KLASIFIKASI FUNGSI SENYAWA AKTIF DATA BERDASARKAN KODE *SIMPLIFIED MOLECULAR INPUT LINE ENTRY SYSTEM* (*SMILES*) MENGGUNAKAN METODE *MODIFIED K-NEAREST NEIGHBOR*

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:
Yunita Dwi Alfiyanti
NIM: 145150201111092

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

PENGESAHAN

PENGESAHAN

KLASIFIKASI FUNGSI SENYAWA AKTIF DATA BERDASARKAN KODE *SIMPLIFIED MOLECULAR INPUT LINE ENTRY SYSTEM* (*SMILES*) MENGGUNAKAN METODE *MODIFIED K-NEAREST NEIGHBOR*

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:
Yunita Dwi Alfiyanti
NIM: 145150201111092

Skripsi ini telah diuji dan dinyatakan lulus pada
19 Desember 2018

Telah diperiksa dan disetujui oleh :

Dosen Pembimbing I

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

Dosen Pembimbing II

Syaiful Anam, S.Si, M.T,Ph.D
NIP. 19780115 200212 1 003

Mengetahui
Ketua Jurusan Teknik Informatika

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

ii

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 perundangundangan yang berlaku (UU No.20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70)

Malang, 18 Desember 2018

Yunita Dwi ALfiyanti

NIM: 145150201111092

PRAKATA

Puji syukur kehadirat Allah SWT yang selalu melimpahkan rahmat dan karunia-Nya kepada penulis sehingga penulis dapat menyelesaikan skripsi dengan judul "Klasifikasi Fungsi Senyawa Aktif Data berdasarkan Kode *Simplified Molecular Input Line Entry Sistem* (*SMILES*) menggunakan Metode *Modified K-Nearest Neighbor*" 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. Allah SWT yang Maha Pemurah yang telah memberikan seluruh cinta kasihnya sehingga penulis dapat menyelesaikan skripsi dengan sebaik mungkin
2. Almarhum kakek dan almarhumah nenek yang telah mendukung selama penulis menjalani perkuliahan di Universitas Brawijaya Malang yang tidak sempat melihat kelulusan penulis, semoga tenang di sisi-Nya.
3. Ayah tercinta Khoirul Anam dan Mama tercinta Alfi Syayyidah yang selalu memberikan segenap kasih sayang, tenaga, do'a dan dukungan materi maupun mental yang tak terhingga kepada penulis dalam menyelesaikan skripsi ini.
4. Kakak dan adik Tercinta, Alfian Amiruddin dan Andrian Maulana yang selalu memberikan semangat, doa dan dukungan kepada penulis dalam menyelesaikan skripsi ini.
5. Keluarga besar di Surabaya dan Sidoarjo yang selalu mendoakan dan memberikan semangat dalam menyelesaikan skripsi ini.
6. 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.
7. 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.
8. Bapak dan Ibu dosen Fakultas Ilmu Komputer Universitas Brawijaya yang telah memberikan ilmu selama penulis melaksanakan kegiatan perkuliahan di kampus tercinta ini.
9. Staff dan karyawan Fakultas Ilmu Komputer yang telah membantu proses selama perkuliahan dan penulisan skripsi ini.
10. Bapak Andika yang telah memberikan nasihat dan dukungan untuk penulis mengambil kuliah di Jurusan Informatika.
11. Teman-teman seperjuangan dalam menyusun skripsi Suhhy Ramzini, Muhammad Iskandar A.R, Nur Khilmiyatul, Sherly Witanto, Nyimas Ayu Widi Indriana, Raden Rizky Widdie Tigusti yang telah saling membantu dan berdiskusi dalam penyelesaian skripsi ini.

12. Hamim Fathul Aziz yang sudah banyak meluangkan waktu untuk membantu dalam penyelesaian skripsi ini
13. Keluarga baru selama di perantauan Novita Krisma Diarti yang selalu memberikan semangat, menemani dan membantu untuk penyelesaian skripsi ini.
14. Sahabat SMP Dwi Kartika Sari beserta Keluarga yang sangat mendukung keberhasilan penyelesaian skripsi ini.
15. Sahabat-sahabat Perempuan Ceria Andini Aprilia Wardani, Dewi Rachmawati Saputri, Dian Arimurti, dan Teesa Wijayanti yang menemani sejak SMA dan memberikan dukungan dan motivasi yang luar biasa agar penulis dapat cepat lulus.
16. Sahabat-sahabat kecil Sayyidatul Eka Putri Rosalinda, Maulida Alfi Chabbah, Nur Fitriyah yang telah selalu mendengarkan keluh kesah penulis dalam penyelesaian skripsi ini.
17. Teman-teman PK2MABA 2016 khususnya Ali Hafidz, M. Fakhruddin Farid, Tahtri Nadia Utami, M.Irfanul Hadi, Dwi Wahyu Puji Lestari, Muliyahati Sutejo, Tafarrara Irsa dan semua staff yang tidak bisa penulis sebutkan satu persatu yang telah memberikan berbagai pengalaman selama berpanitia bersama di Fakultas Ilmu Komputer Universitas Brawijaya.
18. Teman-teman Panitia PK2MABA 2015, Seminar Profesi IT 2016, PEMILWA 2016, Panitia Bina Desa 2016 yang tidak bisa penulis sebutkan namanya satu persatu, terimakasih telah memberikan berbagai pengalaman baru selama kepanitian di Fakultas Ilmu Komputer Universitas Brawijaya.
19. Teman-teman Informatika FILKOM UB 2013, 2014, dan 2015 yang mengenal dan dikenal penulis yang tidak bisa penulis sebutkan satu per satu yang telah memberikan kesan dan pesan selama penulis menempuh perkuliahan di Fakultas Ilmu Komputer Universitas Brawijaya.
20. Keluarga Bapak dan Ibu kos yang selalu menjaga dan memberikan dukungan kepada penulis selama menempati kamar kosan di Malang.
21. Serta semua pihak 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, 18 Desember 2018

Yunita Dwi Alfiyanti

Yunitadwi64@gmail.com

ABSTRAK

Yunita Dwi Alfiyanti. 2018. Klasifikasi Fungsi Senyawa Aktif Data Berdasarkan Kode *Simplified Molecular Input Line Entry System* (*SMILES*) Menggunakan Metode *Modified K-Nearest Neighbor*. Skripsi Program Informatika / Ilmu Komputer, Fakultas Ilmu Komputer, Universitas Brawijaya

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

Senyawa merupakan zat tunggal kimia dari dua atau lebih unsur kimia yang membentuk ikatan dan dapat diuraikan. Senyawa dibagi menjadi senyawa aktif dan senyawa tidak aktif. Senyawa aktif adalah senyawa kimia yang memiliki banyak fungsi. Senyawa memiliki susunan yang sulit diolah pada komputer, untuk itu diciptakan kode yang mudah untuk diproses menggunakan komputer. Kode tersebut adalah *SMILES* (*Simplified Molecular Input Line Entry System*) yang merupakan notasi kimia modern yang dapat disimpan pada variabel string sehingga memudahkan proses klasifikasi pada sistem. Karakteristik pada *SMILES* didapat dengan melakukan *preprocessing* dengan hasil berupa 11 fitur yang terdiri dari atom B, C, N, O, P, S, F, Cl, Br, I dan OH. Fitur-fitur tersebut kemudian digunakan untuk proses klasifikasi menggunakan metode *Modified K-Nearest Neighbor* yang merupakan pengembangan performansi dari metode KNN yang terdiri dari dua pemrosesan, validasi data latih dan pembobotan. Klasifikasi fungsi senyawa aktif bertujuan untuk mempermudah pengelompokkan senyawa aktif berdasarkan farmakologinya melalui bantuan teknologi informasi dan perosesan ilmu komputer, dimana selama ini pada bidang kedokteran memerlukan waktu yang lama dalam penentuannya karena menggunakan tes laboratorium. Pengujian pada penelitian ini menggunakan dataset sebanyak 260 yang terbagi menjadi 2 kelas kategori yaitu kelas Saraf dan kelas Jantung yang terdiri dari data latih dan data uji sebesar 90% (234 data) dan 10% (26 data). Hasil dari pengujian didapatkan nilai akurasi sebesar 73% dengan nilai k sebesar 3, sedangkan pada pengujian *k-fold cross validation* nilai akurasi didapatkan rata-rata sebesar 62,69%.

Kata kunci: *Senyawa Aktif, SMILES, Modified K-Nearest Neighbor*

ABSTRACT

Yunita Dwi Alfiyanti. 2018. Klasifikasi Fungsi Senyawa Aktif Data Berdasarkan Kode *Simplified Molecular Input Line Entry System* (*SMILES*) Menggunakan Metode *Modified K-Nearest Neighbor*. Skripsi Program Informatika / Ilmu Komputer, Fakultas Ilmu Komputer, Universitas Brawijaya

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

Compound is a single chemical substance from two or more chemical elements that make up bonds and can be decomposed. The compound is divided into active compounds and inactive compounds. Active compounds are chemical compounds that have many functions. Compounds have an arrangement that is difficult to process on a computer, for which code is made easy to make using a computer. The code is SMILES (Simple Input System for Molecular Input) which is a modern chemical notation that can be stored in string variables to facilitate the classification process in the system. The characteristics of the SMILES are made by doing preprocessing with the results consisting of 11 features consisting of B, C, N, O, P, S, F, Cl, Br, I and OH atoms. These features are then used for the classification process using the Modified K-Nearest Neighbor method which is the performance development of the KNN method which consists of two developments, training data validation and weighting. Classification of the active compilation function for active complement compilation based on its pharmacology through the help of information technology and computer science degeneration, where so far in the medical field it takes a long time in its determination to use laboratory tests. The test in this study used a dataset of 260 which were divided into 2 categories of classes, namely the Neural class and the Heart class which consist of training data and test data of 90% (234 data) and 10% (26 data). The results of the test were obtained a test value of 73% with a k value of 3, while in the k-fold test cross validation the value of accuracy obtained an average of 62.69%.

Keywords: *Active Compounds, SMILES, Modified K-Nearest Neighbor*

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

BAB 1 PENDAHULUAN

1.1 Latar Belakang

Bioinformatika adalah salah satu cabang baru ilmu biologi yang merupakan perpaduan antara biologi dan teknologi informasi. Bioinformatika merupakan ilmu multidisipliner yang memadukan pendekatan biologi molekuler dan teknik informatika (Searls, 2012). Pembelajaran Bioinformatika mulai berkembang sebagai akibat dari kemajuan berbagai metode baru yang menghasilkan data besar dan didukung oleh teknologi penyimpanan, manajemen, dan pertukaran data melalui media komputer. Perwujudan hal ini dapat dilaukukan dengan adanya data-data yang yang menjadi kunci penentu tingkah polah gejala alam tersebut, yaitu gen yang meliputi DNA (*Deoxyribose-Nucleic Acid)* atau RNA (*Ribose Nucleic Acid*).

DNA atau RNA dalam ilmu biologi tersusun atas berbagai macam atom, molekul dan ion. Beberapa susunan molekul dan atom membentuk senyawa. Senyawa merupakan zat tunggal kimia yang dapat membentuk ikatan dan dapat diuraikan. Senyawa tersebut dapat dikategorikan menjadi senyawa aktif dan senyawa tidak aktif. Dalam studinya, senyawa aktif adalah senyawa kimia tertentu yang terdapat dalam tumbuhan dan hewan sebagai bahan obat yang mempunyai efek fisiologis terhadap organisme lain, atau sering disebut sebagai senyawa bioaktif (Salni, Marisa, & Mukti, 2011). Senyawa tidak aktif mempunyai peran sebagai zat tambahan atau pengikat.

Bagi orang awam, mengetahui kegunaan dari senyawa-senyawa tersebut adalah hal yang sulit, tetapi orang dengan latar belakang ilmu pendidikan pengetahuan (kimia dan Biologi) dapat mengerti kegunaan dari senyawa tersebut. Dari permasalahan diatas beberapa peneliti menemukan cara untuk dapat mengkonversi senyawa tersebut kedalam bentuk yang mudah untuk diproses menggunakan komputer. Kode tersebut diberi nama *SMILES* (*Simplified Molecular Input Line Entry System*). Kode ini mengkonversi senyawa dalam bentuk notasi menggunakan karakter-karakter *ASCII* (*American Standart Code for Information Interchange*) untuk menggambarkan senyawa kimia sehingga mampu mempermudah orang awam dalam mengenali senyawa kimia (Weininger, 1987).

Data-data senyawa kimia tersebut dapat mendukung bidang kedokteran, salah satunya dalam pengelompokkan senyawa yang memiliki farmakologi (sebagai obat suatu penyakit) tertentu. Data berupa kode *SMILES* tersebut dapat diolah terlebih dahulu melalui proses *preprocessing* untuk mendapatkan beberapa fitur yang dibutuhkan dalam pengelolaan sistem klasifikasi kelas anti penyakit tertentu tanpa melalui tes laboratorium, namun menggunakan bantuan dari teknologi informasi dan pemrosesan ilmu komputer. Pengelolaan atau pemrosesan data dengan metode yang baik dapat mengurangi pekerjaan

sehingga dapat mengefisiensi waktu maupun tenaga. Melihat besarnya jumlah data yang diproses, tidak mudah dalam mengelola data secara manual dan penggunaan metode yang kurang efisien, maka dari perkembangan IT (*Information Technology*) dan penggunaan langkah-langkah metode yang baik dapat memberi peluang untuk mempermudah pengelolaan data.

Sistem klasifikasi senyawa aktif dapat di implementasikan menggunakan metode yang dikembangkan dalam studi data mining. Salah satu metode data mining yang populer digunakan untuk klasifikasi adalah *K-Nearest Neighbor* (KNN). Algoritma ini mengklasifikasikan suatu objek baru berdasarkan kedekatan jarak suatu data dengan data yang lain. Algortima KNN memiliki kelemahan, salah satunya yaitu kelas objek baru ditentukan berdasarkan voting mayoritas kelas pada K jarak terdekat. Berdasarkan kelemahan tersebut, solusi untuk memperbaiki kinerja dari algoritma KNN dalam melakukan klasifikasi dilakukan beberapa modifikasi pada algoritma KNN. Modifikasi algoritma KNN yang telah diperkenalkan adalah algoritma *Modified K-Nearest Neighbor* (MKNN). Algoritma MKNN merupakan pengembangan performansi dari metode KNN yang terdiri dari dua pemrosesan, pertama validasi data latih dan yang kedua adalah menerapkan pembobotan KNN (Parvin, Alizadeh, & Minaei-Bidgoli, 2008). Nilai K sangat berpengaruh terhadap hasil keakurasian data. Tingkat akurasi MKNN terbukti lebih baik jika dibandingkan dengan metode sebelumnya yaitu KNN.

Mengkaji dari beberapa penelitian terdahulu, metode klasifikasi dengan tingkat akurasi baik yang sering digunakan adalah metode *K-Nearest Neighbor* yang selanjutnya dimodifikasi. Dari penelitian yang dilakukan oleh Leidiyana (2013) dengan objek data konsumen yang menggunakan jasa keuangan kredit kendaraan bermotor didapatkan bahwa Hasil testing untuk mengukur performa algoritma ini menggunakan metode Cross Validation, Confusion Matrix dan Kurva ROC dan menghasilkan akurasi dan nilai AUC berturut-turut 81,46 % dan 0,984 dan masih berada ditingkat baik. Begitu pula dengan penelitian yang dilakukan oleh Wafiyah, Hidayat, & Perdana (2017) menemukan bahwa berdasarkan hasil pengujian terhadap perubahan nilai K, perubahan jumlah data latih dan perubahan komposisi data latih didapatkan rata-rata akurasi untuk pengujian pengaruh nilai K terhadap akurasi sebesar 88.55%. Penelitian ini diterapkan untuk klasifikasi penyakit demam dengan mempelajari pola dari data hasil pemeriksaan sebelumnya berdasarkan 15 gejala penyakit.

Berdasarkan uraian di atas, peneliti mencoba memberikan penyelesaian klasifikasi senyawa aktif dengan menerapkan metode dalam penelitian terkait. Penelitian ini memberikan fitur dalam pengolahan data *SMILES* menggunakan metode MKNN untuk mengklasifikasikan jenis farmakologi suatu senyawa dengan mudah dan sistem yang efisien.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dipaparkan, maka dapat diambil sebuah rumusan masalah sebagai berikut:

1. Bagaimana melakukan *preprocessing* pada data *SMILES* (*Simplified Molecular Input Line Entry System*)?
2. Bagaimana mengimplementasikan metode *Modified K- Nearest Neighbor* untuk klasifikasi dalam menentukan fungsi senyawa aktif?
3. Berapa tingkat akurasi dari perhitungan klasifikasi senyawa aktif menggunakan metode *Modified K- Nearest Neighbor*?

1.3 Tujuan

Tujuan adanya penelitian ini yaitu:

1. Melakukan *preprocessing* pada data senyawa dalam notasi *SMILES* (*Simplified Molecular Input Line Entry System*)
2. Mengimplementasikan metode *Modified K- Nearest Neighbor* untuk klasifikasi fungsi senyawa aktif.
3. Mengetahui tingkat akurasi dari perhitungan klasifikasi senyawa aktif menggunakan metode M*odified K- Nearest Neighbor*.

1.4 Manfaat

1.4.1 Manfaat Teoritis

Bagi penulis, pemilik instansi ataupun pengembang, penelitian ini diharapkan dapat menambah wawasan dan pengetahuan tentang metode *Modified K- Nearest Neighbor* untuk klasifikasi fungsi senyawa aktif.

1.4.2 Manfaat Praktis

Dengan adanya penelitian ini diharapkan dapat mempermudah bidang kedokteran untuk menentukan fungsi senyawa aktif dan farmakologinya tanpa melakukan tes laboratorium dengan waktu dan biaya yang relative rendah.

1.5 Batasan Masalah

Batasan penelitian yang digunakan adalah :

1. Data diperoleh dari website Pubchem berupa notasi kode *SMILES* senyawa aktif yang memiliki fungsi farmakologis.
2. Fitur yang digunakan meliputi jumlah O, jumlah OH, jumlah masing-masing senyawa penyusunnya dan panjang kode baris *SMILES* yang digunakan.

1.6 Sistematika Pembahasan

Gambaran secara garis besar pembahasan dari keseluruhan isi Skripsi untuk setiap bab adalah sebagai berikut:

BAB I **PENDAHULUAN**

Pada bab pendahuluan berisi uraian dari latar belakang, rumusan masalah, tujuan, manfaat, batasan masalah dan sistematika penulisan.

BAB II **LANDASAN KEPUSTAKAAN**

Pada 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* dan metode *Modified K- Nearest Neighbor.*

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 Analisis**

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

BAB VII **Kesimpulan dan Saran**

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 Kepustakaan

Penelitian ini didasarkan pada studi pustaka buku dan beberapa penelitian yang pernah dilakukan yang ada kaitannya dengan klasifikasi fungsi senyawa aktif berdasarkan kode *SMILES* menggunakan metode *Modified K-Nearest Neighbor*. Studi literatur ini dilakukan dengan tujuan untuk memperkuat pemahaman domain permasalahan dan juga untuk mencari penyelesaian masalah terbaik. Beberapa penelitian yang dijadikan sebagai sumber penelitian terangkum dalam Tabel 2.1 Kajian Kepustakaan.

Penelitian pertama merupakan penelitian yang dilakukan oleh Dzikrulloh et al. (2017), penelitian ini memberikan sebuah keluaran berupa bobot kriteria untuk menentukan calon guru dan karyawan tata usaha dalam sebuah tes penerimaan. Masukan yang digunakan dalam penelitian ini berupa nilai dari surat lamaran beserta IPK (Indeks Prestasi Komulatif) rata-rata, hasil tes akademik, tes pengetahuan umum dan tes wawancara. Penelitian ini menggunakan implementasi Metode Gabungan *K-Nearest Neighbor* dan *Weighted Product*, sehingga menghasilkan nilai akurasi sebesar 94% dengan *Precission* dan *recall* sebesar 80%.

Penelitian kedua dilakukan oleh Wafiyah et al. (2017) untuk klasifikasi penyakit demam, Dimana objek masukan yang digunakan berupa 15 gejala dari penyakit beserta bobotnya dan menggunakan total data sebanyak 133 pasien penderita penyakit demam berdarah, tifoid dan malaria. Penelitian ini memanfaatkan Metode *Weighted K-NN* untuk mendapatkan hasil klasifikasi yang memiliki akurasi sebesar 88,55% untuk rata-rata pengujian dari pengaruh nilai K.

Penelitian Selanjutnya digunakan untuk klasifikasi penyakit pada tanaman kedelai Dimana penelitian dilakukan oleh Simanjuntak et al. (2017). Berdasarkan dataset berupa penyakit tanaman kedelai dari situs *center for machine learning and intelligent systems*, para Peneliti menggunakan Metode *Modified K-Nearest Neighbor* untuk proses klasifikasinya. Proses ini menghasilkan akurasi sebesar 98.83% dengan jumlah data latih 170 dan 70,23% saat jumlah data latih 30. Hasil akurasi tersebut membuktikan bahwa jumlah data latih sangat berpengaruh terhadap hasil klasifikasi karena metode *MKNN* menerapkan *supervised Learning* untuk proses klasifikasinya.

Penelitian terakhir yang digunakan sebagai rujukan merupakan penelitian yang dilakukan oleh Astuti dkk (2017) untuk klasifikasi penyakit pada kucing. Data latih yang digunakan merupakan data gejala penyakit yang dimiliki kucing, data ini diperoleh dari studi literatu dan wawancara dengan pakar. Hasil klasifikasi menunjukkan Metode *MKNN* teroptimasi menggunakan Algoritma Genetika dengan K optimal 1 mendapatkan nilai akurasi sebesar 100%.

Tabel 2.1 Kajian Kepustakaan

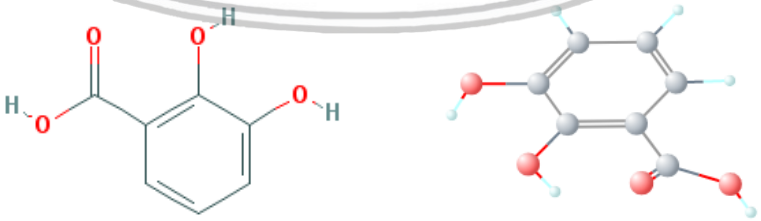
No	Peneliti	Objek (*input*)	Algoritma	Hasil (*Keluaran*)
1	Dzikrulloh, Indriati, & Setiawan (2017)	Masukan berupa nilai dari 4 aspek yaitu: Surat lamaran dan lampiran IPK, Tes akademik, Tes Pengetahuan umum IPTEK, dan Tes wawancara.	Metode *K–Nearest Neighbor* (KNN) dan Metode *Weighted Product* (WP)	Hasil dari pengujian pengaruh nilai K terbaik diperoleh nilai akurasi 94%, *precision* 80%, dan nilai *recall* 80%
2	Wafiyah, Hidayat, & Perdana (2017)	15 gejala penyakit beserta bobotnya dengan total data sebanyak 133 data pasien penderita penyakit demam berdarah, tifoid dan malaria.	Algoritma *Weighted K- Nearest Neighbor*	hasil pengujian terhadap perubahan nilai k dan jumlah data latih didapatkan rata-rata akurasi 88.55%
3	Simanjuntak, Mahmudy, & Sutrisno (2017)	dataset penyakit tanaman kedelai dari situs Center for Machine Learning and Intelligent Systems	Algoritma *Modified K-Nearest Neighbor*	Rata-rata akurasi maksimum 98,83% pada saat jumlah data latih 170 data dan akurasi minimum 70,23% pada saat jumlah data latih 30 data.
4	Astuti, Ratnawati, & Widodo (2017)	data gejala penyakit dan data kucing penderita penyakit diperoleh dari studi literature dan wawancara dengan pakar	Algoritma *Modified K-Nearest Neighbor* Teroptimasi	Hasil akurasi *Modified K-Nearest Neighbor* dengan algoritma genetika untuk k optimal 1 adalah 100%
5	Yunita (2018)	Data berupa senyawa aktif yang memiliki farmakologi dalam bentuk kode *SMILES* (*Simplified Molecular Input Line Entry System*)	Algoritma *Modified K-Nearest Neighbor*	Keluaran yang dihasilkan berupa kelas data senyawa aktif (fungsinya sebagai obat untuk mengatasi penyakit tertentu pada manusia).

2.2 Senyawa dan Cara Merepresentasikannya

Senyawa merupakan zat tunggal kimia yang terdiri dari dua atau lebih unsur kimia sehingga dapat membentuk ikatan serta dapat diuraikan menjadi zat yang lebih sederhana. Senyawa tersebut dapat dikategorikan menjadi senyawa aktif dan senyawa tidak aktif. Senyawa aktif adalah suatu zat yang mempunyai daya atau kemampuan melakukan pencegahan atau penyembuhan saat terjadinya berbagai macam kondisi buruk tubuh dalam proses metabolisme, senyawa aktif adalah senyawa kimia tertentu yang terdapat dalam tumbuhan dan hewan sebagai bahan obat yang mempunyai efek fisiologis terhadap organisme lain, atau sering disebut sebagai senyawa bioaktif (Salni, Marisa, & Mukti, 2011). Berbeda dengan senyawa aktif, senyawa tidak aktif tidak memiliki daya atau kemampuan pencegahan dan penyembuhan di dalam struktur atom penyusunnya. Merujuk pada beberapa pendapat mengenai senyawa aktif sebelumnya maka dapat dikatakan senyawa aktif adalah bahan obat yang berguna untuk menjaga kesehatan pada fisik manusia.

2.2.1 Struktur Senyawa

Senyawa memiliki struktur representasi yang berbeda-beda dimana rumus struktur dari suatu senyawa adalah representasi grafis dari senyawa yang menunjukkan bagaimana atom tersusun. Rumus struktur dapat menunjukkan pengaturan atom dalam ruang tiga dimensi dengan cara yang mungkin tidak dapat dilakukan oleh rumus kimia. Geometri molekuler suatu senyawa kimia dapat digambarkan secara kasar oleh rumus struktur, dengan menggunakan model 2 Dimensi dan 3 Dimensi. Perbedaan sangat jelas terdapat pada struktur senyawa 2D dan 3D adalah dimana pada rumus struktur senyawa 2D digambarkan dengan visualisasi dimana ikatan antar molekul digambarkan dengan jelas. Struktur senyawa 3D merupakan geometri dari senyawa 2D yang digambarkan dalam bentuk animasi dan dapat digerakkan. Senyawa 2 dimensi dan 3 dimensi di contohkan pada Gambar 2.1.



Gambar 2.1 Struktur 2 dimensi dan 3 dimensi dari Senyawa 2, 3-*dihydroxybenzoic acid*

Sumber: https://pubchem.ncbi.nlm.nih.gov/search/search.cgi#

Sistem penulisan senyawa memiliki beberapa format 'penamaan' kimia yang sistematis, seperti dalam basis data kimia digunakan yang setara dan sekuat struktur geometris. Sistem tata nama senyawa kimia ini termasuk *SMILES, InChI dan CML*.

2.2.2 Tata Nama Senyawa

Tata nama senyawa kimia adalah serangkaian aturan persenyawaan kimia yang disusun secra sistematis. Tata nama senyawa kimia disusun berdasarkan aturan IUPAC (*Internatonal Union of Pure And Applied Chemistry*). Sebuah aturan dalam memberi nama senyawa dapat memudahkan dalam penyebutan senyawa tersebut, mengingat banyaknya jumlah senywa di alam semesta dan terus bertambah dengan ditemukannya senyawa buatan.

1. Tata nama senyawa anorganik
   a. Senyawa biner dari logam dan non-logam
      Senyawa biner tersebut umumnya merupakan senyawa ion. Logam membentuk ion positif (kation) dan non-logam membentuk ion negatif (anion). Jika senyawa biner terdiri atas atom logam dan non-logam dengan logam memiliki satu muatan atau bilangan oksidasi maka tata namanya: Nama(logam) + nama(non-logam)akhiran –ida, contoh : NaCl = Natrium Klorida dan CaO = Kalsium Oksida. Jika atom logam memiliki lebih dari satu muatan atau bilangan oksidasi maka penamaannya: nama(logam) + bilangan oksidasi logam + nama(non-logam)akhiran –ida, contohnya FeO = Besi(II)oksida dan PbO_2 = Timbal(IV)oksida.
   b. Senyawa biner dari non-logam dan non-logam
      Penulisan rumus senyawa mengikuti aturan penulisan unsur yang terlebih dahulu dibaca sesuai urutan unsur berikut: B-Si-C-S-As-P-N-H-S-I-Br-Cl-O-F, sebagai contoh rumus kimia dari Amonia adalah Nh_3 bukan H_3N. Jika pasangan unsur yang bersenyawa membentuk lebih dari satu jenis senyawa, maka senyawa-senyawa itu dibedakan dengan menyebutkan angka indeks dalam bahasa yunani sebagai berikut: 1 = mono, 2 = Di, 3 = Tri, 4 = Tetra, 5 = Penta, 6 = Heksa, 7 = Hepta, 8 = Okta, 9 = Nona, 10 = Deka.
      Indeks satu tidak perlu disebutkan kecuali untuk Karbon Dioksida, contohnya Co = Karbon Monoksida dan CO_2 = Karbon dioksida
   c. Senyawa anorganik poliatomik
      Senyawa ini merupakan senyawa ion yang terbentuk dari kation monoatmik dengan anion poliatomik atau kation poliatomik dengan anion monoatomic/poliatomik. Penamaan dimulai dengan menyebutkan kation + anion. Contoh dari penamaan senyawa anorganik poliatomik misalnya Na_2CO_3 = Natrium karbonat dan $KMnO_4$ = Kalium permanganate
   d. Senyawa Asam
      Senyawa asam merupakan zat kimia yang ada dalam air dan melepas ion H+, contohnya HCl. Penamaan senyawanya dengan menyebut asam + anionnya. Contoh penamaan pada senyawa asam yaitu: HCl = Asam Klorida, HNO_3 = Asam Nitrat, dan H_2SO_4 = Asam sulfat.
2. Tata nama senyawa organic
   Jumlah senyawa organik sangat banyak dan tata namanya lebih kompleks sehingga tidak dapat ditentukan dari rumus kimia saja tetapi menggunakan

rumus struktur juga gugus fungsinya. Contoh penamaan senyawa organic yang sederhana misalnya: Metana = CH_4 dan Etana = C_2H_6.

3. Tata nama senyawa yang memiliki nama umum
   Senyawa yang memiliki nama umum boleh saja untuk tidak menggunakan tata nama menurut IUPAC dalam penyebutannya. Contoh senyawa ini adalah NaCl = Garam dapur, $CaCO_3$ = batu kapur dan $NaHCO_3$ = Soda kue.

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

Simplified Molecular Input Line Entry System (*SMILES*) merupakan suatu cara membaca notasi kimia yang di desain untuk melakukan pengenalan senyawa dan informasi kimia dengan cara modern. *SMILES* diciptakan oleh David Weininger pada tahun 1980 menggunakan konsep *graph*. Notasi *SMILES* menggunakan karakter dari kode ASCII sehingga dapat disimpan dalam variabel string. Variabel dari notasi kimia tersebut lebih mudah diproses oleh komputer dan tidak membutuhkan banyak memori. Penggunaan kode *SMILES* yang sederhana memungkinkan pengguna mengkodekan struktur kiia yang mudah digunakan (Weininger, 1987). Sistem penamaan senyawa dalam bentuk kode *SMILES* dapat dilihat pada Gambar 2.2.

C1=CC(=C(C(=C1)O)O)C(=O)O

Gambar 2.2 Contoh Kode *SMILES* untuk Senyawa 2, 3-*dihydroxybenzoic acid*
Sumber: https://pubchem.ncbi.nlm.nih.gov/search/search.cgi#

2.3.1 Aturan Spesifikasi *SMILES*

SMILES menunjukkan struktur molekul sebagai grafik yang pada dasarnya digambarkan oleh gambar kimia dua dimensi yang berorientasi valensi untuk menggambarkan molekul. SMILES dapat direpresentasikan dengan berbagai cara, yaitu:

1. Penulisan Atom

Penulisan atom menyesuaikan simbol atom senyawa yang sebenarnya. Penulisan ini dilakukan dengan cara menuliskan huruf besar atau Kapital, namun jika dalam penulisannya terdapat simbol yang memiliki lebih dari satu huruf maka huruf yang pertama dituliskan dengan huruf besar dan huruf selanjutnya dituliskan dengan huruf kecil. Unsur-unsur dalam "*organic ubset*", B, C, N, 0, P, S, F, Cl, Br, dan I, dapat ditulis tanpa tanda kurung jika jumlah hidrogen yang dilekatkan sesuai dengan valensi normal terendah yang konsisten dengan ikatan eksplisit. Atom dalam cincin aromatik ditentukan oleh huruf kecil misalnya karbon normal diwakili oleh huruf C, karbon aromatik oleh c. Karena hidrogen terpasang dengan tidak adanya tanda kurung, simbol atom berikut ini adalah notasi SMILES yang valid. Beberapa contoh penerapannya dapat dilihat pada Tabel 2.2.

Tabel 2.2 Tabel Penerapan Penulisan Atom

Atom	Penamaan
CH_4	metana
NH_3	ammonia
H_2O	air
PH_3	phosphine
H_2S	hidrogen sulfida
HCl	hidrogen klorida

Hidrogen yang memiliki muatan formal selalu dituliskan di dalam tanda kurung. Jumlah hidrogen ditunjukkan oleh simbol H diikuti oleh digit opsional dan muatan formal ditunjukkan oleh salah satu simbol + atau – diikuti dengan digit opsional. Jika tidak ditentukan, jumlah hidrogen dan muatan diasumsikan nol. Contoh Penerapan penulisan penamaan atom Hidrogen yang memiliki muatan dapat dilihat pada Tabel 2.3.

Tabel 2.3 Tabel Penerapan Penulisan Atom Hidrogen bermuatan

Atom	Penulisan
[H+]	Proton
[OH-]	Anion hidroksil
[OH_3+]	Kation hydronium
[Fe_2+]	Besi(II) kation
[NH_4+]	Kation amonium

2. Penulisan Ikatan

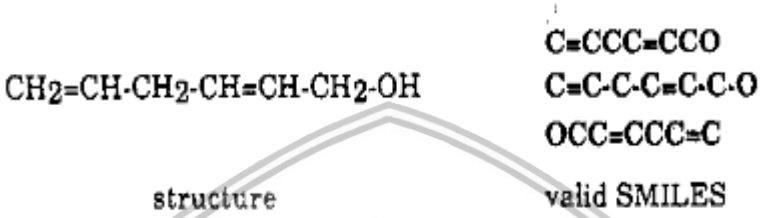
Ikatan antar atom terbagi menjadi tiga macam. Ikatan pertama disebut sebagai ikatan tunggal yang memiliki notasi "-". Tipe kedua dalam ikatan atom yaitu ikatan rangkap yang dilambangkan dengan notasi "=". Tipe ketiga adalah ikatan rangkap tiga yang memiliki notasi "#". Beberapa contoh penulisan ikatan ditunjukkan pada Tabel 2.4,

Tabel 2.4 Tabel Penerapan Penulisan Ikatan Atom

Kode *SMILES*	Penulisan
CC	*ethane* (CH_3 CH_3)
C==C	*ethylene* (CH_2 == CH_2)
COC	*dimethyl ether* (CH_30CH_3)
CCO	*ethanol* (CH_3CH_2OH)

O=C=O	*carbon dioxide* (CO_2)
C#N	*hydrogen cyanide* (HCN)
[H][H]	*molecular hydrogen* (H_2)

untuk struktur *linier*, notasi *SMILES* sesuai dengan notasi diagram konvensional kecuali bahwa hidrogen dapat dihilangkan. Misalnya, 6-hidroksi-1, 4-heksadiena dapat diwakili oleh tiga *SMILES* yang sama-sama valid. Struktur penulisannya dapat dilihat pada Gambar 2.3.

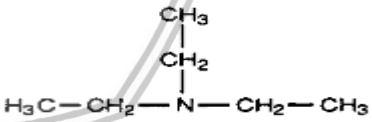
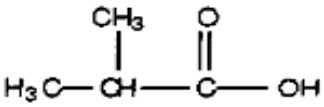
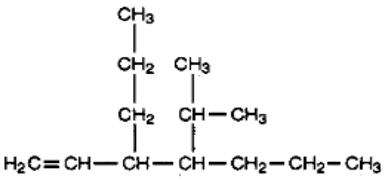


Gambar 2.3 Struktur linier notasi SMILES

3. Penulisan Percabangan

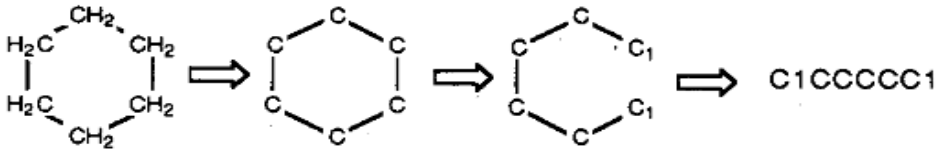
Penulisan notasi kimia pada percabangan ditandai dengan tanda kurung buka dan kurung tutup "()".Cabang dapat disarangkan atau ditumpuk, seperti yang ditunjukkan untuk 3-propil-4-isopropil-1-hepten. Contoh percabangan dapat dilihat pada Tabel 2.5.

Tabel 2.5 Tabel Penerapan Penulisan Percabangan Atom

Kode *SMILES*	Penamaan	Struktur
CCN(CC)CC	*Triethylamine*	
CC(C)C(=O	*Isobutyric acid*	
C=CC(CCC)C(C(C)C)C	3-propil-4-isopropil-1-hepten	

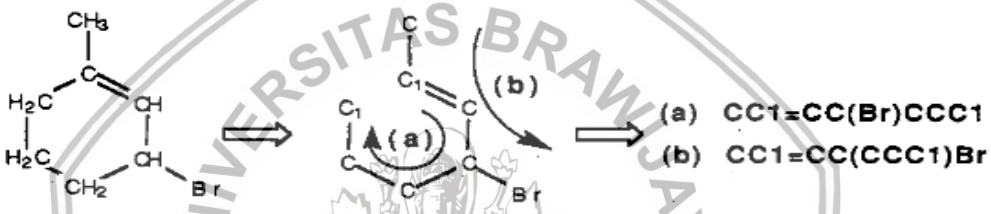
4. Struktur siklik

Struktur siklik diwakili oleh putus satu ikatan tunggal di setiap cincin yang diberi nomor dalam urutan apapun untuk menunjukkan pembukaan cincin (atau cincin-penutupan). Contoh penerapan struktur siklik dapat dilihat pada Gambar 2.4,



Gambar 2.4 Struktur Siklik pada *Cyclohexane*

biasanya ada banyak deskripsi yang berbeda tetapi sama-sama valid dari struktur yang sama, Misalnya dapat dilihat pada Gambar 2.5.



Gambar 2.5 Struktur Siklik pada 1-metil-3-bromo-sikloheksena

Aturan *SMILES* subset empat aturan yang bahkan lebih sederhana sudah cukup untuk sebagian besar senyawa organik. Subset ini hanya menggunakan simbol H, C, N, 0, P, S, F, C1, Br, I, dan (,) dan digit, dengan empat aturan berikut: (1) Atom diwakili oleh simbol atom. (2) Ikatan rangkap dan rangkap diwakili oleh = dan #, secara spektra. (3) Percabangan ditandai dengan tanda kurung. (4) Penutupan cincin ditunjukkan dengan angka yang cocok ditambahkan ke simbol.

2.4 Data Mining

Data mining merupakan kegiatan yang meliputi pengumpulan, pemakaian data historis untuk menemukan keteraturan, pola dan hubungan dalam set data berukuran besar. Data mining terbentuk karena adanya beberapa alasan, yaitu adanya kebutuhan akan pengetahuan pada data yang besar, adanya evolusi atau perkembangan teknologi *database* dan pentinganya analisa data. Ketersediaan data yang besar dan perkembangan teknologi membuat pengolahan data dilakukan secara modern menggunakan berbagai Metode menggunakan teknologi agar tidak menimbulkan kesan membosankan jika dilakukan secara manual. Kebutuhan data yang besar dalam suatu proses pengolahan data menyebabkan perlunya teknologi untuk memberikan waktu singkat dan efisien dalam prosesnya. Beberapa proses yang dapat dilakukan dalam data mining adalah *preprocessing,* klasifikasi*,* klusterisasi dan lainnya.

2.4.1 Preprocessing

Preprocessing merupakan suatu proses yang berguna untuk melakukan berubahan pada bentuk data yang awalnya belum terstruktur kemudian menjadi bentuk data yang terstruktur sesuai dengan kebutuhan, *preprocessing* juga berguna untuk mengetahui letak dan banyak suatu huruf atau kata (*the number of terms*) (Manning, Raghavan, & Schütze, 2009). Pada penelitian ini dilakukan *preprocessing* terhadap notasi *SMILES* dengan mencari jumlah huruf atau masing-masing unsur yang ada dalam senyawa tersebut. *Preprocessing* terhadap notasi *SMILES* adalah pemrosesan notasi *SMILES* untuk mencari dan mengambil lambang atom yang ada pada notasi agar dapat ditentukan panjang dari senyawa (notasi *SMILES*) dan untuk mengetahui jumlah dari masing-masing atom. Nantinya panjang notasi *SMILES* dan jumlah masing-masing atom dijadikan sebagai *input* dalam proses perhitungan klasifikasi.

2.4.2 Klasifikasi

Klasifikasi merupakan salah satu teknik data mining yang memiliki kemampuan untuk melakukan proses klasifikasi data. Klasifikasi bisa digunakan untuk menemukan model atau fungsi yang membedakan kelas data. Klasifikasi bertujuan untuk memprediksi kelas dari suatu objek yang labelnya tidak diketahui. Klasifikasi merupakan suatu teknik dengan melihat tingkah laku dan atribut dari kelompok yang telah didefinisikan. Teknik ini dapat memberikan klasifikasi pada data baru dengan memanipulasi data yang ada yang telah diklasifikasi dan dengan menggunakan hasilnya untuk memberikan sejumlah aturan. Teknik ini menggunakan *supervised induction,* yang memanfaatkan kumpulan pengujian dari record yang terklasifikasi untuk menentukan kelas-kelas tambahan (Kurniawan, 2017).

Klasifikasi merupakan proses untuk menyatakan suatu objek ke dalam salah satu kategori kelas yang telah didefinisikan sebelumnya. Tujuan dari klasifikasi adalah data-data yang sebelumnya belum termasuk dalam kategori kelas dapat dinyatakan kelasnya secara akurat. Tahapan-tahapan klasifikasi terdiri dari:

1. Pembangunan Model
   Pada tahapan ini dibuat model untuk menyelesaikan masalah klasifikasi data, model ini dibangun berdasarkan data pelatihan.
2. Penerapan Model
   Pada tahapan ini model yang sudah dibangun sebelumnya digunakan untuk menentukan atribut atau kelas dari sebuah data baru yang atibut atau kelasnya belum diketahui.
3. Evaluasi
   Pada tahapan ini hasill dari tahapan sebelumnya dievaluasi menggunakan parameter terstruktur untuk menentukan apakah model tersebut dapat diterima.

2.4.3 *K-Nearest Neighbor* (KNN)

Algoritma *K-Nearest Neighbor* merupakan metode klasifikasi objek yang mengklasifikasikan berdasarkan data yang jaraknya paling dekat dengan objek. Data diporyeksikan ke ruang berdimensi dan direpresentasikan fitur datanya yang kemudian dibagi berdasarkan klasifikasi data. Algoritma KNN mencari K *training record* yang memiliki jarak terdekat dari *record* baru, untuk memprediksi kelas dari *record* baru tersebut (Cahyaningtyas, Ridok, & Dewi, 2013). Pada metode ini memiliki beberapa kelebihan seperti tangguh terhadap data pelatihan yang *noisy* dan efektif apabila data pelatihan berjumlah besar. Metode ini memiliki kekurangan juga diantaranya perlu ditentukan nilai K yang paling optimal yang menyatakan jumlah tetangga terdekat dan biaya komputasi yang cukup tinggi karena perhitungan jarak harus dilakukan pada setiap *query instance* secara bersama-sama dengan seluruh instan dari *training sample*.

Prinsip kerja *K-Nearest Neighbor* (KNN) adalah mencari jerak terdekat antara data yang dievaluasi dengan K tetangga terdekatnya dalam data pelatihan. Dekat jauhnya data dapat dihitung dengan *Euclidean distance.* Persamaan perhitungan untuk mencari jarak dengan menggunakan rumus *Euclidean distance* ditunjukkan pada Persamaan 2.1

$$\mathbf{d_{xj,y}} = \sqrt{\sum_{\mathbf{i=0}}^{\mathbf{n}} (\mathbf{x_{j,i}} - \mathbf{y_i})^2}, \qquad (2.1)$$

dimana,

$\boldsymbol{x_j}$ = Data Latih ke-j, j= 1,2,3,… N

$\boldsymbol{y}$ = Data Uji

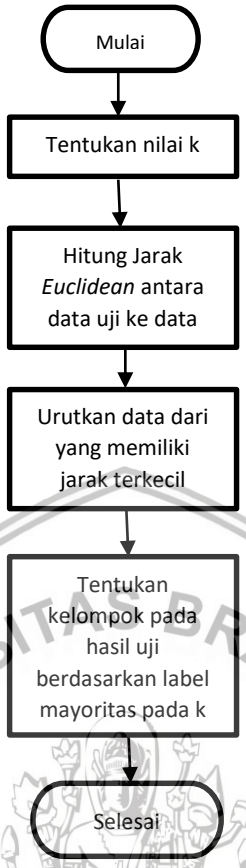
i, j = indeks Data

n = banyak fitur

d = Jarak

N= banyak data latih,

sebelum melakukan perhitungan dengan metode KNN, terlebih dahulu harus menentukan data latih dan data uji. Kemudian dilakukan proses perhitungan untuk mencari jarak terdekat menggunakan rumus Persamaan 2.1. Setelah mendapatkan jarak tetangga terdekat sesuai nilai k, dilakukan tahapan perhitungan dengan metode KNN yang digambarkan dengan diagram alur seperti pada Gambar 2.3.



Gambar 2.6 Proses Metode K-Nearest Neighbor

(Sumber: Alfian 2014)

2.4.4 *Modified K-Nearest Neighbor* (MKNN)

Modified K-Nearest Neighbor (MKNN) adalah menempatkan label kelas data sesuai dengan k divalidasi poin data yang sudah ditetapkan dengan perhitungan *K-Nearest Neighbor* (KNN) tertimbang (Parvin, Alizadeh, & Minaei-Bidgoli, 2008). Dalam algoritma MKNN, setiap data pada data *training* harus melalui proses *validitas* terlebih dahulu pada awalnya. *Validitas* setiap data tergantung pada setiap tetangganya. Tujuan utama yang menjadi dasar modifikasi pada metode *KNN* ini adalah menentukan kelas label dari *query instance* ke dalam K data latih yang telah divalidasi. Setelah menentukan *Validitas*, *weighted* KNN dilakukan pada setiap data uji. Secara garis besar terdapat dua proses utama dalam metode ini, yaitu (Parvin, Alizadeh, & Minaei-Bidgoli, 2008):

1. *Validitas* data Latih

Proses *validitas* dilakukan untuk semua data pada data *training*. *Validitas* setiap data tergantung pada setiap tetangganya. Setelah dihitung *validitas* tiap data maka nilai v*aliditas* tersebut digunakan sebagai informasi lebih mengenai data tersebut. Untuk menghitung v*aliditas* dari data pada data *training*, tetangga terdekatnya perlu dipertimbangkan. Di antara tetangga terdekat dengan data, *validitas* digunakan untuk menghitung jumlah titik dengan label yang sama untuk

data tersebut. Persamaan yang digunakan untuk menghitung *validitas* dari setiap titik pada data *training* adalah seperti pada Persamaan 2.2

$$\mathbf{Validitas(v)} = \frac{\mathbf{1}}{\mathbf{K}}\sum_{\mathbf{i-1}}^{\mathbf{K}} \mathbf{S(lbl(v), (lbl(N_i(v)))}\ , \qquad (2.2)$$

dimana:

K = Jumlah titik terdekat

$lbl(v)$ = Kelas v

$N_i(v)$ = Label kelas titik terdekat v,

fungsi S pada Persamaan 2.2 digunakan untuk menghitung kesamaan antara titik x dan data ke-I dari tetangga terdekat. Fungsi S dituliskan dengan Persamaan 2.3

$$\mathbf{S(a,b)} = \begin{cases} \mathbf{1} & \mathbf{a = b} \\ \mathbf{0} & \mathbf{a \neq b}, \end{cases} \qquad (2.3)$$

dimana,

a = Kelas a pada data training

b = Kelas lain selain kelas a pada data training,

melalui Persamaan 2.3 dapat diketahui bahwa a dan b adalah label kelas kategori suatu data latih. S bernilai 1 jika label kategori a sama dengan kabel kategori b, sedangkan S bernilai 0 jika label kategori a tidak sama dengan label kategori b.

2. Weight voting

Weight voting adalah salah satu variasi metode *KNN* yang menggunakan K tetangga terdekat dan hasil perhitungan dari jarak masing-masing data. Pada metode *MKNN*, masing-masing K tetangga terdekat dihitung menggunakan Persamaan 2.2 dan Persamaan 2.3. Nilai *validitas* yang dihasilkan dari setiap data yang dihitung sebelumnya kemudikan dikaliakan dengan hasil *Weight voting* berdasarkan jarak. Sehingga dalam metode *MKNN*, nilai rumus persamaan untuk menghitung *Weight voting* dinyatakan pada Persamaan 2.4

$$\mathbf{W(w)} = \mathbf{Validitas(v)}\frac{\mathbf{1}}{\mathbf{d(x,y)+0{,}5}}\ , \qquad (2.4)$$

dimana:

$W(w)$ = Nilai *Weight voting*

$d(\boldsymbol{x}, \boldsymbol{y})$ = Jarak *Euclidean*

$0{,}5$ = Konstanta.

Menurut penelitian Parvin (2010) teknik *Weight voting* ini berpengaruh terhadap data yang mempunyai nilai v*aliditas* lebih tinggi dan paling dekat dengan data. Pada tahapan perkalian *validitas* dengan jarak data dapat mengatasi kelemahan antara jarak setiap data dengan *weight* yang memiliki banyak masalah dalam *outlier.* Oleh karena itu, metode *MKNN* secara signifikan

lebih kuat daripada metode KNN tradisional yang hanya berdasarkan jarak (Pervin, 2010).

2.5 Evaluasi

2.5.1 Akurasi Sistem

Akurasi merupakan seberapa akurat nilai hasil suatu pengukuran terhadap nilai sebenarnya. Dalam penelitian ini akurasi untuk hasil klasifikasi senyawa aktif dihitung dari jumlah hasil klasifikasi yang tepat dibagi dengan jumlah data yang diujikan. Tingkat akurasi dihitung dengan menggunakan Persamaan 2.5 berikut,

$$Tingkat\ Akurasi = \frac{\sum Data\ uji\ benar}{\sum Total\ data\ uji} x\ 100\% \quad (2.5)$$

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

Bab ini menjelaskan tentang metodologi yang digunakan untuk penelitian klasifikasi senyawa aktift menggunakan Metode *Modified K- Nearest Neighbor* Dalam metodologi penelitian ini juga menjelaskan tentang studi kepustakaan, teknik pengumpulan data yang digunakan dalam penelitian, lokasi penelitian, strategi dalam penelitian, tipe penelitian, implementasi algoritme *M-KNN* dan tentang teknik analisis data.

3.1 Studi Kepustakaan

Studi kepustakaan dalam penelitian ini untuk memehami dan mempelajari konsep tentang permasalahan pada penelitian ini. Sehingga pada penelitian ini dibutuhkan referensi yang relevan terkait dengan penelitian yang sedang dilakukan. Informasi yang didapatkan bisa diperoleh dari buku, jurnal, internet ataupun dari dosen pebimbing dan mendapatkan informasi, diantaranya:

1. Fungsi senyawa dengan menggunakan kode *SMILES* (*Simplified Molecular Input Line Entry System*)
2. Metode *Modified K-Nearest Neighbor* untuk klasifikasi

Kemudian teori yang bersangkutan dengan penelitian yang telah didapatkan akan disertakan dalam dokumen penelitian studi kepustakaan yang dilakukan untuk pembelajaran.

3.2 Tipe Penelitian

Penelitian ini menggunakan penelitian dengan tipe nonimplementatif. Penelitian tipe nonimplementatif adalah proses penelitian yang menggali informasi yang terjadi akibat dari fenomena atau sebuah kejadian yang bertujuan untuk mengidentifikasi sebuah elemen penting daari sebuah objek penelitian, penelitian nonimplementatif ini lebih mengutamakan pendekatan deskriptif dan analitik.

a. Pendekatan deskriptif adalah sebuah penelitian yang berdasarkan kejadian tertentu berdasarkan hasil dari sebuah analisis terhadap data yang digunakan. Pendekatan deskriptif ini digunakan untuk menjelaskan karakteristik dari sebuah objek dan hasilnya yaitu investigasi.

b. Pendekatan analitik adalah sebuah penelitian yang berdasarkan kejadian tertentu berdasarkan hasil sebuah analisi terhadap data yang digunakan. Pendekatan analitik ini digunakan untuk menjelaskan hubungan antar elemen pada sebuah objek dan hasilnya yaitu analisis.

3.3 Strategi Penelitian

Pada penelitian ini menggunakan metode eksperimen. Penelitian eksperimen secara umum adalah penelitian yang megandung sebab akibat

dengan menggunakan uji coba oleh peneliti. Pada umumnya penelitian yang menggunakan metode eksperimen dilakukan di laboratorium.

3.4 Lokasi Penelitian

Pada penelitian ini dilakukan di Laboratorium Komputasi Cerdas Fakultas Ilmu Komputer Universitas Brawijaya

3.5 Teknik Pengumpulan Data

Data yang digunakan dalam penelitian ini diambil dari situs Pubchem dengan menggunakan senyawa yang memiliki fungsi farmakologi. Data yang diambil berupa kode *SMILES* senyawa aktif yang dapat dikomputasikan. Data yang digunakan berjumalh 260 data yang terbagi menjadi dua kelas klasifikasi yaitu kelas saraf dan kelas jantung.

3.6 Analisis Kebutuhan

Pada tahap ini dilakukan tahap analisis metode MKNN 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

3.7 Perancangan Sistem

Perancangan sistem pada penelitian ini dilakukan setelah analisis kebutuhan sistem sudah terpenuhi. Perancangan sistem ini dilakukan agar mempermudah dalam mengimplementasikan, pengujian dan menganalisis. Pada penelitian sistem ini terdapat diagram alir untuk menjelaskan tahapan pada sistem, perancangan *interface* sistem, dan perancangan pengujian pada sistem. Pertama melakukan *preprocessing* data terhadap data latih dan data uji, kemudian melakukan tahap training dengan menggunakan metode *modified K-NN,* dan hasil dari outpunya yaitu klasifikasi senyawa aktif berdasarkan farmakologinya.

3.8 Implementasi Sistem

Implementasi sistem mengacu pada rancangan sistem yang sudah dibuat. Pada tahap implementasi, sistem menggunakan metode *Modified 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.

3.9 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.

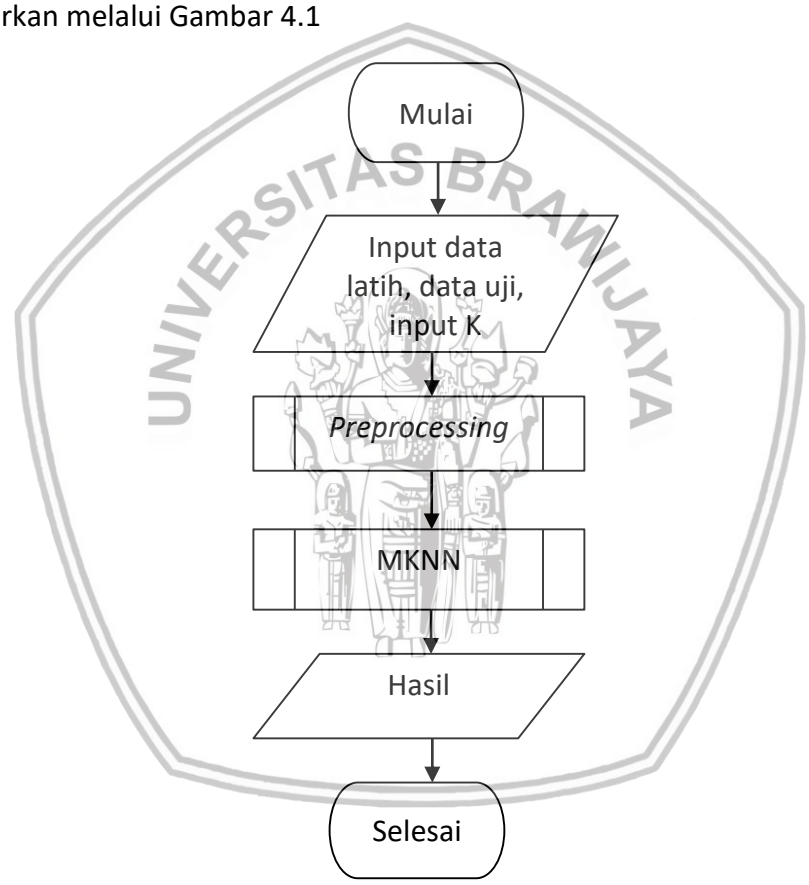
3.10 Kesimpulan dan Saran

Penarikan kesimpulan dilakukan setelah semua tahapan analisis dan proses perhitungan telah selesai dilakukan. Penarikan kesimpulan dilakukan untuk menjawab rumusan permasalahan yang sudah ditetapkan. Tahap terakhir dari penulisan adalah saran yang bertujuan memperbaiki kesalahan-kesalahan yang terjadi serta untuk memberikan masukan dan pertimbangan untuk penelitian selanjutnya.

BAB 4 PERANCANGAN

4.1 Deskripsi Umum Sistem

Sistem yang dibuat bertujuan untuk mengetahui penerapan metode *Modified K-Nearest Neighbor* pada permasalahan klasifikasi pada fungsi aktif senyawa data *SMILES*. Fitur yang digunakan sebagai masukan ada 11, yaitu jumlah masing-masing elemen atom yang kemudian dibagi dengan panjang kode *SMILES*. SIstem ini mempunyai 2 proses, yaitu proses pelatihan dan proses pengujian. Pada proses pelatihan dan pengujian masing-masing memerlukan masukan berupa data latih dan data uji. Proses pelatihan dan pengujian secara keseluruhan digambarkan melalui Gambar 4.1



Gambar 4.1 Diagram Alir Sistem

4.2 Perancangan Sistem

4.2.1 Basis Pengetahuan

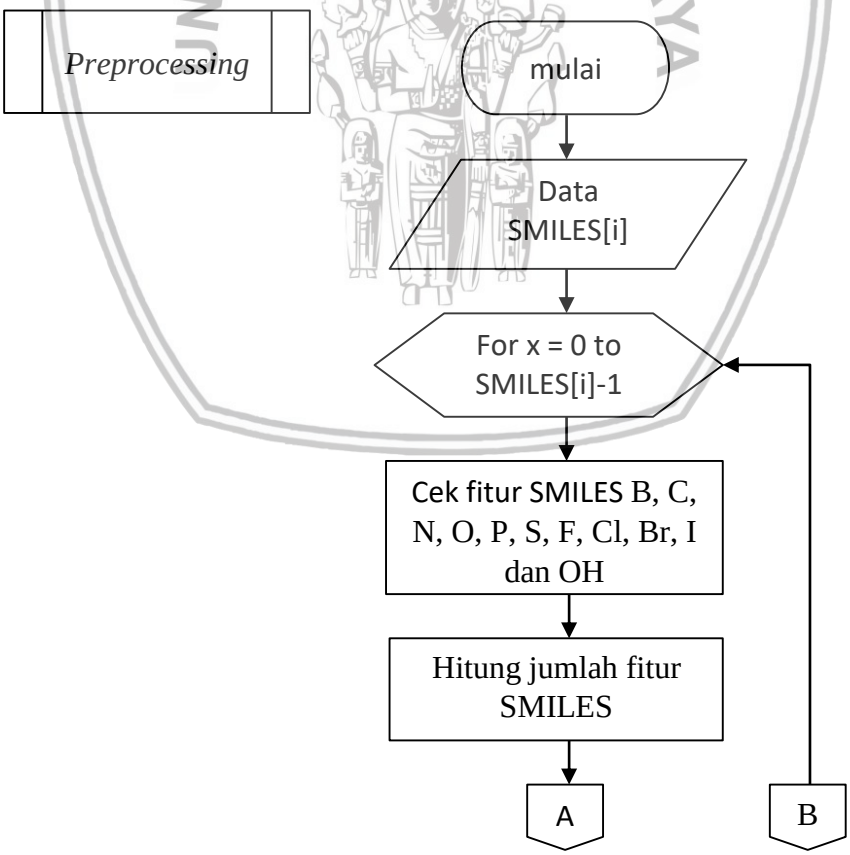
Basis pengetahuan berisi fakta, pemikiran, teori maupun prosedur untuk merumuskan dan memecahkan suatu masalah. Basis pengetahuan tersebut terdiri dari dua bentuk pendekatan yaitu pendekatan berbasis aturan dan pendekatan berbasis kasus. Basis pengetahuan merupakan representasi pengetahuan dari hasil analisis data *SMILES*. Terdapat 11 fitur masukan yang

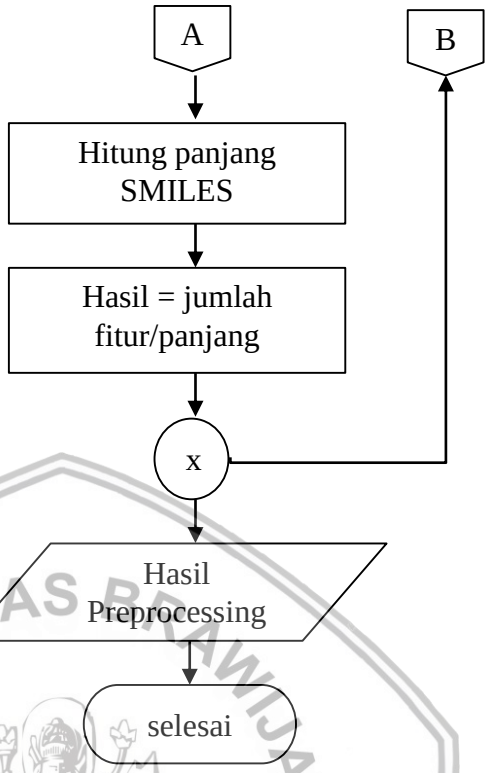
digunakan sebagai perhitungan klasifikasi senyawa kode *SMILES* meliputi jumlah masing-masing elemen B, C, N, O, P, S, F, Cl, Br, I dan OH. Masing-masing fitur dibagi dengan panjang senyawa kode *SMILES* sebelum diproses.

Pada penelitian ini digunakan dataset dengan jumlah 260 data. Pembagian data tersebut menjadi 2 kelas yaitu kelas Jantung dan kelas Saraf. Pada masing-masing kelas, memiliki jumlah data sebanyak 100 untuk data kelas jantung dan 160 untuk data kelas saraf. Selanjutnya kelas-kelas tersebut dibagi kembali sesuai kebutuhan pada saat pengujian sistem.

4.2.2 Preprocessing

Pada tahap ini, setiap data *SMILES* terdiri atas 260 data yang dikelompokkan menjadi 2 kelas kategori farmakologi senyawa aktif yaitu kelas Saraf dan Jantung. Data senyawa aktif tersebut terdiri dari notasi *SMILES* yang harus di konversikan menjadi numerik untuk dapat diolah membentuk perhitungan dengan Algoritma MKNN. Proses konversi data tersebut terdapat pada tahap *preprocessing* untuk menghasilkan nilai numerik pada setiap elemen fitur yang digunakan dalam perhitungan. Hasil dari *preprocessing* tersebut mengisi tiap-tiap elemen fitur meliputi jumlah masing-masing elemen B, C, N, O, P, S, F, Cl, Br, I dan OH. Proses *preprocessing* dapat dilihat pada Gambar 4.2.





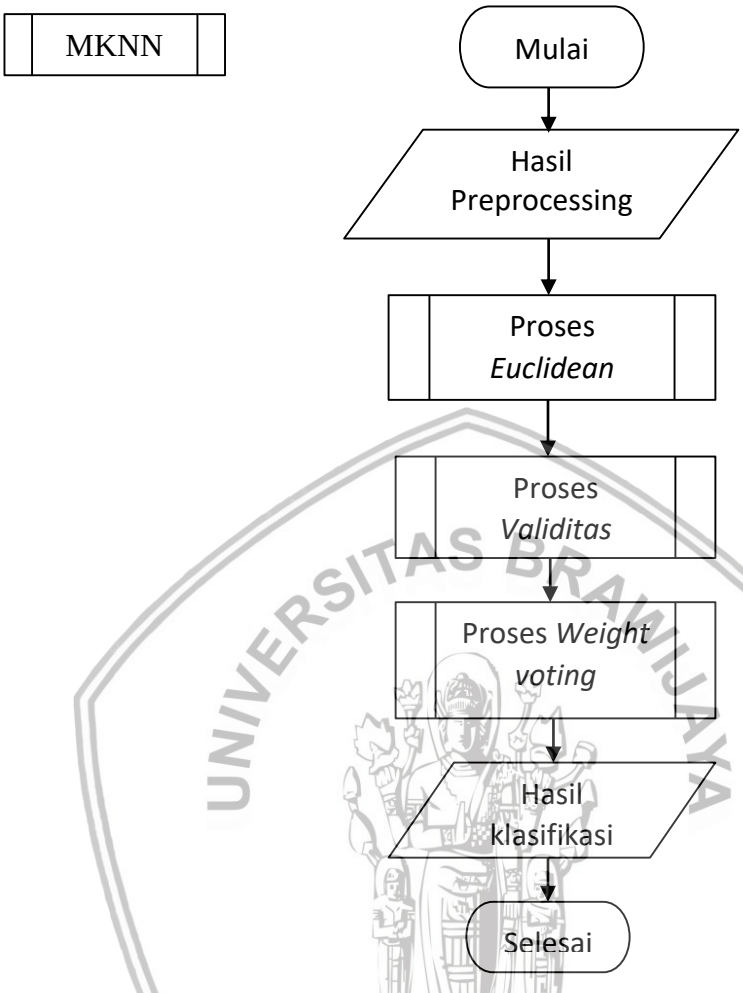
Gambar 4.2 *Flowchart* Proses *Preprocessing*

4.2.3 Klasifikasi Algoritma MKNN

Pada penelitian ini proses klasifikasi dengan MKNN menggunakan 2 kelas kategori. Proses MKNN dijabarkan sebagai berikut:

1. Memasukkan data kode *SMILES* berupa jumlah masing-masing elemen beserta panjang kode *SMILES*.
2. Menginisialisasi nilai K tetangga.
3. Melakukan perhitungan jarak untuk setiap data latih menggunakan rumus Persamaan 2.1 yaitu *eucledian distance*.
4. Melakukan perhitungan nilai *Validitas* setiap data latih menggunakan rumus Persamaan 2.2.
5. Melakukan perhitungan jarak untuk setiap data uji dengan data latih menggunakan rumus Persamaan 2.1.
6. Melakukan perhitungan *Weight voting* pada setiap data uji menggunakan rumus Persamaan 2.4
7. Menentukan kelas klasifikasi dari data uji sesuai nilai *Weight voting* terbesar.

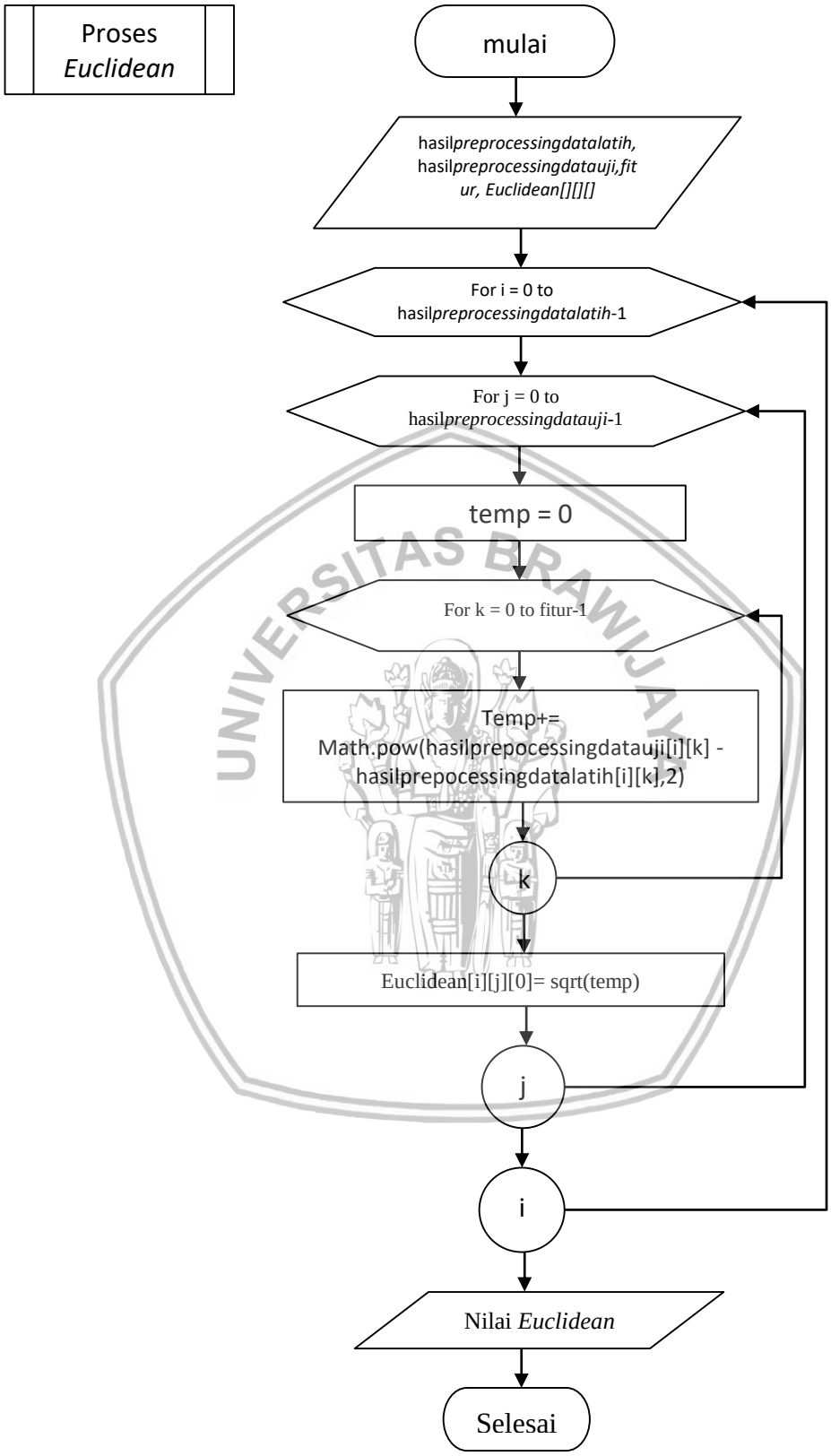
Proses Klasifikasi *Modified K-Nearest Neighbor* digambarkan pada Gambar 4.3



Gambar 4.3 *Flowchart* Proses *Modified K-Nearest Neighbor*

4.2.4 Proses *Euclidean*

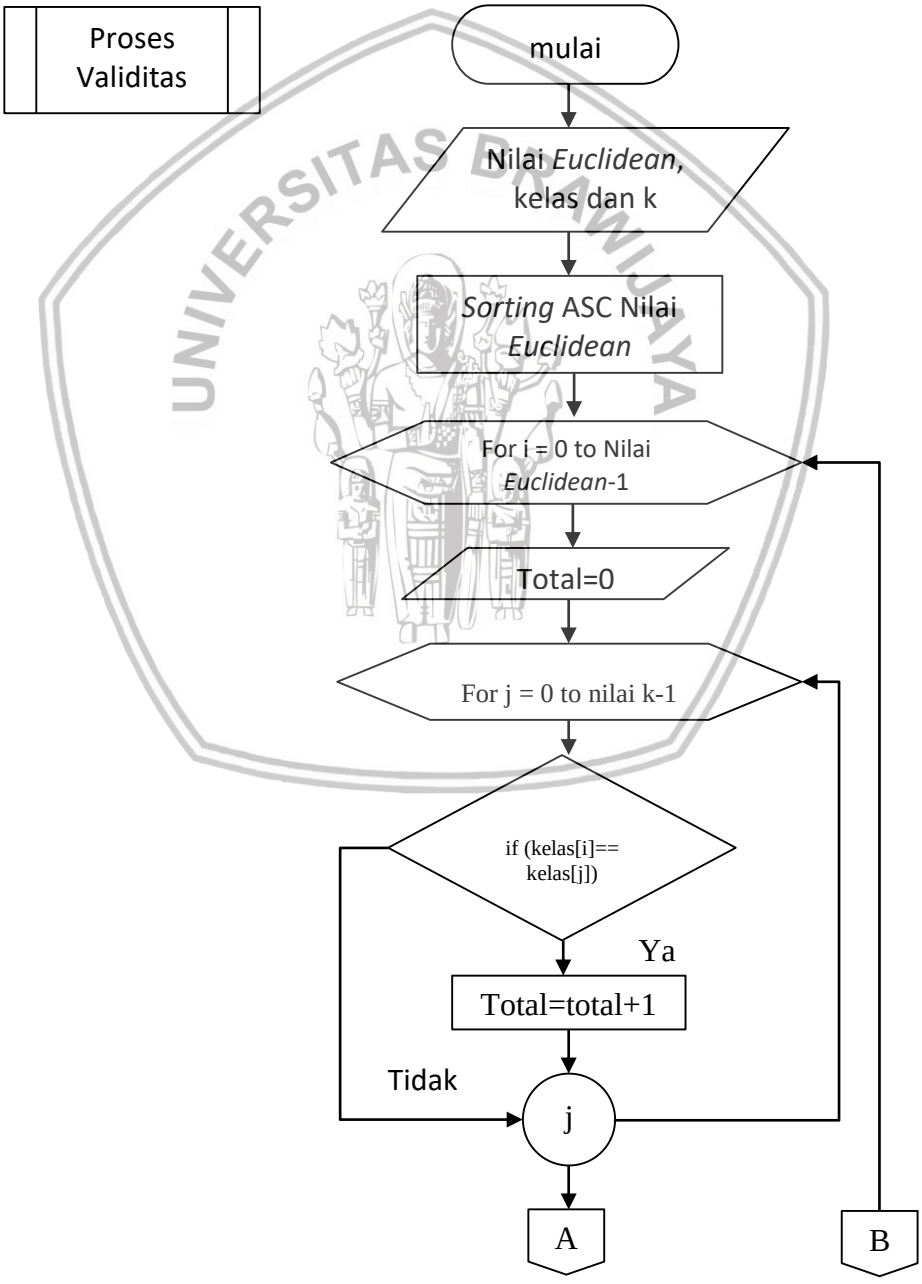
Pada tahap ini dilakukan proses perhitungan nilai jarak kedekatan tetangga antara setiap data latih dan antara data latih dengan data uji menggunakan Persamaan 2.1 Alur proses *Euclidean* dijabarkan pada Gambar 4.4

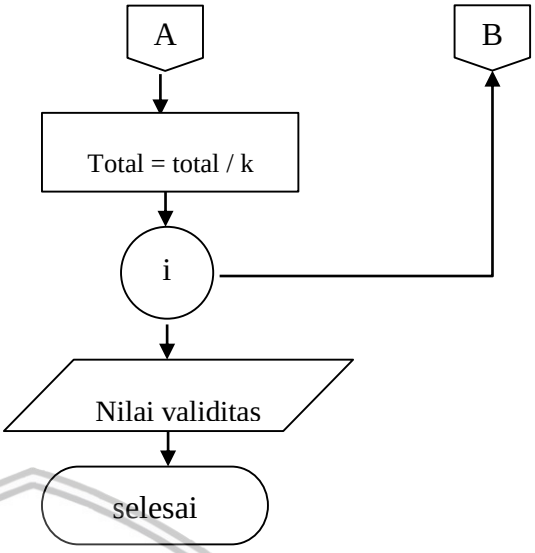


Gambar 4.4 *Flowchart* Proses *Euclidean*

4.2.5 Proses Validitas

Proses *Validitas* dijelaskan alur tahapan yang terdiri dari beberapa tahapan yaitu menginputkan data latih dan input nilai k nya. Setelah menginputkan keduanya dilakukan perhitungan menggunakan Persamaan 2.2. Pada saat hasil nilai tersebut didapatkan dilakukan perhitungan *Validitas* dengan membandingkan kelas-kelas pada data latihnya. Perbandingan ini menggunakan ketentuan jika kelasnya sama maka nilainya 1 dan jika sebaliknya makan nilai 0 seperti pada aturan Persamaan 2.3. Proses pengambilan nilai validitas didasarkan pada nilai terdekat dari jarak *Euclidean* yang terdekat yang diambil sebanyak nilai k yang dimasukkan. Alur proses *Validitas* dijabarkan pada Gambar 4.5.

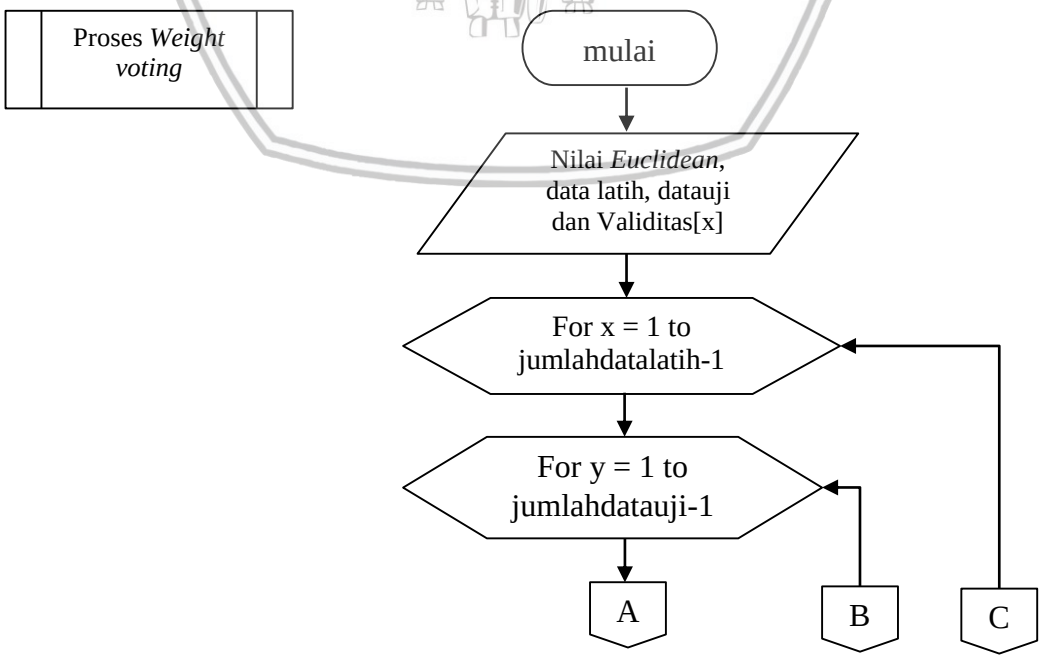


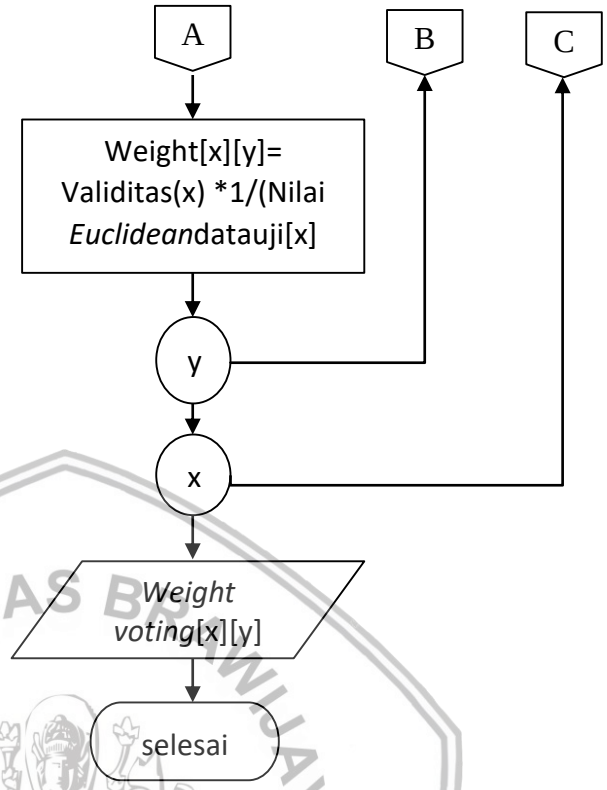


Gambar 4.5 *Flowchart* Proses Validitas

4.2.6 Proses *Weight Voting*

Dalam proses ini hal yang dilakukan pertama adalah memasukkan nilai *Validitas* data latih dan nilai hasil *Euclidean* dari data uji dan data latih. Nilai masukan tersebut kemudian dihitung menggunakan persamaan 2.4 dan menghasilkan nilai *keluaran weigh voting*. Hasil keuaran pada proses ini Selanjutnya akan di urutkan berdasarkan nilai tertinggi dan diambil nilai sebanyak K yang di masukan. Nilai *Weight voting* Selanjutnya dijumlahkan sesuai kelas kategori dari data latih untuk menentukan kelas prediksi sebagai hasil akhir keputusan sistem. Alur Proses *Weight voting* dijabarkan pada Gambar 4.6.





Gambar 4.6 *Flowchart* Proses *Weight voting*

4.3 Perhitungan Manual

Dalam proses perhitungan manual menggunakan 30 data set yang dibagi menjadi 24 data latih dan 6 data uji. Pada 24 data latih yang digunakan terdiri dari 2 kelas dengan yaitu kelas Saraf dan kelas Jantung. Pada data uji yang digunakan terdiri dari 2 kelas dengan pembagian 3 data pada masing-masing kelas Jantung dan Saraf. Data Latih dapat dilihat pada Tabel 4.1 sedangakan data uji dapat dilihat pada Tabel 4.2.

Tabel 4.1 Data Latih Manualisasi

Data	*SMILES*	Farmakologi
SML1	CC(C(=O)O)C(=O)O	Saraf
SML2	C(CC(C(=O)O)N)CN=C(N)N	Saraf
SML3	C(CC(=O)O)C(=O)O	Saraf
SML4	CC1=C(SC=[N+]1CC2=CN=C(N=C2N)C)CCOP(=O)(O)OP(=O)(O)O	Saraf
SML5	C(CC(=O)O)C(C(=O)O)N	Saraf
SML6	C(CCN)CC(C(=O)O)N	Saraf
SML7	CC(C(C(=O)O)N)O	Saraf
SML8	C(CS(=O)(=O)O)N	Saraf
SML9	CC(=O)C1(CCC2C1(CCC3C2C=C(C4=CC(=O)CCC34C)Cl)C)OC(=O)C	Jantung

SML10	C1=C2C(=CC(=C1Cl)S(=O)(=O)N)S(=O)(=O)N=CN2	Jantung
SML11	C1=CC=C2C(=C1)C(=O)NC2(C3=CC(=C(C=C3)Cl)S(=O)(=O)N)O	Jantung
SML12	C1CN=C(N1)NC2=C(C=CC=C2Cl)Cl	Jantung
SML13	C1CCC(C1)CC2NC3=CC(=C(C=C3S(=O)(=O)N2)S(=O)(=O)N)Cl	Jantung
SML14	C1C2CC(C1C=C2)C3NC4=CC(=C(C=C4S(=O)(=O)N3)S(=O)(=O)N)Cl	Jantung
SML15	C1CN(CC2=CC=CC=C21)C(=N)N	Jantung
SML16	CC[N+](CC)(CC)CCOC1=C(C(=CC=C1)OCC[N+](CC)(CC)CC)OCC[N+](CC)(CC)CC.[I-].[I-].[I-]	Saraf
SML17	CC1CC2C3CCC(C3(CC(C2C4(C1=CC(=O)C=C4)C)O)C)(C(=O)CO)O	Saraf
SML18	CC(C)CC(C(=O)O)NC(=O)C(CC1=CC=CC=C1)NC(=O)CNC(=O)CNC(=O)C(CC2=CC=C(C=C2)O)N	Saraf
SML19	C1=CC=C(C=C1)N2C3=CC=CC=C3C(C2=O)(CC4=CC=NC=C4)CC5=CC=NC=C5	Saraf
SML20	CN(C)CCOC(=O)COC1=CC=C(C=C1)Cl.Cl	Saraf
SML21	CC(CC1=CC=CC=C1)N(C)CC#C	Saraf
SML22	C1CN(CCN1)C2=NC3=CC=CC=C3OC4=C2C=C(C=C4)Cl	Saraf
SML23	CCN(CC)CCOC1=CC=C(C=C1)C(=NO)C	Saraf
SML24	COC1=C(C=C2C(=C1)C(=NC(=N2)N3CCN(CC3)C(=O)C4=CC=CO4)N)OC	Jantung

Tabel 4.2 Data Uji Manualisasi

Data	*SMILES*	Farmakologi
SM1	CC(C(=O)O)O.C1CCN(CC1)CCC(C2CC3CC2C=C3)(C4=CC=CC=C4)O	saraf
SM2	C1CCN(CC1)CCC(C2CCCC2)(C3=CC=CC=C3)O	saraf
SM3	CN(C)CCOC(C1=CC=CC=C1)C2=CC=CC=C2	saraf
SM4	CC(=O)C1(CCC2C1(CCC3C2C=C(C4=CC(=O)CCC34C)Cl)C)OC(=O)C	Jantung
SM5	C1=C2C(=CC(=C1Cl)S(=O)(=O)N)S(=O)(=O)N=CN2	Jantung
SM6	C1=CC=C2C(=C1)C(=O)NC2(C3=CC(=C(C=C3)Cl)S(=O)(=O)N)O	Jantung

Langkah 1 Melakukan *Preprocessing* data *SMILES*

Data latih dan data uji yang akan digunakan terlebih dahulu dilakukan *preprocessing* untuk mendapatkan nilai dari masing-masing fitur yang akan dipakai pada proses perhitungan selanjutnya. Data *SMILES* masing-masing dicari dan dijumlahkan sesuai banyak senyawa pembentuknya. Kemudian dihitung panjang *SMILES*, setelah semua nilai didapatkan masing-masing fitur dibagi dengan panjang data *SMILES*. Nilai pembagian tersebut yang Selanjutnya digunakan untuk melanjutkan proses klasifikasi. Data *SMILES* yang telah dilakukan *preprocessing* dapat dilihat pada Tabel 4.3 dan Tabel 4.4.

Tabel 4.3 Data Latih Manualisasi Hasil *Preprocessing*

Data	B	C	N	O	OH	P	S	F	CL	Br	I	Farmakologi
SML1	0	0.2500	0.0000	0.1250	0.1250	0	0	0	0	0	0	Saraf
SML2	0	0.2727	0.1818	0.0455	0.0455	0	0	0	0	0	0	Saraf
SML3	0	0.2500	0.0000	0.1250	0.1250	0	0	0	0	0	0	Saraf
SML4	0	0.2500	0.0833	0.0833	0.0625	0.0417	0	0	0	0	0	Saraf
SML5	0	0.2500	0.0500	0.1000	0.1000	0	0	0	0	0	0	Saraf
SML6	0	0.3529	0.1176	0.0588	0.0588	0	0	0	0	0	0	Saraf
SML7	0	0.2667	0.0667	0.0667	0.1333	0	0	0	0.0000	0	0	Saraf
SML8	0	0.1333	0.0667	0.1333	0.0667	0	0	0	0.0000	0	0	Saraf
SML9	0	0.5111	0.0000	0.0889	0.0000	0	0	0	0.0222	0	0	Jantung
SML10	0	0.1892	0.0811	0.1081	0.0000	0	0	0	0.0270	0	0	Jantung
SML11	0	0.3111	0.0444	0.0667	0.0222	0	0	0	0.0222	0	0	Jantung
SML12	0	0.4091	0.1364	0.0000	0.0000	0	0	0	0.0909	0	0	Jantung
SML13	0	0.2955	0.0682	0.0909	0.0000	0	0	0	0.0227	0	0	Jantung
SML14	0	0.3043	0.0652	0.0870	0.0000	0	0	0	0.0217	0	0	Jantung
SML15	0	0.4762	0.1429	0.0000	0	0	0	0	0	0	0	Jantung
SML16	0	0.3797	0.0380	0.0380	0	0	0	0	0	0	0.0380	Saraf
SML17	0	0.4889	0.0000	0.0667	0.0444	0	0	0	0	0	0	Saraf
SML18	0	0.3944	0.0704	0.0704	0.0282	0	0	0	0	0	0	Saraf
SML19	0	0.5306	0.0612	0.0204	0	0	0	0	0	0	0	Saraf
SML20	0	0.4138	0.0345	0.1034	0	0	0	0	0.0690	0	0	Saraf
SML21	0	0.5909	0.0455	0	0	0	0	0	0	0	0	Saraf
SML22	0	0.5152	0.0909	0.0303	0.0000	0.0000	0	0	0.0303	0	0	Saraf
SML23	0	0.5000	0.0714	0.0714	0.0000	0.0000	0	0	0	0	0	Saraf
SML24	0	0.3958	0.1042	0.0833	0.0000	0.0000	0	0	0	0	0	Jantung

Tabel 4.4 Data Uji Manualisasi Hasil *Preprocessing*

Data	B	C	N	O	OH	P	S	F	CL	Br	I	Farmakologi
SM1	0	0.5333	0.0222	0.0444	0.0444	0	0	0	0	0	0	saraf
SM2	0	0.6333	0.0333	0.0000	0.0333	0	0	0	0	0	0	saraf
SM3	0	0.5862	0.0345	0.0345	0	0	0	0	0	0	0	saraf
SM4	0	0.5111	0	0.0889	0	0	0	0	0.0222	0	0	Jantung
SM5	0	0.1892	0.0811	0.1081	0	0	0	0	0.0270	0	0	Jantung
SM6	0	0.3111	0.0444	0.0607	0.0222	0	0	0	0.0222	0	0	Jantung

Langkah 2 Menentukan nilai K

Nilai K yang ditentukan dalam perhitungan manual adalah 2.

Langkah 3 Melakukan perhitungan jarak *Euclidean* pada data latih

Perhitungan jarak *Euclidean* menggunakan rumus Persamaan 2.1 pada data latih yang dilakukan untuk menentukan tetangga terdekat sebanyak K yang kemudian digunakan dalam perhitungan nilai *Validitas*. Perhitungan jarak *Euclidean* abtar data latih yaitu sebagai berikut:

$$d_{x,y} = \sqrt{\sum_{i=0}^{n}(x_{ij} - y_{ij})^2}$$

$$d_{SML1.SML2} = \sqrt{\begin{matrix}(0-0)^2 + (0.25-0.2727)^2 + (0-0.1818)^2 + (0.125-0.0455)^2 + \\ (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 \\ +(0-0)^2 + (0-0)^2\end{matrix}}$$

$$= 0.2150$$

Pada Tabel 4.5 menunjukkan hasil perhitungan jarak *Euclidean* dari setiap data latih.

Tabel 4.5 Hasil Perhitungan Jarak *Euclidean* Antar Data Latih

Data	SML1	SML2	SML3	SML4	SML5	SML6	SML7	.....	SML22	SML23	SML24
SML1	0	0.215	0	0.1197	0.0612	0.1822	0.0905		0.3227	0.2934	0.2224
SML2	0.215	0	0.215	0.1169	0.1544	0.1044	0.1464		0.2651	0.2581	0.1571
SML3	0	0.215	0	0.1197	0.0612	0.1822	0.0905		0.3227	0.2934	0.2224
SML4	0.1197	0.1169	0.1197	0	0.0673	0.1188	0.0871		0.2824	0.2616	0.1653
SML5	0.0612	0.1544	0.0612	0.0673	0	0.1362	0.0527		0.2963	0.2716	0.1857
SML6	0.1822	0.1044	0.1822	0.1188	0.1362	0	0.125		0.1796	0.1655	0.078
SML7	0.0905	0.1464	0.0905	0.0871	0.0527	0.125	0		0.287	0.2688	0.19
.....											
SML22	0.3227	0.2651	0.3227	0.2824	0.2963	0.1796	0.287		0	0.0567	0.1348
SML23	0.2934	0.2581	0.2934	0.2616	0.2716	0.1655	0.2688		0.0567	0	0.1099
SML24	0.2224	0.1571	0.2224	0.1653	0.1857	0.078	0.19		0.1348	0.1099	0

Setelah jarak *Euclidean* dari semua data latih dihasilkan, langkah selanjutnya adalah menggambil 2 jarak terdekat dari masing-masing data latih. Kedua jarak tersebut menentukan kesamaan kelas dari setiap data latih seperti pada Tabel 4.6 sampai dengan Tabel 4.29. Setiap kelas pada data latih ditentukan kesamaannya dengan aturan jika kelas dengan jarak terdekat memiliki kesamaan maka diberikan nilai 1 dan jika tidak sama maka diberikan nilai 0.

Tabel 4.6 Hasil Perhitungan Jarak *Euclidean* Terdekat Data Latih SML1

D(x,y)	Jarak	Kelas(x)	Kelas(y)	Kesamaan dengan kelas data SML1
SML1,SML3	0	Saraf	Saraf	1
SML1,SML5	0.0612	Saraf	Saraf	1

Tabel 4.7 Hasil Perhitungan Jarak *Euclidean* Terdekat Data Latih SML2

D(x,y)	Jarak	Kelas(x)	Kelas(y)	Kesamaan dengan kelas data SML2
SML2,SML6	0.1044	Saraf	Saraf	1
SML2,SML4	0.1169	Saraf	Saraf	1

Tabel 4.8 Hasil Perhitungan Jarak *Euclidean* Terdekat Data Latih SML3

D(x,y)	Jarak	Kelas(x)	Kelas(y)	Kesamaan dengan kelas data SML3
SML3,SML1	0	Saraf	Saraf	1
SML3,SML5	0.0612	Saraf	Saraf	1

Tabel 4.9 Hasil Perhitungan Jarak *Euclidean* Terdekat Data Latih SML4

D(x,y)	Jarak	Kelas(x)	Kelas(y)	Kesamaan dengan kelas data SML4
SML4,SML5	0.0673	Saraf	Saraf	1
SML4,SML7	0.0871	Saraf	Saraf	1

Tabel 4.10 Hasil Perhitungan Jarak *Euclidean* Terdekat Data Latih SML5

D(x,y)	Jarak	Kelas(x)	Kelas(y)	Kesamaan dengan kelas data SML5
SML5,SML7	0.0527	Saraf	Saraf	1
SML5,SML1	0.0612	Saraf	Saraf	1

Tabel 4.11 Hasil Perhitungan Jarak *Euclidean* Terdekat Data Latih SML6

D(x,y)	Jarak	Kelas(x)	Kelas(y)	Kesamaan dengan kelas data SML6
SML6,SML18	0.0709	Saraf	Saraf	1
SML6,SML24	0.078	Saraf	Jantung	0

Tabel 4.12 Hasil Perhitungan Jarak *Euclidean* Terdekat Data Latih SML7

D(x,y)	Jarak	Kelas(x)	Kelas(y)	Kesamaan dengan kelas data SML7
SML7,SML5	0.0527	Saraf	Saraf	1
SML7,SML4	0.0871	Saraf	Saraf	1

Tabel 4.13 Hasil Perhitungan Jarak *Euclidean* Terdekat Data Latih SML8

D(x,y)	Jarak	Kelas(x)	Kelas(y)	Kesamaan dengan kelas data SML8
SML8,SML10	0.0956	Saraf	Jantung	0
SML8,SML5	0.1269	Saraf	Saraf	1

Tabel 4.14 Hasil Perhitungan Jarak *Euclidean* Terdekat Data Latih SML9

D(x,y)	Jarak	Kelas(x)	Kelas(y)	Kesamaan dengan kelas data SML9
SML9,SML17	0.0587	Jantung	Saraf	0
SML9,SML23	0.0776	Jantung	Saraf	0

Tabel 4.15 Hasil Perhitungan Jarak *Euclidean* Terdekat Data Latih SML10

D(x,y)	Jarak	Kelas(x)	Kelas(y)	Kesamaan dengan kelas data SML10
SML10,SML8	0.0956	Jantung	Saraf	0
SML10,SML4	0.1034	Jantung	Saraf	0

Tabel 4.16 Hasil Perhitungan Jarak *Euclidean* Terdekat Data Latih SML11

D(x,y)	Jarak	Kelas(x)	Kelas(y)	Kesamaan dengan kelas data SML11
SML11,SML14	0.0372	Jantung	Jantung	1
SML11,SML13	0.0435	Jantung	Jantung	1

Tabel 4.17 Hasil Perhitungan Jarak *Euclidean* Terdekat Data Latih SML12

D(x,y)	Jarak	Kelas(x)	Kelas(y)	Kesamaan dengan kelas data SML12
SML12,SML15	0.1132	Jantung	Jantung	1
SML12,SML24	0.1281	Jantung	Jantung	1

Tabel 4.18 Hasil Perhitungan Jarak *Euclidean* Terdekat Data Latih SML13

D(x,y)	Jarak	Kelas(x)	Kelas(y)	Kesamaan dengan kelas data SML13
SML13,SML14	0.0101	Jantung	Jantung	1
SML13,SML11	0.0433	Jantung	Jantung	1

Tabel 4.19 Hasil Perhitungan Jarak *Euclidean* Terdekat Data Latih SML14

D(x,y)	Jarak	Kelas(x)	Kelas(y)	Kesamaan dengan kelas data SML14
SML14,SML13	0.0101	Jantung	Jantung	1
SML14,SML11	0.0372	Jantung	Jantung	1

Tabel 4.20 Hasil Perhitungan Jarak *Euclidean* Terdekat Data Latih SML15

D(x,y)	Jarak	Kelas(x)	Kelas(y)	Kesamaan dengan kelas data SML15
SML15,SML22	0.0779	Jantung	Saraf	0
SML15,SML19	0.1003	Jantung	Saraf	0

Tabel 4.21 Hasil Perhitungan Jarak *Euclidean* Terdekat Data Latih SML16

D(x,y)	Jarak	Kelas(x)	Kelas(y)	Kesamaan dengan kelas data SML16
SML16,SML18	0.0675	**Saraf**	Saraf	1
SML16,SML11	0.0894	**Saraf**	Jantung	0

Tabel 4.22 Hasil Perhitungan Jarak *Euclidean* Terdekat Data Latih SML17

D(x,y)	Jarak	Kelas(x)	Kelas(y)	Kesamaan dengan kelas data SML17
SML17,SML9	0.0587	**Saraf**	Jantung	0
SML17,SML23	0.0849	**Saraf**	Saraf	1

Tabel 4.23 Hasil Perhitungan Jarak *Euclidean* Terdekat Data Latih SML18

D(x,y)	Jarak	Kelas(x)	Kelas(y)	Kesamaan dengan kelas data SML18
SML18,SML24	0.0459	**Saraf**	Jantung	0
SML18,SML16	0.0675	**Saraf**	Saraf	1

Tabel 4.24 Hasil Perhitungan Jarak *Euclidean* Terdekat Data Latih SML19

D(x,y)	Jarak	Kelas(x)	Kelas(y)	Kesamaan dengan kelas data SML19
SML19,SML22	0.0462	**Saraf**	Saraf	1
SML19,SML23	0.0603	**Saraf**	Saraf	1

Tabel 4.25 Hasil Perhitungan Jarak *Euclidean* Terdekat Data Latih SML20

D(x,y)	Jarak	Kelas(x)	Kelas(y)	Kesamaan dengan kelas data SML20
SML20,SML18	0.0912	Saraf	Saraf	1
SML20,SML24	0.1017	Saraf	Saraf	1

Tabel 4.26 Hasil Perhitungan Jarak *Euclidean* Terdekat Data Latih SML21

D(x,y)	Jarak	Kelas(x)	Kelas(y)	Kesamaan dengan kelas data SML21
SML21,SML19	0.0656	Saraf	Saraf	1
SML21,SML22	0.0981	Saraf	Saraf	1

Tabel 4.27 Hasil Perhitungan Jarak *Euclidean* Terdekat Data Latih SML22

D(x,y)	Jarak	Kelas(x)	Kelas(y)	Kesamaan dengan kelas data SML22
SML22,SML19	0.0462	Saraf	Saraf	1
SML22,SML23	0.0567	Saraf	Saraf	1

Tabel 4.28 Hasil Perhitungan Jarak *Euclidean* Terdekat Data Latih SML23

D(x,y)	Jarak	Kelas(x)	Kelas(y)	Kesamaan dengan kelas data SML23
SML23,SML22	0.0567	Saraf	Saraf	1
SML23,SML19	0.0603	Saraf	Saraf	1

Tabel 4.29 Hasil Perhitungan Jarak *Euclidean* Terdekat Data Latih SML24

D(x,y)	Jarak	Kelas(x)	Kelas(y)	Kesamaan dengan kelas data SML24
SML24,SML18	0.0459	Jantung	Saraf	0
SML24,SML6	0.078	Jantung	Saraf	0

Langkah 4 Melakukan perhitungan nilai *validitas* pada data latih

Pada Persamaan 2.1 setelah didapatkan 2 jarak tetangga terdekat masing-masing data latih serta nilai kesamaan kelasnya lalu dilakukan perhitungan *Validitas* sebagai berikut:

$$Validitas(x) = \frac{1}{K}\sum_{i-1}^{K} S(lbl(x), (lbl(N_i(x))))$$

$$Validitas(SML1) = \frac{1}{2}\sum_{i-1}^{3} S(lbl(x), (lbl(N_i(x))))$$

$$= \frac{1}{2}(1+1)$$

$$= 1$$

Pada Tabel 4.6 kemudian digunakan dalam perhitungan nilai *Validitas* menggunakan persamaan 2.2. Dimana $\mathrm{lbl(x)}$ merupakan kelas dari data latih SML1 sedangkan $\mathrm{lbl(Ni(x))}$ merupakan kelas dari data terdekat pada data latih 1. Tabel 4.30 adalah hasil perhitungan nilai *Validitas* untuk seluruh data latih.

Tabel 4.30 Hasil Perhitungan Nilai Validitas

Data	Kesamaan kelas dengan tetangga ke-i		SUM nilai kesaman	Validitas
	k=1	k=2		
SML1	1	1	2	1.0000
SML2	1	1	2	1.0000
SML3	1	1	2	1.0000
SML4	1	1	2	1.0000
SML5	1	1	2	1.0000
SML6	1	0	1	0.5000
SML7	1	1	2	1.0000
SML8	0	1	1	0.5000
SML9	0	0	0	0.0000
SML10	0	0	0	0.0000

SML11	1	1	2	1.0000
SML12	1	1	2	1.0000
SML13	1	1	2	1.0000
SML14	1	1	2	1.0000
SML15	0	0	0	0.0000
SML16	1	0	1	0.5000
SML17	0	1	1	0.5000
SML18	0	1	1	0.5000
SML19	1	1	2	1.0000
SML20	1	0	1	0.5000
SML21	1	1	2	1.0000
SML22	1	1	2	1.0000
SML23	1	1	2	1.0000
SML24	0	0	0	0.0000

Langkah 5 melakukan perhitungan jarak *Euclidean* antara data latih dan data uji

Perhitungan jarak *Euclidean* menggunakan rumus Persamaan 2.1 pada data latih dan data uji untuk mendapatkan hasil jarak yang selanjutnya digunakan dalam perhitungan nilai *Weight voting*. Perhitungan jarak tersebut secara rinci adalah sebagai berikut:

$$d_{x,y} = \sqrt{\sum_{i=0}^{n}(x_{ij} - y_{ij})^2}$$

$$d_{SM1.SML1} = \sqrt{\begin{array}{c}(0-0)^2 + (0.5333-0.25)^2 + (0.0222-0)^2 + (0.0444-0.125)^2 + \\ (0.0444-0.125)^2 + (0-0)^2 + (0-0)^2 + (0-0)^2 + \\ +(0-0)^2 + (0-0)^2\end{array}}$$

$$= 0.3062$$

Pada Tabel 4.31 ditunjukkan hasil perhitungan jarak *Euclidean* dari setiap data latih dengan data uji

Tabel 4.31 Hasil Perhitungan Jarak *Euclidean* antar Data Latih dan Data Uji

	SM1	SM2	SM3	SM4	SM5	SM6
SML1	0.3062	0.4148	0.3715	0.2926	0.1641	0.1446
SML2	0.3056	0.3928	0.3495	0.3071	0.1544	0.1470
SML3	0.3062	0.4148	0.3715	0.2926	0.1641	0.1446
SML4	0.2959	0.3987	0.3513	0.2851	0.1034	0.0980
SML5	0.2953	0.4021	0.3572	0.2851	0.1243	0.1089
SML6	0.2051	0.2997	0.2557	0.2091	0.1864	0.0946
SML7	0.2854	0.3872	0.3492	0.2880	0.1626	0.1239
SML8	0.4128	0.5196	0.4694	0.3925	0.0956	0.1996

SML9	0.0737	0.1598	0.1014	0.0000	0.3325	0.2080
SML10	0.3587	0.4616	0.4073	0.3325	0.0000	0.1377
SML11	0.2266	0.3302	0.2789	0.2073	0.1358	0.0060
SML12	0.2017	0.2651	0.2263	0.2040	0.2592	0.1642
SML13	0.2516	0.3539	0.2989	0.2261	0.1085	0.0471
SML14	0.2420	0.3441	0.2892	0.2168	0.1182	0.0408
SML15	0.1476	0.1944	0.1582	0.1733	0.3140	0.2040
SML16	0.1652	0.2614	0.2100	0.1524	0.2127	0.0877
SML17	0.0544	0.1629	0.1169	0.0587	0.3175	0.1860
SML18	0.1502	0.2519	0.2004	0.1421	0.2125	0.0908
SML19	0.0638	0.1134	0.0633	0.0965	0.3541	0.2260
SML20	0.1570	0.2544	0.1981	0.1143	0.2332	0.1231
SML21	0.0883	0.0553	0.0365	0.1297	0.4184	0.2880
SML22	0.0902	0.1422	0.0957	0.1085	0.3353	0.2128
SML23	0.0789	0.1595	0.1008	0.0776	0.3143	0.1937
SML24	0.1706	0.2636	0.2085	0.1571	0.2111	0.1107

Langkah 6 Melakukan perhitungan *Weight voting*

Pada proses ini, nilai *validitas* pada setiap data latih dan jarak *Euclidean* pada Langkah 4 digunakan sebagai parameter perhitungan *Weight voting*. Perhitungan secara rinci dari Persamaan 2.4 sebagai berikut:

$$W(x) = Validitas(x)\frac{1}{d(i)+0{,}5}$$

$$W(SM1) = Validitas(SM1)\frac{1}{d(i)+0{,}5}$$

$$W(SM1) = 1\frac{1}{0.3062+0{,}5}$$

$$= 1.2404$$

Perhitungan nilai *Weight voting* secara keseluruhan dapat dilihat pada Tabel 4.32.

Tabel 4.32 Hasil Perhitungan *Weight voting*

data	SM1	SM2	SM3	SM4	SM5	SM6
SML1	1.2404	1.0931	1.1474	1.2617	1.5059	1.5514
SML2	1.2413	1.1201	1.1771	1.2389	1.5282	1.5455
SML3	1.2404	1.0931	1.1474	1.2617	1.5059	1.5514
SML4	1.2564	1.1127	1.1746	1.2737	1.6573	1.6721
SML5	1.2574	1.1086	1.1666	1.2737	1.6017	1.6424
SML6	0.7091	0.6252	0.6616	0.7051	0.7284	0.8410
SML7	1.2732	1.1271	1.1776	1.2691	1.5093	1.6029
SML8	0.5478	0.4904	0.5158	0.5602	0.8394	0.7147

SML9	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
SML10	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
SML11	1.3763	1.2046	1.2838	1.4139	1.5729	1.9763
SML12	1.4252	1.3071	1.3769	1.4204	1.3172	1.5055
SML13	1.3305	1.1712	1.2517	1.3771	1.6433	1.8279
SML14	1.3478	1.1847	1.2671	1.3950	1.6176	1.8492
SML15	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
SML16	0.7516	0.6567	0.7042	0.7664	0.7016	0.8508
SML17	0.9018	0.7543	0.8105	0.8949	0.6116	0.7288
SML18	0.7690	0.6650	0.7139	0.7786	0.7017	0.8464
SML19	1.7736	1.6304	1.7753	1.6765	1.1709	1.3774
SML20	0.7611	0.6627	0.7163	0.8140	0.6819	0.8024
SML21	1.6997	1.8009	1.8639	1.5879	1.0889	1.2690
SML22	1.6943	1.5572	1.6787	1.6433	1.1972	1.4028
SML23	1.7273	1.5164	1.6645	1.7313	1.2281	1.4416
SML24	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

Langkah 7 Menentukan kelas dari data uji

Pada proses ini, sebelum menentukan kelas baru dari data uji nilai *weigh voting* pada langkah 5 diurutkan terlebih dahulu dari nilai terbesar hingga nilai terkecil. Berdasarkan hasil pengurutan tersebut, untuk menentukan kelas dari data uji diperlukan perhitungan jumlah *weight* dari K- tetangga terdekat untuk masing-masing kelas kemudian kelas dengan nilai tertinggi merupakan kelas baru dari data uji. Hasil pengurutan nilai *weight* setiap data uji dapat dilihat pada Tabel 4.33 sampai 4.38.

Tabel 4.33 Nilai *Weight voting* Tertinggi Data Uji SM1

Data	SM1	kelas
SML19	1.7734	Saraf
SML23	1.7272	Saraf

Tabel 4.34 Nilai *Weight voting* Tertinggi Data Uji SM2

Data	SM2	kelas
SML21	1.8008	Saraf
SML19	1.6303	Saraf

Tabel 4.35 Nilai Weight voting Tertinggi Data Uji SM3

Data	SM3	kelas
SML21	1.8639	Saraf
SML19	1.7753	Saraf

Tabel 4.36 Nilai *Weight voting* Tertinggi Data Uji SM4

Data	SM4	kelas
SML23	1.7313	Saraf
SML19	1.6764	Saraf

Tabel 4.37 Nilai *Weight voting* Tertinggi Data Uji SM5

Data	SM5	kelas
SML4	1.6573	Saraf
SML13	1.6434	Jantung

Tabel 4.38 Nilai *Weight voting* Tertinggi Data Uji SM6

Data	SM6	kelas
SML11	2	Jantung
SML14	1.8615	Jantung

Berdasarkan hasil pengurutan diatas, berikut Ini adalah penentuan kelas dari data uji SM1:

$$jumlah\ weight\ kelas\ Saraf\ =\ weight(SML19) + weight(SML23)$$
$$= 3.5006$$
$$jumlah\ weight\ kelas\ Jantung\ =\ 0$$

melihat perhitungan diatas, jumlah *weight* terbesar adalah kelas Saraf sehingga dapat diprediksi bahwa kelas baru untuk data uji SM1 adalah kelas Saraf. Berikut hasil keseluruhan perhitungan penentuan kelas baru untuk data uji ditunjukkan pada Tabel 4.39.

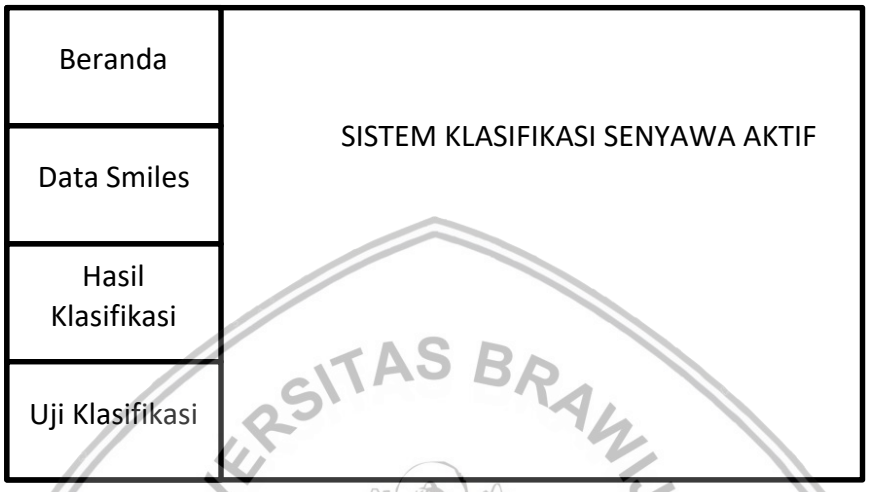
Tabel 4.39 Penentuan Kelas Prediksi

Data	Jumlah weight		Kelas Sebenarnya	Kelas Prediksi
	Saraf	Jantung		
SM1	3.5006	0	Saraf	Saraf
SM2	3.4311	0	Saraf	Saraf
SM3	3.6393	0	Saraf	Saraf
SM4	3.4077	0	Jantung	Saraf
SM5	1.6573	1.6434	Jantung	Saraf
SM6	0	3.8615	Jantung	Jantung
Akurasi			4/6	
%			67	

4.4 Perancangan Antarmuka

4.4.1 Halaman Beranda

Halaman beranda menggambarkan tampilan awal ketika sistem mulai dijalankan. Halaman ini digambarkan pada ilustrasi Gambar 4.7,

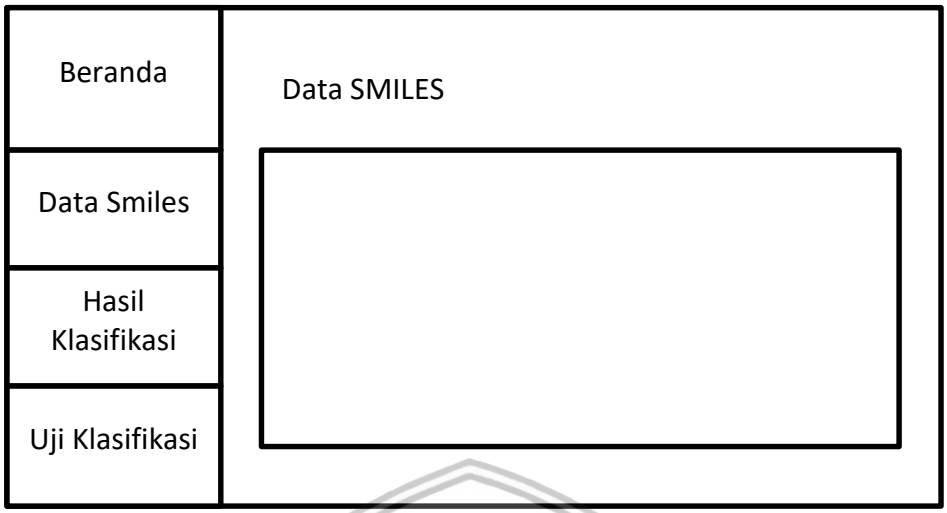


Gambar 4.7 Ilustrasi Halaman Beranda

dimana pada perancangan tampilan di halaman awal terdapat beberapa bagian yaitu: *Menu bar* yang berisi pilihan menu untuk diakses pada sistem yang dibangun. menu terdiri dari menu beranda, menu Data *SMILES* yang akan menampilkan seluruh data yang digunakan dalam sistem, menu hasil klasifikasi yang akan menampilkan hasil klasifikasi dengan data yang sudah disiapkan sistem dan menu uji klasifikasi Dimana user bisa menguji sebuah data untuk mendapatkan hasil prediksinya.

4.4.2 Halaman Data

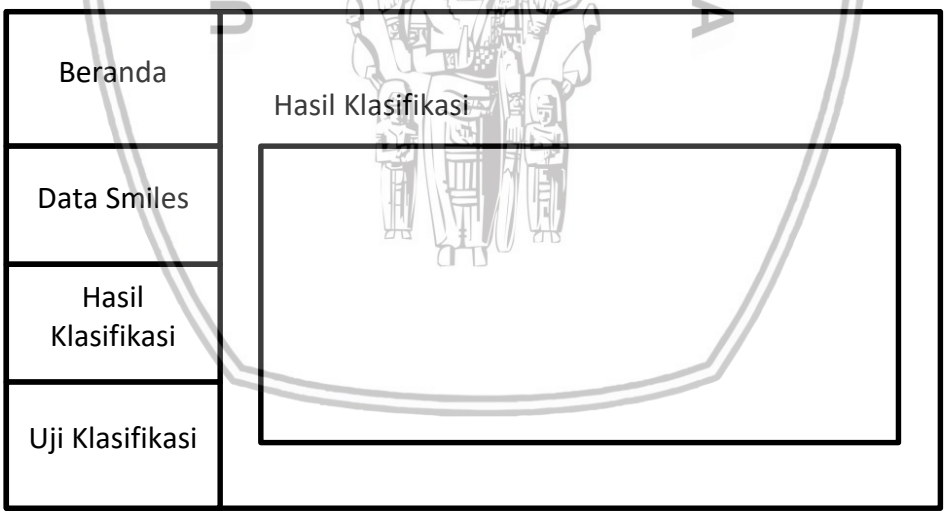
Pada halaman berikutnya terdapat perancangan halaman untuk menampilkan data latih. Perancangan dari halaman tersebut dapat dilihat pada Gambar 4.8.



Gambar 4.8 Ilustrasi Halaman Data *SMILES*

4.4.3 Halaman Hasil Klasifikasi

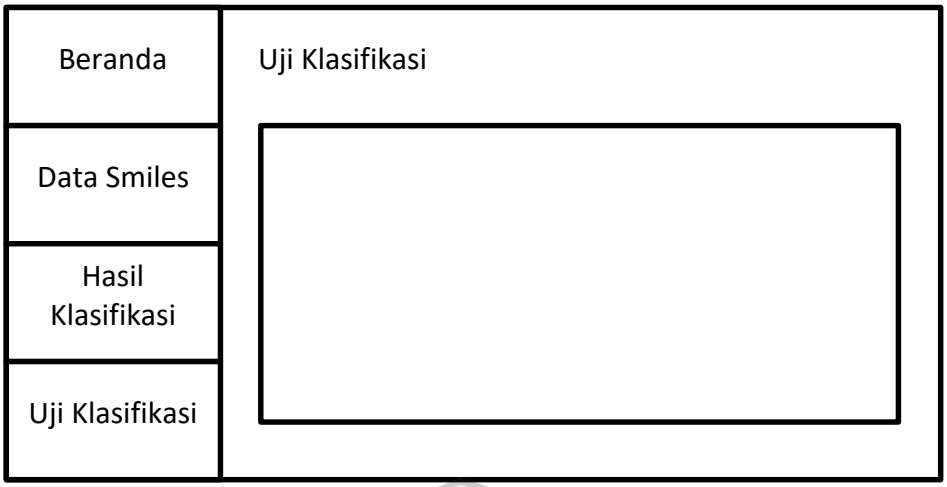
Halaman hasil klasifikasi menampilkan hasil klasifikasi dengan data latih dan data uji yang disiapkan sistem dengan user memasukkan nilai K. Rancangan dari halaman hasil klasifikasi ditunjukkan pada Gambar 4.9.



Gambar 4.9 Ilustrasi Halaman Hasil Klasifikasi

4.4.4 Halaman Uji Klasifikasi

Halaman uji klasifikasi menampilkan hasil dari uji klasifikasi dengan data latih yang disiapkan sistem namun dengan masukan data uji dan nilai K dari user. Rancangan dari halaman hasil klasifikasi ditunjukkan pada Gambar 4.10.



Gambar 4.10 Ilustrasi Halaman Uji Klasifikasi

4.5 Perancangan Pengujian

Pada tahap ini dilakukan pengujian dengan kondisi yang berbeda-beda untuk menguji kemampuan metode *Modified k-nearest neighbor* dalam melakukan proses klasifikasi. Pada perancangan pengujian ini terdapat tiga kondisi pengujian yang berbeda yaitu pengujian berdasarkan variasi nilai k, variasi jumlah data latih yang digunakan dan pengujian *cross validation*. Hasil akhir dari pengujian ini dilihat berapa nilai akurasi yang didapat.

4.5.1 Pengujian *Validasi* Program

Pengujian Akurasi Program dimaksudkan untuk mengetahui apakah keluaran yang dihasilkan oleh program mengeluarkan akurasi yang sama dengan keluaran yang dihasilkan dari perhitungan manualisasi. Dimana pada proses perhitungan manual menghasilkan nilai akurasi dengan jumlah data benar 4.

4.5.2 Perancangan Pengujian Variasi Nilai k

Skenario pengujian yang pertama yaitu adalah pengujian dengan variasi inputan nilai k yang berbeda. Pada perancangan pengujian dengan variasi nilai K terdapat beberapa kondisi yaitu jika nilai k=2, k=3, k=5, k=7 dan k=9 dengan menggunakan seluruh data latih dan data uji dengan pembagian 90% data latih atau 234 data dan 10% data uji atau sebanyak 26 data.

4.5.3 Perancangan Pengujian *Holdout* Validation

Scenario pengujian yang kedua adalah variasi pengaruh jumlah data latih terhadap akurasi. Pada pengujian ini 260 dataset yang digunakan terbagi menjadi empat kondisi yaitu menggunakan 90%, 80%, 70%, dan 60% dari data latih dan 10%, 20%, 30% dan 40% dari data uji dengan menggunakan nilai K terbaik dari pengujian sebelumnya.

4.5.4 Perancangan Pengujian *K- Fold Cross Validation*

Pengujian *k-fold cross validation* dilakukan untuk mengetahui akurasi program jika diuji dengan menggunakan sampel data yang acak. Teknik ini menggunakan dataset yang dibagi menjadi sejumlah K-buah partisi secara acak dan dilakukan sejumlah K-kali eksperimen, dimana masing-masing eksperimen menggunakan data partisi ke-K sebagai data testing dan memanfaatkan sisa partisi lainnya sebagai data training. Pada metode ini dataset dibagi menjadi sebanyak 4 kelompok data yang berjumlah sama yaitu 65 buah data. Pada setiap kelompok data memiliki fungsi yang berganti sebagai data latih atau data uji. Pengujian ini bertujuan agar setiap data memiliki kesempatan sebagai data latih maupun data uji.

BAB 5 IMPLEMENTASI

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 Keras

Spesifikasi perangkat keras dalam proses klasifikasi fungsi senyawa menggunakan metode *fuzzy k-nearest neighbor* ditunjukkan pada Tabel 5.1. dan spesifikasi *minimum* perangkat keras agar sistem dapat berjalan ditunjukkan pada Tabel 5.2.

Tabel 5.1 Spesifikasi Perangkat Keras yang Digunakan

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

Tabel 5.2 Spesifikasi Kebutuhan *Minimum* Perangkat Keras

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

5.1.2 Spesifikasi Perangkat Lunak

Penelitian ini menggunakan perangkat lunak untuk mengimplementasikan metode *modified k-nearest neighbor* untuk klasifikasi fungsi senyawa. Spesifikasi perangkat lunak yang digunakan pada penelitian ini ditunjukkan pada Tabel 5.3. dan spesifikasi *minimum* dari sistem agar dapat berjalan ditunjukkan pada Tabel 5.4.

Tabel 5.3 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 5.4 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

5.2 Implementasi Program

Sistem diimplementasi menggunakan Metode modified K-Nearest Neighbor dimana pada Metode ini terdapat berbagai langkah penyelesaian. Masing-masing langkah pada Metode Moedified K-Nearest Neighbor dijabarkan pada sub bab berikut,

5.2.1 Proses *Euclidean*

Proses *Euclidean* merupakan proses perhitungan jarak terdekat antar data latih dan data uji. Proses ini melibatkan nilai k yang diinputkan sebelumnya untuk mendapatkan nilai *Euclidean* terdekat setiap data. Potongan kode program untuk proses perhitungan *Euclidean* dapat dilihat pada Tabel 5.5.

Tabel 5.5 Kode Program *Euclidean*

	Kode Program
1	`$dist = array();`
2	`for ($i=0; $i < count($table); $i++) {`
	`$arr = array();`
3	`for ($j=0; $j < count($table); $j++) {`
	`$jarak = 0;`
	`for ($k=0; $k < count($table[$i]); $k++) {`
4	`$temp = round($table[$i][$k], 4) - round($table[$j][$k], 4);`
5	`$temp = pow($temp, 2);`
	`$jarak = $jarak + $temp;}`
	`$jarak = sqrt($jarak);`
6	`array_push($arr, $jarak);}`
	`array_push($dist, $arr);`

5.2.2 Proses validitas

Proses Validitas merupakan proses perhitungan jumlah validitas data pada setiap kelas datanya. Proses ini melibatkan nilai k yang diinputkan sebelumnya untuk mendapatkan nilai Euclidean terdekat setiap data. Dari setiap jarak

terdekat yang didapatkan ditentukan kesamaan kelas datanya. Nilai kesamaan tersebut kemudian dijumlahkan dan dibagi dengan nilai k yang diinputkan. Potongan kode program untuk proses perhitungan Euclidean dapat dilihat pada Tabel 5.6.

Tabel 5.6 Kode Program Validitas

No	Kode Program
1	$klasifikasi_tetangga = array();
2	if (isset($_POST["input"])) {
3	for ($i=0; $i < count($tetangga); $i++) {
4	$keys = array_keys($tetangga[$i]);
5	$temp = array();
6	for ($j=0; $j < $_POST["input"]; $j++) {
7	$sub = substr($keys[$j], 3);
8	array_push($temp, $klas[$sub-1]); }
9	array_push($klasifikasi_tetangga, $temp);
10	} }
11	$validitas = array();
12	for ($i=0; $i < count($klasifikasi_tetangga); $i++) {
13	echo "<tr>";
14	echo "<td>";
15	echo "SML" .($i+1);
16	echo "</td>";
17	$total = 0;
18	for ($j=0; $j < count($klasifikasi_tetangga[$i]); $j++) {
19	echo "<td>";
20	if ($klasifikasi_tetangga[$i][$j] == $klas[$i])
21	{
22	echo "1"; $total++;
23	} else { echo "0"; }

5.2.3 Proses *Weight voting*

Proses *Weight Voting* merupakan proses perhitungan bobot KNN pada *Euclidean* data uji dan data latih. Proses ini melibatkan nilai k yang diinputkan sebelumnya untuk mendapatkan nilai Euclidean terdekat setiap data. Nilai weight voting yang didapatkan dari perkalian dengan nilai validitas selanjutnya diurutkan dan diambil nilai terbesar sebanyak k inputan. Penjumlahan nilai bobot

masing-masing kelas klasifikasi tersebut merupakan hasil akhir dari proses klasifikasi. Potongan kode program untuk proses perhitungan Euclidean dapat dilihat pada Tabel 5.7.

Tabel 5.7 Kode Program *Weight voting*

No	Kode Program
1	$jarakUji = array();
2	for ($i=0; $i < count($table2); $i++) {
3	$temp = array();
4	for ($j=0; $j < count($table); $j++) {
5	$total = 0;
6	for ($k=0; $k < count($table2[$i])-1; $k++) {
7	$total += ($table2[$i][$k] - $table[$j][$k])*($table2[$i][$k] - $table[$j][$k]); }
8	$temp[$j] = $total; }
9	array_push($jarakUji, $temp); }
10	for ($j=0; $j < count($jarakUji); $j++) {
11	$jarakUji[$j][$i] = $validitas[$i]/($jarakUji[$j][$i]+0.5);
12	echo "<td>";
13	echo round($jarakUji[$j][$i],4);
14	echo "</td>"; }

5.2.4 Proses klasifikasi

Proses klasifikasi merupakan proses akhir dari sistem yang dijalankan. Proses ini melibatkan nilai penjumlahan terbesar dari hasil pembobotan KNN pada proses *weight voting*. Nilai penjumlahan terbesar merupakan kelas klasifikasi untuk data baru yang dimasukkan pada sistem. Potongan kode program untuk proses perhitungan Euclidean dapat dilihat pada Tabel 5.8.

Tabel 5.8 Kode Program Proses Klasifikasi

No	Kode program
1	$Weighted_voting = array();
2	$cobak = 0;
3	for ($i=0; $i < count($jarakUji); $i++) {
4	arsort($jarakUji[$i]);
5	$index = 0;
6	$total_kanker = 0;
7	$total_Metabolisme = 0;
8	foreach ($jarakUji[$i] as $key => $value) {

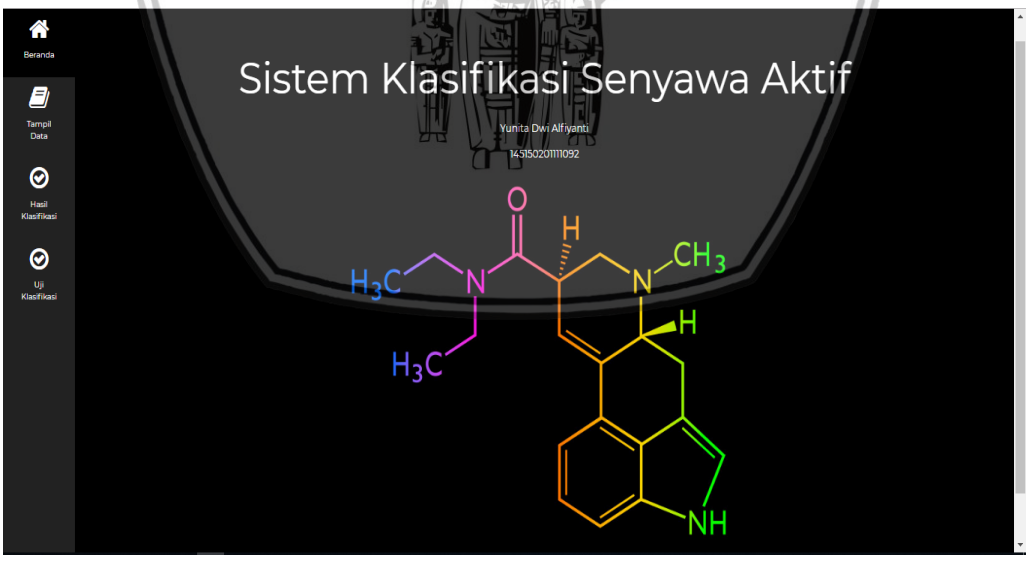
```
9                   if ($index < $_POST["input"]) {
10                      if ($table[$key][count($table[$key])-1] == "Jantung") {
11                          $total_Jantung += $value;
                        } elseif ($table[$key][count($table[$key])-1] == "Saraf") {
12                          $total_Saraf += $value;}}
13                  $index++; }
                $cobak += 1;
14              $temp = array($total_Jantung, $total_Saraf);
15              array_push($Weighted_voting, $temp);
```

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 Beranda

Tampilan antarmuka pada halaman awal menampilkan judul dari sistem yang dibuat, menu bar yang terdiri dari menu Beranda, tampil data, hasil klasifikasi dan uji klasifikasi untuk langsung memulai proses klasifikasi. Hasil dari implementasi antarmuka halaman awal ditunjukkan pada Gambar 5.1.

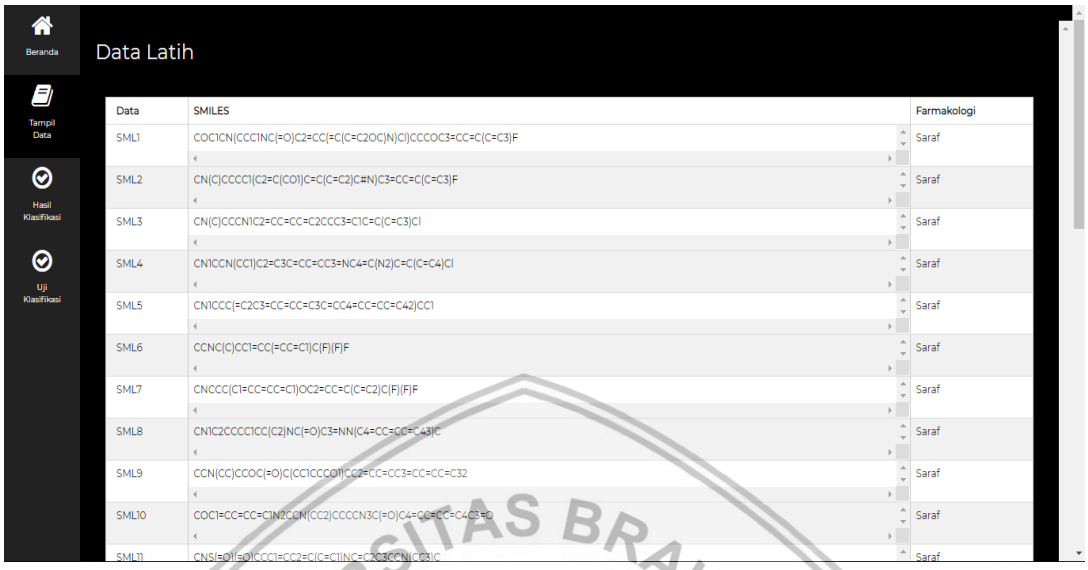


Gambar 5.1 Halaman Beranda

5.3.2 Halaman Tampilan Data

Tampilan antarmuka pada halaman tampilan data latih berisi data-data yang digunakan dalam sistem klasifikasi senyawa aktif yang meliputi data *SMILES*

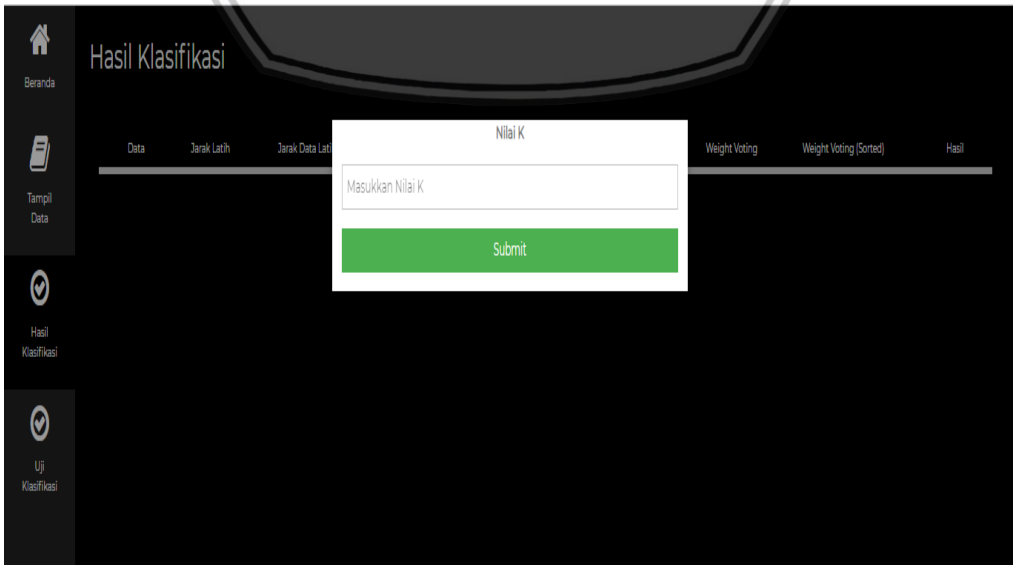
beserta kelas farmakologinya. Hasil implementasi dari antarmuka tersebut dapat dilihat pada Gambar 5.2.



Gambar 5.2 Halaman Tampil Data

5.3.3 Halaman Hasil Klasifikasi

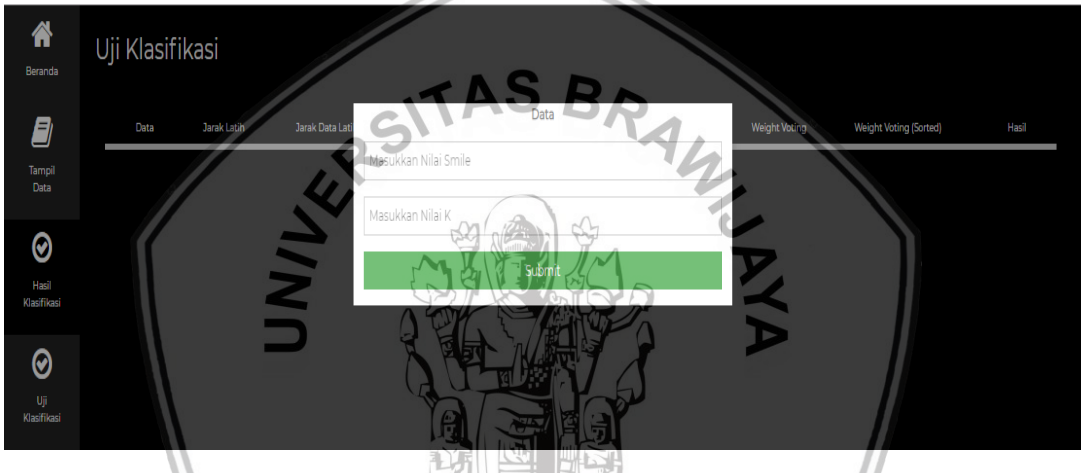
Halaman antarmuka hasil klasifikasi merupakan halaman penting dalam sistem yang dibuat, dimana pada halaman ini terdapat hasil klasifikasi yang merupakan percobaan untuk menguji besar akurasi sistem yang dihasilkan. Untuk melihat proses pada halaman ini user menekan ikon hasil klasifikasi terlebih dahulu. User diminta memasukkan nilai K untuk percobaan tersebut, kemudian sistem akan menampilkan hasil klasifikasi dengan data set dari sistem yang diambil dari database. Hasil implementasi halaman berikut ditunjukkan pada Gambar 5.3.



Gambar 5.3 Halaman Hasil Klasifikasi (Input Nilai K)

5.3.4 Halaman Uji Klasifikasi

Halaman antarmuka hasil uji klasifikasi merupakan halaman penting dalam sistem yang dibuat, dimana pada halaman ini terdapat hasil klasifikasi yang ditujukan untuk pengguna yang ingin mencoba data baru. Untuk melihat proses pada halaman ini user menekan ikon hasil klasifikasi terlebih dahulu. User diminta memasukkan nilai K dan data *SMILES* baru untuk percobaan tersebut, kemudian sistem akan menampilkan hasil klasifikasi dengan data set dari sistem yang diambil dari database. Hasil implementasi halaman berikut ditunjukkan pada Gambar 5.4.



Gambar 5.4 Halaman Uji Klasifikasi (Input *SMILES* dan Nilai K)

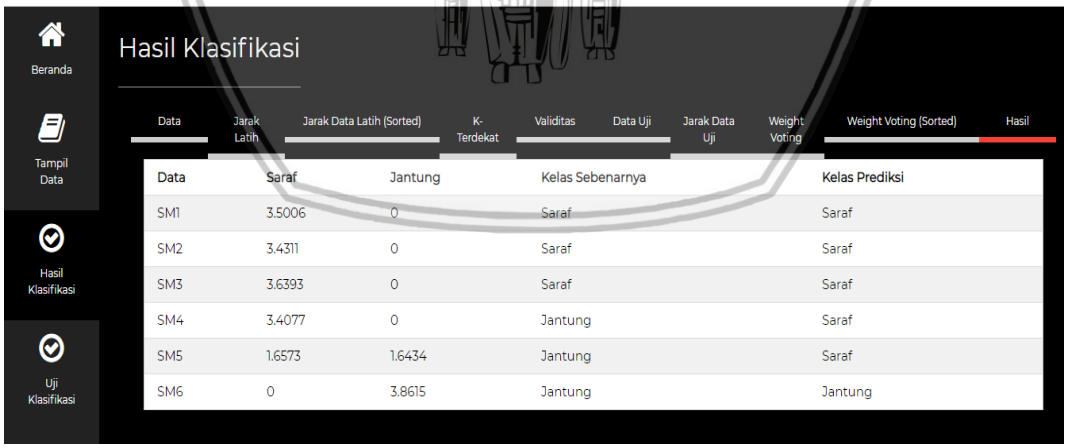
BAB 6 PENGUJIAN DAN ANALISIS

6.1 Pengujian Validasi Sistem

Pengujian akurasi program menggunakan data latih sebanyak 30 dan data uji sebanyak 6. Kelas yang digunakan pada pengujian ini yaitu kelas Saraf dan Jantung. Data pada pengujian ini masing-masing berjumlah 16 data latih Saraf dan 14 data latih Jantung. Data uji yang digunakan pada pengujian ini sama dengan yang digunakan pada perhitungan manualisasi. Pada Tabel 6.1 berikut merupakan perbandingan antara hasil klasifikasi manual dengan hasil klasifikasi sistem.

Tabel 6.1 Perbandingan Hasil Klasifikasi Manual dan Sistem

Data	*SMILES*	Hasil Klasifikasi Manual	Hasil Klasifikasi Sistem
SM1	CC(C(=O)O)O.C1CCN(CC1)CCC(C2CC3CC2C=C3)(C4=CC=CC=C4)O	Saraf	Saraf
SM2	C1CCN(CC1)CCC(C2CCCC2)(C3=CC=CC=C3)O	Saraf	Saraf
SM3	CN(C)CCOC(C1=CC=CC=C1)C2=CC=CC=C2	Saraf	Saraf
SM4	CC(=O)C1(CCC2C1(CCC3C2C=C(C4=CC(=O)CCC34C)Cl)C)OC(=O)C	Saraf	Saraf
SM5	C1=C2C(=CC(=C1Cl)S(=O)(=O)N)S(=O)(=O)N=CN2	Saraf	Saraf
SM6	C1=CC=C2C(=C1)C(=O)NC2(C3=CC(=C(C=C3)Cl)S(=O)(=O)N)O	Jantung	Jantung



Gambar 6.1 Hasil Klasifikasi pada Sistem

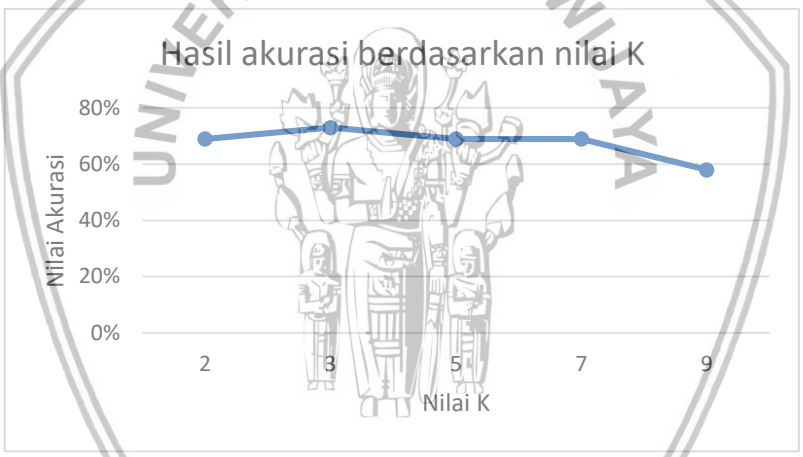
Pada Tabel 6.1 dan Gambar 6.1 menunjukkan hasil keluaran dari perhitungan manual dengan hasil keluaran sistem memiliki nilai akurasi yang sama. Berdasarkan hasil tersebut dapat dihasilkan bahwa program dapat dikatakan valid.

6.2 Pengujian Variasi Nilai k

Pengujian pengaruh nilai k terhadap akurasi menggunakan nilai k yang berbeda. Pada pengujian ini dilakukan 5 kali pengujian dengan menggunakan nilai k sebesar k = 2, k = 3, k= 5, k=7 dan k =9. Tujuan dari penelitian ini adalah untuk mengetahui pengaruh banyaknya nilai tetangga terdekat terhadap nilai akurasi. Dalam scenario pengujian ini digunakan data sebanyak 260 data yang terbagi menjadi latih dan data uji dengan presentase data sebesar 90% yaitu sebanyak 234 data untuk data latih dan 10% yaitu 26 data untuk data uji. Hasil dari pengujian ditunjukkan pada Tabel 6.2

Tabel 6.2 Hasil Akurasi Berdasarkan Nilai k (Jumlah Tetangga Terdekat)

Skenario Ke-	Nilai k	Nilai akurasi
1	2	69%
2	3	73%
3	5	69%
4	7	69%
5	9	58%



Gambar 6.2 Grafik Hasil Pengujian Berdasarkan Nilai K

Dari hasil pengujian variasi nilai K yang ditunjukkan pada Tabel 6.2 dan Gambar 6.2 dapat diketahui bahwa akurasi terbesar diraih pada nilai K=3 yaitu sebesar 73%. 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. 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.

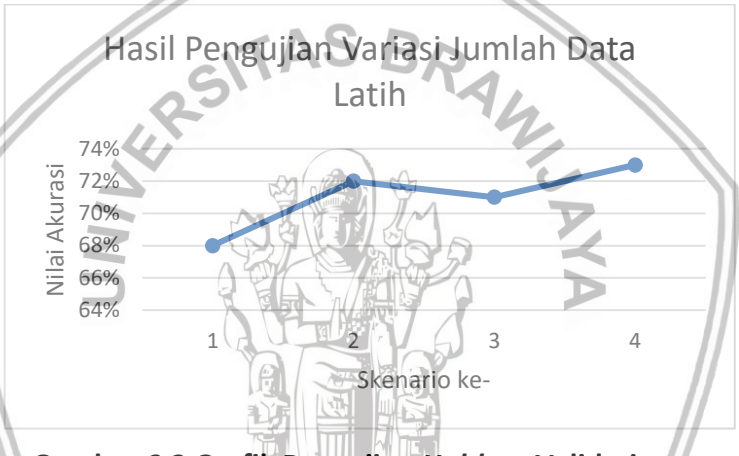
6.3 Pengujian *Holdout Validation* terhadap Data Latih

Pengujian *holdout validation* terhadap data latih digunakan untuk mengetahui apakah jumlah data latih dapat mempengaruhi nilai akurasi. Dalam Metode *holdout*, data awal yang diberi label dibagi ke dalam dua himpunan secara random yang dinamakan data latih dan data uji dengan jumlah keseluruhan data

sebesar 100%. Pada pengujian ini digunakan seluruh data yaitu 260 data dan memiliki presentase jumlah yang berbeda-beda. Skenario pengujian Data latih ini terdapat 4 pengujian dengan jumlah data latih dan data uji yang berbeda-beda, dimana dalam scenario pengujian ini menggunakan nilai *k* dari pengujian sebelumnya yaitu *k*=3. Jumlah masing-masing data untuk pengujian ini dapat dilihat pada Tabel 6.3.

Tabel 6.3 Hasil Pengujian *Holdout* Validation

Skenario Ke-	Jumlah Data Latih	Jumlah Data Uji	Nilai akurasi
1	60%	40%	68%
2	70%	30%	72%
3	80%	20%	71%
4	90%	10%	73%



Gambar 6.3 Grafik Pengujian *Holdout* Validation

Pada Tabel 6.3 dan Gambar 6.3 hasil pengujian variasi jumlah data latih menunjukkan bahwa pada pengujian pertama dengan menggunakan data latih 90% dan data uji sebesar 10% tersebut mendapatkan hasil akurasi mencapai nilai 73%. Berdasarkan penjelasan tersebut dapat disimpulkan bahwa secara umum jika semakin banyak jumlah data latih yang digunakan pada proses klasifikasi maka nilai akurasi semakin tinggi karena sistem melakukan proses pembelajaran lebih banyak.

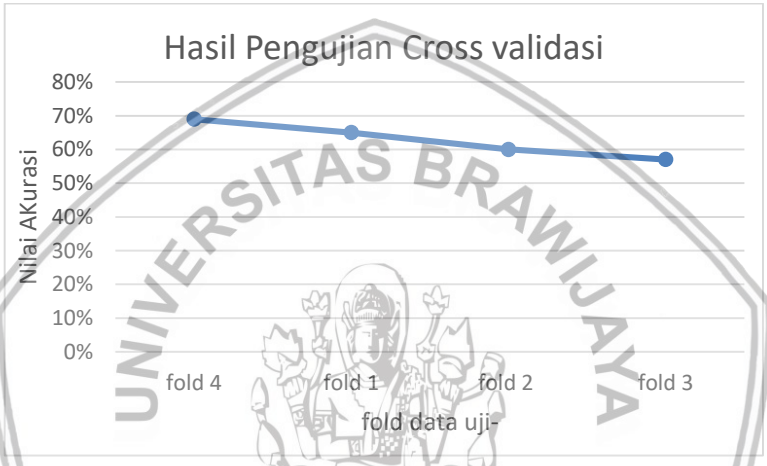
6.4 Pengujian *K- Fold cross validation*

K-Fold cross validation adalah metode yang digunakan untuk mengetahui tingkat keberhasilan dan kelayakan data dalam perancangan sistem klasifikasi. Dalam metode k-fold, dataset yang dibagi menjadi sejumlah k-buah partisi secara acak ke dalam k partisi yang berukuran sama dan dilakukan sejumlah k-kali eksperimen. Pengujian ini dilakukan dengan melakukan pembagian dataset menjadi 4 kelompok (4 fold) dengan banyak data yang sama yaitu 65 data untuk masing-masing kelompok, kemudian setiap kelompok akan dibagi menjadi data latih dan data uji secara bergantian dan diujikan menggunakan nilai *k*=3.

Skenario pengujian ini terbagi menjadi 4 yang hasilnya dapat dilihat pada Tabel 6.4.

Tabel 6.4 Hasil Pengujian K-*fold cross validation*

Skenario Ke-	*fold* Data Latih	*fold* Data Uji	Nilai akurasi
1	1,2,dan 3	4	69%
2	2,3,dan 4	1	65%
3	1,3 dan 4	2	60%
4	1,2 dan 4	3	57%
Rata-rata akurasi			62.69%



Gambar 6.4 Grafik Pengujian K-*Fold cross validation*

Seperti ditunjukkan pada Tabel 6.4 dan Gambar 6.4 Hasil dari pengujian tersebut menghasilkan akurasi tertinggi sebesar 69% pada pengujian *fold* data uji 4 dan terendah sebesar 57% pada pengujian *fold* data uji 3. Sedangkan rata-rata dari akurasi pengujian menggunakan metode *cross validation* adalah sebesar 62.69%. Berdasarkan penjelasan tersebut dapat disimpulkan bahwa penggunaan data yang acak pada klasifikasi memberikan hasil yang berbeda dikarenakan setiap kelompok data memiliki karakteristik data yang berbeda sehingga menghasilkan nilai akurasi yang berbeda.

BAB 7 KESIMPULAN DAN SARAN

7.1 Kesimpulan

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

1. Cara melakukakan *preprocessing* terhadap data *SMILES* adalah mencari atau menemukan karakter dari masing-masing *SMILES* kemudian menghitung panjang notasi dan jumlah masing-masing atom penyusunnya. Langkah Selanjutnya adalah membagi setiap jumlah atom yang sudah diketahui dengan panjang notasi *SMILES* sehingga dapat dijadikan fitur untuk proses klasifikasi. Preprocessing dari *SMILES* tersebut mendapatkani fitur berupa jumlah atom *B, C, N, O, P, S, F, Cl, Br, I, dan OH.*
2. Cara kerja dari metode *Modified k-nearest neighbor* yaitu menghitung selisih nilai dari masing-masing 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 *validitas* dari setiap data latih tersebut. Perhitungan *Euclidean distance* juga dilakukan pada data latih dengan data uji dan diurutkan untuk diambil nilai tertinggi, kemudian dari proses tersebut dilakukan perhitungan weigh voting untuk mendapatkan kelas klasifikasi berdasarkan nilai bobot kelas tertingginya. Kelas klasifikasi yang digunakan pada penelitian ini yaitu Saraf dan Jantung.
3. Pada penelitian ini metode *Modified 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 4 kali percobaan menghasilkan nilai akurasi tertinggi sebesar 69% dengan rata-rata akurasi dari seluruh percobaan sebesar 62.69%.
    c. Pengujian variasi nilai *k* menghasilkan nilai akurasi tertinggi sebesar 73% dengan nilai K=3.
    d. Pengujian *Holdout Validation* terhadap jumlah data latih menghasilkan nilai akurasi tertinggi sebesar 73% dengan menggunakan 90% data latih dan 10% data uji.

7.2 Saran

Untuk pengembangan penelitian selanjutnya saran yang dapat diambil berdasarkan penelitian ini sebaiknya untuk meningkatkan akurasi pada sistem

diharapkan pada penelitian selanjutnya untuk menggunakan fitur yang lebih banyak dan bervariasi karena Metode ini membutuhkan fitur spesifik yang membedakan pada masing-masiing data setia kelasnya. Penggunaan data latih yang lebih bervariasi dan dalam jumlah yang tepat memudahkan proses klasifikasi menggunakan Metode *Modified K- Nearest Neighbor.*

DAFTAR REFERENSI

Astuti, F. D., Ratnawati, D. E., & Widodo, A. W. (2017). Deteksi Penyakit Kucing dengan Menggunakan Modified K-Nearest Neighbor Teroptimasi (Studi Kasus: Puskeswan Klinik Hewan dan Satwa Sehat Kota Kediri). *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer, 1*(11), 1295-1301.

Cahyaningtyas, Y., Ridok, A., & Dewi, C. (2013). Penerapan Fuzzy K-Nearest Neighbor untuk Menentukan Status Evaluasi Kinerja Karyawan. *Repositori Jurnal Mahasiswa PTIIK UB, 1*(4).

Dzikrulloh, N. N., Indriati, & Setiawan, B. D. (2017). Penerapan Metode K–Nearest Neighbor (KNN) dan Metode Weighted Product (WP) Dalam Penerimaan Calon Guru Dan Karyawan Tata Usaha Kejuruan Muhammadiyah 2 Kediri). *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer, 1*(5), 378-385.

Junaedi, H. (2011). Penggambaran Rantai Karbon Dengan Menggunakan Simplified Molecular Input Line System (*SMILES*). *Prosiding Konferensi Nasional "Inovasi dalam Desain dan Teknologi" . IDe aTech 2011* .

Kurniawan, E. (2017). Analisa Data Rekam Medis Menggunakan Teknik Data Mining Association Rules Dengan Algoritma Clustering. *Seminar Nasional Pendidikan, Sains dan Teknologi* .

Leidiyana, H. (2013). Penerapan Algoritma K-Nearest Neighbor Untuk Penentuan Resiko Kredit Kepemilikan Kendaraan Bemotor. *Jurnal Penelitian Ilmu Komputer, System Embedded & Logic, 1*(1), 65-76.

Manning, C. D., Raghavan, P., & Schütze, H. (2009). *An Introduction to Information Retrieval.* England: Cambridge University Press.

Muliantara, A. (2009). Penerapan Regular Expression Dalam Melindungi Alamat Email Dari Spam Robot Pada Konten Wordpress. *Jurnal Ilmu Komputer, 2*(1).

Parvin, H., Alizadeh, H., & Minaei-Bidgoli, B. (2008). Mknn: Modified K-Nearest Neighbor. *Proceedings of the World Congress on Engineering and Computer Science 2008*.

Salni, Marisa, H., & Mukti, R. W. (2011). Isolasi Senyawa Antibakteri dari Daun Jengkol (Pithecolobium Lobatum Benth) dan Penentuan Nilai KHM-nya. *Jurnal Penelitian Sains, 14*(1(D)), 38-41.

Searls, D. B. (2012). A New Online Computational Biology Curriculum. *PLoS Comput Biol, 10*(6). doi:e1003662

Simanjuntak , T. H., Mahmudy, W. F., & Sutrisno. (2017). Implementasi Modified K-Nearest Neighbor Dengan Otomatisasi Nilai K Pada Pengklasifikasian

Penyakit Tanaman Kedelai. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer, 1*(2), 75-79.

Wafiyah, F., Hidayat, N., & Perdana, R. S. (2017). Implementasi Algoritma Modified K-Nearest Neighbor (MKNN) untuk Klasifikasi Penyakit Demam. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer, 1*(10), 1210-1219 .

Weininger, D. (1987). *SMILES*, a Chemical Language and Information System. 1. Introduction to Methodology and Encoding Rules. *J. Chem. Inf. Comput. Sci, 28*(1), 31-36.