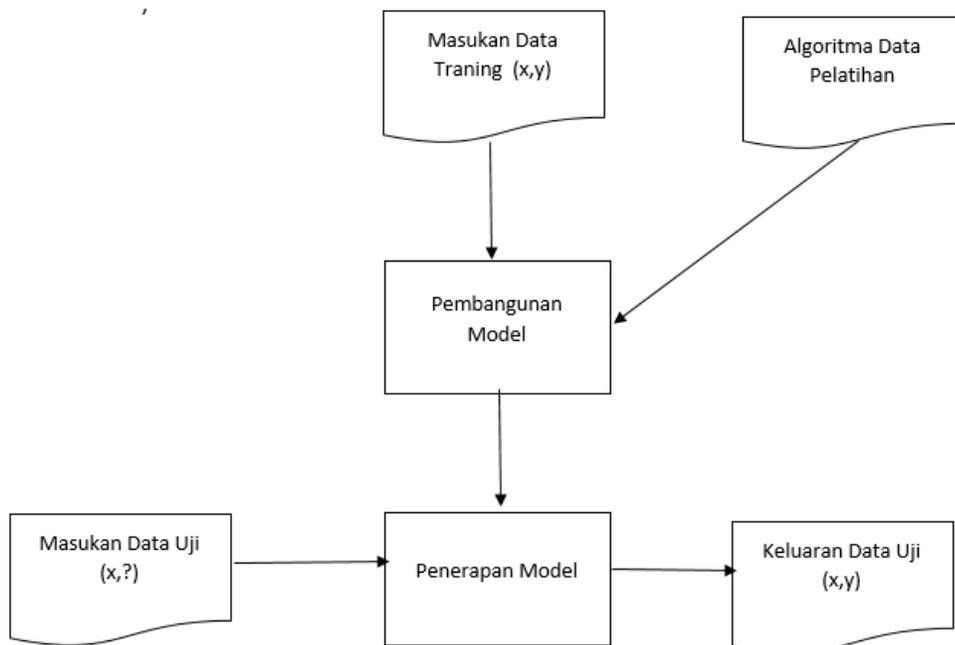


BAB II LANDASAN TEORI

2.1 Konsep Klasifikasi

Menurut (Prasetyo, 2014) menyatakan Bahwa klasifikasi dapat didefinisikan secara detail sebagai suatu pekerjaan yang menggunakan pelatihan atau pembelajaran terhadap fungsi target f yang memetakan setiap vektor (sub fitur) x ke dalam satu dari sejumlah label kelas y yang tersedia. Pekerjaan pelatihan tersebut akan menghasilkan suatu model yang kemudian disimpan sebagai memori.



Gambar 2.2 Proses Pekerjaan Klasifikasi (Prasetyo,2014).

Kerangka kerja klasifikasi ditunjukkan pada gambar 2.2. Pada gambar tersebut disediakan sejumlah data latih (x,y) untuk digunakan sebagai data pembangun model, kemudian menggunakan model tersebut untuk memprediksi kelas dari data uji $(x,?)$ sehingga data uji $(x,?)$ diketahui kelas y yang seharusnya.

Model yang sudah dibangun pada saat pelatihan kemudian dapat digunakan untuk memprediksi label kelas dari data baru yang belum diketahui label kelasnya. Dalam

pembangunan model selama proses pelatihan tersebut diperlukan adanya suatu algoritma untuk membangunnya yang disebut sebagai algoritma pelatihan. Ada banyak algoritma pelatihan yang sudah dikembangkan oleh para peneliti seperti *Decision Tree*, *K-Nearest Neighbor*, *Artificial Neural Network* dan sebagainya. Setiap algoritma mempunyai prinsip yang sama yaitu melakukan suatu pelatihan sehingga di akhir pelatihan model dapat memetakan (memprediksi) setiap vektor masukan ke label keluaran yang benar.

2.2 Algoritma Naive Bayes

Algoritma *Naive Bayes* berbasiskan perhitungan probabilistik dengan asumsi bahwa setiap fitur yang digunakan saling lepas. *Naive Bayes* merupakan metode klasifikasi teks yang paling populer digunakan. Algoritma ini memiliki kelebihan dari sisi kecepatan pembelajaran dan toleransinya terhadap nilai yang hilang dari fitur. Untuk menangani data numerik, algoritma ini menggunakan *probability density function*, artinya data dianggap mengikuti distribusi normal untuk kemudian dihitung nilai rata-rata dan simpangan bakunya.

Untuk merepresentasikan sebuah kelas, terdapat karakteristik petunjuk yang dibutuhkan untuk melakukan klasifikasi yang berguna untuk menjelaskan bahwa peluang masuknya sampel karakteristik tertentu kedalam kelas *posterior*. Peluang munculnya suatu kelas (sebelum masuknya sampel tersebut, seringkali disebut *prior*), dikali dengan peluang kemunculan karakteristik sampel secara global disebut juga *evidence*. Nilai *evidence* selalu tetap untuk setiap kelas pada satu sampel. Nilai *posterior* tersebut dibandingkan dengan nilai *posterior* kelas lainnya untuk menentukan ke kelas apa suatu sampel .

Klasifikasi *Naive Bayes* diasumsikan bahwa ada atau tidak ciri tertentu dari sebuah kelas tidak ada hubungannya dengan ciri dari kelas lainnya. Persamaan dari teorema *bayes* adalah sebagai berikut.

$$P(H|X) = \frac{P(X|H)P(H)}{P(X)} \dots\dots\dots (2-1)$$

Dengan:

X: Data dengan kelas yang belum diketahui;

H: Hipotesis data X merupakan suatu label kelas tertentu;

P(H|X): Probabilistik hipotesis H berdasarkan kondisi X (*posteriori probability*);

P(H): Probabilistik hipotesis H (*prior probability*);

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

P(X): Probabilistik X;

Untuk menjelaskan teorema *naive bayes*, perlu diketahui bahwa proses klasifikasi memerlukan sejumlah petunjuk untuk menentukan kelas apa yang cocok bagi sampel yang dianalisis tersebut. Karena itu, teorema *bayes* tersebut akan disesuaikan sebagai berikut.

$$P(C|F_1 \dots F_n) = \frac{P(C)P(F_1 \dots F_n|C)}{P(f_1 \dots F_n)} \dots(2-2)$$

Dengan

C = Sebuah kelas; F₁ ... F_N = Karakteristik Petunjuk.

$$\text{Posterior} = \frac{\text{Prior} \times \text{likelihood}}{\text{evidence}} \dots(2-3)$$

Penjabaran lebih lanjut sebagai Berikut :

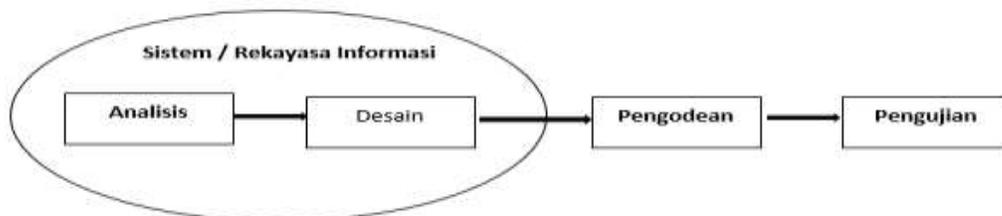
$$\begin{aligned} P(C|F_1, \dots, F_n) &= P(C) P(F_1, \dots, F_n | C) \\ &= P(C) P(F_1|C) \\ &= P(C) P(F_1|C) \\ &= P(C)P(F_1|C)P(F_2|C,F_1)P(F_3|C,F_1,F_2),P(F_4, \dots, F_n|C,F_1,F_2) \\ &= P(C)P(F_1|C)P(F_2|C,F_1)P(F_3|C,F_1,F_2), \dots, (F_n|C,F_1,F_2, \dots, F_{n-1}) \end{aligned}$$

Adapun langkah – langkah pada *naïve bayes* yang akan dilakukan menggunakan rumus diatas adalah sebagai berikut (Suyanto,2017):

1. Menentukan atribut yang akan digunakan.
2. Menentukan data set pada data training atau data testing.
3. Menghitung seluruh atribut menggunakan perhitungan algoritma *naïve bayes* sehingga menghasilkan perhitungan per-atribut yang digunakan.
4. Menentukan kasus baru yang akan di klasifikasikan menggunakan metode *naïve bayes*.
5. Mengklasifikasikan kasus baru berdasarkan kasus yang sama dengan kasus yang lama.
6. Mengalikan semua kelas variable untuk mendapatkan nilai dari masing – masing kelas.
7. Membandingkan hasil perkalian dari masing – masing kelas maka akan diketahui nilai warning untuk status *dropout* ya atau tidak, berdasarkan hasil nilai terbesar dari masing – masing kelas.

2.3 Metode Waterfall

Menurut (Rosa dan Shalahudin, 2016) menjelaskan bahwa model SDLC air terjun (*waterfall*) sering juga disebut model sekuensial linier (*sequential linier*) atau alur hidup klasik (*clasic life cycle*). Model air terjun menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau berturut dimulai dari analisis, desain, pengkodean, pengujian dan tahap pendukung (*support*). Berikut ini adalah gambar model air terjun :



Gambar 2.4 Metode *Waterfall*(Roda dan Sahalahuddin,2016).

1). Analisis kebutuhan perangkat lunak

Proses pengumpulan kebutuhan dilakukan secara intensif untuk memspezifikasikan kebutuhan perangkat lunak agar dapat dipahami perangkat lunak seperti apa yang dibutuhkan oleh *user*. Spesifikasi kebutuhan perangkat lunak pada tahap ini perlu untuk di dokumentasikan.

2). Desain

Desain perangkat lunak proses multi langkah yang fokus pada desain pembuatan program perangkat lunak termasuk struktur data, arsitektur perangkat lunak, representasi antarmuka dan prosedur pengkodean. Tahap ini mentranslasi desain agar dapat di implementasikan menjadi program pada tahap selanjutnya. Desain perangkat lunak yang dihasilkan pada tahap ini juga perlu didokumentasikan.

3). Pembuatan kode program

Desain harus ditranslasikan ke dalam program perangkat lunak. Hasil dari tahap ini adalah program komputer sesuai dengan desain yang telah dibuat pada tahap desain.

4). Pengujian

Pengujian fokus pada perangkat lunak secara dari segi logik dan fungsional dan memastikan bahwa semua bagian sudah diuji. Hal ini dilakukan untuk meminimalisir kesalahan (*error*) dan memastikan keluaran yang dihasilkan sesuai dengan yang diinginkan.

2.4 Database

Sebuah *database* adalah sebuah struktur yang umumnya dikategorikan dalam 2 hal: Sebuah *database flat* dan sebuah *database relational*. *Database relational* lebih disukai karena lebih masuk akal dibandingkan *database flat* MySQL adalah sebuah *database relational*. Pada database yang memiliki struktur relasional. Ada tabel-tabel yang menyimpan data. Setiap tabel terdiri dari kolom dan baris. Sebuah kolom mendefinisikan jenis informasi apa yang akan disimpan. Diperlukan kolom khusus untuk setiap jenis informasi yang ingin di simpan (misalnya umur, tinggi, berat, alamat).(Abdul Kadir,2002).

Kalau kolom mendefinisikan jenis informasi apa yang akan disimpan, maka sebuah baris adalah data aktual yang disimpan. Setiap baris dari tabel adalah masukan dari tabel tersebut dan berisi nilai-nilai untuk setiap kolom tabel tersebut. (contoh masukannya adalah Jojon: umur 35 tahun, 65 kg dan 165cm).

2.4.1 DBMS

Menurut Rosa dan Shalahuddin (2011), DBMS (*Database Management System*) adalah suatu sistem aplikasi yang digunakan untuk menyimpan, mengelola, dan menampilkan data. Suatu sistem aplikasi disebut DBMS jika memenuhi persyaratan minimal sebagai berikut :

- a. Menyediakan fasilitas untuk mengelola akses data
- b. Mampu menangani integritas data
- c. Mampu menangani akses data
- d. Mampu menangani backup data

Karena pentingnya data bagi suatu organisasi atau perusahaan, maka hampir sebagian besar perusahaan memanfaatkan DBMS dalam mengelola data yang mereka miliki. Pengelola DBMS sendiri biasanya ditangani oleh tenaga ahli yang spesialis menangani DBMS yang disebut sebagai DBA (*Database Administrator*).

Berikut ini adalah 4 macam DBMS versi komersia; yang paling bnyak digunakan di dunia saat ini, yaitu :

- a. Oracle
- b. Microsoft SQL Server
- c. IBM DB2
- d. Microsoft Access

Sedangkan DBMS versi *open source* yang cukup berkembang dan paling banyak digunakan saat ini adalah sebagai berisku :

- a. MySQL
- b. PostgreSQL
- c. Firebird

Hampir semua DBMS mengadopsi SQL sebagai bahasa untuk mengelola data pada DBMS.

2.4.2 SQL

(Rosa dan Shalahuddin, 2014) SQL (Structured Query Language) adalah bahasa yang digunakan untuk mengelola data pada RDBMS. SQL awalnya dikembangkan berdasarkan teori aljabar relasional dan kalkulus. MySQL adalah sebuah implementasi dari sistem manajemen basis data relasional (RDBMS) yang didistribusikan secara gratis dibawah lisensi GPL (*General Public License*). Setiap pengguna dapat secara bebas menggunakan MySQL, namun dengan batasan perangkat lunak tersebut tidak boleh dijadikan produk turunan yang bersifat komersial. MySQL sebenarnya merupakan turunan salah satu konsep utama dalam basis data yang telah ada sebelumnya yakni SQL (*Structured Query Language*). SQL adalah sebuah konsep pengoperasian basis data, terutama untuk pemilihan atau seleksi dan pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis (Herwanto, 2012).

2.4.3 Conceptual Data Model (CDM)

CDM adalah model yang dibuat berdasarkan anggapan bahwa dunia nyata terdiri dari koleksi obyek-obyek dasar yang dinamakan entitas serta hubungan antara entitas-entitas itu. Biasanya CDM direpresentasikan dalam bentuk Entity Relationship Diagram. Adapun manfaat penggunaan CDM dalam perancangan database dapat memberikan gambaran yang lengkap mengenai struktur basis data yaitu arti, hubungan, dan batasan-batasan dalam memodelkan struktur logis dari keseluruhan aplikasi data, tidak tergantung pada software atau pertimbangan model struktur data serta menggambarkan secara detail struktur basis data dalam bentuk logik. (Shandy Ardianto, 2012).

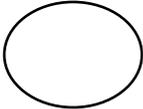
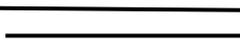
Jenis – jenis objek dalam CDM adalah sebagai berikut

1. Entity
2. Relationship
3. Inheritance

2.4.4 Data Flow Diagram (DFD)

Menurut (Swastika Windra, 2006) Data Flow Diagram adalah suatu model logika data atau proses yang dibuat lebih mendetail dibanding diagram konteks yang diperbolehkan, bisa dicapai dengan mengembangkan diagram. Sisa diagram asli dikembangkan ke dalam gambaran yang lebih terperinci yang melibatkan tiga sampai sembilan proses dan menunjukkan penyimpanan data dan aliran data baru pada level yang lebih rendah. Menurut Rosa A.S dan M. Shalahudin (2013:70) Data Flow Diagram (DFD) atau dalam Bahasa Indonesia menjadi Diagram Alir Data (DAD) adalah representasi grafik yang menggambarkan aliran informasi dan transformasi informasi yang diaplikasikan sebagai data yang mengalir dari masukan (*input*) dan keluaran (*output*). DFD dapat digunakan untuk merepresentasikan sebuah sistem atau perangkat lunak pada beberapa level yang lebih detail untuk merepresentasikan 16 aliran informasi lebih detail. DFD menyediakan mekanisme untuk pemodelan fungsional ataupun pemodelan aliran informasi. Oleh karena itu, DFD lebih sesuai digunakan untuk memodelkan fungsi-fungsi pada perangkat lunak yang akan diimplementasikan menggunakan pemrograman terstruktur dengan fungsi dan prosedur. Komponen penyusun DFD.

Tabel 2.1 Simbol-simbol DFD

Komponen	Keterangan
	Process merupakan sebuah bentuk kerja sistem yang dilakukan secara terkomputersasi
	Eclips mewakili atribut
	Garis mewakili relasi dan entitas atau relasi dan entitas dengan atribut
	<i>File</i> atau basis data atau penyimpanan (<i>storage</i>) pada pemodelan perangkat lunak yang akan di implementasikan.

2.5 Aplikasi Web

Menurut (Abdul Kadir, 2014) *World Wide Web* (WWW) adalah sistem pengakses informasi dalam internet yang biasa dikenal dengan istilah web. Web menggunakan protokol yang disebut HTTP (*HyperText Transfer Protocol*) yang berjalan pada TCP/IP. Dengan menggunakan HyperText, pemakai dapat melompat dari suatu dokumen ke dokumen lain dengan mudah, dengan cukup mengklik text-text khusus yang pada awalnya ditandai dengan garis bawah. Penggunaan HyperText pada web juga telah dikembangkan lebih jauh menjadi HyperMedia. Dengan menggunakan pendekatan HyperMedia, tidak hanya text yang dapat dikaitkan, melainkan juga gambar, suara, dan bahkan video.

2.5.1 PHP

Menurut (Arief, 2011) PHP adalah Bahasa *server side scripting* yang menyatu dengan HTML untuk membuat halaman web yang dinamis. Karena PHP merupakan *server side scripting* maka sintaks dan perintah-perintah PHP akan diesksekusi diserver kemudian hasilnya akan dikirimkan ke browser dengan format HTML. Dengan demikian kode program yang ditulis

dalam PHP tidak akan terlihat oleh user sehingga keamanan halaman web lebih terjamin. PHP dirancang untuk membuat halaman web yang dinamis, yaitu halaman web yang dapat membentuk suatu tampilan berdasarkan permintaan terkini, seperti menampilkan isi basis data ke halaman web. PHP juga dapat berjalan pada berbagai *web server* seperti IIS (*Internet Information Server*), PWS (*Personal Web Server*), *Apache*, *Xitami*. PHP juga mampu berjalan di banyak sistem operasi yang beredar saat ini, diantaranya : Sistem Operasi Microsoft Windows (semua versi), Linux, Mac Os, Solaris. PHP dapat dibangun sebagai modul *web server Apache* dan sebagai binary yang dapat berjalan sebagai CGI (*Common Gateway Interface*). PHP dapat mengirim HTTP header, dapat mengatur cookies , mengatur authentication dan redirect user. Salah satu keunggulan yang dimiliki PHP adalah kemampuannya untuk melakukan koneksi ke berbagai macam software sistem manajemen basis data atau *Database Management Sistem (DBMS)*, sehingga dapat menciptakan suatu halaman web dinamis. PHP mempunyai koneksitas yang baik dengan beberapa DBMS seperti *Oracle*, *Sybase*, *mSQL*, *MySQL*, *Microsoft SQL Server*, *Solid*, *PostgreSQL*, *Adabas*, *FilePro*, *Velocis*, *dBase*, *Unix dbm*, dan tidak terkecuali semua database ber-interface ODBC.

2.5.2 HTML

Menurut (Shelly dan Velmaart 2011) HTML (*Hypertext Markup Language*) adalah bahasa format khusus yang programmer gunakan untuk memformat dokumen untuk ditampilkan di web. Lalu menurut Chaffey (2009: 96) HTML atau Hyper Text Markup language adalah halaman standar web presentasi dengan menggunakan format untuk menentukan pesan dan tata letak halaman web. Berdasarkan pendapat para ahli di atas, penulis menyimpulkan bahwa HTML atau Hyper Text Markup Language adalah bahasa pemrograman yang digunakan untuk memformat dokumen untuk ditampilkan di web.

2.5.3 Javascript

Menurut (Kadir, 2009) menyatakan bahwa Javascript adalah bahasa skrip (bahasa yang kodenya ditulis menggunakan teks biasa) yang ditempelkan pada dokumen HTML dan diproses pada sisi klien. Dengan adanya bahasa ini, kemampuan dokumen HTML menjadi semakin luas. Sebagai contoh, dengan menggunakan JavaScript dimungkinkan untuk memvalidasi masukan – masukan pada formulir sebelum formulir dikirim ke *server*. Selain itu, dengan menggunakan JavaScript juga dimungkinkan untuk mengimplementasikan tugas yang bersifat interaktif tanpa berhubungan dengan *server*. Beberapa contoh yang bisa dilakukan melalui JavaScript :

1. Menampilkan jam lokal pada halaman web.
2. Mengatur warna latar belakang halaman web.
3. Mengganti gambar ketika pemakai menempatkan petunjuk *mouse* ke suatu gambar.
4. Memvalidasi keabsahan data yang dimasukan oleh pemakai.

2.5.4 CSS

Menurut (Prasetio, 2012) “*Casading Style Sheet (CSS)* adalah suatu teknologi yang digunakan untuk memperindah halaman *website* (situs)”. CSS mempunyai 2 bagian utama yaitu selectors dan deklarasi. Yang dimaksud selectrors biasanya element HTML yang ingin diubah, sedangkan deklarasi biasanya terdiri dari properti dan nilai. Properti sendiri adalah atribut style yang ingin diubah, dan setiap properti memiliki nilai. *Cascading Style Sheet (CSS)* merupakan aturan untuk mengatur beberapa komponen dalam sebuah web sehingga akan lebih tersstruktur dan seragam.

2.5.5 MySQL

MySQL adalah sebuah program *database server* yang mampu menerima dan mengirimkan datanya dengan sangat cepat, *multi user* serta menggunakan perintah standar SQL (*Structured Query Language*). *MySQL* juga telah mendukung bahasa pemrograman berfitur API seperti Java sehingga memudahkan para programmer java untuk berkoneksi dengan menggunakan *MySQL*. (Wahana Komputer, 2012).

Berikut beberapa keistimewaan yang dimiliki *MySQL* antara lain:

1. *Portabilitas* yaitu *MySQL* dapat berjalan secara stabil pada berbagai macam sistem operasi.
2. *Open Source* yaitu *MySQL* di distribusikan secara gratis.
3. *Multi User* yaitu *MySQL* dapat digunakan dengan banyak *user* tanpa memiliki masalah.
4. *Performance Turning* yaitu *MySQL* memiliki tipe kolom yang sangat kompleks.
5. Perintah dan fungsi yaitu *MySQL* operator dan fungsi yang secara penuh perintah *Select* dan *Where* dalam *query*.
6. Keamanan yaitu *MySQL* memiliki beberapa lapisan keamanan dengan sistem perizinan yang detail serta sandi yang terenkripsi.
7. Skalabilitas dan pembatasan yaitu *MySQL* mampu menangani *database* berskala besar dengan jumlah *record* lebih dari 50 juta, tabel lebih dari 60 ribu dan baris lebih dari 5 milyar.
8. *Konektivitas* yaitu *MySQL* dapat melakukan koneksi dengan klien menggunakan *TCP/IP*, *UNIX* dan *Named Pipes*.
9. Lokalisasi yaitu *MySQL* dapat mendeteksi kesalahan pada klien dengan menggunakan lebih dari dua puluh bahasa meskipun bahasa Indonesia belum masuk di dalamnya.
10. Antarmuka yaitu *MySQL* memiliki antarmuka yang mendukung berbagai bahasa pemrograman yang menggunakan fungsi API.

11. Klien dan Peralatan yaitu *MySQL* dilegkapi dengan berbagai *tool* yang dapat digunakan secara *online*.
12. Struktur tabel yaitu *MySQL* memiliki strukutr tabel yang *fleksibel* dalam menangani *alter table* jika dibandingkan dengan bahasa pemrograman lainnya.

2.6 Black-Box Testing (Pengujian Kotak Hitam)

Menurut (Rosa dan Shalahudin,2016) *Black-box testing* yaitu menguji perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode program. Pengujian dimaksudkan untuk mengetahui apakah fungsi – fungsi, masukan, dan keluaran dari perangkat lunak sesuai dengan spesifikasi yang dibutuhkan. Kasus uji yang dibuat untuk melakukan pengujian kotak hitam harus dibuat dengan kasus benar dan kasus salah, misal untuk proses *login* maka kasus uji yang dibuat adalah:

1. Jika *user* memasukan nama pemakai (*username*) dan kata sandi (*password*) yang benar.
2. Jika *user* memasukan nama pemakai (*username*) dan kata sandi (*password*) yang salah, misalnya nama pemakai benar tapi kata sandi salah, atau sebaliknya, atau keduanya salah.