KLASIFIKASI DNA-*BINDING* PROTEIN MENGGUNAKAN ALGORITME *RANDOM FOREST*

(Skripsi)

Oleh

Ridho Alrafi

1817051009

ILMU KOMPUTER

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM

UNIVERSITAS LAMPUNG

2022

ABSTRAK

KLASIFIKASI DNA-*BINDING* PROTEIN MENGGUNAKAN ALGORITME *RANDOM FOREST*

Oleh

Ridho Alrafi

Mengklasifikasikan fungsi protein dan struktur dari urutan adalah salah satu tantangan penting untuk kombinasi biologi. Urutan protein masih susah untuk diklasifikasikan karena semakin beragamnya protein yang ditemukan, dan juga banyaknya metode yang bisa digunakan untuk menentukan urutan protein. Pada penelitian ini berfokus terhadap protein pengikat DNA. Protein pengikat DNA memainkan peran penting dalam beberapa bagian besar proses seluler. Oleh karena itu, perlu dilakukannya pengklasifikasian untuk mengidentifikasi protein pengikat DNA berdasarkan urutan protein menggunakan ekstrasi fitur dan random forest. Tujuan dari penelitian ini untuk menganalisa sensitivitas, spesifisitas, akurasi, dan matthew correlation coefficient dari protein pengikat DNA. Dataset protein pengikat DNA diperoleh dari PDB (Protein Data Bank) dengan mencari kata kunci "DNA-binding protein", memiliki jumlah protein 1075 dengan 525 data positif dan 550 data negatif serta memiliki panjang protein terpendek, yaitu 51 amino acids. Dataset dibagi menjadi 2, yaitu 80% data latih 20% data uji dan 90% data latih 10% data uji. Ekstraksi fitur yang digunakan protein descriptor menggunakan R package BioSeqClass versi 1.44.0, yaitu AAIndex, CTD, dan PseAAC, dengan jumlah total 440 fitur. Hasil yang didapatkan diolah kembali menggunakan klasifikasi algoritme random forest dengan mtry 10, 21, dan 42 lalu ntree dipilih secara acak 100, 250, 500, dan 1000. Hasil yang didapatkan paling tinggi didapatkan pada pembagian dataset 90% data latih 10% data uji dengan mtry 42 ntree 1000 sebesar 89.97% sensitivitas, 92.79% spesifisitas, 80.76% MCC, dan 90.42% akurasi. Hasil yang didapatkan menggunakan ekstraksi fitur dan algoritme random forest bisa mengklasifikasikan protein pengikat DNA.

Kata Kunci: Protein pengikat DNA, fitur ekstraksi, klasifikasi, random forest.

ABSTRACT

CLASSIFICATION DNA-BINDING PROTEIN USING RANDOM FOREST ALGORITHM

By

Ridho Alrafi

Classifiying the function and structure of proteins from sequences is one of the most important challenges for combination biology. Protein sequences are still difficult to classify because of the increasing variety of proteins found, as well as the many methods that can be used to determine protein sequences. In this research, we focus on DNA-binding proteins. DNA-binding proteins plays an important role in several major cellular processes. Therefore, it is necessary to classify to identify DNA-binding proteins based on protein sequences using feature extraction and random forest. The purpose of this research was to analyze the sensitivity, specificity, accuracy, and Matthew Correlation Coefficient of DNA-binding protein. The DNA-binding protein dataset was obtained from PDB (Protein Data Bank) by searching the keyword "DNA-binding protein", it has 1075 proteins with 525 positive data and 550 negative data and has the shortest protein length of 51 amino acids. The dataset separation into 2, 80% training data 20% test data and 90% training data 10% test data. Feature extraction used by protein descriptors using R package BioSeqClass version 1.44.0, using AAIndex, CTD, and PseAAC, with a total of 440 features. The results obtained were reprocessed using the random forest algorithm classification with mtry 10, 21, and 42 then ntree was selected at random 100, 250, 500 (default), and 1000. The highest results obtained in the separation of the dataset 90% training data 10% test data with mtry 42 ntree 1000, result 89.97% sensitivity, 92.79% specificity, 80.76% MCC, and 90.42% accuracy. The results obtained using feature extraction and random forest algorithms can classification of DNA binding proteins.

Keywords: DNA-binding protein, feature extraction, classification, random forest.

KLASIFIKASI DNA-*BINDING* PROTEIN MENGGUNAKAN ALGORITME *RANDOM FOREST*

Oleh

Ridho Alrafi

Skripsi

Sebagai Salah Satu Syarat untuk Memperoleh Gelar

SARJANA ILMU KOMPUTER

Pada

Jurusan Ilmu Komputer

Fakultas Matematika dan Ilmu Pengetahuan Alam

ILMU KOMPUTER

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM

UNIVERSITAS LAMPUNG

BANDAR LAMPUNG

2022

Judul Skripsi	:	**KLASIFIKASI DNA-*BINDING* PROTEIN MENGGUNAKAN ALGORITME *RANDOM FOREST***
Nama Mahasiswa	:	Ridho Alrafi
Nomor Pokok Mahasiswa	:	1817051009
Jurusan	:	Ilmu Komputer
Fakultas	:	Matematika dan Ilmu Pengetahuan Alam

MENYETUJUI

1. Komisi Pembimbing

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

2. Ketua Jurusan Ilmu Komputer

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

MENGESAHKAN

1. Tim Penguji

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

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

Penguji II
Penguji Pembahas : **Dr. Ir. Kurnia Muludi, M.S.Sc.** .....

2. Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam

Dr. Eng. Suripto Dwi Yuwono, S.Si., M.T.
NIP 19740705 200003 1 001

Tanggal Lulus Ujian Skripsi : **27 Juli 2022**

PERNYATAAN

Saya yang bertanda tangan di bawah ini:

Nama: Ridho Alrafi

NPM: 1817051009

Dengan ini menyatakan bahwa skripsi saya yang berjudul "KLASIFIKASI DNA-BINDING PROTEIN MENGGUNAKAN ALGORITME RANDOM FOREST" adalah benar hasil 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 oleh orang lain, maka saya bersedia menerima sanksi sesuai dengan peraturan yang berlaku.

Bandar Lampung, 19 Agustus 2022

Penulis

Ridho Alrafi

NPM. 1817051009

RIWAYAT HIDUP

Penulis dilahirkan di Bandar Lampung pada tanggal 16 Januari 2000 sebagai anak ketiga dari tiga bersaudara dari pasangan Bapak Zaberdas (Alm.) dan Ibu Yenni. Penulis menyelesaikan Pendidikan Sekolah Dasar (SD) di SDN 2 Palapa Bandar Lampung (Sampai kelas 2 SD) dan pindah ke SDN 47 Kota Jambi (Sampai kelas 6 SD) pada tahun 2012. Kemudian melanjutkan Pendidikan Sekolah Pertama (SMP) di SMPN 8 Bukittinggi yang diselesaikan pada tahun 2015. Kemudian melanjutkan Pendidikan Sekolah Menengah Atas (SMA) di SMAN 3 Teladan Bukittinggi yang diselesaikan pada tahun 2018.

Penulis terdaftar sebagai mahasiswa Jurusan Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung pada tahun 2018 melalui jalur Seleksi Nasional Masuk Perguruan Tinggi Negeri (SNMPTN). Selama menjadi mahasiswa, penulis melakukan beberapa kegiatan antara lain.

1. Menjadi anggota Adapter Himpunan Mahasiswa Jurusan Ilmu Komputer pada periode 2018/2019.
2. Menjadi anggota pengurus di Bidang Kaderisasi Himpunan Mahasiswa Jurusan Ilmu Komputer (Himakom) pada periode 2018/2019.
3. Menjadi Asisten Dosen Jurusan Ilmu Komputer untuk mata kuliah Dasar-Dasar Pemrograman pada periode semester ganjil tahun ajaran 2019/2020.

4. Menjadi Asisten Dosen Jurusan Ilmu Komputer untuk mata kuliah Sistem Operasi pada periode semester genap tahun ajaran 2019/2020.
5. Menjadi Koor Asisten Dosen Jurusan Ilmu Komputer untuk mata kuliah Komunikasi Data dan Jaringan Komputer pada periode semester ganjil tahun ajaran 2021/2022.
6. Melaksanakan Kerja Praktik di Way Kanan pada tahun 2021.
7. Melaksanakan Kuliah Kerja Nyata (KKN) pada tahun ajaran 2021/2022 di Desa Pesawahan, Teluk Betung Selatan, Bandar Lampung, Lampung.

MOTTO

لَا يُكَلِّفُ اللّٰهُ نَفْسًا اِلَّا وُسْعَهَا

“Allah tidak membebani seseorang melainkan sesuai dengan kesanggupannya.”

(Al-Baqarah: 286)

"Kamu seharusnya tidak menyerah terhadap apapun yang terjadi padamu. Maksudku, kamu seharusnya menggunakan apapun yang terjadi padamu sebagai alat untuk naik, bukan turun."

(Bob Marley)

“Barang siapa keluar untuk mencari sebuah ilmu, maka ia akan berada di jalan Allah hingga ia kembali.”

(HR Tirmidzi)

“Kamu tidak harus menjadi hebat untuk mendapatkannya, tetapi kamu harus mendapatkannya untuk menjadi hebat.”

(Penulis)

PERSEMBAHAN

Alhamdulillahirobbilalamin

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

Aku persembahkan karya ini kepada:

Papa dan Mama

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

Kakakku Martha Zani Putri dan **Abangku** Jufri Yandes

Terima kasih telah memberikan semangat, dukungan, dan doa.

Seluruh Keluarga Besar, Sahabat, dan Teman-teman yang selalu memberikan semangat dan dukungan

Almamater Tercinta, Universitas Lampung

SANWACANA

Puji syukur kehadirat Allah SWT, karena telah memberikan rahmat dan hidayahNya kepada saya sehingga saya dapat menyelesaikan skripsi dengan judul "Klasifikasi DNA-Binding Protein Menggunakan Algoritme Random Forest". Saya berharap skripsi ini dapat menambah pengetahuan bagi pembaca tentang dna-binding protein, fitur ekstraksi dan algoritme random forest.

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

1. Papa, Mama, Kakak, Abang dan keluarga saya yang selalu mendoakan, menyemangati, membiayai serta mendukung saya baik secara moral maupun material. Terima kasih atas doa yang kalian berikan untuk keberhasilan dan kesuksesanku.
2. Bapak Favorisen R. Lumbanraja, Ph. D. sebagai pembimbing utama yang telah membimbing saya, memberikan kritik dan saran serta membina dalam menyelesaikan skripsi ini yang dapat diselesaikan dengan baik.
3. Bapak M. Reza Faisal, S.T., M.T., Ph.D. sebagai pembahas pertama yang telah membimbing saya dalam memberikan ide, kritik, saran sehingga penulisan skripsi ini dapat diselesaikan dengan baik.
4. Bapak Dr. Ir. Kurnia Muludi, M.S.Sc. sebagai pembahas kedua yang telah membimbing saya dalam memberikan ide, kritik, saran sehingga penulisan skripsi ini dapat diselesaikan dengan baik.

5. Bapak Dr. Eng. Admi Syarif sebagai pembimbing akademik saya yang telah membimbing saya selama perkuliahan saya di Jurusan Ilmu Komputer.
6. Bapak Dr. Suripto Dwi Yuwono, M.T. selaku Dekan FMIPA Universitas Lampung.
7. Bapak Didik Kurniawan, S.Si., M.T., selaku Ketua Jurusan Ilmu Komputer Universitas Lampung.
8. Bapak Dr. rer. nat. Akmal Junaidi, M. Sc. selaku sekretaris Jurusan Ilmu Komputer FMIPA Universitas Lampung.
9. Bapak dan Ibu Dosen Jurusan Ilmu Komputer FMIPA Universitas Lampung yang telah memberikan ilmu dan pengalaman dalam hidup untuk menjadi lebih baik.
10. Ibu Ade Nora Maela, Bang Zainuddin, dan Mas Nofal yang telah membantu segala urusan administrasi dan segala jenis izin penulis di Jurusan Ilmu Komputer.
11. Teman-teman Heemahoora yang telah menemani saya dalam menjalani hari-hari dan saling memberi semangat selama masa perkuliahan.
12. Teman seperbimbingan: Intan, Ica, Ajeng, Sucihik, Fajru, dan Sepryan yang saling membantu dan menyemangati satu sama lain serta Rika dan Arbi teman seperjuangan dalam mengurus berkas administrasi akhir perkuliahan..
13. Rekan-rekan *discord* “Prog(r)ammer” yang tidak bisa disebutkan satu persatu yang telah meringankan beban pikiran dengan bermain *game* bersama.
14. Keluarga Ilmu Komputer 2018 yang tidak bisa penulis sebut satu persatu. Keluarga kedua penulis yang telah memberikan pengalaman tak ternilai semasa duduk di bangku kuliah.
15. Seluruh kakak tingkat Ilmu Komputer yang tidak bisa disebutkan satu persatu yang telah membantu selama masa perkuliahan penulis.
16. Semua pihak yang telah berpartisipasi baik secara langsung maupun tidak langsung dalam membantu penyusunan skripsi ini.

Penulis menyadari bahwa dalam penulisan skripsi ini masih terdapat banyak kekurangan karena keterbatasan kemampuan, pengalaman serta pengetahuan penulis. Oleh karena itu, saran dan kritik yang membangun sangat diharapkan

sebagai bahan evaluasi untuk kedepannya. Semoga skripsi ini dapat bermanfaat bagi semua pihak.

Bandar Lampung, 19 Agustus 2022

Ridho Alrafi
NPM. 1817051009

DAFTAR ISI

Halaman

DAFTAR TABEL

Tabel Halaman

DAFTAR GAMBAR

Gambar Halaman

DAFTAR *PSEUDOCODE*

Pseudocode Halaman

I. PENDAHULUAN

1.1. Latar Belakang

Memprediksi fungsi protein dan struktur dari urutan adalah salah satu tantangan penting untuk komputasi biologi. Pada era pasca-genomik, dengan perkembangan teknologi sekuensing yang cepat, data urutan biologis (DNA, RNA dan urutan protein) tumbuh dengan cepat. Pada urutan protein itu masih susah untuk diklasifikasikan karena semakin beragamnya protein yang ditemukan, dan juga banyaknya metode yang bisa digunakan untuk menentukan urutan potein.

Interaksi protein memainkan peran kunci dalam berbagai proses dan fungsi biologis. Dalam sel-sel hidup, termasuk siklus metabolisme,transkripsi DNA dan replikasi, dan kaskade sinyal. Dengan demikian, mengidentifikasi dan mengkarakterisasi interaksi protein dengan benar sangat penting untuk memahami mekanisme molekuler di dalam sel (You et al., 2015).

Feature extraction adalah metode yang lebih umum di mana seseorang mencoba mengembangkan transformasi ruang input ke subruang berdimensi rendah yang menyimpan sebagian besar informasi yang relevan (Khalid et al., 2014). *Feature extraction* dapat digunakan dalam konteks ini untuk mengurangi kompleksitas dan memberikan representasi sederhana dari data yang mewakili setiap variabel dalam ruang fitur sebagai kombinasi linier dari variabel input asli.

Untuk urutan protein, *feature extraction* merupakan solusi yang banyak digunakan untuk melakukan fase praproses ini. Urutan protein adalah rantai residu asam amino di mana setiap residu diwakili oleh karakter dalam alfabet

ukuran 20. Fase *preprocessing* ini adalah langkah kunci menuju proses penemuan pengetahuan yang andal karena secara langsung mempengaruhi kualitas hasil yang diperoleh. Setiap urutan protein dijelaskan oleh satu set motif untuk mencapai vektor nilai di mana nilai-nilai tergantung pada sifat fungsi deskripsi yang mengikat urutan dan motif seperti, kehadiran/tidak adanya motif dalam urutan, jumlah kejadian motif dalam urutan, dan posisi pertama motif dalam urutan (Saidi et al., 2012).

Penelitian ini dibangun dengan f*eature extraction* serta menggunakan algoritme *random forest* (RF) untuk menentukan urutan protein. *Random forest* adalah model pohon yang menghasilkan satu set sejumlah tertentu dari kumpulan pohon secara acak (Saifullah et al., 2017).

1.2. Rumusan Masalah

Berdasarkan latar belakang yang sudah dijelaskan diatas, terdapat masalah dalam penelitian ini adalah sebagai berikut.

1. Mengklasifikasikan *DNA-binding* protein menggunakan algoritme *random forest* berdasarkan dataset yang dimiliki.
2. Menggunakan dan membuktikan *random forest* akan mendapatkan urutan yang memberikan akurasi terbaik *DNA-binding protein sequences*.

1.3. Batasan Masalah

Batasan masalah dalam penelitian ini adalah.

1. Metode untuk mengklasifikasikan *DNA-binding* protein ini menggunakan *feature extraction* dan algoritme *random forest*.
2. Penelitian ini hanya berupa sebauh model, bukan sistem.
3. Data yang digunakan diperoleh dari dataset *benchmark* yang terdapat 1057 *protein sequences* yang dimana terdapat 525 *DNA-binding proteins* (*positive samples*) dan 550 *DNA-binding proteins* (*negative samples*).

1.4. Tujuan Penelitian

Tujuan dari penelitian ini adalah sebagai berikut.

1. Mengukur dan mengklasifikasi hasil dari *feature extraction* dan algoritme *random forest.*
2. Membandingkan hasil yang diperoleh terhadap penelitian terdahulu yang menggunakan dataset yang sama dengan berbagai metode klasifikasi yang digunakan.

1.5. Manfaat Penelitian

Adapun manfaat yang diperoleh adalah sebagai berikut.

1. Menambah pengetahuan mengenai cara kerja *feature extraction* dan *random forest*.
2. Serta mengetahui tingkat keberhasilan dan akurasi dengan menggunakan metode-metode yang dibutuhkan.

II. TINJAUAN PUSTAKA

2.1. Penelitian Terdahulu

Beberapa penelitian yang digunakan adalah sebagai berikut.

Tabel 1. Deskripsi mengenai penelitian terdahulu terkait *DNA-binding protein*

No	Penelitian	Data	Metode	Hasil
1	DNA binding protein identification by combining pseudo amino acid composition and profile-based protein representation (Liu et al., 2015).	***Dataset PDB1075*** Jumlah: 1075 Positif:525 Negatif:550	*Support Vector Machine*	Akurasi : 76,76% Sensitivitas : 75,62% Spesifisitas : 77,45% MCC : 53%
2	Local-DPP: An Improved DNA-binding Protein Prediction Method by Exploring Local Evolutionary Information (Wei, L., Tang, J., & Zou, Q., 2017).		*Random Forest*	Akurasi : 79,20% Sensitivitas : 84% Spesifisitas : 74,50% MCC : 59%
3	iDNAProt-ES: Identifcation of DNA-binding Proteins Using Evolutionary and Structural Features (Chowdhury, S. Y., Shatabda, S., & Dehzangi, A., 2017).		*Support Vector Machine*	Akurasi : 90,18% Sensitivitas : 90,38% Spesifisitas : 90% MCC : 80%
4	DPP-PseAAC: A DNA-binding protein prediction model using Chou's general PseAAC (Rahman, M. S et al., 2018).		*Support Vector Machine*	Akurasi : 95,91% Sensitivitas : 94,10% Spesifisitas : 97,64% MCC : 92%

Berdasarkan Tabel 1, didapatkan penjelasan mengenai setiap penelitian terdahulu yang digunakan, yaitu.

2.1.1. DNA binding protein identification by combining pseudo amino acid composition and profile-based protein representation

Penelitian ini dilakukan oleh Liu , Wang dan Xiaolong yang diterbitkan pada tahun 2015. Penelitian ini menggunakan dataset benchmark dengan 1057 protein *sequences* dengan 525 DNA-*binding proteins (positive samples)* dan 550 *non* DNA-*binding proteins (negative samples)*. Dataset bisa dilihat di https://static-content.springer.com/esm/art%3A10.1038%2Fsrep15479/MediaObjects/41598_2015_BFsrep15479_MOESM1_ESM.pdf berikut. Penelitian ini bertujuan untuk mengidentifikasi protein pengikat DNA dengan mengkombinasikan komposisi pseudo amino acid dan berbasis profil representasi protein. Metode yang digunakan ialah Support Vector Machine dengan nama predictor iDNAPro-PseACC. Hasil yang didapatkan ialah akurasi 76,76%, sensitivitas 75,62%, spesifisitas 77,45%, dan MCC 53%.

2.1.2. Local-DPP: an improved DNA-binding protein prediction method by exploring local evolutionary information

Penelitian ini dilakukan oleh Wei, Tang, dan Zou yang diterbitkan pada tahun 2016. Menggunakan *dataset benchmark* PDB1075 dengan 525 DNA-*binding proteins (positive samples)* dan 550 *non* DNA-*binding proteins (negative samples)*.

Menggunakan *jackknife* dan pengklasifikasian dengan *random forest* didapatkan sebuah predictor dengan nama Local-DPP serta hasil yang didapatkan ialah akurasi 79,20%, sensitivitas 84%, spesifisitas 74,50%, dan MCC 59%.

2.1.3. iDNAProt-ES: identification of DNA-binding protein using evolutionary and structural features

Penelitian ini dilakukan oleh Chowdhury, Shatabda, dan Dehzangi yang diterbitkan pada tahun 2017. Menggunakan *dataset benchmark*, PDB1075

dengan 525 DNA-*binding proteins (positive samples)* dan 550 *non* DNA-*binding proteins (negative samples)*.

Menggunakan *jackknife* dan pengklasifikasian dengan SVM didapatkan sebuah predictor dengan nama iDNAProt-ES serta hasil yang didapatkan sebesar akurasi 90,18%, sensitivitas 90,38%, spesifisitas 90%, dan MCC 80%.

2.1.4. DPP-PseAAC: a DNA-binding protein prediction model using Chou's general PseAAC

Penelitian ini dilakukan oleh Rahman yang diterbitkan pada tahun 2018. Menggunakan *dataset benchmark* PDB1075 dengan 525 DNA-*binding proteins (positive samples)* dan 550 *non* DNA-*binding proteins (negative samples)*.

Menggunakan *jackknife* dan pengklasifikasian dengan SVM didapatkan sebuah predictor dengan nama DPP-PseAAC dengan hasil yang diperoleh sebesar akurasi 95,91%, sensitivitas 94,10%, spesifistias 97,64%, dan MCC 92%.

2.2. Protein *Sequence*

Protein adalah urutan asam amino yang terdapat 20 jenis, dan setiap asam amino memiliki kode satu huruf diantaranya A, C, D…. Contohnya seperti DYMQKREVDLHN yang dimana ini mewakili urutan protein (Mardia, 2013).

Protein adalah makromolekul yang secara fisik dan fungsional kompleks yang melakukan beragam peran penting. Protein mengalami beberapa perubahan, yaitu perubahan fisik dan fungsional. Dimana masing-masing perubahan itu mencerminkan siklus hidup organisme tempat protein itu berada (Murray, 2014).

Protein tersusun atas rantai-rantai asam amino yang terdiri dari unsur-unsur karbon (C), hidrogen (H), oksigen (O), dan nitrogen (N) yang tidak memiliki lemak atau karbohidrat.

Interaksi protein memainkan peran kunci dalam berbagai proses dan fungsi biologis. Dalam sel-sel hidup, termasuk siklus metabolisme, transkripsi DNA

dan replikasi, dan kaskade sinyal. Dengan demikian, mengidentifikasi dan mengkarakterisasi interaksi protein dengan benar sangat penting untuk memahami mekanisme molekuler di dalam sel (You et al., 2015).

Struktur protein sangat penting untuk fungsinya, dan oleh karena itu, fitur struktur yang diprediksi banyak digunakan untuk menganalisis fungsi protein, seperti interaksi protein-protein, prediksi situs dan identifikasi protein yang tidak teratur (Liu, 2018).

2.3. *Preprocessing Data*

Preprocessing data adalah proses mengubah data ke dalam sebuah format yang mudah dipahami, lebih efektif, dan bisa dibilang sesuai dengan kebutuhan pengguna. Indikator yang dapat digunakan sebagai referensi adalah hasil yang akurat, waktu komputasi yang lebih pendek, juga data menjadi lebih kecil tanpa mengubah informasi di dalamnya (Saifullah et al., 2017).

Preprocessing itu sendiri adalah teknik paling awal saat kita ingin mengubah data mentah dari sumber yang kita dapatkan menjadi sebuah informasi yang bisa dimengerti dan bisa digunakan untuk keperluan selanjutnya.

2.3.1. Pemotongan Data

Dataset yang didapat berupa *sequences protein* yang dimana panjang dari setiap rantai itu berbeda-beda. Maka dari itu dilakukannya pemotongan data menjadi 50 panjang *sequences*. Tujuan memotong menjadi 50 panjang *sequences* saja ialah berdasarkan penelitian Liu et al., 2015 dimana *sequences* terpendek dikurangi satu ($L - 1$). L adalah panjang *sequences* terpendek.

2.4. *Feature Extraction*

Feature extraction merupakan masalah penting dalam klasifikasi data dengan dimensi besar. Tujuan dari *feature extraction* adalah untuk memproses dalam mengklasifikasikan suatu objek untuk menemukan pemetaan fitur yang asli

kedalam fitur yang baru, sementara untuk tetap mempertahankan karakteristik data asli yang cukup untuk mengklasifikasikan data (Annisa, 2017). *Feature extraction* pada penelitian ini berfungsi untuk membuat sebuah data tidak terstruktur menjadi data terstruktur dengan maksud, yaitu membuat kolom menjadi sama jumlahnya. Ini bertujuan karena rata-rata algoritma membutuhkan kolom yang sama agar klasifikasi bisa berjalan dan diproses sesuai data terstruktur yang sudah ada. Dalam penelitian ini, *feature extraction* yang digunakan protein deskriptor menggunakan *package* BioSeqClass versi 1.44.0.

Protein deskriptor adalah *tools* untuk melakukan *feature extraction* pada sebuah protein *sequences*. Terdapat dua jenis utama protein deskriptor berdasarkan urutan asam amino dari subjek protein, yaitu *protein-level global descriptor* dan *amino acid-level descriptors*. *Protein-level global descriptor* merangkum dari ciri-ciri protein sedangkan *amino acid-level descriptors* menggambarkan sifat asam amino di sepanjang urutan (Xu et al., 2020).

Protein deskriptor yang digunakan untuk melakukan *feature extraction* ini berdasarkan dari *static length* dan *dynamic length*. *Static length* adalah protein deskriptor yang nilainya (fitur) tidak berubah sepanjang apapun *sequences*-nya. *Feature extraction* pada *package* BioSeqClass yang digunakan pada penelitian ini adalah *featureCTD*. Misal terdapat *sequences* “KRRAI” dan “GGGARR” dengan nilai setiap amino acids nya (*numeric*) ditentukan sesuai jumlah fitur yang terdapat pada *featureCTD*, yaitu 21 fitur dan tidak dipengaruhi panjangnya. *Dynamic length* adalah protein deskriptor yang nilainya (fitur) berubah-ubah sesuai panjang, urutan dan parameter yang digunakan. *Package* BioSeqClass yang menggunakan *dynamic length* untuk melakukan *feature extraction* pada penelitian ini adalah *featureAAIndex* dan *featurePseudoAAComp*. Contohnya terdapat *sequences* “GMAFA” dan “RMAAGI” yang dimana *sequences* ini akan menghasilkan output berupa *numeric* yang berbeda-beda sesuai dari urutan, panjang dan parameter yang digunakan. Berikut penjelasan dari *feature extraction* yang digunakan.

2.4.1. AAIndex

AAIndex adalah kumpulan database yang berisi informasi asam amino. AAindex dapat digunakan dalam kombinasi dengan analisis kemometrik dan QSAR, pola kesamaan, simulasi prediksi dalam percobaan docking (Carrasco-Castilla et al., 2012). AAIndex memiliki website dimana tampilannya seperti Gambar 1.

Gambar 1. Website database AAIndex (https://www.genome.jp/aaindex/).

AAIndex memiliki rumus yang dapat dilihat pada Persamaan 1.

$$aaindex(i) = \frac{\sum_{n=1}^{N} AAIndex_i(aa_n)}{N} \quad \text{.....(1)}$$

Di mana i menunjukkan x indeks $AAIndex_i$, aa_n menunjukkan asam amino pada posisi n dan N adalah jumlah residu dalam urutan. Untuk saat ini, AAIndex memiliki 3 bagian, yaitu AAIndex1 yang digunakan memiliki indeks asam amino dari 20 nilai *numeric* dengan jumlah keseluruhan sebanyak 544 indeks asam amino (untuk versi 9.2 saat ini sudah memiliki 566 indeks asam amino berdasarkan website database AAIndex), AAIndex2 berfungsi untuk matriks mutasi asam amino yang memiliki 67 matriks simetris dan 27 matriks non-simetris dan AAIndex3 untuk potensi kontak protein statistik yang memiliki 47 matriks asam amino potensial, terdiri dari 44 matriks simetris dan 3 matriks non-simetris. Semua data berasal dari literatur yang diterbitkan (Kawashima et al., 2008). AAIndex yang digunakan pada *feature extraction* ini, yaitu AAIndex1 yang bertujuan untuk memberikan informasi mengenai urutan *sequences* yang diberikan.

AAIndex1 yang digunakan saat ini memiliki 544 indeks asam amino. Setiap indeks memiliki nomor aksesi, deskripsi singkat terhadap indeksnya, informasi referensi, dan nilai *numeric* untuk sifat 20 asam amino. AAIndex memiliki parameter yang berfungsi untuk memanggil properti nama sifat fisikokimia dan biokimia. Jika yang digunakan parameter aaindex.name=”all”, maka seluruh nama sifatnya akan dipertimbangkan serta setiap baris mewakili fitur dari satu urutan. Percobaan parameter (aaindex.name) yang digunakan dalam penelitian ini memakai 7 sifat fisikokimia dan biokimia dalam *amino acids* sesuai dengan penelitian terdahulu antara lain, “NOZY710101”, “HOPT810101”, “TSAJ990101”, “CHOC760101”, “PRAM820101”, “GEIM800101”, dan “FINA770101”. Berikut kode program AAIndex1 dapat dilihat pada Pseudocode 1.

```
featureAAindex(seq,aaindex.name=”all”)
```

Pseudocode 1. Kode program *featureAAindex*.

Keterangan dari Pseudocode 1.

seq : vektor string untuk urutan protein, DNA, atau RNA.

aaindex.name : string nama sifat fisikokimia dan biokimia di AAIndex.

Diberikan contoh *sequences* berikut.

Input : “ATFEIVNRCS” dan “SNMWVIGKSK”

Output dapat dilihat setelah diolah menggunakan Pseudocode 1 dengan menggunakan aaindex.name=“NOZY710101” pada Tabel 2.

Tabel 2. Contoh *output* dari *featureAAindex*

Output	**NOZY710101_1**	**NOZY710101_2**	**NOZY710101_n**
ATFEIVNRCS	0.5	0.4	…
SNMWVIGKSK	0	0	…

Keterangan dari Tabel 2.

NOZY710101 : nama sifat aaindex1 yang digunakan.

n : informasi mengenai panjang dan urutan *sequences* yang diberikan.

Berikut *input* dan *output* jika memberikan panjang *sequences*-nya 20 dan 50 dapat dilihat pada Tabel 3.

Tabel 3. Perbandingan hasil *featureAAindex* jika panjang *sequences* berbeda

***Input Sequences* (File)**	***Output***
AI_20	Num [1:525, 1:20]
AI_50	Num [1:525, 1:50]

Keterangan dari Tabel 3.

AI_20 dan AI_50 : 20 dan 50 adalah panjang *sequences* yang diberikan.

num : data bertipe *numeric*.

1:525 : 525 baris data positif yang diberikan.

1:20 dan 1:50 : jumlah kolom sesuai panjang *sequences* yang diberikan.

Dengan ini, bahwa AAIndex akan menghasilkan fitur sesuai urutan dan panjang *sequences* yang diberikan.

Dataset yang dimiliki mempunyai panjang *sequences* yang berbeda-beda dan dikarenakan AAIndex1 memiliki ketentuan harus mempunyai panjang yang sama setiap *sequencesnya* serta mengikuti rumus *L*-1 dari penelitian terdahulu bahwa panjang *sequences* terpendek dikurangi 1. Jadi panjang *sequences* yang digunakan adalah 50. Dan aaindex.name yang digunakan terdapat 7 properti, jadi fitur berjumlah 7 * 50 = 350 fitur.

2.4.2. *Composition, Transition and Distribution* (CTD)

Composition (C) adalah jumlah asam amino dari properti tertentu dibagi dengan total jumlah asam amino. *Transition* (T) mendeskripsikan frekuensi persen dengan mana asam amino dari properti tertentu diikuti oleh asam amino dari properti yang berbeda. *Distribution* (D) mengitung panjang rantai di mana yang pertama, 25, 50, 75 dan 100% dari asam amino dari properti tertentu terletak masing-masing (Govindan & Nair, 2011).

Composition memiliki rumus yang dapat dilihat pada Persamaan 2.

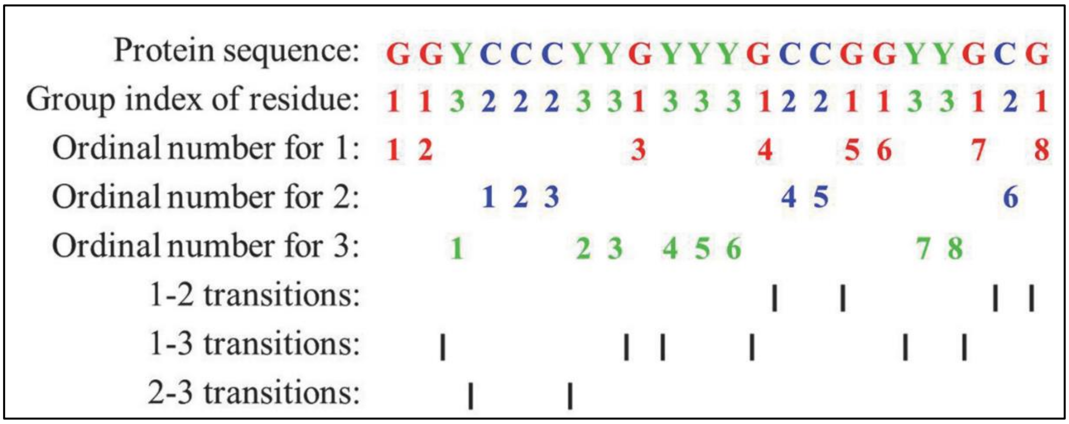
$$C(i) = \frac{ni}{N} \quad i = 1, 2, 3 \qquad (2)$$

Di mana ni menunjukkan frekuensi grup i dalam barisan, dan N adalah total panjang barisan.

Transition memiliki rumus yang dapat dilihat pada Persamaan 3.

$$T(ij) = \frac{nij+nji}{N-1} \quad ij = 12, 13, 23 \qquad (3)$$

Di mana nij dan nji masing-masing menunjukkan frekuensi transisi dari grup i ke j dan j ke i dalam barisan dan N adalah panjang barisan total.



Gambar 2. Contoh dari CTD (You et al., 2015).

CTD memiliki kode program yang dapat dilihat pada Pseudocode 2.

```
featureCTD(seq,class=aaclass("aaV"))

featureCTD(seq,class=elements("aminoacid"))
```

Pseudocode 2. Kode program *featureCTD*.

Keterangan dari Pseudocode 2.

seq : vektor string untuk urutan protein, DNA, atau RNA.

class : daftar untuk kelas sifat biologi, yang terdiri dari *aaclass* dan *elements*.

aaclass : daftar kelompok asam amino tergantung pada sifat fisik kimianya.

elements : daftar elemen dasar urutan biologis.

Diberikan contoh *sequences* berikut.

Input : "KKEKSPKGKS" dan "MIKLRMPAGG".

Output dapat dilihat setelah diolah menggunakan kode program Pseudocode 2 yang hanya menggunakan *aaclass*("aaV") pada Tabel 4.

Tabel 4. Contoh *output* dari *featureCTD*

Output	**C_small**	**T_small_medium**	**D_small_1st**	***x_y_z***
KKEKSPKGKS	0.33	0	0.56	…
MIKLRMPAGG	0.33	0	0.78	…

Keterangan dari Tabel 4.

C,T,D : Composition, Transition, Distribution.

x : C atau T atau D.

y : *small* atau *medium* atau *large*.

z : keterangan tambahan jika *x* adalah T atau D.

Berikut *input* dan *output* jika memberikan panjang *sequences*-nya 20 dan 50 dapat dilihat pada Tabel 5.

Tabel 5. Perbandingan hasil *featureCTD* jika panjang *sequences* berbeda

***Input Sequences* (File)**	***Output***
CTD_20	Num [1:525, 1:21]
CTD_50	Num [1:525, 1:21]

Berikut keterangan Tabel 5.

CTD_20 dan CTD_50 : 20 dan 50 adalah panjang *sequences* yang diberikan.

num : data bertipe *numeric*.

1:525 : 525 baris data positif yang diberikan.

1:21 : jumlah kolom sesuai panjang *sequences* yang diberikan.

Dengan ini, bahwa CTD akan menghasilkan jumlah fitur yang sama yaitu 21 fitur, walaupun panjang *sequences* yang diberikan berbeda.

Dataset yang dimiliki mempunyai panjang *sequences* yang berbeda-beda dan dikarenakan AAIndex1 memiliki ketentuan harus mempunyai panjang yang sama setiap *sequencesnya* serta mengikuti rumus *L*-1 dari penelitian terdahulu bahwa panjang *sequences* terpendek dikurangi 1, *featureCTD* mengikuti agar panjang *sequences* sama. Jadi panjang *sequences* yang digunakan adalah 50. Dan aaclass yang digunakan, yaitu "aaV", jadi fitur berjumlah 21 fitur. Fitur yang dihasilkan tidak dipengaruhi oleh panjang *sequences*.

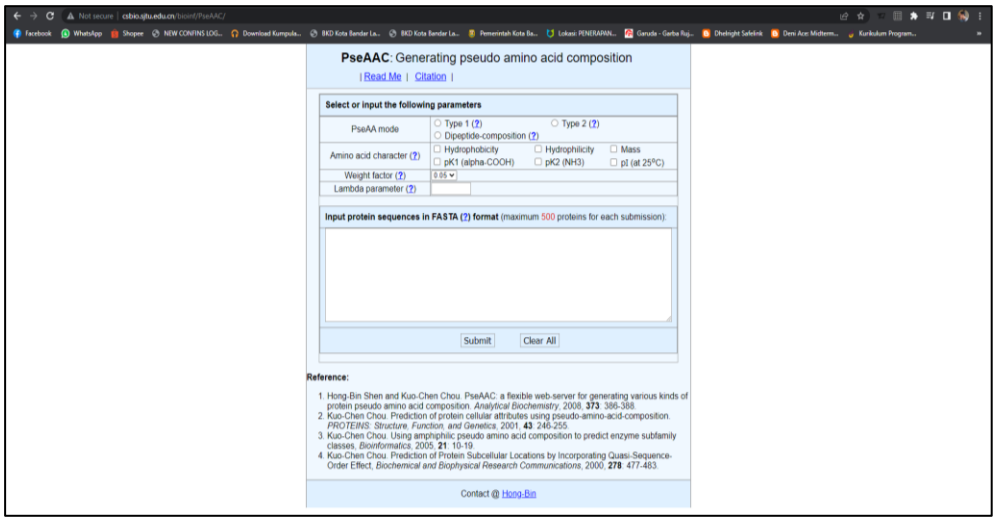
2.4.3. *Pseudo Amino Acid Composition* (PseAAC*)*

PseAAC pertama kali diusulkan oleh Chou pada tahun 2001. Pseudo Amino Acid Composition (PseAAC) adalah suatu metode yang berfungsi sebagai alat untuk merumuskan urutan biologis dengan vektor, tetapi masih menyimpan informasi urutan-urutannya. PseAAC protein berisi lebih 20 faktor yang berbeda. 20 faktor pertama terkait dengan komponen-komponennya seperti komposisi asam amino konservatif, sedangkan faktor tambahan menggabungkan informasi urutan-urutan mereka melalui berbagai metode. Tiga faktor yang sering digunakan untuk menghasilkan berbagai jenis PseAAC seperti komposisi asam amino, faktor bobot dan tingkat korelasi. PseAAC mendukung enam fitur asam amino, yaitu hidrofobisitas, hidrofilisitas, massa rantai samping, pK1 (alfa-COOH), pK2 (NH3) dan pI

(Titik isoelektrik). Karakter-karakter ini diterapkan untuk menilai efek dari perbedaan lokasi asam amino di sepanjang urutan peptide (Chou, 2001).

Peringkat atau tingkatan yang dihitung dari korelasi sebuah panjang *sequences* biasanya diwakili oleh λ (lambda). Dengan catatan, λ harus lebih kecil dari panjang *sequences* yang diinputkan, λ harus bilangan bulat positif, dan saat λ bernilai 0, maka *output* dari PseAAC mengalami degenerasi menjadi komposisi AA konvensional diskrit (Shen & Chou, 2008).

PseAAC memiliki 3 tipe. Tipe pertama PseAAC atau *parallel correlation type*. Hasi *output* dari tipe pertama ini mempunyai (20 + λ) nomor diskrit. Tipe kedua PseAAC atau *the series corelation type*. Hasil *output* dari tipe kedua ini mempunyai (20 + Ɛ x λ) nomor diskrit, dimana Ɛ adalah jumlah karakter AA yang dipilih. Tipe ketiga PseAAC atau *the dipeptide type*. Hasil *output* dari tipe ketiga ini mempunyai (20 + 400 = 420) nomor diskrit (Shen & Chou, 2008). PseAAC memiliki web server sendiri yang dapat dilihat pada Gambar 3.



Gambar 3. *Web server generating* PseAAC (http://www.csbio.sjtu.edu.cn/bioinf/PseAAC/).

Penelitian ini hanya menggunakan PseAAC tipe pertama karena pada *package* BioSeqClass 1.44.0 hanya mempunyai tipe pertama saja dengan nama *featurePseudoAAComp*. 20 faktor dari *amino acids* adalah "RKEDQNWGASTPHYCVLIMF". Kode program *featurePseudoAAComp* dapat dilihat pada Pseudocode 3.

```
featurePseudoAAComp(seq,d)
```

Pseudocode 3. Kode program *featurePseudoAAComp*.

Keterangan dari Pseudocode 3.

seq : vektor string untuk urutan protein, DNA, atau RNA.

d : parameter bertipe integer dengan nilai (d>=1). Nilai maksimalnya tidak boleh sama/melebihi panjang *sequences* yang diberikan.

Diberikan contoh *sequences* berikut.

Input : "KKEKSPKGKS" dan "MIKLRMPAGG".

Output dapat dilihat setelah diolah menggunakan kode program Pseudocode 3 dengan menggunakan nilai *d*=9 pada Tabel 6.

Tabel 6. Contoh *output* dari *featurePseudoAAComp*

Output	**PAC:R**	**PAC:K**	**PAC:*x***	**PAC:1**	**PAC:*y***
KKEKSPKGKS	0	0.28	…	0.04	…
MIKLRMPAGG	0.05	0.05	…	0.02	…

Keterangan dari Tabel 6.

x : 20 faktor pertama PseAAC.

y: integer yang digunakan dan tidak boleh sama atau melebihi panjang *sequences*.

Berikut *input* dan *output* jika memberikan panjang *sequences*-nya 20 dan 50 dapat dilihat pada Tabel 7.

Tabel 7. Perbandingan hasil *featurePseudoAAComp* jika panjang *sequences* berbeda

***Input Sequences* (File)**	***Output***
PseAAC_20	Num [1:525, 1:39]
PseAAC_50	Num [1:525, 1:69]

Keterangan dari Tabel 7.

PseAAC_20 dan PseAAC_50	: 20 dan 50 panjang *sequences* yang diberikan.
num	: data bertipe *numeric*.
1:525	: 525 baris data positif yang diberikan.
1:39	: 39 kolom dengan *input* 20 panjang *sequences* (20 +19).
1:69	: 69 kolom dengan *input* 50 panjang *sequences* (20 + 49).

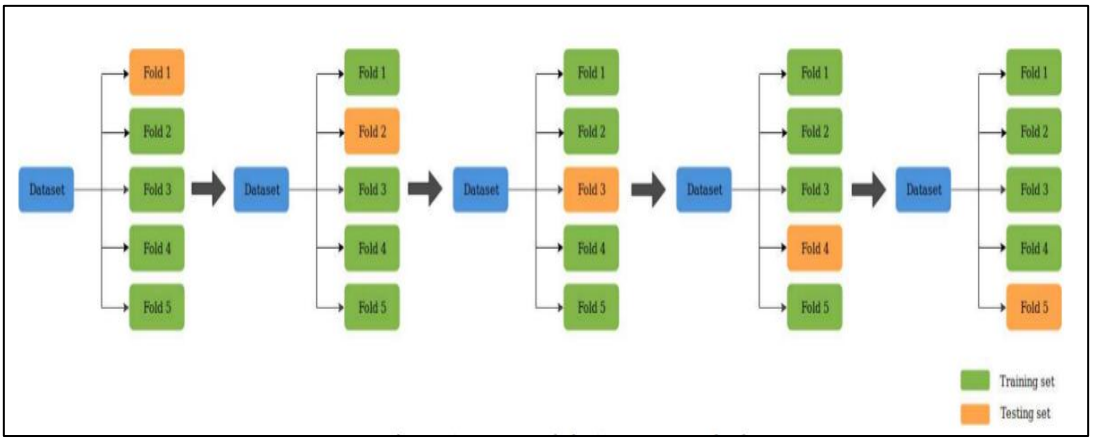
Dapat disimpulkan, PseAAC memiliki jumlah fitur berbeda-beda tergantung dari panjang *sequences* yang diberikan.

Dataset yang dimiliki mempunyai panjang *sequences* yang berbeda-beda dan dikarenakan AAIndex1 memiliki ketentuan harus mempunyai panjang yang sama setiap *sequencesnya* serta mengikuti rumus *L*-1 dari penelitian terdahulu bahwa panjang *sequences* terpendek dikurangi 1, *featurePseudoAAComp* mengikuti agar panjang *sequences* sama. Jadi panjang *sequences* yang digunakan adalah 50. Dan parameter *d* nya adalah 49, jadi fitur berjumlah (20 + 49 = 69 fitur) .

Jika digabungkan, jumlah keseluruhan fitur ekstrasinya ialah sebanyak 440 fitur.

2.5. *Cross-Validation*

Cross-Validation adalah teknik yang ditetapkan untuk memperkirakan keakuratan sebuah pengklasifikasi dan biasanya dilakukan menggunakan sejumlah data *test/train* secara acak, atau bisa juga menggunakan *k-fold cross-validation* (Mullin & Sukthankar, 2000). *k-fold cross-validation* adalah sebuah metode dimana data pertama kali dipartisi menjadi segmen yang hampir berukuran sama. Selanjutnya *k* dari *training* dan *validation* dilakukan seterusnya sehingga setiap *fold* bertahan untuk validasi, sedangkan sisanya *k*-1 *fold* digunakan untuk pembelajaran (Refaeilzadeh et al., 2016). Kumpulan data dibagi menjadi 5 fold. Pada iterasi pertama, fold pertama digunakan untuk menguji model dan sisanya digunakan untuk melatih model. Pada iterasi kedua, fold kedua digunakan sebagai set pengujian sementara sisanya berfungsi sebagai set pelatihan. Proses ini diulangi sampai setiap *fold* dari k-*fold* telah digunakan sebagai set pengujian.



Gambar 4. Prosedur dari *fold cross-validation* (Peryanto, Yudhana, & Umar, 2020).

2.6. *Random Forest*

Random forest ini terdiri dari kumpulan dari *decision tree* (*ensemble*). DT memiliki 2 jenis pohon, yaitu *classification tree* dan *regression tree*. *Classification tree* memiliki variabel bertipe kategorik sedangkan *regression tree* memiliki variabel bertipe numerik. *Decision tree* bisa terjadi suatu kondisi, yaitu overlap jika kelas yang digunakan sangat banyak. Dalam hal akumulasi, sering terjadi *error* dalam jumlah besar dan juga DT kesulitan dalam

mendesain pohon yang optimal. Untuk mengatasi itu dibutuhkannya *random forest* untuk mengatasi *overlap* tersebut (Salman & Galih, 2020).

Random forest adalah satu set sejumlah tertentu pohon *random* yang menghasilkan *forest* (hutan; kumpulan pohon) acak. Model yang dihasilkan adalah model suara pilihan dari semua pohon. Operator *random forest* menghasilkan satu set pohon acak. Model hutan yang dihasilkan mengandung sejumlah model pohon acak. Jumlah pohon parameter menentukan jumlah yang diperlukan pohon. Model yang dihasilkan adalah model pilihan dari semua pohon acak (Saifullah et al., 2017).

Model *random forest* adalah algoritma klasifikasi *ensemble* (kumpulan dari pohon keputusan) yang menggunakan koleksi pohon keputusan untuk mengurangi varians output dari pohon individu dan dengan demikian meningkatkan stabilitas dan akurasi klasifikasi. Model *random forest* saat ini merupakan salah satu teknik pembelajaran mesin yang paling sering digunakan (You et al., 2015).

Disebut *random forest* karena itu adalah hutan pohon keputusan yang dibuat secara acak. Setiap node di pohon keputusan bekerja pada subset fitur acak untuk menghitung output. *Random forest* kemudian menggabungkan output dari *decision trees* untuk menghasilkan output akhir.

Random forest menggunakan *decision trees* sebagai pengklasifikasi dasar. Berdasarkan dari mengumpulkan dan ada 2 cara menggunakannya, yaitu satu dalam pemilihan sampel data pelatihan untuk melatih setiap pohon dasar dan yang lainnya melakukan melakukan pemilihan atribut untuk induksi pohon (Kulkarni et al., 2015).

Dalam *random forest* terdapat sebuah penambahan pada *random sub sampling* atau pemilihan *m* variabel yang digunakan dalam membangun pohon. Proses ini sama seperti *decision tree* hanya saja tidak melakukan *pruning* (pemangkasan).

Strength (kekuatan) adalah rata-rata atau ekspektasi ukuran kekuatan akurasi pohon tunggal. Nilai *s* yang semakin besar tandanya bahwa akurasi semakin

membaik. Nilai *s* memiliki rumus yang dapat dilihat di Persamaan 4 (Breiman, 2001).

$$s = E_{X,Y} mg(X,Y) \quad \text{.....(4)}$$

Rata-rata korelasi ($\bar{p}$) antar pasangan dugaan dari dua pohon tunggal dalam *random forest* dapat dilihat di Persamaan 5 (Breiman, 2001).

$$\bar{p} = \frac{E_{\Theta,\Theta'}\left(p(\Theta,\Theta')sd(\Theta)(\Theta')\right)}{E_{\Theta,\Theta'}(sd(\Theta)sd(\Theta))} \quad \text{.....(5)}$$

Dimana *p(Θ,Θ')sd(Θ)sd(Θ')* merupakan korelasi antar pohon.

Batasan besarnya kesalahan *(*Σ_{RF}*)* oleh *random forest* dapat dilihat di Persamaan 6.

$$\Sigma_{RF} \leq \bar{p}\left(\frac{1-s^2}{s^2}\right) \quad \text{.....(6)}$$

Mtry adalah jumlah berbagai prediktor untuk dicoba pada setiap *node* dan *node size* adalah ukuran minimum terminal *node*. *Ntree* adalah jumlah *regression tree* (nilai bawaannya adalah 500 pohon) (Azqhandi et al., 2017). Untuk mendapatkan *ntree* yang optimal, pohon harus dibangun sampai errornya kecil dengan cara memiliki korelasi kecil dan memperkuat *strength* (Fadilah, 2018). Pemilihan *mtry* dan *ntree* sangat berpengaruh dalam *random forest*. Sebagai contoh, jika *mtry* besar, kemungkinan pada setiap generasi, output pohon regresi mungkin tumbuh terlalu besar.

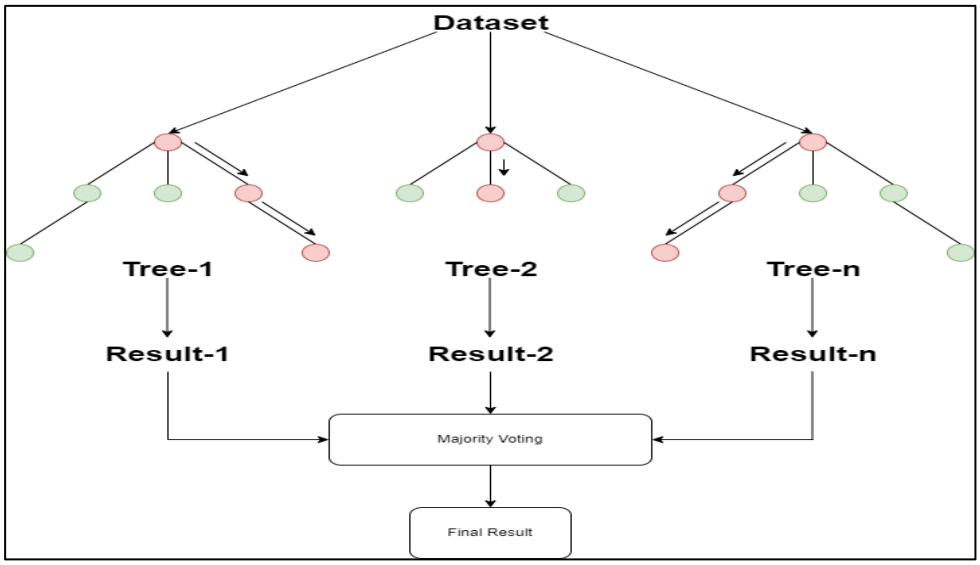
Berikut ini adalah algoritma *random forest* (Hestie et al., 2008).

1. Buat suatu pengambilan sampel *Z* dengan pengembalian (*replacement*) dari suatu ukuran *N* dari gugus data.
2. Pilih *m* variabel secara acak dari *p* variabel, dimana $m \leq p$. Biasanya ukuran *m* terbaik dipilih dari akar kuadrat dari total jumlah *p* variabel, yaitu $|\sqrt{p}|$. Menurut Breiman, nilai *m* juga dapat diperoleh dari 2x nilai akar kuadrat dari total jumlah *p* variabel yang dapat dilihat di Persamaan 7.
   $m = 2|\sqrt{p}|$ .....(7)

dan setengah dari nilai akar kuadrat dari total jumlah *p* variabel yang dapat dilihat di Persamaan 8.

$$m = \frac{1}{2}|\sqrt{p}| \quad \text{.....(8)}$$

3. Selesai melakukan pemilihan *m* secara acak, selanjutnya pohon ditumbuhkan tanpa pemangkasan (*pruning*).
4. Langkah 1-3 dilakukan sebanyak *n* kali hingga terbentuk suatu *forest* sebanyak *n* pohon.
5. Setelah *forest* terbentuk, selanjutnya mencari nilai misklasifikasi error (*out of bag error*) untuk mendapatkan *mtry* optimal dan diperoleh tingkat kepentingan variabel (*variable importance*) yang lebih stabil.
6. Untuk prediksi suatu kelas dilakukan dengan *majority vote* (suara terbanyak).



Gambar 5. Contoh *random forest*.

Random forest menggunakan gini index untuk menentukan kelas akhir disetiap pohon. Pada final class setiap pohon dikumpulkan serta dipilih oleh nilai-nilai weight untuk membangun *classifier* akhir. *Random forest* menggunakan gini index diambil dari sistem CART untuk membangun *decision trees*. Berikut gini index dapat dilihat pada Persamaan 9.

$$Gini(T) = 1 - \sum_{i=1}^{m}(\frac{T_i}{T})^2 \quad \text{.....(9)}$$

Jika T dataset dibagi menjadi 2 himpunan bagian yaitu T1 dan T2 dengan ukuran M1 dan M2 masing-masing, index pada split data berisi contoh-contoh dari kelas m, index gini (T) didefinisikan pada Persamaan 10.

$$Gini(T) = \frac{M_1}{M} gini(T_1) + \frac{M_2}{M} gini(T_2) \quad \text{.....(10)}$$

Berikut contoh perhitungan *random forest*.

Terdapat dataset yang singkat dengan menunjukkan pembangunan *single tree*. Terdapat 2 atribut, yaitu *average_value* dan *description*. Dapat dilihat pada Tabel 8.

Tabel 8. Contoh soal *random forest*

Record	Atribut		Class
	average_value	description	
1	34	3	1
2	22	3	0
3	5	1	1
4	17	4	1
5	10	1	0

Asumsikan atribut pertama yang akan di split adalah *average_value*. Split yang mungkin untuk *average_value* dalam rentang *node* kiri ialah $5<x<34$, di mana *x* adalah nilai split. Seluruh nilai lain pada setiap dari *node* anak kanan. Berikut split untuk atribut *average_value* dalam dataset adalah.

- *average_value* < 5
- *average_value* < 10
- *average_value* < 17
- *average_value* < 22
- *average_value* < 34

Ambil split pertama untuk melakukan perhitungan yang dapat dilihat pada Tabel 9.

Tabel 9. Perhitungan soal *random forest*

Atribut	Number Of Record		N= 5
	0	**1**	
Average_value* <= *5	0	1	N1= 1
Average_value* > *5	2	2	N2 = 4

Kemudian Gini T1, T2, dan gini split dapat dihitung pada Persamaan 11, 12, dan 13.

$$Gini(average\_value \leq 5) = 1 - \left(\left(\frac{0}{1}\right)^2 + \left(\frac{1}{1}\right)^2\right) = 0 \text{.....(11)}$$

$$Gini(average\_value > 5) = 1 - \left(\left(\frac{2}{4}\right)^2 + \left(\frac{2}{4}\right)^2\right) = 0{,}5 \text{.....(12)}$$

$$Gini\ Split = \left(\frac{1}{5}\right)\ x\ 0 + \left(\frac{4}{5}\right)\ x\ 0{,}5 = 0{,}4 \text{.....(13)}$$

Lakukan tahap tersebut untuk mencari gini split <10, <17, <22, dan <34. Sehingga akan mendapatkan keseluruhan gini split yang dapat dilihat pada Tabel 10.

Tabel 10. Nilai gini split pada setiap kemungkinan

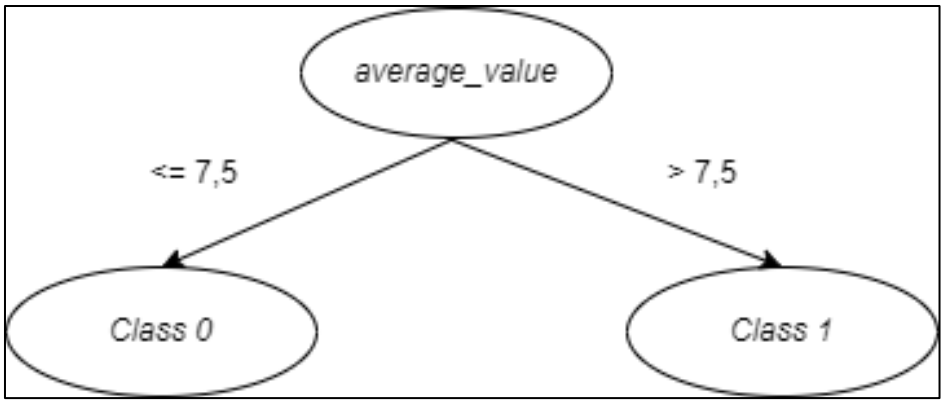
Gini Split	Value
***average_value* <= 5**	**0,4**
average_value <= 10	0,466
average_value <=17	0,466
average_value <= 22	0,41
average_value <=34	0,52

Dari Tabel 10 didapatkan Gini Index terendah pada *average_value* <= 5, yaitu 0,4. Dalam *random forest*, split dimana gini index terendah dipilih pada nilai split, namun karena nilai-nilai atribut *average_value* bersifat *continue*, titik

tengah setiap pasangan nilai berturut dipilih sebagai titik *best* split. *Best* split pada contoh ini dapat dilihat pada Persamaan 14.

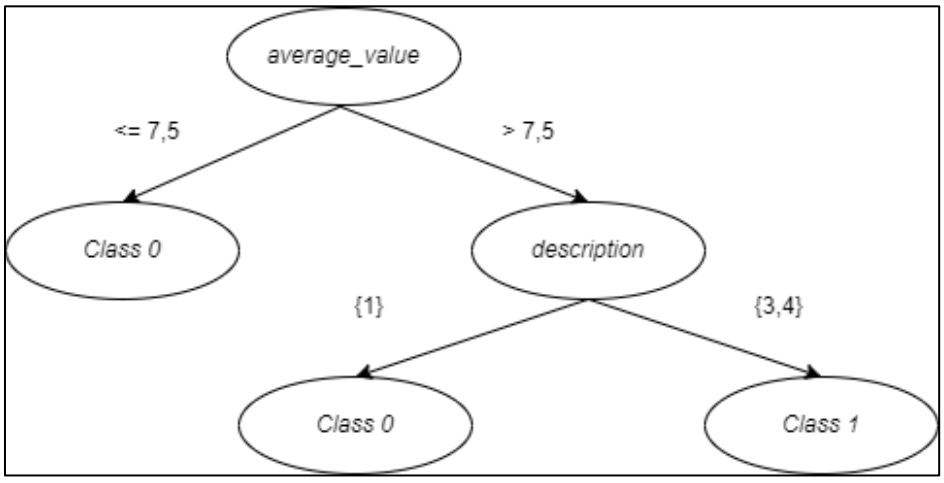
$$average\_value = \left(\frac{5+10}{2}\right) = 7{,}5 \quad \text{.....(14)}$$

Jadi *best* split bukan pada *average_value* <= 5. Berikut *decision tree* yang pertama dapat dilihat pada Gambar 6.



Gambar 6. *Tree* soal *random forest* variabel pertama

Langkah-langkah yang telah dilakukan, diulangi kembali untuk menentukan sisa atribut pada dataset yang ada untuk mencari Gini Index dan Gini Split. Sehingga *decision trees* akhir akan ditunjukkan seperti *tree* pada Gambar 7.



Gambar 7. *Tree* soal *random forest* seluruh variabel

2.6.1. Tingkat Misklasifikasi (*Out of Bag Error*)

Salah satu fitur yang ada didalam *random forest* adalah *out-of-bag* (OOB). OOB ialah sepertiga amatan gugus data asli yang tidak termuat dalam contoh pada setiap iterasinya. Error OOB bergantung pada korelasi antar pohon dan kekuatan (*strength*) masing-masing pohon dalam *random forest* dimana peningkatan korelasi dapat meningkatkan error OOB sedangkan peningkatan pohon dapat menurunkan error OOB. Error OOB dihitung dari proporsi misklasifikasi hasil *random forest* dari seluruh data asli (Breiman, 2001).

2.6.2. *Variable Importance*

Salah satu tingkat kepentingan variabel bebas yang dihasilkan oleh *random forest* adalah *Mean Decrease Accuracy* (MDA), dimana MDA memperlihatkan besar tambahan observasi yang mengalami misklasifikasi jika satu persatu variabel bebas tidak diikutsertakan dalam pengujian. Adapun namanya *Mean Decrease Gini* (MDG), yaitu salah satu ukuran tingkat kepentingan peubah penjelas yang dihasilkan metode *random forest* (Breiman & Cutler., 2003). Misalkan terdapat *p* peubah penjelas dengan h = 1,2,…,*p* maka MDG mengukur tingkat kepentingan peubah penjelas X_h dapat dilihat pada Persamaan 14.

$$MDG_h = \frac{1}{k}\sum_t [d(h,t)I(h,t)] \quad \text{.....(14)}$$

Keterangan dari Persamaan 14.

d(h,t) : besar penurunan indeks Gini untuk peubah penjelas X_h.

I(h,t) : memilah simpul *t.*

k : banyaknya pohon dalam *random forest.*

2.6.2.1. Mean Decrease Accuracy (MDA)

MDA berfungsi untuk seberapa besar akurasi model yang hilang dengan mengecualikan setiap variabel. Semakin mendapatkan akurasi, semakin penting variabel untuk klasifikasi yang sukses. Variabel disajikan dari tingkat kepentingan yang menurun.

2.6.2.2. Gini Impurity Index (GII)

Didalam *random forest*, terdapat sebuah fitur untuk memperlihatkan fitur-fitur yang mempengaruhi hasil akhir dari sebuah klasifikasi. Gini impurity index digunakan untuk menentukan *variable importance*. GII bekerja dengan cara memilih fitur secara acak dari kumpulan data yang akan diklasifikasikan secara tidak benar, jika diklasifikasikan secara acak berdasarkan distribusi kelas dalam subset (Lumbanraja et al., 2018). Dalam penelitian ini digunakannya GII untuk me-*rangkin*g 20 fitur teratas yang digunakan dalam klasifikasi *random forest*.

2.7. Evaluasi

Tujuan evaluasi adalah untuk melihat hasil akhir dari penelitian yang dilakukan. Berikut metode yang digunakan.

2.7.1. *Confusion Matrix*

Confusion Matrix adalah sebuah evaluasi untuk pengukuran akurasi dari penelitian yang kita teliti.

Tabel 11. *Confusion matrix* (Annisa, 2017)

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

Adapun istilah yang digunakan pada *confusion matrix* berdasarkan Tabel 11, diantaranya.

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

Adapun formulasi penghitungan dari *confusion matrix* diantaranya adalah sensitivitas (Sn), spesifisitas (Sp), akurasi (Acc) dan *Matthew Correlation Coefficient* (Mcc).

a. Sensitivitas (Sn)

   Sn adalah persentase pasangan protein berinteraksi yang diidentifikasi dengan benar (You et al., 2015). Rumus dapat dilihat di Persamaan 15.

$$SN = \frac{TP}{TP+FN} \quad \text{.....(15)}$$

b. Spesifisitas (Sp)

   Sp adalah persentase pasangan protein non-berinteraksi yang diidentifikasi dengan benar (You et al., 2015). Rumus dapat dilihat di Persamaan 16.

$$SP = \frac{TN}{TN+FP} \quad \text{.....(16)}$$

c. Akurasi (ACC)

   ACC adalah persentase pasangan protein berinteraksi dan non-berinteraksi yang diidentifikasi dengan benar (You et al., 2015). Rumus dapat dilihat di Persamaan 17.

$$ACC = \frac{TP+TN}{TP+FP+TN+FN} \quad \text{.....(17)}$$

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 (Bekkar et al., 2013). Rumus dapat dilihat di Persamaan 18.

$$MCC = \frac{TP\ x\ TN - FP\ x\ FN}{\sqrt{(TP+FN)\ x\ (TN+FP)\ x\ (TP+FP)\ x\ (TN+FN)}} \quad \text{....(18)}$$

III. DATA DAN METODOLOGI

3.1. Tempat dan Waktu

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 dilakukan pada bulan Desember 2021 di semester tujuh ganjil hingga penyelesaian pada bulan Juni 2022. Untuk penjelasan kegiatannya ada di Tabel 12.

Tabel 12. Kegiatan dan Jadwal Penelitian

Kegiatan	2021												2022																			
	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
Pengumpulan Data																																				
Preprocessing Data																																				
Feature Extraction																																				
• AAIndex																																				
• CTD																																				
• PseAAC																																				
K-fold Cross-Validation																																				
Pemodelan *random forest*																																				
Evaluasi Kinerja																																				

Berdasarkan Tabel 12, penelitian ini dibagi menjadi 2 tahapan, antara lain.

1. Pelaksanaan Penelitian
   a. Pengambilan atau Pengumpulan Data

      Data *benchmark* telah dibuat oleh Liu, Wang dan Xiaolong pada tahun 2015 dan telah dipublikasikan. Data penelitian dapat diakses melalui link *https://static-content.springer.com/esm/art%3A10.1038%2Fsrep15479/MediaObjects/41598_2015_BFsrep15479_MOESM1_ESM.pdf*.

   b. *Preprocessing Data*

      Melakukan pemotongan data terhadap *sequences* dari dataset menjadi 50 panjang *sequences* yang digunakan agar sama dan bisa diolah.

   c. *Feature extraction*

      Dilakukan *feature extraction* menggunakan CTD, PseAAC, dan Aaindex.

   d. *k-fold Cross Validation*

      Menggunakan 10-*fold* dan 3-*fold* dengan membagi dataset menjadi 10% data *testing*, 90% data *training* dan 30% data *testing*, 70% data *training*.

   e. Pembentukan Model Data

      Penelitian ini dibuat sebuah model *random forest*.

2. Evaluasi Penelitian

   Dibagian ini, dilakukan evaluasi kinerja model menggunakan *Confusion Matrix* dengan 4 formulasi perhitungan yaitu sensitivitas, spesifisitas, akurasi dan *Matthew Correlation Coefficient.*

3.2. Data dan Alat

Berikut data dan alat yang digunakan selama penelitian.

3.2.1. Data

Tabel 13. *Dataset* yang digunakan

Dataset	PDB1075	Persentase
Positive (S^+)	525	48,837%
Negative (S^-)	550	51,163%
Total (S)	**1075**	**100%**

Berikut adalah keterangan dari Tabel 13.

S^+ : Protein pengikat DNA *positive.*

S^- : Protein pengikat DNA *negative*.

PDB : Protein Data Bank.

DNA-*binding protein sequences* didapatkan dari PDB dengan mencari kata kunci "DNA *binding protein*", "*protein*-DNA *complex*" dan kata kunci lainnya yang memiliki penjelasan yang sama. Setelah didapatkan, data harus disaring terlebih dahulu dengan 2 kriteria agar data yang didapatkan berkualitas tinggi dan tidak berlebihan. Kriteria pertama, protein dengan panjang kurang dari 50 *amino acids* dihapus dan kriteria kedua mengurangi redundansi dan bias homologi, dengan cara memotong data menggunakan PISCES yang memiliki kesamaan urutan dibawah 25%. Setelah itu didapatkan lah 525 DNA *binding protein positive*, sedangkan 550 DNA *binding protein negative* dipilih secara acak dari PDB dengan memperhatikan kriteria yang ada (Liu et al., 2015).

Dataset yang digunakan berjumlah 525 (48,837%) positif dan 550 (51,163%) negatif dengan total 1075 (100%).

Panjang dari masing-masing *sequences* dataset tersebut terdiri dari 51 sampai 1248 panjang *sequences*. Berikut bentuk dari dataset yang didapatkan dapat dilihat pada Gambar 8.

Gambar 8. Dataset *DNA-Binding protein sequences*.

Keterangan dari Gambar 8.

1AKHA dan 1UHAA : nama *sequences*.

1AKHA|1 : angka 1 setelah tanda | berarti *sequences* positif.

1UHAA|2 : angka 2 setelah tanda | berarti *sequences* negatif.

3.2.2. Alat

Berikut alat-alat yang digunakan selama penelitian.

3.2.2.1. Perangkat Keras (*Hardware)*

Perangkat keras yang digunakan untuk penelitian ini adalah sebagai berikut.

a. *Processor* : Intel(R) Core i7-4720HQ CPU @ 2.60GHz (8 CPUs), 6M Cache, 4 Cores, 8 Threads
b. *Random Access Memory* (RAM) : 8.00 GB, DDR3, 1600 MHz
c. *Storage* : SSD mSATA 256 GB. HDD 1 TB 7200rpm
d. *Network Interface* : Intel Dual Band Wireless-AC 3160
e. *Video Graphics Array* (VGA) : NVIDIA GeForce GTX 950M GDDR5 2GB

3.2.2.2. Perangkat Lunak (*Software*)

Perangkat lunak yang digunakan untuk penelitian ini adalah sebagai berikut :

a. *Operating System* : Windows 10 Home 64-Bit.
b. *Tools*
   a) R Studio 2021.09.1-372

      R Studio adalah aplikasi yang digunakan untuk Bahasa pemrograman R dan bersifat *opensource*.
   b) R Programming 3.6.1

      Bahasa pemrograman *opensource* yang digunakan untuk penelitian ini.

c) Microsoft Office Professional Plus 2016

Menggunakan Excel sebagai penelitian agar dapat mengolah data.

c. *Packages*

a) *Library* BioSeqClass 1.44.0

Mengekstrak Fitur dari Urutan dan Bangunan Biologis

Model Klasifikasi. BioSeqClass versi 1.44.0 ini dikembangkan oleh Hong.

b) *Library* randomForest 4.6-1.4

Klasifikasi dan regresi berdasarkan hutan pohon menggunakan input acak. randomForest versi 4.6-1.4 ini dikembangkan oleh Breiman and Cutler.

c) *Library* caret 6.0-86

Fungsi lain-lain untuk melatih dan merencanakan klasifikasi dan model regresi. Versi 6.0-86 ini dikembangkan oleh Kuhn.

d) *Library* protr 1.6-2

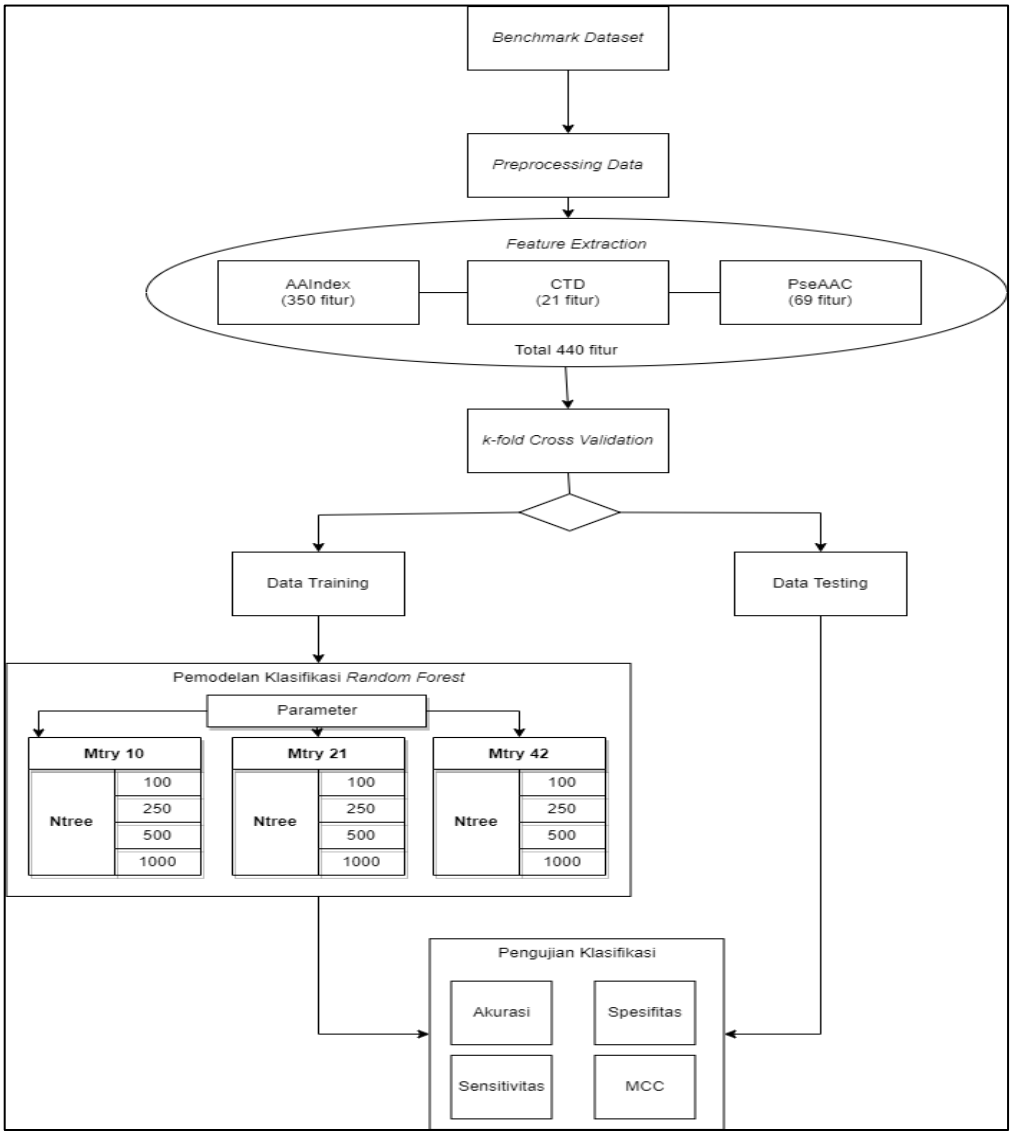
Toolkit untuk menghasilkan dan analisis berbagai numerik fitur urutan protein. protr versi 1.6.-2 ini dikembangkan oleh Xiao.

e) *Library* MCC 1.0.0

Salah satu *package* yang berfungsi untuk menghitung nilai *Matthew Corelation Coefficient* (MCC). MCC versi 1.0.0 ini dikembangkan oleh Zhou.

3.3. Metodologi

Alur kerja penelitian ini didasari oleh penelitian -penelitian sebelumnya yang membahas mengenai DNA-*binding protein*.



Gambar 9. Alur pengerjaan penelitian klasifikasi DNA-*binding protein* menggunakan algoritme *random forest*.

Berdasarkan Gambar 9, penelitian dimulai dari *benchmark dataset*/pengumpulan data dilanjutkan dengan *preprocessing data*. Selanjutnya melakukan *feature extraction* menggunakan AAIndex, CTD, dan PseAAC. Selesai dilakukannya *feature extraction*, data dibagi menjadi 10% *testing*, 90% *training* dan 30% *testing*, 70% *training* dengan menggunakan 10-*fold* dan 3-*fold cross validation*. Setelah dibagi, dilakukannya pemodelan menggunakan

data *training* dengan klasifikasi *random forest* dengan parameter *mtry* dan *ntree*. Terakhir setelah dilakukannya pemodelan, maka akan di evaluasi data *testing* dan data *training* dengan hasil berbentuk *confusion matrix*.

3.3.1. Pengumpulan Data

Pada tahap ini dilakukan dengan mengambil *benchmark dataset* dari jurnal yang digunakan, yakni 525 data positif dan 550 data negatif. Setelah itu data ini akan dilanjutkan ke tahap berikutnya yaitu pemotongan data.

3.3.2. *Preprocessing Data*

Setelah didapatkannya dataset, dilakukannya *preprocessing data* dengan melakukan pemotongan data menjadi 50 panjang *sequences*.

3.3.3. *Feature Extraction*

Tahap ini dilakukan untuk mendapatkan ciri dari suatu kelas dengan cara mengekstrak fitur utama dari masing-masing data. Proses ini memiliki tiga tahap, yaitu AAIndex, *Composition, Transition and Distribution* (CTD) dan *Pseudo Amino Acid Composition* (PseAAC) dengan total 440 fitur.

3.3.4. Pembagian Data

Di tahap ini digunakannya 10-*fold* dan 3-*fold cross validation* dengan membagi 10% *testing*, 90% *training* dan 30% *testing*, 70% *training*.

3.3.5. Klasifikasi

Pada tahap ini dilakukan pengklasifikasian data dengan algoritme *random forest* (RF). Terdapat 2 parameter yang digunakan pada klasifikasi *random forest* ini, yaitu *mtry* dan *ntree*. Nilai *mtry* yang digunakan ialah 10, 21, dan 42, sedangkan masing-masing dari *mtry* tersebut menggunakan *ntree* 100, 250, 500, dan 1000.

3.3.6. Evaluasi

Nilai pengujian atau evaluasi ini berasal dari *k-fold cross-validation* dan *random forest* yang menghasilkan nilai *confussion matrix* dengan nilai akurasi, sensitifitas, spesifisitas dan *Matthew Correlation Coefficient*.

V. PENUTUP

5.1. Simpulan

Dari penelitian dan pembahasan yang sudah dilakukan mengenai klasifikasi *dna-binding* protein menggunakan algoritme *random forest* dapat diambil beberapa kesimpulan, antara lain.

1. *Dna-binding* protein dapat di klasifikasikan dengan menggunakan *random forest* menggunakan parameter *mtry* 10, 21, dan 42 serta *ntree* 100, 250, 500, dan 1000. Berikut hasil-hasil yang didapatkan.
   A. Didapatkannya hasil tertinggi pada 3-*fold* menggunakan *mtry* 42 *ntree* 1000 dengan nilai MCC 78,02%, sensitivitas 87,19%, spesifisitas 90,75%, dan akurasi 89,02%.
   B. Hasil tertinggi pada 10-*fold* menggunakan *mtry* 42 *ntree* 1000 dengan nilai MCC 80,76%, sensitivitas 87,97%, spesifisitas 92,79%, dan akurasi 90,42%.
2. Setelah dilakukannya perbandingan dengan penelitian sebelumnya, ternyata penelitian (Rahman et al., 2018) yang menggunakan metode SVM memiliki nilai akhir yang baik. Dengan ini membuktikan bahwa menggunakan *random forest* masih memiliki hasil akhir yang kecil disaat sudah dipengaruhi oleh *feature extraction* serta parameter yang digunakan.

5.2. Saran

Saran pada penelitian ini, antara lain.

1. Penelitian ini dapat menggunakan metode klasifikasi lainnya seperti *Artificial Neural Network* dikarenakan terdapat 3 layer yang digunakan

untuk mendapatkan hasil maksimal,*Xgboost* karena dapat meningkatkan generalisasi model dan meningkatkan algoritma *ensemble*.

2. Penelitian ini dapat menggunakan *feature extraction* lain seperti *hydrophobicity*, *quasi-sequence-order* (QSO), dan lain-lain.

DAFTAR PUSTAKA

Ahmadi Azqhandi, M. H., Ghaedi, M., Yousefi, F., & Jamshidi, M. 2017. Application of random forest, radial basis function neural networks and central composite design for modeling and/or optimization of the ultrasonic assisted adsorption of brilliant green on ZnS-NP-AC. *Journal of Colloid and Interface Science*, 505, 278–292.

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

Bekkar, M., Djemaa, H. K., & Alitouche, T. A. 2013. Evaluation Measures for Models Assessment over Imbalanced Data Sets. Journal of Information Engineering and Applications.

Breiman, Leo. 2001. Random Forest. *Machine Learning*, 45, 5–32.

Breiman, L., & Cutler, A. 2003 Prediction of Protein Cellular Attributes Using Pseudo- Amino Acid Composition V4.0.

Carrasco-Castilla, J., Hernández-Álvarez, A. J., Jiménez-Martínez, C., Gutiérrez-López, G. F., & Dávila-Ortiz, G. 2012. Use of Proteomics and Peptidomics Methods in Food Bioactive Peptide Science and Engineering. *Food Engineering Reviews*, *4*(4), 224–243.

Chou, K. 2001. Prediction of Protein Cellular Attributes Using Pseudo- Amino Acid Composition. 246–255.

Chowdhury, S. Y., Shatabda, S., & Dehzangi, A. 2017. iDNAProt-ES: Identification of DNA-binding proteins using evolutionary and structural features. Scientific reports, 7(1), 1-14.

Fadilah, Laili. 2018. Klasifikasi Random Forest Pada Data Imbalanced. (Skripsi). Universitas Islam Negeri Syarif Hidayatullah, Jakarta, 51.

Govindan, G., & Nair, A. S. 2011. Composition, Transition and Distribution (CTD) - A dynamic feature for predictions based on hierarchical structure of cellular sorting. *Proceedings - 2011 Annual IEEE India Conference: Engineering Sustainable Solutions, INDICON-2011*, 1–6.

Hestie, Trevor, Robert Tibshirani, & Jerome Friedman. 2008. The Elements of Statistical Learning: Data Mining. Inference, and Prediction Second Edition. *Springer*.

Kawashima, S., Pokarowski, P., Pokarowska, M., Kolinski, A., Katayama, T., & Kanehisa, M. 2008. AAindex: Amino acid index database, progress report 2008. Nucleic Acids Research.

Khalid, S., Khalil, T., & Nasreen, S. 2014. A survey of feature selection and feature extraction techniques in machine learning. *Proceedings of 2014 Science and Information Conference, SAI 2014*, 372–378.

Kulkarni, V. Y., Sinha, P. K., & Petare, M. C. 2015. Weighted Hybrid Decision Tree Model for Random Forest Classifier. *Journal of The Institution of Engineers (India) : Series B*,97(2), 209-217

Liu, B. 2018. BioSeq-Analysis: A platform for DNA, RNA and protein sequence analysis based on machine learning approaches. *Briefings in Bioinformatics*, *20*(4), 1280–1294.

Liu, B., Wang, S., & Wang, X. 2015. DNA binding protein identification by combining pseudo amino acid composition and profile-based protein representation. *Scientific Reports*, *5*(May), 1–11.

Liu, B., Wang, S., & Wang, X. 2015. DNA binding protein identification by combining pseudo amino acid composition and profile-based protein representation. *Scientific Reports*. https://static-content.springer.com/esm/art%3A10.1038%2Fsrep15479/MediaObjects/41598_2015_BFsrep15479_MOESM1_ESM.pdf. Diakses 4 Desember 2021.

Lumbanraja, F. R., Nguyen, N. G., Phan, D., Faisal, M. R., Abipihi, B., Purnama, B., ... & Satou, K. 2018. Improved protein phosphorylation site prediction by a new combination of feature set and feature selection. Journal of Biomedical Science and Engineering, 11(6), 144-157.

Mardia, K. V. 2013. Statistical approaches to three key challenges in protein structural bioinformatics. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, *62*(3), 487–514.

Mullin, M., & Sukthankar, R. 2000. Complete Cross-Validation for Nearest Neighbor Classifiers Matthew. *ICML*, 1–8.

Murray, R. K. 2014. Biokimia Harper Edisi 27. In Igarss 2014.

Peryanto, A., Yudhana, A., & Umar, R.. 2020. Klasifikasi Citra Menggunakan Convolutional Neural Network dan K Fold Cross Validation. *Journal of Applied Informatics and Computing (JAIC)*, 4(1), 45–51.

Rahman, M. S., Shatabda, S., Saha, S., Kaykobad, M., & Rahman, M. S. (2018). DPP-PseAAC: a DNA-binding protein prediction model using Chou's general PseAAC. Journal of theoretical biology, 452, 22-34.

Refaeilzadeh, P., Tang, L., & Liu, H. 2016. Cross-Validation. *Encyclopedia of Database Systems*, 1–7.

Saidi, R., Aridhi, S., Maddouri, M., & Nguifo, E. M. 2012. Feature extraction in protein sequences classification : A new stability measure. *2012 ACM Conference on Bioinformatics, Computational Biology and Biomedicine, BCB 2012*, 683–689.

Saifullah, S., Zarlis, M., Zakaria, Z., & Sembiring, R. W. 2017. Analisa Terhadap Perbandingan Algoritma Decision Tree Dengan Algoritma Random Tree Untuk Pre-Processing Data. *J-SAKTI (Jurnal Sains Komputer Dan Informatika)*, *1*(2), 180–185.

Salman, Afan Galih. 2020. Konsep Decision Tree & Random Forest. https://socs.binus.ac.id/2020/05/26/konsep-decision-tree-random-forest/ Diakses pada 31 Januari 2022

Shen, H. B., & Chou, K. C. 2008. PseAAC: a flexible web server for generating various kinds of protein pseudo amino acid composition. *Analytical biochemistry, 373(2), 386-388.*

Wei, L., Tang, J., & Zou, Q. 2017. Local-DPP: An improved DNA-binding protein prediction method by exploring local evolutionary information. Information Sciences, 384, 135-144.

Xu, Y., Verma, D., Sheridan, R. P., Liaw, A., Ma, J., Marshall, N. M., ... & Johnston, J. M. 2020. Deep dive into machine learning models for protein engineering. *Journal of chemical information and modeling, 60(6), 2773-2790.*

You, Z. H., Chan, K. C. C., & Hu, P. 2015. 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), 1–19.