KLASIFIKASI DNA-*BINDING* PROTEIN MENGGUNAKAN METODE *BIDIRECTIONAL GATED RECURRENT UNIT* (BIGRU)

(Skripsi)

Oleh

JIHAN CAHYA FATIMAH

1917051016

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG
2023

ABSTRAK

KLASIFIKASI DNA-*BINDING* PROTEIN MENGGUNAKAN METODE *BIDIRECTIONAL GATED RECURRENT UNIT* (BIGRU)

Oleh

Jihan Cahya Fatimah

Protein pengikat DNA merupakan protein yang dapat mengikat DNA dan berinteraksi dengan DNA dalam membantu memodulasi fungsi DNA. Mengidentifikasi protein pengikat DNA menjadi salah satu fokus penelitian di bioinformatika karena signifikansinya dalam kegiatan biologis di dalam sel seperti membantu transkripsi sintesis protein, replikasi dan rekombinasi DNA. Protein pengikat DNA biasanya diidentifikasi dengan teknik eksperimental. Namun saat ini peneliti mengidentifikasi protein pengikat DNA menggunakan pembelajaran mesin. Tujuan penelitian ini untuk membuat sebuah model pembelajaran mesin yang digunakan untuk mengklasifikasi protein pengikat DNA. Metode yang digunakan yaitu metode *Bidirectional Gated Recurrent Unit* (BiGRU). Terdapat dua skenario arsitektur percobaan yaitu BiGRU *single layer* dan BiGRU *multi layer*, serta tiga skenario pembagian data yaitu 90% *training* 10% validasi, 80% *training* 20% validasi dan 70% *training* 30% validasi. Hasil penelitian yang didapatkan akan dianalisis nilai akurasi, nilai sensitivitas, nilai spesifisitas, dan nilai *Matthew Correlation Coefficient* untuk mengukur kinerja model yang telah dibuat. Data yang digunakan terdiri dari dua kelas yaitu protein pengikat DNA dan non protein pengikat DNA. Data tersebut diperoleh dari *Protein Data Bank* (PDB) yaitu dataset PDB1075 yang digunakan sebagai data *training* dan dataset PDB186 yang digunakan sebagai data *testing.* Setelah beberapa skenario percobaan dilakukan, didapatkan hasil tertinggi pada arsitektur BiGRU *single layer* yang mendapatkan hasil akurasi 81,72%, sensitivitas 90,32%, spesifisitas 73,11%, MCC 64,40%. Hasil penelitian menunjukkan bahwa metode BiGRU mampu mengklasifikasi protein pengikat DNA.

Kata Kunci : Untaian Protein, Protein Pengikat DNA, Klasifikasi, BiGRU.

ABSTRACT

CLASSIFICATION OF DNA-BINDING PROTEINS USING BIDIRECTIONAL GATED RECURRENT UNIT (BIGRU)

By

Jihan Cahya Fatimah

DNA-binding proteins are proteins that binds to DNA and interact with DNA to support modulate DNA function. Identifying DNA-binding proteins is one of the focuses research in bioinformatics because of the significant role of DNA-binding protein in biological activities cells such as transcription proses, DNA replication and recombination. DNA-binding proteins are usually identified by experimental techniques. However in recent years researchers are identifying DNA-binding proteins using machine learning. Purpose of this research is to create a machine learning model for classifying DNA-binding proteins. The used method is Bidirectional Gated Recurrent Unit (BiGRU). There are two architecture scenarios, BiGRU single layer and BiGRU multi layer, and then three data separation scenarios, 90% training `10% validation, 80% training 20% validation and 70% training 30% validation. The results of the research will be analyzed for accuracy score, sensitivity score, specificity score, and Matthew Correlation Coefficient score to measure the performance of the model. Dataset consists of two classes, DNA-binding proteins and non-DNA-binding proteins. Dataset is retrieved from the Protein Data Bank (PDB), which is a PDB1075 dataset used as training data and a PDB186 dataset used as testing data. After several experimental scenarios, the highest result is on the single layer BiGRU, the result are 81,72% accuracy, 90,32% sensitivity, 73,11% specificity, 64,40% MCC. The results showed that BiGRU was able to classify DNA-binding proteins.

Keywords: Protein Sequences, DNA-Binding Proteins, Classification, BiGRU.

KLASIFIKASI DNA-*BINDING* PROTEIN MENGGUNAKAN METODE *BIDIRECTIONAL GATED RECURRENT UNIT* (BIGRU)

Oleh

JIHAN CAHYA FATIMAH

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
2023

Judul Skripsi	: **KLASIFIKASI DNA-*BINDING* PROTEIN MENGGUNAKAN METODE *BIDIRECTIONAL GATED RECURRENT UNIT* (BIGRU)**
Nama Mahasiswa	: Jihan Cahya Fatimah
Nomor Pokok Mahasiswa	: 1917051016
Jurusan	: Ilmu Komputer
Fakultas	: Matematika dan Ilmu Pengetahuan Alam

MENYETUJUI

1. Komisi Pembimbing

Favorisen R. Lumbanraja, Ph.D.
NIP 19830110 200812 1002

2. Ketua Jurusan Ilmu Komputer

Didik Kurniawan, S.Si.,M.T.
NIP 19800419 200501 1004

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 19711001 200501 1002

Tanggal Lulus Ujian Skripsi : **21 November 2023**

PERNYATAAN

Saya yang bertanda tangan di bawah ini :

Nama : Jihan Cahya Fatimah

NPM : 1917051016

Dengan ini menyatakan bahwa skripsi saya yang berjudul "KLASIFIKASI DNA-*BINDING* PROTEIN MENGGUNAKAN METODE *BIDIRECTIONAL GATED RECURRENT UNIT* (BIGRU) adalah benar karya sendiri dan bukan orang lain. Seluruh tulisan yang tertuang dalam skripsi ini telah mengikuti kaidah penulisan karya ilmiah Universitas Lampung. Jika di kemudian hari terbukti skripsi saya adalah hasil penjiplakan atau dibuat orang lain, maka saya bersedia menerima sanksi sesuai dengan peraturan yang berlaku.

Bandar Lampung, 5 Desember 2023

Penulis

Jihan Cahya Fatimah

NPM 1917051016

RIWAYAT HIDUP

Penulis dilahirkan di Desa Wonokarto, Kecamatan Gading Rejo, Kabupaten Pringsewu pada tanggal 28 Januari 2001 sebagai anak kedua dari dua bersaudara dari pasangan Bapak Faizal Muhtar dan Ibu Siti Makmuroh. Penulis memulai menempuh pendidikan pertama di TK Aisiyah Wonokarto, kemudian melanjutkan pendidikan dasar di SDN 7 Gading Rejo dan menyelesaikan pendidikan pada tahun 2013. Kemudian penulis melanjutkan pendidikan di SMPN 1 Gading Rejo dan dapat diselesaikan pada tahun 2016 serta langsung melanjutkan pendidikan ke SMAN 1 Gading Rejo hingga selesai pada tahun 2019.

Pada tahun 2019, penulis terdaftar sebagai mahasiswa jurusan Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung melalui jalur SNMPTN. Selama menjadi mahasiswa, penulis beberapa kali terlibat dalam kegiatan sebagai berikut.

1. Menjadi anggota Printer Himpunan Mahasiswa Jurusan Ilmu Komputer periode 2019/2020.
2. Melaksanakan Karya Wisata Ilmiah (KWI) di Lampung Timur.
3. Menjadi Koordinator Asisten Dosen Jurusan Ilmu Komputer untuk mata kuliah Struktur Data dan Algoritma pada periode semeter ganjil tahun ajaran 2021/2022.
4. Menjadi Koordinator Asisten Dosen Jurusan Ilmu Komputer untuk mata kuliah Basis Data pada periode semester genap tahun ajaran 2021/2022.
5. Menjadi Asisten Dosen Jurusan Ilmu Komputer untuk mata kuliah Bioinformatika pada periode semester genap tahun ajaran 2023/2024.
6. Melaksanakan Kerja Praktik di Badan Pusat Statistik Kabupaten Pringsewu pada tahun 2022.
7. Melaksanakan Kuliah Kerja Nyata (KKN) pada tahun 2022 di Desa Banding Agung,Kecamatan Talang Padang, Kabupaten Tanggamus.

MOTO

Hanya ada dua pilihan untuk memenangkan kehidupan: keberanian, atau keikhlasan. Jika tidak berani, ikhlaslah menerimannya. jika tidak ikhlas, beranilah mengubahnya.

Lenang Manggala

"Boleh jadi kamu membenci sesuatu, padahal dia amat baik bagimu, dan boleh jadi (pula) kamu menyukai sesuatu, padahal dia amat buruk bagimu; Allah mengetahui, sedang kamu tidak mengetahui."

(QS. Al Baqarah: 216)

Jangan takut jatuh, karena yang tidak pernah memanjatlah yang tidak pernah jatuh. Jangan takut gagal, karena yang tidak pernah gagal hanyalah orang-orang yang tidak pernah melangkah. Jangan takut salah, karena dengan kesalahan yang pertama kita dapat menambah pengetahuan untuk mencari jalan yang benar pada langkah yang kedua.”

Hamka

PERSEMBAHAN

Alhamdulillahirobbilalamin

Puji dan syukur atas kehadirat Allah SWT atas segala Rahmat dan Karunia-Nya serta tidak lupa shalawat teriring salam selalu tersampaikan kepada Nabi Muhammad SAW sehingga saya dapat menyelesaikan skripsi ini.

Kupersembahkan karya ini kepada :

Papa dan Mama tercinta yang selalu mendoakan, mendukung, dan memberikan segala yang terbaik untukku. Terima kasih atas kasih sayang, perjuangan, pengorbanan, dan kesabaran yang telah diberikan selama ini, selalu memberikan semangat di setiap langkah yang kujalani hingga saat ini dalam mendidik dan membesarkanku. Teruntuk kakak-kakak serta keponakan tersayangku, keluarga besarku, terima kasih selalu memberikan semangat dan motivasi kepadaku.

Sahabat-sahabatku tersayang yang selalu menemani, mendukung, dan memberikan kenangan yang indah dalam hidupku.

Keluarga Besar Ilmu Komputer 2019

Almamater Tercinta Universitas Lampung

SANWACANA

Puji syukur kehadirat Allah SWT atas rahmat, berkah, dan karunia-Nya, shalawat serta salam semoga senantiasa tercurahkan kepada junjungan kita Nabi Muhammad SAW, sehingga penulis dapat menyelesaikan skripsi yang berjudul "Klasifikasi DNA-*binding* Protein Menggunakan Metode *Bidirectional Gates Recurrent Unit* (BiGRU) dengan baik.

Terima kasih penulis ucapkan kepada semua pihak yang telah membantu dan berperan penting dalam menyusun skripsi ini, antara lain.

1. Allah SWT yang telah memberikan rahmat, berkah, dan karunia-Nya.
2. Kedua orang tua serta kakak tersayang yang selalu memberikan doa, semangat, motivasi, dukungan, dan kasih sayang yang tak terhingga. Semoga Allah SWT memberikan keberkahan, kesehatan, dan kebahagiaan dalam kehidupan di dunia maupun di akhirat.
3. Bapak Favorisen R. Lumbanraja, Ph.D sebagai pembimbing utama yang selalu membimbing dengan sabar, memberikan arahan, serta saran kepada penulis sehingga dapat menyelesaikan skripsi dengan baik.
4. Ibu Fatma Indriani, S.T., MIT., Ph.D sebagai pembahas yang telah memberikan kritik dan masukan yang sangat membantu dalam memperbaiki skripsi sehingga skripsi dapat diselesaikan dengan baik.
5. Bapak Dr. rer. nat. Akmal Junaidi, M.Sc sebagai pembahas yang telah memberikan kritik dan masukan yang sangat membantu dalam memperbaiki skripsi sehingga skripsi dapat diselesaikan dengan baik.
6. Ibu Yunda Heningtyas, M.Kom sebagai pembimbing akademik yang telah membimbing penulis selama menjalani perkuliahan di Jurusan Ilmu Komputer.

7. Bapak Dr. Eng. Heri Satria, S.Si., M.Si., selaku Dekan FMIPA Universitas Lampung.
8. Bapak Didik Kurniawan, S.Si., M.T., selaku Ketua Jurusan Ilmu Komputer Universitas Lampung.
9. Bapak dan Ibu Dosen Jurusan Ilmu Komputer yang telah memberikan ilmu yang bermanfaat selama menjalani perkuliahan di Jurusan Ilmu Komputer.
10. Ibu Ade Nora Maela, Bang Zainuddin, dan Mas Nofal yang telah membantu segala urusan izin dan administrasi yang diperlukan penulis di Jurusan Ilmu Komputer.
11. Sahabat tersayang, Azahra dan Dina, yang selalu ada disaat suka maupun duka, tempat berkeluh kesah, bercerita, saling membantu, dan memberikan banyak kenangan indah selama di perkuliahan.
12. Rekan seperbimbingan, Ardella dan Fajar, yang saling menyemangati dan membantu satu sama lain, tempat berdiskusi dan bertukar pikiran.
13. Keluarga Ilmu Komputer 2019 yang tidak bisa penulis sebut satu persatu yang telah menemani melewati masa perkuliahan yang penuh warna.
14. Semua pihak yang telah berpartisipasi baik secara langsung maupun tidak langsung dalam membantu penyusunan skripsi.

Bandar Lampung, 5 Desember 2023

Jihan Cahya Fatimah
NPM. 1917051016

DAFTAR ISI

Halaman

DAFTAR TABEL

Tabel Halaman

DAFTAR GAMBAR

Gambar Halaman

DAFTAR KODE PROGRAM

Kode Program Halaman

I. PENDAHULUAN

1.1 Latar Belakang

Protein pengikat DNA merupakan protein yang dapat mengikat DNA dan berinteraksi dengan DNA dalam membantu memodulasi fungsi DNA (Rahman, et al., 2018). Protein pengikat DNA berperan penting dalam komposisi struktural DNA dan regulasi gen. Selain itu protein pengikat DNA berperan membantu dalam proses transkripsi sintesis protein, replikasi DNA, rekombinasi, modifikasi, dan perbaikan DNA dalam sel makhluk hidup (Sang, et al., 2020). Dengan peran penting protein pengikat DNA dalam proses regulasi dalam makhluk hidup, mengidentifikasi protein pengikat DNA menjadi salah satu fokus penelitian di bioinformatika karena signifikansinya dalam kegiatan biologis di dalam sel.

Mengidentifikasi protein pengikat DNA sangat penting untuk memahami mekanisme aktivitas biologis yang terjadi di dalam sel. Memahami bagaimana interaksi antara protein dan DNA akan meningkatkan pemahaman kita tentang gen sehingga didapatkan gambaran lengkap tentang interaksi keduanya yang memungkinkan kita memahami karakterisasi gen terhadap lingkungan yang berubah secara dinamis (Jingna, Rui dan Rongling, 2015). Dalam satu dekade terakhir, berbagai metode komputasi telah dikembangkan untuk memprediksi urutan protein sehingga semakin banyak protein yang telah teridentifikasi. Menurut The Universal Protein Resource Knowledgebase (Uniprot), repositori urutan protein terus meningkat dan mengalami pertumbuhan yang eksplosif. Hal tersebut menyebabkan pengklasifikasian urutan protein sulit dilakukan karena semakin beragamnya urutan protein. Oleh karena itu, mengklasifikasi urutan protein terus

dilakukan dengan berbagai metode komputasi agar dapat mengelompokkan urutan protein secara tepat.

Protein pengikat DNA biasanya diidentifikasi dengan teknik eksperimental, seperti uji pengikatan filter, analisis genetik, kristalografi Xray, analisis ChIP, dan magnetik nuklir resonansi (NMR) (W.Lou, et al., 2014). Namun, teknik eksperimental sangat memakan waktu dan tenaga sehingga berbagai penelitian dilakukan menggunakan algoritma *machine* learning dalam beberapa tahun terakhir dalam mengidentifikasi protein pengikat DNA. Salah satu di antaranya yaitu penelitian Zaman, et al., (2017) bertujuan untuk memprediksi protein pengikat DNA dengan menggunakan metode *Support Vector Machine.* Selain itu, penelitian oleh Shadab, et al., (2020) yang bertujuan untuk mengidentifikasi protein pengikat DNA dengan menggunakan metode yaitu *Convolutional Neural Network* dan *Artificial Neural Network.* Serta penelitian yang dilakukan oleh Wei, Tang dan Zou, (2017) bertujuan mengklasifikasi protein pengikat DNA dengan menggunakan metode *Random Forest*.

Pada penelitian ini mengusulkan menggunakan salah satu metode *Recurrent Neural Network* yaitu *Bidirectional Gated Recurrent Unit* dalam mengklasifikasi protein pengikat DNA. *Bidirectional Gated Recurrent Unit* (BiGRU) adalah salah satu metode *Recurrent Neural Networks* (RNN) untuk memproses data yang berurutan (data berbentuk sekuensial). Metode ini dinilai sesuai dengan bentuk data protein *sequence* yang terdiri dari variasi urutan asam amino yang tersusun secara terurut. Beberapa penelitian menggunakan metode serupa untuk memprediksi data sekuensial dan menghasilkan prediksi yang baik. Salah satunya penelitian yang dilakukan Shen, Bao dan Huang, (2018) telah berhasil memprediksi situs pengikatan faktor transkipsi menggunakan metode RNN dengan akurasi sebesar 96,20%. Selain itu, penelitian tentang prediksi *enhancers* oleh Yang, et al., (2017) berhasil memprediksi *enchancers* dan mendapatkan akurasi sebesar 95,60%.

1.2 Rumusan Masalah

Rumusan masalah pada penelitian ini sebagai berikut :

1. Apakah metode *Bidirectional Gated Recurrent Unit* (BiGRU) dapat diimplementasikan untuk membuat model klasifikasi DNA-*binding* protein?
2. Berapa hasil evaluasi kinerja yang didapatkan dari metode *Bidirectional Gated Recurrent Unit* (BiGRU) dalam mengklasifikasikan DNA-*binding* protein?
3. Apakah hasil yang didapatkan pada penelitian ini mampu mencapai kinerja yang lebih baik dari penelitian terdahulu?

1.3 Batasan Masalah

Batasan masalah pada penelitian ini sebagai berikut :

1. Penelitian ini menggunakan dataset *benchmark* PDB1075 yang berjumlah 1075 data yang digunakan dalam penelitian Liu, et al., (2014) serta dataset *independent* yang berjumlah 186 data yang digunakan dalam penelitian Lou, et al., (2014).
2. Penelitian ini menggunakan metode *Bidirectional Gated Recurrent Unit* (BiGRU) dalam mengklasifikasi DNA-*binding* protein.
3. Hasil klasifikasi hanya terdiri dari dua kelas yaitu DNA-*binding* protein dan non-DNA-*binding* protein.

1.4 Tujuan Penelitian

Adapun tujuan yang hendak dicapai dari penelitian ini sebagai berikut :

1. Mengevaluasi kinerja metode *Bidirectional Gated Recurrent Unit* (BiGRU) dalam mengklasifikasi DNA-*binding* protein.
2. Membandingkan hasil yang diperoleh pada penelitian ini dengan hasil penelitian terdahulu yang menggunakan sumber data yang sama.

1.5 Manfaat Penelitian

Manfaat yang didapat dari penelitian ini sebagai berikut :

1. Menambah pengetahuan mengenai cara kerja algoritma *Bidirectional Gated Recurrent Unit* (BiGRU).
2. Penelitian ini dapat dijadikan informasi untuk penelitian mengenai klasifikasi DNA-*binding* protein.

II. TINJAUAN PUSTAKA

2.1 Penelitian Terdahulu

Penelitian terdahulu yaitu penelitian yang telah dilakukan peneliti lain yang membahas tentang klasifikasi DNA-*binding* protein. Penelitian terdahulu digunakan untuk membandingkan hasil yang didapatkan pada penelitian ini. Adapun beberapa penelitian terdahulu terkait klasifikasi protein pengikat DNA dapat dilihat pada Tabel 1.

Tabel 1. Tabel penelitian terdahulu terkait prediksi protein pengikat DNA

No	Penelitian	Data	Metode	Hasil
1	HMMBinder: *DNA-Binding Protein Prediction Using HMM Profile Based Features* (Zaman, et al., 2017).	Dataset PDB1075 Positif : 525 Negatif : 550 Dan Dataset PDB186 Positif : 93 Negatif : 93	*Support Vector Mechine Using HMM Profile Based Features*	Akurasi : 69,02% Sensitifitas : 61,53% Spesifisitas : 76,34% MCC : 39,40%
2	DeepDBP: *Deep neural networks for identification of DNA-binding proteins* (Shadab, et al., 2020).		*Convolutional Neural Network*	Akurasi : 84,31% Sensitifitas : 83,00% Spesifisitas : 75,00% MCC : 98,10%
3	*DNA-binding protein prediction based on deep transfer learning* (Jun, et al., 2022)		*LSTM and Convolutional Neural Network*	Akurasi : 78,00% Sensitifitas : 71,00% Spesifisitas : 90,00% MCC : 58,00%
4	*Local-DPP: An Improved DNA-binding Protein Prediction Method by Exploring Local Evolutionary Information (Wei, L., Tang, J., & Zou, Q., 2017).*		*Random Forest*	Akurasi : 79,00% Sensitivitas : 92,00% Spesifisitas : 65,60% MCC : 63,00%

Berdasarkan Tabel 1, penjabaran mengenai masing-masing penelitian terdahulu yang digunakan sebagai berikut.

2.1.1 HMMBinder: *DNA-Binding Protein Prediction Using HMM Profile Based Features*

Penelitian Zaman, et al., (2017) bertujuan untuk memprediksi protein pengikat DNA dengan menggunakan fitur monogram dan bigram yang diekstrak dari profil HMM urutan protein. Metode prediksi ini dinamakan HMMBinder. Penelitian ini menggunakan *Support Vector Mechine* sebagai teknik untuk mengklasifikasikan protein pengikat DNA. Dataset yang digunakan yaitu Dataset *Benchmark* PDB1075 yang terdiri 1075 protein *sequence* dengan 525 DNA-*binding proteins (positive samples)* dan 550 non-DNA-*binding proteins (negative samples).* Hasil penelitian ini yaitu metode HMMBinder dapat memprediksi protein pengikat DNA dengan akurasi 69,02%, sensitivitas 61,53%, spesifisitas 76,34%, dan MCC 39,40%.

2.1.2 DeepDBP: Deep neural networks for identification of DNA-binding proteins

Penelitian Shadab, et al., (2020) bertujuan untuk mengidentifikasi protein pengikat DNA dengan menggunakan salah satu metode *Deep Neural Network* yaitu *Convolutional Neural Network.* Metode prediksi ini dinamakan DeepDBP-CNN. Dataset yang digunakan yaitu Dataset *Benchmark* PDB1075 yang terdiri 1075 protein *sequence* dengan 525 DNA-*binding proteins (positive samples)* dan 550 non-DNA-*binding proteins (negative samples).* Hasil penelitian ini yaitu metode DeepDBP-CNN dapat memprediksi protein pengikat DNA dengan akurasi 84,31%, sensitivitas 83,00%, spesifisitas 75,00%, dan MCC 98,10%.

2.1.3 DNA-binding protein prediction based on deep transfer learning

Penelitian Jun, et al., (2022) bertujuan untuk memprediksi protein pengikat DNA dengan menggunakan konjungsi antara *Deep Learning*

dan *Transfer Learning.* Algoritma *Transfer Learning* digunakan untuk mengekstrak data sedangkan data di *training* menggunakan metode LSTM dan *Convolutional Neural Network.* Dataset yang digunakan yaitu Dataset *Benchmark* PDB1075 yang terdiri 1075 protein *sequence* dengan 525 DNA-*binding proteins (positive samples)* dan 550 non-DNA-*binding proteins (negative samples).* Hasil penelitian ini yaitu metode *deep transfer learning* dapat memprediksi protein pengikat DNA dengan akurasi 78,00%, sensitivitas 71,00%, spesifisitas 90,00%, dan MCC 58,00%.

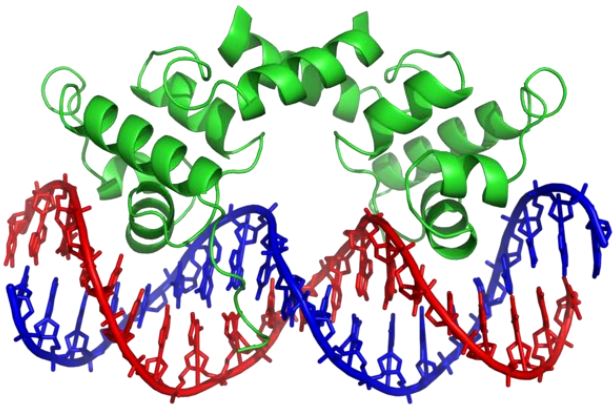
2.1.4 Local-DPP: An Improved DNA-binding Protein Prediction Method by Exploring Local Evolutionary Information

Penelitian Wei, Tang, dan Zou, (2017) bertujuan untuk memprediksi DNA-*binding* protein dengan menggunakan metode *random forest.* Dataset yang digunakan yaitu Dataset *Benchmark* PDB1075 yang terdiri 1075 protein *sequence* dengan 525 DNA-*binding proteins (positive samples)* dan 550 non-DNA-*binding proteins (negative samples).* Hasil penelitian yang didapatkan yaitu metode *random forest* dapat memprediksi protein pengikat DNA dengan akurasi 79,00%, sensitivitas 92,00%, spesifisitas 65,60%, dan MCC 63,00%.

2.2 Protein Pengikat DNA (DNA-*binding* Protein)

Protein pengikat DNA merupakan protein yang dapat mengikat DNA dan berinteraksi dengan DNA dalam membantu memodulasi fungsi DNA (Rahman, et al., 2018). DNA-*binding* protein termasuk ke dalam kategori protein spesifik yang artinya protein yang hanya dapat berinteraksi dengan untaian DNA tertentu dengan motif tertentu. DNA-*binding* protein mengandung gugus fungsi yang dapat mengidentifikasi pasangan basa dan memungkinkan berinteraksi dengan alur utama DNA. Berdasarkan fungsinya, protein pengikat DNA dibagi menjadi empat yaitu faktor transkripsi (*trancription factors*), faktor replikasi DNA (DNA *replications factors*), faktor perbaikan DNA (DNA *repair factors*), dan histon. Faktor transkripsi

terlibat dalam regulasi transkripsional, seperti aktivator transkripsi. Faktor replikasi DNA bertugas melakukan sintesis pada seluruh genom atau fragmen DNA. Faktor perbaikan berperan dalam menghilangkan pasangan basa tunggal atau oligonukleotida tertentu dan mengisi celah tersebut dengan nukleotida yang sesuai. Histon terlibat dalam transkripsi dan pengemasan kromosom inti sel. Interaksi antara protein pengikat DNA dengan DNA target dapat dilihat pada Gambar 1.



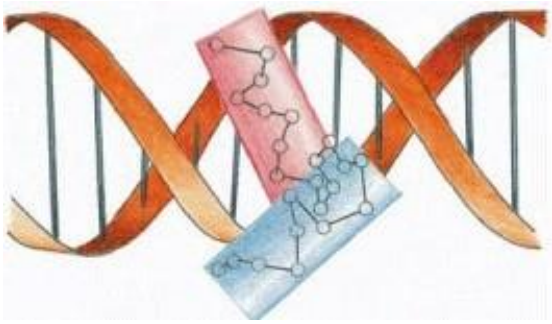
Gambar 1. Interaksi Protein dengan DNA (Jen dan Travers, 2013).

DNA-*binding* protein mengandung suatu domain pengikat DNA yang melakukan interaksi antara DNA dan protein. Interaksi antara DNA dan protein yang kompleks menjalankan fungsi berbeda berdasarkan jenis kompleks yang terbentuk antara fragmen DNA dan protein berbeda. Interaksi antara protein dan DNA inilah yang akan memulai serangkaian transisi biokimia yang mengatur semua fungsi biologis utama dalam sel hidup. Beberapa dari jenis protein pengikat DNA terlibat dalam replikasi dan rekombinasi DNA, sementara yang lain terlibat dalam transkripsi dan perbaikan DNA. Adapun pengertian dari proses replikasi DNA adalah proses sintesis untai DNA baru dengan menggunakan untai DNA lama sebagai cetakan. Berbeda dengan faktor transkripsi, protein ini mempunyai ikatan DNA non-urutan spesifik (*non-sequence-specific*). Jenis protein ini tidak memerlukan seperangkat nukleotida tertentu untuk menjalankan fungsinya tetapi bekerja berdasarkan struktur DNA. Contohnya termasuk enzim DNA polimerase. Faktor transkripsi adalah kelompok DNA-*binding* protein

terbesar. Jenis DNA-*binding* protein ini mengontrol laju transkripsi dengan mengikat urutan DNA tertentu. Contohnya termasuk protein aktivator atau represor yang memandu RNA polimerase ke gen spesifik untuk aktivasi atau represinya. Faktor transkripsi menghidupkan dan mematikan gen dengan mengikat situs promotor, yang memungkinkan RNA polimerase memulai transkripsi atau mencegahnya menjalankan fungsinya. Perbaikan DNA adalah proses menghilangkan pasangan basa atau untaian nukleotida yang rusak dari DNA dan kemudian mengisi celah tersebut dengan pasangan basa yang benar. Contohnya yaitu untuk memperbaiki mutasi yang terjadi pada genom organisme hidup. DNA-*binding* protein memiliki kemampuan untuk mengenali dan mengikat rangkaian DNA tertentu dengan menggunakan motif yang berbeda. Terdapat beberapa jenis motif DNA-*binding* protein yaitu, *α-helix-turn-α-helix, zinc-finger and loop, zipper motifs.*

A. *α-Helix-Turn-α-Helix* (HTH motif)

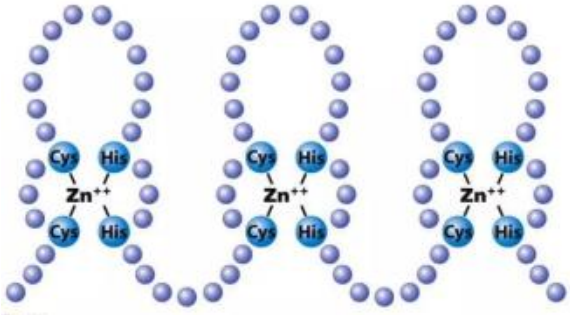
Motif HTH terdiri dari lebih kurang 20 residu asam amino dengan residu 1-7 membentuk *α-helix* pertama dan residu 12-20 membentuk *α-helix* kedua. Kedua *α-helix* ini dihubungkan oleh belokan pendek sehingga cenderung berhadapan satu sama lain sebesar 120 derajat. *α-helix* kedua dikenali sebagai pembuat sebagian besar kontak spesifik ke DNA dan terletak pada alur utama dupleks *B-form.* Mayoritas protein HTH memiliki sejumlah interaksi langsung antara heliks pengenalan dan alur utama DNA urutan operator, yang mempertahankan konformasi tipe-B. Ilustrasi protein pengikat DNA dengan motif HTH dapat dilihat pada Gambar 2.



Gambar 2. Protein Pengikat DNA dengan Motif HTH (Alberts, et al., 2002).

B. *zinc-finger and loop*

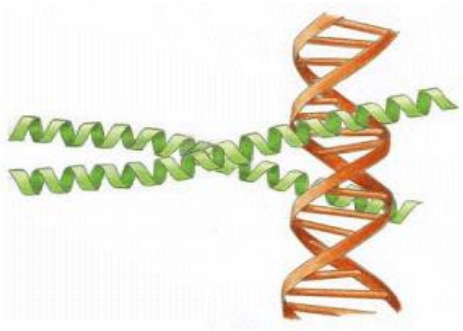
Zinc-finger merupakan kelompok terbesar dalam protein pengikat DNA yang memiliki keragaman yang cukup besar pada arsitekturnya. Protein *zinc-finger* memiliki setidaknya dua unit seperti jari. Setiap jari membuat kontak langsung ke untai kaya guanin, dengan pola interaksi rantai samping dari α-heliks yang melibatkan pengenalan arginin–guanin dan histidin–guanin yang sangat mirip dengan yang terlihat pada protein HTH (Neidle, 2008). Ilustrasi protein pengikat DNA dengan motif *zinc finger* dapat dilihat pada Gambar 3.



Gambar 3. Protein Pengikat DNA dengan motif *zinc finger* (Neidle, 2008).

C. *zipper motifs*

zipper motifs merupakan motif struktural pada banyak protein pengikat DNA dimana dua *helix α* protein terpisah dan bergabung bersama membentuk kumparan melingkar yang berbentuk seperti resleting, membentuk dimer protein. Ilustrasi protein pengikat DNA dengan motif *zipper motifs* dapat dilihat pada Gambar 4.



Gambar 4. Protein Pengikat DNA dengan Motif Zipper (Alberts, et al., 2002).

2.3 Protein *Sequence*

Protein merupakan sebuah rantai atau urutan dari gabungan 20 jenis asam amino yang berbeda, yang masing-masing jenis protein mempunyai jumlah dan ciri khas *sequence* yang berbeda (Handamari, et al., 2011). Protein *sequence* memiliki kompleksitas lima kali lebih besar daripada DNA *sequence* yang hanya memiliki 4 jenis nukleotida. Hal ini menjadikan protein sequence lebih banyak menampung informasi dan sensitif daripada DNA *sequence* (Hartono, Setyorini dan Karimah, 2021). Terdapat 20 jenis asam amino pada tubuh manusia yang dapat dilihat pada Tabel 2.

Tabel 2. Jenis-jenis Asam Amino

No	Asam Amino	Singkatan	Simbol	No	Asam Amino	Singkatan	Simbol
1	*Alanine*	Ala	A	11	*Leucine*	Leu	L
2	*Arginine*	Arg	R	12	*Lysine*	Lys	K
3	*Asparagine*	Asn	N	13	*Methionine*	Met	M
4	*Asparic acid*	Asp	D	14	*Phenylalanine*	Phe	F
5	*Cysteine*	Cys	C	15	*Proline*	Pro	P
6	*Glutamic acid*	Glu	E	16	*Serine*	Ser	S
7	*Glutamine*	Gln	Q	17	*Threonine*	Thr	T
8	*Glycine*	Gly	G	18	*Tryptophan*	Trp	W
9	*Histidine*	His	H	19	*Tyrosine*	Tyr	Y
10	*Isoleucine*	Ile	I	20	*Valine*	Val	V

2.4 Data *Preprocessing*

Data *preprocessing* adalah proses mengubah data mentah yang seringkali tidak lengkap, tidak konsisten, dan redundan menjadi format yang mudah dimengerti (Agarwal, 2015). Tujuan data *preprocessing* yaitu untuk mentransformasi dari data mentah menjadi format yang dapat dimengerti mesin untuk analisis selanjutnya. Sebelum mulai menganalisis data, data *preprocessing* merupakan tahapan pertama dan tahapan yang penting karena proses data *preprocessing* dapat meningkatkan kualitas data. Terdapat beberapa teknik pada data *preprocessing* yaitu data *cleaning*, data

integration, data *transformation*, dan data *reduction* (Han dan Kamber, 2006).

2.5 *Tokenization*

Tokenisasi merupakan proses membagi teks menjadi potongan-potongan yang lebih kecil yang disebut sebagai token (Mullen, et al., 2018). Token tersebut akan menjadi representasi vektor yang mewakili dokumen tersebut. Tujuan proses tokenisasi yaitu memecah data teks bahasa alami menjadi potongan-potongan informasi yang dapat dianggap sebagai elemen diskrit. Dengan kata lain tokenisasi mengubah string tidak terstruktur (dokumen teks) menjadi struktur data numerik yang cocok untuk pembelajaran mesin.

Tokenisasi dibagi menjadi tiga yaitu *word tokenization, character tokenization, subword tokenization* (Alkaoud dan Syed, 2020). *Word tokenization* merupakan proses tokenisasi dengan membagi sebuah kalimat menjadi beberapa kata berdasarkan pembatas kata seperti spasi atau tanda baca. Token kata yang dihasilkan dapat berbeda-beda tergantung pembatas kata yang diterapkan sebagai acuan. Berbeda dengan *word tokenization*, *subword tokenization* merupakan jenis tokenisasi yang menjadi solusi di antara dua jenis tokenisasi lainnya. Prinsip *subword tokenization* yaitu tidak membagi kata yang sering digunakan menjadi sub-kata yang lebih kecil dan pisahkan kata yang unik dan berbeda menjadi sub-kata yang lebih kecil. *Character tokenization* merupakan proses tokenisasi dengan membagi sebuah kalimat menjadi karakter individu. Jenis tokenisasi ini baik digunakan jika dalam suatu *sequence* memiliki banyak kata namun memiliki jumlah karakter yang tetap. Jenis tokenisasi ini cukup sederhana dan dapat mengurangi kompleksitas memori dan waktu secara signifikan. Berdasarkan penjabaran di atas, tipe tokenisasi yang sesuai untuk data protein pengikat DNA yaitu *character tokenization* karena dalam satu data protein terdapat 20 jenis asam amino yang berbeda. Jenis tokenisasi *character tokenization* dapat dilihat pada Tabel 3.

Tabel 3. *Character-based Tokenization*

	***Sequence* Protein Sebelum Tokenisasi**	**Sesudah Tokenisasi**
Character -based	SNLNVERVLSVH	‘S’: 1, ’N’ : 2, ’L’ : 3, ’V : 4’, ’E’ : 5, ’R’ : 6, ’H’ : 7
Hasil		[1 2 3 1 4 5 6 4 3 1 4 7]

2.6 *Padding*

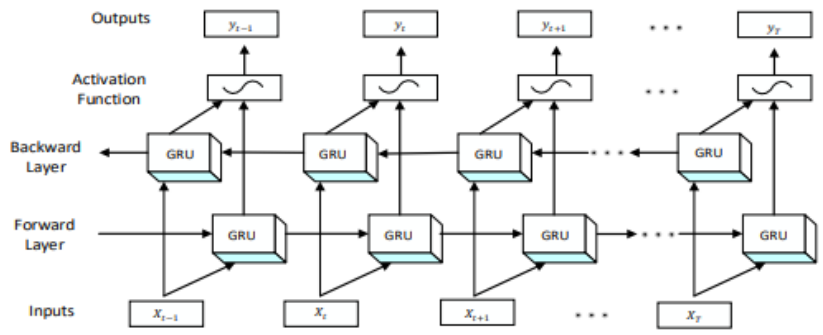
Padding merupakan proses selanjutnya yang harus dilakukan setelah proses tokenisasi selesai. Tujuan proses *padding* yaitu membuat setiap kalimat yang sebelumnya memiliki panjang yang berbeda-beda menjadi sama panjang (Rishita, et al., 2018). *Padding* dapat dilakukan dengan menambahkan angka 0 secara otomatis tepat di sebelum *sequence* maupun di sesudah *sequence*. Penambahan angka 0 sebelum *sequence* disebut dengan *pre-padding* sedangkan penambahan angka 0 sesudah *sequence* disebut dengan *post-padding.* Penggunaan *pre-padding* dan *post-padding* dapat mempengaruhi akurasi pada model (Reddy, 2019). *Pre-padding* dapat lebih optimal dalam meningkatkan akurasi dibandingkan dengan *post-padding* karena *pre-padding* memungkinkan model dapat mengingat memori *sequence* sebenarnya yang berada diakhir *sequence,* yang mana memori ini merupakan informasi yang penting dalam proses pembelajaran model. Sedangkan penggunaan *post-padding,* memori *sequence* sebenarnya berada di awal *sequence* sehingga kemungkinan model akan melupakan informasi yang seharusnya disimpan. *Pada* proses *padding* ini juga dapat diatur panjang maksimal dari masing-masing *sequence,* yang disebut sebagai parameter *max_length.* Misalkan parameter *max_length* diisi dengan nilai 5, maka panjang masing-masing kalimat secara otomatis tidak akan melebihi 5. Perbedaan antara *pre* dan *post padding* dapat dilihat pada Tabel 4.

Tabel 4. Jenis-jenis *Padding*

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

2.7 *Bidirectional Gated Recurrent Unit (BiGRU)*

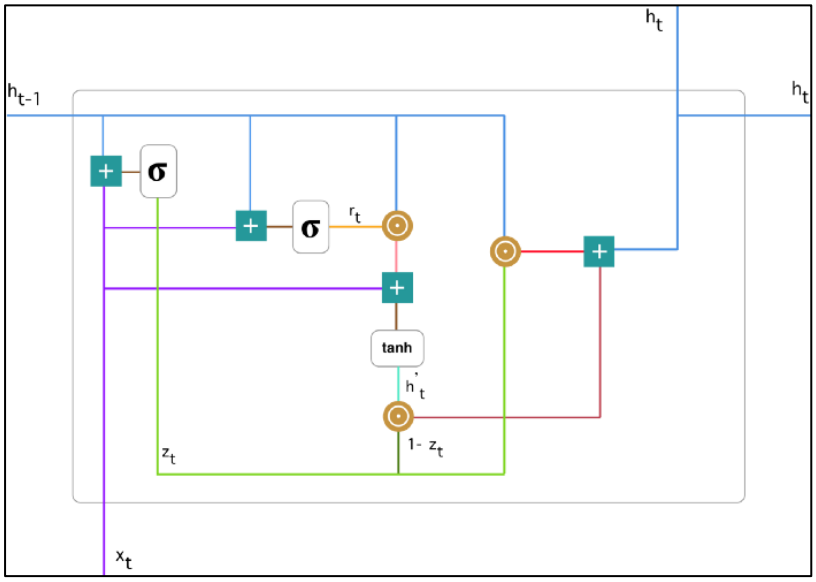
Bidirectional Gated Recurrent Unit (BiGRU) adalah salah satu metode *Recurrent Neural Networks* (RNN) untuk memproses data yang berurutan (data berbentuk sekuensial). *Bidirectional* GRU terdiri dari dua GRU yang ditumpangkan di atas satu sama lain. GRU merupakan versi perbaikan dari metode *Recurrent Neural Network* yang menggunakan gerbang reset (*reset gate*) dan gerbang pembaruan (*update gate*) untuk memutuskan cara membuang informasi yang tidak diperlukan dan memperbarui informasi untuk disimpan dalam memori (Le, Yapp dan Yeh, 2019). BiGRU menghubungkan dua lapisan tersembunyi dengan arah transmisi yang berlawanan ke lapisan output yang sama sehingga lapisan output dapat memperoleh informasi dari keadaan masa lalu dan masa depan. Ini berarti bahwa model BiGRU dapat memperoleh informasi dari dua arah yang berbeda, sehingga melatih model yang lebih akurat dan membuat prediksi lebih akurat (Cao, et al., 2021). Struktur *Bidirectional* GRU dapat dilihat pada Gambar 5.



Gambar 5. Struktur *Bidirectional* GRU (Ju, Zhang dan Zhu, 2019).

Dapat dilihat dari Gambar 5, *Bidirectional* GRU memiliki dua GRU yang tumpang tindih satu sama lain. Hal ini menyebabkan model dapat memperoleh informasi dari kedua arah. Untuk lebih spesifiknya, satu GRU mengalir ke arah depan dan menghitung *hidden state* depan. Sementara itu, yang lain mengalir ke arah belakang dan menghitung *hidden state* belakang. Oleh karena itu, setiap lapisan yang menggunakan BiGRU pada satu langkah waktu tertentu dapat menangkap informasi masa lalu dan masa depan secara bersama-sama (Deng, et al., 2019).

Metode ini merupakan salah satu metode khusus untuk memproses data yang berurutan (data berbentuk sekuensial) seperti pengenalan ucapan, pemodelan bahasa dan lain-lain (Amajd, 2017). Selain itu, metode ini memungkinkan untuk mengenali pola data dengan baik dengan menyimpan memori/ ingatan (*feedback loop*) yang kemudian menggunakannya untuk membuat prediksi akurat yang membantu memprediksi *input* selanjutnya. GRU berhasil memecahkan masalah ketergantungan jangka panjang yang dimiliki metode RNN biasa. Jika dibandingkan dengan arsitektur *Recurrent Neural Network* yang lain, GRU memiliki lebih sedikit parameter sehingga dari waktu komputasi GRU lebih cepat dan efisien. Ilustrasi alur kerja arsitektur GRU dapat dilihat pada Gambar 6.



Gambar 6. Cara Kerja Arsitektur GRU (Kostadinov, 2017).

Beberapa ketentuan masukan pada metode GRU yaitu mengikuti fungsi berikut.

a. Gerbang pembaruan *(update gate)* membantu model menentukan banyaknya informasi pada langkah sebelumnya yang perlu diteruskan. Fungsi perhitungan pada *update gate* dapat dilihat pada Persamaan (1).

$$z_t = \sigma(W_{iz}x_t + W_{hz}h_{(t-1)}) \quad \text{.....(1)}$$

b. Gerbang reset *(reset gate)* membantu model menentukan berapa banyak informasi pada langkah sebelumnya yang perlu dilupakan atau dihapus. Perhitungan secara matematik *(reset gate)* dapat dilihat pada Persamaan (2).

$$r_t = \sigma(W_{ir}x_t + W_{hr}h_{(t-1)}) \quad \text{..... (2)}$$

c. Konten memori yang sedang digunakan *(current memory)* menggunakan gerbang reset untuk menyimpan informasi yang relevan dari langkah sebelumnya. Perhitungan secara matematik *(current memory)* dapat dilihat pada Persamaan (3).

$$h'_t = tanh\left(W_{in}x_t + r_h \odot W_{hn}h_{(t-1)}\right) \quad \text{..... (3)}$$

d. Memori akhir pada langkah yang sedang digunakan *(final memory)* menghitung vektor HT yang menyimpan informasi untuk unit yang sedang digunakan dan meneruskannya ke jaringan dengan menggunakan gerbang pembaruan *(update gate).* Perhitungan secara matematik *(final memory)* dapat dilihat pada Persamaan (4).

$$h_t = (1 - z_t) \odot h'_t + z_t \odot h_{(t-1)} \quad \text{..... (4)}$$

Makna dari masing-masing simbol yang terdapat pada persamaan GRU dapat dilihat pada Tabel 5.

Tabel 5. Makna Simbol Pada Persamaan BiGRU

Simbol Persamaan	Makna
z_t	*Update gate*
W_{iz} , W_{ir} , W_{in}	*Weight input*
W_{hz} , W_{hr} , W_{hn}	*Weight hidden state*
x_t	*Input*
r_t	*forget gate*
$h_{(t-1)}$	*Hidden state*
σ	*Sigmoid*

Adapun langkah-langkah pemrosesan data menggunakan algoritma *Bidirectional* GRU, misalkan ada data sepotong protein *sequence* "LSVHLSVH". Selanjutnya akan diprediksi karakter yang akan muncul setelah kata "LSVHLSVH". Pertama, tokenisasi kalimat *input* menjadi kata dengan *index* yang berbeda-beda.

'H': 0, 'S' : 1, 'V' : 2, 'L' : 3

Sehingga kata berubah menjadi *encode* seperti di bawah ini.

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

Selanjutnya mengubah *encode sequence* ke dalam bentuk matriks dengan membagi *sequence* dengan *batch size* 3 dan konversi ke dalam bentuk *one-hot encoding*.

$$\begin{bmatrix} 3 & 1 & 2 \\ 0 & 3 & 1 \end{bmatrix} \rightarrow \begin{bmatrix} 0001 & 0100 & 0010 \\ 1000 & 0001 & 0100 \end{bmatrix}$$

$$\downarrow \quad \downarrow \quad \downarrow$$

X_{t-1} X_t X_{t+1}

Selanjutnya, menghitung *update gate* pada *input* pertama yaitu X_{t-1} dengan menggunakan Persamaan (1). Dimana x_t adalah input pada waktu t, $h_{(t-1)}$ adalah *hidden state* dari lapisan sebelumnya pada waktu (t-1) atau pada saat awal *hidden state* pada waktu 0. W adalah berat *(weight)*, dan σ adalah fungsi sigmoid. Nilai W diinisiasikan secara acak.

$$\mathbf{z}_{(Xt-1)} = \begin{bmatrix} 0001 \\ 1000 \end{bmatrix} \bullet \begin{bmatrix} 0.66 & 0.26 \\ 0.06 & 0.62 \\ 0.45 & -0.16 \\ -1.52 & 0.38 \end{bmatrix} + \begin{bmatrix} 0.32 & -0.47 \\ 1.37 & 2.52 \end{bmatrix} \bullet \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

$$\mathbf{z}_{(Xt-1)} = \sigma \begin{bmatrix} -1.522 & 0.381 \\ 0.661 & 0.269 \end{bmatrix}$$

Kemudian agar hasil perhitungan matriks berada pada kisaran 0 sampai 1, selanjutnya hasil dimasukkan ke dalam fungsi sigmoid seperti pada Persamaan (1).

$$\mathbf{z}_{(Xt-1)} = \begin{bmatrix} \frac{1}{1+e^{1.522}} & \frac{1}{1+e^{-0.381}} \\ \frac{1}{1+e^{-0.661}} & \frac{1}{1+e^{-0.269}} \end{bmatrix}$$

$$\mathbf{z}_{(Xt-1)} = \begin{bmatrix} 0.179 & 0.594 \\ 0.659 & 0.566 \end{bmatrix}$$

Update gate bertujuan untuk menentukan banyaknya informasi pada langkah sebelumnya yang perlu diteruskan. Nilai yang mendekati bilangan 1 akan diteruskan sedangkan nilai yang mendekati bilangan 0 akan berarti hanya informasi baru yang disimpan. Setelah menghitung *update gate* pertama, selanjutnya menghitung *reset gate* dengan cara yang sama menggunakan Persamaan (2).

$$\mathbf{r}_{(Xt-1)} = \begin{bmatrix} 0001 \\ 1000 \end{bmatrix} \bullet \begin{bmatrix} 0.56 & 0.16 \\ 0.06 & 0.42 \\ 0.45 & -0.18 \\ -1.54 & 0.38 \end{bmatrix} + \begin{bmatrix} 0.42 & -0.37 \\ 1.37 & 2.32 \end{bmatrix} \bullet \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

$$\mathbf{r}_{(Xt-1)} = \sigma \begin{bmatrix} -1.54 & 0.38 \\ 0.56 & 0.16 \end{bmatrix}$$

$$\mathbf{r}_{(Xt-1)} = \begin{bmatrix} \frac{1}{1+e^{1.54}} & \frac{1}{1+e^{-0.38}} \\ \frac{1}{1+e^{-0.56}} & \frac{1}{1+e^{-0.16}} \end{bmatrix}$$

$$\mathbf{r}_{(Xt-1)} = \begin{bmatrix} 0.176 & 0.593 \\ 0.636 & 0.539 \end{bmatrix}$$

Reset gate bertujuan untuk menentukan dan menyimpan berapa banyak informasi pada langkah sebelumnya yang perlu dilupakan atau dihapus. Selanjutnya, menghitung *current memory* dengan cara yang sama menggunakan Persamaan (3).

$$h'_{(Xt-1)} = \begin{bmatrix} 0001 \\ 1000 \end{bmatrix} \bullet \begin{bmatrix} -0.16 & 0.36 \\ 0.46 & 0.42 \\ 0.45 & -0.18 \\ -1.24 & 0.58 \end{bmatrix} + \begin{bmatrix} 0.176 & 0.593 \\ 0.636 & 0.539 \end{bmatrix} \odot \begin{bmatrix} -0.22 & -0.51 \\ 1.42 & 0.62 \end{bmatrix} \bullet$$

$$\begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

$$h'_{(Xt-1)} = \begin{bmatrix} -1.24 & 0.58 \\ -0.16 & 0.36 \end{bmatrix}$$

Hasil yang didapatkan selanjutnya dimasukkan ke dalam fungsi tanh, sehingga nilainya berada pada kisaran -1 sampai dengan 1. Fungsi tanh dapat dilihat pada Persamaan (5).

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

$$h'_{(Xt-1)} = \begin{bmatrix} \frac{exp^{-1.24} - exp^{-1.24}}{exp^{-1.24} + exp^{-1.24}} & \frac{exp^{0.58} - exp^{0.58}}{exp^{0.58} + exp^{0.58}} \\ \frac{exp^{-0.16} - exp^{-0.16}}{exp^{-0.16} + exp^{-0.16}} & \frac{exp^{0.36} - exp^{0.36}}{exp^{0.36} + exp^{0.36}} \end{bmatrix}$$

$$h'_{(Xt-1)} = \begin{bmatrix} -0.84 & 0.52 \\ -0.16 & 0.34 \end{bmatrix}$$

Selanjutnya yaitu menghitung *output* dari *sequence* 1 pada *batch* pertama yang akan digunakan sebagai *input* pada *sequence* 2 pada *batch* pertama. Perhitungan *output* dilakukan dengan menggunakan Persamaan (4).

$$h_{(Xt-1)} = \left(1 - \begin{bmatrix} 0.179 & 0.594 \\ 0.659 & 0.566 \end{bmatrix} \right) * \begin{bmatrix} -0.84 & 0.52 \\ -0.16 & 0.34 \end{bmatrix} + \begin{bmatrix} 0.0179 & 0.594 \\ 0.659 & 0.566 \end{bmatrix} * \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

$$h_{(Xt-1)} = \begin{bmatrix} -0.69 & 0.21 \\ -0.05 & 0.15 \end{bmatrix}$$

Langkah yang sama dilakukan untuk *sequence* kedua dan ketiga. Setelah semua *sequence* dihitung, selanjutnya untuk melakukan prediksi yaitu dengan mengubah *output* menggunakan *linear layer* yang dapat dihitung menggunakan Persamaan (6). *Input* awal terdiri dari 4 karakter unik sehingga *output* yang diharapkan juga memiliki ukuran yang sama. *Input* dimasukkan ke dalam *dense layer* sehingga memiliki ukuran *input* yang sama panjang dengan jumlah karakter unik pada *sequence* dan diteruskan ke fungsi *activation softmax* sehingga menghasilkan suatu prediksi. Fungsi *activation softmax* dapat dilihat pada Persamaan (7).

$$linear = W_y h_{(t-1)} + b_y \quad \text{.....(6)}$$

$$linear = \begin{pmatrix} -0.69 & 0.21 \\ -0.05 & 0.15 \end{pmatrix} \begin{bmatrix} 0.75 & -0.34 & -0.69 & 0.68 \\ 0.14 & -0.67 & 0.87 & 0.53 \end{bmatrix} + (0\ 0\ 0\ 0)$$

$$= \begin{bmatrix} -0.4881 & 0.0939 & 0.6588 & -0.3579 \\ -0.0165 & -0.0835 & 0.165 & 0.0455 \end{bmatrix}$$

$$softmax = \frac{\exp(xi)}{\sum \exp(xj)} \quad \text{.....(7)}$$

Untuk mendapatkan nilai x_i dengan cara mengurangi semua elemen pada hasil perhitungan *linear* dengan nilai maksimum pada yang didapatkan pada perhitungan *linear* yaitu sebesar 0,9021. Nilai ini berfungsi untuk mencegah terjadi *exploding value* pada perhitungan *linear* yang besar.

$$\exp\left(y_{linear} - y_{linear\_max}\right) = exp\begin{pmatrix} -0.4881 & 0.0939 & -0.6588 & -0.3579 \\ -0.0165 & -0.0835 & 0.165 & 0.0455 \end{pmatrix} - 0.9021$$

$$= \exp\begin{pmatrix} -0.4881 - 0.9021 & 0.0939 - 0.9021 & -0.6588 - 0.9021 & -0.3579 - 0.9021 \\ -0.0165 - 0.9021 & -0.0835 - 0.9021 & 0.165 - 0.9021 & 0.0455 - 0.9021 \end{pmatrix}$$

$$= \begin{pmatrix} exp^{-1.3902} & exp^{-0.8082} & exp^{-1.5609} & exp^{-1.26} \\ exp^{-0.9186} & exp^{-0.9856} & exp^{-0.7371} & exp^{-0.8566} \end{pmatrix} = \begin{pmatrix} 0.2490 & 0.4456 & 0.2099 & 0.2836 \\ 0.3990 & 0.3732 & 0.4784 & 0.4246 \end{pmatrix}$$

Setelah didapatkan nilai xi, selanjutnya untuk mendapatkan nilai xj dengan menjumlahkan semua elemen pada nilai xi.

$$\sum exp^{\exp\left(y_{linear} - y_{linear\_max}\right)} = \begin{pmatrix} 0.2490 & + & 0.4456 & + & 0.2099 & + & 0.2836 \\ 0.3990 & + & 0.3732 & + & 0.4784 & + & 0.4246 \end{pmatrix}$$

$$= \begin{pmatrix} 1.1881 \\ 1.6752 \end{pmatrix}$$

Setelah mendapatkan nilai xi dan xj, selanjutnya dimasukkan ke dalam fungsi *softmax*.

$$Softmax = \begin{pmatrix} \frac{0.2490}{1.1881} & \frac{0.4456}{1.1881} & \frac{0.2099}{1.1881} & \frac{0.2836}{1.1881} \\ \frac{0.3990}{1.6752} & \frac{0.3732}{1.6752} & \frac{0.4784}{1.6752} & \frac{0.4246}{1.6752} \end{pmatrix} = \begin{pmatrix} 0.2095 & 0.3750 & 0.1766 & 0.2387 \\ 0.2381 & 0.2227 & 0.2855 & 0.2534 \end{pmatrix}$$

Dari hasil tersebut didapatkan prediksi karakter yang akan muncul adalah karakter S karena probabilitas tertinggi terdapat pada karakter S.

2.8 *Dropout Layer*

Dropout layer adalah *layer* pada aristektur *neural network* yang dapat membantu mencegah *overfitting* pada model dengan cara menonaktifkan nilai-nilai tertentu pada *layer* dengan mengubah nilainya menjadi 0 (Asghar, et al., 2020). Pemilihan secara acak pada *dropout layer* dapat mengabaikan *node hidden layer* dalam proses pelatihan, sehingga setiap kali proses pelatihan berjalan model akan menampilkan jaringan pelatihan berbeda-beda. Oleh karena itu, setiap pelatihan dapat dianggap sebagai model baru (Yang dan Yang, 2018).

2.9 *Flatten Layer*

Flatten layer digunakan untuk mengubah data yang multi dimensi menjadi satu dimensi sebagai *input* untuk *layer* selanjutnya. *Flatten layer* biasanya terhubung dengan *layer* terakhir yaitu *fully connected layer* yang hanya dapat menerima *input* berupa satu dimensi saja untuk selanjutnya diklasifikasikan.

2.10 *Dense Layer*

Dense layer (Fully Connected Layer) merupakan salah satu model tradisional *neural network* untuk melakukan klasifikasi sesuai kategori kelas pada *output*. *Dense layer* memiliki *input* dan *output* yang jumlahnya tergantung dengan kategori kelas yang diprediksi (Andros, et al., 2015). *Dense layer* biasa digunakan pada tahap akhir pada model *neural network*.

2.11 *Confusion Matrix*

Confusion Matrix merupakan sebuah metode untuk mengevaluasi kinerja algoritma klasifikasi dengan menggunakan tabel matriks yang menyatakan jumlah data uji yang benar dan jumlah data uji yang salah dalam proses klasifikasi (Indriani, 2014). Contoh *Confusion Matrix* dapat dilihat pada Tabel 6.

Tabel 6. Tabel *Confusion Matrix*

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

Penjabaran makna dari istilah yang digunakan pada *confusion matrix* berdasarkan Tabel 6 sebagai berikut.

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

Pengukuran yang paling sering digunakan dalam mengevaluasi algoritma klasifikasi yaitu akurasi (ACC), sensitivitas (SN), spesifisitas (SP), dan

Matthew Correlation Coefficient (MCC). Berikut ini persamaan yang digunakan pada masing-masing matriks.

a. Akurasi (ACC)

Akurasi adalah persentase pasangan DNA-*binding* protein dan non DNA-*binding* protein yang diidentifikasi dengan benar (You, Chan dan Hu, 2015). Rumus yang digunakan dalam menghitung akurasi dapat dilihat pada Persamaan (8).

$$ACC = \frac{TP+TN}{TP+FP+TN+FN} * 100 \quad \text{..... (8)}$$

b. Sensitivitas (SN)

Sensitivitas adalah persentase pasangan DNA-*binding* protein yang diidentifikasi dengan benar (You, Chan dan Hu, 2015). Rumus yang digunakan dalam menghitung sensitivitas dapat dilihat pada Persamaan (9).

$$SN = \frac{TP}{TP+FN} * 100 \quad \text{.....(9)}$$

c. Spesifisitas (SP)

Spesifisitas adalah persentase pasangan non DNA-*binding* protein yang diidentifikasi dengan benar (You, Chan dan Hu, 2015). Rumus yang digunakan dalam menghitung spesifisitas dapat dilihat pada Persamaan (10).

$$SP = \frac{TN}{TN+FP} * 100 \quad \text{.....(10)}$$

d. *Matthew Correlation Coefficient* (MCC)

MCC adalah ukuran akurasi prediksi yang lebih ketat dengan memperhitungkan nilai positif dan nilai negatif yang bernilai salah dan nilai benar (Zhu, 2020). Rumus yang digunakan dalam menghitung MCC dapat dilihat pada Persamaan (11).

$$MCC = \frac{TP\ x\ TN - FP\ x\ FN}{\sqrt{(TP+FN)\ x\ (TN+FP)\ x\ (TP+FP)\ x\ (TN+FN)}} * 100 \quad \text{.....(11)}$$

III. DATA DAN METODOLOGI

3.1 Tempat dan Waktu

Adapun penjabaran terkait tempat dan waktu penelitian dijelaskan sebagai berikut.

3.1.1 Tempat

Penelitian dilaksanakan di Lab RPL Jurusan Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam (FMIPA) Universitas Lampung.

3.1.2 Waktu dan Jadwal Penelitian

Penelitian dimulai pada bulan Desember 2022 di semester tujuh ganjil hingga penyelesaian pada bulan September 2023. Untuk penjelasan kegiatannya terdapat pada Tabel 7.

Tabel 7. Alur Pelaksanaan Penelitian

Kegiatan	2022												2023																							
	Des				Jan				Feb				Mar				April				Mei				Jun				Jul				Ags				Sep		
	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
Perencanaan penelitian																																							
Pengumpulan Data																																							
Preprocessing Data																																							
-Tokenization																																							
-Padding																																							
-One-hot encoding																																							
Pemodelan BiGRU																																							
Evaluasi Kinerja Model																																							
Penyusunan Skripsi																																							

Pada Tabel 7 menggambarkan alur waktu pengerjaan penelitian yang secara garis besar dibagi menjadi 3 tahapan, yaitu :

1. Perencanaan

   Tahap perencanaan penelitian diawali dengan melakukan studi literatur yaitu dengan mencari jurnal referensi untuk menentukan topik dan metode penelitian yang akan digunakan. Selanjutnya pada minggu kedua, mencari sumber data yang akan digunakan untuk penelitian.
2. Pelaksanaan

   Pada tahap pelaksanaan penelitian dimulai pada minggu ke tiga pada bulan Desember dengan melakukan pengumpulan data. Pada minggu setelahnya, melakukan *preprocessing* pada data. Setelah dilakukan *preprocesing*, pada minggu kedua bulan Januari mulai melakukan proses tokenisasi dan *padding* pada data. Selanjutnya pada minggu pertama bulan Febuari 2023 mulai melakukan pemodelan dengan menggunakan metode *Bidirectional* GRU.
3. Evaluasi

   Tahap ini adalah tahap terakhir penelitian dimana model yang dihasilkan akan dievaluasi kinerjanya dalam mengklasifikasi protein pengikat DNA. Evaluasi kinerja model dinilai dengan menggunakan *Confusion Matrix* dengan empat formulasi yang digunakan yaitu akurasi, sensitivitas, spesifisitas, dan *Matthew Corelation Coefficient.*

3.2 Data dan Alat

Adapun data dan alat yang digunakan selama proses penelitian yaitu sebagai berikut.

3.2.1 Data

Data *training* yang digunakan merupakan dataset *benchmark* PDB1075 yang diekstrak oleh Liu, et al., (2014). Dataset terdiri dari 525 data positif dan 550 data negatif. Dataset protein pengikat DNA diekstrak dari Protein Data *Bank* (PDB). Panjang masing-masing *sequence*

berkisar di antara 51 – 1323 *amino acid.* Data dapat diakses melalui https://static-content.springer.com/esm/art3A10.1038%2Fsrep15479/MediaObjects/41598_2015_BFsrep15479_MOESM1_ESM.pdf.

Data *testing* yang digunakan merupakan dataset *benchmark* PDB186 yang telah diekstrak oleh Lou, et al., (2014). Dataset terdiri dari 93 urutan protein pengikat DNA positif dan 93 urutan protein pengikat DNA negatif. Panjang masing-masing *sequence* berkisar di antara 64 – 1323 *amino acid.* Data dapat diaskes melalui tautan berikut. https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0086703#pone.0086703.s001.

Adapun contoh bentuk dari data yang telah didapatkan dapat dilihat pada Gambar 7 dan ringkasan dataset dapat dilihat pada Tabel 8.

Gambar 7. Dataset DNA-*binding* Protein *Sequence*.

Tabel 8. Ringkasan Dataset Penelitian

Nama Dataset	**Panjang *sequence***	**Jumlah data protein positif**	**Jumlah data protein negatif**	**Jumlah total**
PDB1075	51-1323 *amino acid*	525	550	1075
PDB186	64-1323 *amino acid*	93	93	186

Berdasarkan Tabel 8 didapatkan bahwa jumlah total keseluruhan data urutan protein pengikat DNA yang digunakan pada penelitian ini berjumlah 1261 data.

3.2.2 Alat

A. Perangkat Keras

Perangkat keras yang digunakan selama proses penelitian yaitu sebagai berikut.

a. *System Manufacture* : PC
b. *Processor* : AMD Ryzen 5 3400G with Radeon Vega Graphics x 8
c. *Storage* : WDC WDS500G2B0C-00PXH0

B. Perangkat Lunak

Perangkat lunak yang digunakan selama proses penelitian yaitu sebagai berikut.

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

   Jupyter Notebook merupakan aplikasi yang dipakai oleh data *scientist* untuk menganalisis data, membuat visualisasi data dan membagikan dokumen yang memiliki kode, perhitungan, dan teks. Jupyter singkatan dari tiga bahasa pemrograman, yakni Julia (Ju), Python (Py), dan R. Tiga bahasa pemrograman tersebut sangat penting bagi seorang data *scientist*.
c. Python 3.9

   Python merupakan salah satu bahasa yang digunakan ilmuwan untuk mengaplikasikan ilmu data. Ilmuwan data menggunakan bahasa pemrograman Python untuk memanipulasi data seperti membuat visualisasi data, membersihkan data, dan membangun sebuah model. Kelebihan bahasa pemrograman Python dalam pemrosesan data yaitu mudah dan sederhana dalam penggunaan. Selain itu bahasa pemrograman Python menyediakan berbagai *library* yang membantu memecahkan masalah analisi data.

d. *Library* Pandas 1.3.4

Pandas merupakan salah satu *library* yang populer untuk memanipulasi dan analisis data. *Library* ini mempermudah dalam membaca data yang ada di dalam dokumen ke dalam bentuk tabulasi. Pandas dapat membantu memanipulasi, agregasi, dan visualisasi sejumlah besar data terstruktur dengan cepat dan mudah.

e. *Library* Numpy 1.20.3

Numpy merupakan singkatan dari *Numerical* Python, yaitu *library* pada Python yang menyediakan fungsi matematika dengan berbagai metode seperti fungsi array, matriks, dan aljabar linier yang digunakan untuk menangani array berdimensi besar. *Library* ini menyediakan vektorisasi operasi matematika pada jenis array Numpy yang meningkatkan kinerja dan mempercepat eksekusi.

f. *Library* Tensorflow 2.9.1

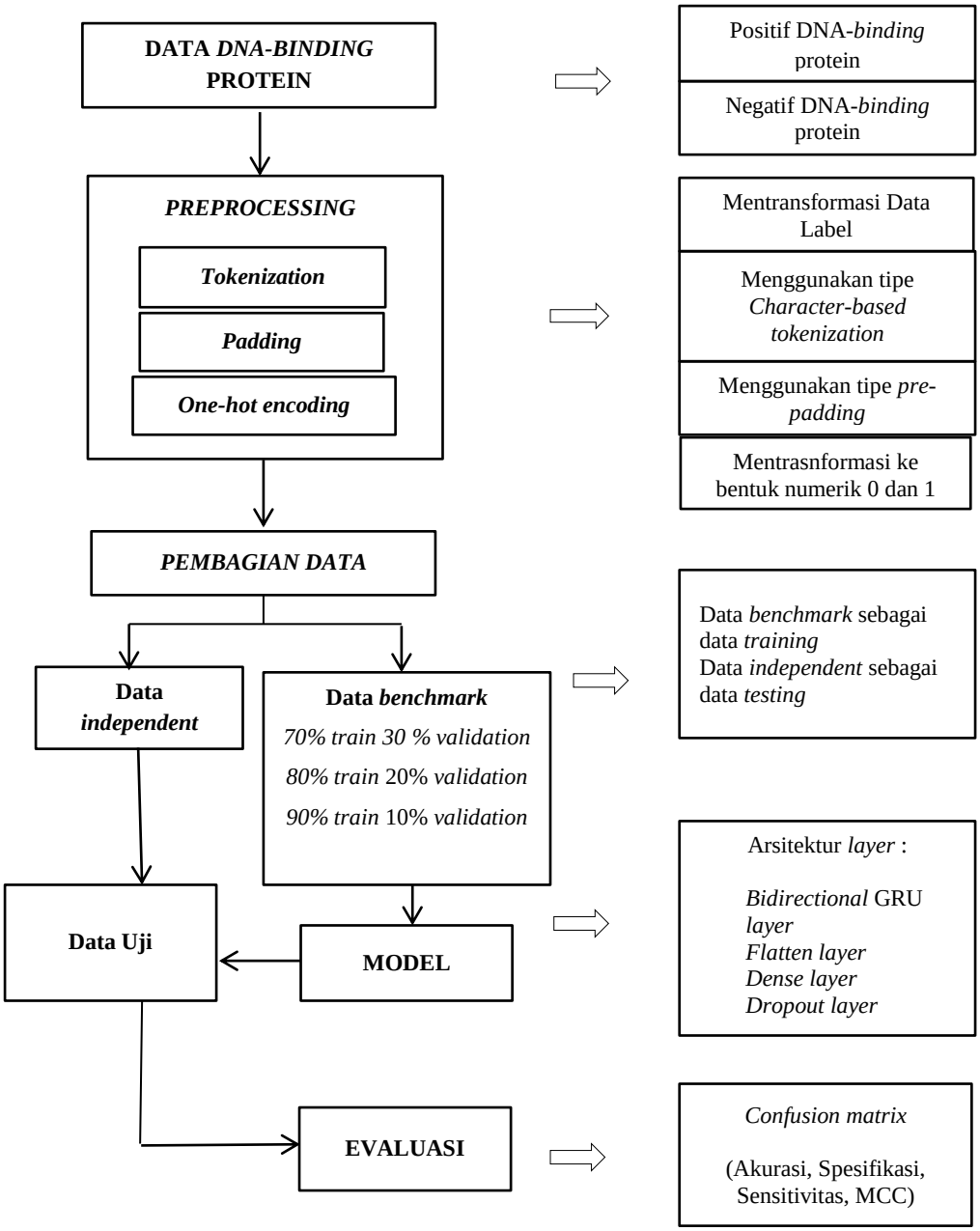
Tensoflow merupakan *library* pada Python yang digunakan untuk melakukan pemodelan pada *deep learning. Library* ini berfokus pada proses *training, interfacing,* dan *deep neural network. Library* Tensorflow dikembangkan oleh Google pertama kali pada tahun 2015.

g. *Library* Scikit-learn 1.2.0

Scikit-learn atau Sklearn merupakan *library* yang menyediakan berbagai fungsi dan algoritma yang digunakan untuk pembelajaran mesin. Sklearn menyediakan *tools* yang sederhana dan mudah untuk proses data *mining* dan analisis data. *Library* ini dapat membantu menerapkan algoritma populer pada data yang akan dianalisis.

3.3 Metodologi

Alur kerja penelitian yang digunakan didasari oleh penelitian-penelitian sebelumnya terkait dengan klasifikasi urutan protein menggunakan metode *Bidirectional Gated Recurrent Unit* (BiGRU). Diagram alur kerja penelitian dapat dilihat pada Gambar 8.



Gambar 8. Alur Kerja Penelitian.

Berdasarkan Gambar 8, penelitian dimulai dengan mengumpulkan data *training* dan data *testing* yang berupa data urutan protein pengikat DNA positif dan data urutan protein pengikat DNA negatif. Selanjutnya dilakukan *preprocessing* pada data dengan cara mentransformasi data label. Data selanjutnya dilakukan proses tokenisasi dan kemudian dilakukan proses *padding* dengan nilai maksimal yang telah ditentukan agar *input sequences* memiliki panjang yang sama. Kemudian dilakukan proses *one-hot encoding* untuk mentransformasi token *sequence* menjadi bentuk numerik yang bernilai 0 dan 1. Langkah selanjutnya yaitu data dipisah menjadi data *training* dan data *testing*, di mana data *training* dibagi menjadi tiga skenario yaitu 90% data *training* 10% data *validation*, 80% data *train* 20% data *validation*, 70% data *train* 30% data *validation*. Kemudian kedua data tersebut dilatih dengan menggunakan aristektur *Bidirectional Gates Recurrent Unit* (BiGRU). Setelah model *training* mendapatkan akurasi, spesifisitas, sensitivitas, dan MCC yang cukup baik maka model akan diuji coba menggunakan data *testing* dan kemudian akan dievaluasi hasil prediksi yang didapatkan.

3.3.1 Pengumpulan Data

Data urutan protein yang digunakan pada penelitian ini menggunakan dua dataset yaitu data untuk proses *training* dan data untuk proses *testing.* Data training didapat dari dataset *benchmark* PDB1075 yang terdiri dari 525 urutan protein pengikat DNA positif dan 550 urutan protein pengikat DNA negatif. Sedangkan data *testing* didapat dari dataset *benchmark* PDB186 yang terdiri dari 93 urutan protein pengikat DNA positif dan 93 urutan protein pengikat DNA negatif.

3.3.2 *Tokenization*

Setelah data label menjadi data bentuk numerik, selanjutnya data *sequence* dilakukan proses tokenisasi yang bertujuan untuk mengubah data tektual menjadi data numerik. Proses ini mengubah setiap asam amino menjadi bentuk numerik dengan menggunakan tipe tokenisasi *character-based tokenization.*

3.3.3 *Padding*

Data *sequence* yang sudah diubah menjadi bentuk numerik selanjutnya dilakukan proses *padding* yang bertujuan agar input *sequence* memiliki panjang yang sama. Proses *padding* diatur dengan menggunakan *padding_type* yaitu '*pre*' yang artinya *padding* dilakukan pada sebelum *sequence* protein.

3.3.4 *One-hot encoding*

Setelah data dilakukan *padding*, selanjutnya data akan ditransformasi menjadi bentuk numerik yang bernilai 0 dan 1 dengan menggunakan teknik *one-hot encoding*. Data *input* yang telah dilakukan *one-hot encoding* akan memiliki ukuran 3 dimensi yang sesuai dengan ketentuan *input* pada *layer Bidirectional* GRU. Hal ini bertujuan agar data *input* dapat langsung masuk ke dalam *layer Bidirectional* GRU pada proses pemodelan.

3.3.5 Pemodelan

Setelah semua data menjadi bentuk data numerik selanjutnya dataset *training* akan dipisah menjadi dua bagian yaitu data *train* dan data *validation*. Selanjutnya kedua data tersebut dilatih dan diklasifikasi menggunakan metode *Bidirectional Gated Recurrent Unit* (BiGRU). Pemodelan dilakukan dengan menggunakan arsitektur *layer* yaitu *bidirectional* GRU *layer*, *dense layer*, *dropout layer*, serta menggunakan *activation sigmoid* dan *optimizer* Adam.

3.3.6 Evaluasi dan *Testing*

Setelah model dilatih selanjutnya model akan dievaluasi kinerjanya dengan parameter evaluasi *confusion matrix* yaitu akurasi, sensitivitas, spesifisitas, dan MCC. Jika model *training* dapat mengklasifikasi protein pengikat DNA dengan baik, maka selanjutnya model akan diuji coba dengan menggunakan data *testing* yang telah disediakan.

V. PENUTUP

5.1 Simpulan

Berdasarkan penelitian dan penjabaran mengenai klasifikasi DNA-*binding* protein menggunakan metode *Bidirectional* GRU yang telah dilakukan, beberapa hal yang dapat disimpulkan antara lain.

1. Metode *Bidirectional* GRU dapat diimplementasikan untuk membuat model klasifikasi DNA-*binding* protein dengan menggunakan beberapa parameter pendukung yaitu *optimizer* Adam, *learning rate* 0,001, 80 *epochs,* dan 64 *batch size*.
2. Hasil tertinggi yang didapatkan pada penelitian ini, yaitu pada arsitektur BiGRU *single layer* dengan pembagian data 80% data *train* 20% data validasi dengan nilai akurasi 81,72%, sensitivitas 90,32%, spesifisitas 73,11%, dan MCC 64,40%.
3. Hasil perbandingan dengan penelitian terdahulu menunjukkan bahwa penelitian oleh Shadab, et al pada tahun 2020 yang menggunakan metode *Convolutional Neural Network* (CNN) memiliki hasil klasifikasi yang lebih baik dengan nilai akurasi 84,31%. Meskipun hasil penelitian ini mendapatkan hasil yang cukup baik, namun hasil yang didapatkan belum mampu mengungguli penelitian oleh Shadab, et al., (2020). Dengan ini membuktikan bahwa penggunaan metode *Bidirectional* GRU belum dapat menghasilkan hasil klasifikasi yang lebih baik dari penelitian terdahulu.

5.2 Saran

Adapun saran yang diberikan dalam penelitian ini adalah sebagai berikut.

1. Penelitian ini dapat dilanjutkan dengan menambahkan proses *hyperparameter optimization* dengan menggunakan metode *Grid Search* sehingga dapat dihasilkan akurasi yang lebih optimal dengan mencoba semua alternatif yang dapat dilakukan.

2. Penelitian ini dapat dilanjutkan dengan menggunakan metode klasifikasi lainnya sehingga dapat memperoleh hasil klasifikasi yang lebih baik sebagai bahan perbandingan.

DAFTAR PUSTAKA

Agarwal, Vivek. "Research on Data Preprocessing and Categorization Technique for Smartphone Review Analysis." *International Journal of Computer*, 2015.

Alberts, B, A Johnson, and J Lewis. *Molecular Biology of the Cell. 4th edition.* New York: Garland Science, 2002.

Alkaoud, Mohamed, and Mairaj Syed. "On the Importance of Tokenization in Arabic Embedding Models." *Arabic Natural Language Processing*, 2020.

Amajd, Z.K, Maaz. "Text Classification with Deep Neural Networks." *Industrial Research*, 2017.

Andros, Dimas Prawita, Juan Karsten, and Maldy Vinandar. "Perbandingan Algoritma Pendeteksian Spam." *Jurnal Teknologi Terpadu*, 2015.

Asghar, Muhammad Zubair, et al. "Senti-eSystem: A sentiment-based eSystem-using hybridized fuzzy and deep neural network for measuring customer satisfaction." *Software: Practice and Experience*, 2020.

Cao, Yudong, Minping Jia, Peng Ding, and Yifei Ding. "Transfer learning for remaining useful life prediction of multi-conditions bearings based on bidirectional-GRU network." *Measurement*, 2021.

Deng, Yaping, Lu Wang, Hao Jia, Xiangqian Tong, and Feng Li. "Sequence-to-Sequence Deep Learning Architecture Based on Bidirectional GRU for Type Recognition and Time Location of Combined Power Quality Disturbance." *Transactions on Industrial Informatics*, 2019.

Han, J, and M Kamber. *Data Mining: Concepts and Techniques.* Second Edition, 2006.

Handamari, Endang Wahyu , Kwardiniya A, Mila Kurniawaty, and Emilia S I. "Prediksi Profil Asam Amino Pada Family Protein Menggunakan Hidden Markov Model." *Jurnal Pointer*, 2011.

Hartono, Devina Adinda, Setyorini, and Siti Amatullah Karimah. "Model Komputasi BLAST pada Lingkungan Hadoop." *e-Proceeding of Engineering*, 2021.

Hu, Xia, Lingyang Chu, Jian Pei, Weiqing Liu, and Jiang Bian. "Model Complexity of Deep Learning." *Knowledge and Information System*, August 2021.

Indriani, Aida. "Klasifikasi Data Forum dengan menggunakan Metode Naïve Bayes Classifier." *Jurnal SNATI*, 2014.

Jen, K -Y, and A Travers. "DNA-Binding Proteins." *Brenner's Encyclopedia of Genetics*, 2013.

Jingna, Si, Zhao Rui, and Wu Rongling. "An Overview of the Prediction of Protein DNA-Binding Sites." *International Journal of Molecular Sciences*, 2015.

Jun, Yan, et al. "DNA-binding protein prediction based on deep transfer learning." *Mathematical Biosciences and Engineering*, 2022.

Katsnelson, Zachary. *Kaggle*. 2021. https://www.kaggle.com/code/zakarii/dna-sequence-classification-cnn-gru (accessed March 23, 2023).

Kostadinov, Simeon. *Medium*. December 16, 2017. https://towardsdatascience.com/understanding-gru-networks-2ef37df6c9be (accessed July 25, 2023).

Le, N.Q.K., E.K.Y Yapp, and H.Y Yeh. "ET-GRU: using multi-layer gated recurrent units to identify electron transport proteins." *BMC Bioinformatics*, 2019.

Liu, Bin, et al. "iDNA-Prot|dis: Identifying DNA-Binding Proteins by Incorporating Amino Acid Distance-Pairs and Reduced Alphabet Profile into the General Pseudo Amino Acid Composition." *Plos One*, 2014.

Lou, Wangchao, Xiaoqing Wang, Fan Chen, Yixiao Chen, Bo Jiang, and Hua Zhang. "Sequence Based Prediction of DNA-Binding Proteins Based on Hybrid Feature Selection Using Random Forest and Gaussian Naïve Bayes." *Plos One*, 2014.

Mullen, Lincoln A., Kenneth Benoit, Os Keyes, Dmitry Selivanov, and Jeffrey Arnold. " Consistent Tokenization of Natural Language Text." *Journal of Open Source Software*, 2018.

Neidle, S. "Principles of Protein-DNA Recognition." *Principles of Nucleic Acid Structure*, 2008.

Rahman, M. Saifur, Swakkhar Shatabda, Sanjay Saha, M Kaykobad, and M. Sohel Rahman. "DPP-PseAAC: A DNA-binding Protein Prediction Model Using Chou's." *Journal of Theoretical Biology*, 2018.

Reddy, Dwarampudi Mahidhar, and N V Subba Reddy. "Effect Of Padding On LSTMS And CNNS." *Computation and Language*, 2019.

Rishita, Middi Venkata Sai, Middi Appala Raju, and Tanvir Ahmed Harris. "Machine translation using natural language." *JCMME*, 2018.

Sang, Xiuzhi, Wanyue Xiao, Huiwen Zheng, Yang Yang, and Taigang Liu. "HMMPred: Accurate Prediction of DNA-Binding Proteins Based on HMM Profiles and XGBoost Feature Selection." *Computational and Mathematical Methods in Medicine*, 2020.

Shadab, Shadman, Md Tawab Alam Khan, Nazia Afrin Neezi, Sheikh Adilina, and Swakkhar Shatabda. " Deep neural networks for identification of DNA-binding proteins ." *Informatics in Medicine*, 2020.

Shen, Zhen, Wenzheng Bao, and De-Shuang Huang. "Recurrent Neural Network for Predicting Transcription Factor Binding Sites." *Scientific Report*, 2018.

W.Lou, X.Wang, F.Chen, Y.Chen, B.Jiang, and H.Zhang. "Sequence based prediction of DNA-binding proteins based on hybrid feature selection using random forest and gaussian naive bayes." *PLoS One*, 2014.

Wei, Leyi, Jijun Tang, and Quan Zou. "Local-DPP: An improved DNA-binding protein prediction method by exploring local evolutionary information." *Information Sciences*, 2017.

Yang, Bite, et al. "BiRen: predicting enhancers with a deep-learning-based model using the DNA sequence alone." *Bioinformatics*, 2017.

Yang, Jing, and Guanci Yang. "Modified Convolutional Neural Network Based on Dropout and the Stochastic Gradient Descent Optimizer." *Algorithms*, 2018.

You, Z. H, K.C.C Chan, and P. Hu. "Predicting protein-protein interactions from primary protein sequences using a novel multi-scale local feature representation scheme and the random forest." *PLoS ONE, 10(5)*, 2015.

Zaman, Rianon, Shahana Yasmin Chowdhury, Mahmood A Rashid, Alok Sharma, Abdollah Dehzangi, and Swakkar Shatabda. "DNA-Binding Protein Prediction Using HMM Profile Based Features." *BioMed Research International*, 2017.

Zhu, Qiuming. "On the performance of Matthews correlation coefficient (MCC) for imbalanced dataset." *Pattern Recognition Letters*, 2020: 71-80.