

## **BAB II**

### **LANDASAN TEORI**

#### **2.1. Sistem Pakar**

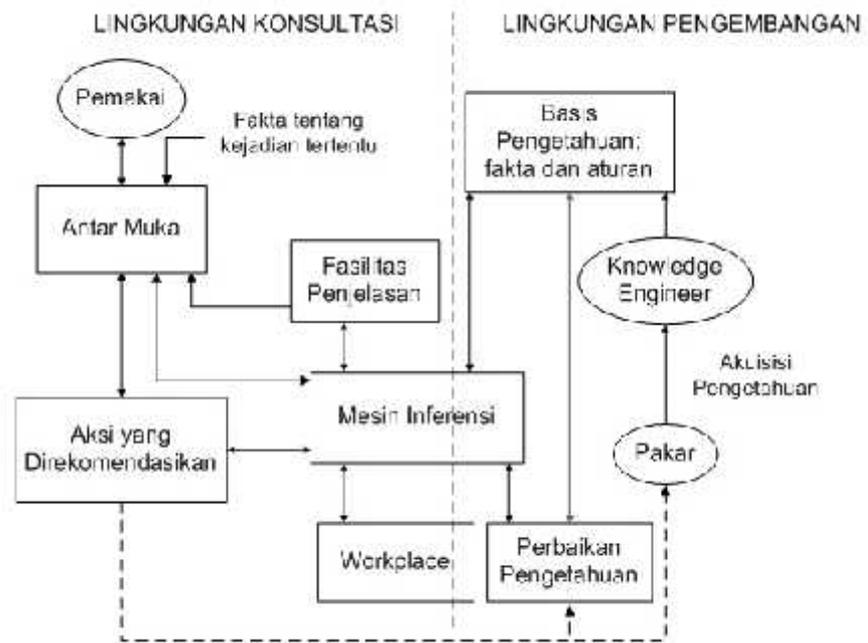
Sistem pakar kadang disebut sebagai sistem berbasis pengetahuan, merupakan sebuah sistem komputer yang dirancang untuk menganalisis data dan menghasilkan rekomendasi, diagnosis, atau keputusan berdasarkan seperangkat fakta dan aturan. Fakta-fakta dan aturan untuk sistem pakar biasanya berasal dengan mewawancarai seorang ahli atau pakar, dan kemudian dimasukkan ke dalam basis pengetahuan. Basis pengetahuan disimpan dalam file komputer dan dapat dimanipulasi oleh perangkat lunak yang disebut mesin inferensi. Proses perancangan, masuk dan menguji aturan dalam sistem mengerahkan disebut sebagai rekayasa pengetahuan. (Parsons dan Oja,2013).

System pakar memiliki banyak manfaat yang dapat diambil yakni:

1. Menyimpan pengetahuan dan keahlian para pakar.
2. Dapat melakukan proses secara berulang secara otomatis
3. Meningkatkan output dan produktifitas.
4. Meningkatkan kualitas.

Selain memiliki manfaat, system pakar juga memiliki kelemahan yakni biaya yang dibutuhkan untuk membuat dan memeliharanya sangat mahal, sulit dikembangkan. Hal itu tentu saja erat kaitannya dengan ketersediaan pakar bidangnya, system pakar tidak 100% benar nilainya. Daya kerja dan produktifitas manusia menjadi kurang karena semuanya dilakukan otomatis oleh system. System pakar memiliki dua bagian pokok

yakni lingkungan pengembangan (*development environment*) yakni untuk memasukan pengetahuan pakar kedalam lingkungan system pakar dan lingkungan konsultasi (*consultation environment*) yang digunakan oleh pengguna yang bukan pakar guna memperoleh pengetahuan pakar (Arhami, 2004 Dalam Dimas, 2014).



**Gambar 2.1. Arsitektur Sistem Pakar ( Arhami, 2004 Dalam Dimas, 2014 )**

Pembuatan Sistem pakar dapat dengan menggunakan bahasa pemrograman komputer, akan tetapi sistem pakar menawarkan satu set alat yang dirancang untuk menyederhanakan proses pembangunan. Sebuah sistem pakar adalah sebuah perangkat lunak yang berisi mesin inferensi dan user interface yang pengembang gunakan untuk memasukkan fakta-fakta dan aturan untuk basis pengetahuan. Sebuah

sistem pakar juga memiliki alat untuk menguji pengetahuan dasar untuk memastikan menghasilkan keputusan yang akurat.

### 2.1.1 Komponen Sistem Pakar

Sistem pakar merupakan aplikasi yang tersusun dari beberapa komponen, bagian-bagian system pakar (Arhami, 2014 dalam Dimas, 2014) , terdiri dari:

a. Representasi Pengetahuan

Logika merupakan suatu pengajian ilmiah tentang serangkaian penalaran. System kaidah dan prosedur yang membantu proses penalaran. Dalam melakukan penalaran, komponen harus dapat menggunakan proses penalaran deduktif dan induktif kedalam bentuk yang sesuai dengan manipulasi computer, yaitu berupa logika simbolik dan logika Matematik. Jaring semantic merupakan teknik representasi kecerdasan buatan klasik yang digunakan untuk informasi proporsional.

b. Basis Pengetahuan ( *Knowlegde Base* )

Basis pengetahuan merupakan bagian yang mengandung semua fakta-faktbaik fakta awal pada saat system ulai beroperasi maupun fakta-fakta yang didapatkan pada saat pengambilan kesimpulan sedang dilaksanakan.

c. Inferensi

Inferensi merupakan proses untuk menghasilkan informasi dari fakta yang diketahui atau diasumsikan. Inferensi adalah konklusi logis ( *logical conclusion* ) atau implementasi berdasarkan informasi yang tersedia.

#### d. Antar Muka

Antar muka Aplikasi untuk bagian penghubung antara system pakar dengan pemakai. Pada bagian ini akan terjadi dialog antara program dengan pemakai. Program akan mengajukan beberapa pertanyaan hasil Laboratorium . program system pakar akan mengampil kesimpulan berdasarkan jawaban-jawaban dari pemakai.

### 2.1.2 Ciri – Ciri Sistem Pakar

Ciri-ciri system pakar yaitu :

1. Terbatas pada bidang keahlian yang spesifik,
2. Dapat mengemukakan rangkaian alasan-alasan yang diberikan dengan cara yang dapat dipahami,
3. Dapat memberikan penalaran untuk data-data yang tidak lengkap atau tidak pasti,
4. Berdasarkan rule atau kaidah tertentu,
5. Dirancang untuk dikembangkan secara bertahap,
6. Keluaran bersifat anjuran atau nasehat,
7. Keluaran tergantung dari dialog dengan user.

### 2.2.2 Kelebihan Sistem Pakar

- a. Membuat seorang yang awam dapat bekerja seperti layaknya seorang pakar,
- b. Dapat bekerja dengan informasi yang tidak lengkap atau tidak pasti
- c. Meningkatkan output dan produktivitas.

## 2.2. Metode Naive Bayes

Merupakan pengklasifikasian dengan metode probabilitas dan statistik yang dilakukan oleh ilmuwan Inggris Thomas Bayes, yaitu memprediksi peluang dimasa depan berdasarkan pengalamana dimasa sebelumnya sehingga dikenal sebagai Thomas Bayes. Naive bayes untuk setiap kelas keputusan, menghitung probabilitas dengan syarat bahwa kelas keputusan adalah benar, mengingat vektor informasi obyek. Algoritma ini mengasumsikan bahwa atribut obyek adalah independen. Probabilitas yang terlibat data memproses perkiraan akhir dihitung sebagai jumlah frekuensi dari “master” table keputusan (Olson dan Delen, 2008 Dalam Dimas, 2014).

Pengklasifikasian Bayesian memiliki tingkat kesalahan minimal dibandingkan dengan klasifikasi lainnya. Namun, dalam prakteknya hal ini tidak selalu terjadi, karena ketidakakuratan asumsi yang dibuat untuk penggunaannya, seperti kondisi kelas independen, dan kurangnya data probabilitas yang tersedia. Pengklasifikasian Bayesian juga berguna dalam memberikan pembenaran teoritis untuk pengklasifikasian lain yang tidak secara eksplisit menggunakan teorema Bayes.

Menurut Han dan Kamber (2011) Proses dari The Naive Bayesian Classifier atau Simple Bayesian Classifier, sebagai berikut :

1. Variable D menjadi pelatihan set tuple dan label yang terkait dengan kelas. Seperti biasa, setiap tuple diwakili oleh vektor atribut

n-dimensi,  $X = (x_1, x_2, \dots, x_n)$ , ini menggambarkan pengukuran n dibuat pada tuple dari atribut n, masing-masing,  $A_1, A_2, \dots, A_n$ .

2. Misalkan ada kelas  $m$ ,  $C_1, C_2, \dots, C_m$ . Diberi sebuah *tuple*,  $X$ , *classifier* akan memprediksi  $X$  yang masuk kelompok memiliki probabilitas posterior tertinggi, kondisi-disebutkan pada  $X$ . Artinya, *classifier naive bayesian* memprediksi bahwa  $X$  *tuple* milik kelas  $C_i$  jika dan hanya jika :

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

(2.1)

Jadi memaksimalkan  $P(C_i | X)$ .  $C_i$  kelas yang  $P(C_i | X)$  dimaksimalkan disebut hipotesis posteriori maksimal. Dengan teorema Bayes :

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

(2.2)

Keterangan :

$P(C_i|X)$  = Probabilitas hipotesis  $C_i$  jika diberikan fakta atau *record X (Posterior probability)*

$P(X|C_i)$  = mencari nilai parameter yang memberi kemungkinan yang paling besar (*likelihood*)

$P(C_i)$  = Prior probability dari  $X$  (*Prior probability*)  $P(X)$  = Jumlah probability

*tuple* yg muncul

3. Ketika  $P(X)$  adalah konstan untuk semua kelas, hanya  $P(X | C_i) P(C_i)$  butuh dimaksimalkan. Jika probabilitas kelas sebelumnya tidak diketahui, maka umumnya diasumsikan ke dalam kelas yang sama, yaitu,  $P(C_1) = P(C_2) = \dots = P(C_m)$ , maka dari itu akan memaksimalkan  $P(X | C_i)$ . Jika tidak, maka akan memaksimalkan  $P(X | C_i) P(C_i)$ . Perhatikan bahwa probabilitas sebelum kelas dapat diperkirakan oleh  $P(C_i) = |C_i, D| / |D|$ , dimana  $|C_i, D|$  adalah jumlah *tuple* pelatihan kelas  $C_i$  di  $D$ .
4. Mengingat *dataset* mempunyai banyak atribut, maka akan sangat sulit dalam mengkomputasi untuk menghitung  $P(X|C_i)$ . Agar dapat mengurangi perhitungan dalam mengevaluasi  $P(X|C_i)$ , asumsi *naïve* independensi kelas bersyarat dibuat. Dianggap bahwa nilai-nilai dari atribut adalah kondisional independen satu sama lain, diberikan kelas label dari *tuple* (yaitu bahwa tidak ada hubungan ketergantungan diantara atribut) dengan demikian :

$$\begin{aligned}
 P(X|C_i) &= \prod_{k=1}^n P(x_k|C_i) \\
 &= P(x_1|C_i) \times P(x_2|C_i) \times \dots \times P(x_n|C_i).
 \end{aligned}
 \tag{2.3}$$

Maka dapat dengan mudah memperkirakan probabilitas  $P(x_1 | C_i)$ ,  $P(x_2 | C_i)$ , . . . ,  $P(x_n | C_i)$  dari pelatihan *tuple*. Ingat bahwa di sini  $x_k$  mengacu pada nilai atribut  $A_k$  untuk *tuple*  $X$ . Untuk setiap atribut, dilihat dari apakah atribut tersebut

kategorikal atau *continuous-valued* Misalnya, untuk menghitung  $P(X | C_i)$  mempertimbangkan hal-hal berikut:

- a) Jika  $A_k$  adalah kategorikal, maka  $P(X_k | C_i)$  adalah jumlah *tuple* kelas  $C_i$  di  $D$  memiliki nilai  $X_k$  untuk atribut  $A_k$ , dibagi dengan  $|C_i, D|$ , jumlah *tuple* kelas  $C_i$  di  $D$ .
- b) Jika  $A_k$  *continuous-valued*, maka perlu melakukan sedikit lebih banyak pekerjaan, tapi perhitungannya cukup sederhana. Sebuah atribut *continuous-valued* biasanya diasumsikan memiliki distribusi *Gaussian* dengan rata-rata  $\mu$  dan standar deviasi  $\sigma$ , didefinisikan oleh

$$g(x, \mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}, \quad (2.4)$$

sehingga :

$$P(x_k | C_i) = g(x_k, \mu_{C_i}, \sigma_{C_i}). \quad (2.5)$$

Setelah itu hitung  $\mu_{C_i}$  dan  $\sigma_{C_i}$ , yang merupakan deviasi *mean* (rata-rata) dan standar masing-masing nilai atribut  $A_k$  untuk *tuple* pelatihan kelas  $C_i$ . Setelah itu gunakan kedua kuantitas dalam Persamaan, bersama-sama dengan  $x_k$ , untuk memperkirakan

$P(x_k | C_i)$ .

5. Untuk memprediksi label kelas  $x$ ,  $P(X|C_i)P(C_i)$  dievaluasi untuk setiap kelas  $C_i$ . *Classifier* memprediksi kelas label dari *tuple*  $x$  adalah kelas  $C_i$ , jika

$$P(X|C_i)P(C_i) > P(X|C_j)P(C_j) \text{ for } 1 \leq j \leq m, j \neq i. \quad (2.)$$

6) Dengan kata lain, label kelas diprediksi adalah  $C_i$  yang mana  $P(X | C_i) P(C_i)$  adalah maksimal.

### 2.2.1 Contoh Penggunaan Naïve Bayes

Berikut referensi jurnal kasus penerapan algoritma naïve bayes :

1. Menggunakan Metode Naïve Bayes classifier pada Aplikasi Perpustakaan oleh Selvia Lorena Br Ginitn,S.Si, MT , Reggy Pasya Trinanda dari jurusan Teknik Komputer Unikom Bandung.
2. Penerapan Algoritma Naïve Bayes untuk Penentuan Status Turn-Over Pegawai oleh Yeffriansjah Salim dari progdi Sistem Informasi STMIK Indonesia Banjarmasin.
3. Penerapan Algoritma Naïve Bayes untuk Mengklasifikasikan data Nasabah Asuransi oleh Bustami Dosen Teknik Informatika Unviersitas Malikusaleh.
4. Penggunaan Algoritma naïve bayes pada aplikasi Sistem Pakar untuk Menentukan Rumah Tangga Miskin (RTM) oleh Dimas Pamilih dari Jurusan Teknik Informatika STMIK Sinar Nusantara Surakarta.
5. Penggunaan Algoritma naïve bayes pada aplikasi Sistem Pakar untuk Mendiagnosa Penyandang Tunagrhita oleh Afif Fredi Kurniawan jurusan Teknik Infirmatika STMIK SInar Nusantara.

### 2.2.1 Contoh Penghitungan Naïve Bayes

#### 1. Data Cuaca bermain golf

Outlook	Temperature	Humidity	Windy	Play
Sunny	Hot	High	False	No
Sunny	Hot	High	True	No
Overcast	Hot	High	False	Yes
Rainy	Mild	High	False	Yes
Rainy	Cool	Normal	False	Yes
Rainy	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Sunny	Mild	High	False	No
Sunny	Cool	Normal	False	Yes
Rainy	Mild	Normal	False	Yes
Sunny	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
rainy	Mild	High	True	No

**Gambar 2.2 Data Cuaca bermain golf**

#### 2. Data cuaca dengan jumlah dan probabilitas

Outlook			Temperature			Humidity			Windy			Play	
Sun	Yes	No	Hot	Yes	No	High	Yes	No	False	Yes	No	Yes	No
ny	2	3	Hot	2	2	High	3	4	False	6	2	9	5
Overcast	4	0	Mild	4	2	Normal	6	1	True	3	3		
Rainy	3	2	Cool	3	1								
Sunny	2/9	3/5	Hot	2/9	2/5	High	3/9	4/5	False	6/9	2/5	9/1	5/1
Overcast	4/9	0/5	Mild	4/9	2/5	Normal	6/9	1/5	True	3/9	3/5	4	4
Rainy	3/9	2/5	Cool	3/9	1/5								

**Gambar 2.3 Data Cuaca dengan jumlah probabilitas**

#### 3. Kondisi cuaca dilain hari

Outlook	Temperature	Humidity	Windy	Play
Sunny	Cool	High	True	?

**Gambar 2.4 Data Cuaca dilain hari**

4. Menari apakah dengan kondisi cuaca Outlook = sunny, Temperature = cool, Humidity = high, dan Windy = true, akan bermain golf?
5. Mencari Nilai parameter yang member kemungkinan yang paling besar (*likelihood*)
  - *Likelihood of "yes"* = (sunny=yes) (cool=yes) \* (high=yes)\*(true=yes)\*(play=yes)  
 $2/9 * 3/9 * 3/9 * 3/9 * 9/14 = 0.0053$
  - *Likelihood of "no"* = (sunny=no) \* (cool=no) \* (high=no) \* (true=no) \* (play=no)  
 $3/5 * 1/5 * 4/5 * 3/5 * 5/14 = 0.0206$
  - *Probability of yes* = (Likelihood of "yes") / ((Likelihood of "yes") + (Likelihood of "no"))  
 $0.0053 / (0.0053 + 0.0206) = 20.5\%$
  - *Probability of no* = (Likelihood of "no") / ((Likelihood of "yes") + (Likelihood of "no"))  
 $0.0206 / (0.0053 + 0.0206) = 79.5\%$
6. Membandingkan nilai *Probability of yes* dan *Probability of no* .  
 Karena lebih besar *Probability of no* , hasil tidak bermain golf pada hari itu

### 2.3 Personal Hypertext Preprocessor (PHP)

Pada awalnya PHP merupakan singkatan dari personal Home Page tools, yang gunanya untuk memonitor pengunjung web. PHP mula-mula dikembangkan oleh Rasmus Lerdorf. Kemudian istilah PHP mengacu pada

Hypertext Preprocessor. PHP merupakan bahasa berbentuk script yang ditempatkan dalam server dan diproses di server. Hasilnya akan dikirim ke client, tempat pemakai menggunakan browser. PHP dikenal sebagai bahasa scripting, yang menyatu dengan tag-tag HTML, dieksekusi di server, dan digunakan untuk membuat halaman web yang dinamis seperti hanya Active Server Pages (ASP) atau Java Server Pages (JSP). PHP adalah bahasa pemrograman web yang terintegrasi dengan HTML dan berada pada server, PHP banyak dipakai untuk membuat situs web dinamis (Anhar, 2010)

Kelebihan PHP dari bahasa pemrograman lainnya yaitu :

- a. Bahasa pemrograman PHP adalah sebuah bahasa script yang tidak melakukan sebuah kompilasi dalam penggunaannya.
- b. Web Server yang mendukung PHP dapat ditemukan dimana-mana dari mulai apache, IIS, Lighttpd, hingga Xitami dengan konfigurasi yang relative mudah.
- c. Dalam sisi pengembangan lebih mudah, karena banyaknya milis-milis dan developer yang siap membantu dalam pengembangan.
- d. Dalam sisi pemahaman, PHP adalah bahasa scripting yang paling mudah karena memiliki referensi yang banyak.

## 2.4 MySQL

MySQL merupakan software yang tergolong database server dan bersifat Open Source. Open Source menyatakan bahwa Software ini dilengkapi dengan source code (kode yang dipakai untuk membuat MySQL), selain tentu saja bentuk executable -nya atau kode yang dapat dijalankan

secara langsung dalam sistem operasi , dan dapat diperoleh dengan cara mengunduh di Internet secara gratis (Anhar, 2010)

Pada bulan Mei 1996, MySQL versi 1.0 berhasil dirilis namun penggunaanya terbatas 4 orang saja, namun dibulan Oktober ditahun yang sama versi 3.11.0 dilepaskan ke public tapi belum bersifat open source. Bulan Juni 2000, MySQL AB mengumumkan bahwa sejak versi 3.23.19, MySQL adalah merupakan software database yang bebas berlisensi GPL atau General Public License yang open source. Mulanya MySQL hanya berjalan disistem operasi linu namun pada saat MySQL versi 3.22 tahun 1998-1999 sudah tersedia diberbagai platform termasuk windows. Ini terjadi karena MySQL menjadi semakin populer dan dilirik banyak orang karena kestabilan dan kecepatan yang meningkat. Beberapa keunggulan dari MySQL adalah :

- a. Mampu menangani jutaan user dalam waktu bersamaan
- b. Mampu menanggung lebih dari 50.000.000 record.
- c. Sangat cepat mengeksekusi perintah
- d. Memiliki user privilege yang mudah dan efisien

## **2.5. CSS**

CSS merupakan singkatan dari Cascading Stle Sheet. Kegunaannya adalah untuk mengatur tampilan dokumen HTML, contohnya seperti pengaturan jarak antar baris. Teks, warna dan format border bahkan penampilan file gambar. CSS dikembangkan oleh W3C, organisasi yang mengembangkan teknologi internet. Tujuan tak lain untuk mempermudah proses penataan web, perlu diingat CSS hanyalah berupa kumpulan script

yang tujuannya bukan untuk menggantikan HTML, melainkan pelengkap agar dokumen HTML bisa tampil lebih cantik dan dinamis.

Sejak ditemukan CSS pada awal decade 90an, CSS terus dikembangkan dan diserap oleh web developer. Hingga sekarang telah tercapai versi ke-3. Kode CSS bersifat lintas platform, yang berarti script ini dapat dibaca berbagai macam system operasi dan browse. Hanya saja browse seperti Internet Explorer, seringkali salah mengartikan script CSS yang menyebabkan ketidaksempurnaannya tampilan dokumen oleh HTML. Script CSS perlu dioptimalkan agar tampilan maksimal pada browser Internet Explorer (Jayan. 2010)