KLASIFIKASI *IMBALANCED DATA* UNTUK *POST-TRANSLATIONAL MODIFICATION* (PTM) PADA *SEQUENCES* PROTEIN LISIN MENGGUNAKAN *ADAPTIVE SYNTHETIC* (ADASYN) *SAMPLING* DENGAN METODE *LIGHT GRADIENT BOOSTING MACHINE* (LightGBM)

(Skripsi)

Oleh

Ardella Dean Awalia

1917051024

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG
2023

ABSTRAK

KLASIFIKASI *IMBALANCED DATA* UNTUK *POST-TRANSLATIONAL MODIFICATION* (PTM) PADA *SEQUENCES* PROTEIN LISIN MENGGUNAKAN *ADAPTIVE SYNTHETIC* (ADASYN) *SAMPLING* DENGAN METODE *LIGHT GRADIENT BOOSTING MACHINE* (LightGBM)

Oleh

ARDELLA DEAN AWALIA

Crotonylation merupakan salah satu jenis modifikasi pascatranslasi (PTM) berupa penambahan gugus asil pada asam amino lisin. Modifikasi ini memiliki kemampuan dalam mengatur ekspresi gen dan ditemukan terlibat pada beberapa penyakit, seperti depresi, ginjal kronis, hingga kanker. Identifikasi situs PTM menjadi hal krusial mengingat perannya dalam siklus sel. Metode machine learning untuk klasifikasi situs PTM dapat digunakan sebagai alternatif dalam mengenali situs PTM, namun membutuhkan data yang relatif banyak agar dapat memberikan hasil yang andal. Penelitian ini dilakukan menggunakan metode klasifikasi LightGBM dan teknik *oversampling* ADASYN untuk menangani ketersediaan data protein *crotonylation* yang cukup sedikit. Data yang digunakan diperoleh dari situs UniProt terdiri dari 159 data positif dan 847 data negatif. Ekstraksi fitur menggunakan *binary encoding, position weight amino acid, encoding based on grouped weight, k-nearest neighbors,* dan *pseudo-position specific scoring matrix* menghasilkan 833 fitur. 5-*fold cross-validation* digunakan pada proses *training* untuk mencari kombinasi *hyperparameter* terbaik. Hasil penelitian menunjukkan bahwa pembagian data menggunakan 90% data sebagai data latih dan 10% data sebagai data uji memberikan hasil tertinggi dengan nilai *accuracy* sebesar 96,04%, *sensitivity* sebesar 87,50%, *specificity* sebesar 97,65%, MCC sebesar 85,15%, dan AUC sebesar 98,90%.

Kata kunci: modifikasi pascatranslasi, *crotonylation,* ADASYN, LightGBM

ABSTRACT

CLASSIFICATION OF IMBALANCED DATA FOR POST-TRANSLATIONAL MODIFICATION ON LYSINE PROTEIN SEQUENCES USING ADAPTIVE SYNTHETIC SAMPLING WITH LIGHT GRADIENT BOOSTING MACHINE

By

ARDELLA DEAN AWALIA

Crotonylation is a type of post-translational modification (PTM). It is an addition of acyl group to the lysine residues. This modification has the ability to regulate gene expression and has been found to be involved in several diseases, such as depression, chronic kidney disease, and cancer. Identification of PTM sites is crucial considering their role in the cell cycle. Machine learning methods for classifying PTM sites can be used as an alternative for recognizing PTM sites, but they require relatively large amounts of data to provide reliable results. This research was carried out using the LightGBM classification method and the ADASYN oversampling technique to handle the limited availability of crotonylated protein sequences. The data was obtained from the UniProt website consisting of 159 positive data (crotonylated) and 847 negative data (noncrotonylated). Feature extraction by using binary encoding, position weight amino acid, encoding based on grouped weight, k-nearest neighbors, and pseudo-position specific scoring matrix produced 833 features. 5-fold cross validation was used in the training process to find best hyperparameter combinations. The results showed that the highest result was obtained by using 90% of the data as training data with the application of ADASYN oversampling (n_neighbors=9) and 10% of the data as test data with 96,04% accuracy, 87,50% sensitivity, 97.65% specificity, 85,15% MCC, and 98,90% AUC.

Keywords: post-translational modification, *crotonylation*, ADASYN, LightGBM

KLASIFIKASI *IMBALANCED DATA* UNTUK *POST-TRANSLATIONAL MODIFICATION* (PTM) PADA *SEQUENCES* PROTEIN LISIN MENGGUNAKAN *ADAPTIVE SYNTHETIC* (ADASYN) *SAMPLING* DENGAN METODE *LIGHT GRADIENT BOOSTING MACHINE* (LightGBM)

Oleh

ARDELLA DEAN AWALIA

Skripsi

Sebagai Salah Satu Syarat Untuk Memperoleh Gelar
SARJANA KOMPUTER

Pada

Jurusan Ilmu Komputer
Fakultas Matematika dan Ilmu Pengetahuan Alam

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG
2023

Judul Skripsi	:	**KLASIFIKASI *IMBALANCED DATA* UNTUK *POST-TRANSLATIONAL MODIFICATION* (PTM) PADA *SEQUENCES* PROTEIN LISIN MENGGUNAKAN *ADAPTIVE SYNTHETIC* (ADASYN) *SAMPLING* DENGAN METODE *LIGHT GRADIENT BOOSTING MACHINE* (LightGBM)**
Nama Mahasiswa	:	Ardella Dean Awalia
Nomor Pokok Mahasiswa	:	1917051024
Program Studi	:	S1-Ilmu Komputer
Fakultas	:	Matematika dan Ilmu Pengetahuan Alam

MENYETUJUI

1. Komisi Pembimbing

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

2. Ketua Jurusan Ilmu Komputer

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

MENGESAHKAN

1. Tim Penguji

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

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

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

2. Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam

Dr. Eng. Heri Satria, S.Si., M.Si.
NIP 19711001 200501 1 002

Tanggal Lulus Ujian Skripsi: **23 November 2023**

PERNYATAAN

Saya yang bertanda tangan di bawah ini:

Nama : Ardella Dean Awalia

NPM : 1917051024

Dengan ini menyatakan bahwa skripsi saya yang berjudul "KLASIFIKASI *IMBALANCED DATA* UNTUK *POST-TRANSLATIONAL MODIFICATION* (PTM) PADA *SEQUENCES* PROTEIN LISIN MENGGUNAKAN *ADAPTIVE SYNTHETIC* (ADASYN) *SAMPLING* DENGAN METODE *LIGHT GRADIENT BOOSTING MACHINE* (LightGBM)" adalah benar hasil karya sendiri dan bukan karya orang lain. Seluruh tulisan yang tertulis 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, 8 Desember 2023

Ardella Dean Awalia

NPM 1917051024

RIWAYAT HIDUP

Penulis dilahirkan di Tangerang pada tanggal 18 Juli 2001 sebagai anak pertama dari dua bersaudara Penulis menyelesaikan pendidikan Sekolah Dasar (SD) di SDIT Al-Fatih 1 pada tahun 2013, pendidikan Sekolah Menengah Pertama (SMP) di SMP Negeri 1 Curug tahun 2016, dan pendidikan Sekolah Menengah Atas (SMA) di SMA Negeri 3 Kabupaten Tangerang pada tahun 2019.

Pada tahun 2019, penulis lulus Seleksi Nasional Masuk Perguruan Tinggi Negeri (SNMPTN) dan terdaftar sebagai mahasiswa Jurusan Imu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung. Selama menjadi mahasiswa, penulis turut serta berpartisipasi dalam berbagai kegiatan, antara lain sebagai berikut.

1. Anggota Bidang Kaderisasi Himpunan Mahasiswa Jurusan Ilmu Komputer (Himakom) pada tahun 2020 dan 2021.
2. Asisten Praktikum di Jurusan Ilmu Komputer pada tahun 2021 hingga 2023.
3. Peserta program Kredensial Mikro Mahasiswa Indonesia (KMMI) di Universitas Wijaya Kusuma Surabaya pada tahun 2021.
4. Peserta uji kompetensi *Junior Web Developer* yang diselenggarakan oleh Badan Nasional Sertifikasi Profesi (BNSP) pada tahun 2022.
5. Peserta Kuliah Kerja Nyata (KKN) Kebangsaan ke-X pada tahun 2022 di Desa Tamban Jaya, Kecamatan Tamban Catur, Kabupaten Kapuas, Kalimantan Tengah.

MOTTO

"Sebaik-baik manusia adalah yang paling bermanfaat bagi manusia (yang lain)"

(HR. Ahmad)

"Dan bahwasanya seorang manusia tiada memperoleh selain apa yang telah diusahakannya"

(QS. An-Najm: 39)

"Hatiku tenang mengetahui bahwa apa yang melewatkanku tidak akan pernah menjadi takdirku, dan apa yang ditakdirkan untukku tidak akan pernah melewatkanku"

(Umar bin Khattab)

"But I have promises to keep, and miles to go before I sleep, and miles to go before I sleep."

(Robert Frost, dalam *Stopping by Woods on a Snowy Evening)*

PERSEMBAHAN

Alhamdulillahi rabbil'aalamiin

Puji syukur ke hadirat Allah Subhanahu Wa Ta'ala atas segala rahmat, berkah, dan karunia-Nya sehingga saya dapat menyelesaikan skripsi ini dengan sebaik-baiknya. Shalawat serta salam selalu tercurah kepada Nabi Muhammad Shalallahu 'Alaihi Wasallam serta keluarga dan para sahabatnya.

Dengan ini, saya persembahkan karya ini kepada:

Kedua Orang Tua

Ibu dan Bapak yang selalu mendukung, mendoakan, dan memberikan yang terbaik. Terima kasih untuk semua kasih sayang, didikan, dukungan, doa, serta pengorbanan yang telah Ibu dan Bapak lakukan. Semoga tulisan ini dapat menjadi sedikit bentuk tanggung jawab dan rasa terima kasih saya atas semua yang telah kalian berikan.

SANWACANA

Puji syukur kehadirat Allah SWT atas rahmat, berkah, hidayah, dan karuniaNya, shalawat serta salam senantiasa tercurahkan kepada Nabi Muhammad SAW beserta keluarga dan para sahabatnya, sehingga penulis dapat menyelesaikan skripsi ini yang berjudul "Klasifikasi *Imbalanced Data* untuk *Post Translational Modification* (PTM) pada *Sequences* Protein Lisin Menggunakan *Adaptive Synthetic* (ADASYN) *Sampling* dengan *Light Gradient Boosting Machine* (LightGBM)". Semoga skripsi ini dapat menambah pengetahuan bagi pembaca tentang klasifikasi data sekuens protein menggunakan teknik *oversampling* dengan metode LightGBM.

Terima kasih penulis ucapkan kepada seluruh pihak yang turut berkontribusi dalam penyusunan skripsi ini hingga selesai, antara lain.

1. Kedua orang tua dan keluarga yang selalu mendoakan, menyayangi, dan mendukung secara moral maupun material.
2. Bapak Favorisen R. Lumbanraja, Ph.D., sebagai pembimbing utama yang selalu membimbing, memberikan arahan, dan saran sehingga skripsi ini dapat diselesaikan dengan baik.
3. Bapak Reza Faisal, S.T., M.T., Ph.D., sebagai pembahas pertama yang telah memberikan saran penelitian yang bermanfaat dalam memperbaiki kekurangan skripsi ini sehingga dapat diselesaikan dengan baik.
4. Bapak Dr. rer. nat. Akmal Junaidi, M.Sc., sebagai pembahas kedua yang telah memberikan saran penelitian yang bermanfaat dalam memperbaiki kekurangan skripsi ini sehingga dapat diselesaikan dengan baik.
5. Bapak Dwi Sakethi, M.Kom., selaku pembimbing akademik.
6. Bapak Dr. Eng. Heri Satria, S.Si., M.Si., selaku Dekan FMIPA Universitas Lampung.

7. Bapak Didik Kurniawan, S.Si., M.T., selaku Ketua Jurusan Ilmu Komputer Universitas Lampung.
8. Ibu Anie Rose Irawati selaku sekretaris Jurusan Ilmu Komputer yang telah membantu proses administrasi pelaksanaan seminar dan ujian skripsi.
9. Bapak dan Ibu Dosen Jurusan Ilmu Komputer yang telah memberikan ilmu yang bermanfaat selama perkuliahan dan selalu memotivasi untuk menjadi individu yang lebih baik.
10. Ibu Ade Nora Maela, Bang Zainuddin, dan Mas Nofal yang telah membantu urusan administrasi di Jurusan Ilmu Komputer.
11. Rekan seperbimbingan yang selalu saling membantu, menyemangati, menghibur, dan memotivasi.
12. Rekan-rekan Ilmu Komputer angkatan 2019 yang telah memberi pengalaman berarti selama menjalankan studi di Jurusan Ilmu Komputer Universitas Lampung.
13. Semua pihak yang terlibat secara langsung maupun tidak langsung selama masa perkuliahan berlangsung dan penyusunan skripsi yang tidak dapat disebutkan satu persatu.

Skripsi ini tentu masih banyak terdapat kekurangan yang disebabkan keterbatasan pengetahuan dan pengalaman penulis. Kritik dan saran sangat penulis harapkan sebagai pembelajaran. Semoga skripsi ini dapat bermanfaat bagi para pembaca.

Bandar Lampung, 8 Desember 2023
Penulis

Ardella Dean Awalia
NPM 1917051024

DAFTAR ISI

Halaman

DAFTAR TABEL

Tabel Halaman

DAFTAR GAMBAR

Gambar Halaman

DAFTAR KODE PROGRAM

Kode Program Halaman

I. PENDAHULUAN

1.1. Latar Belakang

Modifikasi pascatranslasi atau biasa disebut sebagai *post-translational modification* (PTM) adalah perubahan biokimia yang terjadi pada asam amino setelah melewati tahap translasi sintesis protein (Carter & Shieh, 2015). PTM memiliki peran yang signifikan dalam aktivitas, fungsi, dan mengatur struktur protein. Identifikasi PTM terkait menjadi dasar yang mendalam pada mekanisme biologis, pengobatan penyakit, dan desain obat. Metode untuk mengidentifikasi situs PTM dapat dilakukan dengan cara yang beragam, seperti *radioactive label method, chromatin immunoprecipitation (ChIP), mass spectrometry (MS), dan liquid chromatography* yang memerlukan biaya, waktu dan tenaga manusia cukup banyak (Dou *et al*., 2021).

Salah satu jenis PTM adalah *lysine crotonylation.* PTM jenis ini berperan dalam transkripsi gen, pemrosesan RNA, metabolisme asam nukleat, ekspresi gen, dan siklus sel (Wei *et al*., 2017). Beberapa studi juga menyebutkan bahwa *lysine crotonylation* berasosiasi dengan beberapa penyakit, seperti latensi HIV (Zhao *et al*., 2022), ginjal kronis dan akut, depresi, dan kanker. Identifikasi *lysine crotonylation* pada protein menjadi penting untuk melihat karakter dan mengatur mekanisme biologis primer (Tng *et al*., 2022). Namun, permasalahan umum yang sering terjadi adalah ketersediaan data yang tidak seimbang, dimana data yang akan diteliti jumlahnya sangat sedikit.

Imbalanced data dapat diartikan sebagai ketidakseimbangan jumlah sampel data yang merepresentasikan sebuah kelas, sehingga jauh lebih rendah dibandingkan dengan kelas lainnya pada suatu *dataset* (Asniar *et al*., 2022). Masalah ini sering terjadi pada hal yang jarang ditemui, namun memiliki hubungan yang erat di dunia

nyata, misalnya deteksi penipuan (Branco *et al*., 2016), diagnosa kesehatan, *return* yang tidak biasa di pasar saham, dan lain sebagainya (H. Chen *et al*., 2019). Kelas yang jumlah datanya lebih rendah disebut kelas minoritas, sedangkan kelas yang representasi datanya lebih banyak disebut kelas mayoritas. Klasifikasi menggunakan data yang tidak seimbang dapat menyebabkan *bias*, yaitu kecenderungan model klasifikasi dalam menjatuhkan pilihan kepada data mayoritas saat menentukan kelas dari suatu sampel.

Beberapa pendekatan sebagai solusi yang tersedia saat ini dalam mengatasi *imbalanced data* dapat dibagi menjadi 3, yaitu berdasarkan data (*data-level methods),* algoritma *(algorithm methods)*, dan campuran (*hybrid-level methods)*. Pendekatan berdasarkan data dapat dilakukan dengan mengolah ulang sampel agar didapatkan jumlah data yang seimbang dengan cara *undersampling, oversampling, dan hybridsampling*. Pendekatan berdasarkan algoritma yaitu pengklasifikasi secara langsung disesuaikan agar dapat mengurangi *bias* prediksi, sedangkan pendekatan *hybrid* adalah gabungan antara pendekatan berdasarkan data dan pendekatan berdasarkan algoritma (Dou *et al*., 2021).

Adaptive Syntethic Sampling (ADASYN) adalah salah satu teknik *oversampling* untuk menangani ketidakseimbangan data. ADASYN didasarkan pada ide untuk membuat data sintetis secara adaptif berdasarkan kepadatan distribusinya. ADASYN dapat secara otomatis memustuskan jumlah data sintetis yang dibuat untuk setiap data minoritas dengan mengubah *weight* secara adaptif dari setiap data minoritas yang berbeda. ADASYN tidak hanya dapat mengurangi kecenderungan pembelajaran, tetapi juga dapat menggeser batas keputusan secara adaptif untuk fokus pada sampel-sampel yang sulit dipelajari (He *et al*., 2008).

Penelitian Liu *et al*. (2020) mengidentifikasi situs *lysine crotonylation* menggunakan teknik *oversampling* SMOTE dan *classifier* LightGBM. Data protein Lisin yang digunakan terdiri dari 159 data positif dan 847 data negatif. Sekuens protein diekstrak menggunakan gabungan beberapa metode ekstraksi fitur. Metode tersebut adalah *binary encoding, position-weight amino acid, encoding based grouped weight, k-nearest neighbor,* dan *pseudo-position specific scoring matrix.*

Pengukuran hasil klasifikasi dilakukan dengan beberapa metrik evaluasi, yaitu akurasi sebesar 98,99%, MCC sebesar 0,9798, dan AUC sebesar 0,9996.

Karena biaya yang tinggi dan banyaknya waktu yang dibutuhkan untuk melakukan teknik *high-throughput sequencing*, klasifikasi situs PTM membutuhkan metode alternatif dengan biaya, waktu, dan tenaga yang lebih sedikit tetapi tetap dapat menghasilkan akurasi yang tepat untuk menghasilkan rekomendasi yang dapat diandalkan. Pemanfaatan *machine learning* dalam hal ini dilakukan sebagai pendekatan yang efektif untuk mengenali secara cepat situs PTM yang memiliki potensi termodifikasi. Penerapan metode lain dalam menangani ketidakseimbangan data, seperti ADASYN, diperlukan untuk melihat pengaruh yang dihasilkan dari penerapan metode tersebut. Dengan demikian, diperoleh metode yang dapat membantu meningkatkan performa *classifier* dan memberi hasil yang dapat diandalkan. Penelitian yang dilakukan oleh Liu *et al*., (2020) menjadi acuan dalam penelitian ini. Metode yang diusulkan dalam penelitian ini adalah teknik *oversampling* ADASYN dan *classifier Light Gradient Boosting Machine* (LightGBM) pada *crotonylation* protein lisin menggunakan data yang diperoleh dari penelitian Liu *et al*., (2020). ADASYN digunakan untuk melihat perbandingan hasil klasifikasi yang diperoleh dalam menangani *imbalanced data.*

1.2. Rumusan Masalah

Berdasarkan latar belakang tersebut, rumusan masalah dalam penelitian ini adalah seperti berikut.

1. Apakah penerapan *oversampling* ADASYN pada ketidakseimbangan data *lysine crotonylation* dapat memengaruhi hasil klasifikasi menggunakan metode klasifikasi LightGBM?
2. Apakah kinerja metode klasifikasi LightGBM dengan penerapan *oversampling* ADASYN dalam menangani ketidakseimbangan data *lysine crotonylation* lebih baik dari penelitian sebelumnya?

1.3. Batasan Masalah

Batasan dalam penelitian ini adalah sebagai berikut.

1. Penelitian ini menggunakan *dataset* yang diperoleh dari penelitian Liu *et al*. (2020) berupa protein Lisin sebanyak 1006 data.
2. Penelitian ini dilakukan dengan menggunakan teknik *oversampling* ADASYN dan metode klasifikasi LightGBM dengan ekstraksi fitur BE, PWAA, EBGW, KNN, dan PsePSSM.

1.4. Tujuan Penelitian

Tujuan penelitian ini adalah sebagai berikut.

1. Mengukur kinerja metode klasifikasi LightGBM dengan atau tanpa penerapan *oversampling* ADASYN dalam menangani ketidakseimbangan data *lysine crotonylation.*
2. Membandingkan kinerja model yang diperoleh dalam penelitian ini dengan penelitian Liu *et al*. (2020).

1.5. Manfaat Penelitian

Manfaat dari penelitian ini adalah sebagai berikut.

1. Menambah pengetahuan tentang ketidakseimbangan data pada *lysine crotonylation.*
2. Mengetahui kinerja metode klasifikasi LightGBM dengan atau tanpa penerapan *oversampling* ADASYN dalam menangani ketidakseimbangan data pada *lysine crotonylation.*

II. TINJAUAN PUSTAKA

2.1. Penelitian Terdahulu

Penelitian terdahulu digunakan sebagai acuan dalam melakukan perbandingan hasil klasifikasi. Penelitian pembanding memiliki topik mengenai *imbalanced data* dan prediksi protein lisin pada situs *crotonylation.* Tabel 1 adalah gambaran umum penelitian terdahulu sebagai acuan dalam penelitian ini.

Tabel 1. Penelitian Terdahulu Terkait *Lysine Crotonylation*

No	Penelitian	Data	Metode	Hasil
1	*Prediction of protein crotonylation sites through LightGBM classifier based on SMOTE and elastic net* (Liu *et al*., 2020)	**Protein Lisin** Data Latih Jumlah: 1.006 Negatif: 847 Positif: 159 Sumber: UniProt *Database*	Metode klasifikasi: LightGBM Seleksi Fitur: *Elastic Net* Algoritma SMOTE	Acc: 98,99% MCC: 0,9798 AUC: 0,9996
2	*Identification of Protein Lysine Crotonylation Sites by a Deep Learning Framework with Convolutional Neural Networks* (Zhao *et al*., 2020)	**Protein Lisin** Data Latih Jumlah: 32.418 Negatif: 29.676 Positif: 2.742 Data Independen Jumlah: 8.169 Negatif: 7.458 Positif: 711	Metode klasifikasi: *Convolutional Neural Network* (CNN)	Acc: 86,78% MCC: 0,3339 AUC: 0,8555 Acc: 85,64% MCC: 0,3335 AUC: 0,8553
3	*iKcr-PseEns: Identify lysine crotonylation sites in histone proteins with pseudo components and ensemble classifier* (Qiu *et al*., 2018)	**Protein Lisin** Data Latih Jumlah: 1.035 Negatif: 866 Positif: 169 Sumber: UniProt *Database*	Metode klasifikasi: *Ensemble Random Forest*	Acc: 94,49% MCC: 0,8260 AUC: 0,9753

2.1.1. *Prediction of protein crotonylation sites through LightGBM classifier based on SMOTE and elastic net* (2020)

Liu *et al*. (2020) menggunakan 101 protein yang diperoleh dari *Universal Resource of Protein* (UniProt) *database* pada studi ini. *Dataset* terdiri dari protein lisin dengan data positif sebanyak 159 dan data negatif sebanyak 847. LightGBM digunakan sebagai *classifier* penentu dengan bantuan algoritma SMOTE untuk *oversampling.* Penelitian ini juga menggunakan algoritma *Elastic Net* untuk melakukan *feature selection.*

Metode seleksi fitur menggunakan *Elastic Net* memperoleh nilai performa tertinggi daripada lima lainnya dengan AUC 98,15%. Metode seleksi fitur lainnya adalah *lasso, extra-trees, singular value decomposition* (SVD), *local linear embedding* (LLE), dan *multiple dimensional scaling* (MDS). Sedangkan, metode klasifikasi LightGBM memperoleh performa yang paling tinggi diantara 6 lainnya yaitu Naïve Bayes, AdaBoost, KNN, XGBoost, SVM, dan Random Forest. Hasil yang diperoleh LightGBM adalah 98,99% untuk akurasi, 0,9798 untuk MCC, dan 0,9996 untuk AUC.

2.1.2. *Identification of Protein Lysine Crotonylation Sites by a Deep Learning Framework with Convolutional Neural Networks* (2020)

Penelitian ini dilakukan oleh Zhao *et al*. (2020) menggunakan protein lisin sebanyak 1.188 protein untuk data latih dan 295 protein untuk data uji. Data latih terdiri dari data positif sebanyak 2.742 dan data negatif sebanyak 29.676. Sedangkan data uji terdiri dari data positif sebanyak 711 dan data negatif sebanyak 7.458. *Dataset* dapat diunduh di http://www.bioinfogo.org/pkcr/download.php.

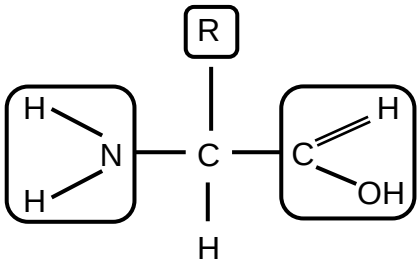
Metode *deep learning* yaitu *Convolutional Neural Network* (CNN) digunakan untuk melakukan klasifikasi protein dalam penelitian ini. Pada data latih, CNN memiliki hasil akurasi sebesar 86,78%, MCC sebesar 0,3339, dan AUC sebesar 0,8555. Sedangkan pada data uji, CNN menghasilkan 85,64%, 0,3335, dan 0,8553 untuk akurasi, MCC, dan AUC.

2.1.3. *iKcr-PseEns: Identify lysine crotonylation sites in histone proteins with pseudo components and ensemble classifier* (2018)

Studi ini menggunakan *Random Forest* untuk mengidentifikasi situs *crotonylation* pada protein lisin (Kcr). Qiu *et al*. (2018) melibatkan data protein lisin sebanyak 1.035 yang diperoleh dari *database* UniProt untuk melatih model. Data tersebut terdiri dari 169 data positif dan 866 data negatif. Metode *Ensemble Random Forest* menghasilkan akurasi sebesar 94,49%, MCC sebesar 0,8260, dan AUC sebesar 0,9753.

2.2. Protein

Protein tersusun dari rangkaian asam amino yang membentuk ikatan peptida sehingga disebut sebagai makromolekul. Sedangkan asam amino merupakan senyawa organik yang terdiri dari gugus amino (NH_2), gugus karboksil (COOH), dan gugus rantai samping (biasa dilambangkan dengan huruf R) (Suprayitno & Sulistiyati, 2017). Rantai samping inilah yang membedakan antara satu asam amino dengan asam amino lainnya. Gambar 1 memperlihatkan susunan gugus fungsi pada asam amino.



Gambar 1. Struktur Asam Amino (Urry et al., 2016).

Protein dibentuk di ribosom melalui proses yang disebut sintesis protein. Sintesis protein dilakukan dalam dua tahap, yaitu transkripsi dan translasi. Pada transkripsi, informasi pada DNA ditransfer ke *messenger* RNA (mRNA) dengan bantuan enzim RNA polimerase. Kemudian, pada tahap translasi, informasi pada mRNA digunakan untuk menggabungkan asam amino menjadi rantai polipeptida (Urry *et al*., 2016). Asam amino yang dibutuhkan untuk sintesis protein pada tubuh manusia dapat dilihat pada Tabel 2. Perbedaan urutan, jenis asam amino, dan interaksi

kompleks yang dapat dilakukan oleh rantai ini menentukan keunikan protein yang terbentuk (Lopez & Mohiuddin, 2022). Setelah proses translasi selesai, rantai polipeptida yang terbentuk dapat dimodifikasi melalui proses *post-translational modification* (PTM) (Wu, 2009).

Tabel 2. Asam Amino (Suprayitno & Sulistiyati, 2017)

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

2.3. *Post-Translational Modification*

Modifikasi pascatranslasi protein merupakan modifikasi biokimia yang terjadi pada satu atau lebih asam amino pada protein setelah melewati tahap translasi di ribosom (Carter & Shieh, 2015). PTM mungkin dibutuhkan sebelum protein dapat memulai tugasnya di dalam sel. Modifikasi asam amino dapat berupa penambahan gugus gula, lipid, fosfat, dan lain sebagainya (Urry *et al*., 2016). PTM memiliki peran yang sangat berarti dalam berbagai proses seluler yang mengatur pengaruh fisik, sifat kimia, pelipatan, stabilitas, dan aktivitas protein sehingga dapat mengubah fungsi protein itu sendiri (W. Chen *et al*., 2018). Modifikasi pada protein lisin, sering disebut juga sebagai K-PTM merupakan jenis PTM yang paling banyak diamati. Beberapa contoh jenis PTM tersebut adalah *acetylation*, *biotinylation*, *butyrylation*, *crotonylation*, *methylation*, *propionylation*, *succinylation*, *ubiquitination*, dan modifikasi lain seperti ubiquitin (Qiu *et al*., 2016).

2.4. *Crotonylation*

Crotonylation utamanya terjadi pada gugus asam amino Lisin pada histon dan baru-baru ini juga dilaporkan terjadi pada residu Serin (S. Wang *et al*., 2021). *Lysine*

crotonylation (Kcr) adalah jenis PTM yang tergolong baru teridentifikasi, terjadi pada protein histon dan non-histon di berbagai organisme. Jenis PTM ini ditemukan beberapa tahun lalu oleh Tan *et al*. (2011) menggunakan pendekatan proteom terintegrasi berbasis *mass-spectrometry*. PTM jenis baru ini dikelompokan sebagai *reversible acylation modification* yang diregulasi oleh berbagai *acylase, deacylase,* dan konstentrasi substrat *crotonyl-CoA* intraseluler (Zhao *et al*., 2022).

Kcr telah muncul sebagai jenis PTM yang penting dalam regulasi transkripsi gen melalui mekanisme epigenetik (Bos & Muir, 2018). Analisis bioinformatika mengungkapkan bahwa protein *crotonylation* terlibat dalam pemrosesan RNA, metabolisme asam nukleat, organisasi kromosom, ekspresi gen, dan juga dapat menghambat replikasi DNA sehingga memengaruhi regulasi siklus sel (Wei *et al*., 2017). Studi yang dilakukan oleh Tan *et al*. (2011) menyimpulkan bahwa Kcr histon adalah indikator kuat promotor aktif dan dapat menjadi sinyal penting dalam kontrol diferensiasi sel benih jantan. Kcr histon juga ditemukan berasosiasi dengan banyak penyakit seperti cedera ginjal akut, depresi, latensi HIV, dan proses kanker (Zhao *et al*., 2022).

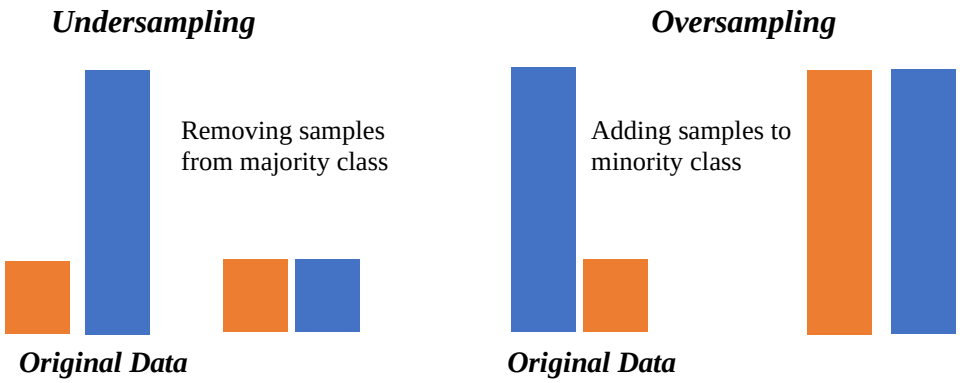
2.5. *Imbalanced Data*

Imbalanced data, secara teknis dapat berarti keadaan dimana sebuah *dataset* menunjukkan distribusi yang jomplang antar kelas satu dengan kelas lainnya (He & Garcia, 2009). Pada kasus yang memiliki dua kelas *(binary)*, kelas mayoritas adalah keadaan dimana kelas tersebut memiliki persentase sampel yang lebih besar dibanding kelas lainnya. Sebagai contoh, orang yang menderita HIV lebih sedikit dibanding orang yang sehat, sehingga kelas "penderita HIV" adalah kelas minoritas dan kelas "sehat" adalah kelas mayoritas. Dalam hal ini, *classifier* biasanya cenderung memprediksi sampel uji ke dalam kelas mayoritas dan mengabaikan keberadaan kelas minoritas (Yen & Yue-Shi Lee, 2009). Kecenderungan ini dapat menghasilkan ketidakseimbangan derajat akurasi yang parah, kelas mayoritas dapat memiliki akurasi mendekati seratus persen, tetapi tidak dengan akurasi kelas minoritas (He & Garcia, 2009).

Solusi yang digunakan untuk mengatasi klasifikasi kelas tidak seimbang ada tiga cara menurut Dou *et al*. (2021), yaitu:

1. Pendekatan berdasarkan data, bertujuan menyeimbangkan jumlah sampel kedua kelas. Pendekatan ini dibagi menjadi tiga cara:
   a. *Undersampling,* yaitu mengurangi jumlah data kelas mayoritas sehingga jumlahnya setara dengan kelas minoritas. Contoh algoritmanya adalah NearMiss, RUS, Tomek-Links, dan ENN.
   b. *Oversampling,* yaitu menambah jumlah data kelas minoritas sehingga jumlahnya setara dengan kelas mayoritas. Contoh algoritmanya adalah ROS, SMOTE, ADASYN, dan RWO.
   c. *Hybridsampling,* yaitu gabungan teknik *undersampling* dan *oversampling.* Contoh algoritmanya adalah SMOTE-Tomek dan SMOTE-ENN.

   Gambar 2 menunjukkan perbandingan antara *undersampling* dan *oversampling.*



Gambar 2. *Undersampling* dan *Oversampling* (Mohammed et al., 2020).

2. Pendekatan berdasarkan algoritma dilakukan dengan memodifikasi algoritma yang sudah ada atau membuat algoritma baru. Cara untuk memodifikasi algoritma dibagi menjadi tiga:
   a. *Cost-Sensitive,* yaitu mendefinisikan *cost-matrix* untuk merevisi sampel yang salah diklasifikasi.
   b. *Ensemble,* yaitu melatih banyak algoritma yang sudah ada dan menggabungkannya untuk menghasilkan hasil akhir. Contoh metode ini adalah *bagging, boosting,* dan gabungan keduanya.

c. *One-class,* membuat model untuk sampel minoritas dengan kasus yang sangat tidak seimbang.

3. Pendekatan gabungan *(hybrid-level)* yaitu menggabungkan pendekatan berdasarkan data dan algoritma. Contoh algoritma yang termasuk jenis ini adalah RUSBoost dan SMOTEBoost.

2.6. *Adaptive Synthetic Sampling*

Metode *Adaptive Synthetic* adalah salah satu metode yang digunakan untuk menangani ketidakseimbangan data. Metode ini diusulkan oleh He *et al*. (2008) berdasarkan ide untuk membuat data sintetis secara adaptif menurut penyebarannya. Tujuan utama dari algoritma ini adalah mengurangi bias dan pembelajaran yang dapat beradaptasi. ADASYN menggunakan distribusi kepadatan (*density*) sebagai kriteria untuk memutuskan banyak data sintetis yang akan dibuat untuk setiap data minoritas (Zhu *et al*., 2020). Algoritma ADASYN dapat dilihat pada Kode Program 1 dan contoh data tidak seimbang seperti pada Tabel 3.

Tabel 3. Contoh Data Tidak Seimbang

Data	Fitur #1	Fitur #2	Fitur #3	Fitur #4	Fitur #5	Label
x_1	1	0	1	2	0	1
x_2	2	1	0	1	1	1
x_3	3	2	2	2	0	0
x_4	3	1	0	0	2	0
x_5	2	3	1	0	1	0
x_6	1	4	3	1	3	0
x_7	5	2	2	4	3	0
x_8	5	1	4	3	2	0
x_9	2	2	5	3	4	0
x_{10}	2	0	3	2	5	0

Langkah pertama, ADASYN menghitung derajat ketidakseimbangan *d* antara kelas minoritas (label 1) dan kelas mayoritas (label 0) dengan nilai antara 0 sampai 1. Derajat ketidakseimbangan *d* merupakan perbandingan antara banyak data kelas minoritas (m_s) dan banyak data kelas mayoritas (m_l). Langkah kedua, jika *d* kurang dari batas maksimal toleransi (d_{th}), algoritma akan membuat data buatan untuk mengimbangi data mayoritas. Pertama, algoritma akan menghitung jumlah data

sintetis yang akan dibuat yaitu $G = (m_l - m_s) * \beta$. Kedua, setiap sampel minoritas akan dicari *k* tetangga terdekatnya berdasarkan *Euclidean distance*, lalu dihitung perbandingan antara banyak sampel mayoritas yang merupakan anggota tetangga terdekat dan banyak tetangga terdekat. Ketiga, rasio dari tiap sampel dinormalisasi. Keempat, hitung jumlah data sintetis yang harus dibuat untuk tiap-tiap sampel pada data minoritas. Kelima, data baru dibuat untuk setiap sampel data minoritas yang ada.

Input: Training data set D_{tr} with m samples $\{x_i, y_i\}$, i = 1, ..., m, where x_i is an instance in the n dimensional feature space X and $y_i \in Y = \{1, -1\}$ is the class identity label associated with xi. Define m_s and m_l as the number of minority class examples and the number of majority class examples, respectively. Therefore, $m_s \leq m_l$ and $m_s + m_l = m$.

Procedure

(1) Calculate the degree of class imbalance: $d = m_s/m_l$ where $d \in (0, 1]$.

(2) **If** $d < d_{th}$ then (d_{th} is a preset threshold for the maximum tolerated degree of class imbalance ratio):

   a) Calculate the number of synthetic data examples that need to be generated for the minority class: $G = (m_l - m_s) \times \beta$. Where $\beta \in [0, 1]$ is a parameter used to specify the desired balance level after generation of the synthetic data. $\beta = 1$ means a fully balanced data set is created after the generalization process.

   b) For each example $x_i \in$ *minorityclass*, find k nearest neighbors based on the Euclidean distance in n dimensional space, and calculate the ratio r_i defined as: $r_i = \Delta_i/K$, i = 1, ..., ms. Where Δ_i is the number of examples in the K nearest neighbors of x_i that belong to the majority class, therefore $r_i \in [0, 1]$;

   c) Normalize r_i according to $\hat{r}_i = r_i / \sum_{i=1}^{m_s} r_i$, so that $\hat{r}_i$ is a density distribution $\sum_i \hat{r}_i = 1$.

   d) Calculate the number of synthetic data examples that need to be generated for each minority example x_i: $g_i = \hat{r}_i \times G$ where G is the total number of synthetic data examples that need to be generated for the minority class.

   e) For each minority class data example x_i, generate g_i synthetic data examples according to the following steps:

      Do the Loop from 1 to g_i:

      i) Randomly choose one minority data example, x_{zi}, from the K nearest neighbors for data x_i.

      ii) Generate the synthetic data example:
      $s_i = x_i + (x_{zi} - x_i) \times \lambda$, where $(x_{zi} - x_i)$ is the difference vector in n dimensional spaces, and λ is a random number: $\lambda \in [0, 1]$.

      End Loop

Kode Program 1. Algoritme ADASYN (He et al., 2008).

Berikut contoh kasus penerapan algoritme ADASYN pada data yang memiliki dua fitur dan dua kelas, data ditampilkan pada Tabel 4.

Tabel 4. Contoh Data Tidak Seimbang Dua Fitur

Sampel	Fitur 1	Fitur 2	Kelas
x_1	1	0	1
x_2	3	3	0
x_3	3	1	0
x_4	2	3	0
x_5	1	4	0
x_6	2	1	1
x_7	5	2	0
x_8	5	1	0
x_9	2	2	0
x_{10}	2	-1	0

Langkah 1. Hitung derajat ketidakseimbangan *d* dengan batas maksimal toleransi d_{th} = 0,8 (nilai 0,8 menandakan bahwa data minoritas adalah sebanyak 80% dari data mayoritas).

$$d = \frac{m_s}{m_l} = \frac{2}{8} = 0{,}25$$

Langkah 2. Jika $d < d_{th}$, lanjut ke Langkah 2. Karena 0,25 < 0,8, maka lanjut ke proses berikutnya untuk membuat data baru.

Langkah 2a. Hitung banyak data baru yang akan dibuat sehingga jumlah data seimbang antara minoritas dan mayoritas. β adalah parameter yang menunjukkan tingkat keseimbangan yang diinginkan, dalam hal ini β = 1.

$$\begin{aligned} G &= (m_l - m_s) \times \beta \\ &= (8 - 2) \times 1 \\ G &= 6 \end{aligned}$$

Langkah 2b. Cari *k* tetangga terdekat (*k=3*) untuk setiap data minoritas menggunakan jarak *Euclidean,* $dis\,(x_1, x_2) = \sqrt{\sum_{i=1}^{n}(x_{1i} - x_{2i})^2}$.

Cari tetangga terdekat untuk x_1,

$dis\,(x_{1,}\, x_2) = \sqrt{(1-3)^2 + (0-3)^2} = \sqrt{13}$

$dis\,(x_{1,}\, x_3) = \sqrt{(1-3)^2 + (0-1)^2} = \sqrt{5}$

$dis\,(x_{1,}\, x_4) = \sqrt{(1-2)^2 + (0-3)^2} = \sqrt{10}$

$dis\,(x_{1,}\, x_5) = \sqrt{(1-1)^2 + (0-4)^2} = \sqrt{16}$

$dis\,(x_{1,}\, x_6) = \sqrt{(1-2)^2 + (0-1)^2} = \sqrt{2}$

$dis\,(x_{1,}\, x_7) = \sqrt{(1-5)^2 + (0-2)^2} = \sqrt{20}$

$dis\,(x_{1,}\, x_8) = \sqrt{(1-5)^2 + (0-1)^2} = \sqrt{17}$

$dis\,(x_{1,}\, x_9) = \sqrt{(1-2)^2 + (0-2)^2} = \sqrt{5}$

$dis\,(x_{1,}\, x_{10}) = \sqrt{(1-2)^2 + (0-(-1))^2} = \sqrt{2}$

Fungsi NearestNeighbors() pada *library* sklearn secara *default* memilih sampel yang ditemui lebih dulu pada *training set* jika diharuskan memilih satu antara dua sampel dengan jarak yang sama. Pada contoh di atas, jarak ($x_{1,}\, x_3$) dan ($x_{1,}\, x_9$) memiliki jarak yang sama. Karena sampel x_3 muncul lebih dulu daripada x_9, maka tiga tetangga terdekat x_1 adalah x_{10}, x_6, dan x_3. Dari tetangga terdekat tersebut, dua sampel termasuk anggota mayoritas. Perbandingan banyak anggota mayoritas pada lingkungan tetangga x_1 adalah $r_1 = \frac{\Delta_1}{k} = \frac{2}{3}$, dimana Δ_1 merupakan banyak sampel anggota mayoritas pada lingkungan tetangga x_1.

Cari tetangga terdekat untuk $x_{6,}$

$dis\,(x_{6,}\, x_1) = \sqrt{(2-1)^2 + (1-0)^2} = \sqrt{2}$

$dis\,(x_{6,}\, x_2) = \sqrt{(2-3)^2 + (1-3)^2} = \sqrt{5}$

$dis\,(x_{6,}\, x_3) = \sqrt{(2-3)^2 + (1-1)^2} = \sqrt{1}$

$dis\,(x_{6,}\, x_4) = \sqrt{(2-2)^2 + (1-3)^2} = \sqrt{4}$

$dis\,(x_{6,}\, x_5) = \sqrt{(2-1)^2 + (1-4)^2} = \sqrt{10}$

$dis\,(x_{6,}\, x_7) = \sqrt{(2-5)^2 + (1-2)^2} = \sqrt{10}$

$dis\,(x_{6,}\, x_8) = \sqrt{(2-5)^2 + (1-1)^2} = \sqrt{9}$

$dis\,(x_{6,}\, x_9) = \sqrt{(2-2)^2 + (1-2)^2} = \sqrt{1}$

$dis\,(x_{6,}\, x_{10}) = \sqrt{(2-2)^2 + (1-(-1))^2} = \sqrt{4}$

Tiga tetangga terdekat x_6 adalah x_9, x_3, dan x_1, dua sampel termasuk anggota mayoritas. Maka, perbandingan banyak anggota mayoritas pada lingkungan tetangga x_6 adalah $r_2 = \frac{\Delta_2}{k} = \frac{2}{3}$, dimana Δ_2 merupakan banyak sampel anggota mayoritas pada lingkungan tetangga x_6.

Langkah 2c. Normalisasi nilai *r* setiap data minoritas, $\widehat{r_i} = \frac{r_i}{\sum_{i=1}^{m_s} r_i}$.

$$\widehat{r_1} = \frac{r_1}{r_1 + r_2} = \frac{\frac{2}{3}}{\frac{2}{3} + \frac{2}{3}} = \frac{1}{2} \qquad \widehat{r_2} = \frac{r_2}{r_1 + r_2} = \frac{\frac{2}{3}}{\frac{2}{3} + \frac{2}{3}} = \frac{1}{2}$$

Langkah 2d. Hitung banyak data yang akan dibuat untuk tiap sampel data minoritas.

$g_i = \widehat{r_i} \times G, \qquad i = 1, \dots, m_s$

$g_1 = \widehat{r_1} \times 6 = 3 \rightarrow$ akan dibuat 3 data baru di sekitar x_1

$g_2 = \widehat{r_2} \times 6 = 3 \rightarrow$ akan dibuat 3 data baru di sekitar x_6

Langkah 2e. Buat data baru di sekitar tiap sampel data minoritas, $s_{ij} = x_i + (x_{zi} - x_i) \times \lambda$, dimana j=1, …, g_i. X_{zi} adalah sampel minoritas lain yang ada dalam lingkungan tetangga x_i, dan λ adalah nilai acak antara 0 sampai 1.

Buat data baru di sekitar x_1,

$s_{11} = [1 \quad 0] + ([2 \quad 1] - [1 \quad 0]) \times 0{,}2$

$s_{11} = [1 \quad 0] + ([1 \quad 1]) \times 0{,}2$

$s_{11} = [1 \quad 0] + [0{,}2 \quad 0{,}2]$

$s_{11} = [1{,}2 \quad 0{,}2]$

$s_{12} = [1 \quad 0] + ([2 \quad 1] - [1 \quad 0]) \times 0{,}7$

$s_{12} = [1 \quad 0] + ([1 \quad 1]) \times 0{,}7$

$s_{12} = [1 \quad 0] + [0{,}7 \quad 0{,}7]$

$s_{12} = [1{,}7 \quad 0{,}7]$

$s_{13} = [1 \quad 0] + ([2 \quad 1] - [1 \quad 0]) \times 1$

$s_{13} = [1 \quad 0] + ([1 \quad 1]) \times 1$

$s_{13} = [1 \quad 0] + [1 \quad 1]$

$s_{13} = [2 \quad 1]$

Buat data baru di sekitar x_6,

$s_{21} = [2 \quad 1] + ([1 \quad 0] - [2 \quad 1]) \times 0{,}4$

$s_{21} = [2 \quad 1] + ([-1 \quad -1]) \times 0{,}4$

$s_{21} = [2 \quad 1] + [-0{,}4 \quad -0{,}4]$

$s_{21} = [1{,}6 \quad 0{,}6]$

$s_{21} = [2 \quad 1] + ([1 \quad 0] - [2 \quad 1]) \times 0{,}6$

$s_{21} = [2 \quad 1] + ([-1 \quad -1]) \times 0{,}6$

$s_{21} = [2 \quad 1] + [-0{,}6 \quad -0{,}6]$

$s_{21} = [1{,}4 \quad 0{,}4]$

$s_{21} = [2 \quad 1] + ([1 \quad 0] - [2 \quad 1]) \times 0{,}9$

$s_{21} = [2 \quad 1] + ([-1 \quad -1]) \times 0{,}9$

$s_{21} = [2 \quad 1] + [-0{,}9 \quad -0,9]$

$s_{21} = [1{,}1 \quad 0{,}1]$

Jadi, setelah data baru diperoleh, dataset baru akan menjadi seperti pada Tabel 5.

Tabel 5. Contoh Data Hasil *Oversampling* ADASYN

Sampel	Fitur 1	Fitur 2	Kelas
x_1	1	0	1
x_2	3	3	0
x_3	3	1	0
x_4	2	3	0
x_5	1	4	0
x_6	2	1	1
x_7	5	2	0
x_8	5	1	0
x_9	2	2	0
x_{10}	2	-1	0
x_{11}	1,2	0,2	1
x_{12}	1,7	0,7	1
x_{13}	2	1	1
x_{14}	1,6	0,6	1
x_{15}	1,4	0,4	1
x_{16}	1,1	0,1	1

2.7. Light Gradient Boosting Machine (LightGBM)

LightGBM adalah metode yang dikembangkan oleh Ke *et al*. (2017) berdasarkan *Gradient Boosting Decision Tree* (GBDT). Metode ini menambahkan teknologi berbasis *gradient-based one-side sampling* (GOSS) dan *exclusive feature bundling* (EFB). LightGBM dapat menyelesaikan kekurangan yang dimiliki GBDT tradisional yaitu *time-consuming* dan kurang efisien ketika menangani data dengan dimensi yang besar (Dou *et al*., 2021). Model *ensemble* dari *decision tree* ini dapat digunakan untuk prediksi klasifikasi dan regresi. LightGBM menunjukkan kinerja yang unggul dalam presisi prediksi, stabilitas model, dan efisiensi komputasi melalui serangkaian tes (Yan *et al*., 2021).

Ide utama dari GOSS adalah memberi fokus lebih kepada data yang tidak terlatih cukup baik *(under-trained).* Gradien digunakan sebagai tolok ukur karena dapat menunjukan informasi penting untuk pengambilan sampel data. Sampel dengan gradien kecil berarti memiliki eror yang kecil, sehingga dapat dikatakan terlatih dengan baik. Sebaliknya, sampel dengan gradien besar memiliki eror yang besar dan dapat dikatakan tidak terlatih dengan cukup baik. Karena itu, GOSS menggunakan semua sampel yang memiliki gradien besar dan memilih acak pada sampel dengan gradien kecil untuk dijadikan subset baru.

Sedangkan, EFB memanfaatkan keadaan yang disebut *sparse data* (*features*), biasanya terdapat pada data berdimensi tinggi. *Sparse data* adalah ketika fitur-fitur pada suatu data banyak memiliki nilai nol *(zero values)* (Shukla, 2022). Jumlah fitur dan jumlah sampel yang digunakan pada dataset akan sebanding dengan kompleksitas komputasi sehingga mengarah pada lamanya waktu pelatihan. Tujuan EFB adalah mereduksi jumlah fitur dengan melakukan penggabungan fitur yang saling eksklusif ke dalam fitur tunggal. Penggabungan beberapa fitur ekslusif ini menjadikan dataset memiliki *dense features* (fitur yang memiliki kebanyakan nilai selain nol) yang lebih sedikit, dengan begitu dapat menghindari komputasi yang tidak perlu untuk nilai fitur nol. EFB digambarkan pada Gambar 3. Berikut langkah-langkah untuk membuat *feature bundle.*

Langkah 1. Tetapkan fitur *reference* dan fitur yang akan di-*bundle.*

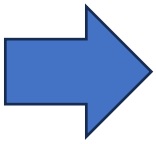
Langkah 2. Untuk setiap nilai, ketika ada konflik (salah satu nilai fitur bukan 0), nilai *feature bundle* merupakan nilai fitur yang akan di-*bundle* ditambah nilai maksimum fitur *reference*.

Langkah 3. Untuk setiap nilai, jika nilai fitur *reference* adalah 0, nilai pada *feature bundle* adalah nilai fitur yang akan di-*bundle* ditambah nilai maksimum fitur *reference*.

Langkah 4. Untuk setiap nilai, jika nilai fitur yang akan di-*bundle* adalah 0, maka nilai *feature bundle* adalah nilai fitur *reference.*

Langkah 5. Jika nilai fitur *reference* dan nilai fitur yang akan di-*bundle* sama-sama bernilai 0, maka nilai *feature bundle* adalah 0.

F1	**F2**
1	0
0	1
1	0
0	3
2	0
3	0
0	0
1	4
1	0
2	0



F1_F2
1
4
1
6
2
3
0
7
1
2

Gambar 3. *Exclusive Feature Bundling.*

Fitur F1 dan F2 akan di-*bundle,* nilai pada *feature bundle* didapatkan sebagai berikut. Fitur F1 merupakan fitur *reference.* Nilai maksimum pada F1 adalah 3 dan nilai maksimum pada F2 adalah 4.

F1[1] = 1

F2[1] = 0, maka nilai pada *feature bundle* F1_F2[1] = 1

F1[2] = 0

F2[2] = 1, maka nilai pada *feature bundle* F1_F2[2] = 1+3 = 4

F1[3] = 1

F2[3] = 0, maka nilai pada *feature bundle* F1_F2[3] = 1

F1[4] = 0

F2[4] = 3, maka nilai pada *feature bundle* F1_F2[4] = 3+3 = 6

F1[5] = 2

F2[5] = 0, maka nilai pada *feature bundle* F1_F2[5] = 2

F1[6] = 3

F2[6] = 0, maka nilai pada *feature bundle* F1_F2[6] = 3

F1[7] = 0

F2[7] = 0, maka nilai pada *feature bundle* F1_F2[7] = 0

F1[8] = 1

F2[8] = 4, pada indeks ini terjadi konflik, maka nilai pada *feature bundle*

F1_F2[8] = 4+3 = 7

F1[9] = 1

F2[9] = 0, maka nilai pada *feature bundle* F1_F2[9] = 1 = 1

F1[10] = 2

F2[10] = 0, maka nilai pada *feature bundle* F1_F2[10] = 2

2.8. Ekstraksi Fitur

Ekstraksi fitur adalah teknik yang digunakan untuk melakukan transformasi data mentah ke dalam fitur yang dapat merepresentasikan dan mendeskripsikan data tersebut (Salau & Jain, 2019). Teknik ekstraksi fitur berperan dalam mencari fitur yang paling penting dan informatif dari sekumpulan fitur sehingga dapat meningkatkan efisiensi dalam penggunaan memori dan pada saat pemrosesan berlangsung (Guyon *et al*., 2008). Ekstraksi fitur pada sekuens protein menggunakan *tools* yang dikenal sebagai *protein descriptor. Protein descriptor* digunakan untuk mendapatkan informasi ikatan kimia dan sifat yang relevan dari sebuah sekuens protein (Emonts & Buyel, 2023). Berdasarkan panjang hasil yang diperoleh, *protein descriptor* dibagi menjadi dua tipe, yaitu *fixed length (static)* dan *dynamic length protein descriptor*.

Static length protein descriptor merupakan *descriptor* yang menghasilkan jumlah fitur yang selalu sama dan tidak bergantung pada panjang sekuens protein. *Dynamic length protein descriptor* merupakan *descriptor* yang bergantung pada panjang sekuens protein, sehingga menghasilkan jumlah fitur yang berbeda-beda berdasarkan sesuai panjang sekuens. Pada penelitian ini, p*rotein descriptor* yang termasuk ke dalam tipe *static length* adalah *position weight amino acid composition*

(PWAA), *encoding based group weight* (EBGW), *k-nearest neighbour* (KNN), dan *pseudo-position specific scoring matrix* (PsePSSM), sedangkan *protein descriptor* yang termasuk ke dalam tipe *dynamic length* adalah *binary encoding* (BE).

2.8.1. *Binary Encoding (BE)*

Metode ini mentransformasikan sekuens ke dalam bentuk biner sehingga dikenal juga dikenal sebagai *one-hot encoding*. Asam amino diurutkan berdasarkan 'ACDEFGHIKLMNPQRSTVWYX' dan menghasilkan vektor 21 dimensi untuk setiap asam amino. Sebagai contoh, untuk asam amino lisin yang dilambangkan dengan 'K' direpresentasikan dengan urutan '000000001000000000000'. Maka dari itu, sekuens dengan panjang 31 akan menghasilkan 31 * 21 dimensi fitur vektor (Sohrawordi & Al Mehedi Hasan, 2020). Berikut ini adalah contoh penerapan BE pada sekuens 'MMARTKQTARK'.

	A	**C**	**D**	**E**	**F**	**G**	**H**	**I**	**K**	**L**	**M**	**N**	**P**	**Q**	**R**	**S**	**T**	**V**	**W**	**Y**	**X**
M	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
M	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
A	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
T	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
K	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
Q	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
T	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
A	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
K	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0

Jika diberikan sebuah data sebanyak *n* sekuens dengan panjang masing-masing sekuens adalah m, penerapan BE menghasilkan matriks dengan ukuran $n \times m \times 21$.

2.8.2. *Position Weight Amino Acid Composition (PWAA)*

Menurut Shi *et al*. (2012), PWAA mengekstrak informasi posisi asam amino dalam sekuens protein. Jika sekuens protein dengan panjang $2n + 1$, a_i (i = 1, 2, …, 20) adalah asam amino dan informasi posisinya dapat dihitung dengan Persamaan (1).

$$C_i = \frac{1}{L\ (L+1)} \sum_{j=-L}^{L} x_{i,j} \left(j + \frac{|j|}{L}\right) \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(1)$$

Dimana L adalah banyak *upstream* atau *downstream* pada sekuens protein. Jika a_i adalah residu ke-j dari sekuens tersebut, maka $x_{i,j} = 1$, selain itu $x_{i,j}, = 0$.

Berikut ini adalah contoh penerapan PWAA pada sekuens 'MMARTKQTARK' dengan L = 5. Informasi posisi asam amino A (*i* = 1) dapat dihitung sebagai berikut.

$$\begin{aligned}
\sum_{j=-L}^{L} X_{i,j} \left(j + \frac{|j|}{L}\right) = & \left(X_{1,-5}\left(-5 + \frac{|-5|}{5}\right)\right) + \left(X_{1,-4}\left(-4 + \frac{|-4|}{5}\right)\right) + \left(X_{1,-3}\left(-3 + \frac{|-3|}{5}\right)\right) \\
& + \left(X_{1,-2}\left(-2 + \frac{|-2|}{5}\right)\right) + \left(X_{1,-1}\left(-1 + \frac{|-1|}{5}\right)\right) + \left(X_{1,0}\left(0 + \frac{|0|}{5}\right)\right) \\
& + \left(X_{1,1}\left(1 + \frac{|1|}{5}\right)\right) + \left(X_{1,2}\left(2 + \frac{|2|}{5}\right)\right) + \left(X_{1,3}\left(3 + \frac{|3|}{5}\right)\right) \\
& + \left(X_{1,4}\left(4 + \frac{|4|}{5}\right)\right) + \left(X_{1,5}\left(5 + \frac{|5|}{5}\right)\right) \\
= & \left(0\left(-5 + \frac{|-5|}{5}\right)\right) + \left(0\left(-4 + \frac{|-4|}{5}\right)\right) + \left(1\left(-3 + \frac{|-3|}{5}\right)\right) \\
& + \left(0\left(-2 + \frac{|-2|}{5}\right)\right) + \left(0\left(-1 + \frac{|-1|}{5}\right)\right) + \left(0\left(0 + \frac{|0|}{5}\right)\right) \\
& + \left(0\left(1 + \frac{|1|}{5}\right)\right) + \left(0\left(2 + \frac{|2|}{5}\right)\right) + \left(1\left(3 + \frac{|3|}{5}\right)\right) \\
& + \left(0\left(4 + \frac{|4|}{5}\right)\right) + \left(0\left(5 + \frac{|5|}{5}\right)\right) \\
= & (0 + 0 + (2,4) + 0 + 0 + 0 + 0 + 0 + 3,6) \\
= & 1,2
\end{aligned}$$

$$C_1 = \frac{1}{5(5+1)} \sum_{j=-L}^{L} x_{i,j} \left(j + \frac{|j|}{L}\right)$$

$$= \frac{1}{30}(1,2) = 0,04$$

Langkah di atas diulang hingga *i* = 20. Dengan begitu, hasil ekstraksi fitur PWAA untuk sekuens 'MMARTKQTARK' adalah [0,04, …, $PWAA_{20}$].

2.8.3. *Encoding Based Group Weight (EBGW)*

Metode *encoding based group weight* (EBGW) diusulkan oleh Zhang *et al*. (2006) untuk mengekstrak karakteristik sifat fisikokimia protein. EBGW didasarkan pada sifat fisikokimia urutan asam amino menggunakan konsep yang disebut *'coarse graining'* untuk mengonversi sekuens protein ke dalam bentuk sekuens karakteristik biner. Sekuens protein terdiri dari 20 asam amino melalui serangkaian reaksi biokimia, sehingga asam amino yang berbeda pasti memiliki sifat fisikokimia yang berbeda. Berdasarkan perbedaan tersebut, asam amino dapat dibagi menjadi empat kategori, yaitu asam amino asam C_1 = {A, F, G, I, L, M, P, V, W}, asam amino basa C_2 = {Q, N, S, T, Y, C}, asam amino netral dan polar C_3 = {D, E}, dan asam amino netral dan hidrofobik C_4 = {H, K, R}. Keempat tipe asam amino dikombinasikan satu persatu untuk mendapatkan tiga metode pembagian baru, yaitu {C_1, C_2}, dan {C_3, C_4}, {C_1, C_3} dan {C_2, C_4}, {C_1, C_4} dan {C_2, C_3}. Setelah protein dipetakan berdasarkan tiga kelompok ini, sekuens protein direduksi menjadi sekuens fitur biner.

Diberikan sekuens protein $P = p_1 p_2 \dots p_n$ ditransformasikan ke dalam tiga sekuens biner menurut Persamaan (2) – (5).

$$H_i(P) = H_i(p_1)H_i(p_2) \dots H_i(p_n), \quad (i = 1, 2, 3) \qquad \dots(2)$$

$$H_1(P_j) = \begin{cases} 1 \ if \ p_j \in \{C_1, C_2\} \\ 0 \ if \ p_j \in \{C_3, C_4\} \end{cases}, \quad (j = 1, \dots, n) \qquad \dots(3)$$

$$H_2(P_j) = \begin{cases} 1 \ if \ p_j \in \{C_1, C_3\} \\ 0 \ if \ p_j \in \{C_2, C_4\} \end{cases}, \quad (j = 1, \dots, n) \qquad \dots(4)$$

$$H_3(P_j) = \begin{cases} 1 \ if \ p_j \in \{C_1, C_4\} \\ 0 \ if \ p_j \in \{C_2, C_3\} \end{cases}, \quad (j = 1, \dots, n) \qquad \dots(5)$$

dimana $H_i(p)$ merepresentasikan karakteristik sekuens protein. Jika diberikan fitur sekuens $H_i(p)$ dengan panjang n, *weight* (w) merujuk pada jumlah kejadian dari '1' pada sekuens, dan *weight* yang dinormalisasi $w(n)$ merujuk pada frekuensi kejadian. Fitur ini dapat didapatkan dengan cara membagi sekuens mentah ke dalam subsekuens L. kemudian, $w(n)$ dari setiap subsekuens dihitung untuk mendapatkan

vektor *L-dimensional* dari H_i. Transformasi dari $H_i(p)$ ke H_i adalah pengkodean bobot fitur sekuens. Ketiga fitur sekuens digunakan untuk mendapatkan tiga vektor yang dikombinasikan untuk mendapatkan vektor *3L-dimensional*, dinotasikan sebagai $W = [w_{1,} w_2, \dots, w_{3L}]$, dimana *W* merujuk pada pengkodean bobot kelompok sekuens protein *P*.

Berikut ini contoh penerapan EBGW pada sekuens 'MMARTKQTARK' dengan subsekuens L = 1 dan L = 2.

Langkah 1. Transformasikan sekuens protein ke dalam tiga bentuk urutan biner *H*(*n*). Kemudian hitung *weight w* untuk masing-masing bentuk biner. *W* adalah banyak digit 1 pada urutan tersebut. Setelah itu, hitung *weight* yang dinormalisasi *w*(*n*), yaitu frekuensi digit 1 pada tiap bentuk biner.

$H_1(n) = 11101011100, \quad w_1 = 7, \quad w_1(n) = \frac{7}{11}$

$H_2(n) = 11100000100, \quad w_2 = 4, \quad w_2(n) = \frac{4}{11}$

$H_3(n) = 11110100111, \quad w_3 = 8, \quad w_3(n) = \frac{8}{11}$

Langkah 2. Partisi *H*(*n*) ke dalam *L* subsekuens.

Jika $H\left(\left\lfloor\frac{kn}{L}\right\rfloor\right)(k = 1, 2, \dots, L)$ adalah subsekuens dari *H*(*n*) dengan panjang sekuens $\left\lfloor\frac{kn}{L}\right\rfloor(k = 1, 2, \dots, L)$. Maka, $w\left(\left\lfloor\frac{kn}{L}\right\rfloor\right)(k = 1, 2, \dots, L)$ adalah *weight* yang dinormalisasi dari subsekuens $H\left(\left\lfloor\frac{kn}{L}\right\rfloor\right)(k = 1, 2, \dots, L)$.

Ketika L = 1,

$H_1\left(\left\lfloor\frac{1.11}{1}\right\rfloor\right) = 11101011100, \quad w_1\left(\left\lfloor\frac{1.11}{1}\right\rfloor\right) = \frac{7}{11} = 0{,}6363$

$H_2\left(\left\lfloor\frac{1.11}{1}\right\rfloor\right) = 11100000100, \quad w_2\left(\left\lfloor\frac{1.11}{1}\right\rfloor\right) = \frac{4}{11} = 0{,}3636$

$H_3\left(\left\lfloor\frac{1.11}{1}\right\rfloor\right) = 11110100111, \quad w_3\left(\left\lfloor\frac{1.11}{1}\right\rfloor\right) = \frac{8}{11} = 0{,}7272$

Maka, $W_1 = [0{,}6363]$, $W_2 = [0{,}3636]$, dan $W_3 = [0{,}7272]$. Fitur EBGW yang dihasilkan ketika L = 1 adalah $X = [0{,}6363;\ 0{,}3636;\ 0{,}7272]$.

Ketika L = 2,

$$H_1\left(\left\lfloor\frac{1.11}{2}\right\rfloor\right) = 11101, \qquad w_1\left(\left\lfloor\frac{1.11}{2}\right\rfloor\right) = \frac{4}{5} = 0{,}8$$

$$H_1\left(\left\lfloor\frac{2.11}{2}\right\rfloor\right) = 11101011100, \qquad w_1\left(\left\lfloor\frac{2.11}{2}\right\rfloor\right) = \frac{7}{11} = 0{,}6363$$

$$H_2\left(\left\lfloor\frac{1.11}{2}\right\rfloor\right) = 11100, \qquad w_2\left(\left\lfloor\frac{1.11}{2}\right\rfloor\right) = \frac{3}{5} = 0{,}6$$

$$H_2\left(\left\lfloor\frac{2.11}{2}\right\rfloor\right) = 11100000100, \qquad w_2\left(\left\lfloor\frac{2.11}{2}\right\rfloor\right) = \frac{4}{11} = 0{,}3636$$

$$H_3\left(\left\lfloor\frac{1.11}{2}\right\rfloor\right) = 11110, \qquad w_3\left(\left\lfloor\frac{1.11}{2}\right\rfloor\right) = \frac{4}{5} = 0{,}8$$

$$H_3\left(\left\lfloor\frac{2.11}{2}\right\rfloor\right) = 11110100111, \qquad w_3\left(\left\lfloor\frac{2.11}{2}\right\rfloor\right) = \frac{8}{11} = 0{,}7272$$

Maka, $W_1 = [0{,}8;\ 0{,}6363]$, $W_2 = [0{,}6;\ 0{,}3636]$, dan $W_3 = [0{,}8;\ 0{,}7272]$. Fitur EBGW yang dihasilkan ketika L = 2 adalah $X = [0{,}8;\ 0{,}6363;\ 0{,}6;\ 0{,}3636;\ 0{,}8;\ 0{,}7272]$.

2.8.4. *K-Nearest Neighbour (KNN)*

Menurut Gao *et al*. (2010), metode ini mengekstrak fitur sekuens protein berdasarkan kesamaan sekuens lokal. Pertama, dihitung jarak antara situs *test* dan seluruh situs yang diketahui. Jika ada dua sekuens protein lokal, yaitu $S_1 = (S_1(1), S_1(2), \ldots, S_1(N))$ dan $S_2 = (S_2(1), S_2(2), \ldots, S_2(N))$, jarak antara dua protein tersebut dapat didefinisikan seperti Persamaan (6).

$$Dist(S_1, S_2) = 1 - \frac{\sum_{j=1}^{L} sim\big(S_1(j), S_2(j)\big)}{N} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(6)$$

Dengan *N* adalah banyak asam amino dalam fragmen sekuens protein. Matriks skor kemiripan (*similarity*) asam amino dapat didefinisikan seperti Persamaan (7).

$$sim\big(S_1(j), S_2(j)\big) = \frac{Matrix\big(S_1(j), S_2(j)\big) - \min\{matrix\}}{\max\{Matrix\} - \min\{Matrix\}} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots(7)$$

Dimana $S_i(j)$ *(i = 1, 2)* merepresentasikan asam amino dari segmen sekuens protein. *Matrix* adalah BLOSUM62 matriks substitusi, sedangkan *max{Matrix}* atau

min{*Matrix*} adalah nilai maksimal atau minimal pada *Matrix.* Berikut ini adalah contoh penerapan ekstraksi fitur KNN dengan *k = 3*.

Langkah 1. Bentuk kumpulan data yang terdiri dari data positif dan negatif.

ID	Sekuens	Kelas
S1	KQTARKSTGGK	1
S2	GKAPRKQLATK	1
S3	RKRSRKESYSI	1
S4	KSTGGKAPRKQ	0
S5	ATGGVKKPHRY	0
S6	ELLIRKLPFQR	0

Langkah 2. Hitung jarak antara sekuens uji (*Sx*) dan tiap sekuens dari kumpulan yang terbentuk pada Langkah 1, lalu urutkan dari yang terkecil.

Hitung *similarity* antara tiap asam amino *Sx* = MMARTKQTARK dan S1.

$$sim(M,K) = \frac{(-1)-(-4)}{11-(-4)} = \frac{3}{15} \qquad sim(Q,S) = \frac{0-(-4)}{11-(-4)} = \frac{4}{15}$$

$$sim(M,Q) = \frac{0-(-4)}{11-(-4)} = \frac{4}{15} \qquad sim(T,T) = \frac{5-(-4)}{11-(-4)} = \frac{9}{15}$$

$$sim(A,T) = \frac{0-(-4)}{11-(-4)} = \frac{4}{15} \qquad sim(A,G) = \frac{0-(-4)}{11-(-4)} = \frac{4}{15}$$

$$sim(R,A) = \frac{(-1)-(-4)}{11-(-4)} = \frac{3}{15} \qquad sim(R,G) = \frac{(-2)-(-4)}{11-(-4)} = \frac{2}{15}$$

$$sim(T,R) = \frac{(-1)-(-4)}{11-(-4)} = \frac{3}{15} \qquad sim(K,K) = \frac{5-(-4)}{11-(-4)} = \frac{9}{15}$$

$$sim(K,K) = \frac{5-(-4)}{11-(-4)} = \frac{9}{15}$$

Maka, jarak antara *Sx* dan S1 adalah,

$$Dist(Sx,S1) = 1 - \frac{\frac{3}{15}+\frac{4}{15}+\frac{4}{15}+\frac{3}{15}+\frac{3}{15}+\frac{9}{15}+\frac{4}{15}+\frac{9}{15}+\frac{4}{15}+\frac{2}{15}+\frac{9}{15}}{11} = 0{,}67$$

Setelah jarak *Sx* dan semua sekuens dihitung, jarak terdekat dengan *Sx* adalah S2, S1, S5, S4, S6, dan S3.

Langkah 3. Pilih *k* tetangga terdekat, dalam contoh ini *k = 3.*

Tiga tetangga terdekat dari *Sx* adalah S2, S1, dan S4. Nilai fitur KNN untuk sekuens *Sx* adalah persentase kelas positif dari tetangga tersebut, yaitu $\frac{2}{3}$ atau 0,67.

2.8.5. *Pseudo-Position Specific Scoring Matrix (PsePSSM)*

PsePSSM yang diusulkan oleh Chou & Shen. (2007) digunakan untuk mengekstrak informasi evolusi sekuens protein. Sekuens protein sepanjang *N* dibandingkan dengan basis data nonredundant *Swiss-prot* dengan metode iteratif PSI-BLAST untuk mendapatkan matriks PSSM dengan dimensi *N* × *20*. Kemudian, matriks tersebut diproses sebagai berikut:

a) Menyaring baris vektor matriks. Nilai maksimum *MAXpi* setiap baris dibandingkan dengan nilai tetap *T* dan baris vektor yang nilainya kurang dari *T* akan dihapus.
b) Melakukan standarisasi matriks yang dihasilkan dan setiap elemen dalam matriks tersebut diskalakan ke dalam angka antara 0 dan 1 menurut rumus pada Persamaan (8).

$$f(x) = \frac{1}{(1 + e^{-x})} \quad \text{.....(8)}$$

c) Kolom vektor dari matriks yang dinormalisasi dihitung rata-ratanya dan model PSSM-AAC diperoleh dengan Persamaan (9).

$$P_{PSSM-AAC} = [\overline{x_1}, \overline{x_2}, \dots, \overline{x_{20}}] \quad \text{.....(9)}$$

dimana $\overline{x_j}(j = 1, 2, \dots, 20)$ merepresentasikan komponen asam amino tipe *j* di PSSM.

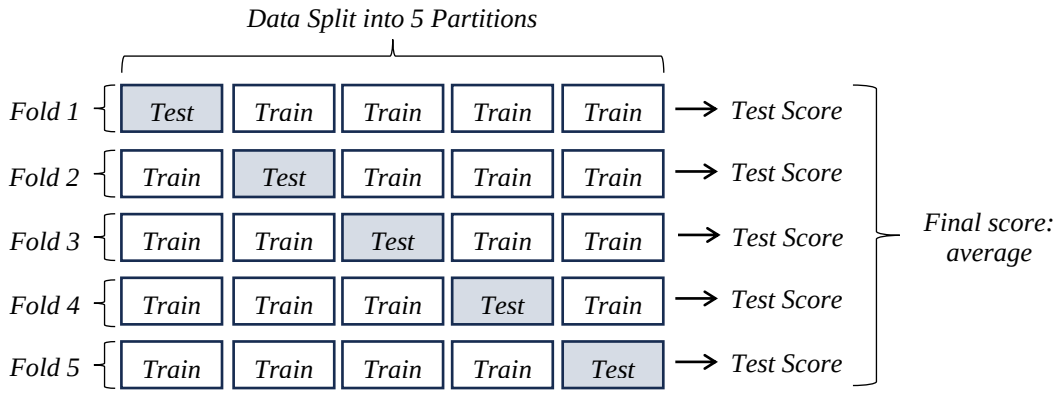
d) Hitung faktor korelasi urutan dari asam amino tipe *j* dengan Persamaan (10).

$$\theta_j^{\varepsilon} = \frac{1}{L - \varepsilon} \sum_{i=1}^{L-\varepsilon} (P_{i,j} - P_{i+\varepsilon j})^2 \; (j = 1, 2, \dots, 20;\ \varepsilon \neq 0;\ \varepsilon < L) \quad \text{.....(10)}$$

Setelah itu, sekuens protein ditransformasikan ke dalam vektor $P_{PSSM} = (x_1, x_2, \dots, x_{20,} \theta_1^1, \theta_2^1, \dots, \theta_{20}^1, \theta_1^{\varepsilon} \theta_2^{\varepsilon}, \dots, \theta_{20}^{\varepsilon})$.

2.9. *Cross-Validation*

Cross-validation merupakan bentuk teknik *resampling* berbasis *bootstrap* untuk estimasi prediksi eror (Bates *et al*., 2022), menggunakan semua data yang tersedia sebagai *training* dan *test*. Penggunaan *training set* dan *test set* dengan cara melatih algoritme sebanyak *k* dengan pembagian 1/*k* dari *training set* dipisahkan untuk tujuan pengujian (Bengio & Grandvalet, 2004). Jenis CV yang umum digunakan yaitu *hold-out*, *k-fold*, dan *leave-one-out* CV. Prosedur *hold-out* membagi data secara acak menjadi dua bagian, *training set* untuk sesi pelatihan dan *hold-out set* untuk mengukur performa model. *K-fold* digunakan dengan harapan lebih akurat dibandingkan *hold-out* tanpa mengurangi jumlah data latih. Prosedur *k-fold* membagi data secara acak ke dalam *fold* sebanyak *k* dengan jumlah yang sama. Kemudian, untuk *fold* ke-*k*, latih model menggunakan *k-1 fold* dan hitung prediksi error dari model yang telah dilatih Ketika memprediksi *fold* ke-*k* tersebut. Hal ini diulang untuk setiap *fold* yang ada, dan estimasi hasil *k-fold* merupakan rata-rata dari hasil setiap *fold* (Blum *et al*., 1999). Sedangkan prosedur *leave-one-out* sama seperti *k*-fold dengan *k*=*n*. Gambar 4 menggambarkan proses dalam *k-fold cross-validation.*



Gambar 4. *K-Fold Cross-Validation* (Phung & Rhee, 2019).

2.10. Metrik Evaluasi

Metrik evaluasi yang digunakan untuk mengukur hasil klasifikasi pada penelitian ini adalah *Accuracy* (Acc), *Sensitivity* (Sp), *Specificity* (Sn)*, Matthew's Correlation Coefficient* (MCC), *Area Under Receiver Operator Characteristic* (AUC).

2.10.1. *Confusion Matrix*

Confusion matrix digunakan untuk mengevaluasi performa kasus metode klasifikasi. *Confusion matrix* mengandung informasi yang membandingkan hasil prediksi dengan data sebenarnya (Rahmad *et al*., 2020). Pada *dataset* dengan kelas sebanyak *n*, maka *confusion matrix* berukuran *n x n* (He & Garcia, 2009). Tabel 6 menggambarkan *confusion matrix* untuk data dengan dua kelas.

Tabel 6. *Confusion Matrix* (Sun et al., 2009)

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

Berikut adalah istilah yang digunakan dalam *confusion matrix:*

1. *True Positive* (TP), yaitu data positif diprediksi dengan benar.
2. *False Positive* (FP), yaitu data positif yang salah diprediksi.
3. *True Negative* (TN), yaitu data negatif diprediksi dengan benar.
4. *False Negative* (FN), yaitu data negatif yang salah diprediksi.

2.10.2. *Accuracy*

Akurasi adalah metrik evaluasi yang paling umum digunakan untuk menilai performa klasifikasi (Sun *et al*., 2009). Persamaan (11) menunjukkan persamaan akurasi.

$$Acc = \frac{TP + TN}{TP + TN + FP + FN} \quad \text{.....(11)}$$

2.10.3. *Sensitivity/True Positive Rate*

Sensitivity atau *Recall* adalah ukuran akurasi untuk kelas positif (berapa banyak sampel dari kelas positif yang diprediksi benar) (Bekkar *et al*., 2013). Persamaan (12) menunjukkan persamaan *sensitivity.*

$$Sn = \frac{TP}{TP + FN} \quad \text{.....(12)}$$

2.10.4. *Specificity*

Specificity adalah probabilitas kelas negative diprediksi benar (Bekkar *et al*., 2013). Persamaan (13) adalah persamaan untuk menghitung *specificity.*

$$Sp = \frac{TN}{TN + FP} \quad \text{……...……..……..……(13)}$$

2.10.5. *Matthew's Correlation Coefficient*

MCC adalah metrik pengukuran performa yang kurang dipengaruhi oleh ketidakseimbangan data, karena MCC mempertimbangkan akurasi dan *error rate* pada kedua kelas dan mengikutsertakan semua nilai pada *confusion matrix* (Bekkar *et al*., 2013). Persamaan (14) adalah rumus MCC.

$$MCC = \frac{TP * TN - FP * FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \quad \text{……..…(14)}$$

2.10.6. ROC-AUC

Receiver Operator Characteristic (ROC) adalah grafik berbentuk kurva probabilitas yang menunjukkan performa model klasifikasi pada semua *threshold* menggunakan nilai *True Positive Rate* (TPR) dan *False Positive Rate* (FPR). Rumus FPR dapat dilihat pada Persamaan (15). Sedangkan *Area Under Curve* (AUC) adalah area dibawah kurva ROC sebagai ukuran kemampuan *binary classifier* dalam membedakan kelas.

$$FPR = \frac{TN}{TN + FP} = 1 - Specificity \quad \text{…….……..……(15)}$$

III. METODOLOGI PENELITIAN

3.1. Tempat dan Waktu Penelitian

Penjelasan mengenai tempat dan waktu penelitian adalah sebagai berikut.

3.1.1. Tempat Penelitian

Penelitian dilakukan di Laboratorium Rekayasa Perangkat Lunak (RPL), Jurusan Ilmu Komputer, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Lampung. Lokasi beralamat di Jalan Prof. Dr. Soemantri Brodjonegoro No.1, Gedung Meneng, Kota Bandar Lampung, Lampung 35145.

3.1.2. Waktu Penelitian

Penelitian dimulai pada bulan Desember tahun 2022 hingga bulan Agustus tahun 2023. Tahap pertama merupakan tahap studi literatur dan pengumpulan data yang dilakukan selama kurang lebih 13 minggu. Tahap kedua yaitu tahap pengerjaan riset, mulai dari melakukan ekstraksi fitur selama 11 minggu hingga evaluasi model yang akan dilakukan selama 8 minggu. Penyusunan laporan dilakukan sejak awal penelitian hingga akhir penelitian. Tabel 7 menampilkan alur waktu penelitian yang dilakukan.

3.2. Data dan Alat

3.2.1. Data

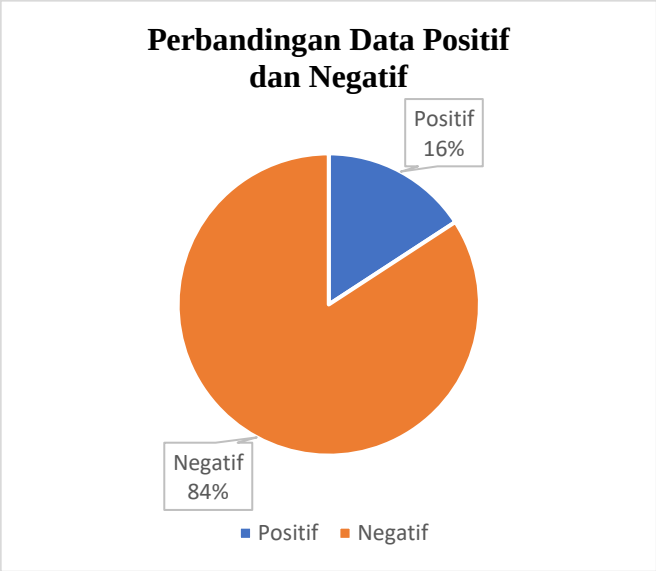
Dataset yang digunakan dalam penelitian ini diperoleh dari penelitian Liu *et al*. (2020). Data ini terdiri dari 159 sampel positif dan 847 sampel negatif yang dikumpulkan dari *database* Uniprot oleh Qiu *et al*. (2017) dengan kata kunci "*histone human and mouse*". Contoh data yang digunakan dapat dilihat pada Tabel 8. Persentase perbandingan antara data positif dan negatif ditampilkan pada Gambar 5.

Tabel 7. Alur Waktu Penelitian

Kegiatan	2022																2023																			
	Desember				Januari				Februari				Maret				April				Mei				Juni				Juli				Agustus			
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
Studi Literatur																																				
Pengumpulan Data																																				
Ekstraksi Fitur																																				
BE																																				
PWAA																																				
KNN																																				
EBGW																																				
PsePSSM																																				
Pemodelan dan																																				
Klasifikasi LightGBM																																				
Oversampling																																				
ADASYN																																				
Evaluasi Model																																				
Penyusunan Laporan																																				

Tabel 8. Contoh Data dari Penelitian Liu et al. (2020)

Sekuens	Label
SKRATQKTRAMMART**K**QTARKSTGGKAPRKQ	Positif
QKTRAMMARTKQTAR**K**STGGKAPRKQLATKA	Positif
MMARTKQTARKSTGG**K**APRKQLATKAARKSA	Negatif
ATKAARKSAPATGGV**K**KPHRYRPGTVALREI	Negatif
TKAARKSAPATGGVK**K**PHRYRPGTVALREIR	Negatif



Gambar 5. Perbandingan Data Positif dan Negatif.

3.2.2. Alat

Berikut alat yang digunakan dalam mendukung penelitian ini:

a. Perangkat Keras *(Hardware)*

Perangkat keras yang digunakan berupa laptop dengan spesifikasi:

1) *Processor:* Intel® Core™ i3-6006U CPU @ 2.00GHz,
2) *RAM:* 8 GB,
3) *Storage: SSD* 480 GB.

b. Perangkat Lunak *(Software)*

Perangkat lunak yang digunakan dalam penelitian ini yaitu:

1) Sistem Operasi: Windows 10 Pro 64-bit dan Ubuntu 22.04.2 LTS 64-bit
2) Python 3.10.9

Python termasuk dalam bahasa pemrograman tingkat tinggi yang sering digunakan dalam *data science.* Bahasa pemrograman berorientasi objek

ini diciptakan oleh Guiddo van Rossum. Python memungkinkan pemrogram untuk menulis kode dalam baris yang lebih sedikit dibanding bahasa pemrograman lainnya (Kalaiarasi & Gopinath, 2021).

3) Jupyter Notebook 6.5.4

   Jupyter Notebook adalah sebuah alat berbasis *browser* yang bersifat *open source* dan berfungsi untuk mendukung alur kerja, kode, data, dan visualisasi dalam proses riset (Randles *et al*., 2017).

4) *Library* NumPy 1.24.3

   NumPy merupakan representasi dari *Numerical Python* yang dapat mengimplementasikan komputasi numerik secara efisien. NumPy memiliki kumpulan fungsi matematika yang banyak digunakan secara luas oleh akademisi maupun industri (Van der Walt & Aivazis, 2011).

5) *Library* Pandas 1.5.3

   Pandas adalah *tools* yang digunakan untuk data terstruktur, seperti statistik, keuangan, ilmu sosial, dan bidang lainnya. *Library* ini menyediakan fungsi terintegrasi untuk memanipulasi data dan analisis pada data tersebut (McKinney, 2011).

6) *Library* Matplotlib 3.7.1

   Matplotlib merupakan *package* untuk melakukan visualisasi data seperti grafik. Matplotlib didesain untuk dapat membuat visualisasi dengan sedikit perintah (Ari & Ustazhanov, 2014).

7) *Library* Scikit-learn 1.2.2

   Scikit-learn merupakan modul Python yang mengintegrasikan algoritma-algoritma *machine-learning*. *Package* ini berfokus dalam mengenalkan *machine learning* untuk nonspesialis (Pedregosa *et al*., 2011).

8) *Library* Imbalanced-learn 0.10.1

   Imbalanced-learn adalah *library open-source* yang menyediakan berbagai metode untuk menangani ketidakseimbangan data seperti ADASYN dalam kasus-kasus *machine learning* dan pengenalan pola (Lemaître *et al*., 2017).

9) *Library* LightGBM 3.3.5

LightGBM adalah *framework* yang menggunakan algoritma pembelajaran berbasis *gradient boosting decision tree.* Metode ini dapat mempercepat proses pembelajaran dibanding GBDT konvensional juga lebih baik dalam hal komputasi dan memori dibanding XGBoost dan SGB (Ke *et al*., 2017).

10) NCBI BLAST+ 2.13.0

BLAST+ merupakan program yang digunakan untuk mendapatkan informasi hubungan fungsional atau evolusioner sekuens dan membantu mengidentifikasi kemiripan gen. Jenis program yang digunakan pada penelitian ini adalah PSI-BLAST (Altschul *et al*., 1997) untuk mengekstrak profil PSSM setiap sekuens.

11) *Library* pssmpro 0.0.2

Pssmpro memiliki 21 fitur yang mampu mengodekan sekuens protein menggunakan profil PSSM (Banerjee, 2021). *Library* ini digunakan untuk mendapatkan file PSSM untuk setiap sekuens dalam dataset.

12) POSSUM *Standalone Toolkit* 1.0

POSSUM adalah alat yang memungkinkan penggunanya mendapatkan berbagai representasi numerik (fitur) berdasarkan PSSM untuk sekuens protein (J. Wang *et al*., 2017). POSSUM dalam hal ini digunakan untuk mengekstrak fitur PsePSSM.

13) ProFeatX *Standalone Toolkit*

ProFeatX merupakan *web server* yang memiliki 32 deskriptor protein, versi independennya menawarkan 50 deskriptor tanpa batasan ukuran file (Guevara-Barrientos & Kaundal, 2023). ProFeatX digunakan untuk mengekstrak fitur EBGW.

3.3. Alur Kerja Penelitian

Gambar 6 menggambarkan alur penelitian dengan penjelasan sebagai berikut.

1) Data Protein

Data protein Lisin diperoleh dari penelitian Liu *et al*. (2020) sebanyak 1006 data berupa 159 data positif dan 847 data negatif. Panjang sekuens protein adalah 31.

2) Ekstraksi Fitur

   Ekstraksi fitur dilakukan untuk mengubah sekuens protein yang merupakan tipe data *string* menjadi numerik. Fitur yang digunakan adalah BE, PWAA, EBGW, KNN, dan PsePSSM.

3) Pembagian Data

   Pembagian data dilakukan dalam tiga skema. Skema pertama, data latih sebanyak 70% data dan data uji sebanyak 30% data. Skema kedua, data latih sebanyak 80% data dan data uji sebanyak 20% data. Skema ketiga, data latih sebanyak 90% data dan data uji sebanyak 10% data.

4) *Oversampling*

   Penerapan *oversampling* menggunakan algoritme ADASYN dilakukan pada data latih. Algoritme tersebut digunakan untuk membuat data baru pada kelas minoritas agar jumlahnya seimbang dengan jumlah data pada kelas mayoritas. Parameter ADASYN yang digunakan merupakan parameter *n_neighbors* dengan nilai 5 (*default*), 7, dan 9. Parameter lainnya menggunakan parameter *default*.

5) Klasifikasi

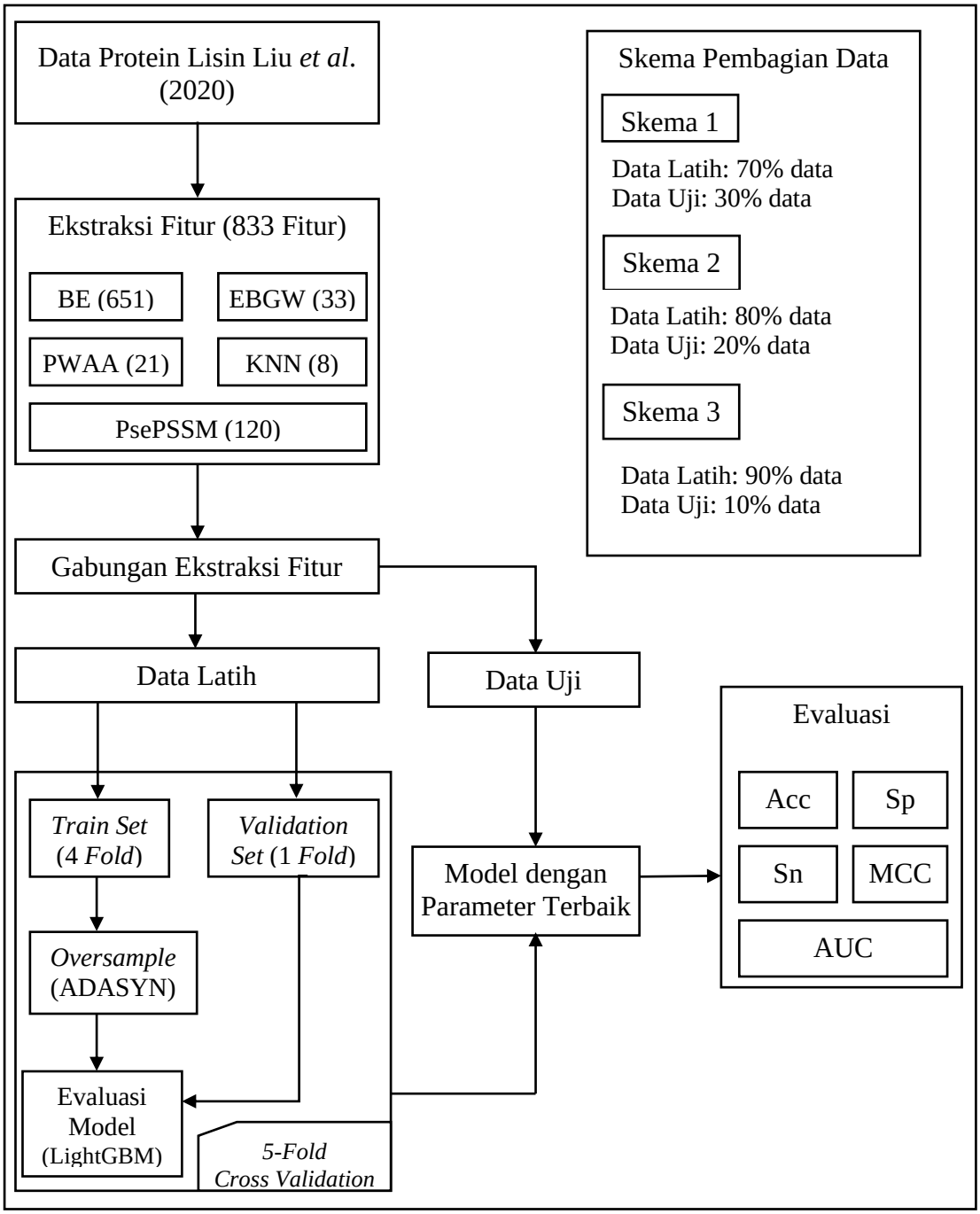
   Metode klasifikasi yang digunakan merupakan algoritme LightGBM. *Hyperparameter* LightGBM yang digunakan ada dua, yaitu *max_depth* dan *learning_rate*. *max_depth* merupakan tingkat kedalaman maksimum pada suatu *tree.* Semakin tinggi nilai *max_depth,* maka *tree* yang dibuat akan semakin dalam sehingga rentan terhadap *overfitting,* sedangkan *learning_rate* merupakan nilai yang menentukan kontribusi setiap pohon terhadap hasil prediksi akhir dan mengontrol seberapa cepat model belajar. Semakin kecil nilai *learning_rate* maka model akan semakin baik dalam memprediksi hasil, namun membutuhkan waktu yang lebih lama. Nilai dari setiap parameter yang digunakan dapat dilihat pada Tabel 9.

Tabel 9. *Hyperparameter* LightGBM

Hyperparameter	**Nilai**
max_depth	4, 6, 10, 12
learning_rate	0.05, 0.1, 0.15

6) Evaluasi

Evaluasi model diukur dengan beberapa metrik evaluasi, diantaranya adalah *accuracy*, *specificity*, *sensitivity*, MCC, dan AUC.



Gambar 6. Alur Kerja Penelitian.

V. PENUTUP

5.1. Simpulan

Berdasarkan pembahasan mengenai penelitian yang telah dilakukan mengenai klasifikasi *imbalanced data* pada *lysine crotonylation* menggunakan algoritme *oversampling* ADASYN dan metode klasifikasi LightGBM dapat diperoleh kesimpulan sebagai berikut.

1. Penerapan *oversampling* ADASYN pada ketidakseimbangan data *lysine crotonylation* tidak memberi pengaruh signifikan terhadap hasil klasifikasi menggunakan metode klasifikasi LightGBM.
2. Perbandingan penelitian ini dengan penelitian sebelumnya menunjukan bahwa kinerja metode klasifikasi LightGBM dengan penerapan *oversampling* ADASYN dalam menangani ketidakseimbangan data *lysine crotonylation* tidak lebih baik dari penelitian sebelumnya yang telah dilakukan oleh Liu *et al*. (2020) dengan menggunakan algoritme *oversampling* SMOTE dan metode klasifikasi LightGBM.

5.2. Saran

Saran pada penelitian ini antara lain sebagai berikut.

1. Mengekplorasi pendekatan lainnya dalam menangani ketidakseimbangan data, baik pendekatan berdasarkan data, algoritme, ataupun gabungan keduanya.
2. Mengeksplorasi penggunaan metode klasifikasi *gradient boosting* lainnya seperti XGBoost dan CatBoost, atau metode klasifikasi berbasis *Deep Learning* untuk mencari hasil yang lebih optimal.
3. Menambahkan jumlah data yang digunakan dalam penelitian untuk meningkatkan keakuratan model.

DAFTAR PUSTAKA

Altschul, S. F., Madden, T. L., Schäffer, A. A., Zhang, J., Zhang, Z., Miller, W., & Lipman, D. J. (1997). Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. In *Nucleic Acids Research* (Vol. 25, Issue 17). Oxford University Press.

Ari, N., & Ustazhanov, M. (2014). Matplotlib in python. *2014 11th International Conference on Electronics, Computer and Computation (ICECCO)*, 1–6. https://doi.org/10.1109/ICECCO.2014.6997585

Asniar, Maulidevi, N. U., & Surendro, K. (2022). SMOTE-LOF for noise identification in imbalanced data classification. *Journal of King Saud University - Computer and Information Sciences*, *34*(6), 3413–3423. https://doi.org/10.1016/J.JKSUCI.2021.01.014

Banerjee, D. (2021). *deeprob/pssmpro: Latest-release* [Computer software]. https://doi.org/10.5281/ZENODO.5032505

Bates, S., Hastie, T., & Tibshirani, R. (2022). *Cross-validation: What does it estimate and how well does it do it?* (arXiv:2104.00673). arXiv. http://arxiv.org/abs/2104.00673

Bekkar, M., Djemaa, H. K., & Alitouche, T. A. (2013). Evaluation Measures for Models Assessment over Imbalanced Data Sets. *Journal of Information Engineering and Applications*, *3*(10), 27–38.

Bengio, Y., & Grandvalet, Y. (2004). No Unbiased Estimator of the Variance of K-Fold Cross-Validation. *Journal of Machine Learning Research*, *5*, 1089–1105.

Blum, A., Kalai, A., & Langford, J. (1999). Beating the hold-out: Bounds for K-fold and progressive cross-validation. *Proceedings of the Twelfth Annual Conference on Computational Learning Theory*, 203–208. https://doi.org/10.1145/307400.307439

Bos, J., & Muir, T. W. (2018). A Chemical Probe for Protein Crotonylation. *Journal of the American Chemical Society*, *140*(14), 4757–4760. https://doi.org/10.1021/jacs.7b13141

Branco, P., Torgo, L., & Ribeiro, R. P. (2016). A Survey of Predictive Modeling on Imbalanced Domains. *ACM Comput. Surv.*, *49*(2). https://doi.org/10.1145/2907070

Carter, M., & Shieh, J. (2015). Biochemical Assays and Intracellular Signaling. In *Guide to Research Techniques in Neuroscience* (pp. 311–343). Elsevier. https://doi.org/10.1016/b978-0-12-800511-8.00015-0

Chen, H., Li, T., Fan, X., & Luo, C. (2019). Feature selection for imbalanced data based on neighborhood rough sets. *Information Sciences*, *483*, 1–20. https://doi.org/10.1016/J.INS.2019.01.041

Chen, W., Tang, D., Xu, Y., Zou, Y., Sui, W., Dai, Y., & Diao, H. (2018). Comprehensive analysis of lysine crotonylation in proteome of maintenance hemodialysis patients. *Medicine*, *97*(37). https://journals.lww.com/md-journal/Fulltext/2018/09140/Comprehensive_analysis_of_lysine_crotonylation_in.10.aspx

Chou, K.-C., & Shen, H.-B. (2007). MemType-2L: A Web server for predicting membrane proteins and their types by incorporating evolution information through Pse-PSSM. *Biochemical and Biophysical Research Communications*, *360*(2), 339–345. https://doi.org/10.1016/j.bbrc.2007.06.027

Dou, L., Yang, F., Xu, L., & Zou, Q. (2021). A comprehensive review of the imbalance classification of protein post-translational modifications. *Briefings in Bioinformatics*, *22*(5), 1–18. https://doi.org/10.1093/bib/bbab089

Emonts, J., & Buyel, J. F. (2023). An overview of descriptors to capture protein properties – Tools and perspectives in the context of QSAR modeling. *Computational and Structural Biotechnology Journal*, *21*, 3234–3247. https://doi.org/10.1016/j.csbj.2023.05.022

Gao, J., Thelen, J. J., Dunker, A. K., & Xu, D. (2010). Musite , a Tool for Global Prediction of General and Kinase-specific Phosphorylation Sites * □. *Molecular and Cellular Proteomics*, *9*(12), 2586–2600. https://doi.org/10.1074/mcp.M110.001388

Guevara-Barrientos, D., & Kaundal, R. (2023). ProFeatX: A parallelized protein feature extraction suite for machine learning. *Computational and Structural Biotechnology Journal*, *21*, 796–801. https://doi.org/10.1016/j.csbj.2022.12.044

Guyon, I., Gunn, S., Nikravesh, M., & Zadeh, L. A. (2008). *Feature extraction: Foundations and applications* (Vol. 207). Springer.

He, H., Bai, Y., Garcia, E. A. G., & Li, S. (2008). ADASYN: Adaptive synthetic sampling approach for imbalanced learning. In IEEE International Joint

Conference on Neural Networks, 2008. *IJCNN 2008.(IEEE World Congress on Computational Intelligence) (Pp. 1322– 1328), 3*, 1322–1328.

He, H., & Garcia, E. A. (2009). Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering*, *21*(9), 1263–1284.

Kalaiarasi, K., & Gopinath, R. (2021). *Stochastic Lead Time Reduction for Replenishment Python-Based Fuzzy Inventory Order EOQ Model with Machine Learning Support*. *11*(10), 1982–1991. https://doi.org/10.34218/IJARET.11.10.2020.188

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., & Liu, T.-Y. (2017). LightGBM: A Highly Efficient Gradient Boosting Decision Tree. *Advances in Neural Information Processing Systems*, *30*.

Lemaître, G., Nogueira, F., & Aridas, C. K. (2017). Imbalanced-learn: A Python Toolbox to Tackle the Curse of Imbalanced Datasets in Machine Learning. *Journal of Machine Learning Research*, *18*(17), 1–5.

Liu, Y., Yu, Z., Chen, C., Han, Y., & Yu, B. (2020). Prediction of protein crotonylation sites through LightGBM classifier based on SMOTE and elastic net. *Analytical Biochemistry*, *609*(January), 113903. https://doi.org/10.1016/j.ab.2020.113903

Lopez, M. J., & Mohiuddin, S. S. (2022). *Biochemistry, Essential Amino Acids.*

McKinney, W. (2011). pandas: A foundational Python library for data analysis and statistics. *Python for High Performance and Scientific Computing*, *14*(9), 1–9.

Mohammed, R., Rawashdeh, J., & Abdullah, M. (2020). Machine Learning with Oversampling and Undersampling Techniques: Overview Study and Experimental Results. *2020 11th International Conference on Information and Communication Systems (ICICS)*, 243–248.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., & others. (2011). Scikit-learn: Machine learning in Python. *The Journal of Machine Learning Research*, *12*, 2825–2830.

Phung, V. H., & Rhee, E. J. (2019). A High-Accuracy Model Average Ensemble of Convolutional Neural Networks for Classification of Cloud Image Patches on Small Datasets. *Applied Sciences*, *9*(21). https://doi.org/10.3390/app9214500

Qiu, W. R., Sun, B. Q., Tang, H., Huang, J., & Lin, H. (2017). Identify and analysis crotonylation sites in histone by using support vector machines. *Artificial Intelligence in Medicine*, *83*, 75–81. https://doi.org/10.1016/j.artmed.2017.02.007

Qiu, W. R., Sun, B. Q., Xiao, X., Xu, Z. C., & Chou, K. C. (2016). iPTM-mLys: Identifying multiple lysine PTM sites and their different types. *Bioinformatics*, *32*(20), 3116–3123. https://doi.org/10.1093/bioinformatics/btw380

Qiu, W. R., Sun, B. Q., Xiao, X., Xu, Z. C., Jia, J. H., & Chou, K. C. (2018). iKcr-PseEns: Identify lysine crotonylation sites in histone proteins with pseudo components and ensemble classifier. *Genomics*, *110*(5), 239–246. https://doi.org/10.1016/j.ygeno.2017.10.008

Rahmad, F., Suryanto, Y., & Ramli, K. (2020). Performance Comparison of Anti-Spam Technology Using Confusion Matrix Classification. *IOP Conference Series: Materials Science and Engineering*, *879*(1). https://doi.org/10.1088/1757-899X/879/1/012076

Randles, B. M., Pasquetto, I. V., Golshan, M. S., & Borgman, C. L. (2017). Using the Jupyter Notebook as a Tool for Open Science: An Empirical Study. *2017 ACM/IEEE Joint Conference on Digital Libraries (JCDL)*, 1–2. https://doi.org/10.1109/JCDL.2017.7991618

Salau, A. O., & Jain, S. (2019). Feature Extraction: A Survey of the Types, Techniques, Applications. *2019 International Conference on Signal Processing and Communication (ICSC)*, 158–164. https://doi.org/10.1109/ICSC45622.2019.8938371

Shi, S. P., Qiu, J. D., Sun, X. Y., Suo, S. B., Huang, S. Y., & Liang, R. P. (2012). A method to distinguish between lysine acetylation and lysine methylation from protein sequences. *Journal of Theoretical Biology*, *310*, 223–230. https://doi.org/10.1016/j.jtbi.2012.06.030

Shukla, P. (2022, October 20). *Dealing with Sparse Datasets in Machine Learning*. https://www.analyticsvidhya.com/blog/2022/10/dealing-with-sparse-datasets-in-machine-learning/

Sohrawordi, M., & Al Mehedi Hasan, M. (2020). LyFor:Prediction of lysine formylation sites from sequence based features using support vector machine. *2020 IEEE Region 10 Symposium, TENSYMP 2020*, *June*, 250–253. https://doi.org/10.1109/TENSYMP50017.2020.9230689

Sun, Y., Wong, A. K. C., & Kamel, M. S. (2009). Classification of imbalanced data: A review. *International Journal of Pattern Recognition and Artificial Intelligence*, *23*(4), 687–719. https://doi.org/10.1142/S0218001409007326

Suprayitno, E., & Sulistiyati, T. D. (2017). *Metabolisme protein*. Universitas Brawijaya Press.

Tan, M., Luo, H., Lee, S., Jin, F., Yang, J. S., Montellier, E., Buchou, T., Cheng, Z., Rousseaux, S., Rajagopal, N., Lu, Z., Ye, Z., Zhu, Q., Wysocka, J., Ye, Y., Khochbin, S., Ren, B., & Zhao, Y. (2011). Identification of 67 histone

marks and histone lysine crotonylation as a new type of histone modification. *Cell*, *146*(6), 1016–1028. https://doi.org/10.1016/j.cell.2011.08.008

Tng, S. S., Le, N. Q. K., Yeh, H.-Y., & Chua, M. C. H. (2022). Improved Prediction Model of Protein Lysine Crotonylation Sites Using Bidirectional Recurrent Neural Networks. *Journal of Proteome Research*, *21*(1), 265–273. https://doi.org/10.1021/acs.jproteome.1c00848

Urry, L. A., Cain, M. L., Wasserman, S. A., Reece, J. B., Minorsky, P. V., & Campbell, N. A. (2016). *Campbell Biology 11th Edition*. Pearson Education, Incorporated.

Van der Walt, S., & Aivazis, M. (2011). The NumPy Array: A Structure for Efficient Numerical Computation, Computing in Science & Engineering. *Computing in Science and Engineering*, *13*(2), 22–30.

Wang, J., Yang, B., Revote, J., Leier, A., Marquez-Lago, T. T., Webb, G., Song, J., Chou, K. C., & Lithgow, T. (2017). POSSUM: A bioinformatics toolkit for generating numerical sequence feature descriptors based on PSSM profiles. *Bioinformatics*, *33*(17), 2756–2758. https://doi.org/10.1093/bioinformatics/btx302

Wang, S., Mu, G., Qiu, B., Wang, M., Yu, Z., Wang, W., Wang, J., & Yang, Y. (2021). The Function and related Diseases of Protein Crotonylation. *Int. J. Biol. Sci.*, *17*.

Wei, W., Mao, A., Tang, B., Zeng, Q., Gao, S., Lu, L., Li, W., Du, J. X., Li, J., Wong, J., & Liao, L. (2017). Large-Scale Identification of Protein Crotonylation Reveals Its Role in Multiple Cellular Functions. *Journal of Proteome Research*, *16*(4), 1743–1752. https://doi.org/10.1021/acs.jproteome.7b00012

Wongvorachan, T., He, S., & Bulut, O. (2023). A Comparison of Undersampling, Oversampling, and SMOTE Methods for Dealing with Imbalanced Classification in Educational Data Mining. *Information*, *14*(1), 54. https://doi.org/10.3390/info14010054

Wu, G. (2009). Amino acids: Metabolism, functions, and nutrition. *Amino Acids*, *37*(1), 1–17. https://doi.org/10.1007/s00726-009-0269-0

Yan, J., Xu, Y., Cheng, Q., Jiang, S., Wang, Q., Xiao, Y., Ma, C., Yan, J., & Wang, X. (2021). LightGBM: accelerated genomically designed crop breeding through ensemble learning. *Genome Biology*, *22*(1), 1–24. https://doi.org/10.1186/s13059-021-02492-y

Yen, S.-J. & Yue-Shi Lee. (2009). Cluster-based under-sampling approaches for imbalanced data distributions. *Expert Systems with Applications*, *36*(3), 5718–5727. https://doi.org/10.1016/j.eswa.2008.06.108

Zhang, Z. H., Wang, Z. H., Zhang, Z. R., & Wang, Y. X. (2006). A novel method for apoptosis protein subcellular localization prediction combining encoding based on grouped weight and support vector machine. *FEBS Letters*, *580*(26), 6169–6174. https://doi.org/10.1016/j.febslet.2006.10.017

Zhao, Y., Hao, S., Wu, W., Li, Y., Hou, K., Liu, Y., Cui, W., Xu, X., & Wang, H. (2022). Lysine Crotonylation: An Emerging Player in DNA Damage Response. *Biomolecules*, *12*(10). https://doi.org/10.3390/biom12101428

Zhao, Y., He, N., Chen, Z., & Li, L. (2020). Identification of Protein Lysine Crotonylation Sites by a Deep Learning Framework with Convolutional Neural Networks. *IEEE Access*, *8*, 14244–14252. https://doi.org/10.1109/aCCESS.2020.2966592

Zhu, Y., Jia, C., Li, F., & Song, J. (2020). Inspector: A lysine succinylation predictor based on edited nearest-neighbor undersampling and adaptive synthetic oversampling. *Analytical Biochemistry*, *593*(October 2019), 113592. https://doi.org/10.1016/j.ab.2020.113592