KLASIFIKASI *ASETILASI* DAN *METILASI* PADA DNA RAGI MENGGUNAKAN METODE *LONG SHORT-TERM MEMORY* (LSTM)

(Skripsi)

Oleh

MOHAMMAD FAJAR
NPM 1917051014

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG
2023

ABSTRAK

KLASIFIKASI *ASETILASI* DAN *METILASI* PADA DNA RAGI MENGGUNAKAN METODE *LONG SHORT-TERM MEMORY* (LSTM)

Oleh

MOHAMMAD FAJAR

DNA merupakan molekul di dalam sel yang mengandung informasi genetik yang bertanggung jawab atas perkembangan dan fungsi suatu organisme. DNA merupakan sebuah cetak biru untuk membuat protein. Terdapat protein yang disebut *Histone*. *Histone* terdiri dari empat jenis, yaitu H2A, H2B, H3, dan H4 dan biasanya DNA berada di sekeliling *Histone*. DNA sebagai struktur dasar pembentuk protein, pada prosesnya akan terjadi proses transkripsi yang mengubah DNA menjadi RNA kemudian menjadi protein. Pada protein terjadi modifikasi pada asam amino saat proses, yaitu *Asetilasi* dan *Metilasi*. Pada Tahun 2005 dilakukan penelitian mengenai klasifikasi DNA oleh Pokholok et al., dan didapatkan data DNA dari organisme Ragi. Data tersebut berisikan sekuens DNA ada atau tidaknya *Histone* atau modifikasi DNA. Untuk melakukan klasifikasi DNA diperlukan sumber daya yang besar sehingga untuk mengatasi hal tersebut digunakanlah metode komputasi. Penelitian ini akan dilakukan pengklasifikasian biner terhadap DNA Ragi. DNA Ragi akan diklasifikasikan dengan menentukan apakah terdapat *Histone*, terjadi *Asetilasi* atau *Metilasi*. Metode yang digunakan adalah *Long Short-Term Memory* (LSTM). Penelitian ini bertujuan untuk melakukan *Experiment* seberapa baik kinerja pengklasifikasian secara spasial pada data DNA. *Dataset* yang digunakan diperoleh dari penelitian Pokholok et al., (2005) yang berisikan sepuluh *dataset*. Setiap *dataset* memiliki dua kelas, yaitu positif dan negatif. Pembagian data dilakukan dengan skenario, yaitu 80% *training* dan 20% *testing*. Evaluasi model dilakukan menggunakan *Confusion Matrix* dengan metrik evaluasi yang digunakan yaitu *Accuracy*, *Precision*, dan *Recall*. Hasil klasifikasi DNA Ragi terbaik didapatkan dari *Experiment* 4 yang memiliki rerata *Accuracy* Pengujian yang terbaik dengan rerata *Accuracy* sebesar 63,10%, *Precision* sebesar 62,95%, dan *Recall* sebesar 77,80%.

Kata Kunci: DNA Ragi, *Histone*, *Asetilasi* dan *Metilasi*, LSTM.

ABSTRACT

CLASSIFICATION OF ACETYLATION AND METHYLATION IN YEAST DNA USING LONG SHORT-TERM MEMORY (LSTM) METHOD

By

MOHAMMAD FAJAR

DNA is a molecule inside cells that contains genetic information responsible for the development and function of an organism. DNA is a blueprint for making proteins. There are proteins called Histones. Histone consists of four types, namely H2A, H2B, H3, and H4 and usually DNA is around Histone. DNA as the basic structure of protein formation, in the process there will be a transcription process that converts DNA into RNA and then into protein. In proteins, there are modifications to amino acids during the process, namely acetylation and methylation. In 2005, research was conducted on DNA classification by Pokholok et al., and obtained DNA data from yeast organisms. The data contains DNA sequences of the presence or absence of Histone or DNA modifications. To perform DNA classification, large resources are needed so that to overcome this, computational methods are used. This research will do a binary classification of yeast DNA. Yeast DNA will be classified by determining whether there is Histone, Acetylation or Methylation. The method used is Long Short-Term Memory (LSTM). This research aims to experiment how well the spatial classification performance on DNA data. The dataset used is obtained from the research of Pokholok et al., (2005) which contains ten datasets. Each dataset has two classes, namely positive and negative. Data division is done by scenario, which is 80% training and 20% testing. Model evaluation is done using Confusion Matrix with evaluation metrics used, namely Accuracy, Precision, and Recall. The best Yeast DNA classification results were obtained from Experiment 4 which had the best average Testing Accuracy with an average Accuracy of 63.10%, Precision of 62.95%, and Recall of 77.80%.

Keywords: Yeast DNA, Histone, Acetylation and Methylation, LSTM.

KLASIFIKASI *ASETILASI* DAN *METILASI* PADA DNA RAGI MENGGUNAKAN METODE *LONG SHORT-TERM MEMORY* (LSTM)

Oleh

MOHAMMAD FAJAR

Skripsi

Sebagai Salah Satu Syarat Untuk Memperoleh Gelar
SARJANA ILMU KOMPUTER

Pada

Jurusan Ilmu Komputer
Fakultas Matematika dan Ilmu Pengetahuan Alam

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG
2023

Judul Skripsi	:	**KLASIFIKASI *ASETILASI* DAN *METILASI* PADA DNA RAGI MENGGUNAKAN METODE *LONG SHORT-TERM MEMORY* (LSTM)**
Nama Mahasiswa	:	*Mohammad Fajar*
Nomor Pokok Mahasiswa	:	1917051014
Program Studi	:	S1-Ilmu Komputer
Fakultas	:	Matematika dan Ilmu Pengetahuan Alam

MENYETUJUI

1. Komisi Pembimbing

Favorisen R. Lumbanraja, Ph.D.
NIP 19830110 200812 1 002

2. Ketua Jurusan Ilmu Komputer

Didik Kurniawan, S.Si., M.T.
NIP 19800419 200501 1 004

MENGESAHKAN

1. Tim Penguji

   Ketua : **Favorisen R. Lumbanraja, Ph.D.** .....

   Penguji : **M. Reza Faisal, S.T., M.T., Ph.D.** .....

   Penguji Pembahas : **Tristiyanto, S.Kom., M.I.S., Ph.D.** .....

2. Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam

Dr. Eng. Heri Satria, S.Si., M.Si.

NIP 19711001 200501 1 002

Tanggal Lulus Ujian Skripsi: **04 Desember 2023**

PERNYATAAN

Saya yang bertanda tangan di bawah ini:

Nama : Mohammad Fajar
NPM : 1917051014

Dengan ini menyatakan bahwa skripsi saya yang berjudul "KLASIFIKASI *ASETILASI* DAN *METILASI* PADA DNA RAGI MENGGUNAKAN METODE *LONG SHORT-TERM MEMORY* (LSTM)" adalah benar hasil karya saya sendiri dan bukan karya orang lain. Seluruh tulisan yang tertulis dalam skripsi ini telah mengikuti kaidah penulisan karya tulis ilmiah Universitas Lampung. Jika di kemudian hari terbukti skripsi saya adalah hasil penjiplakan atau dibuat oleh orang lain, maka saya bersedia menerima sanksi sesuai dengan peraturan yang berlaku.

Bandar Lampung, 19 Desember 2023
Penulis,

Mohammad Fajar
NPM. 1917051014

RIWAYAT HIDUP

Penulis dilahirkan di Tanjung Pinang pada tanggal 19 Maret 2001 sebagai anak pertama dan satu-satunya dari pasangan Bapak Bardian dan Ibu Meli. Penulis menyelesaikan Pendidikan Sekolah Dasar (SD) di SDIT Darul Hikmah Jakarta Timur (Hanya kelas 1 SD), lalu pindah ke SDN Percontohan Cipinang Muara 05 Pagi Jakarta Timur (Hanya sampai kelas 2 SD Semester 1), kemudian kembali pindah ke SDN 03 Pandeglang Banten (Sampai kelas 5 SD Semester 2), dan terakhir pindah ke SDN 02 Palapa Bandar Lampung (Sampai kelas 6 SD) yang diselesaikan pada tahun 2013. Kemudian melanjutkan Pendidikan Sekolah Menengah Pertama (SMP) di SMPN 01 Bandar Lampung yang diselesaikan pada tahun 2016. Kemudian melanjutkan Pendidikan Sekolah Menengah Atas (SMA) di SMA YP Unila Bandar Lampung yang diselesaikan pada tahun 2019.

Penulis terdaftar sebagai mahasiswa Jurusan Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung pada tahun 2019 melalui jalur Seleksi Nasional Masuk Perguruan Tinggi Negeri (SNMPTN) pada saat itu. Selama menjadi mahasiswa, penulis melakukan beberapa kegiatan antara lain.

1. Menjadi anggota Adapter Himpunan Mahasiswa Jurusan Ilmu Komputer pada periode 2019/2020.
2. Menjadi anggota pengurus di Bidang Dana Usaha (Danus) Unit Kegiatan Mahasiswa Sains Dan Teknologi (UKM-U Saintek) pada periode 2019/2020.

3. Menjadi anggota pengurus di Bidang Biro Dana dan Usaha (Danus) Unit Kegiatan Mahasiswa Rohani Islam Fakultas MIPA Unila (UKM-F Rois FMIPA) pada periode 2020/2021.
4. Menjadi anggota pengurus di Bidang Keilmuan Himpunan Mahasiswa Jurusan Ilmu Komputer (Himakom) pada periode 2019/2020.
5. Menjadi anggota pengurus di Bidang Biro Kesekretariatan Himpunan Mahasiswa Jurusan Ilmu Komputer (Himakom) pada periode 2020/2021.
6. Melaksanakan Praktek Kerja Lapangan (PKL) di Akademi Keperawatan Bunda Delima (Akper Bunda Delima) pada tahun ajaran 2021/2022.
7. Melaksanakan Kuliah Kerja Nyata (KKN) pada tahun ajaran 2022/2023 di Desa Sidomakmur, Kecamatan Melinting, Kabupaten Lampung Timur, Lampung.
8. Peserta ujian kompetensi *Junior Web Developer* yang diselenggarakan oleh Badan Nasional Sertifikasi Profesi (BNSP) pada tahun 2020.
9. Mengikuti ujian sertifikasi dan mendapatkan sertifikat *Junior Web Developer* (JWD) yang diselenggarakan oleh Badan Nasional Sertifikasi Profesi (BNSP) pada tahun 2022.
10. Menjadi Asisten Dosen Jurusan Biologi, Prodi S1 Biologi Terapan untuk mata kuliah Dasar-Dasar Bioinformatika pada periode semester genap tahun ajaran 2022/2023.

MOTTO

دُوْنِهٖ مِّنْ لَهُمْ وَمَا ۚ وَالٍ مِنْ لَّهُمْ مَّرَدَّ فَلَا سُوْۤءًا بِقَوْمٍ اللّٰهُ اَرَادَ اِذَا وَ بِاَنْفُسِهِمْ مَا يُغَيِّرُوْا حَتّٰى بِقَوْمٍ مَا يُغَيِّرُ لَا اللّٰهَ اِنَّ

"Sesungguhnya Allah tidak akan mengubah nasib suatu kaum sehingga mereka mengubah keadaan yang ada pada diri mereka sendiri."

(Q.S Ar-Ra'd: 11)

"Waktu bagaikan pedang. Jika engkau tidak memanfaatkannya dengan baik (untuk memotong), maka ia akan memanfaatkanmu (dipotong)."

(HR. Muslim)

"Disiplin itu melakukan apa yang kamu benci untuk kamu lakukan, tetapi kamu melakukannya seperti kamu menyukainya. Lakukan apa yang kamu benci, tidak peduli sebagus apapun kamu dalam segala hal, jika kamu tidak memiliki kedisiplinan, kamu bukan siapa-siapa, kamu bukan apa-apa tanpa kedisiplinan. Karena kamu akan mudah menyerah tanpa kedisiplinan 100%."

(Mike Tyson)

"Bagaimana mungkin aku mendapatkan hal yang luar biasa, sedangkan aku belum mengubah kebiasaan-kebiasaan burukku."

(Penulis)

"*Humility Wins The World*, Kerendahan Hati Memenangkan Dunia."

(Penulis)

PERSEMBAHAN

Alhamdulillahirobbilalamin

Puji syukur kepada Allah SWT yang telah memberikan rahmat dan hidayah-Nya sehingga saya dapat menyelesaikan penulisan skripsi ini. Sholawat dan salam saya sanjungkan kepada Nabi Muhammad SAW.

Aku persembahkan karya ini kepada:

Ayah dan Bunda

Sebagai tanda terimakasihku kepada Ayah dan Bunda yang tercinta dan yang tersayang. Terima kasih telah mendidik dan membesarkanku dengan kasih sayang kalian. Terima kasih selalu mendukungku dan mendoakanku dalam segala pilihanku. Terima kasih atas semua pengorbanan, perjuangan, dan doa kalian yang tiada hentinya. Terima kasih Ayah dan Bunda.

Seluruh Keluarga Besar, Sahabat, dan Teman-teman

Terima kasih telah memberikan semangat, dukungan, bantuan dan doa.

Dosen Pembimbing dan Pembahas

Yang senantiasa membimbing, mengarahkan dan memberi motivasi sejak awal hingga terselesaikannya skripsi ini.

Almamater Tercinta, Universitas Lampung

SANWACANA

Puji syukur kehadirat Allah SWT, karena telah memberikan rahmat dan hidayah-Nya kepada saya sehingga dapat menyelesaikan skripsi dengan judul "Klasifikasi *Asetilasi* Dan *Metilasi* Pada DNA Ragi Menggunakan Metode *Long Short-Term Memory* (LSTM)". Saya berharap skripsi ini dapat menambah pengetahuan bagi pembaca tentang DNA Ragi, *Asetilasi* dan *Metilasi*, tokenisasi dan algoritme LSTM.

Selama proses penulisan skripsi ini tidak terlepas dari dukungan banyak pihak yang telah membimbing, membantu dan memberi semangat kepada saya, sehingga pada kesempatan ini saya ingin menyampaikan ungkapan terima kasih kepada:

1. Ayah, bunda, dan keluarga saya yang selalu mendoakan, menyemangati, membiayai serta mendukung saya baik secara moral maupun material. Terima kasih atas doa yang kalian berikan untuk keberhasilan dan kesuksesan saya.
2. Bapak Favorisen R. Lumbanraja, Ph.D., sebagai dosen pembimbing utama yang telah membimbing saya, memberikan kritik dan saran, serta membina dalam menyelesaikan skripsi ini sehingga skripsi ini dapat diselesaikan dengan baik dan tepat waktu.
3. Bapak M. Reza Faisal, S.T., M.T., Ph.D., sebagai dosen pembahas pertama yang telah membimbing saya dalam memberikan ide, kritik, dan saran yang sangat bermanfaat sehingga penulisan skripsi ini dapat diselesaikan dengan baik.
4. Bapak Tristiyanto, S.Kom., M.I.S., Ph.D., sebagai dosen pembahas kedua yang telah membimbing saya dalam memberikan ide, kritik, dan saran yang sangat bermanfaat sehingga penulisan skripsi ini dapat diselesaikan dengan baik.
5. Ibu Yunda Heningtyas, S.Kom., M.Kom., sebagai dosen pembimbing akademik saya yang telah membimbing selama masa perkuliahan saya di Jurusan Ilmu Komputer.

6. Bapak Dr. Eng. Heri Satria, S.Si., M.Si., selaku Dekan FMIPA Universitas Lampung.
7. Bapak Didik Kurniawan, S.Si., M.T., selaku Ketua Jurusan Ilmu Komputer Universitas Lampung.
8. Ibu Anie Rose Irawati, ST, M.Cs., selaku Sekretaris Jurusan Ilmu Komputer Universitas Lampung.
9. Bapak dan Ibu Dosen Jurusan Ilmu Komputer FMIPA Universitas Lampung yang telah memberikan ilmu dan pengalaman dalam hidup untuk menjadi lebih baik.
10. Ibu Ade Nora Maela, Bang Zainudin, Mas Sam, dan Mas Ardi Nofalian yang telah membantu segala urusan administrasi dan segala jenis izin penulis di Jurusan Ilmu Komputer.
11. Teman-teman tutor sebaya: Ale Habsyi Arwendi, Azahra Alya, Siever Geoffrey Kalele, dan Tegar Jayanaga yang telah menemani saya dalam menjalani hari-hari dan saling memberi semangat selama masa perkuliahan.
12. Teman seper bimbingan skripsi: Ardella, Azahra, Dina, dan Jihan yang saling membantu, memberikan ide selama penyusunan skripsi dan menyemangati satu sama lain.
13. Rekan-rekan *discord* "Kelompok SUE" yang tidak bisa disebutkan satu persatu yang telah menemani untuk meringankan penat dan tempat berbagi cerita.
14. Teman-teman Ilmu Komputer 2019 yang telah menjadi keluarga dalam memberikan pengalaman dan berbagi macam memori yang seru dan tak ternilai selama menjalankan masa studi di Jurusan Ilmu Komputer.
15. Seluruh kakak tingkat Ilmu Komputer yang tidak bisa disebutkan satu persatu yang telah membantu selama masa perkuliahan penulis.
16. Semua pihak yang telah berpartisipasi baik secara langsung maupun tidak langsung dalam membantu penyusunan skripsi ini.

Penulis menyadari bahwa dalam penulisan skripsi ini tentunya masih terdapat banyak kekurangan karena keterbatasan kemampuan, pengalaman serta pengetahuan penulis. Oleh karena itu, kritik dan saran yang membangun dari para pembaca sangat diharapkan sebagai bahan pembelajaran dan evaluasi untuk penulis. Semoga skripsi ini dapat bermanfaat bagi para pembaca.

Bandar Lampung, 19 Desember 2023

Mohammad Fajar

NPM. 1917051014

DAFTAR ISI

Halaman

DAFTAR TABEL

Tabel Halaman

DAFTAR GAMBAR

Gambar Halaman

DAFTAR KODE PROGRAM

Kode Program Halaman

I. PENDAHULUAN

1.1. Latar Belakang

DNA merupakan molekul di dalam sel yang mengandung informasi genetik yang bertanggung jawab atas perkembangan dan fungsi suatu organisme. Molekul DNA memungkinkan informasi ini diteruskan dari satu generasi ke generasi berikutnya. DNA merupakan sebuah cetak biru untuk membuat protein (Mahmoud & Guo, 2021). Cetak biru setiap organisme hidup adalah DNA (*Deoxyribonucleic Acid*). Terdapat protein yang disebut *Histone*. *Histone* terdiri dari empat jenis yaitu H2A, H2B, H3, dan H4. Biasanya DNA berada di sekeliling *histone*. Pada *nukleus*, DNA akan tersimpan dalam keadaan terikat dengan *Histone Octamer*. *Histone Octamer* dibentuk dari delapan buah *Histone*, yaitu dua buah untuk setiap jenis H2A, H2B, H3 dan H4. DNA sebagai struktur dasar pembentuk protein, pada prosesnya akan terjadi proses transkripsi yang mengubah DNA menjadi RNA kemudian menjadi protein. Di protein terkadang terjadi modifikasi pada *asam amino* saat proses ini. *Asetilasi* dan *Metilasi* adalah contoh dari modifikasi yang dapat terjadi. *Asetilasi* merupakan modifikasi dengan menambahkan gugus asetil pada *asam amino*. *Metilasi* merupakan proses modifikasi dengan menambahkan gugus metil ke dalam *asam amino*.

Dengan munculnya teknologi pengurutan yang efektif, adalah mungkin untuk melakukan pengurutan yang ditargetkan untuk identifikasi organisme yang diperoleh dari sampel lingkungan yang berbeda (Achtman et al., 2002). Karena fenomena ini, lebih banyak proyek *sequencing* atau *dataset* telah diserahkan dalam *database National Center for Biotechnology Information (NCBI)* dan telah tersedia untuk umum. Jumlah data dari *untaian* DNA juga terus meningkat, seperti pada GenBank yang diperkirakan telah mencapai 220 juta sekuens pada bulan Juni 2021 (GenBank and WGS Statistics).

Studi *metagenomik* akhir-akhir ini menarik banyak perhatian karena tidak memerlukan kultur sel untuk mengkarakterisasi spesies (Woo et al., 2008). Pendekatan yang jauh lebih standar belum ditetapkan untuk memproses kumpulan data sebesar itu tanpa menimbulkan bias dalam analisis ini. Dengan munculnya algoritme pembelajaran mesin dalam dekade terakhir, berbagai penelitian muncul yang melibatkan pendekatan *Naive Bayes* seperti pengklasifikasi *Ribosomal Database Project (RDP)*, *Hierarchical Clustering*, *Random Forests*, dan *Support Vector Machines*. Sampai batas tertentu semua studi ini mengandalkan pencocokan kesamaan urutan.

Sebelumnya telah dilakukan penelitian mengenai klasifikasi DNA. Pada tahun 2005, penelitian Pokholok et al., didapatkan data DNA dari organisme Ragi. Data tersebut berisikan sekuens DNA ada atau tidaknya *Histone* atau modifikasi DNA. Selanjutnya pada tahun 2008 Higashihara et al., melakukan penelitian. Pada penelitian tersebut diklasifikasikan data DNA dengan algoritme *Support Vector Machine (SVM)*, dimana sebelum diklasifikasikan fitur, pada datanya telah lebih dulu diseleksi dan di-*ranking* dengan algoritme *Random Forest*. Kemudian pada tahun 2016, Nguyen et al., melakukan penelitian. Penelitian tersebut mengklasifikasikan data DNA menggunakan algoritme *seq-CNN*. Lalu ada penelitian terbaru yang dilakukan oleh Mahmoud dan Guo pada tahun 2021. Penelitian tersebut dilakukan dengan tahapan mengubah data DNA menjadi gambar, kemudian diklasifikasikan dengan gabungan beberapa model pra-latih *deep learning*.

Klasifikasi merupakan salah satu tugas yang sering dilakukan oleh *machine learning*. *Deepface* merupakan salah satu hasil dari klasifikasi yang dilakukan *machine learning* mampu memprediksi hingga 97.35% (Taigman et al., 2014). Klasifikasi organisme memiliki berbagai *denominasi*, yang bersifat *hierarkis*. Di sisi lain, karena kemajuan inovasi pengurutan DNA dengan aksesibilitas perangkat *Keras* komputasi yang unggul (misalnya GPU) yang diperlukan untuk menangani sekuens DNA dan menghasilkan data yang bermanfaat tentangnya, karakterisasi DNA menjadi teknik yang umum digunakan dengan kecepatan tinggi (Hebert & Gregory, 2005). Dengan sumber daya komputasi yang dapat diakses dan repositori data publik yang besar, maka teknik *deep learning* dibutuhkan untuk tugas

klasifikasi yang melibatkan analisis citra medis, *genomik* kanker, serta mengoreksi kesalahan pengurutan.

Terdapat juga algoritme yang dapat mengklasifikasikan suatu sekuens seperti DNA. Salah satu algoritme untuk klasifikasi sekuens adalah *Long Short-Term Memory* (LSTM). Penelitian ini akan dilakukan pengklasifikasian *biner* terhadap DNA Ragi. DNA Ragi akan diklasifikasikan dengan menentukan apakah terdapat profil *Histone*, terjadi *Asetilasi* atau *Metilasi*. Algoritme yang digunakan pada penelitian ini adalah *Long Short-Term Memory* (LSTM). DNA akan ditransformasi menggunakan teknik tokenisasi sebelum dijadikan *input* pada model LSTM. Penelitian ini bertujuan untuk melakukan *Experiment* seberapa baik kinerja pengklasifikasian secara *spasial* pada data DNA.

1.2. Rumusan Masalah

Berdasarkan pemaparan pada latar belakang, maka rumusan masalah dalam penelitian ini adalah sebagai berikut:

a. Apakah pembuatan model untuk mengklasifikasikan data DNA dapat diimplementasikan dengan menggunakan metode *Long Short-Term Memory* (LSTM)?
b. Seberapa baik kinerja klasifikasi dari metode *Long Short-Term Memory* (LSTM) dalam mengklasifikasikan data DNA?
c. Apakah kinerja klasifikasi dengan menggunakan metode *Long Short-Term Memory* (LSTM) lebih baik jika dibandingkan dengan metode dari penelitian terdahulu dalam mengklasifikasikan data DNA?

Berdasarkan rincian masalah tersebut, maka didapatkan rumusan masalah utama, yaitu memodelkan Klasifikasi *Asetilasi* dan *Metilasi* Pada DNA Ragi Menggunakan Metode *Long Short-Term Memory* (LSTM). Penelitian ini juga dapat dijadikan informasi untuk penelitian lanjutan mengenai Klasifikasi *Asetilasi* dan *Metilasi* Pada DNA Ragi dengan menggunakan metode LSTM.

1.3. Batasan Masalah

Batasan masalah dalam penelitian ini adalah sebagai berikut:

a. Membuat model klasifikasi data DNA menggunakan algoritme *Long Short-Term Memory* (LSTM).
b. Data yang akan digunakan adalah Peta *Genom Nukleus Asetilasi* dan *Metilasi* pada Ragi. *Dataset* diambil dari penelitian yang dilakukan oleh Pokholok et al., (2005). Sumber data dapat diakses melalui link berikut. (https://drive.google.com/drive/folders/1xOjKwfgZ1GGPOSK1BGKKlu5vEPAepE34?usp=drive_link).

1.4. Tujuan Penelitian

Tujuan dari penelitian ini adalah sebagai berikut:

a. Mengetahui hasil kinerja klasifikasi dari algoritme LSTM dalam mengklasifikasi *untaian* DNA.
b. Menentukan parameter yang terbaik untuk algoritme LSTM dalam memprediksi *untaian* DNA.
c. Membandingkan performa kinerja dan hasil algoritme LSTM dengan algoritme klasifikasi DNA pada penelitian terdahulu yang menggunakan *dataset* yang sama dalam mengklasifikasi *untaian* DNA.

1.5. Manfaat Penelitian

Adapun manfaat penelitian ini adalah sebagai berikut:

a. Menambah pengetahuan dan wawasan mengenai cara kerja algoritme LSTM dalam memecahkan permasalahan klasifikasi *Asetilasi* dan *Metilasi* pada DNA Ragi.
b. Mengetahui performa kinerja dan hasil algoritme LSTM dalam mengklasifikasi *untaian* DNA.
c. Mendapatkan perbandingan hasil kinerja antara klasifikasi menggunakan LSTM dengan penelitian terdahulu.

II. TINJAUAN PUSTAKA

2.1. Penelitian Terdahulu

Penelitian ini berdasarkan penelitian sebelumnya yang telah dilakukan, sehingga penelitian yang akan dilakukan memiliki hubungan antara persamaan dan perbedaan objek yang diteliti. Ringkasan dari penelitian terdahulu dapat dilihat pada Tabel 1.

Tabel 1. Penelitian terdahulu yang terkait dengan klasifikasi DNA

No	Penelitian	Metode	Data	Hasil
1.	*Application of a Feature Selection Method to Nucleosome Data: Accuracy Improvement and Comparison with Other Methods* (Higashihara et al., 2008)	Pengklasifikasian DNA Ragi dengan fitur *ranking* dan fitur seleksi menggunakan algoritme klasifikasi *SVM*.	Data yang digunakan adalah 10 *dataset* dari penelitian Pokholok et al., (2005), yaitu H3, H4, H3K9ac, H3K14ac, H4ac, H3K4me1, H3K4me2, H3K4me3, H3K35me3, H3K79me3 • Rata-rata data positif: 15.599. • Rata-rata data negatif: 13.141.	Menyeleksi fitur penting berhasil meningkatkan akurasi. Rata-rata akurasi yang didapatkan: 75,72%

No	Penelitian	Metode	Data	Hasil
2.	*DNA Sequence Classification by Convolutional Neural Network* (Nguyen et al., 2016)	Menggunakan algoritme *seq-CNN* untuk memprediksi terjadinya *Asetilasi* dan *Metilasi*.	Data yang digunakan adalah 10 *dataset* dari penelitian Pokholok et al., (2005), yaitu H3, H4, H3K9ac, H3K14ac, H4ac, H3K4me1, H3K4me2, H3K4me3, H3K35me3, H3K79me3 • Rata-rata data positif: 15.599. • Rata-rata data negatif: 13.141.	Rata-rata akurasi yang didapat: 79,41%
3.	*DNA sequence classification based on MLP with PILAE algorithm* (Mahmoud & Guo, 2021)	Mengubah *untaian* DNA menjadi gambar dan diklasifikasikan dengan menggabungkan beberapa model pra-latih.	Menggunakan lima *dataset* H3, H4, H3K9ac, H3K14ac, dan H4ac dari penelitian Pokholok et al., (2005). • Rata-rata data positif: 13.349. • Rata-rata data negatif: 11.550.	Rata-rata akurasi yang didapat: 98,04%

Berdasarkan pada Tabel 1, dapat dilihat penelitian terdahulu mengenai klasifikasi DNA, berikut merupakan penjelasan selengkapnya:

2.1.1. *Application Of A Feature Selection Method To Nucleosome Data: Accuracy Improvement And Comparison With Other Methods*

Dalam studi yang dilakukan oleh Higashihara et al., (2008) ini berusaha untuk membedakan antara penggunaan seleksi fitur dan tidak pada pengklasifikasian DNA.

Dalam studi yang dilakukan oleh Pokholok et al., (2005), seleksi fitur tidak dipilih dari sekuens DNA yang ada, melainkan hanya dilakukan pengklasifikasian. Hampir di semua *dataset* yang digunakan mengalami peningkatan akurasi sebagai hasilnya. Akurasi tidak mengalami peningkatan kecuali pada *dataset* H3K14ac dan H4ac. *Random Forest* adalah algoritme fitur seleksi yang digunakan. Metode *Support Vector Machine (SVM)* digunakan untuk melanjutkan pengklasifikasian setelah melakukan seleksi fitur. *RBF Kernel* adalah kernel yang digunakan pada *SVM*. Setelah hasil klasifikasi diperoleh, selanjutnya dilakukan perbandingan dengan metode lainnya dan rata-rata hasil lebih baik dan unggul yang ditunjukkan.

2.1.2. *DNA Sequence Classification By Convolutional Neural Network*

Pada penelitian yang dilakukan oleh Nguyen et al., (2016) ini menggunakan data yang sama pada penelitian yang dilakukan oleh Higashihara et al., (2008) untuk melakukan pengklasifikasian. *Dataset Splice* dan *Promoter* dari *UCI machine learning repository* adalah *dataset* tambahan. Untuk menjaga informasi mengenai urutan *nukleotida* tetap tersimpan, sekuens DNA yang ada akan direpresentasikan sebagai *one-hot vector*. Pada akhirnya, hasil klasifikasi berhasil mengalami peningkatan sebesar 6%. Pada penelitian Nguyen et al., (2016) algoritme yang digunakan adalah *seq-CNN*. Penelitian ini menggunakan versi modifikasi dari algoritme *CNN* yang digunakan untuk mengklasifikasikan sekuens. *One-hot vector* digunakan untuk mentransformasikan sekuens DNA agar memiliki karakteristik yang sama dengan data teks. Kemudian Model *CNN* akan diuji untuk mendapatkan keakuratan dari model dengan data DNA sebagai *input* setelah ditransformasi. Hasil akhirnya pada *dataset* H3K4me3, model ini memiliki kinerja hingga 6,12% lebih baik, sedangkan pada *dataset* H4 mengalami peningkatan terkecil, yaitu sebesar 0,77%.

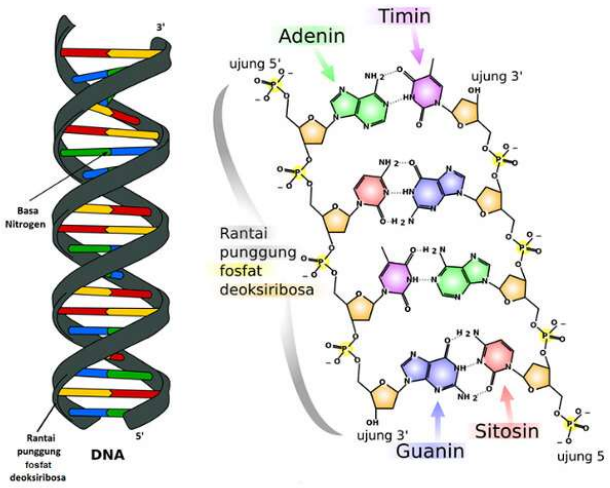
2.1.3. *DNA Sequence Classification Based On MLP With PILAE Algorithm*

Penelitian oleh Mahmoud dan Guo pada tahun 2021 ini tidak menggunakan *one-hot encoder* untuk merepresentasikan urutan DNA. Gambar digunakan sebagai cara untuk merepresentasikan DNA. DNA Data *Visualization*, yang dikembangkan oleh Neugebauer et al., (2015) digunakan untuk menghasilkan representasi gambar.

Pseudoinverse Learning Autoencoder (PILAE) akan digunakan untuk mengklasifikasikan gambar akhir. Hasilnya menunjukkan peningkatan akurasi yang signifikan. Selain itu, metode ini secara signifikan meningkatkan efisiensi komputasi. Langkah pertama dalam penelitian yang dilakukan oleh (Mahmoud & Guo, 2021) adalah menggunakan DNA Data *Visualization* untuk mengubah data DNA menjadi gambar (Neugebauer et al., 2015). Dengan menggunakan model pra-latih seperti *Xception* dan *VGG*, seleksi fitur akan dilakukan setelah data diubah menjadi gambar. *PILAE* baru bisa digunakan untuk klasifikasi setelah fitur-fitur dipilih. Hasilnya pada *dataset* H3K9ac, didapatkan akurasi yang mencapai 98,57%. Pada *dataset* H4ac, peningkatan akurasi mencapai 14,59%.

2.2. DNA

Deoxyribonucleic Acid (disingkat DNA) adalah molekul di dalam sel yang membawa informasi *genetik* untuk perkembangan dan fungsi suatu organisme. DNA terbuat dari dua helai saling terhubung yang melilit satu sama lain menyerupai tangga bengkok dengan bentuk yang dikenal sebagai *Double Helix*. Setiap helai memiliki tulang punggung yang terbuat dari gula bolak-balik (*Deoksiribosa*) dan gugus fosfat. DNA terdiri dari empat *nukleotida* yaitu *Adenine* (A), *Guanine* (G), *Cytosine* (C), dan *Thymine* (T). Struktur dari DNA ditunjukkan pada Gambar 1.



Gambar 1. Struktur DNA (Kelas Pintar, 2020).

DNA merupakan *polimer nukleotida* berarti rantai DNA yang berbentuk heliks ganda (*Double Helix*) itu terdiri atas senyawa organik *nukleotida* yang berulang-ulang dan tersusun rangkap.

Dari gambar DNA di atas, dapat dilihat jika setiap *nukleotida* terdiri dari tiga gugus molekul sebagai berikut:

a. Gula *Pentosa* atau *Deoksiribosa* dan Gugus Fosfat.
b. Basa Nitrogen yang terdiri dari golongan purin yaitu Adenin (*Adenine* = A) dan Guanin (*Guanine* = G), serta golongan Pirimidin yaitu Sitosin (*Cytosine* = C) dan Timin (*Thymine* = T).

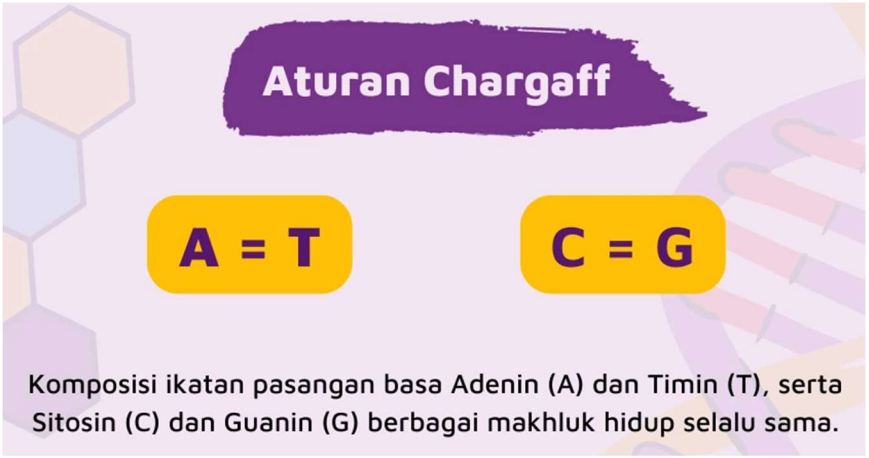
DNA dapat berbentuk spiral karena pada struktur molekul kimianya. Salah satu penyusun DNA dalam *nukleotida* adalah gula *deoksiribosa*. Jenis gula ini kehilangan satu atom oksigen dan mempunyai lima atom karbon. Berdasarkan gugus *keton*-nya, atom karbon akan menghasilkan bentuk *pentagon* atau segi lima. Pada formasi seperti ini, atom karbon nomor satu akan berikatan dengan basa *nukleotida* dan atom karbon nomor lima selalu berikatan dengan gugus fosfat. DNA memiliki sifat antiparalel, artinya jika salah satu rantai *nukleotida*-nya punya arah ke atas, *nukleotida* lainnya akan menuju ke bawah. Bisa dikatakan bahwa rantai *nukleotida* dalam DNA ini mirip dengan jalur lalu lintas dua arah, mereka saling berlawanan. Oleh karena saling berlawanan, penghitungan gula *deoksiribosa nukleotida*-nya juga akan berlawanan, tetapi aturannya akan tetap sama (Aryani, 2022).

2.2.1. Aturan Chargaff

Pada tahun 1940, seorang ilmuwan *biokimia*, Erwin Chargaff, berhasil mendata persentase basa nitrogen yang berbeda dari satu spesies dengan spesies lainnya. Meski begitu, pada setiap molekul DNA, komposisi Adenin akan selalu sama dengan jumlah Timin (A = T). Demikian juga dengan komposisi Sitosin yang selalu sama dengan Guanin (C = G).

Fenomena inilah yang kemudian dinamakan sebagai aturan Chargaff atau *Chargaff's rule*. Tidak hanya sampai di penelitian Chargaff, James Watson dan Francis Crick yang setuju dengan penemuan tersebut kemudian mencoba

mengkajinya lebih dalam. Keduanya berusaha untuk menemukan bukti lainnya tentang pasangan Adenin-Timin dan Sitosin-Guanin dalam DNA (Aryani, 2022). Aturan Chargaff bisa dilihat pada Gambar 2.



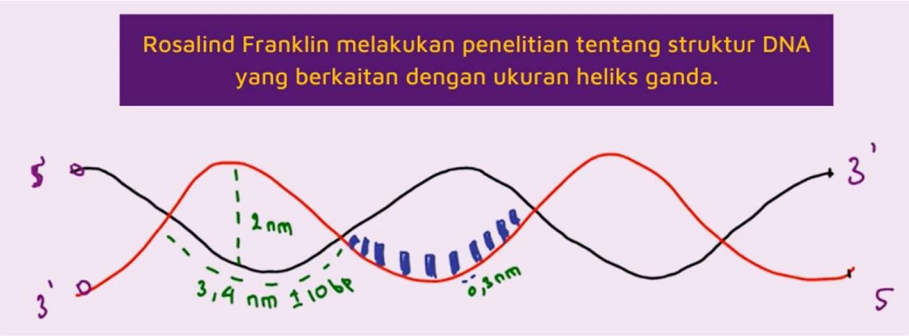
Gambar 2. Aturan Chargaff atau *Chargaff's rule* tentang komposisi basa nitrogen DNA (Aryani, 2022).

Watson dan Crick membuktikan pasangan basa nitrogen tersebut berdasarkan ukuran DNA yang ditemukan oleh Rosalind Franklin. Sebelumnya, Franklin mengungkapkan bahwa *Double Helix* DNA mempunyai lebar sekitar 2 nm dengan panjang satu lekukan sebesar 3,4 nm. Berdasarkan data tersebut, Watson dan Crick kemudian melakukan penelitian lanjutan dan memperoleh beberapa kesimpulan, di antaranya:

a. Pasangan golongan basa nitrogen purin dan purin akan menghasilkan lebar *nukleotida* lebih dari 2 nm. Berarti data ini tidak sesuai dengan hasil penelitian Franklin.

b. Pasangan golongan basa nitrogen pirimidin dan pirimidin menghasilkan lebar *nukleotida* kurang dari 2 nm. Berarti data ini juga kurang sesuai sama apa yang dikemukakan Franklin.

c. Pasangan golongan basa nitrogen purin dan pirimidin menghasilkan lebar *nukleotida* sebesar 2 nm. Artinya, data ini sesuai dengan hasil penelitian yang dilakukan Franklin. Penelitian tentang struktur DNA oleh Rosalind Franklin bisa dilihat pada Gambar 3.



Gambar 3. Penemuan Rosalind Franklin tentang ukuran *nukleotida* DNA (Aryani, 2022).

Setiap *nukleotida* akan memiliki pasangan berdasarkan pasangan dasar Watson-Crick (A-T dan G-C). Kedua helai dihubungkan oleh ikatan kimia antara basa: ikatan Adenin dengan Timin, dan ikatan Sitosin dengan Guanin. Urutan basa di sepanjang tulang punggung DNA mengkodekan informasi biologis, seperti instruksi untuk membuat molekul protein atau RNA.

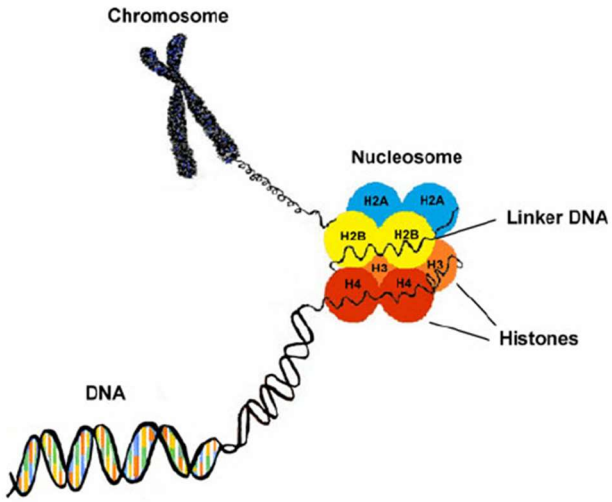
2.2.2. DNA *Sequence*

DNA *Sequence* mengacu pada teknik laboratorium umum untuk menentukan urutan yang tepat dari *nukleotida* atau basa, dalam molekul DNA. Urutan basa (sering disebut dengan huruf pertama dari nama kimianya: A, T, C, dan G) mengkodekan informasi biologis yang digunakan sel untuk berkembang dan beroperasi. Mengurutkan DNA berarti menentukan urutan empat blok bangunan kimia, disebut "Basa" yang membentuk molekul DNA. Urutan tersebut memberi tahu para ilmuwan jenis informasi *genetik* yang dibawa dalam *segmen* DNA tertentu. Informasi urutan DNA penting bagi para ilmuwan yang menyelidiki fungsi gen. Memahami urutan DNA dapat diterapkan dalam berbagai pengaturan seperti sekarang ini, pengurutan DNA dapat membentuk dasar penelitian pada *biologi* dan diterapkan dalam *bioteknologi*, *biologi forensik*, *virologi*, serta diagnosis medis.

2.3. *Histone*

Protein dasar yang disebut *Histon*e berinteraksi dengan DNA untuk membentuk *nukleosom*, yang pada organisme *eukariotik* membentuk *untaian kromatin* yang membentuk *kromosom*. Pita ganda DNA dililitkan di sekitar pusat protein yang terbuat dari protein-protein yang berinteraksi erat satu sama lain karena susunan

Histone. DNA membuat sekitar 1,7 putaran di sekitar pusat *Histone*, yang berbentuk seperti cakram. DNA dapat terikat pada pusat protein berbentuk *Histone* di setiap *nukleosom* melalui ikatan hidrogen ganda. Sebagian besar waktu, ikatan ini terbentuk antara tulang punggung gula-fosfat DNA dan kerangka *asam amino Histon*e. Ikatan *ionik* dan beberapa interaksi *hidrofobik* juga terlibat. Masuknya mesin transkripsi ke dalam DNA yang terkandung dalam *nukleosom* dimungkinkan oleh pemecahan dan pembentukan ikatan pengikat antara DNA dan *Histone* oleh protein yang dikenal sebagai "Kompleks Renovasi *Kromatin*". Terlepas dari kenyataan bahwa asam *nukleat* dekat dengan pusat protein *Histone*, pusat-pusat ini diatur sedemikian rupa sehingga faktor transkripsi dan protein lain yang terlibat dalam ekspresi atau pembungkaman gen dapat masuk jika perlu. *Histone* dapat diubah dalam berbagai cara, menghasilkan banyak varian yang berbeda. Hal ini memungkinkan adanya berbagai jenis *kromatin* yang berbeda, yang masing-masing memiliki kemampuan untuk mengubah bagaimana gen diekspresikan dengan cara yang berbeda (Thpanorama, 2023).



Gambar 4. *Histone* (Creative Proteomics, 2019).

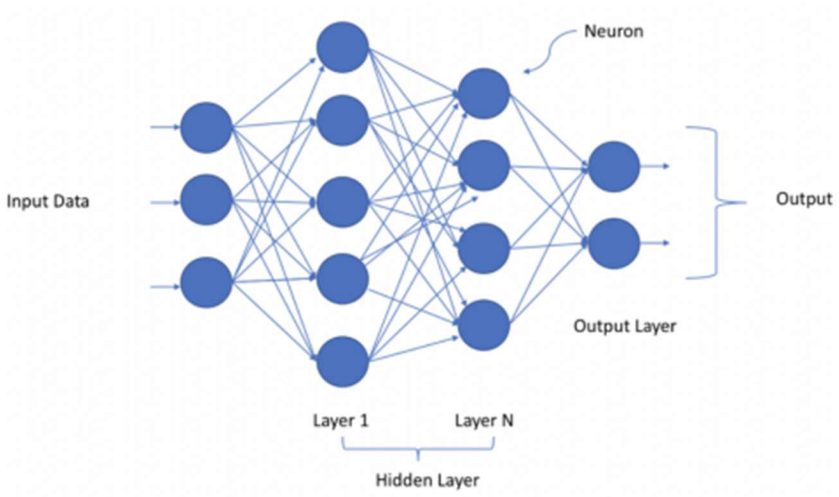
Berdasarkan pada Gambar 4, *Histone* adalah protein yang ditemukan pada inti sel *eukariota* yang terbungkus DNA, yang kemudian bersama DNA menyusun struktur *nukleosom*. Terdapat lima subunit *Histone* yaitu H1, H2A, H2B, H3 dan H4. Subunit-subunit ini kaya akan *asam amino* yang bermuatan positif atau bersifat basa. *Histone* bereaksi dengan *asam deoksiribonukleat* melalui interaksi antara

protein yang bermuatan positif dengan *fosfodiester* dari *asam deoksiribonukleat* yang bermuatan negatif kemudian membentuk *nukleosom*. Tiap inti *nukleosom* terdiri atas suatu kompleks dari delapan protein *Histone*, yang disebut juga *Histone Oktamer*, pada DNA rantai ganda dengan panjang 147 pasang *nukleotida*. Kompleks *Histone Oktamer* yang membentuk inti *nukleosom* ini masing-masing terdiri atas dua molekul *Histone* H2A, H2B, H3, dan H4. Modifikasi *Histone* memengaruhi perubahan bentuk *kromatin*. Ada berbagai macam modifikasi *Histone*, diantaranya *Asetilasi, Metilasi*, dan *Fosforilasi*.

2.4. *Artificial Neural Networks*

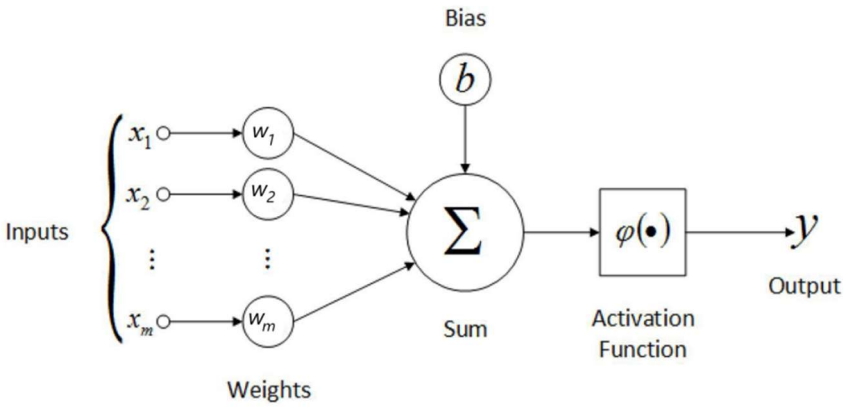
Artificial Neural Networks (ANN) pada dasarnya adalah model komputasi paralel masif yang didasarkan pada cara kerja otak manusia. Banyak prosesor sederhana yang terhubung dengan koneksi berbobot membentuk *ANN*. *Node* pemrosesan dapat disebut "*neuron*" dengan *analogi*. Informasi yang disimpan di dalam *node* atau ditransmisikan melalui koneksi berbobot hanya mempengaruhi *output* dari *node* tersebut. Setiap unit menerima *input* dari banyak *node* lain dan mentransmisikan *output* nya ke *node* lain. Satu elemen pemrosesan yang tidak terlalu kuat, menghasilkan *output* skalar dengan nilai *numerik* tunggal, yang merupakan fungsi *non-linear* sederhana dari *input* nya dengan sendirinya. Perbedaan antara respons yang diinginkan dan *output* sistem adalah yang membentuk kesalahan.

Sistem menyesuaikan parameter sistem dengan cara yang sistematis menggunakan informasi kesalahan ini (aturan pembelajaran). Sampai kinerja dapat diterima, prosedur ini diulangi. Penjelasan ini sangat jelas bahwa data memainkan peran penting dalam kinerja. *ANN* mungkin bukan pilihan terbaik jika data tidak mencakup sebagian besar kondisi operasi atau sibuk. Di sisi lain, *ANN* adalah pilihan yang baik jika ada banyak data dan masalahnya tidak cukup dipahami untuk mendapatkan model perkiraan. Bandingkan prosedur operasi ini dengan desain teknik konvensional yang terdiri dari semua spesifikasi subsistem dan protokol untuk interkomunikasi (Dongare et al., 2012). Ilustrasi dari *ANN* ditunjukkan pada Gambar 5.



Gambar 5. Ilustrasi dari *Artificial Neural Network* (C'edric, 2020).

Salah satu metode *Machine Learning* yang mampu melakukan prediksi dan klasifikasi dengan mempelajari pola-pola pada data yang telah disediakan adalah *Artificial Neural Networks*. Semakin banyak pola yang dapat dipelajari menggunakan metode ini dengan lebih banyak data yang diberikan. Data yang dipelajari sistem merupakan salah satu faktor yang mempengaruhi untuk mendapatkan hasil yang terbaik. *Backpropagation* adalah algoritme untuk mengubah bobot berdasarkan *error* yang didapat dari *output*, sedangkan *Feed Forward* adalah algoritme untuk melakukan penghitungan nilai *output* berdasarkan *input*. Sebuah mesin akan belajar lebih cepat jika tingkat pembelajarannya lebih tinggi *(learning rate)*, tetapi akan sulit untuk mencapai hasil yang optimal. Namun, ada peluang yang lebih besar untuk mencapai hasil terbaik. Berikut gambaran proses pada *ANN* dapat dilihat pada Gambar 6.



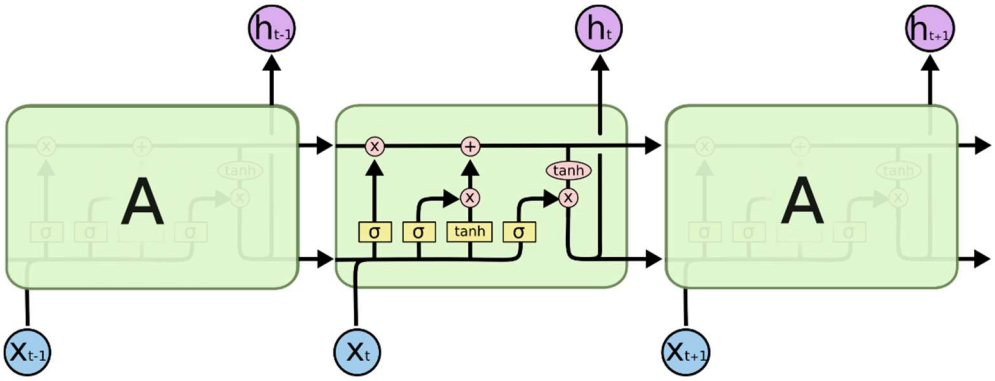
Gambar 6. Ilustrasi perhitungan pada *ANN* (C'edric, 2020).

2.5. *Long Short-Term Memory* (LSTM)

Long Short-Term Memory Network (LSTM) adalah salah satu modifikasi dari *Recurrent Neural Network* atau *RNN*. Banyak modifikasi dari *RNN*, tetapi LSTM merupakan salah satu yang populer di antaranya. LSTM hadir untuk melengkapi kekurangan *RNN* yang tidak dapat memprediksi kata berdasarkan informasi lampau yang disimpan dalam jangka waktu lama. Dengan demikian, LSTM mampu mengingat kumpulan informasi yang telah disimpan dalam jangka waktu panjang, sekaligus menghapus informasi yang tidak lagi relevan. LSTM lebih efisien dalam memproses, memprediksi, sekaligus mengklasifikasikan data berdasarkan urutan waktu tertentu (Algoritma, 2022).

2.6. Arsitektur Algoritme LSTM

Arsitektur LSTM dikembangkan sebagai solusi dari masalah *Vanishing Gradient* yang ditemui pada *RNN* konvensional. *Vanishing gradient* disebabkan karena *gradien* yang semakin mengecil hingga *layer* terakhir membuat nilai bobot tidak berubah sehingga menyebabkan tidak pernah memperoleh hasil yang lebih baik atau *konvergen*. Sebaliknya *gradien* yang semakin membesar menyebabkan nilai bobot pada beberapa *layer* juga ikut membesar sehingga algoritme optimasi menjadi *divergen* atau disebut *Exploding Gradient*. Struktur algoritme LSTM terdiri atas *neural network* dan beberapa blok memori yang berbeda. Blok memori ini disebut sebagai *cell*. *State* dari *cell* dan *hidden state* akan diteruskan ke *cell* berikutnya. Bangun berbentuk persegi panjang adalah ilustrasi *cell* pada LSTM. Informasi yang dikumpulkan oleh algoritme LSTM kemudian akan disimpan oleh *cell* dan manipulasi memori dilakukan oleh komponen yang disebut dengan *Gate*, seperti yang ditunjukkan pada Gambar 7 (Trivusi, 2022).



Gambar 7. Arsitektur LSTM (Trivusi, 2022).

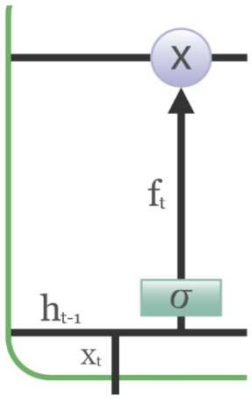
Berikut adalah penjelasan *notasi* arsitektur metode *Long Short-Term Memory* (LSTM) seperti yang ditunjukkan pada Tabel 2 (Smagulova & James, 2020).

Tabel 2. *Notasi* arsitektur metode *Long Short-Term Memory* (LSTM)

$\boldsymbol{x_t}$	***Input Vector***
h_{t-1}	*Output* dari *cell* sebelumnya
C_t	Memori *cell* dari kondisi saat ini
C_{t-1}	Memori *cell* dari *cell* sebelumnya
$\tilde{\boldsymbol{C}}_t$	Kandidat ke memori *cell*
i_t	*Input gate*
o_t	*Output gate*
f_t	*Forget gate*
g_t	*Input gate*
σ	Fungsi *sigmoid*
Tanh	*Hyperbolic tangent function*
$W^{(*)}$, $U^{(*)}$, $V^{(*)}$	Matriks bobot
b^*	Bias

Ada tiga jenis *Gate* pada algoritme LSTM, di antaranya *Forget Gate*, *Input Gate*, dan *Output Gate*.

2.6.1. *Forget Gate*



Gambar 8. *Forget Gate* (Trivusi, 2022).

Dilihat pada Gambar 8, gerbang pertama dalam LSTM disebut dengan *Forget Gate*. Mudahnya, gerbang ini bertugas untuk melupakan beberapa informasi yang tidak relevan dan sudah tidak diperlukan oleh sebuah sistem. Alhasil, LSTM dapat menyajikan kumpulan informasi yang lengkap, tetapi tetap aktual sesuai dengan kebutuhan. *Forget gate* berfungsi untuk menghapus informasi yang tidak lagi digunakan pada *cell*. Caranya adalah dengan mengevaluasi *output biner* dari dua *input x(t)* dan *output cell* sebelumnya *h(t-1)* dikalikan dengan *matriks* bobot kemudian ditambahkan dengan nilai bias. Nilai yang didapat kemudian dilewatkan melalui fungsi aktivasi dan menghasilkan *output biner*. Apabila *output*-nya bernilai 0, maka informasi dianggap tidak lagi berguna dan bisa dihapus. Begitupun sebaliknya, apabila *output*-nya bernilai 1 maka informasi tersebut disimpan untuk penggunaan di masa mendatang (Trivusi, 2022). *Forget gate* didefinisikan pada Persamaan (1) (Smagulova & James, 2020).

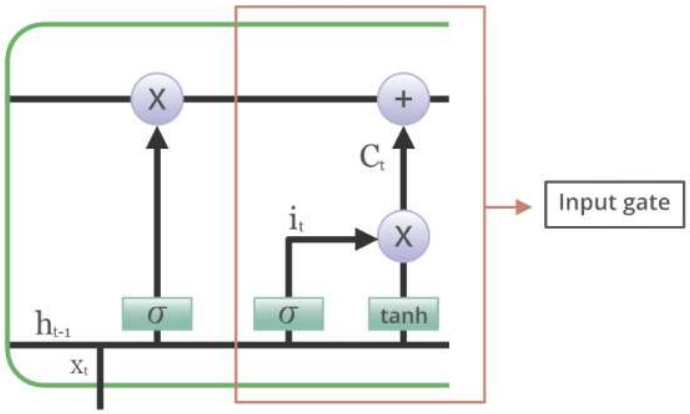
$$ft = \sigma(w(f)\, xt + U(f)\, ht\text{-}1 + b(f)) \quad \text{(1)}$$

2.6.2. *Input Gate*

Berikutnya, ada gerbang kedua, yakni *Input Gate* yang bertugas untuk memasukkan informasi yang berguna untuk mendukung keakuratan data. Tugas *input gate* adalah untuk menambahkan informasi yang sebelumnya telah diseleksi terlebih dahulu melalui *forget gate*. Gerbang ini tidak dimiliki oleh *RNN* yang hanya memungkinkan satu *input* data untuk satu *output* data. Dalam *input gate* kemudian dikenal istilah *input modulation gate* yang sering tidak ditulis dalam beberapa

ulasan tentang LSTM. Sesuai namanya, *input modulation gate* berfungsi untuk memodulasi informasi yang ada, sehingga dapat mengurangi kecepatan *konvergensi* dari data *zero-mean*. Penambahan informasi yang berguna ke *cell state* dilakukan oleh *input gate*. Pertama, informasi diatur menggunakan fungsi *sigmoid* dan menyaring nilai yang akan disimpan, prosesnya mirip dengan *forget gate* yang menggunakan *input h(t-1)* dan *x(t)*. Setelah itu, sebuah *vektor* dibuat menggunakan fungsi *tanh* yang memberikan *output* dari -1 hingga +1, yang berisi semua kemungkinan nilai dari *h(t-1)* dan *x(t)*. Terakhir, nilai-nilai *vektor* dan nilai-nilai yang diatur dikalikan untuk mendapatkan informasi yang berguna, seperti yang ditunjukkan pada Gambar 9 (Trivusi, 2022). *Input gate* didefinisikan pada Persamaan (2) (Smagulova & James, 2020).

$it = \sigma(w(i)\ xt + U(i)\ ht\text{-}1 + b(i))$ ……(2)



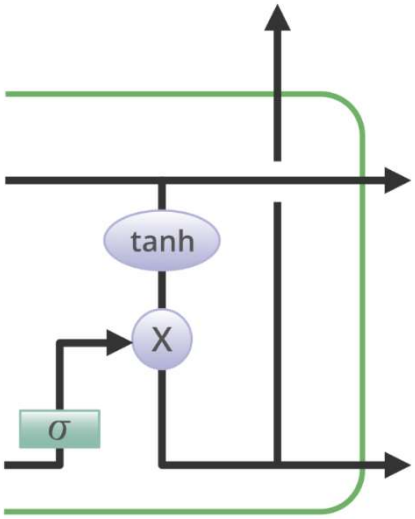
Gambar 9. *Input Gate* (Trivusi, 2022).

2.6.3. *Output Gate*

Terakhir adalah *Output Gate* yang menjadi gerbang terakhir untuk menghasilkan informasi data yang komplet dan aktual. Gerbang ini bisa menjadi yang terakhir atas sebuah informasi atau hanya menjadi bagian dari tahap pertama saja, sebelum akhirnya informasi akan diproses lewat *input gate* di *cell* berikutnya. Tugas mengekstraksi informasi yang berguna dari *cell state* saat ini untuk disajikan sebagai nilai keluaran dilakukan oleh *output gate*. Pertama, sebuah *vektor* dibangkitkan dengan menerapkan fungsi *tanh* pada *cell*. Kemudian, informasi tersebut diatur menggunakan fungsi *sigmoid* dan menyaring nilai-nilai yang akan disimpan menggunakan *input h(t-1)* dan *x(t)*. Terakhir, nilai *vektor* dan nilai yang

diatur dikalikan untuk dikirim sebagai *output* dan *input* ke *cell* berikutnya. Berikut ditunjukkan pada Gambar 10 (Trivusi, 2022). *Output gate* didefinisikan pada Persamaan (3) (Smagulova & James, 2020).

$ot = \sigma(w(o)\ xt + U(o)\ ht\text{-}1 + b(o))$…….(3)



Gambar 10. *Output Gate* (Trivusi, 2022).

Seperti yang sempat disinggung sebelumnya, LSTM *Network* memiliki *gate* yang dapat mengatur aliran atau arus informasi dengan lebih baik. Secara total, ada empat *gate* yang membentuk struktur LSTM, yaitu *forget gate (f)*, *input gate (i)*, *input modulation gate (g)*, dan *output gate (o)*. *Forget gate* berperan untuk menghapus informasi yang tidak lagi berguna dalam *cell state*. *Input gate* berperan untuk penambahan informasi berguna ke *cell state*. Informasi yang masuk diatur menggunakan fungsi *sigmoid*. Sedangkan, ekstraksi informasi dari situasi *cell* saat ini yang disajikan sebagai *output*, dilakukan oleh *input modulation gate*. Kemudian, *output gate* mengatur *output* dari unit LSTM.

2.7. Aplikasi LSTM

Pengaplikasian LSTM *Network* dapat ditemukan dalam kehidupan sehari-hari. Berikut adalah beberapa contoh aplikasi atau implementasi dari LSTM.

2.7.1. Pemodelan Bahasa

Untuk memodelkan bahasa di tingkat karakter menggunakan LSTM, cukup dengan mengganti masukan dari kata menjadi karakter. Keluaran dari LSTM sekarang dapat dianggap sebagai prediksi karakter berikutnya dengan diberikan kemunculan karakter-karakter sebelumnya (Kurniawan et al., 2020).

2.7.2. Mesin Penerjemah

LSTM digunakan dalam proses penerjemahan bahasa seperti menerjemahkan bahasa Inggris kedalam bahasa Indonesia yang dapat mengatasi permasalahan *vanishing gradient* yang menyebabkan gagalnya pembelajaran *long-term dependency* pada *RNN* (Yustiana et al., 2022).

2.7.3. *Image Captioning*

Image captioning adalah proses untuk menghasilkan suatu kalimat atau lebih untuk menjelaskan konten *visual* dari suatu gambar. *Image captioning* bermanfaat untuk kebutuhan di masa mendatang untuk membantu kegiatan manusia memahami konten *visual* seperti keterangan pada citra medis, interaksi manusia dengan robot dan membantu mendeskripsikan gambar kepada *tunanetra* (Nugroho & Hidayatulah, 2021).

2.7.4. *Handwriting Generation*

Handwriting Generation adalah proses *numerik* untuk menerjemahkan gambar teks tulisan tangan ke dalam rangkaian karakter (Suraj et al., 2020).

2.7.5. *Chatbot*

Chatbot atau mesin *bot* untuk tanya-jawab merupakan program atau aplikasi yang dirancang agar dapat melakukan obrolan *online* dengan manusia. *Chatbot* menerima *input* melalui teks, ucapan-ke teks, atau pemilihan opsi dan memberikan tanggapan dari serangkaian opsi yang telah ditentukan (Wintoro et al., 2022).

2.8. Contoh Perhitungan LSTM

Berikut ini contoh implementasi perhitungan *Forward Propagation* pada LSTM. Sebuah *cell* untuk melihat bagaimana *forward propagation* melalui waktu dalam LSTM direpresentasikan.

Data *input* pada *timestep* $t0$ dan $t1$:

$x_0 = \begin{bmatrix} 0{,}25 \\ 0{,}30 \end{bmatrix}$ dengan label $\begin{bmatrix} 0{,}5 \\ 0{,}3 \end{bmatrix}$

$x_1 = \begin{bmatrix} 1 \\ 0{,}80 \end{bmatrix}$ dengan label $\begin{bmatrix} 1{,}25 \\ 1 \end{bmatrix}$

Nilai bobot *matriks* yang sesuai yaitu:

$w^{(f)} = \begin{bmatrix} 0{,}11 & 0{,}32 \\ 0{,}42 & 0{,}19 \end{bmatrix}$, $w^{(i)} = \begin{bmatrix} 0{,}60 & 0{,}17 \\ 0{,}16 & 0{,}17 \end{bmatrix}$, $w^{(g)} = \begin{bmatrix} 0{,}46 & 0{,}74 \\ 0{,}75 & 0{,}65 \end{bmatrix}$,

$w^{(o)} = \begin{bmatrix} 0.98 & 0.08 \\ 0.15 & 0.54 \end{bmatrix}$

Nilai bobot *matriks* yang tersembunyi yaitu:

$U^{(f)} = \begin{bmatrix} 0{,}87 & 0{,}50 \\ 0{,}23 & 0{,}67 \end{bmatrix}$, $U^{(i)} = \begin{bmatrix} 0{,}30 & 0{,}89 \\ 0{,}64 & 0{,}65 \end{bmatrix}$, $U^{(g)} = \begin{bmatrix} 0{,}60 & 0{,}12 \\ 1{,}00 & 0{,}01 \end{bmatrix}$,

$U^{(o)} = \begin{bmatrix} 0{,}98 & 0{,}08 \\ 0{,}15 & 0{,}54 \end{bmatrix}$

$b^{(f)} = \begin{bmatrix} 0{,}30 & 0{,}1 \end{bmatrix}$, $b^{(i)} = \begin{bmatrix} 0{,}67 & 0{,}13 \end{bmatrix}$, $b^{(g)} = \begin{bmatrix} 0{,}47 & 0{,}07 \end{bmatrix}$, $b^{(o)} = \begin{bmatrix} 0{,}75 & 0{,}09 \end{bmatrix}$

Forward Propagation

1. Pertama-tama, dihitung t_0 menggunakan Persamaan (1) dan Persamaan (4) yang didefinisikan sebagai berikut:

$g_t = \tilde{C}_t = tanh\,(W^{(g)}x_t + U^{(g)}h_{t-1} + b^{(g)}$…….(4)

Perhitungan *step* pertama yaitu:

$$g_o = tanh\left(\begin{bmatrix} 0{,}46 & 0{,}75 \\ 0{,}74 & 0{,}65 \end{bmatrix}.\begin{bmatrix} 0{,}25 \\ 0{,}3 \end{bmatrix} + \begin{bmatrix} 0{,}6 & 1{,}00 \\ 0{,}12 & 0{,}01 \end{bmatrix}.\begin{bmatrix} 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0{,}47 & 0{,}07 \end{bmatrix}\right) = \begin{bmatrix} 0{,}66959 \\ 0{,}42190 \end{bmatrix};$$

Atau

$$g_o = tanh\left(\begin{bmatrix} 0{,}46 & 0{,}75 & 0{,}6 & 1{,}00 & 0{,}47 \\ 0{,}74 & 0{,}65 & 0{,}12 & 0{,}01 & 0{,}07 \end{bmatrix}.\begin{bmatrix} 0{,}25 \\ 0{,}3 \\ 0 \\ 0 \\ 1 \end{bmatrix}\right) = tanh\left(\begin{bmatrix} 0{,}81 \\ 0{,}45 \end{bmatrix}\right) = \begin{bmatrix} 0{,}66959 \\ 0{,}42190 \end{bmatrix}$$

dan serupa dalam menghitung,

$f_0 = \begin{bmatrix} 0{,}61147 \\ 0{,}55897 \end{bmatrix}$, $i_0 = \begin{bmatrix} 0{,}70432 \\ 0{,}55564 \end{bmatrix}$, $o_0 = \begin{bmatrix} 0{,}73885 \\ 0{,}56758 \end{bmatrix}$.

Karena tidak ada $C_{t\text{-}1}$, *Memory state* pada waktu t_0 adalah $C_0 = \begin{bmatrix} 0{,}47161 \\ 0{,}23442 \end{bmatrix}$.

1. Akhirnya, *Cell output* pada t_0 adalah $h_0 = \begin{bmatrix} 0{,}32472 \\ 0{,}13067 \end{bmatrix}$.

2. Selanjutnya, setelah mengulangi langkah 1 - 4 untuk *timestep* t_1, maka

$f_1 = \begin{bmatrix} 0{,}74248 \\ 0{,}69481 \end{bmatrix}$; $i_1 = \begin{bmatrix} 0{,}82911 \\ 0{,}69219 \end{bmatrix}$; $o_1 = \begin{bmatrix} 0{,}88740 \\ 0{,}69463 \end{bmatrix}$; $g_1 = \begin{bmatrix} 0{,}95231 \\ 0{,}87876 \end{bmatrix}$.

Kemudian, didapat hasil $C_1 = \begin{bmatrix} 1{,}13973 \\ 0{,}77115 \end{bmatrix}$, $h_1 = \begin{bmatrix} 0{,}72263 \\ 0{,}44984 \end{bmatrix}$.

2.9. *Preprocessing* Data

Data *preprocessing* adalah salah satu tugas data *mining* yang meliputi persiapan dan transformasi data ke dalam bentuk yang sesuai dengan prosedur *mining*. Data *preprocessing* bertujuan untuk mengurangi ukuran data, menemukan hubungan antara data, menormalkan data, menghapus *outlier* dan mengekstrak fitur data. Hal ini mencakup beberapa teknik seperti pembersihan data, integrasi, transformasi dan reduksi (Alasadi & Baya, 2017).

2.10. Tokenisasi

Menurut Gregory., (1999), Tokenisasi adalah salah satu langkah awal dalam transformasi selama pemrosesan bahasa alami. Tokenisasi berarti membagi teks *input*, yang bagi komputer hanya berupa satu rangkaian karakter yang panjang, menjadi subunit-subunit, yang disebut *token*. *Token-token* ini kemudian

dimasukkan ke dalam langkah-langkah pemrosesan bahasa alami yang berurutan seperti analisis *morfologi*, penandaan kelas kata, dan penguraian. Karena proses-proses selanjutnya ini biasanya dirancang untuk bekerja pada kalimat-kalimat individual, tugas tambahan dari tokenisasi sering kali adalah mengidentifikasi batas-batas kalimat dan juga batas-batas *token*. Meskipun jarang dibahas, dan dengan cepat terlewatkan, tokenisasi dalam sistem pemrosesan teks otomatis menimbulkan sejumlah pertanyaan pelik, beberapa di antaranya memiliki jawaban yang benar-benar sempurna. Proses tokenisasi sangat bergantung pada jenis teks yang sedang diproses, sehingga analisis masalah tokenisasi pada jenis teks tertentu harus dilakukan (dengan cara memeriksa status pemisah/non-pemisah karakter).

Tokenisasi adalah langkah penting dalam banyak tugas pemrosesan bahasa alami karena menyederhanakan data *input* dan membuatnya lebih mudah untuk diproses. Hal ini juga memungkinkan algoritme untuk menganalisis dan memahami makna yang mendasari teks dengan memecahnya menjadi unit-unit yang lebih kecil dan lebih mudah dikelola.

Terdapat tiga teknik tokenisasi dengan masing-masing teknik bekerja secara berbeda dan memiliki kelebihan dan kekurangannya sendiri. Berikut ketiga teknik tersebut:

a. Tokenisasi berbasis kata (*word-based tokenization*)
   Ini adalah teknik tokenisasi yang paling umum digunakan. Teknik ini membagi sepotong teks menjadi kata-kata berdasarkan pembatas. Pembatas yang paling umum digunakan adalah spasi. Selain itu, dapat memisahkan teks menggunakan lebih dari satu pembatas, seperti spasi dan tanda baca. Bergantung pada pembatas yang digunakan, akan didapatkan token tingkat kata yang berbeda.

b. Tokenisasi berbasis karakter (*character-based tokenization*)
   Tokenizer berbasis karakter membagi teks mentah menjadi karakter individual. Logika di balik tokenisasi ini adalah bahwa suatu bahasa memiliki banyak kata yang berbeda tetapi memiliki jumlah karakter yang tetap. Ini menghasilkan kosa kata yang sangat sedikit.

c. Tokenisasi berbasis subkata (*subword-based tokenization*)

   Tokenisasi populer lainnya adalah tokenisasi berbasis subkata yang merupakan solusi antara tokenisasi berbasis kata dan karakter. Ide utamanya adalah untuk memecahkan masalah yang dihadapi oleh tokenisasi berbasis kata (ukuran kosakata yang sangat besar, jumlah *token Out-Of-Vocabulary* (*OOV*) yang banyak, dan arti berbeda dari kata yang sangat mirip) dan tokenisasi berbasis karakter (urutan yang sangat panjang dan *token* individu yang kurang bermakna).

Pada penelitian ini menggunakan *character-based tokenization* untuk memproses data DNA, contoh implementasi menggunakan *character-based tokenization* ditunjukkan pada Tabel 3.

Tabel 3. Implementasi *character-based tokenization*

Sebelum Tokenisasi	**Sesudah Tokenisasi**
ATCG	{'A'= 1, 'T'= 2, 'C'= 3, 'G'= 4}

Dari tokenisasi di atas diberikan data *sequence* untuk kemudian diterapkan, hasilnya dapat dilihat pada Tabel 4.

Tabel 4. Hasil Tokenisasi

Sequence	**Hasil Tokenisasi**
AATTTTTATA	[1 1 2 2 2 2 2 1 2 1]
CAAAGATTTC	[3 1 1 1 4 1 2 2 2 3]
ATGACTGGAG	[1 2 4 1 3 2 4 4 1 4]

2.11. *Padding*

Masalah utama dari *one-hot encoding* adalah bahwa setiap data *sequence* DNA memiliki panjang yang berbeda, sementara semua *vektor input* harus memiliki ukuran yang sama untuk dimasukkan ke dalam model. Untuk mengatasi masalah ini, *padding* biasanya diterapkan. Hal ini berarti menerapkan panjang yang sama (*max_length*) untuk semua *sequence* DNA dan kemudian memotong DNA yang lebih panjang hingga panjang tersebut atau mengisi DNA yang lebih pendek dengan

karakter “buatan” sepanjang *sequence* yang ditentukan (Del Rio, et al., 2020). Dalam proses *padding* tersebut juga, *max_length* sebagai parameter untuk mengatur panjang maksimal dari tiap *sequence*.

Misalkan parameter *max_length* diisi dengan nilai 5, maka panjang masing-masing kalimat secara otomatis tidak akan melebihi 5. Proses melengkapi *sequence* ini disebut *padding* dan karakter yang digunakan untuk mengisi dapat berupa karakter apa pun yang tidak digunakan dalam *sequence* itu sendiri. Untuk masalah ini, karakter nol (“0”) adalah yang paling umum digunakan. Penambahan angka 0 sebelum *sequence* disebut dengan *pre-padding* sedangkan penambahan angka 0 sesudah *sequence* disebut dengan *post-padding*. Perbedaan antara *pre* dan *post padding* dapat dilihat pada Tabel 5.

Tabel 5. Jenis-jenis *Padding*

Sebelum *padding*	***Pre-padding***	***Post-padding***
sequence 1: [1, 2, 3]	*sequence* 1 : [0, 0, 1, 2, 3]	*sequence* 1 : [1, 2, 3, 0, 0]
sequence 2: [2, 3, 4, 5]	*sequence* 2 : [0, 2, 3, 4, 5]	*sequence* 2 : [2, 3, 4, 5, 0]

2.12. *Embedding Layer*

Embedding layer merupakan *hidden layer* yang menerima tiga argumen, yang pertama yaitu *input dimension* yang menerima sebuah nilai dari banyaknya variasi *token*/kata dalam *dataset*, yang kedua yaitu *output dimension* yang menerima sebuah nilai sebagai sebuah *vektor* dari setiap *token*/kata yang mana akan menjadi nilai *output* dari *layer* ini, dan yang ketiga yaitu *input length* yang menerima nilai dari panjang sekuens *dataset*. *Embedding layer* akan mentransformasikan setiap *token* menjadi *vektor* dengan panjang yang telah ditentukan (Fauzi & Romadhony, 2021).

2.13. *Flatten Layer*

Flatten layer adalah *layer* yang digunakan untuk mengubah data multi-dimensi menjadi *array* 1 dimensi untuk dimasukkan ke *layer* berikutnya. Meratakan *output* dari *convolutional layer* untuk membuat *vektor* fitur tunggal yang panjang. *Vektor* fitur ini terhubung ke model klasifikasi akhir, yang disebut *Fully-connected layer*.

Flatten layer digunakan untuk membuat *input* multi-dimensi menjadi satu dimensi, biasanya digunakan dalam transisi dari *convolution layer* ke *Fully-connected layer*. Berdasarkan apakah *TensorSpace* Model memuat model yang sudah dilatih sebelum diinisialisasi, mengkonfigurasi *layer* dengan cara yang berbeda.

2.14. *Dropout Layer*

Dropout layer adalah *layer* pada arsitektur *neural network* yang pada tujuannya adalah untuk mengontrol fenomena *overfitting* pada model. Kisaran *Dropout layer* berada di antara 0-1. *Dropout layer* dimasukkan setelah *embedding layer*, untuk menonaktifkan aktivasi *neuron* secara acak pada *embedding layer*. *Dropout layer* secara acak menonaktifkan nilai-nilai tertentu pada *embedding layer* dengan mengubah nilainya menjadi 0 (Asghar, et al., 2020).

2.15. *Dense Layer*

Dense layer (*Fully Connected Layer*) merupakan salah satu model tradisional *neural network* yang biasa digunakan pada tahap akhir pemodelan. *Layer* ini digunakan untuk melakukan klasifikasi sesuai kategori kelas pada *output* dan membantu mengubah dimensi *output* dari *layer* sebelumnya sehingga model dapat dengan mudah menentukan hubungan antara nilai data tempat model bekerja. *Dense layer* pada tahap terakhir digunakan untuk mengklasifikasikan fitur *input* ke dalam kelas (Ullah, et al., 2022). *Dense layer* memiliki *input* dan *output* yang jumlahnya tergantung dengan kategori kelas yang diprediksi (Andros, et al., 2015).

2.16. *Confusion Matrix*

Confusion Matrix adalah salah satu metode yang sering digunakan dalam evaluasi model *machine learning* seperti mengukur performa klasifikasi (Kulkarni et al., 2020). *Confusion Matrix* dapat mengevaluasi *binary classification* ataupun *multi-class classification*. *Confusion Matrix* adalah tabel dengan empat kombinasi berbeda dari nilai prediksi dan nilai aktual. Ada empat istilah untuk pengukuran kinerja pada *Confusion Matrix*, diantaranya: *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN). Contoh evaluasi *Confusion Matrix* ditunjukkan pada Tabel 6.

Tabel 6. *Confusion Matrix* Pada Klasifikasi Dua Kelas (Annisa, 2017)

	Predicted Positive	***Predicted Negative***
Actual Positive	TP *(True Positive)*	FN *(False Negative)*
Actual Negative	FP *(False Positive)*	TN *(True Negative)*

Adapun istilah yang digunakan pada *Confusion Matrix* berdasarkan Tabel 6, diantaranya (Chicco et al., 2021):

a. *True Positive* (TP):
   Diklasifikasikan jumlah data positif dengan benar oleh sistem.
b. *True Negative* (TN):
   Diklasifikasikan jumlah data negatif dengan benar oleh sistem.
c. *False Positive* (FP):
   Diklasifikasikan jumlah data positif dengan salah oleh sistem.
d. *False Negative* (FN):
   Diklasifikasikan jumlah data negatif dengan salah oleh sistem.

Beberapa *matriks* pengukuran yang digunakan untuk mengevaluasi kinerja prediksi model pada penelitian ini yaitu:

a. *Accuracy*/ Akurasi
   Accuracy merupakan perbandingan prediksi benar berdasarkan hasil positif dan negatif dengan total data. *Accuracy* adalah *matriks* yang paling umum untuk melakukan evaluasi klasifikasi. *Accuracy* bekerja dengan cara menghitung nilai *probabilitas* berdasarkan nilai yang benar dari label kelas. Perumusan *Accuracy* dapat dilihat pada Persamaan (5) (Bekkar et al., 2013).

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad \text{.....(5)}$$

b. *Precision*/ Presisi
   Precision merupakan nilai kebenaran dari prediksi yang dilakukan oleh *classifier* dengan label kelas yang sudah ada. Perumusan *Precision* dapat dilihat pada Persamaan (6) (Bekkar et al., 2013).

$$Precision = \frac{TP}{TP+FP} \quad \text{.....(6)}$$

c. *Recall*/ Sensitivitas

Recall atau *Sensitivity* adalah keakuratan dari nilai positif yang ada. Perumusan *Sensitivity* dapat dilihat pada Persamaan (7) (Bekkar et al., 2013).

$$Sensitivity = \frac{TP}{TP+FN} \quad \text{.....(7)}$$

III. METODOLOGI PENELITIAN

3.1. Tempat dan Waktu Penelitian

Berikut adalah pemaparan tempat penelitian dan waktu serta jadwal penelitian:

3.1.1. Tempat Penelitian

Penelitian dilakukan di Laboratorium Rekayasa Perangkat Lunak yang bertempat di gedung MIPA Terpadu Jurusan Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung. Penelitian ini dilaksanakan tepatnya di Laboratorium Rekayasa Perangkat Lunak menggunakan beberapa fasilitas komputasi yang telah disediakan, yaitu untuk menginformasikan beberapa PC RPL.

3.1.2. Waktu dan Jadwal Penelitian

Penelitian dimulai pada bulan Desember 2022 di semester tujuh ganjil hingga target penyelesaian pada bulan Agustus 2023. Alur waktu pengerjaan dapat dilihat pada Tabel 7.

Tabel 7. Alur Waktu dan Jadwal Penelitian

No	Kegiatan	2022																2023																							
		Desember				Januari				Februari				Maret				April				Mei				Juni				Juli				Agustus							
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1	Studi Literatur																																								
2	Pengumpulan Data																																								
4	*Preprocessing* Data																																								
5	*Tokenization*																																								
6	*Padding*																																								
6	Penggunaan 50% Data																																								
7	Pemodelan LSTM																																								
8	Evaluasi Kinerja Awal																																								
9	Penggunaan 100% Data																																								
10	Pemodelan LSTM																																								
11	Evaluasi Kinerja Akhir																																								
12	Penyusunan Laporan																																								

3.2. Data dan Alat

Berikut data dan alat yang digunakan selama penelitian:

3.2.1. Data

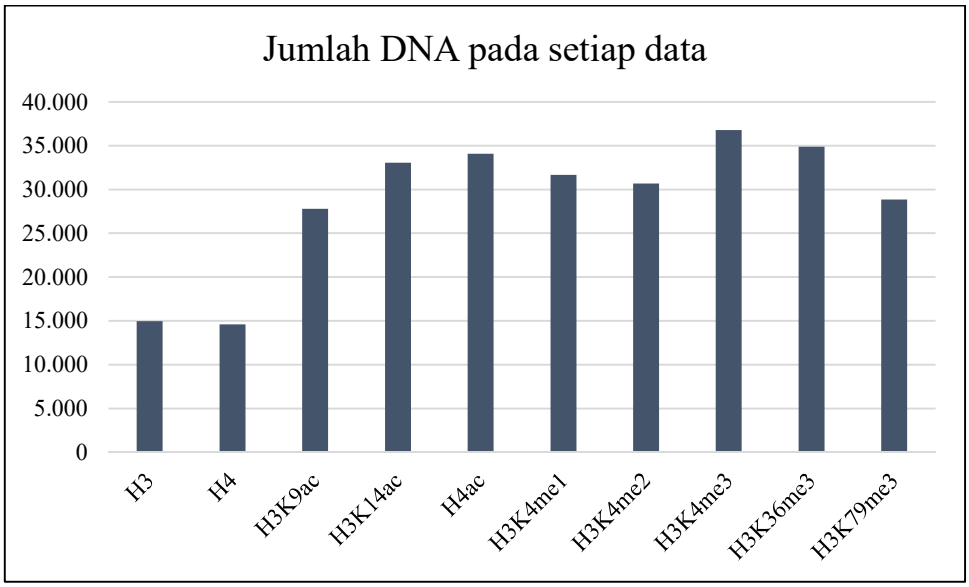
Data yang digunakan adalah data pada penelitian Pokholok et al., (2005). Data berisi kumpulan *untaian* DNA yang membungkus protein *Histone* dengan panjang *nukleotida*, yaitu 500. "H3" dan "H4" yang terdapat dalam setiap *dataset* menunjukkan jenis *Histone*. "K" dan angka yang muncul setelahnya menunjukkan *asam amino* yang dimodifikasi (misalnya, "K14" menunjukkan *asam amino* ke-14, dan "K" telah dimodifikasi). "ac," dan "me," menandakan jenis modifikasi yang terjadi (*Asetilasi* atau *Metilasi*) dan angka setelah "me" menunjukkan waktu *Metilasi* (misalnya, "me2" menandakan terjadinya *Dimetilasi*). Di setiap *dataset*, sampel adalah urutan dengan panjang 500 pasangan basa dan termasuk dalam kelas "Positif" atau "Negatif". Sampel dalam kelas "Positif" mengandung daerah yang membungkus protein *Histone*. Sebaliknya, sampel di kelas "Negatif" tidak mengandungnya. Dengan *dataset* ini, kita dapat memprediksi profil *Histone* dari sekuens dengan tingkat akurasi tertentu, dan dapat membantu untuk memahami tentang pola ekspresi gen. Data tersebut juga telah digunakan pada penelitian (Higashihara et al., 2008; Nguyen et al., 2016; dan Mahmoud & Guo, 2021). Data tersebut bisa dilihat pada Tabel 8.

Tabel 8. Data DNA Ragi (Pokholok et al., 2005)

No	Dataset	Deskripsi	Positif	Negatif	Total	Rentang Panjang Sekuens
1	H3	Keberadaan H3	7.667	7.298	14.965	290-500
2	H4	Keberadaan H4	6.480	8.121	14.601	310-500
3	H3K9ac	*Asetilasi* H3K9 terhadap H3	15.415	12.367	27.782	290-500
4	H3K14ac	*Asetilasi* H3K14 terhadap H3	18.771	14.277	33.048	290-500
5	H4ac	*Asetilasi* H4 terhadap H3	18.410	15.685	34.095	290-500
6	H3K4me1	*Monometilasi* H3K4 terhadap H3	17.266	14.441	31.677	290-500

No	Dataset	Deskripsi	Positif	Negatif	Total	Rentang Panjang Sekuens
7	H3K4me2	*Dimetilasi* H3K4 terhadap H3	18.143	12.540	30.683	290-500
8	H3K4me3	*Trimetilasi* H3K4 terhadap H3	19.604	17.195	36.799	290-500
9	H3K36me3	*Trimetilasi* H3K36 terhadap H3	18.892	15.988	34.880	310-500
10	H3K79me3	*Trimetilasi* H3K79 terhadap H3	15.337	13.500	28.837	310-500

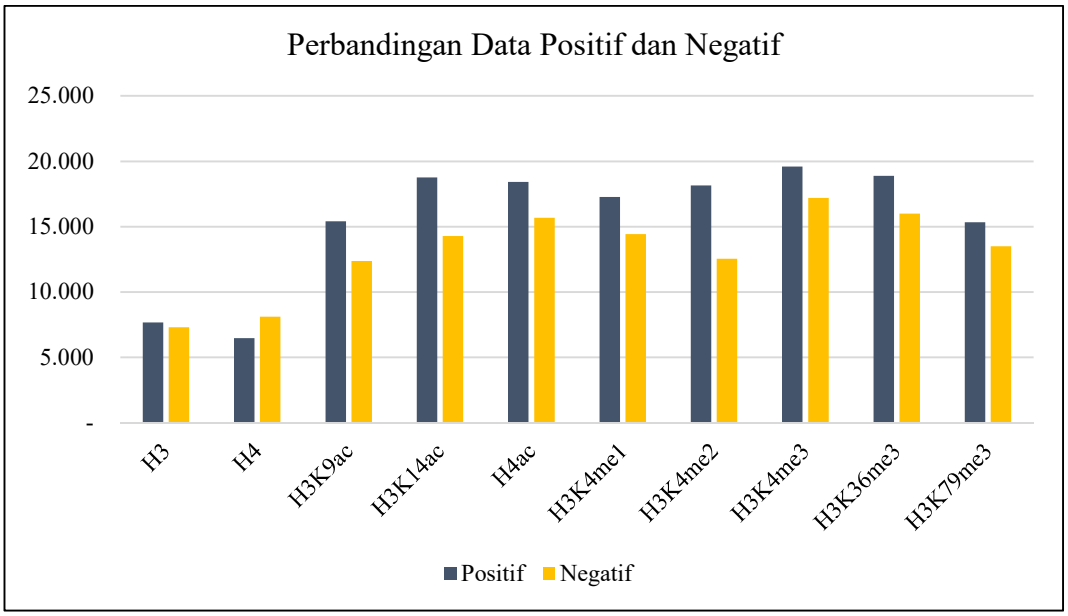
Berdasarkan Tabel 8, menunjukkan data yang digunakan pada penelitian ini, yaitu sepuluh *dataset* yang akan digunakan pada penelitian. Data tersebut diantaranya adalah H3, H4, H3K9ac, H3K14ac, H4ac, H3K4me1, H3K4me2, H3K4me3, H3K36me3, dan H3K79me3 dengan rerata panjang sekuens dari setiap *dataset* adalah 500 *nukleotida*. Data diambil dari penelitian Pokholok et al., pada tahun 2005. Berikut adalah perbandingan total setiap data, ditunjukkan pada Gambar 11.



Gambar 11. Perbandingan Total Setiap Data.

Gambar 11, menunjukkan total *untaian* DNA dari setiap *dataset* yang akan digunakan. Dapat dilihat *dataset* dengan jumlah data terbesar adalah *dataset* H3K4me3 dengan jumlah data sebanyak 36.799. Sementara *dataset* dengan jumlah data terkecil adalah *dataset* H4 dengan data DNA sebanyak 14.601. Lalu untuk rata-rata jumlah data positif dari setiap *dataset* adalah sebanyak 15.599, dan rata-rata

jumlah data negatif dari setiap *dataset* adalah sebanyak 13.141, sedangkan rata-rata keselurahan data adalah sebanyak 14.370. Grafik perbandingan data positif dan negatif pada setiap *dataset* ditunjukkan pada Gambar 12 berikut.



Gambar 12. Perbandingan Data Positif dan Negatif pada setiap *dataset*.

Gambar 12, menunjukkan jumlah data positif dan negatif DNA dari setiap *dataset* yang akan digunakan. Dapat dilihat bahwa setiap *dataset* memiliki jumlah data positif dan negatif yang hampir seimbang sehingga tidak perlu dilakukan *oversampling* ataupun *undersampling*. Dapat dilihat juga *dataset* dengan selisih data positif dan negatif terbesar adalah *dataset* H3K4me2 dengan selisih data sebanyak 5.603. Sementara *dataset* dengan selisih data positif dan negatif terkecil adalah *dataset* H3 dengan selisih data DNA sebanyak 369. Data positif terbesar terdapat di *dataset* H3K4me3 sebesar 19.604 dan Data negatif terbesar adalah sebanyak 17.195 juga di *dataset* H3K4me3. Kemudian Data positif terkecil terdapat di *dataset* H4 sebesar 6.480 sedangkan Data negatif terkecil adalah sebanyak 7.298 di *dataset* H3.

3.2.2. Alat

Berikut perangkat penelitian yang digunakan:

3.2.2.1. Perangkat Keras (*Hardware*)

Perangkat keras yang digunakan untuk penelitian ini adalah sebagai berikut:

a. *Processor*: AMD Ryzen 5 3500U with Radeon Vega Mobile Gfx 2.10 GHz
b. *Random Access Memory* (RAM): 8,00 GB 2400 MHz
c. *System Type*: 64-bit operating system, x64-base processor
d. *Storage*: SSD 250 GB
e. *Network Interface*: Realtek RTL8821CE 802.11ac PCIe Adapter
f. *Video Graphics Array* (VGA) : AMD Radeon (TM) Vega 8 Graphics

3.2.2.2. Perangkat Lunak (*Software*)

Perangkat lunak yang digunakan untuk penelitian ini adalah sebagai berikut:

a. *Operating System*:
   - Windows 10 Home Single Language 64-bit operating system

b. *Tools*:
   - *Jupyter Notebook*

     Jupyter Notebook adalah sebuah aplikasi *web open-source* yang memungkinkan pengguna untuk membuat dan berbagi dokumen interaktif yang berisi kode, teks, gambar, *visualisasi*, dan *output* dari eksekusi kode. Dokumen ini disebut "*notebook*" dan biasanya digunakan untuk melakukan pemrosesan data, analisis data, *visualisasi*, serta penelitian ilmiah.

   - *Python 3.11.1*

     Python adalah bahasa pemrograman tingkat tinggi yang diinterpretasikan dan banyak digunakan dalam pengembangan perangkat lunak, data *science*, dan *machine learning* serta pemrosesan data. *Python* dikenal karena keterbacaan dan kemudahan penggunaannya, yang membuatnya menjadi bahasa yang populer untuk pemula maupun pengembang berpengalaman. Salah satu kekuatan *Python* adalah *library* standarnya yang luas, yang

menyediakan berbagai macam modul dan fungsi yang dapat digunakan untuk berbagai tugas, mulai dari *input/output* file dasar hingga jaringan yang kompleks dan pengembangan *web*. Selain itu, ada banyak *library* pihak ketiga yang tersedia untuk *Python*, yang memperluas fungsinya lebih jauh lagi.

c. *Packages*:

- *Library Pandas 1.5.3*

  Pandas merupakan *library* yang digunakan untuk menyimpan data dalam bentuk *tabulasi*. *Library* ini juga dapat mempermudah dalam pemrosesan data seperti pada saat pembacaan file ke dalam program.

- *Library Numpy 1.23.5*

  Numpy merupakan sebuah *library* yang mendukung komputasi dengan kecepatan tinggi untuk *array*. *Library Pandas* juga menggunakan *numpy* untuk menyimpan data *array* pada tabelnya.

- *Library Tensorflow 2.10.0*

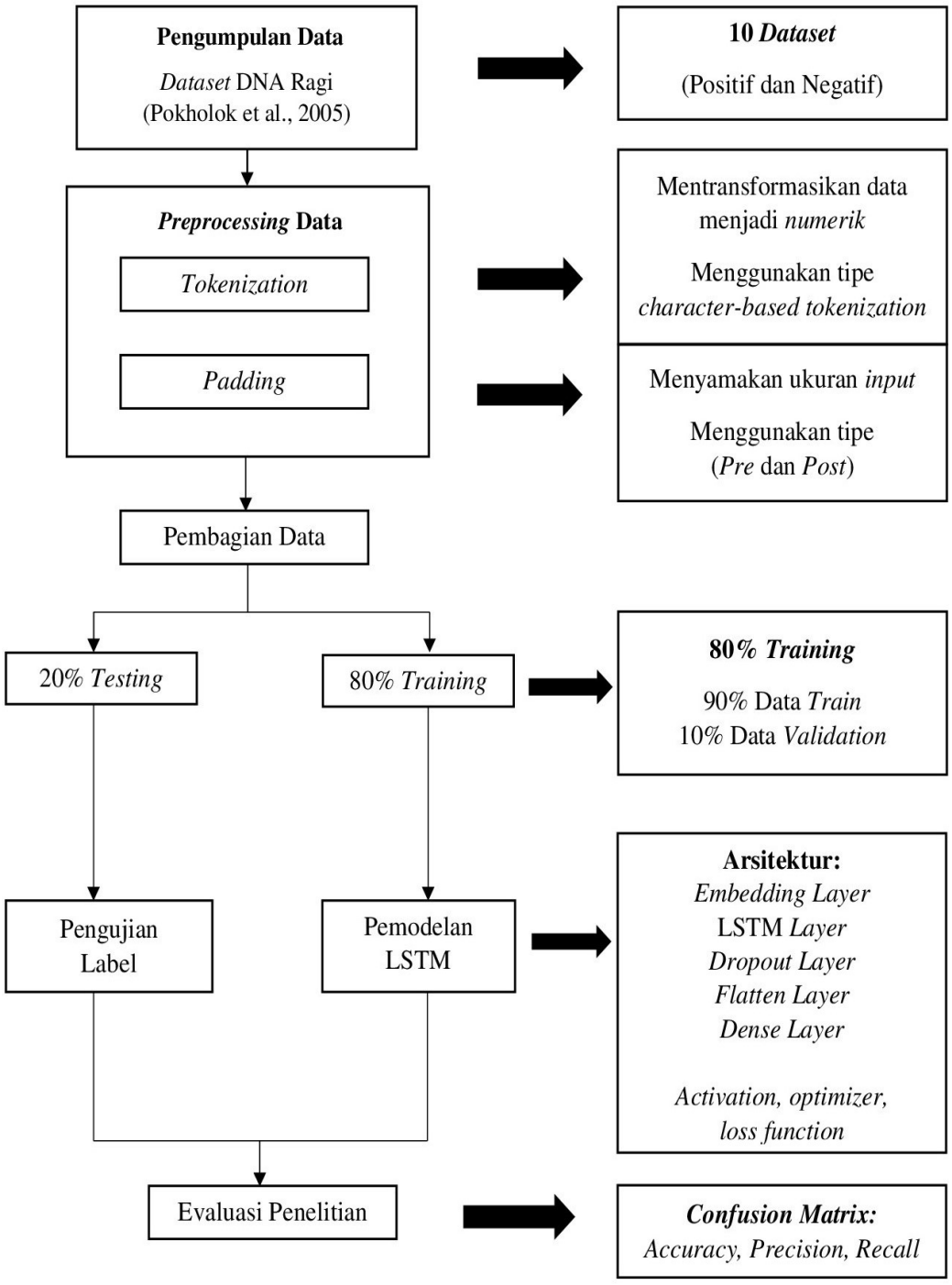
  Tensorflow adalah *library* yang dibuat oleh *google* untuk melakukan pemodelan *deep learning* dengan mudah. *Tensorflow* merupakan *library* yang berjalan pada bahasa pemrograman yang lebih rendah (mendekati bahasa mesin) sehingga memiliki performa komputasi yang baik. *Tensorflow* juga telah memuat *Keras* yang sering digunakan juga untuk pemodelan *deep learning*.

- *Library Scikit-learn 1.2.2*

  Scikit-learn merupakan modul yang dibangun berdasarkan *Numpy*, *SciPy* dan *Matplotlib*. *Scikit-learn* memudahkan dalam *processing* data ataupun *training*. *Library* ini juga biasa digunakan untuk membagi data (*split data*) untuk *training* dan *testing*.

3.3. Metodologi

Alur kerja penelitian ini berdasarkan penelitian sebelumnya dan akan melalui beberapa tahapan. Diagram alur kerja penelitian tentang klasifikasi pada DNA Ragi dapat dilihat pada Gambar 13.



Gambar 13. Alur Kerja Penelitian Klasifikasi Pada DNA Ragi Menggunakan Metode LSTM.

Pada Gambar 13, alur kerja penelitian memiliki enam proses utama yaitu pengumpulan data, *preprocessing* data, transformasi data, pembagian atau pemisahan data, pemodelan dan klasifikasi, serta evaluasi penelitian. Berikut penjelasan dari setiap tahap pada Gambar 13.

3.3.1. Pengumpulan Data

Pada tahap pertama, data awal akan dikumpulkan terlebih dahulu. Penelitian yang dilakukan oleh (Pokholok et al., 2005) merupakan sumber data yang diperoleh, yaitu *Dataset* DNA Ragi yang terdiri dari sepuluh *dataset*, yaitu H3, H4, H3K9ac, H3K14ac, H4ac, H3K4me1, H3K4me2, H3K4me3, H3K36me3, dan H3K79me3. Dari masing-masing *dataset* berisikan kelas data positif dan negatif dari keseluruhan jumlah data.

3.3.2. *Preprocessing* Data

Pada tahap ini, data yang telah dikumpulkan akan dilakukan *preprocessing* data. Awalnya *dataset* yang digunakan masih dalam bentuk format text lalu kemudian diubah menjadi dalam bentuk format data CSV. Selanjutnya dilakukan pembacaan pada *dataset*, kemudian mengecek panjang *sequence* DNA dari masing-masing *dataset*. Tujuan mengubah file ke dalam bentuk format data CSV adalah agar pada saat teks dan angka yang disimpan dalam file CSV, mudah untuk memindahkannya dari satu program ke program lain dan memungkinkan data disimpan dalam format tabel terstruktur.

3.3.3. Pembagian Data

Setelah dilakukan *preprocessing* data, tahap selanjutnya yaitu dari tiap *dataset* akan dilakukan pembagian atau pemisahan data yang terdiri dari data *training* (latih) dan *testing* (uji). Data *training* akan digunakan untuk proses pemodelan, sedangkan data *testing* akan digunakan untuk pengujian setelah pemodelan selesai. Pada Data *training* akan digunakan 80% dari *dataset*, sedangkan pada data *testing* akan digunakan 20% dari *dataset*. Pembagian data *training* dan data *testing* dengan *rasio* 80-20 digunakan agar eksperimen memiliki *rasio* yang sama dengan penelitian sebelumnya. Sebelum dilanjutkan tahap transformasi data, data *training* yang

sebelumnya 80% akan dibagi lagi dengan skema pembagian yaitu 90% untuk data *train* dan 10% untuk data *validation*.

3.3.4. Transformasi Data

Tujuan dari transformasi data ini adalah untuk memungkinkan model LSTM menerima *input* dari data *untaian* DNA. Selanjutnya adalah tahapan untuk representasi *untaian* atau *sequence* DNA dengan dua tahapan, yaitu pertama melakukan tokenisasi pada data *sequence* DNA. Proses tokenisasi bertujuan untuk mengubah *sequence* DNA yang sebelumnya berupa karakter *string* menjadi *numerik*. Proses ini menggunakan *library* bernama *tokenizer* dari *tensorflow*. Lalu untuk tipe tokenisasi yang digunakan adalah tipe tokenisasi *character-based tokenization*. Kemudian tahap kedua adalah melakukan *padding* yang bertujuan untuk menyamakan ukuran *input* sebelum masuk ke dalam model. Data *sequence* yang sudah diubah menjadi bentuk *numerik* selanjutnya dilakukan proses *padding* yang bertujuan agar *input sequence* memiliki panjang yang sama. Proses *padding* dilakukan pada *sequence* DNA menggunakan tipe *pre* dan *post padding* dengan panjang maksimumnya sejumlah *sequence* terpanjang, yaitu 500 DNA.

3.3.5. Pemodelan dan Klasifikasi

Untuk melakukan pengklasifikasian pada data, maka selanjutnya model LSTM akan dibuat. Sebanyak 50% dari data akan digunakan di tahap awal. Semua data yang tersedia akan digunakan untuk pelatihan setelah model yang dievaluasi menunjukkan akurasi yang baik. Model yang akan dibuat kemudian dilatih menggunakan data *training* (latih). Parameter model LSTM akan disesuaikan pada tahapan ini. Sebelumnya pada tahap pembagian data, data *training* telah kembali dibagi menjadi data *train* dan *validation*, yang digunakan dalam proses ini. Sebanyak 90% dari data *training* adalah data *train*, sedangkan 10% dari data *training* adalah data *validation*.

Pembagian data *validation* dari data *training* bertujuan untuk memonitor proses pembelajaran model. Selanjutnya kedua data tersebut dilatih dan diklasifikasi menggunakan metode *Long Short-Term Memory* (LSTM). Pemodelan dilakukan dengan menggunakan beberapa *layer* yaitu *Embedding layer*, LSTM *layer*, *Dropout layer*, *Flatten layer* dan *Dense layer*, serta menggunakan *activation function* ReLU dan *optimizer Adam*. Proses pelatihan akan berhenti saat akurasi pada data validasi tidak bertambah baik lagi. Parameter yang digunakan menggunakan rujukan dari penelitian oleh (Vazhayil, et al., 2018) ditunjukkan pada Tabel 9.

Tabel 9. Parameter Model

Type	**Jumlah/Ukuran**
Embedding	128
LSTM	128/64
Dropout	0,5
Dense	2

Embedding layer merupakan lapisan pertama setelah proses *padding*. Dalam *Embedding layer* terdapat 3 parameter yaitu *input* dimensi, *output* dimensi dan *input length*. *Input* dimensi adalah ukuran kosa kata dimana pada penelitian ini memiliki 4 karakter *nukleotida* (A, T, G, C) dan karakter *padding* (0) sehingga *input* dimensinya adalah 5. *Output* dimensi adalah panjang vektor dalam setiap karakter, pada rujukan arsitektur pada Tabel 9, jumlah *output* dimensinya adalah 128. *Input length* adalah panjang maksimum urutan, panjang maksimum urutan ini mengikuti jumlah panjang maksimum pada *padding*. Pada *layer* LSTM di Tabel 9, *neuron* yang digunakan adalah 128 dan 64 yang diikuti jumlah *dropout* 0,5. Selanjutnya diikuti dengan *fully connected layer* (*Dense Layer*) dengan jumlah kelasnya yaitu 2.

3.3.6. Evaluasi Penelitian

Setelah model dilakukan *Training* dan *Testing*, berikutnya model akan dievaluasi kinerjanya dengan menggunakan parameter evaluasi *Confusion Matrix* yaitu *Accuracy*, *Precision*, dan *Recall*. *Precision* adalah kecocokan antara bagian data yang diambil dengan informasi yang dibutuhkan. *Recall* merupakan tingkat

keberhasilan sistem dalam menemukan kembali sebuah informasi. *Accuracy* adalah tingkat kedekatan antara nilai yang didapat terhadap nilai sebenarnya. *Accuracy*, *Precision*, dan *Recall* dapat dihitung menggunakan *Confusion Matrix*. Ukuran besaran *Accuracy*, *Precision*, dan *Recall* biasanya diberi nilai dalam bentuk presentase antara 1 sampai 100%. Sebuah sistem akan dianggap baik jika tingkat *Accuracy*, *Precision*, dan *Recall* -nya tinggi. Hasil yang didapatkan akan dibandingkan dengan penelitian terdahulu.

V. SIMPULAN DAN SARAN

5.1. Kesimpulan

Dari penelitian dan pembahasan yang sudah dilakukan mengenai klasifikasi DNA Ragi menggunakan metode *Long Short-Term Memory* (LSTM) dapat diambil beberapa kesimpulan, antara lain:

A. Dengan melakukan sebanyak enam kali *Experiment* didapatkan hasil sebagai berikut:
   - Pada Pelatihan *Experiment* 1 menghasilkan rerata *Accuracy* sebesar 61,40%, *Precision* sebesar 66,17%, dan *Recall* sebesar 63,52%. Pada Pengujian *Experiment* 1 menghasilkan rerata *Accuracy* sebesar 61,62%, *Precision* sebesar 65,79%, dan *Recall* sebesar 63,82%. Tidak terdapat perbedaan yang cukup besar antara hasil Pelatihan dan hasil Pengujian, tetapi rerata *Accuracy* Pengujian lebih baik dari Pelatihan.

   - Pada Pelatihan *Experiment* 2 menghasilkan rerata *Accuracy* sebesar 60,70%, *Precision* sebesar 63,12%, dan *Recall* sebesar 71,77%. Pada Pengujian *Experiment* 2 menghasilkan rerata *Accuracy* sebesar 60,86%, *Precision* sebesar 62,61%, dan *Recall* sebesar 71,76%. Tidak terdapat perbedaan yang cukup besar antara hasil Pelatihan dan hasil Pengujian, tetapi rerata *Accuracy* Pengujian lebih baik dari Pelatihan.

   - Pada Pelatihan *Experiment* 3 menghasilkan rerata *Accuracy* sebesar 62,55%, *Precision* sebesar 62,51%, dan *Recall* sebesar 81,61%. Pada Pengujian *Experiment* 3 menghasilkan rerata *Accuracy* sebesar 62,92%, *Precision* sebesar 62,27%, dan *Recall* sebesar 82,08%. Tidak terdapat

perbedaan yang cukup besar antara hasil Pelatihan dan hasil Pengujian, tetapi rerata *Accuracy* Pengujian lebih baik dari Pelatihan.

- Pada Pelatihan *Experiment* 4 menghasilkan rerata *Accuracy* sebesar 62,69%, *Precision* sebesar 63,16%, dan *Recall* sebesar 77,50%. Pada Pengujian *Experiment* 4 menghasilkan rerata *Accuracy* sebesar 63,10%, *Precision* sebesar 62,95%, dan *Recall* sebesar 77,80%. Tidak terdapat perbedaan yang cukup besar antara hasil Pelatihan dan hasil Pengujian, tetapi rerata *Accuracy* Pengujian lebih baik dari Pelatihan.

- Pada Pelatihan *Experiment* 5 menghasilkan rerata *Accuracy* sebesar 62,43%, *Precision* sebesar 63,19%, dan *Recall* sebesar 87,79%. Pada Pengujian *Experiment* 5 menghasilkan rerata *Accuracy* sebesar 62,42%, *Precision* sebesar 62,82%, dan *Recall* sebesar 87,80%. Pada *Experiment* ini rerata *Accuracy* dari Pengujian sedikit lebih rendah dari rerata *Accuracy* Pelatihan.

- Pada Pelatihan *Experiment* 6 menghasilkan rerata *Accuracy* sebesar 62,39%, *Precision* sebesar 62,69%, dan *Recall* sebesar 90,93%. Pada Pengujian *Experiment* 6 menghasilkan rerata *Accuracy* sebesar 62,16%, *Precision* sebesar 62,09%, dan *Recall* sebesar 91,03%. Pada *Experiment* ini rerata *Accuracy* dari Pengujian sedikit lebih rendah dari rerata *Accuracy* Pelatihan.

B. Dapat disimpulkan pada penelitian ini bahwa hasil klasifikasi DNA Ragi terbaik didapatkan dari *Experiment* 4 yang memiliki rerata *Accuracy* Pengujian yang terbaik dibandingkan dari *Experiment* lainnya.

C. Dibandingkan dengan penelitian terdahulu, hasil *Experiment* pada penelitian ini ternyata belum berhasil meningkatkan *Accuracy* dari proses klasifikasi. Hal ini terjadi karena diduga pada model mengalami *Underfitting*.

5.2. Saran

Berikut adalah beberapa saran yang dapat dilakukan untuk penelitian lanjutan mengenai klasifikasi DNA Ragi menggunakan metode *Long Short-Term Memory* (LSTM).

A. Penelitian selanjutnya dapat mencoba untuk tidak hanya menggunakan nilai ukuran kata 1 *nukleotida* saja tetapi dengan jumlah *nukleotida* yang lebih besar.

B. Penelitian selanjutnya dapat mencoba berbagai teknik untuk meminimalisir terjadinya *Underfitting* atau *Overfitting* dengan mencari parameter yang tepat sehingga dapat meningkatkan hasil klasifikasi dalam membangun model.

C. Penelitian selanjutnya dapat mencoba melakukan klasifikasi dengan memperhatikan urutan dari *untaian* DNA dengan menggunakan algoritme yang berbeda seperti BiLSTM, GRU dan BiGRU, dan metode lain dengan kelengkapan fitur di dalamnya sehingga dapat diperoleh hasil klasifikasi yang lebih baik sebagai bahan perbandingan.

DAFTAR PUSTAKA

Achtman, M., Dougan, G., Kidgell, C., Reichard, U., Wain, J., Linz, B., & Mia Torpdahl. (2002). Salmonella typhi, the causative agent of typhoid fever, is approximately 50,000 years old. *Elsevier*, 39-45.

Alasadi, S., & Bhaya, W. (2017). Review of Data Preprocessing Techniques in Data Mining. *Journal of Engineering and Applied Sciences*, 4102-4107.

Algoritma. (2022, Maret 24). *Algoritma*. Retrieved February 19, 2023, from https://algorit.ma/blog/lstm-network-adalah-2022/

Andros, Prawita, D., Karsten, J., & Vinandar, M. (2015). Perbandingan Algoritma Pendeteksian Spam. *Jurnal Teknologi Terpadu*, 1-5.

Anisa, N. (2022, Februari 14). *Binus University School Of Information System*. Retrieved February 19, 2023, from https://sis.binus.ac.id/2022/02/14/mengenal-artificial-neural-network/

Annisa, R. (2017). Pendekatan Metode Feature Extraction Dengan Algoritma Naive Bayes. *Konferensi Nasional Ilmu Sosial & Teknologi (KNiST)*, 19-24.

Aryani, T. (2022, Juli 25). *Zenius*. Retrieved February 19, 2023, from https://www.zenius.net/blog/konsep-dna

Asghar, M. Z., Subhan, F., Ahmad, H., Khan, W. Z., Hakak, S., Gadekallu, T. R., & Alazab, M. (2020). Senti-eSystem: A sentiment-based eSystem-using

hybridized fuzzy and deep neural network for measuring customer satisfaction. *Software: Practice and Experience*, 1-24.

Bekkar, M., Djemaa, D. K., & Alitouche, D. A. (2013). Evaluation Measures for Models Assessment over Imbalanced Data Sets. *Journal of Information Engineering and Applications*, 27-39.

Brownlee, J. (2017, Juli 28). *Machine Learning Mastery*. Retrieved Agustus 15, 2023, from https://machinelearningmastery.com/why-one-hot-encode-data-in-machine-learning/

C´edric, S. M. (2020). Audio frame reconstruction from incomplete observations using Deep Learning techniques. *Université de Liège, Liège, Belgique*, 1-72.

Chicco, D., Tötsch, N., & Jurman, G. (2021). The Matthews correlation coefficient (MCC) is more reliable than balanced accuracy, bookmaker informedness, and markedness in two-class confusion matrix evaluation. *BioData Mining*, 1-22.

Del Rio, A. L., Martin, M., Lluna, A. P., & Saidi, R. (2020). Efect of sequence padding on the performance of deep learning models in archaeal protein functional prediction. *Scientific Reports*, 1-14.

Desai, H. P., Parameshwaran, A. P., Sunderraman, R., & Weeks, M. (2020). Comparative Study Using Neural Networks for 16S. *Journal Of Computational Biology*, 248-258.

Dongare, A., Kharde, R., & Kachare, A. (2012). Introduction to artificial neural network. *International Journal of Engineering and Innovative Technology (IJEIT)*, 189-194.

EDUTECH. (2020, January 15). *Kelas Pintar*. Retrieved February 19, 2023, from https://www.kelaspintar.id/blog/edutech/mengenal-dna-dan-rna-dalam-pewarisan-sifat-2879/

Fauzi, R. A., & Romadhony, A. (2021). Ekstraksi Aspek menggunakan BiLSTM-CRFs pada Ulasan Lipstik Bahasa Indonesia. *e-Proceeding of Engineering*, 10350.

Grefenstette, G. (1999). TOKENIZATION. *Kluwer Academic Publishers*, 117-133.

Gunasekaran, H., Ramalakshmi, K., Arokiaraj, A. M., Kanmani, S., Venkatesan, C., & Dhas, C. G. (2021). Analysis of DNA Sequence Classification Using CNN and Hybrid Models. *Hindawi*, 1-12.

Hebert, P. N., & Gregory, T. R. (2005). The Promise of DNA Barcoding for Taxonomy. *SYSTEMATIC BIOLOGY*, 852-859.

Higashihara, M., Mendez, J., Yamada, Y., & Satou, K. (2008). Application of a Feature Selection Method to Nucleosome Data: Accuracy Improvement and Comparison with Other Methods. *WSEAS TRANSACTIONS on BIOLOGY and BIOMEDICINE*, 96-105.

Kulkarni, A., Batarseh, F., & Chong, D. (2020). Foundations of data imbalance and solutions for a data democracy. In Data Democracy: At the Nexus of Artificial Intelligence, Software Development, and Knowledge Engineering. *arxiv*, 1-20.

Kurniawan, A. A., & Mustikasari, M. (2020). Implementasi Deep Learning Menggunakan Metode CNN dan LSTM untuk Menentukan Berita Palsu dalam Bahasa Indonesia. *Jurnal Informatika Universitas Pamulang* , 544-552.

Mahmoud, M., & Guo, P. (2021). DNA sequence classification based on MLP with PILAE algorithm. *Soft Computing*, 4003-4014.

Neugebauer, T., Bordeleau, E., Burrus, V., & Brzezinski, R. (2015). DNA Data Visualization (DDV): Software for Generating Web-Based Interfaces Supporting Navigation and Analysis of DNA Sequence Data of Entire Genomes. *PLoS ONE*, 1-16.

Nguyen , N. G., Tran, V. A., Ngo, D. L., Phan, D., Lumbanraja, F. R., Faisal, M. R., . . . Sato, K. (2016). DNA Sequence Classification by Convolutional Neural Network. *J. Biomedical Science and Engineering*, 280-286.

Nugroho, A. M., & Hidayatullah, A. F. (2021). Keterangan Gambar Otomatis Berbahasa Indonesia dengan CNN dan LSTM. *Jurnal UII*, 1-4.

Pokholok , D., Harbison, C., Levine , S., Cole, M., Hannett, N., Lee, T., . . . Young, R. (2005). Genome-wide Map of Nucleosome Acetylation and Methylation in Yeast. *Nucleosome Acetylation and Methylation Map*, 517-527.

Proteomics, C. (2019, Desember 24). *Creative Proteomics Blog*. Retrieved Agustus 15, 2023, from https://www.creative-proteomics.com/blog/index.php/strategies-for-analyzing-histone-modifications/

Rahayu, S., Adji, T. B., & Setiawan, N. A. (2017). Analisis Perbandingan Metode Over-Sampling Adaptive Synthetic-Nominal (ADASYN-N) dan Adaptive Synthetic-kNN (ADSYN-kNN) untuk Data dengan Fitur Nominal-Multi Categories. *CITEE*, 296-300.

Suraj. B., Sneha. R., & Sugamya. K. (2020). Realistic Handwriting Generation Using Recurrent Neural Networks and Long Short-Term Networks.

Proceedings of the Third International Conference on Computational Intelligence and Informatics, 651-661.

Smagulova, K., & James, A. P. (2019). Chapter 11 Overview of Long Short-Term Memory Neural Networks. *Modeling and Optimization in Science and Technologies*, 139-153.

Taigman , Y., Yang , M., Ranzato, M., & Wolf, L. (2014). DeepFace: Closing the Gap to Human-Level Performance in Face Verification. *IEEE Conference on Computer Vision and Pattern Recognition*, 1701-1708.

Thpanorama. (2023). *Thpanorama*. Retrieved February 19, 2023, from https://id.thpanorama.com/articles/biologa/histonas-caractersticas-estructura-tipos-y-funciones.html

Trivusi. (2022, September 17). *Trivusi Web*. Retrieved February 19, 2023, from https://www.trivusi.web.id/2022/07/algoritma-lstm.html

Ullah, K., Rashad, A., Khan, M., Ghadi, Y., Aljuaid, H., & Nawaz, Z. (2022). A Deep Neural Network-Based Approach for Sentiment Analysis of Movie Reviews. *Hindawi*, 1-9.

Valueva, M., Nagornov, N., Lyakhov, P., Valuev, G., & Chervyakov, N. (2020). Application of the residue number system to reduce hardware costs of the convolutional neural network implementation. *Mathematics And Computers In Simulation*, 232-243.

Vazhayil, A., Ravi, V., & P, S. K. (2018). DeepProteomics: Protein family classification using Shallow and Deep Networks. *Arxiv*, 1-17.

Wintoro, P. B., H. H., Muda, M. A., & Y. M. (2022). Implementasi Long Short-Term Memory pada Chatbot Informasi Akademik Teknik Informatika Unila. *Jurnal Manajemen Sistem Informasi dan Teknologi*, 68-75.

Woo, Lau, Teng, Tse, & Yuen. (2008). Then and now: use of 16S rDNA gene sequencing for bacterial identification and discovery of novel bacteria in clinical microbiology laboratories. *Journal Compilation*, 908-934.

Yustiana. F., Ridwan. I., & Fatan. K. (2022). Mesin Penterjemah Bahasa Indonesia-Bahasa Sunda Menggunakan Recurrent Neural Networks. *JURNAL TEKNOINFO*, 313-322.