IMPLEMENTASI *FUZZY K-NEAREST NEIGHBOR* (FK-NN) UNTUK MENGKLASIFIKASI FUNGSI SENYAWA BERDASARKAN *SIMPLIFIED MOLECULAR INPUT LINE ENTRY SYSTEM (SMILES)*

SKRIPSI

Untuk memenuhi sebagian persyaratan
Memperoleh gelar Sarjana Komputer

Disusun oleh:
Raden Rizky Widdie Tigusti
NIM: 145150201111134

PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2018

PENGESAHAN

IMPLEMENTASI FUZZY K-NEAREST NEIGHBOR (FK-NN) UNTUK MENGKLASIFIKASI FUNGSI SENYAWA BERDASARKAN SIMPLIFIED MOLECULAR INPUT LINE ENTRY SYSTEM (SMILES)

SKRIPSI

Untuk memenuhi sebagian persyaratan
Memperoleh gelar Sarjana Komputer

Disusun oleh:
Raden Rizky Widdie Tigusti
NIM: 145150201111134

Skripsi ini telah diuji dan dinyatakan lulus pada
31 Juli 2018
Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I	Dosen Pembimbing II
Dian Eka Ratnawati, S.Si, M.Kom	Syaiful Anam, S.Si.,MT.,Ph.D
NIP. 19730619 200212 2 001	NIP. 19780115 200212 1 003

Mengetahui
Ketua Jurusan Teknik Informatika

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

PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, didalam 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 pustaka.

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 sesusai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70)

Malang, 16 Juli 2018

Raden Rizky Widdie Tigusti

NIM: 145150201111134

KATA PENGANTAR

Puji syukur kehadirat Allah SWT yang selalu melimpahkan rahmat dan karunia-Nya kepada penulis sehingga penulis dapat menyelesaikan skripsi dengan judul "Implementasi *fuzzy k-nearest neighbor* (Fk-Nn) Untuk Mengklasifikasi Fungsi Senyawa Berdasarkan *Simplified Molecular Input Line Entry System* (SMILES)" sebagai syarat dalam memperoleh gelar sarjana pada Program Studi Informatika, Fakultas Ilmu Komputer, Universitas Brawijaya. Dalam proses penyusunan skripsi penulis mendapatkan bantuan berupa moral maupun materil dari banyak pihak. Maka dari itu, penulis ingin mengucapkan rasa terima kasih sebanyak-banyaknya kepada:

1. Ibu Dian Eka Ratnawati , S.Si, M.Kom selaku dosen pembimbing I yang telah memberikan arahan, masukan ilmu dan bimbingan sehingga penulis dapat menyelesaikan skripsi ini.
2. Bapak Syaiful Anam, S.Si.,MT.,Ph.D selaku dosen pembimbing II yang juga turut memberikan arahan, masukan ilmu dan bimbingan sehingga penulis dapat menyelesaikan skripsi ini.
3. Bapak dan Ibu dosen Fakultas Ilmu Komputer Universitas Brawijaya yang telah memberikan ilmu selama penulis melaksanakan kegiatan perkuliahan di kampus tercinta ini.
4. Staff dan karyawan Fakultas Ilmu Komputer yang telah membantu proses selama perkuliahan dan penulisan skripsi ini.
5. Ayahanda tercinta R. Iman Rumawat yang selalu memberikan segenap kasih sayang, tenaga, do'a dan dukungan yang tak terhingga kepada penulis dalam menyelesaikan skripsi ini.
6. Ibunda tercinta Umiyanti yang selalu memberikan segenap kasih sayang, tenaga, do'a dan dukungan yang tak terhingga kepada penulis dalam menyelesaikan skripsi ini.
7. Kakak dan adik Tercinta, Wina Nurdiani, Abang Irfan, dan R. Sekar Ayulindra Tigusti yang selalu memberikan semangat, doa dan dukungan kepada penulis dalam menyelesaikan skripsi ini.
8. Keluarga besar di Cirebon dan Bandung yang selalu mendoakan dan memberikan semangat dalam menyelesaikan skripsi ini.
9. Teman-teman seperjuangan dalam menyusun skripsi Suhhy Ramzini, Muhammad Iskandar A.R, Nur Khilmiyatul, Sherly Witanto, Nyimas Ayu Widi Indriana, Yunita Dwi Alfianti yang telah saling membantu dan berdiskusi dalam penyelesaian skripsi ini.
10. Dhika Rozqi Anggitama, Kevin Dwiki Saputra, Hermawan Wijaya, David Christanto, Yosua Tito Sumbogo, Sherly Witanto, Maria Rantikasari, dan Reka Suryani Sidauruk yang telah memberikan pengalaman, diskusi dan motivasi selama penullis melaksanakan perkuliahan di Fakultas Ilmu Komputer Universitas Brawijaya.

11. Teman-teman Informatika FILKOM UB 2014 khususnya Dwi Qunita Putri, Mahdarani Dwi Laxmi, Risailin, Sofi Hidyah Anggraeni, Moh. Chaliffilardhy, Firhad Rinaldi Saputra, Hendro Febrian, Shindy Maria Ulfah, Kelas K Informatika 2014 dan seluruh pihak yang tidak bisa penulis sebutkan satu per satu yang telah memberikan kesan dan pesan selama penulis menempuh perkuliahan di Fakultas Ilmu Komputer Universitas Brawijaya.
12. Perkumpulan mahasiswa asal Cirebon, Ikatan Mahasiswa Keluarga Cirebon khususnya Indriani Dwi Rahayu, Alan Pratama, Olsen Hersaputra, Rijalkhairidin, Sri Indah Sari, M. Yusuf, Mugni Labib, Shella Oktavia, Ria Amelia, Naufal Firdaus dan Saqila Citra yang telah memberikan dukungan dan motivasi kepada penulis dalam penulisan skripsi
13. Serta semua phiak yang tidak bisa penulis sebutkan satu per satu yang juga turut memberikan dukungan, doa, semangat dan motivasi kepada penulis.

Penulis menyadari betul bahwa skripsi ini masih memiliki kekurangan dan jauh dari kata sempurna. Oleh sebab itu, penulis mengharapkan adanya saran dan kritik guna membangun dan memperbaiki kekurangan. Akhir kata, penulis sangat berharap skripsi ini bisa memberikan manfaat kepada semua pihak.

Malang, 15 Juni 2018

Raden Rizky Widdie Tigusti
rizkywiddie@yahoo.com

ABSTRAK

Raden Rizky Widdie Tigusti. 2018. Implementasi Fuzzy K-Nearest Neighbor (FK-NN) untuk Mengklasifikasi Fungsi Senyawa Berdasarkan *Simplified Molecular Input Line Entry System (SMILES)*. Skripsi Program Informatika / Ilmu Komputer, Fakultas Ilmu Komputer, Universitas Brawijaya

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

Senyawa aktif adalah senyawa kimia yang memiliki banyak fungsi. Salah satu fungsi dari senyawa aktif yaitu sebagai obat. Senyawa aktif memiliki karakteristik khusus yang menentukan fungsi sebagai obat. Untuk mendapatkan nilai karakteristik pada senyawa aktif digunakan notasi *SMILES* sebagai masukan agar dapat dengan mudah dilakukan proses komputasi pada sistem. Notasi *SMILES* merupakan notasi kimia modern yang dapat disimpan pada variabel string sehingga memudahkan proses klasifikasi pada sistem. Untuk mendapatkan karakteristik pada senyawa notasi *SMILES* yang dijadikan sebagai masukan akan dibagi kedalam 12 fitur yang terdiri dari atom B, C, N, O, P, S, F, Cl, Br, I, OH dan panjang dari notasi *SMILES*. Nilai dari setiap fitur didapatkan dari proses *preprocessing* terhadap notasi *SMILES* yang dilakukan pada awal proses klasifikasi.

Pada proses klasifikasi fungsi senyawa aktif menggunakan metode *fuzzy k-nearest neighbor* yang memiliki kelebihan dalam pemrosesan data dalam jumlah besar. Metode *fuzzy k-nearest neighbor* merupakan penggabungan dari dua metode yaitu metode *fuzzy* dan *k-nearest neighbor*. Langkah penting dari proses klasifikasi dengan menggunakan metode *fuzzy k-nearest neighbor* yaitu dimulai dengan menghitung jarak dari setiap data uji terhadap data latih dengan menggunakan metode *euclidean distance*, mengambil nilai sebanyak *k* dan menghitung *fuzzy*. Pengujian pada penelitian ini menggunakan dataset sebanyak 631 dan dibagi menjadi 2 yaitu data latih dan data uji dengan masing-masing komposisi data latih dan data uji sebesar 80% (503 data) dan 20% (128 data). Hasil dari pengujian didapatkan nilai akurasi sebesar 71% dengan nilai *k* sebesar 15, sedangkan pada pengujian *k-fold cross validation* nilai akurasi terbesar yang didapatkan yaitu sebesar 77%.

Kata Kunci: Senyawa Aktif, *SMILES*, *Fuzzy K-Nearest Neighbor*

ABSTRACT

The active compound is a chemical compound that has many functions. One of the functions of the active compound is as a medicine. Active compounds have special characteristics that determine function as a drug. To obtain a characteristic value on the active compound, SMILES notation is used as input system. SMILES notation is a modern chemical notation that can be stored on string variables to use for the process of computing. To obtain the characteristic on the compound the SMILES notation will be divided into 12 features consisting of B, C, N, O, P, S, F, Cl, Br, I, OH and the length from SMILES notation. The value of each feature is obtained from the preprocessing process against the SMILES notation made at the beginning of the classification process.

In the process of classifying the function of active compounds, the Fuzzy K-Nearest Neighbor method is used because it can do process by using large amounts of data. The Fuzzy K-Nearest Neighbor method is a combination of two methods namely Fuzzy and K-Nearest Neighbor. An important step of the classification process using the Fuzzy K-Nearest Neighbor is to calculate the distance from each test data to the train data or so-called by euclidean distance, pick value as much as k value and calculate the fuzzy. Testing process is using 631 dataset and the dataset will be divided in 2 class. Each composition of data training and data testing are 80% (503 data) and 20% (128 data). The result of the accuracy is 71% with the value of k = 15, in other test by using k-fold cross validation the biggest accuracy is 77%.

Keywords: Active Compounds, SMILES, Fuzzy K-Nearest Neighbor

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

BAB 1 PENDAHULUAN

1.1 Latar Belakang

Senyawa aktif merupakan senyawa kimia yang terdapat dalam tumbuhan atau hewan dan digunakan sebagai bahan obat yang memiliki efek fisiologis terhadap organisme lain, atau sering disebut senyawa bioaktif (Marisa, et al., 2011). Sedangkan pada pengertian yang lain menyebutkan bahwa senyawa aktif merupakan zat yang memiliki kemampuan untuk mencegah terjadinya berbagai kondisi buruk tubuh saat proses metabolisme atau mencegah masalah kesehatan dan menjaga kesehatan manusia (Suharto, et al., 2012). Senyawa aktif berperan penting sebagai obat yang dapat digunakan dalam proses pencegahan atau penyembuhan penyakit, terutama dalam bidang medis dan farmasi.

Proses pengenalan fungsi senyawa aktif pada bidang kimia pada umumnya yaitu dengan cara melalui proses uji laboratorium. Menurut penelitian yang dilakukan oleh Rosidah R Radam dan Erni Purnamasari pada tahun 2016, dilakukan penelitian untuk mengetahui fungsi dari senyawa kimia pada akar nipah melalui proses uji fitokimia. Pada proses uji fitokimia pada akar nipah dilakukan serangkaian uji laboratorium untuk mengidentifikasi senyawa aktif. Pada penelitian tersebut digunakan beberapa bahan kimia pendukung dan alat-alat laboratorium seperti tabung reaksi, *Hot Plate, Waterbath*, dll (Radam & Purnamasari, 2016).

Hal tersebut tentu membuat proses pengenalan fungsi dari suatu senyawa cenderung berjalan lebih lama. Para kimiawan harus terlebih dahulu melakukan uji laboratorium untuk mengetahui fungsi dari senyawa aktif. Maka dari itu pada penelitian ini penulis membuat sistem klasifikasi fungsi senyawa aktif untuk mempermudah kimiawan dalam proses pengenalan fungsi senyawa. Proses klasifikasi fungsi senyawa pada sistem juga dilakukan dengan mempelajari karakteristik dari rumus senyawa aktif.

Senyawa aktif terdiri dari susunan dan ikatan atom-atom yang bentuknya sangat beragam. Untuk dapat mengetahui atom apa saja yang ada pada senyawa aktif maka digunakan notasi *SMILES*. *SMILES* merupakan sebuah notasi kimia *modern* yang diciptakan oleh David Weininger yang terdiri dari karakter-karakter *ASCII (American Standart Code for Information Interchange)* sehingga dapat disimpan pada variabel *string* dan dapat diolah secara komputasi untuk dilakukan proses klasifikasi.

Fuzzy K-Nearest Neighbor (FK-NN) merupakan pengembangan dari *K-Nearest Neighbor* (K-NN) yang digabung dengan teory *fuzzy* (Prasetyo, 2012). FK-NN merupakan salah satu metode klasifikasi yang bersifat *lazy-learning* yang mengakibatkan proses pelatihan berlangsung sangat cepat. Selain itu penggunaan konsep derajat keanggotaan juga menambah keakuratan dalam proses klasifikasi (Prasetyo, 2012). Dengan menggunakan *fuzzy* maka setiap data memiliki nilai keanggotaan pada setiap kelas dengan memperhatikan nilai derajat keanggotaan pada nilai interval [0,1].

Metode *k-nearest neighbor* merupakan metode klasifikasi data dengan cara yang lebih sederhana namun kurang sesuai jika diterapkan pada model klasifikasi yang mengatur jumlah data latih yang memiliki nilai mayoritas. Hal tersebut dikarenakan prinsip klasifikasi dari metode *k-nearest neighbor* yaitu berdasarkan dengan data mayoritas yang ada pada nilai *k* (Wisdarianto, et al., 2013). Berdasarkan uraian tersebut maka pada penelitian ini proses klasifikasi menggunakan metode *fuzzy k-nearest neighbor*.

Penelitian dengan menggunakan metode *fuzzy k-nearest neighbor* sebelumnya sudah banyak dilakukan. Beberapa diantaranya pada tahun 2016 penelitian yang dilakukan oleh Septyan Teguh Mahendra yang berjudul "Klasifikasi Penyakit Ginjal Kronis Berdasarkan *Chronic Kidney Disease Dataset* Menggunakan Metode *Fuzzy K-Nearest Neighbor* (FK-NN)". Pada penelitian tersebut berhasil didapatkan nilai akurasi sebesar 96.67% dengan pengujian data latih sebanyak 60 dan 80.

Pada judul yang berbeda terdapat penelitian dilakukan oleh Nabil Syahiwa pada Tahun 2016 dengan judul "Implementasi *Fuzzy K-Nearest Neighbor* Untuk Diagnosis Penyakit Tiroid". Pada penelitian tersebut berhasil mendapatkan nilai akurasi sebesar 100% dengan menguji pengaruh keseimbangan kelas pada variasi jumlah data latih yang terdiri dari tiga kelas klasifikasi yaitu normal, *hipertiroid* dan *hipotiroid.*

Penelitian lain dilakukan oleh Hadistria Massalli pada tahun 2016 dengan judul "Penerapan *Fuzzy K-Nearest Neighbor* (FK-NN) Untuk Mengklasifikasi Penyakit Jantung Berdasarkan SPECTF Heart Dataset". Pada penelitian tersebut menggunakan 44 parameter dan berhasil mendapatkan akurasi terbesar 79% dengan 187 data uji.

Berdasarkan uraian tersebut, maka penulis melakukan penelitian dengan judul "**IMPLEMENTASI *FUZZY K-NEAREST NEIGHBOR* (FK-NN) UNTUK MENGKLASIFIKASI FUNGSI SENYAWA BERDASARKAN *SIMPLIFIED MOLECULAR-INPUT LINE-ENTRY SYSTEM (SMILES)***". Data yang digunakan untuk melakukan proses klasifikasi berupa *SMILES* yang terdiri dari atom penyusunnya dan panjang dari notasi *SMILES*. Kelas yang digunakan pada penelitian ini sebanyak 2 kelas penyakit yaitu kelas kanker dan metabolisme.

1.2 Rumusan Masalah

Rumusan masalah dalam penelitian ini yaitu:

1. Bagaimana cara kerja metode *fuzzy k-nearest neighbor* dalam klasifikasi fungsi senyawa aktif menggunakan notasi *SMILES*?
2. Bagaimana tingkat akurasi penggunaan metode *fuzzy k-nearest neighbor* dalam klasifikasi fungsi senyawa aktif menggunakan notasi *SMILES*?

1.3 Tujuan

1. Dapat mengenali fungsi pada suatu senyawa dengan menggunakan metode *fuzzy k-nearest neighbor* berdasarkan *SMILES*.
2. Dapat menghitung akurasi penggunaan metode *fuzzy k-nearest neighbor* untuk pengenalan fungsi senyawa berdasarkan *SMILES*.

1.4 Manfaat

Penelitian ini dilaksanakan untuk mengetahui apakah metode *fuzzy k-nearest neighbor* dapat memberikan hasil yang baik dengan menggunakan model data input *SMILES*. Apabila metode dapat memberikan hasil yang baik maka metode *fuzzy k-nearest neighbor* dapat digunakan untuk penelitian selanjutnya dengan menggunakan model data input *SMILES*.

1.5 Batasan Masalah

Pada penelitian ini penulis memberikan batasan masalah yaitu:

1. Senyawa yang digunakan merupakan senyawa aktif yang memiliki fungsi untuk penyembuhan penyakit.
2. Sistem dibuat dalam bentuk web dengan menggunakan Bahasa pemrograman PHP dan basis data *MySQL*.

1.6 Sistematika Penulisan

Sistematika pembahasan yang menjadi langkah-langkah proses penyusunan tugas akhir yaitu:

BAB I Pendahuluan

bab pendahuluan menjelaskan mengenai latar belakang, rumusan masalah, tujuan, manfaat, batasan masalah dan sistematika penulisan.

BAB II Landasan Kepustakaan

Bab landasan kepustakaan berisi penjelasan mengenai teori, konsep, model, metode yang berkaitan dengan penelitian yang sudah dilaksanakan sebelumnya. Dasar teori yang digunakan untuk mendukung penelitian ini adalah *smiles,* metode *fuzzy k-nearest neighbor* (fk-nn)*, k-nearest neighbor,* bahasa pemrograman php dan basis data *mysql*.

BAB III Metodologi

Bab metodologi membahas mengenai metode yang akan digunakan untuk mendapatkan hasil pada penelitian yang dilakukan. Bab metodologi terdiri studi literatur, pengumpulan data, analisis kebutuhan, perancangan sistem, implementasi sistem, pengujian dan analisis sistem kemudian diakhiri dengan kesimpulan dan saran.

BAB IV Perancangan

Pada bab perancangan berisi perancangan sistem, perhitungan manual, dan perancangan antarmuka. Pada bab ini juga dijelaskan analisis kebutuhan perangkat yang akan digunakan, yaitu perangkat keras dan lunak.

BAB V Implementasi

Pada bab implementasi dijelaskan mengenai spesifikasi sistem, implementasi kode program dan antarmuka sistem. Implementasi pada sistem mengikuti perancangan sistem yang ada pada bab sebelumnya.

BAB VI Pengujian dan Analisa

Bab ini menjelaskan mengenai hasil dari pengujian sistem dan analisis dari pengujian.

BAB VII Penutup

Bab ini menjelaskan kesimpulan dan saran dari penulis yang diperoleh dari pembuatan, pengujian dan analisis perangkat lunak untuk klasifikasi fungsi senyawa pada skripsi ini.

BAB 2 LANDASAN KEPUSTAKAAN

2.1 Kajian Pustaka

Kajian pustaka merupakan literatur yang digunakan sebagai referensi untuk mendukung penggunaan metode *fuzzy k-nearest neighbor* pada penelitian ini. Pada kajian pustaka dilakukan perbandingan pada masukan, proses dan keluaran. Perbandingan referensi dijelaskan pada Tabel 2.1.

Penelitian pertama yang berjudul "Klasifikasi Penyakit Ginjal Kronis Berdasarkan *Chronic Kidney Disease Dataset* Menggunakan Metode *Fuzzy K-Nearest Neighbor* (FK-NN)" oleh Septyan Teguh Mahendra. Pada penelitian tersebut menggunakan 24 atribut dengan 2 kelas *output*. Nilai akurasi dari penelitian tersebut sebesar 96.67% dengan jumlah data latih yang digunakan sebanyak 60 dan 80.

Penelitian kedua dilakukan oleh Nabil Syahiwa pada tahun 2016 dengan judul "Implementasi *Fuzzy K-Nearest Neighbor* Untuk Diagnosis Penyakit Tiroid". Dari jumlah data latih sebanyak 140 dan data uji sebanyak 75 diperoleh akurasi sebesar 100%. Penelitian ini memiliki atribut yaitu nilai T3RU, *thyroxin, triiodothyronine, TSH* dan *thyrotropin*.

Penelitian ketiga adalah penlitian yang dilakukan oleh Hadistria massalli dengan judul "Penerapan *Fuzzy K-Nearest Neighbor* (FK-NN) Untuk Mengklasifikasi Penyakit Jantung Berdasarkan SPECTF Heart Dataset" yang dilakukan pada tahun 2016. Atribut yang digunakan merupakan data penyakit jantung sebanyak 44 yang dikodekan menjadi F1R – F22R dan F1S – F22S. kelas output pada penelitian ini dibagi menjadi 2 kelas yaitu normal dan abnormal. Dari hasil penelitian diperoleh nilai akurasi sebesar 79% dari 20 data uji.

Tabel 2.1 Rincian metode

No	Judul	Obyek (Masukan)	Metode (Proses)	Hasil (Keluaran)
1	Klasifikasi Penyakit Ginjal Kronis Berdasarkan *Chronic Kidney Disease Dataset* Menggunakan Metode *Fuzzy K-Nearest Neighbor* (FK-NN)	Nilai atribut sebanyak 24 yang terdiri dari *age, blood pressure, specific gravity, albumin, sugar, red blood cells, pus cell, pus cell clumps, bacteria, blood glucose random, blood urea, serum*	Menggunakan metode FK-NN dengan alur proses sebagai berikut: 1. Menginputkan data latih dan data uji 2. Menormalisasikan setiap atribut pada data input 3. Memproses data yang telah diinput dengan k-NN	Terdapat 2 kelas output yaitu ckd dan notckd

No	Judul	Obyek (Masukan)	Metode (Proses)	Hasil (Keluaran)
		creatinine, sodium, potassium, hemoglobin, packed cell volume, white blood cell count, red blood cell count, hypertension, diabetes mellitus, coronary artery disease, appetite, pedal edema dan *anemia.*	4. Mentransformasikan hasil perhitungan k-NN kedalam *fuzzy* 5. Menampilkan hasil klasifikasi kelas penyakit ginjal kronis	
2	Implementasi *Fuzzy K-Nearest Neighbor* Untuk Diagnosis Penyakit Tiroid	Atribut yang digunakan berjumlah 5 yang terdiri dari *T3RU, thyroxin, triiodothyronine, TSH* dan *thyrotropin.*	Metode yang digunakan yaitu FK-NN dengan alur sebagai berikut: 1. Memasukkan data file yang berisi data latih, data uji dan nilai k 2. Normalisasi data latih dan data uji 3. Proses inisialisasi *fuzzy* 4. Proses k-NN menggunakan persamaan *Euclidian distance* 5. Proses *fuzzy k-nearest neighbor* dengan mencari nilai keanggotaan data 6. Menampilkan hasil keluaran berupa kelas klasifikasi	Terbagi menjadi 3 kelas klasifikasi yaitu normal, *hipertiroid* dan *hipotiroid.*

No	Judul	Obyek (Masukan)	Metode (Proses)	Hasil (Keluaran)
3	Penerapan *Fuzzy K-Nearest Neighbor* (FK-NN) Untuk Mengklasifikasi Penyakit Jantung Berdasarkan SPECTF Heart Dataset	Terdapat 42 atribut yang didapat dari data penyakit jantung yang dikodekan menjadi F1R – F22R dan F1S – F22S.	Menggunakan metode FK-NN dengan alur proses sebagai berikut: 1. Memasukkan data latih, data uji dan nilai k 2. Menghitung *Euclidean distance* *3.* menghitung *fuzzy k-nearest neighbor* 4. Menampilkan hasil keluaran klasifikasi	Hasil keluaran dibagi menjadi 2 kelas yaitu Normal dan Abnormal.
4.	Implementasi *Fuzzy K-Nearest Neighbor* (Fk-Nn) Untuk Mengklasifikasi Fungsi Senyawa Berdasarkan *Simplified Molecular Input Line Entry System (SMILES)*	Input berupa notasi SMILES (*Simplified Molecular Input Line System*)	Menggunakan metode FK-NN dengan alur dari proses yaitu: 1. Menginputkan data latih dan data uji 2. Melakukan preprocessing 3. Melakukan proses termweighting 4. Melakukan proses klasifikasi menggunakan FK-NN Menampilkan hasil klasifikasi	Berupa kelas penyakit dari senyawa aktif yang menunjukkan fungsi senyawa dalam penyembuhan penyakit.

2.2 *Simplified Molecular Input Line Entry System* (*SMILES*)

SMILES merupakan singkatan dari *Simplified Molecular Input Line Entry Sistem* merupakan sistem notasi kimia yang didesain untuk melakukan proses penyimpanan informasi kimia secara modern. *SMILES* diciptakan oleh David Weininger pada akhir tahun 1980 dengan konsep *graph*. Notasi *SMILES* terdiri dari karakter *ASCII* sehingga dapat disimpan kedalam variable *string*. Hal tersebut membuat notasi *SMILES* lebih mudah untuk diproses oleh computer dan cenderung memakan memori yang lebih sedikit (Junaedi, 2011).

Terdapat beberapa aturan untuk membuat notasi *SMILES* yaitu (Junaedi, 2011):

1. Penulisan Atom
   Atom direpresentasikan dengan simbol atomiknya masing-masing. Cara penulisannya yaitu hanya dengan menuliskan symbol dengan penulisan huruf besar. Apabila terdapat beberapa atom yang memiliki simbol lebih dari satu huruf, maka huruf pertama ditulis dengan huruf besar dan huruf selanjutnya ditulis dengan huruf kecil. Sebagai contoh untuk penulisan atom *carbon* adalah "C" dan *Bromium* adalah "Br".
2. Penulisan Ikatan
   Antar atom pasti memiliki ikatan. Ikatan atom terbagi menjadi tiga yaitu ikatan tunggal, ikatan rangkap dan rangkap tiga. Notasi untuk rangkap tunggal dilambangkan dengan "-", ikatan rangkap dilambangkan dengan "=" dan ikatan rangkap tiga dilambangkan dengan "#".
3. Penulisan Percabangan
   Notasi untuk percabangan pada ikatan dilambangkan dengan tanda kurung buka dan kurung tutup "()". Tanda kurung buka menandakan dimulainya percabangan sedangkan kurung tutup menandakan akhir dari percabangan.

2.3 Klasifikasi

Klasifikasi menurut (Kusnawi, 2007), *data mining* memiliki suatu fungsionalitas yang dapat menghasilkan model untuk melakukan prediksi kategori atau kelas dari objek-objek dalam baris data yang disebut klasifikasi. Proses klasifikasi memiliki dua tahapan yaitu tahap pembelajaran dan tahap klasifikasi.

Tahap pembelajaran data latih akan dianalisis untuk membangun sebuah model klasifikasi dengan menggunakan sebuah algoritma klasifikasi. Pada tahap ini dilakukan pembentukan fungsi atau pemetaan $y = f(x)$ dimana y merupakan kelas hasil prediksi dan x merupakan objek atau *tuple* yang akan diprediksi kelasnya.

Tahap selanjutnya adalah proses klasifikasi, yaitu akan dilakukan proses klasifikasi terhadap data menggunakan model yang telah dihasilkan dari proses pelatihan. Data uji digunakan untuk melakukan pengujian terhadap model guna mengetahui akurasinya. Data latih yang digunakan haruslah berbeda dengan data yang digunakan sebagai data uji. Apabila data yang digunakan dalam data latih sama dengan data uji maka akan menghasilkan nilai akurasi yang tinggi sedangkan jika menggunakan data yang berbeda untuk data latih dan ata uji maka hasilnya akan berpengaruh pada karakteristik data.

2.4 *Fuzzy K-Nearest Neighbor*

Fuzzy K-Nearest Neighbor (FK-NN) merupakan gabungan dari dua metode yaitu metode *fuzzy* dan metode *k-nearest neighbor*. Metode FK-NN menetapkan nilai keanggotaan sebagai pola jarak atau kesamaan dari sejumlah tetangga terdekat pada data *testing* dengan pemberian nilai keanggotaan pada kelas tertentu. Jika pada proses *k-nearest neighbor* hasil klasifikasi ditentukan dari banyaknya jumlah ketetanggaan yang lebih dominan, dengan ditambahkannya

metode *fuzzy* maka hasil akhirnya yaitu adalah kelas dengan nilai keanggotaan tertinggi.

Proses klasifikasi dengan menggunakan metode *fuzzy k-nearest neighbor* terdapat beberapa tahapan yaitu: (Prasetyo, 2012)

1. Input nilai data latih dan data uji
2. Melakukan normalisasi apabila rentang nilai antar parameter terlalu jauh.
3. Menghitung *euclidean distance* untuk mendapatkan nilai jarak antara data uji dan data latih. Persamaan e*uclidean distance* terdapat pada persamaan 2.1

$$d = \sqrt{\sum_{i=1}^{p}(x_{2i} - x_{1i})^2} \quad (2.1)$$

Keterangan:

x_1 = data latih

x_2 = data uji

i = variabel data

d = jarak terdekat

p = dimensi data

4. Jarak diurutkan dari nilai terkecil ke nilai terbesar dan mengambil data sebanyak *k*. Dalam penelitian ini proses pengurutan pada program menggunakan metode pengurutan *bubble sort*.
5. Menghitung nilai keanggotaan setiap data terhadap setiap kelas yang terdapat dalam dataset sejumlah *k* dengan menggunakan Persamaan 2.2

$$u_{ij} = \begin{cases} 0{,}51 + \left(\frac{n_j}{n}\right) \times 0{,}49, \ jika\ j = 1 \\ \left(\frac{n_j}{n}\right) \times 0{,}49, \ jika\ j = 0 \end{cases} \quad (2.2)$$

Keterangan:

u_{ij} = Nilai *membership* pada data ke i kelas j

n_j = Jumlah anggota kelas j pada suatu data latih n

n = Jumlah seluruh data latih yang digunakan

j = Kelas data

6. Mengambil data sebanyak nilai k, apabila nilai k = 5 maka memilih 5 jarak yang sudah diurutkan dari nilai terkecil ke nilai terbesar. Dari data yang sudah diambil sebanyak k maka akan dihitung nilai *membership* untuk setiap kelas klasifikasi. Persamaan yang digunakan untuk menghitung nilai *membership* terdapat pada Persamaan 2.3.

$$u_i(x) = \frac{\sum_{j=1}^{k} u_{ij}(||x - x_j||^{-2/(m-1)})}{\sum_{j=1}^{k}(||x - x_j||^{-2/(m-1)})} \quad (2.3)$$

Keterangan:

$u_i(x)$ = nilai keanggotaan *fuzzy*

$u_i(j)$ = nilai keanggotaan dalam tetangga i terhadap kelas j

k = banyaknya nilai ketetanggaan terdekat yang diambil, berupa bilangan integer positif

$x - x_j$ = selisih jarak antara data uji x ke data latih x_j dalam k tetangga terdekat

m = bobot pangkat yang besarnya $m > 1$

7. Hasil klasifikasi ditetapkan berdasarkan nilai *membership* data dengan nilai kelas terbesar maka data tersebut dianggap termasuk kedalam kelas tersebut.

2.5 *Preprocessing*

Preprocessing merupakan proses yang digunakan untuk mengubah bentuk data yang semula tidak terstruktur menjadi terstruktur (Manning, et al., 2008), *preprocessing* juga digunakan untuk melakukan jumlah dan letak huruf atau kata pada sebuah paragraf. Sedangkan *regular expression* (*Regex*) adalah sebuah fungsi yang digunakan untuk mencari pola pada sebuah kalimat (Muliantara, 2009). *Regular Expression* dapat diimplementasikan pada banyak bahasa program seperti *Python, PHP, Javascript,* dan lain sebagainya. Pada sistem ini *regex* digunakan untuk melakukan *preprocessing* pada data *SMILES* untuk mengetahui nilai fitur yang akan digunakan untuk melakukan proses klasifikasi.

2.6 *PHP Hypertext Processor* (PHP)

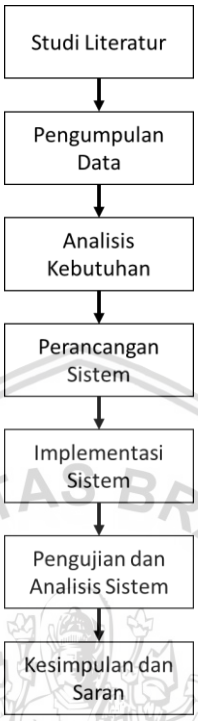
Menurut (Anhar, 2010), PHP singkatan dari PHP : *Hypertext Preprocessor* yaitu bahasa pemrograman *web server-side* yang bersifat *open source*. PHP merupakan kode program yang terintegrasi dengan HTML dan berada pada *server* (*server side HTML embedded scripting*). PHP adalah kode yang digunakan untuk membuat halaman *website* yang dinamis. Dinamis berarti halaman yang akan ditampilkan dibuat saat halaman itu diminta oleh *client*. Mekanisme ini menyebabkan informasi yang diterima client selalu yang terbaru. Semua kode PHP dieksekusi pada *server* yang mana kode tersebut dijalankan. Dapat dikatakan juga bahwa PHP *Hypertext Prepocessor*, yaitu bahasa pemrograman yang digunakan secara luas untuk penanganan pembuatan dan pengembangan sebuah situs *web*.

2.7 Basis Data *MySQL*

MySQL adalah suatu sistem manajemen data rasional (*RDBMS*) yang mampu bekerja secara cepat, kokoh dan mudah digunakan (Kadir, 2008). *Database* memungkinkan menyimpan, menelusuri, dan mengurutkan data secara efisien. *Server MySQL* yang membantu melakukan fungsionalitas tersebut. Bahasa yang digunakan *MYSQL* adalah *SQL*, standar bahasa *database* yang rasional di seluruh dunia saat ini.

BAB 3 METODOLOGI

Blok diagram pada metodologi dapat dilihat pada Gambar 3.1.



Gambar 3.1 Blok diagram metode penelitian

Pada Gambar 3.1 dijelaskan bahwa alur dari penelitian yaitu dimulai dengan melakukan studi literatur, pengumpulan data yang akan digunakan, analisis kebutuhan yang dibutuhkan sistem, perancangan sistem, pengujian dan analisis sistem, kemudian tahap akhir yaitu kesimpulan dan saran.

3.1 Studi Literatur

Pada tahap ini dilakukan pembelajaran menggunakan literatur dari berbagai bidang ilmu yang berhubungan dengan identifikasi fungsi senyawa antara lain:

1. *Fuzzy K-Nearest Neighbor* (FK-NN)
2. Fungsi Senyawa

3.2 Pengumpulan Data

Data yang digunakan dalam penerapan *fuzzy k-nearest neighbor* diambil dari laman https://pubchem.ncbi.nlm.nih.gov/ yang diambil dari dataset *PubChem BioAssay* dengan memilih senyawa yang memiliki fungsi sebagai obat. Data pada senyawa yang diambil merupakan kode *SMILES* agar dapat mudah dilakukan proses komputasi. Data terdiri dari notasi *SMILES*, 11 fitur yang terdiri dari atom *B, C, N, O, P, S, F, Cl, Br, I*, *OH* dan panjang dari notasi *SMILES* tersebut. Nilai panjang dari notasi *SMILES* dihitung berdasarkan seluruh atom yang ada pada notasi

SMILES, notasi percabangan senyawa, dan notasi ikatan rangkap. Angka pada notasi *SMILES* tidak dihitung dalam panjang *SMILES*.

3.3 Analisis Kebutuhan

Pada tahap ini dilakukan tahap analisis metode FK-NN yang akan digunakan mengenai parameter masukan yang akan digunakan. Kebutuhan yang harus dipenuhi oleh program yaitu dapat menerima masukan berupa notasi *SMILES*, melakukan proses *preprocessing* dan klasifikasi, kemudian menampilkan keluaran berupa hasil klasifikasi.

Sistem yang akan dibangun memiliki kebutuhan agar sistem dapat berjalan dengan baik. Kebutuhan terdiri dari kebutuhan perangkat keras, kebutuhan perangkat lunak dan kebutuhan data. Sistem dapat dijalankan selama pengguna dapat memenuhi kebutuhan yang dibutuhkan sistem.

3.4 Perancangan Sistem

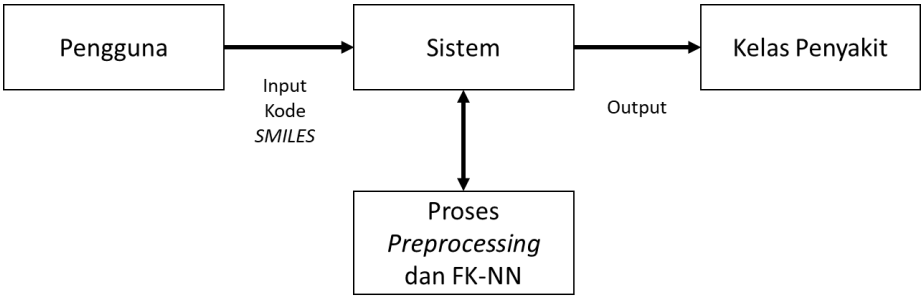
Perancangan sistem dilakukan untuk mempermudah proses implementasi. Perancangan sistem merupakan proses implementasi dari teori-teori yang ada, dan data yang digunakan serta ilmu yang didapatkan untuk merancang sistem klasifikasi fungsi senyawa. Beberapa tahap yang terdapat dalam perancangan meliputi diagram alir sistem, perhitungan manual dan perancangan antarmuka.

3.4.1 Deskripsi Umum Sistem

Sistem klasifikasi fungsi senyawa merupakan sistem yang digunakan untuk mengklasifikasikan fungsi senyawa berdasarkan fungsi sebagai obat terhadap suatu penyakit. Pada sistem ini metode yang digunakan untuk proses klasifikasi yaitu menggunakan metode *fuzzy k-nearest neighbor*. Selain itu inputan yang akan diberikan oleh user adalah berupa notasi *SMILES*. Setelah menerima inputan sistem akan melakukan proses *preprocessing* untuk mendapatkan nilai dari setiap fitur dan mengolah nilai tersebut untuk dilakukan proses klasifikasi. Setelah proses klasifikasi selesai sistem memberikan hasil keluaran berupa hasil klasifikasi.

3.4.2 Cara Kerja Sistem

Sistem memiliki alur kerja yaitu dimulai oleh pengguna yang akan memasukkan notasi *SMILES* yang kemudian akan diproses oleh sistem yang dimulai dengan proses *preprocessing* untuk mendapatkan nilai fitur. Setelah mendapatkan nilai fitur maka akan dilakukan proses klasifikasi dengan menggunakan metode *fuzzy k-nearest neighbor.* Setelah proses klasifikasi berhasil dilakukan maka sistem akan menampilkan keluaran berupa kelas klasifikasi penyakit. Alur kerja sistem dapat dilihat pada Gambar 3.2.



Gambar 3.2 Blok diagram alur kerja sistem

3.5 Implementasi Sistem

Implementasi sistem mengacu pada rancangan sistem yang sudah dibuat. Pada tahap implementasi, sistem akan menggunakan metode *fuzzy k-nearest neighbor* dalam proses klasifikasi. Implementasi antarmuka dan penyimpanan data juga diterapkan untuk memudahkan penggunaan sistem. Sistem diimplementasikan dengan menggunakan Bahasa pemrograman PHP dengan *tools* pendukung lainnya.

3.6 Pengujian dan Analisis Sistem

Tahap pengujian dan analisis merupakan tahap yang dilakukan apabila tahap implementasi telah selesai dilakukan. Pada tahap ini dilakukan pengujian terhadap sistem yang telah dibangun dengan tujuan untuk mengukur tingkat keberhasilan implementasi metode dalam penyelesaian masalah.

Analisis dilakukan untuk mengetahui berapa tingkat akurasi sistem dalam menentukan fungsi senyawa. Cara menghitung akurasi yaitu dengan menghitung seberapa banyak keberhasilan dari data yang diujikan.

$$Akurasi = \frac{Jumlah\ output\ program\ yang\ benar}{Total\ seluruh\ data\ uji} * 100\% \quad (3.1)$$

Pengujian sistem dilakukan untuk mengetahui apakah sistem berjalan sesuai dengan yang diharapkan dan tidak terjadi *error*. Pengujian dilakukan menggunakan metode *White Box Testing* yaitu dengan cara menelusuri secara detil algoritma yang ada pada program untuk meminimalisir *error* dan kesalahan logika.

3.7 Kesimpulan dan Saran

Setelah melalui beberapa tahap yang meliputi perancangan, implementasi dan pengujian algoritma maka pada tahap akhir penelitian dapat ditarik kesimpulan yang diambil dari hasil pengujian dan analisis. Penulisan saran juga dilakukan yang dapat dijadikan sebagai pertimbangan untuk penelitian selanjutnya.

BAB 4 PERANCANGAN

Bab ini akan membahas mengenai implementasi pada sistem klasifikasi fungsi senyawa menggunakan metode *fuzzy k-nearest neighbor* secara lebih detail. Pada bab ini juga terdapat contoh perhitungan manualisasi menggunakan metode *fuzzy k-nearest neighbor*.

4.1 Deskripsi Umum Sistem

Secara umum sistem memiliki proses masukan berupa notasi *SMILES* dan keluaran berupa kelas klasifikasi penyakit. Pada sistem ini kelas klasifikasi terdiri dari dua kelas penyakit yaitu kelas kanker dan metabolisme. Metode yang digunakan pada proses klasifikasi fungsi senyawa aktif yaitu *fuzzy k-nearest neighbor*. Untuk mengetahui lebih lanjut mengenai hasil dari implementasi metode *fuzzy k-nearest neighbor* pada penelitian ini maka dilakukan pengujian diantaranya pengujian variasi nilai *k*, variasi jumlah data latih dan variasi *k-fold cross validation*.

4.2 Analisis Kebutuhan Perangkat Lunak

Pada sub bab ini akan dijelaskan mengenai tahapan analisa kebutuhan apa saja yang dibutuhkan dalam menyelesaikan masalah untuk menentukan fungsi senyawa dengan menggunakan metode *fuzzy k-nearest neighbor*. Pada analisis kebutuhan perangkat lunak akan dibagi menjadi tiga yaitu kebutuhan *input*, proses dan *output*.

4.2.1 Analisis Kebutuhan *Input*

Analisis kebutuhan input merupakan kebutuhan masukkan yang digunakan pada perangkat lunak agar dapat mengetahui klasifikasi fungsi senyawa. Data yang digunakan sebagai *input* pada penelitian ini yaitu berupa notasi *SMILES*.

4.2.2 Analisis Kebutuhan Proses

Analisis kebutuhan proses merupakan tahap dimana perangkat lunak melakukan perhitungan dalam menentukan fungsi senyawa menggunakan metode *fuzzy k-nearest neighbor*. Proses yang dilakukan pada perangkat lunak yaitu:

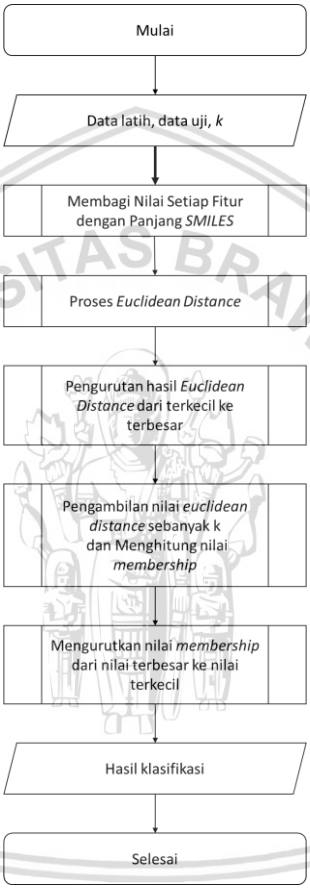
1. Dapat melakukan *import* data latih dan data uji dari file berbentuk *excel*.
2. Sistem mampu melakukan proses *preprocessing* untuk mendapatkan nilai fitur dari notasi *SMILES*.
3. Sistem mampu melakukan proses klasifikasi menggunakan metode *fuzzy k-nearest neighbor.*
4. Sistem dapat menampilkan hasil klasifikasi yaitu berupa fungsi dari senyawa aktif.

4.2.3 Analisis Kebutuhan *Output*

Setelah diketahui kebuthan *input* dan kebutuhnan proses maka langkah terakhir yaitu kebutuhan *output* dari sistem yang merupakan hasil klasifikasi dari fungsi senyawa. Kebutuhan *output* diharapkan dapat sesuai dengan apa yang diharapkan penulis mengenai klasifikasi fungsi senyawa.

4.3 Perancangan Sistem

Proses klasifikasi menggunakan metode *fuzzy k-nearest neighbor* memiliki beberapa tahapan yang akan dijelaskan melalui diagram pada Gambar 4.1.



Gambar 4.1 Alur proses klasifikasi dengan metode *Fuzzy K-Nearest Neighbor*

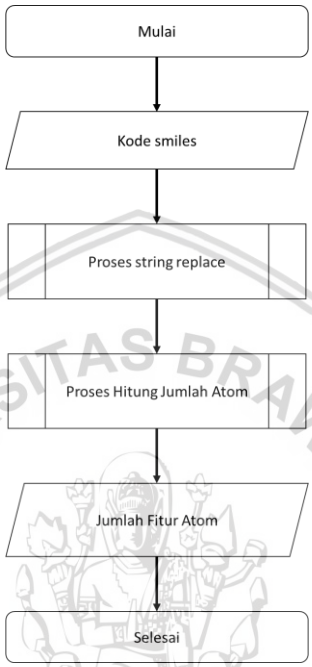
Berdasarkan Gambar 4.1 proses klasifikasi menggunakan metode *fuzzy k-nearest neighbor* memiliki tahapan yaitu:

1. Inisialisasi data latih, data uji dan nilai k
2. Membagi nilai setiap fitur dengan panjang *SMILES*
3. Menghitung jarak *euclidean distance*
4. Mengurutkan nilai *euclidean distance* dari nilai terendah ke nilai tertinggi
5. Menghitung nilai *membership* dengan menggunakan nilai *euclidean distance* sebanyak *k*
6. Mengurutkan nilai *membership* dari nilai tertinggi ke nilai terendah.

7. Menetapkan nilai *membership* terbesar di setiap kelas sebagai hasil klasifikasi.

4.3.1 Proses *Preprocessing*

Preprocessing pada sistem digunakan untuk mendapatkan nilai dari setiap atom yang akan digunakan sebagai fitur pada proses klasifikasi fungsi senyawa. Alur dari proses *preprocessing* ditunjukkan pada Gambar 4.2.

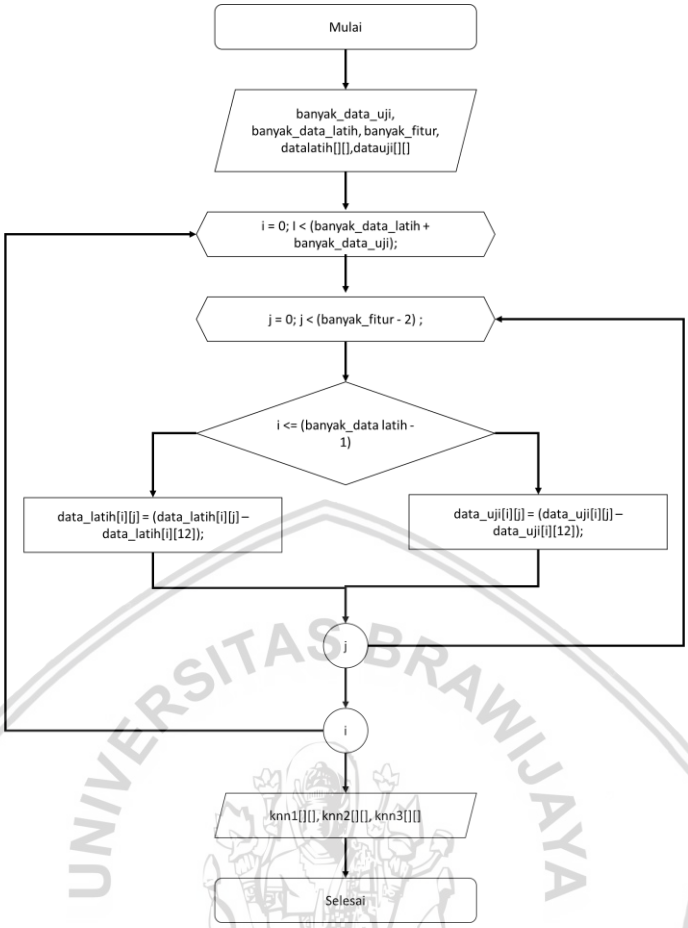


Gambar 4.2 Alur proses *Preprocessing*

Proses *preprocessing* dimulai dengan menerima masukan berupa notasi *SMILES*. Setelah menerima masukan maka proses selanjutnya adalah proses *string replace* yaitu untuk menghilangkan angka karena angka tidak diperhitungkan pada proses perhitungan jumlah setiap atom dan panjang *SMILES*. Kemudian proses selanjutnya yaitu menghitung jumlah atom dan menyimpan hasil perhitungan kedalam *database*.

4.3.2 Proses Membagi Seluruh Nilai Fitur dengan Nilai Panjang *SMILES*

Setelah didapatkan nilai dari setiap fitur melalui proses *preprocessing,* maka selanjutnya yaitu membagi nilai dari setiap fitur dengan nilai dari panjang *SMILES*. Alur proses dari membagi setiap nilai fitur dengan panjang *SMILES* ditunjukkan pada Gambar 4.3.

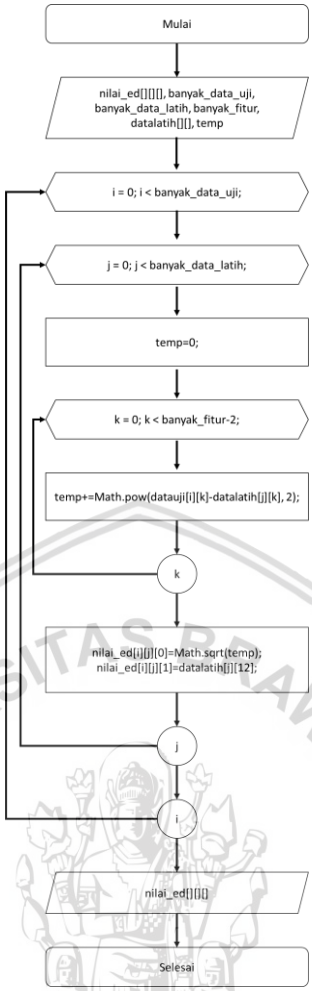


Gambar 4.3 Alur proses membagi nilai setiap fitur dengan panjang *SMILES*

Pada proses membagi nilai setiap fitur dengan panjang *SMILES* proses dimulai dengan melakukan perulangan sebanyak jumlah fitur dan jumlah seluruh data. Kemudian sistem akan membagi nilai setiap fitur dari data latih dan data uji dan kembali menyimpan nilai tersebut pada variabel data latih dan data uji.

4.3.3 Proses *Euclidean Distance*

Euclidean distance merupakan nilai jarak dari setiap data uji terhadap masing-masing data latih. Alur dari proses menghitung nilai *euclidean distance* ditunjukkan pada Gambar 4.4.

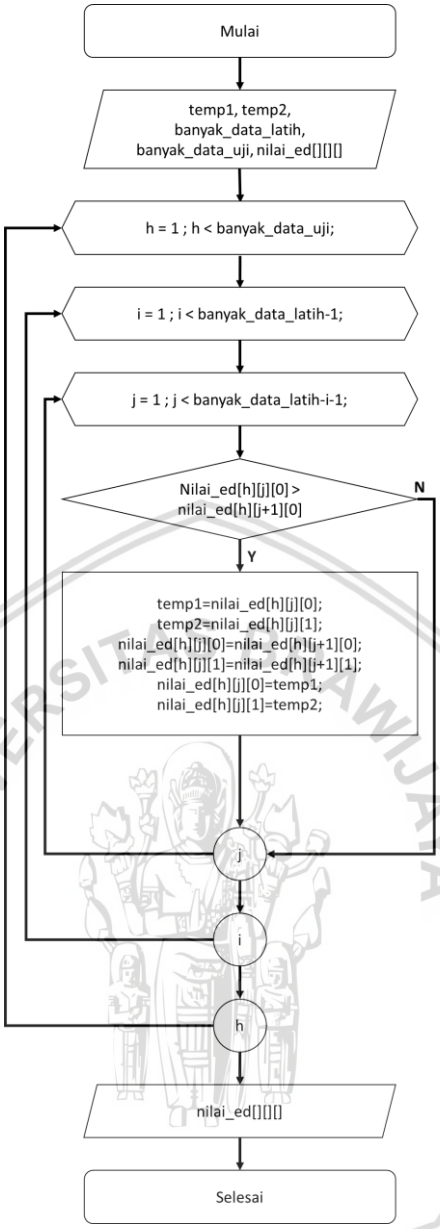


Gambar 4.4 Alur proses *euclidean distance*

Pada proses perhitungan *euclidean distance* setiap nilai fitur data uji akan dihitung selisih nilainya dengan seluruh nilai fitur data latih. Proses perhitungan dimulai dari indeks pertama hingga indeks terakhir dari masing-masing data uji dan data latih. Proses menghitung nilai *euclidean distance* terus dilakukan hingga masing-masing data uji dan data latih mencapai indeks terakhir.

4.3.4 Proses Pengurutan Nilai *Euclidean Distance*

Proses selanjutya yaitu mengurutkan nilai hasil perhitungan *euclidean distance* dari nilai terendah ke nilai tertinggi. Alur pada proses pengurutan tersebut dapat dilihat pada Gambar 4.5.

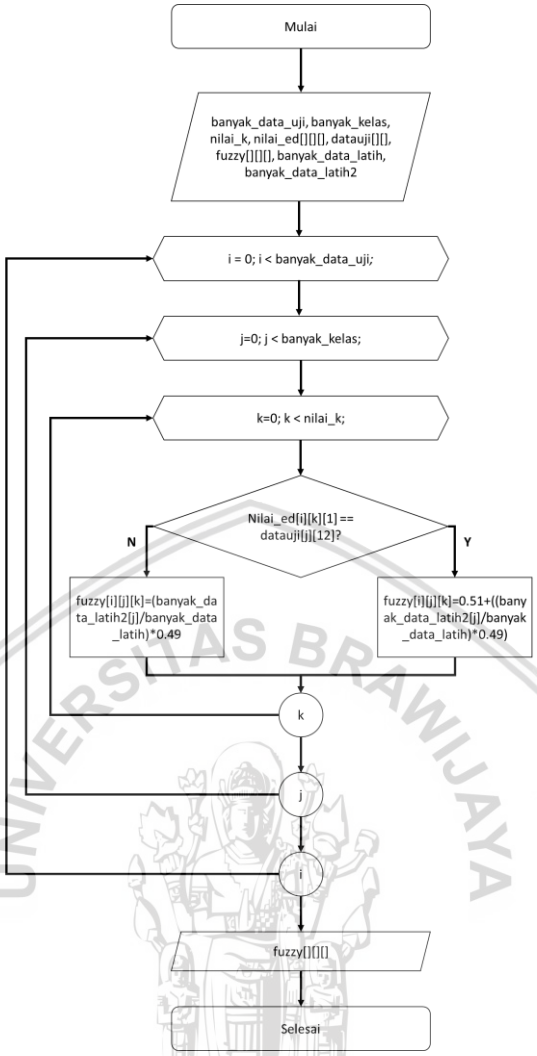


Gambar 4.5 Alur proses pengurutan hasil perhitungan *euclidean distance* dari nilai terkecil ke nilai terbesar

Proses pengurutan nilai *euclidean distance* menggunakan metode *sorting bubble sort*. Proses dimulai dengan inisialisasi nilai, melakukan perulangan dan membandingkan setiap data hingga data memiliki urutan dari nilai terendah ke nilai tertinggi.

4.3.5 Proses Menghitung Nilai *Fuzzy*

Proses selanjutnya yaitu menghitung nilai *fuzzy* untuk setiap kelas. alur pada proses menghitung nilai *fuzzy* ditunjukkan pada Gambar 4.6.

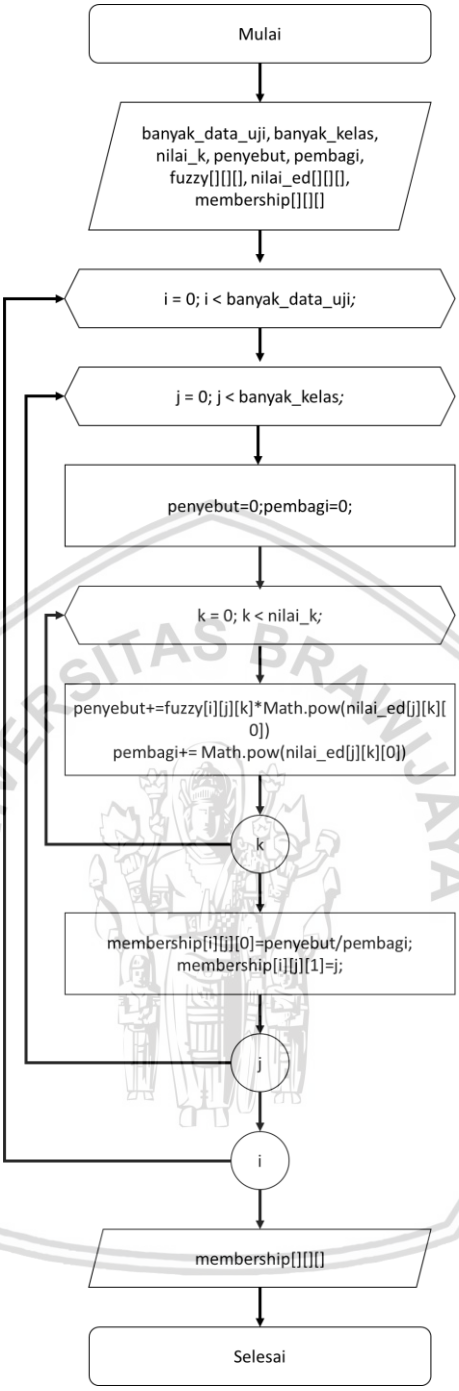


Gambar 4.6 Alur proses menghitung nilai *fuzzy* setiap kelas

Proses dalam menghitung nilai *fuzzy* setiap kelas yaitu dengan menghitung berapa jumlah data yang termasuk kedalam masing-masing kelas klasifikasi dan jumlah seluruh data latih. Setelah jumlah tersebut diketahui maka langkah berikutnya adalah mengambil hasil klasifikasi dari data yang sudah diambil sebanyak *k*. Pada perhitungan nilai *fuzzy* digunakan Persamaan 2.2 pada bab sebelumnya.

4.3.6 Proses Menghitung Nilai *Membership*

Proses selanjutnya yaitu menghitung nilai *membership* untuk setiap kelas. alur pada proses menghitung nilai *membership* ditunjukkan pada Gambar 4.7.

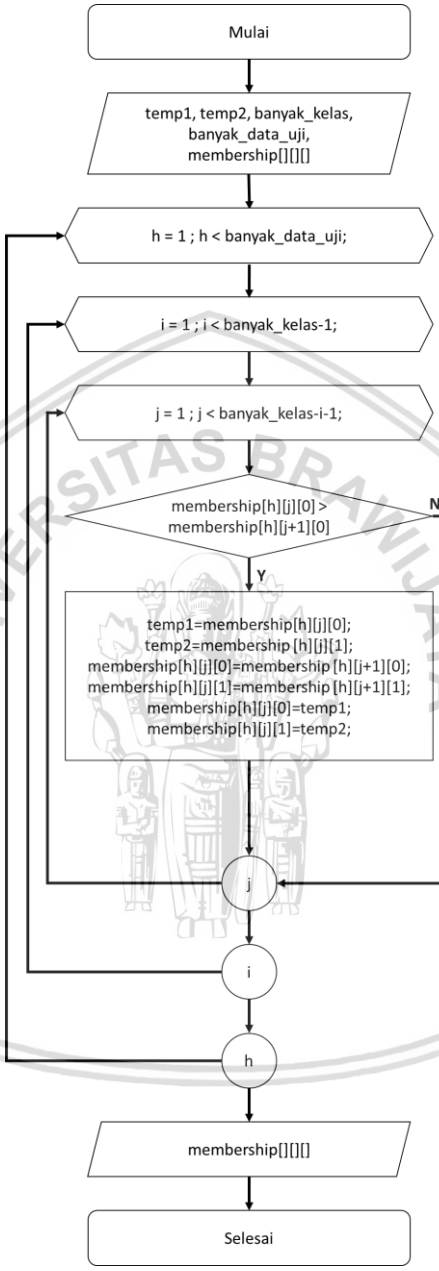


Gambar 4.7 Alur proses menghitung nilai *membership*

Proses dalam perhitungan nilai *membership* yaitu dengan melakukan perhitungan terhadap nilai fuzzy pada setiap kelas dan nilai *euclidean distance*. Nilai lain yang digunakan yaitu menggunakan banyak kelas, banyak data dan nilai *m* yang merupakan bobot pangkat. Pada proses perhitungan nilai *membership* terdapat 3 perulangan yang digunakan untuk mengulang perhitungan sebanyak jumlah banyak data, banyak kelas dan nilai *k*. Perhitungan nilai *membership* menggunakan Persamaan 2.3 pada bab sebelumnya.

4.3.7 Proses Mengurutkan Nilai *Membership* dan Menentukan Kelas Klasifikasi

Proses selanjutnya yaitu mengurutkan nilai *membership* untuk setiap kelas. alur pada proses menghitung nilai *membership* ditunjukkan pada Gambar 4.8.



Gambar 4.8 Alur proses mengurutkan nilai *membership* dan menentukan kelas klasifikasi

Pada proses mengurutkan nilai *membership* metode yang digunakan yaitu metode pengurutan *bubble sort*. Untuk menentukan kelas hasil klasifikasi yaitu dengan memilih kelas dengan nilai *membership* paling tinggi. Nilai *membership* diurutkan dari nilai terbesar ke nilai terkecil dari setiap kelas klasiikasinya. Nilai

terbesar dari hasil pengurutan nilai membership dipilih sebagai hasil dari klasifikasi.

4.4 Proses Perhitungan Manualisasi

Pada tahap ini akan dilakukan proses perhitungan secara manual menggunakan metode *fuzzy k-nearest neighbor* yang dimulai dari proses menghitung nilai *euclidean distance* hingga menentukan kelas klasifikasi. Pada perhitungan manualisasi menggunakan data sebanyak 30 data latih, 4 data uji dan 2 kelas klasifikasi yaitu kelas kanker dan metabolisme. Rincian mengenai data latih, data uji dan kelas klasifikasi akan ditunjukkan pada Tabel 4.1 dan Tabel 4.2.

Tabel 4.1 Data latih

No	Kode *SMILES*	Fitur												Klasifikasi
		Jumlah Atom											Panjang *SMILES*	
		B	C	N	O	P	S	F	Cl	Br	I	OH		
1	C1=CC(=C2C(=C1NCCNCCO)C(=O)C3=C(C=CC(=C3C2=O)O)O)NCCNCCO	0	22	4	4	0	0	0	0	0	0	2	50	Kanker
2	C1=CC=C(C=C1)C(=O)NCC(=O)O	0	9	1	2	0	0	0	0	0	0	1	24	Kanker
3	C1=CC=C(C=C1)C2=CC=CC=C2O	0	12	0	0	0	0	0	0	0	0	1	21	Kanker
4	C1=CC=C2C(=C1)C(=CN2)CC3=CNC4=CC=CC=C43	0	17	2	0	0	0	0	0	0	0	0	31	Kanker
5	C1=CC=C2C(=C1)C(=CN2)CO	0	9	1	1	0	0	0	0	0	0	0	19	Kanker
6	C1=CC=C2C(=C1)C(=O)C3=C(C2=O)C(=CC=C3)O	0	14	0	2	0	0	0	0	0	0	1	33	Kanker
7	C1C(O1)C2CO2	0	4	0	1	0	0	0	0	0	0	1	8	Kanker
8	C1C2=CC=CC=C2C3=CC=CC=C31	0	13	0	0	0	0	0	0	0	0	0	19	Kanker
9	C1C2N1C(=O)N=C2N	0	4	3	1	0	0	0	0	0	0	0	12	Kanker
10	C1CC(N2C=NC=C2C1)C3=CC=C(C=C3)C#N	0	14	3	0	0	0	0	0	0	0	0	27	Kanker

No	Kode *SMILES*	Fitur												Klasifikasi
		Jumlah Atom											Panjang *SMILES*	
		B	C	N	O	P	S	F	Cl	Br	I	OH		
11	C1CC1C(=O)N2CCN(CC2)C(=O)C3=C(C=CC(=C3)CC4=NNC(=O)C5=CC=CC=C54)F	0	24	4	3	0	0	1	0	0	0	0	54	Kanker
12	C1CC2C3C(C1O2)C(=O)OC3=O	0	8	0	3	0	0	0	0	0	0	1	19	Kanker
13	C1COCCN1N=O	0	4	2	2	0	0	0	0	0	0	0	10	Kanker
14	CC(=O)[O-].CC(=O)[O-].C1=CC(=C2C(=C1NCC[NH2+]CCO)C(=O)C3=C(C=CC(=C3C2=O)O)O)NCC[NH2+]CCO	0	26	4	8	0	0	0	0	0	0	4	80	Kanker
15	CC(=O)C1=CC=C(C=C1)[N+](=O)[O-]	0	8	1	3	0	0	0	0	0	0	0	29	Kanker
16	C1=CC=C(C=C1)[Si](Cl)(Cl)Cl	0	6	0	0	0	1	0	3	0	0	0	22	Metabolisme
17	C1=CC=C(C=C1)C#N	0	7	1	0	0	0	0	0	0	0	0	14	Metabolisme
18	C1=CC=C(C=C1)C(=O)NC(C(=O)O)O	0	9	1	2	0	0	0	0	0	0	2	27	Metabolisme
19	C1=CC=C(C=C1)C(C(=O)O)O	0	8	0	1	0	0	0	0	0	0	2	21	Metabolisme
20	C1=CC=C(C=C1)C(CCN)OC2=CC=C(C=C2)C(F)(F)F	0	16	1	1	0	0	3	0	0	0	0	37	Metabolisme
21	C1=CC=C(C=C1)C=CC=O	0	9	0	1	0	0	0	0	0	0	0	18	Metabolisme
22	C1=CC=C(C=C1)C=CCO	0	9	0	1	0	0	0	0	0	0	0	16	Metabolisme

No	Kode *SMILES*	Fitur												Klasifikasi
		Jumlah Atom											Panjang *SMILES*	
		B	C	N	O	P	S	F	Cl	Br	I	OH		
23	C1=CC=C(C=C1)C2=C(OC(=N2)CCC(=O)O)C3=CC=CC=C3	0	18	1	2	0	0	0	0	0	0	1	39	Metabolisme
24	C1=CC=C(C=C1)C2=CC(=O)C3=C(O2)C=CC=C3OC4C(C(C(C(O4)CO)O)O)O	0	21	0	3	0	0	0	0	0	0	5	51	Metabolisme
25	C1=CC=C(C=C1)C2=CC(=O)C3=CC=CC=C3O2	0	15	0	2	0	0	0	0	0	0	0	29	Metabolisme
26	C1=CC=C(C=C1)C2=NC3=CC=CC=C3C(=C2)C(=O)O	0	16	1	1	0	0	0	0	0	0	1	34	Metabolisme
27	C1=CC=C(C=C1)CCCCOCCCCCCNCC(C2=CC(=C(C=C2)O)CO)O	0	25	1	2	0	0	0	0	0	0	2	44	Metabolisme
28	C1=CC=C(C=C1)CCO	0	8	0	1	0	0	0	0	0	0	0	14	Metabolisme
29	C1=CC=C(C=C1)CN=C=S	0	8	1	0	0	1	0	0	0	0	0	17	Metabolisme
30	C1=CC=C(C=C1)COC(=O)C2=CC=CC=C2	0	14	0	2	0	0	0	0	0	0	0	27	Metabolisme

Tabel 4.2 Data uji

No	Kode *SMILES*	Fitur												Klasifikasi
		Jumlah Atom											Panjang *SMILES*	
		B	C	N	O	P	S	F	Cl	Br	I	OH		
1	CC1=C2C(C(=O)C3(C(CC4C(C3C(C(C2(C)C)(CC1OC(=O)C(C(C5=CC=CC=C5)NC(=O)OC(C)(C)C)O)O)OC(=O)C6=CC=CC=C6)(CO4)OC(=O)C)O)C)O	0	43	1	10	0	0	0	0	0	0	4	106	Kanker
2	CCCCC(C)C(C(CC(C)CCCCCCC(CC(C)N)O)O)OC(=O)CC(CC(=O)O)C(=O)O)OC(=O)CC(CC(=O)O)C(=O)O	0	34	1	8	0	0	0	0	0	0	6	85	Kanker
3	CC(C)C(C1=CC=C(C=C1)Cl)C(=O)OC(C#N)C2=CC(=CC=C2)OC3=CC=CC=C3	0	25	1	3	0	0	0	1	0	0	0	53	Metabolisme
4	CC(CC1=CC=C(C=C1)O)NCC(C2=CC(=CC(=C2)O)O)O	0	17	1	0	0	0	0	0	0	0	4	38	Metabolisme

Tabel 4.3 Hasil fitur data latih setelah dibagi dengan panjang *SMILES*

No	Fitur												Klasifikasi
	Jumlah Atom											Panjang *SMILES*	
	B	C	N	O	P	S	F	Cl	Br	I	OH		
1	0	0,4400	0,0800	0,0800	0	0	0	0	0	0	0,0400	50	Kanker
2	0	0,3750	0,0417	0,0833	0	0	0	0	0	0	0,0417	24	Kanker
3	0	0,5714	0	0	0	0	0	0	0	0	0,0476	21	Kanker
4	0	0,5484	0,0645	0	0	0	0	0	0	0	0	31	Kanker
5	0	0,4737	0,0526	0,0526	0	0	0	0	0	0	0	19	Kanker
6	0	0,4242	0	0,0606	0	0	0	0	0	0	0,0303	33	Kanker
7	0	0,5000	0	0,1250	0	0	0	0	0	0	0,1250	8	Kanker
8	0	0,6842	0	0	0	0	0	0	0	0	0	19	Kanker
9	0	0,3333	0,2500	0,0833	0	0	0	0	0	0	0	12	Kanker
10	0	0,5185	0,1111	0	0	0	0	0	0	0	0	27	Kanker
11	0	0,4444	0,0741	0,0556	0	0	0,0185	0	0	0	0	54	Kanker
12	0	0,4211	0	0,1579	0	0	0	0	0	0	0,0526	19	Kanker
13	0	0,4000	0,2000	0,2000	0	0	0	0	0	0	0	10	Kanker
14	0	0,3250	0,0500	0,1000	0	0	0	0	0	0	0,0500	80	Kanker
15	0	0,2759	0,0345	0,1034	0	0	0	0	0	0	0	29	Kanker
16	0	0,2727	0	0	0	0,0455	0	0,1364	0	0	0	22	Metabolisme
17	0	0,5000	0,0714	0	0	0	0	0	0	0	0	14	Metabolisme

No	Fitur												Klasifikasi
	Jumlah Atom											Panjang *SMILES*	
	B	C	N	O	P	S	F	Cl	Br	I	OH		
18	0	0,3333	0,0370	0,0741	0	0	0	0	0	0	0,0741	27	Metabolisme
19	0	0,3810	0	0,0476	0	0	0	0	0	0	0,0952	21	Metabolisme
20	0	0,4324	0,0270	0,0270	0	0	0,0811	0	0	0	0	37	Metabolisme
21	0	0,5000	0	0,0556	0	0	0	0	0	0	0	18	Metabolisme
22	0	0,5625	0	0,0625	0	0	0	0	0	0	0	16	Metabolisme
23	0	0,4615	0,0256	0,0513	0	0	0	0	0	0	0,0256	39	Metabolisme
24	0	0,4118	0	0,0588	0	0	0	0	0	0	0,0980	51	Metabolisme
25	0	0,5172	0	0,0690	0	0	0	0	0	0	0	29	Metabolisme
26	0	0,4706	0,0294	0,0294	0	0	0	0	0	0	0,0294	34	Metabolisme
27	0	0,5682	0,0227	0,0455	0	0	0	0	0	0	0,0455	44	Metabolisme
28	0	0,5714	0	0,0714	0	0	0	0	0	0	0	14	Metabolisme
29	0	0,4706	0,0588	0	0	0,0588	0	0	0	0	0	17	Metabolisme
30	0	0,5185	0	0,0741	0	0	0	0	0	0	0	27	Metabolisme

Tabel 4.4 Hasil fitur data uji setelah dibagi dengan panjang *SMILES*

No	Fitur												Klasifikasi
	Jumlah Atom											Panjang *SMILES*	
	B	C	N	O	P	S	F	Cl	Br	I	OH		
1	0	0,4057	0,0094	0,0943	0	0	0	0	0	0	0,0377	106	Kanker
2	0	0,4000	0,0118	0,0941	0	0	0	0	0	0	0,0706	85	Kanker
3	0	0,4717	0,0189	0,0566	0	0	0	0,0189	0	0	0	53	Metabolisme
4	0	0,4474	0,0263	0	0	0	0	0	0	0	0,1053	38	Metabolisme

4.4.1 Proses Menghitung Nilai *Euclidean Distance*

Proses awal adalah proses menghitung nilai selisih antara data uji dan data latih atau disebut dengan nilai *euclidean distance*. Dalam proses perhitungan nilai *euclidean distance* digunakan Persamaan 2.1 pada sub bab sebelumnya . Berikut adalah contoh perhitungan nilai *euclidean distance* untuk data uji ke 1 terhadap data latih ke 1.

$$d = \sqrt{\begin{matrix}(0-0)^2+(0{,}4400-0{,}4057)^2+(0{,}0800-0{,}0094)^2+\\(0{,}1724-0{,}0943)^2+(0-0)^2+\\(0-0)^2+(0-0)^2+\ (0-0)^2+(0-0)^2+(0{,}0400-0{,}0377)^2\end{matrix}}$$

$$= 0{,}0798$$

Cara yang sama dilakukan hingga data latih ke 30 . Setelah mencapai indeks data latih terakhir maka indeks data uji akan bertambah dan indeks data latih dimulai kembali dari awal. Data hasil perhitungan nilai *euclidean distance* dari data uji ke 1 terhadap seluruh data latih dapat dilihat pada Tabel 4.5.

Tabel 4.5 Hasil perhitungan *euclidean distance* data uji ke-1

No	*Euclidean Distance*	Klasifikasi
1	0,0798	Kanker
2	0,0460	Kanker
3	0,1912	Kanker
4	0,1837	Kanker
5	0,0983	Kanker
6	0,0403	Kanker
7	0,1325	Kanker
8	0,2967	Kanker
9	0,2543	Kanker
10	0,1828	Kanker
11	0,0946	Kanker
12	0,0677	Kanker
13	0,2212	Kanker
14	0,0913	Kanker
15	0,1378	Kanker
16	0,2208	Metabolisme
17	0,1519	Metabolisme
18	0,0879	Metabolisme

19	0,0787	Metabolisme
20	0,1164	Metabolisme
21	0,1092	Metabolisme
22	0,1647	Metabolisme
23	0,0734	Metabolisme
24	0,0709	Metabolisme
25	0,1209	Metabolisme
26	0,0943	Metabolisme
27	0,1704	Metabolisme
28	0,1718	Metabolisme
29	0,1430	Metabolisme
30	0,1211	Metabolisme

Setelah berhasil menghitung nilai *euclidean distance* data diurutkan dari nilai terkecil hingga ke nilai terbesar. Kemudian setelah berhasil diurutkan data diambil sebanyak k untuk dilakukan proses perhitungan *fuzzy*. Pada penelitian ini nilai k yang digunakan yaitu sebesar 5. Data lengkap untuk nilai *euclidean distance* data uji ke 1 yang sudah diurutkan ditunjukkan pada Tabel 4.6 berikut.

Tabel 4.6 Data nilai *euclidean distance* data uji ke-1 yang sudah diurutkan

No	*Euclidean Distance*	Klasifikasi
6	0,0403	Kanker
2	0,0460	Kanker
12	0,0677	Kanker
24	0,0709	Metabolisme
23	0,0734	Metabolisme

Berdasarkan pada Tabel 4.6 dapat disimpulkan data latih yang menunjukkan jarak terdekat dengan data uji berdasar nilai *euclidean distance* yaitu data ke 6, 2, 12, 24 dan 23. Dari hasil pengambilan nilai sebanyak *k* didapatkan kelas klasifikasi yaitu kelas kanker = 3 dan kelas metabolisme = 2.

4.4.2 Proses Menghitung Nilai *Fuzzy*

Proses *fuzzy m*erupakan proses selanjutnya setelah diketahui nilai *euclidean distance*. Pada proses *fuzzy* akan dilakukan perhitungan nilai *membership*. Proses dari menghitung nilai *fuzzy* menggunakan Persamaan 2.3 pada sub bab sebelumnya. Proses mengihtung nilai *fuzzy* dengan menggunakan 5

data hasil dari pengurutan nilai *euclidean distance.* Sebelum melakukan proses perhitungan nilai *membership* terlebih dahulu akan dilakukan proses inisialisasi *fuzzy.* Contoh perhitungan nilai inisialisasi *fuzzy* untuk kelas kanker adalah sebagai berikut.

- untuk kelas Kanker

$$u_{1(0)} = 0{,}51 \times \left(\frac{15}{30}\right) \times 0{,}49 = 0{,}7550$$

- untuk kelas bukan Kanker

$$u_{1(0)} = \left(\frac{10}{30}\right) \times 0{,}49 = 0{,}2450$$

Cara perhitungan yang sama juga dilakukan untuk menghitung inisialisasi *fuzzy* di kelas lainnya. Untuk nilai *fuzzy* pada semua kelas klasifikasi ditunjukkan pada Tabel 4.7.

Tabel 4.7 Nilai *fuzzy* kelas klasifikasi

Membership kelas Kanker ya	0,7550
Membership kelas Kanker tidak	0,2450
Membership kelas metabolisme ya	0,7550
Membership kelas metabolisme tidak	0,2450

Setelah diperoleh nilai *fuzzy* dari setiap kelas klasifikasi maka proses selanjutnya yaitu menghitung nilai *membership*. Nilai *membership* didapatkan dari hasil perhitungan nilai inisialisasi *fuzzy* dan nilai *euclidean distance*. Nilai inisialisasi *fuzzy* ditentukan dengan ketentuan apabila perhitungan dilakukan pada kelas kanker, data yang memiliki kelas klasifikasi kanker diberi nilai *fuzzy* 0,7550 dan selain data yang memiliki kelas klasifikasi kanker akan diberi nilai 0,2450. Ketentuan tersebut berlaku juga untuk kelas lainnya seperti kelas metabolism. Contoh perhitungan nilai *membership* pada seluruh kelas ditunjukkan pada perhitungan berikut.

1. Perhitungan *membership* kelas kanker

   Pada perhitungan *membership* untuk kelas kanker terdapat tiga data yang memiliki kelas klasifikasi kanker, sehingga hanya ada tiga data yang memiliki nilai inisialiasi *fuzzy* sebesar 0,7550 dan sisanya memiliki nilai *fuzzy* 0,2450. Hasil dari perhitungan *membership* kelas kanker didapatkan nilai 0,6389. Data dari nilai *euclidean distance* yang digunakan dapat dilihat pada Tabel 4.6 sedangkan nilai inisialisasi *fuzzy* yang digunakan dapat dilihat pada Tabel 4.7.

$$U_1(x) = \frac{\left(0{,}7550\left(0{,}0403^{\frac{2}{2-1}}\right)\right) + \left(0{,}7550\left(0{,}0460^{\frac{2}{2-1}}\right)\right) + \left(0{,}7550\left(0{,}0677^{\frac{2}{2-1}}\right)\right) + \left(0{,}2450\left(0{,}0709^{\frac{2}{2-1}}\right)\right) + \left(0{,}2450\left(0{,}0734^{\frac{2}{2-1}}\right)\right)}{0{,}0403^{\frac{2}{2-1}} + 0{,}0460^{\frac{2}{2-1}} + 0{,}0677^{\frac{2}{2-1}} + 0{,}0709^{\frac{2}{2-1}} + 0{,}0734^{\frac{2}{2-1}}}$$

$$= 0{,}6389$$

2. Perhitungan *membership* kelas metabolisme

   Pada perhitungan *membership* untuk kelas metabolisme terdapat dua data yang memiliki kelas metabolisme, sehingga terdapat dua data yang memiliki nilai inisialisasi *fuzzy* sebesar 0,7550 dan sisanya memiliki nilai 0,2450. Hasil dari perhitungan nilai *membership* untuk kelas metabolisme sebesar 0,3611. Data nilai *euclidean distance* yang digunakan dapat dilihat pada Tabel 4.6 sedangkan nilai inisialisasi *fuzzy* yang digunakan dapat dilihat pada Tabel 4.7.

$$U_1(x) = \frac{\left(0{,}2450\left(0{,}0403^{\frac{2}{2-1}}\right)\right) + \left(0{,}2450\left(0{,}0460^{\frac{2}{2-1}}\right)\right) + \left(0{,}2450\left(0{,}0677^{\frac{2}{2-1}}\right)\right) + \left(0{,}7550\left(0{,}0709^{\frac{2}{2-1}}\right)\right) + \left(0{,}7550\left(0{,}0734^{\frac{2}{2-1}}\right)\right)}{0{,}0403^{\frac{2}{2-1}} + 0{,}0460^{\frac{2}{2-1}} + 0{,}0677^{\frac{2}{2-1}} + 0{,}0709^{\frac{2}{2-1}} + 0{,}0734^{\frac{2}{2-1}}}$$

$$= 0{,}3611$$

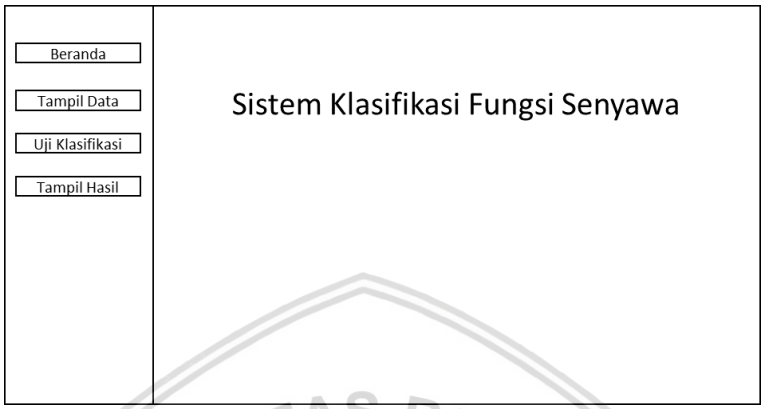
Dari perhitungan tersebut berhasil didapatkan nilai *membership* untuk masing-masing kelas. Pada akhir perhitungan didapatkan nilai terbesar pada kelas kanker yaitu dengan nilai 0.6389, sehingga dapat disimpulkan bahwa data uji termasuk kedalam kelas kanker. Dari seluruh data uji dan data latih yang digunakan dalam proses manualisasi didapatkan hasil akurasi sebesar 100%. Hasil klasifikasi untuk seluruh data uji ditampilkan pada Tabel 4.8.

Tabel 4.8 Hasil klasifikasi data uji

No	Kode SMILES	Fitur												Klasifikasi	Hasil Klasifikasi
		Jumlah Atom											Panjang Smiles		
		B	C	N	O	P	S	F	Cl	Br	I	OH			
1	CC1=C2C(C(=O)C3(C(CC4C(C3C(C(C2(C)C)(CC1OC(=O)C(C(C5=CC=CC=C5)NC(=O)OC(C)(C)C)O)O)OC(=O)C6=CC=CC=C6)(CO4)OC(=O)C)O)C)O	0	43	1	5	0	0	0	0	0	0	4	106	Kanker	Kanker
2	CCCCC(C)C(C(CC(C)CCCCCC(CC(C(C)N)O)O)OC(=O)CC(CC(=O)O)C(=O)O)OC(=O)CC(CC(=O)O)C(=O)O	0	34	1	8	0	0	0	0	0	0	6	85	Kanker	Kanker
3	CC(C)C(C1=CC=C(C=C1)Cl)C(=O)OC(C#N)C2=CC(=CC=C2)OC3=CC=CC=C3	0	25	1	3	0	0	0	1	0	0	0	53	Metabolisme	Metabolisme
4	CC(CC1=CC=C(C=C1)O)NCC(C2=CC(=CC(=C2)O)O)O	0	17	1	0	0	0	0	0	0	0	4	38	Metabolisme	Metabolisme
akurasi														100	%

4.5 Perancangan Antarmuka

Perancangan antarmuka akan menjelaskan mengenai tampilan yang akan dibuat pada sistem untuk implementasi metode *fuzzy k-nearest neighbor*. Pada perancangan antarmuka terdapat beberapa bagian yang memiliki fungsi masing-masing. Rancangan pada halaman awal ditunjukkan pada Gambar 4.9.



Gambar 4.9 Tampilan rancangan antarmuka halaman awal

Pada perancangan tampilan di halaman awal terdapat beberapa bagian yaitu:

a. *Menu bar* yang berisi pilihan menu untuk diakses pada sistem yang dibangun. menu terdiri dari menu *home*, uji yang berupa *dropdown* yang akan menampilkan tiga pilihan lainnya yaitu menampilkan pilihan uji klasifikasi, uji variasi k dan uji variasi data latih.

b. Judul dari sistem yang dibangun

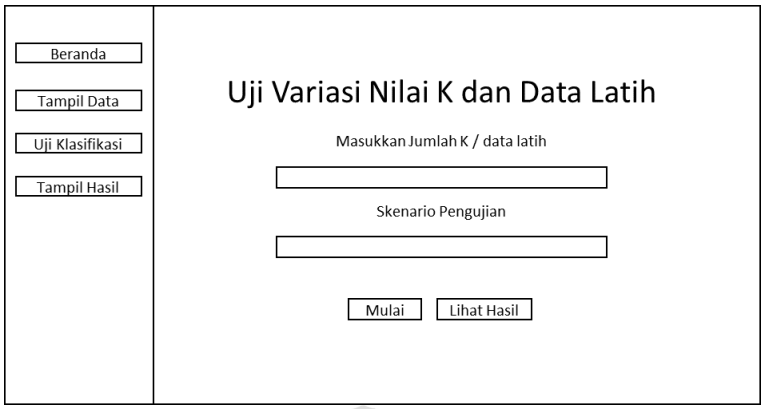
c. Tombol mulai untuk melakukan proses uji klasifikasi.

Selanjutnya adalah halaman untuk uji klasifikasi, setelah memilih melalui *dropdown* uji maka sistem akan menampilkan halaman uji klasifikasi. Rancangan mengenai tampilan halaman uji klasifikasi ditunjukkan pada Gambar 4.10.

Gambar 4.10 Tampilan rancangan antarmuka uji klasifikasi

Pada laman uji klasifikasi masih sama pada halaman awal yaitu masih terdapat *menu bar* dan judul, pada halaman uji klasifikasi ditambahkan kolom untuk memasukkan kode *SMILES* dan tombol mulai untuk memulai proses

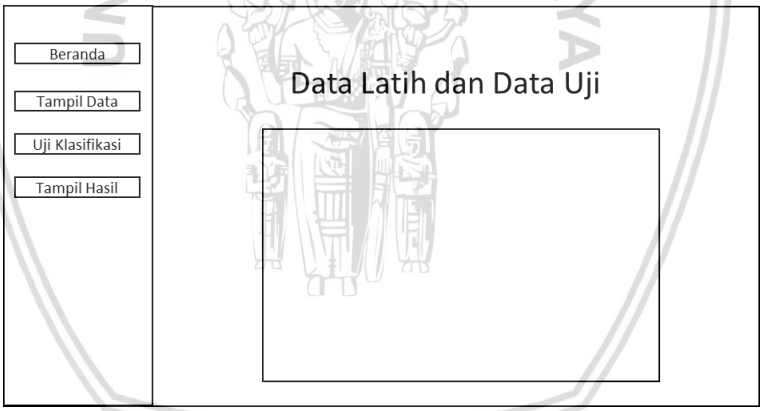
klasifikasi. Berikutnya adalah tampilan halaman untuk uji variasi nilai *k* dan data latih, rancangan dari halaman tersebut ditampilkan pada Gambar 4.11.



Gambar 4.11 Tampilan rancangan antarmuka uji variasi nilai *k* dan data latih

Pada halaman uji variasi nilai *k* sama dengan halaman uji klasifikasi namun pada halaman ini ditambahkan satu kolom untuk memasukkan berapa nilai *k* dan tombol untuk melihat hasil keseluruhan dari uji variasi nilai *k*.

Pada halaman berikutnya terdapat perancangan halaman untuk menampilkan data latih dan data uji. Perancangan dari halaman tersebut dapat dilihat pada Gambar 4.12.



Gambar 4.12 Tampilan rancangan antarmuka untuk menampilkan data latih dan data uji

Pada halaman tersebut terdapat tabel untuk menampilkan seluruh data latih dan data uji yang ada pada proses klasifikasi. Susunan tampilan lainnya masih sama dengan halaman-halaman sebelumnya.

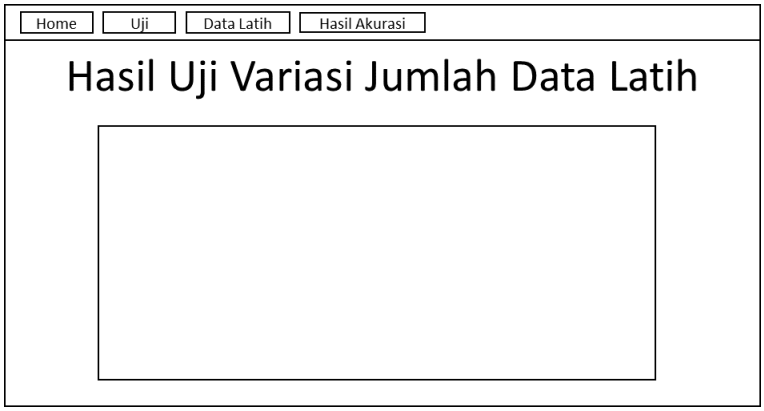
Berikutnya adalah halaman untuk menampilkan hasil dari uji klasifikasi. Terdapat dua rancangan dari halaman hasil klasifikasi yaitu hasil dari seluruh data uji dan hasil dari notasi *SMILES* yang dimasukkan oleh pengguna. Rancangan dari halaman hasil klasifikasi ditunjukkan pada Gambar 4.13.

Gambar 4.13 tampilan rancangan antarmuka untuk menampilkan hasil klasifikasi seluruh data uji

Halaman untuk menampilkan hasil uji klasifikasi pada Gambar 4.13 menampilkan tabel dan nilai akurasi pada akhir tabel. Pada halaman berikutnya adalah rancangan dari halaman hasil uji variasi *k* yang dapat dilihat pada Gambar 4.14.

Gambar 4.14 Tampilan rancangan antarmuka untuk menampilkan hasil uji variasi *k* dan data latih

Pada halaman tersebut terdapat tabel untuk menampilkan no, nilai *k* dan nilai akurasi. Tampilan yang ada pada halaman tersebut masih sama dengan halaman sebelumnya. Pada halaman berikutnya yaitu hasil uji variasi jumlah data latih. Rancangan dari halaman tersebut dapat dilihat pada Gambar 4.15.



Gambar 4.15 Tampilan rancangan antarmuka untuk menampilkan hasil uji variasi nilai *k* dan jumlah data latih

Pada halaman tersebut memilki tata letak yang sama dengan halaman sebelumnya, namun data yang ditampilkan berbeda. Pada halaman hasil uji variasi jumlah data latih menampilkan jumlah data latih yang digunakan dan akurasi yang dihasilkan dari proses klasifikasi.

4.6 Perancangan Pengujian

Pada tahap ini akan dilakukan pengujian dengan kondisi yang berbeda-beda untuk menguji kemampuan metode *fuzzy k-nearest neighbor* dalam melakukan proses klasifikasi. Terdapat berapa skenario pengujian diantaranya kondisi yang berbeda yaitu pengujian berdasarkan variasi nilai *k* dan variasi jumlah data latih yang digunakan. Hasil akhir dari pengujian ini akan dilihat berapa nilai akurasi yang didapat.

4.6.1 Pengujian Validitas Program

Pengujian validitas program dimaksudkan untuk mengetahui apakah keluaran yang dihasilkan oleh program sama dengan keluaran yang dihasilkan dari perhitungan manualisasi. Data latih dan data uji yang digunakan pada pengujian validitas program menggunakan data yang sama dengan data yang digunakan pada perhitungan manualisasi. Program dapat dikatakan valid apabila keluaran yang dihasilkan oleh program sama dengan keluaran yang dihasilkan dari perhitungan manualisasi.

4.6.2 Perancangan Pengujian Variasi Nilai K

Skenario pengujian berikutnya yaitu adalah pengujian dengan variasi inputan nilai *k* yang berbeda. Pada perancangan pengujian dengan variasi nilai *k* terdapat beberapa kondisi yaitu jika nilai *k*=15, *k*=25 dan *k*=35 dengan menggunakan seluruh data latih dan data uji. Tujuan dari pengujian ini yaitu untuk mengetahui pengaruh besaran nilai *k* terhadap nilai akurasi.

4.6.3 Perancangan Pengujian Pengaruh Jumlah Data Latih Terhadap Akurasi

Skenario pengujian yang terakhir adalah variasi pengaruh jumlah data latih terhadap akurasi. Pada pengujian ini terdapat tiga kondisi yaitu menggunakan 25% dari total masing-masing data latih dan data uji, 50% dari total masing-masing data latih, 75% dari total masing-masing data latih dan menggunakan seluruh data latih dan data uji. Pada percobaan ini nilai *k* yang digunakan yaitu sebesar 9. Tujuan dari pengujian ini yaitu untuk mengetahui pengaruh jumlah data latih yang digunakan terhadap nilai akurasi.

4.6.4 Pengujian *K-Fold Cross Validation*

Pengujian *k-fold cross validation* dilakukan untuk mengetahui akurasi program jika diuji dengan menggunakan sampel data yang acak. Pada metode ini dataset dibagi menjadi sebanyak *k* dataset dan dilakukan perulangan sebanyak *k*. pada setiap perulangan setiap *k* akan memiliki fungsi yang berganti sebagai data latih atau data uji. Pada penelitian ini dataset dibagi menjadi 10 fold. Masing-masing fold berisi data sebanyak 65 kecuali pada fold ke-10 berisi data sebanyak 56.

BAB 5 IMPLEMENTASI

Bab ini akan menjelaskan mengenai implementasi perhitungan *fuzzy k-nearest neighbor* kedalam kode program. Bab ini juga akan menjelaskan mengenai pengujian dan analisis dalam proses implementasi perhitungan *fuzzy k-nearest neighbor* dalam kode program.

5.1 Spesifikasi Sistem

Untuk dapat melakukan implementasi pada penelitian ini dibutuhkan spesifikasi perangkat lunak dan perangkat keras sebagai pendukung dalam membangun perangkat lunak. Spesifikasi sistem yang dibutuhkan yaitu perangkat lunak dan perangkat keras.

5.1.1 Spesifikasi Perangkat Lunak

Penelitian ini membuat perangkat lunak untuk mengimplementasikan metode *fuzzy k-nearest neighbor* untuk klasifikasi fungsi senyawa. Spesifikasi perangkat lunak yang digunakan pada penelitian ini ditunjukkan pada Tabel 5.1. dan spesifikasi *minimum* dari sistem agar dapat berjalan ditunjukkan pada Tabel 5.2.

Tabel 5.1 Spesifikasi Perangkat Lunak yang Digunakan

Nama Komponen	Spesifikasi
Sistem Operasi	Microsoft Windows 10 64bit
Bahasa Pemrograman	PHP
Tools Pemrograman	Sublime Text 3
Tools Database	MySQL
Aplikasi *Browser*	Google Chrome

Tabel 5.2 Spesifikasi Kebutuhan *Minimum* Perangkat Lunak

Nama Komponen	Spesifikasi
Sistem Operasi	Microsoft Windows 7 32bit
Bahasa Pemrograman	PHP
Tools Database	MySQL
Aplikasi *Browser*	Google Chrome, Mozilla Firefox, Opera

5.1.2 Spesifikasi Perangkat Keras

Spesifikasi perangkat keras dalam proses klasifikasi fungsi senyawa menggunakan metode *fuzzy k-nearest neighbor* ditunjukkan pada Tabel 5.3. dan spesifikasi *minimum* perangkat keras agar sistem dapat berjalan ditunjukkan pada Tabel 5.4.

Tabel 5.3 Spesifikasi Perangkat Keras yang Digunakan

Nama Komponen	Spesifikasi
Processor	Intel Core B950 2.1 GHz
RAM	6GB
Harddisk	500GB

Tabel 5.4 Spesifikasi Kebutuhan *Minimum* Perangkat Keras

Nama Komponen	Spesifikasi
Processor	Processor 1.5 GHz
RAM	1GB
Harddisk	80 GB

5.2 Implementasi Program

Tahap selanjutnya dari penelitian ini yaitu melakukan proses implementasi. Proses implementasi mengikuti proses perancangan yang telah dilakukan pada bab sebelumnya. Proses implementasi klasifikasi menggunakan metode *fuzzy k-nearest neighbor* dimulai dengan proses *preprocessing* untuk mendapatkan nilai fitur dari notasi *SMILES* yang telah dimasukkan kedalam *database*. Kemudian proses selanjutnya yaitu menghitung nilai *euclidean distance*, kemudian mengurutkan nilai *euclidean distance* dari nilai terendah ke nilai tertinggi menggunakan metode *sorting bubble sort,* menghitung nilai *fuzzy,* menghitung nilai *membership*, dan mengurutkan nilai *membership* dari nilai tertinggi ke nilai terendah. Hasil klasifikasi yaitu merupakan kelas dengan nilai *membership* tertinggi.

5.2.1 Implementasi Proses *Preprocessing*

Setelah data berhasil disimpan dalam *database* maka langkah awal yaitu melakukan proses *preprocessing* untuk mengetahui jumlah atom pada notasi *SMILES* agar dapat disimpan pada fitur sehingga bisa dilakukan proses klasifikasi. Kode program untuk proses *preprocessing* ditunjukkan pada Tabel 5.5.

Tabel 5.5 Kode program proses *preprocessing*

Algoritma 1: Preprocessing

```
1.  $SMILES = $kode_smiles;
2.  $resultB=0;$resultC=0;$resultN=0;$resultO=0;$resultP=0;
3.  $resultOH=0;$resultBr=0;$resultCl=0;$resultF=0;$resultI=0;$resul
4.  tS=0;
5.  $x=0; $countAngka=0;
6.  $hasilBr=0;$hasilCl=0;$hasilOH=0;$hasilC=0;$hasilB=0;$hasilN=0;$
7.  hasilP=0;$hasilF=0;$hasilI=0;$hasilO=0;$hasilS=0;
8.  //aturan perubahan untuk preprocesssing data
9.  $precOH    =    preg_replace("/[O][0-9][)]|[O][)]|[O]$/",    "OH",
10. $SMILES);
11. //print_r($precOH);
12. $OH = preg_replace("/[=][O][H]/", "=O)", $precOH);
13. //print_r($OH);
14. $tambahOH = preg_replace("/[O][H]/", "OH)", $OH);
15. //print_r($tambahOH);
16. $ubahOH = preg_replace("/[O][H][)]$/", "OH", $tambahOH);
17. //print_r($ubahOH);
18. //$ubahN = preg_replace("/[=]+[N]+[O]+[H]/", "=NO", $ubahOH);
19. //print_r($ubahN);
20. $ubahC = preg_replace("/[C]{1}[O][H]/", "CO", $ubahOH);
21. //print_r($ubahC);
    $preproc    =    preg_replace("/|([O][H]|[A-Z]|[A-Z][a-z]|[A-Z][a-
22. z]|[A-Z]{1}[\d]{1,}|[A-Z]|[()-+=.])/", ",$1", $ubahC);
23. //print_r($preproc);
24. $proccomma = preg_replace("/^[,]/", '', $preproc);
25. //print_r($proccomma);
26. //$procH = preg_replace("/[O]$/", 'OH', $proccomma);
27. $procbreak = explode(",", $proccomma); //memecah menjadi array
28. //print_r($procbreak);
29. $kurung= preg_replace("/\)|\(/", "", $procbreak); //menghilangkan
30. kurung
31. $rangkap= preg_replace("/\=/", "", $kurung); //menghilangkan tanda
32. rangkap
33. $rantai['panjang'] = count($procbreak);
34. for ($i = 0; $i < $rantai['panjang']; $i++) { //perulangan untuk
35. menghilangkan angka
36.             $arrayhuruf[$i]    =    preg_replace("/[1-9]/",    '',
37. $procbreak[$i]);
38.         }
39. $hasil=implode($arrayhuruf); //menggabungkan string
40. #echo "Hasil : $hasil <br>";
41. $ubahBr = str_replace("Br", "R", $hasil); //ubah Br jadi R
42. $ubahH =str_replace("OH", "H", $ubahBr); //ubah OH jadi H
43. $ubahCl =str_replace("Cl", "l", $ubahH); //ubah Cl jadi l
44. $Hasilpreprop=$ubahCl;
45. #echo "Hasil Preprocessing : $Hasilpreprop <br>";
46. $pj=strlen($Hasilpreprop);
47. #echo "Panjang SMILES : $pj<br>" ;
48. for ($x=0; $x <= strlen($Hasilpreprop)-1; $x++){
49.       if(substr($Hasilpreprop,$x,1) == "R"){
50.             $resultBr=substr_count($Hasilpreprop, "R");
51.             $hasilBr=$resultBr/$pj;
52.        }
53.       elseif (substr($Hasilpreprop, $x,1) == "l") {
54.             $resultCl=substr_count($Hasilpreprop, "l");
55.             $hasilCl=$resultCl/$pj;
56.       }
57.       elseif (substr($Hasilpreprop,$x,1) == "H"){
58.             $resultOH=substr_count($Hasilpreprop, "H");
59.             $hasilOH=$resultOH/$pj;
60.       }
61.       elseif (substr($Hasilpreprop,$x,1)== "C") {
```

Algoritma 1: Preprocessing

```
62.                 $resultC=substr_count($Hasilpreprop, "C");
63.                 $hasilC=$resultC/$pj;
64.           }
65.           elseif (substr($Hasilpreprop,$x,1)== "B") {
66.                 $resultB=substr_count($Hasilpreprop, "B");
67.                 $hasilB=$resultB/$pj;
68.           }
69.           elseif (substr($Hasilpreprop,$x,1)== "N") {
70.                 $resultN=substr_count($Hasilpreprop, "N");
71.                 $hasilN=$resultN/$pj;
72.           }
73.           elseif (substr($Hasilpreprop,$x,1) == "P"){
74.                 $resultP=substr_count($Hasilpreprop, "P");
75.                 $hasilP=$resultP/$pj;
76.           }
77.           elseif (substr($Hasilpreprop,$x,1)== "F"){
78.                 $resultF=substr_count($Hasilpreprop, "F");
79.                 $hasilF=$resultF/$pj;
80.           }
81.           elseif (substr($Hasilpreprop,$x,1)== "I"){
82.                 $resultI=substr_count($Hasilpreprop, "I");
83.                 $hasilI=$resultI/$pj;
84.           }
85.           elseif (substr($Hasilpreprop,$x,1)== "O"){
86.                 $resultO=substr_count($Hasilpreprop, "O");
87.                 $hasilO=$resultO/$pj;
88.           }
89.           elseif (substr($Hasilpreprop,$x,1)== "S"){
90.                 $resultS=substr_count($Hasilpreprop, "S");
91.                 $hasilS=$resultS/$pj;
92.           }
93.     }
94.     $kode_smiles = $SMILES;
95.     $fitur_b = $resultB;
96.     $fitur_c = $resultC;
97.     $fitur_n= $resultN;
98.     $fitur_o = $resultO;
99.     $fitur_p = $resultP;
100.    $fitur_s = $resultS;
101.    $fitur_f = $resultF;
102.    $fitur_cl = $resultCl;
103.    $fitur_br = $resultBr;
104.    $fitur_i = $resultI;
105.    $fitur_oh = $resultOH;
106.    $panjang_smiles = $pj;
107.    $kelas_klasifikasi = $kelas;
108.    $data = array(
109.           'kode_smiles' => $kode_smiles,
110.           'fitur_b' => $fitur_b,
111.           'fitur_c' => $fitur_c,
112.           'fitur_n' => $fitur_n,
113.           'fitur_o' => $fitur_o,
114.           'fitur_p' => $fitur_p,
115.           'fitur_s' => $fitur_s,
116.           'fitur_f' => $fitur_f,
117.           'fitur_cl' => $fitur_cl,
118.           'fitur_br' => $fitur_br,
119.           'fitur_i' => $fitur_i,
120.           'fitur_oh' => $fitur_oh,
121.           'panjang_smiles' => $panjang_smiles,
122.           'kelas_klasifikasi' => $kelas_klasifikasi
123.    );
124.    if($pilihan==0)
125.    {
```

Algoritma 1: Preprocessing	
126.	$this->m_data->input_data($data,'data_latih');
127.	}
128.	else if($pilihan==1)
129.	{
130.	$this->m_data->input_data($data,'data_uji');
131.	}
132.	else if($pilihan==2)
133.	{
134.	$this->m_data->input_data($data,'dataset');
135.	}

Penjelasan dari potongan program diatas yaitu sebagai berikut:

1. Pada baris ke-1 sampai pada baris ke-7 dilakukan deklarasi variabel untuk menyimpan data pada proses *preprocessing*.
2. Pada baris ke-8 sampai baris ke-47 dilakukan proses *preprocessing* untuk menampilkan huruf yang akan dihitung menggunakan fungsi `preg_replace` dan `str_replace`.
3. Pada baris ke-48 hingga baris ke-93 merupakan proses untuk menghitung berapa jumlah huruf yang muncul yang dihasilkan dari proses sebelumnya. Perulangan akan terus berjalan sebanyak jumlah dari nilai panjang *SMILES*.
4. Pada baris ke-94 sampai dengan baris ke-123 merupakan proses untuk menyimpan jumlah nilai setiap fitur kedalam *database*.
5. Baris ke-124 hingga baris ke-135 merupakan proses seleksi kondisi untuk menentukan nilai fitur akan disimpan kedalam *database* data_latih, data_uji atau dataset.

5.2.2 Implementasi Proses Klasifikasi Menggunakan Metode *Fuzzy K-Nearest Neighbor*

Tahap berikutnya dalam implementasi sistem klasifikasi fungsi senyawa aktif yaitu membagi setiap nilai fitur dengan nilai panjang dari notasi *SMILES*. Kode program untuk membagi setiap fitur dengan panjang *SMILES* dapat dilihat pada Tabel 5.6.

Tabel 5.6 Kode program proses membagi fitur dengan panjang *smiles*

Algoritma 2: Membagi setiap nilai fitur dengan panjang smiles	
136.	for($i=0;$i<$this->banyak_data_latih+$this-
137.	>banyak_data_uji;$i++){
138.	for($j=1;$j<$this->banyak_fitur-2;$j++){
139.	if ($i <= $this->banyak_data_latih-1) { $this->data_latih[$i][$j]=$this-
140.	>data_latih[$i][$j]/$this->data_latih[$i][12];}
141.	else{ $this->data_uji[$i-$this- >banyak_data_latih][$j]=$this->data_uji[$i-$this- >banyak_data_latih][$j]/$this->data_uji[$i-$this-
142.	>banyak_data_latih][12];
143.	}
144.	}
	}

Penjelasan dari potongan program diatas yaitu sebagai berikut:

1. Perulangan pada baris ke-1 untuk mendapatkan nilai dari variabel `i` dari indeks ke- 0 sampai dengan kurang dari banyaknya data latih yang ada pada sistem. Nilai variabel `i` akan digunakan sebagai index pada *array* `data_latih[][]`.
2. Perulangan pada baris ke-2 untuk mendapatkan nilai dari variabel `j` dari indeks ke- 0 sampai dengan kurang dari banyaknya fitur yang ada pada data latih. Nilai variabel `j` akan digunakan sebagai *index* pada *array* `data_latih[][]`.
3. Seleksi kondisi `if else` pada baris ke-3 digunakan untuk menyeleksi kondisi jika nilai pada variabel `i` memiliki nilai kurang dari nilai variabel `banyak_data_latih - 1` maka akan dilakukan proses perhitungan pembagian setiap nilai fitur pada data latih. Pada *array* `data_latih[][]` indeks nilai fitur disimpan pada indeks ke 1 hingga indeks ke 11, sedangkan pada indeks ke 0 berisi kode *SMILES* dan indeks ke 12 berisi kelas klasifikasi. Jika kondisi tidak memenuhi ketentuan maka akan dilakukan proses perhitungan pembagian nilai fitur pada data uji dengan cara penyimpanan indeks yang sama dengan data latih.

Setelah membagi nilai fitur dengan panjang *SMILES* tahap berikutnya adalah menghitung nilai jarak *euclidean distance*. Implementasi kode program untuk menghitung nilai *euclidean distance* ditunjukkan pada Tabel 5.7.

Tabel 5.7 Kode program proses menghitung *euclidean distance*

Algoritma 3: menghitung *euclidean distance*

```
1.  $temp=0;
2.  for($i=0;$i<$this->banyak_data_uji;$i++){
3.        for($j=0;$j<$this->banyak_data_latih;$j++){
4.              for($k=1;$k<$this->banyak_fitur-2;$k++){
5.                    $temp+=pow($this->data_uji[$i][$k]-$this->
    data_latih[$j][$k], 2);}
6.              $this->nilai_ed[$i][$j][0]=sqrt($temp);
7.              $this->nilai_ed[$i][$j][1]=$this->
    data_latih[$j][13];
8.              $temp=0;
9.        }
10. }
```

Penjelasan dari potongan program diatas yaitu sebagai berikut:

1. Pada baris ke-0 dilakukan deklarasi variabel `temp` dengan nilai 0. Variabel tersebut akan digunakan untuk menyimpan nilai sementara dari perhitungan nilai *euclidean distance*.
2. Pada baris ke-2 sampai baris ke-10 terdapat perulangan untuk menghitung nilai *euclidean distance*. terdapat 3 buah perulangan yaitu pada baris ke-2 untuk mengulang hingga sebanyak jumlah data uji, baris ke-3 untuk

mengulang hingga sebanyak jumlah data latih dan baris ke-4 untuk mengulang hingga sebanyak jumlah banyak fitur yang sudah dikurangi 2.

3. Pada baris ke-5 hingga baris ke-7 merupakan perhitungan dari nilai *euclidean distance*. pada baris ke-5 dilakukan proses perhitungan untuk menghitung jarak dari nilai setiap fitur dari data uji terhadap data latih yang akan disimpan pada variabel `temp`. proses ini terus berulang mengikuti perulangan pada baris ke-4. Sedangkan untuk baris ke-6 merupakan proses untuk menyimpan nilai dari penjumlahan variabel `temp` dan baris ke-7 untuk menyimpan hasil kelas klasifikasi. Kedua nilai pada baris ke-6 dan ke-7 disimpan pada variabel `nilai_ed` dengan indeks ke -0 untuk nilai *euclidean distance* dan indeks ke-1 untuk kelas klasifikasi.

Setelah mendapatkan nilai *euclidean distance* maka langkah selanjutnya yaitu mengurutkan nilai *euclidean distance* dari nilai terkecil hingga nilai terbesar menggunakan algoritma *sorting bubble sort*. Implementasi dari *sorting* nilai *euclidean distance* ditunjukkan pada Tabel 5.8.

Tabel 5.8 Kode program proses mengurutkan nilai *euclidean distance* dari nilai terendah ke nilai tertinggi

Algoritma 4: mengurutkan nilai *euclidean distance* dari nilai terendah ke tertinggi	
1.	`$temp1=0;$temp2=0;`
2.	`for($h=0;$h<$this->banyak_data_uji;$h++){`
3.	`for($i=0;$i<$this->banyak_data_latih-1;$i++){`
4.	`for($j=0;$j<$this->banyak_data_latih-$i-1; $j++){`
5.	`if($this->nilai_ed[$h][$j][0]> $this->nilai_ed[$h][$j+1][0]){`
6.	`$temp1=$this-> nilai_ed[$h][$j][0];`
7.	`$temp2=$this-> nilai_ed[$h][$j][1];`
8.	`$this->nilai_ed[$h][$j][0]=$this->nilai_ed[$h][$j+1][0];`
9.	`$this->nilai_ed[$h][$j][1]=$this->nilai_ed[$h][$j+1][1];`
10.	`$this->nilai_ed[$h][$j+1][0]= $temp1;`
11.	`$this->nilai_ed[$h][$j+1][1]= $temp2;`
12.	`}`
13.	`}`
14.	`}`
15.	`}`

Penjelasan dari potongan program diatas yaitu sebagai berikut:

1. Pada baris ke-1 dilakukan deklarasi variabel `temp1` dan `temp2` yang akan digunakan untuk menyimpan nilai sementara dari proses pengurutan nilai *euclidean distance*.
2. Dilakukan perulangan pada baris ke-2 hingga ke-4. Pada baris ke-2 dilakukan perulangan hingga sebanyak jumlah data uji, pada baris ke-3 dilakukan perulangan hingga sebanyak jumlah data latih dikurangi 1 dan pada baris ke-4 dilakukan perulangan hingga sebanyak jumlah data latih.
3. Pada baris ke-5 dilakukan seleksi kondisi untuk mengetahui apakah nilai pada indeks tersebut lebih besar dari nilai yang ada pada indeks berikutnya. Jika

hasil dari seleksi kondisi bernilai benar maka akan terjadi proses pemindahan indeks yang dilakukan pada baris ke-6 hingga baris ke-12.

Setelah nilai *euclidean distance* berhasil diurutkan selanjutnya adalah proses untuk menghitung nilai *fuzzy*. Proses untuk menghitung nilai *fuzzy* ditunjukkan pada Tabel 5.9.

Tabel 5.9 Kode program proses menghitung nilai *fuzzy*

Algoritma 5: menghitung *fuzzy*	
1.	`$this->bagi_data_latih();`
2.	`for($i=0;$i<$this->banyak_data_uji;$i++){`
3.	`for($j=0;$j<$this->banyak_kelas;$j++){`
4.	`for($k=0;$k<$this->nilai_k;$k++){`
5.	`if($this->nilai_ed[$i][$k][1]==$this->data_uji[$j][13]){`
6.	`$this->fuzzy[$i][$j][$k]=0.51+((($this->banyakdatalatih2[$j]/$this->banyak_data_latih)*0.49);}`
7.	`else{`
8.	`$this->fuzzy[$i][$j][$k]=(($this->banyakdatalatih2[$j]/$this-> banyak_data_latih)*0.49);}`
9.	`}`
10.	`}`
11.	`}`

Penjelasan dari potongan program diatas yaitu sebagai berikut:

1. Pada baris ke-1 dilakukan proses pemanggilan *method* `bagi_data_latih()` yang akan melakukan proses pengelompokan data latih menurut kelas klasifikasinya.
2. Dilakukan perulangan pada baris ke-2 hingga ke-4. Pada baris ke-2 dilakukan perulangan hingga sebanyak jumlah data uji, pada baris ke-3 dilakukan perulangan hingga sebanyak kelas dan pada baris ke-4 dilakukan perulangan hingga sebanyak nilai *k*.
3. Pada baris ke-5 dilakukan seleksi kondisi apabila nilai pada variabel `nilai_ed` memiliki nilai yang sama dengan variabel `data_uji` maka akan dilakukan proses perhitungan pada baris ke-6. Sedangkan apabila nilai tidak sama maka akan dilakukan perhitungan pada baris ke-8.
4. Proses perhitungan pada baris ke-6 dan ke-8 mengikuti rumus yang terdapat pada Persamaan 2.4. pada bab sebelumnya.

Setelah mendapatkan nilai *fuzzy* maka proses selanjutnya adalah menghitung nilai *membership* untuk setiap kelas klasifikasi. Implementasi kode program untuk menghitung nilai *membership* ditunjukkan pada Tabel 5.10.

Tabel 5.10 Kode program proses menghitung nilai *membership*

Algoritma 6: menghitung *membership*	
1.	`$penyebut=0;$pembagi=0;`
2.	`for ($i=0; $i < $this->banyak_data_uji; $i++) {`
3.	`for ($j=0; $j < $this->banyak_kelas; $j++) {`
4.	`for ($k=0; $k < $this->nilai_k; $k++) {`
5.	

Algoritma 6: menghitung *membership*	
6.	$penyebut+=$this->fuzzy[$i][$j][$k]*pow($this->nilai_ed[$j][$k][0], $this->m);
7.	$pembagi+=pow($this-> nilai_ed[$j][$k][0], $this->m);}
8.	$this->membership[$i][$j][0]= $penyebut/$pembagi;
9.	$this->membership[$i][$j][1]=$this->data_uji[$j][13];
10.	$penyebut=0;$pembagi=0;
11.	}
12.	$penyebut=0;$pembagi=0;
	}

Penjelasan dari potongan program diatas yaitu sebagai berikut:

1. Pada baris ke-1 dilakukan deklarasi variabel `penyebut` dan variabel `pembagi` yang masing-masing bernilai 0. Kedua variabel tersebut berfungsi untuk menyimpan nilai sementara pada proses perhitungan nilai *membership*.
2. Dilakukan perulangan pada baris ke-2 hingga ke-4. Pada baris ke-2 dilakukan perulangan hingga sebanyak jumlah data uji, pada baris ke-3 dilakukan perulangan hingga sebanyak kelas dan pada baris ke-4 dilakukan perulangan hingga sebanyak nilai *k*.
3. Pada baris ke-5 dan ke-6 dilakukan nilai perhitungan untuk menghitung nilai pada variabel `penyebut` dan variabel `pembagi` dan terus diulang mengikuti perulangan pada barisk ke-4.
4. Pada baris ke-7 dan ke-8 merupakan tahap akhir dari proses perhitungan nilai *membership*. Pada baris ke-7 nilai dari variabel `penyebut` akan dibagi dengan variabel `pembagi` dan disimpan pada variabel `membership` indeks ke-0. Sedangkan pada baris ke-8 dilakukan proses menyimpan data kelas klasifikasi yang akan disimpan pada pada variabel `membership` indeks ke-1.

Untuk dapat menentukan hasil klasifikasi maka nilai *membership* yang sudah dilakukan perhitungan perlu diurutkan dari nilai tertinggi ke nilai terendah *(descending)*. Pada pengurutan nilai tertinggi ke nilai terendah digunakan algoritma *sorting bubble sort*. Implementasi dari kode program *sorting bubble sort* ditunjukkan pada Tabel 5.11.

Tabel 5.11 Kode program proses mengurutkan nilai *membership* dari nilai tertinggi ke nilai terendah

Algoritma 7: mengurutkan nilai *membership* dari nilai tertinggi ke nilai terendah	
1.	$temp1=0;$temp2=0;
2.	for($h=0;$h<$this->banyak_data_uji;$h++){
3.	for($i=0;$i<$this->banyak_kelas-1;$i++){
4.	for($j=0;$j<$this->banyak_kelas-$i-1;$j++){
5.	if($this->membership[$h][$j][0]<$this->membership[$h][$j+1][0]){
6.	$temp1=$this->membership[$h][$j][0];
7.	$temp2=$this->membership[$h][$j][1];
8.	$this->membership[$h][$j][0]=$this->membership[$h][$j+1][0];
9.	

Algoritma 7: mengurutkan nilai *membership* dari nilai tertinggi ke nilai terendah

```
                        $this->      membership[$h][$j][1]=$this->
10.     membership[$h][$j+1][1];
                        $this-> membership[$h][$j+1][0]=$temp1;
11.                     $this->membership[$h][$j+1][1]=$temp2;
12.                     }
13.                 }
14.             }
        }
```

Penjelasan dari potongan program diatas yaitu sebagai berikut:

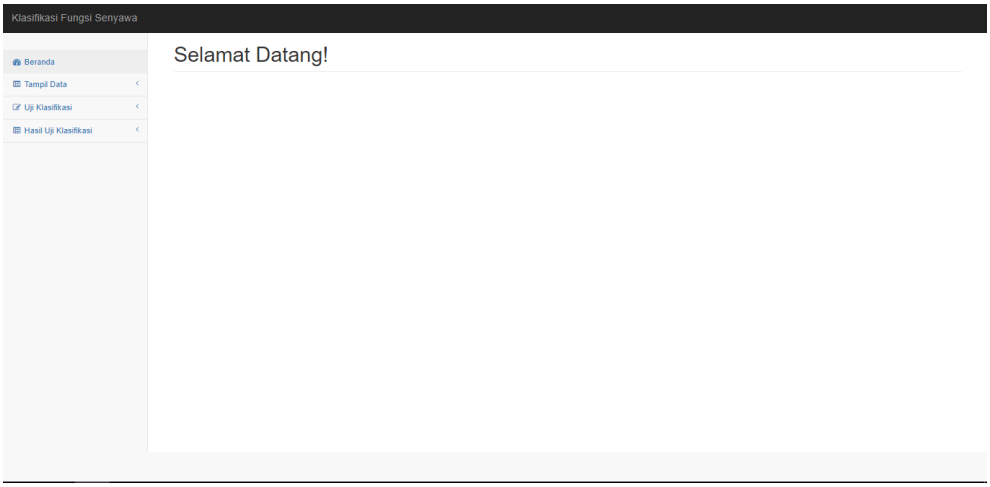
1. Pada baris ke-1 dilakukan deklarasi variabel `temp1` dan `temp2` yang akan digunakan untuk menyimpan nilai sementara dari proses pengurutan nilai *membership*.
2. Dilakukan perulangan pada baris ke-2 hingga ke-4. Pada baris ke-2 dilakukan perulangan hingga sebanyak jumlah data uji, pada baris ke-3 dilakukan perulangan hingga sebanyak kelas dikurangi 1 dan pada baris ke-4 dilakukan perulangan hingga sebanyak kelas.
3. Pada baris ke-5 dilakukan seleksi kondisi untuk mengetahui apakah nilai pada indeks tersebut kurang dari nilai yang ada pada indeks berikutnya. Jika hasil dari seleksi kondisi bernilai benar maka akan terjadi proses pemindahan indeks yang dilakukan pada baris ke-6 hingga baris ke-12.

5.3 Implementasi Antarmuka

Implementasi antarmuka merupakan hasil tampilan yang ada pada sistem yang berhasil dibuat yang merujuk pada perancangan antarmuka pada bab sebelumnya.

5.3.1 Halaman Awal

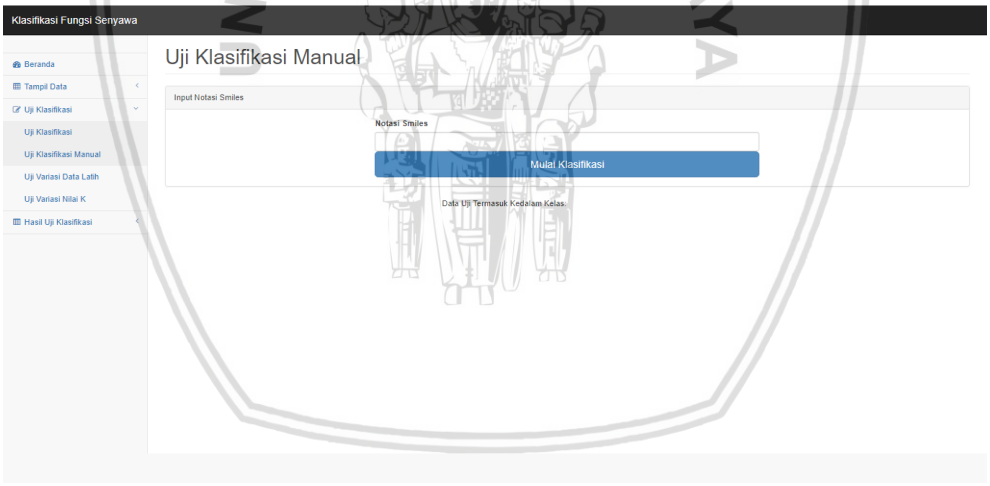
Tampilan antarmuka pada halaman awal menampilkan judul dari sistem yang dibuat, menu bar yang terdiri dari menu *home*, uji, data latih dan data uji, dan tombol mulai untuk langsung memulai proses klasifikasi. Hasil dari implementasi antarmuka halaman awal ditunjukkan pada Gambar 5.1.



Gambar 5.1 Implementasi halaman awal

5.3.2 Halaman Uji Klasifikasi

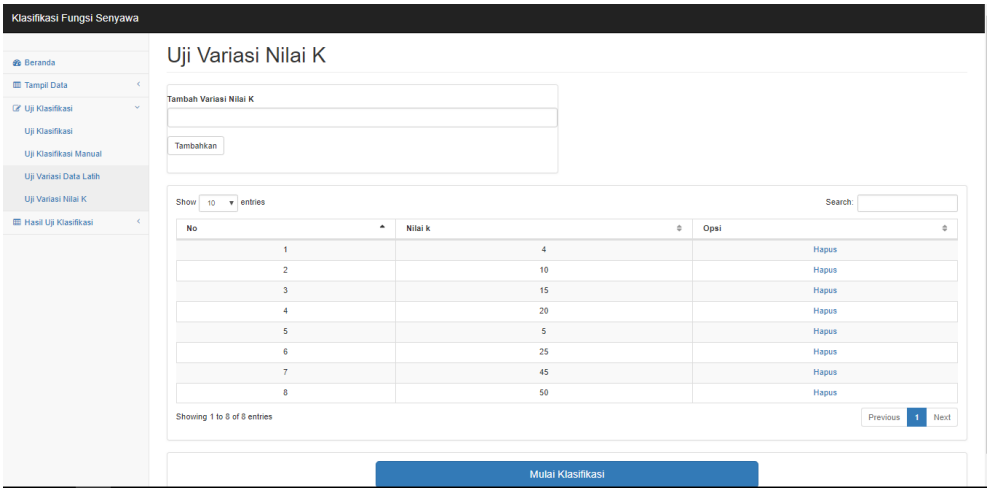
Tampilan antarmuka pada halaman uji klasifikasi menampilkan kolom masukan agar pengguna dapat memasukkan kode *smiles* yang akan digunakan dalam proses klasifikasi. Juga terdapat tombol mulai untuk memulai proses klasifikasi. Hasil dari implementasi antarmuka halaman uji klasifikasi ditunjukkan pada Gambar 5.2.



Gambar 5.2 Implementasi halaman uji klasifikasi

5.3.3 Halaman Uji Variasi Nilai K

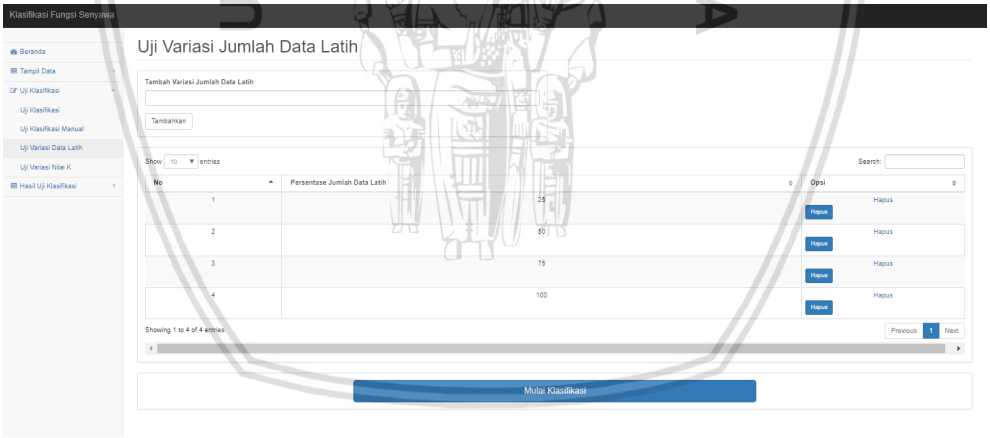
Tampilan antarmuka pada halaman uji variasi nilai *k* sama dengan halaman uji klasifikasi hanya saja terdapat satu kolom tambahan yang digunakan untuk memasukkan besaran nilai *k* yang akan digunakan dalam proses klasifikasi. Juga terdapat tambahan satu buah tombol untuk melihat hasil dari klasifikasi sebelumnya. Hasil dari implementasi antarmuka halaman uji variasi nilai *k* ditunjukkan pada Gambar 5.3.



Gambar 5.3 Implementasi halaman uji variasi nilai k

5.3.4 Halaman Uji Variasi Jumlah Data Latih

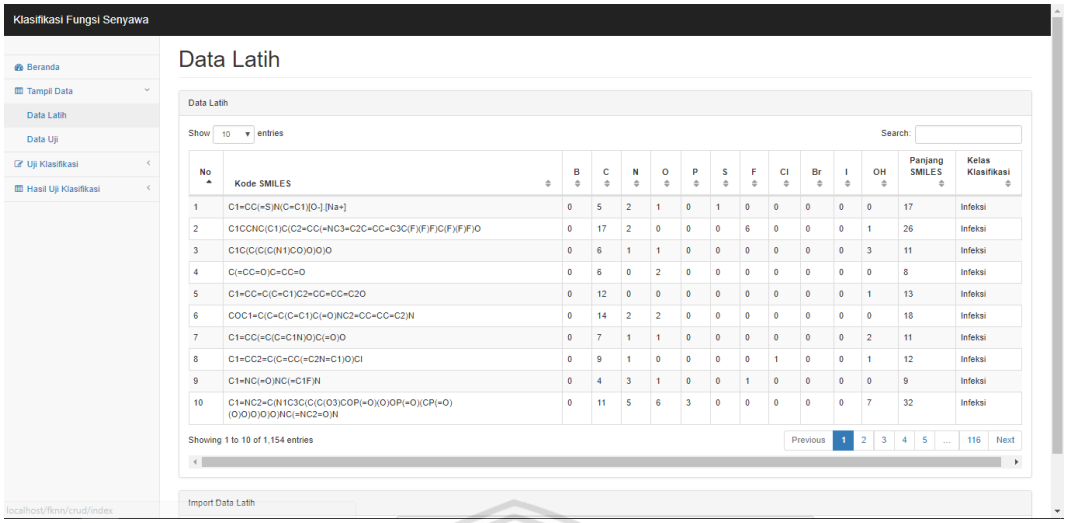
Tampilan antarmuka pada halaman uji variasi jumlah data latih sama dengan halaman uji klasifikasi hanya saja terdapat satu kolom tambahan yang digunakan untuk memasukkan besaran nilai persentase data latih yang akan digunakan dalam proses klasifikasi. Juga terdapat tambahan satu buah tombol untuk melihat hasil dari klasifikasi sebelumnya. Hasil dari implementasi antarmuka halaman uji variasi nilai *k* ditunjukkan pada Gambar 5.4.



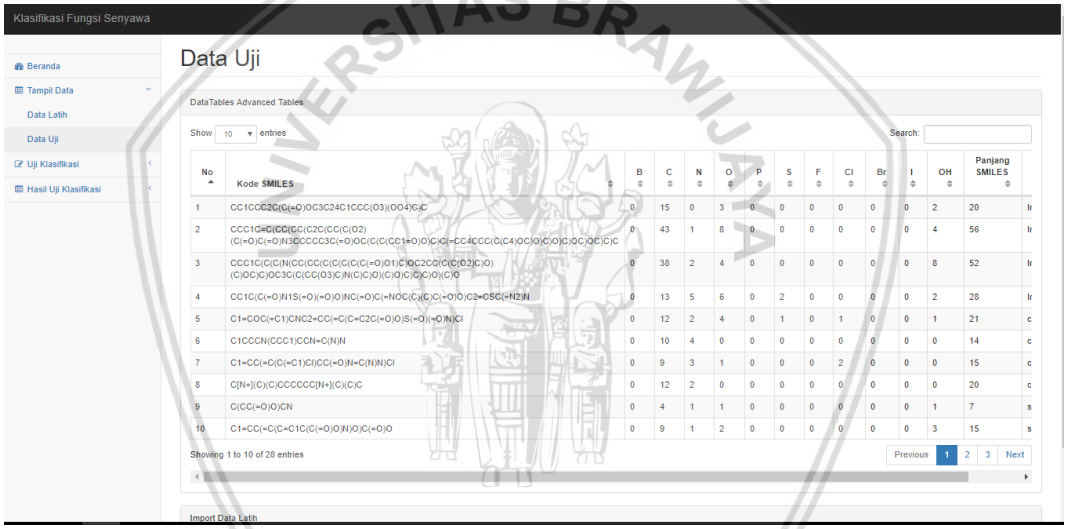
Gambar 5.4 Implementasi halaman uji variasi jumlah data latih

5.3.5 Halaman Menampilkan Data Latih dan Data Uji

Tampilan antarmuka pada halaman untuk menampilkan data latih dan data uji terdiri dari tabel yang akan menampilkan seluruh data latih dan data uji beserta fitur-fitur dan kelas klasifikasinya. Hasil dari implementasi antarmuka halaman untuk menampilkan data latih dan data uji ditunjukkan pada Gambar 5.5 dan Gambar 5.6.



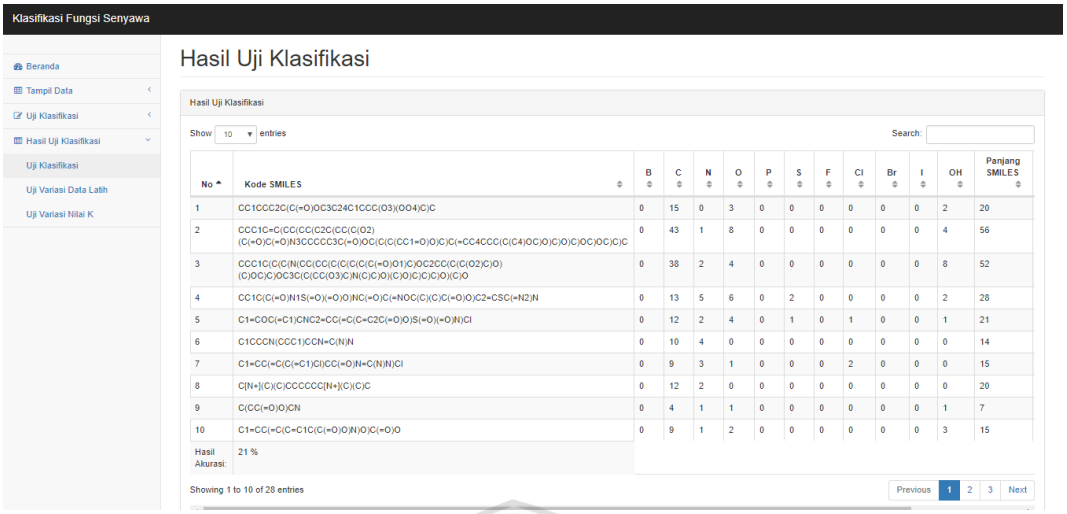
Gambar 5.5 Implementasi halaman menampilkan data latih



Gambar 5.6 Implementasi halaman menampilkan data uji

5.3.6 Halaman Menampilkan Hasil Klasifikasi

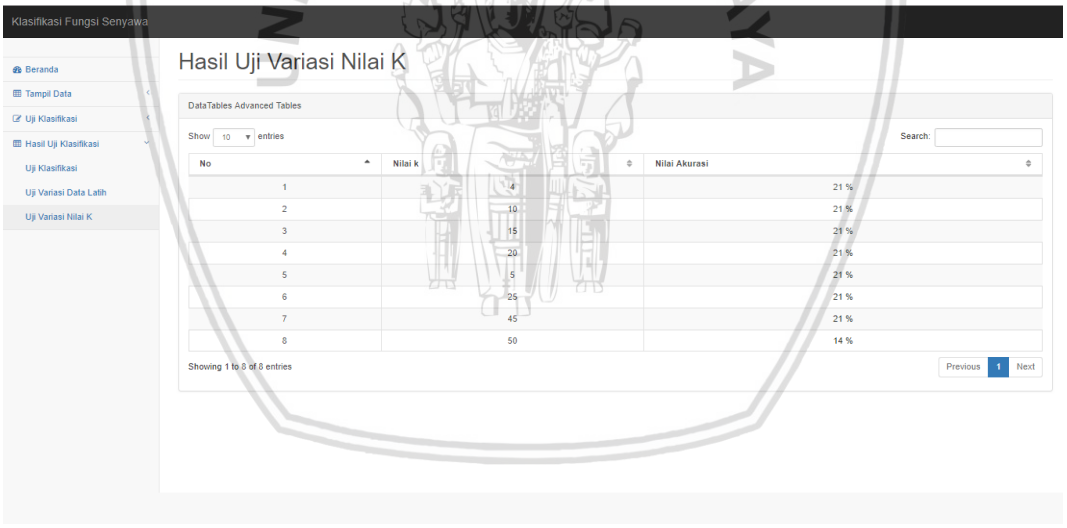
Tampilan antarmuka pada halaman untuk menampilkan hasil klasifikasi terdiri dari *text* yang akan menjelaskan senyawa tersebut termasuk kedalam salah satu kelas klasifikasi. Hasil dari implementasi antarmuka halaman uji variasi nilai *k* ditunjukkan pada Gambar 5.7.



Gambar 5.7 Implementasi halaman menampilkan hasil klasifikasi

5.3.7 Halaman Menampilkan Hasil Uji Variasi Nilai K

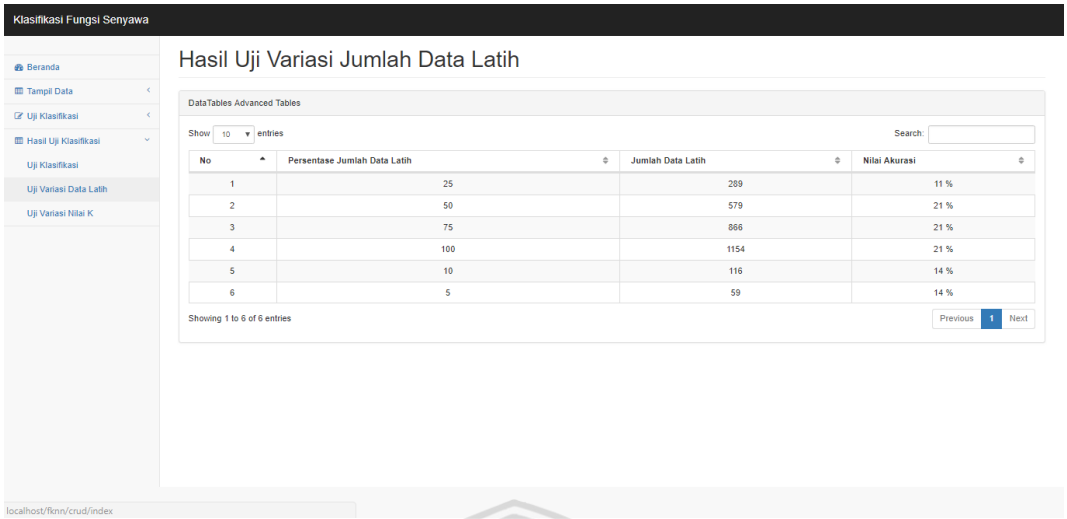
Tampilan antarmuka pada halaman hasil uji variasi nilai *k* akan menampilkan tabel yang terdiri dari nilai k yang digunakan dan nilai akurasi yang didapat dari hasil perhitungan klasifikasi. Hasil dari implementasi antarmuka halaman uji variasi nilai *k* ditunjukkan pada Gambar 5.8.



Gambar 5.8 Implementasi halaman menampilkan hasil uji variasi nilai k

5.3.8 Halaman Menampilkan Hasil Uji Variasi Jumlah Data Latih

Tampilan antarmuka pada halaman hasil uji variasi jumlah data latih akan menampilkan tabel yang terdiri dari jumlah data latih yang digunakan dan nilai akurasi yang didapat dari hasil perhitungan klasifikasi. Hasil dari implementasi antarmuka halaman uji variasi nilai *k* ditunjukkan pada Gambar 5.9.



Gambar 5.9 Implementasi halaman menampilkan hasil uji variasi jumlah data latih

BAB 6 PENGUJIAN DAN ANALISIS

Proses pengujian terhadap metode *fuzzy k-nearest neighbor* terdiri dari beberapa skenario pengujian. Pengujian pada penelitian ini terbagi menjadi beberapa skenario pengujian yaitu pengujian validasi program, pengaruh nilai *k* terhadap akurasi, pengujian pengaruh jumlah data latih terhadap akurasi dan *k-fold cross validation*.

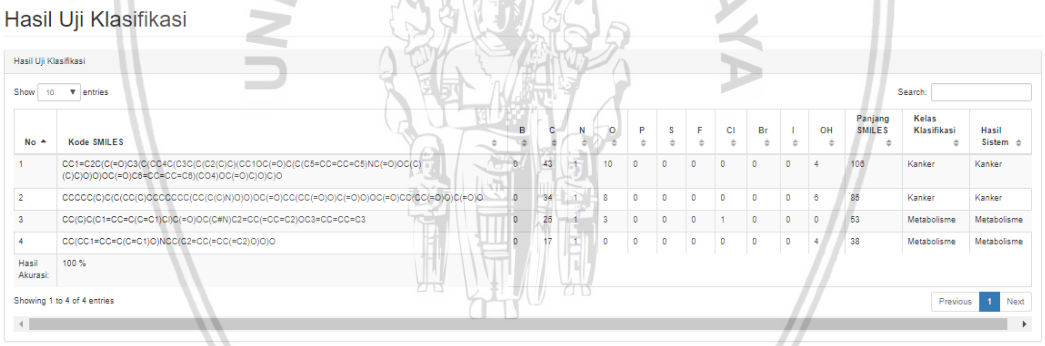
6.1 Pengujian Validitas Program

Pengujian validitas program menggunakan 30 data latih, 4 data uji yang diambil dari 2 kelas yang berbeda. Setiap kelas diambil 10 data untuk data latih dan 2 data untuk data uji. Kelas yang digunakan yaitu kelas kanker dan metabolisme. Data latih, data uji dan kelas yang digunakan pada pengujian ini sama dengan yang digunakan pada perhitungan manualisasi. Hasil dari perhitungan manualisasi dan hasil keluaran sistem ditunjukkan pada Tabel 6.1 dan Gambar 6.1.

Tabel 6.1 Hasil perhitungan manualisasi

No	Kode SMILES	Fitur											Klasifikasi	Hasil Klasifikasi
		Jumlah Atom										Panjang Smiles		
		B	C	N	O	P	S	F	Cl	Br	I	OH		
1	CC1=C2C(C(=O)C3(C(CC4C(C3C(C(C2(C)C)(CC1OC(=O)C(C(C5=CC=CC=C5)NC(=O)OC(C)(C)C)O)O)OC(=O)C6=CC=CC=C6)(CO4)OC(=O)C)O)C)O	0	43	1	#	0	0	0	0	0	0	4	106	Kanker	Kanker
2	CCCCC(C)C(C(CC(C)CCCCCCC(CC(C(C)N)O)O)OC(=O)CC(CC(=O)O)C(=	0	34	1	8	0	0	0	0	0	0	6	85	Kanker	Kanker

	O)O)OC(=O)CC(CC(=O)O)C(=O)O														
3	CC(C)C(C1=CC=C(C=C1)Cl)C(=O)OC(C#N)C2=CC(=CC=C2)OC3=CC=CC=C3	0	25	1	3	0	0	0	1	0	0	0	53	Metaboli sme	Meta bolis me
4	CC(CC1=CC=C(C=C1)O)NCC(C2=CC(=CC(=C2)O)O)O	0	17	1	0	0	0	0	0	0	0	4	38	Metaboli sme	Meta bolis me
akurasi														100	%



Gambar 6.1 Hasil keluaran program

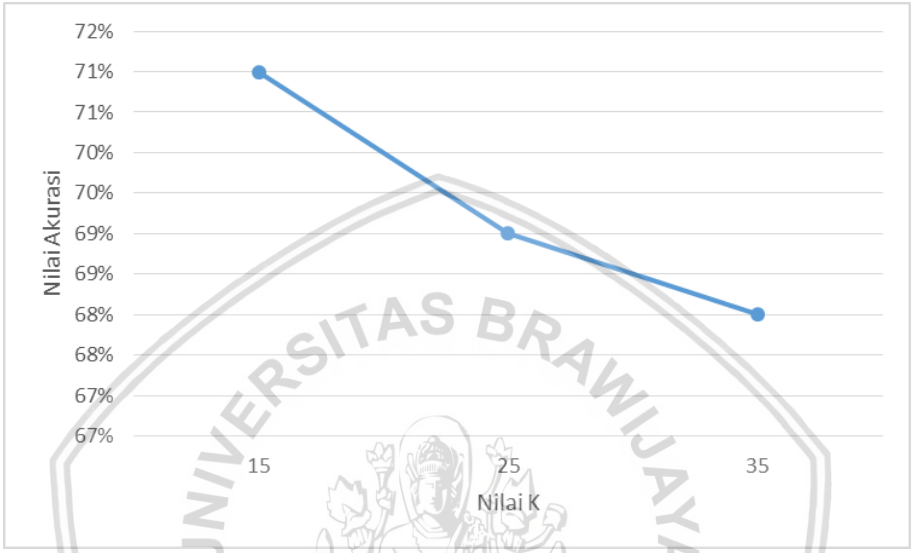
Pada Tabel 6.1 dan Gambar 6.1 menunjukkan hasil keluaran dan nilai akurasi yang sama. Berdasarkan hasil tersebut dapat dihasilkan bahwa program dapat dikatakan valid.

6.2 Pengujian Pengaruh Nilai K Terhadap Akurasi

Pengujian pengaruh nilai *k* terhadap akurasi menggunakan nilai *k* yang berbeda. Pada pengujian ini dilakukan 3 kali pengujian dengan menggunakan nilai *k* sebesar *k* = 15, *k* = 25 dan *k* =35. Tujuan dari penelitian ini adalah untuk mengetahui pengaruh banyaknya nilai tetangga terdekat terhadap nilai akurasi. Hasil dari pengujian ditunjukkan pada Tabel 6.2 dan Gambar 6.2.

Tabel 6.2 Hasil pengujian variasi nilai k

Skenario Ke-	Nilai *k*	Nilai akurasi
1	15	71%
2	25	69%
3	35	68%



Gambar 6.2 Grafik hasil pengujian variasi nilai *k*

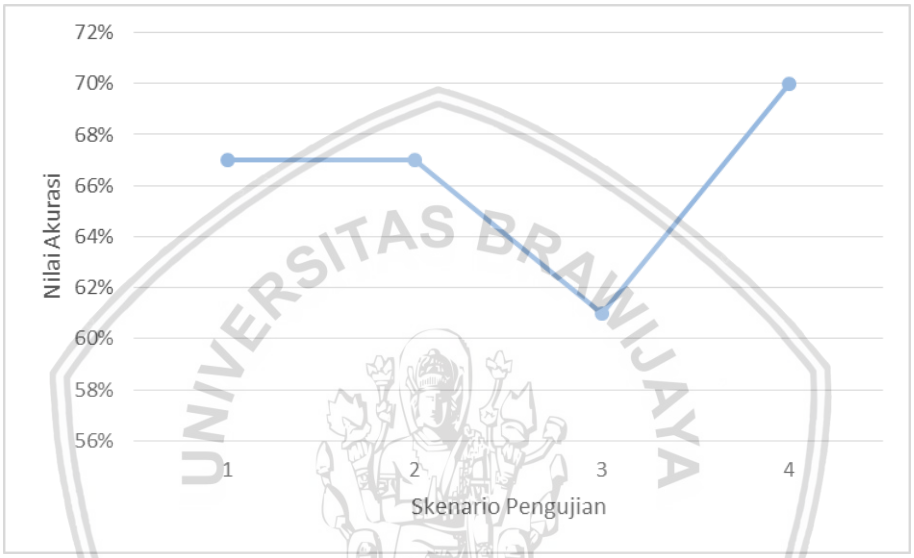
Dari hasil pengujian variasi nilai *k* yang ditunjukkan pada Tabel 6.2 dapat diketahui bahwa akurasi terbesar diraih pada nilai k = 15 yaitu sebesar 71%. Setiap pengujian dengan menggunakan nilai *k* yang berbeda menghasilkan nilai akurasi yang berbeda. Kesimpulan dari pengujian ini adalah nilai *k* dapat mempengaruhi besarnya nilai akurasi pada pengujian. Besarnya nilai *k* berbanding terbalik dengan besarnya nilai akurasi. Hal ini dikarenakan jika nilai *k* semakin besar maka semakin banyak data yang tidak relevan diikutkan dalam pengambilan keputusan hasil klasifikasi sehingga nilai akurasi menurun.

6.3 Pengujian Pengaruh Jumlah Data Latih Terhadap Akurasi

Pengujian pengaruh jumlah data latih terhadap akurasi digunakan untuk mengetahui apakah jumlah data latih dapat mempengaruhi nilai akurasi. Pada pengujian ini menggunakan persentase jumlah data latih yang akan digunakan karena jumlah data latih pada setiap kelas memiliki jumlah yang berbeda-beda. Skenario pengujian pada pengujian variasi jumlah data latih terdapat tiga kali pengujian dengan persentase penggunaan jumlah data latih yang berbeda-beda yaitu 25%, 50%, 75% dan 100%. Hasil dari pengujian variasi jumlah data latih dapat dilihat pada Tabel 6.3 dan Gambar 6.3.

Tabel 6.3 Hasil pengujian variasi jumlah data latih

Skenario Ke-	Persentase Jumlah Data Latih	Jumlah Data Latih	Nilai akurasi
1	25	126	67%
2	50	252	67%
3	75	378	69%
4	100	503	70%



Gambar 6.3 Grafik hasil pengujian variasi jumlah data latih

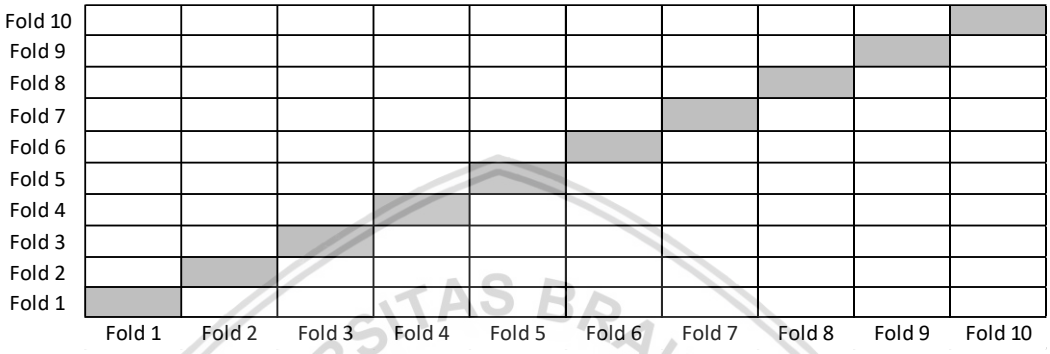
Pada hasil pengujian variasi jumlah data latih menunjukkan bahwa pada pengujian keempat dengan menggunakan seluruh jumlah data latih memberikan nilai akurasi sebesar 70%. Nilai akurasi yang dihasilkan dari pengujian keempat masih lebih baik daripada pengujian lainnya yang masing-masing menggunakan data latih sebanyak 25%, 50% dan 75% dari seluruh jumlah data latih. Nilai akurasi yang dihasilkan dari percobaan pertama dan kedua yaitu sebesar 67%, kemudian percobaan ketiga sebesar 69%. Berdasarkan penjelasan tersebut dapat disimpulkan bahwa jika semakin banyak jumlah data latih yang digunakan pada proses klasifikasi maka nilai akurasi akan semakin baik karena sistem akan melakukan proses pembelajaran lebih banyak sehingga meminimalisir kesalahan pada proses klasifikasi.

6.4 Pengujian *K-Fold Cross Validation*

K-Fold Cross Validation adalah metode yang digunakan untuk mengetahui tingkat keberhasilan dalam perancangan sistem klasifikasi. Pengujian ini dilakukan dengan melakukan pembagian dataset sebanyak *k*, kemudian setiap *k* akan dibagi menjadi data latih dan data uji secara bergantian. Proses pengujian *k-fold cross validation* dilakukan berulang sebanyak *k-fold* nya. Tujuan dari pengujian ini

adalah untuk mengetahui apakah sistem dapat diuji dengan menggunakan atribut acak.

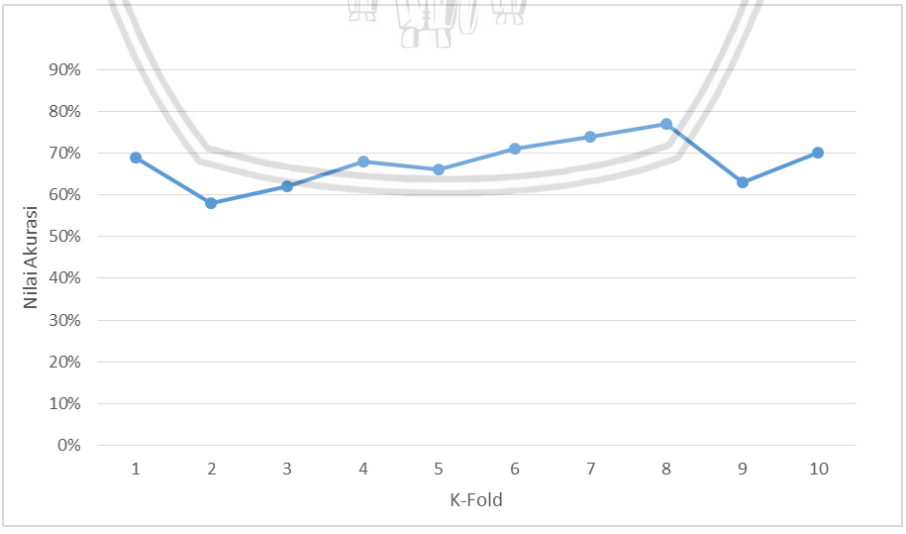
Pada pengujian pada penelitian ini dataset dibagi menjadi 10 *k* atau biasa disebut dengan 10 fold. Skenario pengujian dari metode *k-fold cross validation* adalah dengan melakukan pengujian sebanyak *k* dengan mengganti tugas dari masing-masing *k*. Pembagian dataset sebanyak *k fold* ditunjukkan pada Gambar 6.4 dan hasil dari pengujian metode *k-fold cross validation* ditunjukkan pada Tabel 6.4 dan Gambar 6.5.



Gambar 6.4 Pembagian *dataset* sebanyak *k-fold*

Tabel 6.4 Hasil pengujian menggunakan metode *k-fold cross validation*

Akurasi	K-Fold									
	1	2	3	4	5	6	7	8	9	10
	69%	58%	62%	68%	66%	71%	74%	77%	63%	70%
Rata-Rata:									68%	
Standar Deviasi									5,73	



Gambar 6.5 Grafik hasil pengujian akurasi menggunakan metode *k-fold cross validation*

Seperti ditunjukkan pada Gambar 6.4 kolom yang berwarna gelap merupakan fold yang digunakan sebagai data uji, dan sisanya sebagai data latih. Hasil dari pengujian tersebut menghasilkan akurasi tertinggi sebesar 77% pada fold ke 8 dan terendah sebesar 58% pada fold ke 2. Sedangkan rata-rata dari akurasi pengujian menggunakan metode *k-fold cross validation* adalah sebesar 68%. Berdasarkan penjelasan tersebut dapat disimpulkan bahwa penggunaan data yang acak pada klasifikasi memberikan hasil yang berbeda dikarenakan setiap fold memiliki karakteristik data yang berbeda sehingga menghasilkan nilai akurasi yang berbeda.

BAB 7 PENUTUP

Bab ini akan menjelaskan mengenai kesimpulan dan saran berdasarkan penelitian ini meliputi perancangan, implementasi dan pengujian yang telah dilakukan.

7.1 Kesimpulan

Kesimpulan yang diambil berdasarkan penelitian mengenai penerapan metode *fuzzy k-nearest neighbor* dalam proses klasifikasi fungsi senyawa menggunakan kode *SMILES* adalah:

1. Cara kerja dari metode *fuzzy k-nearest neighbor* yaitu menghitung selisih nilai dari data uji dan data latih atau disebut *euclidean distance*. Kemudian nilai *euclidean distance* diurutkan dari nilai terendah ke nilai tertinggi. Setelah nilai *euclidean distance* diurutkan data diambil sebanyak *k* dan dihitung nilai *fuzzy* setiap kelas. Kelas dengan nilai *fuzzy* tertinggi ditetapkan sebagai hasil klasifikasi. Dalam proses klasifikasi data yang digunakan berupa notasi SMILES. Notasi *SMILES* memiliki fitur berupa atom *B,C,N,O,P,S,F,Cl,Br,I,OH* dan panjang dari notasi *SMILES*. Kelas klasifikasi yang digunakan pada penelitian ini yaitu kanker dan metabolisme.
2. Pada penelitian ini metode *fuzzy k-nearest neighbor* memberikan hasil akurasi yang cukup baik dalam melakukan proses klasifikasi fungsi senyawa menggunakan kode *SMILES*. Setelah melakukan beberapa pengujian hasil yang didapatkan adalah sebagai berikut:
    a. Pengujian validasi program untuk memastikan hasil perhitungan manualisasi dan hasil keluaran sistem memberikan hasil yang valid.
    b. Pengujian *k-fold cross validation* sebanyak 10 kali percobaan menghasilkan nilai akurasi tertinggi sebesar 77% dengan rata-rata akurasi dari seluruh percobaan sebesar 68%.
    c. Pengujian variasi nilai *k* menghasilkan nilai akurasi tertinggi sebesar 71% dengan nilai *k* = 15.
    d. Pengujian variasi jumlah data latih menghasilkan nilai akurasi tertinggi sebesar 70% dengan menggunakan seluruh data latih.

7.2 Saran

Untuk pengembangan penelitian selanjutnya saran yang dapat diambil berdasarkan penelitian ini adalah:

1. Untuk meningkatkan akurasi pada sistem diharapkan pada penelitian selanjutnya untuk menggunakan data latih dengan jumlah yang lebih banyak.

2. Menggunakan nilai *k* yang tepat karena nilai *k* pada proses *Fuzzy K-Nearest Neighbor* memiliki peran yang sangat penting dan berpengaruh pada hasil akurasi.

DAFTAR PUSTAKA

Anhar, 2010. *Panduan Menguasai PHP & MySQL Secara Otodidak.* Jakarta Selatan: Mediakita.

Cahyaningtyas, Y., 2013. *Penerapan Fuzzy K-Nearest Neighbor Untuk Menentukan Status Evaluasi Kinerja Karyawan,* S1: Universitas Brawijaya.

Junaedi, H., 2011. *Penggambaran Rantai Karbon Dengan Menggunakan Simplified Molecular Input Line System (SMILES).* Surabaya, Sekolah Tinggi Teknik Surabaya.

Kadir, A., 2008. *Belajar Database Menggunakan MySQL.* Yogyakarta, Andi Publisher.

Kusnawi, 2007. *Pengantar Solusi Data Mining.* Yogyakarta, Seminar Nasional Teknologi.

Kusumadewi, S., 2003. *Artificial Intelligence (Teknik dan Aplikasinya).* Yogyakarta, Graha Ilmu.

Larose, D. T., 2005. *Discovering Knowledge in Data.* New Jersey, Wiley-Interscience.

Manning, C. D., Raghavan, P. & Schutze, H., 2008. *Introduction to Information Retrieval.* Cambridge: Cambridge University Press.

Marisa, H., Mukti, R. W. & Salni, 2011. Isolasi Senyawa Antibakteri Dari Daun Jengkol (Pithecolobium lobatum Benth) dan Penentuan Nilai KHM-nya. *Jurnal Penelitian Sains,* Volume 14, pp. 38-41.

Massalli, H., 2016. *Penerapan Fuzzy K-Nearest Neighbor (FK-NN) Untuk Mengklasifikasi Penyakit Jantung Berdasarkan SPECTF Heart Dataset,* S1: Universitas Brawijaya.

Muliantara, A., 2009. *Penerapan Regular Expression Dalam Melindungi Alamat Email Dari Spam Robot Pada Konten Wordpress.* Denpasar: Universitas Udayana.

Prasetyo, E., 2012. *Data Mining: Konsep Dan Aplikasi Menggunakan Matlab.* Yogyakarta, Andi Offset.

Prasetyo, E., 2012. *Fuzzy K-Nearest Neighbor In Every Class Untuk Klasifikasi Data.* Surabaya, Universitas Pembangunan Nasional Veteran, pp. 57-60.

Radam, R. R. & Purnamasari, E., 2016. Uji Fitokimia Senyawa Kimia Aktif Akar Nipah (Nyfa Fruticans WURMB) Sebagai Tumbuhan Obat Di Kalimantan Selatan. *Jurnal Hutan Tropis,* Volume 4, pp. 28-34.

Robiansyah, C., 2016. *Klasifikasi Dokumen Twitter Menggunakan Metode Fuzzy K-Nearest Neighbor (FK-NN) Dengan Term Weighting Berbasis Indeks Dokumen dan Kelas,* S1: Universitas Brawijaya.

Suharto, M. A. P. S., Edy, H. J. & Dumanauw, J. M., 2012. ISOLASI DAN IDENTIFIKASI SENYAWA SAPONIN DARI EKSTRAK METANOL BATANG PISANG AMBON(Musa paradisiaca var. sapientum L.). *Jurnal Ilmiah Pharmacon,* Volume Vol. 1, pp. 86-92.

Syahiwa, N., 2016. *Implementasi Fuzzy K-Nearest Neighbor Untuk DIagnosis Penyakit Tiroid,* S1: Universitas Brawijaya.

Turban, E., Aronson, J. E. & Liang, T.-P., 2007. *Decision Support Systems And Intelligent System.* 7th Edition ed. New Delhi: Prentice'Hall of India.

Wisdarianto, A., Ridok, A. & Rahman, M. A., 2013. Penerapan Metode Fuzzy K-Nearest Neighbor (FK-NN) Untuk Pengklasifikasian Spam Email. *Jurnal Teknologi Informasi dan Ilmu Komputer Program Studi Ilmu Komputer Universitas Brawijaya,* Volume 1 - Number 6.