BAB II
LANDASAN TEORI

2.1 Data Mining

Secara sederhana, *data mining* merupakan ekstraksi informasi yang tersirat dalam sekumpulan data. Data mining merupakan sebuah proses untuk menggali kumpulan data dan menemukan informasi di dalamnya [2]. Data mining merupakan proses pengekstrakan informasi dari jumlah kumpulan data yang besar dengan menggunakan algoritma dan teknik gambar dari statistik, mesin pembelajaran dan sistem manajemen *database*. Penggalian data ini dilakukan pada sekumpulan data yang besar untuk menemukan pola atau hubungan yang ada dalam kumpulan data tersebut [6]. Hasil penemuan yang diperoleh setelah proses penggalian data ini, kemudian dapat digunakan untuk analisis yang lebih lanjut.

Data mining yang disebut juga dengan *Knowledge-Discovery in Database* (KDD) adalah sebuah proses secara otomatis atas pencarian data di dalam sebuah memori yang amat besar dari data untuk mengetahui pola dengan menggunakan alat seperti klasifikasi, hubungan (*association*) atau pengelompokan (*clustering*). Proses KDD ini terdiri dari langkah-langkah sebagai berikut [4]:

1. *Data Cleaning*, proses menghapus data yang tidak konsisten dan kotor.
2. *Data Integration*, penggabungan beberapa sumber data.
3. *Data Selection*, pengambilan data yang akan dipakai dari sumber data.
4. *Data Transformation*, proses dimana data ditransformasikan menjadi bentuk yang sesuai untuk diproses dalam data mining.
5. *Data Mining*, suatu proses yang penting dengan melibatkan metode untuk menghasilkan suatu pola data.
6. *Pattern Evaluation*, proses untuk menguji kebenaran dari pola data yang mewakili *knowledge* yang ada didalam data itu sendiri.
7. *Knowledge Presentation*, proses visualisasi dan teknik menyajikan *knowledge* digunakan untuk menampilkan *knowledge* hasil *mining* kepada *user*.

2.2 Metode Data Mining

Pada umumnya metode *data mining* dapat dikelompokkan kedalam dua kategori yaitu *deskriptif* dan *prediktif*. Metode *deskriptif* bertujuan untuk mencari pola yang dapat dimengeti oleh manusia yang menjelaskan karakteristik dari data. Metode *prediktif* menggunakan ciri-ciri tertentu dari data. Pada umumnya metode *data mining* dapat dikelompokkan menjadi dua untuk melakukan prediksi.

Metode-metode yang ada dalam *data mining* adalah sebagai berikut [3]:

1. *Classification*

   Klasifikasi (*Classification*) merupakan proses untuk menemukan sekumpulan model yang menjelaskan dan membedakan kelas-kelas data, sehingga model tersebut dapat digunakan untuk memprediksi nilai suatu kelas yang belum diketahui pada sebuah objek. Untuk mendapatkan model, kita harus melakukan analisis terhadap data latih (*training set*). Sedangkan data uji (*test set*) digunakan untuk mengetahui tingkat akurasi dari model yang dihasilkan. Klasifikasi dapat digunakan untuk memprediksi nama atau nilai kelas dari suatu obyek data. Metode inilah yang digunakan dalam tugas akhir ini.

2. *Clustering*

   Pengelompokan (*Clustering*) merupakan proses untuk melakukan segmentasi. Digunakan untuk melakukan pengelompokan secara alami terhadap atribut suatu set data, termasuk kedalam *supervised task*. Contoh *clustering* seperti mengelompokkan dokumen berdasarkan topiknya.

3. *Assosiation*

   Tujuan dari metode ini untuk menghasilkan sejumlah *rule* yang menjelaskan sejumlah data yang berhubung kuat satu dengan yang lainnya. Sebagai contoh *assosiation analysis* dapat digunakan untuk menentukan produk yang datang secara bersamaan oleh banyak pelanggan, atau bisa juga disebut dengan *basket analysis*.

4. *Regression*

   Regression mirip dengan klasifikasi. Perbedaan utamanya adalah terletak pada atribut yang diprediksi berupa nilai yang kontinyu.

5. *Forecasting*

   Prediksi (*Forecasting*) berfungsi untuk melakukan kejadian yang akan datang berdasarkan data sejarah yang ada.

6. *Sequence Analysis*

   Tujuan dari metode ini adalah untuk mengenali pola dari data diskrit. Sebagai contoh adalah menemukan kelompok gen dengan tingkat ekpresi yang mirip.

7. *Deviation Analysis*

   Tujuan dari metode ini adalah untuk menemukan penyebab perbedaan antara data yang satu dengan data yang lain dan biasa disebut sebagai *oulier detection*. Sebagai contoh adalah apakah sudah terjadi penipuan terhadap pengguna kredit dengan melihat catatan transaksi yang tersimpan dalam basis data perusahaan kartu kredit.

2.3 Klasifikasi

Klasifikasi merupakan suatu pekerjaan menilai objek data untuk memasukkannya ke dalam kelas tertentu dari sejumlah kelas yang tersedia. Dalam klasifikasi ada dua pekerjaan utama yang dilakukan, yaitu (1) pembangunan model sebagai prototipe untuk disimpan sebagai memori dan (2) penggunaan model tersebut untuk melakukan pengenalan/klasifikasi/prediksi pada suatu objek data lain agar diketahui dikelas mana objek data tersebut dalam model yang sudah disimpannya. Model dalam klasifikasi mempunyai arti yang sama denga kotak hitam, dimana ada suatu model yang menerima masukan, kemudian mampu melakukan pemikiran terhadap masukan tersebut dan memberikan jawaban sebagai keluaran dari hasil pemikiannya [8].

Tahapan dari klasifikasi dalam data mining terdiri dari [4] :

1. Pembangunan Model

   Pada tahapan ini dibuat sebuah model untuk menyelesaikan masalah klasifikasi class atau atribut dalam data. Tahap ini merupakan fase pelatihan, dimana data latih dianalisis menggunakan algoritma klasifikasi,

sehingga model pembelajaran direpresentasikan dalam bentuk aturan klasifikasi.

2. Penerapan Model

   Pada tahapan ini model yang sudah dibangun sebelumnya digunakan untuk menentukan atribut/kelas dari sebuah data baru yang atribut/kelasnya belum diketahui sebelumnya. Tahap ini digunakan untuk memperkirakan keakuratan aturan klasifikasi terhadap data uji. Jika model dapat diterima, maka aturan dapat diterapkan terhadap klasifikasi data baru.

2.4 Decision Tree

2.4.1 Pengertian Decision Tree

Decision tree merupakan metode klasifikasi *data mining*. *Decision tree* dalam istilah pembelajaran merupakan sebuah struktur pohon dimana setiap *node* pohon mempresentasikan atribut yang telah diuji. Setiap cabang merupakan suatu pembagian hasil uji dan *node* daun (*leaf*) mempresentasikan kelompok kelas tertentu. [5]. Level *node* teratas dari sebuah *Decision Tree* adalah node akar (*root*) yang biasanya berupa atribut yang paling memiliki pengaruh terbesar pada suatu kelas tertentu. Pada umumnya *Decision Tree* melakukan strategi pencarian secara *top-down* untuk solusinya. Pada proses mengklasifikasi data yang tidak diketahui, nilai atribut akan diuji dengan cara melacak jalur dari *node* akar (*root*) sampai *node* akhir (daun) dan kemudian akan diprediksi kelas yang dimiliki oleh suatu data baru tertentu.

2.4.2 Jenis - Jenis Decision Tree

Beberapa model *decision tree* yang sudah dikembangkan antara lain C4.5 atau ID3 dan CART. Berikut ini akan dijelaskan model dari decision tree tersebut :

1. C4.5 atau ID3

   Decision Tree menggunakan algoritma ID3 atau C4.5, yang diperkenalkan dan dikembangkan pertama kali oleh Quinlan yang

merupakan singkatan dari *Iterative Dichotomiser 3* atau *Induction of Decision 3.* Algoritma ID3 membentuk pohon keputusan dengan metode *divide-and-conquer* data secara rekursif dari atas ke bawah. Strategi pembentukan Decision Tree dengan algoritma ID3 adalah:

A. Pohon dimulai sebagai *node* tunggal (akar/*root*) yang merepresentasikan semua data.
B. Sesudah *node root* dibentuk, maka data pada *node* akar akan diukur dengan *information gain* untuk dipilih atribut mana yang akan dijadikan atribut pembaginya.
C. Sebuah cabang dibentuk dari atribut yang dipilih menjadi pembagi dan data akan didistribusikan ke dalam cabang masing-masing.
D. Algoritma ini akan terus menggunakan proses yang sama atau bersifat rekursif untuk dapat membentuk sebuah *Decision Tree*. Ketika sebuah atribut telah dipilih menjadi *node* pembagi atau cabang, maka atribut tersebut tidak diikutkan lagi dalam penghitungan nilai *information gain.*
E. Proses pembagian rekursif akan berhenti jika salah satu dari kondisi dibawah ini terpenuhi :
    a. Semua data dari anak cabang telah termasuk dalam kelas yang sama.
    b. Semua atribut telah dipakai, tetapi masih tersisa data dalam kelas yang berbeda. Dalam kasus ini, diambil data yang mewakili kelas yang terbanyak untuk menjadi label kelas pada *node* daun. Tidak terdapat data pada anak cabang yang baru. Dalam kasus ini, *node* daun akan dipilih pada cabang sebelumnya dan diambil data yang mewakili kelas terbanyak untuk dijadikan label kelas.

Metode C4.5 dan ID3 memiliki perbedaan dalam nilai tiap atribut. Metode C4.5 menggunakan atribut yang bernilai kategorikal dan numerikal, sedangkan metode ID3 menggunakan atribut yang bernilai kategorikal. Metode *decision tree C4.5* inilah yang digunakan dalam tugas akhir ini.

2. CART

CART adalah singkatan dari *Classification And Regression Tree*. Dalam CART ada dua langkah penting yang harus diikuti untuk mendapatkan *tree* dengan performansi yang optimal. Yang pertama adalah pemecahan objek secara berulang berdasarkan atribut tertentu. Yang kedua, *prunning* (pemangkasan) dengan menggunakan data validasi.

Misalkan kita mempunyai variabel independent x1,x2,x3,.....xn dan variabel dependent atau output y. Pemecahan secara berulang berarti kita bagi objek ke dalam kotak-kotak berdasarkan nilai variabel x1,x2 atau xr. Cara ini diulang sehingga dalam suatu kotak sebisa mungkin berisi observasi dalam kelompok atau kelas yang sama.

Langkah berikutnya sesudah dilakukan pemecahan objek atau data secara berulang adalah melakukan *prunning*. Dalam *prunning* kita ingin memangkas *tree* yang mungkin terlalu besar dan terjadi fenomena *overfitting*. *Overvitting* merupakan sebuah satu buah pengelompokkan yang mungkin hanya berisi satu data yang memungkinkan data tersebut merupakan *noise* yang ada di data training dan bukan pola yang mungkin terjadi dalam data testing atau data validasi. *Prunning* terdiri dari beberapa langkah pemilihan secara berulang simpul yang akan dijadikan simpul daun. Dengan mengubah simpul menjadi simpul daun artinya tidak akan dilakukan pemecahan lagi sesudah itu. Dengan demikian ukuran *tree* akan berkurang. [7]

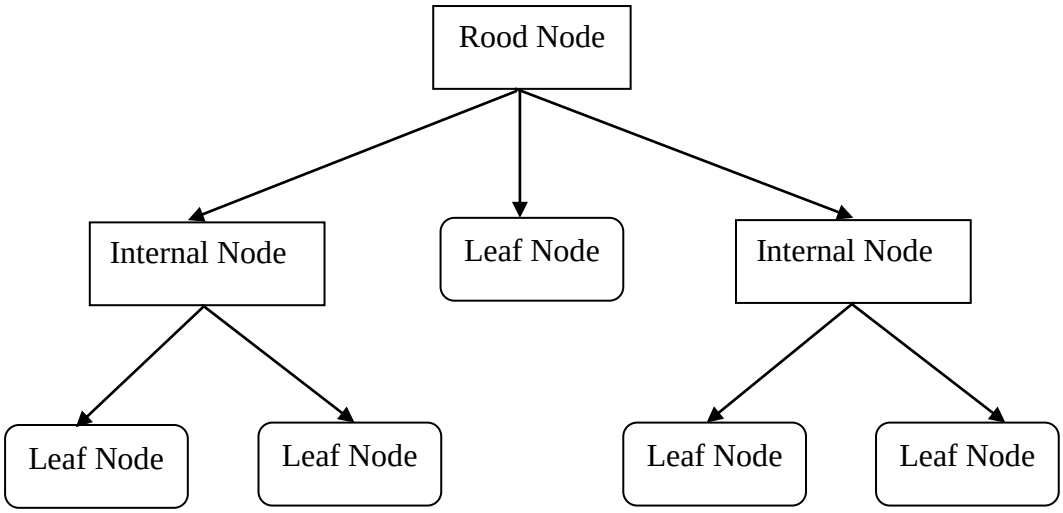
2.4.3 Model Decision Tree

Decision tree adalah *flow-chart* seperti *struktur tree*, dimana tiap *internal node* menunjukkan sebuah test pada sebuah atribut, tiap cabang menunjukkan hasil dari test, dan *leaf node* menunjukkan *class-class* atau *class distribution*.

Selain karena pembangunannya relatif cepat, hasil dari model yang dibangun mudah untuk dipahami. Pada *decision tree* terdapat 3 jenis *node*, yaitu:

a. *Root Node*, merupakan *node* paling atas, pada *node* ini tidak ada *input* dan bisa tidak mempunyai *output* atau mempunyai *output* lebih dari satu.
b. *Internal Node*, merupakan *node* percabangan, pada *node* ini hanya terdapat satu *input* dan mempunyai *output* minimal dua.
c. *Leaf node* atau *terminal node*, merupakan *node* akhir, pada *node* ini hanya terdapat satu *input* dan tidak mempunyai *output*.

Contoh dari model pohon keputusan yaitu seperti pada gambar 2.1 berikut:



Gambar 2.1 Model *Decision Tree*

2.5 Algoritma Decision Tree C4.5

Algoritma C4.5 diperkenalkan oleh Quinlan pada tahun 1996 sebagai versi perbaikan dari ID3. Dalam ID3, induksi decision tree hanya bisa dilakukan pada fitur bertipe kategorikal (nominal atau ordinal), sedangan tipe numerik (interval atau rasio) tidak dapat digunakan.

Yang menjadi hal penting dalam induksi decision tree adalah bagaimana menyatakan syarat pengujian pada node. Ada 3 kelompok penting dalam syarat pengujian node :

1. Fitur biner

   Adalah Fitur yang hanya mempunyai dua nilai berbeda. Syarat pengujian ketika fitur ini menjadi node (akar maupun interval) hanya punya dua pilihan cabang.

2. Fitur kategorikal

   Untuk fitur yang nilainya bertipe kategorikal (nominal atau ordinal) bisa mempunyai beberapa nilai berbeda. Secara umum ada 2 pemecahan yaitu pemecahan biner (*binary splitting*) dan (*multi splitting*).

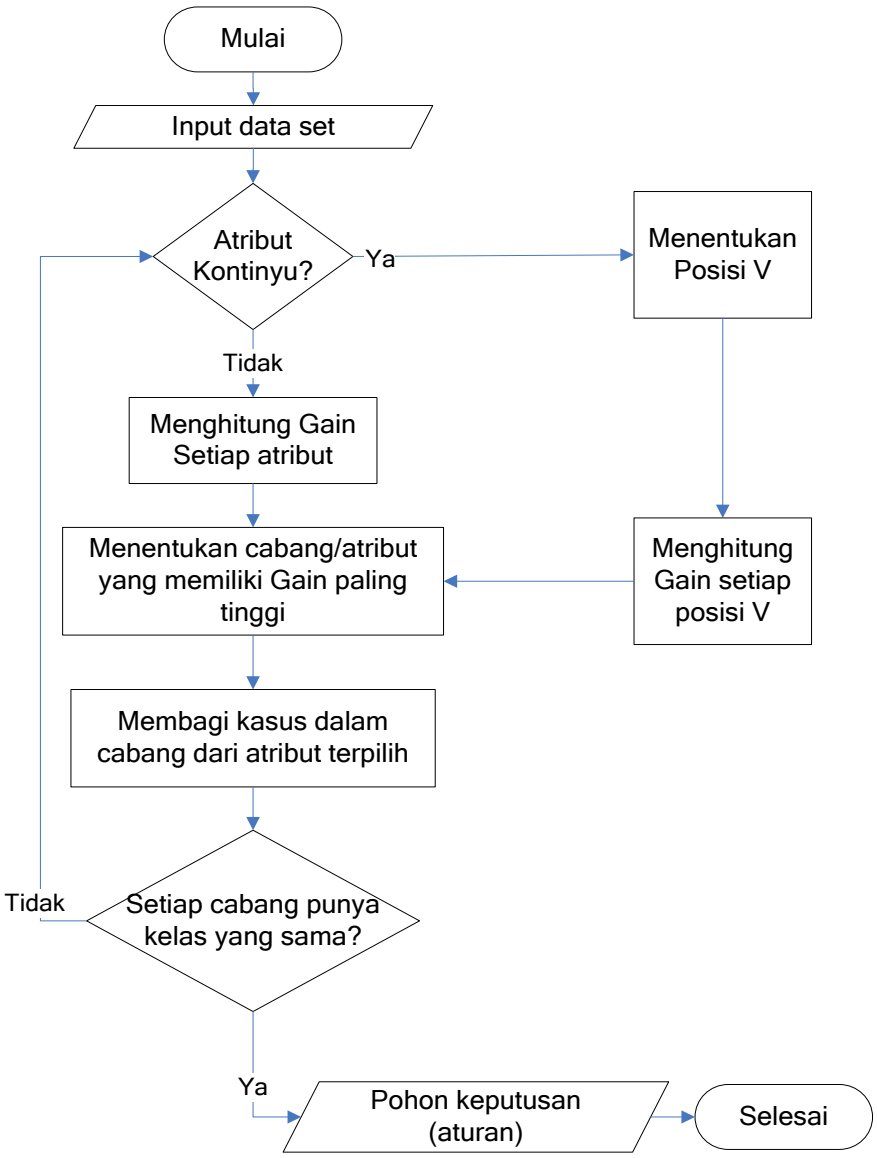
3. Fitur numerik

   Untuk fitur bertipe numerik, Syarat pengujian dalam node (akar maupun internal) dinyatakan dengan pengujian perbandingan ($A \leq V$) atau ($A > V$) dengan hasil biner.

Secara umum algoritma C4.5 untuk membangun pohon keputusan adalah sebagai berikut:

1. Pilih atribut sebagai simpul akar.
2. Buat cabang untuk tiap-tiap nilai.
3. Bagi kasus dalam cabang.
4. Ulangi proses untuk setiap cabang sampai semua kasus pada cabang memiliki kelas yang sama.

Berikut ini akan dijelaskan secara lebih detail algoritma C4.5 menggunakan *flowchart* yang disajikan pada gambar 2.2.



Gambar 2.2 *Flowchart* algoritma *Decision Tree C4.5*

Untuk memilih atribut sebagai simpul akar (*root node*) atau simpul dalam (*internal node*), didasarkan pada nilai *information gain* tertinggi dari atribut-atribut yang ada. Sebelum perhitungan *information gain*, akan dilakukan perhitungan *entropy*. *Entropy* merupakan distribusi probabilitas dalam teori informasi dan diadopsi kedalam algoritma *C*4.5 untuk mengukur tingkat homogenitas distribusi kelas dari sebuah himpunan data (*data set*). Semakin tinggi tingkat *entropy* dari sebuah data maka semakin homogen distribusi kelas pada

data tersebut. Perhitungan *information gain* menggunakan rumus 2.1, sedangkan *entropy* menggunakan rumus 2.2.

$$Gain(S,A) = Entropy\ (S) - \sum_{i=1}^{n} \frac{|S_i|}{|S|} * Entropy(S_i) \quad \text{.....(2.1)}$$

dimana,

S : Himpunan kasus

A : Atribut

n : Jumlah partisi atribut A

|Si|: Jumlah kasus pada partisi ke i

|S| : Jumlah kasus dalam S

$$Entropy(S) = -\sum_{i=1}^{n} p_i * log_2 p_i \quad \text{.....(2.2)}$$

dimana,

S : Himpunan kasus

n : Jumlah partisi S

p_i : Proporsi dari S_i terhadap S

Selain *Information Gain* kriteria yang lain untuk memilih atribut sebagai pemecah adalah *Rasio Gain*. Perhitungan rasio gain menggunakan rumus 2.3, sedangkan split information menggunakan rumus 2.4.

$$GainRasio\ (S,A) = \frac{Gain\ (S,A)}{SplitInformation(S,A)} \quad \text{.....(2.3)}$$

$$SplitInformation(S,A) = -\sum_{i=1}^{c} \frac{S_i}{S} log_2 \frac{S_i}{S} \quad \text{.....(2.4)}$$

dimana S1 sampai Sc adalah c subset yang dihasilkan dari pemecahan S dengan menggunakan atribut A yang mempunyai sebanyak c nilai.

2.6 Penelitian Sebelumnya

Penilitian sebelumnya yang menggunakan metode Naïve Bayes dilakukan oleh Faridatul Choiriyah (Universitas Muhammadiyah Gresik, 2015) dengan judul "Implementasi Metode Naïve Bayes Sebagai Penentu Status Gizi Balita (Study Kasus puskesmas Dukuh Kupang Surabaya)". Algoritma yang digunakan adalah

naïve bayes. Data yang dijadikan inputan dalam sistem klasifikasi status gizi balita diperoleh dari data balita yang terdapat dipegawai bidan KIA puskesmas dukuh kupang surabaya, bulan september tahun 2014 sebanyak 131 balita. Atribut yang terdapat pada tabel mewakili fitur data yang digunakan meliputi jenis kelamin, usia, berat badan, tinggi badan, dan lingkar kepala. Jumlah data yang digunakan sebanyak 131 *record* dengan kelas Baik dan Kurang. Penelitian ini diuji sebanyak 2 kali pengujian dengan rata-rata akurasi sebesar 94%.

Penelitian selanjutnya adalah tentang metode Decision Tree C4.5 dalam penelitian berjudul "Aplikaksi Klasifikasi Penentuan Penerimaan Beras Miskin (Raskin) ini Ds.Sidomulyo Kec.Deket Kab.Lamongan dengan Metode Decision Tree C4.5", dibuat oleh M.Basroni Rizal (Universitas Muhammadiyah Gresik, 2016). Penelitian dilakukan untuk mengklasifikasi penentuan penerimaan beras miskin. Dari 600 data kepala keluarga di Ds. Sidomulyo Kec. Deket Kab. Lamongan, data tersebut diambil 40 % yang akan dijadikan sebagai data uji dan 60 % akan menjadi data latih. Jadi jumlah pembagiannya adalah 240 data sebagai data latih dan 360 data untuk data uji. Penelitian ini diuji sebanyak 7 kali pengujian dengan rata-rata akurasi sebesar 92%.

Penelitian selanjutnya adalah tentang metode *Decision Tree C4.5* dalam penelitian yang berjudul "*System prediksi prestasi akademik mahasiswa menggunakan metode decision tree C4.5 (Studi kasus:Jurusan Teknik informatika UNMUH GRESIK)"*, dibuat oleh Aunur Rasyid (Universitas Muhammadiyah Gresik, 2014). Tujuan dari penelitian tersebut adalah untuk menghasilkan informasi perkiraan kategori prestasi mahasiswa menggunakan metode *Decision Tree C4.5* sebagai peringatan dini dan motivasi mahasiswa dalam mendapatkan prestasi yang maksimal. Atribut-atribut yang digunakan adalah instansi sekolah asal (SMK,SMA atau MA), satatus sekolah asal (Negri atau Swasta), jurusan sekolah asal (IPA,IPS,Bahasa,Teknik,Administrasi), nilai rata-rata UN, status kerja (Sudah atau Belum), dan pihak yang mempengaruhi mahasiswa dalam memilih kuliah (Sendiri,Orang tua atau Orang lain). Hasil dari penelitian tersebut, Sistim Prediksi mahasiswa yang dirancang menggunakan algoritma C4.5 dapat

memprediksi prestasi mahasiswa agar mampu mempertahankan kondisinya atau melakukan perbaikan utuk mencapai prestasi yang maksimal. Hasil akurasi dari penelitian tersebut adalah 90%.