Available at https://jtsiskom.undip.ac.id (31 January 2022)
DOI:10.14710/jtsiskom.2022.13984

Jurnal Teknologi dan Sistem Komputer, 10(1), 2022, 1-11

Prediksi interaksi protein-protein berbasis sekuens protein menggunakan fitur *autocorrelation* dan *machine learning*

Sequence-based prediction of protein-protein interaction using autocorrelation features and machine learning

Syahid Abdullah[1)], Wisnu Ananta Kusuna[1,2*)], Sony Hartono Wijaya[1)]

[1)]*Departemen Ilmu Komputer, Fakultas Matematika dan IPA, Institut Pertanian Bogor*
Jl. Raya Dramaga, Kampus IPB Dramaga, Bogor 16680, Indonesia

[2)]*Pusat Studi Biofarmaka Tropika, Institut Pertanian Bogor*
Jl. Taman Kencana No. 3, Bogor 16128, Indonesia

Cara sitasi: S. Abdullah, W. A. Kusuma, and S. H. Wijaya, "Prediksi protein-protein interaction berbasis sekuens protein menggunakan fitur autocorrelation dan machine learning," Jurnal Teknologi dan Sistem Komputer, vol. 10, no. 1, pp. 1-11, 2022. doi: 10.14710/jtsiskom.2022.13984, [Online].

Abstract* - Protein-protein interaction (PPI) can be used to define a protein's function by knowing the protein's position in a complex network of protein interactions. The number of PPIs that have been identified is relatively small. Therefore, several studies were conducted to predict PPI using protein sequence information. This research compares the performance of three autocorrelation methods: Moran, Geary, and Moreau-Broto, in extracting protein sequence features to predict PPI. The results of the three extractions are then applied to three machine learning algorithms, namely k-Nearest Neighbor (KNN), Random Forest, and Support Vector Machine (SVM). The prediction models with the three autocorrelation methods can produce predictions with high average accuracy, which is 95.34% for Geary in KNN, 97.43% for Geary in RF, and 97.11% for Geary and Moran in SVM. In addition, the interacting protein pairs tend to have similar autocorrelation characteristics. Thus, the autocorrelation method can be used to predict PPI well.

Keywords *- autocorrelation; machine learning; protein-protein interaction; protein sequence*

Abstrak* - Interaksi protein-protein atau protein-protein interaction (PPI) dapat digunakan untuk mendefinisikan fungsi sebuah protein dengan mengetahui posisi protein tersebut dalam sebuah jaringan kompleks interaksi protein. Jumlah PPI yang berhasil diidentifikasi relatif masih sedikit. Oleh karena itu, beberapa penelitian dilakukan untuk memprediksi PPI menggunakan informasi sekuens protein. Dalam penelitian ini, dilakukan perbandingan kinerja tiga metode autocorrelation, yaitu Moran, Geary, dan Moreau-Broto dalam mengekstraksi fitur sekuens protein untuk memprediksi PPI. Hasil ekstraksi ketiganya diterapkan pada tiga algoritme machine learning, yaitu k-Nearest Neighbor (KNN), Random Forest, dan Support Vector Machine (SVM). Setelah dilakukan prediksi, diketahui bahwa fitur yang dihasilkan oleh ketiga metode autocorrelation tersebut dapat menghasilkan prediksi dengan rerata akurasi yang tinggi, yaitu sebesar 95,34% untuk Geary di KNN, 97,43% untuk Geary di RF, dan 97,11% untuk Geary dan Moran di SVM. Selain itu, dari penelitian ini juga diketahui bahwa pasangan protein yang berinteraksi cenderung memiliki fitur autocorrelation yang mirip. Dengan demikian, metode autocorrelation dapat dipertimbangkan sebagai metode yang dapat memprediksi PPI dengan baik.

Kata kunci *- autocorrelation; machine learning; interaksi protein-protein; sekuens protein*

I. PENDAHULUAN

Protein merupakan molekul yang memiliki peranan penting dalam menjalankan fungsi sel suatu organisme. Untuk menjalankan fungsinya, protein seringkali berinteraksi dengan protein lain [1]. Interaksi antar protein atau *protein-protein interaction* (PPI) adalah hubungan molekuler antara satu protein dengan protein lainnya yang terjadi di dalam sebuah sel dalam suatu organisme hidup [2]. PPI melibatkan dua protein yang saling mengikat. Interaksi yang terjadi banyak menimbulkan fungsi-fungsi biologis tertentu [3]. Fungsi-fungsi biologis ini mengendalikan proses-proses biologis yang terjadi di dalam sel, seperti respons imun dan organisasi seluler. Himpunan dari PPI biasa divisualisasikan dalam bentuk graf dan disebut sebagai jejaring PPI (*PPIs network*).

Gambar 1 adalah ilustrasi jejaring PPI yang melibatkan beberapa protein dengan setiap node merepresentasikan suatu protein sedangkan setiap verteks menunjukkan adanya interaksi antara dua protein. Interaksi antar protein atau PPI terjadi antara P1 dan P3, P2 dan P3, serta P2 dan P4. Adapun P1 dan P2 serta P1 dan P4 merupakan pasangan protein yang tidak

[*)] Penulis korespondensi (Wisnu Ananta Kusuma)
Email: ananta@apps.ipb.ac.id

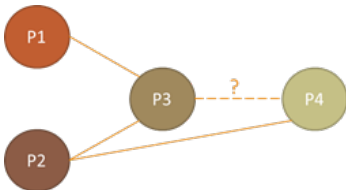
Submitted: 20 November 2020; Revised: 14 September 2021; Accepted: 4 January 2022; Published: 31 January 2022

saling berinteraksi. P3 dan P4 adalah pasangan protein yang belum diketahui apakah ada interaksi antara keduanya. Dari jejaring PPI tersebut, dapat diperoleh data PPI dengan dua kelas, yaitu kelas positif dan negatif seperti pada Tabel 1 dan Tabel 2.

Penelitian mengenai PPI cukup penting untuk mengungkap proses pengaturan seluler [4]. PPI dapat digunakan untuk mendefinisikan fungsi suatu protein dengan mengetahui posisi protein tersebut dalam sebuah jejaring PPI. Posisi protein dalam suatu jejaring PPI menentukan seberapa signifikan peran protein tersebut dalam jejaring PPI. Identifikasi terhadap protein yang signifikan tersebut dapat bermanfaat dalam berbagai penelitian, salah satunya adalah untuk penemuan obat baru [1], [5]. Hal tersebut dapat dilakukan dengan mencari protein-protein yang berasosiasi dengan suatu penyakit tertentu. Setelah itu, dicari jejaring PPI dari setiap protein yang berasosiasi tersebut. Setelah dilakukan identifikasi protein yang signifikan dalam jejaring PPI tersebut, dapat dilakukan pemberian senyawa obat yang dapat menormalkan fungsi protein-protein tersebut.

Untuk mengukur seberapa signifikan suatu protein, diperlukan suatu jejaring PPI yang lengkap. Sampai saat ini, data jejaring PPI belum lengkap karena masih banyak interaksi antar protein atau PPI yang belum diketahui. Oleh karena itu, banyak metode penelitian yang dilakukan untuk mengidentifikasi PPI baru. Metode-metode tersebut berhasil mengidentifikasi banyak PPI baru, namun membutuhkan banyak biaya dan waktu. Selain itu, jumlah PPI yang berhasil diidentifikasi relatif masih sedikit karena menurut estimasi ada sekitar 300 ribu PPI yang ada di manusia [6]. Shen dkk. [7] berhasil melakukan prediksi PPI dengan hanya menggunakan informasi sekuens protein. Sekuens protein merupakan struktur dasar dari protein yang terdiri dari susunan asam amino yang berurutan [8]. Metode yang digunakan adalah *Support Vector Machine* (SVM) untuk memprediksi PPI dengan terlebih dahulu menggunakan metode *conjoint triad* (CT) untuk mengekstraksi sekuens protein untuk memperoleh fitur yang diinginkan. Selanjutnya, Guo dkk. [9] menggunakan SVM yang dikombinasikan dengan *autocovariance* (AC) sebagai metode ekstraksi fitur. Setelah itu, banyak penelitian yang dilakukan untuk memprediksi PPI berbasis sekuens protein dengan menerapkan metode *machine learning* lainnya, seperti Pan dkk. [6] yang menggunakan *Random Forest* (RF) dengan CT dan Yang dkk. [10] dengan *k-Nearest Neighbors* (KNN) sebagai metode klasifikasi dan *local descriptor* (LD) untuk ekstraksi fitur.

Menurut Pevsner [11], interaksi antar protein atau PPI cenderung akan terjadi antara sepasang protein yang memiliki kesamaan tertentu. Metode-metode ekstraksi fitur tersebut digunakan untuk merepresentasikan sekuens dari dua buah protein dalam bentuk numerik sehingga dapat diamati kesamaan dari dua protein tersebut. Selain CT, AC, dan LD, metode ekstraksi fitur yang populer digunakan adalah *autocorrelation*. Xia dkk. [12] dan You dkk. [13] menggunakan metode



Gambar 1. Contoh data sekuens protein

Tabel 1. Data interaksi antar protein dengan kelas negatif

No	Protein 1	Protein 2
1	P1	P2
2	P1	P4

Tabel 2. Data interaksi antar protein dengan kelas positif

No	Protein 1	Protein 2
1	P1	P3
2	P2	P3
3	P2	P4

autocorrelation untuk memprediksi PPI dengan *rotation forest* dan SVM. *Autocorrelation* merupakan metode yang digunakan untuk menghitung korelasi atau hubungan antar asam amino dalam suatu sekuens protein berdasarkan sifat-sifat tertentu pada masing-masing asam amino. Penelitian-penelitian dengan menggunakan fitur *autocorrelation* tersebut menunjukkan hasil prediksi dengan tingkat akurasi yang cukup baik. Hal tersebut menunjukkan bahwa interaksi antar protein atau PPI dapat diduga dengan melihat korelasi antar asam amino yang menyusun pasangan protein yang terlibat. Sejauh ini metode *autocorrelation* yang digunakan adalah Moran autocorrelation.

Selain Moran autocorrelation, terdapat metode lain untuk menghitung autocorrelation, yaitu Geary *autocorrelation* dan Moreau-Broto *autocorrelation* [14]. Ketiga fitur *autocorrelation* tersebut memiliki formulasi yang berbeda dalam merepresentasikan korelasi antar asam amino dalam suatu protein. Penelitian ini bertujuan untuk membandingkan kinerja Moran, Geary, dan Moreau-Broto *autocorrelation* dalam mengekstraksi fitur dari sekuens protein untuk memprediksi PPI. Hasil ekstraksi dari ketiga metode tersebut digunakan dalam prediksi PPI beberapa algoritme *machine learning*, yaitu KNN, RF, dan SVM. Ketiga algoritme tersebut merupakan algoritme yang telah banyak digunakan untuk menyelesaikan berbagai permasalahan klasifikasi. Selain untuk mengetahui kinerja ketiga metode *autocorrelation* tersebut, penelitian ini juga dilakukan untuk mengetahui kondisi *autocorrelation* seperti apa yang membuat dua protein berinteraksi, yang mana hal tersebut belum pernah dibahas pada penelitian-penelitian sebelumnya.

Data protein yang digunakan dalam penelitian ini dihimpun dari berbagai sumber basis data, di antaranya

Indonesia Jamu-Herbs (IJAH), STRING, UniProt, dan LR-PPI. STRING merupakan sebuah basis data PPI yang juga dapat memprediksi PPI antar protein. Selain STRING, telah banyak *pipeline* yang dapat digunakan untuk memprediksi PPI. Sebagai catatan, penelitian ini dilakukan untuk menganalisa penggunaan fitur *autocorrelation* dan *machine learning* dalam memprediksi PPI. Hasil dari penelitian ini diharapkan dapat berguna untuk mengembangkan *pipeline-pipeline* prediksi protein yang sudah ada.

Dalam bidang bioinformatika, KNN telah berhasil digunakan untuk memprediksi fungsi gen [15] dan ekspresi gen [16]. RF telah banyak diterapkan di antaranya untuk memprediksi prediksi kanker oesophageal [17], prediksi khasiat jamu [18], dan klasifikasi penyakit stroke otak [19]. SVM juga telah banyak diterapkan seperti pada permasalahan klasifikasi berbagai macam kanker [20] dan juga pada prediksi khasiat jamu [18]. Dalam kajian ini, hasil prediksi dari ketiga algoritme klasifikasi tersebut dievaluasi untuk mengetahui apakah metode-metode tersebut dapat digunakan untuk memprediksi PPI dengan baik.

II. METODE PENELITIAN

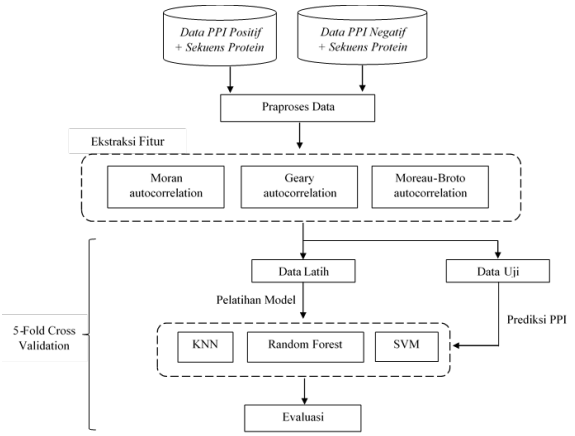
Penelitian ini dimulai dengan mengumpulkan data PPI kelas positif dan negatif beserta sekuens masing-masing protein dari beberapa basis data. Praproses data dilakukan untuk menyeimbangkan data serta menghapus protein yang mengandung asam amino yang tidak dikenali. Setelah itu sekuens protein dari setiap pasangan protein diekstraksi untuk mendapatkan fitur yang diperlukan menggunakan metode Moran, Geary, dan Moreau-Broto *autocorrelation*. Fitur yang diperoleh digunakan sebagai masukan algoritme klasifikasi KNN, RF, dan SVM untuk memprediksi PPI. Selanjutnya, hasil prediksi dievaluasi dengan beberapa pengukuran untuk mengetahui seberapa baik metode yang digunakan dalam penelitian ini. Penelitian ini dilakukan menggunakan perangkat keras dengan spesifikasi: Intel(R) Core(TM) i3-4030U CPU 1.90GHz, RAM 6 GB, dan HDD 500 GB. Perangkat lunak yang digunakan adalah sistem operasi Windows 10 64-bit dengan bahasa pemrograman R versi 3.4.3 dan RStudio versi 1.1.419. Tahapan pada penelitian ini dapat dilihat pada Gambar 2.

A. Data PPI dan sekuens protein

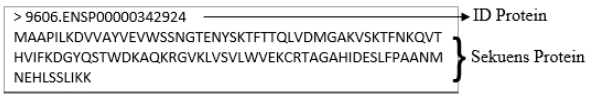
Data yang digunakan berupa sekuens protein yang terdiri dari susunan asam amino yang telah dikodekan dalam bentuk kode IUPAC. Data untuk pelatihan dan pengujian model dalam penelitian ini merupakan gugus data protein yang diperoleh dari basis data IJAH Analytics, STRING, UniProt, dan LR-PPI. Basis data IJAH Analytics[1] hanya terdiri dari data protein sejumlah 3.334 protein dan tidak memiliki data interaksi antar protein (PPI) [21]. Oleh karena itu, data PPI dari kumpulan protein IJAH diperoleh dari basis data

Tabel 3. Kode IUPAC asam amino [23]

Kode 3-huruf	Kode IUPAC	Asam Amino
A	Ala	Alanine
C	Cys	Cysteine
D	Asp	Aspartic Acid
E	Glu	Glutamic Acid
F	Phe	Phenylalanine
G	Gly	Glycine
H	His	Histidine
I	Ile	Isoleucine
K	Lys	Lysine
L	Leu	Leucine
M	Met	Methionine
N	Asn	Asparagine
P	Pro	Proline
Q	Gln	Glutamine
R	Arg	Arginine
S	Ser	Serine
T	Thr	Threonine
V	Val	Valine
W	Trp	Tryptophan
Y	Tyr	Tyrosine



Gambar 2. Tahapan penelitian



Gambar 3. Contoh data sekuens protein

STRING[2] yang memuat PPI dengan kelas positif [22]. Dari basis data tersebut, diperoleh pasangan protein dengan jumlah 6.628 pasang. Gugus data tersebut disebut sebagai data PPI kelas positif, di mana setiap pasangan protein dalam kelas tersebut merupakan pasangan protein yang berinteraksi. Jumlah pasangan protein yang diperoleh adalah 68.628 pasang.

Data sekuens protein yang terlibat dalam PPI tersebut diperoleh dari basis data UniProt[3]. UniProt memuat informasi rinci mengenai protein termasuk sekuens protein terkait [23]. Sekuens protein terdiri dari beberapa asam amino yang saling berurutan. Data

[1]http://ijah.apps.cs.ipb.ac.id

[2]http://string-db.org
[3]https://www.uniprot.org

sekuens protein yang digunakan merupakan susunan asam amino yang telah dikodekan dalam bentuk kode IUPAC [24]. Masing-masing asam amino diwakili oleh sebuah alfabet seperti pada Tabel 3, sedangkan Gambar 3 adalah contoh data sekuens protein yang terdiri dari ID protein dan deretan asam amino yang menyusunnya.

Untuk data PPI kelas negatif, yaitu pasangan protein yang tidak berinteraksi, diperoleh dari basis data LR-PPI[4]. Gugus data tersebut telah memuat data PPI beserta sekuens proteinnya [6].

B. Praproses data

Praproses data dilakukan dengan terlebih dahulu mengintegrasikan data PPI dengan data sekuens protein. Pada tahapan ini, gugus data PPI yang memuat pasangan ID protein disubstitusi menjadi sekuens protein dengan merujuk pada data sekuens protein. Selanjutnya, dilakukan pengecekan apakah setiap sekuens pada setiap pasangan protein tersebut hanya mengandung asam amino yang termasuk dalam 20 kode IUPAC (Tabel 3). Selain 20 kode asam amino tersebut, terdapat 6 kode asam lain yang mungkin terdapat dalam gugus data. Kode tersebut di antaranya adalah B, J, O, U, X, dan Z seperti yang terdapat pada Tabel 4.

Jika ditemukan pasangan PPI di mana salah satu proteinnya mengandung salah satu dari enam asam amino tersebut, pasangan PPI tersebut akan dihapus dari gugus data. Keenam asam amino tersebut merupakan asam amino yang masih ambigu seperti asam amino B, J, X, dan Z, atau asam amino yang baru ditemukan seperti asam amino O dan U yang belum diketahui nilai atribut *physicochemical*-nya [25]. Akibatnya, ekstraksi fitur *autocorrelation* tidak dapat dilakukan terhadap protein tersebut.

Setelah itu, karena jumlah data negatif jauh lebih besar dari data positif, dilakukan proses *random undersampling* agar data seimbang. Data dengan sebaran kelas yang tidak seimbang dapat menyebabkan model prediksi yang dibangun menjadi kurang baik. *Undersampling* digunakan untuk menyeimbangkan gugus data dengan sebaran kelas yang tidak seimbang, di mana suatu kelas memiliki memiliki jumlah data yang jauh lebih besar dibandingkan data pada kelas lain. *Random undersampling* dilakukan dengan mengeliminasi data dengan kelas yang dominan secara acak sampai kedua kelas memiliki jumlah data yang seimbang [26].

Gugus data yang telah berupa pasangan sekuens protein dari kelas positif dan negatif dengan jumlah data yang seimbang selanjutnya digabungkan menjadi sebuah gugus data. Penggabungan dilakukan secara acak untuk memastikan sebaran kelas pada gugus data tersebut merata.

C. Ekstraksi fitur

Setelah dilakukan praproses data, masing-masing pasangan sekuens protein diekstraksi untuk memperoleh fitur yang diperlukan. Ada beberapa metode yang bisa dilakukan untuk mengekstraksi fitur sekuens protein. Salah satu metode ekstraksi fitur yang banyak digunakan adalah metode *autocorrelation*. *Autocorrelation* digunakan untuk mengukur korelasi antar objek dalam suatu data terurut berdasarkan sifat atau atribut tertentu untuk mengetahui seberapa besar keterkaitan antar objek dalam data tersebut [14]. Dalam kasus ekstraksi sekuens protein, *autocorrelation* berguna untuk menghitung korelasi antar asam amino dalam setiap sekuens protein berdasarkan atribut *physicochemical* tertentu. Atribut *physicochemical* merupakan sifat fisika-kimia yang melekat pada suatu molekul. Sebagai sebuah molekul, asam amino juga memiliki atribut *physicochemical* yang berbeda satu sama lain.

Terdapat tiga metode penghitungan fitur autocorrelation, yaitu Moran, Geary, dan Moreau-Broto autocorrelation [27]. Moran autocorrelation didefinisikan dalam (1) [28], Geary dalam (2) [29], sedangkan Moreau-Broto dalam (3) [30], di mana $j=1,2,\ldots,n$, $lag=1,2,\ldots,m$, N merupakan panjang sekuens protein, $P_{i,j}$ dan $P_{i+lag,j}$ menyatakan nilai atribut ke-j asam amino pada posisi ke-i dan $i+lag$ pada suatu sekuens protein, $\overline{P}_j$ adalah rerata nilai atribut ke-j asam amino dari posisi 1 sampai N pada sekuens protein, sedangkan *lag* adalah jarak antara suatu asam amino dengan asam amino lainnya. Nilai maksimum parameter *lag* adalah panjang sekuens terpendek pada gugus data yang digunakan.

$$M(j,lag)=\frac{\frac{1}{N-lag}\sum_{i=1}^{N-lag}\left(P_{i,j}-\overline{P}_j\right)\left(P_{i+lag,j}-\overline{P}_j\right)}{\frac{1}{N}\sum_{i=1}^{N}\left(P_{i,j}-\overline{P}_j\right)} \quad (1)$$

$$G(j,lag)=\frac{\frac{1}{2(N-lag)}\sum_{i=1}^{N-lag}\left(P_{i,j}-P_{i+lag,j}\right)^2}{\frac{1}{N-1}\sum_{i=1}^{N}\left(P_{i,j}-\overline{P}_j\right)^2} \quad (2)$$

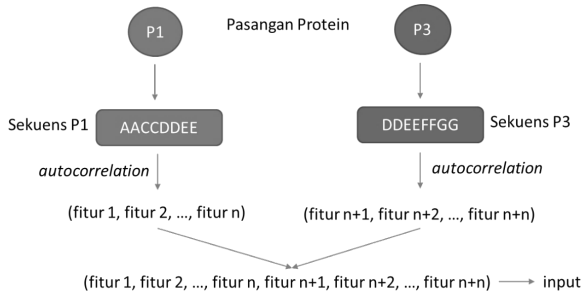
$$MB(j,lag)=\frac{1}{N-lag}\sum_{i=1}^{N-lag}P_{i,j}P_{i+lag,j} \quad (3)$$

Secara matematis, ketiga metode tersebut memiliki formulasi berbeda dalam menghitung fitur *autocorrelation*. Jika Moran menggunakan kovarian dari nilai atribut asam amino dengan *lag*-nya, Geary menghitung fitur *autocorrelation* menggunakan kuadrat

Tabel 4. Asam amino yang tidak termasuk ke dalam 20 kode IUPAC

Kode 3-huruf	Kode IUPAC	Asam Amino
B	Ala	Alanine
J	Cys	Cysteine
O	Asp	Aspartic Acid
U	Glu	Glutamic Acid
X	Phe	Phenylalanine
Z	Gly	Glycine

[4]http://www.csbio.sjtu.edu.cn/bioinf/LR_PPI/Data.htm



Gambar 4. Skema ekstraksi fitur menggunakan metode *autocorrelation*

dari selisih atribut asam amino dengan *lag*-nya pada sebuah sekuens. Penghitungan fitur *autocorrelation* pada Moreau-Broto didasari perkalian nilai atribut antar asam amino dengan *lag*-nya dalam suatu sekuens protein [14]. Nilai *autocorrelation* yang dihasilkan ketiga metode tersebut juga memiliki rentang yang berbeda-beda. Nilai *autocorrelation* yang dihasilkan Moran berada pada rentang [-1, 1], pada Geary nilainya berkisar [0, ∞], sedangkan nilai yang dihasilkan Moreau-Broto berada pada rentang [-∞, ∞]. Jika nilai korelasinya mendekati 0, hubungan antar asam amino dalam suatu sekuens dapat dikatakan semakin rendah. Sebaliknya, hubungan antar asam amino dalam suatu sekuens dapat dikatakan tinggi jika nilainya menjauhi 0.

Menurut Guo dkk. [9], terdapat empat macam interaksi yang terjadi antara dua protein, yaitu interaksi elektrostatis, interaksi *hydrophobic*, interaksi *steric* (berkaitan dengan susunan atom), dan ikatan hidrogen. Oleh karena itu, dalam penelitian ini ada enam atribut *physicochemical* yang dapat menggambarkan jenis-jenis interaksi tersebut, yaitu *hydrophobicity*, *volume of side chains*, *polarity*, *polarizability*, *solvent accessible surface area*, dan *net charge index of side chains*.

Hydrophobicity (H) adalah sifat fisis suatu asam amino untuk menolak air [31], [32], *volume of side chains* (VSC) merupakan volume rantai samping asam amino yang mengikat asam amino lain [33], *polarity* (P1) menyatakan kepolaran atau kemampuan setiap asam amino untuk memisahkan muatan listrik [34], *polarizability* (P2) berkaitan dengan interaksi antara elektron asam amino dan nukleus sel tempat protein berada [35], *solvent accessible surface area* (SASA) adalah luas area permukaan asam amino yang dapat dijangkau oleh molekul lain [36], dan *net charge index of side chains* (NCISC) sesuai [37]. Keenam atribut *physicochemical* tersebut memiliki nilai seperti yang tercantum pada Tabel 5.

Sebelum digunakan pada (1)-(3), nilai atribut *physicochemical* tersebut terlebih dahulu distandarisasi menggunakan (4) [14], di mana P_i merupakan atribut *physicochemical* ke-*j* asam amino ke-*i*, sedangkan $\overline{P}$ dan σ_j adalah rerata dan simpangan baku nilai atribut *physicochemical* ke-*j* dari 20 asam amino.

Tabel 5. Nilai atribut *physicochemical* pada setiap asam amino

Asam Amino	H	VSC	P1	P2	SASA	NCISC
A	0,62	27,5	8,1	0,046	1,181	0,007187
C	0,29	44,6	5,5	0,128	1,461	-0,03661
D	-0,9	40	13	0,105	1,587	-0,02382
E	-0,74	62	12,3	0,151	1,862	0,006802
F	1,19	115,5	5,2	0,29	2,228	0,037552
G	0,48	0	9	0	0,881	0,179052
H	-0,4	79	10,4	0,23	2,025	-0,01069
I	1,38	93,5	5,2	0,186	1,81	0,021631
K	-1,5	100	11,3	0,219	2,258	0,017708
L	1,06	93,5	4,9	0,186	1,931	0,051672
M	0,64	94,1	5,7	0,221	2,034	0,002683
N	-0,78	58,7	11,6	0,134	1,655	0,005392
P	0,12	41,9	8	0,131	1,468	0,239531
Q	-0,85	80,7	10,5	0,18	1,932	0,049211
R	-2,53	105	10,5	0,291	2,56	0,043587
S	-0,18	29,3	9,2	0,062	1,298	0,004627
T	-0,05	51,3	8,6	0,108	1,525	0,003352
V	1,08	71,5	5,9	0,14	1,645	0,057004
W	0,81	145,5	5,4	0,409	2,663	0,037977
Y	0,26	117,3	6,2	0,298	2,368	0,023599

$$P'_{ij} = \frac{P_{ij} - \overline{P}_j}{\sigma_j} \quad (4)$$

Banyaknya fitur yang dihasilkan oleh ketiga metode tersebut ditentukan oleh banyaknya atribut *physicochemical* dan parameter *lag* yang digunakan. Menurut Xia dkk. [12] dan You dkk. [13] parameter *lag* yang paling banyak digunakan untuk memperoleh fitur *autocorrelation* adalah 30. Oleh karena itu, penelitian ini menggunakan parameter *lag* sebesar 30 dengan 6 atribut *physicochemical* (Tabel 5). Gambar 4 merupakan ilustrasi tentang proses yang terjadi pada tahapan ekstraksi fitur menggunakan metode *autocorrelation* pada sepasang protein, yaitu P1 dan P3.

D. Prediksi PPI

Dalam penelitian ini, setiap ekstraksi fitur *autocorrelation* dari metode Moran, Geary, dan Moreau-Broto digunakan sebagai input untuk membuat model KNN, RF, dan SVM. Untuk itu data dibagi menjadi latih dan data uji dengan 5-*fold cross validation* (5-*fold* CV) atau validasi silang. Validasi silang dilakukan untuk menghindari *overfitting* dan generalisasi data [38]. Data dibagi secara acak ke dalam 5 partisi, kemudian dilakukan 5 kali percobaan. Dalam setiap percobaan data yang sudah dipartisi tersebut dibagi menjadi data latih dan data uji. Data latih digunakan sebagai data untuk melatih model pada algoritme KNN, RF, dan SVM. Setelah dilakukan pelatihan model, model tersebut digunakan untuk memprediksi kelas data uji untuk kemudian dibandingkan dengan kelas data uji yang sebenarnya.

KNN merupakan salah satu algoritme yang banyak digunakan untuk permasalahan klasifikasi dan regresi. Sebagai algoritme klasifikasi, KNN menunjukkan kinerja yang sangat baik pada data yang berukuran besar [39]. Berbeda dengan metode klasifikasi pada umumnya, KNN tidak memerlukan proses pelatihan

model untuk melakukan prediksi terhadap data baru [40]. KNN hanya memperhitungkan kedekatan suatu data baru (data uji) dengan data yang sudah ada (data latih) untuk menentukan kelas dari data tersebut [41]. Pertama, KNN menghitung jarak dari suatu data uji dengan semua data yang sudah ada. Selanjutnya, KNN memilih sejumlah k data latih terdekat.

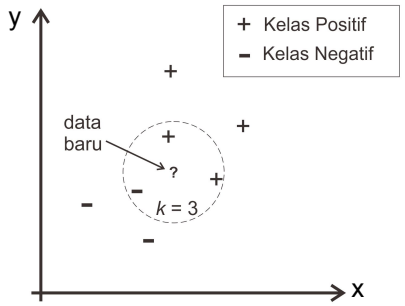
Penentuan kelas dari data uji tersebut dengan sistem kelas mayoritas, yaitu kelas dengan kemunculan terbanyak akan dipilih sebagai kelas dari data uji tersebut. Namun, jika terdapat beberapa kelas dengan jumlah *voting* yang sama, akan dipilih salah satu kelas secara acak. Gambar 5 merupakan ilustrasi penentuan k terdekat pada KNN dengan nilai $k=3$.

Kinerja dari KNN ditentukan oleh nilai k [42]. Nilai k yang terlalu kecil dapat menyebabkan hasil dari KNN terlalu kaku. Sebaliknya, jika k terlalu besar akan mempersulit proses *voting* karena terlalu banyak data yang dilibatkan rentan terhadap derau. Oleh karena itu, dalam penelitian ini akan dilakukan pengujian KNN pada beberapa nilai k yaitu 1, 3, dan 5 untuk dapat memperoleh hasil prediksi yang optimum. Adapun dalam penentuan jarak untuk mencari k terdekat, salah satu metode yang banyak digunakan adalah jarak Euclidean, seperti pada (5), di mana $p=(p_1,p_2,\ldots,p_m)$ dan $q=(q_1,q_2,\ldots,q_m)$ adalah data latih dan data uji, p_i dan q_i merupakan variabel data latih dan data uji ke-i, serta m adalah jumlah variabel atau fitur data.

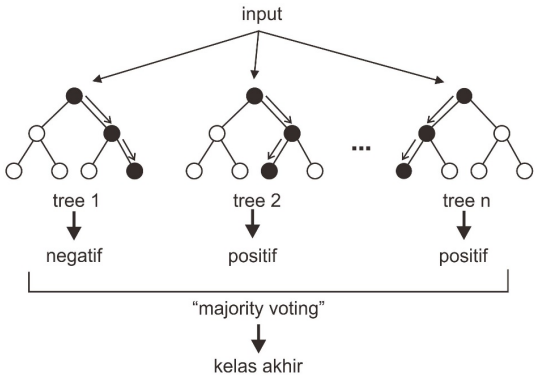
$$d(p,q)=\left(\sum_{i=1}^{m}(p_i-q_i)^2\right)^{\frac{1}{2}} \quad (5)$$

RF adalah algoritme klasifikasi yang terdiri dari beberapa *decision tree* sebagai sebuah model prediksi yang dapat divisualisasikan sebagai sebuah pohon. Dalam mengklasifikasikan data, sebuah *decision tree* memutuskan kelas sebuah data baru dengan membagi variabel data latih menjadi beberapa sub himpunan yang lebih kecil [19]. *Decision tree* pada RF dibentuk dengan metode *bootstrap resampling* [18], yaitu penarikan contoh acak pada data latih untuk membentuk n-*tree* dengan variabel yang juga diambil secara acak berukuran *m-try*. Proses tersebut diulang berkali-kali sampai terbentuk beberapa *tree* yang membentuk *forest*. Seperti halnya pada KNN, penentuan kelas data uji pada RF juga ditentukan oleh *majority voting*. Kelas dengan kemunculan terbanyak dari semua tree akan dipilih sebagai kelas dari data baru [6]. Gambar 6 merupakan ilustrasi pembentukan *tree* dan penentuan kelas pada RF.

Pembentukan *tree* pada model RF dalam penelitian ini menggunakan metode *GINI Index*. Perhitungan nilai *GINI Index* dinyatakan dalam (6) [18], di mana v menyatakan node pada *tree*, $p(i|v)$ merupakan peluang kelas i berada pada node v, sedangkan m adalah banyaknya kelas. Karena dalam penelitian ini menggunakan dua kelas, nilai m adalah 2. Hasil prediksi model RF sangat dipengaruhi oleh penentuan nilai parameter *n-tree* dan *m-try* [6]. Untuk itu, pada



Gambar 5. Ilustrasi penentuan k terdekat pada KNN



Gambar 6. Ilustrasi algoritme *random forest*

percobaan ini akan dilakukan pada beberapa nilai *n-tree,* yaitu 10, 30, dan 50 dengan nilai parameter *m-try* 5.

$$GINI(v)=1-\sum_{i=0}^{m-1}(p(i|v))^2 \quad (6)$$

SVM sering dianggap sebagai algoritme yang lebih baik dibandingkan algoritme lain untuk menyelesaikan permasalahan klasifikasi dalam berbagai kasus [42]. Ide dasar dari SVM adalah membangun *hyperplane* untuk memisahkan data berdasarkan kelasnya [43]. SVM membentuk *hyperplane* dari data latih yang sedemikian rupa sehingga dapat memprediksi kelas dari data uji. Misalkan terdapat sebuah gugus data $\{x_i,y_i\}$ dengan dua kelas di mana x_i adalah vektor atribut atau fitur data ke-i dan $y_i\in\{-1,\ 1\}$ merupakan kelas data ke-i, *hyperlane* pada gugus data tersebut dapat didefinisikan pada (7) [44], di mana w adalah vektor yang tegak lurus dengan *hyperplane* dan b adalah bias. Dengan demikian, sebuah data baru dianggap termasuk ke dalam kelas 0 jika $w\cdot x_i+b\leftarrow 1$ dan termasuk ke dalam kelas 1 jika $w\cdot x_i+b>1$.

$$w\cdot x_i+b=0 \quad (7)$$

Hyperplane yang baik ditempatkan di antara kelas dengan mengoptimumkan jarak antar dirinya dengan vektor-vektor terluar (*support vector*) dari masing-masing kelas. Jarak antara *hyperplane* dan *support vector* disebut sebagai margin. Karena w tegak lurus dengan *hyperplane*, maka margin dalam sebuah SVM

dapat dirumuskan sebagai $\|w\|^{-1}$. Gambar 7 merupakan ilustrasi penentuan *hyperplane* SVM pada data berdimensi dua.

Hyperplane yang optimal dapat diperoleh dengan menyelesaikan permasalahan optimasi pada (8) dengan kendala seperti pada (9) [44], di mana C merupakan parameter penalti yang dikenakan pada tiap kesalahan klasifikasi dan ξ_i adalah peubah *slack* yang menyatakan banyaknya data yang diperbolehkan berada dalam margin.

$$min\frac{1}{2}\|w\|^2+C\sum_{i=1}^{n}\xi_i;\xi_i\geq 0 \quad (8)$$

$$y_i\left(w^T\cdot x_i+b\right)\geq 1-\xi_i;\xi_i\geq 0 \quad (9)$$

Pada penelitian ini, model SVM dibentuk menggunakan kernel *radiant basis function* (RBF) yang dinyatakan dalam (10) [44], di mana x_i merupakan data latih kelas positif, sedangkan x_j adalah data latih dengan kelas negatif. Penentuan parameter optimum menggunakan metode *grid search* dengan mencari nilai γ terbaik di sekitar $1/n$, dengan n adalah jumah total fitur, yaitu $\{1/n, 2/n\}=\{1/360, 2/360\}$ serta C di sekitar nilai {1, 2, 3}.

$$K\left(x_i,x_j\right)=e^{-\gamma\left|x_i-x_j\right|^2},\gamma>0 \quad (10)$$

E. Evaluasi

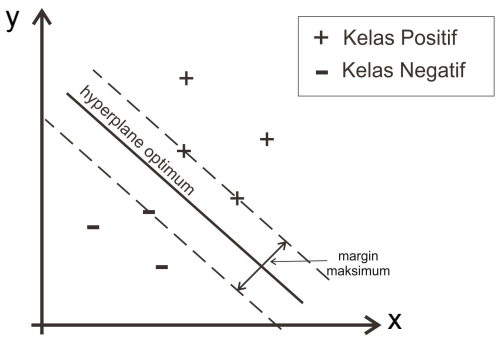
Hasil dari prediksi masing-masing model pada setiap *fold* kemudian dievaluasi untuk mengetahui tingkat akurasi, spesifisitas, sensitivitas, dan presisi yang didefinisikan pada (11)-(14) [45], di mana TP (*true positive*) merupakan banyaknya pasangan protein dengan kelas positif yang berhasil diprediksi, TN (*true negative*) adalah jumlah pasangan protein dengan kelas negatif yang berhasil diprediksi, FP (*false positive*) adalah banyaknya pasangan protein kelas negatif yang diprediksi sebagai positif, sedangkan FN (*false negative*) merupakan jumlah pasangan protein dengan kelas positif yang diprediksi sebagai kelas negatif. Akurasi menunjukkan persentase ketepatan hasil prediksi pada dua kelas, spesifisitas adalah persentase pasangan protein dengan kelas negatif yang diprediksi secara tepat, sensitivitas adalah pasangan protein dengan kelas positif yang berhasil diprediksi, sedangkan presisi menunjukkan nilai hasil prediksi pada kelas positif.

$$Akurasi=\frac{TP+TN}{TP+TN+FP+FN} \quad (11)$$

$$Spesifisitas=\frac{TN}{TP+TN} \quad (12)$$

$$Sensitivitas=\frac{TP}{TP+FN} \quad (13)$$

$$Presisi=\frac{TP}{TP+FP} \quad (14)$$



Gambar 7. Ilustrasi penentuan *hyperplane* pada SVM

Tabel 6. Perbandingan jumlah protein sebelum dan sesudah praproses

	Jumlah Pasangan Protein	
	Kelas Positif	**Kelas Negatif**
Sebelum Praproses	68.628	36.480
Sesudah Praproses	32.364	32.364

III. HASIL DAN PEMBAHASAN

A. Data PPI dan sekuens protein

Dari basis data IJAH dan STRING, diperoleh sebuah gugus data yang berisi ID pasangan protein yang saling berinteraksi (PPI kelas positif). Jumlah pasangan protein yang diperoleh adalah 68.628 pasang. Adapun dari basis data UniProt, diperoleh sebuah gugus data yang memuat ID protein yang terlibat dalam PPI kelas positif beserta sekuensnya yang terdiri dari 3.268 protein. Dari basis data LR-PPI diperoleh dua gugus data, yaitu gugus data ID pasangan protein yang saling tidak berinteraksi (PPI kelas negatif) yang terdiri dari 36.480 pasang protein serta gugus data yang berisi ID dan sekuens protein yang terlibat sejumlah 11.205 protein.

B. Praproses data

Dalam data yang telah dihimpun, ditemukan beberapa pasangan protein yang mengandung asam amino yang tidak termasuk ke dalam 20 kode IUPAC, yaitu asam amino yang terdapat pada Tabel 4. Selain itu, terdapat pasangan protein yang tidak ditemukan sekuens proteinnya. Oleh karena itu, pasangan protein dengan dua kondisi tersebut dihapus dari gugus data. Tabel 6 menunjukkan perbandingan jumlah pasangan protein sebelum dan sesudah dilakukan praproses data.

Setelah dilakukan praproses, data kelas positif dan negatif digabungkan sehingga diperoleh sebuah gugus data PPI dengan jumlah 32.364 pasang protein pada masing-masing kelas atau sejumlah 64.728 pasang pada keseluruhan gugus data.

Tabel 7. Hasil prediksi PPI dengan beragam metode *autocorrelation* dan *machine learning*

Algoritme	Metode *Autocorrelation*	Parameter	*Fold* ke-	Akurasi (%)	Spesifisitas (%)	Sensitivitas (%)	Presisi (%)
KNN	Moran	$k = 1$	1	94,18	95,29	93,08	95,24
			2	94,62	95,78	93,45	95,61
			3	94,32	94,91	93,73	94,81
			4	94,36	95,36	93,35	95,21
			5	94,65	96,17	93,17	96,14
			Rataan	94,41±0,30	93,30±0,49	95,53±0,23	93,44±0,44
		$k = 3$	Rataan	95,08±0,23	94,06±0,46	96,11±0,21	94,18±0,37
		$k = 5$	Rataan	95,31±0,19	94,24±037	**96,37±018**	94,36±0,30
	Geary	$k = 1$	Rataan	94,43±0,18	95,50±0,43	93,36±0,23	95,40±0,45
		$k = 3$	Rataan	95,13±0,21	96,07±0,34	94,19±0,24	95,99±0,34
		$k = 5$	Rataan	**95,34±0,22**	**96,39±0,37**	94,29±0,09	**96,31±0,39**
	Moreau-Broto	$k = 1$	Rataan	94,02±0,17	95,28±0,12	92,78±0,32	95,15±0,12
		$k = 3$	Rataan	94,55±0,08	95,73±0,11	93,37±0,17	95,62±0,15
		$k = 5$	Rataan	94,55±0,13	95,95±0,14	93,16±0,24	95,83±0,17
Random Forest	Moran	*n-tree* = 10	Rataan	96,20±0,15	97,01±0,25	95,39±0,28	96,96±0,24
		n-tree = 30	Rataan	97,26±0,10	97,79±0,18	96,71±0,19	97,77±0,17
		n-tree = 50	Rataan	97,41±0,14	**97,95±0,19**	96,85±0,20	**97,93±0,18**
	Geary	*n-tree* = 10	Rataan	96,26±0,19	96,17±0,24	96,34±0,14	96,18±0,28
		n-tree = 30	Rataan	97,22±0,08	96,91±0,11	97,52±0,12	96,93±0,12
		n-tree = 50	Rataan	**97,43±0,11**	97,10±0,15	**97,76±0,10**	97,12±0,16
	Moreau-Broto	*n-tree* = 10	Rataan	96,02±0,25	95,84±0,30	96,19±0,27	95,85±0,31
		n-tree = 30	Rataan	96,97±0,14	96,65±0,09	97,28±0,25	96,67±0,12
		n-tree = 50	Rataan	97,29±0,11	96,99±0,10	97,60±0,13	97,00±0,09
SVM	Moran	(*grid*	Rataan	**97,19±0,11**	**97,44±0,24**	96,94±0,08	**97,43±0,27**
	Geary	*search*)	Rataan	**97,19±0,20**	97,11±0,32	**97,26±0,30**	97,12±0,33
	Moreau-Broto		Rataan	96,12±0,65	96,26±0,43	95,99±0,88	96,25±0,46

C. Ekstraksi fitur

Dalam bahasa pemrograman R, proses ekstraksi fitur dapat dilakukan dengan pustaka *protr cran*[5] yang telah menyediakan perintah untuk mengekstraksi baik fitur Moran, Geary, maupun Moreau-Broto dari sebuah sekuens protein. Karena dalam penelitian ini menggunakan parameter *lag* 30 dan 6 atribut *physicochemical*, melalui Persamaan (1)-(3) masing-masing metode *autocorrelation* akan merepresentasikan setiap sekuens protein oleh vektor sepanjang 180 (30×6) sehingga setiap pasang protein direpresentasikan oleh vektor sepanjang 360 yang dijadikan sebagai masukan untuk setiap *machine learning* pada penelitian ini.

D. Prediksi PPI dan evaluasi

Tabel 7 menunjukkan hasil prediksi PPI yang dihasilkan oleh tiga metode *autocorrelation* pada tiga algoritme *machine learning*. Dari hasil prediksi menggunakan KNN, prediksi dengan nilai akurasi terbaik diperoleh melalui fitur yang dihasilkan metode Geary pada *k* bernilai 5 dengan rerata akurasi sebesar 95,34%. Pada KNN dengan fitur Moran, hasil prediksi terbaik juga diperoleh pada *k* bernilai 5 dengan rerata akurasi sebesar 95,31%. Adapun melalui fitur Moreau Broto, hasil prediksi terbaik diperoleh pada k bernilai 3 dan 5 dengan rerata akurasi sebesar 94,55%. Meskipun demikian, hasil prediksi KNN pada tiga fitur tersebut dapat dikatakan sebanding karena perbedaan ketiganya tidak signifikan.

Prediksi yang dihasilkan algoritme RF menggunakan fitur Moran, Geary, dan Moreau-Broto menunjukkan hasil yang sangat baik secara akurasi, spesifisitas, sensitivitas, dan presisi. Dalam hal akurasi, metode Geary menunjukkan nilai akurasi yang sedikit lebih baik dengan rerata nilai 97,43% pada n-tree bernilai 50. Moran menghasilkan prediksi dengan rerata akurasi 97,41% pada nilai n-tree yang sama. Adapun Moreau-Broto menunjukkan hasil prediksi dengan tingkat akurasi rata-ratanya adalah 97,31% yang diperoleh pada nilai n-tree 50. Adapun percobaan dengan menggunakan SVM menunjukkan bahwa Moran memiliki hasil prediksi dengan nilai rerata akurasi yang sama persis dengan Geary yaitu 97,19%. Adapun hasil pengujian pada fitur Moreau-Broto menunjukkan hasil prediksi dengan rerata akurasi yang sedikit lebih rendah, namun tidak berbeda signifikan dengan Moran dan Geary, yaitu sebesar 96,12%.

Dari seluruh percobaan yang dilakukan, diketahui bahwa hasil prediksi menggunakan KNN, RF, dan SVM menunjukkan rerata akurasi yang tinggi baik pada fitur Moran, Geary, maupun Moreau-Broto. Rataaan akurasi yang diperoleh di atas 94% pada KNN, sedangkan pada RF dan SVM rata-rata akurasinya di atas 96%. Hal ini menunjukkan bahwa metode Moran, Geary, dan Moreau-Broto *autocorrelation* dapat digunakan untuk mengekstraksi fitur sekuens sepasang protein yang dapat digunakan untuk memprediksi interaksi antara kedua protein tersebut. Dengan kata lain, kelas PPI dari dua buah protein dapat diprediksi menggunakan nilai korelasi atau hubungan antar asam amino dalam

[5]Dapat diunduh dari http://cran.r-project.org/package=protr

masing-masing sekuens dari dua protein tersebut. Hal itu selaras dengan penelitian-penelitian sebelumnya, bahwa fitur *autocorrelation* dari sebuah protein dapat digunakan untuk memprediksi interaksi suatu protein dengan protein lainnya dengan baik.

Menurut Pevsner [11], interaksi antar protein atau PPI cenderung akan terjadi antara dua protein yang memiliki kesamaan tertentu. Dengan demikian, dari hasil prediksi yang telah dilakukan, diasumsikan bahwa sepasang protein akan akan dianggap sebagai pasangan PPI kelas positif jika memiliki kemiripan fitur *autocorrelation*. Untuk mengetahui karakteristik dari sekuens protein yang digunakan pada penelitian ini, dilakukan pengamatan terhadap salah satu pasangan protein yang tidak diprediksi secara tepat. Salah satu contoh pasangan protein yang salah diprediksi pada model RF dengan nilai *n-tree* 50 menggunakan fitur Geary *autocorrelation* diambil secara acak, yaitu pasangan protein 9606.ENSP00000260653 (*Homeobox protein SIX3*) dan 9606.ENSP00000295934 (*Homeobox expressed in ES cells 1*). Kedua protein tersebut merupakan protein dalam tubuh manusia yang berfungsi dalam perkembangan otak depan. Pasangan protein tersebut seharusnya berinteraksi dan berada pada kelas positif, namun diprediksi sebagai kelas negatif (*false negative*) oleh algoritme RF. Fitur Geary yang dihasilkan dari pasangan protein tersebut adalah [0.9674913 0.9332728 0.9703942 0.9591933 ... 1.089882]. Fitur tersebut merupakan vektor dengan panjang 360, di mana 180 fitur pertama diperoleh dari protein pertama sedangkan 180 fitur kedua diperoleh dari protein kedua. Ketika dibandingkan, fitur dari kedua protein tersebut memiliki keragaman sebesar 0.001883344. Nilai tersebut jauh lebih besar dibandingkan pasangan-pasangan PPI lain dengan kelas positif. Hal tersebut juga terjadi pada pasangan *false positive*, di mana pasangan yang seharusnya berada pada kelas negatif namun diklasifikasikan ke dalam kelas positif oleh *machine learning* memiliki keragaman fitur *autocorrelation* antar pasangan-pasangan tersebut cenderung rendah dibandingkan pasangan kelas negatif yang lainnya.

Ada beberapa kemungkinan mengapa hal tersebut dapat terjadi. Pertama, keenam atribut *physicochemical* yang digunakan dalam penelitian ini kurang mampu merepresentasikan jenis interaksi antara kedua protein tersebut. Yang kedua, pada beberapa kasus kelas dari PPI memang tidak dapat diduga hanya dengan mengandalkan informasi *autocorrelation* atau korelasi antar asam amino yang menyusun kedua protein tersebut. Meskipun demikian, fitur *autocorrelation* tetap merupakan fitur yang sangat baik untuk memprediksi PPI karena memiliki akurasi yang tinggi ketika diuji coba pada beberapa algoritme klasifikasi *machine learning*.

IV. KESIMPULAN

Pada penelitian ini, metode Moran, Geary, dan Moreau-Broto *autocorrelation* berhasil diterapkan untuk mengekstraksi fitur sekuens protein pada PPI. Fitur yang dihasilkan oleh ketiga metode tersebut dapat digunakan untuk memprediksi PPI pada beberapa algoritme *machine learning*, yaitu KNN, RF, dan SVM. Diketahui bahwa hasil prediksi menggunakan fitur yang dihasilkan Moran, Geary, dan Moreau-Broto dapat menghasilkan prediksi dengan tingkat akurasi yang tinggi. Pada KNN nilai rerata akurasi tertingginya tidak kurang dari 94% baik pada fitur Moran, Geary, maupun Moreau-Broto. Sementara itu, RF dan SVM menunjukkan hasil prediksi yang sedikit lebih baik, di mana rata-rata akurasi tertinggi mencapai 97% pada hampir semua fitur, kecuali fitur Moreau-Broto pada SVM sebesar 96,12%. Hal ini menunjukkan bahwa *autocorrelation* adalah fitur yang sangat baik untuk memprediksi PPI pada algoritme KNN, RF, dan SVM.

Selain itu, dari penelitian ini diketahui bahwa interaksi antar protein juga cenderung terjadi pada dua protein yang fitur *autocorrelation*-nya yang mirip atau memiliki keragaman yang rendah. Dengan demikian, kelas PPI dari sepasang protein dapat diprediksi menggunakan fitur *autocorrelation* atau nilai korelasi antar asam amino dalam masing-masing sekuens dari dua protein tersebut.

Untuk meningkatkan hasil prediksi perlu ditambahkan atribut *physicochemical* lain yang mungkin dapat merepresentasikan interaksi antar protein. Selain itu, penelitian ini menggunakan KNN, RF, dan SVM pada beberapa parameter yang terbatas sehingga perlu dilakukan pengujian pada parameter yang lebih banyak untuk memperoleh hasil prediksi yang lebih baik.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada Kementerian Riset, Teknologi, dan Pendidikan Tinggi yang telah memberikan bantuan melalui program hibah Penelitian Dasar Unggulan Perguruan Tinggi (PDUPT) tahun 2019.

DAFTAR PUSTAKA

[1] A. Athanasios, V. Charalampos, T. Vasileios, and G. M. Ashraf, "Protein-Protein Interaction (PPI) network: recent advances in drug discovery", *Current Drug Metabolism*, vol. 18, pp. 5-10, 2017.

[2] J. D. L. Rivas and C. Fontanillo, "Protein-protein interactions essentials: key concepts to building and analyzing interactome networks", *PLOS Computational Biology*, vol. 6, 2010.

[3] S. Jones and J. M. Thornton, "Principles of protein-protein interactions", *Proceedings of the National Academy of Sciences of the United States of America*, vol. 93, pp. 13-20, 1996.

[4] Z. H. You Y. K. Lei, L. Zhu, J. Xia, and B. Wang. 2013, "Prediction of protein-protein interactions from amino acid sequences with ensemble extreme learning machines and principal component analysis", *BMC Bioinformatics*, vol. 14, 2013.

[5] L. Skrabanek, H. K. Saini, G. D. Bader, and A. J. Enright, "Computational prediction of protein-protein interactions", *Molecular Biotechnology*, vol. 38, pp. 1-17, 2008.
[6] X. Y. Pan, Y. N. Zhang, and H. B. Shen. 2010. "Large-Scale prediction of human protein-protein interactions from amino acid sequence based on latent topic features", *Journal of Proteome Research*, vol. 9, pp. 4992-5001, 2010.
[7] J. Shen, J. Zhang, X. Luo, W. Zhu, K. Yu, K. Chen, Y. Li, and H. Jiang, "Predicting protein-protein interactions based only on sequences information", *Proceedings of the National Academy of Sciences of the United States of America*, vol. 104, pp. 4337-4341, 2007.
[8] H. S. Stoker, *Organic and Biological Chemistry,* 7th ed. Boston, US: Cengage Learning, 2015.
[9] Y. Z. Guo, L. Z. Yu, Z. N. Wen, and M. L. Li, "Using support vector machine combined with auto covariance to predict protein-protein interactions from protein sequences", *Nucleic Acids Research*, vol. 36, pp. 3025-3030, 2008.
[10] L. Yang, J. F. Xia, and J. Gui, "Prediction of protein-protein interactions from protein sequence using local descriptors", *Protein & Peptide Letters*, vol. 17, pp. 1085-1090, 2010.
[11] J. Pevsner, *Bioinformatics and Functional Genomics*, 2nd ed. New Jersey, US: John Wiley & Sons, 2009.
[12] J. Xia, K. Han, and D. Huang, "Sequence-Based prediction of protein-protein interactions by means of rotation forest and autocorrelation descriptor", *Protein & Peptide Letters*, vol. 17, pp. 137-145, 2010.
[13] Z. You, J. Yu, L. Zhu, S. Li, and Z. Wen, "Neurocomputing a MapReduce based parallel SVM for large-scale predicting protein-protein interactions, *Neurocomputing*, vol. 145, pp. 37-43, 2014.
[14] S. A. K. Ong, H. H. Lin, Y. Z. Chen, Z. R. L, and Z. Cao, "Efficacy of different protein descriptors in predicting protein functional families", *BMC Bioinformatics*, vol. 14, pp.1-14, 2007. doi: 10.1186/1471-2105-8-300
[15] L. Lan, N. Djuric, Y. Guo, and S. Vucetic, "MS-*k*NN: protein function prediction by integrating multiple data sources", *BMC Bioinformatics*, vol. 14, 2013.
[16] R. M. Parry *et al*., "k-Nearest neighbor models for microarray gene expression analysis and clinical outcome prediction", *The Pharmacogenomics Journal*, vol. 10, pp. 292-309, 2010.
[17] G. Paul, R. Sua, M. Romaina, V. Sebastien, V. Pierre, and G. Isabellea, "Feature selection for outcome prediction in oesophageal cancer using genetic algorithm and random forest classier", *Computerized Medical Imaging and Graphics*, vol. 16, 2016
[18] S. H. Wijaya, I. Batubara, T. Nishioka, M. A. U. Amin, and S. Kanaya, "Metabolomic studies of indonesian jamu medicines: prediction of jamu efficacy and identification of important metabolites", *Molecular Informatics*, vol. 36, 2017.
[19] A. Subudhi, M. Dash, and S. Sabut. "Automated segmentation and classification of brain stroke using expectation-maximization and random forest classifier", *Biocybernetics and Biomedical Engineering*, pp.1-13, 2019.
[20] S. Huang, N. Cai, P. P. Pacheco, S. Narrandes, Y. Wang, and W. Xu, "Applications of support vector machine (SVM) learning in cancer genomics", *Cancer Genomics & Proteomics*, vol. 15, pp. 41-51, 2018.
[21] N. S. Ramadhanti, W. A. Kusuma, and R Heryanto, "Development of Jamu formula prediction system module of Ijah analytics based on pharmacology activity and particular efficacy target," in *IOP Conference Series: Earth and Environmental Science,* vol. 335, 012003, 2019.
[22] D. Szklarczyk et al., "STRING v10: protein-protein interaction networks, integrated over the tree of life", *Nucleic Acids Research*, vol. 43, pp. 447-452, 2015.
[23] [TUC] The UniProt Consortium, "UniProt: The Universal Protein Knowledgebase", *Nucleic Acids Research*, vol. 45, pp. 158–169, 2016.
[24] [IUPAC-IUB] International Union of Pure and Applied Chemistry Commission on Biochemical Nomenclature, "A one-letter notation for amino acid sequences: Tentative rules", *Biochemical Journal*, vol. 113, pp. 1-4, 1968.
[25] D. R. Flower, "On the utility of alternative amino acid scripts", *Bioinformation*, vol. 8, pp. 539–542, 2012. doi: 10.6026/97320630008539
[26] G. E. A. Batista, R. C. Prati, and M. C. Monard, "A study of the behavior of several methods for balancing machine learning training data", *SIGKDD Explore*, vol. 6, pp. 20-29, 2004.
[27] Y. Liang, S. Liu, and S. Zhang, "Prediction of protein structural class based on different autocorrelation descriptors of position-specific scoring matrix", *MATCH Communications in Mathematical and in Computer Chemistry*, vol. 73, pp. 765-784, 2015.
[28] P. A. P. Moran, "Notes on continuous stochastic phenomena", *Biometrika*, vol. 37, pp. 17-23, 1950.
[29] R. C. Geary, "The contiguity ratio and statistical mapping", *The Incorporated Statistician*, vol. 5, pp. 115-145, 1954.
[30] G. Moreau and P. Broto, "Autocorrelation of molecular structures application to SAR studies", *Nour J Chim,* vol. 4, pp. 757-767, 1980.
[31] C. Tanford, "Contribution of hydrophobic interactions to the stability of the globular conformation of proteins", *Journal of the American Chemical Society*, vol.84, pp. 4240-4247, 1962. doi: 10.1021/ja00881a009
[32] A. Ben-Naim, *Hydrophobic Interactions*, New York, US: Springer, 1980.

[33] W. R. Krigbaum and A. Komoriya, "Local interactions as a structure determinant for protein molecules: II", *Biochimica et Biophysica Acta (BBA)-Protein Structure*. vol. 576, pp. 204-228, 1979. doi: 10.1016/0005-2795(79)90498-7
[34] R. Grantham, "Amino acid difference formula to help explain protein evolution", *Science,* vol. 185, pp. 862-864, 1974. doi: 10.1126/science.185.4154.862
[35] M. Charton and B. I. Charton, "The structural dependence of amino acid hydrophobicity parameters", *Journal of Theoretical Biology*, vol. 99, pp. 629-644, 1982. doi: 10.1016/0022-5193(82)90191-6
[36] G. Rose, A. Geselowitz, G. Lesser, R Lee, and M. Zehfus, "Hydrophobicity of amino acid residues in globular proteins", *Science*, vol. 229, pp. 834-838, 1985. doi: 10.1126/science.4023714
[37] P. Zhou, F. F. Tian, B. Li, S. R. Wu, and Z. L. Li, "Genetic algorithm base virtual screening of combinative mode for peptide/protein", *Acta Chim Sinica*, vol. 64, pp. 691-697, 2006.
[38] M. W. Browne, "Cross-validation methods", *Journal of Mathematical Psychology*, vol. 44, pp. 108-132, 2000.
[39] S. Zhang, X. Li, M. Zong, X. Zhu, and D. Cheng, "Learning k for kNN classification", *ACM Trans Intell Syst Technol*, vol. 8, 2017.
[40] S. Zhang, X. Li, M. Zong, X. Zhu X, R. Wang, "Efficient kNN classification with different numbers of nearest neighbors", *IEEE Transactions on Neural Networks and Learning Systems,* vol. 29, pp. 1774-1785, 2018.
[41] T. M. Cover and P. E. Hart, "Nearest neighbor pattern classification", *IEEE Transactions on Information Theory*, vol. 13, pp. 21-27, 1967.
[42] X. Wu, V. Kumar, J. R. Quinlan, J. Ghosh, Q. Yang, H. Motoda, J. H. McLachlan, A. Ng, B. Liu, B. P. Yu, et al., "Top 10 algorithms in data mining", *Knowledge Information System*, vol. 14, pp. 1-37, 2008.
[43] R. Romero, E. L. Iglesias, L. Borrajo, "A Linear-RBF multikernel SVM to classify big text corpora", *BioMed Research International*, 2015.
[44] C. Cortes and V. Vapnik, "Support-vector network", *Machine Learning*, vol. 20, pp. 273-297, 1995.
[45] J. Han, M. Kamber, J. Pei, *Data Mining Concepts and Techniques,* 3rd ed. Waltham (US): Elsevier, 2012.