

BAB II

LANDASAN TEORI

2.1 DATA MINING

Data mining dikenal sejak tahun 1990-an, ketika adanya suatu pekerjaan yang memanfaatkan data menjadi suatu hal yang lebih penting dalam berbagai bidang, seperti marketing dan bisnis, sains dan teknik, serta seni dan hiburan. Sebagian ahli menyatakan bahwa data mining merupakan suatu langkah untuk menganalisis pengetahuan dalam basis data atau biasa disebut Knowledge Discovery in Database (KDD). Data mining merupakan proses untuk menemukan pola data dan pengetahuan yang menarik dari kumpulan data yang sangat besar. Sumber data dapat mencakup database, data warehouse, web, repository, atau data yang dialirkan ke dalam sistem dinamis[3].

Data mining adalah proses berulang di mana kemajuan ditentukan oleh penemuan, baik melalui metode otomatis atau manual. Data mining paling berguna dalam sebuah skenario analisis eksplorasi di mana tidak ada gagasan yang telah ditentukan sebelumnya apa yang akan merupakan hasil yang "menarik". Data mining adalah pencarian baru, berharga, dan informasi nontrivial dalam volume data yang besar. Ini adalah koperasi usaha manusia dan komputer. Hasil terbaik dicapai dengan menyeimbangkan pengetahuan pakar manusia dalam menjelaskan masalah dan tujuan dengan kemampuan pencarian komputer[4].

Data mining merupakan serangkaian proses untuk menggali nilai tambah dari suatu kumpulan data berupa pengetahuan yang selama ini tidak diketahui[5].

Data mining adalah sebuah proses pencarian secara otomatis informasi yang berguna dalam tempat penyimpanan data berukuran besar. Teknik data mining digunakan untuk memeriksa basis data berukuran besar sebagai cara untuk menemukan pola yang baru dan berguna. Namun tidak semua pekerjaan pencarian informasi dapat dinyatakan sebagai data mining[6].

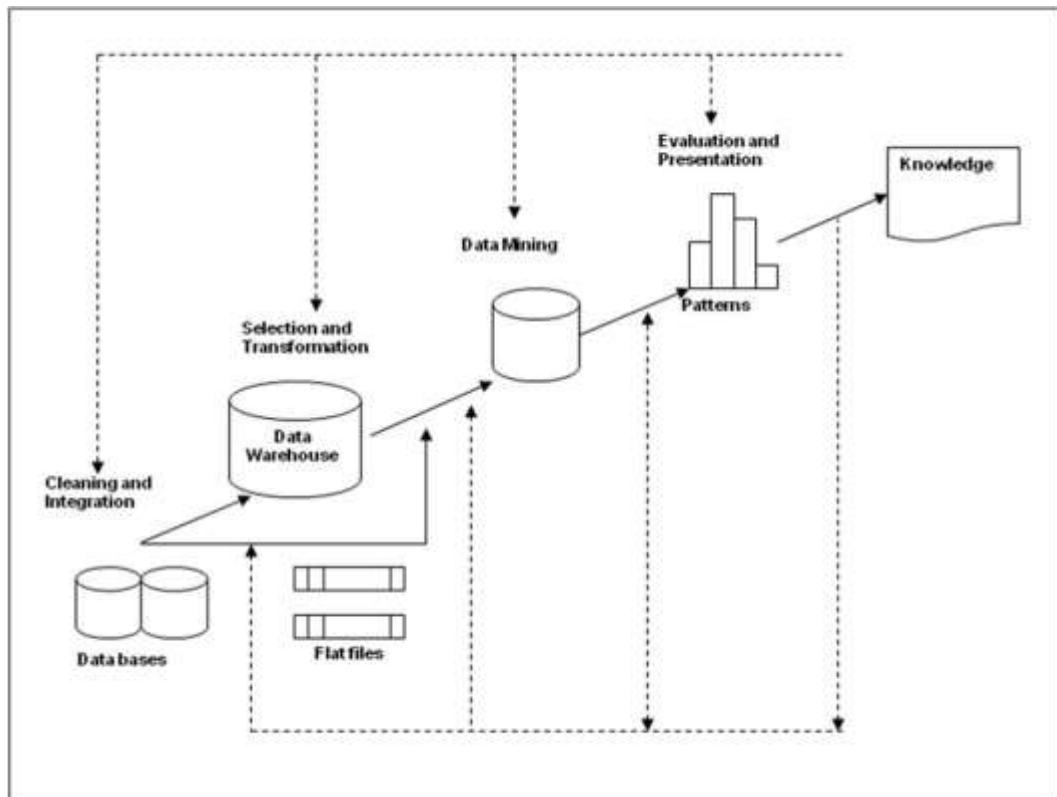
Data mining adalah proses penemuan korelasi, pola, dan tren baru yang bermakna dengan memilah-milah sejumlah besar data disimpan dalam repositori, menggunakan teknologi pengenalan pola serta statistik dan teknik matematika.

Dalam penerapannya, dua tujuan utama dari *data mining* adalah prediksi dan deskripsi. Prediksi melibatkan penggunaan beberapa variabel atau bidang dalam kumpulan data untuk memprediksi nilai yang tidak diketahui atau masa depan dari variabel lain diminati. Namun, deskripsi, berfokus pada menemukan pola yang menggambarkan data yang dapat diinterpretasikan manusia. Oleh karena itu, dimungkinkan untuk menempatkan aktivitas *data mining* ke dalam salah satu dari dua aktivitas dibawah ini[4]:

1. *Data mining prediktif*, yang menghasilkan model sistem yang dijelaskan oleh kumpulan data yang diberikan, atau
2. *Data mining deskriptif*, yang menghasilkan informasi baru yang berbasis nontrivial pada kumpulan data yang tersedia

2.1.1 Tahapan Data Mining

Beberapa tahapan proses *data mining* yang sesuai dengan proses KDD seperti yang digambarkan pada gambar 2.1 [7]:



Gambar 2. 1 Tahapan Data Mining

1. Cleaning and Integration

a. *Data Cleaning* (Pembersih Data)

Data cleaning (Pembersihan data) adalah proses yang dilakukan untuk menghilangkan noise pada data yang tidak konsisten atau bisa disebut tidak relevan. Data yang diperoleh dari database suatu perusahaan maupun hasil eksperimen yang sudah ada, tidak semuanya memiliki isian yang sempurna misalnya data yang hilang, data yang tidak valid, atau bisa

juga hanya sekedar salah ketik. Data yang tidak relevan itu dapat ditangani dengan cara dibuang atau sering disebut dengan proses cleaning. Proses cleaning dapat berpengaruh terhadap performa dari teknik data mining

b. *Data Integration* (Integrasi data)

Integrasi data merupakan proses penggabungan data dari berbagai database sehingga menjadi satu database baru. Data yang diperlukan pada proses data mining tidak hanya berasal dari satu database tetapi juga dapat berasal dari beberapa database.

2. Selection and Transformation

a. *Data Selection* (Seleksi Data)

Tidak semua data yang terdapat dalam database akan dipakai, karena hanya data yang sesuai saja yang akan dianalisis dan diambil dari database. Misalnya pada sebuah kasus market basket analysis yang akan meneliti faktor kecenderungan pelanggan, maka tidak perlu mengambil nama pelanggan, cukup dengan id pelanggan saja.

b. *Data Transformation* (Transformasi Data)

Transformasi data merupakan proses pengubahan data dan penggabungan data ke dalam format tertentu. Data mining membutuhkan format data khusus sebelum diaplikasikan. Misalnya metode standar seperti analisis asosiasi dan clustering hanya bisa menerima input data yang bersifat kategorikal. Karenanya data yang berupa angka numerik apabila

mempunyai sifat kontinyu perlu dibagi-bagi menjadi beberapa interval.

Proses ini sering disebut dengan transformasi data.

3. Proses Mining

Proses mining dapat disebut juga sebagai proses penambangan data. Proses mining merupakan proses utama yang menggunakan metode untuk menemukan pengetahuan berharga yang tersembunyi dari data.

4. Evaluation and Prediction

a. Evaluasi pola

Evaluasi pola bertugas untuk mengidentifikasi pola-pola yang menarik ke dalam *knowledge based* yang ditemukan. Pada tahap ini dihasilkan polapola yang khas dari model klasifikasi yang dievaluasi untuk menilai apakah hipotesa yang ada memang tercapai. Bila ternyata hasil yang diperoleh tidak sesuai dengan hipotesa, terdapat beberapa alternatif yang bisa diambil seperti menjadikannya umpan balik untuk memperbaiki proses data mining, atau mencoba metode *data mining* lain yang lebih sesuai.

b. Presentasi Pengetahuan

Knowledge presentation merupakan visualisasi dan penyajian pengetahuan mengenai metode yang digunakan untuk memperoleh pengetahuan atau informasi yang telah digali oleh pengguna. Tahap terakhir dari proses *data mining* adalah memformulasikan keputusan dari hasil analisis yang didapat.

Berdasarkan pengertian diatas dapat disimpulkan bahwa *data mining* merupakan sebuah proses pencarian secara otomatis informasi yang berguna dalam tempat penyimpanan data berukuran besar. Teknik data mining digunakan untuk memeriksa basis data berukuran besar sebagai cara untuk menemukan pola yang baru dan berguna.

2.1.2 Teknik Data Mining

Teknik data mining merupakan suatu proses utama yang digunakan saat metode diterapkan untuk menemukan pengetahuan berharga dan tersembunyi dari data[7]. Berdasarkan peran data mining dalam melakukan proses prediksi dan mendeskripsikan data, tugas data mining dapat dibagi ke dalam empat kelompok utama, yaitu[6]:

1. *Estimation* (Estimasi)

Teknik untuk melakukan estimasi terhadap sebuah data baru yang tidak memiliki keputusan berdasarkan histori data yang telah ada, dimana variabel target estimasi lebih ke arah numerik daripada ke arah kategori. Model dibangun menggunakan record lengkap yang menyediakan nilai dari variabel target sebagai nilai prediksi. Selanjutnya, pada peninjauan berikutnya estimasi nilai dari variabel target dibuat berdasarkan nilai dari variabel prediksi. Contohnya melakukan estimasi tekanan darah sistolik pada pasien rumah sakit berdasarkan umur pasien, jenis kelamin, indeks berat badan, dan level sodium darah. Hubungan antara tekanan darah sistolik dan nilai variabel prediksi dalam proses

pembelajaran akan menghasilkan model estimasi. Model estimasi yang dihasilkan dapat digunakan untuk memprediksi kasus baru lainnya.

2. *Classification* (Klasifikasi)

Suatu teknik dengan melihat pada kelakuan dan atribut dari kelompok yang telah didefinisikan. Teknik ini dapat memberikan klasifikasi pada data baru dengan memanipulasi data yang ada yang telah diklasifikasi dan dengan menggunakan hasilnya untuk memberikan sejumlah aturan. Salah satu contoh yang mudah dan populer adalah dengan decision tree yaitu salah satu teknik klasifikasi yang paling populer karena mudah untuk interpretasi seperti Algoritma C4.5, ID3 dan lain-lain. Contoh pemanfaatannya misalnya pada bidang akademik terkait klasifikasi siswa yang layak masuk kedalam kelas unggulan atau akselerasi di sekolah tertentu.

3. *Clustering* (Klasterisasi)

Klasterisasi merupakan teknik untuk mengelompokkan data dan membentuk kelas objek-objek yang memiliki kemiripan. Klaster adalah kumpulan record yang memiliki kemiripan satu dengan yang lainnya dan memiliki ketidakmiripan dengan record-record dalam klaster lain. Proses klasterisasi berbeda dengan proses klasifikasi yaitu tidak adanya variabel target dalam pengklusteran. roses klasterisasi mencoba untuk melakukan pembagian terhadap keseluruhan data menjadi kelompok-kelompok yang memiliki kemiripan (homogen), dimana kemiripan data dalam satu kelompok tinggi (maksimal) dan kemiripan data dengan data pada

kolompok lain rendah (minimal). Contoh klasterisasi misalkan proses mendapatkan kelompok-kelompok konsumen untuk target pemasaran dari suatu produk bagi perusahaan yang tidak memiliki dana pemasaran yang besar.

4. *Association* (Asosiasi)

Teknik untuk mengenali kelakuan dari kejadian-kejadian khusus atau proses dimana hubungan asosiasi muncul pada setiap kejadian. Adapun teknik pemecahan masalah yang sering digunakan seperti Algoritma Apriori. Contoh pemanfaatan Algoritma apriori yaitu pada bidang Marketing ketika sebuah Minimarket melakukan tata letak produk yang dijual berdasarkan produk-produk mana yang paling sering dibeli konsumen, selain itu seperti tata letak buku yang dilakukan pustakawan di perpustakaan.

Secara garis besar terdapat 2 pendekatan untuk melakukan teknik - teknik data mining di atas, yaitu:

1. *Supervised Learning* yaitu pembelajaran menggunakan guru dan biasanya ditandai dengan adanya class/label/target pada himpunan data. Supervised learning merupakan sebuah pendekatan dimana sudah terdapat data yang dilatih, dan terdapat variabel yang ditargetkan sehingga tujuan dari pendekatan ini adalah mengelompokkan suatu data ke data yang sudah ada. Adapun teknik-teknik yang digunakan yang bersifat supervised learning seperti Teknik Prediksi dan Klasifikasi seperti Algoritma C4.5, Teknik Rough Set dan lain-lain.

2. *Unsupervised Learning* yaitu pembelajaran tanpa menggunakan guru dan biasanya ditandai pada himpunan datanya tidak memiliki atribut keputusan atau class/label/target. Perbedaan dengan supervised learning, pada unsupervised learning tidak memiliki data latih, sehingga dari data yang ada dikelompokkan menjadi 2 bagian atau 3 bagian dan seterusnya. Adapun teknik-teknik yang bersifat Unsupervised Learning yaitu Teknik Estimasi, Clustering, dan Asosiasi seperti Regresi Linier, Analytical Hierarchy Clustering dan lain-lain.

2.2 KLASIFIKASI

Klasifikasi adalah suatu teknik dengan melihat pada kelakuan dan atribut dari kelompok yang telah didefinisikan. Teknik ini dapat memberikan klasifikasi pada data baru dengan memanipulasi data yang ada yang telah diklasifikasi dan dengan menggunakan hasilnya untuk memberikan sejumlah aturan. Salah satu contoh yang mudah dan populer adalah dengan Decision tree yaitu salah satu metode klasifikasi yang paling populer karena mudah untuk diinterpretasi. Decision tree adalah model prediksi menggunakan struktur pohon atau struktur berhirarki[8].

Klasifikasi adalah bentuk analisis data yang mengekstraksi model yang menggambarkan kelas data penting. Model seperti itu, disebut pengklasifikasi, memprediksi label kelas kategoris (diskrit, tidak berurutan)[3].

Klasifikasi merupakan proses untuk menemukan fungsi dan model yang dapat membedakan atau menjelaskan konsep atau kelas data dalam tujuan memperkirakan kelas yang tidak diketahui dari suatu objek. Dalam proses pengklasifikasian biasa terdapat dua proses yang harus dilakukan yaitu[3]:

1. Proses Training

Pada Proses ini akan digunakan data training set atau data sampel yang telah diketahui label-label atau atribut dari data sampel tersebut untuk membangun model.

2. Proses Testing

Pada proses testing ini dilakukan untuk mengetahui model yang telah dibuat pada training maka dibangun data yang disebut dengan data testing untuk pengklasifikasian label-labelnya.

Beberapa algoritma yang dapat digunakan dalam klasifikasi *data mining* adalah sebagai berikut:

A. *Neural Network*

Neural Network adalah prosesor tersebar paralele yang sangat besar dan memiliki kecenderungan untuk menyimpan pengetahuan yang bersifat pengalaman dan membuatnya siap untuk digunakan.

B. *Decision Tree*

Decision tree merupakan adalah pembelajaran pohon keputusan dari tupel pelatihan berlabel kelas. Pohon keputusan adalah struktur pohon seperti diagram alir, di mana setiap simpul internal (simpul non-daun) menunjukkan pengujian pada atribut, setiap cabang mewakili hasil pengujian, dan setiap simpul daun (atau simpul terminal) memegang label kelas.

C. *Naïve Bayes*

Naïve bayes merupakan sebuah model klasifikasi statistic yang dapat digunakan untuk memprediksi probabilitas keanggotaan suatu kelas. *Naïve bayes* didasarkan pada teorema bayes yang memiliki kemampuan klasifikasi serupa dengan *decision tree* dan *neural network*.

D. *K-Nearest Neighbor*

Klasifikasi *Nearest Neighbor* digunakan ketika semua nilai atribut kontinu, meskipun dapat dimodifikasi untuk menangani atribut kategori. Idennya adalah untuk memperkirakan klasifikasi dari instance yang tidak terlihat menggunakan klasifikasi dari instance atau instance yang paling dekat dengannya, dalam beberapa hal yang perlu kita definisikan[5].

E. *Logistic Regreesion*

Regresi logistic (*Logistic regression*) adalah bagian dari analisis regresi yang digunakan Ketika variable dependen (respon) merupakan variable dikotomi. Variabel dikotomi biasanya terdiri atas dua nilai, yang mewakili kemunculan atau tidak adanya suatu kejadian yang biasanya diberi angka 0 atau 1.

Berdasarkan penjelasan diatas dapat disimpulkan bahwa klasifikasi merupakan kegiatan yang digunakan untuk mengelompokkan data kedalam bentuk kelas-kelas tertentu dalam pengerjaan klasifikasi dapat menggunakan beberapa metode algoritma seperti *Neural Network*, *Decision Tree*, *Naïve Bayes*, *K-Nearest Neighbor*, *Logistic Regression*.

2.3 ALGORITMA NAÏVE BAYES

2.3.1 Teorema Bayes

Naïve bayes merupakan pengklasifikasian statistic yang dapat digunakan untuk memprediksi probabilitas keanggotaan suatu class. *Naïve bayes* didasarkan pada teorema bayes yang memiliki kemampuan klasifikasi berupa dengan decision tree dan neural network. *Naïve bayes* terbukti memiliki akurasi dan kecepatan yang tinggi saat diaplikasikan ke dalam database dengan data yang besar[9].

Prediksi *naïve bayes* didasarkan pada formula teorema *bayes* dengan formula umum sebagai berikut :

$$P(H|X) = \frac{P(X|H) \cdot P(H)}{P(X)}$$

Keterangan :

- X : Data dengan class yang belum diketahui
- H : Hipotesis data X merupakan suatu class spesifik
- $P(H|X)$: Probabilitas hipotesis H berdasarkan kondisi X (posterior)
- $P(H)$: Probabilitas hipotesis H (prior probability)
- $P(X|H)$: Probabilitas X berdasarkan kondisi pada hipotesis H
- $P(X)$: Probabilitas X

Ide dasar dari aturan Bayes adalah bahwa hasil dari hipotesis atas peristiwa (H) dapat diperkirakan berdasarkan pada beberapa bukti (E) yang diamati. Ada beberapa hal penting dalam aturan Bayes tersebut yaitu,

- a. Sebuah probabilitas awal/prior H atau $P(H)$ adalah probabilitas dari suatu hipotesis sebelum bukti diamati.
- b. Sebuah probabilitas akhir H atau $P(H|E)$ adalah probabilitas dari suatu hipotesis setelah bukti diamati.

Penentuan *class* dilakukan dengan cara membandingkan nilai probabilitas suatu sampel berada di *class* yang satu dengan nilai probabilitas suatu sampel berada di *class* yang lain. Untuk menentukan *class* yang sesuai dari suatu sampel dilakukan dengan membandingkan nilai posterior untuk masing-masing *class*, dan mengambil *class* dengan nilai posterior tertinggi[10].

2.3.1 Naïve Bayes Classification

Naïve Bayes Classifier (NBC) merupakan salah satu metode pembelajaran mesin yang memanfaatkan perhitungan probabilitas dan statistic yang dikemukakan oleh ilmuwan Inggris Thomas Bayes, yaitu memprediksi probabilitas di masa depan berdasarkan pengalaman di masa sebelumnya. Tahapan persamaan *naive bayes* untuk klasifikasi sebagai berikut[3]:

Setiap contoh data diwakilkan dengan n dimensional *feature vector*, $X = (x_1, x_2, \dots, x_n)$, dimana n dibuat dari contoh n atribut, berturut-turut $A_1, A_2, A_3, \dots, A_n$.

Misalkan ada m *class*, $C_1, C_2, C_3, \dots, C_m$. Diberikan sebuah contoh data, X (yang belum diketahui label *class* nya), selanjutnya *classifier* akan melakukan prediksi X ke dalam *class* yang memiliki nilai probabilitas posterior paling tinggi berdasarkan kondisi X . *Naïve bayes classifier* akan menetapkan bahwa sampel data X yang belum diketahui sebelumnya ke dalam *class* C_i hanya jika:

$$P(C_i|X) > P(C_j|X) \text{ untuk } 1 \leq j \leq m, \text{ dimana } j \neq i$$

Berdasarkan kondisi tersebut, $P(C_i | X)$ harus dimaksimalkan. Pada $P(C_i | X)$ class C_i merupakan nilai terbesar atau disebut maksimum *posteri hypothesis*,

$$P(C_i | X) = \frac{P(X|C_i)P(C_i)}{P(X)} \quad (1)$$

Penjelasan dari persamaan tersebut adalah sebagai berikut:

X : Kriteria suatu kasus berdasarkan masukan.

C_i : Kelas solusi pola ke- i dimana i adalah jumlah label kelas.

$P(C_i|X)$: Probabilitas kemunculan label kelas C_i dengan kriteria masukan X .

$P(X|C_i)$: Probabilitas kriteria masukan X dengan label kelas C_i .

$P(C_i)$: Probabilitas label kelas C_i .

Nilai $P(X)$ tetap pada semua *class*, yang perlu dimaksimalkan hanya $P(X | C_i)P(C_i)$. Jika probabilitas *prior class* belum diketahui, umumnya ditetapkan nilai setiap *class* sama, yaitu $P(C_1)=P(C_2)=P(C_3)\dots=P(C_m)$, dan berikutnya yaitu hitung nilai $P(X | C_i)$ dan hitung nilai $P(X | C_i)P(C_i)$. Probabilitas *class prior* dapat dihitung dengan cara:

$$P(C_i) = \frac{s_i}{s} \quad (2)$$

s_i merupakan jumlah *training sample* pada *class* C_i di dalam *data training*, sedangkan s merupakan jumlah total *data training sample*.

Jika dataset mencakup banyak atribut, hal ini dapat membuat proses komputasi akan rumit untuk mengestimasi $P(X|C_i)$. Dalam rangka pengurangan beban komputasi dalam mengevaluasi $P(X|C_i)$, *naïve bayes* menetapkan pembuatan

class bersifat independent antara atribut diatur agar bersifat bebas atau independent antara atribut satu dengan atribut lainnya, serta tidak terdapat relasi dependensi antar atribut.

$$P(X|C_i) = \prod_{k=1}^n P(X_k|C_i) = P(x_1|C_i) \times P(x_2|C_i) \times \dots \times P(x_n|C_i) \quad (3)$$

Proses estimasi probabilitas $P(x_1|C_i)$, $P(x_2|C_i)$, $P(x_3|C_i)$... $P(x_n|C_i)$ dari data training akan lebih mudah dilakukan. Dimana x_k adalah nilai pada atribut A_k pada data X . Pada tiap atribut, dilihat atribut tersebut apakah bernilai kontinyu atau kategorikal, sehingga untuk melakukan perhitungan $P(X|C_i)$ mengikuti aturan seperti berikut:

1. Jika nilai A_k bersifat kategorikal, perhitungan $P(X|C_i)$ menjadi

$$P(x_k|C_i) = \frac{s_{ik}}{s_i} \quad (4)$$

s_{ik} merupakan jumlah *training sample* pada *class* C_i yang memiliki nilai x_k untuk A_k sedangkan s_i merupakan jumlah *training sample* yang masuk ke dalam *class* C_i .

2. Jika nilai A_k bersifat kontinyu, terdapat sedikit perbedaan. Sebuah atribut bernilai kontinyu umumnya memiliki distribusi gaussian yang memiliki *mean* (rata-rata) μ serta standar deviasi σ , yang didefinisikan dengan

$$g(x_k, \mu_{C_i}, \sigma_{C_i}) = \frac{1}{\sqrt{2\pi}\sigma_{C_i}} e^{-\frac{(x_k - \mu_{C_i})^2}{2\sigma_{C_i}^2}} \quad (5)$$

sehingga

$$P(X_k|C_i) = g(x_k, \mu_{C_i}, \sigma_{C_i}) \quad (6)$$

dimana,

$$\mu = \frac{1}{n} \sum_{i=1}^n (x_i - \mu)^2 \quad (7)$$

$$\sigma = \frac{1}{n-1} \sum_{i=1}^n (x_i - \mu)^2 \quad (8)$$

Persamaan (6) merupakan fungsi gaussian pada atribut A_k dengan μ_{C_i} serta σ_{C_i} masing-masing merupakan *mean* (rata-rata) dan standar deviasi dari nilai atribut A_k pada *training sample class* C_i .

Proses estimasi label *class* pada *sample* X , $P(X | C_i) P(C_i)$ dievaluasi setiap *class* C_i . *Classifier* mengestimasi label *class* pada *sample* X merupakan *class* C_i hanya jika

$$P(X|C_i)P(C_i) > P(X|C_j)P(C_j) \text{ untuk } 1 \leq j \leq m, j \neq i \quad (9)$$

Persamaan (9) menunjukkan bahwa hasil perkiraan label *class* merupakan *class* C_i pada $P(X | C_i) P(C_i)$ yang memiliki nilai maksimum.

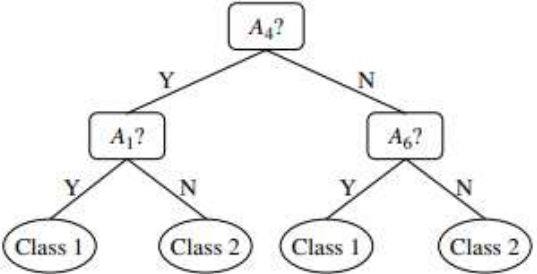
Berdasarkan penjelasan diatas dapat disimpulkan bahwa algoritma *Naïve Bayes* adalah salah satu metode algoritma yang sangat independen (naif) yang digunakan untuk memprediksi probabilitas masa depan melalui pengalaman masa lalu.

2.4 GREEDY FORWARD SELECTION

Greedy Forward Selection merupakan salah satu metode dalam sebuah algoritma *greedy* atas metode seleksi atribut atau seleksi fitur. Metode seleksi atribut *greedy* terdapat beberapa metode seperti : *greedy forward selection*, *greedy backward elimination* dan kombinasi *forward selection* dan *backward elimination*. Mengenai metode seleksi fitur *greedy*, dimana *greedy* mencari atribut terbaik atau terburuk dari suatu data dengan menggunakan pengukuran *information gain*[3].

Han et al. [3] menyatakan bahwa “Algoritma *greedy* dapat disebut juga dengan *stepwise* dan weka menggunakan algoritma ini dengan sebutan *stepwise*”

Atribut terbaik dan terburuk biasanya ditentukan dengan menggunakan uji signifikansi statistik, yang mengasumsikan bahwa atribut tidak bergantung satu sama lain. Banyak ukuran evaluasi atribut lainnya dapat digunakan seperti ukuran perolehan informasi[3]. Metode heuristik dasar pemilihan subset atribut mencakup teknik-teknik berikut, beberapa di antaranya diilustrasikan pada Gambar 2.2.

Forward selection	Backward elimination	Decision tree induction
Initial attribute set: $\{A_1, A_2, A_3, A_4, A_5, A_6\}$ Initial reduced set: $\{\}$ $\Rightarrow \{A_1\}$ $\Rightarrow \{A_1, A_4\}$ $\Rightarrow$ Reduced attribute set: $\{A_1, A_4, A_6\}$	Initial attribute set: $\{A_1, A_2, A_3, A_4, A_5, A_6\}$ $\Rightarrow \{A_1, A_3, A_4, A_5, A_6\}$ $\Rightarrow \{A_1, A_4, A_5, A_6\}$ $\Rightarrow$ Reduced attribute set: $\{A_1, A_4, A_6\}$	Initial attribute set: $\{A_1, A_2, A_3, A_4, A_5, A_6\}$  $\Rightarrow$ Reduced attribute set: $\{A_1, A_4, A_6\}$

Gambar 2. 2 Metode *greedy* (heuristik) untuk pemilihan subset atribut

Algoritma *greedy* dengan seleksi subset atribut sebagai berikut[3]:

1. *Stepwise forward selection*

Prosedur dimulai dengan himpunan kosong dari atribut sebagai set yang dikurangi, atribut yang terbaik dari atribut asli ditentukan dan ditambahkan

pada set yang kurang. Pada setiap iterasi berikutnya yang terbaik dari atribut asli yang tersisa ditambahkan ke set.

2. *Stepwise backward elimination*

Prosedur dimulai dengan kumpulan atribut lengkap. Di setiap langkah, ini menghilangkan atribut terburuk yang tersisa di set

3. Kombinasi dari *forward selection* dan *backward elimination*

Stepwise forward selection dan *stepwise backward elimination* dapat digabungkan sehingga, pada setiap langkah, prosedur memilih atribut terbaik dan menghilangkan yang terburuk dari atribut yang tersisa.

4. *Decision tree induction*

Algoritma *decision tree* awalnya ditujukan untuk klasifikasi. Induksi *decision tree* membangun struktur seperti *flowchart* dimana setiap node internal (non-daun) menunjukkan tes pada atribut, setiap node eksternal (daun) menunjukkan prediksi kelas. Di setiap node, algoritma memilih atribut “terbaik” untuk mempartisi data ke dalam kelas individual. Ketika induksi *decision tree* digunakan dalam membangun pohon keputusan untuk klasifikasi. Pada *forward selection*, dengan dimulai dengan tidak ada fitur: $F = \emptyset$. Pada setiap Langkah, untuk semua kemungkinan x_i , melatih model pada set pelatihan dan menghitung $E(F \cup x_i)$ pada set validasi. Kemudian pilih input x_j yang menyebabkan kesalahan paling kecil[11].

$$j = \operatorname{argmin} E(F \cup X_i)$$

dan

$$\text{tambah } X_j \text{ ke } E(F \cup X_j) < E(F)$$

Prosedur *Forward Selection* [12]:

1. *Forward selection* dimulai pada saat tidak ada variable pada model.
2. Untuk variable pertama yang masuk ke model, pilih predictor yang paling berkorelasi tinggi dengan target. (tanpa kehilangan keumuman, menunjukan variable ini) Jika hasilnya model tidak signifikan, berhenti dan laporkan bahwa tidak ada variable mungkin F dan F. Pilih variable dengan sekuensial terbesar F-statistik.

Berdasarkan penjelasan diatas dapat disimpulkan bahwa metode *greedy forward selection* merupakan metode seleksi fitur dimana *greedy* mencari atribut terbaik atau terburuk dengan menggunakan *information gain*.

2.5 TOOLS DATA MINING

2.5.1 RAPID MINER

RapidMiner merupakan perangkat lunak yang bersifat terbuka (*open source*). *RapidMiner* adalah sebuah solusi untuk melakukan analisis terhadap data mining, text mining dan analisis prediksi. *RapidMiner* menggunakan berbagai teknik deskriptif dan prediksi dalam memberikan wawasan kepada pengguna sehingga dapat membuat keputusan yang paling baik. *RapidMiner* memiliki kurang lebih 500 operator data mining, termasuk operator untuk input, output, data preprocessing dan visualisasi. *RapidMiner* merupakan software yang berdiri sendiri untuk analisis data dan sebagai mesin data mining yang dapat diintegrasikan pada produknya sendiri. *RapidMiner* ditulis dengan menggunakan bahasa java sehingga dapat bekerja di semua sistem operasi[8].

RapidMiner memiliki beberapa sifat sebagai berikut[8]:

- Ditulis dengan Bahasa pemrograman java sehingga dapat dijalankan di berbagai sistem operasi.
- Proses peneuan pengetahuan dimodelkan sebagai operator trees.
- Representasi XML internal untuk memastikan format standar pertukaran data.
- Bahasa scripting memungkinkan untuk eksperimen skala besar dan otomatisasi eksperimen..
- Konsep multi-layer untuk menjamin tampilan data yang efisien dan menjamin penanganan data.
- Memiliki GUI, command line mode, dan Java API yang dapat dipanggil dari program lain.

Terdapat beberapa fitur yang ada pada *rapidminer*, antara lain[8]:

- Banyaknya algoritma data mining, seperti decision treee dan self-organization map.
- Bentuk grafis yang canggih, seperti tumpang tindih diagram histogram, tree chart dan 3D Scatter plots.
- Banyaknya variasi plugin, seperti text plugin untuk melakukan analisis teks.
- Menyediakan prosedur data mining dan machine learning termasuk: ETL (extraction, transformation, loading), data preprocessing, visualisasi, modelling dan evaluasi
- Proses data mining tersusun atas operator-operator yang nestable, dideskripsikan dengan XML, dan dibuat dengan GUI

- Mengintegrasikan proyek data mining Weka dan statistika R

2.5.2 *Waikato Environment for Knowledge Analysis (Weka)*

Weka merupakan sebuah *tools data mining open source* yang berbasis java dan berisi kumpulan algoritma *machine learning* dan data *pre-processing*. *Waikato Environment for Knowledge Analysis* adalah kepanjangan dari Weka yang dibuat di Universitas Waikato, New Zealand yang digunakan untuk pendidikan, penelitian, dan aplikasi eksperimen *data mining*. Weka mampu menyelesaikan masalah-masalah *data mining* yang ada di dunia, khususnya klasifikasi yang mendasari pendekatan *machine learning*. Weka dapat digunakan dan diterapkan pada beberapa tingkatan yang berbeda. Weka menyediakan pengimplementasian algoritma pembelajaran *state of the art* yang dapat diterapkan pada *dataset* dari *command line*. [7]

Dalam Weka terdapat *tools* untuk *preprocessing data*, *clustering*, aturan asosiasi, klasifikasi, regresi, dan visualisasi. Adanya *tools* yang sudah tersedia pengguna dapat melakukan *preprocess* pada data, kemudian memasukkannya dalam sebuah skema pembelajaran, dan menganalisis *classifier* yang dihasilkan serta performanya. Semua itu dilakukan tanpa menulis kode program sama sekali. Contoh penggunaan weka adalah menerapkan sebuah metode pembelajaran ke *dataset* dan menganalisis hasilnya untuk memperoleh informasi tentang data, atau menerapkan beberapa metode dan membandingkan performa untuk dipilih. [7]



Gambar 2. 3 Tampilan Halaman Utama Weka

Tampilan awal ketika aplikasi weka dijalankan maka terlihat seperti pada Gambar 2.3, pada tampilan awal weka terdapat empat menu utama diantaranya *program*, *visualisation*, *tools*, dan *help* serta lima tombol diantaranya *explorer*, *experimenter*, *knowledgeflow*, *woekbench*, dan *simple CLI*.

1. Program

Menu program mempunyai 4 sub menu diantaranya:

a. *LogWindow*(Shortcut CTRL+L)

Sub menu *LogWindow* digunakan untuk menampilkan log yang merekap semua yang tercetak untuk stdout dan stderr.

b. *Memory usage* (Shortcut CTRL+M)

Sub menu *memory usage* digunakan untuk menampilkan penggunaan memori pada saat aplikasi weka digunakan.

c. *Setting*

Sub menu *setting* digunakan untuk mengatur tampilan pada *user interface*.

d. *Exit (Shortcut CTRL+E)*

Submenu *exit* digunakan untuk keluar dari aplikasi weka

2. *Visualisation*

Menu *visualisation* merupakan sarana yang digunakan untuk memvisualisasikan data dengan aplikasi weka. Menu ini mempunyai lima submenu, diantaranya:

a. *Plott (Shortcut CTRL+P)*

Sub menu plot digunakan untuk menampilkan plot 2D dari sebuah *dataset*.

b. *ROC (Shortcut CTRL+R)*

Sub menu ROC digunakan untuk menampilkan kurva ROC yang telah disimpan sebelumnya.

c. *TreeVisualizer (Shortcut CTRL+T)*

Sub menu *TreeVisualizer* digunakan untuk menampilkan graf berarah, contohnya: *decision tree*

d. *Graph Visualizer (Shortcut CTRL+G)*

Sub menu *graph visualizer* digunakan untuk memvisualisasikan format grafik XML, BIF, atau DOT, contohnya sebuah jaringan bayesian.

e. *Boundary Visualizer (Shortcut CTRL+B)*

Sub menu boundary visualizer bertugas untuk mengizinkan visualisasi dari batas keputusan classifier dalam plot 2D.

3. *Tools*

Menu tools menampilkan aplikasi lainnya yang berguna bagi pengguna. Pada menu ini terdapat empat sub menu, yaitu:

a. *Package Manager (Shortcut CTRL+U)*

b. *ArffViewer* (Shortcut CTRL+A)

Sebuah aplikasi MDI yang menampilkan *file* Arff dalam format *spreadsheet*.

c. *SqlViewer* (Shortcut CTRL+S)

Merepresentasikan sebuah lembar kerja SQL, untuk melakukan query database via JDBC.

d. *Bayes net editor* (Shortcut CTRL+N)

Sebuah aplikasi untuk mengedit, memvisualisasikan dan mempelajari bayes net.

4. Help

Menu *help* mempunyai empat sub menu, diantaranya:

a. *Weka Homepage* (Shortcut CTRL+H)

b. *HOWTOs, code snippets, etc.* (Shortcut CTRL+W)

c. *Weka on sourceforge* (Shortcut CTRL+F)

d. *System info* (Shortcut CTRL+I)

Tools yang dapat digunakan untuk *preprocessing dataset* membuat pengguna dapat berfokus pada algoritma yang digunakan tanpa terlalu memperhatikan detail seperti pembacaan data dari *file-file*, penyediaan kode untuk evaluasi hasil, dan implementasi algoritma *filtering*. [7]

2.6 STROKE

Stroke atau *Cerebrovascular disease* menurut *World Health Organization* (WHO) adalah tanda-tanda klinis yang berkembang cepat akibat gangguan fungsi otak fokal atau global karena adanya sumbatan atau pecahnya pembuluh darah di otak dengan gejala-gejala yang berlangsung selama 24 jam atau lebih [13] .

Klasifikasi penyakit stroke terdiri dari beberapa kategori, diantaranya: berdasarkan patologis, secara garis besar stroke dibagi dalam 2 tipe yaitu: *ischemic stroke* disebut juga *ifark* atau *non-hemorrhagic* disebabkan oleh gumpalan atau penyumbatan dalam arteri yang menuju ke otak yang sebelumnya sudah mengalami proses aterosklerosis. *Ischemic stroke* terdiri dari tiga macam yaitu *embolic stroke*, *thrombotic stroke* dan *hipoperfusi stroke*. Tipe kedua adalah *hemorrhagic stroke* merupakan kerusakan atau “ledakan” dari pembuluh darah di otak, pendarahan dapat disebabkan lamanya tekanan darah tinggi dan aneurisma otak. Ada dua tipe stroke *hemorrhagic*: *subarchnoid* dan *intraserbral* [14].

Penyebab utama stoke diantaranya pasien stroke yang tebiasa mengkonsumsi makanan yang mengandung lemak jenuh yang menimbulkan aterosklerosis, yaitu menyempitnya pembuluh arteri disebabkan lemak menempel pada dinding arteri. Para ahli menganggap bahwa aterosklerosis merupakan penyebab utama stroke pada umumnya. Di zaman sekarang, pengobatan dan pencegahan stroke sudah semakin maju walaupun masih tetap mahal[15].

Secara garis besar faktor risiko stroke dibagi atas faktor risiko yang dapat dimodifikasi (*modifiable*) dan yang tidak dapat dimodifikasi (*nonmodifiable*). Faktor risiko stroke yang dapat dimodifikasi diantaranya adalah hipertensi, penyakit jantung (fibrilasi atrium), diabetes melituus, merokok, konsumsi alkohol, hiperlipidemia, kurang aktifitas, dan stenosis arteri karotis. Sedangkan faktor risiko stroke yang tidak dapat dimodifikasi antara lain usia, jenis kelamin, ras/suku, dan faktor genetik[14].

Gejala-gejala ringan stroke dapat dikenali seperti seringnya kesemutan ringan tanpa sebab, sakit kepala atau vertigo ringan, tiba-tiba menggerakkan mulut dan sulit berbicara, lumpuh sebelah serta mendadak pikun dan cadel. Bagi mereka yang pernah mengalami serangan stroke lalu dikemudian hari terkena serangan stroke yang kedua, maka serangan stroke ulangan ini lebih berbahaya dan dapat menyebabkan kematian[16].

2.7 KAJIAN PUSTAKA

Penulis memulai penelitian ini dengan terlebih dahulu melakukan studi kepustakaan berdasarkan kajian dan sumber terkait. Beberapa penelitian yang serupa telah dilakukan dengan menggunakan teknik data mining untuk mengungkap berbagai informasi.

Table 2. 1 Kajian Pustaka

Peneliti dan Tahun	Masalah	Metode	Hasil Penelitian
Vina Adelina, Dian Eka Ratnawati, dan M. Ali Fauzi (2018) [17]	Masalah penyakit stroke ini memerlukan perhatian yang serius karena jumlah kasus yang terus meningkat dan mempunyai angka kematian yang tinggi.	GA-Fuzzy Tsukamoto dan Algoritma Genetika	Berdasarkan hasil analisis yang telah dilakukan dalam proses tersebut diketahui bahwa akurasi sistem menggunakan metode Fuzzy Tsukamoto-GA dari hasil dilakukannya pengoptimalan batasan fungsi keanggotaan adalah 86.66% yang didapatkan. Dengan parameter terbaik yang mempunyai hasil

			optimal yaitu jumlah popsize sebesar 500, kemudian banyak generasi 1000 serta gabungan $C_r=0,5$ dan $M_r=0,6$.
Anissa Puspitawuri, Edi Santoso, dan Candra Dewi (2019) [18]	Penyakit stroke dapat terjadi karena beberapa faktor, diantaranya tekanan darah, riwayat fibrilasi atrium, kolesterol, diabetes dan lain sebagainya. Selama ini penanganan penyakit stroke dilakukan secara manual	<i>K-Nearest Neighbor</i> dan <i>Naïve Bayes</i>	Hasil penelitian menunjukkan nilai akurasi tertinggi yang diperoleh pada data kelas seimbang adalah 96.67% dengan data latih 45, data uji 30 dan nilai $K=15-22$. Sedangkan pada data latih tidak seimbang, menunjukkan akurasi tertinggi sebesar 100% dengan jumlah data latih 60, data uji 30 dan nilai $K=20-30$.
Olivia Aulia Nastiti (2016) [19]	Mendiagnosis penyakit stroke membutuhkan ketelitian yang tinggi karena proses identifikasi cukup sulit yaitu dalam menganalisis dan menelusuri gejala-gejala yang diderita oleh pasien serta melakukan pengambilan keputusan dalam melakukan Tindakan selanjutnya dalam kurun waktu yang sempit.	<i>Naïve Bayes</i> dan <i>Certainty Factor</i>	Berdasarkan analisis terhadap 25 data uji terdapat 105 data latih diperoleh akurasi dari metode <i>Naïve Bayes Classifier</i> sebesar 96% dari metode <i>Certainty Factor</i> sebesar 84%. Program system pakar menggunakan metode <i>Naïve Bayes Classifier</i> dan <i>Certainty Factor</i> ini dapat menjadi alat bantu diagnosis untuk penyakit stroke.

Kenny Riva Sulaeman, Casi Setianingsih, dan Randy Erfa Saputra (2022) [20]	Stroke adalah masalah kesehatan global yang umum dan serius. Di sebagian besar negara, stroke adalah yang kedua atau ketiga penyebab kematian paling umum dan salah satu penyebab utama cacat dewasa yang didapat	<i>Support Vector Machine</i>	Program ini menggunakan klasifikasi SVM yang mendapatkan hasil akurasi tertinggi dari data unbalance pada kernel linear yaitu 76% dan polynomial sebesar 80%. Untuk data yang balanced penulis mendapatkan hasil akurasi pada kernel linear 77%, dan di polynomial 76%.
Nur Diana Saputri, Khalid, Dwi Rolliawati (2022) [21]	Stroke merupakan penyakit tidak menular dan sangat berbahaya karena disebabkan oleh gangguan fungsional otak yang disebabkan oleh tersumbatnya aliran peredaran darah. Penyakit ini tergolong ke dalam <i>cerebrovascular disease</i> karena membutuhkan penanganan selama 24 jam, jika tidak ditangani secara cepat dapat menyebabkan kematian.	Algoritma C4.5, Metode <i>Bagging</i> dan <i>Adabost</i>	Hasil pengujian klasifikasi menggunakan Confusion Matrix dan k-fold cross validation untuk algoritma C4.5 menghasilkan akurasi sebesar 92.87%. Kemudian hasil akurasi dari algoritma C4.5 dengan metode bagging meningkat menjadi 95.02% dan ketika dikombinasikan dengan metode Adaboost nilai akurasinya juga meningkat menjadi 94.63%.

Berdasarkan penelitian sejenis diatas, yang membedakan penelitina ini dengan penelitian sebelumnya adalah:

Pada penelitian [17] memiliki perbedaan pada metode yang digunakan yaitu, penulis akan menggunakan metode *Naïve Bayes* dan *Greedy Forward Selection*. Sedangkan persamaannya terletak pada meneliti prediksi penyakit stroke.

Pada penelitian [18] memiliki perbedaan pada metode yang digunakan yaitu, penulis akan menggunakan metode *Greedy Forward Selection* untuk seleksi atribut. Sedangkan persamaannya terletak pada meneliti prediksi penyakit stroke menggunakan metode *Naïve Bayes*.

Pada penelitian [19] memiliki perbedaan pada metode yang digunakan yaitu penulis akan menggunakan metode *Greedy Forward Selection* untuk seleksi atribut dan *RapidMiner* sebagai alat bantu. Sedangkan persamaannya terletak pada meneliti prediksi penyakit stroke menggunakan metode *Naïve Bayes*.

Pada penelitian [20] memiliki perbedaan pada metode yang digunakan yaitu, penulis akan menggunakan metode *Naïve Bayes* dan *Greedy Forward Selection*. Sedangkan persamaannya terletak pada meneliti prediksi penyakit stroke.

Pada penelitian [21] memiliki perbedaan pada metode yang digunakan yaitu, penulis akan menggunakan metode *Naïve Bayes* dan *Greedy Forward Selection*. Sedangkan persamaannya terletak pada meneliti prediksi penyakit stroke.