PENERAPAN ALGORITMA C4.5 UNTUK KLASIFIKASI FUNGSI SENYAWA AKTIF MENGGUNAKAN KODE *SIMPLIFIED MOLECULAR INPUT LINE SYSTEM* (SMILES)

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh:
Mochammad Iskandar Ardiyansyah Rochman
NIM: 145150201111026

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

PERSETUJUAN

PENERAPAN ALGORITMA C4.5 UNTUK KLASIFIKASI FUNGSI SENYAWA AKTIF MENGGUNAKAN KODE *SIMPLIFIED MOLECULAR INPUT LINE SYSTEM* (SMILES)

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :
Mochammad Iskandar Ardiyansyah Rochman
NIM: 145150201111026

Telah diperiksa dan disetujui oleh :

Dosen Pembimbing I	Dosen Pembimbing II
Dian Eka Ratnawati,S.Si, M.Kom	Syaiful Anam, S.Si, MT, Ph.D
NIP:19730619 200212 2 001	NIK. 197801152002121003

PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar 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 sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 13 Juli 2018

Mochammad Iskandar Ardiyansyah Rochman
NIM: 145150201111026

KATA PENGANTAR

Ungkapan rasa syukur kepada Allah SWT yang telah memberikan rahmat dan hidayahNya kepada penulis sehingga bisa menyelesaikan skripsi yang berjudul "Penerapa Algoritma C4.5 untuk Klasifikasi Senyawa Aktif Menggunakan Kode *Simplified Molecular Input Line System*" dengan baik.

Penulis melakukan penulisan skripsi ini dengan tujuan untuk memenuhi salah satu syarat bagi mahasiswa dalam memperoleh Ijazah Sarjana Komputer Program Studi Teknik Informatika Universitas Brawijaya.

Dalam menyelesaikan skripsi ini, penulis mendapatkan banyak bantuan dan dukungan dari berbagai pihak. Oleh karena itu penulis menghaturkan ucapan terima kasih kepada:

1. Ibu Dian Eka Ratnawati, S.Si,M.Kom dan Syaiful Anam, S.Si, MT, Ph.D. selaku dosen Pembimbing skripsi yang telah membimbing dan mengarahkan penulis sehingga skripsi dapat selesai dengan baik.
2. Bapak Agus Wahyu Widodo, S.T, M.Sc selaku Ketua Program Studi Teknik.
3. Bapak Tri Astoto Kurniawan, S.T, M.T, Ph.D selaku Ketua Jurusan Teknik Informatika.
4. Kedua orang tua dan adik saya yang selalu mendo'akan, menyemangati dan membantu kelancaran penulisan skripsi.
5. Adhiajeng Putri Ananda Pradana yang selalu menemani, memberikan semangat, dukungan dan bantuan dalam proses penulisan skripsi.
6. Teman-teman satu angkatan Teknik Informatika 2014 atas kebersamaan dan saling membantu dalam perjuangan menyelesaikan penulisan skripsi.
7. Semua pihak yang turut membantu penulisan skripsi yang tidak dapat disebutkan satu per satu.

Penulis menyadari bahwa skripsi ini masih terdapat kekurangan. Oleh karena itu, kritik dan saran yang membangun penulis harapkan untuk penyempurnaan skripsi ini. Besar harapan semoga skripsi ini dapat bermanfaat bagi semua pihak.

Malang, 13 Juli 2018

Penulis
iskandardotardian@gmail.com

ABSTRAK

Senyawa merupakan hal yang kerap kali ditemukan didunia ini, dengan wujud zat yang merupakan kumpulan dari senyawa (Berpendidikan, 2015). Senyawa sendiri terbagi atas senyawa aktif dan tidak aktif. Senyawa tersebut memiliki fungsi yang mungkin dapat dimanfaatkan untuk beberapa aspek bila memiliki suatu fungsi seperti obat ataupun perangsang suatu hormon bekerja. notasi SMILES (*Simplified Molecular Masukan Line System)* oleh David Weininger pada tahun 1980. Notasi SMILES memanfaatkan karakter yang ada pada ASCII yang sangat mudah untuk diproses oleh komputer. Proses klasifikasi notasi SMILES akan sangat bermanfaat untuk mengetahu kelas fungsi dari senyawa tersebut. Penelitian ini dilakukan untuk mengklasifikasi fungsi dari senyawa memanfaatkan notasi SMILES dengan menerapkan algoritma C4.5 sedangkan objeknya adalah 2 kelas fungsi senyawa, diantaranya adalah kelas kanker dan metabolisme. Fitur yang diuji dari penelitian sebanyak 11 fitur. Hasil dari pengujian terbaik ketika teknik diskritisasi yang dilakukan menggunakan teknik diskritisasi *entropy based*, melakukan pembagian nilai panjang notasi SMILES pada setiap atribut fitur, dan penggunaan data latih sebanyak mungkin yaitu akan menghasilkan nilai akurasi sebesar 79,34%. Sedangakan akurasi dari pengujian *cross validation* menunjukan angka akurasi sebesar 70,18%.

Kata kunci : Notasi SMILES, Algoritma C4.5, Kanker, Metabolisme

ABSTRACT

Compounds are things that are often found in this world, with a substance that is a collection of compounds (Educated, 2015). The compound itself is divided into active and inactive compounds. The compound has a function that may be utilized for some aspect if it has a function like a drug or a stimulating hormone work. notation of SMILES (Simplified Molecular Input Line System) by David Weininger in 1980. SMILES notation takes advantage of ASCII characters that are very easy to process by the computer. SMILES notation classification process will be very useful to know the function class of the compound. This study was conducted to classify the function of the compound utilizing the SMILES notation by applying the C4.5 algorithm while the object is 2 classes of compound function, including the class of cancer and metabolism. Features tested from research as many as 11 features. The results of the best tests when the discretization technique is performed using entropy based discretization techniques, dividing the SMILES notation values on each feature attribute, and the use of practicable data as much as possible will result in an accuracy of 79.34%. While the accuracy of the cross validation test shows an accuracy of 70.18%.

***Keywords**: SMILES Notation, Algorithm C4.5, Cancer, Metabolism*

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

BAB 1 PENDAHULUAN

1.1 Latar Belakang

Senyawa merupakan suatu hal yang kerap kali dijumpai didunia ini, dengan wujud yang tersusun dari kumpulan senyawa yang membentuk suatu bentuk zat. Senyawa sendiri merupakan gabungan dari beberapa unsur yang memiliki sifat berbeda dengan unsur pembentuknya (Berpendidikan, 2015).

Dalam dunia ini terdapat berbagai jenis senyawa baik yang aktif maupun tidak aktif (Darusman et al, 2011). Senyawa sangat penting untuk keilmuan dan pekerjaan di bidang kimia, biologi, dan kedokteran dikarenakan bidang tersebut mempelajari tentang makhluk hidup yang tentunya berinteraksi langsung dengan berbagai senyawa yang ada di muka bumi. Senyawa tersebut bila diteliti tentunya akan didapatkan memiliki dampak toksik atau bermanfaat bagi makhluk hidup terutama manusia. Dengan penelitian beragam jenis senyawa yang ada di muka bumi tentunya sangat bermanfaat untuk pencegahan dan pengobatan bagi manusia.

Senyawa kimia memiliki rumus struktur 2 dimensi yang tentunya akan sulit untuk disimpan dalam komputer sehingga diciptakannya notasi yang memudahkan proses yang berjalan secara digital menggunakan media komputer yaitu notasi SMILES (*Simplified Molecular Masukan Line System)*. Kode SMILES sendiri berhasil diciptakan oleh David Weininger pada tahun 1980 silam dengan memanfaatkan konsep dari teknik *graph*. SMILES merupakan sebuah sistem dalam menotasikan notasi kimia yang dimanfaatkan untuk berbagai proses pada kimia modern. Berpedoman pada dasar teori *molecular graph,* kode SMILES mampu melakukan proses spesifikasi struktrur secara mudah dan tepat dengan mempergunakan struktur bahasa yang umum dan mudah dipahami. Dalam proses spesifikasi di SMILES, atom memiliki peran sebagai *node* atau titik pada *graph*, sementara ikatan akan didefinisikan sebagai edge atau garis pada *graph*, sehingga representasi *graph* akan dapat tercipta dari sebuah molekul. Sistem SMILES juga memiliki tingkat kecocokan yang sangat baik untuk berjalan pada mesin yang memiliki kecepatan yang sangat tinggi. Notasi SMILES juga memanfaatkan karakter yang ada pada ASCII untuk melakukan proses spesifikasi molekul dan menciptakan representasi dari atom dan molekul (Junaedi, 2011: 222-223).

Metode penulisan kimia modern dengan kode SMILES sangat menunjang bagi bidang ilmu yang mempelajari senyawa, namun tentunya dibutuhkan metode yang cepat dan akurat untuk mendeteksi fungsi dari senyawa aktif yang

berhasil ditemukan. Dengan mendeteksi tingkat pola yang ada dari beberapa senyawa maka dapat diklasifikasikan senyawa tersebut menjadi beberapa klasfifikasi yang dapat menentukan secara cepat senyawa yang dijadikan masukan.

Salah satu metode klasifikasi yang bisa dipergunakan untuk proses klasifikasi notasi SMILES adalah algoritma C4.5. Algoritma C4.5 merupakan salah satu teknik yang dapat digunakan untuk menjalankan proses klasifikasi dengan representasi penggambaran menggunakan pohon keputusan. Pohon keputusan merupakan salah satu metode klasifikasi dan prediksi yang sudah terbukti sangat efektif dalam menangani berbagai permasalahan dan mudah untuk dimengerti proses dari pohon keputusan akan diawali dari noda akar yang akan di cek hingga noda daun yang dilakukan secara terus menerus atau rekursif dengan kondisi setiap cabang akar hingga daun merepresentasikan kondisi dari atribut sedangkan kondisi dimana tidak ada cabang dari pohon lagi maka akan menyatakan kondisi keputusan atau klasifikasi yang dipilih. Algoritma C4.5 sudah cukup sering digunakan dan juga memiliki banyak keunggulan salah satunya adalah karena algoritma C4.5 memiliki kemampuan untuk mengolah data yang bersifat numerik maupun diskrit, menghasilkan rumus yang mudah diaplikasikan dan juga memiliki peforma yang lebih superior dan proses yang lebih cepat dibandingkan dengan metode lainya (Haryanto dan Hansun, 2017: 97-98).

Berdasarkan penelitian yang dilakukan oleh Teguh Budi Santoso tahun 2017 dalam penelitian berjudul "ANALISA DAN PENERAPAN ALGORITMA C4.5 UNTUK PREDIKSI LOYALITAS PELANGGAN" menunjukan bahwa akurasi sebesar 93,3%, kemudian penelitian yang dilakukan oleh Swastina tahun 2013 dengan judul "Penerapan algoritma C4.5 Untuk Penentuan Jurusan Mahasiswa" menunjukan bahwa akurasi algoritma C4.5 sebesar 93,1% untuk kesesuaian jurusan Mahasiswa dan 82,64% untuk rekomendasi jurusan, serta penelitian yang dilakukan oleh Prajarini tahun 2016 dengan judul "Perbandingan algoritma Klasifikasi *Data Mining* Untuk Prediksi Penyakit Kulit" dengan obejek penyakit kulit didapatkan akurasi sebesar 94,7%, merujuk pada penelitian sebelumnya dengan tingkat keakuratan yang sangat baik maka tentunya akan sangat baik bila diterapkan dalam klasifikasi senyawa aktif kode SMILES.

Algoritma C4.5 memiliki keunggulan dengan perwujudan visualisasi dari penyelesaian masalah yang dapat dikelola dengan memanfaatkan konsep dari *Data Mining* (penggalian data) yang menciptakan suatu protokol hasil klasifikasi atau prediksi yang mudah diamati dan dipahami, sehingga algoritma C4.5 mampu membentuk formula terbaik dalam menentukan keputusan bercabang

yang berpengaruh nantinya pada hasil klasifikasi kode SMILES (Haryanto dan Hansun, 2017: 97-98).

Dengan menggunakan algoritma C4.5 dengan representasi penggambaran pohon keputusan maka akan diharapkan agar metode ini dapat secara optimal dalam melakukan proses klasifikasi dengan objek data kode SMILES sehingga dapat ditentukan nama senyawa aktif yang ada pada kode SMILES tersebut dan akan bermanfaat bagI perkembangan penelitian dalam bidang senyawa aktif dapat dilakukan secara digital menggunakan komputer serta turut secara inovatif untuk berkembang mengikuti perkembangan zaman.

1.2 Hipotesis

Algoritma C4.5 dapat digunakan untuk proses klasifikasi dengan menggunakan objek notas SMILES dengan akurasi yang baik.

1.3 Rumusan Masalah

Berdasarkan latar belakang tersebut di atas maka rumusan masalahnya adalah sebagai berikut:

1. Bagaimana proses pada algoritma C4.5 dalam mengklasifikasi notasi SMILES?
2. Bagaimana tingkat keakuratan algoritma C4.5 dalam mengklasifikasi notasi SMILES?

1.4 Tujuan Penelitian

Tujuan Umum

1. Membantu mempercepat proses pencarian manfaat dan kandungan dari senyawa yang dilambangkan dengan notasi SMILES.

Tujuan Khusus

1. Untuk mengetahui bagaimana proses pada algoritma C4.5 dalam mengklasifikasi notasi SMILES.
2. Untuk mengetahui tingkat keakuratan algoritma C4.5 dalam mengklasifikasi notasi SMILES.

1.5 Manfaat Penelitian

Adapun manfaat penelitian ini sebagai berikut:

1. Bermanfaat untuk mempercepat proses pencarian manfaat dan kandungan dari senyawa yang dilambangkan dengan notasi SMILES.
2. Membantu bidang keilmuan kimia tentang senyawa aktif untuk terus berkembang dan berinovasi dengan mengikuti perkembangan digital dengan memanfaatkan media komputer.
3. Dapat menjadi refrensi untuk penelitian lebih lanjut.

1.6 Batasan Masalah

Penelitian ini hanya mengambil data berbagai notasi SMILES dari website pubchem.ncbi.nlm.nih.gov/compound/152306#section=Depositor-Provided PubMed-Citations, serta hanya melakukan proses klasifikasi terhadap 2 kelas obat diantaranya adalah kanker dan metabolisme.

1.7 Sistematika Pelaporan

Format laporan yang digunakan pada proposal skripsi ini adalah format proposal Implementasi yang memiliki format ataupun sistematika seperti berikut:

Bab 1 Pendahuluan

Mencakup tentang latar belakang yang menjadi dasar untuk dilakukanya penelitian, rumusan masalah yang menjadi inti bahan yang diteliti, tujuan dari penelitian, manfaat yang akan didapatkan dari penelitian, dan batasan masalah yang ditentukan serta terdapat juga sistematika pembahasan.

Bab 2 Kajian Pustaka

Membahas tentang kajian pustaka untuk rujukan seperti beberapa penelitian terdahulu dan pustaka yang berkaitan dengan penelitian yang dilakukan sehingga dapat dijadikan sebagai kajian pustaka untuk mendukung pelaksanaan penelitian klasifikasi fungsi senyawa aktif menggunakan algoritma C4.5 dengan menggunakan objek notasi SMILES.

Bab 3 Metode Penelitian

Membahas tentang metode beserta rencana untuk analisis data yang digunakan pada penelitian, sumber pengumpulan data serta pengolahan data, rencana metode untuk pengujian dan rencana untuk analisa hasil penellitian.

Bab 4 Implementasi

Membahas tentang bagaimana cara pengumpulan dan sumber dari data berserta proses menyiapkan data agar dapat diolah dan diproses serta implementasi untuk mengolah data klasifikasi notasi SMILES.

Bab 5 Pengujian

Membahas tentang cara menguji data untuk memastikan apakah hasil penelitian merepresentasi hasil yang validitas sehingga akan menunjukan tingkat kecocokan algoritma C4.5 dengan objek notasi SMILES senyawa aktif.

Bab 6 Analisis Hasil

Membahas tentang analisa mengapa algoritma C4.5 dapat berjalan baik ataupun tidak baik, karena akan dianalisa faktor-faktor yang menyebabkan data tersebut cocok ataupun tidak cocok dengan objeknya yaitu notasi SMILES senyawa aktif.

Bab 7 Kesimpulan dan Saran

Membahas tentang penarikan kesimpulan untuk menjawab pertanyaan-pertanyaan yang ada di latar belakang serta saran bagi berbagai pihak.

Daftar Pustaka

Berisi daftar rujukan dan sitasi yang digunakan pada penelitian ini.

BAB 2 KAJIAN PUSTAKA

2.1 Perbandingan Objek dan Penelitian Sebelumnya dengan Algoritma C4.5

Untuk menunjukan bahwa algoritma C4.5 memiliki kualitas yang baik dan telah teruji untuk digunakan maka dibutuhkan perbandingan objek dan hasil penelitian sebelumnya sehingga juga akan dapat asumsi bahwa juga akan teruji dengan baik bila dengan menggunakan objek data notasi SMILES senyawa aktif. Berikut Tabel 2.1 yang menjelaskan tentang Objek dan Hasil Penelitian Sebelumnya dengan algoritma C4.5 sedangkan Tabel 2.2 tentang penelitian yang sedang dijalankan.

Tabel 2.1 Perbandingan Objek dan Penelitian Sebelumnya dengan Algoritma C4.5

No	Penulis	Objek	Metode	Hasil Penelitian
1.	(Santoso, 2017)	Objek : *Pelanggan* *Masukan* : Kualitas Pelayanan, Harga, Citra Perusahaan, Kepercayaan.	Metode : *C4.5* Proses : 1. Menghitung nilai Entropi. 2. Menghitung nilai Gain. 3. Menggambarkan formula pohon keputasan. 4. Melakukan klasifikasi data dengan pohon keputusan.	*Output* : • Hasil loyalitas pelanggan dengan akurasi 93,3%
2.	(Swastina, , 2013)	Objek : Mahasiswa. *Masukan* : IPK Semester 1, IPK Semester 2, IPK Semester 3.	Metode : *C4.5* Proses : 1. Menghitung nilai Entropi. 2. Menghitung nilai Gain. 3. Menggambarkan formula pohon keputusan. 4. Melakukan klasifikasi data dengan pohon keputusan.	*Output* : Kesesuaian jurusan dengan akurasi 93,31% dan rekomendasi 82,64%.
3	(Prajarini, 2016)	Objek : Penyakit Kulit. *Masukan* : Gejala penyakit kulit	Metode : *C4.5* Proses : 1. Menghitung nilai Entropi. 2. Menghitung nilai Gain. 3. Menggambarkan formula	*Output* : Klasifikasi penyakit kulit dengan akurasi 94,7%

No	Penulis	Objek	Metode	Hasil Penelitian
			pohon keputusan. 4. Melakukan klasifikasi data dengan pohon keputusan.	

Tabel 2.2 Penelitian yang dijalankan

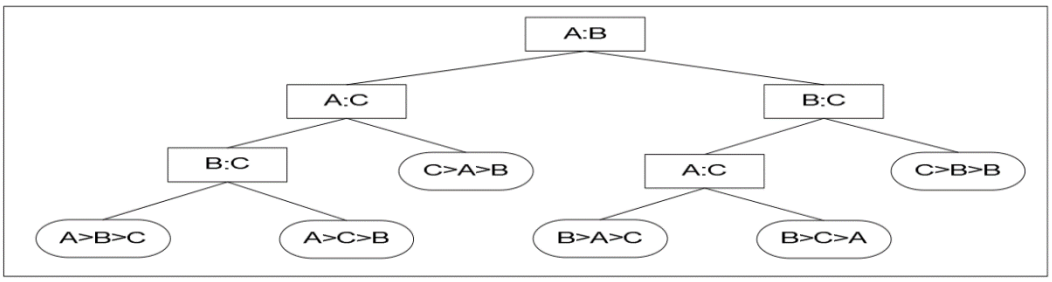
No	Penulis	Objek	Metode	Hasil Penelitian
1.	Mochammad Iskandar Ardiansayh Roochman	Objek : notasi *SMILES* (*Simplified Molecular Masukan Line System)* *Masukan* : Panjang notasi *SMILES*, jumlah dari masing-masing unsur yang ada pada notasi *SMILES*.	Metode : *C4.5* Proses : 1. Menghitung nilai Entropi. 2. Menghitung nilai Gain. 3. Menggambarkan formula pohon keputasan. 4. Melakukan klasifikasi data dengan pohon keputusan.	*Output* : • Hasil klasifikasi senyawa aktif dengan notasi SMILES

2.2 Algoritma C4.5

Manusia cenderung selalu menghadapi berbagai jenis masalah dari segala jenis bidang kehidupan. Masalah yang dihadapi ini pula mempunyai beragam tingkatan kesulitan. Untuk memecahkan masalah, manusia memulai untuk melakukan pengembangan suatu sistem yang mampu mempermudah mereka memecahkan masalah yang ada, salah satu konsep buatan manusia tersebut merupakan pohon keputusan. Pohon keputusan sendiri merupakan sebuah metode klasifikasi bersifat prediksi yang memiliki efektifitas *powerfull* dan kerap kali digunakan. Metode pohon keputusan ini berperan untuk mengkonversi fakta-fakta menjadi bentuk pohon keputusan yang berperan untuk merepresentasikan suatu formula yang sangat mudah untuk dipahami oleh bahasa alami atau bahasa manusia (Kalsum, 2009).

Pohon keputusan merupakan suatu data yang terstruktur dengan memiliki bagian simpul (noda) dan garis (edge). Noda yang ada pada pohon keputusan dikategorikan menjadi 3 diantaranya, simpul bagian akar, simpul bagian cabang, dan simpul akhir tree (Kalsum, 2009).

Secara umum pohon keputusan merupakan sebuah gambaran yang dimodelkan dengan acuan suatu permasalahan dan terbagi atas serangkaian serangkaian keputusan sehingga dapat mengarah ke hasil solusi (Kalsum, 2009).



Gambar 2.1 Pohon keputusan untuk 3 bilangan A,B dan C

Pohon keputusan yang ada pada Gambar 2.1 diatas dibaca mulai dari atas hingga bawah. Simpul yang terletak di bagian paling atas dinamakan sebagai simpul akar sedangkan simpul yang terletak di bagian paing akhir dinamakan simpul keputusan atau daun. Cabang yang terletak di bagian kanan dan juga kiri merupakan representasi dari banyaknya alternatif klasifikasi atau keputusan yang dapat diambil berdasarkan syarat dan rule pohon keputusan yang ada. Hasil keputusan yang diambil dari pohon keputusan dalam satu waktu hanyalah sebanyak 1 keputusan sehingga pohon keputusan tidak akan mungkin untuk memilki banyak hasil (Kalsum, 2009). Beberapa pohon keputusan, juga sering terdapat simpul probabilitas. Simpul probabilitas ditandai dengan adanya gambar sebuah lingkaran kecil yang juga disertai dengan adanya angka kemungkinan yang merepresentasikan kemungkinan munculnya sebuah keputusan yang ada pada cabang tersebut.

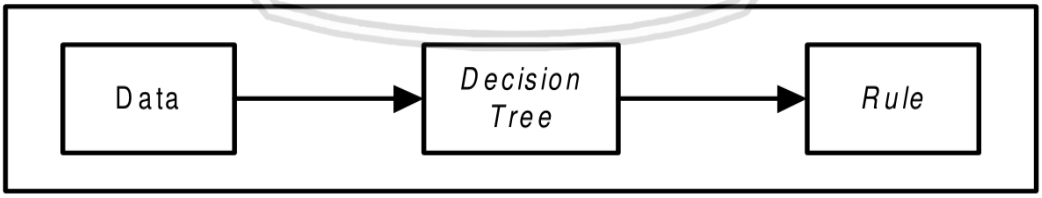
Pada konsep pohon keputusan mampu memberi keunggulan dengan perwujudan visualisasi dari penyelesaian masalah yang dapat dikelola dengan memanfaatkan konsep dari *Data Mining* (penggalian data) yang menciptakan suatu protokol hasil klasifikasi atau prediksi yang mudah diamati dan dipahami, sehingga menyebabkan konsep *Data Mining* pohon keputusan fleksibel dan efisien. Metode pohon keputusan sendiri telah sering kali digunakan untuk menyelesaikan masalah yang ada pada beragam bidang keilmuan, diantaranya seperti pada bidang kesehatan dan kedokteran yang dimanfaatkan untuks diagnosa penyakit pasien, ilmu computer pada struktur data, psikologi untuk teori pengambilan keputusan (Kalsum, 2009).

Walau memiliki banyak sekali keunggulan namun menurut kajian dari Trisiiant tahun 2016 beberapa kekurangan diantaranya berikut ini:

- Memungkinkan untuk terjadinya keadaan overlap bila atribut yang digunakan memiliki jumlah besar. Beserta akan juga berakibat adanya pertambahan waktu atau waktu eksekusi untuk proses klasifikasi dan juga memori untuk menampung data yang diproses semakin tinggi.
- Membuat desain pohon keputusan dengan sempurna merupakan hal yang sulit.
- Kualitas keputusan atau akurasu yang didapatkan memuliki ketertergantungan pada bagaimana proses mendesai pohon tersebut dan juga penentuan nilai gain dan entropi (Abdillah, 2011).

Pohon keputusan dapat pula diartikan sebagai sebuah tools yang dapat digunakan untuk membentuk suatu rule yang memiliki pedoman pada model dari keputusan dan akibat yang berdampak dari keputusan yang ada, seperti peluang untuk terjadinya suatu keputusan dan juga biaya yang akan dibutuhkan untuk suatu penggunaan. Sehingga dengan adanya pohon keputusan dapat menjadi strategi terbaik untuk menghitung peluang sebuah kejadian yang disertai dengan analisis faktor yang melatarbelakangi mengapa keputusan tersebut diambil oleh pohon keputusan (Kalsum, 2009).

Pohon keputusan merupakan suatu metode belajar dengan tingkat kepopuleran yang tinggi dan banyak dimanfaatkan dikarenakan memiliki konsep yang cukup praktis. Metode pohon keputusan adalah sebuah metode yang berusaha untuk menemukan fungsi pendekatan yang memiliki nilai diskrit. Konsep yang ada pada pohon keputusan adalah dengan mengubah data pada tabel keputusan kemudian dikoversi menjadi sebuah aturan pada pohon keputusan. Berikut merupakan konsep dan gambaran dari metode pohon keputusan ditunjukan pada Gambar 2.2 (Kalsum, 2009).



Gambar 2.2 Konsep dari pohon keputusan

Dalam salah satu algoritma pohon keputusan yaitu adalah algoritma C4.5, yang memiliki cara penyelesaian dengan menggunakan konsep pohon keputusan. Algoritma yang ada pada *Data Mining* ini merupakan suatu algoritma yang manfaatkan untuk proses klasifikasi atau prediksi maupun pengelompokan yang cenderung mampu memiliki sifat prediktif. Cabang-cabang dari pohon keputusan adalah percabangan untuk menentukan klasifikasi kemudian daun-

daunnya pohon keputusan adalah kelas dimana data masukan akan masuk salah satu kelas daun tersebut. Rumus yang digunakan untuk menentukan nilai entropy yang ada di algoritma C4.5 adalah:

2.2.1 *Entropy*

Entropi merupakan distribusi probabilitas dalam teori informasi dan diadopsi kedalam algoritma C4.5 untuk mengukur tingkat homogenitas distribusi kelas dari sebuah himpunan data (*dataset*). Sebagai ilustrasi, semakin tinggi tingkat entropi dari sebuah *dataset* maka semakin homogen distribusi kelas pada *dataset* tersebut. Rumus untuk menghitung entropi pada pohon keputusan C4.5 ditujukan pada Rumus 2.1 (Raditya, 2009):

$$Entropi\ (S) = \sum_{i=0}^{k} -phi\ \log_2 phi \quad \text{.....(2.1)}$$

Keterangan :

S : Himpunan (*dataset*) kasus

k : Banyaknya partisi S

phi : Probabilitaas yang didapat dari *Sum*(Ya) atau *Sum*(Tidak) dibagi total kasus

2.2.2 *Gain*

Setelah membagi *dataset* berdasarkan sebuah atribut kedalam subset yang lebih kecil, entropi dari data tersebut akan berubah. Perubahan entropi ini dapat digunakan untuk menentukan bagus tidaknya pembagian data yang telah dilakukan. Perubahan entropi ini disebut dengan information gain dalam algoritma C4.5. Information gain ini diukur dengan menghitung selisih antara entropi *dataset* sebelum dan sesudah pembagian (splitting) dilakukan. Pembagian yang terbaik akan menghasilkan entropi subset yang paling kecil, dengan demikian berdampak pada information gain yang terbesar. Rumus untuk menghitung nilai Gain pada pohon keputusan ditujukan pada Rumus 2.2 (Raditya, 2009):

$$Gain\ (A) = Entropi\ (S) - \sum_{i=1}^{k} \frac{|s|}{|S_i|}\ x\ Entropi\ S_i \quad \text{.....(2.2)}$$

Keterangan :

S : himpunan kasus

A : atribut

n : jumlah partisi atribut A

$|Si|$: jumlah kasus pada partisi ke-i

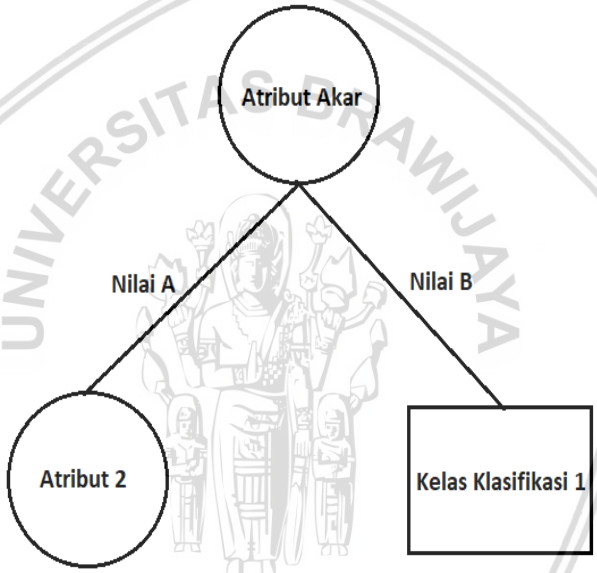
$|S|$: jumlah kasus dalam S (Abdillah, 2011)

2.2.3 Pembuatan Pohon Keputusan

Pada umumnya langkah-langkah pada algoritma C4.5 untuk membuat aturan berupa pohon keputusan adalah :

a. Memilih atribut yang berperan sebagai akar dengan melihat nilai gain terbesar.
b. Buat cabang dengan nilai aturan rule dengan aturan percabangan dengan nilai kondisi pada atribut akar.
c. Membagi kasus kembali pada cabang.
d. Mengulangi kasus hingga cabang memiliki kelas yang sama.

Untuk memvisualisasi proses pembuatan pohon keputusan pada algoritma C4.5 maka dapat digambarkan (Mashlahah, 2013):



Gambar 2.3 Pembuatan pohon keputusan

2.3 *SIMPLIFIED MOLECULAR MASUKAN LINE SYSTEM* (SMILES)

Notasi SMILES sendiri berhasil diciptakan oleh David Weininger pada tahun 1980 silam dengan memanfaatkan konsep dari teknik *graph*. SMILES merupakan sebuah sistem dalam menotasikan Notasi kimia yang dimanfaatkan untuk berbagai proses pada kimia modern. Berpedoman pada dasar teori *molecular graph,* notasi SMILES mampu melakukan proses spesifikasi struktrur secara mudah dan tepat dengan mempergunakan struktur bahasa yang umum dan mudah dipahami. Dalam proses spesifikasi di SMILES, atom memiliki peran sebagai *node* atau titik pada *graph*, sementara ikatan akan didefinisikan sebagai edge atau garis pada *graph*, sehingga representasi *graph* akan dapat tercipta dari sebuah molekul. Sistem SMILES juga memiliki tingkat kecocokan yang sangat baik untuk berjalan pada mesin yang memiliki kecepatan yang sangat tinggi. Notasi

SMILES juga memanfaatkan karakter yang ada pada ASCII untuk melakukan proses spesifikasi molekul dan menciptakan representasi dari atom dan molekul.

Penulisan rumus SMILES memiliki sifat casesensitif yang berati memiliki arti yang berbeda bila terdapat huruf kapital maupun kecil. Secara umum atom dapat direpresntasikan dengan huruf namun kerapkali juga dijumpai atom yang direpresentasikan dengan 2 huruf. Aturan yang diterapkan bagi atom dengan kondisi lambang 2 huruf adalah dengan menuliskan huruf pertama dari lambang atom tersebut dengan huruf kapital. Kemudian aturan untuk huruf yang mengikuti huruf dibelakangnya akan dilambangkan menggunakan huruf non kapital. Sedangkan aturan untuk atom yang direpresentasikan hanya dengan 1 huruf maka akan digunakan huruf kapital untuk melambangkanya.

Molekul yang terdiri atas rantai karbon memiliki 3 jenis ikatan pada rantainya diantaranya adalah ikatan tunggal, rangkap dan juga rangkap 3. Ikatan tunggal dalam notasi dilambangkan dengan simbol "-", sedangkan ikatan rangkap memiliki simbol "=", dan juga rangkap tiga memiliki simbol "≡". Sama halnya dengan rumus SMILES untuk rangkap satu juga memiliki simbol "-" namun khusus untuk rangkap satu diperbolehkan untuk tidak dituliskan. Untuk rangkap dua juga sama disimbolkan dengan simbol "=" dan untuk rangkap tiga disimbolkan dengan simbol "#". Struktur dari rantai karbon dapat memiliki struktur linier (tidak memiliki cabang) dan juga ada yang berstruktur bercabang. Aturan penulisan bagi rantai karbon yang bercabang adalah ketika menuliskan posisi karbon yang bercabang maka sebelum penulisan rantai karbon tersebut diharuskan meletakan simbol "(" terlebih dahulu, setelah itu maka akan dilanjutkan dengan menuliskan rumus karbon yang tersusun pada cabangnya dan ditutup dengan simbol ")". Cabang dalam suatu molekul memungkinkan untuk memiliki cabang kembali, dengan aturan yang sama untuk menuliskan cabang perulangan dan berganda.

Disamping itu terdapat pula atom yang memiliki ikatan dan ikatan yang terbentuk adalah lingkaran dengan istilah siklik (atom yang memiliki ikatan diujungnya dan saling berikatan) . Untuk kasus ini maka ikatan siklik pada rantai karbon akan dilambangkan dengan simbol angka setelah atom karbon contohnya "C1" berikut merupakan salah satu contoh penulisan notasi SMILES (Junaedi, 2011: 222-223).

Notasi Kimia :
CH2CH2CH2CH2CH2CH2
Notasi SMILES :
C1H2CH2CH2CH2CH2C1H2

Atau
C1CCCCC1
Atau
C1H2-CH2-CH2-CH2-CH2-C1H2
Atau
C1-C-C-C-C-C1

Atom pada SMILES direpresentasikan dengan simbol atom mereka sendiri. SMILES memiliki atom organik yang berpengaruh pada SMILES diantaranya B, C, N, O, P, S, F, Cl, Br, dan I. Sedangkan atom H yang mengikutinya tidak perlu dituliskan. Atom tersebut mengisi bagian bagian dari notasi SMILES yang melambangkan representasi senyawa secara 2 dimensi (Weinigner, 1987).

2.4 *Regular Expression* (REGEX)

REGEX atau *Regular Expression* merupakan sekumpulan string yang dikodekan secara khusus yang dimanfaatkan sebagai pola untuk melakukan pecocokan sebuah set string. REGEX mulai muncul pada sekitar tahun 1940 sebagai suatu cara untuk merepresentasikan bahasa reguler, namun REGEX mulai dimanfaatkan dalam dunia pemrograman pada sekitar tahun 1970.

REGEX kemudian menjadi salah satu bagian penting dari tools yang muncul pada Sistem operasi Unix-ed, sed dan vi (vim) editor, grep, AWK dan lain lain. Contoh dari aplikasi regex adalah bila kita bermaksud untuk mencocokan data dengan jenis semua angka telepon yang artinya mengambil data berupa seluruh angka maka REGEX yang tepat untuk mengatasi masalah tersebut adalah [0-9] yang memiliki arti untuk mengambil seluruh angka atau lebih tepatnya digit. REGEX dari struktur [0-9] akan mengatakan kepada prosessor adalah, Cocokkan angka yang Anda temukan di kisaran 0 sampai 9. Namun untuk mengambil seluruh angka dari nomer telepon maka diperluan limit berapa angka dari telepon tersebut sehingga REGEX yang dapat digunakan untuk menangkap nomer telepon adalah [0-9][0-9][0-9]-[0-9][0-9][0-9]-[0-9][0-9][0-9][0-9]. Namun cara tersebut terlalu tidak efektif dikarenakan ada shortcut untuk memanggil seluruh digit angka yaitu \d\d\d-\d\d\d-\d\d\d\d.

Untuk uji regex secara online dapat dilakukan di website www.regexpal.com, dimana notasi regex dapat ditesing berdasarkan data kalimat yang ada pada dokumen berserta fungsi replacenya. Regex sendiri dapat berjalan diberbagai pemrograman seperti php, javascript, python, c++ dan sangat berguna untuk penelitian di bidang pemrosesan bahasa alami, text mining dan

sistem temu kembali informasi dimana dibutuhkan suatu kondisi string khusus yang akan dicari (fitzgerald, 2012:1-4).

2.5 PHP (*PHP Hypertext Processor*)

PHP yang merupakan singkatan dari PHP Hypertext Processor. PHP sendiri merupakan sebuah bahasa pemrograman yang memiliki script disisi server (SSS / Server Side Scripting). *Database* yang didukung oleh PHP diantaranya adalah MySQL, Generic ODBD, PostgreSQL, Oracle,Solid, dan Sybase. PHP merupakan sebuah perangkat lunak yang bersifat open source yang bebas diunduh oleh siapa saja secara gratis (Erawan, 2014: 5).

File PHP dapat saja berisi tag HTML, script ataupun teks. File PHP akan dikembalikan ke browser pada bentuk HTML asli. Sedangkan untuk ekstensi dari PHP diantaranya .php, .php3, .phtml. Kelebihan yang dimiliki PHP diantaranya adalah mampu berjalan diberbagai sistem operasi seperti Mac OS, Linux, dan Windows, Kommpatibel dan cocok digunakan untuk berbagai jenis server yang ada saat ini, dapat secara bebas untuk di unduh dari halaman resmi yaitu www.php.net, kemudian mudah untuk dipelajari dan sangat efisien untuk digunakan (Erawan, 2014: 5).

Untuk menggunakan perangkat lunak PHP maka diharuskan untuk menginstall web server apache pada komputer ataupun server yang digunakan kemudian meninstall MySQL yang berfungsi sebagai *database* atau dengan menggunakan layanan yang ada pada hosting yang telah tersedia PHP dan MySQL. Terdapat tiga jenis penggunaan PHP diantaranya adalah:

- Aplikasi Berbasis Web Dan Website
- Comand Line Scripting
- Aplikasi Dekstop atau Ghrapic User Interface

Jenis yang pertama merupakan jenis penggunaan yang paling umum untuk digunakan dan membutuhkan browser, server website, dan juga PHP. Untuk pengguna OS Linux dan Mac OS, dapat menggunakan sever apache dan untuk windows dapat memanfaatkan server IIS. Browser web yang dapat digunakan diantaranya Internet Explorer, Mozilla FireFox, Google Chrome, Opera dan lain lain. Disamping itu juga dapat memanfaatkan layanan yang ada di hosting, sehingga yang perlu untuk dilakukan adalah dengan menulis script PHP, melakukan upload data ke server kemudian melihat hasil script di browser yang dimiliki (Erawan, 2014: 5).

2.6 HTML (*Hypertext Markup Language*)

Segala jenis halaman website yang kerap kali di buka di browser merupakan hasil tampilan menggunakan HTML. Sehingga HTML dapat didefinisikan sebagai bahasa dasar yang dimanfaatkan untuk menampilkan isi dari web pada browser website. Sederhananya HTML merupakan suatu perangkat lunak yang difungsikan untuk me markup sebuah dokumen seperti dokumen dengan ekstensi .doc atau .docx yang di markup oleh perangkat lunak Microsoft Word (Ariona, 2013: 10-12).

2.7 CSS (*Casading Style Sheet*)

CSS merupakan sebuah teknologi perangkat lunak yang bermanfaat untuk mempermudah serta mempercantik proses pembuatan suatu website. Dengan memanfaatkan fitur dari CSS maka programmer akan mampu mengaplikasikan pengaturan style pada tag HTML tertentu. Terlebih juga memungkinkan untuk menempatkan CSS pada sebuah file namun dapat diaplikasikan kek banyak halaman sepenuhnya. Tag yang dimiliki oleh CSS adalah <style></style> sementara letak dari tag tersebut adalah berada di dalam tag HTML <head></head>. Dengan CSS juga programmer memungkinkan untuk menyisipkan beberapa komentar pada CSS tersebut sehingga mempermudah pemahaman kode yang memanfaatkan CSS yang sangat banyak. Sedangkan untuk deklarasi komentar tersebut dapat dilakukan dengan diapit dengan sintaks /**/ (Astamal, 2008).

2.8 IDE NETBEANS

IDE NetBeans merupakan perangkat lunak gratis, open source, dan merupakan suatu IDE yang memungkinkan untuk membangun berbagai aplikasi dekstop, mobile dan aplikasi berbasis website. IDE Netbeans juga mendukung berbagai jenis bahasa pemrogaman, termasuk java, C++, HTML5, dan PHP. IDE NetBeans juga menyediakan dukungan untuk integrasi siklus pembangunan aplikasi, mulai dari pembuatan program hingga proses debuging, profilling dan deployment. IDE NetBeans mendukung berbagai sistem operasi seperti Windows, Mac OS, Linux, dan UNIX OS lainya (NetBeans, 2016).

IDE NetBeans menyediakan dukungan untuk teknologi JDK 8 (Java Development Kit) dan dukungan untuk perbaikan dan update terbaru dari Java. IDE Netbeans merupakan IDE pertama yang menyediakan dukungan untuk JDK 8, Java FX2 dan Java EE. IDE NetBeans menyediakan dukungan penuh terhadap Java

EE dengan menggunakan standar terbaru Java, Web Service, XML, SQL, dukungan penuh untuk server GlassFish, refrensi pengaplikasian Java (NetBeans, 2016).

2.9 Framework Code Igniter

Code igniter merupakan suatu framework yang terbaik dan terpopuler pada saat ini. Telah cukup banyak website yang terkenal dan populer menggunakan framework code igniter sebagai komponen utama dalam membangun website. Code igniter sendiri merupakan sebuah perangkat lunak web applicaton service framework dengan sifat bebas pakai dan open source yang dimanfaatkan untuk pembangunan suatu aplikasi PHP yang bersifat dinamis. Tujuan dibangunya framework Code igniter adalah untuk membantu berbagai programmer dalam membangun aplikasi dengan lebih mudah dan cepat terlebih dari menulis program secara langsung. Framework Codeigniter dipublikasikan pada 28 februari 2006 (Daqiqil, 2011).

Framework CodeIgniter dibangun mengusung konsep dari MVC atau Model View Controller, dan terkenal sebagai framework yang sangat cepat dibandingkan framework yang lain. Kelebihan dari Codeigniter adalah sangat ringan, mudah untuk dipelajari dan terstruktur. Terlebih Codeigniter juga mempunyai fitur yang dapat mempermudah pengguna diantaranya:

- Memanfaatkan Pola dari MVC sehingga kode yang dibangun akan lebih jelas dan terstruktur.
- URL Friendly.
- Kemudahan mempelajari sintax dengan user guide.
- Framework tercepat berdasarkan hasil benchmark.
- Mudah dimodifikasi dan diadaptasi.

MVC pada CodeIgniter merupakan sebuah konsep yang diterapkan oleh Codeigniter. MVC merupakan sebuah singkatan dari konsep model view controller. MVC merupakan sebuah teknik pemrograman yang memisahkan alur pikir, penyimpanan data, dan antarmuka atau tampilan dari aplikasi, secara sedehana dapat diartikan framework CodeIgniter menggunkan konsep dengan memiisahkan antara proses, data dan juga desain (Daqiqil, 2011).

2.10 Diskritisasi

Diskritisasi juga disebut dengan *Bining,* yang mengkonversi nilai numeric menjadi nilai diskrit. Diskritisasi kerap kali digunakan untuk data mining algoritma data mining yang tidak dapat mengolah nilai numeral. Diskritisasi juga berguna untuk mengurangi banyak nya nilai dari suatu atribut. Terutama ketika disana ada noise pada proses perhitungan maka diskritisasi akan mengurangi dan mengabaikan nilai yang tidak relevan tersebut.

Contohnya, ketika ada nilai numeral atribut X, dan ada sampel acak $\{\mathrm{x\,i}\}\,\mathrm{n\,i} = 1$ sebesar n yang berasal dari atribut X, diskritisasi dilakukan untuk membagi nilai dari atibut X pada k kelompok, dengan menemukan $K - 1$ batasan nilai yang disimbolkan dengan $\mathrm{v1}$, $\mathrm{v2}$, … , v (Zaki dan Meirra, 2014: 89-90):

$$w = {X_{max} - X_{min}}/{K}$$

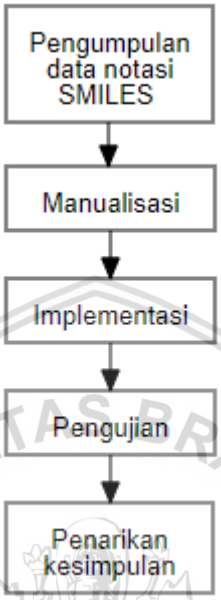
Keterangan :
X_{max} : Nilai maksimum dari *dataset* suatu atribut
X_{min} : Nilai minimum dari *dataset* suatu atribut
K : banyaknya kelompok diskrit

$$v_i = X_{min} + i \times w$$

Keterangan :
X_{min} : Nilai minimum dari *dataset* suatu atribut
i : nilai batas dari 1 hingga $k - 1$
w : nilai pembobotan

BAB 3 METODE PENELITIAN

3.1 Studi literatur/Kajian pustaka/Dasar teori



Gambar 3.1. Diagram penelitian

Penelitian ini merupakan jenis penelitian jenis analisis dengan alur penelitian seperti pada Gambar 3.1. Pengumpulan data uji berupa notasi SMILES dilakukan dengan pencarian data yang berasal dari website http://pubchem.ncbi.nlm. nih.gov/compound/1523076886#section=Depositor-Provided-PubMed-Citations sedangkan data literatur ataupun pustaka dilakukan dengan cara penelusuran kajian dan literatur terhadap berbagai jenis jurnal terbitan nasional dan internasional, skripsi ataupun thesis mahasiswa, dan teori-teori dari buku. Pengujian yang dilakukan adalah dengan mengukur tingkat keakuratan algoritma C4.5 pada pohon keputusan dalam mengklasifikasi notasi SMILES. Segala informasi yang didapatkan oleh penulis selanjutnya akan dilakukan perbandingan terhadap data asli berupa nama senyawa aktif yang ada pada kode SMILES tersebut.

3.2 Analisis Kebutuhan

Adapun Kebutuhan Fungsional yang digunakan dalam klasifikasi algoritma C4.5 pada pohon keputusan dalam mengklasifikasi notasi SMILES adalah berikut:

(1) Sistem harus mampu mendeteksi tingkat gain yang dimiliki oleh kumpulan *dataset* SMILES untuk menentukan tingkat atribut yang paling berpengaruh terhadap hasil klasifikasi yang akan menjadi akar.
(2) Sistem harus menyediakan database untuk menampung data pada *dataset* untuk dikalkulasi nilai gain dan entropinya.
(3) Sistem mampu untuk mengukur nilai keakuratan klasifikasi senyawa aktif kode SMILES.
(4) Sistem mampu untuk menentukan hasil klasfikasi dengan masukan yang berupa kode SMILES.
(5) Sistem harus mampu menormalisasi data dengan perbedaan range jauh untuk meningkatkan keakuratan.

Adapun Spesifikasi Perangkat yang digunakan dalam Penerapan algoritma C4.5 pada pohon keputusan dalam mengklasifikasi notasi SMILES adalah berikut:

(1) HTML 5
(2) PHP
(3) Netbeans IDE 8.2
(4) Laptop Acer One 14 Intel(R) Core(TM) i3-5005U CPU Windows Operating System 64 bit RAM 4 GB
(5) XAMPP Control Panel v3.2.2
(6) Google Chrome Versi 65.0.3325.162 64 bit
(7) Data notasi SMILES

Aplikasi Penerapan algoritma C4.5 pada pohon keputusan dalam mengklasifikasi notasi kimia modern senyawa aktif SMILES

Variabel kontrol : Jenis atribut / Fitur yang diuji
Variabel terikat : Hasil klasifikasi notasi kode SMILES
Variabel bebas : Nilai atribut / Fitur yang diuji

3.3 Pengumpulan Data

Detail data yang dipaergunakan bisa dilihat seperti pada Tabel 3.1.

Tabel 3.1 Sumber data

No.	Item data	Jenis data	Sumber data
1.	Data notasi dan kode SMILES	Sekunder	pubchem.ncbi.nlm.nih.gov/compound/152306#section=Depositor-Provided-PubMed-Citations

No.	Item data	Jenis data	Sumber data
2	Data rumus algoritma C4.5 pohon keputusan	Sekunder	Buku, jurnal dan artikel dan website

3.4 Pengolahan Data

Data yang diolah berdasarkan kelas klasifikasi senyawa. Untuk Penelitian kali ini penulis memfokuskan pada data senyawa aktif dan kode SMILES nya. Masing masing kelas yang akan diolah berdasarkan beberpa parameter yang telah diatur diantaranya adalah

a. Panjang notasi SMILES
b. Banyaknya atom C
c. Banyaknya atom O
d. Banyaknya atom N
e. Banyaknya atom P
f. Banyaknya atom S
g. Banyaknya atom F
h. Banyaknya atom Cl
i. Banyaknya atom Br
j. Banyaknya atom I
k. Banyaknya atom OH

3.5 Perancangan

Pada tahap perancangan maka akan dilakukan perancangan pada sistem dimana sistem tersebut mampu melakukan kesemua jenis kebutuhan fungsional yang telah dipaparkan pada analisis kebutuhan penelitian ini. Perancangan ini didasarkan dari pustaka yang didapat dari studi litertatur, pengumpulan data, dan pengolahan data. Semua komponen tersebut akan dipadukan untuk mendukung proses perancangan dariu sistem untuk klasifikasi fungsi senyawa dari masukan notasi SMILES. Perancangan sistem ini diawali dengan penjabaran melalui *flowchart* yang akan menjelaskan alur dari tahapan *Preprocessing* dan juga tahapan dari pelatihan dan pengujian yang ada di algoritma C4.5. Setelah *flowchart* akan dijabarkan juga perhitungan secara manual dari beberapa *dataset* yang mewakili beberapa kelas klasifikasi senyawa. Disini juga akan dijabarkan perancangan antarmuka dari sistem.

3.6 Implementasi

Pada tahapan ini hasil dari perancangan sistem akan diterapkan menjadi sebuah sistem yang dapat melakukan klasifikasi data notasi SMILES dengan melakukan proses *Preprocessing* data dan Pelatihan data. Berikut merupakan alur dari proses implementasi:

(1) Sistem akan melakukan *Preprocessing* data untuk mengubah data menjadi nilai atribut yang siap pakai oleh sistem.
(2) Sistem akan menyimpan *dataset* hasil *Preprocessing* data untuk datalatih di *database* php mysql.
(3) Sistem akan membuat formula akar dilihat dari nilai gain terbesar yang diolah berdasarkan *dataset* notasi SMILES.
(4) Gain terbesar setelahnya akan menentukan *node* tingkat satu dua dan seterusnya.
(5) Penguji memasukan data masukan berupa kode SMILES untuk dilakukan klasifikasi.
(6) Sistem melakukan proses preprocessing untuk memecah data masukan kode SMILES untuk dilihat jumlah dari banyaknya atom atau atribut yang sesuai dengan atribut yang diuji.
(7) Sistem melakukan klasfikasi senyawa aktif kode SMILES.
(8) Sistem menampilkan output hasil klasfikasi berupa nama senyawa aktif dari kode SMILES tersebut.

3.7 Pengujian

Pada tahap pengujian ini akan dilakukan uji tingkat akurasi hasil klasifikasi notasi SMILES dengan melakukan pengaruh uji kesesuaian dari hasil klasifikasi notasi SMILES dengan data kelas klasifikasi dari data uji. Parameter atribut juga akan diuji seberapa besar pengaruh tersebut pada keputusan yang dihasilkan.

3.8 Kesimpulan

Kesimpulan akan ditarik dengan tingkat akurasi dari penggunaan hasil klasifikasi senyawa aktif kode SMILES. Hasil analisa tingkat keakuratan akan dapat memberi kesimpulan bahwa algoritma C4.5 Pohon keputusan cocok atau untuk digunakan dengan data kode SMILES.

BAB 4 PERANCANGAN

4.1 Perancangan

Perancangan merupakan tahap yang sangat dibutuhkan pada proses penelitian. Dalam proses perancangan maka akan dilakukan beberapa tahapan diantaranya adalah tahapan deskripsi umum sistem, perancangan sistem, contoh kalkulasi manual, Perancangan uji coba pada sistem dan juga perancangan antarmuka pada sistem.

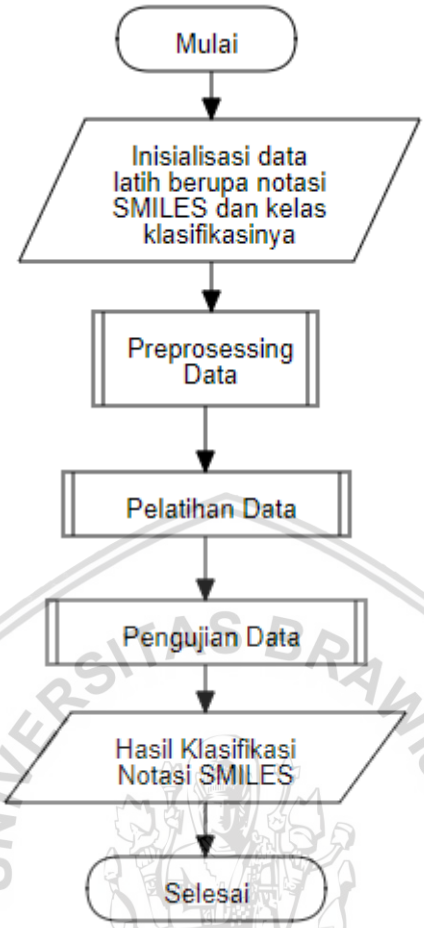
4.1.1 Deskripsi Umum Sistem

Sistem yang akan dibangun pada penelitian ini merupakan sistem yang berguna untuk mengklasifikasi pemilihan kelas pada notasi SMILES. Kelas yang akan diteliti hanya terbatas sebanyak 2 kelas diantaranya adalah saraf, anti radang, cardiovaskular, saluran pernafasan, kanker, infeksi dan metabolisme. Hasil keluaran pada sistem ini berupa prakiraan hasil kelas klasifikasi berdasarkan masukan yang berupa notasi SMILES. Sistem ini berguna untuk melakukan prediksi kelas dan kandungan senyawa dengan cepat berdasarkan masukan notasi SMILES.

Proses pembuatan sistem diawali dengan pencarian data yang berasal dari website http://pubchem.ncbi.nlm.nih.gov/compound//152306#section//=Depositor- Provided-PubMed-Citations kemudian data tersebut di pilah menjadi 7 bagian kelas diantaranya adalah saraf, anti radang, cardiovaskular, saluran pernafasan, kanker, infeksi dan metabolisme. 7 data tersebut sebanyak 70% akan dijadikan data latih dan akan disimpan pada *database* dan diolah oleh sistem dengan menggunakan algoritma C4.5 sedangkan 30% akan dijadikan data uji yang akan diuji hasil akurasi dari rumus pohon keputusan yang telah dibuat.

4.1.2 Perancangan Perangkat Lunak

Pada tahap ini maka akan dilakukan perancangan dari perangkat lunak yang akan dibangun. Proses pembuatan perangkat lunak akan menggunakan bahasa pemrograman HTML5 dan PHP dengan IDE Netbeans dengan menggunakan *database* mysql melalui phpmyadmin yang ada pada XAMPP. Secara umum alur proses dari algoritma C4.5 dalam mengklasifikasi fungsi dan kandungan senyawa aktif berdasarkan masukan notasi SMILES ditunjukan pada Gambar 4.1. Berdasarkan Gambar 4.1, langkah awal dalam penggunaan algoritma C4.5 adalah dengan inisialisasi data latih yang berupa notasi SMILES beserta kelas klasifikasinya. Kemudian melakukan pelatihan data dengan algoritma C4.5. Kemudian dilakukan pengujian tingkat akurasi dari data uji.



Gambar 4.1 Alur proses dari algoritma C4.5

Untuk proses selanjutnya terdapat dua proses utama diantaranya adalah:

1. *Preprocessing* Data

   Preprocessing data merupakan proses pengolahan data sebelum diproses agar nilai atribut masing-masing notasi SMILES siap untuk digunakan sebagai data latih. Pada proses *Preprocessing* data notasi SMILE akan dihilangkan indeks angka dan juga akan dipecah menjadi atribut-atribut yang akan siap digunakan untuk data latih ataupun data uji. Pada *Presprocessing* juga akan dihitung jumlah masing masing atribut yang akan dibagi dengan panjang notasi SMILES.

2. Pelatihan Data

   Proses pelatihan data merupakan proses dimana sistem akan dilatih berdasarkan data latih yang membutuhkan data nilai atribut yang telah di *Preprocessing*. Atribut tersebut merupakan atribut B, C, N, O, P, S, F, Cl, Br, I, dan OH yang telah dibagi dengan panjang notasi SMILES dan dikalikan dengan konstanta yang sama agar menjadi bilangan bulat. Dari atribut tersebut maka akan dicari nilai gain tertinggi yang nanti akan digunakan

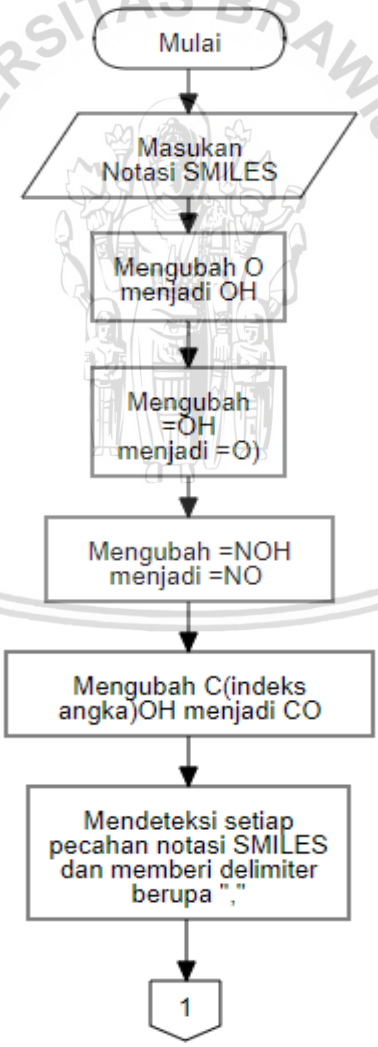
untuk membuat aturan pada desain pohon keputusan. Pohon keputusan dengan update terakhir akan digunakan untuk proses pengujian sistem.

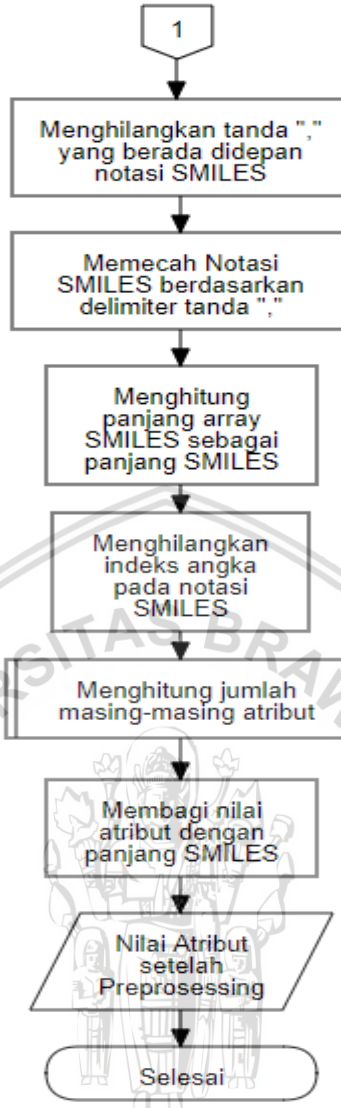
3. Pengujian Data

   Pengujian data merupakan proses menguji masukan yang akan disesuaikan dengan luaran kelas klasifikasinya berdasarkan kelas klasifikasi yang sudah ada. Proses pengujian ini juga akan menghitung akurasi tingkat kesesuaian luaran yang diberikan sistem terhadap data kelas yang aslinya.

4.1.2.1 *Preprocessing* Data

Tahap pertama yang dilakukan adalah tahap *Preprocessing* data dengan masukan notasi SMILES yang akan diolah menjadi nilai atribut yang dapat langsung diproses untuk pelatihan data. Tahapan lebih rinci pada tahap *Preprocessing* data ditunjukan pada Gambar 4.2.





Gambar 4.2 Alur proses dari *Preprocessing* notasi SMILES

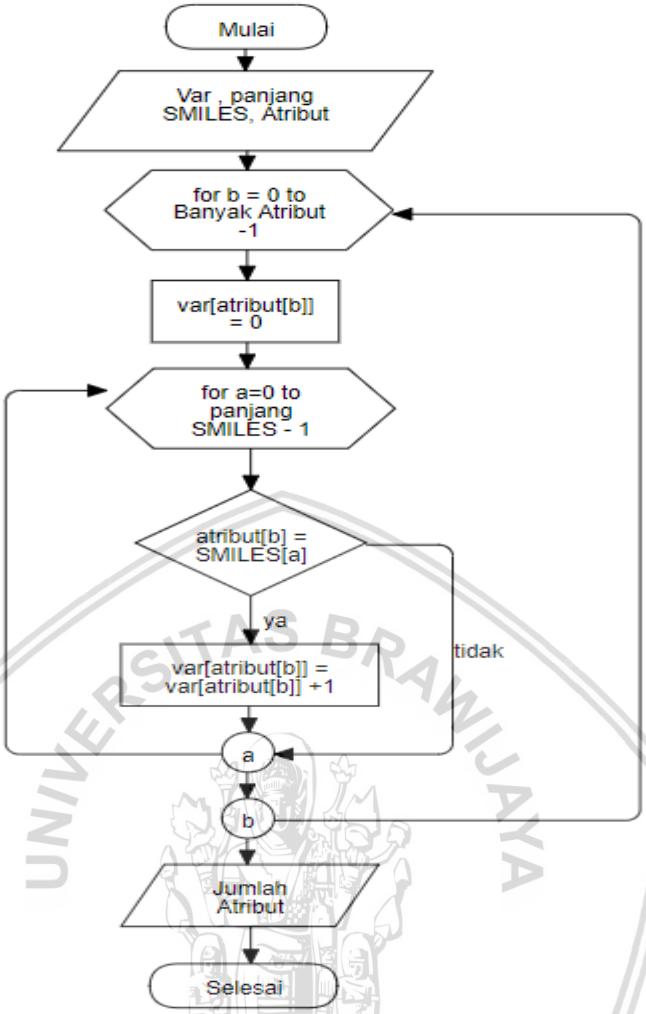
Gambar 4.2 merupakan alur proses dari *Preprocessing* data dengan mendeteksi sebuah pola menggunkanan fitur *Regular Expression* yang tersedia pada bahasa pemrograman PHP. Hasil dari tahap *Preprocessing* ini adalah nilai atribut yang sudah siap digunakan untuk pelatihan data. Keterangan lebih lanjut mengenai Gambar 4.2 adalah sebagai berikut:

1. Memasukan notasi SMILES sebagai masukan yang akan diolah pada tahapan *Preprocessing*.
2. Mendeteksi dengan *Regular Expression* adanya huruf O dan mengubahnya menjadi OH.
3. Mendeteksi dengan *RegEx* adanya pola =OH dan mengubahnya menjadi =O).
4. Mendeteksi dengan *RegEx* pola =NOH dan mengubahnya menjadi =NO.

5. Mendeteksi dengan *Regular Expression* adanya pola =C(indeks angka)OH dan mengubahnya menjadi =CO karena indeks tersebut tidak diperlukan dalam atribut yang digunakan untuk pelatihan data.
6. Mendeteksi dengan *Regular Expression* adanya setiap atom yang ada pada notasi SMILES dan menambahkan simbol "," yang berfungsi sebagai delimiter.
7. Mendeteksi dengan *Regular Expression* adanya simbol "," yang berada didepan dan menghapus simbol tersebut karena akan menyebabkan adanya array atribut bernilai kosong yang akan juga dihitung sebagai salah satu panjang notasi SMILES.
8. Mendeteksi dengan *Regular Expression* setiap atom yang dipisahkan oleh delimiter "," dan mengubahnya menjadi pecahan array
9. Menghitung panjang notasi SMILES dengan menghitung banyaknya pecahan array yang dipisahkan oleh delimiter sebelumnya.
10. Menghilangkan semua indeks angka karena indeks tersebut tidak diperlukan dalam atribut yang digunakan untuk pelatihan data.
11. Menghitung masing masing atribut yang ada pada array.
12. Membagi jumlah setiap atribut dengan panjang dari notasi SMILES.
13. Nilai terakhir siap digunakan untuk proses pelatihan data.

4.1.2.2 Menghitung Jumlah masing-masing Atribut

Menghitung jumlah masing-masing atribut sangat dibutuhkan sebagai nilai dari atribut yang akan dijadikan parameter untuk proses pelatihan data. Tahapan lebih rinci pada tahap Menghitung jumlah masing-masing atribut ditunjukan pada Gambar 4.3.



Gambar 4.3 Alur proses dari menghitung jumlah masing-masing atribut

Gambar 4.3 merupakan alur proses dari tahapan Menghitung jumlah masing-masing atribut diantaranya adalah atribut B, C, N, O, P, S, F, Cl, Br, I, dan OH yang ditampung pada variabel dan akan dikalkulasikan untuk tahapan pelatihan data. Keterangan lebih lanjut mengenai Gambar 4.3 adalah sebagai berikut:

1. Memasukan notasi SMILES yang telah dipecah menjadi array, panjang SMILES dan fitur atribut.
2. Perulangan dari array indeks atribut sesuai banyak atribut hingga indeks akhir dari atribut yang diinputkan, dalam perulangan tersebut terdapat proses:
    3. Inisialisasi jumlah var dari atribut dengan indeks b dengan nilai 0 untuk mereset nilai dari perulangan sebelumnya.
    4. Perulangan dari indeks a hingga panjang akhir dar notasi SMILES yang telah dilakukan *preprocessing*, dalam perulangan tersebut terdapat proses:

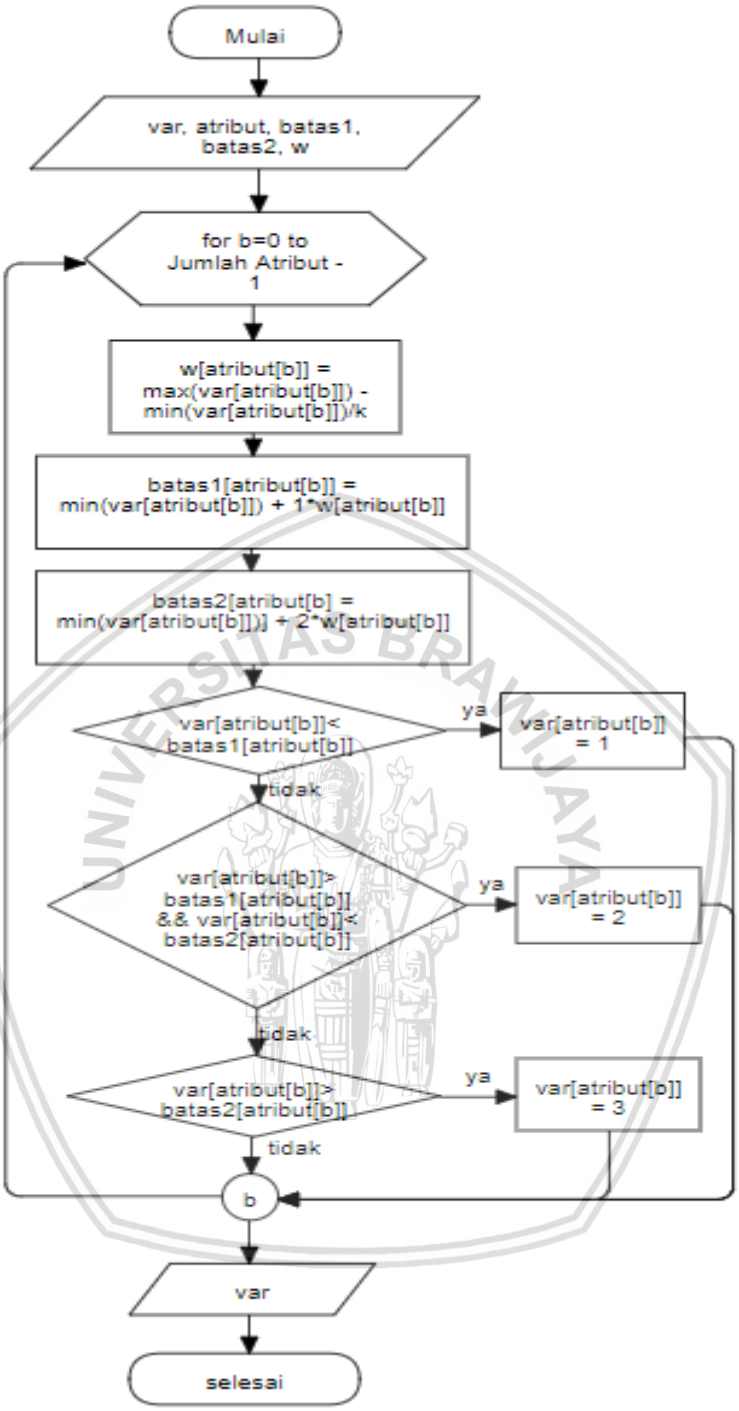
5. Melakukan percabangan di setiap indeks notasi SMILES bila memang nilai array indeks a merupakan nilai dari salah satu atribut dengan indeks b maka variabel atribut tersebut akan ditambah 1, perulangan dilakukan pada atribut C, N, O, P, S, F, Cl, Br, I, dan OH sesuai dengan banyak indeks atribut b.
6. Hasil keluaran berupa nilai atribut yang tersimpan pada var yaitu masing masing jumlah dari atribut B, C, N, O, P, S, F, Cl, Br, I, dan OH pada notasi SMILES yang diinputkan sehingga setiap notasi SMILES akan memiliki nilai masing masing atribut hasil dari *preprocessing* dari program.

4.1.2.3 Pelatihan Data

Tahap kedua setelah data di *Preprocessing* maka keluaran dari masing masing atribut yang pada notasi SMILES akan digunakan pada proses pelatihan data dengan menggunakan algoritma C4.5. Proses pelatihan data yang ada algoritma C4.5 diantaranya adalah proses diskritisasi data dan pembentukan rule pohon keputusan.

4.1.2.3.1 Diskritisasi Data

Pada tahap ini maka akan dilakukan proses pengubahan data kontinyu menjadi kelas atau beberapa kelompok diskrit, pada penelitian ini yang dilakukan adalah dengan mengelompokan menjadi 3 kelas diskrit, sehingga algoritma C4.5 akan lebih mudah dalam mengolah data tersebut karena akan dibentuk sebuah *rule* pohon keputusan maka perlu dibuat kelas diskrit karena akan sangat banyak cabang yang dimiliki bila data tersebut berupa data kontinyu. Untuk tahapan lebih rinci pada tahap diskritisasi data pada algoritma C4.5 akan ditunjukan pada Gambar 4.4 berikut



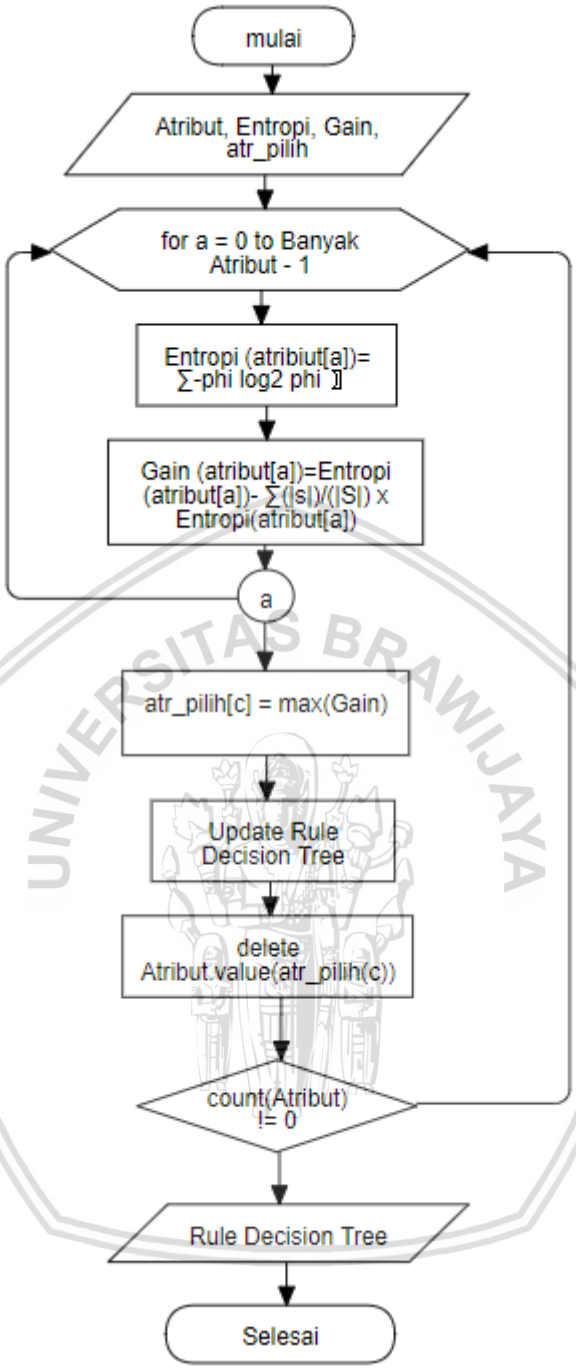
Gambar 4.4 Alur proses dari diskritisasi data

Gambar 4.4 merupakan alur proses tahapan diskritisasi data dari masing-masing atribut diantaranya adalah atribut B, C, N, O, P, S, F, Cl, Br, I, dan OH yang ditampung pada var dan akan dikalkulasikan untuk tahapan perhitungan entropi dan gain untuk penentuan *rule* pohon keputusan. Keterangan lebih lanjut mengenai Gambar 4.4 adalah sebagai berikut:

1. Memasukan var yang akan menampung nilai atribut dari notasi SMILES dan inisialisasi batas 1 dan batas 2 untuk proses diskritisasi data dari bank data notasi SMILES.
2. Perulangan dari array indeks atribut sesuai banyak atribut hingga indeks akhir dari atribut yang diinputkan yaitu yang digunakan adalah atribut B, C, N, O, P, S, F, Cl, Br, I, dan OH, dalam perulangan tersebut terdapat proses:.
    3. Menghitung batas 1 untuk menentukan diskritisasi kelompok 1 dan kelompok 2 diskritisasi.
    4. Menghitung batas 2 untuk menentukan diskritisasi kelompok 2 dan kelompok 3 diskritisasi.
    5. Bila nilai atribut kurang dari batas 1 maka nilai atribut akan masuk kelompok 1 diskrit.
    6. Bila nilai atribut lebih dari batas 1 dan kurang dari batas 2 maka nilai atribut akan masuk kelompok 2 diskrit.
    7. Bila nilai atribut lebih dari batas 2 maka nilai atribut akan masuk kelompok 3 diskrit.
8. Keluaran berupa nilai atribut yang telah dilakukan proses diskritisasi berupa kelompok 1, 2, 3 diskrit.

4.1.2.3.2 Perhitungan Entropi dan Gain

Pada proses perhitungan entropi dan gain maka dilakukan dengan mencari nilai entropi dan juga nilai Gain dari setiap atribut fitur yang digunakan untuk penentuan kasifikasi dari notasi SMILES. Nilai Gain tertinggi dari suatu atribut akan dipilih sebagai akar kemudian entropi dari akar yang merupakan atribut fitur dengan gain tertinggi akan digunakan sebagai entropi total untuk mencari nilai noda di bawahnya dengan cara penentuan yang sama yaitu nilai atribut gain tertinggi dengan syarat atribut yang telah terpilih tidak dihitung kembali pada proses perhitungan atribut ditingkatan bawahnya, proses dilakukan terus menerus hingga semua atribut terpilih menjadi salah satu bagian *rule* dari pohon keputusan. Hasil dari pelatihan data tersebut adalah sebuah *rule* untuk membangun pohon keputusan yang akan digunakan untuk proses pengujian data dalam proses penentuan klasifikasi dari sebuah notasi SMILES. Tahapan lebih rinci pada tahap pelatihan data akan ditunjukan pada Gambar 4.4 berikut.



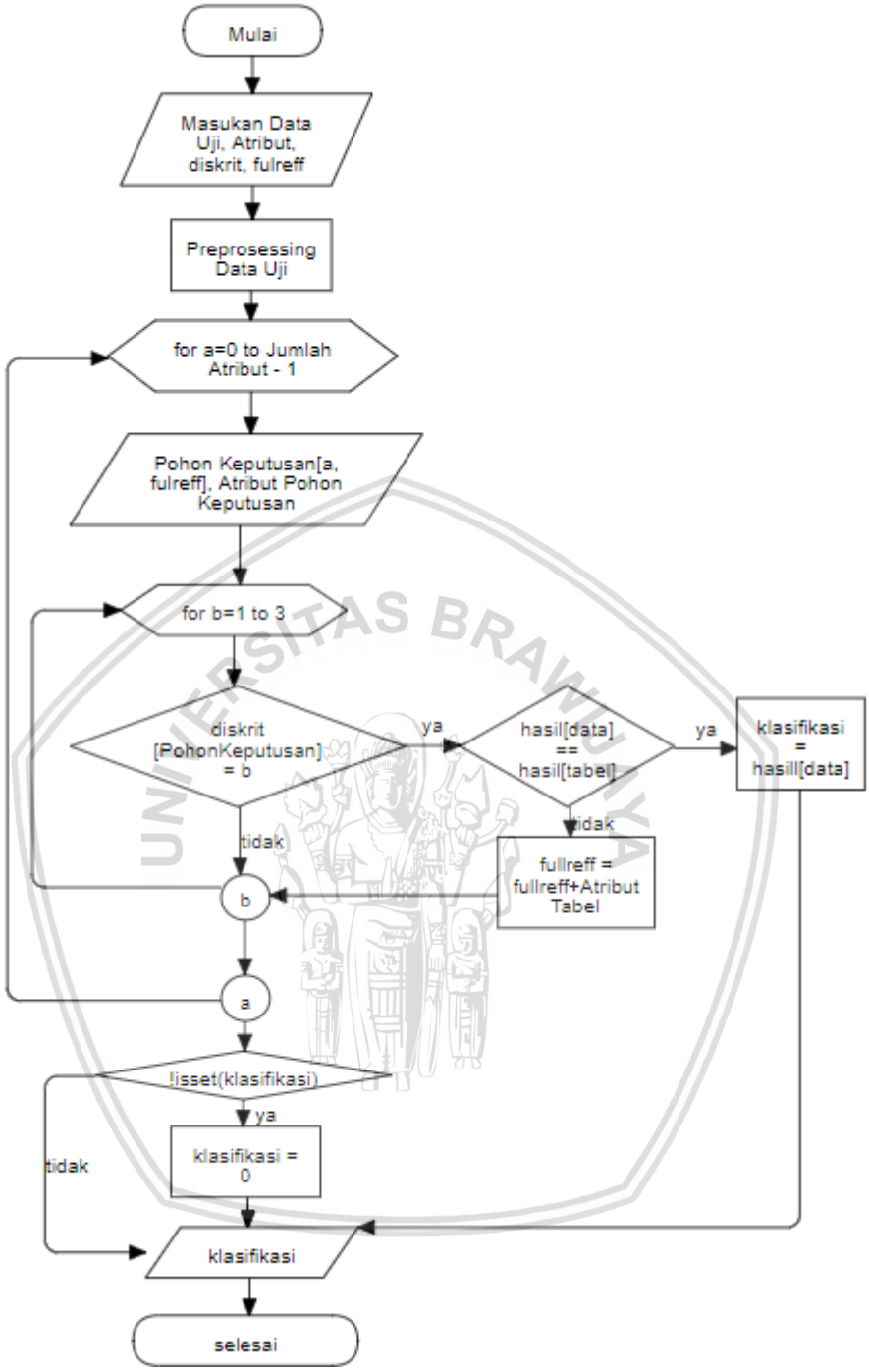
Gambar 4.5 Alur proses perhitungan entropi dan gain

Gambar 4.5 merupakan alur proses dari tahapan perhitungan daru entropi dan gain dari masing-masing atribut diantaranya B, C, N, O, P, S, F, Cl, Br, I, dan OH. Nilai tersebut akan digunakan mencari nilai entropi dan juga nilai Gain dari setiap atribut yang digunakan. Nilai Gain tertinggi akan dipilih sebagai akar kemudian entropi akar akan digunakan sebagai entropi total untuk mencari nilai noda di bawahnya. Keterangan lebih lanjut mengenai Gambar 4.5 adalah sebagai berikut:

1. Memasukan nilai atribut setelah didiskritisasi dan juga kelas klasifikasinya untuk proses pengenalan pada sistem data latih SMILES merupakan kelas klasifikasi tersebut.
2. Perulangan dari array atribut indeks 0 hingga indeks akhir dari akhir dari atribut.
3. Menghitung nilai entropi atribut untuk penentuan Gain dengan rumus $Entropi\ (S) = \sum_{i=0}^{k} -phi\ \log_2 phi$.
4. Menghitung nilai Gain dengan rumus $Gain\ (A) = Entropi\ (S) - \sum_{i=1}^{k} \frac{|s|}{|s_i|} x\ Entropi\ S_i$
5. Mencari nilai Gain terbesar untuk dipilih sebagai akar dan entropi akar akan digunakan sebagai entropi total untuk mencari nilai noda di bawahnya.
6. Membuat dan update pohon keputusan.
7. Menghapus nilai atribut yang terpilih dengan nilai gain terbesar.
8. Bila atribut belum sama dengan 0 maka akan terus melanjutkan proses perulangan.
9. Pohon keputusan paling akhir digunakan untuk proses pengujian data.

4.1.2.4 Pengujian Data

Tahap ketiga setelah proses pelatihan data maka akan dilakukan proses pengujian data menggunakan aturan dari pohon keputusan yang diciptakan berdasarkan proses pelatihan data. Tahapan lebih rinci pada tahap pengujian data akan ditunjukan pada Gambar 4.5.



Gambar 4.6 Alur proses pengujian data

Gambar 4.6 merupakan alur proses dari tahapan pengujian data yang dilakukan dengan masukan data uji notasi SMILES yang telah dilakukan *Preprocessing*, kemudian akan dilakukan pengecekan sesuai aturan pohon keputusan yang telah dibuat sehingga akan muncul hasil klasifikasi oleh sistem. Keterangan lebih lanjut mengenai Gambar 4.6 adalah sebagai berikut:

1. Memasukan nilai atribut setelah di *Preprocessing* dan atribut dan fullreff untuk proses pencarian rule pada database hasil proses peatiahan data.
2. Melakukan perulangan hingga banyak atribut karna rule noda adalah memiliki tingkatan maksimal sebesar banyak atribut dalam perulangan tersebut terdapat proses:.
3. Mengambil *rule* pohon keputusan dari database berdasarkan parameter atribut dan fullreff.
4. Percabangan perulangan sebanyak 3 karena kelompok diskritisasi sebesar 3 sehingga cabang dari pohon keputusan berjumlah 3 dalam perulangan tersebut terdapat proses:.
5. Bila diskrit bernilai sama dengan salah satu indeks cabang antara 1 hingga 3.
6. Bila hasil dari data berupa nilai atribut data sama dengan nilai atribut pada tabel maka klasifikasi notasi SMILE merupakan hasil dari data pada tabel.
7. Keluaran yang didadapatkan adalah hasil klasifikasi dari data uji notasi SMILES.

4.1.3 Contoh Perhitungan Manual

4.1.3.1 *Dataset*

Tabel 4.1 merupakan tabel dari dataset yang digunakan untuk contoh perhitungan manual algoritma C4.5. Pada *dataset* diambil 17 data yang mewakili sebanyak 2 kelas dan terdiri atas 15 data latih dan 2 data uji.

Tabel 4.1 *Dataset* untuk perhitungan manual

No	INPUTAN NOTASI SMILES	C	Cl	B	Br	O	OH	N	S	P	F	I	Kelas
1	CC(C)(C)NCC(COC1=CC=CC2=C1CCC(=O)N2)O.Cl	16	1	0	0	3	0	2	0	0	0	0	1
2	COC1=CC=CC=C1OCCNCC(COC2=CC=CC3=C2C4=CC=CC=C4N3)O	24	0	0	0	3	1	2	0	0	0	0	1
3	CCN(CC)C(=O)NC1=CC(=C(C=C1)OCC(CNC(C)(C)C)O)C(=O)C.Cl	20	1	0	0	3	1	3	0	0	0	0	1
4	CC(C)CC1=CC=C(C=C1)C(C)C(=O)O	13	0	0	0	1	1	0	0	0	0	0	2
5	CC1=C(C2=C(N1C(=O)C3=CC=C(C=C3)Cl	34	1	0	0	4	0	1	0	0	0	0	2

No	INPUTAN NOTASI SMILES	C	Cl	B	Br	O	OH	N	S	P	F	I	Kelas
	)C=CC(=C2)OC)CC(=O)OCC=C(C)CCC=C(C)CCC=C(C)C												
6	CC(C1=CC=CC(=C1)C(=O)C2=CC=CC=C2)C(=O)O	16	0	0	0	2	1	0	0	0	0	0	2
7	CN1CCN(CC1)CCCN2C3=CC=CC=C3SC4=C2C=C(C=C4)Cl	20	1	0	0	0	0	3	0	1	0	0	1
8	CC1=C(C(=O)N2CCCCC2=N1)CCN3CCC(CC3)C4=NOC5=C4C=CC(=C5)F	23	0	0	0	2	0	4	0	0	1	0	1
9	CC(CC1=CC=CC=C1)N.CC(CC1=CC=CC=C1)N.OS(=O)(=O)O	18	0	0	0	3	1	2	0	1	0	0	1
10	CCNC(=O)N(CCCN(C)C)C(=O)C1CC2C(CC3=CNC4=CC=CC2=C34)N(C1)CC=C	26	0	0	0	2	0	5	0	0	0	0	1
11	COC1=C(C=C(C=C1)C(=O)NC2=CC=CC=C2)N	14	0	0	0	2	0	2	0	0	0	0	1
12	CC(=O)OCC(=O)C12C(CC3C1(CC(C4(C3CC(C5=CC(=O)C=CC54C)F)F)O)C)OC(O2)(C)C	26	0	0	0	5	2	0	0	0	2	0	2
13	CC1(C2CCC3(C(C2(CCC1OC4C(C(C(C(O4)C(=O)O)O)O)O)OC5C(C(C(C(O5)C(=O)O)O)O)O)O)C)C(=O)C=C6C3(CCC7(C6CC(CC7)(C)C(=O)O)C)C)C	42	0	0	0	6	10	0	0	0	0	0	2
14	CS(=O)(=O)NC1=C(C=C(C=C1)[N+](=O)[O-])OC2=CC=CC=C2	13	0	0	0	4	0	2	0	1	0	0	2
15	CC1=C(C2=C(N1C(=O)C3=CC=C(C=C3)Cl	34	1	0	0	4	0	1	0	0	0	0	2

No	INPUTAN NOTASI SMILES	C	Cl	B	Br	O	OH	N	S	P	F	I	Kelas
	)C=CC(=C2)OC)CC(=O)OCC=C(C)CCC=C(C)CCC=C(C)C												
1	CC1=C(C(=O)N2CCCCC2=N1)CCN3CCC(C3)C4=NOC5=C4C=CC(=C5)F	23	0	0	0	2	0	4	0	0	1	0	?
2	CC1(C2CCC3(C(C2(CCC1OC4C(C(C(C(O4)C(=O)O)O)O)OC5C(C(C(C(O5)C(=O)O)O)O)O)C)C(=O)C=C6C3(CCC7(C6CC(CC7)(C)C(=O)O)C)C)C)C	43	0	0	0	6	10	0	0	0	0	0	?

Terdapat 11 atribut yang terdapat pada Tabel 4.1, berikut merupakan keterangan dari masing-masing atribut yang akan digunakan pada penelitian menggunakan algoritma C4.5:

1. C = Jumlah atom C pada notasi SMILES
2. Cl = Jumlah atom Cl pada notasi SMILES
3. B = Jumlah atom B pada notasi SMILES
4. Br = Jumlah atom Br pada notasi SMILES
5. O = Jumlah atom O pada notasi SMILES
6. OH = Jumlah atom OH pada notasi SMILES
7. N = Jumlah atom N pada notasi SMILES
8. S = Jumlah atom S pada notasi SMILES
9. P = Jumlah atom P pada notasi SMILES
10. F = Jumlah atom F pada notasi SMILES
11. I = Jumlah atom I pada notasi SMILES

Data tersebut kemudian akan dikonversi dengan membagi setiap nilai atribut tersebut dengan nilai panjang notasi SMILES, kemudian setiap nilai atribut akan dikalikan dengan nilai 100 untuk menghilangkan bilangan desimal, berikut hasil dari nilai masing-masing dataset setelah dibagi dengan nilai panjang notasi SMILES dan setiap nilai atribut dikalikan dengan nilai 100 ditunjukan pada Tabel 4.2 di bawah ini.

Tabel 4.2 *Dataset* setelah dibagi dengan nilai panjang notasi SMILES dan setiap nilai atribut dikalikan dengan nilai 100.

No	INPUTAN NOTASI SMILES	C	Cl	B	Br	O	OH	N	S	P	F	I	Kelas
1	CC(C)(C)NCC(COC1=CC=CC2=C1CCC(=O)N2)O.Cl	46	3	0	0	9	0	6	0	0	0	0	1
2	COC1=CC=CC=C1OCCNCC(COC2=CC=CC3=C2C4=CC=CC=C4N3)O	59	0	0	0	7	2	5	0	0	0	0	1
3	CCN(CC)C(=O)NC1=CC(=C(C=C1)OCC(CNC(C)(C)C)O)C(=O)C.Cl	41	2	0	0	6	2	6	0	0	0	0	1
4	CC(C)CC1=CC=C(C=C1)C(C)C(=O)O	48	0	0	0	4	4	0	0	0	0	0	2
5	CC1=C(C2=C(N1C(=O)C3=CC=C(C=C3)Cl)C=CC(=C2)OC)CC(=O)OCC=C(C)CCC=C(C)CCC=C(C)C	49	1	0	0	6	0	1	0	0	0	0	2
6	CC(C1=CC=CC(=C1)C(=O)C2=CC=CC=C2)C(=O)O	46	0	0	0	6	3	0	0	0	0	0	2
7	CN1CCN(CC1)CCCN2C3=CC=CC=C3SC4=C2C=C(C=C4)Cl	57	3	0	0	0	0	9	0	3	0	0	1
8	CC1=C(C(=O)N2CCCCC2=N1)CCN3CCC(CC3)C4=NOC5=C4C=CC(=C5)F	51	0	0	0	4	0	9	0	0	2	0	1
9	CC(CC1=CC=CC=C1)N.CC(CC1=CC=CC=C1)N.OS(=O)(=O)O	42	0	0	0	7	2	5	0	2	0	0	1
10	CCNC(=O)N(CCCN(C)C)C(=O)C1CC2C(CC3=CNC4=CC=CC2=C34)N(C1)CC=C	50	0	0	0	4	0	10	0	0	0	0	1
11	COC1=C(C=C(C=C1)C(=O)NC2=CC=CC=C2)N	45	0	0	0	6	0	6	0	0	0	0	1

No	INPUTAN NOTASI SMILES	C	Cl	B	Br	O	OH	N	S	P	F	I	Kelas
12	CC(=O)OCC(=O)C12C(CC3C1(CC(C4(C3CC(C5=CC(=O)C=CC54C)F)F)O)C)OC(O2)(C)C	45	0	0	0	9	3	0	0	0	3	0	2
13	CC1(C2CCC3(C(C2(CCC1OC4C(C(C(C(O4)C(=O)O)O)O)OC5C(C(C(C(O5)C(=O)O)O)O)O)C)C(=O)C=C6C3(CCC7(C6CC(CC7)(C)C(=O)O)C)C)C)C	45	0	0	0	6	11	0	0	0	0	0	2
14	CS(=O)(=O)NC1=C(C=C(C=C1)[N+](=O)[O-])OC2=CC=CC=C2	32	0	0	0	10	0	5	0	2	0	0	2
15	CC1=C(C2=C(N1C(=O)C3=CC=C(C=C3)Cl)C=CC(=C2)OC)CC(=O)OCC=C(C)CCC=C(C)CCC=C(C)C	49	1	0	0	6	0	1	0	0	0	0	2
1	CC1=C(C(=O)N2CCCCC2=N1)CCN3CCC(CC3)C4=NOC5=C4C=CC(=C5)F	51	0	0	0	4	0	9	0	0	2	0	?
2	CC1(C2CCC3(C(C2(CCC1OC4C(C(C(C(O4)C(=O)O)O)O)OC5C(C(C(C(O5)C(=O)O)O)O)O)C)C(=O)C=C6C3(CCC7(C6CC(CC7)(C)C(=O)O)C)C)C)C	42	0	0	0	6	10	0	0	0	0	0	?

Untuk proses penunjang dalam perhitungan dan pembentukan pohon keputusan maka dibutuhkan kelas-kelas yang mewakili nilai dari sebuah atribut tersebut. Kelas tersebut akan mengubah nilai atribut yang bersifat *continue* menjadi bersifat diskrit. Metode yang digunakan untuk proses diskritisasi data adalah dengan menggunakan metode *Entropy Based*. Untuk pembobotan maka akan dicek semua kemungkinan yang menghasilkan nilai Gain tertinggi dari suatu atribut fitur, Menurut hasil perhitungan maka akan dihasilkan nilai batas seperti pada Tabel 4.3 berikut:

Tabel 43. Data hasil batas teknik diskritisasi *entropy based*

No	C	Cl	B	Br	O	OH	N	S	P	F	I
1	31	0	0	0	2	2	2	0	0	0	0

Nilai tersebut didapatkan dari perulangan perhitungan dari batas nilai minimal dari suatu fitur atribut hingga batas maksimal dari fitur atribut. Untuk perhitungan manual yang dicontohkan maka akan mengambil contoh fitur atribut dari OH. Untuk menghitung entropi total maka akan digunakan rumus:

$Entropi\ (S) = \sum_{i=0}^{k} -phi\ \log_2 phi$.....(2.1)

Keterangan :

S : Himpunan (*dataset*) kasus

k : Banyaknya partisi S

phi : Probabilitaas yang didapat dari *Sum*(Ya) atau *Sum*(Tidak) dibagi total kasus

Dimana diketahui:

$S : 15$

$k : 2$ (kelas 1, kelas 2)

$phi1 : 0{,}5$

$phi2 : 0{,}5$

$$Entropi\ (S) = (-0{,}5\ \log_2 0{,}5) + (-0{,}5\ \log_2 0{,}5) = 1$$

Untuk selanjutnya merupakan perhitungan entropi setiap atribut, penulis mengambil contoh menghitung nilai dari atribut OH, untuk mencari nilai dari entropi dari atribut OH maka diketahui:

OH kelompok 1

Dimana diketahui:

$S : 9$

$k : 2$ (kelas 1, kelas 2)

$phi1 : 5/9$

$phi2 : 4/9$

$$Entropi\ (OH\ kelompok\ 1) = (-\frac{5}{9}\log_2\frac{5}{9}) + (-\frac{4}{9}\log_2\frac{4}{9}) = 0{,}99107$$

OH kelompok 2

Dimana diketahui:

$S : 3$

$k : 2$ (kelas 1, kelas 2)

$phi1 : \frac{3}{3} = 1$

$phi2 : 0$

$$Entropi\ (OH\ kelompok\ 2) = (-1\log_2 1\ ) + 0 = 0$$

OH kelompok 3

Dimana diketahui:

$S : 3$

$k : 2$ (kelas 1, kelas 2)

$phi1 : \frac{2}{3}$

$phi2 : \frac{1}{3}$

$$Entropi\ (OH\ kelompok\ 3) = (-2/3 \log_2 2/3\ ) + (-1/3\ \log_2 1/3)$$
$$= 0{,}9182$$

$$Gain\ (OH) = 1 - \left(\frac{9}{16} \times\ 0{,}99107 + \frac{3}{16} \times\ 0 + \frac{3}{16} \times\ 0{,}9182\right) = 0{,}27$$

Dengan menggunakan cara yang sama untuk atribut yang lain maka akan dihitung juga nilai masing-masing entropi dan gain dari setiap atribut yang ada. Sehingga menghasilkan batas diskritisasi terbaik. Berikut merupakan konversi dari teknik diskritisasi *entropy based* tercantum pada tabel 4.4.

Tabel 4.4 *Dataset* setelah dilakukan proses diskritisasi

No	INPUTAN NOTASI SMILES	C	Cl	B	Br	O	OH	N	S	P	F	I	Kelas
1	CC(C)(C)NCC(COC1=CC=CC2=C1CCC(=O)N2)O.Cl	46	3	0	0	9	0	6	0	0	0	0	1
2	COC1=CC=CC=C1OCCNCC(COC2=CC=CC3=C2C4=CC=CC=C4N3)O	59	0	0	0	7	2	5	0	0	0	0	1
3	CCN(CC)C(=O)NC1=CC(=C(C=C1)OCC(CNC(C)(C)C)O)C(=O)C.Cl	41	2	0	0	6	2	6	0	0	0	0	1
4	CC(C)CC1=CC=C(C=C1)C(C)C(=O)O	48	0	0	0	4	4	0	0	0	0	0	2
5	CC1=C(C2=C(N1C(=O)C3=CC=C(C=C3)Cl)C=CC(=C2)OC)CC(=O)OCC=C(C)CCC=C(C)CCC=C(C)C	49	1	0	0	6	0	1	0	0	0	0	2

No	INPUTAN NOTASI SMILES	C	Cl	B	Br	O	OH	N	S	P	F	I	Kelas
6	CC(C1=CC=CC(=C1)C(=O)C2=CC=CC=C2)C(=O)O	46	0	0	0	6	3	0	0	0	0	0	2
7	CN1CCN(CC1)CCCN2C3=CC=CC=C3SC4=C2C=C(C=C4)Cl	57	3	0	0	0	0	9	3	0	0	0	1
8	CC1=C(C(=O)N2CCCCC2=N1)CCN3CCC(CC3)C4=NOC5=C4C=CC(=C5)F	51	0	0	0	4	0	9	0	0	2	0	1
9	CC(CC1=CC=CC=C1)N.CC(CC1=CC=CC=C1)N.OS(=O)(=O)O	42	0	0	0	7	2	5	2	0	0	0	1
10	CCNC(=O)N(CCCN(C)C)C(=O)C1CC2C(CC3=CNC4=CC=CC2=C34)N(C1)CC=C	50	0	0	0	4	0	10	0	0	0	0	1
11	COC1=C(C=C(C=C1)C(=O)NC2=CC=CC=C2)N	45	0	0	0	6	0	6	0	0	0	0	1
12	CC(=O)OCC(=O)C12C(CC3C1(CC(C4(C3CC(C5=CC(=O)C=CC54C)F)F)O)C)OC(O2)(C)C	45	0	0	0	9	3	0	0	0	3	0	2
13	CC1(C2CCC3(C(C2(CCC1OC4C(C(C(C(O4)C(=O)O)O)O)O)OC5C(C(C(C(O5)C(=O)O)O)O)O)C)C(=O)C=C6C3(CCC7(C6CC(CC7)(C)C(=O)O)C)C)C)C	45	0	0	0	6	11	0	0	0	0	0	2
14	CS(=O)(=O)NC1=C(C=C(C=C1)[N+]	29	0	0	0	10	0	5	0	0	0	0	2

No	INPUTAN NOTASI SMILES	C	Cl	B	Br	O	OH	N	S	P	F	I	Kelas
	(=O)[O-])OC2=CC=CC=C2												
15	CC1=C(C2=C(N1C(=O)C3=CC=C(C=C3)Cl)C=CC(=C2)OC)CC(=O)OCC=C(C)CCC=C(C)CCC=C(C)C	49	1	0	0	6	0	1	0	0	0	0	2

4.1.3.2 Menghitung Nilai Entropi dan Gain Atribut

Setelah mengumpulkan data dan membentuknya menjadi *dataset* yang telah siap untuk diolah, maka langkah selanjutnya pada algoritma C4.5 adalah menghitung nilai entropi dan gain dari setiap atribut dan membentuk sebuah *rule* atau desain dari pohon keputusan, mulai dari akar hingga noda terakhir dari pohon keputusan.

Pada tabel 4.4 tercantum data nilai entropi dan gain dari masing-masing atribut karena pada tahap awal merupakan penentuan akar maka entropi yang digunakan merupakan entropi total. Untuk menghitung entropi total maka akan digunakan rumus:

$Entropi\ (S) = \sum_{i=0}^{k} -phi\ \log_2 phi$ .....(2.1)

Keterangan :

S : Himpunan (*dataset*) kasus

k : Banyaknya partisi S

phi : Probabilitaas yang didapat dari *Sum*(Ya) atau *Sum*(Tidak) dibagi total kasus

Dimana diketahui:

$S : 8$

$k : 2$ (kelas 1, kelas 2)

$phi1 : 0{,}5$

$phi2 : 0{,}5$

$$Entropi\ (S) = (-0{,}5\ \log_2 0{,}5) + (-0{,}5\ \log_2 0{,}5) = 1$$

Untuk selanjutnya merupakan perhitungan entropi setiap atribut, penulis mengambil contoh menghitung nilai dari atribut N, untuk mencari nilai dari entropi dari atribut N maka diketahui:

N kelompok 1

Dimana diketahui:

$S : 4$

$k : 2$ (kelas 1, kelas 2)

$phi1 : \frac{0}{4} = 0$

$phi2 : \frac{4}{4} = 1$

$$Entropi\ (\text{N kelompok } 1) = 0 + (-1\ \log_2 1) + = 0$$

N kelompok 2

Dimana diketahui:

$S : 1$

$k : 2$ (kelas 1, kelas 2)

$phi1 : \frac{0}{1} = 0$

$phi2 : \frac{1}{1} = 1$

$$Entropi\ (\text{N kelompok } 2) = 0 + (-1\ \log_2 1) = 0$$

N kelompok 3

Dimana diketahui:

$S : 10$

$k : 2$ (kelas 1, kelas 2)

$phi1 : \frac{8}{10} = 0{,}8$

$phi2 : \frac{2}{10} = 0{,}2$

$$Entropi\ (\text{N kelompok } 3) = (-0{,}8 \log_2 0{,}8\ ) + (-0{,}2\ \log_2 0{,}2) = 0{,}72912$$

$$Gain\ (N) = 1 - \left(\frac{0}{15} \times 0 + \frac{0}{15} \times 0 + \frac{10}{15} \times 0{,}72912\right) = 0{,}5153$$

Dengan menggunakan cara yang sama untuk atribut yang lain maka akan dihitung juga nilai masing-masing entropi dan gain dari setiap atribut yang ada. Berikut merupakan tabel dari perhitungan nilai gain dan entropi masing-masing atribut tercantum pada tabel 4.5

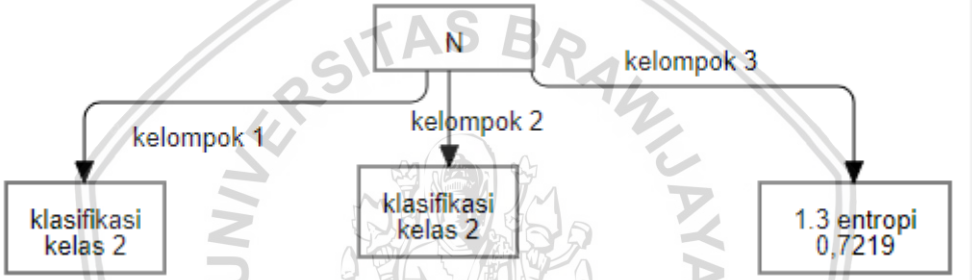
Tabel 4.5 Gain dan Entropi pada tingkatan akar

Tabel Entropi Total						
1						
Atribut	**Kelompok**	**s**	**Kelas 1**	**Kelas 2**	**Entropi**	**Gain**
C						0,854607
	Kelompok 1	0	0,000001	0,000001	0	
	Kelompok 2	1	0,000001	1	0	
	Kelompok 3	14	8	6	0,98522	

Tabel Entropi Total						
1						
Atribut	**Kelompok**	**s**	**Kelas 1**	**Kelas 2**	**Entropi**	**Gain**
Cl						0,16362
	Kelompok 1	13	8	5	0,96123	
	Kelompok 2	0,000001	0,000001	0,000001	0	
	Kelompok 3	2	0,000001	2	0	
B						0
	Kelompok 1	15	7	8	0,99679	
	Kelompok 2	0,000001	0,000001	0,000001	0	
	Kelompok 3	0,000001	0,000001	0,000001	0	
Br						0
	Kelompok 1	15	7	8	0,99679	
	Kelompok 2	0,000001	0,000001	0,000001	0	
	Kelompok 3	0,000001	0,000001	0,000001	0	
O						0,06336
	Kelompok 1	1	1	0,000001	0	
	Kelompok 2	0,000001	0,000001	0,000001	0	
	Kelompok 3	14	7	7	1	
OH						0.08696
	Kelompok 1	11	7	4	0.94566	
	Kelompok 2	0,000001	0,000001	0,000001	0	
	Kelompok 3	4	1	3	0.81127	
N						0,515315
	Kelompok 1	4	0,000001	4	0	
	Kelompok 2	1	0,000001	1	0	
	Kelompok 3	10	8	2	0,7219	
S						0,0131
	Kelompok 1	12	6	6	1	
	Kelompok 2	0,000001	0,000001	0,000001	0	
	Kelompok 3	3	2	1	0,9182	
P						0
	Kelompok 1	15	7	8	0,99679	
	Kelompok 2	0,000001	0,000001	0,000001	0	
	Kelompok 3	0,000001	0,000001	0,000001	0	
F						0,0771
	Kelompok 1	14	8	6	0,9852	
	Kelompok 2	0,000001	0,000001	0,000001	0	
	Kelompok 3	1	0	1	0	

Tabel Entropi Total						
1						
Atribut	**Kelompok**	**s**	**Kelas 1**	**Kelas 2**	**Entropi**	**Gain**
I						0
	Kelompok 1	15	7	8	0,99679	
	Kelompok 2	0,000001	0,000001	0,000001	0	
	Kelompok 3	0,000001	0,000001	0,000001	0	

Berdasarkan data nilai gain dan entropi masing-masing atribut yang terpapar pada Tabel 4.5 maka dapat dilihat bahwa nilai gain tertinggi dimiliki oleh atribut O. Sehingga pohon keputusan sementara akan terbentuk seperti pada Gambar 4.7



Gambar 4.7 Pohon keputusan pada tingkat akar

Setelah berhasil menemukan atribut untuk akar yaitu atribut N maka akan dihitung untuk mencari noda pada tingkat 1 untuk percabangan pohon keputusan yang selanjutnya. Untuk kelompok diskrit 1 dan 2 sudah diketahui hasil akhir berupa kelas klasifikasi 2. Untuk mencari noda tingkat 1 maka harus dicari cabang 1.3 seperti yang nampak pada Gambar 4.7. Pertama kali makan dimulai dengan mencari cabang 1.3 dengan cabang N kelompok 3, bila sebelumnya entropi menggunakan entropi total sebagai parameter maka di cabang 1.3 akan menggunakaan entropi dari N kelompok 3 yaitu sebesar 0,7219 dan menggunakan S atau Himpunan (*dataset*) kasus pada N kelompok 3 yaitu sebanyak 10. Untuk selanjutnya merupakan perhitungan entropi setiap atribut, penulis mengambil contoh menghitung nilai dari atribut C, untuk mencari nilai dari entropi dari atribut C maka diketahui:

$$Entropi\ (N\ Kel.3) = 0{,}7219$$

C kelompok 1

Dimana diketahui:

$S : 0$

$k : 2$ (kelas 1, kelas 2)

$phi1 : 0{,}000001$
$phi2 : 0{,}000001$

$$Entropi\ (C\ Kel.1) = (-0{,}000001 \log_2 0{,}000001) + (-0{,}000001\ \log_2 0{,}000001) + (-0{,}000001 \log_2 0{,}000001) = 0$$

C kelompok 2
Dimana diketahui:
$S : 1$
$k : 1$(kelas 1, kelas 2)
$phi1 : 0{,}000001$
$phi2 : \frac{1}{1} = 1$

$$Entropi\ (C\ Kel.2) = (-0{,}000001 \log_2 0{,}000001) + (-1 \log_2 1) = 3{,}98631\text{E}-05$$

C kelompok 3
Dimana diketahui:
$S : 9$
$k : 2$ (kelas 1, kelas 2)
$phi1 : \frac{8}{9}$
$phi2 : \frac{1}{9}$

$$Entropi\ (S) = (-8/9 \log_2 8/9) + (1/9 \log_2 1/9) = 0{,}2688$$

$$Gain\ (C) = 0{,}7219 - \left(\frac{0}{4}\ x\ 0 + \frac{1}{4}\ x\ 3{,}98631\text{E}-05 + \frac{3}{4}\ x\ 1{,}43444\text{E}-05\right) = 0{,}811262883$$

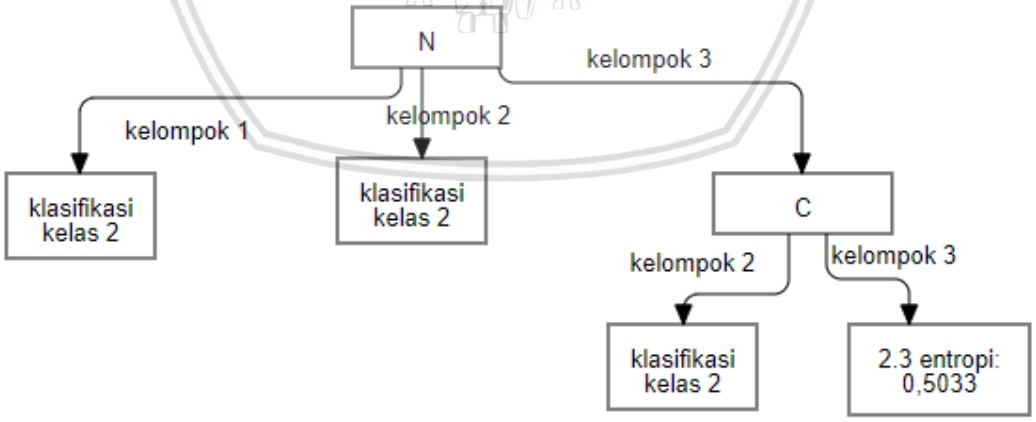
Dengan menggunakan cara yang sama untuk atribut yang lain maka akan dihitung juga nilai masing-masing entropi dan gain dari setiap atribut yang ada. Berikut merupakan tabel dari perhitungan nilai gain dan entropi masing-masing atribut tercantum pada tabel 4.6

Tabel 4.6 Gain dan Entropi pada tingkatan noda 1 N kelompok 3

Tabel N Kelompok 3							
node	**Atribut**	**Kelompok**	**S**	**Kelas 1**	**Kelas 2**	**Entropi**	**Gain**
0	N		10			0,7219	
1	C						0,26883
		Kelompok 1	0,000001	0,000001	0,000001	0	
		Kelompok 2	1	0,000001	1	3,98631E-05	
		Kelompok 3	9	8	1	1,43444E-05	

Tabel N Kelompok 3							
node	Atribut	Kelompok	S	Kelas 1	Kelas 2	Entropi	Gain
	Cl						0.2688
		Kelompok 1	9	8	1	0.50325	
		Kelompok 2	0,000001	0,000001	0,000001	0	
		Kelompok 3	1	0,000001	1	0	
	OH						0,801262883
		Kelompok 1	9	7	2	1,43444E-05	
		Kelompok 2	0,000001	0,000001	0,000001	3,98631E-05	
		Kelompok 3	1	1	0,000001	0	
	S						0.0322
		Kelompok 1	7	6	1	0.9182	
		Kelompok 2	0,000001	0,000001	0,000001	0	
		Kelompok 3	3	2	1	0.59167	
	F						0.7219
		Kelompok 1	10	8	2	0,918303006	
		Kelompok 2	0,00001	0,000001	0,000001	0	
		Kelompok 3	0,000001	0,000001	0,000001	0	

Berdasarkan data nilai gain dan entropi masing-masing atribut yang terpapar pada Tabel 4.6 maka dapat dilihat bahwa nilai gain tertinggi dimiliki oleh atribut C. Sehingga pohon keputusan sementara akan terbentuk seperti pada Gambar 4.8.



Gambar 4.8 Pohon keputusan pada tingkatan noda 1 N kelompok 3

Setelah berhasil menemukan atribut untuk noda 1 yaitu atribut C dan hasilnya adalah bila nilai C masuk pada kelompok 2 maka kelas klasifikasinya merupakan kelas 2 dan bila nilai C masuk pada kelompok 3 maka kelas klasifikasinya merupakan kelas 1 maka akan dihitung untuk mencari noda pada

tingkat 1. Karena tingkatan noda 1 C kelompok 1 merupakan akhir dari noda maka selanjutnya akan kembali lagi untuk menghitung tingkatan noda 1 C kelompok 3. Sedangkan bila sebelumnya entropi menggunakan entropi total sebagai parameter maka di cabang 2.3 akan menggunakaan entropi dari C kelompok 3 yaitu sebesar 0,5033 dan menggunakan S atau Himpunan (*dataset*) kasus pada O kelompok 2 yaitu sebanyak 9. Untuk selanjutnya merupakan perhitungan entropi setiap atribut, penulis mengambil contoh menghitung nilai dari atribut Cl untuk mencari nilai gain dan entropi dari atribut Cl maka diketahui:

$$Entropi\ (C\ Kel.3) = 0{,}5033$$

Cl kelompok 1
Dimana diketahui:
$S : 8$
$k : 2$ (kelas 1, kelas 2)
$phi1 : 1$
$phi2 : 0$

$$Entropi\ (Cl\ Kel.1) = (-1 \log_2 1) + (-0 \log_2 0) = 0{,}0002$$

Cl kelompok 2
Dimana diketahui:
$S : 0$
$k : 2$ (kelas 1, kelas 2)
$phi1 : 0$
$phi2 : 0$

$$Entropi\ (Cl\ Kel.2) = 0$$

Cl kelompok 3
Dimana diketahui:
$S : 1$
$k : 2$ (kelas 1, kelas 2)
$phi1\ 0$
$phi2 : 1$

$$Entropi\ (S) = 0 + (-1 \log_2 1) = 0{,}00132$$

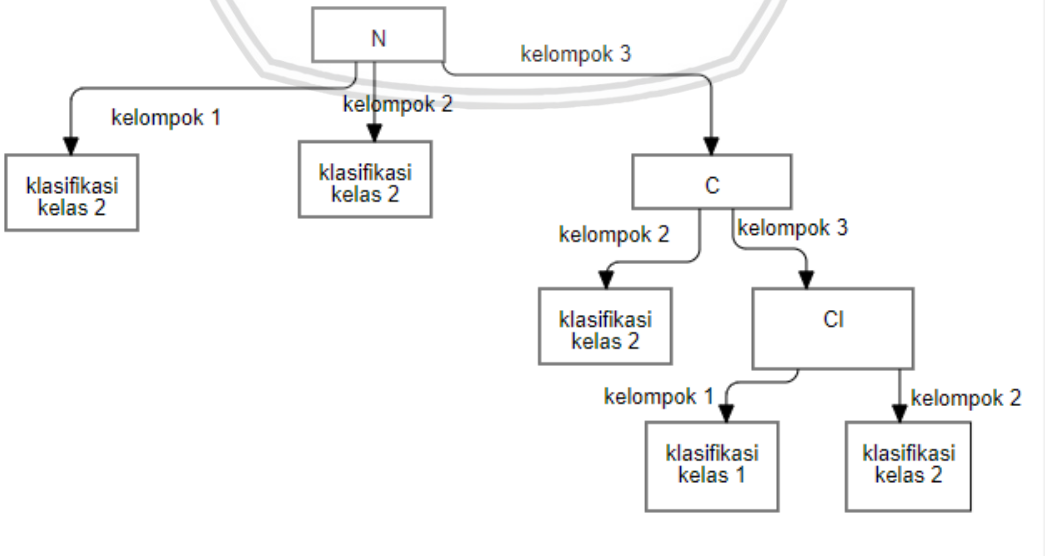
$$Gain\ (Cl) = 0{,}5033 - \left(\frac{8}{9}\ x\ 0{,}0002 + \frac{0}{9}\ x\ 0 + \frac{1}{9}\ x\ 0{,}00132\right) = 0{,}5029$$

Dengan menggunakan cara yang sama untuk atribut yang lain maka akan dihitung juga nilai masing-masing entropi dan gain dari setiap atribut yang ada. Berikut merupakan tabel dari perhitungan nilai gain dan entropi masing-masing atribut tercantum pada tabel 4.7

Tabel 4.7 Gain dan Entropi pada tingkatan noda 3 C kelompok 3

Tabel O Kelompok 2							
Node	**Atribut**	**Kelompok**	**S**	**Kelas 1**	**Kelas 2**	**Entropi**	**Gain**
1	C		9			0,5033	
	Cl						0.5029
		Kelompok 1	8	8	0,000001	0.0002	
		Kelompok 2	0,000001	0,000001	0,000001	0	
		Kelompok 3	1	0,000001	1	0.0013	
	OH						0.0199
		Kelompok 1	8	7	1	0.5435	
		Kelompok 2	0,000001	0,000001	0,000001	0	
		Kelompok 3	1	1	0,000001	0.0013	
	S						0.0429
		Kelompok 1	7	6	1	0.5916	
		Kelompok 2	0,000001	0,00001	0,000001	0	
		Kelompok 3	2	2	0,000001	0.0007	
	F						0
		Kelompok 1	9	8	1	0,503	
		Kelompok 2	0,000001	0,000001	0,000001	0	
		Kelompok 3	0,000001	0,000001	0,000001	0	

Berdasarkan data nilai gain dan entropi masing-masing atribut yang terpapar pada Tabel 4.7 maka dapat dilihat bahwa nilai gain tertinggi dimiliki oleh atribut C. Sehingga pohon keputusan akhir akan terbentuk seperti pada Gambar 4.9



Gambar 4.9 Pohon keputusan pada tingkatan akhir

4.1.3.2 Menghitung Nilai Akurasi pada Data Uji

Berikut merupakan 2 data uji yang akan diujikan pada bentuk pohon keputusan akhir yang telah melalui proses diskritisasi yang tercantum pada Tabel 4.10.

Tabel 4.10 Data Uji

No	INPUTAN NOTASI SMILES	C	Cl	B	Br	O	OH	N	S	P	F	I	Kelas
1	COC1=CC=CC=C1OCCNCC(COC2=CC=CC3=C2C4=CC=CC=C4N3)O	3	1	1	1	1	1	2	1	1	3	1	?
2	CCN(CC)C(=O)NC1=CC(=C(C=C1)OCC(CNC(C)(C)C)O)C(=O)C.Cl	3	1	1	1	1	3	1	1	1	1	1	?

Berikut merupakan data yang telah diklasifikasi melalui rule pada pohon keputusan akhir yang dicocokan tercantum pada Tabel 4.10.

Tabel 4.10 Hasil Pengujian

No	INPUTAN NOTASI SMILES	Kelas	Valid
1	COC1=CC=CC=C1OCCNCC(COC2=CC=CC3=C2C4=CC=CC=C4N3)O	1	Valid
2	CCN(CC)C(=O)NC1=CC(=C(C=C1)OCC(CNC(C)(C)C)O)C(=O)C.Cl	2	Valid

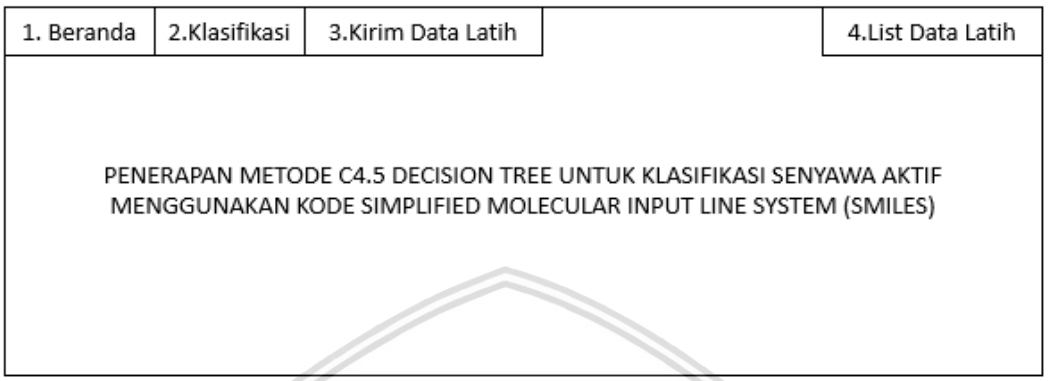
Dari 3 data uji yang dicocokan maka menunjukan bahwa dari 3 data uji, yang benar melalui pohon keputusan adalah 2 dari 2 data uji sehingga prosentase kebenaran untuk pohon keputusan sebesar 100%.

4.1.4 Perancangan Antarmuka

Perancangan antarmuka pada sistem dibutuhkan sebelum tahapan implementasi, hal ini dikarenakan agar diketahui apa saja yang perlu atau tidak untuk ditampilkkan oleh sistem.

4.1.4.1 Perancangan Antarmuka Halaman Beranda

Tampilan halaman awal pada sistem seperti ditunjukan pada Gambar 4.8 dimana pada halaman beranda terdapat menu-menu awal yang dibutuhkan bagi pengguna sistem.



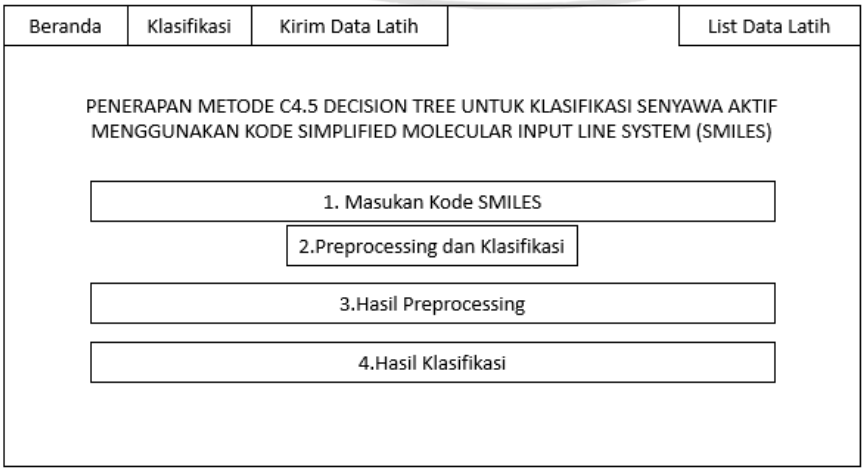
Gambar 4.7 Perancangan Antarmuka Halaman Beranda

Keterangan Gambar 4.7

1. Halaman dan bar untuk Beranda.
2. Halaman dan bar untuk Klasifikasi penentuan kelas SMILES.
3. Halaman dan bar untuk kirim data latih ke database dan proses untuk pelatihan data.
4. Halaman dan bar untuk menampillkan daftar data latih yang ada di database.

4.1.4.2 Perancangan Antarmuka Halaman Klasifikasi

Tampilan halaman klasifikasi pada sistem seperti ditunjukan pada Gambar 4.8 dimana pada halaman klasifikasi terdap masukan dan keluaran pada proses klasifikasi.



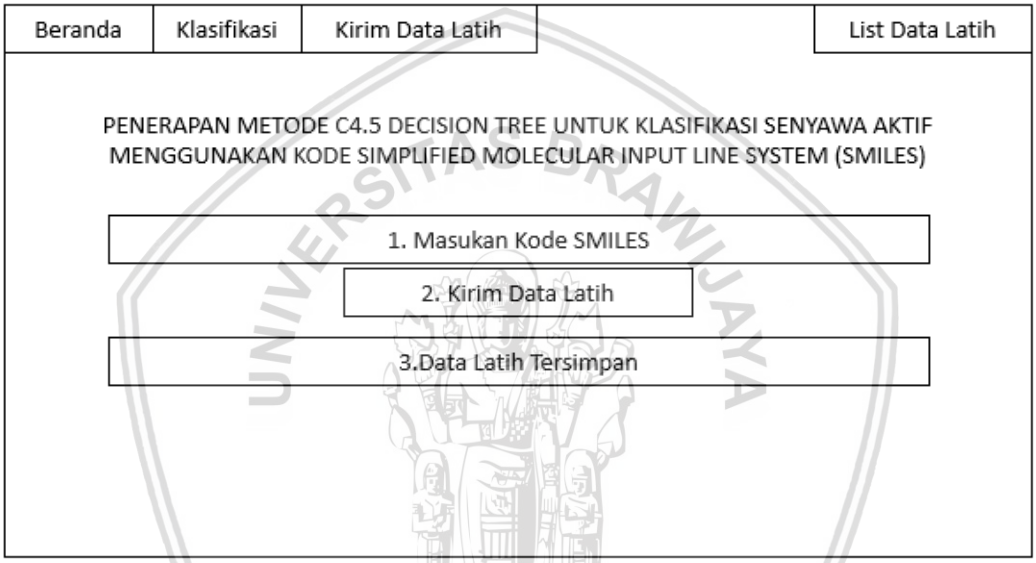
Gambar 4.8 Perancangan Antarmuka Halaman Klasifikasi

Keterangan Gambar 4.8

1. Inputan untuk memasukan notasi SMILES
2. Tombol untuk memuulai preprocessing dan kasifikasi
3. Keluaran input setelah di preprocessing
4. Keluaran hasil klasifikasi

4.1.4.3 Perancangan Antarmuka Halaman Kirim Data Latih

Tampilan halaman Kirim Data Latih pada sistem seperti ditunjukan pada Gambar 4.8 dimana pada halaman Kirim Data Latih terdapat masukan dan keluaran yang proses klasifikasi.



Gambar 4.8 Perancangan Antarmuka Halaman Klasifikasi

Keterangan Gambar 4.8

1. Inputan untuk memasukan notasi SMILES
2. Tombol untuk memulai preprocessing
3. Keluaran bila berhasil disimpan di database

BAB 5 IMPLEMENTASI

Pada bab ini akan membahas tentang implementasi sistem yang menggunakan algoritma C4.5 dalam melakukan klasifikasi notasi SMILES sesuai dengan hasil yang didapatkan pada tahap perancangan sistem.

5.1 Lingkungan Implementasi

Pada sub bab lingkungan implementasi akan dibahas mengenai lingkungan perangkat keras beserta lingkungan perangkat lunak yang digunakan pada proses pengembangan perangkat lunak yang berfungsi untuk proses klasifikasi notasi SMILES menggunakan algoritma C4.5.

5.1.1 Lingkungan Perangkat Keras

Perangkat keras yang digunakan pada proses pengembangan perangkat lunak klasifikasi notasi SMILES dengan menggunakan algoritma C4.5 diantaranya sebagai berikut :

(1) One 14 Intel(R) Core(TM) i3-5005U CPU Windows Operating System 64 bit
(2) Installed Memory (RAM) 4 GB
(3) Hardisk
(4) Monitor 11 inch

5.1.2 Spesifikasi Perangkat Keras

Perangkat lunak yang digunakan pada proses pengembangan perangkat lunak klasifikasi notasi SMILES dengan menggunakan algoritma C4.5 diantaranya sebagai berikut :

(1) Sistem Operasi Windows 10 64 bit
(2) Netbeans IDE versi 8.2
(3) XAMPP Control Panel v3.2.2
(4) Google Chrome Versi 65.0.3325.162 64 bit
(5) MySql
(6) Frameword Codeigniter
(7) HTML 5
(8) PhpMyAdmin

5.1.3 Batasan Implementasi

Batasan implementasi akan menjelaskan tentang batas-batas yang digunakan dalam pembuatan sistem beserta batasan hal yang mampu dilakukan oleh sistem. Berikut merupakan batasan implementasi dari perangkat lunak klasifikasi notasi SMILES dengan menggunakan algoritma C4.5:

(1) Bahasa pemrograman yang digunakan adalah PHP untuk proses dan operasi dan HTML5 untuk tampilan.
(2) Program dijalankan dengan menggunakan server *localhost* apache dari XAMPP Control Panel v3.2.2
(3) Data notasi SMILES disimpan pada database mysql dari server *localhost* phpMyAdmin.
(4) Pada proses preprocessing data menggunakan bantuan dari fungsi Regular Expression yang ada di fitur PHP.
(5) Pada proses pelatihan, masukan yang diterima oleh sistem adalah berupa kumpulan data latih beserta kelas klasifikasinya.
(6) Pada proses pengujian, masukan yang diterima oleh sistem adalah berupa kumpulan data uji.
(7) Pada proses pelatihan, luaran yang dihasilkan oleh sistem adalah berupa rule pohon keputusan untuk proses pengujian data.
(8) Pada proses pengujian, luaran yang dihasilkan oleh sistem adalah berupa hasil klasifikasi dari data uji dan akurasi dari proses pengujian dari kumpulan data uji.

5.2 Implementasi Algoritma

Terdapat 3 proses utama dalam implementasi dari perangkat lunak klasifikasi notasi SMILES dengan menggunakan algoritma C4.5 yaitu proses *preprocessing* data, proses pelatihan data, dan proses pengujian data.

5.2.1 Implementasi Proses Pelatihan Data

Proses pelatihan data merupakan proses dimana sistem akan dilatih berdasarkan data latih yang membutuhkan data nilai atribut yang telah di *Preprocessing*. Atribut tersebut merupakan atribut B, C, N, O, P, S, F, Cl, Br, I, dan OH yang telah dibagi dengan panjang notasi SMILES dan dikalikan dengan konstanta yang sama agar menjadi bilangan bulat. Dari atribut tersebut maka akan dicari nilai gain tertinggi yang nanti akan digunakan untuk membuat aturan pada desain pohon keputusan. Pohon keputusan dengan update terakhir akan digunakan untuk proses pengujian sistem.

5.2.1.1 Implementasi Proses Pengolahan Data Input Fitur

Setelah data berhasil di Preprocessing dan dipecah menjadi array dengan bantuan dari fungsi Regular Expression yang ada pada PHP, maka langkah selanjutnya adalah melakukan pengolahan terhadap input fitur agar siap untukdi proses pada proses pelatihan data, langkah pertama yang dilakukan adalah menghitung jumlah fitur atribut dari notasi SMILES tersebut untuk dijadikan inputan sebagai data latih.

Source Code yang digunakan untuk menghitung masing-masing atribut pada notasi SMILES ditunjukan pada source code 5.1.

```
1  $data['jumlah'] = count($preprop);
2      for ($i = 0; $i < $data['jumlah']; $i++) {
3          $huruf[$i] = preg_replace('/[1-9]/', '', $preprop[$i]);
4      }
5      $data['C'] = 0;
6      $data['Cl'] = 0;
7      $data['B'] = 0;
8      $data['Br'] = 0;
9      $data['O'] = 0;
10     $data['OH'] = 0;
11     $data['N'] = 0;
12     $data['P'] = 0;
13     $data['S'] = 0;
14     $data['F'] = 0;
15     $data['I'] = 0;
16     for ($a = 0; $a < count($huruf); $a++) {
17         if ($huruf[$a] == 'C') {
18             $data['C'] ++;
19         }if ($huruf[$a] == 'Cl') {
20             $data['Cl'] ++;
21         }if ($huruf[$a] == 'B') {
22             $data['B'] ++;
23         }if ($huruf[$a] == 'Br') {
24             $data['Br'] ++;
25         }if ($huruf[$a] == 'O') {
26             $data['O'] ++;
27         }if ($huruf[$a] == 'OH') {
28             $data['OH'] ++;
29         }if ($huruf[$a] == 'N') {
```

```
30              $data['N'] ++;
31          }if ($huruf[$a] == 'S') {
32              $data['S'] ++;
33          }if ($huruf[$a] == 'P') {
34              $data['P'] ++;
35          }if ($huruf[$a] == 'F') {
36              $data['F'] ++;
37          }if ($huruf[$a] == 'I') {
38              $data['I'] ++;
39          }
40      }
41      $data['Cbagi'] = $data['C'] / $data['jumlah'] * 100;
42      $data['Clbagi'] = $data['Cl'] / $data['jumlah'] * 100;
43      $data['Bbagi'] = $data['B'] / $data['jumlah'] * 100;
44      $data['Brbagi'] = $data['Br'] / $data['jumlah'] * 100;
45      $data['Obagi'] = $data['O'] / $data['jumlah'] * 100;
46      $data['OHbagi'] = $data['OH'] / $data['jumlah'] * 100;
47      $data['Nbagi'] = $data['N'] / $data['jumlah'] * 100;
48      $data['Pbagi'] = $data['P'] / $data['jumlah'] * 100;
49      $data['Sbagi'] = $data['S'] / $data['jumlah'] * 100;
50      $data['Fbagi'] = $data['F'] / $data['jumlah'] * 100;
51      $data['Ibagi'] = $data['I'] / $data['jumlah'] * 100;
```

Source Code 5.1. Menghitung masing-masing atribut Notasi SMILES

Penjelasan Source Code 5.1. Menghitung masing-masing atribut Notasi SMILES adalah sebagai berikut:

1. Pada baris 1-4 menjelaskan tentang menghitung panjang array dari notasi SMILES setelah dilakukan preprocessing. Setiap array pada notasi SMILES akan berisi nilai atribut untuk dihitung total atribut pada Notasi SMILES tersebut.
2. Pada baris 6-15 merupakan proses dari inisialisasi awal dari setiap fitur atau atribut dari notasi SMILES.
3. Pada baris 16-40 merupakan proses dalam menghitung bila nilai atribut pada array sama dengan nilai salah satu fitur yang digunakan pada proses pelatihan data, maka nilai fitur tersebut akan ditambah 1.
4. Pada baris 41-51 merupakan proses dalam membagi jumlah fitur dengan panjang dari notasi SMILES dan dikalikan dengan 100 agar nilainya tidak decimal.

Proses selanjutnya adalah melakukan proses diskritisasi yang bermanfaat untuk mengelompokan nilai dari masing-masing fitur yang telah dihitung

jumlahnya pada penjelasan implementasi sebelumnya. Source Code yang digunakan untuk implementasi pada proses diskritisasi ditunjukan pada Source Code 5.2.

```
1  $valid['valid'] = $this->model_PelatihanData->getAllData();
2        $valid['cek'] = TRUE;
3        $C = $this->model_PelatihanData->getCData();
4        $B = $this->model_PelatihanData->getBData();
5        $N = $this->model_PelatihanData->getNData();
6        $O = $this->model_PelatihanData->getOData();
7        $P = $this->model_PelatihanData->getPData();
8        $F = $this->model_PelatihanData->getFData();
9        $S = $this->model_PelatihanData->getSData();
10       $Cl = $this->model_PelatihanData->getClData();
11       $Br = $this->model_PelatihanData->getBrData();
12       $I = $this->model_PelatihanData->getIData();
13       $OH = $this->model_PelatihanData->getOHData();
14       $maxC = (max($C));
15       $minC = (min($C));
16       $valid['wC'] = ($maxC['C'] - $minC['C']) / 3;
17       $valid['v1C'] = $minC['C'] + 1 * $valid['wC'];
18       $valid['v2C'] = $minC['C'] + 2 * $valid['wC'];
19       $save['v1C'] = $minC['C'] + 1 * $valid['wC'];
20       $save['v2C'] = $minC['C'] + 2 * $valid['wC'];
21       $maxB = (max($B));
22       $minB = (min($B));
23       $valid['wB'] = ($maxB['B'] - $minB['B']) / 3;
24       $valid['v1B'] = $minB['B'] + 1 * $valid['wB'];
25       $valid['v2B'] = $minB['B'] + 2 * $valid['wB'];
26       $save['v1B'] = $minB['B'] + 1 * $valid['wB'];
27       $save['v2B'] = $minB['B'] + 2 * $valid['wB'];
28       for ($a = 0; $a < count($valid['valid']); $a++) {
29          if ($valid['valid'][$a]['C'] < $valid['v1C']) {
30             $valid['valid'][$a]['C'] = 1;
31          } elseif ($valid['valid'][$a]['C'] >= $valid['v1C'] &&
32 $valid['valid'][$a]['C'] <= $valid['v2C']) {
33             $valid['valid'][$a]['C'] = 2;
34          } elseif ($valid['valid'][$a]['C'] > $valid['v2C']) {
35             $valid['valid'][$a]['C'] = 3;
```

```
36          }
37          if ($valid['valid'][$a]['B'] < $valid['v1B']) {
38             $valid['valid'][$a]['B'] = 1;
39          } elseif ($valid['valid'][$a]['B'] >= $valid['v1B'] &&
40 $valid['valid'][$a]['B'] <= $valid['v2B']) {
41             $valid['valid'][$a]['B'] = 2;
42          } elseif ($valid['valid'][$a]['B'] > $valid['v2B']) {
43             $valid['valid'][$a]['B'] = 3;
44          }
```

Source Code 5.2. Proses Diskritisasi Setiap Fitur Notasi SMILES

Penjelasan Source Code 5.2. Proses Diskritisasi Setiap Fitur Notasi SMILES adalah sebagai berikut:

1. Pada baris 1 merupakan proses dari pengambilan data senyawa yang telah disimpan di database untuk dilakukan proses diskritisasi.
2. Pada baris 3-13 merupakan proses dari pengambilan nilai fitur atribut secara spesifik dari seluruh data yang ada didatabe.
3. Pada baris 14-15 mengambil nilai maksimal dan minimal nilai fitur dari suatu fitur atribut yang ditampung pada dataset atribut.
4. Pada baris 16 merupakan operasi untuk menghitung nilai w dari suatu fitur yang akan digunakan untuk menentukan batas diskritisasi.
5. Pada baris 17-18 merupakan operasi untuk menghitung batas 1 dan batas 2 dari suatu fitur atribut.
6. Pada baris 29-44 merupakan operasi untuk melakukan percabangan nilai diskrit bila nilai kurang dari batas 1 maka akan masuk kelompok diskrit bernilai 1, sedangkan bila nilai fitur berada diantara nilai batas 1 dan batas 2 maka akan masuk kelompok diskrit bernilai 2, sedangkan bila nilai lebih dari batas 2 maka akan masuk kelompok diskrit bernilai 3.

5.2.1.2 Implementasi Proses Pelatihan Data

Setelah data berhasil di hitung dan masing-masing fitur dan siap digunakan untuk proses pelatihan data, maka langkah selanjutnya adalah melakukan pelatihan data dari kumpulan set datalatih yang sudah disimpan pada database, langkah pertama yang dilakukan adalah menghitung nilai entropi dan gain pada tingkat akar untuk inisialisasi pohon keputusan. Sebelum menghitung nilai gain dam entropi maka dibutuhkan untuk menghitung entropi total. Source Code yang digunakan untuk menghitung entropi total pada tahapan inisialisasi akar ditunjukan pada Source Code 5.3.

```
1   $sTot = count($this->model_PelatihanData->getAllData());
2       $entropiTot = 0;
3       $entropiTemp = 0;
4       $atr = array();
5       $atrTot = array();
6       $entropiS = 0;
7       $entropiSub = array();
8       for ($i = 1; $i < 8; $i++) {
9           $kelas[$i] = count($this->model_PelatihanData-
10  >hitungEntropiTotal($i));
11          $entropi = (-($kelas[$i] / $sTot)) * (log($kelas[$i] / $sTot, 2));
12          $entropiTot = $entropiTot + $entropi;
13      }
14      $atribut = array(0 => 'C', 1 => 'B', 2 => 'N', 3 => 'O', 4 => 'P', 5 => 'S',
15  6 => 'F', 7 => 'Cl', 8 => 'Br', 9 => 'I', 10 => 'OH');
16      $gain0 = array();
17      $hasil = array();
```

Source Code 5.3. Proses Menghitung Entropi Total Pada Tahapan Inisialisasi Akar

Penjelasan Source Code 5.3. Proses Menghitung Entropi Total Pada Tahapan Inisialisasi Akar adalah sebagai berikut:

1. Pada baris 1 merupakan proses dari pengambilan data senyawa yang telah di disktitisasi yang telah disimpan di database untuk dilakukan proses Pelatihan Data.
2. Pada baris 2-7 merupakan proses inisialisasi data awal.
3. Pada baris 8-13 merupakan cara menghitung entropi total dengan rumus – jumlah data di kelas i/ data total dikalikan dengan log basis 2 dari –jumlah data di kelas i/ data total. Kemudian totalnya akan akan dijumlahkan untuk mendapatkan nilai entropi total.
4. Pada baris 14-15 merupakan inisialisasi array atribut fitur yang ada di database.
5. Pada baris 16-17 merupakan proses inisialisasi data awal.

Proses selanjutnya adalah melakukan proses perhitungan entropi dan gain masing-masing atribut yang bermanfaat untuk menentukan atribut yang terpilih untuk proses pemilihan atribut yang akan dijadikan noda pada tingkat akar. Source Code yang digunakan untuk implementasi pada proses perhitungan entropi dan gain ditunjukan pada Source Code 5.4.

```
1  for ($i = 0; $i < count($atribut); $i++) {
2          $gainS = 0;
3          for ($j = 1; $j < 4; $j++) {
4             $atrTot["$atribut[$i]"][$j] = count($this-
5  >model_PelatihanData->sAtrNoda0("$atribut[$i]", $j));
6             if ($atrTot["$atribut[$i]"][$j] < 1) {
7                $atrTot["$atribut[$i]"][$j] = 0.0001;
8             }$entropiS = 0;
9             for ($k = 1; $k < 8; $k++) {
10               $atr["$atribut[$i]"]["$j"][$k] = count($this-
11 >model_PelatihanData->atrNoda0("$atribut[$i]", $j, $k));
12               if ($atr["$atribut[$i]"]["$j"][$k] != 0 &&
13 $atr["$atribut[$i]"]["$j"][$k] == $atrTot["$atribut[$i]"][$j]) {
14                  $hasil["$atribut[$i]"]["$j"] = $k;
15               }
16               if ($atr["$atribut[$i]"]["$j"][$k] < 1) {
17                  $atr["$atribut[$i]"]["$j"][$k] = 0.0001;
18               }
19               $entropiTemp = -($atr["$atribut[$i]"]["$j"][$k] /
20 $atrTot["$atribut[$i]"][$j]) * log(($atr["$atribut[$i]"]["$j"][$k] /
21 $atrTot["$atribut[$i]"][$j]), 2);
22               if (!is_finite($entropiTemp)) {
23                  $entropiTemp = 0;
24               }
25               $entropiS = $entropiS + $entropiTemp;
26            }$entropiSub["$atribut[$i]"][$j] = $entropiS;
27            $gainTemp = ($atrTot["$atribut[$i]"][$j] / $sTot *
28 $entropiSub["$atribut[$i]"][$j]);
29            $gainS = $gainS + $gainTemp;
30         }$gain0["$atribut[$i]"] = $entropiTot - $gainS;
31      }
```

Source Code 5.4. Proses Menghitung Entropi dan Gain Setiap Atribut

Penjelasan Source Code 5.4. Proses Menghitung Entropi dan Gain Setiap Atribut Pada Tahapan Inisialisasi Akar adalah sebagai berikut:

1. Pada baris 1 merupakan perulangan yang mengulang setiap atribut fitur yang diuji oleh sistem.
2. Pada baris 2-3 merupakan proses inisialisasi data awal dan perulangan berdasarkan kelompok diskrit 1 hingga kelompok diskrit 3.

3. Pada baris 4-5 merupakan cara pengambilan nilai yang ada pada database pada atribut dengan indeks array i dan kelompok diskrit bernilai j.
4. Pada baris 6-8 merupakan perancangan bila nilai dari atrTot atribut dengan indeks array i dan kelompok diskrit bernilai j bernilai 0 maka akan diubah menjadu 0,0001 agar tidak terjadi error karena melakukan operasi aritmatika dengan menggunakan nilai 0.
5. Pada baris 9 merupakan proses perulangan sebanyak kelas klasifikasi yang akan dideklarasikan.
6. Pada baris 10-11 merupakan proses mengambil dari database indeks array i dan kelompok diskrit bernilai j dengan klasifikasi fungsi bernilai k.
7. Pada baris 12-15 merupakan proses percabangan bila atribut dengan atribut total indeks array i dan kelompok diskrit bernilai j sama dengan atribut salah satu kelas yang berindeks array i dan kelompok diskrit bernilai j dengan klasifikasi fungsi bernilai k, maka atribut itu hasilnya adalah kelas klasifikasi k.
8. Pada baris 16-18 merupakan perancangan bila nilai dari atr atribut dengan indeks array i dan kelompok diskrit bernilai j dengan klasifikasi fungsi bernilai k bernilai 0 maka akan diubah menjadi 0,0001 agar tidak terjadi error karena melakukan operasi aritmatika dengan menggunakan nilai 0.
9. Pada baris 19-21 merupakan proses perhitungan entropi setiap atribut dengan rumus -(jumlah atribut dengan indeks array i dan kelompok diskrit bernilai j/ dengan indeks array i dan kelompok diskrit bernilai j dengan klasifikasi fungsi bernilai k) x log2 (jumlah atribut dengan indeks array i dan kelompok diskrit bernilai j/ dengan indeks array i dan kelompok diskrit bernilai j dengan klasifikasi fungsi bernilai k).
10. Pada baris 25-26 merupakan proses perhitungan entropi dari atribut indeks i dan kelompok diskrit j dengan menjumlahkan setiap sub entropi setiap kelas klasifikasi k.
11. Pada baris 27-31 merupakan proses perhitungan gain dari suatu atribut. Dengan subgain atrTotal/ total data atribut * dengan entropi masing masing kelas diskrit.

Proses selanjutnya adalah melakukan proses pencarian nilai gain terbesar dan pencarian indeks akar. Source Code yang digunakan untuk implementasi pada proses perhitungan entropi dan gain ditunjukan pada Source Code 5.5.

```
1  $gainTerbesar = 0;
2      $indeksGain = NULL;
3      for ($i = 0; $i < count($gain0); $i++) {
4          if ($gain0[$atribut[$i]] > $gainTerbesar) {
```

```
5             $gainTerbesar = $gain0[$atribut[$i]];
6             $indeksGain = $atribut[$i];
7          }
8       }
9       $hasilfix = array();
10      for ($j = 1; $j < 4; $j++) {
11         if (isset($hasil["$indeksGain"]["$j"])) {
12            $hasilfix[$j] = $hasil["$indeksGain"]["$j"];
13         } else {
14            $hasilfix[$j] = NULL;
15         }
16      }
17      $save['id'] = rand(0, 10000);
18      $save['noda'] = 0;
19      $save['atribut'] = $indeksGain;
20      $save['entropi1'] = $entropiSub["$indeksGain"][1];
21      $save['entropi2'] = $entropiSub["$indeksGain"][2];
22      $save['entropi3'] = $entropiSub["$indeksGain"][3];
23      $save['s1'] = $atrTot["$indeksGain"][1];
24      $save['s2'] = $atrTot["$indeksGain"][2];
25      $save['s3'] = $atrTot["$indeksGain"][3];
26      $save['s1Done'] = FALSE;
27      $save['s2Done'] = FALSE;
28      $save['s3Done'] = FALSE;
29      $save['s1Atr'] = NULL;
30      $save['s2Atr'] = NULL;
31      $save['s3Atr'] = NULL;
32      $save['refrensi'] = NULL;
33      $save['kelompok'] = NULL;
34      $save['fullref'] = NULL;
35      $save['hasil1'] = $hasilfix[1];
36      $save['hasil2'] = $hasilfix[2];
37      $save['hasil3'] = $hasilfix[3];
38      $save['gain'] = $gain0["$indeksGain"];
39      $save['atrDel'] = NULL;
40      for ($i = 0; $i < count($gain0); $i++) {
41         if ($gain0[$atribut[$i]] == 0) {
42            $save['atrDel'] = $save['atrDel'] . $atribut[$i];
43         }
```

```
44          }
45          $this->model_PohonKeputusan->deleteNode0();
46          $this->model_PohonKeputusan->insertTree($save);
```

Source Code 5.5. Proses Pencarian Nilai Gain Terbesar dan Pencarian Indeks Akar

Penjelasan Source Code 5.5. Proses Menghitung Entropi dan Gain Setiap Atribut Pada Tahapan Inisialisasi Akar adalah sebagai berikut:

1. Pada baris 1-2 merupakan proses inisialisasi nilai atribut untuk menampung nilai gain terbesar beserta indeks atribut dari gain terbesar.
2. Pada baris 3-8 merupakan proses pencarian nilai gain terbesar beserta indeks atribut dari gain terbesar.
3. Pada baris 10-16 merupakan proses dimana terdapat isi dari $hasil["$indeksGain"]["$j"] yang menandakan bahwa sudah ada hasil klasifikasi dari entropi akar sehingga tidak diperlukan mencari noda dibawahnya kembali.
4. Pada baris 17-39 merupakan penyimpanan data pada database yang merupakan data penting untuk digunakan pada proses noda selanjutnya ataupun pada proses pemodelan pohon keputusan.
5. Pada baris 40-44 merupakan merupakan penampungan nilai yang berisi atribut yang akan dihapus spada noda selanjutnya.
6. Pada baris 45-46 merupakan proses penyimpanan pada database.

5.2.1.3 Implementasi Proses Pengujian Data

Setelah rule pada pohon keputusan telah terbentuk pada proses peatihan data, maka langkah selanjutnya adalah melakukan pengujian data dari dari rule yang telah tersimpan, dengan memasukan data masukan notasi SMILES maka dengan mengacu pada rule yang telah tersimpan pada database maka sistem akan mampu menentukan kelas klasifikasi dari notasi SMILES yang diinputkan pada sistem. Source Code yang digunakan untuk pengujian data ditunjukan pada Source Code 5.6.

```
1  for ($noda = 0; $noda < 11; $noda++) {
2          $dataPohon[$noda]          =          $this->model_PohonKeputusan-
3  >getTree1($noda, $fullref);
4          if ($dataPohon[$noda] == NULL) {
5             $data['hasil'] = 0;
6             break;
7          }
8          $tabelPohon = $dataPohon[$noda][0];
9          $atribut = $tabelPohon['atribut'];
```

```
10          if ($data[$atribut . 'dis'] == 1) {
11              if (isset($tabelPohon['hasil1']) && $tabelPohon['hasil1'] != 100)
12  {
13                  $data['hasil'] = $tabelPohon['hasil1'];
14                  break;
15              } elseif ($tabelPohon['hasil1'] == 100) {
16                  $data['hasil'] = 0;
17                  break;
18              } else {
19                  $fullref = $fullref . $atribut . $data[$atribut . 'dis'];
20                  continue;
21              }
22          } elseif ($data[$atribut . 'dis'] == 2) {
23              if (isset($tabelPohon['hasil2']) && $tabelPohon['hasil2'] != 100)
24  {
25                  $data['hasil'] = $tabelPohon['hasil2'];
26                  break;
27              } elseif ($tabelPohon['hasil1'] == 100) {
28                  $data['hasi2'] = 0;
29                  break;
30              } else {
31                  $fullref = $fullref . $atribut . $data[$atribut . 'dis'];
32                  continue;
33              }
34          } elseif ($data[$atribut . 'dis'] == 3) {
35              if (isset($tabelPohon['hasil3']) && $tabelPohon['hasil3'] != 100)
36  {
37                  $data['hasi3'] = $tabelPohon['hasil3'];
38                  break;
39              } elseif ($tabelPohon['hasil1'] == 100) {
40                  $data['hasil'] = 0;
41                  break;
42              } else {
43                  $fullref = $fullref . $atribut . $data[$atribut . 'dis'];
44                  continue;
45              }
46          }
47      }
        $data['prepropYES'] = 1;
```

```
        $this->load->view('navigasiProgram');
        $this->load->view('inti_Program', $data);
```

Source Code 5.6. Proses Pengujian Data

Penjelasan Source Code 5.5. Proses Menghitung Entropi dan Gain Setiap Atribut Pada Tahapan Inisialisasi Akar adalah sebagai berikut:

1. Pada baris 1 merupakan perulangan hingga noda 10 karena terdapat 11 atribut sehingga noda yang ada memungkinkan hingga tingat noda 10.
2. Pada baris 2-3 merupakan proses pengambilan data pohon pada noda sesuai perulangan noda dengan fullreff yang default NULL bila noda akar, untuk noda selanjutnya maka nilai fullref merupakan nilai atribut yang telah digunakan di akar ataupun di noda sebelumnya.
3. Pada baris 4-7 menunjukan bahwa bila tidak ada data pohon keputusan pada noda tersebut maka proses dihentikan.
4. Pada baris 11-13 proses bila ada nilai berapapun dari $tabelPohon['hasil1'] dan nilai tersebut bukan 100 yang merepresentasikan bahwa pohon keputusan sudah berada di entropi 0 maka sudah ditemukan kelas klasifikasi dari notasi SMILE yang diinputkan.
5. Pada baris 14-16 proses bila ada nilai dari $tabelPohon['hasil1'] bernilai 100 yang merepresentasikan bahwa pohon keputusan sudah berada di entropi 0 sehingga data SMILES tersebut tidak dapat diklasifikasikan.
6. Pada baris 17-20 merupakan proses bila nilai dari $tabelPohon['hasil1'] bernilai NULL maka akan mencari di noda selanjutnya dengan mengambil atribut dan kelas diskrit saat ini sebagai variabel $fullref
7. Pada baris 21-44 memiliki penjelasan yang sama dengan baris 11-20 namun untuk baris 21-44 merupakan bila kelas diskrit nya masuk pada kelas diskrit 2 dan 3.
8. Pada baris 45-47 merupakan proses menampilkan tampilan antarmuka ke pengguna hasil dari klasifikasi yang diklasifikasi oleh sistem.

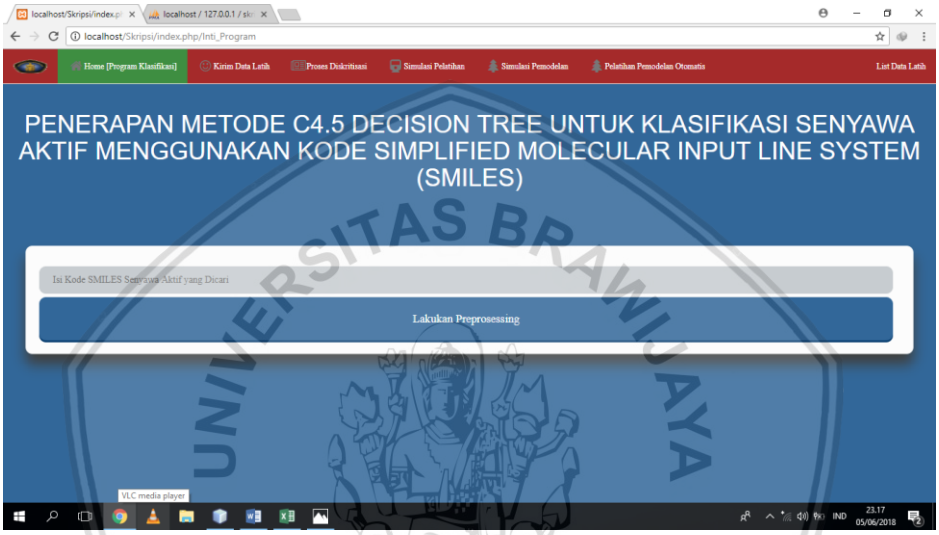
5.3 Implementasi Antarmuka

Terdapat beberapa antarmuka pada sistem klasifikasi notasi SMILES dengan menggunakan algoritma C4.5 yang berfungsi untuk menampung perintah dan inputan dari pengguna dalam melakukan pelatihan data ataupun pengujian data. Pada proses pelatihan data maka antarmuka akan menampilkan inputan untuk mengirim data latih ke database, menampilkan data setelah diprepossessing dan didiskritisasi, menampilkan perhitungan entropi dan gain juga menampilkan model dari rule pohon keputusan yang telah dibuat.

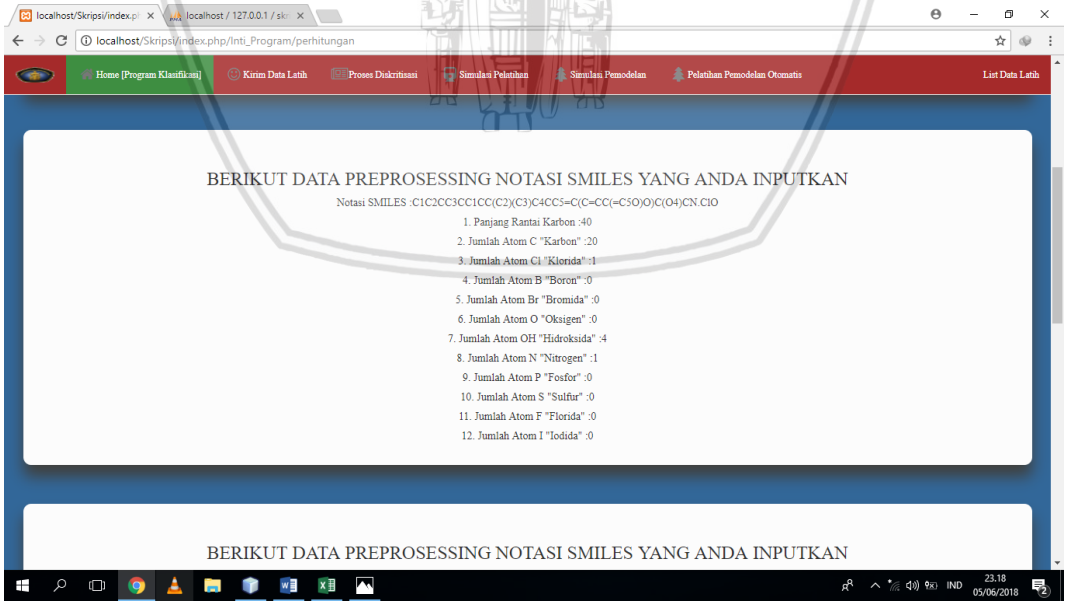
Sedangkan pada saat pengujian maka antarmuka akan menampilkan masukan untuk data uji dan hasil dari klasifikasi dari proses pengujian.

5.3.1 Antarmuka Halaman Beranda

Pada antarmuka halaman beranda akan ditampilkan inti dari fungsi sistem klasifikasi notasi SMILES dengan menggunakan algoritma C4.5 yaitu proses pengujian, pada halaman antarmuka beranda maka akan ditampilkan kolom masukan untuk data uji dan nanti akan diproses dan ditunjukan nilai preprosessing, diskritisasi dan kelas klasifikasinya.



Gambar 5.1 Masukan untuk Data Uji

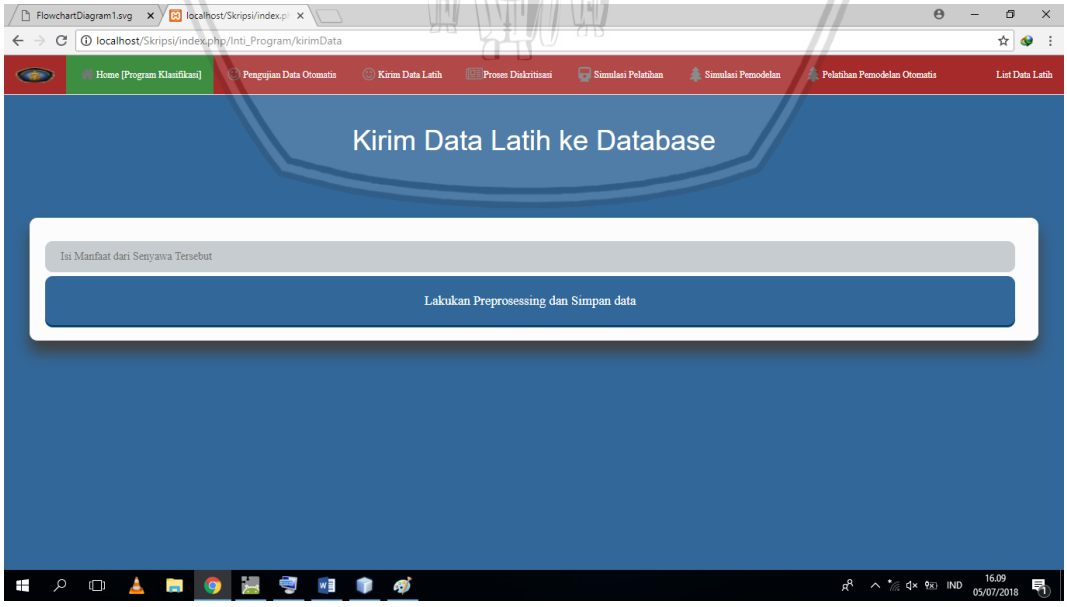


Gambar 5.2 Hasil Preprosessing dari Data Uji

Gambar 5.3 Hasil Diskritisasi dan Kelas Klasifikasi

5.3.2 Antarmuka Kirim Data Latih

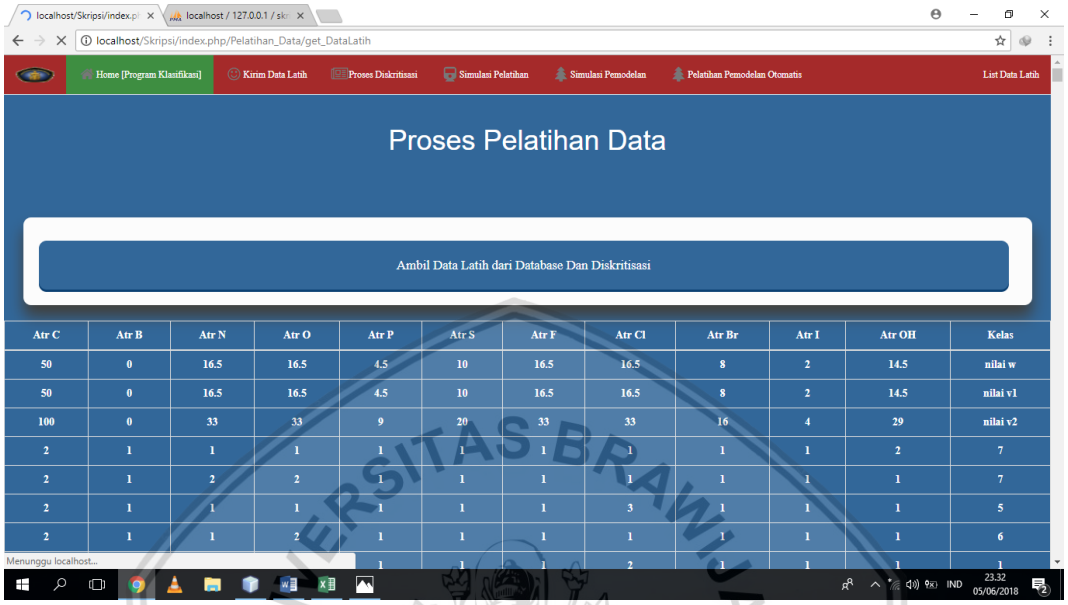
Pada antarmuka halaman kirim data latih maka akan ditampilkan kolom masukan untuk kelas klasifikasi data sedangkan dataset dari kelas klasifikasi tersebut disimpan pada sebuah notepad sehingga akan langsung dapat mengirim banyak dataset ke database akan disimpan pada database dan akan digunakan untuk proses pelatihan data pada sitem klasifikasi notasi SMILES dengan menggunakan algoritma C4.5.



Gambar 5.4 Antarmuka Kirim Data Latih

5.3.4 Antarmuka Diskritisasi Data

Pada antarmuka halaman Diskritisasi Data maka akan ditampilkan data-data yang ada di database setelah dilakukan proses diskritisasi, data diskritisasi adalah data final yang siap untuk digunakan pada proses pelatihan data.

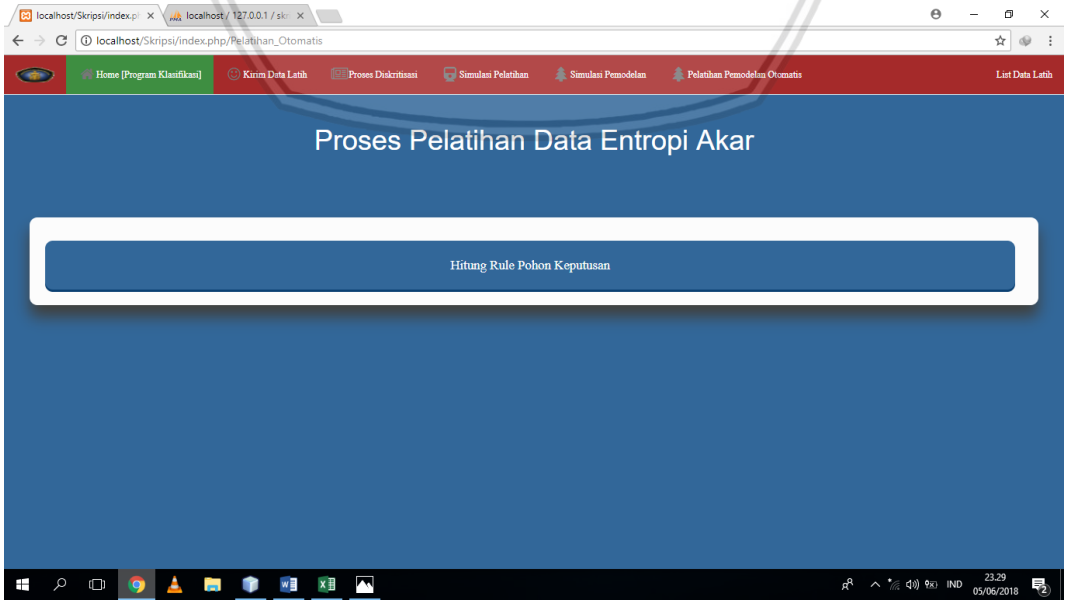


Atr C	Atr B	Atr N	Atr O	Atr P	Atr S	Atr F	Atr Cl	Atr Br	Atr I	Atr OH	Kelas
50	0	16.5	16.5	4.5	10	16.5	16.5	8	2	14.5	nilai w
50	0	16.5	16.5	4.5	10	16.5	16.5	8	2	14.5	nilai v1
100	0	33	33	9	20	33	33	16	4	29	nilai v2
2	1	1	1	1	1	1	1	1	1	2	7
2	1	2	2	1	1	1	1	1	1	1	7
2	1	1	1	1	1	1	3	1	1	1	5
2	1	1	2	1	1	1	1	1	1	1	6
2	1	1	2	1	1	1	2	1	1	1	1

Gambar 5.5 Antarmuka Diskritisasi Data Latih

5.3.5 Antarmuka Pelatihan Data Otomatis

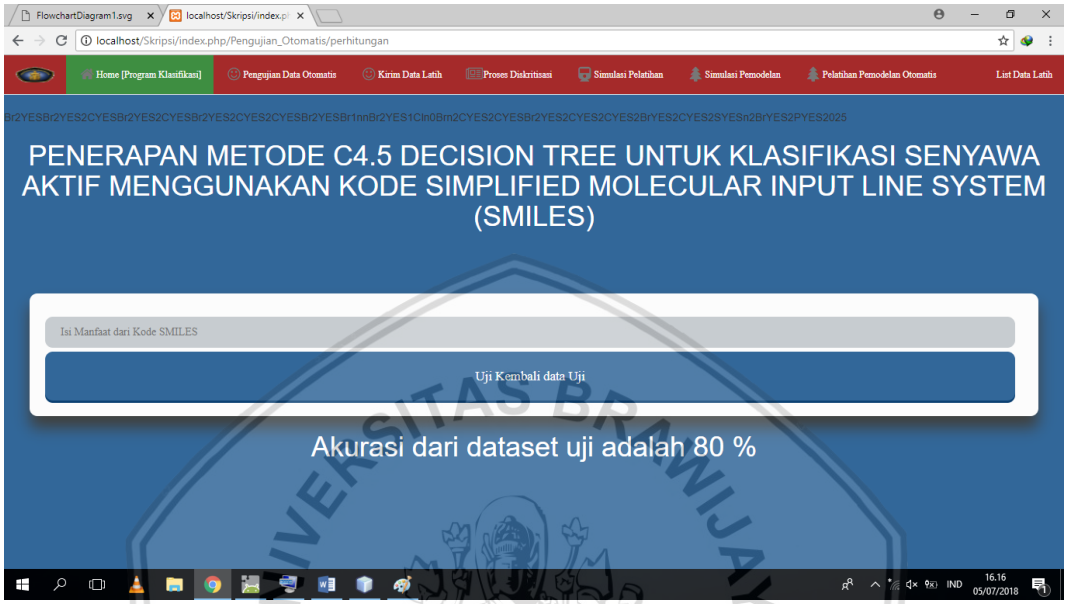
Pada antarmuka halaman Pelatihan Data Otomatis maka pengguna hanya perlu untuk menekan tombol hitung rule pohon keputusan maka akan langsung dihitung proses pembentukan rule dari tingkat akar hingga noda terakhir dari pohon keputusan.



Gambar 5.6 Antarmuka Pelatihan Data

5.3.6 Antarmuka Pengujian Data Otomatis

Pada antarmuka halaman Pengujian Data Otomatis maka pengguna hanya perlu untuk mengisikan kelas dataset yang sekarang sedang diuji sedangkan data pengujian tersimpan pada notepad sehingga akan mampu untuk menguji banyak data secara otomatis.



Gambar 5.7 Antarmuka Pengujian Data Otomatis

BAB 6 PENGUJIAN DAN HASIL PENGUJIAN

Bab ini akan berisi tentang skenario dari pengujian yang akan dilakukan beserta dengan hasil dari analisis yang dilakukan yang dilakukan. Bab ini terdiri dari sub-sub skenario pengujian, hasil dari pengujian, beserta analisa dari hasil pengujian yang dilakukan.

6.1 Skenario Pengujian

Pada sub bab ini akan dibahas skenario pengujian yang akan diuji, terdapat 4 jenis skenario pengujian yang akan dilakukan pada aplikasi untuk mengklasifikasi fungsi notasi SMILES, diantaranya adalah pengujian efektifitas akurasi dari teknik diskritisasi dengan metode *Bining* dan dengan menggunakan metode *Entropi Based*, pengujian efektifitas akurasi dari pembagian panjang notasi SMILES dibandingkan dengan tanpa pembagian panjang notasi SMILES pada setiap fitur atribut yang diuji, efektifitas hasil akurasi dari banyak data latih, pengujian *cross validation* untuk menguji validitas akurasi bila data latih dan data uji diacak secara acak, dan standar deviasi untuk menunjukan tingkat kestabilan dari tingkat akurasi sistem.

6.1.1 Skenario Pengujian Teknik Diskritisasi dengan Metode *Bining* dan dengan Metode *Entropi Based* terhadap Tingkat Akurasi

Pada skenario ini akan diciptakan keadaan pengujian dengan variabel bebas berupa penggunaan teknik diskritisasi yang berbeda diantaranya adalah penggunaan teknik diskritisasi dengan menggunakan teknik diskritisasi metode *Bining* dan teknik diskritisasi dengan metode *Entropi Based.* Sedangkan keadaan yang disamakan atau variabel kontrol yang diterapkan adalah penggunaan data latih yang sama beserta komposisi yang sama yaitu dengan menerapkan data latih sebesar 80% data dan 20% data diterapkan sebagai data uji, data yang digunakan sebanyak 2 kelas diantaranya kelas kanker dan kelas metabolisme. Sedangkan hasil yang akan diamati adalah tingkat akurasi yang dihasilkan dari kedua teknik diskritisasi tersebut.

6.1.2 Skenario Pengujian Pembagian Panjang Notasi SMILES terhadap Tingkat Akurasi

Pada skenario ini akan diciptakan keadaan pengujian dengan variabel bebas berupa fitur atribut dibagi dengan panjang notasi SMILES sedangkan keadaan lainya tanpa pembagian panjang notasi SMILES. Variabel kontrol atau keadaan yang disamakan berupa penggunaan data latih yang sama beserta

komposisi yang sama yaitu dengan menerapkan data latih sebesar 80% data dan 20% data diterapkan sebagai data uji, data yang digunakan sebanyak 2 kelas diantaranya kelas kanker dan kelas metabolisme. Sedangkan hasil yang akan diamati adalah tingkat akurasi yang dihasilkan dari kedua teknik diskritisasi tersebut.

6.1.3 Skenario Pengujian Banyak Data Latih terhadap Tingkat Akurasi

Pada skenario ini maka akan diciptakan keadaan pengujian dengan variabel bebas banyak data latih yang digunakan, data latih yang digunakan diantaranya mulai dari 10% hingga 80% dari set data dengan kelipatan sebesar 10% data latih. Sedangkan variabel kontrol dalam pengujian ini adalah data uji yang dibuat sama dengan komposisi dan jumlah yang sama, dan juga teknik diskritisasi yang digunakan. Variabel hasil yang dicermati dalam pengujian ini adalah tingkat akurasi sistem dalam mengklasifikasi 20% dari data yang akan digunakan sebagai data uji.

6.1.4 Skenario Pengujian *Cross Validation*

Pada skenario ini akan diciptakan keadaan pengujian Untuk menguji tingkat validitas dari akurasi yang dihasilkan. Pengujian *Cross Validation* dilakukan dengan pembuatan set data menjadi 5 bagian masing masing bagian memiliki komposisi data sebesar 20% data, keadaan yang disamakan adalah semua pegujian memiliki komposisi data yang sama yaitu sebesar 80% sbsagai data latih dan 20% sebagai data uji. Data uji sebesar 20% akan menggunakan set data 1, 2, 3 hingga set data 5 dan bagian set data yang tidak digunakan sebagai data uji akan digunakan sebagai data latih.

6.2 Hasil Pengujian dan Analsis

Pada subab ini akan ditampilkan hasil akurasi dari kesemua jenis skenario yang diuji. Dari kesemua jenis skenario tersebut akan dilakukan analisis sehingga dapat ditarik kesimpulan dari kesesuaian metode C4.5 dalam mengklasifikasi notasi SMILES.

6.2.1 Hasil Pengujian dan Analisis Teknik Diskritisasi dengan Metode *Bining* dan dengan Metode *Entropi Based* terhadap Tingkat Akurasi

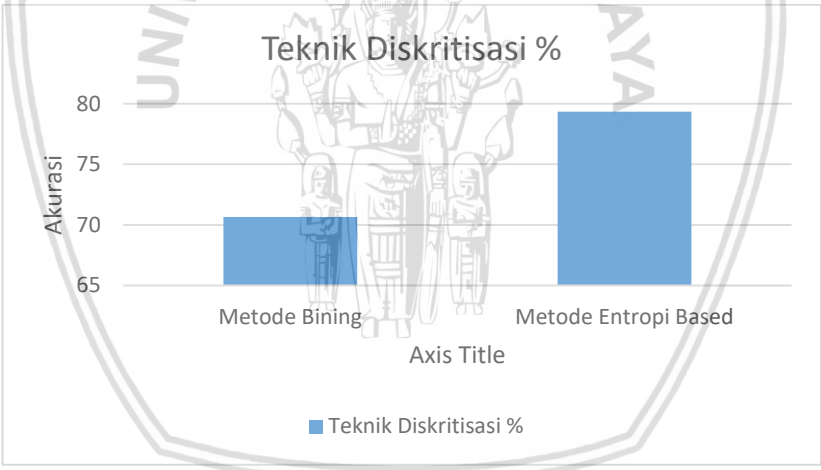
Teknik diskritisasi merupakan teknik mengkonversi bilangan kontinu menjadi bilangan diskrit dikarenakan C4.5 hanya mampu menggunakan data diskrit dalam proses pelatihan dan pengujiannya. Penggunaan metode diskritisasi yang tepat akan sangat bermanfaat dalam membantu mendapatkan akurasi

terbaik dalam mengkasifikasi notasi SMILES. Berikut merupakan grafik dari akurasi yang dihasilkan oleh teknis diskritisasi metode *Bining* dan metode *Entropi Based*. Hasil dari pengujian Teknik Diskritisasi dengan Metode *Bining* dan dengan Metode *Entropi Based* terhadap tingkat akurasi ditunjukan pada Tabel 6.1 berikut.

Tabel 6.1 Hasil dari pengujian Teknik Diskritisasi dengan Metode *Bining* dan dengan Metode *Entropi Based*

No	Jenis Diskritisasi	Akurasi	Waktu Eksekusi
1	Metode *Bining*	70,65%	15,23 detik
2	Metode *Entropy Based*	79,34%	19,55 detik

Dari Tabel 6.1 menunjukan data pengujian teknik diskritisasi dengan metode *bining* dan dengan metode *entropi based* terhadap tingkat akurasi dihasilkan bila dengan metode *bining* menghasilkan akurasi sebesar 70,65% sedangkan bila menggunakan teknik diskritisasi *entropi based* sebesar 79,34%. Bila diamati menggunakan diagram maka ditunjukan seperti pada Gambar 6.1 berikut.



Gambar 6.1 Grafik akurasi teknik diskritisasi metode *bining* dan metode *entropi based*

Teknik diskritisasi *Bining* merupakan teknik diskrit yang mengambil pola nilai tengah dari dataset kontinyu tanpa melihat adanya hubungan kelas data dengan klasifikasinya beserta persebaran data. Sedangkan teknik diskritisasi *Entropy Based* mencari nilai batas berdasarkan tingkat nilai Gain tertinggi yang merepresentasikan nilai pada batas tersebut benar benar memisahkan suatu kelompok data berdasarkan kelas klasifikasinya.

Dari data yang dipaparkan oleh Gambar 6.1 menunjukan bahwa teknik *Bining* yang bersifat mengambil nilai tengah tanpa melihat persebaran data

kepada suatu kelas klasifikasi memiliki klasifikasi yang lebih rendah dari pada teknik diskritisasi *entropi based* dengan memperhatikan persebaran data dan tingkat akurasi, namun dari sisi waktu eksekusi, teknis diskritisasi menggunakan *entropy based* membutuhkan waktu eksekusi yang lebih lama. Hal ini dikarenakan proses pencarian batas pada metode diskritisasi *entropy based* mencari menggunakan perulangan sebanyak *range* nilai minimal suatu fitur atribut hingga nilai maksimal, sedangkan bila menggunakan teknik diskritisasi *bining* hanya dilakukan 1 kali operas aritmatika untuk mendapatkan nilai batas yang merupakan nilai tengah dari fitur atribut.

Proses diskritisasi menjadi komponen yang sangat penting bagi tingkat akurasi, karena dengan proses diskritisasi maka telah menghilangkan nilai asli berupa nilai kontinyu dari suatu atribut fitur sehingga bila nilai diskrit yang merupakan nilai kelompok dari suatu range nilai atribut fitur, kurang merepresentasikan nilai sebenarnya dari atribut fitur, maka akan menyebabkan algoritma C4.5 kurang melakukan pelatihan data dengan baik karena pola sesungguhnya dari suatu notasi SMILES akan hilang. Sehingga ketika pengujian data dilakukan maka sistem akan kurang mengenali beberapa notasi SMILES yang diuji.

Dengan teknik diskritisasi yang tepat maka representasi kelompok diskrit akan mewakili secara tepat nilai sesungguhnya sehingga akurasi yang didapatkan akan semakin lebih besar.

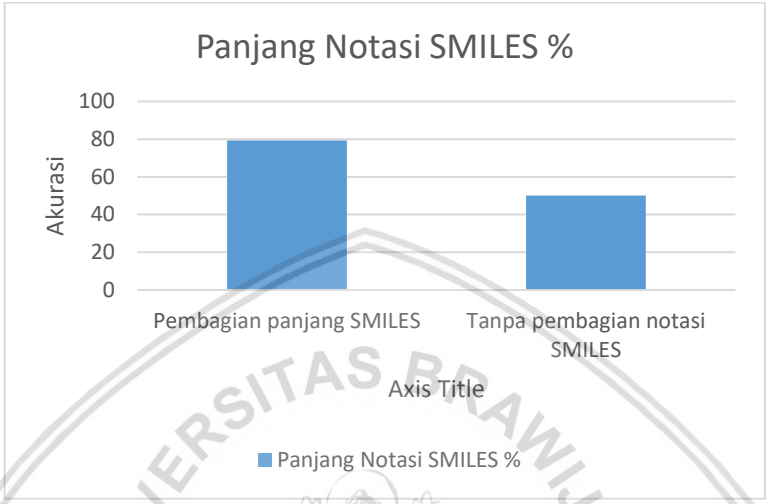
6.2.2 Analisis dan Hasil Pengujian Pembagian Panjang Notasi SMILES terhadap Tingkat Akurasi

Panjang notasi SMILES dapat digunakan sebagai salah satu fitur yang menentukan hasil dari klasifikasi SMILES diantaranya dengan cara membagi setiap fitur atribut dengan panjang dari notasi SMILES. Dikarenakan teknik diskritisasi yang terbaik merupakan teknik diskritisasi *entropi based* dengan skema data 80% data latih dan 20% data uji menghasilkan akurasi sebesar 79,34% maka pada pengujian ini menggunakan teknik diskritisasi *entropi based.* Berikut merupakan grafik dari akurasi yang dihasilkan oleh fitur atribut yang dibagikan dengan panjang SMILES dan tanpa dibagikan dengan panjang SMILES.

Tabel 6.2 Hasil dari pengujian pembagian panjang notasi SMILES

No	Jenis	Akurasi	Waktu Eksekusi
1	Dengan pembagian panjang notasi SMILES	79,34%	19,55 detik
2	Tanpa pembagian panjang notasi SMILES	50%	18,77 detik

Dari Tabel 6.2 menunjukan data pengujian pembagian panjang SMILES dan tanpa pembagian panjang SMILES terhadap tingkat akurasi dihasilkan bila dengan pembagian panjang SMILES menghasilkan akurasi sebesar 79,34% sedangkan bila tanpa pembagian panjang SMILES sebesar 50%. Bila diamati menggunakan diagram maka ditunjukan seperti pada Gambar 6.2 berikut.



Gambar 6.2 Grafik akurasi pembagian panjang SMILES dan tanpa pembagian panjang SMILES

Panjang notasi SMILES menjadi parameter yang harus diperhitungkan dalam proses klasifikasi notasi SMILES dengan algortima C4.5, Hal ini dikarenakan terdapat perbedaan akurasi yang mencolok diantara hasil pengujian tanpa pembagian panjang notasi SMILES dan dengan menerapkan pembagian panjang notasi SMILES. Hal ini menunjukan bahwa panjang notasi SMILES sebagai parameter untuk menormalisasi nilai dari fitut atribut SMILES. Hal ini dikarenakan ada banyak notasi SMILES yang tidak diperhitungkan sebagai atribut fitur sehingga bila menghilangkan data dari atribut panjang notasi SMILES maka data masing masing atribut fitur dari notasi SMILES akan kurang spesifik dikarenakan dengan jumlah atribut fitur yang sama, belum tentu notasi SMILES tersebut memiliki kedekatan nilai bila notasi SMILES yang satu memiliki panjang notasi SMILES yang rendah sedangkan notasi SMILES yang lain memiliki panjang notasi SMILES yang besar. Notasi SMILES dapat dikatakan memiliki kedekatan bila memang notasi SMILES tersebut memiliki atribut yang berdekatan dan juga memiliki panjang notasi SMILES yang sama.

Dengan membagi nilai fitur atribut maka hal tersebut akan membantu sistem dalam proses pelatihan data bahwa suatu notasi SMILES dinyatakan berdekatan atau tidak sehingga sistem akan lebih mampu dalam menentukan

klasifikasi dari notasi SMILES tersebut dan akan meningkatkan hasil akurasi dari sistem dalam mengklasifikasi notasi SMILES.

Sedangkan dari sisi waktu eksekusi tidak terlalu terdapat perbedaan antara dengan dan tanpa pembagian panjang dari notasi SMILES. Hal ini dikarenakan perbadaan dari keduanya hanyalah di operasi perkalian dan pembagian serta tidak adanya perbedaan eksekusi perulangan antara keduanya, sehingga hampir tidak terdapat perbedaan antara penggunaan dan tanpa penggunaan pembagian notasi SMILES pada fitur atribut.

6.2.3 Hasil Pengujian Banyak Data Latih terhadap Tingkat Akurasi

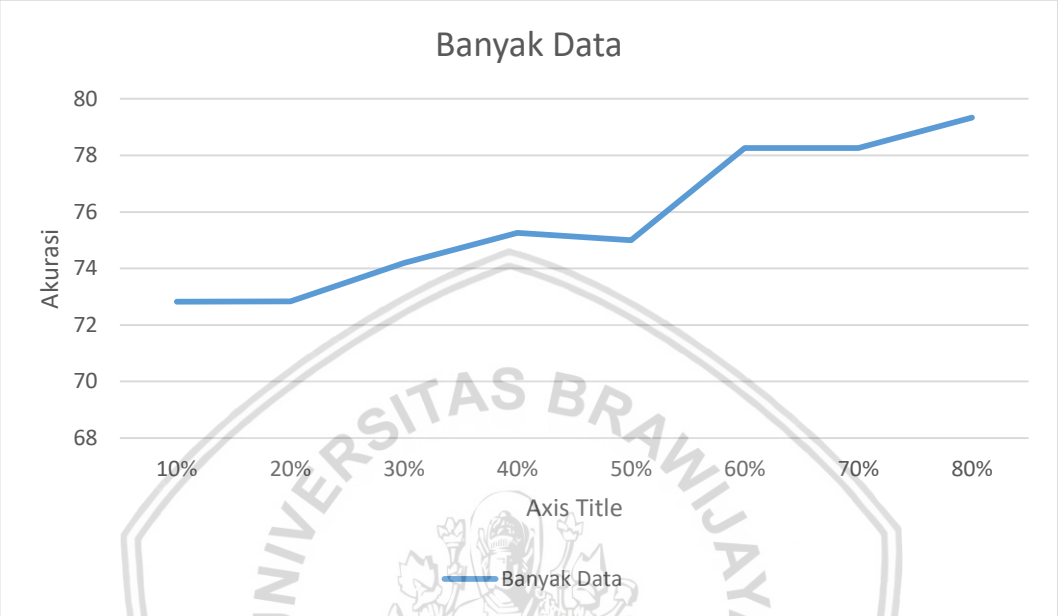
Banyak data latih yang digunakan untuk pelatihan data dapat menjadi salah satu parameter yang digunakan dalam tingkat akurasi yang dihasilkan oleh metode C4.5. Dikarenakan teknik diskritisasi yang terbaik merupakan teknik diskritisasi *entropi based* dengan skema data 80% data latih dan 20% data uji menghasilkan akurasi sebesar 79,34% maka pada pengujian ini menggunakan teknik diskritisasi *entropi based.* Dikarenakan pembagian fitur atribut dengan panjang notasi SMILES yang terbaik merupakan dengan pembagian panjang notasi SMILES dengan hasil akurasi sebesar 79,34% maka pada pengujian ini menggunakan kondisi yang sama yaitu dengan pembagian fitur atribut dengan panjang notasi SMILES. Berikut merupakan grafik dari akurasi yang dihasilkan oleh metode C4.5 berdasarkan jumlah data latih yang berbeda-beda.

Tabel 6.3 Hasil dari pengujian banyak data latih notasi SMILES

No	Banyak Data	Akurasi	Waktu Eksekusi
1	10%	72,82%	6,31 detik
2	20%	72,82%	8,62 detik
3	30%	74,19%	10,64 detik
4	40%	75,26%	12,45 detik
5	50%	75%	13,24 detik
6	60%	78,26%	14,56 detik
7	70%	78,26%	16,78 detik
8	80%	79,34%	19,55 detik

Dari Tabel 6.3 didapat hasil dari pengujian banyak data latih notasi SMILES menunjukan bahwa ketika data latih sebesar 10% dan 20% maka akurasi dari metode C4.5 sebesar 72,82%, sedangkan ketika banyak data latih sebesar 30% maka akurasi dari metode C4.5 sebesar 74,19%. Kemudian ketika banyak data latih sebesar 40% maka akurasi dari metode C4.5 sebesar 75,26%, ketika banyak

data latih sebesar 50% maka akurasi dari metode C4.5 sebesar 75%. Kemudian ketika banyak data latih sebesar 60% dan 70% maka akurasi dari metode C4.5 sebesar 78,26%. Dan yang terakhir bila data yang digunakan sebesar 80% maka akurasi yang dihasilkan sebesar 79,34%. Bila diamati menggunakan diagram maka ditunjukan seperti pada Gambar 6.3 berikut.



Gambar 6.3 Grafik akurasi banyak jumlah data

Dari data yang ditunjukan pada Gambar 6.3 menunjukan bahwa semakin tinggi prosesntase data latih maka terjadi peningkatan akurasi klasifikasi notasi SMILES. Hal ini menunjukan bahwa terdapat korelasi berbanding lurus antara banyak data latih yang digunakan dan hasil akurasi dari pengujian data.

Banyak data latih yang digunakan pada algoritma C4.5 memiliki peran yang sangat penting terhadap kemampuan sistem dalam melakukan klasifikasi notasi SMILES. Hal ini dikarenakan semakin banyak pola data yang dikenali oleh algoritma C4.5 sehingga akurasi yang dihasilkan semakin baik. Bila data latih yang digunakan lebih sedikit maka sistem kurang dalam proses mengenali berbagai macam pola yang mendekati pola dari data latih. Maka dari itu algoritma C4.5 membutuhkan data latih yang banyak sehingga proses pengenalan pola dari suatu data uji akan semakin baik dan sistem akan mendapatkan pengetahuan yang maksimal dalam proses pengujian data latih. Namun terdapat kekurangan bila terlalu banyak data latih, yaitu waktu eksekusi yang akan berjalan lebih lama. Sehingga diperlukan banyak data yang optimal dimana perubahan akurasi jaraknya sangat kecil, sehingga tidak akan memakan waktu yang terlalu lama. Sehingga dapat disimpulkan bahwa banyak data latih berbanding lurus dengan

besar akurasi yang didapatkan namun berbanding terbalik dengan waktu eksekusi progam.

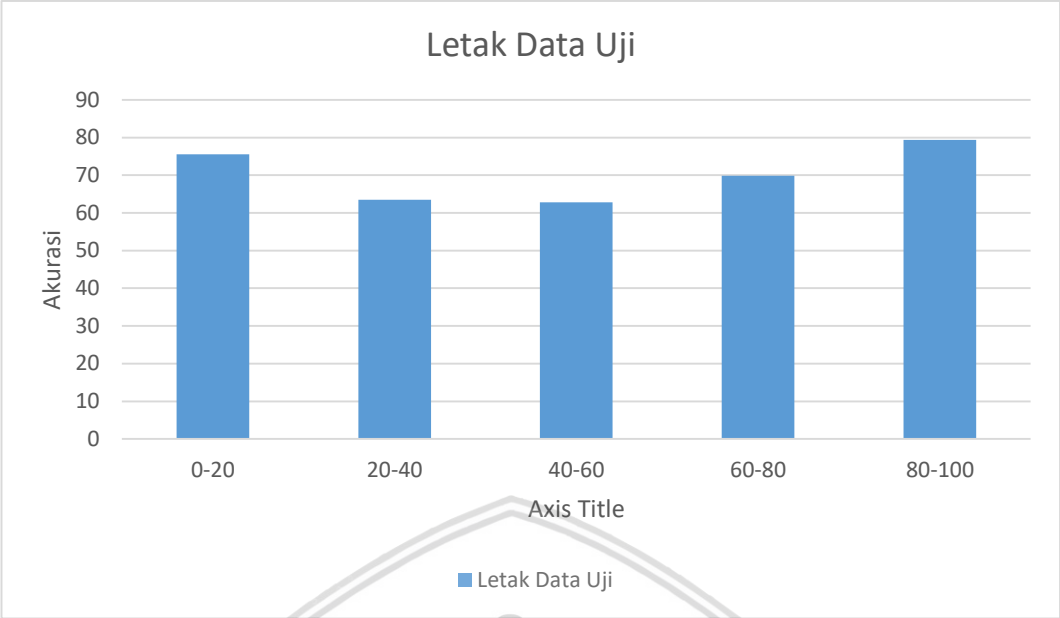
6.2.4 Hasil Pengujian *Cross Validation*

Cross Validation merupakan teknik pengujian untuk memberi validitas kestabilan akurasi bila komponen data diacak. Pengujian *Cross Validation* dapat memberi gambaran tingkat akurasi sebenarnya dari sebuah metode klasifikasi. Pengujian ini, dari 100% dataset maka akan dibagi menjadi 5 *fold* dengan skema 1 *fold* sebagai data uji dan 4 *fold* sebagai data latih. Berikut merupakan grafik dari akurasi yang dihasilkan oleh metode C4.5 berdasarkan uji *cross validation*.

Tabel 6.4 Hasil dari pengujian *cross validation*

No	Letak Data Uji	Akurasi
1	*Fold 1*	75,53%
2	*Fold 2*	63,44%
3	*Fold 3*	62,76%
4	*Fold 4*	69,87%
5	*Fold 5*	79,34%

Dari pengujian *cross validation* pada Tabel 6.4 menunjukan bahwa ketika data uji terletak antara 0% - 20% dan data latih terletak pada jarak 20% - 80% maka akurasi dari metode C4.5 sebesar 75,53%, ketika data uji terletak antara 20% - 40% dan data latih terletak pada jarak 40% - 100% dan 0% - 20% maka akurasi dari metode C4.5 sebesar 63,44%, ketika data uji terletak antara 40% - 60% dan data latih terletak pada jarak 60% - 100% dan 0% - 40% maka akurasi dari metode C4.5 sebesar 62,76%, ketika data uji terletak antara 60% - 80% dan data latih terletak pada jarak 80% - 100% dan 0% - 60% maka akurasi dari metode C4.5 sebesar 69,87%, ketika data uji terletak antara 80% - 100% dan data latih terletak pada jarak 0% - 80% maka akurasi dari metode C4.5 sebesar 79,34%. Bila diamati menggunakan diagram maka ditunjukan seperti pada Gambar 6.4 berikut.



Gambar 6.4 Grafik uji *cross validation*

Dengan pengujian *cross validation* maka akan tampak kestabilan dari algoritma C4.5 dalam mengklasifikasi notas SMILES, karena telah merepresentasikan keseluruhan hasil dari pengujian data dan pelatihan data. Untuk mendapatkan akurasi valid dari pengujian *cross validation* maka prosentasi akurasi dari setiap set data uji memiliki nilai rata-rata sebesar 70,18%.

Pengujian yang dilakukan hanya dengan skema komposisi 0% - 80% sebagai data latih dan 80% - 100% sebagai data uji, kurang merepresentasikan akurasi sebenarnya dari algoritma C4.5 hal ini dikarenakan masih ada kemungkinan bila data uji dari 80% - 100% dataset notasi SMILES digunakan sebagai data latih dan data latih dari 0% - 80% digunkana sebagai data uji, maka sistem akan lebih tidak mengenali atau malah lebih baik dalam mengenali pola data uji yang diujikan. Sehingga masih ada kemungkinan akurasi yang dihasilkan dapat menjadi lebih baik ataupun lebih buruk. Sehingga disinilah peran dari pengujian *cross validation* yang dengan skenario pengujian membagi dataset menjadi 5 set data dan saling mengacak posisi dari data latih dan data uji yang digunakan.

Dengan pengujian *cross validation* maka akan tampak kestabilan dari algoritma C4.5 dalam mengklasifikasi notas SMILES, karena telah merepresentasikan keseluruhan hasil dari pengujian data dan pelatihan data. Untuk mendapatkan akurasi valid dari pengujian *cross validation* maka prosentasi akurasi dari setiap set data uji diambil nilai rata-ratanya. Menurut data hasil pengujian dari Gambar 6.4 maka didapatkan.

BAB 7 PENUTUP

7.1 Kesimpulan

Berdasarkan pengujian dan analisis hasil dari pengujian yang dilakukan maka dapat ditarik kesimpulan sebagai berikut:

1. Proses klasifikasi notasi SMILE diawali dengan proses *preprocessing data* yang akan menghasilkan data nilai setiap fitur atribut. Data setiap fitur atribut tersebut akan dilakukan proses diskritisasi menjadi 3 kelompok diskrit. Data yang telah didiskritisasi akan digunakan pada proses pelatihan data. Proses dari pelatihan data yaitu mencari nilai entropi dan gain dari setiap atribut fitur. Atribut fitur dengan nilai gain tertinggi akan menjadi akar dan dilakukan secara rekursif untuk noda dibawahnya sehingga membentuk sebuah rule pohon keputusan. Hasil dari rule tersebut akan digunakan untuk proses pengujian data.
2. Tingkat akurasi terbaik didapatkan adalah ketika teknik diskritisasi yang dilakukan menggunakan teknik diskritisasi *entropy based*, melakukan pembagian nilai panjang notasi SMILES pada setiap atribut fitur, dan penggunaan data latih sebanyak mungkin yaitu akan menghasilkan nilai akurasi sebesar 79,34%. Sedangkan akurasi dari pengujian *cross validation* menunjukan angka akurasi sebesar 70,18%.

7.2 Saran

Saran untuk pengembangan penelitian selanjutnya diantaranya:

1. Dapat menambah jumlah data latih yang lebih banyak untuk mendapatkan tingkat akurasi maksimal dari algoritma C4.5 dalam mengklasifikasi notasi SMILES.
2. Pengunaan teknik diskritisasi yang lebih efektif untuk mempermudah proses pelatihan data dari algoritma C4.5.

DAFTAR PUSTAKA

Abdillah, S. 2011. Penerapan Algoritma Pohon Keputusan C4.5 Untuk Diagnosa Penyakit Stroke Dengan Klasifikasi *Data Mining* Pada Rumah Sakit Santa Maria Pemalang. Semarang: Program Studi Teknik Informatika – S1 Fakultas Ilmu Komputer Universitas Dian Nuswantoro

Ariona, R. 2013. Belajar HTML dan CSS. Surabaya: Ariona.net

Astamal, R. 2008. DASAR-DASAR WEB PROGRAMMING. Surabaya: Linux User Group STIKOMP Surabaya.

Berpendidikan. 2015. Bahan Kimia dalam Kehidupan Sehari-hari dalam Rumah Tangga beserta Peranan dan Manfaatnya. http://www.berpendidikan.com/2015/12/bahan-kimia-dalam-kehidupan-sehari-hari-dalam-rumah-tangga-beserta-peranan-dan-manfaatnya.html. Diakses tanggal 27 Januari 2018

Darusman, L. K., Batubara, I., Mitsunaga, T., Rahminiwati, M., Djauhari, E., Yamauchi, K. 2012. *Tyrosinase Kinetic Inhibition of Active Compounds from Intsia palembanica.* Bogor: Biopharmaca Research Center, Bogor Agricultural University, Jl Taman Kencana No. 3, Bogor, 16151, Indonesia.

Id, I. D. 2011. FRAMEWORK CODEIGNITER SEBUAH PANDUAN DAN BEST PRACTICE. Pekanbaru: www.koder.web.id.

Junaedi, H. 2011. Penggambaran Rantai Karbon Dengan Menggunakan SIMPLIFIED MOLECULAR MASUKAN LINE SYSTEM (SMILES). Surabaya: Progam Studi Sistem Informasi Sekolah Tinggi Teknik Surabaya.

Erawan, L. 2014. DASAR-DASAR PHP. Depok: Sistem Informasi Fakultas Ilmu Komputer Universitas Indonesia.

Fitzgerad, M. 2012. Introducing *Regular Expression*. United States of America: O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472

Kalsum, U. 2009. Penggunaan Pohon Keputusan (Pohon Keputusan) Untuk Pengambilan Keputusan Dalam Penerimaan Pegawai (Studi Kasus : Perusahaan Asuransi Takaful). Pekanbaru: Jurusan Teknik Informatika Fakultas Sains Dan Teknologi Universitas Islam Negeri Sultan Syarif Kasim Riau Pekanbaru

Mashlahah, S. 2013. Prediksi Kelulusan Mahasiswa Menggunakan Metode Decision Tree dengan Penerapan Algoritma C4.5. Malang: Jurusan Teknis Informatika Fakultas Sains Dan Teknologi Universitas Islam Negeri Maulana Malik Ibraham Malang.

Netbeans. 2016. *NetBeans Developing Applications with NetBeans IDE Release 8.1 E61618-03*. United States of America: Oracle. Inc.

Pambudi, R. H., Setiawan, B. D., Indriati. 2018. Penerapan Algoritma C4.5 dalam Program Untuk Memprediksi Kinerja Siswa Sekolah Menengah. Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer: 2637-2643.

Prajarini, D. 2016. Perbandingan Algoritma Klasifikasi *Data Mining* untuk Prediksi Penyakit Kulit. Yogyakarta: Sekolah Tinggi Seni Rupa Dan Desain Visi Indonesia.

Santoso, T. B. 2017. Analisa Dan Penerapan Algoritma C4.5 untuk Prediksi Loyalitas Pelanggan. Jurnal Ilmiah Fakultas Teknik LIMIT'S :0216 - 1184.

Swastina, L. 2013. Penerapan Algoritma C4.5 Untuk Penentuan Jurusan Mahasiswa. Banjarmasin: Sekolah Tinggi Manajemen Informatika dan Komputer (STMIK) Indonesia Banjarmasin, Indonesia

Raditya, A. 2009. Implementasi *Data Mining* Classification untuk Mencari Pola Prediksi Hujan dengan Menggunakan Algoritma C4.5. Depok: Jurusan Teknik Informatika, Fakultas Teknologi Industri, Universitas Gunadarma

Triisant. 2015. Pohon Keputusan dengan Algoritma C4.5. http://dokumen.tips/documents/algoritma-c45.html. Diakses tanggal 17 Februari 2017.