BAB II

TINJAUAN PUSTAKA

II.1. *Data Mining*

Data mining merupakan proses yang menggunakan teknik statistik, matematika, kecerdasan buatan, dan *machine learning* untuk mengekstraksi dan mengidenfikasi informasi yang bermanfaat dan pengetahuan yang terkait dari berbagai *database* besar/*DataWarehouse* (Tampubolon, dkk, 2013 : 96).

Nama *data mining* sebenarnya mulai terkenal sejak tahun 1990, pekerjaan pemanfaat *data mining* menjadi sesuatu yang penting dalam berbagai bidang, mulai dari bidang akademik, bisnis, hingga medis. *Data mining* dapat diterapkan pada bebagai bidang yang mempunyai sejumlah data, tetapi karena wilayah penelitian dengan sejarah yang belum lama, dan belum melewati masa ‘remaja’, maka *data mining* masih diperdebatkan posisi bidang pengetahuan yang memilikinya. Maka, Daryl Pregibon menyatakan bahwa “*data mining* adalah campuran dari statistik”.

Ada istilah lain yang mempunyai makna yang sama dengan *data mining* yaitu *Knowledge-Discovery in Database* (KDD). Memang *data mining* atau KDD bertujuan untuk memanfaatkan data dalam basis data dengan mengolahnya sehingga menghasilkan informasi baru yang berguna. Jika dilacak keilmuannya, ternyata *data mining* mempunyai empat akar bidang ilmu seperti:

1. Statistik

Bidang ini merupakan akar paling tua, tanpa ada statistik maka *data mining* tidak ada. Dengan menggunakan statistik klasik ternyata data yang diolah dapat diringkas dalam apa yang umum dikenal sebagai *Exploratory Data Analysis* (EDA). EDA berguna untuk mengidentifikasi hubungan sistematis antar variabel/fitur ketika tidak ada cukup informasi alami yang dibawahnya. Teknik EDA klasik yang digunakan dalam *data mining* di antaranya:

a. Metode Komputasional: statistik deskritif (distribusi, parameter statistik klasik (mean, medium, rata-rata, varian, dan sebagainya), korelasi, tabel frekuensi, teknik eksplorasi multivariat (analisis diskriminan, *classification tree*, analisis korespondensi), model linear/nonlinear lanjutan (regresi linear/nonlinear, *time series*/*forecasting*, dan sebagainya).
b. Visualisasi Data: mengarah pada representasi informasi dalam bentuk visual dan dapat dipandang sebagai satu yang paling berguna. Pada saat yang sama, visualisasi data merupakan metode eksplorasi data yang atraktif. Teknik visualisasi yang paling umum yang dikenal adalah histogram semua jenis (kolam, silinder, kerucut, piramida, lingkaran, batang, dan sebagainya), kotak, *scatter*, matriks, ikon, dan sebagainya.

2. Kecerdasaan Buatan atau *Artifical Intelligence*(AI)

Bidang ilmu ini berbeda dengan statistik. Teorinya dibangun berdasarkan teknik heuristik sehingga AI berkontribusi terhadap teknik pengolahan informasi berdasarkan pada model penalaran manual. Salah satu cabang dari AI, yaitu pembelajaran mesin atau *machine learning*, merupakan displin ilmu yang paling

penting yang direpresentasikan dalam pembangun *data mining*, menggunakan teknik dimana sistem komputer belajar dengan ‘pelatihan’.

3. Pengenalan Pola

Sebenarnya *data mining* juga menjadi turunan bidang pengenalan pola, tetapi hanya mengelola data dari basis data. Data yang diambil dari basis data untuk diolah bukan dalam bentuk relasi, melainkan dalam bentuk normal pertama sehingga set data dibentuk normal pertama. Akan tetapi, *data mining* mempunyai ciri khas yaitu pencarian pola asosiasi dan pola sekuensial.

4. Sistem Basis Data

Akar bidang ilmu keempat dari *data mining* yang menyediakan informasi berupa data yang akan ‘digali’ menggunakan metode-metode yang disebutkan sebelumnya.

Kebutuhan ‘penggalian’ informasi dari dalam data dapat dilihat pada kasus dunia nyata, diantaranya sebagai berikut:

a. Ekonomi: Ada jumlah data yang sangat besar yang dikumpulkan dari berbagai bidang, seperti data *web*, e-*commerce*, supermarket, transaksi keuangan dan perbankan, dan sebagainya yang siap dianalisis dengan tujuan untuk mendapatkan keputusan yang optimal terkait tujuan lembaga.
b. Pelayan kesehatan: Saat ini ada banyak basis data berbeda dalam bidang pelayanan kesehatan (medis dan farmasi), yang dianalisis secara parsial, khususnya dengan cara medis sendiri, padahal sebenarnya dalam data tersembunyi banyak informasi yang belum dibuka secara tepat.

c. Riset Pengetahuan: Ada basis data besar yang dikumpulkan betahun-tahun dalam bermacam-macam bidang (astronomi, meteorologi, biologi, linguistik, dan sebagainya) yang tidak dapat dieksporasi menggunakan cara tradisional.

Dari penjelasan diatas jelas bahwa disatu sisi ada sejumlah data dalam jumlah besar yang secara sistematis belum dieksplorasi, dan disisi lain, kekuatan teknik komputasi dan ilmu komputer sudah tumbuh secara eksposional sehingga menyebabkan tekanan pada kebutuhan untuk membuka informasi yang 'tersembunyi' dari data menjadi meningkat. Bidang *data mining* menjadi jawaban untuk menyelesaikan persoalan diatas yang awalnya tidak mungkin untuk dideteksi dengan cara tradisional dan hanya menggunakan kemampuan analisis manusia.

Pengertian *data mining* cukup sulit dijelaskan dengan gambar jika mengingat *data mining* juga merupakan gabungan dari beberapa bidang ilmu. Berikut beberapa pengertian *data mining* yang secara naratif mempunyai beberapa maksud yang mirip:

a. Pencarian otomatis pola dalam basis data besar, menggunakan teknik komputasional campuran dari stastik, pembelajaran mesin, dan pengenalan pola.
b. Pengekstrakan implisit non-trival, yang sebelumnya belum diketahui secara potensial adalah informasi berguna dari data.
c. Ilmu pengekstrakan informasi yang berguna dari data atau basis data besar.
d. Eksplorasi otomatis atau semi otomatis dan analisis data dalam jumlah besar, dengan tujuan untuk menemukan pola yang bermakna.

e. Proses penemuan informasi otomatis dengan mengidentifikasi pola dan hubungan ‘tersembunyi’ dalam data (Prasetyo, 2014 : 2).

II.1.1. Tujuan *Data Mining*

Berawal dari disiplin ilmu, *data mining* bertujuan untuk memperbaiki teknik tradisional sehingga bisa menangani :

a. Jumlah data yang sangat besar.
b. Dimensi yang tinggi.
c. Data yang *heterogen* dan berbeda sifat (Purnomo, dkk, 2014 : 25).
d. Mendapatkan hubungan atau pola yang akan mungkin memberikan indikasi yang bermanfaat (Tampubolon, dkk, 2013 : 96).

II.1.2. Fungsi dan Tugas *Data Mining*

Data mining menganalisis data menggunakan *tool* untuk menemukan pola dan aturan dalam himpunan data. Perangkat lunak bertugas untuk menemukan pola dengan mengidentifikasi aturan dan fitur pada data. *Tool data mining* diharapkan mampu mengenal pola ini dalam data dengan *input* minimal dari *user* (Tampubolon, dkk, 2013 : 97).

II.1.3. Pembagian *Data Mining* Berdasarkan Klasifikasi

1. Klasifikasi Berbasis *Decision Tree*

Decision Tree atau pohon keputusan adalah pohon yang digunakan sebagai prosedur penalaran untuk mendapatkan jawaban dari masalah yang

dimasukkan.Pohon yang dibentuk tidak selalu berupa pohon *biner*. Jika semua fitur dalam *data set* menggunakan dua macam nilai katagorikal maka bentuk pohon yang didapatkan berupa pohon *biner* (Prasetyo, 2014 : 56).

Karateristik dari *Decision tree* dibentuk sejumlah elemen sebagai berikut :

a. Node akar, tidak mempunyai lengan masukan dan mempunyai nol atau lebih lengan keluaran.
b. Node internal, setiap node yang bukan daun (nonterminal) yang mempunyai tepat satu lengan masukan dan dua atau lebih lengan keluaran. Node ini menyatakan pengujian yang didasarkan pada nilai fitur.
c. Lengan, setiap cabang menyatakan nilai hasil pengujian di node bukan daun.
d. Node daun (terminal), node yang mempunyai tepat satu lengan masukan dan tidak mempunyai lengan keluaran. Node ini menyatakan label kelas (keputusan).

Decision Tree mempunyai tiga pendekatan klasik :

1. Pohon klasifikasi, digunakan untuk melakukan prediksi ketika ada data baru yang belum diketahui label kelasnya. Pendekatan ini yang paling banyak digunakan.
2. Pohon regresi, ketika hasil prediksi dinggap sebagai nilai nyata yang mungkin akan didapatkan. Misalnya kasus hanya minyak, kenaikan harga rumah, prediksi inflasi tiap tahun, dan sebagainya.
3. *CART* (atau C&RT), ketika masalah klasifikasi dan regresi digunakan bersama-sama (Prasetyo, 2014 : 58).

Yang termasuk klasifikasi berbasis *decision tree* :

1) Algoritma ID3

Algoritma ID3 (*Iterative Dichotomiser 3*) pertama kali diperkenalkan oleh Quinlan yang digunakan untuk menginduksi *decision tree*. Algoritma ID3 dapat bekerja dengan baik pada semua fitur yang mempunyai tipe data kategorikal (nominal atau ordinal). Dalam perkembangannya, ID3 banyak mengalami perbaikan pada versi berikutnya C4.5 atau C5.0 (Prasetyo, 2014 : 59).

2) Algoritma C4.5

Algoritma C4.5 diperkenalkan oleh Quinlan (1996) sebagai versi perbaikan dari ID3. Dalam ID3, induksi *decision tree* hanya bisa dilakukan pada fitur bertipe kategorikal (nominal atau ordinal), sedangkan tipe numerik (interval atau rasio) tidak dapat digunakan. Perbaikan yang membedakan algoritma C4.5 dari ID3 adalah dapat menangani fitur dengan tipe numerik, melakukan pemotongan (*pruning*) *decision tree*, dan penurunan (*deriving*) *rule set*. Algoritma C4.5 juga menggunakan kriteria *gain* dalam menentukan fitur yang menjadi pemecah node pada pohon yang diinduksi (Prasetyo, 2014 : 65).

2. Klasifikasi Berbasis *Artificial Neural Network*

Artificial Neural Network (ANN) merupakan suatu konsep rekayasa pengetahuan dalam bidang kecerdasan buatan yang di desain dengan mengadopsi sistem saraf manusia, dimana pemrosesan utama sistem saraf manusia ada di otak. Bagian terkecil dari otak manusia adalah sel saraf yang merupakan unit dasar

pemrosesan informasi. Unit ini sering disebut sebagai *neuron* (Prasetyo, 2014 : 85).

Yang termasuk klasifikasi berbasis *decision tree* :

1) ANN *Perceptron*

   ANN *Perceptron* merupakan salah satu jenis ANN dengan *layer* tunggal. Pertama kali diperkenalkan oleh Frank Rosenblatt yang berisi algoritma pelatihan yang digunakan untuk membangun model ANN. *Perceptron* yang paling sederhana menggunakan satu *neuron* pemroses karena hanya dengan satu *neuron* pemroses maka *perceptron* dengan stau *neuron* hanya bisa melakukan klasifikasi dua kelas (Prasetyo, 2014 : 88).

2) ANN *Error Backpropagation* (*Multilayer Perceptron*)

   Multilayer perceptron (MLP) merupakan ANN turunan dari *perceptron*, berupa ANN *feedforward* dengan satu atau lebih *hidden layer*. Biasanya, jaringan terdiri dari satu *layer* masukan, setidaknya satu *layer neuron* komputasi ditengah (*hidden layer*) dan sebuah *layer neuron* komputasi keluaran. Sinyal masuk dipropagasikan dengan arah maju pada *layer* per *layer* (Prasetyo, 2014 : 95).

3) ANN *Learning Vector Quantization*

   ANN *Learning Vector Quantization* (LVQ) merupakan salah satu jenis ANN yang berbasis *competitive learning* atau *winner take all*, di mana dari nilai keluaran yang diberikan *neuron* dalam *layer* keluaran hanya *neuron* pemenang (*neuron* yang mempunyai nilai terkecil) saja yang diperhatikan. *Neuron* yang menang tersebut akan mengalami pembaruan bobot.

Pembaruan bobot yang lain dilakukan pada *neuron* pemenang (karena mendapatkan nilai keluaran paling kecil dibanding yang lain) ini biasa menambah atau mengurangi (Prasetyo, 2014 : 112).

3. Klasifikasi *Support Vector Machine*

Pada SVM, hanya sejumlah data terpilih sajalah yang berkontribusi untuk melakukan model yang digunakan dalam klasifikasi yang akan dipelajari. SVM juga berada pada *Nearest Neighbor* yang menyimpan semua data latih untuk digunakan pada saat prediksi. SVM hanya menyimpan sebagian kecil saja dari data latih untuk digunakan pada saat prediksi. Hal inilah yang menjadi kelebihan SVM karena tidak semua data latih akan dipandang untuk dilibatkan dalam setiap iterasi pelatihannya. Data-data yang berkontribusi tersebut disebut *support vector* sehingga metodenya juga disebut *Support Vector Machine*. SVM terbagi menjadi tiga yaitu : SVM *Linear*, *Hyperplane SVM*, dan SVM *Nonlinear* (Prasetyo, 2014 : 123).

4. Klasifikasi Berbasis *Nearest Neighbor*

Nearest Neighbor (NN) menjadi salah satu metode dalam top 10 metode *data mining* yang paling popular digunakan. Metode NN murni termasuk dalam klasifikasi yang *lazy learner* karena menunda proses pelatihan (atau bahkan tidak melakukan sama sekali) sampai ada data uji yang ingin diketahui label kelasnya, maka metode baru akan menjalankan algoritmanya (Prasetyo, 2014 : 149).

Yang termasuk klasifikasi berbasis *nearest neighbor*:

1) K-*Nearest Neighbor*

   Metode K-*Nearest Neighbor* (K-NN) menjadi salah satu metode berbasis NN yang paling tua dan popular. Nilai K yang digunakan disini menyatakan jumlah tetangga terdekat yang dilibatkan dalam penentuan prediksi label kelas pada data uji. Dari K tetangga terdekat yang terpilih kemudian dilakukan *voting* kelas dari K tetangga terdekat tersebut. Kelas dengan jumlah suara tetangga terbanyaklah yang diberikan sebagai label kelas hasil prediksi pada data uji tersebut (Prasetyo, 2014 : 150).

2) *Support Vector K-Nearest Neighbor Classifier*

   Metode *Support Vector K-Nearest Neighbor Classifier* (SV-KNNC) diperkenalkan oleh Srisawat (2006) dengan konsep campuran antara K-NN, SVM, dan K-Means. Selain mereduksi jumlah data latih, SV-KNNC juga bertujuan untuk meningkatkan hasil prediksi. Ada tiga proses penting dalam SV-KNNC yaitu pemilihan data, pemberian bobot, dan prediksi (Prasetyo, 2014 : 161).

3) K-*Support Vector Nearest Neighbor*

   K-*Support Vector Nearest Neighbor* (K-SVNN) merupakan metode reduksi data latih yang didasarkan pada kedua metode sebelumnya, dengan prinsip K tetangga terdekat pada setiap data latih. Tidak ada proses *clustering* yang dilakukan pada sisi data latih yang dihasilkan dan jug belum ada pembobotan pada latih yang didapatkan sebagai *support vector* sehingga komputasi pada saat pelatihan menjadi lebih cepat. Dengan

berkurangnya data latih yang didapatkan untuk menjadi *support vector*, maka proses prediksi juga dihapkan menjadi lebih cepat dan akurat (Prasetyo, 2014 : 166).

4) Weight K-*Support Vector Nearest Neighbor*

Weight K-*Support Vector Nearest Neighbor* (WK-SVNN) menjadi pengembangan lebih lanjut pada K-SVNN dengan memasukkan bobot pada SV yang didapat, untuk digunakan pada saat prediksi. Pelatihan dalam WK-SVNN meliputi seleksi data latih yang nilai derajat signifikansinya lebih dari atau sama dengan ambang batas yang ditetapkan dan menambahkan bobot pada selama pelatihan tersebut. Selanjutnya diberikan skema proses prediksi yang baru dengan mawarisi algoritma prediksi K-NN klasik (Prasetyo, 2014 : 178).

II.1.4. Proses *Data Mining*

Secara sistematis, ada tiga langkah utama dalam *data mining* :

1. Eksplorasi/pemrosesan awal data

Eksplorasi/pemrosesan awal data terdiri data ‘pembersihan’ data, normalisasi data, transformasi data, penanganan data yang salah, reduksi dimensi, pemilihan subset fitur, dan sebagainya.

2. Membangun model dan melakukan validasi terhadapnya

Membangun model dan melakukan validasi terhadapnya berarti melakukan analisis berbagai model dan memilih model dengan kinerja prediksi yang terbaik. Dalam langkah ini digunakan metode-metode seperti klasifikasi,

regresi, analisis *cluster*, deteksi anomali, analisis asosiasi, analisis pola sekuensial, dan sebagainya. Dalam beberapa teferensi, deteksi anomal juga masuk dalam langkah eksplorasi. Akan tetapi, deteksi anomali juga digunakan sebagai algoritma utama, terutama untuk mencari data-data yang spesial.

3. Penerapan

Penerapan berarti menerapkan model pada data yang baru untuk menghasilkan perkira/prediksi masalah yang diinvestigasi (Prasetyo, 2014 : 7).

II.1.5. Set Data

Bukan *data mining* namanya jika tidak ada set data yang diolah didalamnya. Kata ‘data’ dalam terminologi statistik adalah kumpulan objek dengan atribut-atribut tertentu, dimana objek tersebut adalah individu berupa data dimana setiap data memilih sejumlah atribut. Atribut tersebut berpengaruh pada dimensi dari data, semakin banyak atribut/fitur maka semakin besar dimensi data. Kumpulan data-data membentuk set data. Dalam buku ini kadang menyebut data, kadang menyebut vektor, keduanya mempunyai maksud yang sama.

Berikut tiga jenis set data yang dikenal dan masing-masing penggolongannya:

1. *Record*
    a. Matriks data
    b. Data transaksi
    c. Data dokumen

2. *Graph*
    a. *Word Wide Web* (WWW)
    b. Struktur molekul
3. *Ordered data set*
    a. Data spasial
    b. Data temporal
    c. Data sekuensial
    d. Data urutan genetik (*genetic sequence*) (Prasetyo, 2014 : 7).

II.1.6. Pengelompokan *Data Mining*

Data mining dibagi menjadi beberapa kelompok berdasarkan tugas yang dapat dilakukan (Tampubolon, 2013 : 96), yaitu :

1. Deskripsi

Terkadang peneliti dan analis secara sederhana ingin mencoba mencari data untuk menggambarkan pola dan kecenderungan yang terdapat dalam data. Sebagai contoh, petugas pengumpulan suara mungkin tidak dapat menentukan keterangan atau fakta bahwa siapa yang tidak cukup professional akan sedikit didukung dalam pemilihan presiden. Deskripsi dari pola dan kecenderungan sering memberikan kemungkinan penjelesan untuk suatu pola atau kecenderungan (Tampubolon, 2013 : 96).

2. Estimasi

Estimasi hampir sama dengan klasifikasi, kecuali variabel target estimasi lebih kearah numerik dari pada kearah kategori. Model dibangun menggunakan

record lengkap yang menyediakan nilai dari variabel target sebagai prediksi. Selanjutnya, pada peninjauan berikutnya estimasi nilai dari variabel target dibuat berdasarkan nilai variabel predikasi. Sebagai contoh akan dilakukan 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 kasus baru lainnya (Tampubolon, 2013 : 97).

3. Prediksi

Prediksi hampir sama dengan klasifikasi dan estimasi, kecuali bahwa dalam predikasi nilai dari hasil akan ada dimasa mendatang. Contoh prediksi bisnis dan penelitian adalah:

a. Prediksi harga beras dalam tiga bulan yang akan datang.
b. Prediksi persentasi kenaikan kecelakaan lalu lintas tahun depan jika batas bawah kecepatan dinaikkan.

Beberapa metode dan teknik yang digunakan dalam klasifikasi dan estimasi dapat pula digunakan (untuk keadaan yang tepat) untuk prediksi (Tampubolon, 2013 : 97).

4. Klasifikasi

Dalam klasifikasi, terdapat target variabel kategori. Sebagai contoh, penggolongan pendapatan dapat dipisahkan dalam tiga kategori, yaitu pendapatan tinggi, pendapatan sedang, dan pendapatan rendah. Contoh lain klasifikasi dalam bisnis dan penelitian adalah:

a. Menentukan apakah suatu transaksi kartu kredit merupakan transaksi yang curang atau tidak.
b. Memperkirakan apakah suatu pengajuan hipotek oleh nasabah merupakan suatu kredit yang baik atau buruk.
c. Mendiagnosis penyakit seorang pasien untuk mendapatkan termasuk kategori penyakit apa.

5. Pengklusteran (*Clustering*)

Pengklusteran merupakan pengelompokan *record*, pengamatan, atau memperhatikan dan membentuk kelas objek-objek yang memiliki kemiripan. Kluster adalah kumpulan *record* yang memiliki kemiripan satu dengan yang lainnya dan memiliki ketidakmiripan dengan *record-record* dalam *cluster* lain. Pengklusteran berbeda dengan klasifikasi yaitu tidak adanya variabel target dalam pengklusteran. Pengklusteran tidak mencoba untuk melakukan klasifikasi, mengestimasi, atau memprediksi nilai dari variabel target.Akan tetapi, algoritma pengklusteran mencoba untuk melakukan pembagian terhadap keseluruhan data. menjadi kelompok-kelompok yang memiliki kemiripan (*homogeny*), yang mana kemiripan dalam satu kelompok akan bernilai maksimal, sedangkan kemiripan dengan *record* dalam kelompok lain akan bernilai minimal (Tampubolon, 2013). Contoh pengklusteran dalam bisnis dan penelitian adalah:

a. Mendapatkan kelompok-kelompok konsumen untuk target pemasaran dari satu suatu produk bagi perusahaan yang tidak memiliki dana pemesaran yang besar.

b. Untuk tujuan audit akuntansi, yaitu melakukan pemisahan terhadap perilaku *financial* dalam baik dan mencurigakan.
c. Melakukan pengklusteran terhadap ekspresi dari gen, untuk mendapatkan kemiripan perilaku dari gen dalam jumlah besar.

6. Asosiasi

Tugas asosiasi dalam *data mining* adalah menemukan *attribut* yang muncul dalam satu waktu. Dalam dunia bisnis lebih umum disebut analisis keranjang belanja (Tampubolon, 2013 : 97). Contoh asosiasi dalam bisnis dan penelitian adalah:

a. Meneliti jumlah pelanggan dari perusahaan telekomunikasi seluler yang diharapkan untuk memberikan respon positif terhadap penawaran *upgrade* layanan yang diberikan.
b. Menentukan barang dalam supermarket yang dibeli secara bersamaan dan yang tidak pernah dibeli secara bersamaan.

II.1.7. Algoritma C4.5

Algoritma C4.5 merupakan kelompok algoritma *decision tree*. Algoritma ini mempunyai *input* berupa *training sampl*es dan *samples. Training samples* berupa data contoh yang akan digunakan untuk membangun sebuah *tree* yang telah diuji kebenarannya. Sedangkan *samples* merupakan *field-field* data yang nantinya akan kita gunakan sebagai parameter dalam melakukan klasifikasi data (Sari dan Sindunata, 2014 : 12).

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

a. Pilih atribut sebagai akar

b. Buat cabang untuk masing – masing nilai

c. Bagi kasus dalam cabang

d. Ulangi proses untuk masing–masing cabang sampai semua kasus pada cabang memiliki kelas yang sama (Kamagi dan Hansun, 2014 : 16).

Adapun rumus algoritma C4.5 adalah sebagai berikut :

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

Keterangan :

S : Himpunan kasus

A : Atribut

n : Jumlah Partisi Atribut A

|Si| : Jumlah kasus pada partisi ke-i

|S| : Jumlah Kasus dalam S

Sementara itu, penghitungan nilai *entropy* dapat dilihat pada persamaan berikut :

$$Entropy\ (S) = \sum_{i=1}^{n} - pi * \log_2 pi \quad \text{.....(2)}$$

Keterangan :

S : Himpunan kasus

n : Jumlah partisi S

pi : Proporsi dari Si terhadap S (Kamagi dan Hansun, 2014 : 16).

II.2. Normalisasi

Normalisasi merupakan parameter digunakan untuk menghindari duplikasi terhadap tabel dalam basis data dan juga merupakan proses mendekomposisikan sebuah tabel yang masih memiliki beberapa anomali atau ketidakwajaran sehingga menghasilkan tabel yang lebih sederhana dan struktur yang bagus, yaitu sebuah table yang tidak memiliki *data redundancy* dan memungkinkan *user* untuk melakukan *insert, delete,* dan *update* pada baris (*record*) tanpa menyebabkan inkonsistensi data (Triyono, 2012 : 19).

1. *First Normal Form (1 NF)*

Sudah tidak ada *repeating group* yaitu pengulangan yang terjadi pada beberapa atribut atau kolom dalam sebuah tabel, dan juga setiap atribut harus bernilai tunggal. Atribut *multivaluedm composite, derive* tidak tunggal. Setiap nilai dari atribut hanya mempunyai nilai tunggal.

2. *Second Normal Form (2 NF)*

Untuk menjadikan tabel normal tingkat ke 2 maka sudah 1NF dan setiap atribut yang bukan *primary key* sepenuhnya secara *fungsional* tergantung pada semua atribut pembentuk *primary key*.

3. *Third Normal Form (3 NF)*

Tabel sudah 2NF dan tidak memiliki *transitive dependencies*.*Transitive dependency* adalah ketika ada atribut yang secara tidak langsung tergantung pada *primary key* dan atribut tersebut juga tergantung pada atribut lain yang bukan *primary key*.

4. *Boyce-codd Normal Form (BCNF)*

Tabel dalam BCNF jika sudah 3NF dan semua *determinants* adalah *candidate keys*. Perbedaan 3NF dan BCNF adalah untuk *functional dependency* A&B, 3NF memperbolehkan ketergantungan ada dalam relasi jika B adalah *Primary Key* dan A bukan merupakan *candidate key*. Sedangkan BCNF menuntut untuk ketergantungan tetap ada dalam relasi, A harus menjadi *candidate key*.

5. *Fourth Normal Form* (4 NF)

Relasi berada pada bentuk normal keempat apabila memenuhi syarat BCNF dan tidak mempunyai *multivalue dependency.*

6. *Fifth Normal Form* (5 NF)

Tabel bentuk normal kelima sering disebut PJNF (*Projection Join Normal Form*), penyebutan PJNF karena untuk suatu relasi akan berbentuk normal kelima jika tabel tersebut dapat dipecah atau diproyeksikan menjadi beberapa tabel dan dari proyeksi-proyeksi itu dapat disusun kembali (*join*) menjadi tabel yang sama dengan keadaan semula. Jika penyusunan ini tidak mungkin dilakukan dikatakan padarelasi itu terdapat *join dependencies* dan dikatakan bersifat *lossy join* (Triyono, 2012 : 20).

II.3. *Visual Basic* .NET

Visual Basic .NET adalah salah satu dari kumpulan *tools* pemrograman yang terdapat pada paket *Visual Studio* .NET. Pada *Visual Studio* .NET terdapat beberapa tools pemrograman lain seperti : Visual C++ .NET , Visual C# .NET dan Visual J# .NET. Lingkungan pengembangan VB .NET disebut dengan .NET *Framework Framework* ini menangani bagaimana *.NET programming* membangun tipe intrinsik, *class*, dan antarmuka (Hidayatullah, 2012 : 8).

II.3.1 Kelebihan *Visual Basic* .NET

Aplikasi-aplikasi pemrograman visual yang ada saat ini mempunyai kelebihan dan kelemahan masing-masing. Untuk suatu kasus, bisa jadi menggunakan *Delphi* lebih bagus, tapi untuk kasus yang lain bisa jadi aplikasi VB .NET yang lebih baik. Namun, VB .NET layak untuk dijadikan pilihan karena mempunyai cukup banyak kelebihan.

Beberapa kelebihan VB .NET antara lain :

1. Sederhana dan mudah dipahami

   Seperti pada VB, bahasa yang digunakan pada VB .NET sangat sederhana sehingga lebih mudah dipahami bagi mereka yang masih awam terhadap dunia pemrograman.

2. Mendukung GUI

   VB.NET bisa membuat *software* dengan antarmuka grafis yang lebih *user friendly.*

3. Menyederhanakan *deployment*

VB .NET mengatasi masalah *deployment* dari aplikasi berbasis *Windows* yaitu DLL *Hell* dan registrasi COM (*Component Object Model*). Selain itu tersedia *wizard* yang memudahkan dalam pembuatan *file setup*.

4. Menyederhanakan pengembangan perangkat lunak

Ketika terjadi kesalahan penulisan kode dari sisi *sintaks* (bahasa), maka VB .NET langsung menuliskan kesalahannya pada bagian *Message Windows* sehingga *Programmer* dapat memperbaiki kode dengan lebih cepat. Editor menu bersifat WYSIWYG (*What You See Is What You Get*). Adanya berbagai *Wizard* yang memadu *programmer* dalam membuat *software*. Tersedianya *Crystal Report* (CR) untuk membuat laporan (pada *Visual Studio 2010, Crystal Report* gratis namun harus diinstal secara terpisah). Adanya *Code Snippets* yaitu fitur untuk menyisipkan kode-kode koleksi kita pada program yang sedang kita buat.Di atas adalah hal–hal yang membuat pengembangan perangkat lunak menjadi lebih mudah.

5. Mendukung penuh OOP

Memiliki fitur bahasa pemrograman berorientasi objek seperti *inheritance* (pewarisan), *encapsulation* (pembungkusan), dan *olymorphism* (banyak bentuk).

6. Mempermudah pengembangan aplikasi berbasis Web

Disediakan desainer form Web. Selain itu disediakan layanan Web XML sehingga memungkinkan suatu aplikasi “berkomunikasi” dengan aplikasi lainnya dari berbagai *platform* menggunakan protokol internet terbuka.

7. Migrasi ke VB .NET dapat dilakukan dengan mudah

Jika anda sudah mengembangkan aplikasi di VB, maka konversi ke VB .NET dapat anda jalankan dengan mudah.

Banyak digunakan oleh *programmer–programmer* di seluruh dunia. Salah satu keuntungannya adalah jika kita memiliki masalah/pertanyaan, maka kita bisa tanyakan kepada *programmer–programmer* lain di seluruh dunia melalui forum-forum di internet (Hidayatullah, 2012 : 7).

II.4. SQL *Server* 2008

SQL *Server* 2008 adalah sebuah terobosan baru dari *Microsoft* dalam bidang *database*. SQL *Server* adalah DBMS(*Database Management System*) yang dibuatoleh *Microsoft* untuk ikut berkecimpung dalam persaingan dunia pengolahan data menyusul pendahulunya seperti IBM dan *Oracle*. SQL *Server* 2008 dibuat pada saat kemajuan dalam bidang *hardware* sedemikian pesat. Oleh karena itu sudah dapat dipastikan bahwa SQL *Server* 2008 membawa beberapaterobosan dalam bidang pengolahan dan penyimpanan data (Widya dan Zulkarnaen, 2012 : 3).

II.5. *Unified Modeling Language* (UML)

Menurut Windu Gata (2013) Hasil pemodelan pada OOAD terdokumentasikan dalam bentuk *Unified Modeling Language* (UML). UML adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak. UML merupakan

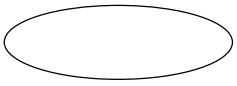
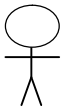
metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem. UML saat ini sangat banyak dipergunakan dalam dunia industri yang merupakan standar bahasa pemodelan umum dalam industri perangkat lunak dan pengembangan sistem (Urva dan Siregar, 2015, Hal : 93).

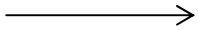
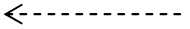
Alat bantu yang digunakan dalam perancangan berorientasi objek berbasiskan UML adalah sebagai berikut:

1. Use Case Diagram

Use case diagram merupakan pemodelan untuk kelakukan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Dapat dikatakan *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Simbol-simbol yang digunakan dalam *use case* diagram dapat dilihat pada tabel II.1 dibawah ini:

Tabel II.1. Simbol *Use Case Diagram*

Gambar	Keterangan
	*Use case* menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal nama *use case*.
	Aktor adalah *abstraction* dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasikan aktor, harus ditentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada konteks target sistem. Orang atau sistem bisa muncul dalam beberapa peran. Perlu dicatat bahwa aktor berinteraksi dengan *use case*, tetapi tidak memiliki control terhadap *use case*.

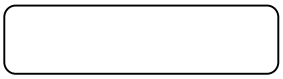
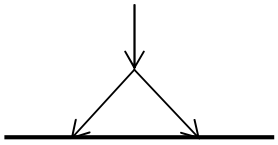
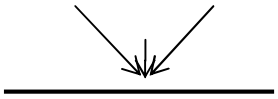
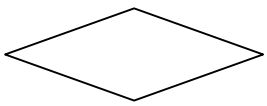
	Asosiasi antara aktor dan *use case*, digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengidikasikan aliran data.
	Asosiasi antara aktor dan *use case* yang menggunakan panah terbuka untuk mengidinkasikan bila aktor berinteraksi secara pasif dengan sistem.
	Include, merupakan di dalam *use case* lain (*required*) atau pemanggilan *use case* oleh *use case* lain, contohnya adalah pemanggilan sebuah fungsi program.
	*Extend*, merupakan perluasan dari *use case* lain jika kondisi atau syarat terpenuhi.

(Sumber : Urva dan Siregar; 2015, Hal : 94)

2. Diagram Aktivitas (*Activity Diagram*)

Activity diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Simbol-simbol yang digunakan dalam *activity diagram* dapat dilihat pada tabel II.2 dibawah ini:

Tabel II.2. Simbol *Activity Diagram*

Gambar	Keterangan
	*Start point*, diletakkan pada pojok kiri atas dan merupakan awal aktifitas.
	*End point*, akhir aktifitas.
	*Activites*, menggambarkan suatu proses/kegiatan bisnis.
	*Fork* (Percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara parallel atau untuk menggabungkan dua kegiatan pararel menjadi satu.
	*Join* (penggabungan) atau rake, digunakan untuk menunjukkan adanya dekomposisi.
	*Decision Points*, menggambarkan pilihan untuk pengambilan keputusan, *true, false.*

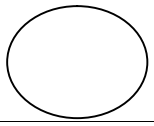
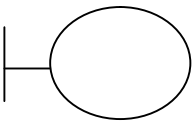
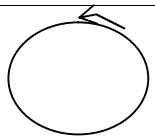
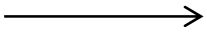
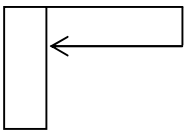
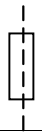
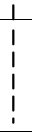
New Swimline	*Swimlane*, pembagian *activity* diagram untuk menunjukkan siapa melakukan apa.

(Sumber : Urva dan Siregar; 2015, Hal : 94)

3. Diagram Urutan (*Sequence Diagram*)

Sequence diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek. Simbol-simbol yang digunakan dalam *sequence* diagram dapat dilihat pada tabel II.3 dibawah ini :

Tabel II.3. Simbol *Sequence Diagram*

Gambar	Keterangan
	*EntityClass*, merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data.
	*Boundary Class*, berisi kumpulan kelas yang menjadi *interface* atau interaksi antara satu atau lebih aktor dengan sistem, seperti tampilan formentry dan *form* cetak.
	*Control class*, suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas, contohnya adalah kalkulasi dan aturan bisnis yang melibatkan berbagai objek.
	*Message*, simbol mengirim pesan antar *class*.
	*Recursive*, menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri.
	*Activation*, *activation* mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivitas sebuah operasi.
	*Lifeline*, garis titik-titik yang terhubung dengan objek, sepanjang *lifeline* terdapat *activation*.

(Sumber : Urva dan Siregar; 2015, Hal : 95)

4. *Class Diagram* (Diagram Kelas)

Class diagram merupakan hubungan antar kelas dan penjelasan detail tiap-tiap kelas di dalam model desain dari suatu sistem, juga memperlihatkan aturan-aturan dan tanggng jawab entitas yang menentukan perilaku sistem. *Class diagram* juga menunjukkan atribut-atribut dan operasi-operasi dari sebuah kelas dan *constraint* yang berhubungan dengan objek yang dikoneksikan. *Class diagram* secara khas meliputi : Kelas (*Class*), Relasi, *Associations*, *Generalization* dan *Aggregation*, Atribut (*Attributes*), Operasi (*Operations/ Method*), *Visibility*, tingkat akses objek eksternal kepada suatu operasi atau atribut. Hubungan antar kelas mempunyai keterangan yang disebut dengan *multiplicity* atau kardinaliti yang dapat dilihat pada tabel II.4 dibawah ini:

Tabel II.4. Simbol *Multiplicity Class Diagram*

Multiplicity	Penjelasan
1	Satu dan hanya satu
0..*	Boleh tidak ada atau 1 atau lebih
1..*	1 atau lebih
0..1	Boleh tidak ada, maksimal 1
n..n	Batasan antara. Contoh 2..4 mempunyai arti minimal 2 maksimum 4

(Sumber : Urva dan Siregar; 2015, Hal : 95)