

**KLASIFIKASI *IMBALANCED DATA* MENGGUNAKAN ALGORITME
SMOTE DAN METODE *SUPPORT VECTOR MACHINE*
(STUDI KASUS: METILASI SEQUENCE PROTEIN ARGININ)**

(Skripsi)

Oleh

ESTER CAROLINE LUMBAN GAOL



**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG
2021**

ABSTRAK

KLASIFIKASI IMBALANCED DATA MENGGUNAKAN ALGORITME SMOTE DAN METODE SUPPORT VECTOR MACHINE (STUDI KASUS: METILASI SEQUENCE PROTEIN ARGININ)

Oleh

ESTER CAROLINE LUMBAN GAOL

Post-translational modification (PTM) merupakan modifikasi kovalen yang terjadi setelah translasi pada sintesis protein. Modifikasi ini dapat mengidentifikasi dan menggambarkan proses PTM seperti metilasi. Metilasi ialah penambahan gugus metil pada gugus protein yang dapat mempengaruhi transduksi sinyal hingga mengikat RNA dalam sitoplasma. Metilasi protein Arginin mengikat gugus metil dan nitrogen Arginin yang menghasilkan basa organik. Umumnya, data biologi memunculkan jumlah data yang tidak seimbang antar kelas. Proses klasifikasi yang melibatkan data tidak seimbang akan menyebabkan penurunan nilai akurasi kelas minoritas dan kualitas metode klasifikasi itu sendiri. Oleh karena itu, masalah ketidakseimbangan data menjadi hal penting untuk ditangani dalam bidang *machine learning*. Untuk itu penggunaan algoritme lain selain metode klasifikasi disarankan menangani ketidakseimbangan data. *Synthetic Minority Oversampling Technique* (SMOTE) merupakan teknik *oversampling* yang membuat salinan data kelas minoritas yang mengimplementasikan algoritme *k-nearest neighbor*. *Support Vector Machine* mengelompokkan data dengan *hyperplane* dan memaksimalkan jarak *margin*. Riset ini menggunakan data Metilasi Arginin yang terdiri dari data latih, data uji, dan data independen. Alur kerja penelitian ini terdiri dari tahap: *preprocessing* yang menghapus dan mereduksional data, ekstraksi fitur, pemodelan SMOTE dan SVM, hingga pengujian klasifikasi. Dengan menerapkan pengujian *10-fold cross validation* dan *confusion matrix* diperoleh keakuratan data latih sebesar 100% pada *kernel RBF*, sedangkan data uji hanya sebesar 64,90% di *kernel linear*. Data independen memiliki rata-rata akurasi yang baik dengan persentase 98,50% pada *kernel linear*.

Kata kunci: *Imbalanced Data; Metilasi; Post-Translational Modification, SMOTE, Support Vector Machine*

ABSTRACT

IMBALANCED DATA CLASSIFICATION USING SMOTE ALGORITHM AND SUPPORT VECTOR MACHINE METHOD (CASE STUDY: AN METHYLATION OF SEQUENCE PROTEIN ARGININ)

By

ESTER CAROLINE LUMBAN GAOL

Post-translational modification (PTM) is a covalent modification that occurs after the translation process in protein synthesis. This modification can identify and describe PTM processes such as methylation. Methylation is the addition of a methyl cluster to a protein cluster that can affect signal transduction and RNA binding in the cytoplasm. Protein methylation of Arginine binds to the methyl and nitrogen cluster of Arginine to produce an organic base. Generally, biological data generates an unbalanced amount of data between classes. The classification process that involves unbalanced data will cause a decrease in the accuracy value of the minority class and the quality of the classification method itself. Therefore, the problem of imbalanced data becomes an crucial thing to be addressed in the field of machine learning. For this reason, it is recommended to use other algorithms besides the classification method to handle imbalanced data. Synthetic Minority Oversampling Technique (SMOTE) is an oversampling technique which creates a copy of the minority class data that implements the k-nearest neighbor algorithm. Support Vector Machine groups the data with hyperplanes and maximizes margin space. This research uses Arginine Methylation data which consists of training data, test data, and independent data. The workflow of this research consists of these stages: preprocessing which removes and redundant data, feature extraction, SMOTE and SVM modeling, and classification testing. By applying the 10-fold cross validation scheme and confusion matrix, the accuracy of the training data is 100% in the RBF kernel, whilst the test data is only 64.90% in the linear kernel. Independent data have a decent accuracy with a percentage of 98.50% in the linear kernel.

Keywords : *Imbalanced Data; Methylation; Post-Translational Modification, SMOTE, Support Vector Machine*

**KLASIFIKASI *IMBALANCED DATA* MENGGUNAKAN ALGORITME
SMOTE DAN METODE *SUPPORT VECTOR MACHINE*
(STUDI KASUS: METILASI SEQUENCE PROTEIN ARGININ)**

Oleh
ESTER CAROLINE LUMBAN GAOL

Skripsi

**Sebagai Salah Satu Syarat untuk Mencapai Gelar
SARJANA KOMPUTER**

Pada

**Jurusan Ilmu Komputer
Fakultas Matematika dan Ilmu Pengetahuan Alam
Universitas Lampung**



**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG
2021**

Judul Skripsi

: **KLASIFIKASI IMBALANCED DATA
MENGGUNAKAN ALGORITME SMOTE DAN
METODE SUPPORT VECTOR MACHINE (STUDI
KASUS: METILASI SEQUENCE PROTEIN
ARGININ)**

Nama Mahasiswa

: **Ester Caroline Lumban Gaol**

No. Pokok Mahasiswa: 1717051002

Jurusan

: Ilmu Komputer

Fakultas

: Matematika dan Ilmu Pengetahuan Alam



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

Dewi Asiah Shofiana, S. Komp., M. Kom.
NIP. 19950929 202012 2 030

2. MENGETAHUI
Ketua Jurusan Ilmu Komputer
FMIPA Universitas Lampung

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

MENGESAHKAN

1. Tim Penguji

Ketua

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

The image shows three handwritten signatures stacked vertically. The top signature is for the Ketua, followed by the Sekretaris, and the bottom one is partially visible.

Sekretaris

: **Dewi Asiah Shofiana, S. Komp., M. Kom.**

Penguji

Bukan Pembimbing

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



2. Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam



Dr. Eng. Suripto Dwi Yuwono, S.Si., M.T.

NIP. 19740705 200003 1 001

Tanggal Lulus Ujian Skripsi : **28 September 2020**

PERNYATAAN

Saya yang bertanda tangan di bawah ini:

Nama : Ester Caroline Lumban Gaol
NPM : 1717051002
Jurusan/Prodi : Ilmu Komputer/S1-Ilmu Komputer

Menyatakan bahwa skripsi saya yang berjudul “Klasifikasi *Imbalanced Data* Menggunakan Algoritme SMOTE dan Metode *Support Vector Machine* (Studi Kasus: Metilasi *Sequence Protein Arginin*)” merupakan karya tulis saya sendiri dan bukan karya orang lain. Semua tulisan yang tertuang di skripsi ini telah mengikuti kaidah penulisan karya ilmiah Universitas Lampung. Apabila di kemudian hari terbukti skripsi saya merupakan hasil penjiplakan atau dibuat orang lain, maka saya bersedia menerima sanksi berupa pencabutan gelar yang saya terima.

Bandar Lampung, 14 Oktober 2021



Ester Caroline Lumban Gaol
NPM. 1717051002

RIWAYAT HIDUP



Penulis merupakan anak pertama dari tiga bersaudara dari pasangan Bapak Sejarah Lumban Gaol (Alm) dan Ibu Sida Manik yang dilahirkan di Bandar Lampung, tepatnya pada tanggal 19 Juli 1999. Penulis telah menyelesaikan pendidikan pertama di TK Sejahtera IV Bandar Lampung pada tahun 2005, dan meneruskan pendidikan dasar di SD Negeri 3 Kemiling Permai yang diselesaikan tahun 2011. Jenjang pendidikan menengah pertama ditempuh di SMP Negeri 28 Bandar Lampung dan selesai pada tahun 2014. Selanjutnya, penulis meneruskan pendidikan menengah atas di SMA Negeri 9 Bandar Lampung dan menyelesaikannya pada tahun 2017.

Tahun 2017, penulis terdaftar menjadi mahasiswa di Jurusan Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung. Dalam masa perkuliahan, penulis aktif mengikuti kegiatan organisasi internal kampus, yaitu Himpunan Mahasiswa Ilmu Komputer (Himakom) periode 2017/2018 hingga 2019/2020 dan Persekutuan Oikumene Mahasiswa (POM) MIPA. Pada bulan Desember 2019, penulis melakukan Kerja Praktik di PT, Kereta Api Indonesia Divisi Regional (Divre) IV Tanjungkarang dan Kuliah Kerja Nyata di Kelurahan Kemiling Permai, Kota Bandar Lampung pada Agustus 2020.

PERSEMBAHAN

Oleh karena kasih Tuhan Yesus Kristus melalui kekuatan, sukacita dan pengharapan, ku persembahkan karya ini untuk mereka yang ku kasih dan mengasihiku.

Teruntuk Mama dan kedua adikku: Rose dan Elisa, serta Opung Boru yang tiada lelah membawa nama ku dalam setiap doanya, yang senantiasa menasihati serta menyemangati ku. Terima kasih untuk segala bentuk kasih sayang dan dukungan yang diberikan.

Teruntuk Bapa yang telah bersama dengan Tuhan Yesus Kristus di Firdaus, terima kasih telah mengajarkan dan membentuk ku menjadi wanita yang tegar menjalani hidup meskipun tanpa dirimu.

Teruntuk teman-teman, terima kasih untuk setiap cerita yang boleh tercipta selama perkuliahan.

Almamater Tercinta,

UNIVERSTAS LAMPUNG

MOTTO

“Karena kita tahu, bahwa kesengsaraan itu menimbulkan ketekunan, dan ketekunan menimbulkan tahan uji dan tahan uji menimbulkan pengharapan. Dan pengharapan tidak mengecewakan , karena kasih Allah telah dicurahkan di dalam hati kita oleh Roh Kudus yang telah dikarunakan kepada kita.”

(Roma 5: 3b-5)

“Segala perkara dapat kutanggung di dalam Dia yang memberi kekuatan kepadaku.”

(Filipi 4:13)

“Siapa yang dibenarkan oleh iman, dia akan meresponnya dengan ucapan syukur dan kasih yang menyempurnakan hukum.”

(Martin Luther)

“Dalam situasi apapun, penyertaan Tuhan hadir dengan cara yang tidak terduga.”

(Penulis)

SANWACANA

Shallom.

Puji syukur kepada Allah Bapa Tuhan Yesus Kristus atas penyertaanNya, skripsi yang berjudul “Klasifikasi *Imbalanced Data* Menggunakan Algoritme SMOTE dan Metode *Support Vector Machine* (Studi Kasus: Metilasi *Sequence Protein Arginin*)” dapat diselesaikan penulis dengan baik. Skripsi ini merupakan salah satu syarat guna menyelesaikan proses perkuliahan dan mendapat gelar Sarjana Komputer di Jurusan Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung.

Dalam hal ini, penulis berterima kasih kepada setiap pihak yang berpartisipasi dalam membantu proses penyelesaian skripsi ini. Dalam kesempatan ini, penulis mengucapkan terima kasih kepada:

1. Mama, Opung Boru dan kedua adikku, Rose dan Elisa yang selalu mendoakan dan mendukung saya.
2. Bapak Dr. Eng. Suripto Dwi Yuwono, S.Si., M.T. selaku Dekan Fakultas MIPA Univeristas Lampung.
3. Bapak Favorisen R. Lumbanraja, Ph.D, selaku dosen pembimbing utama sekaligus dosen pembimbing akademik atas kebaikannya dalam memberikan bimbingan, dukungan dan buah pikirin baik saran maupun kritik dalam penyelesaian skripsi.
4. Ibu Dewi Asiah Shofiana, S.Komp., M.Kom., selaku dosen pembimbing kedua atas keikhlasannya dalam proses bimbingan penyusunan skripsi.
5. Bapak Dr.rer.nat. Akmal Junaidi, M.Sc., selaku dosen penguji yang telah memberikan saran dan kritik guna penyempurnaan skripsi.

6. Bapak Didik Kurniawan, S.Si., M.T., selaku Ketua Jurusan Ilmu Komputer FMIPA Universitas Lampung.
7. Ibu Astria Hijriani, S.Kom., M.Kom., selaku Sekretaris Jurusan Ilmu Komputer FMIPA Universitas Lampung.
8. Seluruh dosen Jurusan Ilmu Komputer FMIPA Universitas Lampung yang telah memberikan pengajaran dalam perkuliahan.
9. Seluruh keluarga & saudaraku yang mendukung proses perkuliahan hingga penyelesaian skripsi yang tak dapat disebutkan satu persatu.
10. Sobat Bodats: Naurah, Reda, Reka, Putri, Ika, dan Caca yang menjadi sahabat karib yang saling mendukung satu sama lain dalam perkuliahan.
11. Teman seperbimbingan: Naurah, Rifky, dan Yulita yang saling menguatkan selama masa penyelesaian skripsi.
12. Rekan-rekan *discord server* “D ickKick dari kehidupan”: Irsyaad, Reza, Sigit, Ahong yang membantu dan memberikan saran, mengajari serta menemani penyusunan skripsi di sela-sela waktu bermain *game*.
13. Teman-teman sepelayanan di POM MIPA atas kerjasama dalam pelayanan kampus, juga semangat saling mendoakan satu sama lain.
14. Teman-teman Jurusan Ilmu Komputer FMIPA Universitas Lampung angkatan 2017 yang telah memberikan cerita dalam masa perkuliahan.
15. Semua pihak yang telah berpartisipasi baik secara langsung maupun tidak langsung dalam membantu penyusunan skripsi ini.

Penulis menyadari bahwa dalam penulisan skripsi ini masih terdapat banyak kekurangan karena keterbatasan kemampuan, pengalaman serta pengetahuan penulis. Oleh karena itu, saran dan kritik yang membangun sangat diharapkan sebagai bahan evaluasi untuk kedepannya. Semoga skripsi ini dapat bermanfaat bagi semua pihak.

Bandar Lampung, 14 Oktober 2021
Penulis,

Ester Caroline Lumban Gaol

DAFTAR ISI

	Halaman
DAFTAR ISI.....	v
DAFTAR TABEL	viii
DAFTAR GAMBAR.....	x
DAFTAR PSEUDOCODE	xi
I. PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	4
1.3 Tujuan Penelitian.....	5
1.4 Manfaat Penelitian.....	5
1.5 Batasan Masalah.....	5
II. TINJAUAN PUSTAKA.....	6
2.1 Penelitian Terkait	6
2.2 Protein	10
2.3 <i>Post-Translational Modifications</i>	12
2.4 Metilasi.....	13
2.5 <i>Imbalanced Data</i>	13
2.6 Redundansi Data	15
2.7 <i>Synthetic Minority Oversampling Technique</i>	15
2.8 Metode Klasifikasi	17
2.8.1 <i>Artificial Neural Network</i>	18
2.8.2 <i>Random Forest</i>	19

2.8.3 <i>Support Vector Machine</i>	19
2.9 <i>Feature Extraction</i>	23
2.9.1 <i>Composition, Transition, dan Distribution</i> (CTD).....	23
2.9.2 AAindex	24
2.9.3 <i>Hydrophobicity</i>	25
2.9.4 <i>Pseudo Amino Acid Composition</i> (PseAAC)	26
2.9.5 <i>Quasi-Sequence-Order</i> (QSO)	26
2.10 <i>Cross-Validation</i>	27
2.11 <i>Confusion matrix</i>	29
III. METODOLOGI PENELITIAN	31
3.1 Tempat dan Waktu Penelitian	31
3.1.1 Tempat Penelitian.....	31
3.1.2 Waktu Penelitian	31
3.2 Data dan Alat.....	33
3.2.1 Data	33
3.2.2 Alat	33
3.3 Alur Kerja Penelitian.....	35
IV. HASIL DAN PEMBAHASAN	37
4.1 Prapemrosesan Data	37
4.2 Ekstraksi Fitur	39
4.3 Pemrosesan Data	42
4.4 Pemodelan Klasifikasi.....	44
4.4.1 Pemodelan SMOTE.....	45
4.4.2 Pembagian Data Independen	45
4.4.3 Klasifikasi <i>Support Vector Machine</i>	46
4.5 Pengujian Hasil Klasifikasi	47
4.5.1 Pengujian Tanpa SMOTE	47
4.5.2 Pengujian Menggunakan SMOTE	53
4.6 Pembahasan	66
4.7 Perbandingan dengan Penelitian Sebelumnya.....	70

V. PENUTUP.....	72
5.1 Simpulan.....	72
5.2 Saran.....	73
DAFTAR PUSTAKA	74

DAFTAR TABEL

Tabel	Halaman
1. Penelitian Terkait	6
2. Jenis dan Standar Simbol Asam Amino (Cristianini & Hahn, 2006)	11
3. Contoh Data Tidak Seimbang	17
4. <i>Confusion Matrix</i> (Kuhn & Johnson, 2013).....	29
5. <i>Gantt Chart</i> Waktu Penelitian.....	32
6. Data Metilasi <i>Sequence</i> Protein Arginin (Kumar et al., 2017)	33
7. Jumlah Data Protein Sebelum dan Sesudah Pembersihan Data.....	37
8. Jumlah Data Protein Sebelum dan Sesudah Redundansi	38
9. Data Protein Setelah Ekstraksi Fitur	42
10. Jumlah Dimensi Ekstraksi Fitur	43
11. Confusion Matrix Data Latih Tanpa SMOTE.....	47
12. Hasil Pengujian Klasifikasi Data Latih Tanpa SMOTE	48
13. <i>Confusion Matrix</i> Data Uji Tanpa SMOTE	48
14. Hasil Pengujian Klasifikasi Data Uji Tanpa SMOTE.....	49
15. <i>Confusion Matrix</i> Data Independen Tanpa SMOTE Pada <i>Kernel Linear</i> ..	49
16. Hasil Pengujian Data Independen Tanpa SMOTE Pada <i>Kernel Linear</i>	50
17. <i>Confusion Matrix</i> Data Independen Tanpa SMOTE Pada <i>Kernel Polynomial</i>	50
18. Hasil Pengujian Data Independen Tanpa SMOTE Pada <i>Kernel Polynomial</i> .	51
19. <i>Confusion Matrix</i> Data Independen Tanpa SMOTE Pada <i>Kernel RBF</i>	51
20. Hasil Pengujian Data Independen Tanpa SMOTE Pada <i>Kernel RBF</i>	52
21. <i>Confusion Matrix</i> Data Independen Tanpa SMOTE Pada <i>Kernel Sigmoid</i> ... <td style="text-align: right;">52</td>	52
22. Hasil Pengujian Data Independen Tanpa SMOTE Pada <i>Kernel Sigmoid</i> .	53
23. <i>Confusion Matrix</i> Data Latih Menggunakan SMOTE.....	54
24. Hasil Pengujian Klasifikasi Data Latih Setelah SMOTE.....	55

25. <i>Confusion Matrix</i> Data Uji Setelah SMOTE	55
26. Hasil Pengujian Klasifikasi Data Uji Setelah SMOTE.....	56
27. <i>Confusion Matrix</i> Data Independen SMOTE-1 Pada <i>Kernel Linear</i>	58
28. Hasil Pengujian Data Independen SMOTE-1 Pada <i>Kernel Linear</i>	58
29. <i>Confusion Matrix</i> Data Independen SMOTE-1 Pada <i>Kernel Polynomial</i> .	59
30. Hasil Pengujian Data Independen SMOTE-1 Pada <i>Kernel Polynomial</i>	59
31. <i>Confusion Matrix</i> Data Independen SMOTE-1 Pada <i>Kernel RBF</i>	60
32. Hasil Pengujian Data Independen SMOTE-1 Pada <i>Kernel RBF</i>	60
33. <i>Confusion Matrix</i> Data Independen SMOTE-1 Pada <i>Kernel Sigmoid</i>	61
34. Hasil Pengujian Data Independen SMOTE-1 Pada <i>Kernel Sigmoid</i>	61
35. <i>Confusion Matrix</i> Data Independen SMOTE-2 pada <i>Kernel Linear</i>	62
36. Hasil Pengujian Data Independen SMOTE-2 Pada <i>Kernel Linear</i>	62
37. <i>Confusion Matrix</i> Data Independen SMOTE-2 Pada <i>Kernel Polynomial</i> .	63
38. Hasil Pengujian Data Independen SMOTE-2 Pada <i>Kernel Polynomial</i>	63
39. <i>Confusion Matrix</i> Data Independen SMOTE-2 Pada <i>Kernel RBF</i>	64
40. Hasil Pengujian Data Independen SMOTE-2 Pada <i>Kernel RBF</i>	64
41. <i>Confusion Matrix</i> Data Independen SMOTE-2 Pada <i>Kernel Sigmoid</i>	65
42. Hasil Pengujian Data Independen SMOTE-2 Pada <i>Kernel Sigmoid</i>	65
43. Ringkasan Hasil Pengujian Klasifikasi Data Independen Setelah SMOTE	66
44. Perbandingan Pengujian Data Latih Dengan Penelitian Sebelumnya.....	70
45. Perbandingan Pengujian Data Uji Dengan Penelitian Sebelumnya ..	71
46. Perbandingan Pengujian Data Independen Dengan Penelitian Sebelumnya ..	71

DAFTAR GAMBAR

Gambar	Halaman
1. Ilustrasi Sintesis Protein (Cristianini & Hahn, 2006).....	11
2. Mekanisme PTM <i>Reversible</i> dan <i>Irreversible</i> (Wang et al., 2014).	12
3. Struktur Dasar <i>Artificial Neural Network</i> (Shanmuganathan, 2016).	18
4. Klasifikasi Dua Kelas Menggunakan <i>Hyperplane</i> (Terzic et al., 2013). ..	19
5. Skema LOOCV (James et al., 2000).....	28
6. Ilustrasi <i>5-Fold Cross Validation</i> (James et al., 2000).....	28
7. Alur Pengerjaan Penelitian.....	35
8. Perbandingan Jumlah Data Latih Sebelum dan Setelah SMOTE.	54
9. Perubahan Jumlah Data Independen Sebelum dan Setelah SMOTE.	57
10. Grafik Hasil Pengujian Data Latih Sebelum dan Setelah SMOTE.....	67
11. Grafik Hasil Pengujian Data Uji Sebelum dan Setelah SMOTE.	68
12. Grafik Hasil Pengujian Data Independen Sebelum dan Setelah SMOTE.	69

DAFTAR PSEUDOCODE

<i>Pseudocode</i>	Halaman
1. Algoritme SMOTE (Chawla et al., 2002).....	16
2. Implementasi <i>Kernel Linear SVM</i> di R	20
3. Implementasi <i>Kernel Polynomial</i> di R.....	21
4. Implementasi <i>Kernel RBF</i> di R.....	22
5. Implementasi <i>Kernel Sigmoid</i> di R.....	22
6. Kode Ekstraksi Fitur CTD.	24
7. Kode Ekstraksi Fitur AAindex.....	25
8. Kode Ekstraksi Fitur <i>Hydrophobicity</i>	25
9. Kode Ekstraksi Fitur PseAAC.	26
10. Kode Ekstraksi Fitur QSO.	26
11. Kode Program <i>Import</i> Data ke RStudio.....	39
12. Kode Program Ekstraksi Fitur CTD.....	39
13. Kode Program Ekstraksi Fitur AAindex.....	40
14. Kode Program Ekstraksi Fitur <i>Hydrophobicity</i>	40
15. Kode Program Ekstraksi Fitur PseAAC.....	41
16. Kode Program Ekstraksi Fitur QSO.....	42
17. Kode Program Gabungan Data Fitur Ekstraksi.....	43
18. Kode Program Pelabelan <i>Class</i> Data.	44
19. Kode Program Import Data Setelah Tahap Ekstraksi Fitur.	44
20. Kode Program Penerapan SMOTE.	45
21. Pembagian Data Dengan Konsep 10-Fold <i>Cross-Validation</i>	46
22. Kode Program Pemodelan Klasifikasi Data Protein dengan SVM.....	46
23. Kode Program Prediksi Data Protein Menggunakan SVM.....	47

I. PENDAHULUAN

1.1 Latar Belakang

Teknologi *high-throughput* dalam komputasi biomolekuler telah mengalami evolusi dalam pengkajian genom (Lloyd, 2000), molekul RNA (transkriptomik) (Velculescu et al., 1997), dan struktur fungsi protein (proteomik) (Anderson & Anderson, 1998). Protein adalah molekul besar yang terbentuk dari molekul asam amino yang dihasilkan dari sintesis protein. Urutan asam amino menentukan fungsi reaksi dan pengikatan molekul pada protein (Cristianini & Hahn, 2006). Protein mengalami banyak jenis modifikasi kovalen, baik selama atau setelah proses translasi yang menghasilkan diversifikasi fungsi protein (Q. Li & Shah, 2017).

Modifikasi pasca-translasi atau lebih dikenal *post-translational modifications* (PTM) adalah kontributor yang berperan penting dalam fungsi protein baik mengatur aktivitas, lokalisasi, atau interaksi protein. Berdasarkan peran PTM tersebut adanya gangguan pada PTM dapat menyebabkan penyakit (Hornbeck et al., 2015). Salah satu yang dapat mempengaruhi proses PTM adalah urutan asam amino dalam struktur protein. Urutan asam amino berpengaruh pada proses identifikasi dan penggambaran proses fosforilasi, asetilasi, glikosilasi, metilasi, ubiquitinasi, nitrosilasi, dan lipidasi (Qiu et al., 2016). Salah satu jenis PTM yang umum terjadi adalah metilasi. Metilasi berperan dalam pembungkaman gen di daerah nonkode dari genom seperti heterokromatin secara transkripsi dan memetilasi secara ekstensif. Daerah yang termetilasi membantu melindungi genom dari virus dan mencegah berintegrasi ke dalam genom inang. Metilasi terjadi pada gugus protein Arginin, Lisin, Histidin,

Prolin, dan Karboksil (Q. Li & Shah, 2017), namun umumnya sering terjadi pada protein Lisin dan Arginin (Guo et al., 2014; Paik et al., 2007).

Arginin adalah asam amino bermuatan positif yang memisahkan ikatan hidrogen dengan interaksi amino aromatik. Nitrogen Arginin dalam polipeptida akan mengandung gugus metil setelah dimodifikasi saat translasi, sehingga disebut sebagai metilasi Arginin. Metilasi Arginin menghasilkan guanidina (basa organik) yang mengikat gugus metil sebanyak satu atau dua pada atom nitrogen Arginin. Selanjutnya, nitrogen Arginin dikatalisis oleh enzim metiltransferase melalui atom nitrogen, karbon, sulfur, serta oksigen dari molekul kecil, lipid, protein, dan asam nukleat (Gary & Clarke, 1998). Sebelum tahun 1999, metilasi Arginin terbukti mempengaruhi transduksi sinyal (Aletta et al., 1998) dan mengikat RNA dalam sitoplasma (Shen et al., 1998).

Dalam beberapa dekade terakhir, banyak studi telah melakukan identifikasi terkait metilasi Arginin. Satu dari penelitian yang ada ialah penelitian yang berjudul “*PRmePRed: A Protein Arginine Methylation Prediction Tool*” yang dilakukan oleh Kumar et al. (2017). Kumar et al mengidentifikasi *screening* cepat serta kemungkinan metilasi pada proteome dengan membandingkan analisis data dan penilaian fitur pada alat prediksi metilasi Arginin. Dalam penelitiannya, Kumar et al melakukan analisis menggunakan metode klasifikasi *Support Vector Machine*. Data protein Arginin yang dipakai dalam penelitian Kumar et al. (2017) diperoleh dari basis data UniProt (*release 2015_06*) yang membagi data menjadi tiga bagian yaitu: data uji, data latih, dan data independen. Hasil penelitian menunjukkan tingkat akurasi sebesar 84,10% untuk data uji, 90% untuk data latih, dan 93% untuk data independen. Hasil sensitivitas sebesar 82,38%, spesifitas 83,77%, dan MCC 66,20%.

Metode *Support Vector Machine* (SVM) yang digunakan oleh Kumar et al adalah metode klasifikasi dalam bidang *machine learning* yang diusulkan oleh Vapnik tahun 1995 (Cortes & Vapnik, 1995). Metode SVM digunakan untuk klasifikasi dan regresi data (Shavers et al., 2006). Konsep SVM adalah

membangun *hyperplane* dan mengoptimalkan jarak pemisah (*margin*) antara dua kelas data untuk memberikan hasil klasifikasi optimal (Terzic et al., 2013). SVM memberikan hasil yang baik dalam pengenalan pola dengan pendekatan *artificial intelligence* seperti pengenalan karakter tulisan tangan (Cortes & Vapnik, 1995), klasifikasi gambar (Schölkopf et al., 2000), deteksi wajah (Chang & Lin, 2001), pemrosesan sinyal, dan pengenalan ucapan (Terzic et al., 2013).

Penelitian sebelumnya juga telah dilakukan oleh Lumbanraja et al. (2019) berdasarkan penelitian Kumar et al. (2017). Lumbanraja et al menggunakan data protein Arginin yang sama dengan Kumar et al, namun metode klasifikasi *random forest* dan *feature extraction* yang berbeda sebagai banding. Hasil akurasi yang diperoleh untuk data uji sebesar 93,76%, data latih 80,32%, dan data independen 98,08%. Metode *random forest* menunjukkan kinerja yang baik pada data uji dan data independen, namun pada data latih akurasi yang diperoleh sedikit lebih kecil dibandingkan hasil yang didapat oleh Kumar et al dengan metode SVM. Akurasi 98,08% pada data independen dinilai tidak efektif karena banyak menggunakan data kelas negatif daripada kelas positif dalam prediksi metilasi protein.

Pada penelitian Kumar et al. (2017) data latih protein yang digunakan memiliki perbandingan rasio data positif dan negatif sebesar 1:4, sedangkan pada penelitian Lumbanraja et al. (2019) data independen yang digunakan memiliki perbandingan data positif:negatif sebesar 8:92. Pembagian rasio seperti ini memungkinkan terjadinya ketidakseimbangan data (*imbalanced data*). *Imbalanced data* adalah keadaan yang menunjukkan jumlah data kelas minoritas lebih sedikit dibandingkan jumlah kelas mayoritas. Akibatnya, kelas minoritas memiliki hasil akurasi prediksi yang lebih rendah (Noorhalim et al., 2019), dan membuat performa metode klasifikasi yang digunakan menjadi buruk. Dalam kasus seperti ini, penggunaan metode klasifikasi saja tidak cukup untuk memperbaiki hasil klasifikasi.

Untuk itu diperlukan penggunaan algoritme lain untuk mengatasi permasalahan klasifikasi pada data tidak seimbang. Metode *Synthetic Minority Oversampling Technique* (SMOTE) adalah metode yang andal mengatasi masalah ketidakseimbangan data (*imbalanced data*). Diperkenalkan oleh Nithes V. Chawla, SMOTE memiliki konsep membuat salinan dari data minoritas yang dikenal dengan data sintetis (*synthetic data*), sehingga memperbaiki performa kinerja metode klasifikasi yang digunakan dan menghasilkan tingkat akurasi yang lebih baik (Chawla et al., 2002). Ketidakseimbangan data metilasi Arginin pada penelitian Kumar et al. (2017) yang berjudul “*PRmePRed: A Protein Arginine Methylation Prediction Tool*” menjadi acuan dalam penelitian ini. Dalam riset ini, data protein Arginin diperoleh dari penelitian Kumar et al, yang dikerjakan menggunakan algoritme SMOTE dan metode klasifikasi SVM. Dengan demikian, penelitian ini berjudul Klasifikasi *Imbalanced Data* Menggunakan Algoritme SMOTE dan Metode *Support Vector Machine* pada Studi Kasus: Metilasi *Sequence* Protein Arginin.

1.2 Rumusan Masalah

Berdasarkan pemaparan latar belakang tersebut, adapun masalah dalam penelitian ini diantaranya:

- a. Ketidakseimbangan penggunaan data metilasi *sequence* protein Arginin pada penelitian Kumar et al. (2017).
- b. Perbandingan kinerja metode *Support Vector Machine* dengan dan tanpa penerapan algoritme SMOTE dalam mengatasi ketidakseimbangan data pada studi kasus metilasi protein Arginin.

Berdasarkan rincian masalah tersebut, rumusan masalah utama adalah memodelkan klasifikasi *imbalanced data* menggunakan algoritme SMOTE & metode *support vector machine* dalam kasus metilasi *sequence* protein Arginin.

1.3 Tujuan Penelitian

Tujuan dari penelitian ini diantaranya:

- a. Mengukur performa algoritme SMOTE dan metode *Support Vector Machine* dalam masalah *imbalanced data* metilasi protein Arginin.
- b. Membandingkan hasil klasifikasi metilasi *sequence* protein Arginin pada penelitian Kumar et al. (2017) dengan judul “*PRmePred: A Protein Arginine Methylation Prediction Tool*” dan penelitian Lumbanraja et al.(2019) dengan judul “Implementasi Metode *Random Forest* Untuk Prediksi Posisi Metilasi Pada Sekuens Protein”.

1.4 Manfaat Penelitian

Manfaat dari penelitian ini adalah mengetahui keakuratan algoritme SMOTE dan metode *support vector machine* dalam mengklasifikasi *imbalanced data* pada studi kasus metilasi protein Arginin.

1.5 Batasan Masalah

Batasan masalah dalam penelitian ini, diantaranya:

- a. Metode klasifikasi yang digunakan berfokus pada metode *Support Vector Machine* dengan algoritme *Synthetic Minority Oversampling Technique* (SMOTE).
- b. *Kernel* yang digunakan pada metode *Support Vector Machine* terdiri dari empat *kernel*, yaitu: Linear, Polynomial, Radial Basis Function (RBF), dan Sigmoid.
- c. Jenis data metilasi protein arginin yang dipakai diperoleh dari riset penelitian Kumar et al. (2017) dengan jumlah 10.912 data.

II. TINJAUAN PUSTAKA

2.1 Penelitian Terkait

Penelitian terkait dalam penelitian ini digunakan sebagai bahan acuan dan perbandingan untuk hasil klasifikasi. Topik penelitian yang menjadi pembanding ialah prediksi metilasi *sequence* protein Arginin dan klasifikasi *imbalanced data*. Secara umum, gambaran mengenai beberapa riset yang digunakan dalam penelitian ini terangkum dalam Tabel 1.

Tabel 1. Penelitian Terkait

No	Penelitian	Data	Metode	Hasil
1	<i>Fast Prediction of Protein Methylation Sites Using a Sequence-based Feature Selection Technique</i> (Wei, Xing, et al., 2017)	Protein Arginin Jumlah: 1.481 Negatif: 1.296 Positif: 185	Metode klasifikasi: <i>Random Forest</i>	Akurasi: 80,7% Sensitivitas: 76,9% Spesifisitas: 86,6% MCC: 61,7%
		Protein Lisin Jumlah: 1.744 Negatif: 1.518 Positif: 225	Fitur Seleksi: <i>Maximal Relevance Maximal-Distance (MRMD) dan Sequential Backward Search (SBS)</i>	Akurasi: 69,9% Sensitivitas: 68,6% Spesifisitas: 71,3% MCC: 39,9%
2	<i>PRmePRed: A Protein Arginine Methylation Prediction Tool</i> (Kumar et al., 2017)	Protein Arginin Data Latih Positif: 1.038 Negatif: 5.190		Akurasi: 90% Sensitivitas: 82,38% Spesifisitas: 83,77% MCC: 66,20%
		Protein Arginine Data Uji Positif: 260 Negatif: 260	Metode klasifikasi: <i>Support Vector Machine</i>	Akurasi: 84,10% Sensitivitas: 82,38% Spesifisitas: 83,77% MCC: 66,20%
		Data Bebas Positif : 1.131 Negatif: 3.033		Akurasi: 93% Sensitivitas: 82,38% Spesifisitas: 83,77% MCC: 66,20%
		Sumber: UniProt Database		

		Protein Arginin	
		Data Latih	
		Positif: 369	Akurasi: 93,76%
		Negatif: 369	
		Data Uji	
		Positif: 162	Akurasi: 80,32%
		Negatif: 153	
		Metode klasifikasi: <i>Random Forest</i>	
		Data Independen	
		Positif: 87	
		Negatif: 1.007	
			Akurasi: 98,08%
		Sumber:	
		UniProt	
		Database	
		Credit Card	
		Fraud	
		Jumlah: 29.976	<i>F-measure:</i> 81%.
		Positif: 23.347	<i>G-mean:</i> 81,8%.
		Negatif: 6.629	Akurasi: 81%.
			Sensitivitas: 79,9%.
		Sumber: <i>UCI Repository</i>	Spesifisitas: 83,6%.
3	Implementasi Metode <i>Random Forest</i> Untuk Prediksi Posisi Metilasi Pada Sekuens Protein Arginin (Lumbanraja et al., 2019)		
4	Klasifikasi Data Tidak Seimbang Menggunakan Algoritme SMOTE dan <i>k-Nearest Neighbor</i> (Siringoringo, 2018)		

Berikut adalah resume penelitian pada Tabel 1.

2.2.1 *Fast Prediction of Protein Methylation Sites Using a Sequence-based Feature Selection Technique (2017)*

Riset ini dikerjakan oleh Wei et al. (2017) yang mengukur performa prediksi situs metilasi protein Arginin dan lisin menggunakan teknik fitur seleksi. Data protein yang digunakan diperoleh dari *Benchmark Dataset* (http://www.jcibioinfo.cn/PMe/PMessi_data.html). Data protein Arginin terdiri dari 1.481 yang terbagi menjadi 185 data positif dan 1.296 data negatif. Selanjutnya, data protein lisin terdiri dari 226 data positif dan 1.518 data negatif dengan jumlah 1.744 data. Data protein Arginin maupun lisin memiliki jumlah data negatif yang lebih banyak daripada data positif. Oleh karena itu, data positif dan negatif dibagi ke dalam perbandingan 1:7.

Dalam alur penelitian, data protein dilakukan *preprocessing* dengan memotong menjadi 11 *windows*. Selanjutnya, data diekstraksi menggunakan enam fitur, diantaranya: *Information Theory Features* (ITF), *Overlapping Property Features* (OPF), *Twenty-Bit Features*

(TBF), *Twenty-One-Bit Features* (TOBF), *Skip Dipeptide Composition Features* (SDCF), dan *Conjoint Triad Features* (CTF). Data protein yang telah diekstraksi selanjutnya dilakukan pemeringkatan dengan metode *Maximal-Relevance-Maximal-Distance* (MRMD) dan pemilihan fitur optimal dengan metode *Sequential Backward Search* (SBS). Fitur yang terpilih digunakan sebagai pemodelan klasifikasi dengan menggunakan metode *random forest*. Hasil klasifikasi data protein Arginin diperoleh akurasi sebesar 80,7%, sensitivitas 76,9%, spesifisitas 86,6%, dan MCC 61,7%. Protein lisin memberikan hasil klasifikasi lebih rendah dengan akurasi 69,9%, sensitivitas 68,6%, spesifisitas 71,3%, dan MCC 39,9%.

2.2.2 PRmePRed: A Protein Arginine Methylation Prediction Tool (2017)

Penelitian pada tahun 2017 ini, dikerjakan oleh Kumar et al. (2017) dengan melakukan identifikasi *screening* cepat dan kemungkinan situs metilasi pada proteome. Kumar et al membandingkan analisis data dan penilaian fitur pada alat prediksi metilasi Arginin. Dalam penelitiannya, data protein diperoleh dari *database* Uniprot (*release 2015_06*), dengan 6754 situs metilasi yang diekstrak dari 2077 *sequence* protein. Selanjutnya, data dilakukan pemotongan dengan panjang *window* yang beragam (7, 11, 15, 19, 23, 27, 31, dan 35). Dalam penggerjaannya, *window* yang dipakai 19, 23, 27, 31, dan 35 dikelompokkan menjadi tiga data percobaan berbeda. Percobaan menggunakan data uji, data latih, dan data independen.

Adapun *feature extraction* yang digunakan berjumlah enam, diantaranya: *Atchley Factors*, *AA Frequency*, *ASA*, *Disorder*, *Hydrophobicity*, dan *Van der Waal's Volume*. Metode klasifikasi yang dipakai adalah metode *support vector machine* yang dilakukan pada empat *Kernel* , yaitu: linear, *Polynomial*, *Radial Basis Function* (RBF), dan *Sigmoid*. Untuk mengukur kinerja performa metode klasifikasi, Kumar et al. menggunakan evaluasi matriks. Parameter evaluasi matriks yang dipakai antara lain: akurasi, sensitivitas, spesifisitas, dan *Matthew Correlation*

Coefficient (MCC). Hasil penelitian menunjukkan *window* dengan panjang 19 memberikan hasil prediksi terbaik dengan akurasi pada data latih dan data independen sebesar 90% dan 93% berturut-turut, sedangkan data uji sebesar 84,10%.

2.2.3 Implementasi Metode *Random Forest* Untuk Prediksi Posisi Metilasi Pada Sekuens Protein Arginin (2019)

Penelitian selanjutnya berjudul “Implementasi Metode *Random Forest* Untuk Prediksi Posisi Metilasi Pada Sekuens Protein Arginin” dikerjakan oleh Lumbanraja et al. (2019). Penelitian ini mengacu pada penelitian Kumar et al. (2017) dalam melakukan prediksi metilasi pada *sequence* protein Arginin. Dalam penggerjaannya Lumbanraja et al. menggunakan data *sequence* protein Arginin yang sama (*database* Uniprot (*release* 2015_06)), dengan implementasi metode *random forest* sebagai pembanding.

Dalam penelitian ini, data protein dilakukan proses *preprocessing* dengan menghapus data asam amino yang tidak terdeteksi dan redundansi data. Selanjutnya, data dibagi menjadi tiga kelompok, yaitu data uji, data latih, dan data independen. Kemudian data dilakukan *feature extraction* dengan menggunakan empat fitur, yaitu: QSO, CTD, PseAAC, dan AAindex. Hasilnya metode *random forest* menunjukkan performa yang baik pada data independen dan data latih dengan akurasi sebesar 98,08% dan 93,76%. Pada data uji didapatkan hasil akurasi yang lebih rendah dari penelitian Kumar et al. sebesar 80,32%.

2.2.4 Klasifikasi Data Tidak Seimbang Menggunakan Algoritme SMOTE dan *k-Nearest Neighbor* (2018)

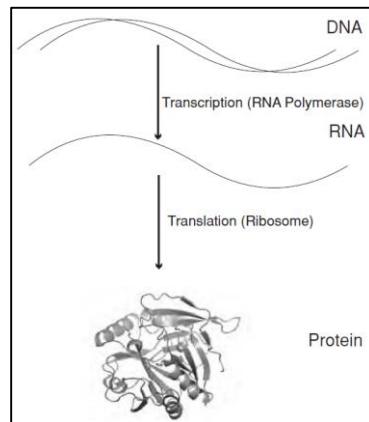
Riset mengenai *imbalanced data* ini dikerjakan oleh Siringoringo (2018) yang bertujuan meningkatkan hasil klasifikasi pada data kartu kredit. *Dataset* yang digunakan adalah *Credit Card Fraud* yang bersumber dari UCI Repository. *Dataset* berjumlah 29.976 data, terdiri dari 23.347 data positif (kelas mayoritas) dan 6.629 data negatif (kelas minoritas). Teknik

klasifikasi yang digunakan adalah SMOTE dan *k-nearest neighbor*. Selanjutnya, data dibagi menjadi *data training* dan *data testing* dalam 10 bagian dengan *10-fold cross validation*.

Data latih diproses menggunakan algoritme SMOTE sebanyak dua kali dengan nilai $k=5$ dan membuat data sintetik pada data negatif (kelas minoritas). Setelah mendapatkan data sintetis, dilakukan klasifikasi menggunakan *k-nearest neighbor* dengan ketentuan nilai $k=1,2,3,5,7$, dan 9. Untuk mengukur kinerja klasifikasi, Siringoringo (2018) menggunakan *F-measure* dan *G-Mean*. *F-measure* digunakan untuk mengukur klasifikasi kelas minoritas pada data tidak seimbang, dan *G-Mean* mengukur performa klasifikasi secara keseluruhan. Hasil penelitian diperoleh *F-measure* sebesar 81% dan *G-mean* sebesar 81,8% dengan rata-rata akurasi, sensitivitas, dan spesifisitas adalah 81%, 79,9%, dan 83,6% berturut-turut.

2.2 Protein

Protein adalah molekul besar (makromolekul) yang terbentuk dari molekul sederhana yang disebut asam amino (Cristianini & Hahn, 2006). Asam amino adalah senyawa yang memiliki gugus karboksi dan gugus amino yang terikat oleh atom karbon (Buxbaum, 2007). Asam amino dibentuk melalui proses yang disebut sintesis protein. Sintesis protein secara konseptual terbagi ke dalam dua tahap: transkripsi (DNA→RNA) dan translasi (RNA→Protein). Transkripsi adalah penyalinan kode DNA menjadi RNA untuk menghasilkan urutan asam amino kolinear. Translasi adalah penerjemahan (*translate*) RNA menjadi asam amino yang kemudian dikirim ke ribosom (Cristianini & Hahn, 2006). Ilustrasi sintesis protein digambarkan pada Gambar 1. Hasil sintesis protein membentuk 20 jenis asam amino berbeda yang disajikan pada Tabel 2.



Gambar 1. Ilustrasi Sintesis Protein (Cristianini & Hahn, 2006).

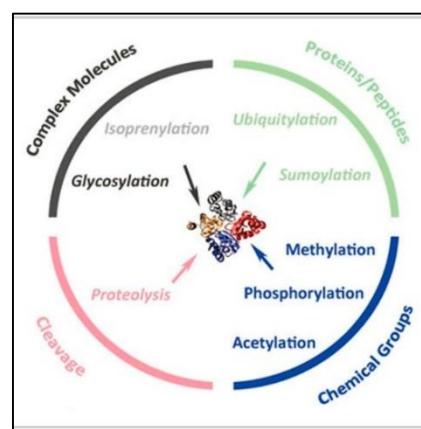
Tabel 2. Jenis dan Standar Simbol Asam Amino (Cristianini & Hahn, 2006)

Simbol	Asam Amino	Simbol	Asam Amino
A	Alanine	L	Leucine
R	Arginine	K	Lysine
N	Asparagine	M	Methionine
D	Aspartic acid	F	Phenylalanine
C	Cysteine	P	Proline
Q	Glutamine	S	Serine
E	Glutamine acid	T	Threonine
G	Glycine	W	Tryptophan
H	Histidine	Y	Tyrosine
I	Isoleucine	V	Valine

Setiap protein terdiri dari 200-300 asam amino, namun tak jarang ditemukan protein dengan kandungan asam amino yang sedikit, berkisar 30-40 asam amino. Setiap urutan asam amino menentukan bentuk protein, fungsi reaksi, dan pengikatan molekul pada protein. Sejumlah protein yang disebut degradase digunakan untuk memotong molekul yang tidak dibutuhkan sel, dan sejumlah lain disebut ligase berperan dalam menggabungkan molekul (Cristianini & Hahn, 2006). Protein mengalami banyak jenis modifikasi kovalen, baik selama atau sesudah proses translasi yang menghasilkan diversifikasi fungsi protein. Modifikasi ini dikatalisis oleh enzim tertentu yang bertanggung jawab atas kompleksitas dan karakteristik fungsi biologis makhluk hidup (Q. Li & Shah, 2017).

2.3 Post-Translational Modifications

Post-Translational Modifications (PTM) adalah perubahan rantai asam amino setelah proses biosintesis (Q. Li & Shah, 2017), yang meningkatkan struktur biokimia sel dalam protein (Vuzman et al., 2012). Modifikasi pasca-translasi mempengaruhi proteome secara fungsional maupun struktural sehingga berdampak pada proses sel (Paik & Kim, 1967). PTM berkontribusi dalam aktivitas sel, lokalisasi, interaksi antar protein (Hornbeck et al., 2015), transduksi sinyal, serta ekspresi gen (Q. Li & Shah, 2017). PTM juga mempengaruhi perubahan sifat protein melalui reaksi biokimia yang terjadi secara *reversible* atau *irreversible* (Wang et al., 2014) seperti pada Gambar 2.



Gambar 2. Mekanisme PTM *Reversible* dan *Irreversible* (Wang et al., 2014).

PTM sudah menjadi riset dalam pengembangan identifikasi situs modifikasi dalam bidang bioinformatika. Pada komputerisasi modern, PTM meningkatkan fungsi proteome dalam mengelola aspek biologi sel berdasarkan ikatan kovalen dari gugus fungsi. Hal ini membuat PTM dapat mengidentifikasi dan menggambarkan proses fosforilasi, asetilasi, glikosilasi, asetilasi, ubiquitinasi, *sumoylation*, dan oksidasi protein (Qiu et al., 2016). PTM yang mengalami gangguan dapat memunculkan jenis penyakit (Hornbeck et al., 2015) seperti alzheimer dan parkinson (Didonna & Benetti, 2016).

2.4 Metilasi

Metilasi adalah jenis modifikasi pasca-translasi yang menambahkan gugus metil ke protein (Schubert et al., 2006). Metilasi banyak mempengaruhi proses biologis maupun fisiologis sel seperti regulasi transkripsi dan epigenetik, metabolisme sel, dan penyakit manusia (Paik et al., 2007; Yu et al., 2012; C. Zhang et al., 2014). Metilasi berperan dalam pembungkaman gen di daerah nonkode seperti heterokromatin secara transkripsi dan memetilasi secara ekstensif. Daerah yang termetilasi membantu melindungi genom dari virus dan mencegah berinteraksi ke dalam genom inang (Q. Li & Shah, 2017). Metilasi terjadi di beberapa tipe asam amino seperti: Lisin (K), Arginin (R), Prolin (P), Histidin (H), Alanin (A), dan Asparagin (N) (Bannister & Kouzarides, 2005; Bedford & Richard, 2005; Lee et al., 2005). Protein yang umum termetilasi adalah Lisin (K) dan Arginin (R) (Guo et al., 2014; Paik et al., 2007).

Arginin merupakan asam amino bermuatan positif yang memisahkan ikatan hidrogen dengan interaksi amino aromatik. Dalam polipeptida, nitrogen Arginin akan mengandung gugus metil ketika dimodifikasi saat translasi sehingga disebut metilasi Arginin. Metilasi Arginin merupakan modifikasi pasca-translasi *reversible* penting dalam sel. Metilasi Arginin menghasilkan guanidina (basa organik) yang mengikat satu atau dua gugus metil pada atom nitrogen Arginin. Selanjutnya, nitrogen Arginin dikatalisis oleh enzim metiltransferase melalui atom nitrogen, karbon, sulfur, dan oksigen dari molekul kecil, lipid, protein, serta asam nukleat (Gary & Clarke, 1998). Sebelum tahun 1999, metilasi Arginin terbukti mempengaruhi transduksi sinyal (Aletta et al., 1998), serta mengikat RNA dalam sitoplasma (Shen et al., 1998).

2.5 *Imbalanced Data*

Imbalanced data adalah keadaan tidak seimbang kelas data tertentu yang memiliki jumlah data lebih banyak atau sedikit daripada kelas data lain. Ketidakseimbangan data dominan terjadi pada dua kelas yang umumnya terdiri atas kelas negatif dan kelas positif. Kelas yang memiliki jumlah data lebih

banyak disebut kelas mayoritas dan yang memiliki jumlah data lebih sedikit disebut kelas minoritas (Faisal, 2016). Dalam *machine learning* dan *data mining*, ketidakseimbangan data menjadi masalah krusial, seperti masalah diagnosis medis (Kothandani, 2015), klasifikasi teks (Cardie, 1997), dan deteksi tumpahan minyak pada data citra satelit (Kubat et al., 1998).

Salah satu kesalahan klasifikasi pada uji sampel adalah mengklasifikasikan data yang seharusnya termasuk dalam kelas minoritas, menjadi data kelas mayoritas. Akibatnya, kelas minoritas memiliki hasil akurasi prediksi yang rendah (Noorhalim et al., 2019) dan membuat kinerja metode klasifikasi yang digunakan menjadi buruk. Menurut Sun et al., (2009) metode klasifikasi seperti *support vector machine*, *k-nearest neighbor*, *naïve bayes*, dan *backpropagation neural network* menunjukkan hasil yang tidak memuaskan dalam masalah *imbalanced data*.

Menurut Faisal (2016) penanganan masalah *imbalanced data* ada 2 cara, yaitu:

- a. Pendekatan data dengan menyeimbangkan jumlah data kedua kelas, yang terbagi dalam 3 cara:
 - 1) *Undersampling*: mengurangi kelas mayoritas agar jumlah data sebanding dengan kelas minoritas.
 - 2) *Oversampling*: menambah kelas minoritas agar seimbang dengan data kelas mayoritas.
 - 3) Gabungan *undersampling* dan *oversampling*.
- b. Pendekatan algoritme dengan membuat algoritme baru atau memodifikasi algoritme yang ada untuk menyelesaikan masalah data tidak seimbang. Pendekatan algoritme antara lain:
 - 1) *Bagging* adalah membuat beberapa model algoritme yang sama dari sub sampel yang berbeda dari *data training*.
 - 2) *Boosting* yaitu membangun model dari beberapa algoritme yang sama untuk memperbaiki kesalahan prediksi model sebelumnya.
 - 3) *Stack* merupakan pengembangan algoritme dari beberapa jenis metode klasifikasi untuk mendapatkan prediksi yang terbaik.

2.6 Redundansi Data

Redundansi data merupakan kemunculan data yang sama secara berulang yang dapat mengakibatkan terjadinya *inconsistency data*. Ketidakkonsistenan data terjadi akibat kesalahan dalam *input* atau *update* data yang dapat berdampak pada hasil pengolahan data yang tidak sesuai dengan fakta (Pamungkas, 2017). *Tools* dalam melakukan redundansi data protein diantaranya adalah CD-HIT yang dikembangkan oleh W. Li & Godzik (2006). CD-HIT menerapkan konsep algoritme *greedy incremental* dalam mengklaster data protein berdasarkan masukkan ambang batas (presentase kemiripan sekuens). Secara singkat, CD-HIT akan membandingkan data protein antara satu sekuens dengan sekuens lainnya hingga mendapat persentase kemiripan antar sekuens. Sekuens yang memiliki persentase kemiripan di atas ambang batas yang ditentukan akan dihapus. Kemiripan antar sekuens dilihat dari setiap urutan protein dalam sekuens (W. Li & Godzik, 2006)

2.7 Synthetic Minority Oversampling Technique

Synthetic Minority Oversampling Technique (SMOTE) merupakan salah satu pendekatan untuk menyelesaikan masalah ketidakseimbangan data. Metode ini pertama kali diusulkan oleh Chawla et al., (2002) dengan tujuan untuk meningkatkan beberapa data kelas minoritas dengan metode interpolasi (Amari, 1987). Chawla et al melakukan pendekatan dengan menambahkan data lebih pada kelas minoritas atau dikenal dengan membuat data sintetis dengan pengambilan sampel secara acak (Chawla et al., 2002). *Data training* yang diklasifikasikan dengan tepat dapat meningkatkan hasil, sehingga dapat meminimalisir kesalahan *overfitting* (Amari, 1987).

SMOTE dapat diuji pada berbagai *dataset* dengan tingkat ketidakseimbangan dan jumlah data yang berbeda pada *data training* (Maimon & Rokach, 2010). Metode SMOTE telah digunakan untuk mendeteksi jaringan (Cieslak et al., 2006), prediksi pembagian spesies (Blagus & Lusa, 2013), atau untuk mendeteksi kanker payudara (Fallahi & Jafari, 2011). Dalam bidang bioinformatika, metode SMOTE juga diaplikasikan dalam prediksi gen *miRNA*

(Batuwita & Palade, 2009; J. Xiao et al., 2011), serta identifikasi spesifisitas ikatan protein (MacIsaac et al., 2006). Chawla et al. membuat konsep algoritme SMOTE dengan memanfaatkan algoritme *artificial intelligence k-nearest neighbor* seperti yang ditunjukkan pada Pseudocode 1.

```
Algorithm SMOTE(T, N, k)
Input: Number of minority class samples T; Amount of SMOTE
N%; Number of nearest neighbors k
Output: (N/100) * T synthetic minority class samples
(*If N is less than 100%, randomize the minority class samples
as only a random percent of them will be SMOTEd.*)
if N < 100
    then Randomize the T minority class samples
        T = (N/100) * T
        N = 100
    endif
    N = (int)(N/100) (*The amount of SMOTE is assumed to be in
integral multiples of 100.*)
    k = Number of nearest neighbors
    numattrs = Number of attributes
    Sample[ ][ ]: array for original minority class samples
    newindex: keeps a count of number of synthetic samples
        generated, initialized to 0
    Synthetic[ ][ ]: array for synthetic samples
    (*Compute k nearest neighbors for each minority class sample only*)
    for i ← 1 to T
        Compute k nearest neighbors for i, and save the indices in
        the nnarray
        Populate(N, i, nnarray)
    endfor
    Populate(N, i, nnarray) (*Function to generate the synthetic
samples.*)
    while N ≠ 0
        Choose a random number between 1 and k, call it nn. This
        step chooses one of the k nearest neighbors of i.
        for attr ← 1 to numattrs
            Compute: dif = Sample[nnarray[nn]][attr]-Sample[i][attr]
            Compute: gap = random number between 0 and 1
            Synthetic[newindex][attr] = Sample[i][attr] + gap * dif
        endfor
        newindex++
        N = N - 1
    endwhile
    return (*End of Populate.*)
```

Pseudocode 1. Algoritma SMOTE (Chawla et al., 2002).

Berikut adalah penjelasan dari Pseudocode 1. Diberikan contoh data tidak seimbang seperti pada Tabel 3 berikut.

Tabel 3. Contoh Data Tidak Seimbang

Tipe	Atribut#1	Atribut#2	Atribut#3
1	14,23	1,71	2,43
1	13,2	1,78	2,14
2	12,37	94	1,36
2	12,33	1,1	2,28
2	12,64	1,36	2,02
2	12,37	1,13	2,16
2	12,17	1,45	2,53
2	12,37	1,21	2,56

Konsep awal algoritme SMOTE adalah mengidentifikasi kelas minoritas dan kelas mayoritas berdasarkan jumlah tipe pada data. Setelah kelas minoritas teridentifikasi (tipe 1), selanjutnya SMOTE akan menghitung data sintetik yang akan terbentuk berdasarkan pada rumus $T_y = (N/100) * T_x$. Variabel T_x merupakan jumlah data kelas minoritas, T_y adalah kelas minoritas akhir (setelah ditambah data sintetis), dan N adalah persentasi *oversampling* dengan nilai kelipatan 100. Sebagai contoh, persentasi *oversampling* yang diberikan adalah 300, sehingga jumlah kelas minoritas akan dinaikkan setara dengan kelas mayoritas yakni 6 dengan perhitungan $T_y = (300/100) * 2$.

Selanjutnya, komputer akan menghitung jumlah k terdekat (nilai k berdasarkan masukkan) dari setiap data kelas minoritas sampai jumlah T_y kali dan menyimpan data dalam *array*. Kemudian, komputer akan memilih secara acak data minoritas, dan menentukan data minoritas terdekat lainnya (sejumlah k) yang dihitung dengan persamaan jarak *euclidean*. Dari jumlah tetangga terdekat yang terdeteksi, maka algoritme SMOTE memilih acak data minoritas tertentu dan akan membentuk data baru (data sintetis) antara dua data tersebut. Berikut, langkah-langkah tersebut diulang sejumlah N kali hingga data minoritas dan mayoritas menjadi seimbang.

2.8 Metode Klasifikasi

Menurut KBBI, klasifikasi adalah penyusunan sistematis dalam suatu kelompok atau golongan menurut kaidah atau standar yang ditetapkan. Tujuan klasifikasi adalah mendapatkan model dari *training set* yang membedakan atribut ke dalam kategori atau kelas yang cocok, selanjutnya digunakan untuk

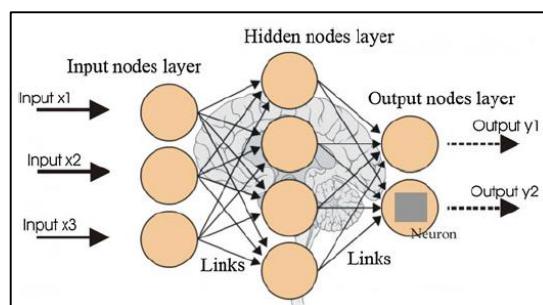
klasifikasi atribut yang kelasnya belum diketahui. Beberapa metode klasifikasi dalam konsep *machine learning* adalah sebagai berikut:

2.8.1 Artificial Neural Network

Artificial Neural Network (ANN) atau Jaringan Syaraf Tiruan (JST) merupakan model komputasi yang terdiri dari elemen pemrosesan yang disebut neuron, dan koneksi yang mengikat koefisien (bobot). Koneksi pada struktur ini adalah pelatihan dan algoritme pengingatan antar neuron, sehingga algoritme ANN disebut juga model koneksiis. Dalam *machine learning*, *neural network* adalah algoritme pembelajaran statistik yang didasarkan pada jaringan saraf biologi (khususnya otak) yang digunakan untuk memperkirakan fungsi tergantung jumlah masukan (*input*). Menurut Shanmuganathan (2016), ANN terdiri dari empat komponen utama, yaitu:

- a. *Node* sebagai unit yang menerima sinyal masuk (*input*);
- b. Koneksi antar *node*;
- c. *Rule* atau aktivitas yang mengubah *node* (*input* menjadi *output*);
- d. Pembelajaran yang mengelola bobot pasangan *input-output*.

ANN digambarkan sebagai kumpulan teknik matematika yang digunakan untuk pemrosesan sinyal, peramalan, dan pengelompokan sehingga disebut sebagai teknik regresi paralel (Shanmuganathan, 2016). Struktur dasar ANN berupa garis, atau bidang datar yang dimodelkan melalui set data untuk menentukan hubungan antara *input* dan *output*, atau identifikasi representasi data pada skala kecil. Struktur dasar ANN diilustrasikan pada Gambar 3.



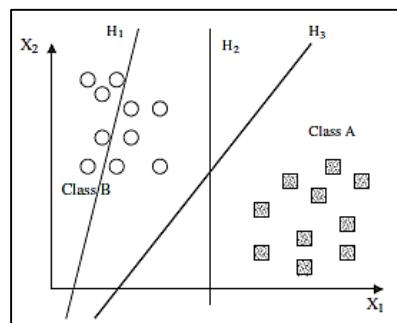
Gambar 3. Struktur Dasar *Artificial Neural Network* (Shanmuganathan, 2016).

2.8.2. Random Forest

Random Forest merupakan metode klasifikasi yang diperkenalkan oleh Breiman (2001) yang mempresentasikan metaanalisis klasifikasi tingkat objek. RF membuat set pengklasifikasi berbasis pohon, yang dilatih oleh himpunan acak dari objek pelatihan. *Random forest* merupakan model prediksi yang berkorespondensi dengan label kelas dan cabang untuk menentukan label kelas. Klasifikasi objek yang tidak diketahui dilakukan dengan pemungutan suara mayoritas atas himpunan pengklasifikasi (Ma et al., 2017). Metode *Random forest* sudah diterapkan dalam bidang komputasi biologi (C. J. Zhang et al., 2016). Penerapan ini diantaranya: prediksi struktur protein (Wei et al., 2015a), prediksi lipatan protein (Wei et al., 2015b), prediksi ikatan DNA-protein (Wei et al., 2017), dan deteksi tubulus (Su et al., 2016).

2.8.3. Support Vector Machine

Support Vector Machine (SVM) adalah metode *supervised learning* yang pertama kali diusulkan oleh Vapnik tahun 1995 (Cortes & Vapnik, 1995). SVM digunakan untuk klasifikasi data dan keperluan regresi yang memiliki keunggulan dapat membagi data linear maupun non-linear secara cepat dalam aplikasi ilmiah dan teknik (Shavers et al., 2006). SVM membuat model sistem untuk memprediksi *output* berdasarkan *data training*. Konsep SVM adalah membuat *hyperplane* (H_2) antara dua kelas data (H_1 dan H_3) dan memaksimalkan jarak *margin* dua kelas untuk memberikan klasifikasi optimal (Terzic et al., 2013). Konsep SVM secara lebih jelas dapat dilihat pada Gambar 4.



Gambar 4. Klasifikasi Dua Kelas Menggunakan *Hyperplane* (Terzic et al., 2013).

SVM memberikan hasil yang baik dalam pengenalan pola seperti pengenalan karakter tulisan tangan (Cortes & Vapnik, 1995), klasifikasi gambar (Schölkopf et al., 2000), deteksi wajah (Chang & Lin, 2001), pemrosesan sinyal, dan pengenalan ucapan (Terzic et al., 2013). Dalam metode SVM terdapat istilah *kernel* yang dipakai dalam analisis regresi. *Kernel* adalah salah satu fungsi pendekatan nonparametrik yang memisahkan data secara linear (Schölkopf, 2002). Berikut empat *kernel* yang digunakan dalam penelitian ini.

2.8.3.1 *Kernel* Linear

Kernel linear merupakan fungsi *kernel* yang diterapkan pada dataset yang terbagi secara linier. Data yang dianalisis menggunakan *kernel* linear dipisahkan dengan satu baris (*hyperplane*). *Kernel* linear digunakan untuk representasi data vektor dan *text mining* khususnya klasifikasi teks. Untuk fungsi *kernel* linear dapat dilihat pada Persamaan 1.

Variabel x_i dan x_j merupakan representasi vektor dari *dataset*. Penerapan Persamaan 1 menggunakan bahasa R pada *class* `cv.glmkappa`, di library `kernlab` terdapat seperti Pseudocode 2.

```

setClass("vanillaKernel ", prototype=structure(.Data=
function() {}, kpar=list()), contains=c("Kernel "))
vanilladot <- function( )
{ rval<- function(x, y = NULL) {
  if(!is(x,"vector")) stop("x must be a vector")
  if(!is(y,"vector")&&!is.null(y)) stop("y must be a
vector")
  if (is(x,"vector") && is.null(y)){
    crossprod(x)
  }
  if (is(x,"vector") && is(y,"vector")){
    if (!length(x)==length(y))
      stop("number of dimension must be the same n
both data points")
    crossprod(x,y)
  }
}
return(new("vanillaKernel
", .Data=rval, kpar=list())))

```

Pseudocode 2. Implementasi *Kernel Linear SVM* di R.

2.8.3.2 Kernel Polynomial

Kernel Polynomial digunakan pada *dataset* yang terpisah secara linear, misalnya pada permasalahan normalisasi *data training*. *Kernel Polynomial* dirumuskan pada Persamaan 2.

Simbol γ dan r merupakan parameter untuk memproses eksekusi program, dan d adalah derajat pangkat *Polynomial*. Implementasi *kernel polynomial* di R menggunakan *library kernlab* pada *class polyKernel* terlihat pada Pseudocode 3.

```

setClass("polyKernel
",prototype=structure(.Data=function() {},kpar=list()
),contains=c("Kernel "))
polydot <- function(degree = 1, scale = 1, offset =
1) { rval<- function(x, y = NULL) {
  if(!is(x,"vector")) stop("x must be a vector")
  if(!is(y,"vector")&&!is.null(y)) stop("y must be a
vector")
  if (is(x,"vector") && is.null(y)) {
    (scale*crossprod(x)+offset)^degree }
  if (is(x,"vector") && is(y,"vector")){
    if (!length(x)==length(y))
      stop("number of dimension must be the same on
both data points")
    (scale*crossprod(x,y)+offset)^degree } }
return(new("polyKernel
",.Data=rval,kpar=list(degree=degree,scale=scale,off
set=offset))) }

```

Pseudocode 3. Implementasi *Kernel Polynomial* di R.

2.8.3.3 Kernel Radial Basis Function (RBF)

Kernel RBF memiliki dua parameter yaitu *Cost* (v) dan *Gamma* (γ). Parameter *Cost* (v) mengoptimalkan hasil klasifikasi *data training*, sedangkan *Gamma* (γ) memastikan pengaruh nilai parameter. Fungsi *kernel RBF* ditunjukkan pada Persamaan 3.

Variabel `exp` merupakan basis logaritma persamaan *kernel* RBF. Implementasi Persamaan 3 menggunakan *class rbfdot* ditunjukkan pada Pseudocode 4.

```

rbfdot<- function(sigma=1) {
  rval <- function(x,y=NULL) {
    if(!is(x,"vector")) stop("x must be a vector")
    if(!is(y,"vector")&&!is.null(y)) stop("y must be a
vector")
    if (is(x,"vector") && is.null(y)){
      return(1)
    }
    if (is(x,"vector") && is(y,"vector")){
      if (!length(x)==length(y))
        stop("number of dimension must be the same on
both data points")
      return(exp(sigma*(2*crossprod(x,y) - crossprod(x)
- crossprod(y))))
      # sigma/2 or sigma ?? } }
  return(new("rbfKernel
",.Data=rval,kpar=list(sigma=sigma )) )
  setClass("rbfKernel
",prototype=structure(.Data=function() {},kpar=list()
),contains=c("Kernel "))
}

```

Pseudocode 4. Implementasi *Kernel RBF* di R.

2.8.3.4 Kernel Sigmoid

Kernel sigmoid merupakan *kernel* yang digunakan untuk menyelesaikan permasalahan data non-linear. Persamaan *Kernel Sigmoid* terlihat pada Persamaan 4.

Tanh merupakan basis logaritma pada persamaan *Kernel Sigmoid*. Implementasi *source code* menggunakan *library kernlab* pada *class tanhdot* ditunjukkan pada Pseudocode 5.

```

tanhdot <- function(scale = 1, offset = 1) {
  rval<- function(x, y = NULL)
  {
    if(!is(x,"vector")) stop("x must be a vector")
    if(!is(y,"vector")&&!is.null(y)) stop("y must
be a vector")
    if (is(x,"vector") && is.null(y)){
      tanh(scale*crossprod(x)+offset)}
    if (is(x,"vector") && is(y,"vector")){
      if (!length(x)==length(y))
        stop("number of dimension must be the same
on both data points")
      tanh(scale*crossprod(x,y)+offset)
    }
  }
  return(new("tanhKernel
",.Data=rval,kpar=list(scale=scale,offset=offset)))
  setClass("tanhKernel
",prototype=structure(.Data=function() {},kpar=list()
),contains=c("Kernel "))
}

```

Pseudocode 5. Implementasi *Kernel Sigmoid* di R.

2.9 Feature Extraction

Feature extraction atau ekstraksi fitur adalah teknik untuk mengurangi korelasi antar prediktor dan membuat hasil antar prediktor menjadi lebih kompleks (Kuhn & Johnson, 2013). Metode ini efisien digunakan untuk mengubah data *string* menjadi data numerik. Metode ini mengekstraksi dua puluh asam amino dari serangkaian fitur pelatihan dan penggabungan sejumlah informasi posisi asam amino dalam sekuen. Fitur diekstraksi dengan mempertimbangkan probabilitas kejadian asam amino di berbagai urutan (Bharill et al., 2015). Urutan asam amino yang secara resmi diwakili oleh himpunan $A = \{A, C, D, E, F, G, H, I, K, L, M, N, P, Q, R, S, T, V, W, Y\}$. Oleh karena itu, algoritme untuk klasifikasi sekuen protein adalah pengkodean dalam hal fitur dan penerapan sebagai *input* untuk setiap algoritme pembelajaran klasifikasi (Bandyopadhyay, 2005). Dalam penelitian ini, fitur ekstraksi yang digunakan berjumlah lima yang diantaranya:

2.9.1 Composition, Transition, dan Distribution (CTD)

Composition, *Transition*, dan *Distribution* merupakan fitur ekstraksi yang merepresentasikan sifat fisikokimia asam amino. Menurut Govindan & Nair (2011), CTD bertujuan untuk memperoleh informasi mengenai panjang *sequence* protein. Perhitungan CTD serupa dengan pengukuran normalisasi *Van der Waals Volume*, polaritas, polarisasi, struktur sekunder dan *solvent accessibility area* (ASA) (Govindan & Nair, 2011). *Composition* (C) merupakan pembagian jumlah asam amino sifat tertentu (Ne) dengan jumlah total asam amino keseluruhan (N). Perhitungan *Composition* (C) dapat dilihat pada Persamaan 5.

$$\text{Composition } (C) = \frac{Ne}{N}(5)$$

Transition (T) mengukur perbandingan perubahan sifat asam amino dengan asam amino dari kelas yang berbeda. *Transition* (T) menghitung jumlah peptida yang dikode ($N_{nm} + N_{mn}$) berbanding panjang *sequence* (N) dikurangi satu. Secara matematis, perhitungan *Transition* (T) dirumuskan seperti Persamaan 6.

$$\text{Transition } (T) = \frac{N_{nm} + N_{mn}}{N - 1}(6)$$

Distribution (D) menggambarkan distribusi setiap atribut terhadap panjang rantai *sequence*. *Distribution* (D) mengukur panjang rantai di residu pertama dengan 25%, 50%, 75%, dan 100% dari atribut asam amino tertentu. Dalam melakukan ekstraksi fitur CTD, dapat menggunakan fungsi *featureCTD* pada *package BioSeqClass*. *FeatureCTD* akan mengembalikan nilai matriks dengan jumlah kolom $M + M * \frac{M-1}{2} + M * 5$ yang dikode dalam dimensi vektor numerik.

Kode program untuk ekstraksi CTD dapat dilihat pada Pseudocode 6.

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

Pseudocode 6. Kode Ekstraksi Fitur CTD.

2.9.2 AAindex

AAindex merupakan basis data sekuens protein yang berisi sifat fisikokimia dan biokimia asam amino. Berdasarkan penelitian Kawashima et al. (2008), basis data AAindex terbagi menjadi tiga bagian, yaitu AAindex1, AAindex2, dan AAindex3. AAindex1 berisi 20 nilai numerik asam amino dengan jumlah keseluruhan sebanyak 544 indeks. AAindex2 berisi 94 tambahan matriks asam amino yang terbagi menjadi 67 matriks simetris dan 27 matriks non-simetris. AAindex3 berisi 47 matriks asam amino potensial yang terdiri dari 44 matriks simetris dan 3 matriks non-simetris (Kawashima et al., 2008b). Dalam mengukur sifat fisikokimia dan biokimia pada AAindex, dapat menggunakan fitur *featureAAindex* pada *package BioSeqClass*. *FeatureAAindex* menghitung atribut fisikokimia dan biokimia pada asam amino berdasarkan basis data AAindex dalam bentuk matriks.

Jika parameter yang digunakan berupa *aaindex.name="all"*, maka parameter akan menghitung semua atribut di AAindex. Setiap baris yang dikode mewakili fitur dengan vektor numerik berdimensi $531 * N$. Jika parameter yang digunakan berupa nama atribut dari AAindex (misalnya *aaindex.name="ANDN920101"*), maka fitur akan mewakili

satu baris vektor numerik berdimensi N . N adalah panjang sekvens yang berjumlah ganjil. Penerapan *featureAAindex* dalam bahasa R dapat dilihat pada Pseudocode 7.

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

Pseudocode 7. Kode Ekstraksi Fitur AAindex.

2.9.3 *Hydrophobicity*

Hydrophobicity merupakan sifat protein yang merujuk pada daya tarik protein ke dalam lapisan lipid. Untuk mengukur persentase tersebut, dikenal istilah skala hidrofobisitas. Skala hidrofobisitas merupakan nilai yang menentukan hidrofobisitas atau hidrofilisitas residu asam amino. Semakin positif nilai, maka semakin hidrofobik suatu residu. Jika menunjukkan negatif, maka residu tersebut hidropik. Analisis hidrofobisitas digunakan untuk memahami identifikasi struktur dasar sekunder protein (Kyte & Doolittle, 1982).

Untuk mengekstraksi fitur hidrofobisitas, dapat menggunakan *featureHydro* pada *library BioSeqClass*. Fungsi *featureHydro* akan mengembalikan nilai pengukuran efek hidrofobisitas menggunakan parameter *hydro.method*. Metode yang mendukung pengkodean fitur hidrofobisitas yaitu “kpm” dan “SARAH1”. Dalam ekstraksi, metode “kpm” menggunakan angka, sedangkan metode “SARAH1” menggunakan dimensi dalam pengukuran efek hidrofobik dengan jumlah dimensi N . N merupakan panjang *sequence* berjumlah ganjil. Kode program ekstraksi disajikan pada Pseudocode 8.

```
featureHydro(seq, hydro.method="SARAH1")
```

Pseudocode 8. Kode Ekstraksi Fitur *Hydrophobicity*.

2.9.4 *Pseudo Amino Acid Composition* (PseAAC)

Pseudo Amino Acid Composition (PseAAC) merupakan metode prediksi lokalisasi subseluler dan prediksi tipe membran protein yang diperkenalkan oleh Chou (2001). PseAAC berisi serangkaian informasi lebih dari 20 atribut, dimana 20 atribut pertama merepresentasikan urutan asam amino. Atribut tambahan berikutnya merupakan gabungan informasi urutan komponen asam amino semu (Chou, 2001). Untuk mengekstraksi asam amino menggunakan metode PseAAC dapat menggunakan fungsi *featurePseudoAACComp* pada *library* BioSeqClass. Fungsi *featurePseudoAACComp* mengukur komposisi asam amino yang dikodekan dalam 20+d dimensi. Dimensi 20+d mewakili 20 jenis asam amino dan atribut tambahan asam amino semu. Kode *featurePseAACComp* disajikan pada Pseudocode 9.

```
featurePseudoAACComp (seq, d, w=0.05)
```

Pseudocode 9. Kode Ekstraksi Fitur PseAAC.

2.9.5 *Quasi-Sequence-Order* (QSO)

Quasi-Sequence-Order merupakan urutan asam amino berdasarkan sifat fisikokimia untuk prediksi lokalisasi subseluler protein dengan pendekatan statistik. Metode QSO yang dikerjakan oleh Chou (2000) memiliki nilai perhitungan deskriptor $20+20+(2*nlag)$. Nilai 20 pertama merepresentasikan komposisi asam amino, sedangkan $20+(2*nlag)$ merepresentasikan pengaruh efek QSO. Variabel nlag merupakan parameter deskriptor yang memiliki nilai *default* 30 atau tidak melebihi panjang *sequence* protein. Adapun ω merupakan bobot faktor QSO dengan nilai *default* 0,1 (Chou, 2000). Ekstraksi fitur QSO dilakukan dengan menerapkan fungsi *extractQSO* pada *library* protr. Pseudocode 10 memperlihatkan kode ekstraksi QSO sebagai berikut.

```
extractQSO(x, nlag = 30, w = 0.1)
```

Pseudocode 10. Kode Ekstraksi Fitur QSO.

2.10 *Cross-Validation*

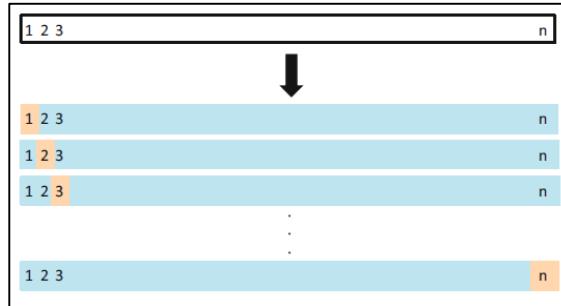
Resampling adalah pengambilan sampel acak berulang dari *data training* untuk menyesuaikan kembali model sampel agar mendapatkan informasi tambahan mengenai model klasifikasi. Pendekatan ini memberi kemungkinan untuk memperoleh informasi yang tidak tersedia dari model sebelumnya yang hanya sekali menggunakan *data testing*. Ada dua metode yang umum digunakan dalam *resampling*, yaitu metode *cross-validation* dan *bootstrap*. Secara garis besar, metode *cross-validation* digunakan untuk mengevaluasi kinerja dan memperkirakan kesalahan pengujian metode *supervised learning* tertentu. Metode *bootstrap* digunakan untuk mengukur akurasi suatu parameter metode *supervised learning*. Pengukuran yang digunakan dalam penelitian ini adalah metode *cross-validation*.

Metode *cross-validation* mengevaluasi dan memperkirakan kesalahan *test error rate* dan *training error rate*. *Test error rate* adalah rata-rata kesalahan pada *data testing*. *Training error rate* merupakan kesalahan pengujian pada *data training*. *Cross-validation* terbagi atas *leave-one-out cross-validation* dan *k-fold cross-validation* (James et al., 2000).

2.10.1 *Leave-One-Out Cross-Validation*

Leave-One-Out Cross-Validation (LOOCV) adalah metode yang digunakan untuk jenis pemodelan prediktif seperti regresi logistik atau analisis diskriminan linear. Konsep LOOCV adalah mengevaluasi kinerja suatu metode dengan memisahkan data menjadi dua seperti pada Gambar 5. Metode ini memiliki keunggulan diantaranya, memberikan hasil bias lebih kecil dan hasil yang sama jika dilakukan perhitungan *data training* maupun *data testing* secara berulang (James et al., 2000). Secara matematis, LOOCV dirumuskan pada Persamaan 7.

Variabel \hat{y}_i adalah nilai ke- i yang digunakan dan h_i adalah nilai tetapan yang dihitung dengan rumus $h_i = \frac{1}{n} + \frac{(x_i - \bar{x})^2}{\sum_{j=1}^n (x_j - \bar{x})^2}$.

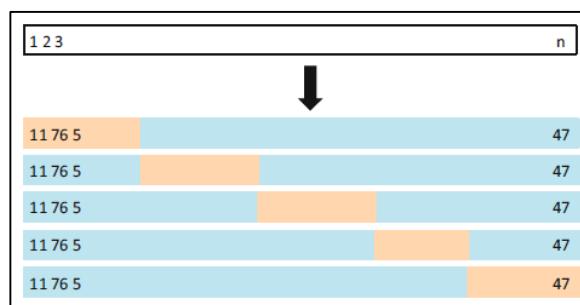


Gambar 5. Skema LOOCV (James et al., 2000).

2.10.2 *K*-Fold Cross-Validation

Pendekatan *K-Fold Cross-Validation* dilakukan dengan cara membagi data secara acak menjadi sejumlah k kelompok atau *fold* dengan ukuran yang sama. *Fold* pertama dijadikan sebagai *data testing*, sedangkan $k - 1$ *fold* lain dijadikan sebagai *data training*. Langkah ini dilakukan berulang sebanyak k kali dan setiap kelompok k dapat menjadi *data testing*. Proses ini menghasilkan perkiraan sebanyak k dari *test error* (James et al., 2000). Perhitungan *K-Fold Cross Validation* dihitung dengan rumus pada Persamaan 8.

Nilai k pada umumnya 5 atau 10 yang memiliki perbedaan pada daya komputasinya. *K-Fold Cross-Validation* memberikan komputasi yang baik dengan melakukan penyesuaian prosedur pembelajaran sebanyak k kali. Hal ini berbeda dengan LOOCV yang melakukan penyesuaian terhadap metode pembelajaran sebanyak k kali (James et al., 2000). Secara visual, konsep *K-Fold Cross Validation* dapat dilihat pada Gambar 6.



Gambar 6. Ilustrasi 5-Fold Cross Validation (James et al., 2000).

2.11 Confusion matrix

Confusion matrix adalah metode untuk menggambarkan performa model klasifikasi. Metode ini berupa matriks ukuran $n \times n$ yang berkaitan dengan *classifier*. *Confusion matrix* menunjukkan hasil prediksi secara aktual, variabel n adalah jumlah kelas yang berbeda. Menurut Kuhn & Johnson (2013) *confusion matrix* memiliki perhitungan matriks evaluasi yang digunakan untuk mengukur performa dari prediktor. Contoh matriks evaluasi untuk dua kelas data dapat dilihat pada Tabel 4.

Tabel 4. *Confusion Matrix* (Kuhn & Johnson, 2013)

<i>Predicted</i>	<i>Actual</i>	
	<i>Non-event</i>	<i>Event</i>
<i>Non-event</i>	TN	FN
<i>Event</i>	FP	TP

Adapun istilah yang digunakan dalam matriks evaluasi, diantaranya:

- a. *True Positive* (TP): data positif yang diklasifikasikan dengan tepat.
- b. *True Negative* (TN): data negatif yang diklasifikasikan dengan tepat.
- c. *False Positive* (FP): data positif yang diklasifikasikan dengan tidak tepat.
- d. *False Negative* (FN): data negatif yang diklasifikasikan tidak tepat.

Ada empat parameter yang digunakan untuk mengukur kinerja model klasifikasi. Parameter tersebut diantaranya: akurasi (ACC), sensitivitas (SE), spesifisitas (SP), dan *Matthew Correlation Coefficient* (MCC). Berikut penjelasan dan rumus perhitungan yang digunakan dalam penelitian ini.

2.11.1 Accuracy (*Error Rate*)

Akurasi adalah pengujian terhadap tingkat kedekatan antara nilai prediksi dengan nilai sebenarnya. Hasil akurasi menentukan performa suatu metode klasifikasi (Bekkar et al., 2013). Persamaan akurasi ditunjukkan pada Persamaan 9.

$$ACC = \frac{TP+TN}{TP+FN+TN+FP} * 100.....(9)$$

2.11.2 Sensitivity (True Positive Rate)

Sensitivitas adalah persentase jumlah informasi positif yang didapat oleh sistem dengan jumlah keseluruhan informasi yang ada (Bekkar et al., 2013). Sensitivitas dihitung dengan Persamaan 10.

2.11.3 Specificity (True Negative Rate)

Spesifisitas merupakan perbandingan informasi proporsi negatif yang didapat terhadap jumlah keseluruhan informasi yang ada baik yang relevan maupun tidak (Bekkar et al., 2013). Perhitungan spesifisitas dilihat pada Persamaan 11.

2.11.4 Matthew Correlation Coefficient (MCC)

Matthew Correlation Coefficient adalah pengukuran terhadap kualitas klasifikasi dengan memperhitungkan nilai positif dan negatif yang bernilai salah dan nilai benar (Bekkar et al., 2013).

$$MCC \equiv \frac{TP * TN - FP * FN}{\sqrt{(TP + FN)(TP + FP)(TN + FN)(TN + FP)}} \quad (12)$$

III. METODOLOGI PENELITIAN

3.1 Tempat dan Waktu Penelitian

3.1.1 Tempat Penelitian

Penelitian dikerjakan di Laboratorium Rekayasa Perangkat Lunak (RPL), Jurusan Ilmu Komputer, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Lampung. Lokasi bertempat di Jalan Prof. Dr. Soemantri Brojonegoro No. 1, Gedung Meneng, Bandar Lampung.

3.1.2 Waktu Penelitian

Penelitian dilakukan pada semester ganjil tepatnya November 2020 hingga penyelesaian bulan Agustus 2021 tahun ajaran 2020/2021. Pengerjaan terbagi menjadi tiga tahap, tahap pertama melakukan pengumpulan dan pemahaman terhadap studi literatur. Setelahnya dilakukan penyusunan draf dan pengumpulan data terkait. Tahap satu memerlukan waktu pengerjaan \pm 17 minggu. Tahap ke-2 merupakan tahap pengerjaan program yang dimulai dari ekstraksi fitur, *k-fold cross validation*, pemodelan klasifikasi menggunakan algoritme SMOTE dan metode SVM hingga tahap pengujian program yang memerlukan waktu \pm 18 minggu. Tahap terakhir merupakan tahap penyusunan hasil pengujian dan analisis program dalam draf laporan dan penyampaian hasil penelitian melalui seminar hasil dan sidang komprehensif \pm 17 minggu. Alur waktu pengerjaan dapat dilihat pada *gantt chart* Tabel 5.

Tabel 5. *Gantt Chart* Waktu Penelitian

3.2 Data dan Alat

3.2.1 Data

Data yang digunakan yang digunakan bersumber dari jurnal Kumar et al. (2017) yang didapatkan dari *database* UniProt (*release 2015_06*). Data protein Arginin terdiri atas data kelas negatif berjumlah 8.483 dan data kelas positif berjumlah 2.429 sehingga jumlah total seluruh data yang digunakan adalah 10.912 data. Data protein terbagi menjadi tiga tipe, yaitu data latih, data uji dan data independen. Rincian data protein Arginin dapat dilihat pada Tabel 6.

Tabel 6. Data Metilasi *Sequence* Protein Arginin (Kumar et al., 2017)

Tipe	Jenis Data	Jumlah <i>Sequence</i> Arginin
Data Latih	Positif	1.038
	Negatif	5.190
Data Uji	Positif	260
	Negatif	260
Data Independen	Positif	1.131
	Negatif	3.033

3.2.2 Alat

Peralatan yang digunakan dalam menunjang penelitian ini antara lain:

a. Perangkat Keras (*Hardware*)

Perangkat keras yang digunakan dalam penelitian ini adalah sebuah laptop dengan spesifikasi sebagai berikut:

- 1) *Processor*: Intel®Core™ i7-8550U CPU.
- 2) *Installed RAM*: DDR4 8.00 GB.
- 3) *Harddisk*: 1 TB 5400 RPM.

b. Perangkat Lunak (*Software*)

Perangkat lunak yang digunakan pada penelitian ini antara lain:

- 1) Sistem operasi: Windows 10 Home Single 64-bit.
- 2) R Programming 3.6.1

Bahasa pemrograman R adalah bahasa tingkat tinggi yang digunakan untuk komputasi statistik dan grafik. R menyediakan banyak pemodelan statistik seperti pemodelan linear dan nonlinear, uji statistik, analisis deret waktu, klasifikasi, serta

teknik grafis. Keunggulan bahasa pemrograman R adalah menghasilkan plot kualitas baik.

3) *R Studio 1.2.5001*

R Studio adalah *integrated development environment* (IDE) *open source* yang digunakan khusus untuk bahasa pemrograman R. Dikembangkan oleh JJ Allaire, *R Studio* tersedia dalam dua edisi yaitu *RStudio Desktop* dan *Server*.

4) *Library Caret 6.0-84*

Library Caret adalah *package* yang digunakan untuk melakukan pemodelan prediksi terhadap klasifikasi dan percobaan regresi suatu data. *Library* ini digunakan untuk mengukur hasil klasifikasi menggunakan *confusion matrix* (Kuhn et al., 2020).

5) *Library e1071 1.7-4*

Library e1071 merupakan *library* untuk melakukan analisis data kelas. Analisis yang dapat dilakukan diantaranya: *fourier transform*, *fuzzy clustering*, *support vector machine*, *bagged clustering*, dan *naïve bayes classifier* (Meyer et al., 2019).

6) *Library DMwR 0.4.1*

Library Data Mining with R (DMwR) menyediakan fitur fungsi dan konsep *data mining* pada bahasa pemrograman R, termasuk fungsi algoritme SMOTE (Torgo, 2013).

7) *Library MCC 0.4.4*

Library *mcc* berguna untuk menghitung nilai *matthew corelation coeficient* (MCC) yang menunjukkan kualitas suatu metode klasifikasi tertentu (Iuchi et al., 2018).

8) *Library protr 1.6-2*

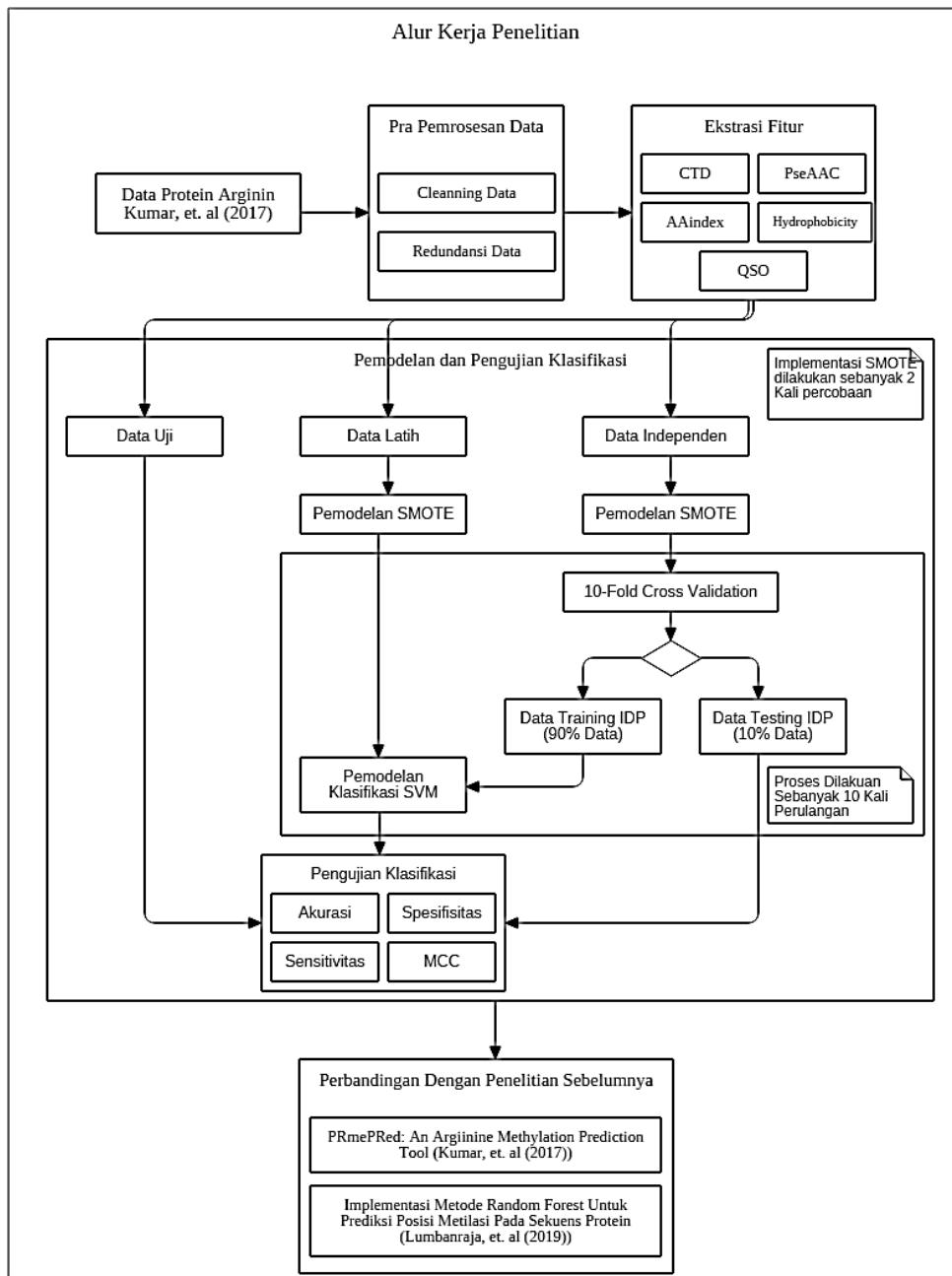
Library ini menyediakan ekstraksi fitur numerik untuk analisis urutan *sequence* protein (Xiao et al., 2015).

9) *Library BioSeqClass 1.30.0*

Library BioSeqClass merupakan *library* yang menyediakan fitur untuk mengekstraksi data *sequence* biologi dengan membangun model klasifikasi (Hong, 2018).

3.3 Alur Kerja Penelitian

Penelitian ini dilakukan berdasarkan alur penelitian pada Gambar 7.



Gambar 7 Alur Pengerjaan Penelitian.

Berdasarkan alur pengerjaan pada Gambar 7, berikut penjelasan setiap tahap.

1) Data Protein Arginin

Data protein Arginin didapatkan dari riset penelitian Kumar et al., (2017) yang terdiri dari 6.581 data negatif dan 4.331 data positif dengan total data 10.912. Panjang *sequence* protein Arginin adalah 19.

2) Prapemrosesan Data

Preprocessing terbagi menjadi dua tahap, yaitu *cleaning* dan redundansi data. Tahap *cleaning data* menghapus X yang bukan termasuk dalam jenis asam amino pada *sequence* protein. Selanjutnya, data yang sudah dibersihkan diredundansi dengan CD-HIT sebesar 40%.

3) Ekstraksi Fitur

Ekstraksi fitur bertujuan mengubah data *string* menjadi numerik. Fitur ekstraksi yang digunakan antara lain: *Composition, Transition, and Distribution* (CTD), AAindex, *Hydrophobicity*, *Pseudo Amino Acid Composition* (PseAAC), dan *Quasi-Sequence-Order* (QSO).

4) Pemodelan dan Pengujian Klasifikasi

Pemodelan klasifikasi dilakukan menggunakan algoritme SMOTE dan *Support Vector Machine*. Pemodelan SMOTE pada data latih dan data independen bertujuan untuk meningkatkan jumlah data kelas minoritas dengan membuat data sintetis. Pemodelan SMOTE dilakukan sebanyak dua kali percobaan. Selanjutnya, terkhusus data independen dilakukan pembagian data menjadi *data training* dan *data testing* menggunakan 10-fold cross validation. Berikutnya, data latih dan data independen dilakukan klasifikasi menggunakan metode SVM. Tiga data percobaan yang telah memperoleh hasil prediksi selanjutnya dilakukan pengujian menggunakan evaluasi matriks. Indikator pengukuran yang digunakan diantaranya: akurasi, sensitivitas, spesifisitas, dan MCC.

5) Perbandingan dengan Penelitian Sebelumnya

Tahap ini melakukan perbandingan hasil pengujian dengan riset berikut:

- a) *PReMed: An Arginine Methylation Prediction Tool* (Kumar et al., 2017).
- b) Implementasi Metode *Random Forest* Untuk Prediksi Posisi Metilasi Pada Sekuens Protein (Lumbanraja et al., 2019).

V. PENUTUP

5.1 Simpulan

Adapun simpulan pada penelitian klasifikasi ketidakseimbangan data metilasi protein Arginin menggunakan algoritme SMOTE dan metode *support vector machine*, sebagai berikut:

1. Data protein Arginin yang digunakan bersumber dari penelitian Kumar et al. (2017) yang terdiri dari tiga data percobaan, yaitu: data latih, data uji, dan data independen. Jumlah data protein Arginin secara keseluruhan adalah 10.912 yang terdiri dari 2.429 data positif dan 8.483 data negatif dengan masing-masing panjang sekuen adalah 19.
2. Data protein tipe data *string* diekstrak berdasarkan sifat fisikokimia maupun biokimia menggunakan lima fitur ekstraksi, yang terdiri dari: *composition*, *transition*, dan *distribution* (CTD), AAindex, *hydrophobicity*, *pseudo amino acid composition* (PseAAC), dan *quasi-sequence-order* (QSO).
3. SMOTE diimplementasikan sebanyak dua kali, yang masing-masing meningkatkan data latih positif sebesar 75% dan 80% dari data awal. Jumlah data positif menjadi 4.072 dan 5.090 dari 1.018 data. Pada data independen, SMOTE menyintesiskan 66,7% dan 75% yang masing-masing berjumlah 1.458 dan 1.944 dari data semula yang berjumlah 486.
4. Hasil akurasi pengujian data latih menggunakan SVM sebesar 100% pada *kernel* RBF lebih baik daripada akurasi pada penelitian Kumar et al. (2017) dan Lumbanraja et al. (2019), masing-masing adalah 90% dan 93,76%. Data independen juga memberikan nilai rata-rata akurasi tertinggi dari dua penelitian sebelumnya sebesar 98,5% pada *kernel*

linear. Akurasi terendah didapat pada data uji dengan tingkat akurasi 64,9% yang diperoleh pada *kernel* linear.

5. Penerapan algoritme SMOTE dan metode SVM pada data metilasi protein Arginin memiliki kinerja yang memuaskan khususnya pada data latih dan data independen. Sebaliknya untuk data uji, implementasi SMOTE masih kurang maksimal.

5.2 Saran

Adapun saran yang dapat diberikan pada penelitian ini adalah sebagai berikut:

1. Dalam meminimalisir kesalahan *overfitting* atau *curse of dimensional* pada data tidak seimbang, dapat menggunakan teknik seleksi fitur yang diimplementasikan dengan algoritme SMOTE atau sejenisnya seperti ADASYN dengan metode klasifikasi tertentu.
2. Dapat menggunakan metode klasifikasi lain seperti XGboost, K-*Nearest neighbors*, atau *Artificial Neural Network* (ANN) untuk mendapatkan hasil klasifikasi pembanding dalam menangani masalah ketidakseimbangan data.

DAFTAR PUSTAKA

- Aletta, J. M., Cimato, T. R., & Ettinger, M. J. (1998). Protein methylation: A signal event in post-translational modification. *Trends in Biochemical Sciences*, 23(3), 89–91. [https://doi.org/10.1016/S0968-0004\(98\)01185-2](https://doi.org/10.1016/S0968-0004(98)01185-2).
- AMARI, S. (1987). Neural Information Processing. *Journal of the Society of Mechanical Engineers*, 90(823), 758–759. https://doi.org/10.1299/jsmemag.90.823_758.
- Anderson, N. L., & Anderson, N. G. (1998). Proteome and proteomics: New technologies, new concepts, and new words. *Electrophoresis*, 19(11), 1853–1861. <https://doi.org/10.1002/elps.1150191103>.
- Bannister, A. J., & Kouzarides, T. (2005). Reversing histone methylation. *Nature*, 436(7054), 1103–1106. <https://doi.org/10.1038/nature04048>.
- Batuwita, R., & Palade, V. (2009). microPred: Effective classification of pre-miRNAs for human miRNA gene prediction. *Bioinformatics*, 25(8), 989–995. <https://doi.org/10.1093/bioinformatics/btp107>.
- Bedford, M. T., & Richard, S. (2005). Arginine methylation: An emerging regulator of protein function. In *Molecular Cell* (Vol. 18, Issue 3, pp. 263–272). <https://doi.org/10.1016/j.molcel.2005.04.003>.
- 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. <http://www.iiste.org/Journals/index.php/JIEA/article/view/7633>.
- Bharill, N., Tiwari, A., & Rawat, A. (2015). A novel technique of feature extraction with dual similarity measures for protein sequence classification. *Procedia Computer Science*, 48(C), 795–801. <https://doi.org/10.1016/j.procs.2015.04.217>.
- Blagus, R., & Lusa, L. (2013). SMOTE for high-dimensional class-imbalanced data. *BMC Bioinformatics*, 14. <https://doi.org/10.1186/1471-2105-14-106>.

- Breiman, L. (2001). Random forests. *Machine Learning*, 45(1), 5–32. <https://doi.org/10.1023/A:1010933404324>.
- Buxbaum, E. (2007). Fundamentals of protein structure and function. In *Fundamentals of Protein Structure and Function*. <https://doi.org/10.1007/978-0-387-68480-2>
- Cardie, C. (1997). Improving minority class prediction using case-specific feature weights. *Proceedings of the Fourteenth International Conference on Machine Learning*, 57–65. <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.118.484>.
- Chang, C. C., & Lin, C. J. (2001). Training v-support vector classifiers: Theory and algorithms. *Neural Computation*, 13(9), 2119–2147. <https://doi.org/10.1162/089976601750399335>.
- Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16, 321–357. <https://doi.org/10.1613/jair.953>.
- Chou, K. C. (2000). Prediction of protein subcellular locations by incorporating quasi-sequence-order effect. *Biochemical and Biophysical Research Communications*, 278(2), 477–483. <https://doi.org/10.1006/bbrc.2000.3815>.
- Chou, K. C. (2001). Prediction of protein cellular attributes using pseudo-amino acid composition. *Proteins: Structure, Function and Genetics*, 43(3), 246–255. <https://doi.org/10.1002/prot.1035>.
- Cieslak, D. A., Chawla, N. V., & Striegel, A. (2006). Combating imbalance in network intrusion datasets. *2006 IEEE International Conference on Granular Computing*, 732–737. <https://doi.org/10.1109/grc.2006.1635905>.
- Cortes, C., & Vapnik, V. (1995). Support-Vector Networks. *Machine Learning*, 20(3), 273–297. <https://doi.org/10.1023/A:1022627411411>.
- Cristianini, N., & Hahn, M. W. (2006). Introduction to computational genomics: A case studies approach. In *Introduction to Computational Genomics: A Case Studies Approach* (Vol. 9780521856). <https://doi.org/10.1017/CBO9780511808982>.
- Didonna, A., & Benetti, F. (2016). Post-translational modifications in neurodegeneration. In *AIMS Biophysics* (Vol. 3, Issue 1, pp. 27–49). <https://doi.org/10.3934/biophy.2016.1.27>.
- Faisal, M. R. (2016). Seri Belajar Data Science: Klasifikasi dengan Bahasa Pemrograman R. In *Indonesia Net Developer Community* (Issue February).

- Fallah, A., & Jafari, S. (2011). An Expert System for Detection of Breast Cancer Using Data Preprocessing and Bayesian Network. *International Journal of Advanced Science and Technology*, 34, 65–70.
- Gary, J. D., & Clarke, S. (1998). RNA and protein interactions modulated by protein arginine methylation. In *Progress in nucleic acid research and molecular biology* (Vol. 61, pp. 65–131). [https://doi.org/10.1016/s0079-6603\(08\)60825-9](https://doi.org/10.1016/s0079-6603(08)60825-9).
- Govindan, G., & Nair, A. S. (2011). Composition, Transition and Distribution (CTD) - A dynamic feature for predictions based on hierarchical structure of cellular sorting. *Proceedings - 2011 Annual IEEE India Conference: Engineering Sustainable Solutions, INDICON-2011*. <https://doi.org/10.1109/INDCON.2011.6139332>.
- Guo, A., Gu, H., Zhou, J., Mulhern, D., Wang, Y., Lee, K. A., Yang, V., Aguiar, M., Kornhauser, J., Jia, X., Ren, J., Beausoleil, S. A., Silva, J. C., Vemulapalli, V., Bedford, M. T., & Comb, M. J. (2014). Immunoaffinity enrichment and mass spectrometry analysis of protein methylation. *Molecular and Cellular Proteomics*, 13(1), 372–387. <https://doi.org/10.1074/mcp.O113.027870>.
- Hong, L. (2018). Using the BioSeqClass Package. In *BioSeqClass: Classification for Biological Sequences*. <https://doi.org/10.18129/B9.bioc.BioSeqClass>.
- Hornbeck, P. V., Zhang, B., Murray, B., Kornhauser, J. M., Latham, V., & Skrzypek, E. (2015). PhosphoSitePlus, 2014: Mutations, PTMs and recalibrations. *Nucleic Acids Research*, 43(D1), D512–D520. <https://doi.org/10.1093/nar/gku1267>.
- Iuchi, H., Sugimoto, M., & Tomita, M. (2018). MICOP: Maximal information coefficient-based oscillation prediction to detect biological rhythms in proteomics data. *BMC Bioinformatics*. <https://doi.org/10.1186/s12859-018-2257-4>.
- James, G., Witten, D., Hastie, T., & Tibshirani, R. (2000). An introduction to Statistical Learning. In *Current medicinal chemistry* (Vol. 7, Issue 10). <https://doi.org/10.1007/978-1-4614-7138-7>.
- Kawashima, S., Pokarowski, P., Pokarowska, M., Kolinski, A., Katayama, T., & Kanehisa, M. (2008a). AAindex: Amino acid index database, progress report 2008. *Nucleic Acids Research*. <https://doi.org/10.1093/nar/gkm998>.
- Kawashima, S., Pokarowski, P., Pokarowska, M., Kolinski, A., Katayama, T., & Kanehisa, M. (2008b). AAindex: Amino acid index database, progress report 2008. *Nucleic Acids Research*, 36(SUPPL. 1). <https://doi.org/10.1093/nar/gkm998>.

- Kothandan, R. (2015). Handling class imbalance problem in miRNA dataset associated with cancer. *Bioinformation*, 11(1), 6–10. <https://doi.org/10.6026/97320630011006>.
- Kubat, M., Holte, R. C., & Matwin, S. (1998). Machine learning for the detection of oil spills in satellite radar images. *Machine Learning*, 30(2–3), 195–215. <https://doi.org/10.1023/a:1007452223027>.
- Kuhn, M., & Johnson, K. (2013). Applied predictive modeling. In *Applied Predictive Modeling*. <https://doi.org/10.1007/978-1-4614-6849-3>.
- Kuhn, M., Wing, J., Weston, S., Williams, A., Keefer, C., Engelhardt, A., Cooper, T., Mayer, Z., Kenkel, B., Team, R. C., Benesty, M., Lescarbeau, R., Ziem, A., Scrucca, L., Tang, Y., Candan, C., & Hunt, T. (2020). Package “caret.” In *The R Journal*.
- Kumar, P., Joy, J., Pandey, A., & Gupta, D. (2017). PRmePRed: A protein arginine methylation prediction tool. *PLoS ONE*, 12(8). <https://doi.org/10.1371/journal.pone.0183318>.
- Kyte, J., & Doolittle, R. F. (1982). A simple method for displaying the hydropathic character of a protein. *Journal of Molecular Biology*, 157(1), 105–132. [https://doi.org/10.1016/0022-2836\(82\)90515-0](https://doi.org/10.1016/0022-2836(82)90515-0).
- Lee, D. Y., Teyssier, C., Strahl, B. D., & Stallcup, M. R. (2005). Role of protein methylation in regulation of transcription. In *Endocrine Reviews* (Vol. 26, Issue 2, pp. 147–170). <https://doi.org/10.1210/er.2004-0008>.
- Li, Q., & Shah, S. (2017). Structure-Based Virtual Screening BT - Protein Bioinformatics: From Protein Modifications and Networks to Proteomics. In *Methods in Molecular Biology* (Vol. 1558, Issue 1558). <https://www.springer.com/us/book/9781493967810>.
- Li, W., & Godzik, A. (2006). Cd-hit: A fast program for clustering and comparing large sets of protein or nucleotide sequences. *Bioinformatics*, 22(13). <https://doi.org/10.1093/bioinformatics/btl158>.
- Lloyd, M. (2000). Genome: the Autobiography of a Species in 23 Chapters. *Briefings in Bioinformatics*, 1(1), 103–103. <https://doi.org/10.1093/bib/1.1.103>.
- Lumbanraja, F. R., Mudyaningsih, W., Hermanto, B., Syarif, A., & Komputer, J. I. (2019). Implementasi Metode Random Forest Untuk Prediksi Posisi Metilasi Pada Sekuens Protein. *Seminar Nasional Sains, Matematika, Informatika, Dan Aplikasinya*, 105–112. <http://jurnal.fmipa.unila.ac.id/snsmap/article/view/2461>.

- Ma, L., Li, M., Ma, X., Cheng, L., Du, P., & Liu, Y. (2017). A review of supervised object-based land-cover image classification. In *ISPRS Journal of Photogrammetry and Remote Sensing* (Vol. 130, pp. 277–293). <https://doi.org/10.1016/j.isprsjprs.2017.06.001>.
- MacIsaac, K. D., Gordon, D. B., Nekludova, L., Odom, D. T., Schreiber, J., Gifford, D. K., Young, R. A., & Fraenkel, E. (2006). A hypothesis-based approach for identifying the binding specificity of regulatory proteins from chromatin immunoprecipitation data. *Bioinformatics*, 22(4), 423–429. <https://doi.org/10.1093/bioinformatics/bti815>.
- Maimon, O., & Rokach, L. (2010). Data Mining and Knowledge Discovery Handbook. In *Data Mining and Knowledge Discovery Handbook*. <https://doi.org/10.1007/978-0-387-09823-4>.
- Meyer, D., Dimitriadou, E., Hornik, K., Weingessel, A., Leisch, F., Chang, C.-C., & Lin, C.-C. (2019). Package ‘e1071’: Misc Functions of the Department of Statistics, Probability Theory Group (Formerly: E1071), TU Wien. In *R package version 1.7-3*.
- Noorhalim, N., Ali, A., & Shamsuddin, S. M. (2019). Handling Imbalanced Ratio for Class Imbalance Problem Using SMOTE. In *Proceedings of the Third International Conference on Computing, Mathematics and Statistics (iCMS2017)* (pp. 19–30). https://doi.org/10.1007/978-981-13-7279-7_3.
- Paik, W. K., & Kim, S. (1967). Enzymatic methylation of protein fractions from calf thymus nuclei. *Biochemical and Biophysical Research Communications*, 29(1), 14–20. [https://doi.org/10.1016/0006-291X\(67\)90533-5](https://doi.org/10.1016/0006-291X(67)90533-5).
- Paik, W. K., Paik, D. C., & Kim, S. (2007). Historical review: the field of protein methylation. In *Trends in Biochemical Sciences* (Vol. 32, Issue 3, pp. 146–152). <https://doi.org/10.1016/j.tibs.2007.01.006>.
- PAMUNGKAS, C. A. (2017). Pengantar dan Implementasi Basis Data/oleh Canggih Ajika Pamungkas. In *Pengantar dan Implementasi Basis Data*.
- Qiu, W. R., Sun, B. Q., Xiao, X., Xu, Z. C., & Chou, K. C. (2016). iPBM-Lys: Identifying multiple lysine PTM sites and their different types. *Bioinformatics*, 32(20), 3116–3123. <https://doi.org/10.1093/bioinformatics/btw380>.
- Schölkopf, B. (2002). Learning with kernels. *Proceedings of 2002 International Conference on Machine Learning and Cybernetics*, 1. <https://doi.org/10.7551/mitpress/4175.001.0001>.
- Schölkopf, B., Smola, A. J., Williamson, R. C., & Bartlett, P. L. (2000). New support vector algorithms. *Neural Computation*, 12(5), 1207–1245. <https://doi.org/10.1162/089976600300015565>.

- Schubert, H. L., Blumenthal, R. M., & Cheng, X. (2006). 1 Protein methyltransferases: Their distribution among the five structural classes of adomet-dependent methyltransferases. *Enzymes*, 24(C), 3–28. [https://doi.org/10.1016/S1874-6047\(06\)80003-X](https://doi.org/10.1016/S1874-6047(06)80003-X).
- Shanmuganathan, S. (2016). Artificial neural network modelling: An introduction. In *Studies in Computational Intelligence* (Vol. 628, pp. 1–14). https://doi.org/10.1007/978-3-319-28495-8_1.
- Shavers, C., Li, R., & Lebby, G. (2006). An SVM-based approach to face detection. *Proceedings of the Annual Southeastern Symposium on System Theory, 2006*, 362–366. <https://doi.org/10.1109/ssst.2006.1619082>.
- Shen, E. C., Henry, M. F., Weiss, V. H., Valentini, S. R., Silver, P. A., & Lee, M. S. (1998). Arginine methylation facilitates the nuclear export of hnRNP proteins. *Genes and Development*, 12(5), 679–691. <https://doi.org/10.1101/gad.12.5.679>.
- Siringoringo, R. (2018). Klasifikasi Data Tidak Seimbang Menggunakan Algoritma SMOTE dan k-Nearest Neighbor. *Jurnal ISD*, 3(1), 44–49.
- Su, R., Zhang, C., Pham, T. D., Davey, R., Bischof, L., Vallotton, P., Lovell, D., Hope, S., Schmoelzl, S., & Sun, C. (2016). Detection of tubule boundaries based on circular shortest path and polar-transformation of arbitrary shapes. *Journal of Microscopy*, 264(2), 127–142. <https://doi.org/10.1111/jmi.12421>.
- 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>.
- Terzic, J., Terzic, E., Nagarajah, R., & Alamgir, M. (2013). Ultrasonic fluid quantity measurement in dynamic vehicular applications: A support vector machine approach. In *Ultrasonic Fluid Quantity Measurement in Dynamic Vehicular Applications: A Support Vector Machine Approach* (Vol. 9783319006). <https://doi.org/10.1007/978-3-319-00633-8>.
- Torgo, L. (2013). *DMwR: Functions and data for “Data Mining with R.”* 2013/08/08.
- Velculescu, V. E., Zhang, L., Zhou, W., Vogelstein, J., Basrai, M. A., Bassett, D. E., Hieter, P., Vogelstein, B., & Kinzler, K. W. (1997). Characterization of the yeast transcriptome. *Cell*, 88(2), 243–251. [https://doi.org/10.1016/S0092-8674\(00\)81845-0](https://doi.org/10.1016/S0092-8674(00)81845-0).
- Vuzman, D., Hoffman, Y., & Levy, Y. (2012). Modulating protein-DNA interactions by post-translational modifications at disordered regions. *Pacific Symposium on Biocomputing*, 188–199. https://doi.org/10.1142/9789814366496_0018.

- Wang, Y. C., Peterson, S. E., & Loring, J. F. (2014). Protein post-translational modifications and regulation of pluripotency in human stem cells. In *Cell Research* (Vol. 24, Issue 2, pp. 143–160). <https://doi.org/10.1038/cr.2013.151>.
- Wei, L., Liao, M., Gao, X., & Zou, Q. (2015a). An improved protein structural classes prediction method by incorporating both sequence and structure information. *IEEE Transactions on Nanobioscience*, 14(4), 339–349. <https://doi.org/10.1109/TNB.2014.2352454>.
- Wei, L., Liao, M., Gao, X., & Zou, Q. (2015b). Enhanced Protein Fold Prediction Method Through a Novel Feature Extraction Technique. *IEEE Transactions on Nanobioscience*, 14(6), 649–659. <https://doi.org/10.1109/TNB.2015.2450233>.
- Wei, L., Tang, J., & Zou, Q. (2017). Local-DPP: An improved DNA-binding protein prediction method by exploring local evolutionary information. *Information Sciences*, 384, 135–144. <https://doi.org/10.1016/j.ins.2016.06.026>.
- Wei, L., Xing, P., Shi, G., Ji, Z. L., & Zou, Q. (2017). Fast prediction of protein methylation sites using a sequence-based feature selection technique. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*. <https://doi.org/10.1109/TCBB.2017.2670558>.
- Xiao, J., Tang, X., Li, Y., Fang, Z., Ma, D., He, Y., & Li, M. (2011). Identification of microRNA precursors based on random forest with network-level representation method of stem-loop structure. *BMC Bioinformatics*, 12. <https://doi.org/10.1186/1471-2105-12-165>.
- Xiao, N., Cao, D. S., Zhu, M. F., & Xu, Q. S. (2015). Protr/ProtrWeb: R package and web server for generating various numerical representation schemes of protein sequences. *Bioinformatics*. <https://doi.org/10.1093/bioinformatics/btv042>.
- Yu, Y., Song, C., Zhang, Q., DiMaggio, P. A., Garcia, B. A., York, A., Carey, M. F., & Grunstein, M. (2012). Histone H3 Lysine 56 Methylation Regulates DNA Replication through Its Interaction with PCNA. *Molecular Cell*, 46(1), 7–17. <https://doi.org/10.1016/j.molcel.2012.01.019>.
- Zhang, C., Gao, S., Molascon, A. J., Liu, Y., & Andrews, P. C. (2014). Quantitative proteomics reveals histone modifications in crosstalk with h3 lysine 27 methylation. *Molecular and Cellular Proteomics*, 13(3), 749–759. <https://doi.org/10.1074/mcp.M113.029025>.
- Zhang, C. J., Tang, H., Li, W. C., Lin, H., Chen, W., & Chou, K. C. (2016). iOri-Human: Identify human origin of replication by incorporating dinucleotide physicochemical properties into pseudo nucleotide composition. *Oncotarget*, 7(43), 69783–69793. <https://doi.org/10.18632/oncotarget.11975>.