SKRIPSI

KLASIFIKASI TEKS SENTIMEN REVIEW E-COMMERCE MENGGUNAKAN ALGORITMA LONG SHORT-TERM MEMORY (LSTM)

Disusun dan diajukan oleh

NURFADILA FIRDANI SALAM

H071171014

PROGRAM STUDI SISTEM INFORMASI

DEPARTEMEN MATEMATIKA

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM

UNIVERSITAS HASANUDDIN

MAKASSAR

30 APRIL 2021

KLASIFIKASI TEKS SENTIMEN REVIEW E-COMMERCE MENGGUNAKAN ALGORITMA LONG SHORT-TERM MEMORY (LSTM)

SKRIPSI

Diajukan sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer pada Program Studi Sistem Informasi Departemen Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Hasanuddin

NURFADILA FIRDANI SALAM

H071171014

PROGRAM STUDI SISTEM INFORMASI

DEPARTEMEN MATEMATIKA

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM

UNIVERSITAS HASANUDDIN

MAKASSAR

30 APRIL 2021

LEMBAR PERNYATAAN KEOTENTIKAN

Yang bertanda tangan di bawah ini:

Nama	: Nurfadila Firdani Salam
NIM	: H071171014
Program Studi	: Sistem Informasi
Jenjang	: S1

Menyatakan dengan ini bahwa karya tulisan saya berjudul

KLASIFIKASI TEKS SENTIMEN REVIEW E-COMMERCE MENGGUNAKAN ALGORITMA LONG SHORT-TERM MEMORY (LSTM)

Adalah benar hasil karya saya sendiri bukan merupakan pengambilan alihan tulisan orang lain dan belum pernah dipublikasikan dalam bentuk apapun.

Apabila dikemudian hari terbukti atau dapat dibuktikan bahwa sebagian atau keseluruhan skripsi ini merupakan hasil karya orang lain, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Makassar, 30 April 2021

Nurfadila Firdani Salam

NIM. H071171014

KLASIFIKASI TEKS SENTIMEN REVIEW E-COMMERCE MENGGUNAKAN ALGORITMA LONG SHORT-TERM MEMORY (LSTM)

Disusun dan diajukan oleh:

NURFADILA FIRDANI SALAM

H071171014

Telah dipertahankan di hadapan Panitia Ujian yang dibentuk dalam rangka Penyelesaian Studi Program Sarjana pada Program Studi Sistem Informasi Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Hasanuddin dan dinyatakan telah memenuhi syarat kelulusan

Menyetujui,

Pembimbing Utama	**Pembimbing Pertama**
Dr. Hendra, S.Si., M.Kom.	**Supri Bin Hj Amir, S.Si., M.Eng.**
NIP. 19760102 200212 1 001	**NIP. 19880504 201903 1 012**

Ketua Program Studi

Dr. Muhammad Hasbi, M.Sc

NIP.196307201989031003

HALAMAN PENGESAHAN

Skripsi ini diajukan oleh:

Nama	: Nurfadila Firdani Salam
NIM	: H071171014
Program Studi	: Sistem Informasi
Judul Skripsi	: Klasifikasi teks sentimen review e-commerce menggunakan algoritma Long Shoer-Term Memory (LSTM)

Telah berhasil mempertahankan di hadapan dewan penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Sarjana Komputer pada Program Studi Sistem Informasi Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Hasanuddin.

DEWAN PENGUJI

		Tanda Tangan
1. Ketua	: Dr. Hendra, S.Si., M.Kom	(.....)
2. Sekretaris	: Supri Bin Hj. Amir, S.Si., M.Eng.	(.....)
3. Anggota	: Dr. Eng. Armin Lawi, M.Eng.	(.....)
4. Anggota	: Edy Saputra, S.Si., M.Si.	(.....)

Ditetapkan di : Makassar
Tanggal : 30 April 2021

KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat Allah SWT, karena atas berkat dan rahmat-Nya sehingga penulis dapat menyelesaikan skripsi yang berjudul **"Klasifikasi Text Sentiment Review E-commerce Menggunakan Algoritma Long Short-Term Memory (LSTM).** Penulisan skripsi ini dilakukan dalam rangka memenuhi salah satu syarat untuk memperoleh gelar Sarjana Komputer pada Program Studi Sistem Informasi Departemen Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Hasanuddin.

Pada kesempatan ini penulis ingin mengucapkan terima kasih yang sebesar-besarnya kepada kedua orang tua tercinta, Ayahanda **Abd. Salam Hanafi** dan Almarhumah Ibunda **Badrah Lahami** yang telah mendidik dengan penuh kesabaran, terimakasih telah mencurahkan kasih sayang yang tak pernah putus, kesungguhan dalam memberikan dukungan moril serta tak kenal lelah dalam memanjatkan doa serta memberikan nasihat dan motivasi kepada penulis selama menjalani proses pendidikan. Untuk kakak-kakak penulis **Ridwan S**, **Welham Prasetya**, **Winda Anugerah**, **Aidil Kurniawan S** dan Adik **Muh. Miqdad Hilal** serta keluarga yang senantiasa memberikan dukungan dan doa, terima kasih atas segala perhatian yang telah kalian berikan kepada penulis. Tugas akhir ini hanya setitik kebahagiaan kecil yang bisa penulis persembahkan kepada kalian.

Penghargaan yang tulus dan ucapan terimakasih dengan penuh keikhlasan juga penulis ucapkan kepada :

1. Pemerintah melalui **Direktorat Jendral Pembelajaran dan Kemahasiswaan (Ditjen Belmawa)**, **Kementrian Riset, Teknologi dan Perguruan Tinggi (Kemenristek Dikti)** yang telah membantu pembiayaan penulis selama perkuliahan melalui program **Bidikmisi**.
2. Bapak **Dr Hendra, S.Si M.Kom** dan Bapak **Supri bin Hj Amir, S.Si, M.Eng** atas kesediaan, kesabaran, dan kesetiaan untuk membimbing dan membagi ilmunya kepada penulis dalam penyusunan skripsi ini.
3. Bapak **Dr. Eng. Armin Lawi, M.Eng** dan Bapak **Edy Saputra, S.Si., M.Si** atas kesediaannya untuk memberikan saran dan arahan kepada penulis dalam penyususunan skripsi ini.

4. Kepada seseorang yang akan berganti nama, sehingga tidak ingin dicantumkan, terima kasih atas bantuan anda selama pengerjaan skripsi ini, tanpa anda penulis tidak akan sehebat ini. Sekali lagi terima kasih.
5. **Dosen pengajar Jurusan Matematika** yang telah membekali ilmu kepada penulis selama menjadi mahasiswa di Jurusan Matematika. Serta Staf Jurusan Matematika yang telah membantu banyak dalam perkuliahan.
6. Rektor Universitas Hasanuddin, Ibu **Prof. Dr. Dwia Aries Tina Pulubuhu** beserta jajarannya.
7. Dekan Fakultas Matematika dan Ilmu Pengetahuna Alam (FMIPA), Bapak **Dr.Eng. Amiruddin,M.Si** beserta jajarannya.
8. Saudara **DISKRIT 2017.** Ketua Angkatan **Ihzanul Wajdi,** saudara **Nur Faathir, Aris Akhyar, Ahmad Ali Winandar,** anggota **GengBel** (Nur Khairunisa, Tasya Wiraz, Khandy Dilsab, Miftahulljanna, Indiralkadry, Krisdayanti, Eka kurnia)**,** serta seluru anggota diskrit, terima kasih telah menjadi rumah ke dua penulis.
9. Seluruh **Anggota Himatika FMIPA Unhas** terima kasih telah menjadi keluarga dan memberikan kenangan yang tidak akan penulis lupakan.
10. Seluruh **Angkatan 2017 SISTEM INFORMASI**, terkhusus kepada **Muhammad Fitrah, A. Amaliah Dwi Ayu, Rigel, Kennedy, Muh Farhan, Muh Amir, Mufliun Naim, Khawaritzmi, Rafly Ahmad Mubin, Muhaimin Anwar**, **Restu Adi Akbar** terima kasih telah banyak membantu mengerjakan tugas dan mengajari materi selama perkuliahan**,** kepada **Gepeng** (**Geby Nionsi, Azzahrah Mubarikah, Eka Fitriani, Siti Rabiatul, Mutiah Amanah, Eka Kurnia, Nur Khairunisa, Fadillah Putri Taha**) terima kasih sudah hadir sejak kita maba sampai sekarang, kalian banyak membantu selama kuliah penulis.
11. Saudara-saudari **MIPA 2017** tanpa terkecuali.
12. Kepada sister **For.efcef (Elsya Mayora, Chindy Claudia, Ema Safira, Muliati)** terima kasih atas segala perhatiannya, kalian bukan lagi sahabat melainkan saudara tak sedarah penulis.

13. Kepada **Jalangkote Paris** terima kasih telah hadir dalam ide penulis, besar harapan penulis untuk bisa menjadikan usaha ini semakian besar.
14. Semua pihak yang tidak dapat disebutkan satu persatu, atas segala bentuk kontribusi, partisipasi, serta motivasi yang diberikan kepada penulis selama ini. Semoga apa yang telah diberikan akan dilipatgandakan oleh Allah.

Makassar, 30 April 2021

Nurfadila Firdani Salam
NIM. H071171014

PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS

Sebagai sivitas akademik Universitas Hasanuddin, saya yang bertanda tangan di bawah ini:

Nama	: Nurfadila Firdani Salam
NIM	: H071171014
Programa Studi	: Sistem Informasi
Departemen	: Matematika
Fakultas	: Matematika dan Ilmu Pengetahuan Alam
Jenis Karya	: Skripsi

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Hasanuddin **Hak Prediktor Royalti Noneksklusif (*Non-exclusive Royalty-Free Right*)** atas tugas akhir saya yang berjudul:

Klasifikasi Teks Sentimen Review E-commerce Menggunakan Algoritma Long Short-Term Memory (LSTM)

beserta perangkat yang ada (jika diperlukan). Terkait dengan hal diatas, maka pihak universitas berhak menyimpan, mengalih-media/format-kan, mengelola dalam bentuk pangkalan data (database), merawat, dan memublikasikan tugas akhir saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian surat pernyataan ini saya buat dengan sebenarnya.

Dibuat di Makassar pada 30 April 2021

Yang menyatakan

(Nurfadila Firdani Salam)

ABSTRAK

Kehadiran teknologi internet pada dunia global saat ini sungguh memberikan dampak yang sangat besar untuk segala bidang, salah satunya bidang penjualan. E-commerce merupakan sarana pendukung yang kuat sehingga dapat membantu perusahaan dalam mencapai tujuannya, penjualan online menyediakan fitur review untuk pelanggan sebagai sarana untuk menyampaikan kekurangan atau kelebihan dari suatu produk yang ditawarkan. Dalam Review terdapat sentiment yang merupakan suatu ungkapaan perasaan pelanggan terhadap produk yang dibeli pada suatu perusahaan. Perusahaan ini membutuhkan alat untuk mengklasifikasikan sentimen secara otomatis. Long short-term memory (LSTM) merupakan bagian dalam algoritma Deep learning yang dapat mengklasifikasi data berupa text sequence, seperti review pelanggan. Pada penelitian ini, peneliti menggunakan model Algoritma Long short-term memory (LSTM) yaitu Vanilla LSTM dan Bidirectional LSTM untuk mengklasifikasi review text e-commerce. Hasil dari penelitian tentang analisis sentimen pada data woman's e-commerce clothing review menggunakan Bidirectional LSTM mampu menghasilkan nilai accuracy dan F1-Score yang lebih baik dibandingkan dengan Vanilla LSTM pada epoch 5 dengan niai Accuracy 87.8%, f1-score negative 51.8%, f1-score neutral 92.6% dan f1-score positive 73.4% .

Kata kunci: Review, Long Short Term Memory, Vanilla LSTM, Bidirectional LSTM

ABSTRACT

The rise of internet technology in current globalization era has a huge impact on all fields, one of which is sales. E-commerce is a strong supporting tool so that it can help companies achieve their goals, online sales provide a review feature for customers as a means of conveying the advantages or disadvantages of a product being offered. In these reviews, there are sentiments which is an expression of how a customer feels about the products purchased at the company. These companies need a tool to automatically classify the sentiments. Long short-term memory (LSTM) is a part of deep learning that can classify data in the form of text sequences, namely the customers review. In this study, the researcher used the Long short-term memory (LSTM) algorithm, which are Vanilla LSTM and Bidirectional LSTM to classify e-commerce text reviews. The results of the research on sentiment analysis on the data of woman's e-commerce clothing review using Bidirectional LSTM were able to produce better accuracy and F1-scores compared to Vanilla LSTM at epoch 5 with an Accuracy of 87.8%, f1-score negative of 51.8%, f1- neutral score of 92.6% and f1-score positive of 73.4%.

Keywords : Review, Long Short Term Memory, Vanilla LSTM, Bidirectional LSTM

DAFTAR ISI

LEMBAR PERNYATAAN KEOTENTIKAN ii
HALAMAN PERSETUJUAN PEMBIMBING iii
HALAMAN PENGESAHAN iv
KATA PENGANTAR vi
PERSETUJUAN PUBLIKASI KARYA ILMIAH vii
ABSTRAK x
ABSTRACT xi
DAFTAR GAMBAR xv
DAFTAR TABEL xvii
BAB I PENDAHULUAN 1
1.1 Latar Belakang 1
1.2 Rumusan Masalah 2
1.3 Tujuan 3
1.4 Manfaat 3
1.5 Batasan Masalah 3
1.6 Organisasi Skripsi 3
BAB II TINJAUAN PUSTAKA 5
2.1 Penelitian Terkait 5
2.2 Deep Learning 6
2.2.1 Multilayer Perception 8
2.2.2 Fungsi Aktivasi 9
2.2.3 Backpropagation 11
2.2.4 Recurrent Neural Network 12
2.2.5 Long Short-Term Memory 14

2.2.6 Bidirectional LSTM 17
2.3 Multi Kelas Klasifikasi 18
2.4 Multiclass Confusion Matrix 20
2.5 Early Stopping 22
2.6 Tensorflow 23
BAB III METODE PENELITIAN 24
3.1 Sumber Data 24
3.2 Waktu dan Tempat 24
3.3 Tahapan Penelitian 24
3.3.1 Tahapan Pra Penelitian (persiapan) 24
3.3.2 Penelitian 24
3.4 Diagram Alur Penelitian 25
BAB IV HASIL DAN PEMBAHASAN 27
4.1 Variabel Data 27
4.2 Preprocessing 28
4.2.1 Lowercase 29
4.2.2 Tokenizer 30
4.2.3 Removing Stop Words 30
4.2.4 Text to Sequence 31
4.2.5 Word Index 31
4.2.6 Padding 32
4.2.7 Membuat y dari review text 32
4.3 Confusion Matrix 35
4.4 Modeling Multi Kelas Klasifikasi 36
4.4.1 Vanilla LSTM 36

4.4.2 Bidirectional LSTM 38
4.4.3 Perbandingan Multi Kelas Klasifikasi 39
4.5 Perbandingan Klasifikasi Biner 45
4.6 Word Cloud 48
4.7 Analisis Kinerja 52
4.7.1 Peforma model Multi Kelas Klasifikasi 52
4.7.2 Peforma Model Klasifikasi Biner 53
BAB V KESIMPULAN DAN SARAN 54
5.1 Kesimpulan 54
5.2 Saran 55
DAFTAR PUSTAKA 56
LAMPIRAN 1 59
LAMPIRAN 2 66
LAMPIRAN 3 69

DAFTAR GAMBAR

Gambar 2.1 Ilustrasi Deep Learning 7
Gambar 2.2 Arsitektur Multilayer Perceptron 9
Gambar 2.3 Grafik Fungsi ReLU 10
Gambar 2.4 Hyperbolic Tangent 10
Gambar 2.5 Arsitektur RNN 13
Gambar 2.6 Persamaan Dasar RNN 13
Gambar 2.7 Visualisasi Persamaan Dasar RNN 14
Gambar 2.8 Arsitektur LSTM 14
Gambar 2.9 Forget Gate 15
Gambar 2.10 Input Gate 16
Gambar 2.11 Memory Cell State Gate 17
Gambar 2.12 Output Gate 17
Gambar 2.13 Arsitektur Bidirectional LSTM (Graves, dkk 2013) 18
Gambar 2.14 Idealized training and validation error curves. Vertical: errors; horizontal: time 22
Gambar 3.1 Diagram Alur Penelitian 26
Gambar 4.1 *Confusion Matrix Vanilla LSTM* 35
Gambar 4.2 *Confusion Matrix Bidirectional LSTM* 36
Gambar 4.3 *Vanilla LSTM Accuracy* 37
Gambar 4.4 *Bidirectional LSTM Accuracy* 38
Gambar 4.5 *Training* dan *validation Accuracy* 39
Gambar 4.6 Vanilla Precision (Training dan Validation) 39
Gambar 4.7 *Bidirectional Precision* (*Training* dan *Validation*) 40
Gambar 4.8 Vanilla Recall (Training dan Validation) 40
Gambar 4.9 *Bidirectional Recall (Training* dan *Validation*) 41
Gambar 4.10 *Vanilla f1 (Training* dan *Validation)* 41
Gambar 4.11 *Bidirectional f1 (Training* dan *Validation*) 42
Gambar 4.12 *Training Score* 43
Gambar 4.13 *Validation Score* 44

Gambar 4.14 *Training Score* 46
Gambar 4.15 *Validation Score* 47
Gambar 4.16 AUROC 48
Gambar 4.17 *Word cloud bigram sentiment positive* 49
Gambar 4.18 *Word cloud bigram sentiment neutral* 50
Gambar 4.19 *Word cloud bigram sentiment negative* 51

DAFTAR TABEL

Tabel 2.1 Contoh Dataset Play Tennis (UCI Mechine learning depository) 19
Table 4.1 *Variabel Data*..... 27
Tabel 4.2 *Before Missing Value*..... 28
Tabel 4.3 *After Missing Value*..... 29
Tabel 4.4 *After Lower Case* 29
Tabel 4.5 *After Tokenaizer*..... 30
Tabel 4.6 After Removing Stop Words..... 30
Tabel 4.7 After Text to Sequence 31
Tabel 4.8 After Word Index..... 31
Tabel 4.9 Padding..... 32
Tabel 4.10 Membuat y dari *review text*..... 33
Tabel 4.11 Model Sequential Vanilla LSTM..... 37
Tabel 4.12 Model Sequential BiLSTM..... 38
Table 4.13 Model Sequential Vanilla LSTM..... 45
Table 4.14 Model Squential BiLSTM..... 45
Table 4.15 Peforma Model Pada Epoch Terbaik 52
Table 4.16 Peforma model pada epoch terbaik..... 53

BAB I
PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi informasi khususnya internet semakin populer, tidak hanya dinikmati oleh kalangan profesional namun juga sudah merambat ke semua lapisan masyarakat. Kehadiran teknologi internet pada dunia global saat ini sungguh memberikan dampak yang sangat besar untuk segala bidang, salah satunya bidang penjualan. Oleh karena itu, pebisnis semakin banyak menggunakan media internet (Agustriana, 2018). *E-commerce* merupakan sarana pendukung yang kuat sehingga dapat membantu perusahaan dalam mencapai tujuannya, seperti dalam pencapaian pendapatan perusahan (Septiyan, 2016). Maka dari itu dalam penjualan online menyediakan *fitur review* untuk pelanggan sebagai sarana untuk menyampaikan kekurangan atau kelebihan dari suatu produk yang ditawarkan, sehingga perusahaan dapat memberikan yang terbaik dari penjualannya.

E-commerce biasanya menyediakan produk yang dilengkapi dengan *review* serta pemberian sentimen yang diberikan oleh konsumen. *Review* yang dituliskan pun bermacam-macam, mulai dari kualitas hingga kekurangan dari perusahaan tersebut (Lila Dini, 2018). Dalam penelitian ini klasifikasi teks sangat dibutuhkan untuk menghitung sentimen pada *review* dari konsumen penjualan pakaian wanita. Klasifikasi teks adalah proses pengelompokan teks ke dalam kelas yang berbeda, dalam tahapannya tiap teks menunjuk pada satu kelas tertentu dan dibutuhkan proses untuk menggali informasi dari teks tersebut (Algina dkk, 2012).

Klasifikasi teks termasuk bagian penting dalam *Natural Language Processing* (Pemrosesan Bahasa Alami) dengan banyak penerapan seperti sentimen analisis, pencarian informasi, perankingan, indexing dan klasifikasi dokumen (Li dkk, 2018). Model klasifikasi teks umumnya dibagi menjadi dua kategori: *machine learning* dan *deep learning*. Baru-baru ini, model berbasis jaringan saraf menjadi semakin populer. Model-model ini mencapai kinerja yang sangat baik dalam praktiknya, cenderung relatif lambat baik pada saat pelatihan maupun pengujian, membatasi penggunaannya pada kumpulan data yang sangat besar. Beberapa

penelitian terbaru menunjukkan bahwa keberhasilan *deep learning* tentang klasifikasi teks sangat tergantung pada *efektivitas word embedding* (Winda, 2020).

Salah satu metode *deep learning* yaitu RNN dengan penerapan arsitektur *Long Short-Term Memory* (LSTM). RNN dapat menggunakan representasi kata yang terdistribusi dengan terlebih dahulu mengubah token yang terdiri dari setiap teks menjadi vektor, yang membentuk matriks. Sedangkan, LSTM dikembangkan untuk menangani masalah *exploding* dan *vanishing gradient* yang dapat dihadapi saat melatih RNN tradisional (Yan, 2018). *Long Short-Term Memory* (LSTM), yang merupakan struktur lanjutan dari *Recurrent Neural Network* (RNN) yang memiliki mekanisme gate termasuk *cell memory*. LSTM memiliki ide utama untuk mengingat nilai-nilai bobot menggunakan *memory cell*. Pada pengaplikasiannya, LSTM memiliki masalah kompleksitas yang sangat tinggi. Hingga saat ini, penelitian terkait pengembangan jaringan LSTM sudah banyak dilakukan.

Winda dkk, 2020 menerapkan metode LSTM dan Word2Vec hanya pada salah satu model LSTM. Pada penelitian ini menggunakan model LSTM yaitu *Vanilla Long Short-Term Memory* dan *Bidirectional Long Short-Term Memory* menggunakan parameter *accuracy, recall, precision* dan F1-*score*.

1.2 Rumusan Masalah

Berdasarkan latar belakang di atas dapat dirumuskan beberapa masalah sebagai berikut:

1. Bagaimana model *Long Short-Term Memory* mengklasifikasikan *sentimen* pada *review e-commerce*?
2. Bagaimana mengevaluasi kinerja model *LSTM Bidirectional* dan model *LSTM Vanilla* pada prediksi *sentimen review e-commerce* berdasarkan tingkat *accuracy, recall, precision* dan F1-*score*?
3. Bagaimana mengetahui kata-kata yang berpengaruh pada sentimen konsumen hasil dari prediksi *sentimen review* pada perusahaan *Women's E-commerce Clothing Review*?

1.3 Tujuan

Tujuan dari penelitian ini adalah sebagai berikut:

1. Untuk mengetahui cara kerja model *Long Short-Term Memory* dalam mengklasifikasikan sentimen pada *review e-commerce.*
2. Untuk mengevaluasi kinerja model LSTM (*Long Short-Term Memory)* yang digunakan pada penelitian ini terhadap prediksi *sentimen review e-commerce* berdasarkan tingkat *accuracy, recall, precision* dan F1-*score*.
3. Untuk mengetahui kata-kata yang berpengaruh pada sentimen konsumen pada perusahaan *Women's E-commerce Clothing Review*.

1.4 Manfaat

Hasil dari penelitian ini diharapkan dapat memberikan manfaat sebagai berikut:

1. Pengembangan LSTM dapat menghasilkan *accuracy, recall, precision* dan F1-*score* dari proses *training* dan *testing* yang lebih optimal.
2. Diharapkan dengan mengklasifikasi multi-kelas teks dengan menggunakan algoritma *LSTM (Long Short-Term Memory)* dapat membantu perusahaan *e-commerce* mengetahui prediksi *sentiment review* pada perusahaan sehingga dapat membuat produk yang dijual semakin baik.
3. Diharapkan dengan melihat hasil prediksi *sentiment review* konsumen yang akan berbelanja pada perusahaan tersebut dapat memilih produk yang baik.

1.5 Batasan Masalah

Adapun batasan masalah yang dilakukan agar pengerjaan Tugas Akhir ini tidak terlalu luas, Maka masalah yang dibahas dibatasi pada:

1. Pada penelitian ini prediksi dilakukan dengan menggunakan model *LSTM bidirectional* dan *LSTM vanilla.*
2. Pada penelitian ini menggunakan dataset *Women's E-commerce Clothing Review.*

1.6 Organisasi Skripsi

Penulisan skripsi ini diorganisasikan sebagai berikut:

Pada Bab I Akan membahas tentang hal yang melatar belakangi secara umum dalam pembuatan penulisan skripsi ini.

Pada Bab II Akan membahas tentang materi – materi yang mendukung penulisan skripsi ini, diantaranya tentang *Deep Learning, Recurrent Neural Network, Long Short-Term Memory*, dan Klasifikasi teks multi kelas.

Pada Bab III Akan membahas tentang hasil dan pembahasan dari penulisan skripsi ini.

Pada Bab IV Berupa hasil dan pembahasan dari penelitian yang dilakukan.

Pada Bab V Berupa penutup yang akan membahas tentang kesimpulan dari hasil penelitian ini dan saran – saran yang dapat membangun pengembangan skripsi ini menjadi lebih baik.

BAB II
TINJAUAN PUSTAKA

2.1 Penelitian Terkait

Dalam penelitian Winda, dkk (2020) yang berjudul *Multilabel Text Classification in News Articles Using Long-Term Memory with Word2Vec*, untuk mendapatkan model yang optimal pada klasifikasi teks dilakukan percobaan *trial* dan *error* menggunakan LSTM dengan *fitur word embedding Word2Vec* 300 dimensi. Dengan *tuning hyperparameter* dan membuat perbandingan delapan model LSTM yang diusulkan dengan dataset skala besar, dan untuk menunjukkan bahwa LSTM dengan fitur Word2Vec dapat mencapai kinerja yang baik dalam klasifikasi teks. Hasil penelitian menunjukkan bahwa klasifikasi teks menggunakan LSTM dengan fitur Word2Vec memperoleh akurasi tertinggi pada model kelima dengan 95,38%, sedangkan rata-rata nilai *precision, recall,* dan F1-*score* adalah 95%.

Dalam penelitian Alejandro, dkk (2020) yang berjudul *Word-Class Embeddings for Multi Class Tect Classification* menunjukkan bukti empiris bahwa WCE menghasilkan peningkatan yang konsisten dalam akurasi klasifikasi multikelas, menggunakan empat arsitektur saraf populer dan enam digunakan secara luas dan tersedia untuk umum dataset untuk klasifikasi teks multikelas.

Dalam penelitian Adam, dkk (2018) yang berjudul *Memristive LSTM network hardware architecture for time-series predictive modeling problems*, menurut mereka *Long Short Term Memory* digunakan untuk mengatasi masalah perkiraan deret waktu. Semua simulasi dilakukan menggunakan teknologi TSMC 0.18 µm CMOS dan HP memristor model. LSTM berbasis tegangan mengurangi *overhead* mengkonversi antara arus dan tegangan yang secara signifikan mengurangi luas dan kompleksitas sirkuit. Selain itu, itu membuat mudah untuk menyesuaikan tegangan *node* perantara sesuai keinginan nilai tegangan selama pembangunan sirkuit secara bertahap. Dari hasil simulasi yang diperoleh, terlihat jelas bahwa prediksi tersebut hasil sangat cocok antara perangkat keras dan perangkat lunak implementasi.

Waktu simulasi selanjutnya dapat dikurangi dengan menyesuaikan sinyal tegangan kontrol dengan memastikan bahwa ada tidak ada masalah konvergensi.

Dalam penelitian Satyo (2020) yang berjudul Analisis Data Time Series Menggunakan *Long Short Term Memory* (LSTM) dan *Autocorrelation Integrated Moving Average* (ARIMA) dalam Bahasa Python, memprediksi data deret waktu menggunakan dua metode, metode pertama yang biasa digunakan adalah statistik *Autocorrelation Integrated Moving Average* (Model ARIMA) dan metode kedua relatif baru, yaitu *Long Short Term Memory* (LSTM). Hasil yang diperoleh nilai RMSE terkecil untuk ARIMA adalah 2% dan nilai RMSE terkecil untuk LSTM adalah 1%. Hasil penelitian ini menggunakan 7 kombinasi proses transformasi, dapat meningkatkan level akurasi prediksi dari ARIMA dan LSTM. Dimana akurasi mesin pembelajaran LSTM oleh menggunakan data saham Telkom memiliki akurasi yang lebih tinggi dari ARIMA.

Dalam penelitian Luthfi, dkk (2020) yang berjudul Prediksi Data Transaksi Penjualan Time Series Menggunakan Regresi, melakukan uji coba prediksi menggunakan teknik *Machine Learning* dengan metode *Regresi Long Short Term Memory* (LSTM). Teknik yang diusulkan diuji coba menggunakan dataset penjualan produk "X" dari PT. Metiska Farma dengan parameter kinerja *Root Mean Squared Error* (RMSE) dan *Mean Absolute Percentage Error* (MAPE). Hasil penelitian ini berupa nilai rata – rata evaluasi error dari pemodelan data *training* dan data *testing*. Di mana hasil menunjukan bahwa Regresi LSTM memiliki nilai prediksi penjualan dengan evaluasi model melalui RMSE sebesar 286.465.424 untuk data training dan 187.013.430 untuk data testing. Untuk nilai MAPE sebesar 787% dan 309% untuk data *training* dan data *testing* secara berurut.

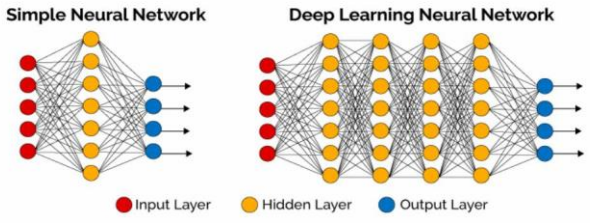
2.2 Deep Learning

Machine learning (ML), bagian dari kecerdasan buatan (*artificial intelligence*), berupa metode untuk mengoptimalkan performa dari sistem dengan mempelajari data sampel atau data histori (Alpaydin, 2010). *Machine learning* merupakan serangkaian teknik yang dapat membantu dalam menangani dan memprediksi data yang sangat besar dengan cara mempresentasikan data-data tersebut dengan algoritma pembelajaran (Danukusumo, 2017).

Deep learning merupakan salah satu bidang dari *machine learning* yang memanfaatkan jaringan syaraf tiruan untuk implementasi permasalahan dengan dataset yang besar. Teknik *deep learning* memberikan arsitektur yang sangat kuat untuk *supervised learning*. Dengan menambahkan lebih banyak lapisan maka model pembelajaran tersebut bisa mewakili data citra berlabel dengan lebih baik. Pada *machine learning* terdapat teknik untuk menggunakan ekstraksi fitur dari data pelatihan dan algoritma pembelajaran khusus untuk mengklasifikasi citra maupun untuk mengenali suara. Namun, metode ini masih memiliki beberapa kekurangan baik dalam hal kecepatan dan akurasi.

Aplikasi konsep jaringan syaraf tiruan yang dalam (banyak lapisan) dapat ditangguhkan pada algoritma *machine learning* yang sudah ada sehingga komputer sekarang bisa belajar dengan kecepatan, akurasi, dan skala yang besar. Prinsip ini terus berkembang hingga *deep learning* semakin sering digunakan pada komunitas riset dan industri untuk membantu memecahkan banyak masalah data besar seperti *Computer vision, Speech recognition, dan Natural Language Processing.*

Feature engineering adalah salah satu fitur utama dari *deep learning* untuk mengekstrak pola yang berguna dari data yang akan memudahkan model untuk membedakan kelas. *Feature engineering* juga merupakan teknik yang paling penting untuk mencapai hasil yang baik pada tugas prediksi. Namun, sulit untuk dipelajari dan dikuasai karena kumpulan data dan jenis data yang berbeda memerlukan pendekatan teknik yang berbeda juga (Danukusumo, 2017).



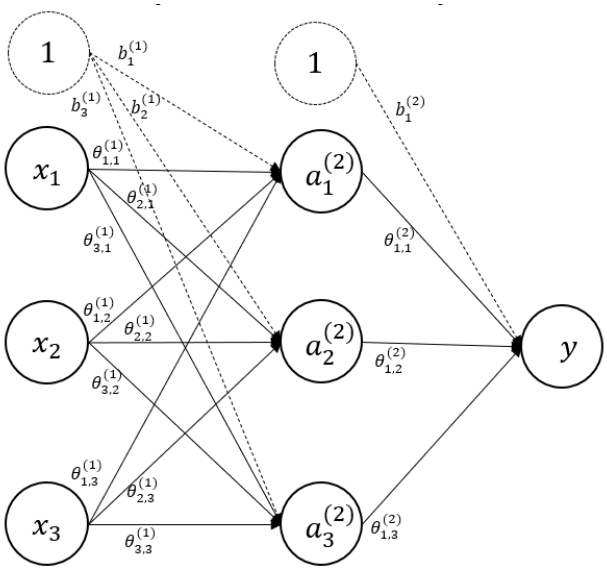
Gambar 2.1 Ilustrasi Deep Learning

Ilustrasi Deep Learning dapat dilihat pada gambar 2.1 terdapat 4 layer dan tiap *layer* mempunyai jumlah *node* yang berbeda-beda. *Deep learning* adalah jaringan yang terdiri dari beberapa *layer*. *Layer-layer* tersebut merupakan kumpulan dari *node-node*. Sebuah *node* hanya sebagai tempat perhitungan terjadi. Sebuah *node* input digabung dengan *weight* (bobot), setelah input dan *weight* ini dijumlahkan dan jumlahnya melewati tahapan yang disebut fungsi aktivasi *node*, untuk menentukan seberapa sejauh sinyal yang berlangsung lebih lanjut melalui jaringan, hal tersebut mempengaruhi hasil akhir (Rizky, 2020).

2.2.1 Multilayer Perception

Ada tiga unit yang dapat dibedakan dari *Multilayer Perceptron* yakni: Lapisan input (*input layer*), lapisan tersembunyi (*hidden layer*), lapisan luaran (*output layer*)(Wikipedia, 2020).

1. Lapisan masukan (*input layer*) terdiri dari *neuron* yang menerima data masukan dari variabel X. Semua *neuron* pada lapis ini dapat terhubung ke *neuron* pada lapisan tersembunyi atau langsung ke lapisan luaran jika jaringan tidak menggunakan lapisan tersembunyi.
2. Lapisan tersembunyi (*hidden layer*) terdiri dari neuron yang menerima data dari lapisan masukan.
3. Lapisan luaran (*output layer*) terdiri dari *neuron* yang menerima data dari lapisan tersembunyi atau langsung dari lapisan masukan yang nilai luarannya melambangkan hasil kalkulasi dari X menjadi nilai Y.



Gambar 2.2 Arsitektur Multilayer Perceptron

Pada gambar 2.2 merupakan Arsitektur Multilayer Perception (MLP) dengan keterangan sebagai berikut:

x : Input vektor x_1, x_2, x_3

θ: Matriks Bobot $\theta_{1,1}$, $\theta_{2,1}$ $\theta_{3,1}$ …

b: Bias $b_1{}^{(1)}$ $b_1{}^{(2)}$

a : Vektor nilai aktivasi

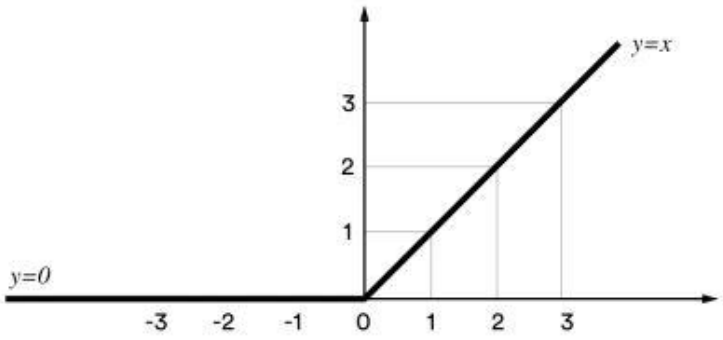
Y: Nilai output hasil perhitungan dari input oleh suatu fungsi aktivasi.

2.2.2 Fungsi Aktivasi

Fungsi aktivasi adalah untuk memberikan kemampuan *network* agar dapat melakukan tugas non-linear. Tanpa fungsi aktivasi, *neural network* hanyalah kombinasi operasi linear yang hanya dapat melakukan tugas-tugas yang linear pula. Padahal kebanyakan kasus nyata di lapangan merupakan kasus non-linear (Umam, 2018). Salah satu fungsi aktivasi yang sering digunakan pada *Neural Network* ialah Fungsi aktivasi *Rectified Linear Unit* (ReLU) (Chen, 2018). Bentuk fungsi ReLU:

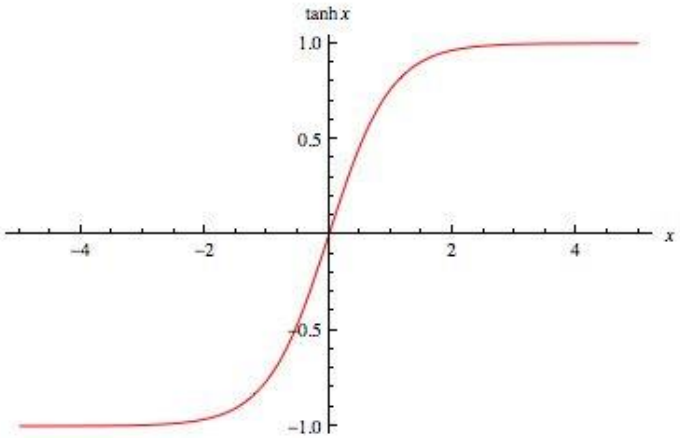
$$f(x) = \begin{cases} x, & x \geq 0 \\ 0, & x < 0 \end{cases}$$

Pada fungsi aktivasi ReLU, semua nilai x negatif akan dipetakan ke 0, seperti pada gambar 2.3:



Gambar 2.3 Grafik Fungsi ReLU

Berdasarkan gambar 2.3 jika nilainya negatif atau di bawah nol, maka hasilnya tetap nol. Tetapi jika nilainya di atas nol, maka hasilnya tetap nilai tersebut.



Gambar 2.4 Hyperbolic Tangent

Bedasarkan gambar 2.4 *Hyperbolic Tangent* memiliki range dari -1 sampai 1 yang dinyatakan pada Persamaan 1.

$$f(x)= \tanh(x) = \frac{2}{1+ e^{-2x}} -1 \quad (1)$$

Fungsi *softmax* menghitung probabilitas dari setiap kelas target atas semua kelas target yang memungkinkan dan akan membantu untuk menentukan kelas target untuk input yang diberikan (Sofia, 2018).

Jika diberikan suatu *convolutional neural network* dengan jumlah *node output* n_o (kelas) maka peluang masing-masing kelas o dapat dinyatakan pada Persamaan 2.

$$P\ (y = o) = \frac{z_o}{\sum_{i=1}^{n_o} z_i} \quad (2)$$

Dengan z_i adalah nilai aktivasi pada *node output* ke i.

2.2.3 Backpropagation

Algoritma *backpropagation* biasanya digunakan oleh *perceptron* dengan banyak lapisan untuk mengubah bobot-bobot yang terhubung dengan neuron-neuron yang ada pada lapisan tersembunyinya. Algoritma *Backpropagation* menggunakan *error output* untuk mengubah nilai bobot-bobotnya dalam arah mundur (*backward).* Untuk mendapatkan error tersebut, tahap perambatan maju (*forward propagation*) harus dikerjakan terlebih dahulu (Fausett, 1994).

Pada dasarnya, pelatihan dengan metode *backpropagtion* terdiri atas tiga langkah, yaitu sebagai berikut:

a. Data dimasukkan ke input jaringan (*feedforward*)
b. Perhitungan dan propagasi balik dari *error* yang bersangkutan
c. Pembaharuan (*adjustment*) bobot.

Jika nilai error yang dihasilkan lebih besar dari batas error yang digunakan dalam sistem, maka akan dilakukan koreksi bobot. Koreksi bobot dapat dilakukan dengan menambah atau menurunkan nilai bobot.

Jika sinyal keluaran terlalu besar dari target yang ditentukan maka bobotnya diturunkan, sebaliknya jika sinyal keluaran terlalu kecil dari target yang ditentukan maka bobotnya dinaikkan. Koreksi bobot akan dilakukan sampai selisih target dan sinyal keluaran sekecil mungkin atau sama dengan batas *error*. Untuk melakukan koreksi bobot akan dilakukan penelusuran ke belakang seperti ditunjukkan dengan tanda panah mundur.

Adapun cara kerja dari *Backpropagation*:

1. Tiap-tiap output menerima target pola yang berhubungan dengan input pembelajaran, hitung informasi errornya Pesamaan 3.

$$\delta_K = \check{y} - y \quad (3)$$

2. Kemudian hitung koreksi bobot (yang akan digunakan untuk memperbaiki nilai bobot (w_{jk}) dengan laju pembelajaran α Persamaan 4.

$$\Delta w_{jk} = \alpha \delta_k Z_j, \text{ (k = 1,2,…,m dan j = 0,1,2,…, p)} \quad (4)$$

3. Tiap-tiap unit tersembunyi (Z_j, j = 1, 2, 3 … . . p) menjumlahkan delta inputnya (dari unit-unit yang ada pada lapisan dibawahnya) Persamaan 5.

$$\delta_{_in\ j} = \sum_{k=1}^{m} \delta_k w_{jk} \quad (5)$$

Hitung informasi errornya, dimana $\theta^{(l)^T}$ adalah matriks bobot yang ada di *hidden* ke *layer* yang ada dibelakangnya $\delta^{(l+1)}$ adalah hidden ke layer ke i+1 dari arah belakang kemudian dikalikan dengan fungsi aktivasi Persamaan 6.

$$\delta_j = \theta^{(l)^T} \theta^{(l+1)} * (z_in_j) \quad (6)$$

Kemudian hitung koreksi bobot (yang nantinya akan digunakan untuk memperbaiki nilai v_{ij}dengan laju pembelajaran α Persamaan 7.

$$\Delta v_{ij}\ \alpha \vartheta_j x_i, (j = 1, 2 \dots p; i = 0,1,2, \dots n) \quad (7)$$

4. Hitung semua perubahan bobot

a. Perubahan bobot garis yang menuju ke unit keluaran Persamaan 8.

$$w_{jk} \text{ (baru)} = w_{jk} \text{(lama)} + \Delta w_{jk} \quad (8)$$

b. Perubahan bobot garis yang menuju ke unit tersembunyi Persamaan 9.

$$v_{ij} \text{ (baru)} = v_{ij} \text{(lama)} + \Delta w_{ij} \quad (9)$$

c. Proses iterasi selesai.

Keterangan symbol:

ϑ_k = Informasi tentang kesalahan pada unit Y_k yang disebarkan kembali ke unit tesembunyi.

ϑ_j= Informasi tentang kesalahan dari lapisan output ke unit tersembunyi.

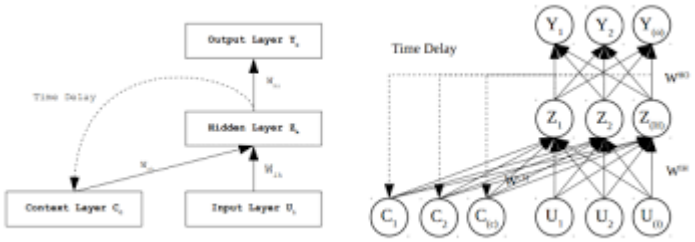
$\boldsymbol{\alpha}$ = Laju Pembelajaran (*Learning Rate*).

2.2.4 Recurrent Neural Network

Recurrent Neural Network merupakan jaringan saraf berulang. Dikatakan jaringan saraf berulang karena nilai *neuron* pada *hidden layer* sebelumnya akan digunakan kembali sebagai data input. Penggunaan *neuron* pada *hidden layer* akan disimpan ke dalam sebuah layer yang dinamakan *context layer*. Nilai *neuron* pada *context layer* akan terus update hingga kondisi RNN terpenuhi (Ahmad, 2017).

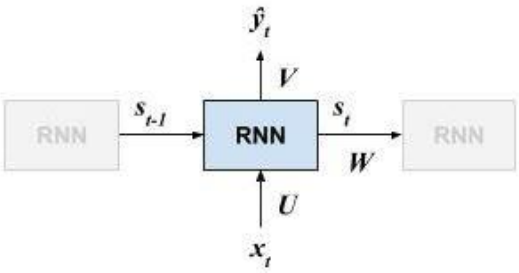
RNN masuk dalam kategori *deep learning* karena data diproses secara otomatis dan tanpa pendefinisian fitur. RNN dapat menggunakan status internal

(memori) untuk memproses urutan masukan. Ini membuatnya dapat diterapkan untuk tugas-tugas seperti pemrosesan bahasa alami (PBA), pengenalan suara, sintesa musik, pemrosesan data finansial seri waktu (Chio, 2007).



Gambar 2.5 Arsitektur RNN

Gambar 2.5 menunjukkan arsitektur RNN yang digunakan. Jumlah *neuron* dalam input *layer* adalah lima buah *neuron* dan sebuah *neuron bias*. Penambahan bias pada *neuron* di input *layer* dan *hidden layer* diberikan untuk membantu proses pembelajaran. Pada dasarnya penambahan bias akan menambah beban perhitungan namun bisa membantu pergerakan neuron pada ambang fungsi aktivasi yang digunakan.



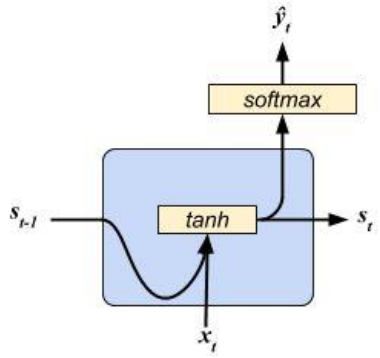
Gambar 2.6 Persamaan Dasar RNN

Persamaan dasar RNN gambar 2.6 menyatakan untuk setiap langkah waktu **t,** kalkulasi *state* s_t dari input (x_t) dan *state* sebelumnya (s_{t-1}), masing-maisng dikalikan dengan parameter **U** dan **W** lalu diproses dengan fungsi aktivasi *tanh* Persamaan 10.

$$s_t = tanh(U \cdot x_t + W \cdot s_{t-1}) \quad (10)$$

Dari s_t kemudian dikalkulasi output $\hat{y}_t$ dengan cara mengalikan dengan parameter ***V*** dan melewatkan pada fungsi aktivasi *softmax* Persamaan 11.

$$\hat{y}_t = softmax(V \cdot s_t) \quad (11)$$

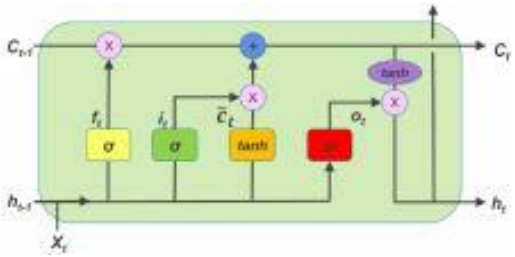


Gambar 2.7 Visualisasi Persamaan Dasar RNN

Pada gambar 2.7 s_t merupakan status internal yang diberikan satu langkah waktu ke langkah waktu berikutnya dinamakan memori dari RNN. $\hat{y}_t$ adalah keluaran kata demi kata yang diberikan oleh RNN. adalah proses masukan kata demi kata. U, V, dan W merupakan parameter yang dimiliki RNN. Sedangkan t adalah waktu. Tanh adalah fungsi aktivasi di lapisan tersembunyi dan *Softmax* di lapisan keluaran.

2.2.5 Long Short-Term Memory

LSTM merupakan salah satu jenis dari *Recurrent Neural Network* (RNN) dimana dilakukan modifikasi pada RNN dengan menambahkan *memory cell* yang dapat menyimpan informasi untuk jangka waktu yang lama (Manaswi, 2018). LSTM diusulkan sebagai solusi untuk mengatasi terjadinya *vanishing gradient* pada RNN saat memproses data *sequential* yang panjang.



Gambar 2.8 Arsitektur LSTM

Setiap unit LSTM gambar 2.8 memiliki sel memori dan status pada waktu t direpresentasikan sebagai c_t. Membaca dan memodifikasi dikendalikan

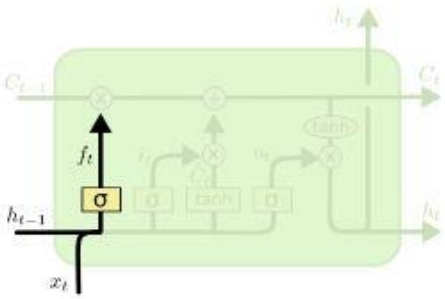
oleh gerbang sigmoid dan berpengaruh pada gerbang masukan i_t, gerbang *forget* f_t dan gerbang keluaran o_t. LSTM dihitung sebagai berikut:

Pada saat momen tanh, model menerima masukan dari dua sumber eksternal (h_{t-1} dan x_t). Status tersembunyi h_t dihitung oleh vektor masukan x_t yang diterima jaringan pada waktu t dan status tersembunyi sebelumnya h_{t-1}. Pada saat menghitung status simpul lapisan tersembunyi, gerbang masukan, keluaran, *forget* dan x_t akan secara bersamaan mempengaruhi keadaan node. Selain itu, setiap gerbang memiliki sumber internal, yaitu, status sel c_{t-1} dari blok selnya. Tautan antara cell dan gerbang sendiri dirujuk ke koneksi peephole (Winda, 2020).

LSTM dikembangkan oleh Hochreiter dan Schmidhuber pada tahun 1997. LSTM peningkatan dari algoritma RNN dasar yang memecahkan masalah yang hilang dengan menambahkan *cell states* untuk mengingat atau melupakan data. *Cell states* mengandung struktur yang disebut *cell gates*. *Cell gates* terdiri dari empat bagian yaitu *input gate, forget gate, memory-cell state gate, dan output gate*. *Input* merupakan *gate* yang digunakan untuk mengontrol input data yang layak disimpan atau tidak. *Forget gate* digunakan untuk mengontrol keadaan tersembunyi sebelumnya yang akan disimpan di *cell memory* keadaan tersembunyi saat ini. *Memory-cell state gate* digunakan untuk perbarui data berdasarkan informasi dari *input gate* dan *forget gate*. *Output gate* digunakan untuk menghitung *output* data dari jaringan berdasarkan *memory-cell state*.

Langkah-langkah *Long Short-Term Memory* (Colah, 2015):

a. Forget Gate

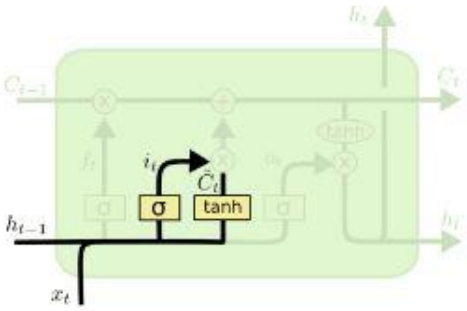


Gambar 2.9 Forget Gate

Pada *forgate gate* gambar 2.9 memutuskan informasi mana yang harus dibuang dan mana yang harus disimpan. Keputusan dibuat oleh lapisan sigmoid yang disebut lapisan forget gate Persamaan 12 dimana keluaran angka antara 0 dan 1 untuk setiap angka dalam status sel $C_t - 1$, angka 1 mewakili "mempertahankan sepenuhnya" sedangkan angka 0 mewakili "hilangkan sepenuhnya".

$$f_t = \sigma\left(W_f . [h_{t-1}, x_t + b_f\right) \quad (12)$$

b. Input Gate



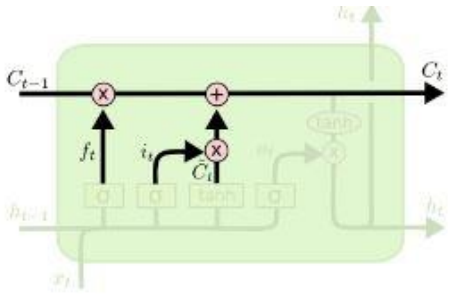
Gambar 2.10 Input Gate

Pada *Input Gate* gambar 2.10 memutuskan nilai mana yang akan diperbarui dengan transformasi nilai antara 0 dan 1. Gerbang ini memiliki dua bagian.Pertama, lapisan sigmoid yang disebut lapisan gerbang masukan memutuskan nilai mana yang harus diperbarui Persamaan 13. Selanjutnya, lapisan tanh membuat vektor dari kandidat baru yang dapat ditambah kedalam status Persamaan 14.

$$i_t = \sigma(W_i \, . [h_{t-1,} x_t + b_i \,) \quad (13)$$

$$c_in_t = tanh(W_c . + [h_{t-1}, x_t + b_c) \quad (14)$$

c. Memory Update

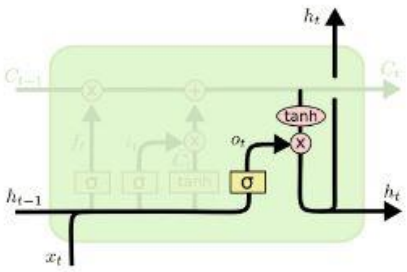


Gambar 2.11 Memory Cell State Gate

Pada *memory cell state gate* gambar 2.11 memperbaharui status sel yang lama dan melepaskan informasi lama kemudian menambahkan dengan informasi baru. Persamaan 15.

$$C_t = (f_t * C_{t-1} + i_t * C_int_t) \quad (15)$$

d. Output Gate



Gambar 2.12 Output Gate

Pada *output gate* gambar 2.12 dilakukan *hidden state* selanjutnya. Mengingat bahwa status tersembunyi terdiri dari informasi pada masukan sebelumnya. Pertama, menjalankan lapisan sigmoid yang memutuskan bagian mana dari *cell state* yang akan menjadi keluaran Persamaan 16. Kemudian, menempatkan *cell state* melalui tanh Persamaan 17.

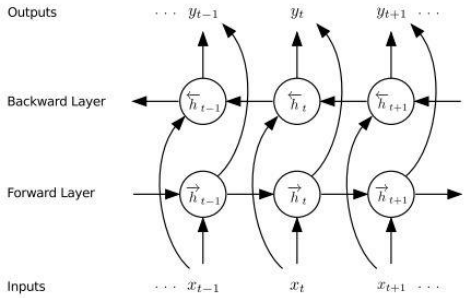
$$O_t = \sigma\,(W_o.[h_{t-1}, x_t + b_o) \quad (16)$$

$$h_t = o_t * \tanh(C_t) \quad (17)$$

2.2.6 Bidirectional LSTM

Bidirectional Long Short-Term Memory adalah perkembangan dari model LSTM dimana terdapat dua lapisan yang prosesnya saling berkebalikan arah,

model ini sangat baik untuk mengenal pola dalam kalimat karena setiap kata dalam dokumen diproses secara sekuensial, karena ulasan dapat dipahami bila pembelajaran secara berurut setiap kata. Lapisan dibawahnya bergerak maju (*forward*), yaitu memahami dan memproses dari kata pertama menuju kata terakhir sedangkan lapisan diatasnya bergerak mundur *(backward*), yaitu memahami dan memproses dari kata terakhir menuju kata pertama. Dengan adanya lapisan dua arah yang saling berlawanan ini maka model dapat memahami dan mengambil perspektif dari kata terdahulu dan kata terdepan, sehingga proses pembelajaran akan semakin dalam yang berdampak pada model akan lebih memahami konteks pada ulasan tersebut.



Gambar 2.13 Arsitektur Bidirectional LSTM (Graves, dkk 2013)

Dapat dilihat pada gambar 2.13 bahwa setiap *hidden* unit keluaran unit h_t pada lapisan bawah dan atas digabungkan membentuk nilai fitur kata tersebut dengan ukuran lebih panjang daripada menggunakan LSTM biasa. Karena lebih panjang nilai fitur, maka informasi yang akan diproses pada tahap selanjutnya yaitu *feed forward neural* akan mengklasifikasikan dengan lebih akurat.

$$\overrightarrow{h_t} = \mathcal{H}(W_{x\vec{h}}\, x_t + W_{\vec{h}\ \vec{h}}\, \overrightarrow{h_{t-1}} + b_{\vec{h}})$$

$$\overleftarrow{h_t} = \mathcal{H}(W_{x\overleftarrow{h}}\, x_t + W_{\overleftarrow{h}\,\overleftarrow{h}}\ \overleftarrow{h_{t+1}} + b_{\overleftarrow{h}})$$

$$y_t = W_{\vec{h}\ y}\, \overrightarrow{h_t} + W_{\overleftarrow{h}\ y}\, \overleftarrow{h_t} + b_y$$

2.3 Multi Kelas Klasifikasi

Menurut Jan Wira (2020), dalam bukunya yang berjudul Pengenalan Konsep Pembelajaran Mesin dan *Deep Lerning* bahwa *Multilayer perceptron* dapat memiliki lebih dari satu *output unit.* Seumpama terdapat empat kelas, dengan demikian dapat dipresentasikan keempat kelas tersebut sebagai empat *output units*.

Kelas pertama direpresentasikan dengan unit pertama, kelas kedua dengan unit kedua, dst. Untuk C kelas, dapat merepresentasikannya dengan C *output units.* Merepresentasikan data yang harus dimasukkan ke kelas mana menggunakan *sparse vector,* yaitu nilai 0 atau 1. Elemen ke-i bernilai 1 apabila data masuk ke kelas c*i,* sementara nilai elemen lainnya adalah 0. Output ANN dilewatkan pada suatau fungsi softmax yang mengembangkan probabilitas *class-assignment; i.e.* ingin *output* agar semirip mungkin dengan *spare vectore (desired output). Cross entropy* cocok digunakan sebagai *ulitity function* ketika output berbentuk distribusi.

Pada *supervised lerning,* diprediksi kelas berdasakan *feature vector* yang merepresentasikan suatu sampel (data/observasi). *Featur vector* bisa diibaratkan sebagai sifat-sifat atau keadaan yang diasosiasikan dengan kelas. Pada *supervised learning,* setiap *feature vector* berkorespondensi dengan kelas tertentu. Mencari kelas yang berkorespondensi terhadap suatu input disebut **klasifikasi (classification).** Contoh klasifikasi adalah mengkategorikan gambar buah (manga, apel, jeruk, dsb).

Multi-class classification yang telah dijelaskan sebelumnya disebut juga sebagai *hard classification,* artinya apabila data diklasifikasikan ke kelas tertentu, maka tidak mungkin data berada di kelas lainnya (ya atau tidak). Apabila kelas-kelas tersebut tidak bersifat *mutually eclusive,* maka dilakukan *multi-label classification*.

Tabel 2.1 Contoh Dataset Play Tennis (UCI Mechine learning depository)

id	Outlook	Temperature	Humidity	windy	play (class)
1	Sunny	Hot	High	False	No
2	Sunny	Hot	High	True	No
3	Overcast	Hot	High	False	Yes
4	Rainy	Mild	High	False	Yes
5	Rainy	Cool	Normal	False	Yes
6	Rainy	Cool	Normal	True	No
7	Overcast	Cool	Normal	True	Yes
8	Sunny	Mild	High	False	No

id	Outlook	Temperature	Humidity	windy	play (class)
9	Sunny	Cool	Normal	False	Yes
10	Rainy	Mild	Normal	False	Yes
11	Sunny	Mild	Normal	True	Yes
12	Overcast	Mild	High	True	Yes
13	Overcast	Hot	Normal	False	Yes
14	Rainy	Mild	High	True	No

Pada Tabel 2.1, tabel ini disebut sebagai dataset, yaitu memuat *entry data* (tiap baris disebut *instanceI).* Hanya ada dua kelas, klasifikasi data ini disebut dengan bonary classification. Apabila kelas klasifikasi lebih dari dua (*mutually exclusive),* disebut ***multi-class classification.*** Apabila kelas-kelas tersebut bersifat *mutually exclusive,* maka dilakukan ***multi-label classification.***

2.4 Multiclass Confusion Matrix

Multiclass confusion matrix merupakan bentuk *confusion matrix* yang memiliki kelas lebih dari dua dan menggambarkan kinerja suatu model pada serangkaian data uji yang nilai sebenarnya diketahui. Pada dasarnya confusion matrix memberikan informasi perbandingan hasil kalsifikasi yang dilakukan oleh sistem (model) dengan hasil klasifikasi sebenarnya.

Pada *confusion matrix*, terdapat 4 istilah representasi dari hasil proses klasifikasi. Kempat istilah tersebut adalah *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN). Kelas klasisifikasi positif yang diprediksi benar disebut *true positive*, kelas klasifikasi positif yang diprediksi negatif disebut *false negative* (salah). Kelas negatif yang diprediksi benar disebut *true negative*, dan kelas negatif yang diprediksi salah disebut *false negative*.

Confusion matrix dapat digunakan dalam menghitung berbagai *performance metrics* untuk mengukur kinerja model yang telah dibuat. *Performance metrics* yang sering digunakan adalah *accuracy, precision, recall,* dan *F1-Score*. *Acuracy* menggambarkan seberapa akurat model dapat mengklasifikasikan dengan benar. *Accuracy* merupakan rasio prediksi benar dengan keseluruhan data. Dengan kata

lain, *accuracy* merupakan tingkat kedekatan nilai prediksi dengan nilai actual (nilai sebenarnya).

Dimisalkan M adalah *confusion matrix* di mana barisnya melambangkan kelas aktual dan kolomnya melambangkan kelas prediksi, dan i melambangkan index baris dan j melambangkan index kolom.

Accuracy menggambarkan seberapa akurat model dapat mengklasifikasikan dengan benar. *Accuracy* merupakan rasio prediksi benar dengan keseluruhan data. Dengan kata lain, accuracy merupakan tingkat kedekatan nilai prediksi dengan nilai actual (nilai sebenarnya) Persamaan 18.

$$\text{Accuracy} = \frac{\sum_i M_{ii}}{\sum_i \sum_j M_{ij}} \tag{18}$$

Precision menggambarkan tingkat kebenaran suatu pengklasifikasi dari seluruh kelas positif yang diprediksi. Maka, precision merupakan rasio prediksi benar positif dibandingkan dengan keseluruhan hasil yang diprediksi positif Persamaan 19.

$$\text{Precision}_i = \frac{M_{ii}}{\sum_i M_{ji}} \tag{19}$$

Recall (Sensitivity) menggambarkan seberapa baik suatu pengklasifikasi dalam mengklasifikasikan kelas positif. *Recall* merupakan rasio prediksi benar positif dibandingkan dengan keseluruhan data yang benar positif Persamaan 20.

$$\text{Recall}_i = \frac{M_{ii}}{\sum_j M_{ij}} \tag{20}$$

F1-score merupakan gabungan antara precision dan recall yang bertujuan untuk melihat seberapa baik performa model dalam memprediksi suatu kelas Persamaan 21.

$$\text{f1-score}_i = \frac{2(\text{Precision}_i \; x \; \text{Recall}_i)}{Precision_i + Recall_i} \tag{21}$$

Keterangan :

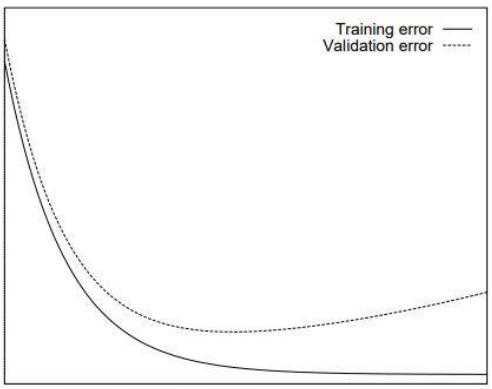
M : Confusion Matrix

i, j : **i,j** $\boldsymbol{\epsilon}$ { negative, netral, positive}

2.5 Early Stopping

Proses pelatihan akan menggunakan ukuran batch 32 dan jumlah epoch menggunakan *early stopping* untuk mencari hasil yang terbaik. *Early stopping* model *training* adalah strategi atau pendekatan untuk menghentikan *training* model untuk mencegah overfitting model. Pada kasus (Simple Linear Regression Menggunakan TensorFLow) hanya melatih model dengan pendekatan sederhana. Hal ini sengaja dilakukan karena saat itu masalah *overfitting* belum diperkenalkan. Sehingga dalam kasus tersebut hanya menentukan jumlah *epoch*. Tidak ada kepastian apakah minimum cost function sudah dicapai. Dan ada kemungkinan model overfit.

Oleh karena itu, pendekatan yang dapat dilakukan berikutnya adalah berhenti melakukan *training* saat *loss function* <= 0.001. Strategi ini memastikan kita mencapai minimum. Namun ada kemungkinan model overfit.



Gambar 2.14 Idealized training and validation error curves. Vertical: errors; horizontal: time

Gambar 2.14 menunjukkan evolusi dari waktu ke waktu kesalahan per contoh pada set pelatihan dan pada set validasi tidak digunakan untuk pelatihan *(the training error curve and the validation error curve)*. Dengan melihat hal ini, jelas bagaimana melakukan *early stopping* pada validasi (Lutz Prechelt, 2000).

2.6 Tensorflow

Tensorflow adalah sebuah *framework* komputasional untuk membuat model *machine learning*. Tensorflow menyediakan berbagai *toolkit* yang memungkinkan untuk membuat model pada tingkat abstraksi yang disukai.

Tensorflow dapat membantu membuat jaringan *neural network* dalam skala besar. *Tensorflow* telah membantu para ilmuwan dalam proyek-proyek seperti pencarian planet baru, dan membantu dokter mencegah kebutaan pada pasien yang menderita penyakit diabetes.