TESIS – ES235401

Prediksi Interaksi Obat-Target Menggunakan Pendekatan *Stacking Ensemble Learning*

VIKO PRADANA PRASETYO

6026231040

DOSEN PEMBIMBING

Dr. Wiwik Anggraeni, S.Si., M.Kom.

NIP 197601232001122002

DEPARTEMEN SISTEM INFORMASI

FAKULTAS TEKNOLOGI ELEKTRO DAN INFORMATIKA CERDAS

INSTITUT TEKNOLOGI SEPULUH NOPEMBER

SURABAYA

2024

(*Halaman ini sengaja dikosongkan*)

THESIS – ES235401

Drug-Target Interactions Prediction Using Stacking Ensemble Learning Approach

VIKO PRADANA PRASETYO
6026231040

ADVISOR
Dr. Wiwik Anggraeni, S.Si., M.Kom.
NIP 197601232001122002

DEPARTEMENT OF INFORMATION SYSTEM
FACULTY OF INTELLIGENT ELECTRICAL AND INOFRMATICS TECHNOLOGY
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2024

(*Halaman ini sengaja dikosongkan*)

LEMBAR PENGESAHAN TESIS

Tesis disusun untuk memenuhi salah satu syarat memperoleh gelar
Magister Sistem Informasi (M.Kom.)
di
Institut Teknologi Sepuluh Nopember

Oleh:
Viko Pradana Prasetyo
NRP: 6026231040

Tanggal Ujian: 26 Juli 2024
Periode Wisuda ITS: 130

Disetujui oleh:
Pembimbing:

Dr. Wiwik Anggraeni, S.Si, M.Kom
NIP: 197601232001122002

Penguji:

Retno Aulia Vinarti, S.Kom., M.Kom., Ph.D
NIP: 1988201812010

Ahmad Mukhlason, S.Kom, M.Sc, Ph.D
NIP: 198203022009121009

Surabaya, 05 Agustus 2024
Kepala Departemen Sistem Informasi
Fakultas Teknologi Elektro dan Informatika Cerdas

LP/P/24/166
DSI-130-F-23-267/9

Dr. Mudjahidin, ST, MT
NIP: 197010102003121001

(*Halaman ini sengaja dikosongkan*)

PERNYATAAN ORISINALITAS

Yang bertanda tangan di bawah ini:

Nama mahasiswa / NRP	:	Viko Pradana Prasetyo / 6026231040
Program studi	:	S2 Sistem Informasi
Dosen Pembimbing / NIP	:	Dr. Wiwik Anggraeni, S.Si, M.Kom / 197601232001122002

dengan ini menyatakan bahwa Tesis dengan judul "Prediksi Interaksi Obat-Target Menggunakan Pendekatan Stacking Ensemble Learning" adalah hasil karya sendiri, bersifat orisinal, dan ditulis dengan mengikuti kaidah penulisan ilmiah.

Bilamana di kemudian hari ditemukan ketidaksesuaian dengan pernyataan ini, maka saya
bersedia menerima sanksi sesuai dengan ketentuan yang berlaku di Institut Teknologi Sepuluh Nopember.

Surabaya, 02 Agustus 2024

Mengetahui

Dosen Pembimbing

Dr. Wiwik Anggraeni, S.Si, M.Kom
NIP. 197601232001122002

Mahasiswa

Viko Pradana Prasetyo
NRP. 6026231040

(*Halaman ini sengaja dikosongkan*)

Prediksi Interaksi Obat-Target Menggunakan Pendekatan *Stacking Ensemble Learning*

Oleh	: Viko Pradana Prasetyo
NRP	: 6026231040
Dosen Pembimbing	: Dr. Wiwik Anggraeni, S.Si., M.Kom.

ABSTRAK

Proses penemuan obat, terutama interaksi obat-target (DTI), memerlukan waktu dan biaya yang besar untuk mendapatkan hasil eksperimen dan persetujuan global. Untuk menghemat waktu dan biaya, digunakan berbagai metode komputasi seperti machine learning dan deep learning. Namun, kedua metode ini memiliki kekurangan seperti kebutuhan data besar dan biaya komputasi tinggi serta rentan terhadap *overfitting*. Metode yang dapat mengatasi masalah ini adalah *Stacking Ensemble Learning* (SEL). Penelitian ini bertujuan untuk menerapkan pendekatan SEL untuk memprediksi DTI.

Studi ini menggunakan SEL dengan algoritme seperti AdaBoost, Gradient Boosting, dan Random Forest. Hasilnya menunjukkan peningkatan signifikan dengan menggunakan teknik *oversampling* SMOTE, seperti peningkatan skor akurasi CV rata-rata dari 0,881 menjadi 0,921 untuk AdaBoost, dari 0,862 menjadi 0,929 untuk Gradient Boosting, dan dari 0,861 menjadi 0,929 untuk Random Forest. Model meta learner yang diuji menunjukkan variasi performa: model 1 memiliki akurasi tertinggi 0,929 dan *precision* 0,960, model 2 memiliki *recall* tertinggi yaitu 0,976, sedangkan model 3 berfokus pada meminimalkan *false positive* dengan *precision tertinggi*, yaitu 0,991. Hasil rekomendasi juga telah divalidasi oleh pakar, yang membuktikan efektivitasnya. Penelitian ini menunjukkan bahwa SEL dapat meningkatkan prediksi DTI dan mempercepat proses penemuan obat.

Kata kunci: akurasi, interaksi obat-target, prediksi, SMOTE, *stacking ensemble learning*

Drug-Target Interactions Prediction Using Stacking Ensemble Learning Approach

By : Viko Pradana Prasetyo
Student ID : 6026231040
Advisor : Dr. Wiwik Anggraeni, S.Si., M.Kom.

ABSTRACT

The process of drug discovery, especially drug-target interactions (DTI), requires significant time and cost to obtain experimental results and global approval. To save time and costs, various computational methods are used, such as machine learning and deep learning. However, these methods have drawbacks, including the need for large data sets, high computational costs, and susceptibility to overfitting. A method that can address these issues is Stacking Ensemble Learning (SEL). This research aims to apply the SEL approach to predict DTI.

The study implemented SEL using base learners including AdaBoost, Gradient Boosting, and Random Forest, achieving significant performance improvements with SMOTE oversampling technique, where AdaBoost's mean CV score rose from 0.881 to 0.921, Gradient Boosting's from 0.862 to 0.929, and Random Forest's from 0.861 to 0.929. The meta learner models demonstrated varying result. Model 1 achieving the highest accuracy of 0.929 and a precision of 0.960, while model 2 improved recall to 0.976, and model 3 focused on minimizing false positives with a precision of 0.991. Additionally, the recommender system's results were validated by expert, confirming its effectiveness. This research highlights the potential of SEL in enhancing DTI prediction, offering a more efficient approach to drug discovery.

Keywords: accuracy, drug-target interaction, prediction, SMOTE, stacking ensemble learning

KATA PENGANTAR

Puji syukur atas kehadiran Allah SWT yang telah melimpahkan rahmat, nikmat, dan berkah sehingga penulis dapat menyelesaikan laporan tugas akhir berjudul "Prediksi Interaksi Obat-Target Menggunakan Pendekatan *Stacking Ensemble Learning*" sebagai syarat untuk menyelesaikan pendidikan di tingkat magister Departemen Sistem Informasi, Fakultas Teknologi Elektro dan Informatika Cerdas, Institut Teknologi Nopember Surabaya. Penulis mengucapkan terima kasih kepada:

1. Bapak Tri Eko Prasetyo, Ibu Elisa Novida Andriani dan Fitri Apriliana yang telah memberikan doa dan dukungan selama penulis menempuh studi magister di Institut Teknologi Sepuluh Nopember Surabaya.
2. Ibu Dr. Wiwik Anggraeni, S.Si., M.Kom. selaku dosen pembimbing yang dengan sabar mengarahkan dan memberi dukungan untuk menyelesaikan tesis ini.
3. Mbak Dr. apt. Wenny Putri Nilamsari, S.Farm.,Sp.FRS. selaku dosen fakultas farmasi Universitas Airlangga yang telah bersedia untuk membantu validasi sistem *recommender* tesis ini.
4. Firsty Putri Novita Sari, seseorang dengan empati dan semangat tinggi telah memberikan penulis kekuatan dalam menyelesaikan tesis ini.
5. Teman-teman Navisatya Fast Track yang telah memberikan pengalaman dan momen berharga selama penulis menempuh studi magister ini.
6. Semua pihak yang tidak bisa disebutkan satu per satu, yang telah memberikan bantuan dan dukungan kepada penulis selama menjadi mahasiswa magister Sistem Informasi, Institut Teknologi Sepuluh Nopember Surabaya.

Surabaya, Agustus 2024

Penulis

(*Halaman ini sengaja dikosongkan*)

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

DAFTAR KODE PROGRAM

BAB 1
PENDAHULUAN

1.1 Latar Belakang

Penemuan obat adalah proses penemuan kandidat obat baru, baik secara kimiawi maupun biologis. Salah satu ranah penemuan obat adalah interaksi obat-protein (*drug-target interaction* atau DTI). Dengan kemajuan teknologi medis, DTI dapat diteliti lebih jauh dan mendalam (Csermely et al., 2013). Karena banyaknya senyawa obat dan protein target, banyak interaksi antara entitas ini yang belum diketahui. Pengetahuan yang lebih lengkap tentang DTI tidak hanya berkontribusi pada pemahaman yang lebih baik mengenai farmakologi obat, tetapi juga relevan untuk memprediksi efek samping dan reposisi obat, yaitu penggunaan obat yang sudah ada untuk mengobati penyakit yang belum pernah disembuhkan sebelumnya.

Faktanya, hanya beberapa lusin obat baru yang disetujui oleh FDA (Departemen Kesehatan Amerika Serikat) setiap tahunnya. Terlebih lagi, biaya rata-rata yang terkait dengan penemuan obat baru adalah sekitar $1,8 miliar, dan prosesnya memakan waktu lebih dari 10 tahun (Buza & Peška, 2017). Prediksi DTI merupakan bagian penting dari reposisi obat yang dapat mengurangi biaya dan waktu pengembangan obat. Oleh karena itu, metode komputasi diperlukan untuk memprediksi DTI untuk mengecilkan *scope*, mengurangi biaya, dan menghemat waktu percobaan (Shang et al., 2021). Berbagai metode komputasi digunakan untuk melakukan prediksi DTI, mulai dari *machine learning* hingga *ensemble learning*.

Machine learning (ML) adalah cabang dari *Artificial Intelligence* (AI) yang bertujuan untuk mengembangkan dan menerapkan algoritma komputer yang belajar dari data mentah yang belum diproses, untuk kemudian melakukan tugas tertentu. Tugas utama yang dilakukan oleh algoritma AI adalah klasifikasi, regresi, pengelompokan, atau pengenalan pola dalam kumpulan data yang besar (Carracedo-Reboredo et al., 2021). Berbagai metode *machine learning* (ML) telah digunakan dalam pemanfaatan sejumlah besar data kompleks berdimensi tinggi untuk memprediksi interaksi, yang dapat memberikan daftar kandidat obat baru sebagai validasi secara eksperimental sehingga mengurangi waktu dan biaya.

Strategi ML di mana hanya sebagian senyawa terfokus yang diuji dan hasil pembacaan eksperimental digunakan untuk menyempurnakan pemilihan molekul untuk *screening* berikutnya, dapat membantu dalam identifikasi target protein. (Reker & Schneider, 2015). ML memiliki kelemahan, seperti tidak bisa menangani *big data* sehingga membutuhkan waktu yang lama. (Elbadawi et al., 2021).

Metode *deep learning* (DL) adalah metode berbasis *artificial neural network* (ANN) dengan banyak *hidden layer* dan prosedur pelatihan parameter yang lebih canggih. DL mulai banyak digunakan karena performa yang relatif lebih baik dan kemampuannya mempelajari representasi data dengan berbagai tingkat abstraksi dibanding ML. Kelebihan DL dibandingkan dengan ML antara lain dapat mempelajari fitur hirarki yang kompleks, dapat menangani data yang tidak terstruktur, dan dapat menyesuaikan model dengan dataset yang sangat besar (Sarker, 2021; Shiri et al., 2023). Namun, ada beberapa kelemahan dari DL, yaitu rawan terjadi *overfitting* dan membutuhkan data dengan kualitas tinggi (Ganaie et al., 2022).

Ensemble learning (EL) atau pembelajaran ansambel adalah sebuah metode menggabungkan berbagai model pembelajaran yang berbeda untuk meningkatkan hasil yang diperoleh dari masing-masing model individu. Ada beberapa tipe EL, salah satunya adalah *stacking ensemble learning* (SEL). SEL dapat mengatasi kekurangan dari DL, seperti performa generalisasi yang lebih baik sehingga mengurangi *overfitting* dan bias serta lebih mudah diinterpretasi. Terdapat beberapa penelitian yang menerapkan SEL, Contohnya adalah penelitian oleh (Muslim et al., 2023) yang diterapkan pada prediksi resiko peminjaman pada perusahaan *peer-to-peer* (P2P) *lending*. Hasilnya menunjukkan akurasi yang tinggi, yaitu lebih dari 90% untuk dua dataset yang berbeda. Contoh lainnya adalah penelitian oleh (Cui et al., 2021) yang diterapkan pada prediksi korban jiwa akibat gempa bumi. Hasilnya menunjukkan bahwa SEL dapat meningkatkan performa model.

Sampai saat ini penerapan SEL dalam rangka penemuan obat berada di ranah interaksi protein-protein (*Protein-Protein Interaction* atau PPI), belum ada penelitian yang menerapkannya pada prediksi DTI. Untuk itu, peneliti mencoba menggunakannya pada penelitian ini untuk menghasilkan prediksi DTI dengan

akurasi tinggi. Diharapkan dengan menggunakan SEL, obat yang aman dikonsumsi oleh manusia dapat diprediksi.

1.2 Rumusan Masalah

Protein yang dapat menghambat atau mendorong timbulnya dan berkembangnya suatu penyakit dapat dianggap sebagai kandidat protein untuk pencegahan dan pengobatan penyakit. Berdasarkan hal tersebut, obat yang relevan dapat difilter atau dikembangkan untuk pencegahan dan pengobatan penyakit. Terapi obat dicapai ketika molekul obat berikatan dengan target dan mengatur aktivitas biologisnya, dan identifikasi DTI bermanfaat untuk pengobatan penyakit selanjutnya. Oleh karena itu, mengidentifikasi DTI penting untuk pengembangan obat. Tujuan memprediksi DTI adalah untuk mengidentifikasi target dan obat baru (Ru et al., 2021).

Identifikasi DTI secara eksperimental membutuhkan waktu yang lama dan biaya yang besar. Untuk itu, metode komputasi digunakan untuk memprediksi DTI. Model ML dan DL dikembangkan untuk menangani permasalahan tersebut. Namun, terdapat beberapa kekurangan dari masing-masing model. Untuk itu, diperlukan sebuah model yang dapat mengatasi kekurangan tersebut. Penelitian ini menggunakan SEL sebagai metode yang digunakan untuk memprediksi DTI. Model SEL merupakan kombinasi dari beberapa model sehingga performa model menjadi lebih tinggi serta hasil yang diharapkan dapat lebih baik dari sebelumnya.

1.3 Tujuan Penelitian

Berdasarkan rumusan masalah yang telah dijelaskan sebelumnya, tujuan dari penelitian ini adalah menerapkan *Stacking Ensemble Learning* (SEL) untuk dapat memprediksi obat yang berinteraksi positif dengan protein tertentu, baik manusia maupun mikroorganisme seperti bakteri dan virus dengan membuat *recommender*.

1.4 Manfaat Penelitian

Manfaat dari penelitian yang dilakukan adalah menghasilkan model yang mampu mengidentifikasi interaksi antara senyawa obat dan protein manusia sehingga dapat memberikan daftar obat yang aman untuk dikonsumsi oleh manusia.

1.5 Kontribusi Penelitian

Secara teoritis, kontribusi dari penelitian ini adalah melakukan prediksi DTI pada dataset daftar obat dan protein dengan menggunakan *Stacking Ensemble Learning* (SEL). Sedangkan secara praktis, kontribusi dari penelitian ini adalah mempersingkat waktu dan mengurangi biaya dalam memprediksi DTI.

1.6 Batasan Penelitan

Pemberian batasan pada penelitian ini ditujukan untuk memberikan fokus dan arah yang jelas terhadap hasil penelitian ini. Batasan pada penelitian ini di antaranya:

1. Data yang digunakan pada penelitian ini adalah data publik yang terdiri dari daftar obat dan protein, struktur kimia obat dan rantai protein dari *database* DrugBank pada tautan *go.drugbank.com*.
2. Algoritma yang digunakan untuk membangun *base learner* SEL adalah Adaptive Boosting, Gradient Boosting, dan Random Forest dengan *meta-learner* yang digunakan adalah *base learner* dengan nilai *cross validation* tertinggi.

BAB 2
KAJIAN PUSTAKA

Pada bab ini dipaparkan kajian penelitian terdahulu dan kajian teori yang menjadi tinjauan pustaka pada penulisan penelitian ini. Pada bab ini dijelaskan tentang penelitian yang pernah dilakukan dalam lingkup penemuan obat..

2.1 Kajian Penelitian Terdahulu

Bagian ini menjelaskan penelitian-penelitian yang pernah dilakukan untuk mencari *gap* atau celah yang dapat diteliti.

2.1.1 Penelitian Sebelumnya

Penelitian pertama yang digunakan sebagai rujukan penelitian ini berjudul "*Prediction of drug-target interactions based on multi-layer network representation learning*". Penelitian oleh (Shang et al., 2021) mengusulkan pendekatan *similarity network* untuk memprediksi DTI. Dataset yang digunakan pada penelitian ini antara lain struktur kimia obat dan interaksinya, relasi obat-penyakit, efek samping obat, serta rantai protein dan interaksinya. Penelitian ini menggunakan metode berbasis *Deep Neural Network* (DNN). *Multilayer network* dibangun berdasarkan similaritasnya, kemudian diubah menjadi vektor untuk dihitung dan dibuat prediksi DTI. Metode yang diusulkan dibandingkan dengan empat metode lain, menunjukkan hasil terbaik dengan nilai lebih dari 90% untuk skor AUPR dan AUROC.

Penelitian berjudul "*Semi-supervised heterogeneous graph contrastive learning for drug–target interaction prediction*" oleh (Yao et al., 2023) mengusulkan metode *Heterogenous Graph Convolutional Network* (HGCN). Serupa dengan penelitian pertama, dataset yang diperoleh dibangun *network* berdasarkan similaritasnya menjadi *node-node*. Representasi *node* dipelajari dengan *contrastive learning* untuk memaksimalkan similaritas di antara pasangan obat-target positif dan meminimalkan similaritas di antara pasangan negatif. Metode yang diusulkan kemudian dibandingkan dengan empat metode lain, menujukkan hasil terbaik untuk skor AUPR dan AUROC.

Penelitian berjudul "*BE-DTI': Ensemble framework for drug target interaction prediction using dimensionality reduction and active learning*" oleh (Sharma & Rani, 2018) mengusulkan *framework ensemble* yang terdiri dari *active learning* (AL) dan reduksi dimensi. AL digunakan untuk meningkatkan performa *ensemble* berbaris *bootstrap aggregation* (*bagging*) dalam menangani *under-sampling*, sedangkan reduksi dimensi digunakan untuk menangani data dengan dimensi tinggi. Dataset yang digunakan adalah interaksi obat-target yang *raw* serta fitur untuk masing-masing obat dan protein. Peneliti menggunakan *Decision Trees* (DT) sebagai *base learner*. Metode yang diusulkan dibandingkan dengan tiga metode lain dan memiliki hasil terbaik dengan masing-masing nilai AUC sebesar 0.927, *sensitivity* sebesar 0.886, *specificity* sebesar 0.864, dan *G-mean* sebesar 0.874.

Penelitian berjudul "*DTi2Vec: Drug–target interaction prediction using network embedding and ensemble learning*" oleh (Thafar et al., 2021) mengusulkan metode *network embedding* dan *ensemble* berbasis *boosting*. Dataset terdiri dari rantai protein dan struktur kima obat. Dari dataset tersebut diterapkan model DTi2Vec, mulai dari membangun *heterogenous network* berdasarkan similaritasnya, kemudian *generate* fitur dari tiap *node* obat dan protein menggunakan teknik *node embedding*. Performa model dievaluasi nilai AUPR-nya menggunakan dua *ensemble learning classifier*, yaitu *AdaBoost* dan *XGBoost*. Terdapat dataset yang memiliki nilai AUPR tinggi dengan *tuning AdaBoost*, ada juga dataset yang nilai AUPR-nya tinggi dengan *tuning XGBoost*. Metode yang diusulkan kemudian dibandingkan dengan lima metode lain, dan DTi2Vec memiliki hasil terbaik untuk tiap dataset yang diuji.

Penelitian berjudul "*EA-based hyperparameter optimization of hybrid deep learning models for effective drug-target interactions prediction*" oleh (Mahdaddi et al., 2021) mengusulkan metode CNN-AbiLSTM (CNN – *Attention-based Bidirectional* LSTM) untuk memprediksi DTI berdasarkan afinitas ikatannya menggunakan kombinasi CNN dengan *attention-based* biLSTM. CNN-AbiLSTM disempurnakan dengan algoritma *Differential Evolution* (DE) yang merupakan salah satu tipe dari *Evolutionary Algorithm* (EA) untuk menemukan konfigurasi *hyperparameter* yang optimal bagi model yang diusulkan. Dataset yang digunakan

adalah struktur kimia obat dan rantai protein dari Davis dan Kiba dataset. Hasilnya dibandingkan dengan enam metode lain, menunjukkan hasil terbaik berdasarkan nilai MSE dan AUPR.

Penelitian berjudul "*PreDTIs: prediction of drug–target interactions based on multiple feature information using gradient boosting framework with data balancing and feature selection techniques*" oleh (Mahmud et al., 2021) mengusulkan metode PreDTI untuk melakukan prediksi DTI. Dataset yang digunakan adalah struktur kimia obat dan rantai protein. Berbagai metode ektraksi fitur digunakan, kemudian dilakukan *data balancing* menggunakan algoritma FastUS (*Fast Under Sampling*) dan seleksi fitur menggunakan algortima MoIFS (*Modified Incremental Feature Selection*). Model *classifier* yang digunakan adalah LightGBM (*Light Gradient Boostin Machine*). Metode yang diusulkan dibandingkan dengan masing-masing tiga metode lain untuk seleksi fitur dan *classifier*, menunjukkan hasil terbaik dari segi akurasi.

Penelitian berjudul "*Using a stacked ensemble learning framework to predict modulators of protein–protein interactions*" oleh (Gao et al., 2023) mengusulkan metode SELPPI (*Stacked Ensemble Learning for Protein-Protein Interaction*) untuk memprediksi modulator baru yang menargetkan interaksi protein-protein (*Protein-Protein Interaction* atau PPI). Dataset yang digunakan adalah pdCSM (*Protein Data Bank-Consensus Scoring Measures*) yang terdiri dari 4965 modulator PPI yang menargetkan 51 PPI yang berbeda. Enam algoritma digunakan sebagai *basic learner*. Enam algoritma tersebut kemudian digunakan secara bergantian sebagai *meta learner*. SEL digunakan untuk memaksimalkan penggunaan berbagai macam model ML, terutama penggunaan fitur beserta kombinasinya. Hasilnya dibagi menajdi klasifikasi dan regresi, menunjukkan hasil yang terbaik jika dibanding dengan enam model dasar.

Penelitian berjudul "*Computational prediction and interpretation of druggable proteins using a stacked ensemble learning framework*" oleh (Charoenkwan et al., 2022) mengusulkan metode SPIDER (*Stacked PredIctor of DruggablE pRoteins*) untuk meningkatkan akurasi prediksi protein yang dapat diobati. Dataset terdiri dari daftar protein *druggable* dan *non-druggable*. Dataset dilatih dengan sepuluh *feature encoding* dan enam algoritma. SEL digunakan agar

data *test* dapat digunakan dan model dapat diinterpretasikan. Hasil menunjukkan bahwa SPIDER memiliki hasil terbaik jika dibandingkan dengan model dasar dalam hal *cross-validation* dan *independent test*.

2.1.2 Analisis *Gap* Penelitian

Penelitian tentang DTI masih terus dikaji dan diteliti dengan berbagai pendekatan dan metode. Tabel di bawah menunjukkan rangkuman analisis *gap* penelitian sebelumnya dari sisi dataset dan metode yang digunakan. Untuk kolom metode, DL adalah *deep learning*, EL adalah *ensemble learning* dan SEL adalah *stacking ensemble learning*.

Tabel 2.1 Analisis *Gap* Penelitian tentang DTI

Penelitian Terdahulu	Metode	Dataset			
		Daftar Obat	Daftar Protein	Struktur Kimia Obat	Rantai Protein
(Shang et al., 2021)	DL	✓	✓	✓	✓
(Yao et al., 2023)	DL	✓	✓	✓	✓
(Sharma & Rani, 2018)	EL			✓	✓
(Thafar et al., 2021)	EL			✓	✓
(Mahdaddi et al., 2021)	DL			✓	✓
(Mahmud et al., 2021)	EL			✓	✓
(Gao et al., 2023)	SEL				✓
(Charoenkwan et al., 2022)	SEL				✓
Penelitian ini	SEL	✓	✓	✓	✓

Penelitian sebelumnya menggunakan metode DL dan EL untuk memprediksi DTI, sedangkan SEL digunakan untuk memprediksi interaksi protein-protein (*Protein-Protein Interaction* atau PPI). Penelitian ini mencoba menggunakan metode SEL untuk memprediksi DTI yang diterapkan pada

kombinasi empat jenis dataset, yaitu daftar obat, daftar protein, struktur kimia obat dan rantai protein.

2.2 Kajian Teori

Bagian ini menjelaskan teori-teori yang digunakan oleh penelitian terdahulu untuk kemudian menjadi tolak ukur dan landasan penelitian ini.

2.2.1 *Drug-Target Interaction*

DTI mengacu pada pengikatan obat ke lokasi target yang mengakibatkan perubahan perilaku atau fungsinya. Obat pada dasarnya mengacu pada senyawa kimia apa pun yang menyebabkan perubahan fisiologis pada tubuh manusia ketika dikonsumsi, disuntikkan atau diserap. Target, juga dikenal sebagai target biologis, adalah bagian mana pun dari organisme hidup yang diikat oleh obat untuk menghasilkan perubahan fisiologis. Target adalah entitas seperti protein atau asam nukleat yang diarahkan untuk perubahan apa pun. Target biologis yang paling umum adalah reseptor nukleus, saluran ion, reseptor G-protein berpasangan, dan enzim. DTI dapat terjadi melalui dua cara. Cara pertama, yang dikenal sebagai inhibitor kompetitif, menempel pada sisi aktif target untuk menghambat reaksi. Cara kedua, yang disebut inhibitor alosterik, berikatan dengan situs alosterik target. Ini mengubah bentuk dan struktur target sehingga substrat tidak dapat mengenalinya. Dengan demikian, reaksi-reaksi tersebut tidak terjadi. Pemblokiran reaksi target dapat memperbaiki ketidakseimbangan metabolisme atau membunuh patogen untuk menyembuhkan penyakit (Sachdev & Gupta, 2019).

2.2.2 K-Mers

K-mers adalah *substring* dari nukleotida dengan panjang k yang terdapat dalam suatu *sequence*. K-mers biasanya digunakan untuk DNA, tetapi konsep ini juga dapat diterapkan pada *sequence* RNA atau protein. Setiap *sequence* genomik dapat didekomposisi menjadi sejumlah k-mers berurutan, dan jumlah ini akan tergantung pada panjang *sequence* (L) dan panjang k-mers (k). Jumlah k-mers dalam suatu *sequence* dengan panjang L adalah sama dengan L - k + 1. Ini adalah prinsip umum yang dapat diterapkan pada setiap *sequence*, terlepas dari panjang atau komposisi *sequence* tersebut (Jenike et al., 2024).

2.2.3 *Jaccard Similarity*

Koefisien *Jaccard similarity*, kadang-kadang disebut sebagai indeks *Jaccard*, pertama kali dirumuskan oleh Paul Jaccard pada tahun 1901 sebagai cara untuk menggambarkan hubungan *co-occurrence* secara umum. Dalam bentuk umumnya, koefisien ini mengukur similaritas antara dua himpunan A dan B, dan dapat dihitung sebagai berikut, di mana |A| menunjukkan kardinalitas himpunan A:

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|} \tag{2.1}$$

Dengan demikian, koefisien *Jaccard similarity* adalah ukuran dari irisan dua himpunan dibagi oleh ukuran gabungan keduanya. Nilai maksimal koefisien adalah satu, menandakan hubungan similaritas yang sempurna di mana gabungan dan irisan memiliki ukuran yang sama. Nilai minimalnya adalah nol, yang terjadi ketika ukuran irisan adalah nol, menunjukkan hubungan similaritas yang sangat negatif di mana kedua himpunan saling eksklusif (Koeneman & Cavanaugh, 2022).

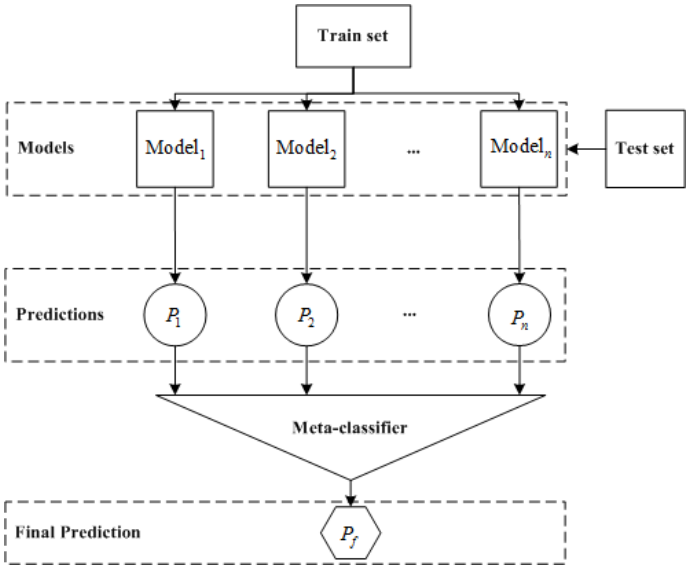
2.2.4 *Stacking Ensemble Learning*

Ensemble learning (EL) atau pembelajaran ansambel adalah sebuah metode menggabungkan berbagai model pembelajaran dalam konteks *machine learning* (ML) yang berbeda untuk meningkatkan hasil yang diperoleh dari masing-masing model individu. EL pertama kali dikembangkan pada tahun 1990 oleh Hansen dan Salamon pada penelitian yang berjudul "Neural Network Ensembles", yang mana terbukti dari beberapa algoritma pembelajaran yang lemah dikonversi menjadi sebuah algoritma pembelajaran yang kuat. EL dibagi menjadi dua tahapan. Pada tahapan pertama, satu set *base learner* didapatkan dari *training* data. Kemudian, pada tahapan kedua, *learner* yang didapat digabungkan untuk mendapatkan satu model yang terpadu. Dengan demikian, *multiple forecast* berdasarkan *base learner* yang berbeda-beda dikombinasikan menjadi satu model yang lebih unggul dari model individu (Divina et al., 2018).

Berdasarkan metode ansambelnya, EL dibagi menjadi tiga, yaitu EL berbasis *bagging*, *boosting*, dan *retraining*. Pada EL berbasis *bagging*, semua model dasar dilatih secara bersamaan. *Output* dari *bagging* didasarkan pada hasil

voting. Untuk EL berbasis *boosting*, model-model yang digunakan bekerja secara berurutan. Setiap model yang dilatih akan diberi bobot untuk menyempurnakan pelatihan model selanjutnya sehingga *output* akhirnya adalah hasil rata-rata dari bobot-bobot yang diperoleh (*weighted mean*), bukan hasil *voting*. Untuk EL berbasis *retraining*, *output* dari model primer dijadikan sebagai *input* untuk pelatihan model sekunder. EL ini disebut *stacked generalization* atau algoritma *stacking ensemble* (SE) (Li et al., 2021).

Stacked generalization dapat mengurangi bias dan *error* dalam generalisasai ketika dibandingkan dengan penggunaan algoritma tunggal. Untuk mencapainya, *stacking* memungkinkan peleburan dari algoritma yang berbeda dan heterogen dengan parameter tertentu. Berbeda dengan kedua metode ansambel lainnya, SE menggunakan *meta-learner* untuk mengagregasi prediksi dari *layer* terakhir dan mendapat performa terbaik (Chatzimparmpas et al., 2021).



Gambar 2.1 Arsitektur *Stacking Ensemble Learning*

2.2.5 Klasifikasi

Klasifikasi didefinisikan sebagai upaya untuk mengkategorikan sebuah data menjadi kelas-kelas yang berbeda (Naser & Alavi, 2023). Klasifikasi bertujuan untuk memprediksi kelas tujuan dengan presisi tertinggi. Algoritma klasifikasi mencari hubungan antara atribut *input* dan *output* untuk membangun model yang mana merupakan proses *training* (Charbuty & Abdulazeez, 2021). Klasifikasi

menyimpulkan beberapa fungsi pemetaan dari kumpulan data *training* dan memprediksi label kelas dengan bantuan fungsi pemetaan untuk entri data baru. Atribut atau fitur adalah parameter yang ditemukan dalam kumpulan masalah tertentu yang cukup membantu membangun model prediksi yang akurat. Dalam klasifikasi, suatu sampel bahkan dapat dipetakan ke lebih dari satu label. Contohnya adalah artikel berita dapat diberi label sebagai artikel olahraga, artikel tentang beberapa pemain, dan artikel tentang tempat tertentu pada saat yang bersamaan (Sen et al., 2020).

2.2.6 *Evaluation Metrics*

Evaluation metrics atau metriks evaluasi adalah pengukuran kuantitatif yang digunakan untuk mengukur performa dan efektivitas dari sebuah model statistik atau ML. Metriks evaluasi memberi gambaran seberapa bagus performa model dan membantu dalam membandingkan berbagai model atau algoritma. Untuk klasifikasi, performa diukur dengan *confusion matrix*. Matriks ini berisi statistik mengenai klasifikasi prediksi dan aktual agar dapat memahami pengukuran akurasi dari pengklasifikasi. Kolom matriks menandakan kejadian yang diprediksi, sedangkan baris matriks menandakan kejadian yang sebenarnya (Naser & Alavi, 2023).

Pengukuran yang terdapat pada *confusion matrix* antara lain *accuracy*, *precision*, *recall*, dan *F1-score*. *Precision* adalah ukuran dari nilai positif sebenarnya (*true positive*) terhadap jumlah total kejadian positif baik yang sebenarnya maupun hasil prediksi (*true positive* + *false positive*). *Recall* adalah ukuran dari jumlah prediksi kelas positif yang sebenarnya (*true positive*) terhadap semua kejadian sebenarnya (*true positive* + *false negative*). *F1-score* adalah ukuran yang menyeimbangkan *precision* dan *recall*. *Accuracy* adalah ukuran dari perbandingan semua hasil sebenarnya terhadap semua hasil baik prediksi maupun sebenarnya (Sambasivam & Opiyo, 2021). Berikut adalah rumus persamaan dari pengukuran dalam *confusion matrix*:

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive} \tag{2.2}$$

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative} \tag{2.3}$$

$$F1\ Score = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{2.4}$$

$$Accuracy = \frac{True\ Positive + True\ Negative}{True\ Positive + False\ Positive + True\ Negative + False\ Negative} \tag{2.5}$$

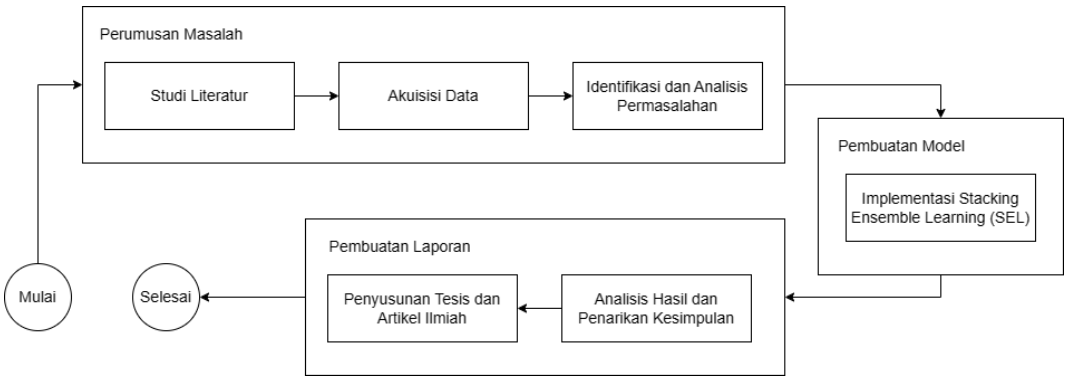
(*Halaman ini sengaja dikosongkan*)

BAB 3
METODOLOGI PENELITIAN

Pada bab ini berisi mengenai penjelasan alur metodologi penelitian dan pengerjaan tesis. Dengan adanya metodologi ini, diharapkan penelitian tesis dapat berjalan dengan tepat sasaran, terarah, dan sistematis. Proses pengerjaan dibagi menjadi tiga tahap besar, yaitu perumusan masalah, pembuatan model, serta pembuatan laporan.

3.1 Diagram Metodologi Penelitian

Tahapan pengerjaan penelitian dimulai dari studi literatur hingga penyusunan tesis dan artikel ilmiah dapat dilihat pada diagram alir seperti pada gambar 3.1.



Gambar 3.1 Diagram Alir Metodologi Penelitian

3.2 Uraian Metodologi Penelitian

Berikut adalah uraian masing-masing tahapan penelitian yang akan dilakukan.

3.2.1 Studi Literatur

Mengumpulkan informasi terkait penelitian untuk membandingkan beberapa metode dan menemukan *research gap* sebagai peluang untuk menyelesaikan permasalahan terkait topik tesis.

3.2.2 Akuisisi Data

Akuisisi data dilakukan dengan mengakses situs basis data DrugBank dengan tautan *go.drugbank.com* dan mengunduh dataset yang diperlukan. DrugBank merupakan basis data obat dan protein *online*, pertama kali dikembangkan pada 2006 oleh Dr. David Wishart di Universitas Alberta, Kanada untuk kebutuhan akademik. Kemudian pada 2011, DrugBank berkolaborasi dengan *The Metabolomics Innovation Center* (TMIC), fasilitas riset kesehatan pemerintah Kanada. Dataset terdiri dari empat data yang berbeda, yaitu daftar obat, daftar protein, struktur kimia obat dan rantai protein. Data yang diambil adalah data *approved* (dianggap aman dan efektif berdasarkan uji dan evaluasi badan regulasi negara-negara maju seperti FDA di Amerika Serikat) dan *rejected* (dilarang secara hukum di sebagian besar negara maju karena potensi efek yang membahayakan).

3.2.3 Identifikasi dan Analisis Permasalahan

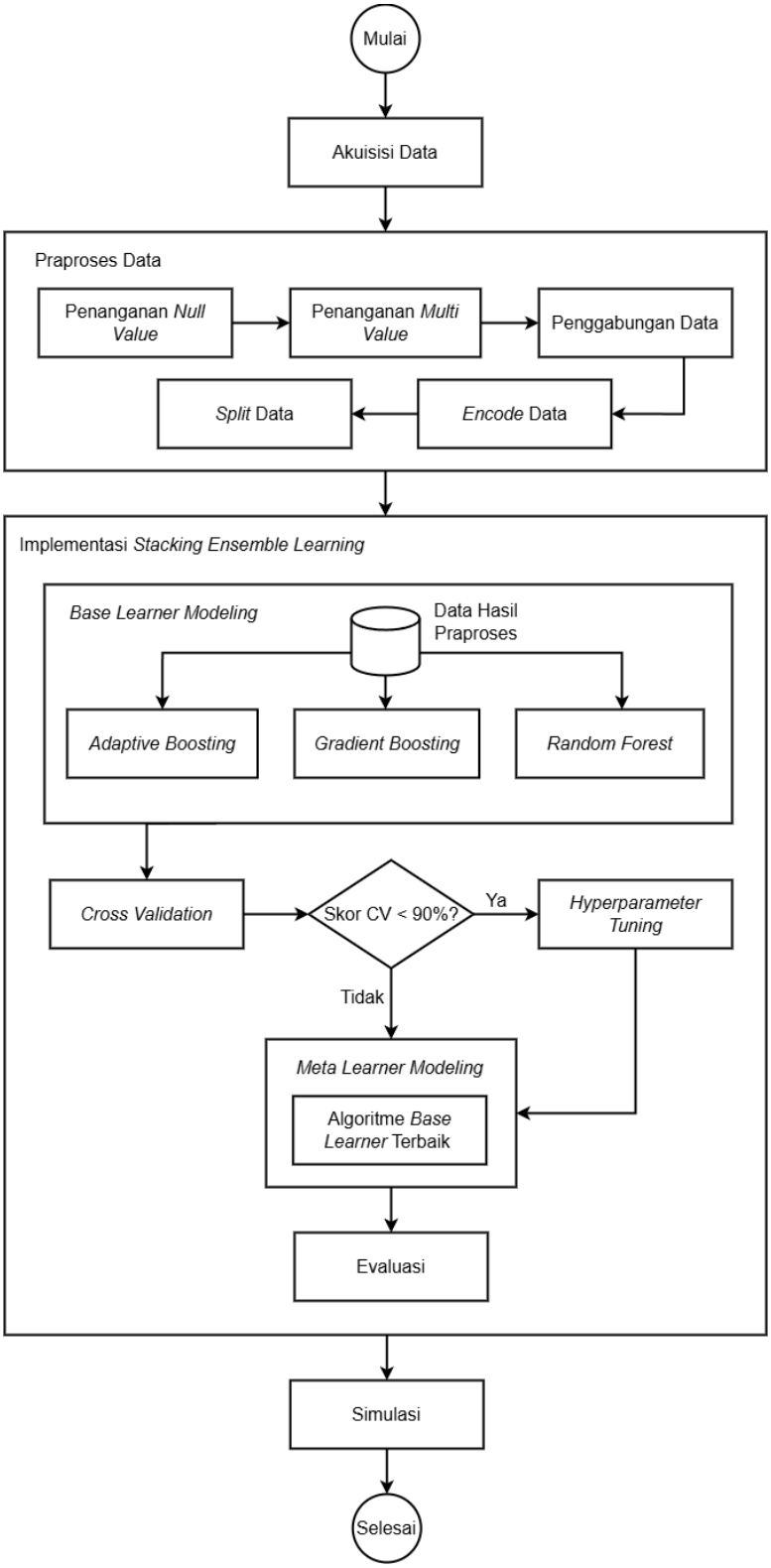
Mengidentifikasi permasalahan dan analisis terhadap model yang diterapkan untuk menyelesaikan permasalahan. Analisis dilakukan berdasarkan informasi yang telah didapat ketika melakukan studi literatur.

3.2.4 Implementasi *Stacking Ensemble Learning* (SEL)

Diagram alir implementasi ditunjukkan pada gambar 3.2. Pertama, dataset dieksplorasi untuk mendapat *insight* dan informasi yang dibutuhkan, seperti statistik atau pengecekan adanya *null value*. Kemudian, dilakukan praproses data yang terdiri dari beberapa tahapan, dimulai dari penanganan *null value* dengan menghilangkan kolom dengan banyak *null value* dan menghilangkan baris-baris yang terdapat *null value*. Untuk kolom yang terdapat *multi value*, data dipisah menjadi baris-baris sehingga hanya terdapat *value* tunggal setiap barisnya. Selanjutnya, dilakukan *merger* data obat dan protein sehingga menjadi dataset tunggal untuk memudahkan proses selanjutnya. Data yang didapat dilakukan *encode* agar dapat dibaca oleh algoritme. Setelahnya data di-*split* menjadi data *training* dan *testing*.

Setelah praproses data dilakukan, selanjutnya adalah pembuatan dan penerapan model *base learner* pada data *training*. Terdapat tiga *base learner* yang

akan digunakan, yaitu Adaptive Boosting (AdaBoost), Gradient Boosting (GBoost), dan Random Forest (RF). Apabila akurasi masing-masing algoritme di bawah 90%, dilakukan *hyperparameter tuning* dan dilakukan *cross validation* untuk mendapat model yang optimal. Satu algoritme dengan hasil *tuning* paling optimal kemudian dijadikan sebagai *final estimator* pada *meta learner* dan dua lainnya sebagai *base estimator*. *Meta learner* dievaluasi hasilnya dengan metriks evaluasi dan *confusion matrix* serta disimulasikan menggunakan data sebelum praproses.



Gambar 3.2 Diagram Alir Implementasi *Stacking Ensemble Learning* (SEL)

3.2.5 Analisis Hasil dan Penarikan Kesimpulan

Hasil akhir dari penerapan SEL dianalisis faktor-faktor yang memengaruhi hasil penelitian. Dari beberapa faktor yang ditemukan, ditarik kesimpulan mengenai prediksi DTI serta solusi yang paling tepat untuk mendapatkan hasil terbaik.

3.2.6 Penyusunan Tesis dan Artikel Ilmiah

Disusun laporan penelitian dalam bentuk buku tugas akhir sebagai dokumentasi dari seluruh rangkaian penelitian mulai dari masukan, proses, hingga luaran penelitian. Dokumentasi pertama adalah buku tesis sesuai pedoman dari ITS dan yang kedua adalah artikel ilmiah sesuai dengan format penerbit skala nasional atau internasional.

(*Halaman ini sengaja dikosongkan*)

BAB 4
PERANCANGAN DAN IMPLEMENTASI

Pada bab ini diuraikan tahap perancangan awal yang diperlukan untuk melakukan proses penelitian dan implementasi *Stacking Ensemble Learning*. Bab ini memuat akuisisi dan eksplorasi data, praproses data, serta melakukan implementasi model *base learner* dan *meta learner* dari tiga algoritme yang kemudian dibandingkan satu sama lain.

4.1 Akuisisi dan Eksplorasi Data

Sebelum melakukan eksplorasi, perlu dilakukan impor *library* dan dataset. *Function* untuk mengimpor *library* dan dataset ditunjukkan pada kode program 4.1.

```
#Import EDA Library
import numpy as np
import pandas as pd
from matplotlib import pyplot as plt
import seaborn as sns
import joblib
from sklearn.preprocessing import OneHotEncoder
from sklearn.model_selection import StratifiedKFold, KFold, train_test_split, cross_val_score
from sklearn.ensemble import AdaBoostClassifier
from sklearn.ensemble import GradientBoostingClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.ensemble import StackingClassifier
from sklearn.model_selection import GridSearchCV
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score, confusion_matrix

#Import Dataset
from google.colab import drive
drive.mount('/content/gdrive/')
```

Kode Program 4.1 *Function* untuk Mengimpor *Library* dan Dataset

Data yang didapat sesuai bab 3.2.2 dieksplorasi menggunakan *python* via *Google Colab*, diawali dengan eksplorasi data struktur kima obat dan rantai protein. Karena kedua dataset tersebut dalam format FASTA, maka digunakan *library* Biopython dengan modul SeqIO untuk dapat membaca data *sequence* seperti pada kode program 4.2. Hasil pembacaan dataset ada pada tabel 4.1.

```python
from Bio import SeqIO

def load_fasta(file_path):
    """
    Load FASTA file and extract sequence IDs and sequences.
    """
    sequences = {'ID': [], 'Sequence': []}
    for record in SeqIO.parse(file_path, 'fasta'):
        sequences['ID'].append(record.id)
        sequences['Sequence'].append(str(record.seq))
    return pd.DataFrame(sequences)
```

Kode Program 4.2 *Function* untuk Membaca Data *Sequence* Obat dan Protein

Tabel 4.1 Cuplikan Struktur Kimia Obat dan Rantai Protein

Data Obat		
No	**ID**	***Sequence***
1	DB00001	LTYTDCTESGQNLCLCEGSNVCGQGNKCILGS…
2	DB00002	QVQLKQSGPGLVQPSQSLSITCTVSGFSLTNY…
3	DB00003	DILLTQSPVILSVSPGERVSFSCRASQSIGTNIH…
4	DB00004	LKIAAFNIQTFGETKMSNATLVSYIVQILSRYD…
5	DB00005	MGADDVVDSSKSFVMENFSSYHGTKPGYVD…
Data Protein		
No	**ID**	***Sequence***
1	P45059	MVKFNSSRKSGKSKKTIRKLTAPETVKQNKPQ…
2	P19113	MMEPEEYRERGREMVDYICQYLSTVRERRVT…
3	Q9UI32	MRSMKALQKALSRAGSHCGRGGWGHPSRSP…
4	P00488	MSETSRTAFGGRRAVPPNNSNAAEDDLPTVE…
5	P35228	MACPWKFLFKTKFHQYAMNGEKDINNNVEK…

Untuk daftar obat dan protein, masing-masing terdiri dari dua subkategori, yaitu *approved* dan *rejected*. Pembacaan data menggunakan *library pandas*. Untuk data *approved*, daftar obat terdiri dari 4219 baris dan 13 kolom, sedangkan daftar obat terdiri dari 21599 baris dan 5 kolom. Kemudian untuk data *rejected*, daftar obat terdiri dari 779 baris dan 13 kolom, sedangkan daftar obat terdiri dari 2447 baris dan 5 kolom. Tabel 4.2 dan 4.3 menunjukkan cuplikan daftar obat dan protein.

```
drug_acc_data_1 = pd.read_csv('/content/gdrive/MyDrive/Tesis/Dataset/drug_approved_carrier_uniprot.csv')
drug_acc_data_2 = pd.read_csv('/content/gdrive/MyDrive/Tesis/Dataset/drug_approved_enzyme_uniprot.csv')
drug_acc_data_3 = pd.read_csv('/content/gdrive/MyDrive/Tesis/Dataset/drug_approved_target_uniprot.csv')
drug_acc_data_4 = pd.read_csv('/content/gdrive/MyDrive/Tesis/Dataset/drug_approved_transporter_uniprot.csv')
drug_acc_data = pd.concat([drug_acc_data_1, drug_acc_data_2, drug_acc_data_3, drug_acc_data_4])
protein_acc_data_1 = pd.read_csv('/content/gdrive/MyDrive/Tesis/Dataset/protein_approved_carrier_polypeptide.csv')
protein_acc_data_2 = pd.read_csv('/content/gdrive/MyDrive/Tesis/Dataset/protein_approved_enzyme_polypeptide.csv')
protein_acc_data_3 = pd.read_csv('/content/gdrive/MyDrive/Tesis/Dataset/protein_approved_target_polypeptide.csv')
protein_acc_data_4 = pd.read_csv('/content/gdrive/MyDrive/Tesis/Dataset/protein_approved_transporter_polypeptide.csv')
protein_acc_data = pd.concat([protein_acc_data_1, protein_acc_data_2, protein_acc_data_3, protein_acc_data_4])
drug_rej_data_1 = pd.read_csv('/content/gdrive/MyDrive/Tesis/Dataset/drug_illicit_carrier_uniprot.csv')
drug_rej_data_2 = pd.read_csv('/content/gdrive/MyDrive/Tesis/Dataset/drug_illicit_enzyme_uniprot.csv')
drug_rej_data_3 = pd.read_csv('/content/gdrive/MyDrive/Tesis/Dataset/drug_illicit_target_uniprot.csv')
drug_rej_data_4 = pd.read_csv('/content/gdrive/MyDrive/Tesis/Dataset/drug_illicit_transporter_uniprot.csv')
drug_rej_data = pd.concat([drug_rej_data_1, drug_rej_data_2, drug_rej_data_3, drug_rej_data_4])
protein_rej_data_1 = pd.read_csv('/content/gdrive/MyDrive/Tesis/Dataset/protein_illicit_carrier_polypeptide.csv')
protein_rej_data_2 = pd.read_csv('/content/gdrive/MyDrive/Tesis/Dataset/protein_illicit_enzyme_polypeptide.csv')
protein_rej_data_3 = pd.read_csv('/content/gdrive/MyDrive/Tesis/Dataset/protein_illicit_target_polypeptide.csv')
protein_rej_data_4 = pd.read_csv('/content/gdrive/MyDrive/Tesis/Dataset/protein_illicit_transporter_polypeptide.csv')
protein_rej_data = pd.concat([protein_rej_data_1, protein_rej_data_2, protein_rej_data_3, protein_rej_data_4])
```

Kode Program 4.3 *Function* untuk Membaca Daftar Obat dan Protein

Tabel 4.2 Cuplikan Daftar Obat

Obat *Approved*					
No	DrugBank ID	Name	…	UniProt ID	UniProt Name
1	DB00059	Pegaspargase	…	P05543	Thyroxine-bind…
2	DB00070	Hyaluronidase...		P02768	Serum albumin
3	DB00080	Daptomycin	…	P02763	Alpha-1-acid…
Obat *Rejected*					
1	DB00237	Butabarbital	…	P02768	Serum albumin
2	DB00349	Clobazam	…	P02763	Alpha-1-acid…
3	DB12319	Benzbromarone	…	O95342	Bile salt export…

Tabel 4.3 Cuplikan Daftar Protein

Protein *Approved*					
No	ID	Name	…	Species	Drug IDs
1	16	Coagulation factor VIII		Humans	DB09130
2	85	Ferritin light chain		Humans	DB06784
3	130	Hemoglobin subunit alpha		Humans	DB09140
Protein *Rejected*					
1	530	Serum albumin	…	Humans	DB00237; DB00327;...

2	925	Alpha-1-acid glycoprotein 1	…	Humans	DB00349; DB00683;…
3	3777	Beta-2-microglobulin	…	Humans	DB11130

Pada cuplikan daftar protein, terdapat satu baris yang teridentifikasi memiliki dua nilai pada kolom ‘Drug IDs’. Ini mengindikasikan bahwa satu protein dapat berinteraksi dengan lebih dari satu obat.

4.2 Praproses Data

Pada tahap ini dilakukan praproses data untuk memperbesar nilai akurasi model dan mengurangi *error* yang mungkin terjadi saat pembuatan model. Praproses yang dilakukan dibagi menjadi tiga, yaitu praproses untuk struktur kimia obat dan rantai protein, praproses untuk daftar obat dan protein, serta praproses untuk data gabungan.

4.2.1 Praproses untuk Struktur Kimia Obat dan Rantai Protein

Praproses diawali dengan menerapkan k-mers untuk merepresentasikan *sequence* obat dan protein dengan membaginya menjadi lebih pendek (*subsequence*). Dengan k-mers dapat diperoleh informasi yang lebih akurat (Jiang et al., 2023). Nilai k-mers yang dipakai adalah tiga dikarenakan umumnya *sequence* terdiri dari tiga huruf (k = 3).

```
def generate_kmers(sequence, k):
    """
    Generate k-mers (subsequences of length k) from a given sequence.
    """
    kmers = [sequence[i:i+k] for i in range(len(sequence) - k + 1)]
    return kmers

def sequence_to_kmer_features(sequence_data, k):
    """
    Convert sequence data into k-mer features.
    """
    kmer_features = []
    for sequence in sequence_data['Sequence']:
        kmers = generate_kmers(sequence, k)
        kmer_features.append(' '.join(kmers))
    return pd.DataFrame({'ID': sequence_data['ID'], 'Kmer_Features': kmer_features})
```

Kode Program 4.4 *Function* untuk Menerapkan K-Mers pada Struktur Kimia Obat dan Rantai Protein

Dengan membaginya menjadi *subsequence*, maka similaritas antara obat dan protein dapat dihitung. Penghitungan similaritas menggunakan *Jaccard similarity*. *Jaccard similarity* didefinisikan sebagai proporsi ukuran irisan (*intersection*) dibanding ukuran gabungan (*union*) dua sampel data (Verma & Aggarwal, 2020). *Jaccard similarity* digunakan dikarenakan presisinya yang tinggi dan mudah untuk diinterpretasikan melalui normalisasi antara 0 dan 1 (Torab-Miandoab et al., 2023). Penghitungan similaritas sesuai dengan kode program 4.5.

```
def jaccard_similarity(set1, set2):
    """
    Calculate Jaccard similarity between two sets.
    """
    intersection = len(set1 & set2)
    union = len(set1 | set2)
    return intersection / union

def calculate_jaccard_similarity(protein_structure_data, drug_structure_data):
    """
    Calculate Jaccard similarity for all pairs of sequences in two DataFrames.
    Assumes that the DataFrames have 'ID' and 'Kmer_Features' columns.
    """
    similarity_matrix = {}
    for i, row1 in protein_structure_data.iterrows():
        similarity_matrix[row1['ID']] = {}
        for j, row2 in drug_structure_data.iterrows():
            kmer_set1 = set(row1['Kmer_Features'].split())
            kmer_set2 = set(row2['Kmer_Features'].split())
            similarity_matrix[row1['ID']][row2['ID']] = jaccard_similarity(kmer_set1, kmer_set2)
    return pd.DataFrame(similarity_matrix)
```

Kode Program 4.5 *Function* untuk Menghitung Similaritas Obat dan Protein

Dataframe hasil penghitungan similaritas berupa matriks. Untuk itu diperlukan pengubahan bentuk *dataframe* menjadi tabel agar dapat digabungkan dengan daftar obat dan protein seperti pada kode program 4.6 berikut.

```
# Calculate Jaccard similarity for the merged data
data_with_similarity = calculate_jaccard_similarity(protein_kmer_features, drug_kmer_features)
protein_drug_interactions = data_with_similarity

# Check the dataset structure
print(protein_drug_interactions.head())
print(protein_drug_interactions.shape)

# Reshape the DataFrame using stack
data_stacked = protein_drug_interactions.stack()

# Convert the Series to a DataFrame
data_melted = data_stacked.reset_index()

# Rename the columns
data_melted.columns = ['Drug ID', 'Protein ID', 'Similarity']

# Display the reshaped DataFrame
print(data_melted.head())
print(data_melted.shape)
```

Kode Program 4.6 *Function* untuk Mengubah Bentuk *Dataframe*

4.2.2 Praproses untuk Daftar Obat dan Protein

Langkah pertama dalam praproses ini adalah menghapus semua ID dari situs lain dan menyisakan ID dan informasi lain dari obat dan protein. Pada daftar protein, dikarenakan kolom ‘Drug IDs’ memiliki nilai jamak, maka dilakukan pemecahan data sehingga semua baris memiliki nilai tunggal.

```
protein_acc_data['Drug IDs'] = protein_acc_data['Drug IDs'].str.split(';')
protein_acc_data = protein_acc_data.explode('Drug IDs')
protein_acc_data = protein_acc_data.reset_index(drop=True)
protein_rej_data['Drug IDs'] = protein_rej_data['Drug IDs'].str.split(';')
protein_rej_data = protein_rej_data.explode('Drug IDs')
protein_rej_data = protein_rej_data.reset_index(drop=True)
```

Kode Program 4.7 *Function* untuk Memecah Data Jamak Menjadi Data Tunggal

Selanjutnya beberapa nama kolom diganti sehingga masing-masing daftar obat dan protein (*approved* dan *rejected*) dapat di-*merge* dengan data similaritas. Sebelum di-*merge*, daftar obat dan protein *approved* ditambahkan kolom ‘Result’ dengan nilai ‘Approved’, sedangkan daftar obat dan protein *rejected* ditambahkan kolom ‘Result’ dengan nilai ‘Rejected’. Hasil akhir penggabungan data dapat dilihat pada tabel 4.4, dengan 1616 baris dan 8 kolom.

Tabel 4.4 Data Obat dan Protein Setelah Praproses

Drug ID	Drug Name	Drug Type	Protein Name	Protein ID	Species	Similarity	Result
DB00008	Peginterferon alfa-2a	BiotechDrug	Interferon alpha/beta receptor 1	P17181	Humans	0.024060	Approved
DB00055	Drotrecogin alfa	BiotechDrug	Vitamin K-dependent protein S	P07225	Humans	0.020520	Rejected
DB00028	Human immunoglobulin G	BiotechDrug	High affinity immunoglobulin gamma Fc receptor IB	Q92637	Humans	0.046875	Approved
DB00055	Drotrecogin alfa	BiotechDrug	Vitamin K-dependent protein S	P07225	Humans	0.020520	Approved
DB00028	Human immunoglobulin G	BiotechDrug	Complement C5	P01031	Humans	0.080863	Approved
...	...	...	...	...	...	...	...
DB14012	Burosumab	BiotechDrug	Fibroblast growth factor 23	Q7Z4T2	Humans	0.002941	Approved
DB15283	Tebentafusp	BiotechDrug	Melanocyte protein PMEL	P40967	Humans	0.021333	Approved
DB09046	Metreleptin	BiotechDrug	Leptin receptor	P48357	Humans	0.029152	Approved
DB00012	Darbepoetin alfa	BiotechDrug	Erythropoietin receptor	P19235	Humans	0.048622	Approved
DB00017	Salmon calcitonin	BiotechDrug	Calcitonin receptor	P30988	Humans	0.007905	Approved

4.2.3 Praproses Data Gabungan

Data yang telah digabung kemudian dibagi menjadi data berisi variabel independen (disimbolkan dengan data X) dan data berisi variabel dependen (disimbolkan dengan data y). Variabel-variabel data X adalah variabel non-*identifier*, antara lain 'Drug Type', 'Species', dan 'Similarity', sedangkan variabel data y adalah variabel target 'Result'.

Karena data gabungan masih berupa *string*, dilakukan *encode* menggunakan *one hot encoder*. Setelah dilakukan *encoding*, Data X dan y dibagi menjadi data *training* dan *testing* dengan rasio 80:20. *Function* untuk *encoding* dan *splitting* data ada pada kode program 4.8.

```
# Prepare data for the stacking ensemble model
X = merged_data[['Drug Type', 'Species', 'Similarity']]
y = merged_data['Result']

# Convert categorical features to numerical using OneHotEncoder
one_hot_encoder = OneHotEncoder(handle_unknown='ignore')
X_encoded = one_hot_encoder.fit_transform(X)

# Save the encoder to Google Drive
encoder_path = '/content/gdrive/MyDrive/Tesis/Model/one_hot_encoder.pkl'
joblib.dump(one_hot_encoder, encoder_path)

# Split dataset into training set and test set
X_train, X_test, y_train, y_test = train_test_split(X_encoded, y, test_size=0.2)
```

Kode Program 4.8 *Function* untuk *Encoding* dan *Splitting* Data

Sebelum diaplikasikan model, dilakukan *balancing* data (Hasan Mahmud et al., 2020). Teknik yang digunakan adalah *oversampling* menggunakan SMOTE

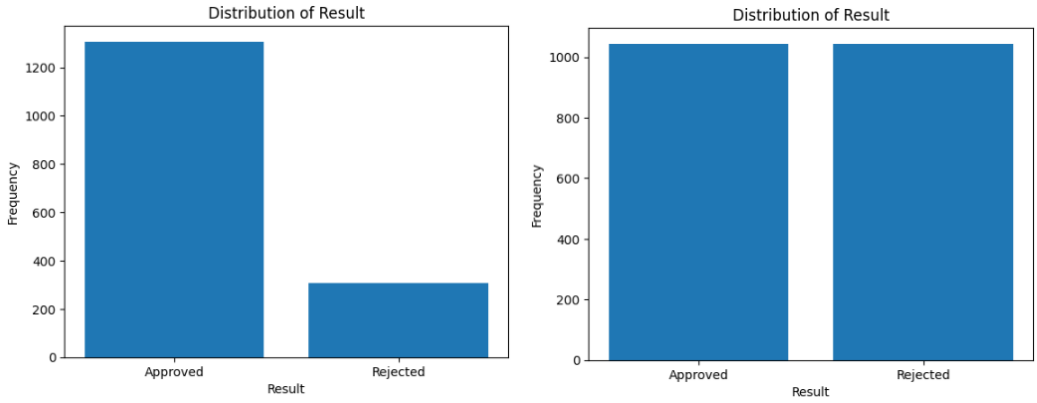
karena jumlah nilai ‘Approved’ dan ‘Rejected’ pada kolom ‘Result’ tidak seimbang. Model *base learner* untuk data sebelum dan setelah *oversampling* dibandingkan untuk membangun model SEL.

```
from imblearn.over_sampling import SMOTE

# Instantiate the SMOTE object
smote = SMOTE(random_state=42)

# Apply SMOTE to generate synthetic samples
X_train_resampled, y_train_resampled = smote.fit_resample(X_train, y_train)
```

Kode Program 4.9 *Function* untuk *Oversampling* Data



Gambar 4.1 Perbandingan Sebelum dan Sesudah *Oversampling*

4.3 Pemodelan Klasifikasi

Arsitektur model SEL dibagi dua, yaitu *base learner* dan *meta learner*. *Base learner* menggunakan tiga algoritme *ensemble learning*, yaitu Adaptive Boosting, Gradient Boosting, dan Random Forest. Masing-masing algoritme dilatih dengan dua jenis data, yaitu data tanpa *oversampling* dan data dengan *oversampling*. Kedua hasil *training* dievaluasi menggunakan *stratified K-Fold cross-validation* (CV). *Stratified K-Fold* digunakan untuk mengatasi distribusi kelas yang tidak seimbang dengan memastikan setiap *fold* mengandung proporsi dua nilai biner yang sama (Prusty et al., 2022). Parameter yang digunakan tiap algoritme adalah *default*.

```
# Create AdaBoost classifier object
abc = AdaBoostClassifier(n_estimators=50, learning_rate=1, random_state=42)
gbc = GradientBoostingClassifier(n_estimators=50, learning_rate=1, random_state=42)
rfc = RandomForestClassifier(n_estimators=100, max_depth=None, random_state=42)

# Perform k-fold cross-validation
skf = StratifiedKFold(n_splits=5)   # Define the value of k for k-fold cross-validation
scores1 = cross_val_score(abc, X_train, y_train, cv=skf)
scores2 = cross_val_score(gbc, X_train, y_train, cv=skf)
scores3 = cross_val_score(rfc, X_train, y_train, cv=skf)

# Print the cross-validation scores
print("Cross-validation scores:", scores1)
print("Mean of CV scores:", scores1.mean())
print("Cross-validation scores:", scores2)
print("Mean of CV scores:", scores2.mean())
print("Cross-validation scores:", scores3)
print("Mean of CV scores:", scores3.mean())
```

Kode Program 4.10 *Function* untuk Pemodelan *Base Learner*

Apabila skor akurasi CV rata-rata algoritme kurang dari 90%, maka dilakukan *hyperparameter tuning*. Masing-masing parameter diuji dengan tiga nilai berbeda hingga memunculkan nilai terbaik. Skor akurasi CV rata-rata dari nilai parameter algoritme terbaik dibandingkan dengan skor akurasi CV rata-rata algoritme *default*.

```
from sklearn.model_selection import GridSearchCV

# Define the parameter grid for grid search
param_grid = {
    'n_estimators': [50, 100, 200],  # Define the values to be searched for n_estimators
    'learning_rate': [0.1, 0.01]  # Define the values to be searched for learning_rate
}

# Create GridSearchCV object
grid_search = GridSearchCV(abc, param_grid, cv=skf)

# Perform grid search
grid_search.fit(X_train_resampled, y_train_resampled)

# Get the best parameters and best score
best_params = grid_search.best_params_
best_score = grid_search.best_score_

# Train Adaboost Classifier with the best parameters
model1 = AdaBoostClassifier(**best_params)
model1.fit(X_train_resampled, y_train_resampled)

# Predict the response for test dataset
y_pred = model1.predict(X_test)

# Print the best parameters and best score
print("Best parameters:", best_params)
print("Best score:", best_score)
```

Kode Program 4.11 Contoh *Function* untuk *Hyperparameter Tuning* pada AdaBoost

Tiap algoritme dengan *tuning* terbaik kemudian dimasukkan ke *meta learner*, di mana algoritme dengan skor akurasi CV rata-rata tertinggi menjadi *final estimator*, sedangkan algoritme lainnya menjadi *base estimator* seperti pada kode program 4.12.

```
# Define the base models for stacking
base_models = [
    ('AdaBoost', AdaBoostClassifier(n_estimators=200, learning_rate=0.1)),
    ('GradientBoost', GradientBoostingClassifier(n_estimators=200, learning_rate=0.1)),
]

# Define the stacking ensemble model
stacked = StackingClassifier(
    estimators=base_models,
    final_estimator=RandomForestClassifier(n_estimators=50),
    cv=skf
)

# Perform k-fold cross-validation
skf = StratifiedKFold(n_splits=5)  # Define the value of k for k-fold cross-validation
scores = cross_val_score(stacked, X_train, y_train, cv=skf)

# Train the stacking ensemble model
rfc_final_model = stacked.fit(X_train, y_train)

# Save the model to a file in Google Drive
joblib.dump(rfc_final_model, rfc_stacked_model_path)

# Load the model from the file in Google Drive
loaded_model = joblib.load(rfc_stacked_model_path)

# Predict the response for the test dataset using the loaded model
y_pred = loaded_model.predict(X_test)

# Print the cross-validation scores
print("Cross-validation scores:", scores)
print("Mean of CV scores:", scores.mean())
```

Kode Program 4.12 Contoh *Function* untuk Pemodelan *Meta Learner*

Model klasifikasi final SEL dibagi menjadi tiga, yaitu model pertama yang dilatih menggunakan data tanpa *oversampling* dan tuning, model kedua yang dilatih menggunakan data tanpa *oversampling* dan dengan *tuning*, serta model ketiga yang dilatih menggunakan data dengan *oversampling* dan *tuning*. Masing-masing model dievaluasi dengan metriks klasifikasi dan *confusion matrix*. Model SEL terbaik menjadi model *recommender*.

4.4 Pembuatan *Recommender*

Tahapan terakhir adalah pembuatan *recommender*. *Recommender* dibuat untuk men-*generate* daftar rekomendasi obat berdasarkan protein yang di-*input*. Parameter yang digunakan antara lain adalah *protein identifier*, data yang berisi informasi obat dan protein, model yang digunakan untuk memprediksi hasil eksperimen, *encoder* untuk mentransformasi data, dan k untuk jumlah rekomendasi

yang ingin ditampilkan. Hasil dari *recommender* divalidasi oleh pakar yang ahli di bidang farmasi. Peneliti bekerja sama dengan dosen fakultas farmasi Universitas Airlangga, Dr. apt. Wenny Putri Nilamsari, S.Farm.,Sp.FRS. untuk memvalidasi hasil *recommender*.

```
def get_top_k_recommendations(protein_identifier, merged_data, model, encoder, k):
    """
    Get top-k drug recommendations for a given protein based on similarity and approval prediction.
    Accepts either protein ID or protein name as input.
    """
    # Filter the data for the given protein_id or protein_name
    protein_data = merged_data[merged_data['Protein Name'] == protein_identifier]

    if protein_data.empty:
        print("Protein not found in the dataset.")
        return pd.DataFrame()  # Return an empty DataFrame if no protein is found

    # Encode the features using the loaded encoder
    # Ensure that the features used here match the ones used during training
    features = protein_data[['Drug Type', 'Species', 'Similarity']]  # Adjust these columns if needed
    features_encoded = encoder.transform(features)

    # Predict the approval for each pair
    protein_data = protein_data.copy()  # Avoid SettingWithCopyWarning by working on a copy
    protein_data.loc[:, 'Prediction'] = model.predict(features_encoded)

    # Sort the data by adjusted similarity in descending order and get the top-k recommendations
    top_k_recommendations = protein_data.sort_values(by='Similarity', ascending=False).head(k)
    return top_k_recommendations
```

Kode Program 4.13 *Function* untuk Pembuatan *Recommender*

BAB 5
HASIL DAN PEMBAHASAN

Pada bab ini dipaparkan dua poin utama. Pertama, dijelaskan performa klasifikasi, mulai dari *base learner*, *hyperparameter tuning*, *meta learner* menggunakan skor akurasi CV rata-rata untuk *base learner* dan *hyperparameter tuning* serta metriks evaluasi dan *confusion matrix* untuk *meta learner*. Kedua, dijelaskan performa dari *recommender* yang hasilnya divalidasi oleh pakar.

5.1 Hasil Pemodelan Klasifikasi

Hasil dari *base learner*, berisi tiga algoritme dengan parameter *default* untuk melatih dua jenis data, ditampilkan pada tabel 5.1. Semua algoritme bekerja dengan lebih baik untuk data dengan *oversampling*, ditandai dengan peningkatan skor akurasi CV rata-rata. Misalnya, Adaptive Boosting (AdaBoost) menunjukkan peningkatan skor akurasi CV rata-rata dari 0,881 tanpa *oversampling* menjadi 0,921 dengan *oversampling*. Demikian pula, skor akurasi CV rata-rata Gradient Boosting (GBoost) meningkat dari 0,862 menjadi 0,929, dan skor Random Forest (RF) dari 0,861 menjadi 0,929. Hal ini menunjukkan bahwa penggunaan metode *oversampling*, yang membantu menyeimbangkan distribusi kelas dalam dataset yang tidak seimbang, secara signifikan meningkatkan performa model.

Terdapat hal lain yang dapat diamati. Apabila diperhatikan, AdaBoost memiliki skor akurasi CV rata-rata tertinggi (0,881) apabila menggunakan data tanpa *oversampling*. Di sisi lain, GBoost dan RF memiliki skor akurasi CV rata-rata tertinggi (masing-masing 0,929) apabila menggunakan data dengan *oversampling*. Ini menunjukkan bahwa meskipun AdaBoost lebih *robust* tanpa *oversampling*, GBoost dan RF lebih unggul jika menggunakan *oversampling*.

Selain itu, sensitivitas algoritme terhadap *oversampling* bervariasi. GBoost dan RF menunjukkan peningkatan yang lebih besar pada skor akurasi CV rata-rata dengan *oversampling* dibandingkan dengan AdaBoost. Skor akurasi CV rata-rata GBoost meningkat sebesar 0,067, RF sebesar 0,068, dan AdaBoost sebesar 0,040. Sensitivitas yang lebih besar ini menunjukkan bahwa GBoost dan RF efektif untuk

menangani ketidakseimbangan kelas bila dikombinasikan dengan teknik *oversampling*.

Tabel 5.1 Perbandingan Model *Base Learner*

Algoritme *Base Learner*	**Skor Akurasi CV Rata-Rata**	
	Data tanpa *Oversampling*	**Data dengan *Oversampling***
Adaptive Boosting	0,881	0,921
Gradient Boosting	0,862	0,929
Random Forest	0,861	0,929

Untuk model dengan data tanpa *oversampling*, dilakukan *hyperparameter tuning* karena skor akurasi CV rata-rata di bawah 90%. Parameter yang di-*tune* ditampilkan pada tabel 5.2. AdaBoost dan GBoost memiliki skor akurasi CV rata-rata tertinggi sebesar 0,899, yang menunjukkan performa yang serupa ketika di-*tuning*. Di sisi lain, RF memiliki skor akurasi CV rata-rata yang sedikit lebih rendah yaitu 0,883. Jika dibandingkan dengan model sebelum *hyperparameter tuning*, semua skor akurasi CV rata-rata model terjadi peningkatan. Ini menunjukkan bahwa *hyperparameter tuning* dapat meningkatkan performa model secara signifikan.

Tabel 5.2 Parameter yang Di-*Tuning* Tiap Model Beserta Skor akurasi CV

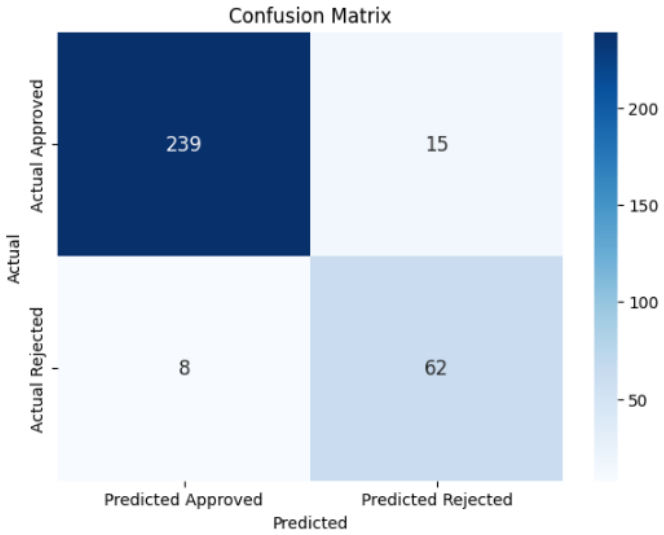
Model	**Parameter**	**Nilai**	**Nilai Terbaik**	**Skor akurasi CV awal**	**Skor akurasi CV akhir**
AdaBooost	n-estimators	50, 100, 200	50	0,881	0,899
	learning rate	1, 0.1, 0.01	0.1		
GBoost	n-estimators	50, 100, 200	100	0,862	0,899
	learning rate	1, 0.1, 0.01	0.1		
RF	n-estimators	50, 100, 200	50	0,861	0,883
	max depth	None, 5, 7	None		

Pemodelan *meta learner* dibuat dengan tiga pengaturan, yaitu tanpa *oversampling* dan *tuning* (model 1), tanpa *oversampling* namun dengan *tuning*

(model 2), dan dengan *oversampling* dan *tuning* (model 3). Ketiga model dievaluasi dengan metriks evaluasi dan *confusion matrix* dan dibandingkan. Hasil evaluasi model ditampilkan pada tabel 5.3 dan gambar 5.1-5.3.

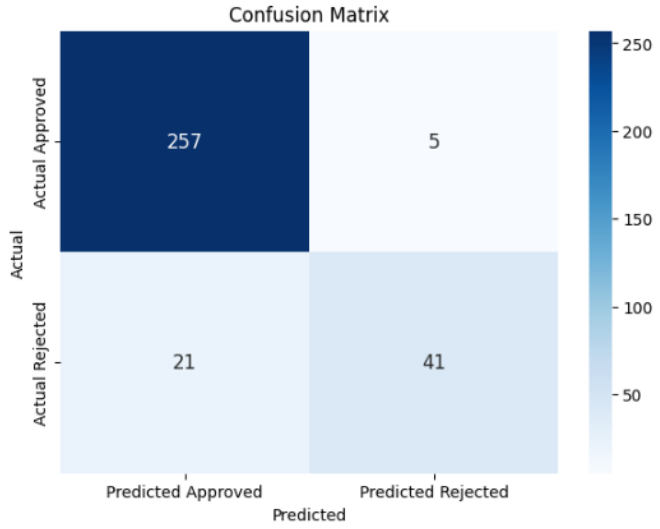
Tabel 5.3 Perbandingan Model *Meta Learner*

Model	**Metriks Evaluasi**			
	Akurasi	***Precision***	***Recall***	**Skor-F1**
Model 1	0,929	0,960	0,949	0,954
Model 2	0,920	0,924	0,980	0,951
Model 3	0,914	0,991	0,898	0,942



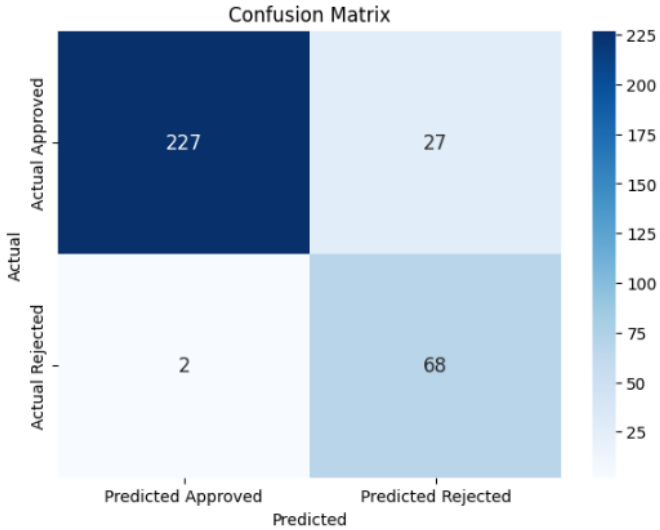
Gambar 5.1 *Confusion Matrix* untuk Model 1

Model 1 menunjukkan performa yang tinggi. Dengan akurasi 0,929, *precision* 0,960, *recall* 0,949, dan skor-F1 0,954, model 1 unggul dalam mengidentifikasi *instance* 'Approved' dan 'Rejected' secara akurat. *Confusion matrix* untuk model 1 (239 *true positive*, 15 *false negative*, 8 *false positive*, and 62 *true negative*) mendukung hasil performa dengan menunjukkan klasifikasi sebagian besar *instance* dengan benar tanpa memerlukan *oversampling* dan *tuning*.



Gambar 5.2 *Confusion Matrix* untuk Model 2

Sebaliknya, model 2 menunjukkan performa yang berbeda. Model ini memiliki nilai *recall* yang tinggi sebesar 0,980, menunjukkan peningkatan dalam mengidentifikasi *true positive*. Dengan akurasi 0,920, *precision* 0,924, dan skor-F1 0,951, model 2 masih memiliki performa yang baik, namun *confusion matrix* (257 *true positive*, 5 *false negative*, 21 *false positive*, and 40 *true negative*) menampilkan *false positive* yang lebih tinggi dibanding model 1. Hal ini menunjukkan bahwa meskipun *tuning* meningkatkan *recall* secara signifikan, hal ini dapat menyebabkan peningkatan *false positive*.



Gambar 5.3 *Confusion Matrix* untuk Model 3

Model 3 berfokus pada pemaksimalan *precision*, sehingga memiliki skor *precision* tertinggi sebesar 0,991. Namun, efeknya adalah akurasi dan *recall* yang lebih rendah, dengan skor masing-masing sebesar 0,898 dan 0,914, serta skor-F1 sebesar 0,942. *Confusion matrix* model 3 (227 *true positive*, 27 *false negative*, 2 *false positive*, and 68 *true negative*) menunjukkan bahwa meskipun model ini sangat efektif dalam meminimalkan *false positive*, model ini menghasilkan jumlah *false negative* yang lebih tinggi dibandingkan model lainnya.

Model diuji coba tanpa menggunakan variabel 'Similarity' untuk membuktikan apakah variabel tersebut berpengaruh terhadap model atau tidak. Tabel 5.4 menunjukkan hasil akurasi CV rata-rata dari *base learner* yang membuktikan bahwa variabel tersebut berpengaruh secara signifikan ketika ditambahkan ke dalam model *base learner*.

Tabel 5.4 Hasil Akurasi CV Rata-Rata untuk Data Tanpa dan Dengan Variabel 'Similarity'

Algoritme *Base Learner*	**Skor Akurasi CV Rata-Rata Tanpa 'Similarity'**		**Skor Akurasi CV Rata-Rata Dengan 'Similarity'**	
	Data tanpa *Oversampling*	**Data dengan *Oversampling***	**Data tanpa *Oversampling***	**Data dengan *Oversampling***
AdaBoost	0.810	0.509	0,881	0,921
GBoost	0.810	0.509	0,862	0,929
RF	0.810	0.508	0,861	0,929

Kemudian algoritme *base learner* dibandingkan dengan algoritme yang lebih sederhana. Algoritme yang digunakan adlaah Decision Tree dan Logistic Regression. Hasilnya ditunjukkan pada tabel 5.5 yang menunjukkan bahwa dengan algoritme yang sederhana, hasil akurasi CV rata-ratanya signifikan dan mirip dengan algoritme yang digunakan pada penelitian ini.

Tabel 5.5 Perbandingan Algoritme Penelitian dengan Algoritme Sederhana

Algoritme *Base Learner*	Skor Akurasi CV Rata-Rata	
	Data tanpa *Oversampling*	Data dengan *Oversampling*
DT	0.865	0.931
LR	0.880	0.930
AdaBoost	0,881	0,921
GBoost	0,862	0,929
RF	0,861	0,929

5.2 Hasil *Recommender*

Recommender diuji dengan menampilkan obat yang berpasangan dengan protein yang di-*input*. Setiap *entry* divalidasi satu per satu oleh Dr. apt. Wenny Putri Nilamsari, S.Farm.,Sp.FRS. selaku pakar apakah pasangan obat-protein *approved* atau *rejected*, didasarkan atas saran yang diberikan oleh pakar. Cuplikan hasil rekomendasi yang telah divalidasi ditampilkan pada tabel 5.4, dan semua hasilnya ditampilkan pada lampiran.

Tabel 5.6 Cuplikan Hasil *Recommender*

Protein yang Di-*input*	Pasangan Obat	Hasil Uji Model	Hasil Validasi Pakar
Erythropoietin receptor	Darbepoetin alfa	*Approved*	*Approved*
Fibrinogen gamma chain	Alteplase	*Approved*	*Approved*
Glucagon-like peptide 2 receptor	Glucagon	*Approved*	*Approved*
Growth hormone receptor	Somatotropin	*Approved*	*Approved*
Proteinase-activated receptor 1	Streptokinase	*Approved*	*Approved*
Macrophage metalloelastase	Urokinase	*Rejected*	*Rejected*

Protein yang Di-*input*	**Pasangan Obat**	**Hasil Uji Model**	**Hasil Validasi Pakar**
Vascular endothelial growth factor A	Bevacizumab	*Approved*	*Approved*
T-lymphocyte activation antigen CD80	Abatacept	*Approved*	*Approved*
Prolactin receptor	Somatotropin	*Approved*	*Rejected*
Plasma serine protease inhibitor	Urokinase	*Rejected*	*Rejected*

Tabel tersebut menunjukkan proses validasi rekomendasi obat melalui pengujian model dan tinjauan pakar. Konsistensi antara hasil validasi model dan pakar yang cukup tinggi menunjukkan efektivitas dan kredibilitas rekomendasi obat yang diberikan. Artinya, sistem *recommender* telah terkalibrasi dengan cukup baik dan selaras dengan data komputatif dan penilaian pakar. Namun, ada beberapa pasangan obat-protein yang salah diprediksi, misalkan Somatotropin dengan Prolactin receptor. Somatotropin berfungsi untuk mengikat *growth hormone receptor*, sedangkan prolactin bukan termasuk golongan *growth hormone*. Contoh lain adalah pasangan Liraglutide dan Neprilysin. Liraglutide berfungsi untuk mengikat Glucagon like peptide 1, sedangkan Neprilysin diikat oleh Sacubitril. Artinya model tidak dapat 100% akurat.

(*Halaman ini sengaja dikosongkan*)

BAB 6
KESIMPULAN DAN SARAN

Pada bab ini membahas rangkuman hasil dari penelitian yang telah dilakukan yang menjawab rumusan masalah dan tujuan penelitian serta saran-saran yang diajukan untuk penelitian serupa di masa depan.

6.1 Kesimpulan

Penelitian ini berfokus pada prediksi DTI menggunakan pendekatan SEL dengan membagi tahapan menjadi tiga tahap besar: perumusan masalah, pengembangan model, dan pembuatan laporan. Tinjauan pustaka mengidentifikasi kesenjangan penelitian dan membantu penentuan metode yang akan digunakan. DrugBank dipilih sebagai sumber data utama karena koleksi informasi obat dan proteinnya yang komprehensif.

Proses akuisisi data melibatkan pengumpulan kumpulan data dari DrugBank, termasuk daftar obat, daftar protein, struktur kimia obat dan rantai protein. Praproses data mencakup penanganan *null value*, penggabungan data obat dan protein, dan *encode* data untuk pelatihan model. Tiga *base learner* (Adaptive Boosting, Gradient Boosting, dan Random Forest) digunakan pada dataset. Model dievaluasi dengan dan tanpa *oversampling* untuk mengatasi ketidakseimbangan kelas, dan *hyperparameter tuning* dilakukan untuk meningkatkan performa.

Penelitian ini menghasilkan beberapa temuan. Semua algoritma menunjukkan peningkatan performa dengan *oversampling*, ditandai dengan peningkatan skor akurasi CV rata-rata AdaBoost dari 0,881 menjadi 0,921, GBoost dari 0,862 menjadi 0,929, dan RF dari 0,861 menjadi 0,929. Sensitivitas terhadap *oversampling* bervariasi, dengan GBoost dan RF menunjukkan peningkatan yang lebih signifikan dibandingkan AdaBoost, dibuktikan dengan peningkatan masing-masing sebesar 0,067 dan 0,068, dibandingkan AdaBoost sebesar 0,040. *Hyperparameter* tuning untuk model tanpa *oversampling* mengakibatkan skor akurasi CV rata-rata meningkat menjadi 0,899 untuk AdaBoost dan GBoost, dan menjadi 0,883 untuk RF. Evaluasi model *meta learner* menunjukkan bahwa model 1, tanpa *oversampling* dan *tuning*, memiliki skor akurasi tertinggi sebesar 0,929

dengan *precision* sebesar 0,960, *recall* sebesar 0,949, dan skor-F1 sebesar 0,954. Model 2, dengan *tuning* tetapi tanpa *oversampling*, memiliki *recall* yang tinggi sebesar 0,976 tetapi meningkatkan *false positive*. Model 3, yang menggabungkan *oversampling* dan *tuning*, memiliki skor *precision* tertinggi yaitu 0,991 tetapi memiliki skor *recall* yang lebih rendah yaitu 0,898. Ini menjadikan model 1 sebagai model terbaik dan digunakan untuk sistem *recommender*.

Sistem *recommender* yang dikembangkan telah divalidasi oleh pakar, menunjukkan konsistensi yang cukup tinggi. Contohnya adalah prediksi untuk Erythropoietin Receptor dengan Darbepoetin Alfa serta Fibrinogen Gamma Chain dengan Alteplase, keduanya *approved* oleh pakar.

6.2 Keterbatasan

Meskipun penelitian ini mencapai hasil yang signifikan, terdapat beberapa keterbatasan. Pertama, hasil pemodelan *meta learner* SEL menunjukkan bahwa model 1 (tanpa *oversampling* dan *tuning*) memiliki performa terbaik dibandingkan model 2 (tanpa *oversampling* namun dengan *tuning*) dan model 3 (dengan *oversampling* dan *tuning*), meskipun kedua teknik tersebut dimaksudkan untuk menyempurnakan model. Hal ini menunjukkan bahwa generalisasi model 2 dan 3 lebih rendah sehingga perbaikan tidak memberikan hasil yang diharapkan. Kedua, sistem *recommender* diuji dan divalidasi hanya pada 40 *entry* oleh seorang pakar karena keterbatasan familiaritas terhadap obat luar negeri, sehingga berpotensi mempengaruhi generalisasi hasil. Ukuran sampel tersebut mungkin tidak mewakili keseluruhan dataset, sehingga berpotensi menimbulkan bias dalam evaluasi performa sistem.

6.3 Saran

Ada beberapa area yang dapat dieksplorasi untuk penelitian lebih lanjut. Pertama, hasil dari model *meta learner* SEL. Penelitian selanjutnya dapat mengeksplorasi teknik *balancing* alternatif atau menggunakan parameter yang berbeda untuk lebih memahami dan mengoptimalkan model. Mengeksplorasi penggunaan dataset yang lebih beragam dari berbagai sumber seperti PubChem atau ChEMBL dapat meningkatkan kemampuan generalisasi model. Selain itu,

menerapkan algoritme *deep learning* seperti Convolutional Neural Network (CNN) atau Recurrent Neural Network (RNN) untuk prediksi DTI dapat dipertimbangkan.

Kedua, proses validasi sistem *recommender*. Memperbanyak jumlah sampel untuk validasi lebih dari 40 *entry* dan melibatkan lebih dari satu pakar dapat mengurangi potensi bias dan memberikan penilaian yang lebih komprehensif terhadap keakuratan dan reliabilitas sistem. Dengan berfokus pada perbaikan yang telah disebutkan, penelitian ini diharapkan dapat mnjadi landasan sehingga menghasilkan prediksi DTI yang lebih akurat dan berkontribusi dalam penemuan obat yang lebih efektif.

(*Halaman ini sengaja dikosongkan*)

DAFTAR PUSTAKA

Buza, K., & Peška, L. (2017). Drug–target interaction prediction with Bipartite Local Models and hubness-aware regression. *Neurocomputing*, *260*, 284–293. https://doi.org/https://doi.org/10.1016/j.neucom.2017.04.055

Carracedo-Reboredo, P., Liñares-Blanco, J., Rodríguez-Fernández, N., Cedrón, F., Novoa, F. J., Carballal, A., Maojo, V., Pazos, A., & Fernandez-Lozano, C. (2021). A review on machine learning approaches and trends in drug discovery. In *Computational and Structural Biotechnology Journal* (Vol. 19, pp. 4538–4558). Elsevier B.V. https://doi.org/10.1016/j.csbj.2021.08.011

Charbuty, B., & Abdulazeez, A. (2021). Classification Based on Decision Tree Algorithm for Machine Learning. *Journal of Applied Science and Technology Trends*, *2*(01), 20–28. https://doi.org/10.38094/jastt20165

Charoenkwan, P., Schaduangrat, N., Lio', P., Moni, M. A., Shoombuatong, W., & Manavalan, B. (2022). Computational prediction and interpretation of druggable proteins using a stacked ensemble-learning framework. *IScience*, *25*(9). https://doi.org/10.1016/j.isci.2022.104883

Chatzimparmpas, A., Martins, R. M., Kucher, K., & Kerren, A. (2021). StackGenVis: Alignment of data, algorithms, and models for stacking ensemble learning using performance metrics. *IEEE Transactions on Visualization and Computer Graphics*, *27*(2), 1547–1557. https://doi.org/10.1109/TVCG.2020.3030352

Csermely, P., Korcsmáros, T., Kiss, H. J. M., London, G., & Nussinov, R. (2013). Structure and dynamics of molecular networks: A novel paradigm of drug discovery: A comprehensive review. In *Pharmacology and Therapeutics* (Vol. 138, Issue 3, pp. 333–408). https://doi.org/10.1016/j.pharmthera.2013.01.016

Cui, S., Yin, Y., Wang, D., Li, Z., & Wang, Y. (2021). A stacking-based ensemble learning method for earthquake casualty prediction. *Applied Soft Computing*, *101*, 107038. https://doi.org/https://doi.org/10.1016/j.asoc.2020.107038

Divina, F., Gilson, A., Goméz-Vela, F., Torres, M. G., & Torres, J. F. (2018). Stacking ensemble learning for short-term electricity consumption forecasting. *Energies*, *11*(4). https://doi.org/10.3390/en11040949

Elbadawi, M., Gaisford, S., & Basit, A. W. (2021). Advanced machine-learning techniques in drug discovery. In *Drug Discovery Today* (Vol. 26, Issue 3, pp. 769–777). Elsevier Ltd. https://doi.org/10.1016/j.drudis.2020.12.003

Ganaie, M. A., Hu, M., Malik, A. K., Tanveer, M., & Suganthan, P. N. (2022). Ensemble deep learning: A review. *Engineering Applications of Artificial Intelligence*, *115*, 105151. https://doi.org/https://doi.org/10.1016/j.engappai.2022.105151

Gao, M., Zhao, L., Zhang, Z., Wang, J., & Wang, C. (2023). Using a stacked ensemble learning framework to predict modulators of protein–protein interactions. *Computers in Biology and Medicine*, *161*. https://doi.org/10.1016/j.compbiomed.2023.107032

Hasan Mahmud, S. M., Chen, W., Jahan, H., Dai, B., Din, S. U., & Dzisoo, A. M. (2020). DeepACTION: A deep learning-based method for predicting novel drug-target interactions. *Analytical Biochemistry*, *610*. https://doi.org/10.1016/j.ab.2020.113978

Jenike, K. M., Campos-Domínguez, L., Boddé, M., Cerca, J., Hodson, C. N., Schatz, M. C., & Jaron, K. S. (2024). *Guide to k-mer approaches for genomics across the tree of life*. https://arxiv.org/abs/2404.01519

Jiang, M., Shao, Y., Zhang, Y., Zhou, W., & Pang, S. (2023). A deep learning method for drug-target affinity prediction based on sequence interaction information mining. *PeerJ*, *11*. https://doi.org/10.7717/peerj.16625

Koeneman, S. H., & Cavanaugh, J. E. (2022). An improved asymptotic test for the Jaccard similarity index for binary data. *Statistics & Probability Letters*, *184*, 109375. https://doi.org/https://doi.org/10.1016/j.spl.2022.109375

Li, H., Jin, Y., Zhong, J., & Zhao, R. (2021). A Fruit Tree Disease Diagnosis Model Based on Stacking Ensemble Learning. *Complexity*, *2021*. https://doi.org/10.1155/2021/6868592

Mahdaddi, A., Meshoul, S., & Belguidoum, M. (2021). EA-based hyperparameter optimization of hybrid deep learning models for effective drug-target interactions prediction. *Expert Systems with Applications*, *185*. https://doi.org/10.1016/j.eswa.2021.115525

Mahmud, S. M. H., Chen, W., Liu, Y., Awal, M. A., Ahmed, K., Rahman, M. H., & Moni, M. A. (2021). PreDTIs: Prediction of drug-Target interactions based on multiple feature information using gradient boosting framework with data balancing and feature selection techniques. *Briefings in Bioinformatics*, *22*(5). https://doi.org/10.1093/bib/bbab046

Muslim, M. A., Nikmah, T. L., Pertiwi, D. A. A., Subhan, Jumanto, Dasril, Y., & Iswanto. (2023). New model combination meta-learner to improve accuracy prediction P2P lending with stacking ensemble learning. *Intelligent Systems with Applications*, *18*, 200204. https://doi.org/https://doi.org/10.1016/j.iswa.2023.200204

Naser, M. Z., & Alavi, A. H. (2023). Error Metrics and Performance Fitness Indicators for Artificial Intelligence and Machine Learning in Engineering and Sciences. *Architecture, Structures and Construction*, *3*(4), 499–517. https://doi.org/10.1007/s44150-021-00015-8

Prusty, S., Patnaik, S., & Dash, S. K. (2022). SKCV: Stratified K-fold cross-validation on ML classifiers for predicting cervical cancer. *Frontiers in Nanotechnology*, *4*. https://doi.org/10.3389/fnano.2022.972421

Reker, D., & Schneider, G. (2015). Active-learning strategies in computer-assisted drug discovery. In *Drug Discovery Today* (Vol. 20, Issue 4, pp. 458–465). Elsevier Ltd. https://doi.org/10.1016/j.drudis.2014.12.004

Ru, X., Ye, X., Sakurai, T., Zou, Q., Xu, L., & Lin, C. (2021). Current status and future prospects of drug–target interaction prediction. *Briefings in Functional Genomics*, *20*(5), 312–322. https://doi.org/10.1093/bfgp/elab031

Sachdev, K., & Gupta, M. K. (2019). A comprehensive review of feature based methods for drug target interaction prediction. *Journal of Biomedical Informatics*, *93*, 103159. https://doi.org/https://doi.org/10.1016/j.jbi.2019.103159

Sambasivam, G., & Opiyo, G. D. (2021). A predictive machine learning application in agriculture: Cassava disease detection and classification with imbalanced dataset using convolutional neural networks. *Egyptian Informatics Journal*, *22*(1), 27–34. https://doi.org/https://doi.org/10.1016/j.eij.2020.02.007

Sarker, I. H. (2021). Deep Learning: A Comprehensive Overview on Techniques, Taxonomy, Applications and Research Directions. In *SN Computer Science* (Vol. 2, Issue 6). Springer. https://doi.org/10.1007/s42979-021-00815-1

Sen, P. C., Hajra, M., & Ghosh, M. (2020). Supervised Classification Algorithms in Machine Learning: A Survey and Review. *Advances in Intelligent Systems and Computing*, *937*, 99–111. https://doi.org/10.1007/978-981-13-7403-6_11

Shang, Y., Gao, L., Zou, Q., & Yu, L. (2021). Prediction of drug-target interactions based on multi-layer network representation learning. *Neurocomputing*, *434*, 80–89. https://doi.org/10.1016/j.neucom.2020.12.068

Sharma, A., & Rani, R. (2018). BE-DTI': Ensemble framework for drug target interaction prediction using dimensionality reduction and active learning. *Computer Methods and Programs in Biomedicine*, *165*, 151–162. https://doi.org/10.1016/j.cmpb.2018.08.011

Shiri, F. M., Perumal, T., Mustapha, N., & Mohamed, R. (2023). A Comprehensive Overview and Comparative Analysis on Deep Learning Models: CNN, RNN, LSTM, GRU. *ArXiv Preprint ArXiv:2305.17473*.

Thafar, M. A., Olayan, R. S., Albaradei, S., Bajic, V. B., Gojobori, T., Essack, M., & Gao, X. (2021). DTi2Vec: Drug–target interaction prediction using network embedding and ensemble learning. *Journal of Cheminformatics*, *13*(1). https://doi.org/10.1186/s13321-021-00552-w

Torab-Miandoab, A., Poursheikh Asghari, M., Hashemzadeh, N., & Ferdousi, R. (2023). Analysis and identification of drug similarity through drug side effects and indications data. *BMC Medical Informatics and Decision Making*, *23*(1). https://doi.org/10.1186/s12911-023-02133-3

Verma, V., & Aggarwal, R. K. (2020). A comparative analysis of similarity measures akin to the Jaccard index in collaborative recommendations: empirical and theoretical perspective. *Social Network Analysis and Mining*, *10*(1). https://doi.org/10.1007/s13278-020-00660-9

Yao, K., Wang, X., Li, W., Zhu, H., Jiang, Y., Li, Y., Tian, T., Yang, Z., Liu, Q., & Liu, Q. (2023). Semi-supervised heterogeneous graph contrastive learning for drug–target interaction prediction. *Computers in Biology and Medicine*, *163*. https://doi.org/10.1016/j.compbiomed.2023.107199

LAMPIRAN

Lampiran 1 Tabel Hasil *Recommender*

Protein yang Di-*input*	**Pasangan Obat**	**Hasil Uji Model**	**Hasil Validasi Pakar**
Erythropoietin receptor	Darbepoetin alfa	*Approved*	*Approved*
Fibrinogen gamma chain	Alteplase	*Approved*	*Approved*
Glucagon-like peptide 2 receptor	Glucagon	*Approved*	*Approved*
Growth hormone receptor	Somatotropin	*Approved*	*Approved*
Proteinase-activated receptor 1	Streptokinase	*Approved*	*Approved*
Interferon alpha/beta receptor 1	Peginterferon alfa-2a	*Approved*	*Approved*
Tumor necrosis factor	Etanercept	*Approved*	*Approved*
Vascular endothelial growth factor A	Bevacizumab	*Approved*	*Approved*
Protein NOV homolog	Insulin human	*Approved*	*Rejected*
T-cell surface glycoprotein CD3 delta chain	Muromonab	*Approved*	*Approved*
Plasminogen activator inhibitor 1	Urokinase	*Rejected*	*Rejected*
T-lymphocyte activation antigen CD80	Abatacept	*Approved*	*Approved*
Granulocyte colony-stimulating factor receptor	Pegfilgrastim	*Approved*	*Approved*

Prolactin receptor	Somatotropin	*Approved*	*Rejected*
Macrophage metalloelastase	Urokinase	*Rejected*	*Rejected*
T-cell surface glycoprotein CD3 delta chain	Elranatamab	*Approved*	*Approved*
T-cell surface glycoprotein CD3 delta chain	Talquetamab	*Approved*	*Approved*
T-cell surface glycoprotein CD3 epsilon chain	Talquetamab	*Approved*	*Approved*
T-cell surface glycoprotein CD3 epsilon chain	Elranatamab	*Approved*	*Approved*
T-cell surface glycoprotein CD3 epsilon chain	Muromonab	*Approved*	*Approved*
T-lymphocyte activation antigen CD80	Abatacept	*Approved*	*Approved*
Granulocyte colony-stimulating factor receptor	Pegfilgrastim	*Approved*	*Approved*
Low affinity immunoglobulin gamma Fc region receptor III-B	Alefacept	*Rejected*	*Rejected*
Low affinity immunoglobulin	Cetuximab	*Approved*	*Rejected*

gamma Fc region receptor III-B			
Complement C4-A	Human immunoglobulin G	*Approved*	*Approved*
Atrial natriuretic peptide receptor 3	Nesiritide	*Approved*	*Approved*
Interleukin-6 receptor subunit alpha	Tocilizumab	*Approved*	*Approved*
Prothrombin	Lepirudin	*Approved*	*Approved*
Insulin-like growth factor-binding protein 7	Insulin human	*Approved*	*Approved*
Low affinity immunoglobulin gamma Fc region receptor II-a	Daclizumab	*Rejected*	*Rejected*
Low-density lipoprotein receptor-related protein 2	Urokinase	*Rejected*	*Rejected*
Interleukin-1 alpha	Rilonacept	*Approved*	*Approved*
High affinity immunoglobulin epsilon receptor subunit beta	Omalizumab	*Approved*	*Approved*
Interleukin-1 receptor type 1	Anakinra	*Approved*	*Approved*
Neprilysin	Liraglutide	*Approved*	*Rejected*
Interleukin-1 receptor antagonist protein	Rilonacept	*Approved*	*Approved*
Thyrotropin receptor	Thyrotropin alfa	*Approved*	*Approved*
Trypsin-1	Aprotinin	*Approved*	*Approved*

Plasma serine protease inhibitor	Urokinase	*Rejected*	*Rejected*
Lymphotoxin-alpha	Etanercept	*Approved*	*Approved*

Lampiran 2 Dokumentasi Validasi Pakar