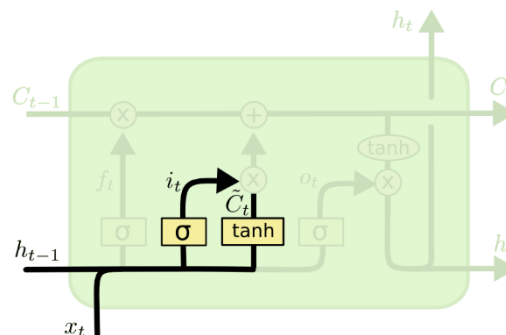
Keterangan berdasarkan Gambar 2.3 adalah sebagai berikut.

- Kotak berwarna kuning merupakan neural network layer yaitu jaringan saraf yang dipelajari.
- Anak panah tanpa cabang atau garis penggabungan adalah vector transfer yaitu garis yang membawa seluruh vektor, dari keluaran satu neuron ke masukan 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.

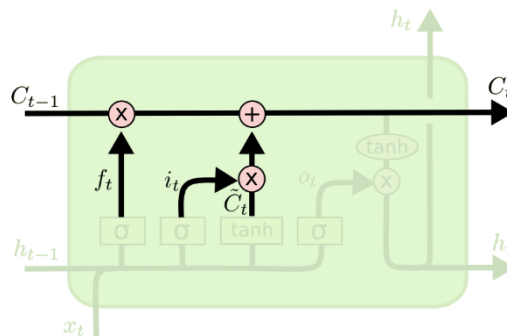
Gambar 2.4 *Forget Gate LSTM*

Persamaan 2.1 adalah persamaan untuk *output forget gate*. Persamaan ini berfungsi untuk memutuskan apakah informasi akan disimpan atau dilupakan dari *cell state* C_{t-1} yang merupakan hasil perkalian dengan *cell state*. Keputusan ini dibuat oleh fungsi sigmoid. Fungsi sigmoid yang menerima input h_{t-1} dan x_t akan menghasilkan angka antara nol dan satu. Angka 1 artinya *cell state* akan disimpan. Angka 0 artinya *cell state* akan dilupakan. Proses ini digambarkan dalam gambar 2.4.

Gambar 2.5 *Input Gate LSTM*

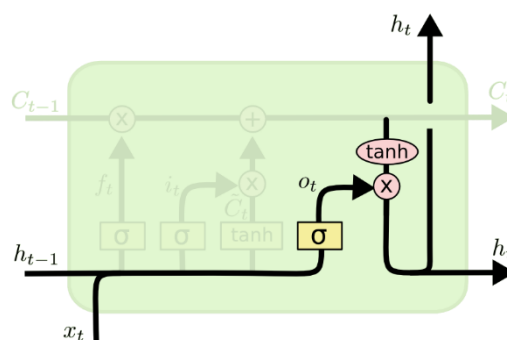
Persamaan 2.2 adalah persamaan gerbang *input LSTM*. Secara sederhana, persamaan ini menciptakan kandidat baru untuk *cell state*. Input gate memiliki dua bagian. Pertama yaitu sigmoid yang akan menentukan nilai yang mana yang akan di perbarui. Selanjutnya yaitu fungsi aktivasi tanh yang menciptakan kandidat baru

yang ditambahkan dengan *cell state*. Pada langkah selanjutnya akan digabungkan untuk membuat pembaruan *cell state*. Proses ini dapat dilihat pada gambar 2.5.



Gambar 2.6 Cell State LSTM

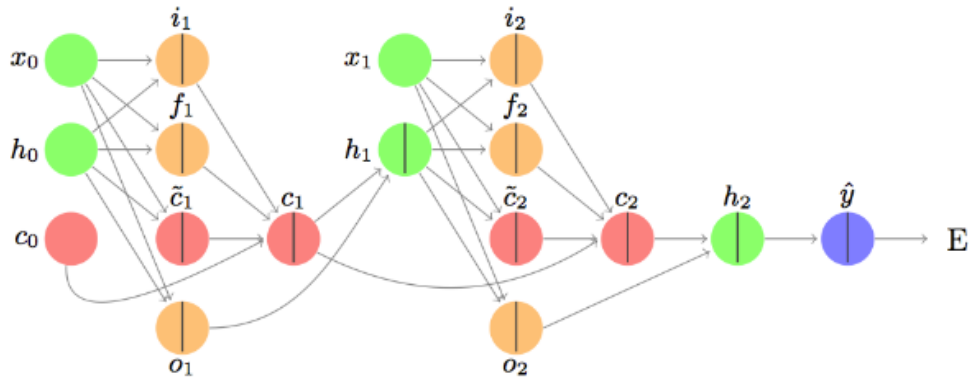
Cell state kemudian diperbarui berdasarkan persamaan 2.3. Sederhananya, *cell state* akan menggabungkan cell state hasil dari forget gate dan kandidat baru hasil dari input gate. *Cell state* hasil dari *forget gate* yang memutuskan apakah akan dilupakan atau diteruskan didapat dengan mengalikan f_t dengan *cell state* sebelumnya. Calon kandidat baru cell state didapat dengan mengalikan i_t dengan C_t . Proses ini dapat dilihat pada gambar 2.6.



Gambar 2.7 Output Gate LSTM

Setelah *cell state* diperbarui, output gate dapat dihitung menggunakan persamaan 2.4 untuk menghasilkan *hidden state*. Hidden state didasarkan pada cell state namun di saring kembali. Pertama, fungsi aktivasi sigmoid akan memutuskan bagian mana dari cell state yang akan dijadikan output. Kemudian cell state dimasukkan dalam

fungsi aktivasi tanh yang menghasilkan nilai antara -1 dan 1 kemudian dikalikan dengan hasil dari sigmoid yang akan menghasilkan output. Gambar lain untuk lebih memahami LSTM dapat dilihat pada gambar 2.8 [25].



Gambar 2.8 Ilustrasi LSTM dengan Dua *Cell*

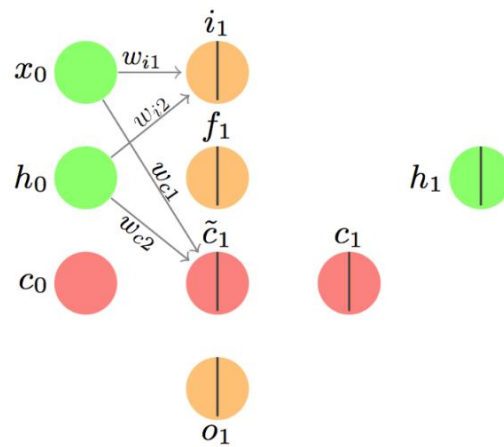
Contoh perhitungan LSTM adalah sebagai berikut.

Misalkan input *sequence* $X_0 = 0.1, X_1 = 0.2$, bobot, dan bias adalah sebagai berikut.

$$W = \begin{bmatrix} W_{i1} & W_{i2} & b_i \\ W_{c1} & W_{c2} & b_c \\ W_{f1} & W_{f2} & b_f \\ W_{o1} & W_{o2} & b_o \\ W_y & 0 & b_y \end{bmatrix} = \begin{bmatrix} 0.5 & 0.25 & 0.01 \\ 0.3 & 0.4 & 0.05 \\ 0.03 & 0.06 & 0.002 \\ 0.02 & 0.04 & 0.001 \\ 0.6 & 0 & 0.025 \end{bmatrix}$$

Maka perhitungan pada cell pertama adalah sebagai berikut.

Input gate akan melakukan perhitungan sebagai berikut.

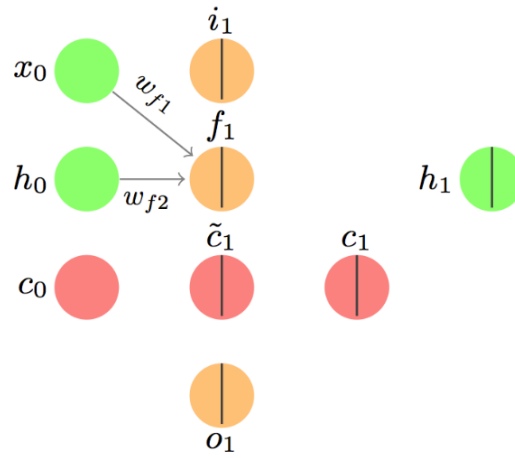
Gambar 2.9 Perhitungan *Input Gate*

$$\begin{aligned}
 i_1 &= \sigma(W_{i1}X_1 + W_{i2}h_0 + b_i) \\
 &= \sigma(0.5 \times 0.1 + 0.25 \times 0 + 0.01) \\
 &= \sigma(0.06) \\
 &= \frac{1}{1 + e^{(-0.06)}} \\
 &= 0.515
 \end{aligned}$$

Input kedua yaitu terkait dengan sel memori yang akan membuat suatu cell state baru.

$$\begin{aligned}
 \tilde{c}_1 &= \tanh(W_{c1}X_1 + W_{c2}h_0 + b_c) \\
 &= \tanh(0.3 \times 0.1 + 0.4 \times 0 + 0.05) \\
 &= \tanh(0.08) \\
 &= \frac{e^{(0.08)} - e^{(-0.08)}}{e^{(0.08)} + e^{(-0.08)}} \\
 &= 0.0798
 \end{aligned}$$

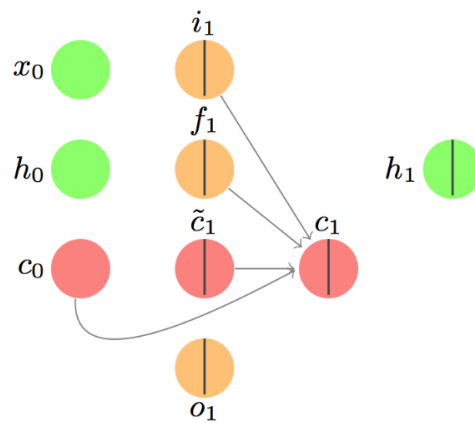
Forget gate akan melakukan perhitungan sebagai berikut.



Gambar 2.10 Perhitungan *Forget Gate*

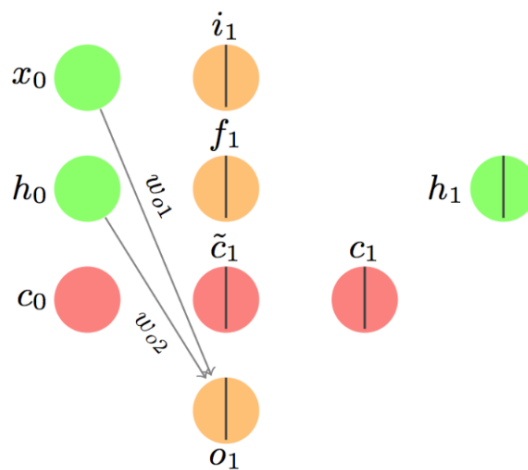
$$\begin{aligned}
 f_1 &= \sigma(W_{f1}X_1 + W_{f2}h_0 + b_f) \\
 &= \sigma(0.03 \times 0.1 + 0.06 \times 0 + 0.02) \\
 &= \sigma(0.05) \\
 &= \frac{1}{1 + e^{(-0.05)}} \\
 &= 0.5012
 \end{aligned}$$

Memory cell akan melakukan perhitungan sebagai berikut.

Gambar 2.11 Perhitungan *Cell State*

$$\begin{aligned}
 C_1 &= i_t \times C_0 + f_1 \times \tilde{c}_1 \\
 &= 1 \times 0.0798 + 1 \times 0 \\
 &= 0.0798
 \end{aligned}$$

Output gate akan melakukan perhitungan sebagai berikut.

Gambar 2.12 Perhitungan *Output Gate*

$$\begin{aligned}
O_1 &= \sigma(W_{o1}X_1 + W_{o2}h_0 + b_o) \\
&= \sigma(0.02 \times 0.1 + 0.04 \times 0 + 0.001) \\
&= \sigma(0.003) \\
&= \frac{1}{1 + e^{(-0.003)}} \\
&= 0.5007
\end{aligned}$$

Hidden state akan melakukan perhitungan sebagai berikut.

$$\begin{aligned}
h_1 &= O_1 \times \tanh(C_1) \\
&= 1 \times 0.0796 \\
&= 0.0796
\end{aligned}$$

Selanjutnya yaitu perhitungan pada cell kedua yaitu sebagai berikut.

Input gate akan melakukan perhitungan sebagai berikut.

$$\begin{aligned}
i_2 &= \sigma(W_{i1}X_2 + W_{i2}h_1 + b_i) \\
&= 0.52875
\end{aligned}$$

Input kedua yaitu terkait dengan sel memori yang akan membuat suatu cell state baru.

$$\begin{aligned}
\tilde{c}_2 &= \tanh(W_{c1}X_2 + W_{c2}h_1 + b_c) \\
&= 0.11768
\end{aligned}$$

Forget gate akan melakukan perhitungan sebagai berikut.

$$\begin{aligned} f_2 &= \sigma(W_{f1}X_2 + W_{f2}h_1 + b_f) \\ &= 0.50231 \end{aligned}$$

Memory cell akan melakukan perhitungan sebagai berikut.

$$\begin{aligned} C_2 &= i_2 \times \tilde{c}_2 + f_2 \times C_1 \\ &= 0.61999 \end{aligned}$$

Output gate akan melakukan perhitungan sebagai berikut.

$$\begin{aligned} O_2 &= \sigma(W_{o1}X_2 + W_{o2}h_0 + b_o) \\ &= 0.50145 \end{aligned}$$

Hidden state akan melakukan perhitungan sebagai berikut.

$$\begin{aligned} h_2 &= O_2 \times \tanh(C_2) \\ &= 0.5511 \end{aligned}$$

Perhitungan akhir untuk mencari prediksi adalah sebagai berikut.

$$\begin{aligned} \hat{y} &= W_y h_2 + b_y \\ \hat{y} &= 0.6 \times 0.511 + 0.025 \\ \hat{y} &= 0.335566 \end{aligned}$$

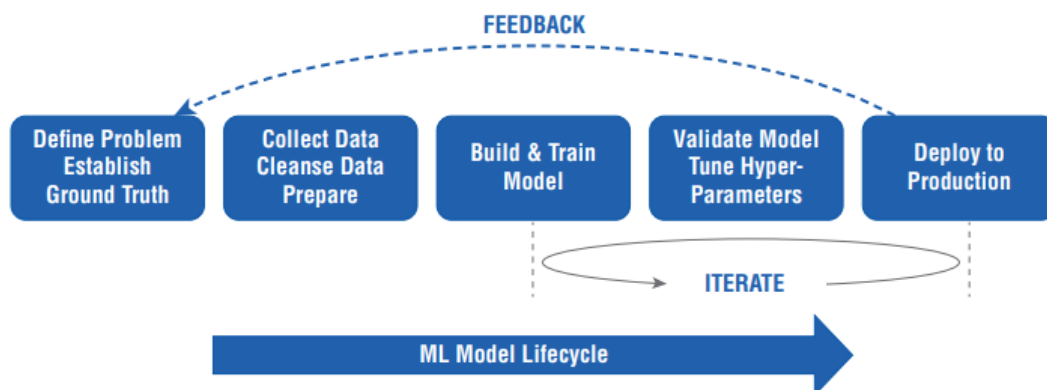
2.9. Dense Layer

Dalam setiap jaringan saraf, *dense layer* adalah lapisan yang sangat terhubung dengan lapisan sebelumnya yang berarti neuron dari lapisan terhubung ke setiap neuron dari lapisan sebelumnya. Lapisan ini merupakan lapisan yang paling umum digunakan dalam jaringan syaraf tiruan.

Neuron dense layer dalam sebuah model menerima output dari setiap neuron lapisan sebelumnya, di mana *neuron dense layer* melakukan perkalian matriks-vektor. Perkalian vektor matriks adalah prosedur di mana vektor baris dari output dari lapisan sebelumnya sama dengan vektor kolom dari lapisan padat. Aturan umum perkalian matriks-vektor adalah bahwa vektor baris harus memiliki banyak kolom seperti vektor kolom. Dense layer sebagai output digunakan untuk melakukan prediksi berdasarkan inputan pada layer sebelumnya [26].

2.10. Machine Learning Development Life Cycle

Machine Learning Development Life Cycle yaitu panduan dalam pengembangan model machine learning dari awal hingga selesai. Berikut merupakan gambar dari MDLC.



Gambar 2.13 Alur MDLC [27]

Langkah pertama adalah mendefinisikan masalah yang akan di selesaikan. Langkah ini diantaranya yaitu seorang machine learning engineer harus memahami dengan jelas domain masalah, bertemu dengan pengguna dan pakar domain, dengan mengajukan pertanyaan sebanyak mungkin. Cari tahu faktor apa saja yang memengaruhi masalah yang dihadapi. Cari tahu elemen apa yang dapat dipelajari dari sistem yang dapat diukur. Kemudian tentukan metrik yang ada dan pengukuran baru yang perlu ditambahkan sebagai tujuan dari masalah yang akan dipecahkan.

Langkah selanjutnya yaitu persiapan data. Langkah ini adalah langkah yang biasanya paling banyak menghabiskan banyak waktu. Data dapat dicari dengan melakukan pengamatan atau observasi berdasarkan masalah yang diangkat atau dapat dicari dari sumber sumber terpercaya di internet. Selanjutnya yaitu tahap pembersihan data. Tahap pembersihan data merupakan tahap yang sangat penting untuk membersihkan data lapangan yang dapat digunakan untuk melatih model. Pembersihan diantaranya yaitu penggantian atau penghilangan data yang buruk atau hilang dari dataset. Setelah mengumpulkan data dan melakukan pembersihan data, langkah selanjutnya yaitu menyiapkan data agar bisa diproses oleh model. Hal ini melibatkan rekayasa fitur dan pemisahan data menjadi data *training* dan *validation*.

Setelah data dibersihkan dan siap untuk diproses oleh model, selanjutnya yaitu membangun dan melakukan pelatihan pada model. Model yang akan dibangun tergantung pada jenis data yang akan di proses, ada arsitektur jaringan saraf tertentu yang telah di standardisasi. Untuk analisis citra, CNN lebih diterima secara universal sebagai arsitektur yang dipilih. Untuk urutan data seperti teks atau audio, standar dalam industri adalah RNN atau LSTM untuk jenis memori jangka pendek. Dalam membangun model, dapat menggunakan pendekatan pemrograman. Bahasa yang digunakan untuk membuat model biasanya yaitu Python, R, atau MATLAB lalu model disimpan dalam format biner untuk deployment.

Langkah selanjutnya yaitu validasi model dan melakukan *tune hyper-parameters*. Setelah model dibuat, model perlu dilatih dan divalidasi terhadap dataset. Untuk mendapatkan hasil pelatihan dan validasi yang baik, harus dilakukan *tune hyper-parameters*. *Hyper-parameters* yang dapat diubah antara lain jumlah layer, neuron dalam layer, learning rate, jenis fungsi aktivasi, dll. Langkah selanjutnya yaitu

melakukan *deployment*. Setelah model dilatih dan divalidasi dengan hasil yang dapat diterima, model dapat diteruskan pada tahap produksi.

2.11. Text Mining

Text mining adalah proses dalam melakukan ekstraksi pola dalam melakukan eksplorasi pengetahuan dari sumber data yang berbentuk teks. Tahap awal dalam proses *text mining* adalah mengumpulkan data yang berasal dari berbagai sumber seperti teks biasa, file pdf, halaman web, ataupun observasi secara langsung. Selanjutnya yaitu melakukan pemrosesan data dan pembersihan data. Proses ini dilakukan untuk memastikan agar dapat diambil makna yang sebenarnya [28].

Text mining adalah proses pencarian pola atau melakukan panggilan informasi baru. Tujuan dari hal ini yaitu agar dapat menemukan informasi yang penting dari suatu teks dengan melakukan perubahan pada teks menjadi data yang dapat digunakan untuk analisa yang lebih lanjut. Tahapan yang dilakukan pada *text mining* diantaranya yaitu sebagai berikut.

1. Case Folding

Case folding yaitu tahapan mengubah seluruh huruf/kata yang ada dalam dataset menjadi huruf kecil dan hanya huruf “a” hingga “z” yang akan diterima. Case folding juga melakukan proses penghilangan karakter lain selain huruf “a” hingga “z” seperti tanda koma, titik, titik dua, titik koma, tanda tanya, dan sebagainya.

2. Tokenizing

Tokenizing adalah memotong tiap kata dalam suatu kalimat dalam dataset dengan menggunakan spasi sebagai delimiter yang dapat menghasilkan token berupa kata.

3. Filtering

Filtering adalah menyaring kata yang dianggap tidak penting atau mungkin tidak memiliki makna dalam proses *text mining* 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.

2.12. Indexing word

Karena bahasa adalah bagian integral dari kehidupan kita dan masyarakat kita, kita secara alami dikelilingi oleh banyak teks. Teks tersedia bagi kita dalam bentuk buku, artikel berita, artikel Wikipedia, tweet dan dalam banyak bentuk lainnya dan dari banyak sumber yang berbeda.

Jumlah teks yang sangat besar ini mungkin merupakan jumlah data terbesar yang tersedia untuk kami. Banyak wawasan dapat diperoleh dari kata-kata tertulis dan informasi yang diekstraksi dapat menghasilkan hasil yang sangat berguna yang dapat digunakan untuk berbagai aplikasi. Namun, semua data ini memiliki satu kekurangan kecil, yaitu, teks adalah bentuk data yang paling tidak terstruktur yang tersedia bagi kita. Ini adalah bahasa murni tanpa implikasi matematis apa pun. Sayangnya, semua algoritma pembelajaran mesin dan pembelajaran mendalam kami bekerja pada angka dan bukan pada teks. Oleh karena itu, kita perlu mengubahnya menjadi data angka yang dapat kita masukkan ke algoritma sehingga mereka dapat menghasilkan model [29].

Salah satu metode yang digunakan untuk mengubah text menjadi integer adalah menggunakan submodule text dari `sklearn.feature_extraction`. `CountVectorizer` adalah kelas dalam modul `sklearn.feature_extraction.text`. Kelas ini berguna untuk membangun kosa kata. Selain itu, menghasilkan representasi numerik dari teks yang kita butuhkan, yaitu vektor Numpy [30].

2.13. Padding

Semua neural network membutuhkan sebuah inputan yang berukuran dan berbentuk sama. Dalam pemrosesan teks sebagai input untuk model seperti dalam

LSTM, tidak semua kalimat memiliki panjang yang sama. Secara alami, sebuah kalimat bisa panjang dan bisa pendek. Oleh karena itu kita perlu mengubah input menjadi teks dengan ukuran yang sama. Mengubah ukuran teks menjadi ukuran yang sama disebut dengan padding.

Kita membutuhkan padding karena setiap kalimat dalam teks tidak mempunyai jumlah kata yang sama. Kita dapat mendefinisikan jumlah maksimal kata untuk setiap kalimat. Jika kalimat lebih panjang dari maksimal yang telah ditentukan, maka sisa kata akan dihapus. Jika kalimat lebih pendek maka kalimat akan diberi penambahan angka 0 di awal atau di akhir kalimat [31].

2.14. One Hot Encoding

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. *One Hot Encoding* membuat fitur biner baru untuk setiap kategori dan memberikan nilai 1 ke fitur setiap sampel yang sesuai dengan kategori aslinya. Hal ini berlaku pada penelitian ini karena penelitian ini bekerja dengan jenis klasifikasi urutan teks menggunakan metode deep learning *Long Short-Term Memory* (LSTM) [32].

One Hot Encoding adalah bagian yang sangat penting dari proses rekayasa fitur dalam pelatihan untuk teknik pembelajaran. Misalnya, variabel seperti warna dan labelnya adalah "merah", "hijau", dan "biru", label dapat dikodekan sebagai vektor biner tiga elemen sebagai Merah: [1, 0, 0], Hijau: [0, 1, 0], dan Biru: [0, 0, 1]. Data Kategoris saat memproses, harus dikonversi ke bentuk numerik. *One Hot Encoding* umumnya diterapkan pada representasi integer dari data. Di sini variabel yang dikodekan integer dihapus dan variabel biner baru ditambahkan untuk setiap nilai integer unik. Selama proses, dibutuhkan kolom yang memiliki data kategorikal, yang telah dikodekan label dan kemudian membagi kolom tersebut menjadi beberapa kolom [33].

2.15. Optimizers

Optimizers memiliki beberapa jenis untuk model deep learning, diantaranya yaitu SGD, Adam, RMSProp, dan banyak lagi. Penelitian ini menerapkan optimizers Adam untuk pelatihan data. Optimizers Adam dapat berguna dalam mengontrol masalah *sparse gradient*. Hal ini yaitu perluasan untuk turunan *gradien stokastik* yang pada saat ini lebih luas dalam aplikasi *deep learning* seperti *natural language processing*. Adam melakukan proses perubahan bobot menggunakan *gradient descent*, yaitu mengubah *learning rate* yang telah ditetapkan secara dinamis sehingga dapat terhindar dari gradient descent yang terjebak pada minimum lokal. Perbandingan performa dengan berbagai jenis optimasi *gradien descent* yang terbaik adalah Adam. Adam lebih mudah dalam implementasi, memori yang dibutuhkan relatif kecil, efisien jika dihitung dalam komputasi. Algoritma adam menghitung rata-rata bergerak eksponensial dari gradien dan gradien kuadrat [34].

2.16. User Validation

User validation adalah metode validasi dimana user dapat mencoba langsung sistem yang telah dibangun dan kemudian user dapat langsung menilai apakah output yang diberikan oleh sistem sudah sesuai dengan kebutuhan dari user. Akurasi sistem dapat dihitung dengan cara membagi jumlah validasi benar dengan total pertanyaan data uji kemudian dikali dengan 100%.

BAB III

METODE PENELITIAN

3.1. Waktu dan Tempat Penelitian

Waktu dan tempat pelaksanaan penelitian ini dilakukan pada:

1. Waktu penelitian : Februari 2022 sampai dengan Juni 2022
2. Tempat penelitian : Laboratorium Teknik Komputer

3.2. Alat dan Bahan

Alat dan bahan yang digunakan dalam penelitian ini adalah sebagai berikut:

Tabel 3.1 Alat yang digunakan dalam penelitian

No	Nama Alat	Spesifikasi	Deskripsi
1	HP Joy XGen 14s-DQ1013TU	Core i3-10051G1 dengan RAM 8 GB dan sistem operasi Windows 10	Perangkat keras yang digunakan dalam data processing, proses pembuatan model serta training model.
2	Google Colaboratory	-	Layanan berbasis cloud yang digunakan dalam data processing, proses pembuatan model serta training model.

3	Visual Studio Code	Versi 1.65.2	Perangkat lunak yang digunakan untuk membangun code untuk deployment model
4	Python	Versi 3.8.0	Bahasa pemrograman yang digunakan untuk processing data, proses pembuatan model, training model, dan deployment.
5	Flask Framework	Versi 2.0.2	Framework yang digunakan untuk deployment model.
6	Tensorflow Framework	Versi 2.8.0	Framework yang digunakan untuk membangun model dan melakukan pelatihan model.
7	Google Drive	-	Tempat penyimpanan dataset.

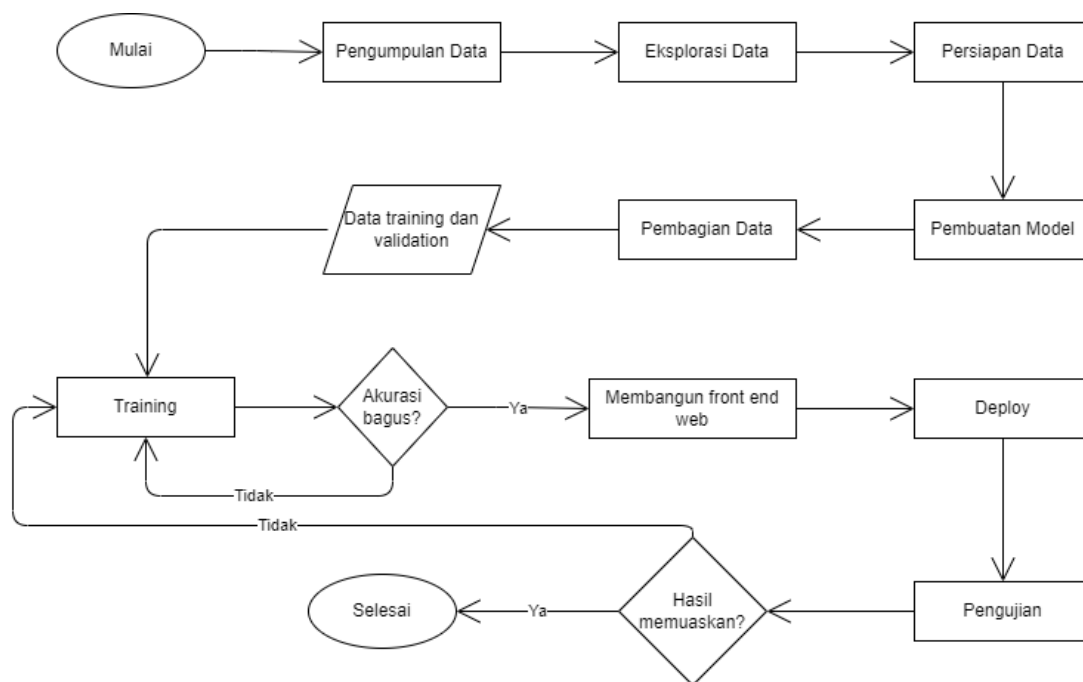
Tabel 3.2 Bahan yang digunakan dalam penelitian

No	Nama Bahan	Spesifikasi	Deskripsi
1	Data Informasi Akademik PSTI	Aturan – aturan dan surat seputar akademik PSTI UNILA	Data yang akan digunakan untuk training dalam model.

3.3. Tahapan Penelitian

Penelitian dengan judul “IMPLEMENTASI LONG SHORT-TERM MEMORY (LSTM) PADA CHATBOT INFORMASI AKADEMIK DI PROGRAM STUDI TEKNIK INFORMATIKA UNIVERSITAS LAMPUNG” 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 3.1 Flowchart Tahapan Penelitian

3.3.1. Pengumpulan dan Eksplorasi Data

Pelatihan model prediktif adalah tentang belajar dari data yang diambil. Oleh karena itu, memiliki data berkualitas tinggi merupakan prasyarat untuk setiap aktivitas pembelajaran mesin. Tahap ini adalah proses 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 di eksplorasi dengan cara memahami dan menganalisis data sehingga dapat menemukan pola untuk menghasilkan ide – ide dan pengetahuan yang akan berguna untuk pemodelan. 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 Program Studi Teknik Informatika Universitas Lampung.

3.3.2. Persiapan Data

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:

- 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.

3.3.3. Pembangunan dan Pelatihan Model

Tahap ini berkaitan dengan pembuatan model dari data yang telah disiapkan pada tahap sebelumnya. 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% untuk data validation dan 80% untuk data training. Tahap selanjutnya yaitu dilakukan penentuan 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 metrik evaluasi yaitu “accuracy”.

3.3.4. Deployment

Sebelum model yang dibuat diuji kepada pengguna, model harus dideploy dan hosting terlebih dahulu agar dapat di akses 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 untuk mempercantik halaman. Web yang dibangun hanya terdiri dari satu halaman yang berisi *bubble*. Ketika bubble tersebut di klik, maka akan tampil menu *chatting*. Pada menu tersebutlah pengguna dapat bertanya mengenai informasi akademik dengan menggunakan bahasa Indonesia.

3.3.5. Pengujian Model

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 mahasiswa 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 diluar dari 5 topik pertanyaan tersebut. Topik didapat dari dataset dengan cara di acak menggunakan bahas pemrograman python. Program python ini menghasilkan 20 file data uji yang tentunya pertanyaan yang diberikan kepada user tidak semuanya sama. User akan melakukan uji coba model dengan memasukkan pertanyaan sesuai dengan topik yang diberikan. Hasil dari pengujian yaitu akurasi. Akurasi dihitung dengan cara membagi jumlah jawaban benar dibagi jumlah pertanyaan kemudian 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.

BAB V

KESIMPULAN DAN SARAN

5.1. Kesimpulan

Berdasarkan hasil penelitian, dapat diperoleh kesimpulan sebagai berikut.

1. 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 yang mendapatkan 100% *accuracy* dan 98,5% *validation accuracy* serta 0,00229% *loss* dan 1,116% *validation loss* dengan hyperparameter terbaik yaitu learning rate 0,1, jumlah neuron output embedding sebanyak 128 dan jumlah neuron untuk LSTM layer sebanyak 256.
2. Model berhasil diimplementasikan pada tahap pengujian karena mendapatkan 99 jawaban benar dari 100 data uji yang diberikan. Dengan hasil tersebut, maka hasil dari pengujian mendapatkan akurasi sebesar 99%.
3. Karena hasil akurasi pada training dan akurasi pada pengujian yang tinggi, model ini layak untuk digunakan sebagai tempat mahasiswa – mahasiswi untuk bertanya seputar informasi akademik PSTI Unila.

5.2. Saran

Berdasarkan hasil penelitian, saran yang dapat digunakan untuk penelitian selanjutnya adalah sebagai berikut.

1. 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.

2. 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.
3. Dapat ditambahkan fitur chat dengan admin mengenai masalah yang tidak bisa dipecahkan oleh chatbot.

DAFTAR PUSTAKA

- [1] T. Harihayati dan L. Kurnia, "SISTEM PAKAR MENDIAGNOSA PENYAKIT UMUM YANG SERING DIDERITA BALITA BERBASIS WEB DI DINAS KESEHATAN KOTA BANDUNG," *Jurnal Komputer dan Informatika (KOMPUTA)*, vol. 1, p. 65, 2012.
- [2] A. S. Oktriwina, "NLP: Kecerdasan Buatan yang Bantu Komputer Pahami Bahasa Manusia," 2 February 2021. [Online]. Available: <https://glints.com/id/lowongan/natural-language-processing-adalah/#.YhTsm4hBxPY>. [Diakses 22 February 2022].
- [3] R. Stauffer, "Understanding & Implementing Chatbots at Your Agency," 20 April 2021. [Online]. Available: <https://blog.hawksoft.com/chatbots>. [Diakses 22 February 2022].
- [4] 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].
- [5] V. R. Prasetyo, N. Benarkah dan V. J. Chrisintha, "Implementasi Natural Language Processing Dalam Pembuatan Chatbot Pada Program Information Technology Universitas Surabaya," *TEKNIKA*, 2021.
- [6] I. M. Untari, "Chatbots and Government Communications in Covid-19 Pandemic," *JURNAL KOMUNIKASI INDONESIA*, vol. IX, 2020.
- [7] 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.
- [8] E. N. S. C. P. a. I. Afrianto, "Rancang Bangun Aplikasi Chatbot Informasi Objek Wisata Kota Bandung Dengan Pendekatan Natural Language Processing," *J. Ilm. Komput. dan Inform*, vol. 4, pp. 49-54, 2015.
- [9] 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].
- [10] Mulyono, "Identifikasi Chatbot dalam Meningkatkan Pelayanan Online," *Jurnal Informatika Ekonomi Bisnis*, 2021.
- [11] A. Y. Chandra, D. Kurniawan dan R. Musa, "Perancangan Chatbot Menggunakan Dialogflow Natural Language," *JURNAL MEDIA INFORMATIKA BUDIDARMA*, 2020.
- [12] L. Nurhayatunnufus, M. P. Ridha dan H. Maulid, ST., M.Sc, "LAPPYBOT: CHATBOT APPLICATION FOR INFORMATION ON SELECTING LAPTOP USING THE NATURAL LANGUAGE PROCESSING (NLP) METHOD," *e-Proceeding of Applied Science*, 2020.
- [13] S. Kusumadewi, *Artificial Inteligence (Teknik dan Aplikasinya)*, Yogyakarta: Graha Ilmu, 2003.

- [14] A. Kaplan dan M. Haenlein, *User Of The World, Unite! The Challenges and Opportunities Of Social Media*, Business Horizon, 2010.
- [15] W. K. & L. S. Härdle, *Applied multivariate statistical analysis* (4th ed.), Cham: Springer, 2015.
- [16] A. J. Dautel, W. K. Hardle, S. Lessmann dan H.-V. Seow, "Forex exchange rate forecasting using deep recurrent," *Springer*, pp. 69-96, 2020.
- [17] G. A. Royani Darma Nurfiti, "IMPLEMENTASI DEEP LEARNING BERBASIS TENSORFLOW UNTUK PENGENALAN SIDIK JARI," *Emitor: Jurnal Teknik Elektro*, vol. 18.
- [18] A. Oliver, "glints blog," glint, 7 Januari 2022. [Online]. Available: <https://glints.com/id/lowongan/deep-learning-adalah/#.Ys2ryXZBxPZ>. [Diakses 13 Juli 2022].
- [19] 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.
- [20] T. Mikolov, K. Chen, G. Corrado dan J. Dean, "Efficient Estimation of Word Representations in Vector Space," *arXiv preprint*, 2013a.
- [21] P. Werbos, "Backpropagation through time: What it does and how to do it," *Proceedings of the IEEE*, p. 78(10), 1550-1560.
- [22] I. B. Y. & C. A. Goodfellow, "Deep learning," *Cambridge: MIT Press.*, 2016.
- [23] J. S. S. Hochreiter, "Long Short-Term Memory". In: *Neural Comput., Massachusetts Institute of Technology*, vol. 9, pp. 1735-1780, 1997.
- [24] P. W. B. N. K. D. R. Tita Lattifia, "Model Prediksi Cuaca Menggunakan Metode LSTM," *JITTER- Jurnal Ilmiah Teknologi dan Komputer*, vol. 3, 2022.
- [25] Caugusta, "statisticalinterference," 1 Juni 2017. [Online]. Available: <https://statisticalinterference.wordpress.com/2017/06/01/lstms-in-even-more-excruciating-detail/>. [Diakses 1 Agustus 2022].
- [26] Y. Verma, "Analytics India Magazine," 19 September 2021. [Online]. Available: <https://analyticsindiamag.com/a-complete-understanding-of-dense-layers-in-neural-networks/>. [Diakses 25 July 2022].
- [27] J. W. & Sons, *Keras to Kubernetes®: The Journey of a Machine Learning Model to Production*, New York: Inc. Published, 2019.
- [28] M. K. S. A. a. F. F. R. Talib, *Text Mining: Techniques, Applications and Issues*, Int. J. Adv. Comput. Sci. Appl., 2016.
- [29] T. Lata, "Towards Data Science," 13 November 2018. [Online]. Available: <https://towardsdatascience.com/getting-your-text-data-ready-for-your-natural-language-processing-journey-744d52912867>. [Diakses 26 Juli 2022].
- [30] B. Klein, "python-course.eu," 17 Februari 2022. [Online]. Available: <https://python-course.eu/machine-learning/natural-language-processing-with-python.php>. [Diakses 26 Juli 2022].

- [31] S. G. a. P. O. Kristensson, "Neural Networks for Text Correction and Completion in Keyboard Decoding," *JOURNAL OF LATEX CLASS FILE*, vol. 14, 2015.
- [32] 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.
- [33] V. Dev, "Analytics India Magazine," 25 Agustus 2021. [Online]. Available: <https://analyticsindiamag.com/when-to-use-one-hot-encoding-in-deep-learning/>. [Diakses 13 Juli 2022].
- [34] D. P. a. J. L. B. Kingma, "Adam: A Method for Stochastic Optimization," *ICLR*, pp. 1-15, 2005.
- [35] C. Olah, "Understanding LSTM Networks," 27 Agustus 2015. [Online]. Available: <http://colah.github.io/posts/2015-08-Understanding-LSTMs/>. [Diakses 22 February 2022].