

repository.ub.ac.id

**KLASIFIKASI SENYAWA KIMIA DENGAN NOTASI *SIMPLIFIED  
MOLECULAR INPUT LINE ENTRY SYSTEM (SMILES)*  
MENGUNAKAN METODE *EXTREME LEARNING MACHINE*  
(ELM)**

**SKRIPSI**

Untuk memenuhi sebagian persyaratan  
memperoleh gelar Sarjana Komputer

Disusun oleh:

Isti Marlisa Fitriani

NIM: 155150200111087



PROGRAM STUDI TEKNIK INFORMATIKA  
JURUSAN TEKNIK INFORMATIKA  
FAKULTAS ILMU KOMPUTER  
UNIVERSITAS BRAWIJAYA  
MALANG  
2019

UNIVERSITAS  
BRAWIJAYA

# PENGESAHAN

**KLASIFIKASI SENYAWA KIMIA DENGAN NOTASI *SIMPLIFIED MOLECULAR INPUT LINE ENTRY SYSTEM (SMILES)* MENGGUNAKAN METODE *EXTREME LEARNING MACHINE (ELM)***

**SKRIPSI**

**Diajukan untuk memenuhi sebagian persyaratan  
memperoleh gelar Sarjana Komputer**

**Disusun Oleh:  
Isti Marlisa Fitriani  
NIM: 155150200111087**

**Skripsi ini telah diuji dan dinyatakan lulus pada  
15 April 2019  
Telah diperiksa dan disetujui oleh:**

**Dosen Pembimbing I**



**Dian Eka Ratnawati, S.Si, M.Kom  
NIP: 19730619 200212 2 001**

**Dosen Pembimbing II**



**Syaiful Anam, S.Si, MT, Ph.D  
NIP: 197801152002121003**

**Mengetahui**

**Ketua Jurusan Teknik Informatika**



**Astoto Kurniawan, S.T, M.T, Ph.D  
NIP: 19710518 200312 1 001**

## PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar referensi.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 15 April 2019



Isti Marlisa Fitriani

NIM: 155150200111087

## PRAKATA

Puji syukur kepada Allah SWT yang telah memberikan rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan skripsi yang berjudul “Klasifikasi Senyawa Kimia Dengan Notasi *Simplified Molecular Input Line Entry System* (SMILES) Menggunakan Metode *Extreme Learning Machine* (ELM)”.

Skripsi ini diajukan untuk memenuhi sebagian persyaratan mencapai gelar Sarjana Komputer (S.Kom) di Fakultas Ilmu Komputer (FILKOM) Universitas Brawijaya Malang. Dalam kesempatan ini penulis menyampaikan rasa terima kasih kepada pihak – pihak yang telah membantu penulis selama penyusunan skripsi ini, diantaranya:

1. Ibu Dian Eka Ratnawati, S.Si, M.Kom selaku Dosen Pembimbing Satu yang telah membimbing dan memberikan banyak saran kepada penulis sehingga penulis dapat menyelesaikan skripsi ini dengan baik.
2. Bapak Syaiful Anam, S.Si, MT, Ph.D selaku Dosen Pembimbing Dua yang telah membimbing dan memberikan banyak saran kepada penulis sehingga penulis dapat menyelesaikan skripsi ini dengan baik.
3. Bapak Wayan Firdaus Mahmudy, S.Si., M.T., Ph.D. selaku Dekan Fakultas Ilmu Komputer Universitas Brawijaya Malang.
4. Bapak Tri Astoto Kurniawan, S.T, M.T, Ph.D. selaku Ketua Jurusan Teknik Informatika Fakultas Ilmu Komputer Universitas Brawijaya Malang.
5. Bapak Agus Wahyu Widodo, ST. M.Cs. selaku Ketua Program Studi Teknik Informatika Fakultas Ilmu Komputer Universitas Brawijaya Malang.
6. Kedua orang tua dan keluarga besar yang telah memberikan dukungan, nasihat, kasih sayang, dan do’a yang tiada henti untuk melancarkan skripsi ini.
7. Teman-teman yang selalu memberikan semangat, motivasi, dan nasihat untuk kelancaran skripsi ini.
8. Seluruh sivitas akademika Teknik Informatika Universitas Brawijaya yang telah memberikan banyak bantuan dan dukungan.

Penulis menyadari masih banyak kekurangan dalam penyusunan skripsi ini, sehingga penulis mengharapkan saran dan kritik yang dapat menjadikan penulis lebih baik lagi. Semoga skripsi ini bermanfaat dan dapat dijadikan sebagai bahan acuan bagi pihak yang membutuhkan.

Malang, 15 April 2019

Penulis

istimarlisa@gmail.com

## ABSTRAK

Isti Marlisa Fitriani, Klasifikasi Senyawa Kimia Dengan Notasi *Simplified Molecular Input Line Entry System (SMILES)* Menggunakan Metode *Extreme Learning Machine (ELM)*

Pembimbing: Dian Eka Ratnawati, S.Si, M.Kom dan Syaiful Anam, S.Si, MT, Ph.D

Indonesia mempunyai potensi alam yang besar dengan ditemukannya berbagai spesies tanaman dan hewan. Hal ini menjadi keuntungan bagi masyarakat Indonesia dalam memanfaatkan kekayaan alamnya terutama dalam bidang farmakologi. Dalam bidang farmakologi, senyawa aktif mempunyai manfaat untuk pencegahan dan penyembuhan penyakit. Oleh karena itu, di dalam bidang informatika dilakukan penelitian dengan membuat sistem klasifikasi senyawa aktif untuk mengetahui manfaat farmakologinya. SMILES merupakan notasi senyawa kimia yang digunakan pada penelitian ini. Fitur SMILES yang digunakan sebanyak 15, yaitu atom B, C, N, O, P, S, F, Cl, Br, I, OH, @, =, #, dan (). ELM adalah metode JST yang dapat melakukan generalisasi dengan baik dalam waktu yang sangat cepat dibandingkan dengan metode *Support Vector Machine (SVM)* dan *Backpropagation*. Pengujian jumlah *hidden neuron* dengan metode *k-fold cross validation* pada 2 kelas menghasilkan akurasi terbaik 85,03% pada skenario kelas Metabolisme dan Radang dengan jumlah 5, 10, dan 15 *hidden neuron*. Pengujian jumlah *hidden neuron* dengan metode *k-fold cross validation* pada 3 kelas menghasilkan akurasi terbaik 55,06% pada skenario kelas Metabolisme, Radang, dan Kanker dengan jumlah 300 *hidden neuron*. Dihilangkan akurasi terbaik pada pengujian 15 fitur sebesar 55,06% dengan jumlah 300 *hidden neuron* sedangkan pada pengujian 11 fitur diperoleh akurasi terbaik sebesar 49,18% dengan jumlah 400 *hidden neuron*.

Kata kunci: senyawa aktif, SMILES, klasifikasi, ELM

## ABSTRACT

**Isti Marlisa Fitriani, *Classification of Chemical Compounds with Simplified Molecular Input Line Entry System (SMILES) Notation Using the Extreme Learning Machine (ELM) Method***

**Supervisors: Dian Eka Ratnawati, S.Si, M.Kom and Syaiful Anam, S.Si, MT, Ph.D**

*Indonesia has a huge natural's potential by the existence of various plants and animals discovery. This issue brings a good for Indonesian people through taking advantage of nature, especially in pharmacology. In pharmacology, active compounds can be used to prevent and cure diseases. Therefore, a research is conducted in informatics's field by making an active compounds' classification system to determine its pharmacological benefits. SMILES is a chemical compound notation used in this research. SMILES's features which are used as many as 15, namely B, C, N, O, P, S, F, Cl, Br, I, OH, @, =, #, and (). ELM is an ANN method that can do a generalization better than conventional methods in a limited time. A number of hidden neurons test which were conducted using k-fold cross validation method in 2 classes produced the best accuracy, 85,03%, in Metabolism and Inflammation class scenario with a total of 5, 10, and 15 hidden neurons. A number of hidden neurons' test using k-fold cross validation method which were conducted in 3 classes produced the best accuracy, 55,06%, in Metabolism, Inflammation, and Cancer class scenario with a total of 300 hidden neurons. The best accuracy was obtained as many as 55,06% by testing 15 features with 300 hidden neurons, while in 11 features's test with 400 hidden neurons was found a number of 49,18% as the best accuracy.*

**Keywords:** active compounds, SMILES, classification, ELM

## DAFTAR ISI

PENGESAHAN .....	ii
PERNYATAAN ORISINALITAS .....	iii
PRAKATA.....	iv
ABSTRAK.....	v
ABSTRACT .....	vi
DAFTAR ISI.....	vii
DAFTAR TABEL.....	x
DAFTAR GAMBAR.....	xii
DAFTAR KODE PROGRAM .....	xiii
<b>BAB 1 PENDAHULUAN.....</b>	<b>1</b>
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Tujuan .....	3
1.4 Manfaat.....	3
1.5 Batasan Masalah.....	3
1.6 Sistematika Pembahasan.....	3
<b>BAB 2 LANDASAN KEPUSTAKAAN .....</b>	<b>5</b>
2.1 Kajian Pustaka .....	5
2.2 Dasar Teori.....	6
2.2.1 Notasi <i>Simplified Molecular Input Line Entry System</i> (SMILES).....	7
2.2.2 Klasifikasi .....	7
2.2.3 Jaringan Saraf Tiruan (JST) .....	7
2.2.4 Fungsi Aktivasi Jaringan Saraf Tiruan (JST).....	8
2.2.5 <i>Extreme Learning Machine</i> (ELM) .....	9
2.2.5.1 Algoritme ELM .....	9
2.2.6 Evaluasi Kinerja Klasifikasi.....	10
<b>BAB 3 METODOLOGI PENELITIAN .....</b>	<b>12</b>
3.1 Tipe Penelitian .....	12
3.2 Strategi Penelitian.....	12
3.3 Objek dan Lokasi Penelitian.....	12



3.4 Pengumpulan Data .....	12
3.5 Teknik Analisis Data .....	13
3.6 Peralatan Pendukung.....	13
3.6.1 Spesifikasi Perangkat Keras.....	13
3.6.2 Spesifikasi Perangkat Lunak .....	13
BAB 4 PERANCANGAN.....	14
4.1 Deskripsi Sistem .....	14
4.2 Arsitektur ELM .....	14
4.3 Perancangan Sistem.....	15
4.3.1 <i>Preprocessing</i> Notasi SMILES .....	15
4.3.2 Proses ELM .....	16
4.3.2.1 <i>Training</i> ELM.....	16
4.3.2.2 <i>Testing</i> ELM .....	22
4.4 Perhitungan Manual .....	25
4.4.1 <i>Preprocessing</i> Notasi SMILES .....	25
4.4.2 Perhitungan <i>Training</i> ELM.....	26
4.4.3 Perhitungan <i>Testing</i> ELM .....	32
4.5 Perancangan Pengujian .....	37
4.5.1 Pengujian Jumlah <i>Hidden Neuron</i> dengan Metode <i>K-fold Cross Validation</i> .....	37
4.5.1.1 Pengujian 2 Kelas.....	37
4.5.1.2 Pengujian 3 Kelas.....	39
4.5.2 Pengujian Jumlah Fitur dengan Metode <i>K-fold Cross Validation</i> .....	40
4.5.2.1 Pengujian 11 Fitur .....	40
BAB 5 IMPLEMENTASI .....	41
5.1 Implementasi Sistem .....	41
5.1.1 Implementasi <i>Preprocessing</i> .....	41
5.1.1.1 Mencari Nilai Dari 15 Fitur .....	41
5.1.1.2 Menyimpan Nilai Fitur.....	44
5.1.2 Implementasi ELM.....	44
5.1.2.1 <i>Training</i> ELM.....	44
5.1.2.2 <i>Testing</i> ELM .....	48

BAB 6 PENGUJIAN DAN ANALISIS.....	50
6.1 Pengujian Jumlah <i>Hidden Neuron</i> dengan Metode <i>K-fold Cross Validation</i> .....	50
6.1.1 Pengujian 2 Kelas .....	50
6.1.1.1 Kelas Metabolisme dan Kanker .....	50
6.1.1.2 Kelas Radang dan Kanker .....	51
6.1.1.3 Kelas Metabolisme dan Radang .....	52
6.1.2 Pengujian 3 Kelas .....	53
6.1.2.1 Kelas Metabolisme, Radang, dan Kanker .....	53
6.2 Pengujian Jumlah Fitur dengan Metode <i>K-fold Cross Validation</i> .....	55
6.2.1 Pengujian 11 Fitur .....	55
6.3 Evaluasi Kinerja Sistem .....	56
6.3.1 Kelas Metabolisme .....	56
6.3.2 Kelas Radang .....	57
6.3.3 Kelas Kanker .....	58
6.3.4 Hasil Evaluasi Kinerja Klasifikasi Multipel Kelas .....	59
BAB 7 PENUTUP .....	60
7.1 Kesimpulan.....	60
7.2 Saran .....	60
DAFTAR PUSTAKA.....	61



## DAFTAR TABEL

Tabel 2.1 Kajian Pustaka .....	6
Tabel 2.2 <i>Confusion Matrix</i> .....	11
Tabel 4.1 Data Latih .....	26
Tabel 4.2 Matriks Bobot ( $W$ ).....	27
Tabel 4.3 Matriks Bobot <i>Transpose</i> ( $W^T$ ) .....	27
Tabel 4.4 Matriks $H_{init}$ .....	28
Tabel 4.5 Matriks $H$ .....	28
Tabel 4.6 Matriks $H^T$ .....	29
Tabel 4.7 Matriks $H^T \cdot H$ .....	30
Tabel 4.8 Matriks $(H^T \cdot H)^{-1}$ .....	30
Tabel 4.9 Matriks $H^\dagger$ .....	31
Tabel 4.10 Matriks $\hat{\beta}$ .....	31
Tabel 4.11 Data Uji.....	32
Tabel 4.12 Matriks $H_{init}$ .....	36
Tabel 4.13 Matriks $H$ .....	36
Tabel 4.14 Matriks $\hat{Y}$ .....	36
Tabel 4.15 Kelas Prediksi.....	37
Tabel 4.16 Perancangan Pengujian Jumlah <i>Hidden Neuron</i> Kelas Metabolisme dan Kanker .....	38
Tabel 4.17 Perancangan Pengujian Jumlah <i>Hidden Neuron</i> Kelas Radang dan Kanker .....	38
Tabel 4.18 Perancangan Pengujian Jumlah <i>Hidden Neuron</i> Kelas Metabolisme dan Radang.....	39
Tabel 4.19 Perancangan Pengujian Jumlah <i>Hidden Neuron</i> Kelas Metabolisme, Radang, dan Kanker .....	40
Tabel 4.20 Perancangan Pengujian 11 Fitur pada Kelas Metabolisme, Radang, dan Kanker .....	40
Tabel 6.1 Perancangan Pengujian Jumlah <i>Hidden Neuron</i> Kelas Metabolisme dan Kanker .....	51
Tabel 6.2 Perancangan Pengujian Jumlah <i>Hidden Neuron</i> Kelas Radang dan Kanker .....	51
Tabel 6.3 Perancangan Pengujian Jumlah <i>Hidden Neuron</i> Kelas Metabolisme dan Radang.....	52



Tabel 6.4 Perancangan Pengujian Jumlah *Hidden Neuron* Kelas Metabolisme, Radang, dan Kanker ..... 54

Tabel 6.5 Perancangan Pengujian 11 Fitur pada Kelas Metabolisme, Radang, dan Kanker ..... 55

Tabel 6.6 *Confusion Matrix* Kelas Metabolisme ..... 56

Tabel 6.7 *Confusion Matrix* Kelas Radang..... 57

Tabel 6.8 *Confusion Matrix* Kelas Kanker..... 58



## DAFTAR GAMBAR

Gambar 2.1 Arsitektur JST.....	8
Gambar 2.2 Arsitektur ELM.....	9
Gambar 4.1 Arsitektur ELM.....	14
Gambar 4.2 Diagram Alir Sistem .....	15
Gambar 4.3 Diagram Alir <i>Preprocessing</i> Notasi SMILES .....	16
Gambar 4.4 Diagram Alir <i>Training</i> ELM .....	17
Gambar 4.5 Diagram Alir Menghitung $W$ .....	18
Gambar 4.6 Diagram Alir Menghitung $W^T$ .....	19
Gambar 4.7 Diagram Alir Menghitung $H_{init}$ .....	20
Gambar 4.8 Diagram Alir Menghitung $H$ .....	21
Gambar 4.9 Diagram Alir Menghitung $\hat{\beta}$ .....	22
Gambar 4.10 Diagram Alir <i>Testing</i> ELM .....	23
Gambar 4.11 Diagram Alir Menghitung Kelas Prediksi .....	24
Gambar 5. 1 <i>Screenshot</i> Keluaran Sistem <i>Preprocessing</i> .....	49
Gambar 5. 2 <i>Screenshot</i> Keluaran Sistem ELM .....	49
Gambar 6.1 Grafik Rata-Rata Akurasi Jumlah <i>Hidden Neuron</i> Terhadap 2 Kelas. 53	
Gambar 6.2 Grafik Rata-Rata Akurasi Jumlah <i>Hidden Neuron</i> pada Kelas Metabolisme, Radang, dan Kanker .....	54
Gambar 6.3 Grafik Rata-Rata Akurasi Jumlah Fitur pada Kelas Metabolisme, Radang, dan Kanker .....	56

## DAFTAR KODE PROGRAM

Kode Program 5.1 Mencari Nilai Dari 15 Fitur .....	41
Kode Program 5.2 Menyimpan Nilai Fitur .....	44
Kode Program 5.3 <i>Training</i> ELM .....	44
Kode Program 5.4 <i>Testing</i> ELM .....	48



## BAB 1 PENDAHULUAN

Pada bab ini dijelaskan pendahuluan pada penelitian yang akan dilakukan. Bab ini terdiri dari latar belakang, rumusan masalah, tujuan, manfaat, batasan masalah, dan sistematika pembahasan.

### 1.1 Latar Belakang

Senyawa adalah zat kimia yang mempunyai minimal dua unsur pembentuk yang nantinya dapat dipecah lagi menjadi unsur terkecil yaitu atom (Brown, dkk, 2009). Di alam semesta ini terdapat jutaan senyawa yang terdiri dari senyawa aktif dan tidak aktif. Senyawa aktif mempunyai banyak manfaat seperti pengobatan, produk kesehatan, suplemen makanan, dan kosmetik (Melissa, & Muchtaridi, 2017). Senyawa aktif dapat diperoleh dari ekstraksi tanaman dan hewan. Indonesia merupakan salah satu negara yang mempunyai potensi alam terbesar dengan ditemukannya berbagai spesies tanaman dan hewan. Hal ini menjadi keuntungan tersendiri bagi masyarakat Indonesia dalam memanfaatkan kekayaan alamnya terutama dalam bidang farmakologi.

Dalam bidang farmakologi, senyawa aktif mempunyai manfaat untuk pencegahan dan penyembuhan suatu penyakit. Namun, senyawa aktif yang sudah diidentifikasi manfaatnya masih sedikit sekali jika dibandingkan dengan keberadaannya di alam semesta ini. Senyawa aktif dapat diidentifikasi manfaatnya dengan serangkaian penelitian di dalam laboratorium oleh kimiawan. Fitokimia merupakan salah satu uji lab yang penting dilakukan untuk mengetahui golongan senyawa kimia dalam tanaman atau hewan yang menjadi objek penelitian. Terdapat dua faktor penting dalam uji fitokimia, yaitu pemilihan pelarut dan metode ekstraksi (Kristanti, dkk, 2008). Serangkaian penelitian tersebut membutuhkan waktu yang tidak sebentar, begitu pun biaya yang diperlukan untuk penelitian juga tidak sedikit. Oleh karena itu, dilakukan penelitian dalam bidang informatika dengan membuat sistem klasifikasi senyawa aktif untuk mengetahui manfaat farmakologinya. Sistem ini mengidentifikasi atom pada senyawa aktif dan mengklasifikasikan senyawa aktif berdasarkan notasi kimianya. Namun, komputer mengalami kendala dalam kecepatan pemrosesan jika input senyawa menggunakan notasi kimia dua dimensi maupun tiga dimensi sehingga dibutuhkan notasi kimia yang lebih sederhana.

*Simplified Molecular Input Line Entry System (SMILES)* adalah notasi kimia satu dimensi yang dirancang sebagai pemrosesan informasi kimia modern untuk menggambarkan struktur kimia yang terdiri dari karakter-karakter American Standard Code for Information Interchange (ASCII) dan disimpan dalam variabel *string* (Weininger, 1988). Notasi SMILES dipilih karena sederhana namun tetap memiliki informasi yang sama dengan notasi senyawa kimia lain sehingga dapat dibaca dengan baik oleh manusia maupun komputer. Oleh karena itu, notasi



SMILES dipilih dalam penelitian ini untuk mempermudah komputasi dalam sistem klasifikasi.

Salah satu tujuan dari penelitian ini adalah mendapatkan akurasi yang maksimal. Pemilihan fitur dan metode yang digunakan dapat memengaruhi hasil akurasi dari penelitian ini. Beberapa penelitian tentang klasifikasi senyawa kimia dengan menggunakan notasi SMILES sudah pernah dilakukan sebelumnya dengan berbagai macam metode. Salah satunya adalah penelitian Witanto pada tahun 2018 untuk pengelompokan fungsi aktif senyawa data SMILES menggunakan metode K-Means dengan inialisasi pusat kluster menggunakan metode Heuristic  $O(N \log N)$  dengan akurasi sebesar 63%. Penelitian selanjutnya dilakukan oleh Tigusti pada tahun 2018 yaitu implementasi Fuzzy K-Nearest Neighbor (FK-NN) untuk mengklasifikasi fungsi senyawa berdasarkan SMILES dengan rata-rata akurasi sebesar 68%. Kedua penelitian tersebut menggunakan 11 fitur yang sama yaitu atom B, S, N, O, I, F, C, P, Cl, Br dan OH. Dalam penelitian ini dilakukan penambahan jumlah fitur sebanyak 15, yaitu atom B, C, N, O, P, S, F, Cl, Br, I, OH, @, =, #, dan () dengan harapan memperoleh akurasi yang lebih baik.

Banyak metode yang dapat digunakan untuk pengklasifikasian, salah satunya adalah metode Jaringan Saraf Tiruan (JST). JST cocok digunakan untuk pengklasifikasian data dengan fitur yang banyak dan jumlah data yang besar. Cara kerja JST adalah dengan meniru kerja otak manusia terutama dalam hal belajar. Berdasarkan penelitian yang telah dilakukan dengan berbagai macam metode yang digunakan, penelitian ini menggunakan salah satu metode di dalam JST yaitu *Extreme Learning Machine* (ELM). ELM merupakan metode jaringan saraf tiruan *feed-forward* dengan satu *hidden layer* (Sun, dkk, 2008). Metode ELM mempunyai kelebihan dalam *learning speed*, serta mempunyai tingkat akurasi yang lebih baik dibandingkan dengan metode konvensional seperti *Moving Average* dan *Exponential Smoothing* (Agustina, 2010). Bahkan ELM mampu melakukan generalisasi dengan baik dalam waktu yang sangat cepat dibandingkan dengan metode *Support Vector Machine* (SVM) dan *Backpropagation* (Huang, dkk, 2006). Beberapa penelitian yang menggunakan metode ELM sudah pernah dilakukan sebelumnya dengan berbagai macam objek. Salah satunya adalah penelitian yang dilakukan oleh Fadilla pada tahun 2018 yaitu klasifikasi penyakit *Chronic Kidney Disease* (CKD) menggunakan metode *Extreme Learning Machine* (ELM) mempunyai akurasi sebesar 96,7%. Oleh karena itu, diharapkan penelitian ini dapat menghasilkan tingkat akurasi yang lebih baik dengan menggunakan metode ELM. Penelitian ini berjudul *Klasifikasi Senyawa Kimia Dengan Notasi Simplified Molecular Input Line Entry System* (SMILES) Menggunakan Metode *Extreme Learning Machine* (ELM).

## 1.2 Rumusan Masalah

Berdasarkan latar belakang di atas, maka rumusan masalah yang menjadi dasar penelitian ini adalah:

1. Bagaimana tingkat akurasi klasifikasi senyawa kimia dengan notasi SMILES menggunakan metode ELM jika dilakukan pengujian jumlah *hidden neuron* yang berbeda?
2. Bagaimana tingkat akurasi klasifikasi senyawa kimia dengan notasi SMILES menggunakan metode ELM jika dilakukan pengujian jumlah fitur atau *input neuron* yang berbeda?

### 1.3 Tujuan

Tujuan yang ingin dicapai dari penelitian ini adalah:

1. Menguji akurasi sistem klasifikasi senyawa kimia dengan notasi SMILES menggunakan metode ELM.
2. Mengetahui manfaat senyawa kimia berdasarkan notasi SMILES, terutama senyawa baru.

### 1.4 Manfaat

Manfaat dari penelitian ini adalah:

1. Membantu memaksimalkan pemanfaatan kekayaan alam di Indonesia dalam bidang farmakologi.
2. Mempermudah kimiawan dalam meneliti suatu senyawa jika dibanding dengan proses penelitian di laboratorium.
3. Membantu mengembangkan ilmu pengetahuan, terutama ilmu kimia.

### 1.5 Batasan Masalah

Batasan dari penelitian ini adalah:

1. Senyawa yang digunakan adalah yang mempunyai notasi SMILES.
2. Fitur yang digunakan ada 15 yaitu *B, C, N, O, P, S, F, Cl, Br, I, OH, @, =, #*, dan *()*. Nilai setiap fitur didapatkan dari jumlah fitur dibagi panjang notasi SMILES.

### 1.6 Sistematika Pembahasan

Sistematika pembahasan berisi deskripsi singkat tentang struktur skripsi mulai bab pendahuluan sampai bab penutup, sebagai berikut:

#### Bab 1 Pendahuluan

Bab ini membahas gambaran umum isi penelitian yang terdiri dari: latar belakang, rumusan masalah, tujuan, manfaat, batasan masalah, dan sistematika pembahasan.

## **Bab 2 Landasan Kepustakaan**

Bab ini dijelaskan uraian tentang kajian pustaka dan dasar teori yang berkaitan dengan judul penelitian “Klasifikasi Senyawa Kimia Dengan Notasi Molecular Input Line Entry System (SMILES) Menggunakan Metode Extreme Learning Machine (ELM)”.

## **Bab 3 Metodologi Penelitian**

Bab ini akan membahas tentang tahapan penelitian klasifikasi senyawa kimia dengan notasi SMILES menggunakan metode ELM. Metodologi penelitian meliputi tipe penelitian, strategi penelitian, objek dan lokasi penelitian, pengumpulan data, teknik analisis data, dan peralatan pendukung.

## **Bab 4 Perancangan**

Bab ini akan dijelaskan tentang perancangan sistem yang terdiri dari deskripsi sistem, arsitektur sistem, perancangan sistem, perhitungan manual, dan perancangan pengujian.

## **Bab 5 Implementasi**

Bab ini membahas tentang implementasi sistem berdasarkan perancangan yang telah dilakukan menggunakan bahasa pemrograman Python versi 2.7. Bab ini terdiri dari implementasi sistem dan *screenshot* keluaran sistem.

## **Bab 6 Pengujian Dan Analisis**

Bab ini berisi hasil pengujian dan analisis terhadap hasil pengujian yang telah dilakukan oleh sistem serta evaluasi kinerja sistem. Pengujian dilakukan berdasarkan perancangan pengujian dengan dilakukan 2 pengujian, yaitu pengujian jumlah *hidden neuron* dengan metode *k-fold cross validation* dan pengujian jumlah fitur dengan metode *k-fold cross validation*.

## **Bab 7 Penutup**

Bab ini berisi kesimpulan dan saran terhadap penelitian yang telah dilakukan.

## BAB 2 LANDASAN KEPUSTAKAAN

Pada bab ini dijelaskan uraian tentang kajian pustaka dan dasar teori yang berkaitan dengan judul penelitian “Klasifikasi Senyawa Kimia Dengan Notasi Molecular Input Line Entry System (SMILES) Menggunakan Metode Extreme Learning Machine (ELM)”. Kajian pustaka berisi tentang penelitian sebelumnya yang terkait dengan penelitian saat ini yang dapat dilihat pada Tabel 2.1.

### 2.1 Kajian Pustaka

Kajian Pustaka pada penelitian sebelumnya dapat digunakan sebagai pengukur keberhasilan. ELM banyak digunakan sebagai metode penelitian, selain itu juga dibutuhkan referensi untuk dapat membantu dalam mengerjakan penelitian.

Penelitian pertama menggunakan metode *Fuzzy K-Nearest Neighbor (FK-NN)* untuk mengklasifikasi fungsi senyawa berdasarkan SMILES yang dilakukan oleh Tigusti pada tahun 2018. Penelitian ini bertujuan untuk mengklasifikasikan fungsi senyawa berdasarkan notasi SMILES dengan menggunakan data latih sebanyak 503 data dan data uji sebanyak 128 data. Fitur yang digunakan ada 12 yaitu atom *B, C, N, O, P, S, F, Cl, Br, I, OH* dan panjang dari notasi *SMILES*. Metode yang digunakan adalah *Fuzzy K-Nearest Neighbor (FK-NN)*. Pengujian *K-Fold Cross Validation* sebanyak 10 kali percobaan menghasilkan nilai akurasi tertinggi sebesar 77% dengan rata-rata akurasi dari seluruh percobaan sebesar 68%. Pengujian variasi nilai *k* menghasilkan nilai akurasi tertinggi sebesar 71% dengan nilai *k* = 15. Pengujian variasi jumlah data latih menghasilkan nilai akurasi tertinggi sebesar 70% dengan menggunakan seluruh data latih.

Penelitian kedua dilakukan oleh Witanto pada tahun 2018. Penelitian tersebut mengelompokkan fungsi aktif senyawa data SMILES menggunakan metode *K-Means* dengan inialisasi pusat kluster menggunakan metode *Heuristic O(N LogN)*. Penelitian ini dilakukan untuk mengelompokkan fungsi senyawa data SMILES dengan menggunakan data latih sebanyak 512 data dan data uji sebanyak 128 data. Fitur yang digunakan sebanyak 11 buah yaitu atom *B, S, N, O, I, F, C, P, Cl, Br* dan *OH* yang ada pada senyawa. Metode yang digunakan dalam penelitian adalah *K-Means* dengan inialisasi pusat kluster menggunakan metode *Heuristic O(N LogN)*. Hasil akurasi pengujian validasi program yaitu 40%. Pengujian seluruh data latih dan data uji memperoleh akurasi 63%. Pengujian *K-Fold Cross Validation* pada *improved K-Means* memperoleh rata-rata akurasi 52,58% dari 10 kali percobaan dan rata-rata akurasi *K-Means* 51%.

Penelitian ketiga dilakukan oleh Fadilla pada tahun 2018 dengan mengelompokkan penyakit *Chronic Kidney Disease (CKD)* menggunakan metode ELM. Digunakan 281 data dengan perbandingan antara data latih dan data uji 70:30 serta menggunakan 24 fitur diperoleh tingkat akurasi sebesar 96,7%.



Tabel 2.1 Kajian Pustaka

Penelitian	Objek	Metode	Hasil
Implementasi <i>Fuzzy K-Nearest Neighbor (FK-NN)</i> Untuk Mengklasifikasi Fungsi Senyawa Berdasarkan <i>Simplified Molecular Input Line Entry System (SMILES)</i> (Tigusti, 2018)	Fungsi Senyawa Berdasarkan <i>Simplified Molecular Input Line Entry System (SMILES)</i>	<i>Fuzzy K-Nearest Neighbor (FK-NN)</i>	Pengujian K-Fold Cross Validation sebanyak 10 kali percobaan menghasilkan nilai akurasi tertinggi sebesar 77% dengan rata-rata akurasi dari seluruh percobaan sebesar 68%.
Pengelompokan Fungsi Aktif Senyawa Data <i>SMILES (Simplified Molecular Input Line Entry System)</i> Menggunakan Metode K-Means Dengan Inisialisasi Pusat Menggunakan Metode <i>Heuristic O(N LogN)</i> (Witanto, 2018)	Fungsi Aktif Senyawa Data <i>SMILES (Simplified Molecular Input Line Entry System)</i>	K-Means Dengan Inisialisasi Pusat Menggunakan Metode <i>Heuristic O(N LogN)</i>	Pengujian seluruh data latih dan data uji yaitu sebanyak 512 data dan 128 data memperoleh akurasi 63%. Pengujian <i>K-Fold Cross Validation</i> pada <i>improved K-Means</i> memperoleh rata-rata akurasi 52,58% dari 10 kali percobaan dan rata-rata akurasi K-Means 51%.
Klasifikasi Penyakit <i>Chronic Kidney Disease (CKD)</i> dengan menggunakan Metode <i>Extreme Learning machine (ELM)</i> (Fadilla, 2018)	Penyakit <i>Chronic Kidney Disease (CKD)</i>	<i>Extreme Learning machine (ELM)</i>	Nilai akurasi yang dihasilkan sebesar 96,7%

## 2.2 Dasar Teori

Bab ini akan menjelaskan landasan teori yang memuat lima hal yaitu Notasi *Simplified Molecular Input Line Entry System (SMILES)*, klasifikasi, Jaringan Saraf Tiruan (JST), fungsi aktivasi jaringan saraf tiruan, *Extreme Learning Machine (ELM)*, dan evaluasi kinerja klasifikasi.



### 2.2.1 Notasi *Simplified Molecular Input Line Entry System* (SMILES)

SMILES merupakan notasi kimia yang dirancang sebagai pemrosesan informasi kimia modern untuk menggambarkan struktur kimia yang terdiri dari karakter-karakter American Standard Code for Information Interchange (ASCII) dan disimpan dalam variabel *string* (Weininger, 1988). Notasi SMILES adalah tatanama senyawa kimia yang sederhana sehingga dapat dibaca dengan baik oleh manusia maupun komputer. Oleh karena itu, notasi SMILES dapat digunakan sebagai pertukaran data dalam bidang informatika. Atom yang digunakan dalam notasi SMILES adalah *B, C, N, O, P, S, F, Cl, Br, I, OH, @, =, #, dan ()*. *B* adalah atom *Boron*, *C* adalah atom *Carbon*, *N* adalah atom *Nitrogen*, *O* adalah atom *Oksigen*, *P* adalah atom *Phosphorus*, *S* adalah atom *Sulfur*, *F* adalah atom *Fluorine*, *Cl* adalah atom *Chlorine*, *Br* adalah atom *Bromine*, *I* adalah atom *Iodine*, *OH* adalah ion *Hidroksida*, *@* adalah stereokimia, *=* adalah rangkap dua, *#* adalah rangkap tiga, dan *()* adalah cabang.

### 2.2.2 Klasifikasi

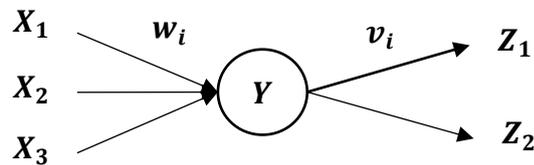
Klasifikasi tidak sama dengan prediksi, dalam *machine learning* klasifikasi adalah bagian dari prediksi. Klasifikasi bertujuan untuk memprediksi suatu kelas (Thabtah, 2007). Klasifikasi menggunakan data latih dan data uji. Data uji dikelompokkan berdasarkan data latih yang sudah ada. Klasifikasi dapat menggunakan beberapa algoritme antara lain *decision tree*, *Naive Bayes*, KNN, algoritme genetik, Jaringan Saraf Tiruan (JST), dan *Support Vector Machine* (SVM).

### 2.2.3 Jaringan Saraf Tiruan (JST)

Jaringan Saraf Tiruan (JST) meniru cara kerja dari jaringan saraf pada makhluk hidup. Otak manusia terdiri dari banyak sel saraf (neuron) yang bertugas untuk memroses informasi. Neuron saling terhubung membentuk jaringan saraf untuk mendukung kemampuan kerja otak manusia. Seperti halnya super komputer yang memiliki kumpulan prosesor saling berhubungan. JST adalah prosesor yang terdistribusi besar-besaran secara paralel yang dibuat dari unit proses sederhana dan mempunyai kemampuan untuk menyimpan pengetahuan berupa pengalaman serta dapat digunakan untuk proses lain (Haykin, 2009). Jadi, JST meniru cara kerja otak manusia terutama kemampuan untuk belajar. Arsitektur JST dapat dilihat pada Gambar 2.1 (Fausett, 1994). Dalam JST terdapat tiga model pembelajaran, yaitu:

1. *Supervised Learning* adalah metode pembelajaran yang menyimpulkan pemetaan data dengan membandingkan pemetaan data saat ini dengan pemetaan data sebelumnya. Pada *Supervised Learning* terdapat beberapa algoritme yang dapat digunakan seperti *Hebb Rule*, *Perceptron*, *Delta Rule*, *Backpropagation*, *Heteroassociative Memory*, *Bidirectional Associative Memory* (BAM), *Learning Vector Quantization* (LVQ), *Extreme Learning Machine* (ELM) (Rochman, & Rachmad, 2017).

2. *Unsupervised Learning* adalah metode pembelajaran yang mengelompokkan unit-unit yang hampir sama dalam area tertentu.
3. *Reinforced Learning* adalah metode yang membuat sistem dapat belajar dari keputusan yang diambil sebelumnya dengan cara memberikan *reward* setiap kali sistem melakukan suatu hal yang benar.



Gambar 2.1 Arsitektur JST

Keterangan:

- $X_1, X_2, X_3$  : *input neuron*
- $Y$  : *hidden neuron*
- $Z_1, Z_2$  : *output neuron*
- $w_i, v_i$  : *bobot*

### 2.2.4 Fungsi Aktivasi Jaringan Saraf Tiruan (JST)

Fungsi aktivasi digunakan untuk menentukan *output* dari neuron, input yang digunakan adalah kombinasi linier input dan bobotnya (Siang, 2009). Macam-macam fungsi aktivasi yang digunakan pada Jaringan Saraf Tiruan adalah:

#### 1. Fungsi Linier

Fungsi linier memungkinkan nilai *output* sama dengan nilai inputnya. Fungsi ini menggunakan Persamaan 2.1.

$$y = f(x) \tag{2.1}$$

#### 2. Fungsi *Threshold*

Fungsi *threshold* atau fungsi tangga biner atau fungsi step digunakan untuk mengubah input variabel kontinu menjadi *output* biner (0 atau 1). Fungsi ini menggunakan Persamaan 2.2.

$$y = f(x) = \begin{cases} 1, & \text{if } x \geq 0 \\ 0, & \text{if } x < 0 \end{cases} \tag{2.2}$$

#### 3. Fungsi *Sigmoid* Biner

Fungsi *sigmoid* biner mempunyai nilai pada interval 0 sampai 1. Fungsi ini menggunakan Persamaan 2.3.

$$y = f(x) = \frac{1}{1+e^{-x}} \tag{2.3}$$

#### 4. Fungsi *Sigmoid* Bipolar

Fungsi *sigmoid* bipolar mempunyai nilai interval 1 sampai -1. Fungsi ini menggunakan Persamaan 2.4

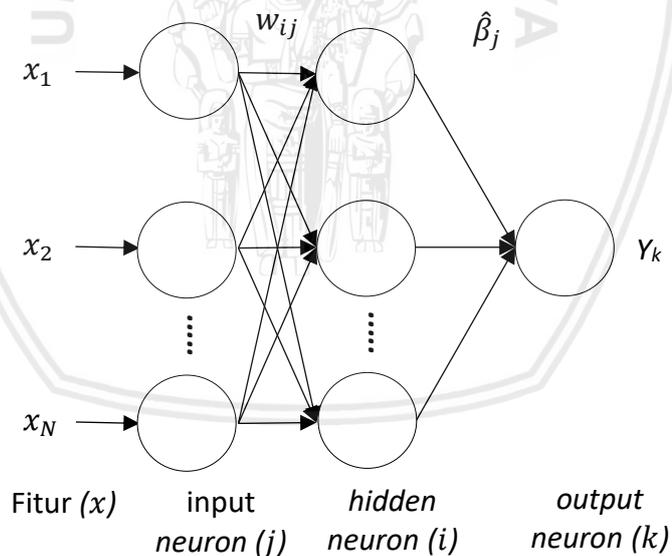


$$y = f(x) = \frac{1-e^{-x}}{1+e^{-x}} \tag{2.4}$$

Fungsi aktivasi yang digunakan pada penelitian ini adalah fungsi *sigmoid* biner karena performa ELM mengalami peningkatan jika menggunakan fungsi aktivasi *sigmoid* (Huang, dkk, 2012).

### 2.2.5 Extreme Learning Machine (ELM)

*Extreme Learning Machine* (ELM) ditemukan oleh Huang pada tahun 2004. *Extreme Learning Machine* (ELM) merupakan metode pembelajaran baru di dalam jaringan saraf tiruan. ELM adalah Jaringan Saraf Tiruan (JST) *feedforward* dengan *single hidden layer* atau dikenal dengan *Single Hidden Layer Feedforward neural Networks* (SLFNs) (Sun, dkk, 2008). Arsitektur ELM dapat dilihat pada Gambar 2.2. Metode pembelajaran ELM dibuat untuk menyelesaikan permasalahan dari jaringan saraf tiruan terutama dalam hal *learning speed*. Berbeda dengan metode lainnya pada JST, ELM tidak melatih bobot input tapi melatih untuk mendapatkan bobot *output* dengan *moore-penrose invers* pada sistem linier secara umum dan mendapatkan *node* yang memiliki nilai *output* maksimal. Nilai bobot dan bias ditentukan secara acak (*random*). Oleh karena itu, algoritme ELM mempunyai *learning speed* dan kinerja generalisasi yang baik.



Gambar 2.2 Arsitektur ELM

#### 2.2.5.1 Algoritme ELM

Algoritme ELM mempunyai tahapan untuk menghitung data latih dan data uji (Huang, dkk, 2006).

Terdapat 5 langkah dalam *training* ELM, yaitu:

**Langkah 1:** Menentukan nilai bobot ( $w_{ij}$ ) secara *random* dengan  $i$  adalah *hidden neuron* dan  $j$  adalah input *neuron*. Nilai  $W$  dibangkitkan antara -0,5 sampai 0,5.

**Langkah 2:** Menghitung matriks inialisasi keluaran *hidden neuron* ( $H_{init}$ ), dengan  $X$  adalah matriks data latih dengan panjang  $j$  input *neuron* (fitur) dikali  $N$  data latih dan  $W^T$  adalah matriks nilai bobot yang ditransposisi.

$$H_{init} = X \cdot W^T \quad (2.5)$$

**Langkah 3:** Menghitung matriks keluaran *hidden neuron* ( $H$ ) menggunakan fungsi aktivasi *sigmoid* biner sesuai Persamaan 2.3 sehingga nilai  $H$  dapat dihitung menggunakan Persamaan 2.6.

$$H = \frac{1}{(1 + \exp(-H_{init}))} \quad (2.6)$$

**Langkah 4:** Menghitung matriks  $H^\dagger$ , dengan  $H^T$  adalah matriks nilai  $H$  yang sudah ditransposisi dan  $(H^T \cdot H)^{-1}$  adalah nilai *invers* dari  $(H^T \cdot H)$ .

$$H^\dagger = (H^T \cdot H)^{-1} \cdot H^T \quad (2.7)$$

**Langkah 5:** Menghitung bobot keluaran ( $\hat{\beta}$ ), dengan  $\hat{\beta}$  adalah bobot keluaran dan  $Y$  adalah target atau kelas.

$$\hat{\beta} = H^\dagger \cdot Y \quad (2.8)$$

Dengan menggunakan nilai bobot ( $w_{ij}$ ) dan  $\hat{\beta}$  yang diperoleh dari *training* ELM, maka lakukan 4 langkah dalam *testing* ELM berikut:

**Langkah 1:** Menghitung matriks inialisasi keluaran *hidden neuron* ( $H_{init}$ ), dengan  $H_{init}$  adalah matriks keluaran *hidden neuron* dan  $X$  adalah matriks data uji dengan panjang  $j$  input *neuron* (fitur) dikali  $N$  data uji dan  $W^T$  adalah matriks nilai bobot yang ditransposisi. Persamaan untuk menghitung  $H_{init}$  sama dengan Persamaan 2.5.

**Langkah 2:** Menghitung matriks keluaran *hidden neuron* ( $H$ ). Persamaan yang digunakan sama dengan Persamaan 2.6.

**Langkah 3:** Menghitung  $\hat{Y}$  prediksi menggunakan Persamaan 2.9.

$$\hat{Y} = H \cdot \hat{\beta} \quad (2.9)$$

**Langkah 4:** Menghitung kelas prediksi dengan mencari selisih terkecil antara  $\hat{Y}$  prediksi dengan kelas sebenarnya ( $Y$ ) yang terdapat pada latih.

$$\text{Kelas prediksi} = |\hat{Y} - Y| \quad (2.10)$$

### 2.2.6 Evaluasi Kinerja Klasifikasi

Evaluasi kinerja klasifikasi digunakan untuk mengukur seberapa akurat sistem untuk mengelompokkan kelas. *Confusion matrix* digunakan untuk membantu perhitungan evaluasi kinerja (Han, dkk, 2012). *Confusion matrix* dapat dilihat pada Tabel 2.2.



Tabel 2.2 *Confusion Matrix*

<i>Predicted Class</i> \ <i>Actual Class</i>	Yes	No
Yes	TP	FN
No	FP	TN

Pada Tabel 2.2 ditampilkan 4 istilah yang akan digunakan untuk menghitung evaluasi kinerja, yaitu:

1. *True Positives* (TP) adalah jumlah data positif yang diklasifikasikan benar oleh sistem.
2. *True Negatives* (TN) adalah jumlah data negatif yang diklasifikasikan benar oleh sistem.
3. *False Positives* (FP) adalah jumlah data negatif yang diklasifikasikan salah oleh sistem sebagai data positif.
4. *False Negatives* (FN) adalah jumlah data positif yang diklasifikasikan salah oleh sistem sebagai data negatif.

Evaluasi kinerja yang digunakan ada 3, yaitu akurasi, *precision*, dan *recall*.

### 1. Akurasi

Akurasi digunakan untuk mengetahui seberapa akurat sistem melakukan klasifikasi dengan baik. Untuk menghitung nilai akurasi digunakan Persamaan 2.11.

$$Akurasi = \frac{TP+TN}{TP+TN+FP+FN} \times 100\% \tag{2.11}$$

### 2. Precision

*Precision* digunakan untuk mengetahui kemampuan sistem agar tidak memanggil kembali data yang tidak relevan atau tidak sesuai dengan apa yang diinginkan pengguna dari basis data. Untuk menghitung nilai *precision* digunakan Persamaan 2.12.

$$Precision = \frac{TP}{TP+FP} \times 100\% \tag{2.12}$$

### 3. Recall

*Recall* digunakan untuk mengetahui kemampuan sistem dalam memanggil kembali data yang dianggap relevan atau sesuai dengan apa yang diinginkan pengguna dari basis data. Untuk menghitung nilai *precision* digunakan Persamaan 2.13.

$$Recall = \frac{TP}{TP+FN} \times 100\% \tag{2.13}$$



## BAB 3 METODOLOGI PENELITIAN

Pada bab ini akan membahas tentang tahapan penelitian klasifikasi senyawa kimia dengan notasi SMILES menggunakan metode ELM. Metodologi penelitian meliputi tipe penelitian, strategi penelitian, objek dan lokasi penelitian, pengumpulan data, teknik analisis data, dan peralatan pendukung.

### 3.1 Tipe Penelitian

Penelitian klasifikasi senyawa kimia dengan notasi SMILES menggunakan metode ELM memakai tipe penelitian non *implementatif* dengan pendekatan analitik. Penelitian non *implementatif* mengutamakan percobaan suatu keadaan tertentu lalu menghasilkan tinjauan ilmiah. Penelitian non *implementatif* juga dapat digunakan dalam pelaksanaan tinjauan dan sintesis pada suatu pengetahuan yang sudah ada. Pendekatan analitik adalah suatu penelitian yang menjelaskan keterkaitan antara elemen dalam objek penelitian dengan fenomena yang sedang diteliti.

### 3.2 Strategi Penelitian

Strategi penelitian yang digunakan pada penelitian klasifikasi senyawa kimia dengan notasi SMILES menggunakan metode ELM adalah studi kasus kumulatif dengan mengumpulkan data yang diperoleh dari *website pubchem*, yaitu *website* yang menyediakan basis data yang berisi informasi dari suatu senyawa kimia seperti nama senyawa, formula molekul, notasi SMILES, manfaat, dan informasi lainnya.

### 3.3 Objek dan Lokasi Penelitian

Objek penelitian yang digunakan adalah senyawa kimia yang memiliki notasi SMILES yang diperoleh dari *website pubchem*. Dalam pencarian data diperlukan analisis untuk memilih data notasi SMILES yang sesuai dengan manfaat yang digunakan sebagai kelas pada sistem. Proses analisis penelitian ini dilakukan di Laboratorium Komputasi Cerdas (KC) Fakultas Ilmu Komputer Universitas Brawijaya.

### 3.4 Pengumpulan Data

Berdasarkan sistem yang dibuat, maka data yang dibutuhkan terdiri dari nama senyawa, formula molekul, notasi SMILES, dan manfaatnya. Senyawa tersebut didapatkan dari basis data yang dapat diakses di link <https://pubchem.ncbi.nlm.nih.gov/search/search.cgi#>.



### 3.5 Teknik Analisis Data

Teknik analisis data yang dilakukan pada penelitian klasifikasi senyawa kimia dengan notasi SMILES menggunakan metode ELM adalah dengan melakukan *preprocessing* pada data notasi SMILES lalu dilakukan perhitungan klasifikasi menggunakan metode ELM. Selanjutnya dilakukan pengujian sistem untuk mengetahui apakah sistem dapat bekerja dengan baik sesuai yang diharapkan oleh peneliti. Pengujian yang dilakukan ada 2, yaitu pengujian jumlah *hidden neuron* dengan metode *k-fold cross validation* dan pengujian jumlah fitur dengan metode *k-fold cross validation*. Setelah itu dilakukan perhitungan evaluasi kinerja untuk mengetahui keakuratan sistem dalam mengelompokkan kelas. Data latih yang digunakan sebanyak 840 yang terdiri dari 501 data kelas Metabolisme, 247 data kelas Kanker, dan 92 data kelas Radang. Data uji yang digunakan sebanyak 209 yang terdiri dari 125 data kelas Metabolisme, 62 data kelas Kanker, dan 22 data kelas Radang.

### 3.6 Peralatan Pendukung

Peralatan pendukung yang diperlukan untuk klasifikasi senyawa kimia dengan notasi SMILES menggunakan metode ELM adalah spesifikasi perangkat keras (*hardware*) dan spesifikasi perangkat lunak (*software*). Peralatan pendukung digunakan untuk mengetahui spesifikasi perangkat apa saja yang diperlukan dalam penelitian. Peralatan pendukung yang digunakan dalam penelitian ini adalah sebagai berikut:

#### 3.6.1 Spesifikasi Perangkat Keras

Spesifikasi perangkat keras yang digunakan adalah:

1. Intel core i3-5005U, 2.0 GHz
2. RAM 4 GB
3. *Hard disk* kapasitas 500 GB

#### 3.6.2 Spesifikasi Perangkat Lunak

Spesifikasi perangkat lunak yang digunakan adalah:

1. Sistem Operasi Windows 10 64 bit
2. Bahasa pemrograman Python(x,y)-2.7.10.0
3. Spyder IDE

## BAB 4 PERANCANGAN

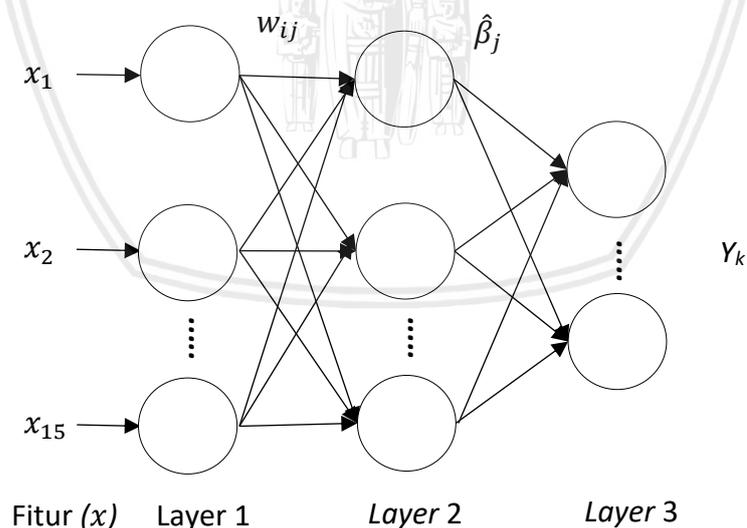
Pada bab ini akan dijelaskan tentang perancangan sistem yang terdiri dari deskripsi sistem, arsitektur sistem, perancangan sistem, perhitungan manual, dan perancangan pengujian.

### 4.1 Deskripsi Sistem

Dalam penelitian ini, dibuat sistem untuk mengklasifikasikan senyawa kimia dengan notasi SMILES menggunakan metode ELM. Sebelum sistem ini dibuat diperlukan perhitungan manual dengan metode ELM. Kelas yang digunakan ada 3, yaitu Metabolisme, Kanker, dan Radang. Input yang digunakan adalah notasi SMILES, lalu *dipreprocessing* untuk mengetahui nilai dari kemunculan 15 fitur / input neuron yang digunakan dalam sistem. 15 fitur yang digunakan yaitu *B, C, N, O, P, S, F, Cl, Br, I, OH, @, =, #, dan ()*. Nilai setiap fitur didapatkan dari jumlah fitur dibagi panjang notasi SMILES. Setelah nilai fitur didapatkan maka dilakukan klasifikasi menggunakan metode ELM.

### 4.2 Arsitektur ELM

Pada arsitektur ELM terdapat 3 *layer* yang memiliki fungsi masing-masing. Arsitektur ELM untuk klasifikasi notasi SMILES dapat dilihat pada Gambar 4.1.



Gambar 4.1 Arsitektur ELM

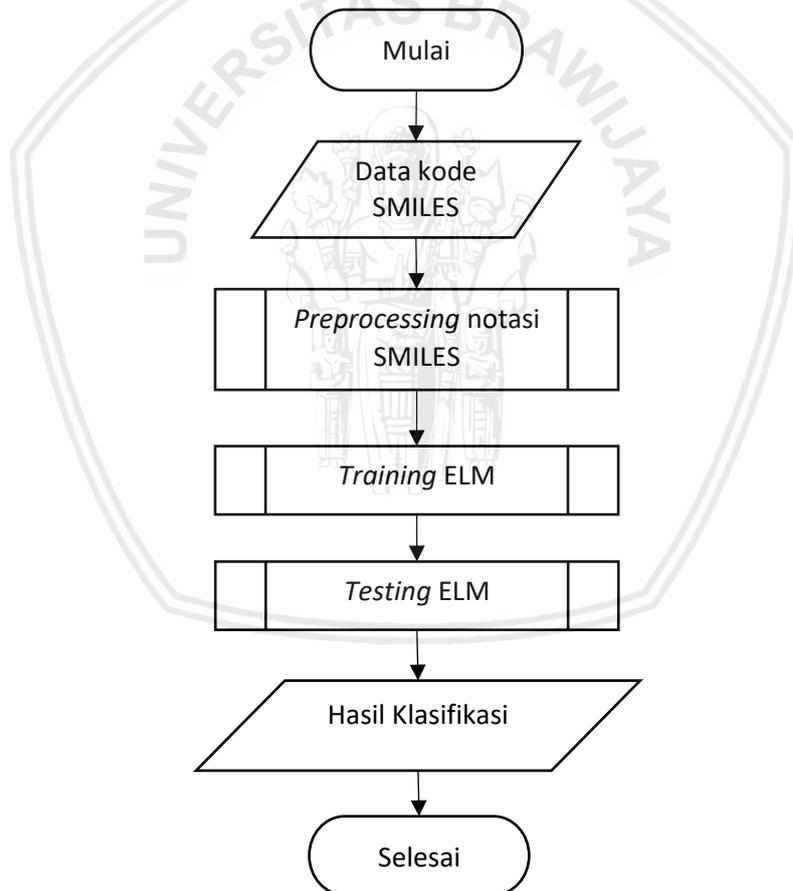
*Layer 1* : Input sistem berupa nilai 15 fitur / input *neuron* yang telah *dipreprocessing*.



- Layer 2* : Input sistem akan dipreses dengan metode ELM menggunakan 5 *hidden neuron*.
- Layer 3* : Hasil akhir dari perhitungan ELM, hasil berupa nilai klasifikasi yaitu kelas yang sesuai dengan notasi SMILES yang dimasukkan, jumlah kelas yang digunakan sebanyak 3.

### 4.3 Perancangan Sistem

Sistem melakukan klasifikasi dengan menerima data notasi SMILES. Sistem kemudian melakukan *preprocessing* notasi SMILES untuk mendapatkan nilai 15 fitur, lalu dilakukan proses perhitungan menggunakan metode ELM yang terdiri dari *training* ELM dan *testing* ELM. Setelah perhitungan selesai maka akan didapatkan hasil klasifikasi berupa kelas dari data notasi SMILES. Diagram alir sistem dapat dilihat pada Gambar 4.2.

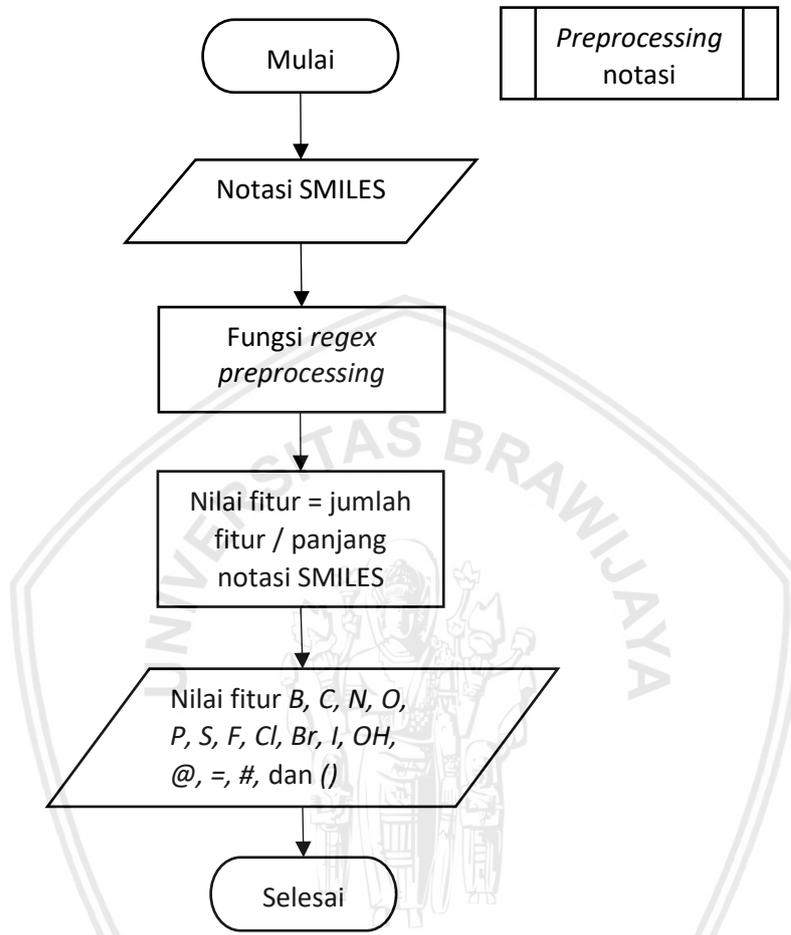


Gambar 4.2 Diagram Alir Sistem

#### 4.3.1 *Preprocessing* Notasi SMILES

Pada tahap ini dilakukan *preprocessing* terhadap notasi SMILES. Notasi SMILES diperiksa setiap karakternya agar dapat diketahui nilai dari 15 fitur yang

telah ditentukan. Setelah itu nilai setiap fitur dibagi dengan panjang notasi SMILES. Tahap *preprocessing* selesai dilakukan dan didapatkan nilai 15 fitur yang nantinya akan digunakan dalam perhitungan ELM. Diagram alir *preprocessing* notasi SMILES dapat dilihat pada Gambar 4.3.



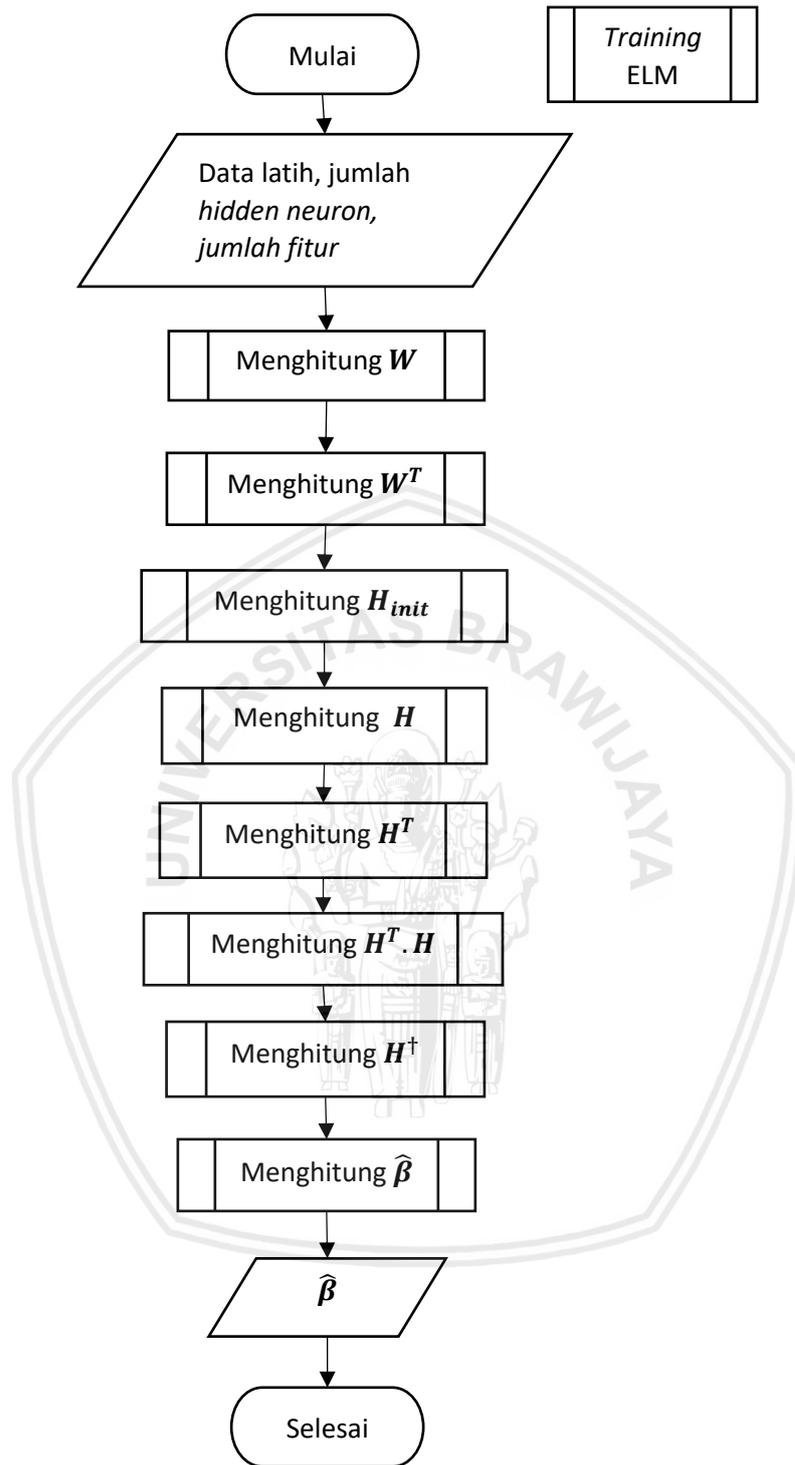
Gambar 4.3 Diagram Alir *Preprocessing* Notasi SMILES

### 4.3.2 Proses ELM

Algoritme ELM mempunyai 2 tahapan, yaitu menghitung data latih dan data uji.

#### 4.3.2.1 *Training* ELM

Tujuan dari proses *training* ELM adalah untuk mendapatkan nilai bobot input ( $W$ ) dan bobot keluaran ( $\hat{\beta}$ ) yang nantinya akan digunakan pada proses *testing*. Diagram alir *training* ELM dapat dilihat pada Gambar 4.4.



Gambar 4.4 Diagram Alir *Training* ELM

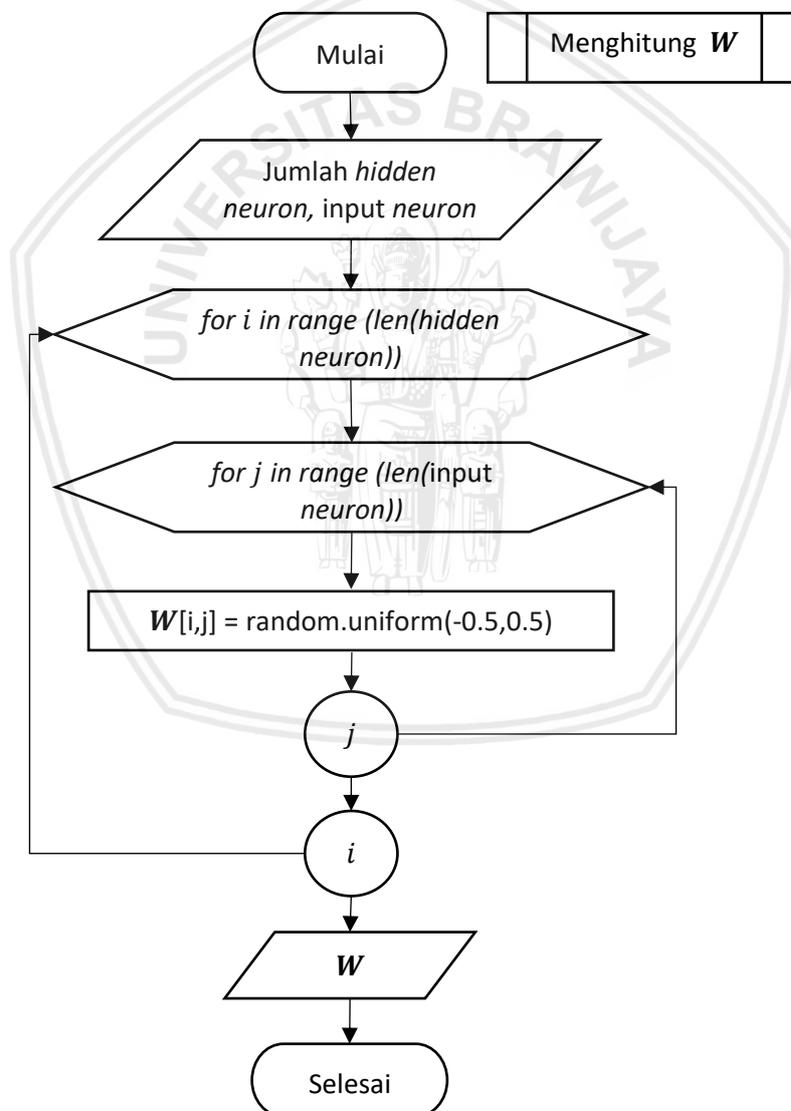
Penjelasan diagram alir *training* ELM berdasarkan Gambar 4.4 adalah:

1. Input data berupa data latih, jumlah *hidden neuron*, dan *jumlah fitur*.
2. Menghitung bobot ( $W$ ).
3. Menghitung bobot *transpose* ( $W^T$ ).

4. Menghitung matriks inialisasi keluaran *hidden neuron* ( $H_{init}$ ).
5. Menghitung matriks keluaran *hidden neuron* ( $H$ ).
6. Menghitung matriks  $H$  transpose ( $H^T$ ).
7. Menghitung perkalian matriks  $H^T$  dengan  $H$  ( $H^T \cdot H$ ).
8. Menghitung matriks  $H^\dagger$ .
9. Menghitung bobot keluaran ( $\hat{\beta}$ ).
10. *Output* yang diperoleh adalah bobot keluaran ( $\hat{\beta}$ ).

Terdapat 5 langkah dalam *training* ELM, yaitu:

**Langkah 1:** Nilai bobot ( $w_{ij}$ ) ditentukan secara *random* sebanyak *hidden neuron* dikali input *neuron* dan  $W$  dibangkitkan antara -0,5 sampai 0,5. Diagram alir menghitung  $W$  dapat dilihat pada Gambar 4.5.



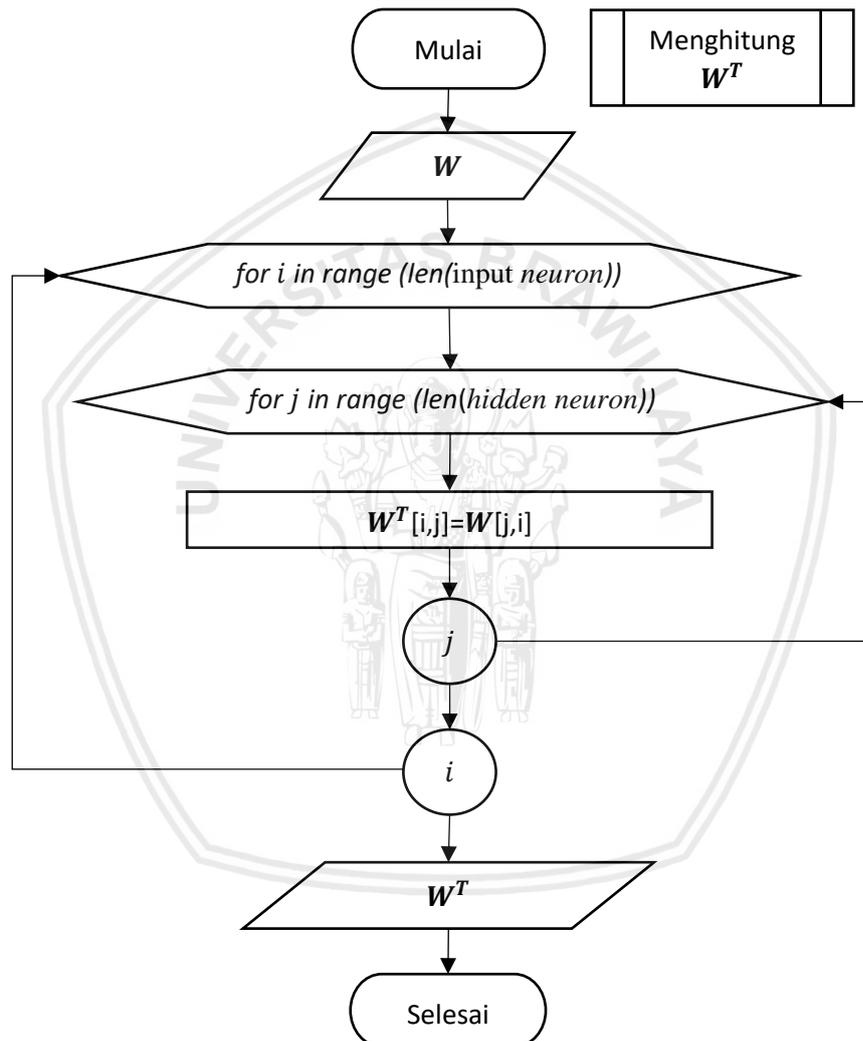
**Gambar 4.5 Diagram Alir Menghitung  $W$**

Perhitungan bobot ( $W$ ) pada Gambar 4.5 adalah:



2. Input data terdiri dari jumlah *hidden neuron* dan input *neuron*.
3. Dilakukan perulangan nilai  $W$  secara *random* sebanyak *hidden neuron* dikali input *neuron*.
4. *Output* yang dihasilkan adalah  $W$ .

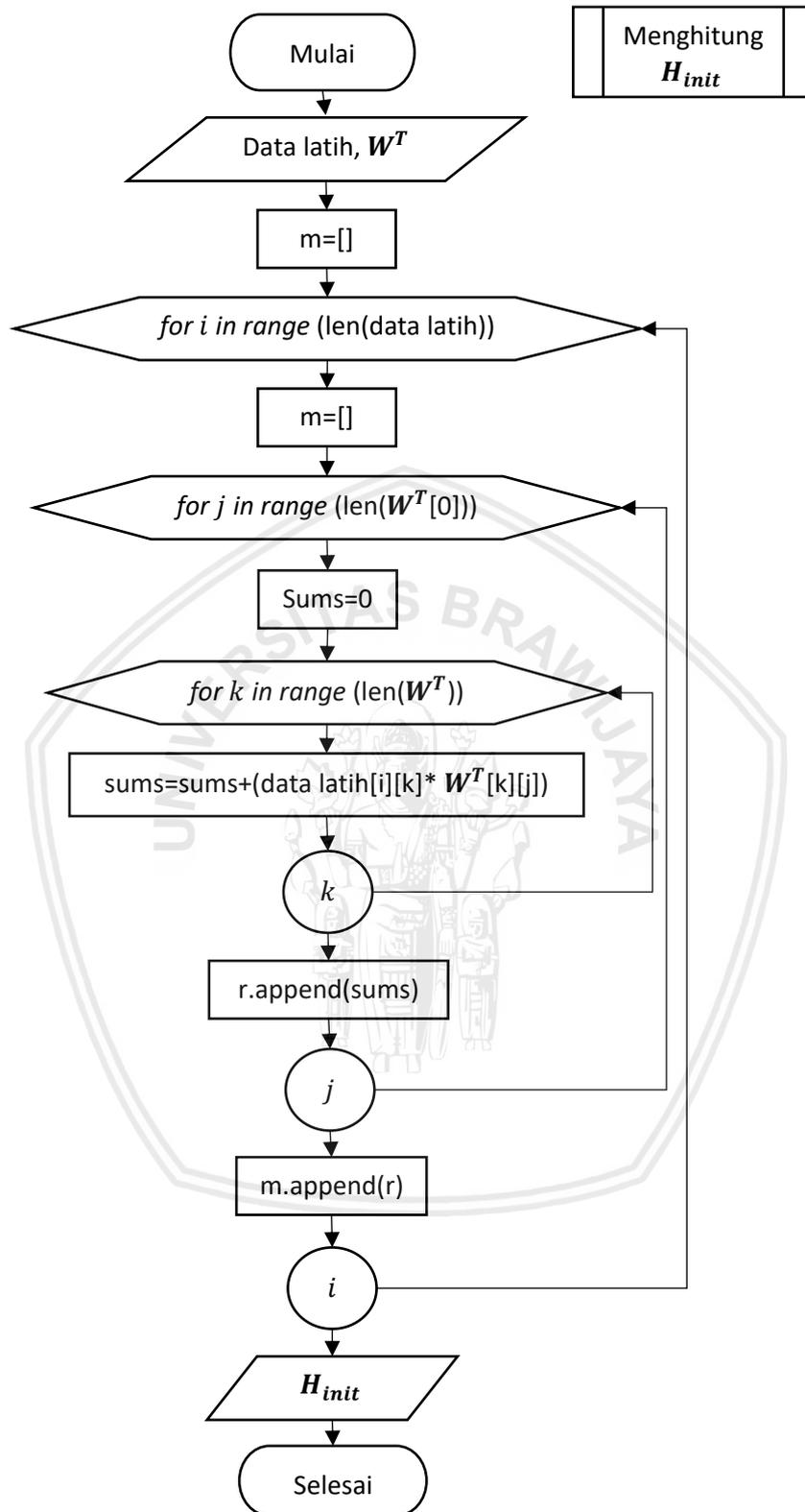
**Langkah 2:** Menghitung matriks inialisasi keluaran *hidden neuron* ( $H_{init}$ ). Perhitungan  $H_{init}$  membutuhkan bobot *transpose* ( $W^T$ ), diagram alir menghitung  $W^T$  dapat dilihat pada Gambar 4.6 dan diagram alir menghitung  $H_{init}$  dapat dilihat pada Gambar 4.7.



**Gambar 4.6 Diagram Alir Menghitung  $W^T$**

Perhitungan  $W^T$  pada Gambar 4.6 adalah:

1. Input data terdiri dari  $W$ .
2. Dilakukan perulangan  $W^T$  sebanyak input *neuron* dikali *hidden neuron*.
3. *Output* yang dihasilkan adalah  $W^T$ .



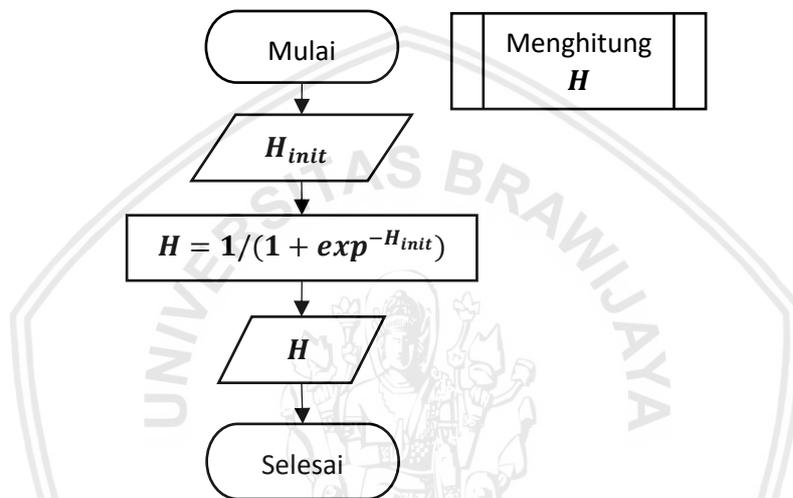
Gambar 4.7 Diagram Alir Menghitung  $H_{init}$

Perhitungan  $H_{init}$  pada Gambar 4.7 adalah:

1. Input data terdiri dari data latih dan  $W^T$ .

2. Perkalian matriks dilakukan dengan menjumlahkan hasil perkalian setiap baris matriks data latih ke setiap kolom matriks  $W^T$ . Perulangan pertama untuk menentukan jumlah baris matriks baru sebanyak jumlah baris pada matriks data latih. Perulangan kedua untuk menentukan jumlah kolom matriks baru sebanyak jumlah kolom pada matriks  $W^T$ . Perulangan ketiga digunakan untuk melakukan proses penjumlahan hasil perkalian baris dan kolom. Proses tersebut akan menghasilkan elemen matriks baru yang disimpan pada koordinat tersebut.
3. *Output* yang dihasilkan adalah  $H_{init}$ .

**Langkah 3:** Menghitung matriks keluaran *hidden neuron* ( $H$ ). Diagram alir dapat dilihat pada Gambar 4.8.



**Gambar 4.8 Diagram Alir Menghitung  $H$**

Perhitungan  $H$  pada Gambar 4.8 adalah:

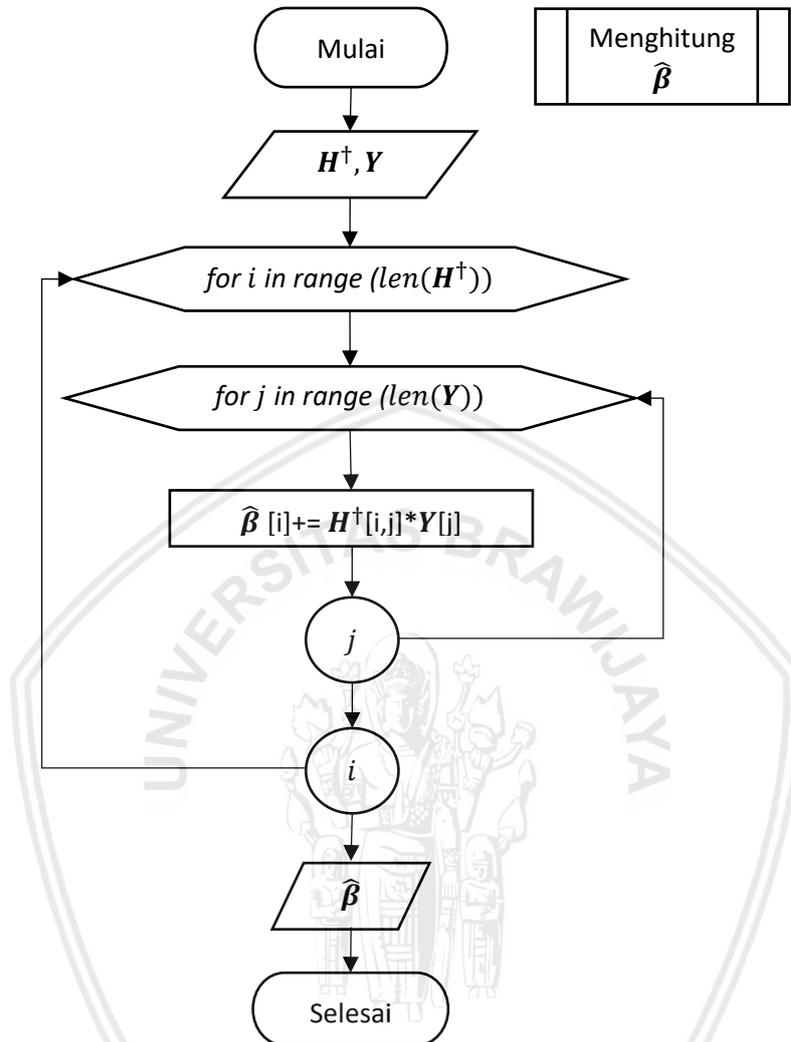
1. Input terdiri dari  $H_{init}$ .
2. Menghitung  $H$  berdasarkan Persamaan 2.6.
3. *Output* yang dihasilkan adalah  $H$ .

**Langkah 4:** Menghitung matriks  $H^\dagger$ . Untuk menghitung  $H^\dagger$  diperlukan perhitungan  $H^T$  dan perkalian  $H^T$  dengan  $H$ . Diagram alir untuk menghitung  $H^T$  sama dengan diagram alir menghitung  $W^T$  pada Gambar 4.6 yang berbeda adalah pada input data yang terdiri dari  $H$  dan *output* yang dihasilkan adalah  $H^T$ .

Diagram alir untuk menghitung perkalian  $H^T$  dengan  $H$  sama dengan diagram alir menghitung  $H_{init}$  pada Gambar 4.7 yang berbeda adalah pada input data yang terdiri dari  $H^T$  dan  $H$ , *output* yang dihasilkan adalah  $H^T \cdot H$ .

Untuk menghitung  $H^\dagger$  berdasarkan Persamaan 2.7. Diagram alir untuk menghitung  $H^\dagger$  sama dengan diagram alir menghitung  $H_{init}$  pada Gambar 4.7 yang berbeda adalah pada input data yang terdiri dari  $H^T$  dan  $H^T \cdot H$ , *output* yang dihasilkan adalah  $H^\dagger$ .

**Langkah 5:** Menghitung bobot keluaran ( $\hat{\beta}$ ). Diagram alir dapat dilihat pada Gambar 4.9.



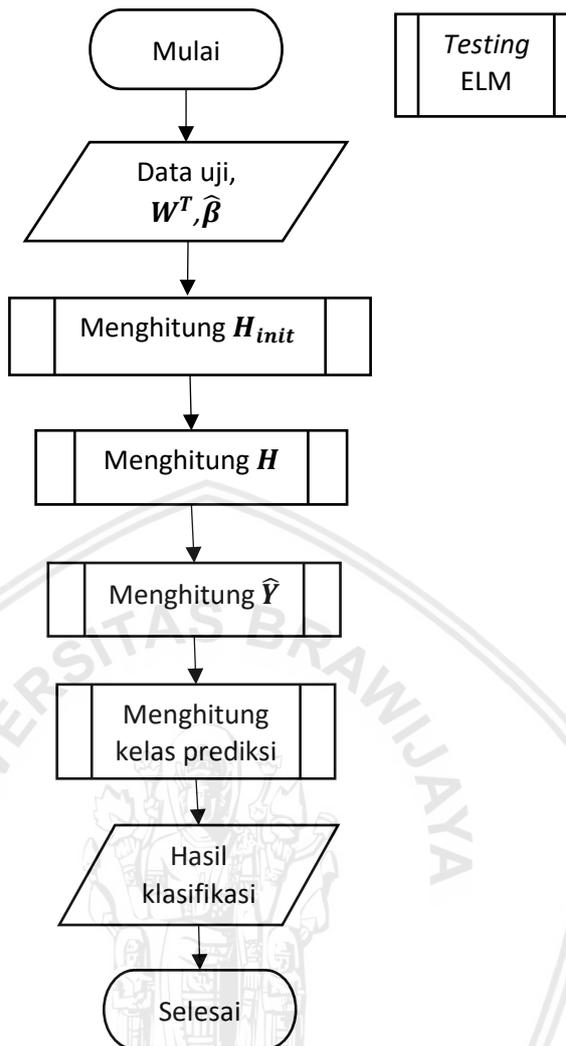
**Gambar 4.9 Diagram Alir Menghitung  $\hat{\beta}$**

Perhitungan  $\hat{\beta}$  pada Gambar 4.9 adalah:

1. Input data terdiri dari  $H^T$  dan  $Y$ .
2. Menghitung  $\hat{\beta}$  berdasarkan Persamaan 2.8 lalu dilakukan perulangan sebanyak  $H^T$  dikali  $Y$ .
3. *Output* yang dihasilkan adalah  $\hat{\beta}$ .

#### 4.3.2.2 Testing ELM

Tujuan dari proses *testing* ELM adalah untuk mendapatkan nilai  $\hat{Y}$  prediksi dari perkalian *hidden neuron* ( $H$ ) dengan bobot keluaran ( $\hat{\beta}$ ) yang diperoleh dari proses *training* ELM, lalu dilakukan perhitungan kelas prediksi untuk mendapatkan hasil klasifikasi. Diagram alir *testing* ELM dapat dilihat pada Gambar 4.10.



**Gambar 4.10 Diagram Alir Testing ELM**

Langkah perhitungan *testing ELM* berdasarkan Gambar 4.10 adalah:

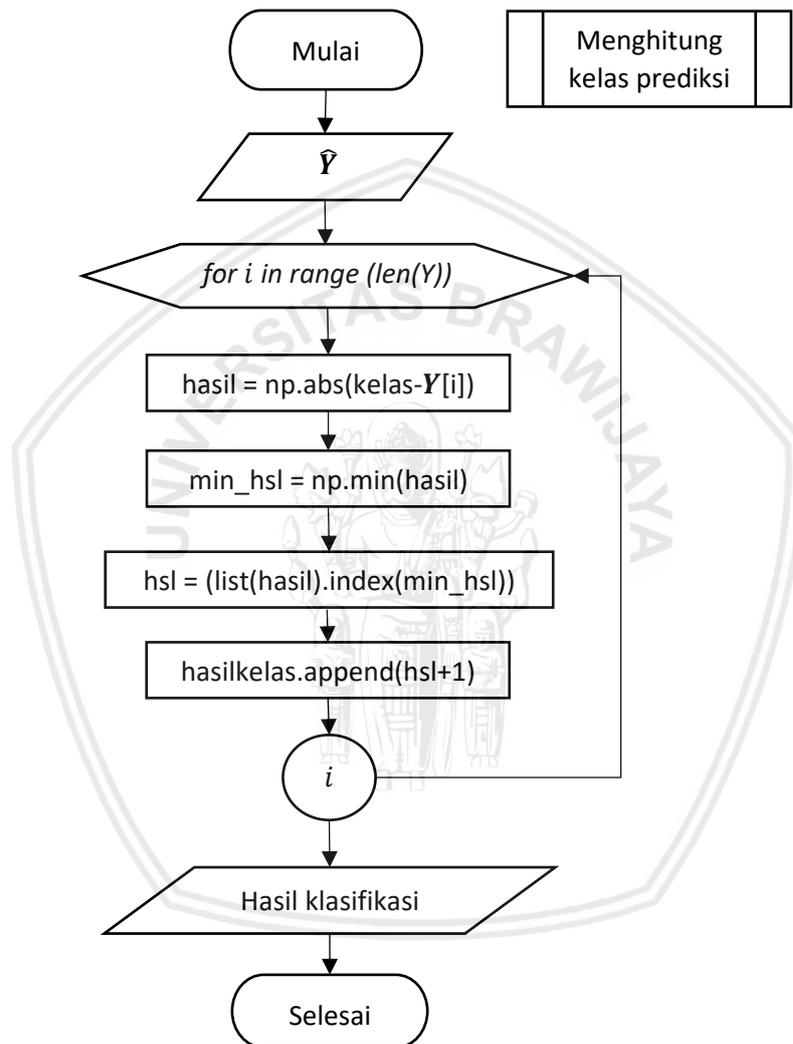
1. Input data terdiri dari data uji,  $W^T$  dan  $\hat{\beta}$ .
2. Menghitung matriks inialisasi keluaran *hidden neuron* ( $H_{init}$ ).
3. Menghitung matriks keluaran *hidden neuron* ( $H$ ).
4. Menghitung  $\hat{Y}$  prediksi.
5. Menghitung kelas prediksi.
6. *Output* yang diperoleh adalah hasil klasifikasi.

**Langkah 1:** Menghitung matriks inialisasi keluaran *hidden neuron* ( $H_{init}$ ) berdasarkan Persamaan 2.5. Diagram alir yang digunakan sama dengan diagram alir menghitung  $H_{init}$  pada Gambar 4.7 yang berbeda adalah pada input data yang terdiri dari data uji dan  $W^T$ , *output* yang dihasilkan adalah  $H_{init}$ .

**Langkah 2:** Menghitung matriks keluaran *hidden neuron* ( $H$ ) berdasarkan Persamaan 2.6. Diagram alir yang digunakan sama dengan diagram alir menghitung  $H$  pada Gambar 4.8 yang berbeda adalah pada input data yang terdiri dari  $H_{init}$  yang diperoleh dari langkah sebelumnya dan *output* yang dihasilkan adalah  $H$ .

**Langkah 3:** Menghitung  $\hat{Y}$  prediksi berdasarkan Persamaan 2.9. Diagram alir yang digunakan sama dengan diagram alir menghitung  $\hat{\beta}$  pada Gambar 4.9 yang berbeda adalah pada input data yang terdiri dari  $H$  dan  $\hat{\beta}$ , *output* yang dihasilkan adalah  $\hat{Y}$ .

**Langkah 4:** Menghitung kelas prediksi dengan diagram alir yang dapat dilihat pada Gambar 4.11.



**Gambar 4.11 Diagram Alir Menghitung Kelas Prediksi**

Perhitungan kelas prediksi pada Gambar 4.11 adalah:

1. Input data terdiri dari  $\hat{Y}$ .
2. Menghitung hasil prediksi berdasarkan Persamaan 2.10 dengan mencari selisih terkecil antara  $\hat{Y}$  prediksi dengan kelas sebenarnya ( $Y$ ) dan dilakukan perulangan sebanyak panjang  $\hat{Y}$ . Kelas yang memiliki selisih terkecil akan menjadi kelas prediksi pada notasi SMILES.
3. *Output* yang dihasilkan adalah hasil klasifikasi.

## 4.4 Perhitungan Manual

Tujuan dari perhitungan manual adalah untuk mengetahui gambaran secara umum proses perhitungan pada sistem klasifikasi senyawa kimia dengan notasi SMILES menggunakan metode ELM. Proses perhitungan pada ELM dibagi menjadi 2, yaitu proses *training* dan *testing*.

Ketentuan dataset perhitungan manual yang digunakan untuk data latih dan data uji adalah:

1. Menggunakan 15 input *neuron* (fitur) pada notasi SMILES.
2. Menggunakan 2 kelas klasifikasi, yaitu Kanker dan Metabolisme.
3. Menggunakan 10 data latih dan 4 data uji.

### 4.4.1 Preprocessing Notasi SMILES

Sebelum melakukan perhitungan ELM, terlebih dahulu dilakukan *preprocessing* terhadap notasi SMILES. *Preprocessing* dilakukan dengan membaca fitur – fitur yang ada pada notasi SMILES, yaitu *B, C, N, O, P, S, F, Cl, Br, I, OH, @, =, #, dan ()*. Setelah itu nilai setiap fitur dibagi dengan panjang notasi SMILES. Hasil *preprocessing* data latih dapat dilihat pada Tabel 4.1.

Untuk mendapatkan nilai fitur notasi SMILES CC(C)(C)C(=O)N pada kelas Kanker (1), maka:

$$\begin{aligned} \text{Nilai fitur } x &= \frac{\text{jumlah fitur}}{\text{panjang fitur}} & \text{Nilai fitur } P &= \frac{0}{11} = 0 \\ \text{Nilai fitur } B &= \frac{0}{11} = 0 & \text{Nilai fitur } S &= \frac{0}{11} = 0 \\ \text{Nilai fitur } C &= \frac{5}{11} = 0,455 & \text{Nilai fitur } F &= \frac{0}{11} = 0 \\ \text{Nilai fitur } N &= \frac{1}{11} = 0,09 & \text{Nilai fitur } Cl &= \frac{0}{11} = 0 \\ \text{Nilai fitur } O &= \frac{1}{11} = 0,09 & \text{Nilai fitur } Br &= \frac{0}{11} = 0 \\ \text{Nilai fitur } I &= \frac{0}{11} = 0 & \text{Nilai fitur rangkap 2(=)} &= \frac{1}{11} = 0,09 \\ \text{Nilai fitur } OH &= \frac{0}{11} = 0 & \text{Nilai fitur } \# &= \frac{0}{11} = 0 \\ \text{Nilai fitur } @ &= \frac{0}{11} = 0 & & \\ \text{Nilai fitur } () &= \frac{3}{11} = 0,273 & & \end{aligned}$$

### 4.4.2 Perhitungan *Training* ELM

**Langkah 1:** Melakukan *preprocessing* pada notasi SMILES yang digunakan sebagai data latih untuk menentukan nilai dari 15 fitur. Kelas yang digunakan adalah kelas 1 sebagai kelas kanker dan kelas 2 sebagai kelas metabolisme. Nilai dari 15 fitur, masing-masing dibagi panjang notasi SMILES tersebut. Berikut adalah hasil dari *preprocessing* data latih:

**Tabel 4.1 Data Latih**

No	Notasi SMILES	Hasil preprocessing														Kelas	
		B	C	N	O	P	S	F	Cl	Br	I	OH	@	=	#		()
1	<chem>CC(C)(C)C(=O)N</chem>	0	0,455	0,09	0,09	0	0	0	0	0	0	0	0	0,09	0	0,273	1
2	<chem>C(C(=O)N)S</chem>	0	0,286	0,143	0,143	0	0,143	0	0	0	0	0	0	0,143	0	0,143	1
3	<chem>CCN(C(=O)N)N=O</chem>	0	0,273	0,273	0,182	0	0	0	0	0	0	0	0	0,182	0	0,091	1
4	<chem>CCCC(=O)N(C)C</chem>	0	0,545	0,091	0,091	0	0	0	0	0	0	0	0	0,091	0	0,182	1
5	<chem>C(CBr)F</chem>	0	0,4	0	0	0	0	0,2	0	0,2	0	0	0	0	0	0,2	1
6	<chem>CC1=CC=C(C=C1)N(SC(F)(Cl)Cl)S(=O)(=O)N(C)C</chem>	0	0,345	0,069	0,069	0	0,069	0,035	0,069	0	0	0	0	0,172	0	0,172	2
7	<chem>COP(=S)(OC)SCN1C(=O)C2=CC=CC=C2C1=O</chem>	0	0,393	0,036	0,143	0,036	0,071	0	0	0	0	0	0	0,214	0	0,107	2
8	<chem>CC(C)OP(=S)(OC(C)C)SCCNS(=O)(=O)C1=CC=CC=C1</chem>	0	0,412	0,029	0,118	0,029	0,088	0	0	0	0	0	0	0,177	0	0,147	2
9	<chem>COP(=S)(OC)S</chem>	0	0,2	0	0,2	0,1	0,2	0	0	0	0	0	0	0,1	0	0,2	2
10	<chem>CC(C=C)C(=O)C</chem>	0	0,546	0	0,091	0	0	0	0	0	0	0	0	0,182	0	0,182	2

**Langkah 2:** Menentukan bobot ( $w_{ij}$ ) secara *random* sebanyak *hidden neuron* dikali *input neuron*. Nilai bobot ( $W$ ) yang digunakan adalah rentang dari -0,5 sampai 0,5. Jumlah *hidden neuron* yang digunakan sebanyak 5 dan jumlah *input neuron* atau fitur yang digunakan pada notasi SMILES sebanyak 15, jadi matriks bobot ( $w_{ij}$ ) yang digunakan adalah 5 x 15. Nilai bobot ( $W$ ) dapat dilihat pada Tabel 4.2.

**Tabel 4.2 Matriks Bobot ( $W$ )**

$W$	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	0,5	-0,1	0,2	0,3	-0,3	0,1	0,2	0,5	-0,1	0,2	0,3	-0,3	0,3	0,4	0,5
2	0,4	0,1	-0,4	-0,5	0,4	-0,4	-0,5	0,4	0,1	-0,4	-0,5	0,4	0,3	0,3	0,2
3	-0,2	0,5	0,1	-0,1	0,2	-0,3	0,1	-0,2	0,5	0,1	-0,1	0,2	0,2	0,4	0,5
4	-0,1	-0,3	0,2	0,5	-0,1	0,2	0,3	-0,3	-0,2	-0,1	-0,2	0,5	0,4	-0,3	-0,5
5	-0,5	-0,4	0,1	0,4	0,1	-0,4	-0,5	0,4	0,5	-0,2	-0,1	0,1	-0,4	0,3	0,4

Matriks bobot  $W$  pada Tabel 4.2 ditransposisi sehingga diperoleh  $W^T$  pada Tabel 4.3.

**Tabel 4.3 Matriks Bobot Transpose ( $W^T$ )**

$W^T$	1	2	3	4	5
1	0,5	0,4	-0,2	-0,1	-0,5
2	-0,1	0,1	0,5	-0,3	-0,4
3	0,2	-0,4	0,1	0,2	0,1
4	0,3	-0,5	-0,1	0,5	0,4
5	-0,3	0,4	0,2	-0,1	0,1
6	0,1	-0,4	-0,3	0,2	-0,4
7	0,2	-0,5	0,1	0,3	-0,5
8	0,5	0,4	-0,2	-0,3	0,4
9	-0,1	0,1	0,5	-0,2	0,5
10	0,2	-0,4	0,1	-0,1	-0,2
11	0,3	-0,5	-0,1	-0,2	-0,1
12	-0,3	0,4	0,2	0,5	0,1
13	0,3	0,3	0,2	0,4	-0,4
14	0,4	0,3	0,4	-0,3	0,3
15	0,5	0,2	0,5	-0,5	0,4

**Langkah 3:** Menghitung matriks  $H_{init}$  berdasarkan Persamaan 2.5, dengan  $W^T$  diperoleh dari Tabel 4.3 dan  $X$  diperoleh dari Tabel 4.1, sehingga diperoleh matriks  $H_{init}$  pada Tabel 4.4.

$$\begin{aligned}
 H_{init} &= X \cdot W^T \\
 h_{init\ 1,1} &= ((0 \times 0,5) + (0,455 \times -0,1) + (0,09 \times 0,2) + (0,09 \times 0,3) + \\
 &\quad (0 \times -0,3) \dots \dots \dots (0,273 \times 0,5)) \\
 &= 0,163
 \end{aligned}$$

Tabel 4.4 Matriks  $H_{init}$

$H_{init}$	1	2	3	4	5
1	0,163	0,0461	0,382	-0,174	-0,0638
2	0,1716	-0,0858	0,2002	0,0286	-0,1001
3	0,182	-0,1001	0,2275	0,091	-0,0455
4	0,1093	0,0363	0,3817	-0,1544	-0,1361
5	0,12	0	0,42	-0,2	-0,08
6	0,186	0,0409	0,2619	-0,0688	-0,121
7	0,1248	0,025	0,268	0,0035	-0,164
8	0,1267	0,0295	0,2854	-0,0468	-0,159
9	0,16	-0,05	0,16	0,01	-0,03
10	0,1183	0,1001	0,3913	-0,1365	-0,182

Langkah 4: Menghitung matriks keluaran *hidden neuron* ( $H$ ) berdasarkan Persamaan 2.6, dengan  $H_{init}$  diperoleh dari Tabel 4.4, sehingga diperoleh matriks  $H$  pada Tabel 4.5.

$$H = \frac{1}{(1 + \exp(-H_{init} 1,1))}$$

$$h_{1,1} = \frac{1}{(1 + \exp(-0,163))}$$

$$= 0,5407$$

Tabel 4.5 Matriks  $H$

$H$	1	2	3	4	5
1	0,5407	0,5115	0,5944	0,4566	0,4841
2	0,5428	0,4786	0,5499	0,5071	0,475
3	0,5454	0,475	0,5566	0,5227	0,4886
4	0,5273	0,5091	0,5943	0,4615	0,466
5	0,53	0,5	0,6035	0,4502	0,48
6	0,5464	0,5102	0,5651	0,4828	0,4698
7	0,5312	0,5062	0,5666	0,5009	0,4591
8	0,5316	0,5074	0,5709	0,4883	0,4603
9	0,5399	0,4875	0,5399	0,5025	0,4925
10	0,5295	0,525	0,5966	0,4659	0,4546



**Langkah 5:** Menghitung matriks  $H^\dagger$ . Matriks  $H^T$  dihitung terlebih dulu dengan men-*transpose* matriks  $H$  yang diperoleh dari Tabel 4.5. Hasil dari perhitungan  $H^T$  dapat dilihat pada Tabel 4.6

**Tabel 4.6 Matriks  $H^T$**

$H^T$	1	2	3	4	5	6	7	8	9	10
1	0,5407	0,5428	0,5454	0,5273	0,53	0,5464	0,5312	0,5316	0,5399	0,5295
2	0,5115	0,4786	0,475	0,5091	0,5	0,5102	0,5062	0,5074	0,4875	0,525
3	0,5944	0,5499	0,5566	0,5943	0,6035	0,5651	0,5666	0,5709	0,5399	0,5966
4	0,4566	0,5071	0,5227	0,4615	0,4502	0,4828	0,5009	0,4883	0,5025	0,4659
5	0,4841	0,475	0,4886	0,466	0,48	0,4698	0,4591	0,4603	0,4925	0,4546



Matriks  $H^T$  dari Tabel 4.6 dikalikan dengan matriks  $H$  dari Tabel 4.5. Hasil dari perhitungan  $H^T \cdot H$  dapat dilihat pada Tabel 4.7.

**Tabel 4.7 Matriks  $H^T \cdot H$**

$H^T \cdot H$	1	2	3	4	5
1	28,785	26,874	30,772	25,966	2,538
2	26,874	25,128	28,771	24,218	23,687
3	30,772	28,771	32,965	27,718	2,713
4	25,966	24,218	27,718	23,466	22,893
5	2,538	23,687	2,713	22,893	22,389

Nilai  $H^T \cdot H$  dari Tabel 4.7 diinvers. Hasil dari perhitungan  $(H^T \cdot H)^{-1}$  dapat dilihat pada Tabel 4.8.

**Tabel 4.8 Matriks  $(H^T \cdot H)^{-1}$**

$(H^T \cdot H)^{-1}$	1	2	3	4	5
1	4914,9177	-1651,1675	-3,8862	-1274,8588	-2516,4411
2	-1651,1634	1393,0674	-696,5168	177,4204	1060,5505
3	-3,8897	-696,5157	706,2957	261,9485	-382,4104
4	-1274,8601	177,4219	261,9476	604,3778	322,0789
5	-2516,4401	1060,5522	-382,4121	322,0780	1865,1454

Menghitung matriks  $H^\dagger$  berdasarkan Persamaan 2.7, dengan  $(H^T \cdot H)^{-1}$  diperoleh dari Tabel 4.8 dan  $H^T$  diperoleh dari Tabel 4.6. Hasil perhitungan  $H^\dagger$  dapat dilihat pada Tabel 4.9.

$$\begin{aligned}
 H^\dagger &= (H^T \cdot H)^{-1} \cdot H^T \\
 h_{1,1}^\dagger &= (4914,9177 \times 0,5407) + (-1651,1675 \times 0,5115) \dots + \\
 &\quad (-2516,4411 \times 0,4841) \\
 &= 10,1701
 \end{aligned}$$

Tabel 4.9 Matriks  $H^+$

$H^+$	1	2	3	4	5	6	7	8	9	10
1	10,1701	33,6248	-2,0021	-12,2985	-5,0160	42,9837	-21,3214	-7,9664	-33,3747	-4,5698
2	0,2664	-18,8399	-15,5472	0,7123	-9,9156	-1,0759	9,3163	6,2158	23,0519	6,2873
3	-4,0958	4,1460	10,2561	5,7799	10,2777	-11,5547	1,1536	-0,3857	-17,0219	1,8353
4	-10,9519	-3,5434	8,1117	2,7647	-2,1664	-14,8823	11,6671	5,1826	1,9310	2,3555
5	4,5672	-19,3740	-1,7860	3,5655	6,1523	-18,1597	1,1911	-2,1718	32,3179	-5,9010

**Langkah 5:** Menghitung bobot keluaran ( $\hat{\beta}$ ) berdasarkan Persamaan 2.8, dengan  $H^+$  diperoleh dari Tabel 4.9 dan  $Y$  adalah kelas sebenarnya yang diperoleh dari Tabel 4.1 pada kolom Kelas. Hasil dari perhitungan bobot keluaran ( $\hat{\beta}$ ) dapat dilihat pada Tabel 4.10.

$$\hat{\beta} = H^+ \cdot Y$$

$$\begin{aligned} \hat{\beta}_{1,1} &= (10,1701 \times 1) + (33,6248 \times 1) + (-2,0021 \times 1) \dots + (-4,5698 \times 2) \\ &= -24,0188 \end{aligned}$$

Tabel 4.10 Matriks  $\hat{\beta}$

	$\hat{\beta}$
1	-24,0188
2	44,2667
3	-25,5830
4	6,7224
5	7,6779

### 4.4.3 Perhitungan *Testing* ELM

Tabel 4.11 Data Uji

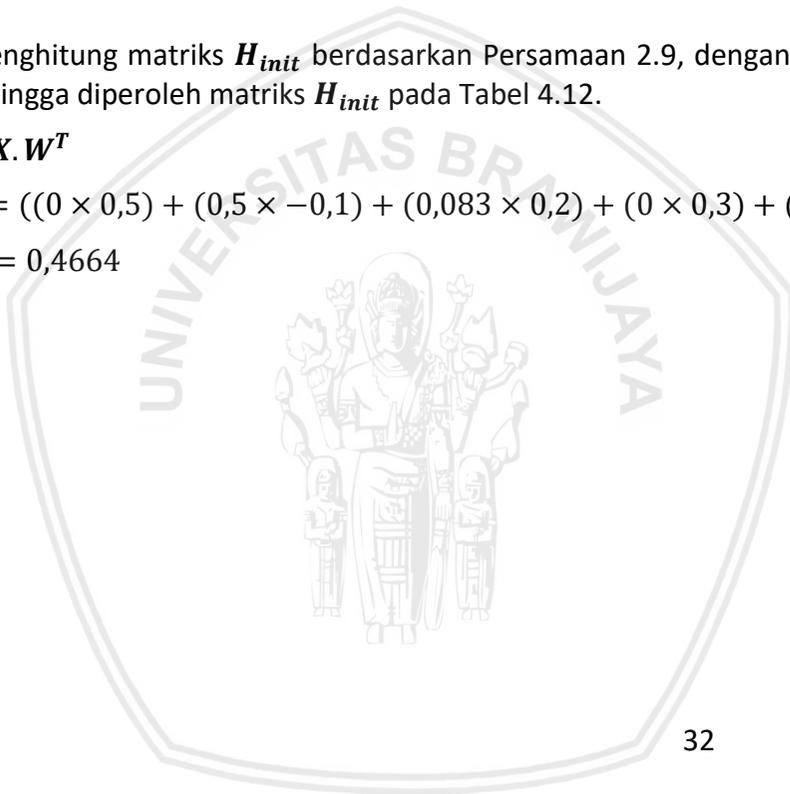
No	Notasi SMILES	Hasil preprocessing														Kelas	
		B	C	N	O	P	S	F	Cl	Br	I	OH	@	:=	#		()
1	<chem>C1=CC=C(C(=C1)N)Br</chem>	0	0,5	0,083	0	0	0	0	0	0,083	0	0	0	0,25	0	0,833	1
2	<chem>CC(=O)NC(C)(C)C</chem>	0	0,5	0,083	0,083	0	0	0	0	0	0	0	0	0,083	0	0,25	1
3	<chem>COC1=CC=C(C=C1)C2=CC=CC=C2</chem>	0	0,619	0	0,048	0	0	0	0	0	0	0	0	0,286	0	0,048	2
4	<chem>CN(C1=CC=CC=C1)N=O</chem>	0	0,467	0,133	0,067	0	0	0	0	0	0	0	0	0,267	0	0,067	2

**Langkah 1:** Menghitung matriks  $H_{init}$  berdasarkan Persamaan 2.9, dengan  $W^T$  diperoleh dari Tabel 4.3 dan data uji ( $X$ ) diperoleh dari Tabel 4.11, sehingga diperoleh matriks  $H_{init}$  pada Tabel 4.12.

$$H_{init} = X \cdot W^T$$

$$h_{init\ 1,1} = ((0 \times 0,5) + (0,5 \times -0,1) + (0,083 \times 0,2) + (0 \times 0,3) + (0 \times -0,3) \dots \dots \dots (0,833 \times 0,5))$$

$$= 0,4664$$



**Tabel 4.12 Matriks  $H_{init}$**

$H_{init}$	1	2	3	4	5
1	0,4664	0,2667	0,7663	-0,4665	0,083
2	0,1414	0,0502	0,3916	-0,1837	-0,0917
3	0,0623	0,1333	0,3859	-0,0713	-0,3236
4	0,1136	0,0535	0,327	-0,0067	-0,2267

**Langkah 2:** Menghitung matriks keluaran *hidden neuron* ( $H$ ) berdasarkan Persamaan 2.5, dengan  $H_{init}$  diperoleh dari Tabel 4.12 sehingga diperoleh matriks  $H$  pada Tabel 4.13.

$$H = \frac{1}{(1 + \exp(-H_{init}))}$$

$$h_{1,1} = \frac{1}{(1 + \exp(-0,4664))}$$

$$= 0,6145$$

**Tabel 4.13 Matriks  $H$**

$H$	1	2	3	4	5
1	0,6145	0,5663	0,6827	0,3854	0,5207
2	0,5353	0,5125	0,5967	0,4542	0,4771
3	0,5156	0,5333	0,5953	0,4822	0,4198
4	0,5284	0,5134	0,5810	0,4983	0,4436

**Langkah 3:** Menghitung  $\hat{Y}$  prediksi berdasarkan Persamaan 2.9, dengan  $H$  diperoleh dari Tabel 4.13 dan  $\hat{\beta}$  diperoleh dari Tabel 4.10. Hasil dari perhitungan  $\hat{Y}$  prediksi dapat dilihat pada Tabel 4.14.

$$\hat{Y} = H \cdot \hat{\beta}$$

$$\hat{Y}_{1,1} = (0,6145 \times -24,0188)$$

$$+ (0,5663 \times 44,2667) + \dots + (0,5207 \times 7,6779)$$

$$= -0,5696$$

**Tabel 4.14 Matriks  $\hat{Y}$**

	$\hat{Y}$
1	-0,5696
2	1,2836
3	2,4581
4	1,9256

**Langkah 4:** Menghitung kelas prediksi berdasarkan Persamaan 2.10 dengan mencari selisih terkecil antara  $\hat{Y}$  prediksi pada Tabel 4.14 dengan kelas sebenarnya ( $Y$ ) pada Tabel 4.11 kolom kelas. Hasil perhitungan kelas prediksi dapat dilihat pada Tabel 4.15

Tabel 4.15 Kelas Prediksi

	$\hat{Y}$	$Y$	$ \hat{Y} - 1 $	$ \hat{Y} - 2 $	Kelas Prediksi
1	-0,5696	1	1,5696	2,5696	1
2	1,2836	1	0,2836	0,7164	1
3	2,4581	2	1,4581	0,4581	2
4	1,9256	2	0,9256	0,0744	2

Berdasarkan perhitungan manual ELM, maka didapatkan hasil perbandingan antara kelas sebenarnya ( $Y$ ) dan kelas prediksi dengan nilai yang sama sebanyak 4 dari 4 data *testing*. Sehingga nilai akurasi dapat dihitung dengan persamaan akurasi sebagai berikut, dengan  $N$  adalah jumlah data *testing*.

$$\begin{aligned}
 \text{Akurasi} &= \frac{\text{hasil perbandingan yang bernilai benar}}{N} \times 100\% \\
 &= \frac{4}{4} \times 100\% \\
 &= 100\%
 \end{aligned}$$

#### 4.5 Perancangan Pengujian

Pada tahap ini dilakukan pengujian terhadap sistem klasifikasi senyawa kimia dengan notasi SMILES menggunakan metode ELM. Pengujian yang dilakukan ada 2, yaitu pengujian jumlah *hidden neuron* dengan metode *k-fold cross validation* dan pengujian jumlah fitur dengan metode *k-fold cross validation*.

##### 4.5.1 Pengujian Jumlah *Hidden Neuron* dengan Metode *K-fold Cross Validation*

Metode *k-fold cross validation* digunakan untuk mengevaluasi kinerja ELM pada data SMILES. Pengujian jumlah *hidden neuron* dengan metode *k-fold cross validation* adalah pengujian jumlah *hidden neuron* menggunakan 20% data uji dari keseluruhan data yang dipilih secara bergantian dengan nilai  $k=5$  sehingga diperlukan 5 kali pengujian dengan data latih dan data uji yang berbeda lalu hasil akurasi dari kelima pengujian tersebut dirata-rata. Pada setiap pengujian dilakukan 5 kali *running* pada program lalu hasilnya dirata-rata.

Pengujian jumlah *hidden neuron* dengan metode *k-fold cross validation* dilakukan pada 2 skenario berbeda, yaitu pengujian jumlah *hidden neuron* untuk 2 kelas dan pengujian jumlah *hidden neuron* untuk 3 kelas. Pengujian jumlah *hidden neuron* untuk 2 kelas terdiri dari 3 skenario yang berbeda, yaitu kelas Metabolisme dengan Kanker, Radang dengan Kanker, dan Metabolisme dengan Radang. Pengujian jumlah *hidden neuron* untuk 3 kelas terdiri dari 1 skenario yaitu kelas Metabolisme, Radang, dan Kanker.

##### 4.5.1.1 Pengujian 2 Kelas

Pengujian 2 kelas terdiri dari 3 skenario yang berbeda, yaitu kelas Metabolisme dengan Kanker, Radang dengan Kanker, dan Metabolisme dengan

Radang. Pengujian jumlah *hidden neuron* dilakukan sebanyak 8 kali, yaitu dengan jumlah *hidden neuron* 5, 10, 15, 20, 25, 30, 35, dan 40. Pengujian *k-fold cross validation* menggunakan 20% data uji dari keseluruhan data yang dipilih secara bergantian dengan nilai  $k=5$  sehingga diperlukan 5 kali pengujian dengan data latih dan data uji yang berbeda lalu hasil akurasi dari kelima pengujian tersebut dirata-rata.

### 1. Kelas Metabolisme dan Kanker

Pada kelas Metabolisme dan Kanker data latih yang digunakan sebanyak 748 dengan 501 data kelas Metabolisme dan 247 data kelas Kanker. Data uji yang digunakan sebanyak 187 dengan 125 data kelas Metabolisme dan 62 data kelas Kanker. Hasil pengujian pada kelas Metabolisme dan Kanker dapat dilihat pada Tabel 4.16.

**Tabel 4.16 Perancangan Pengujian Jumlah *Hidden Neuron* Kelas Metabolisme dan Kanker**

<i>Cross Validation</i> Jumlah <i>Hidden Neuron</i>	1	2	3	4	5	Rata-Rata Akurasi
5						
10						
15						
20						
25						
30						
35						
40						

### 2. Kelas Radang dan Kanker

Pada kelas Radang dan Kanker data latih yang digunakan sebanyak 339 dengan 92 data kelas Radang dan 247 data kelas Kanker. Data uji yang digunakan sebanyak 84 dengan 22 data kelas Radang dan 62 data kelas Kanker. Hasil pengujian pada kelas Radang dan Kanker dapat dilihat pada Tabel 4.17.

**Tabel 4.17 Perancangan Pengujian Jumlah *Hidden Neuron* Kelas Radang dan Kanker**

<i>Cross Validation</i> Jumlah <i>Hidden Neuron</i>	1	2	3	4	5	Rata-Rata Akurasi
5						
10						
15						
20						
25						



30						
35						
40						

### 3. Kelas Metabolisme dan Radang

Pada kelas Metabolisme dan Radang data latih yang digunakan sebanyak 593 dengan 501 data kelas Metabolisme dan 92 data kelas Radang. Data uji yang digunakan sebanyak 147 dengan 125 data kelas Metabolisme dan 22 data kelas Radang. Hasil pengujian pada kelas Metabolisme dan Radang dapat dilihat pada Tabel 4.18.

**Tabel 4.18 Perancangan Pengujian Jumlah *Hidden Neuron* Kelas Metabolisme dan Radang**

<i>Cross Validation</i> Jumlah <i>Hidden Neuron</i>	1	2	3	4	5	Rata-Rata Akurasi
5						
10						
15						
20						
25						
30						
35						
40						

#### 4.5.1.2 Pengujian 3 Kelas

Pengujian 3 kelas terdiri dari 1 skenario yaitu kelas Metabolisme, Radang, dan Kanker. Pengujian jumlah *hidden neuron* dilakukan sebanyak 4 kali, yaitu dengan jumlah *hidden neuron* 100, 200, 300, dan 400. Pengujian *k-fold cross validation* menggunakan 20% data uji dari keseluruhan data yang dipilih secara bergantian dengan nilai  $k=5$  sehingga diperlukan 5 kali pengujian dengan data latih dan data uji yang berbeda lalu hasil akurasi dari kelima pengujian tersebut dirata-rata.

##### 1. Kelas Metabolisme, Radang, dan Kanker

Pada kelas Metabolisme, Radang, dan Kanker data latih yang digunakan sebanyak 840 dengan 501 data kelas Metabolisme, 92 data kelas Radang, dan 247 data kelas Kanker. Data uji yang digunakan sebanyak 209 dengan 125 data kelas Metabolisme, 22 data kelas Radang, dan 62 data kelas Kanker. Hasil pengujian pada kelas Metabolisme, Radang, dan Kanker dapat dilihat pada Tabel 4.19.

**Tabel 4.19 Perancangan Pengujian Jumlah *Hidden Neuron* Kelas Metabolisme, Radang, dan Kanker**

<i>Cross Validation</i> Jumlah <i>Hidden Neuron</i>	1	2	3	4	5	Rata-Rata Akurasi
100						
200						
300						
400						

#### 4.5.2 Pengujian Jumlah Fitur dengan Metode *K-fold Cross Validation*

Pengujian jumlah fitur digunakan untuk menunjukkan hasil akurasi pada jumlah fitur yang berbeda. Pada pengujian jumlah fitur digunakan 2 kondisi yaitu ketika fitur berjumlah 11 dan 15. Jumlah kelas yang digunakan sebanyak 3, yaitu Metabolisme, Radang, dan Kanker. Pada kelas Metabolisme, Radang, dan Kanker data latih yang digunakan sebanyak 840 dengan 501 data kelas Metabolisme, 92 data kelas Radang, dan 247 data kelas Kanker. Data uji yang digunakan sebanyak 209 dengan 125 data kelas Metabolisme, 22 data kelas Radang, dan 62 data kelas Kanker.

Pada metode *k-fold cross validation* menggunakan 20% data uji dari keseluruhan data yang dipilih secara bergantian dengan nilai  $k=5$  sehingga diperlukan 5 kali pengujian dengan data latih dan data uji yang berbeda lalu hasil akurasi dari kelima pengujian tersebut dirata-rata. Pada setiap pengujian dilakukan 5 kali *running* pada program lalu hasilnya dirata-rata.

##### 4.5.2.1 Pengujian 11 Fitur

Pada pengujian 11 fitur digunakan fitur *B, C, N, O, P, S, F, Cl, Br, I, dan OH*. Hasil pengujian 11 fitur dapat dilihat pada Tabel 4.21. Sedangkan hasil pengujian 15 fitur dengan perlakuan pengujian yang sama dengan pengujian 11 fitur dapat dilihat pada Tabel 4.20.

**Tabel 4.20 Perancangan Pengujian 11 Fitur pada Kelas Metabolisme, Radang, dan Kanker**

<i>Cross Validation</i> Jumlah <i>Hidden Neuron</i>	1	2	3	4	5	Rata-Rata Akurasi
100						
200						
300						
400						

## BAB 5 IMPLEMENTASI

Pada bab ini membahas tentang implementasi sistem berdasarkan perancangan yang telah dilakukan menggunakan bahasa pemrograman Python versi 2.7. Bab ini terdiri dari implementasi sistem dan *screenshot* keluaran sistem.

### 5.1 Implementasi Sistem

Implementasi sistem terdapat 2 kelas yaitu kelas *prerprocessing* dan kelas ELM. Dalam metode ELM terdapat 2 tahapan yaitu tahapan untuk pemrosesan data latih dan data uji.

#### 5.1.1 Implementasi *Preprocessing*

Sebelum melakukan perhitungan ELM, sistem akan melakukan *preprocessing* pada data SMILES untuk mendapatkan nilai dari 15 fitur, yaitu *B, C, N, O, P, S, F, Cl, Br, I, OH, @, =, #, dan ()*.

##### 5.1.1.1 Mencari Nilai Dari 15 Fitur

Langkah pertama saat melakukan *preprocessing* adalah dengan mencari nilai dari 15 fitur yang ada pada notasi SMILES.

#### Kode Program 5.1 Mencari Nilai Dari 15 Fitur

Algoritme 1: Mencari Nilai Dari 15 Fitur	
1	<code>import re, numpy as np</code>
2	<code>def hitungBr(senyawa):</code>
3	<code>    Br=len(re.findall(r"[B][r]",senyawa))</code>
4	<code>    return Br</code>
5	<code>def hitungCl(senyawa):</code>
6	<code>    Cl=len(re.findall(r"[C][l]",senyawa))</code>
7	<code>    return Cl</code>
8	<code>def hitungB(senyawa,Br):</code>
9	<code>    B=len(re.findall(r"B",senyawa))-Br</code>
10	<code>    return B</code>
11	<code>def hitungC(senyawa,Cl):</code>
12	<code>    C=len(re.findall(r"C",senyawa))-Cl</code>
13	<code>    return C</code>
14	<code>def hitungN(senyawa):</code>
15	<code>    N=len(re.findall(r"N",senyawa))</code>
16	<code>    return N</code>
17	<code>def hitungOH(senyawa):</code>
18	<code>    OH1=len(re.findall(r"[^=][O]\$",senyawa))</code>
19	<code>    OH2=len(re.findall(r"[^=][O]()",senyawa))</code>
20	<code>    totOH=OH1+OH2</code>
21	<code>    return totOH</code>
22	<code>def hitungO(senyawa,totOH):</code>
23	<code>    O=len(re.findall(r"O",senyawa))-totOH</code>

```
24     return O
25 def hitungP(senyawa):
26     P=len(re.findall(r"P",senyawa))
27     return P
28 def hitungS(senyawa):
29     S=len(re.findall(r"S",senyawa))
30     return S
31 def hitungF(senyawa):
32     F=len(re.findall(r"F",senyawa))
33     return F
34 def hitungI(senyawa):
35     I=len(re.findall(r"I",senyawa))
36     return I
37 def hitungAd(senyawa):
38     ad=len(re.findall(r"@",senyawa))
39     return ad
40 def hitungRangkapDua(senyawa):
41     rangkapDua=len(re.findall(r"=",senyawa))
42     return rangkapDua
43 def hitungRangkapTiga(senyawa):
44     rangkapTiga=len(re.findall(r"#",senyawa))
45     return rangkapTiga
46 def hitungCabang(senyawa):
47     cab1=len(re.findall(r"([BCNOPSFIClBr=1234#@+-.]+)",senyawa))
48     cab2=len(re.findall(r"([C][l]+)",senyawa))
49     cab3=len(re.findall(r"([B][r]+)",senyawa))
50     totCab=cab1+cab2+cab3
51     return totCab
52 def getFeature(senyawa):
53     Br = hitungBr(senyawa)
54     Cl = hitungCl(senyawa)
55     B = hitungB(senyawa,Br)
56     C = hitungC(senyawa,Cl)
57     N = hitungN(senyawa)
58     OH = hitungOH(senyawa)
59     O = hitungO(senyawa,OH)
60     P = hitungP(senyawa)
61     S = hitungS(senyawa)
62     F = hitungF(senyawa)
63     I = hitungI(senyawa)
64     ad = hitungAd(senyawa)
65     rangkapDua = hitungRangkapDua(senyawa)
66     rangkapTiga = hitungRangkapTiga(senyawa)
67     cabang = hitungCabang(senyawa)
```

	fitur =
68	np.array([Br, Cl, B, C, N, OH, O, P, S, F, I, ad, rangkapDua, rangkapTiga, cabang], dtype='float')
69	return fitur
70	print fitur

Penjelasan:

- Baris 2 – 4 adalah metode yang digunakan untuk mencari nilai fitur Br pada notasi SMILES.
- Baris 5 – 7 adalah metode yang digunakan untuk mencari nilai fitur Cl pada notasi SMILES.
- Baris 8 – 10 adalah metode yang digunakan untuk mencari nilai fitur B pada notasi SMILES.
- Baris 11 – 13 adalah metode yang digunakan untuk mencari nilai fitur C pada notasi SMILES.
- Baris 14 – 16 adalah metode yang digunakan untuk mencari nilai fitur N pada notasi SMILES.
- Baris 17 – 21 adalah metode yang digunakan untuk mencari nilai fitur OH pada notasi SMILES.
- Baris 22 – 24 adalah metode yang digunakan untuk mencari nilai fitur O pada notasi SMILES.
- Baris 25 – 27 adalah metode yang digunakan untuk mencari nilai fitur P pada notasi SMILES.
- Baris 28 – 30 adalah metode yang digunakan untuk mencari nilai fitur S pada notasi SMILES.
- Baris 31 – 33 adalah metode yang digunakan untuk mencari nilai fitur F pada notasi SMILES.
- Baris 34 – 36 adalah metode yang digunakan untuk mencari nilai fitur I pada notasi SMILES.
- Baris 37 – 39 adalah metode yang digunakan untuk mencari nilai fitur @ pada notasi SMILES.
- Baris 40 – 42 adalah metode yang digunakan untuk mencari nilai fitur = pada notasi SMILES.
- Baris 43 – 45 adalah metode yang digunakan untuk mencari nilai fitur # pada notasi SMILES.
- Baris 46 – 51 adalah metode yang digunakan untuk mencari nilai fitur () pada notasi SMILES.
- Baris 52 – 70 adalah metode yang digunakan untuk menggabungkan nilai dari 15 fitur yang telah dihitung sebelumnya menggunakan fungsi array.



### 5.1.1.2 Menyimpan Nilai Fitur

Setelah mendapatkan nilai fitur pada notasi SMILES maka nilai tersebut perlu disimpan agar dapat dilakukan perhitungan klasifikasi menggunakan metode ELM.

#### Kode Program 5.2 Menyimpan Nilai Fitur

Algoritme 2: Menyimpan Nilai Fitur	
1	def proses():
	files = np.loadtxt('D:/SKRIPSI
	ISTI/Pengujian/Metabolisme-Radang-Kanker/Cross Validation 3
2	(Uji).csv', dtype='string')
3	fitur = []
4	for i in range(len(files)):
5	hasil = getFeature(files[i])
6	hasil = hasil/hasil.sum()
7	print hasil
8	fitur.append(hasil)
9	fitur = np.array(fitur)
	np.savetxt('D:/SKRIPSI ISTI/Pengujian/Metabolisme-
10	Radang-Kanker/FiturTest.csv', fitur, delimiter=';')
11	proses()

Penjelasan:

- Baris 2 digunakan untuk membaca file yang berisi data notasi SMILES, baik untuk data latih maupun data uji
- Baris 3 – 9 digunakan untuk mendapatkan nilai fitur pada notasi SMILES yang ada pada file. Pada baris 6 nilai setiap fitur dibagi oleh jumlah fitur pada notasi SMILES untuk mendapatkan nilai fitur.
- Baris 10 digunakan untuk menyimpan nilai fitur pada file yang nantinya akan digunakan untuk perhitungan ELM.

### 5.1.2 Implementasi ELM

Setelah dilakukan *preprocessing* dan didapatkan nilai fitur dari notasi SMILES, maka akan dilakukan perhitungan menggunakan ELM. Pada ELM terdapat 2 proses perhitungan yaitu *training* ELM dan *testing* ELM.

#### 5.1.2.1 Training ELM

Pada proses *training* ELM ada beberapa langkah yang dilakukan yaitu inialisasi bobot, menghitung matriks inialisasi keluaran *hidden layer* ( $H_{init}$ ), matriks keluaran *hidden layer* ( $H$ ), matriks  $H^\dagger$ , bobot keluaran ( $\hat{\beta}$ ), dan  $\hat{Y}$  prediksi.

#### Kode Program 5.3 Training ELM

Algoritme 3: Training ELM	
1	def InialisasiBobot(n_hidden, n_fitur):
2	bobot=np.zeros((n_hidden, n_fitur))

```

3     for i in range(n_hidden):
4         for j in range(n_fitur):
5             bobot[i,j] = random.uniform(-0.5,0.5)
6     return bobot
7 def TransposeMatriks(matriks_awal):
8     x,y = matriks_awal.shape
9     hasilTranspose=np.zeros((y,x))
10    for i in range(y):
11        for j in range(x):
12            hasilTranspose[i,j]=matriks_awal[j,i]
13    return hasilTranspose
14 def perkalianMatriks(m1,m2):
15    m=[]
16    for i in range(len(m1)):
17        r=[]
18        for j in range(len(m2[0])):
19            sums=0
20            for k in range(len(m2)):
21                sums=sums+(m1[i][k]*m2[k][j])
22            r.append(sums)
23        m.append(r)
24    return m
25
26 def hitungHinit(data, bobot):
27     Hinit = perkalianMatriks(data,bobot)
28     Hinit=np.array(Hinit)
29     return Hinit
30 def hitungH(Hinit):
31     H = 1/(1+np.exp(-Hinit))
32     return H
33 def hitungHdagger(H):
34     HTranspose=TransposeMatriks(H)
35     HTH=hitungHinit(HTranspose,H)
36     inversHTH=np.linalg.inv(HTH)
37     inversHTH
38     Hdagger=hitungHinit(inversHTH,HTranspose)
39     return Hdagger
40 def hitungBeta(Hdagger,Y):
41     x,y=Hdagger.shape
42     beta=np.zeros(x)
43     for i in range(x):
44         for j in range(y):
45             beta[i]+=Hdagger[i,j]*Y[j]
46     return beta

```

```

47 def hitungYtest(H,B):
48     x,y=H.shape
49     Ytest=np.zeros(x)
50     for i in range (x):
51         for j in range(y):
52             Ytest[i]+=H[i,j]*B[j]
53     return Ytest
54 def hasilpred(Y):
55     hasilkelas=[]
56     kelas = np.array([1,2,3])
57     for i in range(len(Y)):
58         hasil = np.abs(kelas-Y[i])
59         min_hsl = np.min(hasil)
60         hsl = (list(hasil).index(min_hsl))
61         hasilkelas.append(hsl+1)
62     return hasilkelas
63 def akurasi(predKelas, kelasTest):
64     total = 0
65     for i in range(len(predKelas)):
66         if round(predKelas[i])==kelasTest[i]:
67             total +=1
68     akurasi = float(total)/float(len(predKelas))*100
69     return akurasi
70 data = np.loadtxt('D:/SKRIPSI ISTI/Pengujian/Metabolisme-
71 Radang-Kanker/Fitur.csv',delimiter=';')
72 kelas =np.loadtxt('D:/SKRIPSI ISTI/Pengujian/Metabolisme-
73 Radang-Kanker/Kelas (Latih).csv')
74 hidden = input('masukkan hidden neuron : ')
75 bobot=InisialisasiBobot(int(hidden),len(data[0]))
76 def simpanBobot():
77     np.savetxt('D:/SKRIPSI
78 ISTI/bobotFix.csv',bobot,delimiter=';')
79 simpanBobot()
80 bobotTranspose = TransposeMatriks(bobot)
81 Hinit = hitungHinit(data,bobotTranspose)
82 H=hitungH(Hinit)
83 Hdagger=hitungHdagger(H)
84 B=hitungBeta(Hdagger, kelas)
85 Yprediksi=hitungYtest(H,B)

```

Penjelasan:

- Baris 1 – 6 adalah metode yang digunakan untuk inisialisasi bobot secara acak dengan rentang -0,5 sampai 0,5. Panjang matriks bobot adalah  $i \times j$  dengan  $i$  adalah banyak *hidden layer* dan  $j$  adalah banyak fitur.
- Baris 7 – 13 adalah metode yang digunakan untuk *transpose* matriks dengan mengubah baris menjadi kolom dan kolom menjadi baris.
- Baris 14 – 25 adalah metode yang digunakan untuk perkalian matriks.
- Baris 26 – 29 adalah metode yang digunakan untuk menghitung matriks inisialisasi keluaran *hidden layer* ( $\mathbf{H}_{init}$ ) dengan mengalikan matriks data dengan matriks bobot.
- Baris 30 – 32 adalah metode yang digunakan untuk menghitung matriks keluaran *hidden layer* ( $\mathbf{H}$ ) berdasarkan Persamaan 2.6.
- Baris 33 – 39 adalah metode yang digunakan untuk menghitung matriks  $\mathbf{H}^{\dagger}$ , dengan men-*transpose* nilai  $\mathbf{H}$ , melakukan perkalian matriks  $\mathbf{H}$  *transpose* dengan matriks  $\mathbf{H}$  lalu dilakukan *invers* matriks. Setelah itu dilakukan perhitungan matriks  $\mathbf{H}^{\dagger}$  dengan mengalikan hasil *invers* dengan  $\mathbf{H}$  *transpose*.
- Baris 40 – 46 adalah metode yang digunakan untuk menghitung bobot keluaran ( $\hat{\beta}$ ) dengan mengalikan matriks  $\mathbf{H}^{\dagger}$  dengan kelas sebenarnya ( $\mathbf{Y}$ ).
- Baris 47 – 53 adalah metode yang digunakan untuk menghitung  $\hat{\mathbf{Y}}$  prediksi dengan mengalikan matriks  $\mathbf{H}$  dengan bobot keluaran ( $\hat{\beta}$ ).
- Baris 54 – 62 adalah metode yang digunakan untuk menghitung hasil kelas prediksi dengan menentukan selisih terkecil antara  $\hat{\mathbf{Y}}$  prediksi dengan kelas sebenarnya ( $\mathbf{Y}$ ).
- Baris 63 – 69 adalah metode yang digunakan untuk menghitung akurasi sistem dengan rumus:
 
$$Akurasi = \frac{\text{jumlah data sama antara kelas prediksi dan kelas sebenarnya}}{\text{jumlah kelas prediksi}} \times 100\%$$
- Baris 70 digunakan untuk membaca file berisi nilai 15 fitur notasi SMILES pada data latih yang telah di-*preprocessing*.
- Baris 71 digunakan untuk membaca file berisi kelas sebenarnya pada data latih.
- Baris 72 digunakan sebagai masukan *user* untuk menentukan jumlah *hidden layer*.
- Baris 73 digunakan untuk inisialisasi variabel bobot dari perhitungan metode InisialisasiBobot().
- Baris 74 – 76 adalah metode yang digunakan untuk menyimpan nilai bobot ke dalam file untuk menyimpan bobot terbaik yang diperoleh secara *random*.
- Baris 77 digunakan untuk melakukan *transpose* pada matriks bobot dan menyimpannya ke dalam variabel bobotTranspose.
- Baris 78 digunakan untuk menghitung matriks  $\mathbf{H}_{init}$  dan menyimpannya ke dalam variabel Hinit.
- Baris 79 digunakan untuk menghitung matriks  $\mathbf{H}$  dan menyimpannya ke dalam variabel H.
- Baris 80 digunakan untuk menghitung matriks  $\mathbf{H}^{\dagger}$  dan menyimpannya ke dalam variabel Hdagger.



- Baris 81 digunakan untuk menghitung vektor  $\hat{\beta}$  dan menyimpannya ke dalam variabel B.
- Baris 82 digunakan untuk menghitung vektor  $\hat{Y}$  prediksi dan menyimpannya ke dalam variabel Yprediksi.

### 5.1.2.2 Testing ELM

Pada proses *testing* ELM ada beberapa langkah yang dilakukan yaitu menghitung matriks inialisasi keluaran *hidden layer* ( $H_{init}$ ), matriks keluaran *hidden layer* ( $H$ ), dan  $\hat{Y}$  prediksi.

#### Kode Program 5.4 Testing ELM

Algoritme 4: <i>Testing</i> ELM	
1	<code>dataTest= np.loadtxt('D:/SKRIPSI ISTI/Pengujian/Metabolisme-Radang-Kanker/FiturTest.csv',delimiter=';')</code>
2	<code>kelasTest =np.loadtxt('D:/SKRIPSI ISTI/Pengujian/Metabolisme-Radang-Kanker/Kelas (Uji).csv')</code>
3	<code>HinitTest = hitungHinit(dataTest,bobotTranspose)</code>
4	<code>HTest=hitungH(HinitTest)</code>
5	<code>YTest=hitungYtest(HTest,B)</code>
6	<code>predKelas = hasilpred(YTest)</code>
7	<code>print 'Kelas Prediksi : \n', predKelas</code>
8	<code>print 'Kelas Sebenarnya \n: ', kelasTest</code>
9	<code>print 'Akurasi : ', akurasi(predKelas, kelasTest)</code>

Penjelasan:

- Baris 1 digunakan untuk membaca file berisi nilai 15 fitur notasi SMILES pada data uji yang telah di-*preprocessing*.
- Baris 2 digunakan untuk membaca file berisi kelas sebenarnya pada data uji.
- Baris 3 digunakan untuk menghitung matriks  $H_{init}$  pada data uji dan menyimpannya ke dalam variabel HinitTest.
- Baris 4 digunakan untuk menghitung matriks  $H$  pada data uji dan menyimpannya ke dalam variabel HTest.
- Baris 5 digunakan untuk menghitung vektor  $\hat{Y}$  prediksi pada data uji dan menyimpannya ke dalam variabel YTest.
- Baris 6 digunakan untuk menghitung kelas prediksi dan menyimpannya ke dalam variabel predKelas.
- Baris 7 digunakan untuk mencetak kelas prediksi.
- Baris 8 digunakan untuk mencetak kelas sebenarnya.
- Baris 9 digunakan untuk mencetak hasil akurasi.



## BAB 6 PENGUJIAN DAN ANALISIS

Pada bab ini akan dijelaskan tentang hasil pengujian dan analisis terhadap hasil pengujian yang telah dilakukan oleh sistem serta evaluasi kinerja sistem. Pengujian dilakukan berdasarkan perancangan pengujian dengan dilakukan 2 pengujian, yaitu pengujian jumlah *hidden neuron* dengan metode *k-fold cross validation* dan pengujian jumlah fitur dengan metode *k-fold cross validation*.

### 6.1 Pengujian Jumlah *Hidden Neuron* dengan Metode *K-fold Cross Validation*

Metode *k-fold cross validation* digunakan untuk mengevaluasi kinerja ELM pada data SMILES. Pengujian jumlah *hidden neuron* dengan metode *k-fold cross validation* adalah pengujian jumlah *hidden neuron* menggunakan 20% data uji dari keseluruhan data yang dipilih secara bergantian dengan nilai  $k=5$  sehingga diperlukan 5 kali pengujian dengan data latih dan data uji yang berbeda lalu hasil akurasi dari kelima pengujian tersebut dirata-rata. Pada setiap pengujian dilakukan 5 kali *running* pada program lalu hasilnya dirata-rata.

Pengujian jumlah *hidden neuron* dengan metode *k-fold cross validation* dilakukan pada 2 skenario berbeda, yaitu pengujian jumlah *hidden neuron* untuk 2 kelas dan pengujian jumlah *hidden neuron* untuk 3 kelas. Pengujian jumlah *hidden neuron* untuk 2 kelas terdiri dari 3 skenario yang berbeda, yaitu kelas Metabolisme dengan Kanker, Radang dengan Kanker, dan Metabolisme dengan Radang. Pengujian jumlah *hidden neuron* untuk 3 kelas terdiri dari 1 skenario yaitu kelas Metabolisme, Radang, dan Kanker.

#### 6.1.1 Pengujian 2 Kelas

Pengujian 2 kelas terdiri dari 3 skenario yang berbeda, yaitu kelas Metabolisme dengan Kanker, Radang dengan Kanker, dan Metabolisme dengan Radang. Pengujian jumlah *hidden neuron* dilakukan sebanyak 8 kali, yaitu dengan jumlah *hidden neuron* 5, 10, 15, 20, 25, 30, 35, dan 40. Pengujian *k-fold cross validation* menggunakan 20% data uji dari keseluruhan data yang dipilih secara bergantian dengan nilai  $k=5$  sehingga diperlukan 5 kali pengujian dengan data latih dan data uji yang berbeda lalu hasil akurasi dari kelima pengujian tersebut dirata-rata.

##### 6.1.1.1 Kelas Metabolisme dan Kanker

Pada kelas Metabolisme dan Kanker data latih yang digunakan sebanyak 748 dengan 501 data kelas Metabolisme dan 247 data kelas Kanker. Data uji yang digunakan sebanyak 187 dengan 125 data kelas Metabolisme dan 62 data kelas Kanker. Hasil pengujian pada kelas Metabolisme dan Kanker dapat dilihat pada Tabel 6.1.

**Tabel 6.1 Perancangan Pengujian Jumlah *Hidden Neuron* Kelas Metabolisme dan Kanker**

<i>Cross Validation</i> Jumlah <i>Hidden Neuron</i>	1	2	3	4	5	Rata-Rata Akurasi
5	67,70	65,56	67,27	66,63	67,85	67,00
10	66,85	64,92	67,49	64,49	68,71	66,49
15	68,13	63,21	67,81	65,99	67,96	66,62
20	70,80	64,81	69,52	66,85	67,74	67,94
25	69,31	65,95	69,20	68,02	65,59	67,61
30	69,95	65,35	67,70	66,20	65,48	66,94
35	69,20	66,42	68,45	65,99	65,70	67,15
40	66,85	65,13	67,59	67,91	65,91	66,68

Berdasarkan Tabel 6.1 pengujian jumlah *hidden neuron* dilakukan sebanyak 8 kali dengan jumlah *hidden neuron* yang berbeda pada setiap pengujian *k-fold cross validation*. Pengujian *k-fold cross validation* menggunakan 20% data uji dari keseluruhan data yang dipilih secara bergantian dengan nilai  $k=5$  sehingga diperlukan 5 kali pengujian dengan data latih dan data uji yang berbeda lalu hasil akurasi dari kelima pengujian tersebut dirata-rata. Rata-rata akurasi terbaik sebesar 67,94% dengan jumlah 20 *hidden neuron*.

#### 6.1.1.2 Kelas Radang dan Kanker

Pada kelas Radang dan Kanker data latih yang digunakan sebanyak 339 dengan 92 data kelas Radang dan 247 data kelas Kanker. Data uji yang digunakan sebanyak 84 dengan 22 data kelas Radang dan 62 data kelas Kanker. Hasil pengujian pada kelas Radang dan Kanker dapat dilihat pada Tabel 6.2.

**Tabel 6.2 Perancangan Pengujian Jumlah *Hidden Neuron* Kelas Radang dan Kanker**

<i>Cross Validation</i> Jumlah <i>Hidden Neuron</i>	1	2	3	4	5	Rata-Rata Akurasi
5	73,81	73,81	73,33	73,81	73,01	73,55
10	72,62	73,81	73,81	73,81	73,01	73,41
15	72,62	72,86	73,33	73,57	70,12	72,50
20	71,67	72,38	73,10	74,29	69,40	72,17
25	72,38	73,10	73,57	73,57	69,40	72,40
30	74,29	75,24	73,81	74,05	66,75	72,83
35	73,81	74,76	73,81	70,24	65,79	71,68
40	76,90	75,95	74,76	73,09	65,30	73,20

Berdasarkan Tabel 6.2 pengujian jumlah *hidden neuron* dilakukan sebanyak 8 kali dengan jumlah *hidden neuron* yang berbeda pada setiap pengujian *k-fold cross validation*. Rata-rata akurasi terbaik sebesar 73,55% dengan jumlah 5 *hidden neuron*.

### 6.1.1.3 Kelas Metabolisme dan Radang

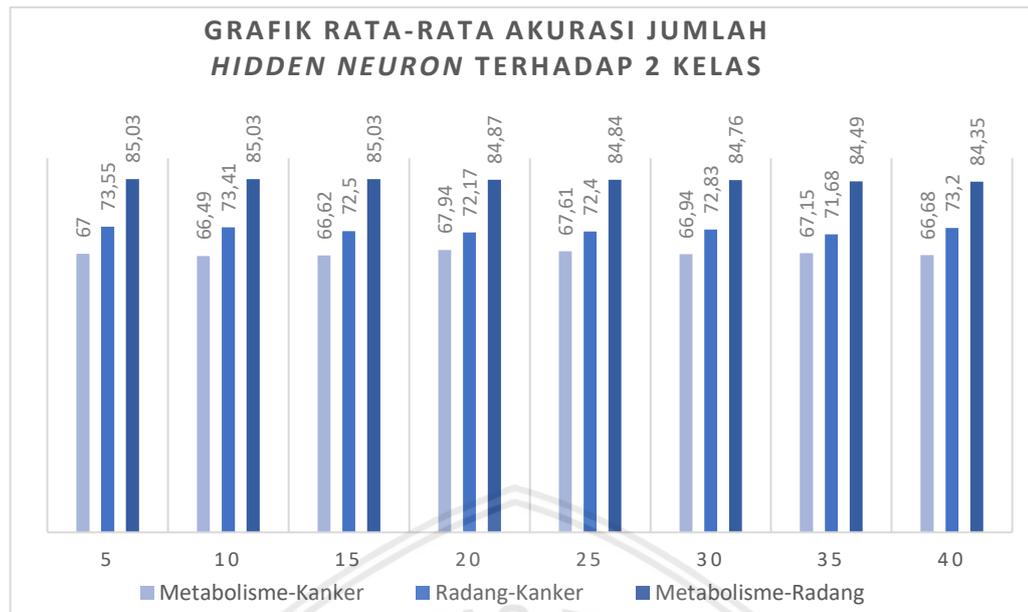
Pada kelas Metabolisme dan Radang data latih yang digunakan sebanyak 593 dengan 501 data kelas Metabolisme dan 92 data kelas Radang. Data uji yang digunakan sebanyak 147 dengan 125 data kelas Metabolisme dan 22 data kelas Radang. Hasil pengujian pada kelas Metabolisme dan Radang dapat dilihat pada Tabel 6.3.

Berdasarkan Tabel 6.3 pengujian jumlah *hidden neuron* dilakukan sebanyak 8 kali dengan jumlah *hidden neuron* yang berbeda pada setiap pengujian *k-fold cross validation*. Rata-rata akurasi terbaik sebesar 85,03% dengan jumlah 5, 10, dan 15 *hidden neuron*.

**Tabel 6.3 Perancangan Pengujian Jumlah *Hidden Neuron* Kelas Metabolisme dan Radang**

<b>Cross Validation</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>Rata-Rata Akurasi</b>
<b>Jumlah Hidden Neuron</b>						
<b>5</b>	85,03	85,03	85,03	85,03	85,03	85,03
<b>10</b>	85,03	85,03	85,03	85,03	85,03	85,03
<b>15</b>	85,03	85,03	85,03	85,03	85,03	85,03
<b>20</b>	84,89	84,49	85,03	85,03	84,89	84,87
<b>25</b>	84,49	84,62	85,03	85,17	84,89	84,84
<b>30</b>	84,49	84,49	84,62	85,17	85,03	84,76
<b>35</b>	84,35	83,81	84,35	85,17	84,76	84,49
<b>40</b>	84,76	83,94	83,40	85,03	84,62	84,35

Dari Tabel 6.1, Tabel 6.2, dan Table 6.3 diperoleh rata-rata akurasi jumlah *hidden neuron* terhadap skenario kelas Metabolisme dan Radang, Radang dan Kanker, serta Metabolisme dan Radang. Untuk perbandingan nilai rata-rata akurasi dari ketiga skenario tersebut dapat dilihat pada Gambar 6.1. Berdasarkan grafik pada Gambar 6.1 diperoleh akurasi tertinggi sebesar 85,03% pada skenario kelas Metabolisme dan Radang dengan jumlah 5, 10, dan 15 *hidden neuron*.



**Gambar 6.1** Grafik Rata-Rata Akurasi Jumlah *Hidden Neuron* Terhadap 2 Kelas

Berdasarkan hasil pengujian jumlah *hidden neuron* dengan metode *k-fold cross validation* pada 2 kelas, ELM dapat melakukan klasifikasi dengan akurasi terbaik sebesar 85,03% pada skenario kelas Metabolisme dan Radang dengan jumlah 5, 10, dan 15 *hidden neuron*. Kelas Metabolisme dan Radang mempunyai akurasi yang lebih tinggi daripada kelas Metabolisme dan Kanker ataupun kelas Radang dan Kanker karena dataset pada kelas Kanker lebih unik daripada dataset pada kelas Radang, sehingga ELM lebih mudah mengenali data pada kelas Radang daripada kelas Kanker.

### 6.1.2 Pengujian 3 Kelas

Pengujian 3 kelas terdiri dari 1 skenario yaitu kelas Metabolisme, Radang, dan Kanker. Pengujian jumlah *hidden neuron* dilakukan sebanyak 4 kali, yaitu dengan jumlah *hidden neuron* 100, 200, 300, dan 400. Pengujian *k-fold cross validation* menggunakan 20% data uji dari keseluruhan data yang dipilih secara bergantian dengan nilai  $k=5$  sehingga diperlukan 5 kali pengujian dengan data latih dan data uji yang berbeda lalu hasil akurasi dari kelima pengujian tersebut dirata-rata.

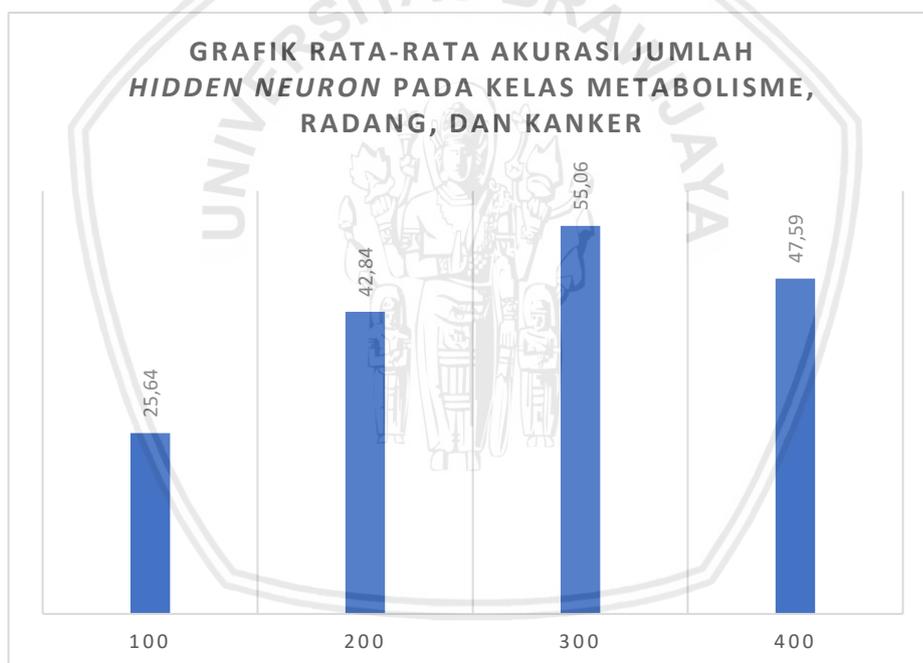
#### 6.1.2.1 Kelas Metabolisme, Radang, dan Kanker

Pada kelas Metabolisme, Radang, dan Kanker data latih yang digunakan sebanyak 840 dengan 501 data kelas Metabolisme, 92 data kelas Radang, dan 247 data kelas Kanker. Data uji yang digunakan sebanyak 209 dengan 125 data kelas Metabolisme, 22 data kelas Radang, dan 62 data kelas Kanker. Hasil pengujian pada kelas Metabolisme, Radang, dan Kanker dapat dilihat pada Tabel 6.4.

Tabel 6.4 Perancangan Pengujian Jumlah *Hidden Neuron* Kelas Metabolisme, Radang, dan Kanker

<i>Cross Validation</i> Jumlah <i>Hidden Neuron</i>	1	2	3	4	5	Rata-Rata Akurasi
100	25,74	24,50	22,87	22,87	32,21	25,64
200	37,99	50,43	44,98	34,93	45,87	42,84
300	53,50	55,98	59,52	51,10	55,19	55,06
400	53,88	44,50	50,81	41,05	47,70	47,59

Berdasarkan Tabel 6.4 pengujian jumlah *hidden neuron* dilakukan sebanyak 4 kali dengan jumlah *hidden neuron* yang berbeda pada setiap pengujian *k-fold cross validation*. Rata-rata akurasi terbaik sebesar 55,06% dengan jumlah 300 *hidden neuron*. Perbandingan rata-rata akurasi dari dari Tabel 6.4 dapat dilihat pada Gambar 6.2.



Gambar 6.2 Grafik Rata-Rata Akurasi Jumlah *Hidden Neuron* pada Kelas Metabolisme, Radang, dan Kanker

Berdasarkan hasil pengujian jumlah *hidden neuron* dengan metode *k-fold cross validation* pada 3 kelas, ELM dapat melakukan klasifikasi dengan akurasi terbaik sebesar 55,06% pada skenario kelas Metabolisme, Radang, dan Kanker dengan 300 *hidden neuron*.

Pada pengujian ini, ELM dapat bekerja lebih maksimal pada 2 kelas daripada 3 kelas. Salah satu faktor kurang maksimalnya akurasi pada 3 kelas adalah fitur pada pengujian perlu ditambah agar ELM lebih mudah membedakan data antar kelas apabila kelas yang digunakan banyak.

## 6.2 Pengujian Jumlah Fitur dengan Metode *K-fold Cross Validation*

Pengujian jumlah fitur digunakan untuk menunjukkan hasil akurasi pada jumlah fitur yang berbeda. Pada pengujian jumlah fitur digunakan 2 kondisi yaitu ketika fitur berjumlah 11 dan 15. Jumlah kelas yang digunakan sebanyak 3, yaitu Metabolisme, Radang, dan Kanker. Pada kelas Metabolisme, Radang, dan Kanker data latih yang digunakan sebanyak 840 dengan 501 data kelas Metabolisme, 92 data kelas Radang, dan 247 data kelas Kanker. Data uji yang digunakan sebanyak 209 dengan 125 data kelas Metabolisme, 22 data kelas Radang, dan 62 data kelas Kanker.

Pada metode *k-fold cross validation* menggunakan 20% data uji dari keseluruhan data yang dipilih secara bergantian dengan nilai  $k=5$  sehingga diperlukan 5 kali pengujian dengan data latih dan data uji yang berbeda lalu hasil akurasi dari kelima pengujian tersebut dirata-rata. Pada setiap pengujian dilakukan 5 kali *running* pada program lalu hasilnya dirata-rata.

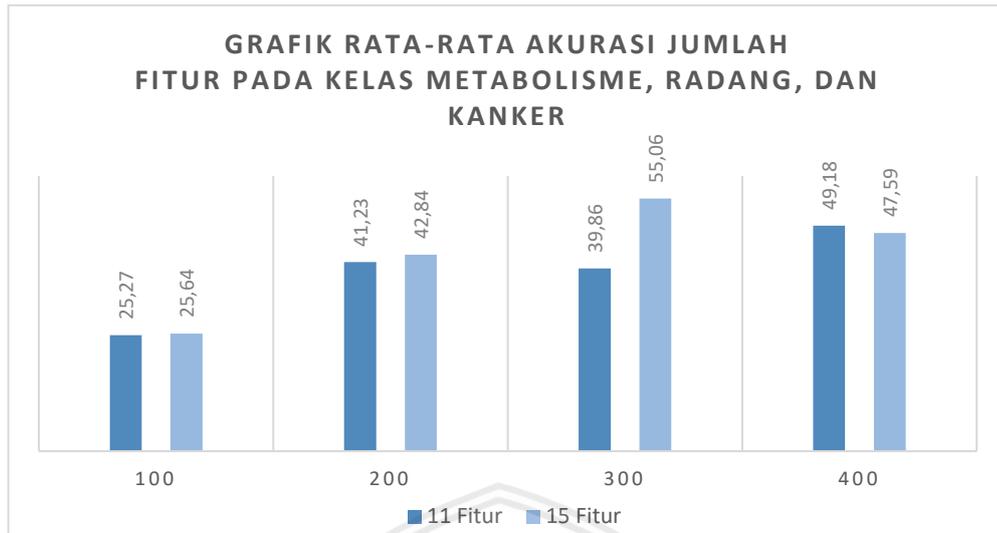
### 6.2.1 Pengujian 11 Fitur

Pada pengujian 11 fitur digunakan fitur *B, C, N, O, P, S, F, Cl, Br, I, dan OH*. Hasil pengujian 11 fitur dapat dilihat pada Tabel 6.5.

**Tabel 6.5 Perancangan Pengujian 11 Fitur pada Kelas Metabolisme, Radang, dan Kanker**

<i>Cross Validation</i> Jumlah <i>Hidden Neuron</i>	1	2	3	4	5	Rata-Rata Akurasi
100	22,87	25,36	23,73	27,56	26,83	25,27
200	41,44	47,36	29,76	40,86	46,73	41,23
300	41,34	37,23	37,70	28,42	54,62	39,86
400	41,05	58,09	46,12	47,37	53,27	49,18

Tabel 6.5 menunjukkan hasil akurasi terbaik pada pengujian 11 fitur adalah pada jumlah *hidden neuron* 400 dengan akurasi sebesar 49,18%. Sedangkan hasil pengujian 15 fitur dengan perlakuan pengujian yang sama dengan pengujian 11 fitur dapat dilihat pada Tabel 6.4. Grafik perbandingan rata-rata akurasi antara Tabel 6.4 dan Tabel 6.5 dapat dilihat pada Gambar 6.3.



**Gambar 6.3 Grafik Rata-Rata Akurasi Jumlah Fitur pada Kelas Metabolisme, Radang, dan Kanker**

Grafik pada Gambar 6.3 menunjukkan bahwa hasil akurasi tertinggi sebesar 55,06% pada pengujian 15 fitur dengan jumlah 300 *hidden neuron*. Hasil analisis pada pengujian ini menunjukkan bahwa semakin banyak fitur yang digunakan maka akurasi yang dihasilkan sistem juga akan semakin baik. Hal ini ditunjukkan bahwa ELM dapat menghasilkan akurasi terbaik pada pengujian 15 fitur sebesar 55,06% dengan jumlah 300 *hidden neuron* sedangkan pada pengujian 11 fitur diperoleh akurasi terbaik sebesar 49,18% dengan jumlah 400 *hidden neuron*.

### 6.3 Evaluasi Kinerja Sistem

Evaluasi kinerja terdiri dari perhitungan akurasi, *precision*, dan *recall*. Hasil pengujian yang dievaluasi adalah pada pengujian 3 kelas yaitu Metabolisme, Radang, dan Kanker dengan 15 fitur dan 300 *hidden neuron*. Perhitungan evaluasi kinerja akan diterapkan pada masing-masing kelas, dan karena menggunakan multipel kelas maka hasil evaluasi dari ketiga kelas akan dirata-rata.

#### 6.3.1 Kelas Metabolisme

Nilai *confusion matrix* yang terdiri dari *True Positives* (TP), *True Negatives* (TN), *False Positives* (FP), dan *False Negatives* (FN) pada kelas Metabolisme dapat dilihat pada Tabel 6.6.

**Tabel 6.6 Confusion Matrix Kelas Metabolisme**

<i>Actual Class</i> \ <i>Predicted Class</i>	<i>Yes</i>	<i>No</i>
	<i>Yes</i>	123
<i>No</i>	82	2

### 1. Akurasi

Untuk menghitung nilai akurasi digunakan Persamaan 2.11.

$$\text{Akurasi} = \frac{TP+TN}{TP+TN+FP+FN} \times 100\%$$

$$\text{Akurasi} = \frac{123 + 2}{123 + 2 + 82 + 2} \times 100\%$$

$$\text{Akurasi} = \frac{125}{209} \times 100\%$$

$$\text{Akurasi} = 59,8\%$$

### 2. Precision

Untuk menghitung nilai *precision* digunakan Persamaan 2.12.

$$\text{Precision} = \frac{TP}{TP+FP} \times 100\%$$

$$\text{Precision} = \frac{123}{123 + 82} \times 100\%$$

$$\text{Precision} = \frac{123}{205} \times 100\%$$

$$\text{Precision} = 60\%$$

### 3. Recall

Untuk menghitung nilai *recall* digunakan Persamaan 2.13.

$$\text{Recall} = \frac{TP}{TP+FN} \times 100\%$$

$$\text{Recall} = \frac{123}{123 + 2} \times 100\%$$

$$\text{Recall} = \frac{123}{125} \times 100\%$$

$$\text{Recall} = 98,4\%$$

## 6.3.2 Kelas Radang

Nilai *confusion matrix* yang terdiri dari *True Positives* (TP), *True Negatives* (TN), *False Positives* (FP), dan *False Negatives* (FN) pada kelas Radang dapat dilihat pada Tabel 6.7.

Tabel 6.7 *Confusion Matrix* Kelas Radang

<i>Actual Class</i> \ <i>Predicted Class</i>	Yes	No
	Yes	No
Yes	1	21
No	63	124

### 1. Akurasi

Untuk menghitung nilai akurasi digunakan Persamaan 2.11.

$$Akurasi = \frac{TP+TN}{TP+TN+FP+FN} \times 100\%$$

$$Akurasi = \frac{1 + 124}{1 + 124 + 63 + 21} \times 100\%$$

$$Akurasi = 59,8\%$$

### 2. Precision

Untuk menghitung nilai *precision* digunakan Persamaan 2.12.

$$Precision = \frac{TP}{TP+FP} \times 100\%$$

$$Precision = \frac{1}{1 + 63} \times 100\%$$

$$Precision = 1,56\%$$

### 3. Recall

Untuk menghitung nilai *recall* digunakan Persamaan 2.13.

$$Recall = \frac{TP}{TP+FN} \times 100\%$$

$$Recall = \frac{1}{1 + 21} \times 100\%$$

$$Recall = 4,55\%$$

## 6.3.3 Kelas Kanker

Nilai *confusion matrix* yang terdiri dari *True Positives* (TP), *True Negatives* (TN), *False Positives* (FP), dan *False Negatives* (FN) pada kelas Kanker dapat dilihat pada Tabel 6.8.

Tabel 6.8 *Confusion Matrix* Kelas Kanker

<i>Actual Class</i> \ <i>Predicted Class</i>	Yes	No
	Yes	No
Yes	1	61
No	23	124

### 1. Akurasi

Untuk menghitung nilai akurasi digunakan Persamaan 2.11.

$$Akurasi = \frac{TP+TN}{TP+TN+FP+FN} \times 100\%$$

$$Akurasi = \frac{1 + 124}{1 + 124 + 23 + 61} \times 100\%$$

$$Akurasi = 59,8\%$$

## 2. Precision

Untuk menghitung nilai *precision* digunakan Persamaan 2.12.

$$Precision = \frac{TP}{TP+FP} \times 100\%$$

$$Precision = \frac{1}{1+23} \times 100\%$$

$$Precision = 4,17\%$$

## 3. Recall

Untuk menghitung nilai *recall* digunakan Persamaan 2.13.

$$Recall = \frac{TP}{TP+FN} \times 100\%$$

$$Recall = \frac{1}{1+61} \times 100\%$$

$$Recall = 1,61\%$$

### 6.3.4 Hasil Evaluasi Kinerja Klasifikasi Multipel Kelas

Untuk mendapatkan nilai hasil evaluasi kinerja klasifikasi multipel kelas maka hasil perhitungan akurasi, *precision*, dan *recall* masing-masing kelas Metabolisme, Radang, dan Kanker dirata-rata.

$$Akurasi = \frac{59,8\% + 59,8\% + 59,8\%}{3} = 59,8\%$$

$$Precision = \frac{60\% + 1,56\% + 4,17\%}{3} = 21,91\%$$

$$Recall = \frac{98,4\% + 4,55\% + 1,61\%}{3} = 34,85\%$$

Berdasarkan hasil perhitungan evaluasi kinerja sistem pada 3 kelas, maka diperoleh nilai akurasi sebesar 59,8%, hal ini menunjukkan bahwa tingkat akurasi sistem cukup baik. Nilai *precision* sebesar 21,91%, hal ini menunjukkan bahwa kemampuan sistem untuk tidak memanggil kembali data yang tidak relevan atau tidak sesuai dengan apa yang diinginkan masih kurang baik. Nilai *recall* sebesar 34,85% hal ini menunjukkan kemampuan sistem untuk memanggil kembali data yang dianggap relevan atau sesuai dengan apa yang diinginkan masih kurang baik.

Hasil evaluasi kinerja pada perhitungan *precision* dan *recall* masih kurang baik karena hasil klasifikasi sistem cenderung menghasilkan kelas 1 atau kelas Metabolisme yang memiliki jumlah data latih lebih banyak daripada kelas Radang dan kelas Kanker. Hal ini dibuktikan dari hasil perhitungan *precision* dan *recall* pada kelas Metabolisme lebih besar yaitu 60% dan 98,4 % dibandingkan hasil perhitungan *precision* dan *recall* pada kelas Radang dan Kanker.

## BAB 7 PENUTUP

Pada bab ini berisi kesimpulan dan saran terhadap penelitian yang telah dilakukan.

### 7.1 Kesimpulan

Berdasarkan pengujian dan analisis yang telah dilakukan, diperoleh kesimpulan sebagai berikut:

1. Berdasarkan hasil pengujian jumlah *hidden neuron* dengan metode *k-fold cross validation* pada 2 kelas, ELM dapat melakukan klasifikasi dengan akurasi terbaik sebesar 85,03% pada skenario kelas Metabolisme dan Radang dengan jumlah 5, 10, dan 15 *hidden neuron*. Dan berdasarkan hasil pengujian jumlah *hidden neuron* dengan metode *k-fold cross validation* pada 3 kelas, ELM dapat melakukan klasifikasi dengan akurasi terbaik sebesar 55,06% pada skenario kelas Metabolisme, Radang, dan Kanker dengan jumlah 300 *hidden neuron*.
2. ELM dapat bekerja lebih maksimal pada 2 kelas daripada 3 kelas, hal ini dikarenakan semakin banyak kelas maka ELM harus lebih banyak belajar untuk membedakan data antar kelas, sehingga diperlukan lebih banyak fitur sebagai pembeda data antar kelas agar data dapat dikenali dengan maksimal. Hal ini ditunjukkan bahwa ELM dapat menghasilkan akurasi terbaik pada pengujian 15 fitur sebesar 55,06% dengan jumlah 300 *hidden neuron* sedangkan pada pengujian 11 fitur diperoleh akurasi terbaik sebesar 49,18% dengan jumlah 400 *hidden neuron*.

### 7.2 Saran

Saran dari penelitian ini digunakan untuk input penelitian selanjutnya agar lebih baik lagi.

1. Jumlah data yang digunakan perlu ditambah agar ELM dapat mengenali pola data lebih baik.
2. Fitur yang digunakan perlu ditambah agar data pada setiap kelas dapat dikenali dengan maksimal oleh ELM sehingga dapat menghasilkan akurasi lebih baik lagi.

## DAFTAR PUSTAKA

- Agustina, I.D., Anggraeni, W., dan Mukhlason A., 2010 *Penerapan Metode Extreme Learning Machine untuk Peramalan Permintaan*, [online] Tersedia di: <<http://digilib.its.ac.id/public/ITS-Undergraduate-9832-Paper.pdf>> [Diakses 20 Agustus 2018]
- Brown, Theodore L., LeMay, H E., Bursten, Bruce E., Murphy, Catherine J., Woodward, dan Patrick M., 2009. *Chemistry: The Central Science*. 11th Edition. Upper Saddle River, NJ: Prentice Hall.
- Fadilla I., Adikara P. P., dan Perdana R. S., 2018. *Klasifikasi Penyakit Chronic Kidney Disease (CKD) dengan menggunakan Metode Extreme Learning machine (ELM)*, [e-journal] 2(10). Tersedia melalui: <<http://j-ptiik.ub.ac.id/index.php/j-ptiik>> [Diakses 20 Oktober 2018]
- Fausett, L., 1994. *Fundamentals of Neural Networks (Architectures, Algorithms, and Applications)*. New Jersey : Prentice Hall.
- Han, J., Kamber M., dan Pei, J., 2012. *Data Mining Concepts and Techniques. Third Edition*. United States of America: Elsevier.
- Haykin, S., 2009. *Neural Networks and Learning Machines. Third Edition*. Upper Saddle River, New Jersey: Pearson Education, Inc.
- Huang, G., Zhou, H., Ding X., dan Zhang R., 2012. *Extreme Learning Machine for Regression and Multiclass Classification*, [online] <<https://ieeexplore.ieee.org/document/6035797> > [Diakses 25 Februari 2019]
- Huang, G., Zhu, Q. dan Siew, C., 2006. *Extreme learning machine: Theory and applications*, [online] <<http://axon.cs.byu.edu/~martinez/classes/678/Presentations/Yao.pdf>> [Diakses 20 Oktober 2018]
- Kristanti, A. N., Aminah, N. S., Tanjung, M., dan Kurniadi, B., 2008. *Buku Ajar Fitokimia*. Surabaya: Airlangga University Press.
- Melissa, dan Muchtaridi M., 2017. *Senyawa Aktif dan Manfaat Farmakologis Ageratum Conyzoides*, [online] Tersedia di: <<http://jurnal.unpad.ac.id/farmaka/article/view/13302/pdf>> [Diakses 4 Desember 2018]
- Rochman, E. M. S., dan Rachmad, A., 2017. *Kecerdasan Komputasional Konsep Dan Aplikasi*. Malang: Media Nusa Creative.
- Siang, J. J., 2009. *Jaringan Syaraf Tiruan Dan Pemrogramannya Menggunakan Matlab*. Yogyakarta: Andi.
- Sun, Z. L., Choi, T. M., Au, K. F. dan Yu, Y., 2008. *Sales Forecasting using Extreme Learning Machine with Application in Fashion Retailing*, [e-journal]. Tersedia melalui <<https://www.sciencedirect.com/>> [Diakses 20 Oktober 2018]
- Thabtah F., 2007. *A Review Of Associative Classification Mining*, [online] Tersedia di: <<http://eprints.hud.ac.uk/id/eprint/269/1/ThabtahReview.pdf>> [Diakses 20 Agustus 2018]

- Tigusti R. R. W., Ratnawati D. E., dan Anam, S., 2018. *Fungsi Senyawa Berdasarkan Simplified Molecular Input Line Entry System (SMILES)*, [e-journal] 3(1). Tersedia melalui: <<http://j-ptiik.ub.ac.id/index.php/j-ptiik>> [Diakses 20 Oktober 2018]
- Weininger, D., 1988. *SMILES, a Chemical Language and Information System. Introduction to Methodology and Encoding Rules*. [online] <<http://organica1.org/seminario/smile1988.pdf>> [Diakses 20 Oktober 2018]
- Witanto S., Ratnawati D. E., dan Anam, S., 2018. *Pengelompokan Fungsi Aktif Senyawa Data SMILES (Simplified Molecular Input Line Entry System) Menggunakan Metode K-Means Dengan Inisialisasi Pusat Kluster Menggunakan Metode Heuristic  $O(N \log N)$* , [e-journal] 3(1). Tersedia melalui: <<http://j-ptiik.ub.ac.id/index.php/j-ptiik>> [Diakses 20 Oktober 2018]

