

BAB II

LANDASAN TEORI

2.1 SEKOLAH MENENGAH ATAS

Sekolah menengah atas adalah salah satu bentuk pendidikan formal yang menyelenggarakan pendidikan umum pada jenjang pendidikan menengah sebagai lanjutan dari SMP, MTs, atau bentuk lain yang sederajat (pasal 1 poin 11 RPP DIKDASMEN). Sebagai suatu instansi pendidikan menengah, SMA memiliki fungsi dan tujuan khusus seperti yang tercantum pada pasal 47 dan 48 RPP DIKDASMEN. Fungsi dari pendidikan menengah adalah menegembangkan nilai-nilai dan sikap rasa keindahan dan harmoni, pengetahuan, kemampuan, dan ketrampilan sebagai persiapan untuk melanjutkan ke pendidikan tinggi dan/atau untuk hidup di masyarakat dalam rangka mencapai tujuan pendidikan nasional. Sedangkan tujuan pendidikan menengah adalah untuk meningkatkan keimanan dan ketakwaan, hidup sehat, memperluas pengetahuan dan seni, memiliki keahlian dan ketrampilan, menjadi anggota masyarakat yang bertanggung jawab, serta mempersiapkan peserta didik untuk mengikuti pendidikan lebih lanjut.

2.2 DATA MINING

Data mining berisi pencarian trend atau pola yang diinginkan dalam database yang besar untuk membantu pengambil keputusan di waktu yang akan datang. KDD (*Knowledge Discovery in Database*) adalah keseluruhan proses konversi data mentah menjadi pengetahuan yang bermanfaat yang terdiri dari serangkaian tahap transformasi meliputi data *preprocessing* dan *postprocessing*. Adapun beberapa pengertian data mining yang diambil dari beberapa pendapat yaitu sebagai berikut :

1. Data Mining adalah suatu istilah yang digunakan untuk menguraikan penemuan pengetahuan di dalam database yang prosesnya menggunakan tehnik statistik, matematika, kecerdasan buatan, dan machine learning, untuk mengekstraksi dan mengidentifikasi informasi yang bermanfaat dan

pengetahuan yang terkait dari berbagai database yang besar (Turban, dkk. 2005).

2. Data mining juga disebut sebagai serangkaian proses untuk menggali nilai tambah berupa pengetahuan yang selama ini tidak diketahui secara manual dari suatu kumpulan data (Pramudiono, 2007).
3. Data mining, sering juga disebut sebagai *knowledge discovery in database* (KDD). KDD adalah kegiatan yang meliputi pengumpulan, pemakaian data, historis untuk menemukan keteraturan, pola atau hubungan dalam set data berukuran besar (Santoso, 2007).

Berdasarkan beberapa pengertian di atas dapat disimpulkan bahwa data mining adalah suatu algoritma untuk menggali informasi berharga yang terpendam atau tersembunyi di dalam suatu koleksi data (*database*) yang sangat besar sehingga ditemukan suatu pola yang menarik yang sebelumnya tidak diketahui.

2.3 TAHAPAN DATA MINING

Sebagai suatu rangkaian proses, data mining dapat dibagi menjadi beberapa tahapan, diantaranya (Han, J. and Kamber, M , 2006) :

1. Pembersihan data (*data cleaning*)

Pembersihan data merupakan proses menghilangkan noise dan data yang tidak konsisten atau data tidak relevan. Pada umumnya data yang diperoleh, baik dari database suatu perusahaan maupun hasil eksperimen, memiliki isian-isian yang tidak sempurna seperti data yang hilang, data yang tidak valid atau juga hanya sekedar salah ketik. Selain itu, ada juga atribut-atribut data yang tidak relevan dengan hipotesa data mining yang dimiliki. Data-data yang tidak relevan itu juga lebih baik dibuang. Pembersihan data juga akan mempengaruhi performansi dari teknik data mining karena data yang ditangani akan berkurang jumlah dan kompleksitasnya.

2. Integrasi data (*data integration*)

Integrasi data merupakan penggabungan data dari berbagai *database* ke dalam satu *database* baru. Tidak jarang data yang diperlukan untuk data mining tidak hanya berasal dari satu *database* tetapi juga berasal dari beberapa *database* atau file teks. Integrasi data dilakukan pada atribut-atribut yang mengidentifikasi entitas-entitas

yang unik seperti atribut nama, jenis produk, nomor pelanggan dan lainnya. Integrasi data perlu dilakukan secara cermat karena kesalahan pada integrasi data bisa menghasilkan hasil yang menyimpang dan bahkan menyesatkan pengambilan aksi nantinya. Sebagai contoh bila integrasi data berdasarkan jenis produk ternyata menggabungkan produk dari kategori yang berbeda maka akan didapatkan korelasi antar produk yang sebenarnya tidak ada.

3. Seleksi Data (*Data Selection*)

Data yang ada pada *database* sering kali tidak semuanya dipakai, oleh karena itu hanya data yang sesuai untuk dianalisis yang akan diambil dari *database*. Sebagai contoh, sebuah kasus yang meneliti faktor kecenderungan orang membeli dalam kasus market basket analysis, tidak perlu mengambil nama pelanggan, cukup dengan id pelanggan saja.

4. Transformasi data (*Data Transformation*)

Data diubah atau digabung ke dalam format yang sesuai untuk diproses dalam data mining. Beberapa metode data mining membutuhkan format data yang khusus sebelum bisa diaplikasikan. Sebagai contoh beberapa metode standar seperti analisis *asosiasi* dan *clustering* hanya bisa menerima input data kategorikal. Karenanya data berupa angka numerik yang berlanjut perlu dibagi-bagi menjadi beberapa *interval*. Proses ini sering disebut transformasi data.

5. Proses mining

Merupakan suatu proses utama saat metode diterapkan untuk menemukan pengetahuan berharga dan tersembunyi dari data.

6. Evaluasi pola (*pattern evaluation*)

Untuk mengidentifikasi pola-pola menarik kedalam *knowledge based* yang ditemukan. Dalam tahap ini hasil dari teknik data mining berupa pola-pola yang khas maupun model prediksi dievaluasi untuk menilai apakah *hipotesa* yang ada memang tercapai. Bila ternyata hasil yang diperoleh tidak sesuai hipotesa ada beberapa alternatif yang dapat diambil seperti menjadikannya umpan balik untuk memperbaiki proses data mining, mencoba metode data mining lain yang lebih sesuai, atau menerima hasil ini sebagai suatu hasil yang di luar dugaan yang mungkin bermanfaat.

7. Presentasi pengetahuan (knowledge presentation)

Merupakan *visualisasi* dan penyajian pengetahuan mengenai metode yang digunakan untuk memperoleh pengetahuan yang diperoleh pengguna. Tahap terakhir dari proses data mining adalah bagaimana memformulasikan keputusan atau aksi dari hasil analisis yang didapat. Ada kalanya hal ini harus melibatkan orang-orang yang tidak memahami data mining. Karenanya presentasi hasil data mining dalam bentuk pengetahuan yang bisa dipahami semua orang adalah satu tahapan yang diperlukan dalam proses data mining. Dalam presentasi ini, *visualisasi* juga bisa membantu mengkomunikasikan hasil data mining (Han, 2006).

2.4 PENGELOMPOKAN DATA MINING

Data mining dibagi menjadi beberapa kelompok berdasarkan tugas yang dapat di lakukan, yaitu (Larose, 2006).

1. Deskripsi

Terkadang peneliti dan analisis secara sederhana ingin mencoba mencari cara untuk menggambarkan pola dan kecenderungan yang terdapat dalam data. Sebagai contoh, petugas pengumpulan suara mungkin tidak dapat menemukan keterangan atau fakta bahwa siapa yang tidak cukup profesional akan sedikit didukung dalam pemilihan presiden. Deskripsi dari pola dan kecenderungan sering memberikan kemungkinan penjelasan untuk suatu pola atau kecenderungan.

2. Estimasi

Estimasi hampir sama dengan klasifikasi, kecuali variabel target estimasi lebih ke arah numerik dari pada 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 variabel prediksi. Sebagai contoh, akan dilakukan estimasi tekanan darah sistolik pada pasien rumah sakit berdasarkan umur pasien, jenis kelamin, 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.

3. Prediksi

Prediksi hampir sama dengan klasifikasi dan estimasi, kecuali bahwa dalam prediksi nilai dari hasil akan ada di masa mendatang.

Contoh prediksi dalam bisnis dan penelitian adalah:

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

Beberapa metode dan teknik yang digunakan dalam klasifikasi dan estimasi dapat pula digunakan (untuk keadaan yang tepat) untuk prediksi.

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 bukan.
- b. Memperkirakan apakah suatu pengajuan hipotek oleh nasabah merupakan suatu kredit yang baik atau buruk.
- c. Mendiagnosa penyakit seorang pasien untuk mendapatkan termasuk kategori apa.

5. Pengklusteran

Pengklusteran merupakan pengelompokan *record*, pengamatan, atau memperhatikan dan membentuk kelas objek-objek yang memiliki kemiripan. *Kluster* adalah kumpulan *record* yang memiliki kemiripan suatu dengan yang lainnya dan memiliki ketidakmiripan dengan *record* dalam *kluster* 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 (*homogen*), yang mana kemiripan dengan *record* dalam kelompok lain akan bernilai minimal.

Contoh pengklusteran dalam bisnis dan penelitian adalah:

- a. Mendapatkan kelompok-kelompok konsumen untuk target pemasaran dari suatu produk bagi perusahaan yang tidak memiliki dana pemasaran yang besar.
- b. Untuk tujuan audit akutansi, yaitu melakukan pemisahan terhadap perilaku finansial dalam baik dan mencurigakan.
- c. Melakukan pengklusteran terhadap ekspresi dari gen, dalam jumlah besar.

6. Asosiasi

Tugas *asosiasi* dalam data mining adalah menemukan atribut yang muncul dalam suatu waktu. Dalam dunia bisnis lebih umum disebut analisis keranjang belanja.

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. Menemukan barang dalam supermarket yang dibeli secara bersamaan dan barang yang tidak pernah dibeli bersamaan.

2.5 KLASIFIKASI

Klasifikasi adalah salah satu masalah mendasar dan tugas utama dalam data mining, dalam klasifikasi sebuah pengklasifikasi dibuat dari sekumpulan data latih dengan kelas yang telah di tentukan sebelumnya. Performa pengklasifikasi biasanya diukur dengan ketepatan atau tingkat galat. Pengklasifikasi Bayesian merupakan pengklasifikasi *statistik* untuk dapat memprediksi *probabilitas* keanggotaan kelas tertentu, menghitung peluang untuk suatu *hipotesis*, menghitung peluang dari suatu kelas dari masing-masing kelompok atribut yang ada, dan menentukan kelas mana yang paling optimal.

2.6 ALGORITMA NAIVE BAYES

Naive Bayes merupakan sebuah pengklasifikasian probabilistik sederhana yang menghitung sekumpulan *probabilitas* dengan menjumlahkan *frekuensi* dan kombinasi nilai dari dataset yang diberikan. Naive Bayes, penelitian tersebut menyimpulkan bahwa metode algoritma Naive Bayes memiliki keunggulan untuk pengembangan DM, yaitu kemudahan konstruksinya dan tidak membutuhkan parameter skema pengulangan yang

kompleks sehingga mudah dalam membaca data dalam jumlah yang besar. Hal ini terjadi karena desain rancangan penuntunan klasifikasi terhadap data. Selain itu, metode ini dinyatakan sebagai algoritma yang mempunyai sifat *simplicity*, *elegance* dan *robustness*. Algoritma menggunakan teorema Bayes dan mengasumsikan semua atribut independen atau tidak saling ketergantungan yang diberikan oleh nilai pada variabel kelas. Definisi lain mengatakan Naive Bayes merupakan pengklasifikasian dengan metode *probabilitas* dan *statistik* yang dikemukakan oleh ilmuwan Inggris Thomas Bayes, yaitu memprediksi peluang di masa depan berdasarkan pengalaman di masa sebelumnya. Naive Bayes didasarkan pada asumsi penyederhanaan bahwa nilai atribut secara kondisional saling bebas jika diberikan nilai *output*. Dengan kata lain, diberikan nilai *output*, *probabilitas* mengamati secara bersama adalah produk dari *probabilitas* individu. Keuntungan penggunaan Naive Bayes adalah bahwa metode ini hanya membutuhkan jumlah data pelatihan (*Training Data*) yang kecil untuk menentukan estimasi parameter yang diperlukan dalam proses pengklasifikasian. Naive Bayes sering bekerja jauh lebih baik dalam kebanyakan situasi dunia nyata yang kompleks dari pada yang diharapkan.

Persamaan Metode Naive Bayes

Persamaan dari teorema *Bayes* adalah:

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

Di mana :

X : Data dengan *class* yang belum diketahui

H : *Hipotesis* data merupakan suatu *class* spesifik

$P(H|X)$: *Probabilitas hipotesis H* berdasar kondisi X (*posteriori probabilitas*)

$P(H)$: *Probabilitas hipotesis H* (*prior probabilitas*)

$P(X|H)$: *Probabilitas X* berdasarkan kondisi pada *hipotesis H*

$P(X)$: *Probabilitas X*

Untuk menjelaskan metode *Naive Bayes*, perlu diketahui bahwa proses klasifikasi memerlukan sejumlah petunjuk untuk menentukan kelas apa yang cocok bagi sampel yang dianalisis tersebut. Karena itu, metode *Naive Bayes* di atas disesuaikan sebagai berikut:

$$P(C|F_1 \dots F_n) = \frac{P(C)P(F_1 \dots F_n|C)}{P(F_1 \dots F_n)}$$

merepresentasikan karakteristik petunjuk yang dibutuhkan untuk melakukan klasifikasi. Maka rumus tersebut menjelaskan bahwa peluang masuknya sampel karakteristik tertentu dalam kelas C (*Posterior*) adalah peluang munculnya kelas C (sebelum masuknya sampel tersebut, seringkali disebut *prior*), dikali dengan peluang kemunculan karakteristik-karakteristik sampel pada kelas C (disebut juga *likelihood*), dibagi dengan peluang kemunculan karakteristik-karakteristik sampel secara global (disebut juga *evidence*). Karena itu, rumus di atas dapat pula ditulis secara sederhana sebagai berikut:

$$\text{Posterior} = \frac{\text{prior} \times \text{likelihood}}{\text{evidence}}$$

Nilai *Evidence* selalu tetap untuk setiap kelas pada satu sampel. Nilai dari *posterior* tersebut nantinya akan dibandingkan dengan nilai-nilai *posterior* kelas lainnya untuk menentukan ke kelas apa suatu sampel akan diklasifikasikan. Penjabaran lebih lanjut rumus *Bayes* tersebut dilakukan dengan menjabarkan $(C|F_1, \dots, F_n)$ menggunakan aturan perkalian sebagai berikut:

$$\begin{aligned} P(C|F_1, \dots, F_n) &= P(C)P(F_1, \dots, F_n|C) \\ &= P(C)P(F_1|C)P(F_2, \dots, F_n|C, F_1) \\ &= P(C)P(F_1|C)P(F_2|C, F_1)P(F_3, \dots, F_n|C, F_1, F_2) \\ &= P(C)P(F_1|C)P(F_2|C, F_1)P(F_3|C, F_1, F_2)P(F_4, \dots, F_n|C, F_1, F_2, F_3) \\ &= P(C)P(F_1|C)P(F_2|C, F_1)P(F_3|C, F_1, F_2) \dots P(F_n|C, F_1, F_2, F_3, \dots, F_{n-1}) \end{aligned}$$

Dapat dilihat bahwa hasil penjabaran tersebut menyebabkan semakin banyak dan semakin kompleksnya faktor - faktor syarat yang mempengaruhi nilai probabilitas, yang hampir mustahil untuk dianalisa satu persatu. Akibatnya, perhitungan tersebut menjadi sulit untuk dilakukan. Di sinilah digunakan asumsi independensi yang sangat tinggi (*naif*), bahwa masing-masing petunjuk ($F_1, F_2 \dots F_n$) saling bebas (*independen*) satu sama lain. Dengan asumsi tersebut, maka berlaku suatu kesamaan sebagai berikut:

$$P(F_i|F_j) = \frac{P(F_i \cap F_j)}{P(F_j)} = \frac{P(F_i)P(F_j)}{P(F_j)} = P(F_i)$$

Untuk $i \neq j$, sehingga

$$P(F_i|C, F_j) = P(F_i|C)$$

Persamaan di atas merupakan model dari teorema *Naive Bayes* yang selanjutnya akan digunakan dalam proses klasifikasi. Untuk klasifikasi dengan data kontinyu digunakan rumus

Densitas Gauss : $P(X_i=x_i|Y=y_j) = \frac{1}{\sqrt{2\pi}\sigma_{ij}} e^{-\frac{(x_i-\mu_{ij})^2}{2\sigma_{ij}^2}}$ (7)

$$P(F_i|F_j) = \frac{1}{\sqrt{2\pi}\sigma_{ij}} e^{-\frac{(x_i-\mu_{ij})^2}{2\sigma_{ij}^2}}$$

Di mana :

P : Peluang

X_i : Atribut ke i

x_i : Nilai atribut ke i

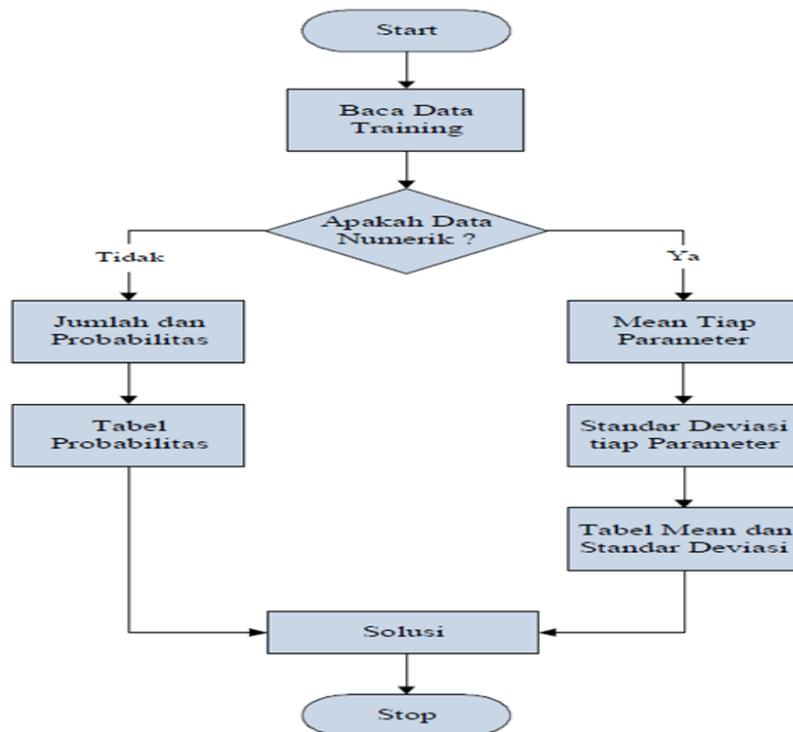
Y : Kelas yang dicari

y_i : Sub kelas Y yang dicari

μ : *mean*, menyatakan rata – rata dari seluruh atribut

σ : *Deviiasi standar*, menyatakan varian dari seluruh atribut.

Alur dari metode *Naive Bayes* dapat dilihat sebagai berikut:



(Alur Metode Naive Bayes)

Adapun keterangan dari gambar 2 di atas sebagai berikut:

1. Baca *data training*
2. Hitung Jumlah dan probabilitas, namun apabila data numerik maka:
 - a. Cari nilai *mean* dan standar deviasi dari masing-masing parameter yang merupakan data numerik.

Adapun persamaan yang digunakan untuk menghitung nilai rata – rata hitung (*mean*) dapat dilihat sebagai berikut :

$$\mu = \frac{\sum_{i=1}^n Xi}{n}$$

atau

$$\mu = \frac{x1 + x2 + x3 + \dots + xn}{n}$$

di mana :

μ : rata – rata hitung (*mean*)

xi : nilai sample ke - i

n : jumlah sampel

Dan persamaan untuk menghitung nilai simpangan baku (*standar deviasi*) dapat dilihat sebagai berikut:

$$\sigma = \sqrt{\frac{\sum_{i=1}^n (Xi - \mu)^2}{n - 1}}$$

di mana :

σ : standar *deviasi*

xi : nilai x ke - i

μ : rata-rata hitung

n : jumlah sampel

- b. Cari nilai *probabilistik* dengan cara menghitung jumlah data yang sesuai dari kategori yang sama dibagi dengan jumlah data pada kategori tersebut.
3. Mendapatkan nilai dalam tabel *mean*, *standard deviasi* dan *probabilitas*.
4. Solusi kemudian dihasilkan.

2.7 SISTEM

Sistem adalah jaringan dari pada elemen-elemen yang saling berhubungan, membentuk satu kesatuan untuk melaksanakan suatu tujuan pokok dari sistem tersebut (Jogiyanto HM, 2005).

2.8 INFORMASI

Informasi adalah data yang diolah menjadi satu bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya (Jogiyanto HM, 2005).

2.9 SISTEM INFORMASI

Sistem informasi adalah suatu sistem di dalam organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan (Jogiyanto HM, 2005).

2.10 ELEMEN SISTEM

Elemen Sistem adalah suatu sistem terdiri dari sejumlah elemen yang saling berinteraksi, yang artinya saling kerjasama membentuk satu kesatuan. Pendekatan suatu sistem yang merupakan suatu jaringan prosedur lebih menekankan pada urutan-urutan operasi di dalam sistem, sedangkan pendekatan yang menekankan pada elemen-elemen atau komponen merupakan interaksi antar elemen atau komponen atau mencapai sasaran atau tujuan sistem. (Jogiyanto HM, 2005).

2.11 KARAKTERISTIK SISTEM

Menurut kusrini (2007:6), untuk mencapai tujuan, suatu sistem harus memiliki sifat-sifat tertentu atau suatu karakteristik seperti berikut :

1. Komponen Sistem

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi dan bekerjasama membentuk suatu komponen sistem atau bagianbagian dari sistem.

2. Batasan Sistem

Merupakan daerah yang membatasi suatu sistem dengan sistem lainnya atau dengan lingkungan kerjanya.

3. SubSistem

Bagian dari sistem yang beraktifitas dan berinteraksi satu sama lain untuk mencapai suatu tujuan dengan sasarannya masing-masing.

4. Lingkungan Luar Sistem

Suatu sistem yang ada diluar lingkungan dari batas sistem yang di pengaruhi oleh operasi sistem.

5. Penghubung Sistem

Media penghubung ini antara suatu subsistem dengan subsistem lainnya. Adanya penghubung ini memungkinkan berbagai sumberdaya mengalir dari suatu subsistem ke subsistem lainnya.

6. Masukan Sistem (*Input*)

Energi yang masuk kedalam sistem, berupa perawatan dan sinyal. Masukan perawatan adalah energi yang dimasukkan suatu sistem tersebut dapat berinteraksi.

7. Keluaran Sistem (*Output*)

Hasil energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna dan sisa pembungan.

8. Pengolahan Sistem (*Proses*)

Suatu sistem dapat mempunyai suatu bagian pengolahan yang akan mengubah masukan menjadi keluaran.

9. Sasaran Sistem (*Object*)

Tujuan yang ingin dicapai oleh sistem, akan dikatakan berhasil apabila mengenai sasaran dan tujuan.

2.12 METODE PENDEKATAN SISTEM

Metode pendekatan yang digunakan adalah pendekatan perancangan berorientasi objek. Menurut Rosa dan M.Shalahuddin (2011 : 82) yang dimaksud dengan Metodologi berorientasi objek adalah suatu strategi pembangunan perangkat lunak yang mengorganisasikan perangkat lunak sebagai kumpulan objek yang berisi data dan operasi yang diberlakukan terhadapnya.

Metodologi ini merupakan suatu cara bagaimana sistem perangkat lunak dibangun melalui pendekatan objek secara *sistematis*. Metode ini didasarkan pada penerapan prinsip-prinsip pengelolaan kompleksitas. Metode ini meliputi rangkaian aktifitas analisis berorientasi objek, perancangan berorientasi objek, pemrograman berorientasi objek, dan pengujian berorientasi objek.

2.13 METODE PENGEMBANGAN SISTEM

Perancangan sistem data mining klasifikasi dengan metode algoritma naive bayes untuk prediksi siswa diterima di perguruan tinggi negeri dirancang dengan menggunakan Model *Waterfall*. Metode ini membutuhkan pendekatan yang *sistematis* dan *sekuensial*. Tahapan dalam metode ini diantaranya sebagai berikut :

1. Analisa Kebutuhan

Langkah ini merupakan analisa terhadap kebutuhan sistem. Pengumpulan data dalam tahap ini bisa melakukan sebuah penelitian, wawancara atau studi literatur. Sistem analis akan menggali informasi sebanyak-banyaknya dari *user* sehingga akan tercipta sebuah sistem komputer yang bisa melakukan tugas-tugas yang diinginkan oleh *user* tersebut. Tahapan ini akan menghasilkan dokumen *user requirement* atau bisa dikatakan sebagai data yang berhubungan dengan keinginan *user* dalam pembuatan sistem. Dokumen ini lah yang akan menjadi acuan sistem analis untuk menerjemahkan ke dalam bahasa pemrogram.

2. Desain Sistem

Tahapan dimana dilakukan penuangan pikiran dan perancangan sistem terhadap solusi dari permasalahan yang ada dengan menggunakan perangkat pemodelan sistem seperti diagram alir data (*data flow diagram*), diagram hubungan entitas (*entity relationship diagram*) serta struktur dan bahasan data.

3. Pengkodean

Pengkodean merupakan penerjemahan *design* dalam bahasa yang bisa dikenali oleh komputer. Dilakukan oleh *programmer* yang akan menerjemahkan transaksi yang diminta oleh *user*. Tahapan ini lah yang merupakan tahapan secara nyata dalam mengerjakan suatu sistem. Dalam artian penggunaan komputer akan dimaksimalkan dalam tahapan ini. Setelah pengkodean selesai maka akan dilakukan *testing* terhadap sistem yang telah dibuat tadi. Tujuan *testing* adalah

menemukan kesalahan-kesalahan terhadap sistem tersebut dan kemudian bisa diperbaiki.

4. Implementasi

Dalam tahap ini dilakukan pemrograman. Pembuatan *software* dipecah menjadi modul-modul kecil yang nantinya akan digabungkan dalam tahap berikutnya. Selain itu dalam tahap ini juga dilakukan pemeriksaan terhadap modul yang dibuat, apakah sudah memenuhi fungsi yang diinginkan atau belum.

5. Pengujian Program

Tahapan akhir dimana sistem yang baru diuji kemampuan dan keefektifannya sehingga didapatkan kekurangan dan kelemahan sistem yang kemudian dilakukan pengkajian ulang dan perbaikan terhadap aplikasi menjadi lebih baik dan sempurna.

6. *Maintenance* / Pemeliharaan

Perangkat lunak yang sudah disampaikan kepada pelanggan pasti akan mengalami perubahan. Perubahan tersebut bisa karena mengalami kesalahan karena perangkat lunak harus menyesuaikan dengan lingkungan (peripheral atau sistem operasi baru), atau karena pelanggan membutuhkan perkembangan fungsional.

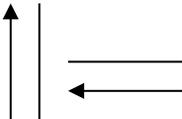
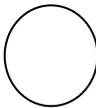
2.14 ALAT BANTU ANALISIS

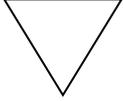
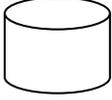
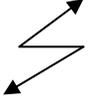
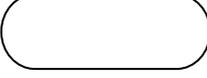
Adapun alat dan teknik yang digunakan pada penelitian ini yaitu bagan alir dokumen (*document flowchart*) yang digunakan untuk menggambarkan arus data sistem yang berjalan, *Unified Modeling Language (UML)* yang digunakan untuk menggambarkan model dari sistem yang diusulkan, *hierarchy plus input process output (HIPO)* yang digunakan sebagai alat desain dan teknik dokumentasi dalam siklus hidup pengembangan sistem, normalisasi sebagai salah satu teknik yang digunakan untuk mendesain *database*, kamus data sebagai alat dokumentasi untuk mendefinisikan struktur data dari *file-file database* dan bagan alir program (*program flowchart*) seperti logika program yang digunakan untuk menggambarkan tiap-tiap langkah di dalam program komputer secara logika.

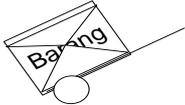
2.14.1 BAGAN ALIR DOKUMEN

Bagan alir dokumen (*document flowchart*) atau disebut juga bagan alir formulir (*form flowchart*) adalah bagan alir yang menunjukkan arus dari laporan dan formulir termasuk tembusan-tembusannya. Berikut ini merupakan simbol-simbol yang terdapat pada bagan alir dokumen

Tabel 2.1 Simbol Bagan Alir Dokumen

Simbol	Keterangan
	<p>Dokumen</p> <p>Simbol ini digunakan untuk menggambarkan semua jenis dokumen yang merupakan formulir atau digunakan untuk merekam data terjadinya suatu transaksi.</p>
	<p>Garis Alir</p> <p>Simbol ini menggambarkan arah proses pengolahan data. Anak panah tidak digambarkan jika arus dokumen mengarah ke bawah dan ke kanan, jika arus dokumen mengalir ke atas atau ke kiri anak panah harus digambarkan.</p>
	<p>Kegiatan Manual</p> <p>Digunakan untuk menggambarkan kegiatan manual.</p>
	<p>Penghubung 1 halaman</p> <p>Arus dokumen digambarkan dari atas kebawah dan dari kiri ke kanan, karena keterbatasan ruang halaman kertas dalam menggambar maka diperlukan simbol penghubung yang memungkinkan aliran dokumen berhenti disuatu lokasi pada halaman tertentu dan kembali berjalan dilokasi lain pada halaman yang sama.</p>
	<p>Penghubung beda halaman</p> <p>Simbol ini digunakan untuk menunjukkan kemana dan bagaimana bagan alir terkait antara yang satu dan yang lainnya.</p>

	<p>Keterangan</p> <p>Simbol ini digunakan untuk menambahkan keterangan agar memperjelas pesan yang disimpan dalam bagan alir.</p>
	<p>Arsip</p> <p>Simbol ini digunakan untuk tempat penyimpanan dokumen yang dokumennya akan diambil kembali dari arsip tersebut dimasa yang akan datang untuk keperluan lebih lanjut terhadap dokumen tersebut.</p>
	<p>Disket</p> <p>Simbol ini digunakan sebagai tempat/media penyimpanan yang menggunakan disket (proses komputerisasi).</p>
	<p>Harddisk</p> <p>Simbol ini digunakan sebagai tempat/media penyimpanan yang menggunakan harddisk (proses komputerisasi).</p>
	<p>Proses Komputerisasi</p> <p>Simbol ini digunakan untuk melakukan proses dengan komputer.</p>
	<p>Display Monitor</p> <p>Simbol ini digunakan untuk menampilkan <i>output</i> melalui komputer.</p>
	<p>Channel Transmisi</p> <p>Simbol ini menerangkan transmisi data dengan jaringan komputer.</p>
	<p>Key In</p> <p>Simbol ini menggambarkan peng-<i>input</i>-an dengan <i>keyboard</i>.</p>
	<p>Catatan Akuntansi</p> <p>Simbol ini menggambarkan catatan yang berhubungan dengan kegiatan akuntansi.</p>
	<p>Terminasi</p> <p>Simbol ini digunakan untuk menunjukkan awal dan akhir suatu proses</p>

	<p>Barang</p> <p>Simbol ini digunakan untuk menggambarkan barang</p>
	<p>Uang</p> <p>Simbol ini digunakan untuk menggambarkan uang</p>

2.14.2 UNIFIED MODELING LANGUAGE (UML)

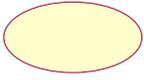
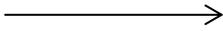
Alat bantu analisis dalam penelitian ini menggunakan UML dengan tools dalam pemodelan sistem diantaranya *use case diagram*, *class diagram* dan *activity diagram*. Berikut penjelasan dari 3 diagram tersebut :

1. Use Case Diagram

Menurut Rosa dan M.Shalahuddin (2011 : 130) yang dimaksud dengan *use case* adalah pemodelan untuk kelakuan sistem informasi yang akan dibuat. Kesimpulannya *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi itu.

Simbol-simbol yang digunakan pada *Use Case Diagram* adalah sebagai berikut:

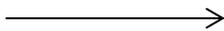
Tabel 2.2 Daftar simbol *Use Case Diagram*

No	Simbol	Keterangan
1		<p>Aktor</p> <p>Menunjukkan <i>user</i> yang akan menggunakan sistem baru</p>
2		<p><i>Use Case</i></p> <p>Menunjukkan proses yang terjadi pada sistem yang baru</p>
3		<p><i>Unidirectional Association</i></p> <p>Menunjukkan hubungan antara actor dengan <i>use case</i> atau antara <i>use case</i></p>

2. Class Diagram

Menurut Rosa dan M.Shalahuddin (2011 : 122) yang dimaksud dengan class diagram adalah suatu kelas yang menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sebuah sistem.

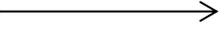
Tabel 2.3 Daftar Simbol Class Diagram

No	Simbol	Keterangan
1		<p><i>Class</i></p> <p>Menunjukkan class-class yang dibangun berdasarkan proses-proses sebelumnya</p>
2		<p><i>Unidirectional Association</i></p> <p>Menunjukkan hubungan antara class pada diagram class</p>

3. Activity Diagram

Secara grafis digunakan untuk menggambarkan rangkaian aliran aktivitas baik proses bisnis maupun *use case*. *Activity diagram* dapat juga digunakan untuk memodelkan action yang akan dilakukan saat sebuah operasi dieksekusi, dan memodelkan hasil dari action tersebut.

Tabel 2.4 Daftar Simbol Activity Diagram

No	Simbol	Keterangan
1		<p>Kondisi awal</p> <p>Menunjukkan awal dari suatu diagram aktivitas</p>
2		<p>Kondisi akhir</p> <p>Menunjukkan akhir dari suatu diagram aktivitas</p>
3		<p>Kondisi transisi</p> <p>Menunjukkan kondisi transisi antara</p>

		aktivitas
4		<i>Swimlane</i> Menunjukkan aktor dari diagram aktivitas yang dibuat
5		Aktivitas Menunjukkan aktivitas-aktivitas yang terdapat pada diagram aktivitas
6		Pengecekan kondisi Menunjukkan pengecekan terhadap suatu kondisi

2.14.3 ENTITY RELATIONSHIP DIAGRAM (ERD)

Entity Relation Diagram (ERD) adalah pemodelan awal basis data yang paling banyak digunakan untuk permodelan basis data relasional (Rosa dan Shalahuddin, 2014).

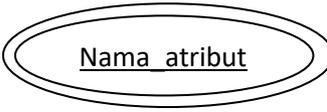
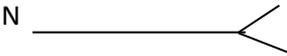
ERD dapat menguji model dengan mengabadikan proses yang harus dilakukan, ERD menggunakan sejumlah notasi dan simbol untuk menggambarkan struktur dan hubungan antara data. Simbol-simbol yang digunakan dalam pembuatan *Entity Relationship Diagram* terdiri dari:

Tabel 2.5 Simbol *Entity Relationship Diagram* (ERD)

Simbol	Keterangan
	Entitas merupakan data inti yang akan disimpan; bakal tabel pada basis data; benda yang memiliki data dan disimpan agar dapat diakses oleh aplikasi komputer; penamaan entitas lebih ke kata benda.
Entitas/ Entity	

 Atribut	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas
---	--

Tabel 2.5 Simbol *Entity Relationship Diagram* (ERD) (Lanjutan)

Simbol	Keterangan
 Atribut Kunci primer	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas dan digunakan sebagai kunci akses <i>record</i> yang diinginkan
 Atribut Multinilai	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas yang dapat memiliki nilai lebih dari satu
 Relasi	Relasi yang menghubungkan antar entitas; diawali kata kerja
 Asosiasi/ Association	Penghubung antar relasi dan entitas

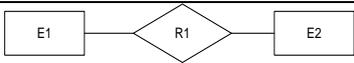
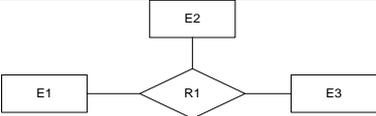
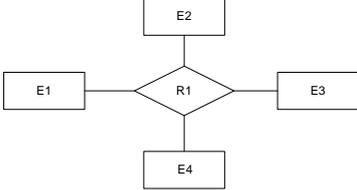
Tahapan-tahapan yang dilakukan ketika akan membuat *Entity Relationship Diagram* (ERD) yaitu sebagai berikut:

1. Menentukan entitas yang diperlukan
2. Menentukan relasi antar entitas

3. Menentukan *cardinality*
4. Menentukan atribut-atribut dan key yang diperlukan dari tiap entitas
5. Membuat *Entity Relationship Diagram* (ERD)

Dalam *Entity Relationship Diagram* relasi yang bisa terjadi antara dua entitas dapat dilihat dibawah ini :

Tabel 2.6 Hubungan Antar Entitas

No	Simbol	Keterangan
1.	 <p style="text-align: center;">Binary</p>	Satu relasi menghubungkan dua entitas
2	 <p style="text-align: center;">Ternery</p>	Satu relasi menghubungkan tiga buah entitas
3.	 <p style="text-align: center;">N-ary</p>	Satu relasi menghubungkan banyak entitas

2.14.4 RELASI ANTAR TABEL

1. One to One :

Mempunyai pengertian "Setiap baris data pada tabel pertama dihubungkan hanya ke satu baris data pada tabel ke dua".

2. One to Many:

Mempunyai pengertian "Setiap baris data dari tabel pertama dapat dihubungkan ke satu baris atau lebih data pada tabel ke dua".

3. Many to Many:

Mempunyai pengertian "Satu baris atau lebih data pada tabel pertama bisa dihubungkan ke satu atau lebih baris data pada tabel ke dua".

2.14.5 SPESIFIKASI FILE

Menjelaskan spesifikasi tabel-tabel yang diperoleh dari hasil normal pada tahap terakhir. Spesifikasi file dibuat berdasarkan DBMS (*Database Management System*) yang akan digunakan dalam membangun suatu sistem.

2.14.6 KAMUS DATA

Kamus data (*Data Dictionary*) adalah kumpulan daftar elemen data yang mengalir pada sistem perangkat lunak sehingga masukan (*input*) dan keluaran (*output*) dapat dipahami secara umum (memiliki standar penulisan). Kamus data dalam implementasi program dapat menjadi parameter masukan atau keluaran dari sebuah fungsi atau prosedur (M. Shalahudin, 2014).

Format Kamus Data

Nama database :

Nama tabel :

Primary Key :

Foreign Key :

Nama Field	Type	Size	Kondisi	Keterangan

Keterangan: Kondisi berisi (contoh: NULL/NOT NULL)

Gambar 2.1 Format Kamus Data

2.14.7 SISTEM KODE

Kode digunakan untuk mengklasifikasikan data, memasukkan data ke dalam komputer dan untuk mengambil bermacam-macam informasi yang berhubungan dengannya. Kode dapat berupa kumpulan angka, huruf dan karakter khusus. Di dalam merancang suatu struktur kode terdapat tipe dari kode yang dapat dibuat. (Jogiyanto HM, 2005).

Maka berdasarkan hal tersebut struktur kode terbagi menjadi :

a. Kode Mnemonik

Kode mnemonik digunakan untuk tujuan supaya mudah diingat. Kode mnemonik dibuat dengan dasar singkatan atau mengambil sebagian karakter

dari item yang akan diwakili dengan kode ini. Misalnya kode “P” untuk mewakili Pria dan kode “W” untuk wanita akan mudah diingat.

b. Kode Urut

Kode urut disebut juga dengan kode seri merupakan kode yang nilainya urut antara satu kode dengan kode berikutnya.

c. Kode Blok

Kode blok mengklasifikasikan item kedalam kelompok blok tertentu yang mencerminkan satu klasifikasi tertentu atas dasar pemakaian maksimum yang diharapkan.

d. Kode Grup

Kode Group merupakan kode berdasarkan *field – field* dan tiap – tiap *field* kode mempunyai arti. Kalau anda mengamati buku – buku teks, maka akan terlihat suatu kode yang disebut dengan ISBN (*International Standard Book Number*) yang terdiri dari 10 digit terbagi dalam 4 *field*. ISBN merupakan kode group yang masing – masing *field* mempunyai arti tertentu.

e. Kode Desimal

Kode desimal mengklasifikasikan kode atas dasar 10 unit angka desimal dimulai dari angka 0 sampai dengan angka 9 atau dari 00 sampai dengan 99 tergantung dari banyaknya kelompok.

2.15 SOFTWARE APLIKASI

Software aplikasi terdiri atas program yang berdiri sendiri yang mampu mengatasi kebutuhan bisnis tertentu. Aplikasi memfasilitasi operasi bisnis atau pengambilan keputusan manajemen maupun teknik sebagai tambahan dalam aplikasi pemrosesan data konvensional. Software aplikasi digunakan untuk mengatur fungsi bisnis secara *real time* (Pressman, 2005).

2.16 BASIS DATA DAN SISTEM MANAJEMEN BASIS DATA (DATABASE AND DATABASE MANAGEMENT SYSTEM)

Database adalah sekumpulan data yang saling berelasi (Elmasri, 2000). *Database* didesain, dibuat, dan diisi dengan data untuk tujuan mendapatkan informasi tertentu. Pendekatan *database* memiliki beberapa keuntungan seperti keberadaan katalog, independensi program-data, mendukung *view* (tampilan) untuk banyak pengguna, dan *sharing data* pada sejumlah transaksi. Selain itu masih ada *fleksibilitas*, ketersediaan *up-to-date* informasi untuk semua pengguna, skala ekonomis. Kategori utama pengguna *database* terbagi menjadi empat kategori, yakni *Administrator*, *Designer*, *End user*, *System Analyst* dan *Application Programmers*. *Administrator* atau *Data Base Administrator* (DBA) bertanggung jawab pada otoritas akses *database*, koordinasi dan monitoring penggunaan, dan pemilihan perangkat keras dan lunak yang dibutuhkan. *Designer* bertanggung jawab pada identifikasi data yang disimpan dalam *database* dan memilih struktur yang tepat untuk menggambarkan dan menyimpan data. *End User* adalah orang yang kegiatannya membutuhkan akses ke *database* untuk melakukan *query*, *update*, dan membuat laporan. *System Analysts* menentukan kebutuhan *End User*. *Application Programmers* mengimplementasikan program sesuai spesifikasi.

Sistem Manajemen Basis Data (SMBD) adalah program yang digunakan pengguna untuk membuat dan memelihara *database*. SMBD memfasilitasi untuk mendefinisikan, mengkonstruksi, dan memanipulasi *database* untuk berbagai aplikasi. Pendefinisian *database* meliputi spesifikasi tipe data, struktur, dan constraint untuk data yang disimpan dalam *database*. Pengkonstruksian *database* adalah proses penyimpanan data itu sendiri pada media penyimpanan. Pemanipulasian *database* meliputi fungsi memanggil *query database* untuk mendapatkan data yang spesifik, *update database*, dan *meng-generate* laporan dari data tersebut (Elmasri, 2000). Keuntungan yang diperoleh menggunakan SMBD meliputi mengontrol *redudansi*, membatasi akses yang tidak berwenang, menyediakan penyimpanan yang persisten, menghasilkan *interface* (antar muka) banyak pengguna, menjaga integritas constraint, menyediakan *backup* dan *recovery*. Dalam SMBD menyediakan perintah yang digunakan untuk mengelola dan mengorganisasikan data, yakni *Data Definition Language* (DDL) dan *Data Manipulation Language* (DML). *Data Definition*

Language adalah bahasa untuk mendefinisikan skema atau dan database fisik ke SMDB. (DDL). *Data Manipulation Language* adalah bahasa untuk memanipulasi data yaitu pengambilan informasi yang telah disimpan, penyisipan informasi baru, penghapusan informasi, modifikasi informasi yang disimpan dalam *database*. Selanjutnya, *query* adalah statemen yang ditulis untuk mengambil informasi. Bagian dari DML yang menangani pengambilan informasi ini disebut bahasa *query*. SQL (dibaca "ess-que-el") singkatan dari *Structured Query Language*. SQL adalah bahasa yang digunakan untuk berkomunikasi dengan *database*. Menurut ANSI (*American National Standards Institute*), bahasa ini merupakan standard untuk *relational database management systems* (RDBMS). Secara prinsip, perintah-perintah SQL (biasa disebut dengan pernyataan) dapat dibagi dalam tiga kelompok, yaitu :

1. DDL (*Data Definition Language*) atau bahasa penerjemah data

Adalah perintah-perintah yang berkaitan dengan penciptaan atau penghapusan objek seperti tabel dan indek dalam database. Versi ANSI mencakup CREATE TABLE, CREATE INDEX, ALTER TABLE, DROP TABLE, DROP VIEW, dan DROP INDEX. Beberapa sistem *database* menambahkan pernyataan DDL seperti CREATE DATABASE dan CREATE SCHEMA.

2. DML (*Data Manipulation Language*) atau bahasa pemanipulasi data

Mencakup perintah-perintah yang digunakan untuk memanipulasi data. Misalnya untuk menambahkan data (INSERT), memperoleh data (SELECT), mengubah data (UPDATE), dan menghapus data (DELETE).

3. DCL (*Data Control Language*) atau bahasa pengendali data

Merupakan kelompok perintah yang dipakai untuk melakukan otorisasi terhadap pengaksesan data dan pengalokasian ruang. Misalnya, suatu data bisa diakses si A, tetapi tidak bisa diakses oleh si B. Termasuk dalam kategori DCL yaitu pernyataan-pernyataan GRANT, REVOKE, COMMIT, dan ROLLBACK (Kadir, 1999).

2.17 PERANGKAT LUNAK (SOFTWARE) PENDUKUNG

Adapun perangkat lunak yang digunakan dalam perencanaan perancangan sistem diantaranya menggunakan NetBeans IDE (versi 6.0) dan MySql sebagai basis data (*database*).

2.17.1 NETBEANS IDE (VERSI 6.0)

NetBeans adalah *Integrated Development Environment* (IDE) berbasis Java dari *Sun Microsystems* yang berjalan di atas *Swing*. *Swing* sebuah teknologi Java untuk pengembangan aplikasi Desktop yang dapat berjalan di berbagai macam *platforms* seperti Windows, Linux, Mac OS X and Solaris. Netbeans merupakan *software development* yang *Open Source*, dengan kata lain software ini di bawah pengembangan bersama, bebas biaya NetBeans merupakan sebuah proyek kode terbuka yang sukses dengan pengguna yang sangat luas, komunitas yang terus tumbuh, dan memiliki hampir 100 mitra. *Sun Microsystems* mendirikan proyek kode terbuka NetBeans pada bulan Juni 2000 dan terus menjadi sponsor utama. Suatu IDE adalah lingkup pemrograman yang diintegrasikan kedalam suatu aplikasi perangkat lunak yang menyediakan pembangun *Graphic User Interface* (GUI), suatu text atau kode editor, suatu *compiler* atau *interpreter* dan suatu *debugger*.

The NetBeans IDE adalah sebuah lingkungan pengembangan – sebuah kaskas untuk pemrogram menulis, mengompilasi, mencari kesalahan dan menyebarkan program. Netbeans IDE ditulis dalam Java - namun dapat mendukung bahasa pemrograman lain. Terdapat banyak modul untuk memperluas Netbeans IDE. Netbeans IDE adalah sebuah produk bebas dengan tanpa batasan bagaimana digunakan. NetBeans IDE mendukung pengembangan semua tipe aplikasi Java (J2SE, web, EJB, dan aplikasi *mobile*). Fitur lainnya adalah sistem proyek berbasis Ant, kontrol versi, dan *refactoring*.

2.17.2 MySQL

MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL (bahasa Inggris: *database management system*) atau DBMS yang *multi thread*, dan *multi-user*. MySQL dimiliki dan di sponsori oleh sebuah perusahaan komersial Swedia MySQL AB, dimana memegang hak cipta hampir atas semua kode sumbernya. Kedua orang Swedia dan satu orang Finlandia yang mendirikan MySQL AB adalah: David Axmark, Allan Larsson, dan Michael "Monty" Widenius.

MySQL adalah *Relational Database Management System* (RDBMS) yang didistribusikan secara gratis dibawah lisensi GPL (*General Public License*). Dimana

setiap orang bebas untuk menggunakan MySQL, namun tidak boleh dijadikan produk turunan yang bersifat komersial. MySQL sebenarnya merupakan turunan salah satu konsep utama dalam database sejak lama, yaitu SQL (*Structured Query Language*). SQL adalah sebuah konsep pengoperasian *database*, terutama untuk pemilihan atau seleksi dan pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis. Contoh MySQL adalah APACHE, XAMPP, WampServer.

2.18 PERANCANGAN APLIKASI

Proses perancangan sistem membagi persyaratan dalam sistem perangkat keras atau perangkat lunak. Kegiatan ini menentukan arsitektur sistem secara keseluruhan. Perancangan perangkat lunak melibatkan identifikasi dan deskripsi abstraksi sistem perangkat lunak yang mendasar dan hubungan-hubungannya (Sommerville, 2003). Sebagaimana persyaratan, desain didokumentasikan dan menjadi bagian dari konfigurasi *software* (Pressman, 1997). Tahap desain meliputi perancangan data, perancangan fungsional, dan perancangan antarmuka.

1. Perancangan data

Perancangan data mentransformasikan model data yang dihasilkan oleh proses analisis menjadi struktur data yang dibutuhkan pada saat pembuatan program (*coding*). Selain itu juga akan dilakukan desain terhadap struktur *database* yang akan dipakai.

2. Perancangan fungsional

Perancangan fungsional mendeskripsikan kebutuhan fungsi-fungsi utama perangkat lunak.

3. Perancangan antarmuka mendefinisikan bagaimana pengguna (*user*) dan perangkat lunak berkomunikasi dalam menjalankan fungsionalitas perangkat lunak.

2.19 IMPLEMENTASI DAN PENGUJIAN UNIT

Pada tahap ini, perancangan perangkat lunak direalisasikan sebagai serangkaian program atau unit program. Kemudian pengujian unit melibatkan verifikasi bahwa setiap unit program telah memenuhi spesifikasinya (Sommerville, 2003). Program sebaiknya dirilis setelah dikembangkan, diuji untuk memperbaiki

kesalahan yang ditemukan pada pengujian untuk menjamin kualitasnya (Padmini, 2005). Terdapat dua metode pengujian yaitu :

- 1) Metode *white box* yaitu pengujian yang berfokus pada logika internal software (*source code program*).
- 2) Metode *black box* yaitu mengarahkan pengujian untuk menemukan kesalahan-kesalahan dan memastikan bahwa input yang dibatasi akan memberikan hasil aktual yang sesuai dengan hasil yang dibutuhkan.

Pada tahap pengujian, penulis melakukan metode *black box* yaitu menguji fungsionalitas dari perangkat lunak saja tanpa harus mengetahui struktur internal program (*source code*).

2.20 FAKTOR-FAKTOR YANG MEMPENGARUHI SISWA DITERIMA DI PERGURUAN TINGGI NEGERI

Keputusan seleksi masuk perguruan tinggi negeri ditentukan oleh beberapa faktor. Faktor tersebut digunakan sebagai penentu siswa dapat diterima oleh perguruan tinggi yang diinginkan. Pada penelitian ini, digunakan beberapa faktor pendukung yang dapat memprediksi tingkat diterimanya siswa di perguruan tinggi negeri. Berikut adalah faktor-faktor yang digunakan dalam penelitian ini, yaitu :

1. Rata-Rata Nilai Raport

Fungsi pokok evaluasi hasil belajar siswa secara umum adalah untuk mengukur tingkat kemajuan siswa dalam belajar, untuk menyusun rencana belajar selanjutnya dan untuk memperbaiki proses pembelajaran (Muhammad Irham dan Novan A.W., 2013: 217). Laporan evaluasi hasil belajar siswa dituliskan pada sebuah dokumen yaitu rapor. Nilai rapor ditulis berdasarkan hasil belajar siswa dalam satu semester dan ditulis pada akhir semester. Sebagai salah satu faktor penentu, rata-rata nilai rapor yang digunakan adalah semester satu sampai semester lima.

2. Nilai Ujian Nasional

Ujian nasional merupakan sesuatu yang diwajibkan bagi para siswa untuk persyaratan kelulusan. Hasil ujian dapat dijadikan bukti kesanggupan siswa berpikir melalui proses yang memenuhi standar kompetensi yang ditentukan dan sesuai

dengan prosedur akademik (Faiz Hidayat, 2013). Nilai ujian nasional digunakan untuk faktor yang berpengaruh masuk perguruan tinggi.

3. Tingkat Pendidikan Orang Tua

Pendidikan merupakan suatu usaha untuk mendewasakan dan memandirikan manusia melalui kegiatan yang terencana dan disadari melalui kegiatan belajar dan pembelajaran yang melibatkan siswa dan guru (Muhammad Irham dan Novan A.W., 2013 : 19). Tingkat pendidikan diukur dari pendidikan terakhir yang ditempuh dari tingkat Sekolah Dasar (SD), Sekolah Menengah Pertama (SMP), Sekolah Menengah Atas (SMA), hingga perguruan tinggi. Kondisi sosial ekonomi berpengaruh terhadap munculnya perbedaan, salah satunya yaitu tingkat pendidikan orang tua. Tingkat pendidikan orang tua berpengaruh pada perkembangan seorang anak sejak kecil hingga dewasa. Perbedaan itu terlihat pada cara pandang terhadap suatu kondisi tertentu. Menurut Esti Setya R. (2012), tingkat pendidikan orang tua akan menentukan cara orang tua dalam membimbing dan mengarahkan anaknya dalam hal pendidikan yang sedang maupun yang akan ditempuh anaknya. Orang tua yang berwawasan luas akan mengarahkan dan membimbing anaknya untuk terus menimba ilmu melebihi orang tuanya. Semakin tinggi tingkat pendidikan orang tua maka semakin tinggi pula minat siswa untuk melanjutkan sekolah ke perguruan tinggi. Tingkat pendidikan orang tua ini dibagi menjadi dua atribut yaitu pendidikan ayah dan pendidikan ibu.

4. Jenis Pekerjaan Orang Tua

Pengertian pekerjaan menurut Poerwodarminta (1996: 180) adalah sesuatu yang dilakukan untuk mencari nafkah dan untuk mengubah dirinya dengan tujuan meningkatkan taraf hidup. Semakin baik jenis pekerjaan orang tua maka semakin tinggi tingkat pendapatan orang tua. Pendapatan orang tua dapat menunjang pendidikan anaknya dalam bentuk materi, baik dalam memberi fasilitas untuk kegiatan pembelajaran, biaya sekolah, dan sebagainya. Orang tua yang mempunyai tingkat pendapatan tinggi akan mendukung penuh anaknya untuk meneruskan studi ke jenjang yang lebih tinggi, dalam hal ini anak yang ingin meneruskan studinya ke perguruan tinggi negeri.

Salah satu aspek sebagai indikator kualitas pada sekolah menengah atas adalah tingkat diterimanya siswa di perguruan tinggi negeri. Dari faktor-faktor yang

berpengaruh yang sudah dijelaskan, maka data tersebut akan dihubungkan dengan tingkat diterimanya siswa di perguruan tinggi negeri.

2.21 PENELITIAN TERKAIT DATA MINING

- a. Dimas Dwi Angen Saputra. (2014). Model Prediksi Tingkat Kelulusan Mahasiswa dengan Teknik Data Mining Menggunakan Metode Decision Tree C4.5. Skripsi Program Studi Matematika Jurusan Pendidikan Matematika Universitas Negeri Yogyakarta.
- b. Esti Setya Rini. (2012). Hubungan Tingkat Pendidikan Orang Tua dan Prestasi Belajar Siswa dengan Minat Siswa Melanjutkan Studi ke Perguruan Tinggi pada Siswa Kelas XI SMA Negeri 1 Kalasan Tahun Ajaran 2011/2012. Skripsi Jurusan Pendidikan Akuntansi Fakultas Ekonomi Universitas Negeri Yogyakarta.
- c. John Fredrik U. (2013). Data Mining Classification untuk Prediksi Lama Masa Studi Mahasiswa Berdasarkan Jalur Penerimaan dengan Metode Naive Bayes. Jurnal Magister Teknik Informatika Universitas Atma Jaya Yogyakarta.
- d. Kabir, M. F., Rahman, C. M., Hossain, A., & Dahal, K. (2011). Enhanced Classification Accuracy on Naive Bayes Data Mining Models. *International Journal of Computer Applications*, Volume 28, No.3.
- e. Taruna R., S., Hiranwal, S., 2013, Enhanced Naive Bayes Algorithm for Intrusion Detection in Data Mining, *International Journal of Computer Science and Information Technologies*, Vol.6, No. 4, Hal 960-962.
- f. Ridwan, M., Suyono, H., Sarosa, M., 2013, Penerapan Data Mining untuk Evaluasi Kinerja Akademik Mahasiswa Menggunakan Algoritma Naive Bayes Classifier, *Jurnal EECCIS*, Vol 1, No. 7, Hal. 59-64.