KLASIFIKASI GEN ESENSIAL PADA SEKUENS PROTEIN *DROSOPHILA MELANOGASTER* MENGGUNAKAN METODE *BIDIRECTIONAL GATED RECURRENT UNIT* (BiGRU)

(Skripsi)

Oleh

Yulia Dwi Putri
2017051016

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG
2024

ABSTRAK

KLASIFIKASI GEN ESENSIAL PADA SEKUENS PROTEIN *DROSOPHILA MELANOGASTER* MENGGUNAKAN METODE *BIDIRECTIONAL GATED RECURRENT UNIT* (BiGRU)

Oleh

YULIA DWI PUTRI

Gen esensial berperan penting dalam kelangsungan hidup organisme, dan identifikasinya memiliki aplikasi luas dalam biologi sintetis dan medis. Namun, teknik eksperimental sering kali mahal dan rumit, sehingga metode komputasional berbasis pembelajaran mesin menjadi alternatif yang lebih efisien. Metode yang digunakan adalah *Bidirectional Gated Recurrent Unit* (BiGRU) dengan menggunakan *Drosophila melanogaster* sebagai organisme model. Tujuan dari penelitian ini adalah untuk mengevaluasi kinerja metode *Bidirectional Gated Recurrent Unit* (BiGRU) dalam sekuens protein pada *Drosophila melanogaster*. Dataset yang digunakan diperoleh dari penelitian Beder, et al, (2021) dengan 2 jenis dataset yaitu *Cellular Essential Gene* (CEG) dan *Organismal Essential Gene* (OEG). Ada dua skema arsitektur model dan dua skenario pembagian data, yaitu 80% *training* 20% *validation* dan 90% *training* 10% *validation*. Pada dataset CEG memiliki distribusi kelas yang tidak seimbang sehingga dilakukan proses *Random Undersampling* (RUS). Hasil kinerja yang paling baik pada dataset OEG didapatkan pada skenario pembagian data 80% *training* 20% *validation* dengan nilai yang didapat adalah 83 % sensitivitas, 79 % spesifisitas, 80% nilai ROC-AUC dan 85% nilai PR-AUC. Hasil yang paling baik pada dataset CEG diperoleh dari pembagian data 90% *training* 10% *validation* dengan nilai yang didapat untuk sensitivitas adalah 73%, spesifisitas 55%, nilai ROC-AUC 64% dan nilai PR-AUC 46%.

Kata Kunci: BiGRU, *Drosophila melanogaster*, Gen Esensial, Sekuens Protein.

ABSTRACT

CLASSIFICATION OF ESSENTIAL GENES IN DROSOPHILA MELANOGASTER PROTEIN SEQUENCES USING BIDIRECTIONAL GATED RECURRENT UNIT (BIGRU) METHOD

By

YULIA DWI PUTRI

Essential genes play a crucial role in the survival of organisms, and their identification has wide applications in synthetic biology and medicine. However, experimental techniques are often expensive and complex, making computational methods based on machine learning a more efficient alternative. The method used in this study is the Bidirectional Gated Recurrent Unit (BiGRU), with Drosophila melanogaster as the model organism. The aim of this research is to evaluate the performance of the BiGRU method in protein sequences of Drosophila melanogaster. The dataset used was obtained from the study by Beder et al. (2021) and consists of two types of datasets: Cellular Essential Gene (CEG) and Organismal Essential Gene (OEG). There are two model architecture schemes and two data-splitting scenarios: 80% training and 20% validation, and 90% training and 10% validation. The CEG dataset has an imbalanced class distribution, so Random Undersampling (RUS) was applied. The best performance for the OEG dataset was achieved with the 80% training and 20% validation split, yielding 83% sensitivity, 79% specificity, 80% ROC-AUC, and 85% PR-AUC. The best performance for the CEG dataset was obtained with the 90% training and 10% validation split, yielding 73% sensitivity, 55% specificity, 64% ROC-AUC, and 46% PR-AUC.

Keywords: *BiGRU, Drosophila melanogaster, Essential Genes, Protein Sequence.*

KLASIFIKASI GEN ESENSIAL PADA SEKUENS PROTEIN *DROSOPHILA MELANOGASTER* MENGGUNAKAN METODE *BIDIRECTIONAL GATED RECURRENT UNIT* (BiGRU)

Oleh

YULIA DWI PUTRI

Skripsi

Sebagai Salah Satu Syarat untuk Mencapai Gelar

SARJANA ILMU KOMPUTER

Pada

Jurusan Ilmu Komputer

Fakultas Matematika dan Ilmu Pengetahuan Alam

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM

UNIVERSITAS LAMPUNG

BANDAR LAMPUNG

2024

Judul Skripsi	:	**KLASIFIKASI GEN ESENSIAL PADA SEKUENS PROTEIN *DROSOPHILA MELANOGASTER* MENGGUNAKAN METODE *BIDIRECTIONAL GATED RECURRENT UNIT* (BiGRU)**
Nama Mahasiswa	:	**Yulia Dwi Putri**
Nomor Pokok Mahasiswa	:	2017051016
Program Studi	:	S1-Ilmu Komputer
Fakultas	:	Matematika dan Ilmu Pengetahuan Alam

1. Komisi Pembimbing

Favorisen R. Lumbanraja, Ph.D.
NIP. 198301102008121002

2. Ketua Jurusan Ilmu Komputer

Dwi Sakethi, S.Si., M.Kom.
NIP. 196806111998021001

MENGESAHKAN

1. Tim Penguji

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

   Penguji I

   Penguji Pembahas : **Fatma Indriani, S.T., MIT, Ph.D.**

   Penguji II

   Penguji Pembahas : **Dr. rer. nat. Akmal Junaidi, M.Sc.**

2. Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam

Dr. Eng. Heri Satria, S.Si., M.Si.
NIP. 197110012005011002

Tanggal Lulus Ujian Skripsi: **2 Oktober 2024**

PERNYATAAN

Saya yang bertanda tangan di bawah ini:

Nama : Yulia Dwi Putri
NPM : 2017051016

Menyatakan bahwa skripsi saya yang berjudul "**Klasifikasi Gen Esensial Pada Sekuens Protein *Drosophila Melanogaster* Menggunakan Metode *Bidirectional Gated Recurrent Unit* (BiGRU)**" merupakan karya saya sendiri dan bukan karya orang lain. Seluruh tulisan yang tertulis dalam skripsi ini telah mengikuti kaidah penulisan karya ilmiah Universitas lampung. Apabila di kemudian hari terbukti skripsi saya merupakan hasil penjiplakan atau dibuat orang lain, maka saya bersedia menerima sanksi berupa pencabutan gelar yang telah saya terima.

Bandar Lampung, 2 Oktober 2024

Yulia Dwi Putri
NPM. 2017051016

RIWAYAT HIDUP

Penulis bernama Yulia Dwi Putri bertempat lahir di Gedong Tataan pada tanggal 10 Juli 2002, sebagai anak kedua dari dua bersaudara dari pasangan Bapak S. Edi Sasmita dan Ibu Ida Royani. Penulis menyelesaikan pendidikan formal di SD Negeri 1 Sukaraja dan selesai pada tahun 2014. Kemudian melanjutkan pendidikan menengah pertama di SMP Negeri 1 Gedong Tataan yang diselesaikan pada tahun 2017, lalu melanjutkan ke pendidikan menengah atas di SMA Negeri 1 Gadingrejo yang diselesaikan pada tahun 2020.

Pada tahun 2020 penulis terdaftar sebagai mahasiswa Jurusan Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung melalui jalur SNMPTN. Selama menjadi mahasiswa, penulis melakukan beberapa kegiatan antara lain.

1. Menjadi anggota Adapter Himpunan Mahasiswa Jurusan Ilmu Komputer pada periode 2020/2021.
2. Menjadi anggota Biro Kesekretariatan Himpunan Mahasiswa Jurusan Ilmu Komputer pada priode 2020/2021.
3. Menjadi bendahara Biro Kesekretariatan Himpunan Mahasiswa Jurusan Ilmu Komputer pada periode 2021/2022.
4. Menjadi Asisten Dosen Jurusan Ilmu Komputer pada mata kuliah Sistem Operasi dan mata kuliah Basis Data tahun 2022, mata kuliah Pemrosesan Data Terdistribusi tahun 2023 serta mata kuliah Bioinformatika tahun 2024.

5. Mengikuti *Course* UI/UX *Designer* Pemula pada Program Kredensial Mikro Mahasiswa Indonesia (KMMI) pada tahun 2021.
6. Melaksanakan Kerja Praktik di PT Jasa Raharja Putera Cabang Bandar Lampung pada periode I tahun 2023.
7. Melaksanakan KKN di Desa Tanjung Agung, Kecamatan Teluk Pandan, Kabupaten Pesawaran pada periode II tahun 2023.

MOTTO

"Dan orang-orang yang bersungguh-sungguh untuk (mencari keridhaan) Kami, benar-benar akan Kami tunjukkan kepada mereka jalan-jalan Kami. Dan sesungguhnya Allah benar-benar beserta orang-orang yang berbuat baik."
(QS. Al-Ankabut: 69)

"Sesungguhnya setiap amalan tergantung pada niatnya. Setiap orang akan mendapatkan apa yang ia niatkan."
(Hadits Riwayat Muslim No. 1907)

PERSEMBAHAN

Alhamdulillahirobbilalamin

Puji dan syukur tercurahkan kepada Allah Subhanahu Wa Ta'ala atas segala Rahmat dan Karunia-Nya sehingga saya dapat menyelesaikan skripsi ini. Shalawat serta salam selalu tercurahkan kepada Nabi Muhammad Shallallahu Alaihi Wasallam.

Kupersembahkan karya ini kepada:

Kedua Orang Tuaku Tercinta

Atas segala pengorbanan, perjuangan, kasih sayang, perhatian, dukungan dan do'a yang selalu menyertaiku. Kuucapkan terima kasih sebesar-besarnya karena telah mendidik dan membesarkanku dengan penuh kasih sayang yang tak akan terbalaskan.

Seluruh Keluarga Besar Ilmu Komputer 2020

Yang senantiasa memberikan semangat dan dukungan.

Almamater Tercinta, Universitas Lampung dan Jurusan Ilmu Komputer

Tempat bernaung mengemban semua ilmu untuk menjadi bekal kehidupan.

SANWACANA

Puji syukur kehadirat Allah *Subhanahu Wa Ta'ala*, karena telah memberikan limpahan nikmat, rahmat dan karunia-Nya. Shalawat serta salam semoga senantiasa tercurahkan kepada junjungan Nabi Muhammad SAW, sehingga penulis dapat menyelesaikan skripsi yang berjudul "**Klasifikasi Gen Esensial Pada Sekuens Protein *Drosophila Melanogaster* Menggunakan Metode *Bidirectional Gated Recurrent Unit* (BiGRU)**" dengan baik dan lancar.

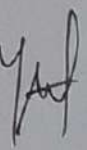
Selesainya skripsi ini tidak terlepas dari bantuan, bimbingan, dan dukungan dari berbagai pihak. Oleh karena itu, ucapan terima kasih ditujukan kepada:

1. Kedua orangtua, Almh. Nenek Budi, kakak serta adik Azlan yang selalu mendoakan yang terbaik, memberi dukungan, kasih sayang dan selalu memberikan semangat baik secara moral maupun material dalam menyelesaikan skripsi ini.
2. Bapak Favorisen R. Lumbanraja, Ph.D. selaku pembimbing akademik serta pembimbing utama dalam penelitian ini yang selalu membimbing, memberikan arahan, masukan dan saran dalam penyelesaian skripsi.
3. Ibu Fatma Indriani, S.T., MIT, Ph.D. selaku pembahas pertama yang telah memberikan masukan serta saran yang bermanfaat dalam perbaikan skripsi ini.
4. Bapak Dr. rer. nat. Akmal Junaidi, M.Sc. sebagai pembahas kedua yang telah memberikan masukan serta saran yang bermanfaat dalam perbaikan skripsi ini.
5. Bapak Dr. Eng. Heri Satria, S.Si., M.Si. selaku dekan FMIPA Universitas Lampung.

6. Bapak Dwi Sakethi, S.Si., M.Kom. selaku Ketua Jurusan Ilmu Komputer FMIPA Universitas Lampung.
7. Ibu Anie Rose Irawati, S.T. M.Cs. selaku Sekretaris Jurusan Ilmu Komputer FMIPA Universitas Lampung.
8. Seluruh Dosen, Staf, dan Karyawan Jurusan Ilmu Komputer yang telah memberikan ilmu, pelajaran, dan bantuan terbaik selama penulis menempuh pendidikan di Jurusan Ilmu Komputer Universitas Lampung.
9. Teman seperjuangan semasa kuliah Aura Husnaini P.Z, Melan Caniadi, Ages Mahesa, Dita Faradila, dan Pynka Aryani Angelia Haryanto yang selalu mendukung, menemani, dan berbagi cerita indah selama masa perkuliahan.
10. Putri Santika Mayangsari, Azahra Alya Hidayah, Mba Mila, Nafasya Rahma Safitra, dan Safiira Rahmah Linisa yang selalu membantu dan memberikan semangat kepada penulis untuk menyelesaikan penelitian ini.
11. Teman-teman Himakom yang sudah mengajarkan banyak hal dalam berorganisasi dan memberikan pengalaman yang berharga.
12. Keluarga Ilmu Komputer 2020 yang telah memberikan pengalaman yang sangat berarti selama menjalankan studi di Jurusan Ilmu Komputer Universitas Lampung.
13. Seluruh pihak yang terlibat secara langsung maupun tidak langsung, atas dukungannya dalam menyelesaikan skripsi.

Penulis menyadari bahwa penyusunan skripsi ini masih jauh dari kata sempurna. Namun penulis sangat mengharapkan skripsi ini dapat bermanfaat bagi para civitas akademik Universitas Lampung pada umumnya dan mahasiswa Ilmu Komputer pada khususnya.

Bandar Lampung, 2 Oktober 2024



Yulia Dwi Putri
NPM. 2017051016

DAFTAR ISI

Halaman

DAFTAR TABEL ..... **iv**
DAFTAR GAMBAR ..... **vi**
DAFTAR KODE PROGRAM ..... **viii**
I. PENDAHULUAN **1**
1.1 Latar Belakang 1
1.2 Rumusan Masalah 3
1.3 Batasan Masalah 3
1.4 Tujuan Penelitian 3
1.5 Manfaat Penelitian 4
II. TINJAUAN PUSTAKA **5**
2.1 Penelitian Terdahulu 5
2.2 Gen Esensial 10
2.3 CEG 10
2.4 OEG 11
2.5 Protein 11
2.6 CRISPR 12
2.7 RNAi 13
2.8 *Drosophila melanogaster* 14
2.9 Tokenisasi 15
2.10 *Padding* 16
2.11 *Random Undersampling* 17
2.12 *Embedding Layer* 17
2.13 BiGRU 17
2.14 *Flatten* 25
2.15 *Dense Layer* 26
2.16 *Dropout* 26
2.17 *Confusion Matrix* 27
2.18 PR AUC 27

2.19 ROC AUC 28
2.20 *Underfitting* 29
2.21 *Overfitting* 29

III. METODE PENELITIAN 30
3.1 Tempat dan Waktu Penelitian 30
3.1.1 Tempat Penelitian 30
3.1.2 Waktu Penelitian 30
3.2 Data dan Alat 32
3.2.1 Data 32
3.2.2 Alat 33
3.3 Metodologi 36

IV. HASIL DAN PEMBAHASAN 39
4.1 *Preprocessing* Data 39
4.1.1 *Cleaning* Data 39
4.1.2 Penggabungan Data Sekuens dan Label 40
4.1.3 Tokenisasi 42
4.1.4 Padding 43
4.2 Pembagian Data 45
4.3 *Random Undersampling* 47
4.4 Klasifikasi *Bidirectional Gated Recurrent Unit* 49
4.5 Pelatihan Data 56
4.5.1 Data OEG 56
4.5.2 Data CEG 68
4.6 Pengujian Hasil Klasifikasi 81
4.6.2 Data OEG 84
4.6.2 Data CEG 93
4.7 Pembahasan 102
4.6.2 Data OEG 103
4.6.2 Data CEG 106
4.8 Perbandingan dengan Penelitian Terdahulu 110

V. SIMPULAN DAN SARAN 114
5.1 Simpulan 114
5.2. Saran 115

DAFTAR PUSTAKA 116

DAFTAR TABEL

Tabel Halaman

1. Penelitian terdahulu terkait dengan gen esensial 5
2. Asam Amino 12
3. *Character-level Tokenization*.....16
4. *Padding* Sekuens.....16
5. Alur Pengerjaan Penelitian.....31
6. Jumlah *Dataset*.....33
7. Data Label.....39
8. Perubahan Jumlah Data Tahap *Cleaning* 40
9. Penggabungan Sekuens dan Label.....42
10. Pembagian Data OEG 46
11. Pembagian Data CEG 47
12. *Random Undersampling* CEG.....48
13. Perbedaan Arsitektur I dan II.....54
14. Hasil Pelatihan 80% *Train* 20% Validasi Arsitektur I OEG.....59
15. Hasil Pelatihan 80% *Train* 20% Validasi Arsitektur II OEG 62
16. Hasil Pelatihan 90% *Train* 10% Validasi Arsitektur I OEG.....65
17. Hasil Pelatihan 90% *Train* 10% Validasi Arsitektur II OEG 68
18. Hasil Pelatihan 80% *Training* 20% Validasi Arsitektur I CEG.....72
19. Hasil Pelatihan 80% *Training* 20% Validasi Arsitektur II CEG.....75
20. Hasil Pelatihan 90% *Training* 10% Validasi Arsitektur I CEG.....78
21. Hasil Pelatihan 90% *Training* 10% Validasi Arsitektur II CEG.....81
22. Hasil Pengujian OEG Arsitektur I Pembagian Data 80% 20%.....86
23. Hasil Pengujian OEG Arsitektur II Pembagian Data 80% 20% 88
24. Hasil Pengujian OEG Arsitektur I Pembagian Data 90% 10%.....91
25. Hasil Pengujian OEG Arsitektur II Pembagian Data 90% 10% 92
26. Hasil Pengujian CEG Arsitektur I Pembagian Data 80% 20%.....94

27. Hasil Pengujian CEG Arsitektur II Pembagian Data 80% 20% 97
28. Hasil Pengujian CEG Arsitektur I Pembagian Data 90% 10%.....99
29. Hasil Pengujian CEG Arsitektur II Pembagian Data 90% 10% 102
30. Hasil Pengujian OEG 80% *Training* 20% Validasi 103
31. Hasil Pengujian OEG Pembagian 90% *Training* 10% Validasi 104
32. Hasil Pengujian CEG 80% *Training* 20% Validasi 107
33. Hasil Pengujian CEG 90% *Training* 10% Validasi 108
34. Perbandingan dengan Penelitian Terdahulu.....111

DAFTAR GAMBAR

Gambar Halaman

1. *Drosophila melanogaster* (Edelsparre et al., 2021). 14
2. Struktur GRU (Yang et al., 2022). 18
3. Struktur BiGRU (Mekruksavanich & Jitpattanakul, 2023).....20
4. *Dropout* (Xie, 2020).....26
5. *Confusion Matrix*. 27
6. Perbandingan *Dataset* CEG dan OEG. 33
7. Alur Kerja Penelitian.....36
8. Data Sekuens Protein. 40
9. Arsitektur I. 50
10. Arsitektur II.....53
11. Grafik OEG Arsitektur I 80% *Train* 20% Validasi *Pre Paddin*g. 57
12. Grafik OEG Arsitektur I 80% *Train* 20% Validasi *Post Padding*.....58
13. Grafik OEG Arsitektur II 80% *Train* 20% Validasi *Pre Padding*.....60
14. Grafik OEG Arsitektur II 80% *Train* 20% Validasi *Post Padding*.....61
15. Grafik OEG Arsitektur I 90% *Train* 10% Validasi *Pre Padding*. 63
16. Grafik OEG Arsitektur I 90% *Train* 10% Validasi *Post Padding*.....64
17. Grafik OEG Arsitektur II 90% *Train* 10% Validasi *Pre Padding*.....66
18. Grafik OEG Arsitektur II 90% *Train* 10% Validasi *Post Padding*.....67
19. Grafik CEG Arsitektur I 80% *Train* 20% Validasi *Pre Padding*.....69
20. Grafik CEG Arsitektur I 80% *Train* 20% Validasi *Post Padding*. 71
21. Grafik CEG Arsitektur II 80% *Train* 20% Validasi *Pre Padding*. 73
22. Grafik CEG Arsitektur II 80% *Train* 20% Validasi *Post Padding*.....74
23. Grafik CEG Arsitektur I 90% *Train* 10% Validasi *Pre Padding*.....76
24. Grafik CEG Arsitektur I 90% *Train* 10% Validasi *Post Padding*. 77
25. Grafik CEG Arsitektur II 90% *Train* 10% Validasi *Pre Padding*. 79
26. Grafik CEG Arsitektur II 90% *Train* 10% Validasi *Post Padding*.....80

27. *Confusion Matrix* OEG 80% 20% Arsitektur I *Pre Padding*.....84
28. *Confusion Matrix* OEG 80% 20% Arsitektur I *Post Padding*. 85
29. *Confusion Matrix* OEG 80% 20% Arsitektur II *Pre Padding*. 87
30. *Confusion Matrix* OEG 80% 20% Arsitektur II *Post Padding*. 88
31. *Confusion Matrix* OEG 90% 10% Arsitektur I *Pre Padding*.....89
32. *Confusion Matrix* OEG 90% 10% Arsitektur I *Post Padding*. 90
33. *Confusion Matrix* OEG 90% 10% Arsitektur II *Pre Padding*. 91
34. *Confusion Matrix* OEG 90% 10% Arsitektur II *Post Padding*. 92
35. *Confusion Matrix* CEG 80% 20% Arsitektur I *Pre Padding*. 93
36. *Confusion Matrix* CEG 80% 20% Arsitektur I *Post Padding*.....94
37. *Confusion Matrix* CEG 80% 20% Arsitektur II *Pre Padding*.....95
38. *Confusion Matrix* CEG 80% 20% Arsitektur II *Post Padding*. 96
39. *Confusion Matrix* CEG 90% 10% Arsitektur I *Pre Padding*. 97
40. *Confusion Matrix* CEG 90% 10% Arsitektur I *Post Padding*.....98
41. *Confusion Matrix* CEG 90% 10% Arsitektur II *Pre Padding*.....100
42. *Confusion Matrix* CEG 90% 10% Arsitektur II *Post Padding*. 101
43. Grafik Hasil Pengujian OEG 80% *Training* 20% Validasi.....104
44. Grafik Hasil Pengujian OEG 90% *Training* 10% Validasi.....105
45. Hasil Pengujian CEG 80% *Training* 20% Validasi. 107
46. Hasil Pengujian CEG 90% *Training* 10% Validasi 108
47. Perbandingan dengan Penelitian Terdahulu. 111

DAFTAR KODE PROGRAM

Kode Program Halaman

1. *Cleaning* Data.....40
2. Penggabungan Sekuens dan Label.....41
3. Tokenisasi.....43
4. Tokenisasi pada Dataset.....43
5. *Padding Dataset* OEG.....44
6. *Padding Dataset* CEG.....44
7. Pembagian Data *Training* dan *Testing* OEG.....45
8. Pembagian Data 80% *Training* 20% Validasi Data OEG.....46
9. Pembagian Data 90% *Training* 10% Validasi Data OEG.....46
10. Pembagian 80% *Training* 20% *Testing* Data CEG.....46
11. Pembagian Data 80% *Training* 20% Validasi Data CEG.....47
12. Pembagian Data 90% *Training* 10% Validasi Data CEG.....47
13. *Random Undersampling* CEG.....48
14. Implementasi Arsitektur I.....51
15. Implementasi Arsitektur II.....53
16. Kompilasi Model.....55
17. Melatih Model.....55
18. *Confusion Matrix*.....81
19. Sensitivitas dan Spesifisitas.....82
20. PR AUC.....83
21. ROC AUC.....83

I. PENDAHULUAN

1.1 Latar Belakang

Gen esensial merupakan gen yang berperan penting untuk kelangsungan hidup organisme. Gen esensial berkaitan dengan aktifitas seluler kritis pada organisme. Gen tersebut berperan dalam mengatur metabolisme pusat, translasi gen, replikasi asam deoksiribonukleat (DNA), struktur seluler dasar serta memfasilitasi transportasi intraseluler dan ekstraseluler. Penghapusan gen esensial dapat mengakibatkan kematian (Rout et al., 2023).

Identifikasi gen esensial memiliki kepentingan teoretis yang substansial dalam biologi sintetis dan memiliki aplikasi praktis dalam biomedis (Yu et al., 2017). Informasi tentang esensialitas gen digunakan dalam berbagai penelitian ilmiah, terutama untuk mengidentifikasi target obat, seperti dalam terapi kanker atau mengidentifikasi target insektisida, serta untuk merancang genom minimal dalam biologi sintetis (Beder et al., 2021).

Identifikasi gen esensial dapat dilakukan melalui pendekatan eksperimental. *Single gene deletion*, *antisense* RNA, *transposon mutagenesis* dan *Clustered Regularly Interspaced Short Palindromic Repeats* (CRISPR) merupakan teknik eksperimental yang sering digunakan untuk mengidentifikasi gen esensial. Namun, teknik tersebut bersifat kompleks, membutuhkan biaya yang mahal dan waktu yang lama (Aromolaran, et al., 2021). Oleh karena itu, pendekatan komputasional dengan metode pembelajaran mesin dikembangkan sebagai alternatif untuk meminimalkan sumber daya yang dibutuhkan dalam identifikasi esensialitas gen. Pada pembelajaran mesin menggunakan data untuk melatih model.

Protein merupakan makromolekul yang terbentuk melalui rangkaian asam amino yang berbeda, yang saling terhubung melalui ikatan peptida dan ikatan kovalen. Struktur utama protein terdiri dari asam amino (Sutanto et al., 2020). DNA berfungsi sebagai pembawa informasi genetik dan mengalami transkripsi menjadi RNA, yang selanjutnya diterjemahkan menjadi protein. Protein merupakan manifestasi dari informasi genetik, bertanggung jawab atas berbagai fungsi biologis, dan mendukung kegiatan metabolisme dalam organisme (Jiang et al., 2013).

Drosophila melanogaster atau yang biasa disebut lalat buah telah digunakan sebagai organisme model dalam penelitian ilmiah dan medis. Ciri fisik lalat buah juga menjadikannya model yang berharga, karena organ dan jaringan lalat dewasa memiliki fungsi yang setara dengan organ dan jaringan manusia (jantung, paru-paru, ginjal, usus, dan saluran reproduksi (Pandey dan Nichols, 2011). Lalat buah dapat dipelihara secara efisien dengan biaya yang rendah di laboratorium dan diakui sebagai model alternatif dibandingkan dengan vertebrata lainnya (Adesola et al., 2021). Sebagian besar gen yang terkait dengan penyakit pada manusia memiliki gen kesamaan fungsional pada *Drosophila melanogaster* yang mencapai sekitar 75% (Bier, 2005).

Pada penelitian ini, gen esensial diklasifikasikan berdasarkan sekuens protein pada *Drosophila melanogaster* menggunakan metode *Bidirectional Gated Recurrent Unit* (BiGRU). *Bidirectional Gated Recurrent Unit* merupakan arsitektur yang menggunakan lapisan rekuren tipe GRU untuk memproses informasi dalam dua arah (Alsanousi et al., 2022). Metode ini digunakan untuk mengolah data sekuensial termasuk sekuens protein**.**

1.2 Rumusan Masalah

Berdasarkan penjelasan latar belakang, adapun masalah dalam penelitian ini adalah sebagai berikut :

1. Apakah metode *Bidirectional Gated Recurrent Unit*(BiGRU) dapat diimplementasikan untuk klasifikasi gen esensial pada sekuens protein *Drosophila melanogaster*?
2. Apakah parameter dan skenario yang terbaik untuk metode *Bidirectional Gated Recurrent Unit* dalam klasifikasi gen esensial pada sekuens protein *Drosophila melanogaster*?
3. Seberapa baik perbandingan hasil evaluasi kinerja antara penelitian ini dengan penelitian terdahulu?

1.3 Batasan Masalah

Batasan masalah dalam penelitian ini adalah :

1. Penelitian ini menggunakan data *Cellular Essential Gene* (CEG) dan *Organismal Essential Gene* (OEG) dari *Drosophila melanogaster* yang diperoleh dari penelitian Beder, et al (2021).
2. Klasifikasi gen esensial berdasarkan pada sekuens protein *Drosophila melanogaster* dan metode *Bidirectional Gated Recurrent Unit* (BiGRU).
3. Hasil klasifikasi hanya terdiri dari dua kelas yaitu esensial dan non-esensial.

1.4 Tujuan Penelitian

1. Mengevaluasi kinerja metode *Bidirectional Gated Recurrent Unit* (BiGRU) dalam mengklasifikasi sekuens protein *Drosophila melanogaster*.
2. Membandingkan hasil evaluasi yang diperoleh dengan penelitian sebelumnya yang menggunakan *dataset* sama.

1.5 Manfaat Penelitian

1. Hasil penelitian dapat dijadikan model untuk klasifikasi gen esensial berdasarkan sekuens protein.
2. Menjadi informasi untuk penelitian selanjutnya dalam masalah klasifikasi gen esensial berdasarkan sekuens protein.

II. TINJAUAN PUSTAKA

2.1 Penelitian Terdahulu

Penelitian terdahulu yang digunakan berkaitan dengan gen esensial. Penelitian tersebut dijadikan sebagai acuan dalam melakukan perbandingan hasil klasifikasi. Tabel 1 menunjukkan gambaran umum dari penelitian terdahulu yang menjadi acuan dalam penelitian ini.

Tabel 1. Penelitian terdahulu terkait dengan gen esensial

No	Penelitian	Data	Metode	Hasil
1	*Identifying essential genes across eukaryotes by machine learning* (Beder, et al., 2021)	OGEE & DEG *Drosophila melanogaster* CEG :11547 *Essential gene* : 1227 *Non Essential gene* : 10320 OEG : 517 *Essential gene* : 246 *Non Essential gene* : 271	*Random Forest* (RF), *Extreme Gradient Boosting* (XGB), *Oversampling* SMOTE	CEG RF : -ROC AUC=84% -PR AUC = 41% -*Sensitivity*=54% -*Specificity*= 88% XGB: -ROC AUC=83% -PR AUC = 40% -*Sensitivity* =55% -*Specificity*=86% OEG RF : -ROC AUC=92%

				-PR AUC = 92% - *Sensitivity*=82% -*Specificity*=82%
				XGB: -ROC AUC=91% -PR AUC =90% -*Sensitivity*=81% -*Specificity*=85%
2	*Essential gene prediction in Drosophila melanogaster using machine learning approaches based on sequence and functional features.* (Aromolaran et al., 2020)	OGEE & DEG *Essential gene* : 441 gen *Non Essential gene* :11.788	*Generalised Linear Model* (GLM), *Support-Vector Machines* (SVM), *Random Forest* (RF), *Artificial Neural Networks* (NNET) dan *Extreme Gradient Boosting* (XGB)	GLM -ROC AUC=89% -PR AUC = 27% -*F1-score* = 28%
				SVM -ROC AUC=88% -PR AUC = 27% -*F1-score* = 30%
				NNET -ROC AUC=85% -PR AUC = 20% -*F1-score* = 24%
				RF -ROC AUC=90% -PR AUC = 29% -*F1-score* = 32%
				XGB -ROC AUC=90% -PR AUC = 30%

				-F1-score = 34%
3	*Performance evaluation of features for gene essentiality prediction* (Aromolaran, Oyelade, et al., 2021)	OGEE & DEG *Saccharomyces. cerevisiae* *Essential gene* : 1037 *Non Essential gene*: 4543 *Schizosaccharomyces pombe* *Essential gene* : 1346 *Non Essential gene*: 3689	*Random Forest* (RF), *Artificial Neural Networks* (NNET), *Extreme Gradient Boosting* (XGB)	Sekuens Protein *S. cerevisiae* RF AUROC = 69,6% AUPRC = 32,1% *Precision* =38,5% *Sensitivity* = 4,1% NNET AUROC = 71,2% AUPRC = 39% *Precision* =40,3% *Sensitivity*=42,2% XGB AUROC = 70,3% AUPRC = 34,3% *Precision* =47,1% *Sensitivity*=12,6% Sekuens Protein *S. pombe.* RF AUROC = 64,7% AUPRC = 37,6% *Precision* =46,7% *Sensitivity* = 6,3% NNET AUROC = 63,2% AUPRC = 39,5%

	Precision=37,9% *Sensitivity*=50,6%
	XGB AUROC = 63,8% AUPRC = 37,9% *Precision*=42,7% *Sensitivity*=20,7%

1. *Identifying essential genes across eukaryotes by machine learning*

Penelitian ini melakukan identifikasi gen esensial pada eukariota dan dilakukan oleh Beder, et al. (2021). *Dataset* diambil dari enam organisme eukariotik diantaranya *Caenorhabditis elegans, D.melanogaster, H. sapiens, M. musculus, S. cerevisiae* dan *S. pombe* yang didapat dari *Online* GEne *Essentiality* (OGEE) dan *Database of Essential Genes* (DEG). Pada penelitian ini, menggunakan 2 *dataset* yaitu CEG (*Celuller Essential Gene*) dan OEG (*Organismal Essential Gene*) untuk *D. melanogaster, M. musculus,* dan *H. sapiens.* Selain itu, hanya data OEG untuk *C. elegans* serta hanya menggunakan data CEG pada *S. cerevisiae* dan *S. pombe*. Pada *D.melanogaster* menggunakan 2 *dataset* yaitu CEG dan OEG.

Total data gen esensial yang didapat sebanyak 11.038 dan 67.035 untuk non-esensial dengan 7 fitur kategori yaitu protein dan *gene sequence, functional domains, topological features, evolution/conservation, subcellular localization,* dan *gene sets* dari *Gene Ontology*.

Klasifikasi gen esensial dilakukan dengan metode *machine learning* yaitu *Random Forest* (RF) dan *Extreme Gradient Boosting* (XGB), dan pada penelitian ini juga dilakukan proses *resampling* yaitu dengan teknik SMOTE. Data dianalisis menggunakan bahasa pemrograman R. Pada CEG, *Random*

Forest menghasilkan ROC AUC sebesar 84%, nilai PR AUC sebesar 40%, *sensitivity* sebesar 54% dan *specificity* sebesar 88%. Sedangkan XGB memperoleh nilai ROC AUC sebesar 83%, PR AUC sebesar 40%, *sensitivity* sebesar 55% dan *specificity* sebesar 86%. Pada OEG, *Random Forest* menghasilkan ROC AUC sebesar 92%, nilai PR AUC sebesar 92%, *sensitivity* sebesar 82% dan *specificity* sebesar 82%, sedangkan XGB memperoleh nilai ROC AUC sebesar 91%, PR AUC sebesar 90%, *sensitivity* sebesar 81% dan *specificity* sebesar 85%.

2. ***Essential gene prediction in Drosophila melanogaster using machine learning approaches based on sequence and functional features***

Penelitian ini dilakukan oleh Aromolaran, et al. (2020) dengan menggunakan kombinasi *dataset* OGEE dan OGE yang terdiri dari 441 gen esensial dan 11.788 gen non-esensial untuk *Drosophila melanogaster*. Data ini mencakup berbagai fitur, seperti urutan protein, urutan gen, domain, topologi (profil transkripsi), topologi (PPI), evolusi, lokalitas, dan himpunan gen.

Dalam penelitian ini, metode yang digunakan mencakup *Generalised Linear Model* (GLM), *Support-Vector Machines* (SVM), *Random Forests* (RF), *Artificial Neural Networks* (NNET), dan *Extreme Gradient Boosting* (XGB). Dari semua metode yang digunakan, *Extreme Gradient Boosting* (XGB) memiliki hasil terbaik dibanding metode lainnya, dengan ROC AUC sebesar 90%, PR AUC sebesar 30%, dan nilai *F1-score* sebesar 34%.

3. ***Performance evaluation of features for gene essentiality prediction***

Penelitian ini dilakukan oleh Oyelade, et al. (2021) untuk mengevaluasi kinerja fitur untuk prediksi gen esensial. Pada penelitian ini menggunakan 2 organisme yaitu *S. cerevisiae* dan *S. pombe* yang diperoleh dari OGEE dan DEG. Prediksi dilakukan dengan mengggunakan sekuens DNA, sekuens protein, ontology dan topologi. Pada *S. cerevisiae* diperoleh data gen esensial sebanyak 1037 dan

non-esensial sebanyak 4543. Data untuk *S. pombe* diperoleh gen esensial sebanyak 1346 dan non-esensial sebanyak 3689.

Metode yang digunakan pada penelitian ini adalah *Random Forest, Artificial Neural Networks* dan *Extreme Gradient Boosting.* Untuk sekuens protein *S. cerevisiae*, metode NNET menghasilkan AUROC sebesar 71,2%, AUPRC sebesar 39%, *precision* sebesar 40,3% serta *sensitivity* sebesar 42,2%. Sedangkan untuk sekuens protein *S. pombe*, metode NNET menghasilkan AUROC sebesar 63,2%, AUPRC sebesar 39,5%, *precision* sebesar 37,9%, dan s*ensitivity* sebesar 50,6%.

2.2 Gen Esensial

Gen sebagai komponen dasar genetik, memainkan peran kunci dalam menentukan fenotipe ketika berinteraksi dengan lingkungan (Jiang et al., 2013). Gen esensial merupakan gen yang diperlukan untuk kelangsungan hidup suatu organisme dan dianggap sebagai dasar kehidupan.

Gen-gen ini memiliki peran dalam mengendalikan metabolisme pusat, translasi gen, replikasi DNA, dan struktur seluler dasar, serta mempermudah transportasi dalam dan luar sel. Gen-gen esensial menjaga informasi genomika yang sangat penting dan mungkin memiliki peran kunci dalam pemahaman mendalam tentang kehidupan dan evolusi. Penelitian mengenai gen-gen esensial telah lama dianggap sebagai topik krusial dalam bidang biologi komputasional karena signifikansinya(Rout et al., 2023).

2.3 CEG

CEG (*Cellular Essential Genes*) adalah gen-gen yang penting untuk kelangsungan hidup dan fungsi dasar sel dalam organisme. Gen ini diidentifikasi melalui eksperimen yang dilakukan pada sistem seluler. CEG berperan dalam proses-proses penting seperti biogenesis makromolekul seluler dan siklus atau proliferasi sel. Sebagai gen yang esensial bagi kelangsungan

hidup sel, diharapkan bahwa gen-gen ini juga penting pada tingkat organisme secara keseluruhan (Beder et al., 2021).

2.4 OEG

OEG (*Organismal Essential Genes*), di sisi lain, adalah gen-gen yang esensial bagi kelangsungan hidup dan perkembangan organisme multiseluler secara keseluruhan. Gen ini sering kali terlibat dalam proses yang lebih kompleks, seperti perkembangan embrio, morfogenesis saraf, dan regulasi pada tingkat organisme. Eksperimen OEG dilakukan pada organisme hidup untuk menentukan gen-gen yang diperlukan untuk fungsi dasar organisme tersebut (Beder et al., 2021).

2.5 Protein

DNA, RNA, protein, dan komponen biologis lainnya membentuk dasar kehidupan. DNA berfungsi sebagai pembawa informasi genetik dan mengalami transkripsi menjadi RNA, yang selanjutnya diterjemahkan menjadi protein. Protein merupakan manifestasi dari informasi genetik, bertanggung jawab atas berbagai fungsi biologis, dan mendukung kegiatan metabolisme dalam organisme (Jiang et al., 2013).

Protein merupakan makromolekul yang terbentuk melalui rangkaian asam amino yang berbeda, yang saling terhubung melalui ikatan peptida dan ikatan kovalen. Struktur utama protein terdiri dari asam amino. Terdapat 20 jenis asam amino yang membentuk protein (Sutanto et al., 2020). Tabel 2 memberikan informasi mengenai nama, singkatan, dan simbol asam amino tersebut.

Tabel 2. Asam Amino

Nama	Singkatan	Simbol	Nama	Singkatan	Simbol
Alanine	Ala	A	*Methionine*	Met	M
Cysteine	Cys	C	*Asparagine*	Asn	N
Aspartic Acid	Asp	D	*Proline*	Pro	P
Glutamic Acid	Glu	E	*Glutamine*	Gln	Q
Phenylalanine	Phe	F	*Arginine*	Arg	R
Glycine	Gly	G	*Serine*	Ser	S
Histidine	His	H	*Threonine*	Thr	T
Isoleucine	Ile	I	*Valine*	Val	V
Lysine	Lys	K	*Tryptophan*	Trp	W
Leucine	Leu	L	*Tyrosine*	Tyr	Y

2.6 CRISPR

CRISPR (*Clustered Regularly Interspaced Short Palindromic Repeats*) adalah teknologi yang memanfaatkan protein Cas9 untuk memotong DNA pada lokasi tertentu yang ditargetkan oleh *single guide* RNA (sgRNA). CRISPR digunakan untuk *skrining* fungsional gen-gen pada skala genom. Melalui teknologi ini, para peneliti dapat melakukan *knock-out* pada gen-gen tertentu, yang memungkinkan mereka untuk mengeksplorasi fungsi gen dengan melihat efek dari hilangnya gen tersebut.

Untuk mengidentifikasi gen-gen yang penting bagi viabilitas sel (essensialitas), CRISPR digunakan untuk membuat pustaka sgRNA yang mencakup seluruh genom. Gen-gen yang penting bagi kelangsungan hidup sel akan menunjukkan pengurangan signifikan dalam jumlah sgRNA yang teramplifikasi, karena hilangnya gen tersebut akan menyebabkan kematian sel. Sebaliknya, gen yang berperan dalam membatasi pertumbuhan sel akan menunjukkan peningkatan sgRNA, karena sel akan bertahan lebih baik tanpa gen tersebut. Dengan cara ini, gen-gen esensial dapat diidentifikasi (Chang et al., 2020).

Penggunaan CRISPR untuk *skrining* gen skala besar dapat memakan biaya yang cukup besar karena perlu mempersiapkan perpustakaan sgRNA yang mencakup seluruh genom, proses pemrosesan data yang melibatkan analisis sekuens lanjutan, dan peralatan bioteknologi yang canggih. Proses ini biasanya memerlukan waktu berminggu-minggu hingga berbulan-bulan, tergantung pada kompleksitas eksperimen dan jumlah sampel yang diproses .

2.7 RNAi

RNA *interference* (RNAi) adalah proses biologis di mana RNA kecil, seperti siRNA atau miRNA, menghambat ekspresi gen dengan mengikat mRNA yang sesuai, mengarahkannya untuk degradasi atau menghambat translasi. Pada dasarnya, RNAi menurunkan atau menghentikan produksi protein dari gen tertentu, sehingga menjadi alat yang berguna dalam studi genetik untuk memahami fungsi gen, terutama dalam mengidentifikasi gen-gen esensial.

Dalam penelitian untuk mengidentifikasi esensialitas gen, RNAi digunakan untuk membungkam ekspresi gen tertentu secara selektif. Pada makalah ini, gen-gen baru pada *Drosophila* diuji dengan menggunakan RNAi, dan hasil *knockdown* menunjukkan bahwa sekitar 30% dari gen yang baru terbentuk adalah gen esensial untuk viabilitas. Gen-gen esensial diidentifikasi melalui hilangnya viabilitas organisme ketika ekspresi gen tersebut dihambat oleh RNAi (Chen et al., 2010).

RNAi memerlukan eksperimen biologis yang melibatkan modifikasi genetik pada organisme hidup, sehingga memakan waktu lebih lama untuk mempersiapkan, menjalankan, dan menganalisis hasil, terutama dalam skala besar seperti skrining genom. Proses ini juga melibatkan biaya untuk bahan biologis, peralatan laboratorium, dan analisis molekuler.

2.8 *Drosophila melanogaster*

Drosophila melanogaster, yang sering disebut lalat buah, menjadi organisme model yang umum digunakan dalam penelitian ilmiah dan medis. Sekitar 75% dari gen yang menyebabkan penyakit pada manusia memiliki kesamaan fungsional dalam *Drosophila melanogaster* (Bier, 2005). *Drosophila melanogaster* dapat dilihat pada Gambar 1.

Gambar 1. *Drosophila melanogaster* (Edelsparre et al., 2021).

Dalam pengklasifikasian taksonomi, lalat buah (*Drosophila melanogaster*) dapat dikelompokkan sebagai berikut (O'Grady & DeSalle, 2018):

Kingdom : *Animalia*
Phylum : *Arthropoda*
Subphylum : *Hexapoda*
Class : *Insecta*
Ordo : *Diphtera*
Family : *Drosophilidae*
Genus : *Drosophila*
Spesies : *Drosophila melanogaster*

D. melanogaster menunjukkan fitur-fitur karakteristik anatomi seperti mata majemuk dan sayap. Lalat buah memiliki siklus hidup yang cepat dari satu

pasangan kawin yang subur, yang dapat menghasilkan ratusan keturunan yang genetik serupa dalam waktu 10 hingga 12 hari pada suhu 25°C. Lalat buah dapat dipelihara secara efisien dengan biaya yang rendah di laboratorium dan diakui sebagai model alternatif dibandingkan dengan vertebrata lainnya (Adesola et al., 2021). Ciri fisik lalat buah juga menjadikannya model yang berharga, karena organ dan jaringan lalat dewasa memiliki fungsi yang setara dengan organ dan jaringan manusia (jantung, paru-paru, ginjal, usus, dan saluran reproduksi (Pandey dan Nichols, 2011).

2.9 Tokenisasi

Tokenisasi mengacu pada proses membagi teks atau data teks menjadi unit-unit yang lebih kecil, yang disebut "token,"(Dotan et al., 2024). Dimana setiap token mewakili satu kata. Dalam proses tokenisasi, setiap kata direpresentasikan oleh angka bulat yang unik. Data yang telah melalui proses tokenisasi dapat digunakan untuk melatih model.

Jenis tokenisasi antara lain *word-level, subword-level* dan *character-level* (Toraman et al., 2023). Dalam tokenisasi *word-level tokenization*, teks dibagi menjadi kata-kata individual. Setiap kata menjadi satu token. Dalam *character-level tokenization*, teks dipecah menjadi karakter individual. Setiap karakter menjadi satu token. *Subword-level tokenization* melibatkan pembagian kata-kata menjadi unit-unit subkata yang lebih kecil, seperti awalan atau akhiran. Subkata tersebut kemudian menjadi token.

Dalam protein, metode tokenisasi yang paling sederhana dan umum adalah menganggap setiap asam amino individual sebagai *character-level tokenization* (Ofer et al., 2021). Tidak seperti bahasa manusia yang memiliki kata-kata terdefinisi dengan baik, protein tidak memiliki unit makna yang setara dengan kata. Oleh karena itu, memperlakukan setiap asam amino sebagai token pada tingkat karakter dapat lebih sesuai. Ilustrasi *character-level tokenization* dapat dilihat pada Tabel 3.

Tabel 3. *Character-level Tokenization*

Sebelum ditokenisasi	**Setelah ditokenisasi**
MAVLRAVLGL	‘M’ : 1, ‘A’ : 2, ‘V’ :3, ‘L’:4, ‘R’ : 5, ‘G’ : 6 [1 2 3 4 5 2 3 4 6 4]

2.10 *Padding*

Padding adalah tahapan yang bertujuan untuk membuat setiap *input* sekuens memiliki panjang yang sama. Data sekuens protein memiliki panjang yang bervariasi dan perlu dilakukan *padding* untuk mengubah *input* tersebut menjadi vektor dengan panjang tetap (Lopez-del Rio et al., 2020).

Ada dua tipe *padding*, yaitu *pre padding* dan *post padding*. Dalam *pre-padding*, nilai *padding* (biasanya 0) ditambahkan di awal sekuens sehingga sekuens menjadi panjang yang diinginkan sebelum memasukkannya ke dalam model. Misalnya, jika panjang maksimum sekuens yang diinginkan adalah 10 dan sekuens asli memiliki panjang 7, tiga nilai *padding* akan ditambahkan di bagian depan sekuens.

Dalam *post-padding*, nilai *padding* ditambahkan di akhir sekuens sehingga sekuens menjadi panjang yang diinginkan setelah sekuens asli. Menggunakan contoh yang sama dengan sebelumnya, jika panjang maksimum sekuens yang diinginkan adalah 10, tiga nilai *padding* akan ditambahkan di bagian belakang sekuens. Contoh pre-padding dan post-padding dapat dilihat pada Tabel 4.

Tabel 4. *Padding* Sekuens

Sekuens Sebelum *padding*	***Pre-padding***	***Post-padding***
[1 2 5 3 4 6 4]	[0 0 0 1 2 5 3 4 6 4]	[1 2 5 3 4 6 4 0 0 0]

2.11 *Random Undersampling*

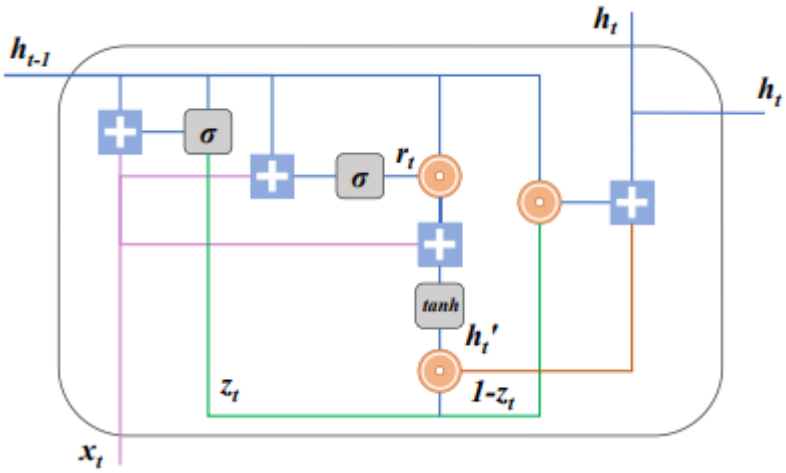
Random undersampling digunakan untuk mengatasi data yang tidak seimbang. Suatu fenomena pada kumpulan data dapat dikatakan tidak seimbang apabila jumlah data pada satu kelas tertentu atau kelas mayor lebih banyak dibandingkan dengan jumlah data pada kelas yang lain atau kelas minor. Pada teknik *under sampling*, proses *sampling* dilakukan dengan mengurangi atau mengeliminasi bagian data pada kelas mayoritas. *Random undersampling* dilakukan dengan menghapus data dari kelas yang dominan secara acak hingga kedua kelas memiliki jumlah data yang seimbang (Batista et al., 2004).

2.12 *Embedding Layer*

Embedding layer digunakan untuk mengubah *input* diskrit menjadi titik-titik dalam ruang vektor, disebut vektor *embedding*. Vektor *embedding* merupakan komponen penting dalam pemrosesan bahasa alami di mana mereka digunakan untuk merepresentasikan kata-kata dalam ruang vektor berdimensi L, di mana L adalah panjang vektor. Hubungan jarak di antara vektor-vektor tersebut adalah representasi dari hubungan satu sama lain. Dalam pemrosesan bahasa alami, vektor-vektor ini merepresentasikan hubungan di antara token *input* (Tawab et al., 2019).

2.13 BiGRU

Bidirectional Gated Recurrent Unit merupakan arsitektur yang menggunakan lapisan rekuren tipe GRU untuk memproses informasi dalam dua arah (Alsanousi et al., 2022). GRU (*Gated Recurrent Unit*) adalah bentuk dari model berurutan yang mengatasi masalah ketergantungan jangka panjang, yang dapat menyebabkan hilangnya gradien pada jaringan saraf yang lebih luas seperti pada jaringan saraf biasa. Struktur GRU dapat dilihat pada Gambar 2.



Gambar 2. Struktur GRU (Yang et al., 2022).

Struktur internal dari GRU mencakup *update gate*, yang menentukan data mana yang akan dihapus dan materi apa yang akan disertakan, sedangkan *reset gate* menentukan seberapa banyak pengetahuan sebelumnya yang akan dilupakan. (Alsanousi et al., 2022).

Pada tahap *update gate*, ditentukan seberapa banyak informasi dari waktu sebelumnya yang dapat dipertahankan sebagai *input* untuk perhitungan di waktu berikutnya. Proses ini untuk mengatur kontribusi informasi masa lalu pada *hidden layer* dan menentukan pengaruhnya terhadap *output* saat ini. Langkah-langkah ini terjadi di dalam lapisan *update gate* dengan menerapkan fungsi aktivasi *sigmoid*. *Output* yang dihasilkan memiliki rentang nilai antara 0 hingga 1. Dalam model GRU, *update gate* dapat dihitung dengan persamaan (1).

$z_t = \sigma\,(w^{(z)}x_t + u^{(z)}h_{t-1} + b)$…….(1)

z_t = *update gate*

σ = fungsi aktivasi *sigmoid*

w,u = *weight*

x = *input*

h_{t-1} = *hidden state*

b = bias

Fungsi aktivasi *sigmoid* dapat dihitung dengan Persamaan (2)

$F(x) = \sigma(x) = \frac{1}{1 + e^{-x}}$……..……(2)

Reset gate berperan dalam menentukan sejauh mana informasi sebelumnya yang tidak relevan akan dihapuskan, menggunakan fungsi aktivasi *sigmoid*. Hasil *output* dari *reset gat*e memiliki rentang nilai antara 0 hingga 1. Ketika nilainya mendekati 0, itu menunjukkan bahwa informasi dari waktu sebelumnya kurang berpengaruh dan kemungkinan akan dihapus. Sebaliknya, jika nilai mendekati 1, itu menandakan bahwa informasi sebelumnya memiliki dampak yang signifikan dan akan tetap disimpan. *Reset gate* dapat dihitung dengan persamaan (3).

$r_t = \sigma\ (w^{(r)}x_t + u^{(r)}h_{t-1} + b)$……..(3)

r_t = *reset gate*
σ = fungsi aktivasi *sigmoid*
w,u = *weight* (bobot)
x = *input*
h_{t-1} = *hidden state*
b = bias

Langkah berikutnya adalah mengukur nilai kandidat *hidden state* atau output pada waktu saat ini (t) berdasarkan informasi relevan dari waktu sebelumnya (t-1) dengan memanfaatkan fungsi aktivasi *tanh*. Ini bukanlah hasil akhir, melainkan representasi memori pada saat sekarang. Penentuan nilai ini dipengaruhi oleh output dari *reset gate*. Kandidat *hidden state* dihitung dengan mengunakan persamaan (4).

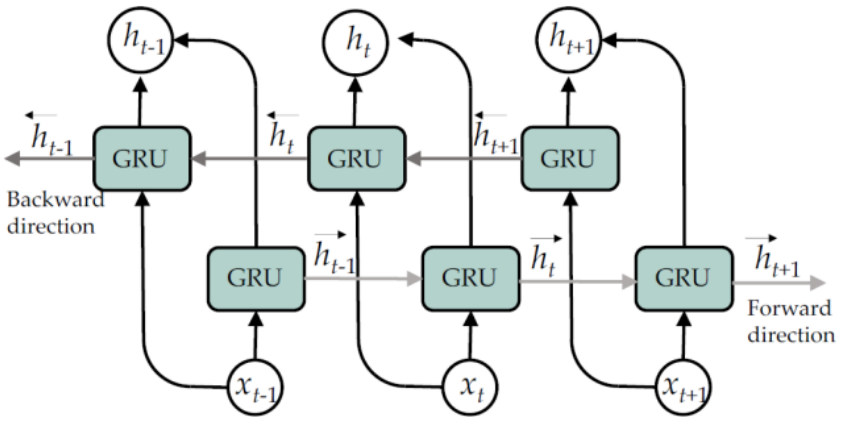
$h'_t = \tanh(w^{(h)}\ x_t + r_t \odot u^{(h)}\ h_{t-1} +\ b)$……(4)

Fungsi aktivasi *tanh*, singkatan dari *hyperbolic tangent*, adalah jenis fungsi aktivasi yang mengonversi *input* numerik menjadi rentang nilai antara -1 dan 1. Persamaan untuk tanh dapat dilihat di persamaan (5).

$$\tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \quad \text{.....(5)}$$

Tahap berikutnya yaitu perhitungan informasi akhir dari unit saat ini, di mana perhitungannya dipengaruhi oleh kandidat *hidden state*, nilai *hidden state* pada waktu sebelumnya, dan output dari *update gate*. Informasi ini selanjutnya disampaikan ke waktu berikutnya sebagai *hidden state*, yang kemudian digunakan kembali untuk menghasilkan output pada unit waktu tersebut. Proses ini berlangsung secara konsisten dan terus berulang ke depan, dengan hanya variasi nilai *input* sebagai perbedaannya.

$$h_t = z_t \odot h_{t-1} + (1 - z_t) \odot h'_t \quad \text{.....(6)}$$



Gambar 3. Struktur BiGRU (Mekruksavanich & Jitpattanakul, 2023).

Ide dasar di balik BiGRU adalah bahwa urutan *input* diproses melalui arah maju dan mundur. Arsitektur dengan dua arah ini dapat menekuni data masa lalu dan data masa yang akan datang untuk setiap sekuen *input* yang sangat berguna untuk bisa mengakses dari informasi sebelum dan setelahnya.

Berikut contoh perhitungan dengan metode GRU :

Input teks :MQLTMQLTMQLTMQLTMQLT

Input tersebut perlu diubah menjadi bentuk numerik dengan ketentuan berikut : {‘T’: 0, ‘Q’: 1, ‘L’: 2, ‘M’: 3}

Maka, *input* akan menjadi sebagai berikut :

[3, 1, 2, 0, 3, 1, 2, 0, 3, 1, 2, 0, 3, 1, 2, 0, 3, 1, 2, 0]

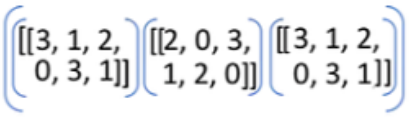
Selanjutnya, *input* dibagi menjadi sekuens dengan panjang tiga.

[(3, 1, 2, 0, 3, 1, 2, 0, 3, 1, 2, 0, 3, 1, 2, 0, 3, 1, 2, 0]

Data dibagi ke dalam *batch size*=2.

[(3, 1, 2, 0, 3, 1, | 2, 0, 3, 1, 2, 0, | 3, 1, 2, 0, 3, 1, | 2, 0]

Dari *batch size* kemudian dibagi menjadi *mini batches* dan di *transpose*.



Setelah dilakukan *transpose* :

$$\begin{pmatrix} \left(\begin{bmatrix}3\\0\end{bmatrix} & \begin{bmatrix}1\\3\end{bmatrix} & \begin{bmatrix}2\\1\end{bmatrix}\right) \\ \left(\begin{bmatrix}2\\1\end{bmatrix} & \begin{bmatrix}0\\2\end{bmatrix} & \begin{bmatrix}3\\0\end{bmatrix}\right) \\ \left(\begin{bmatrix}3\\0\end{bmatrix} & \begin{bmatrix}1\\3\end{bmatrix} & \begin{bmatrix}2\\1\end{bmatrix}\right) \end{pmatrix}$$

Kemudian, melakukan *one hot encoding*

$$\begin{pmatrix} \left(\begin{bmatrix}3\\0\end{bmatrix} & \begin{bmatrix}1\\3\end{bmatrix} & \begin{bmatrix}2\\1\end{bmatrix}\right) \\ \left(\begin{bmatrix}2\\1\end{bmatrix} & \begin{bmatrix}0\\2\end{bmatrix} & \begin{bmatrix}3\\0\end{bmatrix}\right) \\ \left(\begin{bmatrix}3\\0\end{bmatrix} & \begin{bmatrix}1\\3\end{bmatrix} & \begin{bmatrix}2\\1\end{bmatrix}\right) \end{pmatrix}$$

$$\begin{pmatrix} \left(\begin{pmatrix}0&0&0&1\\1&0&0&0\end{pmatrix} & \begin{pmatrix}0&1&0&0\\0&0&0&1\end{pmatrix} & \begin{pmatrix}0&0&1&0\\0&1&0&0\end{pmatrix}\right) \\ \left(\begin{pmatrix}0&0&1&0\\0&1&0&0\end{pmatrix} & \begin{pmatrix}1&0&0&0\\0&0&1&0\end{pmatrix} & \begin{pmatrix}0&0&0&1\\1&0&0&0\end{pmatrix}\right) \\ \left(\begin{pmatrix}0&0&0&1\\1&0&0&0\end{pmatrix} & \begin{pmatrix}0&1&0&0\\0&0&1&0\end{pmatrix} & \begin{pmatrix}0&0&1&0\\0&1&0&0\end{pmatrix}\right) \end{pmatrix}$$

Batch 1

$$\left(\begin{pmatrix}0 & 0 & 0 & 1\\ 1 & 0 & 0 & 0\end{pmatrix}\ \begin{pmatrix}0 & 1 & 0 & 0\\ 0 & 0 & 0 & 1\end{pmatrix}\ \begin{pmatrix}0 & 0 & 1 & 0\\ 0 & 1 & 0 & 0\end{pmatrix}\right)$$

Untuk *weight* diinisiasi dengan random

$$\left(\begin{pmatrix}0.6614 & 0.2669\\ 0.0617 & 0.6213\\ 0.4519 & -0.1661\\ -1.5228 & 0.3817\end{pmatrix}\right)\left(\begin{pmatrix}0.3225 & -0.4791\\ 1.3790 & 2.5286\end{pmatrix}\right)$$

Wz Uz

Berikutnya menghitung *update gate* dengan persamaan sebagai berikut. *Update gate* (z) menentukan seberapa berguna informasi masa lalu bagi keadaan saat ini. Penggunaan fungsi sigmoid menghasilkan nilai *update gate* antara 0 dan 1.

$$z = \sigma\left(\left(\begin{pmatrix}0 & 0 & 0 & 1\\ 1 & 0 & 0 & 0\end{pmatrix}W_z + U_z h_{t-1} + b_z\right)\right)$$

$$z = \sigma\left(\left(\begin{pmatrix}0 & 0 & 0 & 1\\ 1 & 0 & 0 & 0\end{pmatrix}\begin{pmatrix}0.6614 & 0.2669\\ 0.0617 & 0.6213\\ 0.4519 & -0.1661\\ -1.5228 & 0.3817\end{pmatrix} + \begin{pmatrix}0.3225 & -0.4791\\ 1.3790 & 2.5286\end{pmatrix}\begin{pmatrix}0 & 0\\ 0 & 0\end{pmatrix} + \begin{pmatrix}0 & 0\\ 0 & 0\end{pmatrix}\right)\right)$$

$$z = \sigma\begin{pmatrix}-1.5228 & 0.3817\\ 0.6614 & 0.2699\end{pmatrix}$$

$$z = \begin{pmatrix}0.1792 & 0.5943\\ 0.6596 & 0.5663\end{pmatrix}$$

Kemudian, menghitung *reset gate*. Dalam setiap *batch*, *reset gate* akan mengevaluasi kembali kinerja kombinasi dari input sebelumnya dan input yang baru sesuai dengan kebutuhan input baru. Pada fungsi aktifasi sigmoid, nilai yang mendekati 0 berarti harus mengabaikan nilai *hidden state* sebelumnya dan sebaliknya untuk nilai yang mendekati 1.

$r_t = \sigma\ (w^{(r)}x_t + u^{(r)}h_{t-1} + b)$

$$r_t = \begin{pmatrix}0.6041 & 0.5664\\ 0.2653 & 0.3628\end{pmatrix}$$

Lalu, menghitung kandidat *hidden state*.

$h'_t = \tanh(w^{(h)}\ x_t + r_t \odot u^{(h)}\ \ h_{t-1} + \ b)$

$$h'_t = \tanh\begin{pmatrix}1.5987 & -1.2770\\ -1.4212 & -0.5107\end{pmatrix} + \begin{pmatrix}0 & 0\\ 0 & 0\end{pmatrix} + \begin{pmatrix}0 & 0\\ 0 & 0\end{pmatrix}$$

$$h'_t = \begin{pmatrix} 0.9215 & -0.8557 \\ -0.3979 & -0.4705 \end{pmatrix}$$

Langkah selanjutnya adalah menghitung *hidden state*.

$$h_t = z_t \odot h_{t-1} + (1 - z_t) \odot h'_t$$

$$h_t = \begin{pmatrix} 0.1791 & 0.5943 \\ 0.6596 & 0.5663 \end{pmatrix} \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} + \left(1 - \begin{pmatrix} 0.1791 & 0.5943 \\ 0.6596 & 0.5663 \end{pmatrix}\right) \begin{pmatrix} 0.9215 & -0.8557 \\ -0.3979 & -0.4705 \end{pmatrix}$$

$$h_t = \begin{pmatrix} 0.7565 & -0.3472 \\ -0.1355 & -0.2040 \end{pmatrix}$$

Setelah ini, *hidden state* yang baru dibuat akan menjadi $h_{(t-1)}$ yang digunakan untuk menciptakan *hidden state* berikutnya. Proses perhitungan ini akan diulang untuk setiap kelompok data berikutnya. Dengan demikian, nilai *hidden state* untuk setiap *input* dari *batch* pertama hingga *batch* ketiga dapat diperoleh sebagai berikut.

$$\left(\begin{pmatrix} 0.7565 & -0.3472 \\ -0.1355 & -0.2040 \end{pmatrix} \begin{pmatrix} -0.1535 & -0.5712 \\ 0.7664 & -0.5062 \end{pmatrix} \begin{pmatrix} 0.7495 & -0.8616 \\ -0.2399 & -0.6680 \end{pmatrix} \right)$$

$$\left(\begin{pmatrix} 0.9491 & -0.9869 \\ -0.7454 & -0.0868 \end{pmatrix} \begin{pmatrix} 0.1406 & -0.9392 \\ 0.4630 & -0.4591 \end{pmatrix} \begin{pmatrix} 0.8373 & -0.9050 \\ 0.0556 & -0.6569 \end{pmatrix} \right)$$

$$\left(\begin{pmatrix} 0.9054 & -0.9849 \\ -0.2446 & -0.5224 \end{pmatrix} \begin{pmatrix} -0.4335 & -0.8752 \\ 0.7853 & -0.6418 \end{pmatrix} \begin{pmatrix} 0.7948 & -0.8400 \\ -0.3061 & -0.7358 \end{pmatrix} \right)$$

Langkah selanjutnya adalah melakukan perhitungan prediksi untuk setiap langkah waktu (t). Untuk melakukan prediksi pada setiap langkah waktu, langkah awalnya adalah mengubah hasil keluaran menggunakan lapisan linear. *Input* yang diberikan pada awalnya terdiri dari empat karakter unik, dan karena itu, keluaran yang diinginkan juga harus memiliki dimensi yang sama. Oleh

karena itu, digunakan lapisan *dense* atau *fully connected layer* untuk mengubah hasil keluaran tersebut sehingga memiliki dimensi yang serupa dengan input. Lapisan ini kemudian melewati fungsi aktivasi, di mana fungsi aktivasi yang digunakan adalah *softmax*. Persamaan untuk menghitung lapisan linear dijelaskan sebagai berikut.

$\text{Linear} = W_y h_{t-1} + by$ ……(7)

Perhitungan untuk memprediksi *output* dimulai dari mengubah menjadi linear adalah sebagai berikut.

$$\text{Linear}=\begin{pmatrix} 0.7565 & -0.3472 \\ -0.1355 & -0.2040 \end{pmatrix}\begin{pmatrix} 0.8310 & -0.2477 & -0.8029 & 0.2366 \\ 0.2857 & 0.6898 & -0.6331 & 0.8795 \end{pmatrix}+\begin{pmatrix} 0 & 0 & 0 & 0 \end{pmatrix}$$

$$\text{Linear} = \begin{pmatrix} 0.5295 & -0.4269 & -0.3876 & -0.1264 \\ -1.1709 & -0.1072 & 0.2379 & -0.2115 \end{pmatrix}$$

Hasil linear untuk setiap *batch* adalah sebagai berikut.

$$\begin{pmatrix} \begin{pmatrix} 0.5295 & -0.4269 & -0.3876 & -0.1264 \\ -0.1709 & -0.1072 & 0.2379 & -0.2115 \end{pmatrix} \\ \begin{pmatrix} -0.2908 & -0.3560 & 0.4849 & -0.5387 \\ 0.4922 & -0.5390 & -0.2949 & -0.2639 \end{pmatrix} \\ \begin{pmatrix} 0.3767 & -0.7800 & -0.0563 & -0.5805 \\ -0.3902 & -0.4014 & 0.6155 & -0.6443 \end{pmatrix} \end{pmatrix}$$

$$\begin{pmatrix} \begin{pmatrix} 0.5068 & -0.9159 & -0.1373 & -0.6434 \\ -0.6442 & 0.1248 & 0.6534 & -0.2527 \end{pmatrix} \\ \begin{pmatrix} -0.1515 & -0.6827 & 0.4817 & -0.7928 \\ 0.2536 & -0.4314 & -0.0811 & -0.2942 \end{pmatrix} \\ \begin{pmatrix} 0.4373 & -0.8317 & -0.0994 & -0.5978 \\ -0.1415 & -0.4669 & 0.3713 & -0.5646 \end{pmatrix} \end{pmatrix}$$

$$\begin{pmatrix} \begin{pmatrix} 0.4710 & -0.9037 & -0.1035 & -0.6520 \\ -0.3525 & -0.2998 & 0.5271 & -0.5173 \end{pmatrix} \\ \begin{pmatrix} -0.6103 & -0.4963 & 0.9021 & -0.8723 \\ 0.4692 & -0.6372 & -0.2242 & -0.3786 \end{pmatrix} \\ \begin{pmatrix} 0.4205 & -0.7763 & -0.1064 & -0.5507 \\ -0.4646 & -0.4318 & 0.7116 & -0.7186 \end{pmatrix} \end{pmatrix}$$

Kemudian menetapkan aktivasi *softmax* untuk menormalkan *output* menjadi distribusi probabilitas yang berjumlah 1. Fungsi *softmax* ditunjukkan sebagai berikut.

$$\textit{softmax}\ x_i = \frac{\exp(x_i)}{\Sigma_j \exp(x_i)} \quad \text{……(8)}$$

Perhitungan untuk fungsi *softmax* adalah sebagai berikut.

1. Kurangi nilai dari seluruh elemen dengan nilai maksimum

$$\exp\left(y_{linear} - y_{linear\_max}\right) = \exp\begin{pmatrix} 0.5295 & -0.4269 & -0.3876 & -0.1264 \\ -0.1709 & -0.1072 & 0.2379 & -0.2115 \end{pmatrix} - 0.9021$$

$$\exp\left(y_{linear} - y_{linear\_max}\right) = \begin{pmatrix} exp^{-0.3726} & exp^{-1.3290} & exp^{-1.2898} & exp^{-1.0285} \\ exp^{-1.0730} & exp^{-1.0093} & exp^{-0.6642} & exp^{-1.1136} \end{pmatrix}$$

$$\exp\left(y_{linear} - y_{linear\_max}\right) = \begin{pmatrix} 0.6889 & 0.2648 & 0.2753 & 0.3575 \\ 0.3420 & 0.3645 & 0.5147 & 0.3284 \end{pmatrix}$$

2. Jumlahkan seluruh elemen matriks

$$\sum \exp^{\exp\left(y_{linear} - y_{linear\_max}\right)} = \begin{pmatrix} 0.6889 + & 0.2648 + & 0.2753 + & 0.3575 \\ 0.3420 + & 0.3645 + & 0.5147 + & 0.3284 \end{pmatrix}$$

$$\sum \exp^{\exp\left(y_{linear} - y_{linear\_max}\right)} = \begin{pmatrix} 1.5865 \\ 1.5495 \end{pmatrix}$$

3. Bagi setiap elemen dalam matriks dari langkah 1 dengan nilai dari langkah

$$softmax = \begin{pmatrix} \frac{0.6889}{1.5865} & \frac{0.2648}{1.5865} & \frac{0.2753}{1.5865} & \frac{0.3575}{1.5865} \\ \frac{0.3420}{1.5495} & \frac{0.3645}{1.5495} & \frac{0.5147}{1.5495} & \frac{0.3284}{1.5495} \end{pmatrix}$$

$$softmax = \begin{pmatrix} 0.4342 & 0.1669 & 0.1735 & 0.2254 \\ 0.2207 & 0.2352 & 0.3322 & 0.2119 \end{pmatrix}$$

T Q L M

Dari hasil tersebut, didapatkan bahwa probalititas tertinggi adalah T, sehingga hasil prediksi setelah sekuens MQ adalah huruf T.

2.14 *Flatten*

Penempatan layer *flatten* terletak di antara layer BiGRU dan layer *dense* berikutnya. Fungsi utama dari layer *flatten* adalah melakukan proses *reshape* keluaran yang berasal dari layer sebelumnya, sehingga dapat dialirkan ke dalam layer *dense*. Output dari layer sebelumnya berbentuk tensor dengan multidimensi, sedangkan layer dense membutuhkan tensor satu dimensi. Oleh

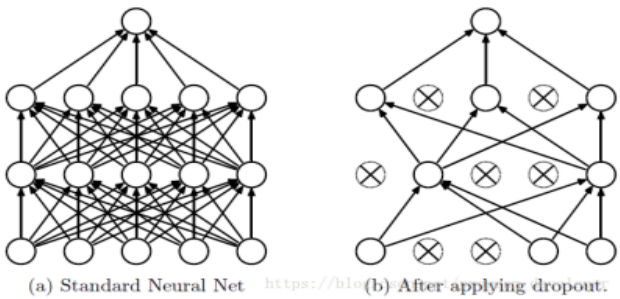
karena itu, layer flatten melakukan transformasi pada tensor multi-dimensi tersebut, menjadikannya tensor satu dimensi (Tan et al., 2023).

2.15 *Dense Layer*

Layer *dense*, yang juga disebut sebagai *fully-connected*. Layer ini menghubungkan setiap neuron pada layer sebelumnya dengan setiap neuron pada layer berikutnya, menciptakan pola konektivitas yang rapat. Layer *dense* menerima keluaran yang telah melalui proses *flatten* dari layer sebelumnya dan menerapkan serangkaian operasi matriks untuk memodelkan hubungan antara *hidden states* dan label kelas (Tan et al., 2023).

2.16 *Dropout*

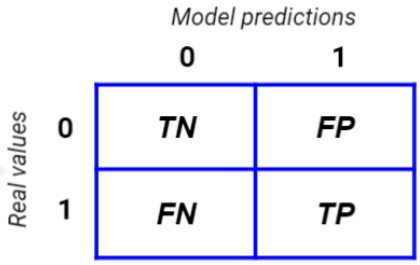
Dropout adalah suatu metode yang digunakan untuk mencegah terjadinya *overfitting* dan meningkatkan kecepatan pembelajaran mesin. Teknik ini melibatkan penghapusan neuron, baik yang berada dalam *hidden layer* maupun *visible layer* di dalam jaringan (Asghar et al., 2021). Dengan menghapus suatu neuron, hal ini menghilangkan secara sementara dari jaringan yang sedang berlangsung. Neuron yang akan dihapus dipilih secara acak pada setiap iterasi pembelajaran. Ilustrasi *dropout* dapat dilihat pada Gambar 4.



Gambar 4. *Dropout* (Xie, 2020).

2.17 *Confusion Matrix*

Confusion matrix digunakan untuk memeriksa kinerja model pembelajaran mesin berbasis klasifikasi. *Confusion matrix* adalah tabel ringkasan dari jumlah prediksi yang benar dan salah yang dihasilkan oleh suatu pengklasifikasi (atau model klasifikasi) dengan matriks berukuran N x N di mana N adalah jumlah kelas target (Karsito & Susanti, 2019).



Gambar 5. *Confusion Matrix.*

- TP: *True Positive*: data positif yang diklasifikasikan dengan benar
- FP: *False Positive*: data negatif namun diklasifikasikan sebagai data positif.
- FN: *False Negative*: data positif namun diklasifikasikan sebagai data negatif.
- TN: *True Negative*: data negatif yang diklasifikasikan dengan benar.

2.18 PR AUC

Kurva *precision-recall* merupakan kurva yang menggambarkan *precision* terhadap *recall* (Rahbari et al., 2017). *Precision* merupakan perbandingan prediksi benar positif dibandingkan dengan keseluruhan hasil yang diprediksi positif. *Recall* merupakan perbandingan prediksi benar positif dibandingkan dengan keseluruhan data yang benar positif.

Untuk perhitungan *precision* ditunjukkan oleh Persamaan (9) dan *recall* ditunjukkan oleh Persamaan (10).

$$Precision = \frac{TP}{TP+FP} \quad (9)$$

$$Recall = \frac{TP}{TP+FN} \quad (10)$$

2.19 ROC AUC

Grafik ROC adalah grafik dua dimensi hubungan antara *True Positive Rate* (TPR) atau *Sensitivity* (sumbu Y) dengan *False Positive Rate* (FPR) atau 1-*Specificity* (sumbu X)(Fawcett, 2006).

TPR juga disebut dengan *sensitivity/recall* mengukur seberapa baik model terbentuk untuk memprediksi secara tepat data testing di kelas positif yang tepat terprediksi ke dalam kelas positif. Persamaan untuk TPR dapat dilihat pada Persamaan (11).

$$TPR = \frac{TP}{TP+FN} \quad (11)$$

False Positive Rate (FPR) merupakan proporsi kasus negatif yang salah diprediksi sebagai positif. Persamaan untuk FPR dapat dilihat pada Persamaan (12).

$$FPR = \frac{FP}{FP+TN} \quad (12)$$

FPR bisa didapat dari *specificity* yaitu, 1 – *specificity*. Untuk persamaan *specificity* ditunjukkan pada Persamaan (13).

$$Specificity = \frac{TN}{FP+TN} \quad (13)$$

2.20 *Underfitting*

Underfitting terjadi ketika model tidak cukup baik dalam menangkap pola dari data pelatihan, sehingga hasilnya kurang optimal baik pada data pelatihan maupun data baru. Ini berarti model terlalu sederhana untuk data yang ada, sehingga tidak dapat memprediksi dengan baik. *Underfitting* dapat diatasi dengan menambah fitur atau variabel yang relevan, atau dengan menggunakan model yang lebih kompleks (Aliferis & Simon, 2024).

2.21 *Overfitting*

Overfitting pada model terjadi ketika model tersebut sangat cocok dengan data pelatihan, namun tidak mampu melakukan generalisasi dengan baik pada data baru yang berasal dari distribusi yang sama. Ini disebabkan oleh beberapa pola yang dipelajari dari data pelatihan tidak mencerminkan populasi secara keseluruhan (Aliferis & Simon, 2024).

Overfitting terjadi karena model terlalu dipaksakan untuk menyesuaikan diri dengan *noise* atau variasi yang tidak relevan dalam data latih. *Overfitting* dapat dihindari dengan menggunakan teknik regulasi.

III. METODE PENELITIAN

3.1 Tempat dan Waktu Penelitian

Pemaparan tempat dan waktu penelitian yaitu sebagai berikut :

3.1.1 Tempat Penelitian

Penelitian dilakukan di Laboratorium Komputasi Dasar Jurusan Ilmu Komputer Fakultas Matematika Dan Ilmu Pengetahuan Universitas Lampung.

3.1.2 Waktu Penelitian

Penelitian dilakukan pada bulan November 2023 di semester tujuh hingga penyelesaian pada bulan Agustus 2024. Alur pengerjaan dapat dilihat di Tabel 5.

Tabel 5. Alur Pengerjaan Penelitian

No	Kegiatan	2023																2024																							
		November				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																																								
3	Penyusunan bab I-III																																								
4	*Cleaning* data																																								
5	Tokenisasi																																								
6	*Padding*																																								
7	*Random Undersampling*																																								
8	Pemodelan BiGRU																																								
9	Evaluasi model																																								
10	Penyusunan bab IV-V																																								

3.2 Data dan Alat

3.2.1 Data

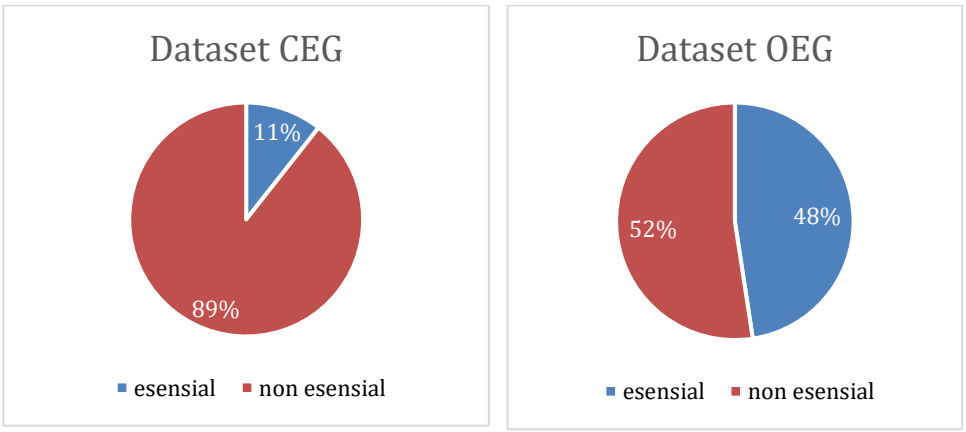
Data yang digunakan dalam penelitian merupakan *dataset* sekuens protein dari *Drosophila melanogaster* yang diperoleh dari penelitian Beder , et al (2021). Data diperoleh dari *Database of Essential Genes* (DEG). DEG (*Database of Essential Genes*) adalah basis data yang berisi gen-gen esensial yang dianggap sangat penting untuk mendukung kehidupan seluler, yang berfungsi sebagai fondasi kehidupan dengan mengkode fungsi-fungsi vital seperti replikasi DNA, transkripsi gen, sintesis protein, dan produksi energi. DEG menyimpan elemen genom esensial dari tiga domain kehidupan yaitu bakteri, arkea, dan eukariota. Gen esensial ini ditemukan melalui berbagai metode, seperti mutagenesis transposon, RNA *antisense*, dan komparatif genomik pada berbagai organisme. DEG (*Database of Essential Genes*) versi 15 adalah pembaruan signifikan dari versi sebelumnya, yang mencakup peningkatan besar dalam jumlah gen esensial yang diidentifikasi (Luo et al., 2021).

Pada penelitian Beder, et al (2021) terdiri dari enam organisme eukariotik diantaranya *Caenorhabditis elegans, D.melanogaster, H. sapiens, M. musculus, S. cerevisiae* dan *S. pombe*. Namun, pada penelitian yang akan dilakukan hanya akan menggunakan data *Drosophila melanogaster*. *Dataset* tersebut terdiri dari data *Cellular Essential Gene* (CEG) dan *Organismal Essential Gene* (OEG). CEG mengacu pada gen-gen yang diperlukan untuk kelangsungan hidup dan fungsi sel individual. Gen-gen ini memainkan peran penting dalam memastikan proses-proses dasar dan vital dalam siklus hidup sel. OEG merujuk pada gen-gen yang penting untuk kelangsungan hidup organisme secara keseluruhan. Ini mencakup gen-gen yang dibutuhkan untuk fungsi-fungsi kritis dalam organisme, seperti pertumbuhan, perkembangan, dan reproduksi. Jumlah *dataset* yang digunakan terdapat pada Tabel 6.

Tabel 6. Jumlah *Dataset*

Gen	CEG	OEG
Esensial	1227	246
Non-esensial	10320	271
Jumlah	11547	517

Berdasarkan Tabel 6, persentase gen esensial dan non-esensial pada *dataset* CEG dan OEG dapat dilihat pada Gambar 6.



Gambar 6. Perbandingan *Dataset* CEG dan OEG.

3.2.2 Alat

1. Perangkat Keras (*Hardware*)

Perangkat keras yang digunakan dalam penelitian ini yaitu:

a. *System Manufacture* : PC
b. *Processor* : AMD Ryzen 5 3400G with Radeon Vega Graphics x
c. *Storage* : WDC WDS500G2B0C-00PXH0

2. Perangkat Lunak (Software)

a. *Operating System* : Ubuntu 22.04.2 LTS 64-Bit.
b. *Tools*

1) *Jupyter Notebook*

Jupyter Notebook adalah aplikasi web interaktif yang memungkinkan pengguna membuat dan berbagi dokumen yang berisi kode sumber, visualisasi, dan teks penjelasan. Dikembangkan sebagai bagian dari proyek *Jupyter*, *notebook* ini mendukung berbagai bahasa pemrograman, dengan *Python* sebagai yang paling umum digunakan. *Jupyter Notebook* sangat berguna untuk *data analysis, machine learning,* dan riset ilmiah karena memungkinkan integrasi langsung antara kode dan *output*, termasuk grafik dan visualisasi data. Selain itu, fitur-fitur seperti penyorotan sintaks, pelaksanaan sel kode individual, dan kemampuan untuk menyimpan dan berbagi notebook dalam format berbagai format menjadikannya alat yang sangat efektif untuk kolaborasi dan dokumentasi ilmiah.

2) *Python* 3.10.13

Python adalah bahasa pemrograman tingkat tinggi. *Python* mendukung pemrograman prosedural, berorientasi objek, dan fungsional, membuatnya sangat fleksibel.

Kelebihan utama *Python* terletak pada kemudahan pembelajaran dan keterbacaan kodenya yang tinggi. *Python* juga memberikan akses ke banyak pustaka (*libraries*).

c. *Packages*

1) *Pandas* 2.2.2

Library Pandas dalam *Python* adalah perpustakaan *open-source* yang memberikan struktur data dan alat analisis data tingkat tinggi, terutama *DataFrame*, yang memungkinkan pengguna untuk menyimpan, memanipulasi, dan menganalisis data tabular dengan mudah.

2) *Numpy* 1.26.4

Numpy (Numerical Python) adalah sebuah perpustakaan *(library)* yang sangat populer dalam ekosistem Python, terutama digunakan untuk manipulasi data numerik dan operasi matematika. Fokus utama *Numpy* adalah menyediakan struktur data array multidimensi yang efisien, bersama dengan berbagai fungsi matematika yang dioptimalkan untuk kinerja tinggi.

3) *Scikit Learn* 1.5.0

Library scikit-learn adalah perpustakaan *machine learning open-source* untuk bahasa pemrograman *Python* yang menyediakan alat dan fungsi untuk keperluan pembelajaran mesin dan analisis data statistik. *Scikit-learn* menyediakan implementasi yang efisien dan mudah digunakan untuk berbagai algoritma *machine learning.*

4) *Tensorflow* 2.12.0

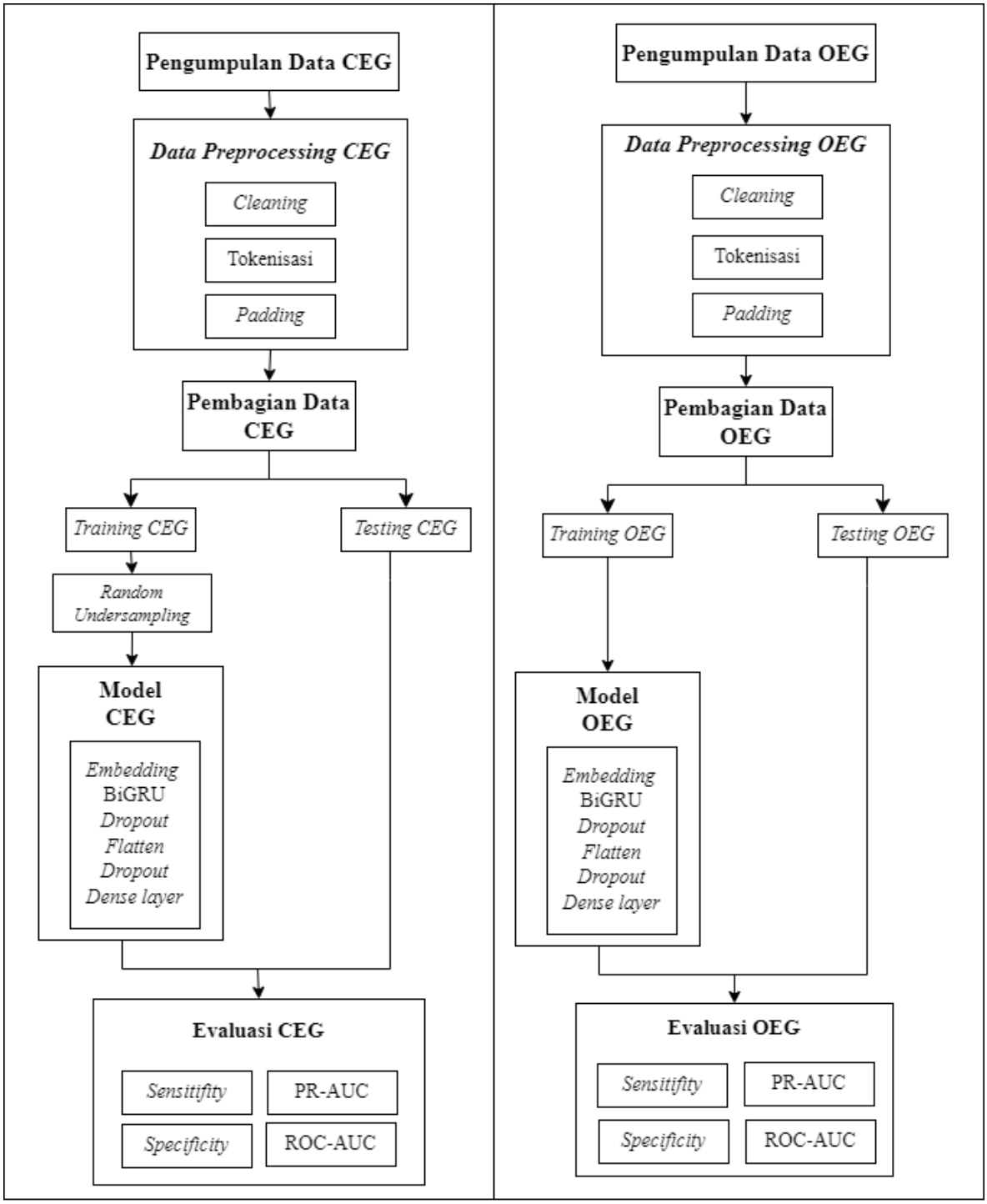
Library TensorFlow adalah perpustakaan *machine learning open-source* yang dikembangkan oleh *Google* untuk mengelola dan melatih model *machine learning* dan *deep learning*. *TensorFlow* menyediakan lingkungan komputasi numerik yang efisien. *TensorFlow* menyediakan lingkungan komputasi numerik yang efisien dengan menggunakan graf komputasi yang terdiri dari node-node yang merepresentasikan operasi matematika.

5) *Matplotlib* 3.7.2

Library Matplotlib adalah pustaka visualisasi data untuk bahasa pemrograman *Python* yang menyediakan antarmuka sederhana dan fleksibel untuk membuat berbagai jenis grafik dan plot. Pustaka ini sangat cocok untuk membuat banyak jenis visualisasi serta dapat digunakan bersama dengan pustaka lain seperti NumPy dan Pandas untuk analisis data yang lebih kompleks.

3.3 Metodologi

Tahapan dalam penelitian ini dapat dilihat pada gambar berikut.



Gambar 7. Alur Kerja Penelitian.

1. Pengumpulan Data

   Data sekuens protein pada *Drosophila melanogaster* didapatkan dari penelitian (Beder et al., 2021). Ada dua jenis dataset yaitu CEG dan OEG. Dataset CEG terdiri dari 12.429 sekuens protein. Untuk dataset OEG juga terdiri dari 12.429 sekuens protein.

2. *Preprocessing*

   Pada tahap ini, dilakukan penghapusan data yang tidak berlabel pada *dataset* CEG dan OEG sehingga menyisakan data-data yang berlabel. Berikutnya dilakukan tokenisasi dengan tipe *character level tokenization* pada data sekuens protein dan memberikan *padding* yang bertujuan untuk menyamakan ukuran *input* sebelum lanjut ke tahap pemodelan. Ada dua jenis padding yang akan dilakukan percobaan serta dilihat perbandingan hasilnya yaitu *pre padding* dan *post padding.*

   Untuk dataset OEG akan menggunakan *padding* dengan panjang maksimal 400 sedangkan dataset CEG akan menggunakan *padding* dengan panjang maksimal 500.

3. Pembagian data

   Pembagian data merupakan proses membagi *dataset* menjadi data *training* dan data *testing*. Data *training* merupakan subset dari *dataset* yang digunakan untuk melatih model, memungkinkan model untuk mempelajari pola dan hubungan antara fitur dan target. Sebaliknya, data *testing* adalah subset terpisah yang tidak terlibat dalam proses pelatihan dan digunakan untuk menguji kinerja model setelah pelatihan. Data *testing* berperan dalam mengukur kemampuan model untuk menggeneralisasi dan membuat prediksi yang akurat pada data yang belum pernah dilihat sebelumnya.

   Pada penelitian ini menggunakan persentase pembagian data yaitu 80% data *training* dan 20% data *testing*. Kemudian, membagi data *training* menjadi *train set* dan *validation set.* Ada dua skema pembagian data yaitu 80% *train set* dan

20% *validation set* serta 90% *train set* dan 10% *validation set.* Skema tersebut akan dilakukan di data CEG maupun OEG.

4. *Random Undersampling*

   Dataset CEG memiliki data yang tidak seimbang dimana data gen non esensial berjumlah 10.230 sedangkan data gen esensial hanya berjumlah 1227. Maka perlu dilakukan *undersampling* dengan teknik *random undersampling*. Proses ini hanya dilakukan pada data *training* CEG.

5. Klasifikasi

   Tahapan selanjutnya adalah melakukan pemodelan atau klasifikasi. Pemodelan dilakukan dengan algoritma *Bidirectional Gated Recurrent Unit* (BiGRU). Pemodelan dilakukan dengan menggunakan arsitektur layer yaitu *embedding layer, bidirectional GRU layer, flatten layer, dense layer,* dan *dropout layer.*

   Arsitektur layer yang digunakan ada dua jenis dengan perbedaan variasi neuron, nilai *dropout*, serta nilai *regularizer*. Arsitektur 1 dengan neuron berjumlah 8, nilai *dropout* 0,1, nilai *regularizer* 0,1 sedangkan Arsitektur II dengan neuron berjumlah 16, nilai *dropout* 0,5, nilai *regularizer* 0,05. Kemudian, setiap arsitektur dilakukan percobaan dengan variasi *epoch* 40,80 dan 120. Kedua arsitektur akan dilakukan percobaan baik di data CEG maupun OEG.

6. Evaluasi

   Tahap ini merupakan tahap terakhir yang akan mengukur performa dari model. Evaluasi dilakukan dengan menghitung ROC AUC, PR AUC, *sensitivity* dan *specificity*. Keempat metrics evaluasi ini juga akan digunakan sebagai perbandingan evaluasi kinerja dengan penelitian sebelumnya dengan *dataset* yang sama.

V. SIMPULAN DAN SARAN

5.1 Simpulan

Berdasarkan penelitian yang telah dilakukan dapat diambil kesimpulan sebagai berikut.

1. Penelitian ini mengimplementasikan metode *Bidirectional Gate Recurrent Unit* (BiGRU) dalam mengklasifikasikan gen esensial pada *Drosophila melanogaster* dengan dua arsitektur, skema pembagian data 80% *training* 20% validasi dan 90% *training* 10% validasi, dan variasi *epoch* (40,80,120).

   Hasil terbaik pada data OEG didapat dari skema dengan pembagian data yaitu 80% *training* dan 20% validasi pada Arsitektur II, *pre-padding* dengan 40 *epoch*. Nilai yang didapat adalah 83% sensitivitas, 79% spesifisitas , 85% nilai PR-AUC dan 80% nilai ROC-AUC. Hasil klasifikasi yang paling baik pada data CEG didapat dari skema dengan pembagian data pelatihan 90% *training* 10% validasi pada Arsitektur I, *pre-padding* dengan 40 *epoch*. Nilai yang didapat untuk sensitivitas adalah 73%, spesifisitas 55%, nilai PR-AUC 46% dan nilai ROC-AUC 64%.

2. Hasil perbandingan dengan penelitian terdahulu menunjukkan bahwa pada penelitian ini memiliki hasil yang lebih baik dari penelitian dengan metode LSTM tetapi memiliki hasil yang lebih rendah dari penelitian Beder, et al, (2021) yang menggunakan Random Forest dan XG Boost. Hal ini berarti metode Bidirectional Gated Recurrent Unit (GRU) pada penelitian ini, belum cukup baik dalam mengklasifikasikan gen esensial pada Drosophila melanogaster dengan parameter yang digunakan.

5.2. Saran

Adapun saran yang diberikan pada penelitian ini adalah sebagai berikut.

1. Penelitian ini dapat dilanjutkan menggunakan metode deep learning lainnya atau metode machine learning.

2. Untuk dataset CEG, penelitian ini dapat dilanjutkan dengan teknik imbalanced data lainnya, oversampling ataupun teknik undersampling yang lain.

DAFTAR PUSTAKA

Aliferis, C., & Simon, G. (2024). Overfitting, Underfitting and General Model Overconfidence and Under-Performance Pitfalls and Best Practices in Machine Learning and AI. *Springer*, *1*, 477–524

Alsanousi, W. A., Ahmed, N. Y., Hamid, E. M., Elbashir, M. K., Musa, M. E. M., Wang, J., Khan, N., & Afnan. (2022). A novel deep learning-assisted hybrid network for plasmodium falciparum parasite mitochondrial proteins classification. *PLoS ONE*, *17*(10), 1–18.

Aromolaran, O., Aromolaran, D., Isewon, I., & Oyelade, J. (2021). Machine learning approach to gene essentiality prediction: A review. In *Briefings in Bioinformatics* (Vol. 22, Issue 5, pp. 1–19). Oxford University Press.

Aromolaran, O., Beder, T., Oswald, M., Oyelade, J., Adebiyi, E., & Koenig, R. (2020). Essential gene prediction in Drosophila melanogaster using machine learning approaches based on sequence and functional features. *Computational and Structural Biotechnology Journal*, *18*, 612–621.

Aromolaran, O., Oyelade, J., & Adebiyi, E. (2021). Performance evaluation of features for gene essentiality prediction. *IOP Conference Series: Earth and Environmental Science*, *655*(1), 1–14.

Asghar, M. Z., Subhan, F., Ahmad, H., Khan, W. Z., Hakak, S., Gadekallu, T. R., & Alazab, M. (2021). Senti-eSystem: A sentiment-based eSystem-using hybridized fuzzy and deep neural network for measuring customer satisfaction. *Software - Practice and Experience*, *51*(3), 571–594.

Batista, G. E. A. P. A., Prati, R. C., & Monard, M. C. (2004). A study of the behavior of several methods for balancing machine learning training data. *ACM SIGKDD Explorations Newsletter*, *6*(1), 20–29.

Beder, T., Aromolaran, O., Dönitz, J., Tapanelli, S., Adedeji, E. O., Adebiyi, E., Bucher, G., & Koenig, R. (2021). Identifying essential genes across eukaryotes by machine learning. *NAR Genomics and Bioinformatics*, *3*(4), 1–13.

Bier, E. (2005). Drosophila, the golden bug, emerges as a tool for human genetics. In *Nature Reviews Genetics* (Vol. 6, Issue 1, pp. 9–23).

Chang, J., Wang, R., Yu, K., Zhang, T., Chen, X., Liu, Y., Shi, R., Wang, X., Xia, Q., & Ma, S. (2020). Genome-wide CRISPR screening reveals genes essential for cell viability and resistance to abiotic and biotic stresses in Bombyx mori. *Genome Research*, *30*(5), 757–767.

Chen, S., Zhang, Y. E., & Long, M. (2010). New genes in Drosophila quickly become essential. *Science*, *330*(6011), 1682–1685.

Dotan, E., Jaschek, G., Pupko, T., Belinkov, Y., Henry, T., & Taub, M. (2024). Effect of Tokenization on Transformers for Biological Sequences. *Bioinformatics*, *40*(4), 1–15.

Edelsparre, A. H., Fitzpatrick, M. J., Rodríguez, M. A., & Sokolowski, M. B. (2021). Tracking dispersal across a patchy landscape reveals a dynamic interaction between genotype and habitat structure. *Oikos*, *130*(1), 79–94.

Fawcett, T. (2006). An introduction to ROC analysis. *Pattern Recognition Letters*, *27*(8), 861–874.

Jiang, R., Zhang, X., & Zhang Editors, M. Q. (2013). Basics of Bioinformatics. *Tsinghua University Press*, *1*, 1–17.

Karsito, & Susanti, S. (2019). KLASIFIKASI KELAYAKAN PESERTA PENGAJUAN KREDIT RUMAH DENGAN ALGORITMA NAÏVE BAYES DI PERUMAHAN AZZURA RESIDENCIA. *SIGMA*, *9*(3), 43–48.

Lopez-del Rio, A., Martin, M., Perera-Lluna, A., & Saidi, R. (2020). Effect of sequence padding on the performance of deep learning models in archaeal protein functional prediction. *Scientific Reports*, *10*(1), 1–15.

Luo, H., Lin, Y., Liu, T., Lai, F. L., Zhang, C. T., Gao, F., & Zhang, R. (2021). DEG 15, an update of the Database of Essential Genes that includes built-in analysis tools. *Nucleic Acids Research*, *49*(D1), D677–D686.

Mekruksavanich, S., & Jitpattanakul, A. (2023). A Hybrid Convolution Neural Network with Channel Attention Mechanism for Sensor-Based Human Activity Recognition. *Scientific Reports*, *13*(1), 1–12.

Ofer, D., Brandes, N., & Linial, M. (2021). The language of proteins: NLP, machine learning & protein sequences. In *Computational and Structural Biotechnology Journal* (Vol. 19, pp. 1750–1758). Elsevier B.V.

O'Grady, P. M., & DeSalle, R. (2018). Phylogeny of the genus Drosophila. *Genetics*, *209*(1), 1–25.

Olamilekan Adesola, R., Taiwo Lawal, J., & Oladele, O. E. (2021). Drosophila melanogaster (Meigen, 1830): A Potential Model for Human Diseases. *World News of Natural Sciences*, *36*, 42–59.

Pandey, U. B., & Nichols, C. D. (2011). Human disease models in drosophila melanogaster and the role of the fly in therapeutic drug discovery. *Pharmacological Reviews*, *63*(2), 411–436.

Rahbari, M., Rahlfs, S., Jortzik, E., Bogeski, I., & Becker, K. (2017). H2O2 dynamics in the malaria parasite Plasmodium falciparum. *PLoS ONE*, *12*(3), 1–13.

Rout, R. K., Umer, S., Khandelwal, M., Pati, S., Mallik, S., Balabantaray, B. K., & Qin, H. (2023). Identification of discriminant features from stationary pattern of nucleotide bases and their application to essential gene classification. *Frontiers in Genetics*, *14*, 1–12.

Sutanto, V. M., Sukma, Z. I., & Afiahayati, A. (2020). Predicting Secondary Structure of Protein Using Hybrid of Convolutional Neural Network and Support Vector Machine. *International Journal of Intelligent Engineering and Systems*, *14*(1), 232–243.

Tan, K. L., Lee, C. P., & Lim, K. M. (2023). RoBERTa-GRU: A Hybrid Deep Learning Model for Enhanced Sentiment Analysis. *Applied Sciences (Switzerland)*, *13*(6), 1–16.

Tawab, M., Khan, A., Adilina, S., Shadab, S., Neezi, N. A., & Shatabda, S. (2019). DeepDBP: Deep Neural Networks for Identification of DNA-binding Proteins. *Informatics in Medicine Unlocked*, *19*, 1–7.

Toraman, C., Yilmaz, E. H., Şahinuç, F., & Ozcelik, O. (2023). Impact of Tokenization on Language Models: An Analysis for Turkish. *ACM Journals*, *22*(4), 1–12.

Xie, J. (2020). A Novel Method of Music Generation Based on Three Different Recurrent Neural Networks. *Journal of Physics: Conference Series*, *1549*(4), 1–8.

Yang, C. H., Chen, B. H., Wu, C. H., Chen, K. C., & Chuang, L. Y. (2022). Deep Learning for Forecasting Electricity Demand in Taiwan. *Mathematics*, *10*(14), 1–19.

Yu, Y., Yang, L., Liu, Z., & Zhu, C. (2017). Gene essentiality prediction based on fractal features and machine learning. *Molecular BioSystems*, *13*(3), 577–584.