BAB 2
TINJAUAN PUSTAKA

2.1 Kecerdasan Buatan

Kecerdasan buatan (*Artificial intelligence*) adalah salah satu cabang ilmu pengetahuan yang berhubungan dengan pemanfaatan mesin untuk memecahkan persoalan yang rumit dengan cara yang lebih manusiawi. Hal ini biasanya dilakukan dengan mengikuti karakteristik dan analogi berpikir dari kecerdasan manusia, dan menerapkannya sebagai algoritma yang dikenal oleh komputer.

Semakin pesatnya perkembangan teknologi menyebabkan adanya perkembangan dan perluasan lingkup yang membutuhkan kehadiran kecerdasan buatan. Karakteristik cerdas sudah mulai dibutuhkan di berbagai disiplin ilmu dan teknologi. Kecerdasan Buatan tidak hanya merambah di berbagai disiplin ilmu yang lain. Irisan antara psikologi dan kecerdasan buatan melahirkan sebuah area yang dikenal dengan nama *cognition* & *psycolinguistics*. Irisan antara teknik elektro dengan kecerdasan buatan melahirkan berbagai ilmu seperti pengolahan citra, teori kendali, pengenalan pola dan robotika.

Kecerdasan buatan digunakan untuk menganalisis pemandangan dalam citra dengan perhitungan simbol-simbol yang mewakili isi pemandangan tersebut setelah citra diolah untuk memperoleh ciri khas. Kecerdasan buatan bisa dilihat sebagai tiga kesatuan yang terpadu yaitu persepsi, pengertian dan aksi. Persepsi menerjemahkan sinyal dari dunia nyata dalam citra menjadi simbol-simbol yang lebih sederhana, pengertian memanipulasi simbol-simbol tersebut untuk memudahkan penggalian suatu informasi tertentu, dan aksi menerjemahkan simbol-simbol yang telah dimanipulasi menjadi sinyal lain yang dapat merupakan hasil akhir atau hasil antara sesuai dengan keperluan [7].

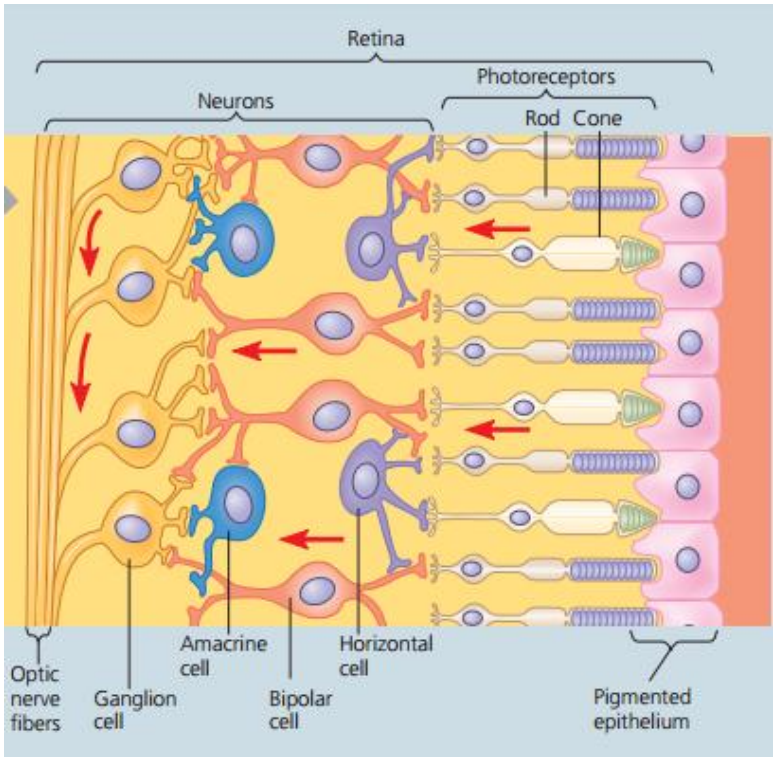
2.2 Retina Mata

Mata adalah organ yang dapat mendeteksi cahaya, secara sederhana mata hanya mendeteksi apakah lingkungan disekitarnya terang atau gelap. Dalam hal yang lebih kompleks mata dapat membedakan bentuk dan warna. Pada mata

manusia, cahaya masuk melalui pupil dan difokuskan pada retina dengan bantuan lensa. Sel-sel saraf sensitif cahaya disebut *rod* (untuk kecerahan) dan *cone* (untuk cahaya) yang beraksi terhadap cahaya. Keduanya berinteraksi satu dengan lainnya dan mengirimkan pesan ke otak yang mengindikasikan kecerahan, warna, kontur.

Retina adalah lapisan terdalam pada mata yang mengandung sel fotoreseptor (*rod* dan *cone*) dan neuron yang berfungsi mentransmisikan bentuk benda yang kita lihat yang dibentuk oleh lensa ke otak melalui saraf optik.

Cahaya masuk kedalam retina melewati sebagian besar lapisan transparan neuron sebelum mencapai rod dan cone, dua jenis fotoreseptor yang berbeda dalam bentuk dan fungsi. Neuron retina kemudian menyampaikan informasi visual yang ditangkap oleh fotoreseptor ke saraf optik dan otak. Setiap sel bipolar memerima informasi dari beberapa *rod* atau *cone* dan setiap ganglion sel mengumpulkan hasil informasi dari beberapa bipolar sel, Horisontal dan Amacrine sel mengintegrasikan informasi di retina. Salah satu daerah retina yaitu optik disk tidak memiliki reseptor sebagai akibatnya daerah ini membentuk "*blind spot*" dimana cahaya tidak terdeteksi [8].



Gambar 2.1 Bagian Pada Retina [8].

2.3 Biometrik

Biometrik adalah suatu cabang keilmuan dengan menggunakan data unik yang terdapat pada anggota tubuh atau tingkah laku manusia untuk tujuan identifikasi. Teknologi biometrik sangat berguna untuk mencegah pemalsuan identitas, karena menggunakan anggota tubuh atau tingkah laku manusia sebagai identitas seseorang tersebut. Anggota tubuh atau tingkah laku manusia yang akan dijadikan untuk sistem biometrik harus memenuhi beberapa syarat diantaranya adalah :

a. *Universality*, bahwa setiap orang memiliki karakteristik.
b. *Uniqueness*, tidak ada kesamaan karakteristik.
c. *Permanence*, bahwa karakteristik konstan dengan waktu.
d. *Collectability*, karakteristik dapat diukur secara kuantitatif.
e. *Performance*, keakuratan dalam mengidentifikasi karakteristik.
f. *Acceptability*, sistem biometrik ini dapat diterima.
g. *Circumvention*, sistem tidak mudah untuk dicurangi.

Anggota tubuh atau tingkah laku yang bisa digunakan untuk sistem biometrik ini diantaranya adalah Wajah, Sidik jari, Geometri tangan, *Keystrokes*, Vena tangan, Iris mata, Retina mata, Tanda tangan, Suara, *Thermograms*, Bau, DNA, Gaya berjalan, dan Telinga. Berikut adalah tabel tingkat kualitas anggota tubuh atau tingkah laku pada sistem biometrik:

Tabel 2.1 Perbandingan Kualitas Biometrik [9]

Biometrik	Universality	Uniqueness	Permanence	Collectability	Performance	Acceptability	Circumvention
Wajah	High	Low	Medium	High	Low	High	Low
Sidik Jari	Medium	High	High	Medium	High	Medium	High
Geometri Tangan	Medium	Medium	Medium	High	Medium	Medium	Medium
Keystrokes	Low	Low	Low	Medium	Low	Medium	Medium
Vena Tangan	Medium	Medium	Medium	Medium	Medium	Medium	High
Iris Mata	High	High	High	Medium	High	Low	High
Retina Mata	High	High	Medium	Low	High	Low	High
Tanda Tangan	Low	Low	Low	High	Low	High	Low

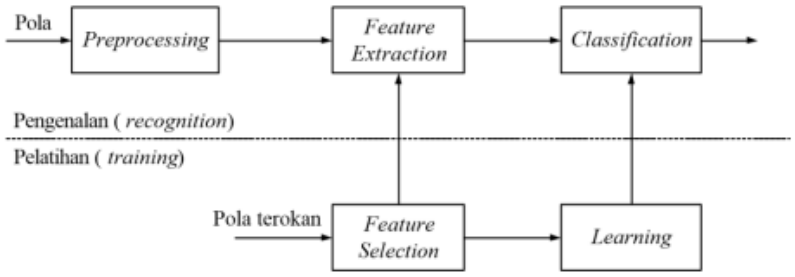
Suara	**Medium**	**Low**	**Low**	**Medium**	**Low**	**High**	**Low**
Thermograms	**High**	**High**	**Low**	**High**	**Medium**	**High**	**High**
Bau	**High**	**High**	**High**	**Low**	**Low**	**Medium**	**Low**
DNA	**High**	**High**	**High**	**Low**	**High**	**Low**	**Low**
Gaya Berjalan	**Medium**	**Low**	**Low**	**High**	**Low**	**High**	**Medium**
Telinga	**Medium**	**Medium**	**High**	**Medium**	**Medium**	**High**	**Medium**

2.4 Pengenalan Pola

Pengenalan pola adalah membedakan suatu objek dengan objek lain. Pengenalan pola bertujuan untuk menentukan kelompok atau kategori pola berdasarkan ciri-ciri yang dimiliki oleh pola tersebut. Terdapat dua pendekatan yang dilakukan dalam pengenalan pola yaitu pendekatan secara statistik dan pendekatan secara sintaktik atau struktural [10].

2.4.1 Pengenalan Pola Secara Statistik

Pengenalan pola dengan pendekatan statistik ini menggunakan teori-teori dalam ilmu peluang dan statistik. . Ciri yang dimiliki oleh suatu pola ditentukan distribusi statistiknya, pola yang berbeda memiliki distribusi yang berbeda pula. Dengan menggunakan teori keputusan di dalam statistik, kita menggunakan distribusi ciri untuk mengklasifikasikan pola. Sistem pengenalan pola dengan pendekatan statistik ditunjukkan pada gambar 2.2.



Gambar 2.2 Pengenalan Pola Dengan Pendekatan Statistik [10].

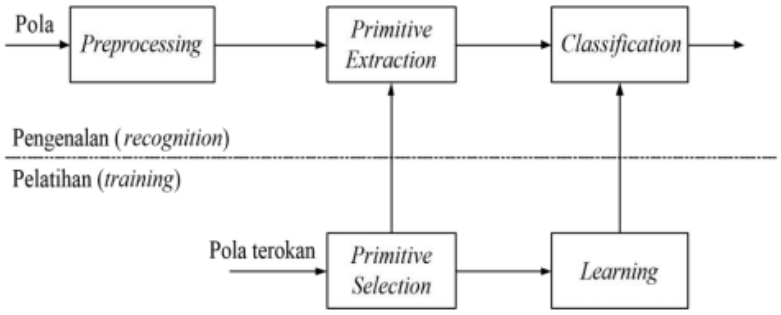
Ada dua fase dalam sistem pengenalan pola yaitu fase pelatihan dan fase pengenalan. Pada fase pelatihan, beberapa contoh citra dipelajari untuk menentukan ciri yang akan digunakan dalam proses pengenalan serta prosedur

klasifikasinya. Pada fase pengenalan, citra diambil cirinya kemudian ditentukan kelas kelompoknya [10].

Pada penelitian ini termasuk sistem pengenalan pola dengan pendekatan statistik, karena ciri-ciri yang dimiliki oleh citra retina mata memiliki pola yang ditentukan distribusi statistiknya. Apabila polanya berbeda maka memiliki distribusi yang berbeda pula. Distribusi ciri digunakan untuk mengklasifikasikan pola dengan memanfaatkan teori keputusan di dalam statistik.

2.4.2 Pengenalan Pola Secara Sintaktik

Pengenalan pola dengan pendekatan sintaktik menggunakan teori bahasa formal. Ciri yang terdapat pada suatu pola ditentukan primitif dan hubungan struktural antara primitif kemudian menyusun tata bahasanya. Dari aturan produksi pada tata bahasa tersebut kita dapat menentukan kelompok pola. Pengenalan pola secara sintaktik lebih dekat ke strategi pengenalan pola yang dilakukan manusia, namun secara praktek penerapannya relatif sulit dibandingkan pengenalan pola secara statistik [10]. Sistem pengenalan pola dengan pendekatan sintaktik ditunjukkkan pada gambar 2.3.



Gambar 2.3 Pengenalan Pola Dengan Pendekatan Sintaktik [10].

2.5 Pengolahan Citra

Pengolahan citra adalah pemrosesan citra, khususnya dengan menggunakan komputer, menjadi citra yang kualitasnya lebih baik. Umumnya, operasi –operasi pada pengolahan citra diterapkan pada citra bila :

1. Perbaikan atau memodifikasi citra perlu dilakukan untuk meningkatkan kualitas penampakan atau untuk menonjolkan beberapa aspek informasi yang terkandung di dalam citra.
2. Elemen di dalam citra perlu dikelompokkan, dicocokkan, atau diukur.
3. Sebagian citra perlu digabung dengan bagian citra yang lain.

Pengolahan citra bertujuan memperbaiki kualitas citra agar mudah diinterpretasi oleh manusia atau mesin (dalam hal ini komputer). Teknik-teknik pengolahan citra mentransformasikan citra menjadi citra lain. Jadi, masukannya adalah citra dan keluarannya juga citra, namun citra keluaran mempunyai kualitas lebih baik daripada citra masukan [10].

2.6 Operasi-operasi Pengolahan Citra

Operasi-operasi yang dilakukan di dalam pengolahan citra banyak ragamnya. Namun, secara umum, operasi pengolahan citra dapat diklasifikasikan dalam beberapa jenis sebagai berikut [10]:

1. Perbaikan kualitas citra (*image enhancement*).

   Jenis operasi ini bertujuan untuk memperbaiki kualitas citra dengan cara memanipulasi parameter-parameter citra. Dengan operasi ini, ciri-ciri khusus yang terdapat di dalam citra lebih ditonjolkan. Contoh operasi perbaikan citra diantaranya adalah perbaikan kontras, perbaikan tepian objek (*edge enhancement*), penajaman (*sharpening*), pemberian warna semu (*pseudocoloring*), dan penapisan derau (*noise filtering*).
2. Pemugaran citra (*image restoration*).

   Operasi ini bertujuan menghilangkan / meminimumkan cacat pada citra. Tujuan pemugaran citra hampir sama dengan operasi perbaikan citra. Bedanya, pada pemugaran citra penyebab degradasi gambar diketahui. Contoh operasi pemugaran citra diantaranya adalah penghilangan kesamaran (*deblurring*) dan penghilangan derau (*noise*).
3. Pemampatan citra (*image compression*).

   Jenis operasi ini dilakukan agar citra dapat direpresentasikan dalam bentuk yang lebih kompak sehingga memerlukan memori yang lebih sedikit. Hal

penting yang harus diperhatikan dalam pemampatan adalah citra yang telah dimampatkan harus tetap mempunyai kualitas gambar yang bagus.

4. Segmentasi citra (*image segmentation*).

   Jenis operasi ini bertujuan untuk memecah suatu citra ke dalam beberapa segmen dengan suatu kriteria tertentu. Jenis operasi ini berkaitan erat dengan pengenalan pola.

5. Pengorakan citra (*image analysis*)

   Jenis operasi ini bertujuan menghitung besaran kuantitif dari citra untuk menghasilkan deskripsinya. Teknik pengorakan citra mengekstraksi ciri-ciri tertentu yang membantu dalam identifikasi objek. Proses segmentasi kadangkala diperlukan untuk melokalisasi objek yang diinginkan dari sekelilingnya. Contoh operasi pengorakan citra diantaranya adalah pendeteksian tepi objek (*edge detection*), ekstraksi batas (*boundary*), dan representasi daerah (*region*).

6. Rekonstruksi citra (*image reconstruction*)

   Jenis operasi ini bertujuan untuk membentuk ulang objek dari beberapa citra hasil proyeksi. Operasi rekonstruksi citra banyak digunakan dalam bidang medis. Misalnya beberapa foto rontgen dengan sinar X digunakan untuk membentuk ulang gambar organ tubuh.

7. Perubahan Model Warna

   Warna adalah persepsi yang dirasakan oleh sistem visual manusia terhadap panjang gelombang cahaya yang dipantulkan oleh objek [7]. Setiap warna mempunyai panjang gelombang yang berbeda. Warna merah mempunyai panjang gelombang paling tinggi, sedangkan warna ungu mempunyai panjang gelombang paling rendah. Warna-warna yang diterima oleh mata merupakan hasil kombinasi cahaya dengan panjang gelombang berbeda. Penelitian memperlihatkan bahwa kombinasi warna yang memberikan rentang warna yang paling lebar adalah red (R), green (G), blue (B).

   Dalam pembentukan citra digital model warna yang umum digunakan antara lain sebagai berikut :

a. Citra RGB

Citra RGB yang biasa disebut juga citra true color, disimpan dalam citra berukuran (m x n) x 3 yang mendefinisikan warna merah (red), hijau (green) dan biru (blue) untuk setiap pikselnya. Warna pada setiap piksel ditentukan berdasarkan kombinasi dari warna red, green dan blue (RGB). RGB merupakan citra 24 bit dengan komponen merah, hijau, dan biru yang masing-masing umumnya bernilai 8 bit sehingga intensitas kecerahan warna sampai 256 level dan kombinasi warnanya kurang dari sekitar 16 juta warna.

b. Citra Keabuan

Citra dengan derajat keabuan berbeda dengan citra RGB, citra ini didefinisikan oleh satu nilai derajat warna. Umumnya bernilai 8 bit sehingga intensitas kecerahan warna sampai 256 level dan kombinasi warnanya 256 varian. Tingkat kecerahan paling rendah yaitu 0 untuk warna hitam dan putih bernilai 255. Untuk mengkonversikan citra yang memiliki warna RGB ke derajat keabuan bisa menggunakan persamaan :

$$Gray = 0.30 * R + 0.59 * G + 0.11 * B \quad (2.1)$$

Dimana :

R = nilai *Red*

G = nilai *Green*

B = nilai *Blue*

2.7 Tekstur

Secara umum tekstur mengacu pada repetisi elemen-elemen tekstur dasar yang sering disebut primitif atau *texel* (*texture element*). Suatu *texel* terdiri dari beberapa *pixel* dengan aturan posisi bersifat periodik, kuasiperiodik, atau acak. Pengertian dari tekstur dalam hal ini adalah keteraturan pola-pola tertentu yang terbentuk dari susunan piksel-piksel dalam citra digital [7].

Untuk membentuk suatu tekstur setidaknya ada dua persyaratan yang harus dipenuhi antara lain :

1. Terdiri dari satu atau lebih piksel yang membentuk pola-pola primitif (bagian-bagian terkecil). Bentuk-bentuk pola primitif ini dapat berupa titik, garis lurus, garis lengkung, luasan dan lain-lain yang merupakan elemen dasar dari sebuah bentuk.
2. Munculnya pola-pola primitif yang berulang-ulang dengan interval jarak dan arah tertentu sehingga dapat diprediksi atau ditemukan karakteristik perulangannya, untuk contoh dari beberapa citra tekstur dapat dilihat dari gambar 2.4.



Gambar 2.4 Contoh Citra Tekstur [7]

Suatu citra memberikan interpretasi tekstur yang berbeda apabila dilihat dengan jarak dan sudut yang berbeda, manusia memandang tekstur berdasarkan deskripsi yang bersifat acak, seperti halus, kasar, teratur, tidak teratur, dan lain sebgainya. Hal ini merupakan deskripsi yang tidak tepat dan non-kuantitatif, sehingga diperlukan adanya suatu deskripsi yang kuantitatif (matematis) untuk memudahkan analisis [7].

2.8 Analisis Tekstur

Analisis tekstur merupakan dasar dari berbagai macam aplikasi, aplikasi dari analisis tekstur antara lain adalah penginderaan jarak jauh, pencitraan medis, identifikasi kualitas suatu bahan (kayu, kulit, tekstil dan lain-lain). Pada analisis citra, dikategorikan menjadi lima kategori utama yaitu statistis, struktural, geometri, model dasar dan pengolahan sinyal. Pendekatan statistis mempertimbangakan bahwa internsitas dibangkitkan oleh medan acak dua dimensi, metode ini berdasar pada frekuensi-frekuensi ruang. Contoh metode statistis adalah fungsi autokorelasi, run-length, matriks kookurensi, tranformasi

fourier, frekuensi tepi. Teknik struktural berkaitan dengan penyusupan bagian-bagian terkecil suatu citra. contoh metode struktural adalah model fraktal. Metode geometri berdasar atas perangkat geometri yang ada pada elemen tekstur. Contoh model dasar adalah medan acak. Sedangkan metode pengolahan sinyal adalah metode yang berdasarkan analisis frekuensi seperti transformasi gabor dan transformasi wavelet [7].

2.9 Metode Run Length

Grey level run length matrix yang biasa disingkat dengan GLRLM merupakan salah satu metode untuk mengekstrak tekstur sehingga diperoleh ciri statistik atau atribut yang terdapat dalam tekstur dengan mengestimasi piksel-piksel yang memiliki derajat keabuan yang sama. Ekstraksi tekstur dengan metode run-length dilakukan dengan membuat rangkaian pasangan nilai (*i*,*j*) pada setiap baris piksel. Perlu diketahui maksud dari run-length itu sendiri adalah jumlah piksel berurutan dalam arah tertentu yang memiliki derajat keabuan/nilai intensitas yang sama. Jika diketahui sebuah matriks run-length dengan elemen matriks $q(i, j \mid \theta)$ dimana i adalah derajat keabuan pada masing-masing piksel, j adalah nilai run-length, dan θ adalah orientasi arah pergeseran tertentu yang dinyatakan dalam derajat. Orientasi dibentuk dengan empat arah pergeseran dengan interval 45^0, yaitu 0^0, 45^0, 90^0, dan 135^0.

Terdapat beberapa jenis ciri tekstur yang dapat diekstraksi dari matriks run-length [11]. Berikut variabel-variabel yang terdapat di dari ekstraksi citra dengan menggunakan metode statistikal *Grey Level Run Length Matrix* :

i = nilai derajat keabuan

j = piksel yang berurutan (*run*)

M = Jumlah derajat keabuan pada sebuah gambar

N = Jumlah piksel berurutan pada sebuah gambar

r(j) = Jumlah piksel berurutan berdasarkan banyak urutannya (*run length)*

g(i) = Jumlah piksel berurutan berdasarkan nilai derajat keabuannya

s = Jumlah total nilai *run* yang dihasilkan pada arah tertentu

p(i,j) = himpunan matrik i dan j

n = jumlah baris * jumlah kolom.

Dimana varibel-variabel tersebut akan digunakan untuk mencari nilai dari atribut-atribut tekstur sebagai berikut:

1. *Short Run Emphasis* (SRE)

   SRE mengukur distribusi short run. SRE sangat tergantung pada banyaknya short run dan diharapkan bernilai besar pada tekstur halus.

$$SRE = \sum_{i=1}^{M}\sum_{j=1}^{N}\frac{p(i,j)/s}{j2} = \sum_{j=1}^{n}\frac{r(j)/s}{j2} \quad (2.2)$$

2. *Long Run Emphasis* (LRE)

   LRE mengukur distribusi long run. LRE sangat bergantung pada banyaknya long run dan diharapkan bernilai besar pada tekstur kasar.

$$LRE = \sum_{i=1}^{M}\sum_{j=1}^{N} j2p(i,j)/s = \sum_{j=1}^{N} r(j)j2/s \quad (2.3)$$

3. *Grey Level Uniformity* (GLU)

   GLU mengukur persamaan nilai derajat keabuan seluruh citra dan diharapkan bernilai kecil jika nilai derajat keabuan serupa diseluruh citra.

$$GLU = \sum_{i=1}^{M}(\sum_{J=1}^{N} p(i,j))2/s = \sum_{i=1}^{M}\frac{g(i)2}{s} \quad (2.4)$$

4. *Run Length Uniformity* (RLU)

   RLU mengukur persamaan panjangnya run diseluruh citra dan diharapkan bernilai kecil jika panjangnya run serupa diseluruh citra.

$$RLU = \sum_{j=1}^{N}(\sum_{I=1}^{M} p(i,j))2/s = \sum_{J=1}^{N}\frac{r(j)2}{s} \quad (2.5)$$

5. *Run Percentage* (RPC)

RPC mengukur kebersamaan dan distribusi run dari sebuah citra pada arah tertentu. RPC bernilai paling besar jika panjangnya run adalah 1 untuk semua derajat keabuan pada arah tertentu.

$$RPC = \sum_{i=1}^{M}\sum_{J=1}^{N} p(i,j)/n = \sum_{J=1}^{N} r(j)/n \quad (2.6)$$

2.10 Metode Naive Bayesian

Naïve bayesian adalah suatu metode pengklasifikasian paling sederhana dengan menggunakan peluang yang ada, dimana diasumsikan bahwa setiap variable X bersifat bebas (*independence*) [4]. Karena asumsi variabel tidak saling terikat, maka didapatkan :

$$P(X \mid Y = y) = \prod_{i=1}^{q} P(X_i \mid Y = y) \quad (2.7)$$

Terdapat beberapa langkah dalam pengklasifikasian menggunakan metode naive bayesian, berikut adalah langkah - langkahnya :

Training :

1. Hitung rata-rata (*mean*) tiap fitur dalam dataset training dengan.

   $$\mu = \frac{\sum x_i}{n} \quad (2.8)$$

   Dimana:

   μ = mean

   n = banyaknya data

   $\sum x_i$ = jumlah nilai data

2. Kemudian hitung nilai varian dari dataset training tersebut seperti pada.

   $$\sigma^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \mu)^2 \quad (2.9)$$

   Dimana:

   σ^2= varians

μ= mean

x_i = nilai data

n = banyaknya data

Testing :

1. Hitung probabilitas (*Prior*) tiap kelas yang ada dengan cara menghitung jumlah data tiap kelas dibagi jumlah total data secara keseluruhan.
2. Selanjutnya menghitung densitas probabilitasnya. Fungsi densitas mengekspresikan probabilitas relatif. Data dengan mean μ dan standar deviasi σ, fungsi densitas probabilitasnya adalah :

$$\varphi_{\mu\sigma}(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad (2.10)$$

Dimana :

x = data masukan

π = 3,14

σ = standar deviasi

μ = mean

3. Setelah didapatkan nilai densitas probabilitasnya, selanjutnya menghitung posterior masing-masing kelas dengan menggunakan persamaan.

$$posterior = \frac{prior\ x\ likehood}{evidence} \quad (2.11)$$

Atau bisa ditulis

$$Posterior = \frac{P(C_i)x\ P(X1 \ldots\ldots\ldots\ldots\ldots\ldots Xn|C_i)}{P(X1 \ldots\ldots\ldots\ldots\ldots\ldots Xn|C_i)} \quad (2.12)$$

4. Setelah didapat nilai posterior masing-masing kelas, maka kelas yang sesuai untuk data masukan adalah kelas yang memiliki nilai posterior terbesar.

2.11 Pengujian *Confusion Matrix*

Pengujian yang dilakukan pada metode klasifikasi terdapat pada bagian akurasi dari hasil klasifikasi. Akurasi sebuah klasifikasi berpengaruh terhadap performa dari suatu metode klasifikasi. Untuk melakukan pengujian akurasi dapat digunakan *confusion matrix* yaitu sebuah matriks dari prediksi yang akan dibandingkan dengan kelas yang asli dari data masukan. Setiap kolom dari

matriks berkorespondensi kepada hasil klasifikasi dan setiap baris pada masukan. Akurasi sebuah klasifikasi dimana i=j menerangkan akurasi dari klasifikasi pada setiap kelas [12]. Berikut contoh *Confusion Matrix* dapat dilihat pada tabel 2.2.

Tabel 2.2 *Confusion Matrix*

Kelas		Hasil Klasifikasi	
		0	1
Target	0	00	01
	1	10	11

Untuk menghitung akurasinya digunakan formula :

$$\frac{jumlah\ akurasi}{total\ data} X\ 100\% = Akurasi\% \quad (2.13)$$

2.12 Pemrograman Berorientasi Objek

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 berorientasi objek merupakan suatu cara bagaimana sistem perangkat lunak dibangun melalui pendekatan objek secara sistematis. Metode berorientasi objek didasarkan pada penerapann prinsip-prinsip pengelolaan kompleksitas. Metode berorientasi objek meliputi rangkaian aktifitas analisis beorientasi objek, perancangan berorientasi objek, pemrograman berorientasi objek, dan pengujian berorientasi objek [13].

Pada saat ini, metode berorientais objek banyak dipilih karena metodologi lama banyak menimbulkan masalah seperti adanya kesulitan pada saat mentranformasi hasil dari satu tahap pengembangan ke tahap berikutnya, misalnya pada metode pendekatan terstruktur, jenis aplikasi yang dikembangkan saat ini berbeda dengan masa lalu. Aplikasi yang dikembangkan saat ini sangat beragam (aplikasi bisnis, *real-time*, *utility* dan sebagainya) dengan platform yang berbeda-beda, sehingga menimbulkan tuntutan kebutuhan metodologi pengembangan yang dapat mengakomodasi ke semua jenis aplikasi.

Keuntungan menggunakan metodologi berorientasi objek adalah sebagai berikut :

a. Meningkatkan Produktivitas

   Karena kelas dan objek yang ditemukan dalam suatu masalah masih dapat dipakai ulang untuk masalah lainnya yang melibatkan objek tersebut (*reuseable*).

b. Kecepatan Pengembangan

   Karena sistem yang dibangun dengan baik dan benar pada saat analisis dan perancangan akan mennyababkan berkurangnya kesalahan pada saat pengkodean

c. Kemudahan Pemeliharaan

   Karena dengan model objek, pola-pola yang cendrung tetap dan stabil dapat dipisahkan dan pola-pola yang mungkin sering diubah-ubah.

d. Adanya Konsistensi

   Karena sifat pewarisan dan penggunaan notasi yang sama pada saat analisis, perancangan maupun pengkodean.

e. Meningkatkan Kualitas Perangkat Lunak

   Karena adanya pendekatan pengembangan lebih dekat dengan dunia nyata dan adanya konsistensi pada saat pengambangannya, perangkat lunak yang dihasilkan akan mampu memenuhi kebutuhan pemakai serta mempunyai sedikit kesalahan.

Berikut beberapa contoh bahasa pemograman yang mendukung pemrograman berorientasi objek adalah :

a. Smalltalk

   Smalltalk adalah salah satu bahasa pemograman yang diekmbangkan untuk mendukung pemrograman beroirentasi objek.

b. Bahasa Pemrograman Eiffel

   Eiffel merupakan bahsa pemrograman yang kembangkan untuk mendukung pemrograman berorientasi objek oleh Bertrand Meyer dan *compiler*.

c. Bahasa Pemrograman (Web) PHP

Php dibuat pertama kali oleh seorang perekayasa perangkat (software engineering) yang bernama Rasmus Lerdoff.

d. Bahasa Pemrograman C++

C++ merupakan pengembangan lebih lanjut dari bahasa pemrograman C untuk mendukung pemrograman berorientasi objek.

e. Bahasa Pemrograman Java

Java dikembangkan oleh perusahaan Sun Microsystem. Java menurut definisi dari Sun Microsystem adalah nama untuk sekumpulan teknologi untuk membuat dan menjalankan perangkat lunak pada komputer *standalone* ataupun pada lingkungan jaringan.

2.12.1 Konsep Dasar Pemrograman Berorientasi Objek

Berikut adalah konsep dasar pemrograman berorientasi objek :

a. Objek (*Object*)

Objek adalah abtraksi dan sesuatu yang mewakili dunia nyata seperti benda, satuan organisasi, tempat, kejadian, struktur, status, atau hal-hal lain yang bersifat abstrak. Objek merupakan suatu entitas yang mampu menyimpan informasi dan mempunyai operasi yang dapat diterapkan atau dapat berpengaruh pada status objeknya.

b. Kelas (*Class*)

Kelas adalah kumpulan objek-objek dengan karakteristik yang sama. Kelas merupakan definisi statik dan himpunan objek yang sama yang mungkin lahir atau diciptakan dalam kelas tersebut.

c. Pembungkusan (*Encapsulation*)

Pembungkusan atribut data dan layanan yang mempunyai objek untuk menyembunyikan implementasi dan objek sehingga objek lain tidak mengetahui cara kerjanya.

d. Pewarisan (*Inheritance*) dan Generalisasi/Spesialisasi

Mekanisme yang memungkinkan satu objek mewarisi sebagian atau seluruh definisi dan objek lain sebagai bagian dirinya.

e. Metode

Operasi atau metode pada sebuah kelas hampir sama dengan fungsi atau prosedur pada metodologi struktural.

f. *Polimorfisme*

Kemampuan suatu objek untuk digunakan dibanyak tujuan yang berbeda dengan nama yang sama sehingga menghemat baris program.

2.12.2 UML (Unified Modelling Language)

Unified Modelling Language (UML) adalah sekumpulan spesifikasi yang dikeluarkan oleh OMG. UML terbaru adalah UML 2.3 yang terdiri dari 4 macam spesifikasi, yaitu : *Diagram Interchange Spesification,* UML *Infrastrukture,* UML *Superstrukture, dan Object Constraint Language* (OCL). Pada UML 2.3 terdiri 13 macam diagram yang dikelompokan pada 3 kategori, yaitu [14] :

A. Struktur Diagram, yaitu kumpulan diagram yang digunakan untuk menggambarkan suatu struktur statis dari sistem yang dimodelkan. Pada Struktur Diagram dibagi menjadi 6 bagian :

1. Diagam Kelas

   Diagram kelas menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut attribut dan metode atau operasi**.**

2. Diagram Objek

   Diagram objek menggambarkan struktur sistem dari segi penamaan objek dan jalannya objek dalam sistem.

3. Diagram Komponen

   Diagram komponen dibuat untuk menunjukan organisasi dan ketergantungan diantara kumpulan komponen dalam sebuah sistem.

4. *Composite Structure Diagram*

   Composite Structure Diagram baru mulai ada pada UML versi 2.0. diagram ini dapat digunakan untuk menggambarkan struktur dari bagian-bagian yang saling terhubung maupun mendeskripsikan struktur pada saat berjalan (*runtime*) .

5. *Package Diagram*

   Package diagram menyediakan cara mengumpulkan elemen-elemen yang saling terkait dalam diagram UML. Hampir semua diagram dalam UML dapat dikelompokan menggunakan *Package Diagram*.

6. *Deployment Diagram*

   Deployment diagram menunjukan konfigurasi komponen dalam proses eksekusi aplikasi.

B. *Behavior Diagram,* yaitu kumpulan diagram yang digunakan untuk menggambarkan kelakuan sistem atau rangkaian perubahan yang terjadi pada sebuah sistem. Pada *Behavior Diagram* dibagi menjadi 3 bagian :

1. *Use Case Diagram*

   Use case diagram merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat,. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat.

2. *Activity Diagram*

   Activity diagram menggambarkan *workflow* atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak.

3. *State Machine Diagram*

   State machine diagram digunakan untuk menggambarkan perubahan status atau transisi status dari sebuah mesin atau sistem atau objek.

C. *Interactions Diagram*, yaitu kumpulan diagram yang digunakan untuk menggambarkan interaksi antar subsistem pada suatu sistem. Pada Interactions Diagram dibagi menjadi 4 bagian :

1. *Sequence Diagram*

   Sequence diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan message yang dikirimkan dan diterima antar objek.

2. *Communication Diagram*

Communication diagram menggambarkan interaksi antar bojek/bagian dalam bentuk urutan pengiriman pesan. Diagram komunikasi merepresentasikan informasi yang diperoleh dari diagram kelas, diagram *sequence*, dan diagram *use case* untuk mendeskripsikan gabungan antara struktur statis dan tingkah laku dinamis dari suatu sistem.

3. *Timing Diagram*

   Timing diagram merupakan diagram yang fokus pada penggambaran terkait batas waktu.

4. *Interaction Overview Diagram*

   Interaction overview diagram mirip dengan diagram aktivitas yang berfungsi untuk menggarbarkan sekumpulan urutan aktivitas, diagram ini adalah bentuk aktivias diagram yang setiap titik merepresentasikan diagram interaksi.

2.12.3 Database

Istilah "basis data" berawal dari ilmu komputer. Meskipun kemudian artinya semakin luas, memasukkan hal-hal di luar bidang elektronika, artikel ini mengenai basis data komputer. Catatan yang mirip dengan basis data sebenarnya sudah ada sebelum revolusi industri yaitu dalam bentuk buku besar, kuitansi dan kumpulan data yang berhubungan dengan bisnis.

Konsep dasar dari basis data adalah kumpulan dari catatan-catatan, atau potongan dari pengetahuan. Sebuah basis data memiliki penjelasan terstruktur dari jenis fakta yang tersimpan di dalamnya: penjelasan ini disebut skema. Skema menggambarkan obyek yang diwakili suatu basis data, dan hubungan di antara obyek tersebut. Ada banyak cara untuk mengorganisasi skema, atau memodelkan struktur basis data: ini dikenal sebagai model basis data atau model data. Model yang umum digunakan sekarang adalah model relasional, yang menurut istilah layman mewakili semua informasi dalam bentuk tabel-tabel yang saling berhubungan dimana setiap tabel terdiri dari baris dan kolom (definisi yang sebenarnya menggunakan terminologi matematika). Dalam model ini, hubungan antar tabel diwakili dengan menggunakan nilai yang sama antar tabel. Model yang

lain seperti model hierarkis dan model jaringan menggunakan cara yang lebih eksplisit untuk mewakili hubungan antar tabel. Istilah *basis data* mengacu pada koleksi dari data-data yang saling berhubungan, dan perangkat lunaknya seharusnya mengacu sebagai *sistem manajemen basis data* (*database management system/DBMS*). Jika konteksnya sudah jelas, banyak administrator dan programer menggunakan istilah basis data untuk kedua arti tersebut [15].

2.12.3.1 MySQL

MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL atau DBMS (*Database Management System*) yang *multithread*, *multi-user*, dengan sekitar 6 juta instalasi di seluruh dunia. MySQL AB membuat MySQL tersedia sebagai perangkat lunak gratis di bawah lisensi GNU *General Public License* (GPL), tetapi mereka juga menjual dibawah lisensi komersial untuk kasus-kasus dimana penggunaannya tidak cocok dengan penggunaan GPL.

Tidak seperti PHP atau Apache yang merupakan software yang dikembangkan oleh komunitas umum, dan hak cipta untuk kode sumber dimiliki oleh penulisnya masing-masing, MySQL dimiliki dan disponsori oleh sebuah perusahaan komersial Swedia yaitu MySQL AB. MySQL AB memegang penuh 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 [15].

Fitur-fitur MySQL antara lain :

1. Relational Database System. Seperti halnya software database lain yang ada di pasaran, MySQL termasuk RDBMS.
2. Arsitektur Client-Server. MySQL memiliki arsitektur client-server dimana server database MySQL terinstal di server. Client MySQL dapat berada di komputer yang sama dengan server, dan dapat juga di komputer lain yang berkomunikasi dengan server melalui jaringan bahkan internet.
3. Mengenal perintah SQL standar. SQL (Structured Query Language) merupakan suatu bahasa standar yang berlaku di hampir semua software database. MySQL mendukung SQL versi SQL:2003.

4. Mendukung Sub Select. Mulai versi 4.1 MySQL telah mendukung select dalam select (sub select).
5. Mendukung Views. MySQL mendukung views sejak versi 5.0
6. Mendukung Stored Prosedured (SP). MySQL mendukung SP sejak versi 5.0
7. Mendukung Triggers. MySQL mendukung trigger pada versi 5.0 namun masih terbatas. Pengembang MySQL berjanji akan meningkatkan kemampuan trigger pada versi 5.1.
8. Mendukung *replication*.
9. Mendukung transaksi.
10. Mendukung foreign key.
11. Tersedia fungsi GIS.
12. Gratis (bebas didownload)
13. Stabil dan tangguh
14. Fleksibel dengan berbagai pemrograman
15. *Security* yang baik, dukungan dari banyak komunitas
16. Perkembangan software yang cukup cepat.

2.12.4 Bahasa Pemograman C#

C# (tanda '#' dibaca "Sharp") merupakan bahasa pemograman baru yang diciptakan Microsoft secara khusus sebagai salah satu bahasa pemrograman dalam teknologi .Net sebagai bahasa baru, C# tidak berevolusi dari bahasa C# versi bukan teknologi .Net. dengan demikian C# dapat memaksimalkan kemampuannya tanpa khawatir dengan masalah kompatibilitas dengan versi-versi sebelumnya. Keharusan sebuah perangkat lunak untuk tetap dapat kompatibel dengan versi-versi sebelumnya sebagaimana yang terjadi pada Visual Basic (VB) maupun C++biasanya menghambat optimalitas kemampuan dari perangkat lunak tersebut [16].

Sejak diluncurkan pada tahun 2000, C# dengan cepat merebut hati progammer C++ bahkan VB. Dengan tata cara penulisan yang mirip C++ dan *interface* mirip VB 6.0 menurut wikipedia, sebuah ensiklopedia gratis di internet pengguna C# .Net pada saat ini sudah melebihi pengguna VB.Net. sementara itu

jumlah pengguna bahasa pemrograman lain masih berada dibawah jumlah pengguna VB .Net. masih menurut wikipedia, jumlah buku C# yang terjual pun berada dikisaran 2 hingga 3 kali lebih banyak dari jumlah buku VB yang terjual. Dari informasi ini dapat disimpulkan bahwa C# merupakan bahasa pemrogrman baru yang sedang berkembang dan dapat diterima dengan baik oleh kebanyakan progammer dan kalangan industri. Di Microsoft sendiri, C# merupakan bahasa pemrograman yang digunakan untuk membuat perangkat lunak yang berteknologi .Net dengan demikian dapat diperkirakan bahwa C# akan menjadi bahasa pemrograman yang akan banyak digunakan di masa-masa mendatang [16].