BAB 2

TINJAUAN PUSTAKA

2.1 Kecerdasan Buatan

Kecerdasan buatan (*Artificial Inteligence*) 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/mencontoh 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 tadi 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.

2.2 Pola

Pola adalah suatu entitas yang terdefinisi dan dapat diidentifikasi serta diberi nama melalui ciri-cirinya (*feature*). Ciri-ciri tersebut digunakan untuk membedakan suatu pola dengan pola yang lainnya. Ciri yang baik adalah ciri yang memiliki daya

pembeda yang tinggi, sehingga pengelompokan pola berdasarkan ciri yang dimiliki dapat dilakukan dengan keakuratan yang tinggi.

Pola adalah komposit/gabungan dari ciri yang merupakan sifat dari sebuah objek[6]. Beberapa contoh pola.

1. Huruf, memiliki ciri-ciri seperti tinggi, tebal, titik sudut, dan lengkungan garis.
2. Suara, memiliki ciri-ciri seperti amplitudo, frekuensi, nada, dan intonasi.
3. Tanda tangan, memiliki ciri-ciri seperti panjang, kerumitan, dan tekanan.
4. Sidik jari, memiliki ciri-ciri seperti lengkungan, dan jumlah garis.

Ciri pada suatu pola diperoleh dari hasil pengukuran terhadap objek uji. Khusus pada pola yang terdapat di dalam citra, ciri-ciri yang dapat diperoleh berasal dari informasi[6].

1. Spasial: intensitas piksel dan histogram.
2. Tepi: arah dan kekuatan.
3. Kontur: garis, elips dan lingkaran.
4. Wilayah/bentuk: keliling, luas dan pusat massa.

Hasil transformasi Fourier: frekuensi.

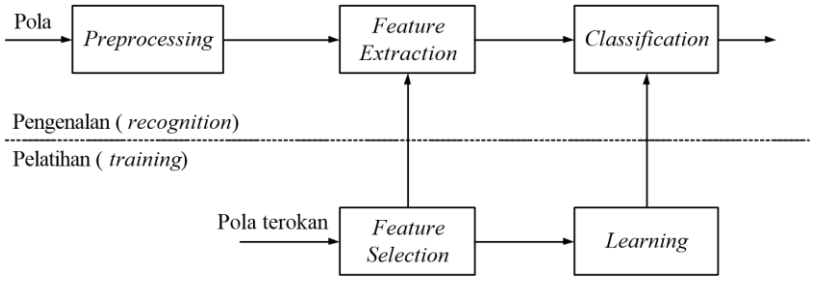
2.2.1 Pengenalan Pola

Pengenalan Pola adalah mengelompokkan data numerik dan simbolik (termasuk citra) secara otomatis oleh mesin (dalam hal ini komputer). Pengenalan pola bertujuan menentukan kelompok atau kategori pola berdasarkan ciri-ciri yang dimiliki oleh pola tersebut. Dengan kata lain, pengenalan pola membedakan suatu objek dengan objek lain. Terdapat dua pendekatan yang dilakukan dalam pengenalan pola: pendekatan secara statistik dan pendekatan secara sintaktik atau structural[6].

a. Pengenalan Pola secara Statistik

Pendekatan ini menggunakan teori-teori ilmu peluang dan statistik. Ciri-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 ditunjukkkan oleh diagram pada gambar 2.1.

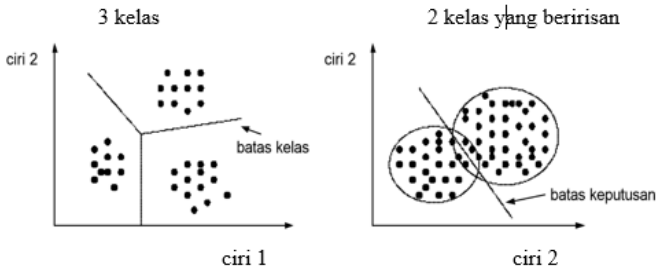


Gambar 2.1 Sistem Pengenalan Pola

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[6].

Pada penelitian ini termasuk sistem pengenalan pola dengan pendekatan statistik, karena ciri-ciri yang dimiliki oleh citra sidik jari 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.

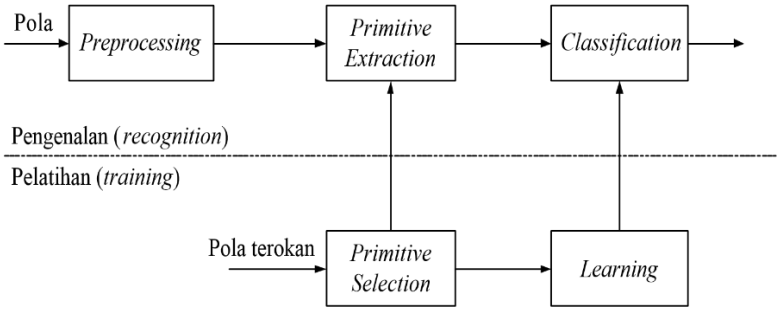
Kumpulan ciri dari suatu pola dinyatakan sebagai vektor ciri dalam ruang bahumatra (multi dimensi). Jadi, setiap pola dinyatakan sebagai sebuah titik dalam ruang bahumatra. Ruang bahumatra dibagi menjadi sejumlah uparuang (sub-ruang). Tiap uparuang dibentuk berdasarkan pola-pola yang sudah dikenali kategori dan ciri-cirinya (melalui fase pelatihan) [6]. Lihat Gambar 2.2.



Gambar 2.2 Contoh Pembagian Kelas Pola

b. Pengenalan Pola Secara Sintaktik

Pendekatan ini menggunakan teori bahasa formal. Ciri-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. Gambar 2.3 memperlihatkan sistem pengenalan pola dengan pendekatan sintaktik. Pengenalan pola secara sintaktik lebih dekat ke strategi pengenalan pola yang dilakukan manusia, namun secara praktek penerapannya relatif sulit dibandingkan pengenalan pola secara statistik[6]. Sistem pengenalan pola dengan pendekatan sintaktik ditunjukkkan oleh diagram pada gambar 2.3.



Gambar 2.3 Sistem pengenalan pola

Pendekatan yang digunakan dalam membentuk tata bahasa untuk mengenali pola adalah mengikuti kontur (tepi batas) objek dengan sejumlah segmen garis terhubung satu sama lain, lalu mengkodekan setiap garis tersebut (misalnya dengan kode rantai). Setiap segmen garis merepresentasikan primitif pembentuk objek[6].

2.3 Pengolahan Citra

Pengolahan citra atau *Image Processing* adalah bidang tersendiri yang sudah cukup berkembang sejak orang mengerti bahwa komputer tidak hanya dapat menangani data teks, tetapi juga citra. Teknik pengolahan citra biasanya digunakan untuk melakukan transformasi dari suatu citra kepada citra lain, sementara tugas perbaikan citra untuk mendapatkan suatu informasi dari citra tersebut merupakan tugas manusia melalui penyusunan algoritmanya[7].

2.3.1 Teknik – Teknik Pengolahan Citra

Secara umum teknik pengolahan citra dibagi menjadi 3 tingkat pengolahan, yaitu sebagai berikut :

1. *Low-Level Processing*, pengolahan ini merupakan operasi dasar dalam pengolahan citra, seperti pengurangan *noise*, perbaikan citra dan restorasi citra.
2. *Mid-Level Processing*, meliputi segmentasi pada citra, deskripsi objek, dan klasifikasi objek secara terpisah.
3. *High-Level Processing*, meliputi analisis citra.

Dari ketiga tingkat pengolahan citra diatas tersebut, dapat dinyatakan suatu gambaran mengenai teknik – teknik pengolahan citra digital dan macam – macamnya antara lain sebagai berikut :

1. *Image Enhancement*, berupa proses perbaiakan citra dengan meningkatkan kualitas itra baik kontras maupun kecerahan.

2. *Image Restoration*, proses memperbaiki model citra, biasanya berhubungan dengan bentuk citra yang sesuai.

3. *Color Image Processing*, suatu proses yang melibatkan citra berwarna, baik berupa image enhancement, image restoration atau yang lainnya.

4. *Wavelet dan Multiresolution Processing*, merupakam suatu proses yang menyatakan citra dalam beberapa resolusi.

5. *Image Compression*, merupakan proses yang digunakan untuk mengubah ukuran data pada citra.

6. *Morpholohical Processing*, proses untuk mendapatkan informasi yang menyatakan deskripsi dari suatu bentuk pada citra.

7. *Segmentataion*, proses untuk membedakan atau memisahkan objek– objek yang ada alam suatu citra, seperti memisahkan objek dengan latar belakangnya.

8. *Object Recognition*, suatu proses yang dialakukan untuk mengenali objek – objek apa saja yang ada dalam suatu citra.

2.3.2 Jenis – Jenis Citra

2.3.1.1 Citra Biner

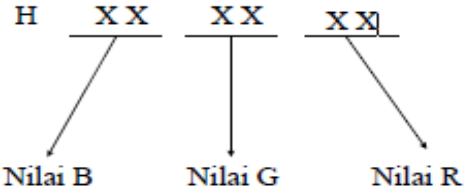
Citra *biner* adalah citra digital yang hanya memiliki dua kemungkinan nilai piksel yaitu hitam dan putih. Citra *biner* disebut juga citra B & W (*Black and White*) atau citra monokrom. Hanya dibutuhkan 1 bit untuk mewakili nilai setiap piksel dari citra *biner*. [8]



Gambar 2.4 Contoh Citra Biner

2.3.1.2 Citra Warna RGB (Red, Green, Blue)

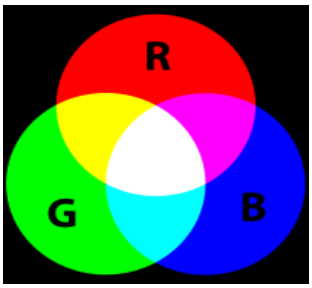
Dalam pengolahan citra, warna dipresentasikan dalam *range* nilai H000000 sampai 0x00ffffff. Warna hitam adalah 0x00000000 dan warna putih adalah Hffffff. Definisi nilai warna di atas adalah sebagai berikut. variabel H menyatakan angka dibelakangnya adalah *hexadecimal*[9].



Gambar 2.5 definisi nilai warna pada citra

Dengan demikian *range* warna yang digunakan adalah (2 8) (2 8) (2 8) = 2 24 (atau yang dikenal dengan istilah *True Colour* pada *Windows*).

Nilai warna yang digunakan di atas merupakan gabungan warna cahaya merah, hijau dan biru seperti yang terlihat pada Gambar 2-6.

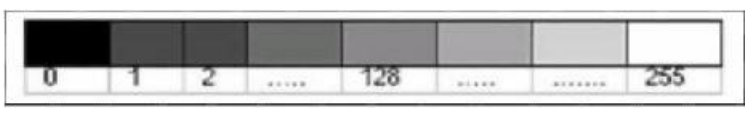


Gambar 2.6 komposisi warna RGB

Dengan demikian, untuk menentukan suatu warna yang bukan warna dasar digunakan gabungan skala kecerahan dari setiap warnanya

2.3.1.3 Citra *Grayscale*

Grayscale adalah warna-warna piksel yang berada dalam rentang gradasi warna hitam dan putih. Intensitas *grayscale* biasanya disimpan sebagai data citra 8 bit atau 256 intensitas warna *gray* dari nilai 0 (hitam) dan 255 (putih). Citra *grayscale* merupakan citra digital yang hanya memiliki satu nilai kanal pada setiap pikselnya, dengan kata lain nilai bagian *RED* = *GREEN* = *BLUE*. Nilai tersebut digunakan untuk menunjukan tingkat intensitas. Warna yang dimiliki adalah warna dari hitam, keabuan, dan putih. Tingkatan keabuan disini merupakan warna abu dengan berbagai tingkatan dari hitam hingga mendekati putih [10]. Model warna ini menyimpan informasi warna sebesar 8 bit. Jadi warna yang bisa ditampilkan pada mode warna *greyscale* berjumlah sampai 256. Skala perhitungan numeriknya bisa dengan pendekatan nilai *brightness* yang berskala 0 sampai 255 seperti pada Gambar 2-7.



Gambar 2.7 Komposisi Warna Grayscale

Untuk mengkonversikan citra yang memiliki warna RGB ke derajat keabuan bisa menggunakan rumus 2.1 :

$X = 0.21 * R + 0.72 * G + 0.07 * B$ [2.1]

Dimana :

X = Citra Greyscale

R = Red

G = Green

B = Blue

R = nilai red * 15 / 255

G = nilai green * 15 / 255

B = nilai blue * 15 / 255

Contoh dari Hasil dari proses *grayscaling* ditunjukkan pada Gambar 2.8 dibawah ini :

(a) (b)



Gambar 2.8 Citra RGB (a) dan citra *Grayscale (b)*

2.3.3 Teknik Pengambilan Citra Digital

Ada beberapa teknik pengambilan citra digital yang bisa dilakukan, antara lain dengan menggunakan kamera digital atau menggunakan *scanner*, atau *barcode scanner*. Hasil dari citra dengan menggunakan teknik kamera atau *scanner* berupa citra *raster* atau citra dengan model matriks. Teknik pengambilan citra selain membuthkan peralatan input, membutuhkan juga suatu *card* yang disebut *frame grabber* yang berupa rangkaian untuk mengolah citra secara *hardware*.

Selain itu pengambilan citra juga bisa dilakukan dengan menggunakan teknik grafika komputer, yaitu dengan membentuk citra komputer sesuai dengan teknik pengambilan citra. Hasil dari citra dengan menggunakan teknik grafika komputer ini bisa berupa citra *raster* atau bisa berupa vektor[11].

2.3.4 Preprocessing

Preprocessing adalah proses pengolahan data asli sebelum data tersebut diolah dengan jaringan syaraf tiruan. Tahap ini merupakan tahap awal penerjemahan. Dalam *preprocessing*, dilakukan penghilangan *noise* agar pola-pola yang tidak dibutuhkan dihilangkan.

Tujuan *preprocessing*, diantaranya*:*

1. Menghilangkan *noise*
2. Memperjelas *features* (fitur) data
3. Memperkecil / memperbesar ukuran data
4. Mengkonversi data asli agar diperoleh data yg sesuai kebutuhan

2.3.5 Thresholding

Thresholding adalah suatu proses dalam pengolahan citra yang berfungsi untuk memisahkan *foreground* dan *background* dari suatu *image*. Proses *tresholding* dilakukan dengan cara melihat perbedaan intensitas warna dari suatu *image*. *Input* dari proses *tresholding* adalah *grayscale* image atau *color image* sedangkan *output* dari proses 26 *tresholding* adalah *binary image* dimana piksel hitam untuk mewakili background dan piksel putih mewakili *foreground*, atau sebaliknya.

Tresholding digunakan untuk mengatur jumlah derajat keabuan yang ada pada suatu citra. Dengan menggunakan *tresholding* maka derajat keabuan dapat diubah seseuai keinginan. *Tresholding* dapat juga digunakan untuk mengubah intensitas piksel menjadi salah satu dari dua nilai, a1 atau a2. Hasil dari proses *thresholding* ditentukan oleh suatu parameter yang disebut nilai *threshold*[12].

Proses *thresholding* menggunakan fungsi sebagai berikut :

$$(x, y, f(x, y)) \rightarrow (x, y, g(x, y))$$

$$g(x, y) = \begin{cases} a_1 & , \quad f(x, y) < T \\ a_2 & , \quad f(x, y) \geq T \end{cases} \qquad [2.2]$$

f(x,y) = Nilai intensitas yang lama

g(x,y) = Nilai intensitas yang baru

T = Nilai tengah dari range 0 – 255 = 128

a1 , a2 = Nilai tetangga

Jika a1=0 dan a2=1, maka proses thresholding mengubah citra *greyscale* menjadi citra b*iner*.

2.4 Tekstur

Tekstur secara umum mengacu pada repetisi elemen-elemen tekstur dasar yang sering disebut primitif atau *texel* (*texture element*). 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.

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[11].

2.4.1 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

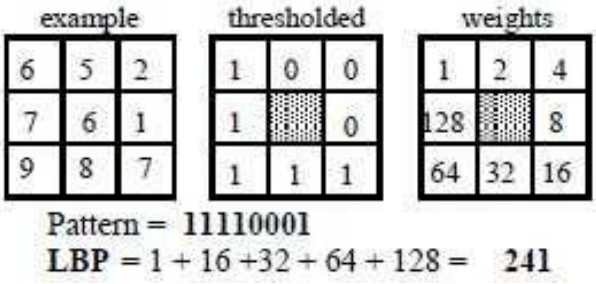
bahwa internsitas dibangkitkan oleh medan acak dua dimensi, metode ini berdasar pada frekuensi-frekuensi ruang. Contoh metode statistis adalah fungsi autokorelasi, run-length, matriks kookurensi, tranformasi fourier, frekuensi tepi. Teknik struktural berkaitan dengan penyusupan bagian-bagian terkecil 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 [6].

2.4.2 Analisis Tekstur Dengan Pendekatan Statistik

Analisis tekstur Metode statistik menggunakan perhitungan statistik distribusi derajat keabuan (histogram) dengan mengukur tingkat kekontrasan, granularitas, dan kekasaran suatu daerah dari hubungan ketetanggaan antar piksel di dalam citra. Metode Statistik ini penggunaannya tidak terbatas untuk teksturtekstur alami yang tidak terstruktur dari sub pola dan himpunan aturan (mikrostruktur). Berdasarkan orde statistiknya, analisis tekstur dapat dikategorikan menjadi 3, yaitu analisis tekstur orde satu, orde dua, dan orde tiga.

2.5 Local Binary Pattern (LBP)

LBP adalah salah satu deskriptor tekstur terbaik dan telah banyak digunakan dalam berbagai aplikasi. LBP telah terbukti sangat diskriminatif dan 15 keuntungan utamanya, yaitu variasi untuk perubahan tingkat abu-abu monoton dan efisiensi komputasi, membuatnya cocok untuk tugas gambar menuntut analisis. *Local Binary Pattern* (LBP) didefinisikan sebagai ukuran tekstur *gray-scale* invarian, berasal dari definisi umum tekstur di daerah sekitar. Operator LBP dapat dilihat sebagai pendekatan kesatuan dengan model statistik dan struktur tradisional berbeda dari analisis tekstur. Secara sederhana, LBP adalah sebuah kode biner yang menggambarkan pola tekstur lokal. Hal ini dibangun dengan lingkungan batas dengan nilai abu-abu dari pusatnya (Ahonen, Hadid, & Pietikainen, 2004, pp. 2-3). Contoh komputasi LBP pada 3x3 piksel pada gambar berikut



Gambar 2.9 Proses perhitungan piksel LBP

Setiap piksel memiliki nilai hasil *grayscale*, kemudian dilakukan *threshold* berpusat pada titik tengah. Piksel yang memiliki nilai sama atau lebih dibandingkan dengan titik tengah diberi nilai 1 selain itu diberi nilai 0. Kemudian nilai LBP didapat dari penjumlahan dua pangkat nilai angka yang bernilai satu. Berikut rumusnya :

$$LBP_{P,R} = \sum_{p=0}^{P-1} s(I_{p,R} - I_c)2^{P-1-p} \qquad [2.7]$$

P = jumlah banyaknya tetangga

R = radius antara titik pusat dan titik tetangga

$LBP_{P,R}$ = nilai desimal hasil konversi nilai biner

I_C = nilai intensitas piksel pusat

$I_{p,R}$ = nilai intensitas piksel tetangga ke-p($p = 0,1, \dots, P-1$) dengan radius R.

Dimana

$$s(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases} \qquad [2.8]$$

s(x) = fungsi *thresholding*.

Kemudian dibuat suatu cara agar pola LBP tidak sensitif terhadap perubahan rotasi *(rotation invariant*).

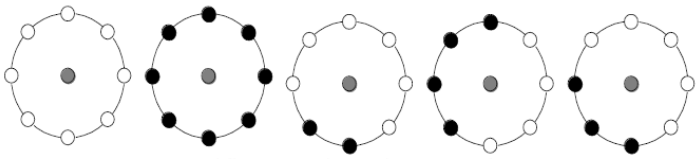
$$LBP^{ri}_{P,R} = \min\{\mathrm{ROR}(LBP_{P,R}, i) \mid i=0,1,\dots,P-1\} \qquad [2.9]$$

ROR(x,i) di atas menerangkan rotasi sirkular kanan dari suatu kelompok bit x sebanyak satu (i) tahap. Pola-pola LBP tertentu memiliki karakteristik utama

dari suatu tekstur. Pola-pola yang memiliki informasi penting ini dinamakan "*uniform patterns*".

Dalam aplikasi analisis tekstur banyak diinginkan untuk memiliki fitur yang invarian atau kuat untuk rotasi gambar input. Sebagai LBP, pola P,R diperoleh dengan sirkuler sampel sekitar piksel pusat, rotasi gambar input memiliki dua efek: setiap lingkungan lokal diputar ke lokasi piksel lainnya, dan dalam masing-masing lingkungan, titik sampling pada lingkaran yang mengelilingi titik pusat diputar ke orientasi yang berbeda. Perpanjangan pada operator aslinya menggunakan *Uniform Pattern*.

LBP dikatakan uniform jika struktur melingkar pola-pola binernya paling banyak terdiri atas dua transisi bit dari 0 ke 1 atau sebaliknya. *Uniform patterns* berfungsi untuk mengidentifikasi noda (spot), flat area atau dark spot, sudut, dan tepi. Hampir 90 persen dari tekstur merupakan *uniform patterns* [12].

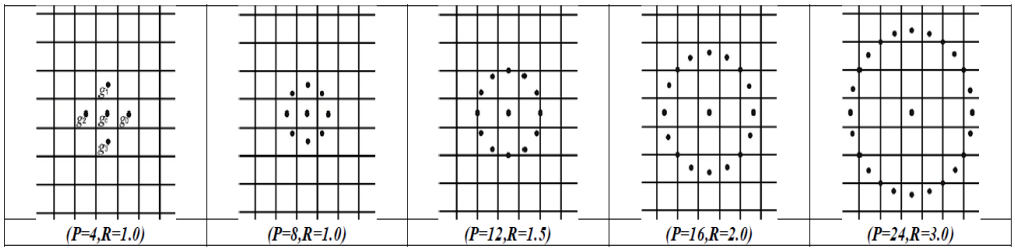


Gambar 2.10 Contoh tekstur *uniform patterns*

Di dalam pemetaan *uniform* LBP ini, terdapat pelabelan yang terpisah untuk setiap pola-pola yang *uniform* dan semua pola-pola yang *non uniform* dimasukkan ke dalam satu label. Sebuah rumusan sederhana untuk menghitung jumlah label pola biner *uniform* dapat terlihat dari P(P-1)+3. P di sini berarti jumlah titik biner yang digunakan. Misalnya untuk pola biner 8 bit, berarti terdapat 59 label pola *uniform* yang berbeda dan 243 label untuk 16 bit.

Penggabungan antara *uniform patterns* dengan *rotation invariant* dilambangkan $LBP_{P,R}^{riu2}$ .Notasi ri menunjukkan *rotation* $LBP_{P,R}^{riu2}$ *invariant* dan u2 untuk *uniform patterns* pada sampling points P dan radius R. merupakan ukuran ketidaksensitifan (invariant) terhadap perubahan *grayscale* dan merupakan ukuran yang digunakan untuk pola spasial [13].

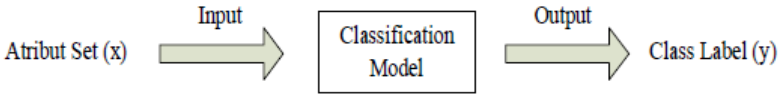
Pola – pola pada LBP memiliki label yang ditandai dengan P dan R. P mewakili jumlah piksel tetangga yang digunakan dalam komputasi sementara R adalah radius antara piksel titik pusat dan piksel tetangga.



Gambar 2.11 varian LBP

2.6 Metode Klasifikasi

Klasifikasi adalah proses untuk menemukan model atau fungsi yang menjelasakan atau membedakan konsep atau kelas data, dengan tujuan untuk dapat memperkirakan kelas dari suatu objek yang labelnya tidak diketahui. Dalam mencapai tujuan tersebut, proses klasifikasi membentuk satu model yang mampu membedakan data kedalam kelas – kelas yang berbeda berdasarkan aturan atau fungsi tertentu. Model itu tersendiri bias berupa aturan "jika-maka", berupa pohon keputusan, atau formula matematis.



Gambar 2.12 Blok Diagram Model Klasifikasi

2.6.1 Klasifikasi Naïve Bayes

Naïve bayes adalah teknik prediksi berbasis probabilistik sederhana yang berdasar pada penerapan teorema Bayes [16].

1. Asumsi independensi (ketidaktergantungan) yang kuat (naif).
2. Model yang digunakan adalah "model fitur independen".

Independensi yang kuat pada fitur adalah bahwa sebuah fitur pada sebuah data tidak ada kaitannya dengan adanya atau tidak adanya fitur yang lain dalam data yang sama. Contoh: kasus klasifikasi hewan dengan fitur: penutup kulit, melahirkan, berat, dan menyusui

1. Dalam dunia nyata, hewan yang berkembang biak dengan cara melahirkan dipastikan hewan tersebut menyusui juga, disini ada ketergantungan pada fitur menyusui karena hewan yang menyusui biasanya melahirkan, atau hewan yang bertelur biasanya tidak menyusui.
2. Dalam Bayes, hal tersebut tidak dipandang, sehingga masing-masing fitur seolah tidak ada hubungan apa-apa.

Kasus lain: prediksi hujan

1. Hujan tergantung angin, cuaca kemarin, kelembaban udara (tidak ada kaitan satu sama lain).
2. Tapi juga tidak boleh memasukkan fitur lain yang tidak ada hubungannya dengan hujan, seperti: gempa bumi, kebakaran, dsb.

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

$$P(X \mid Y = y) = \prod_{i=1}^{q} P(X_i \mid Y = y) \qquad [2.10]$$

Data yang digunakan dapat bersifat kategorial (diskrit) maupun *continue*. Namun, pada tugas akhir ini akan digunakan data *continue*, karena hasil ekstraksi ciri citra merupakan data *continue* berupa angka angka hasil pengukuran tingkat adalah Contrast, Uniformity, Entropy, Inverse Difference Moment (IDM) dan Dissimilarity pada ekstraksi ciri. Maka dari itu untuk data *continue* dapat diselesaikan dengan menggunakan langkah-langkah berikut.

Training :

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

   $\mu = \frac{\sum x_i}{n}$ [2.11]

   Dimana:

   μ = mean

   n = banyaknya data

   $\sum x_i$ = jumlah nilai data

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

   $\sigma^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \mu)^2$ [2.12]

   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 :

   $\varphi_{\mu\sigma}(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$ [2.13]

   Dimana :

   x = data masukan

   π = 3,14

σ = standar deviasi

μ = mean

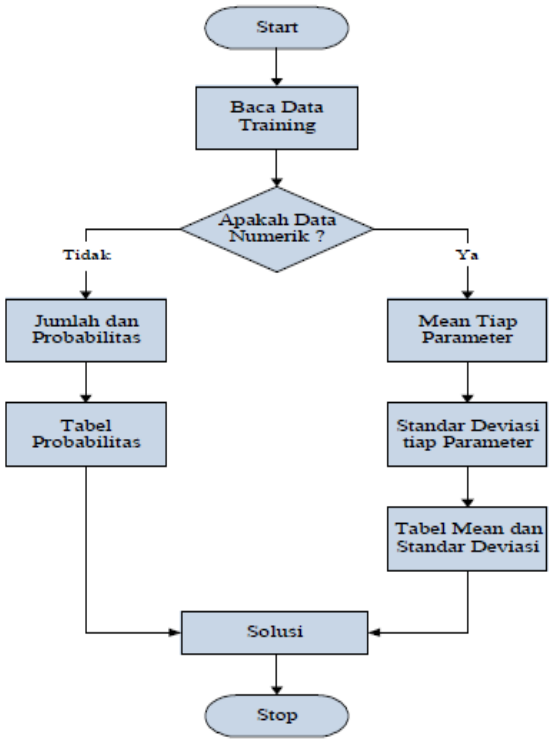
3. Setelah didapatkan nilai densitas probabilitasnya, selanjutnya menghitung posterior masing-masing kelas dengan menggunakan persamaan 2.14.

$$posterior = \frac{prior\ x\ likehood}{evidnce}$$

Atau bisa ditulis

$$Posterior = \frac{P(C_i)x\ P(X1 \dots \dots \dots \dots \dots \dots \dots Xn|C_i)}{P(X1 \dots \dots \dots \dots \dots \dots \dots Xn|C_i)} \quad [2.14]$$

4. Setelah didapat nilai posterior, kelas yang sesuai adalah nilai posterior terbesar.



Gambar 2.13 Skema *Naïve Bayes*

2.7 Anatomi Kuku

Kuku merupakan lapisan keratin yang keras, yang sedikit cembung jika dilihat secara membujur maupun melintang. Kuku menancap pada jaringan lunak yang dipisahkan oleh *periungual grooves* (proksimal, lateral, dan distal). Batang matriks kuku terletak pada bagian proksimal kuku. Lepengan kuku dan matriks sebagian tertutupi oleh lipatan kulit yang dinamakan lipatan kuku proksimal. Pada bagian ini juga terdapat lunula atau half moon, bagian yang terlihat seperti bulan sabit pada kuku. Pada orang dewasa, lunula terlihat pada kuku ibu jari tangan dan kaki.lunula menghubungkan lipatan kuku dengan matriks bagian bawah dimana kuku mulai tumbuh. Kuku tumbuh ke bawah, melewati dasar kuku yang melekat erat dan dipisahkan oleh hyponychium [17].

Bagian kuku lainnya adalah kutikula yang merupakan lapisan tanduk transparan yang melekat pada permukaan kuku dan bertindak sebagai penutup antara lapisan kuku dan lipatan kuku proksimal. Sedangkan pita onychodermal yang berwarna jingga terletak pada bagian bawah kuku. Bagian permukaan atas dari lapisan kuku halus, seadngkan permukaan bawahnya berkerut[17].

Lapisan kuku terbentuk dari lapisan-lapisan keratin, datar, dan sel-sel yang terdiferensiasi sempurna dan tanpa inti. Lapisan kuku bisa dibedakan menjadi tiga zona, yaitu lapisan kuku bagian atas yang membentuk tiga sertiga bagian kuku, lapisan kuku bagian bawah yang membentuk du pertiga bagian kuku, dan keratin subungual. Lapisan keratin subungual ini berhubungan dengan lapisan tanduk yang tebal dan keras dari hyponychium [17].

Dari atas ke bawah, lapisan kuku dibagi menjadi:

1. Lipatan kuku (nail folds)

   Bagian belakangnya merupakan lanjutan epidermis, sedangkan bagian depan merupakan epitel yang datar dan agak tipis dan terkeratinisasi dengan stratum granulosum. Lapisan kutikula menghubungkan bagian tersebut dengan stratum korneum [17].

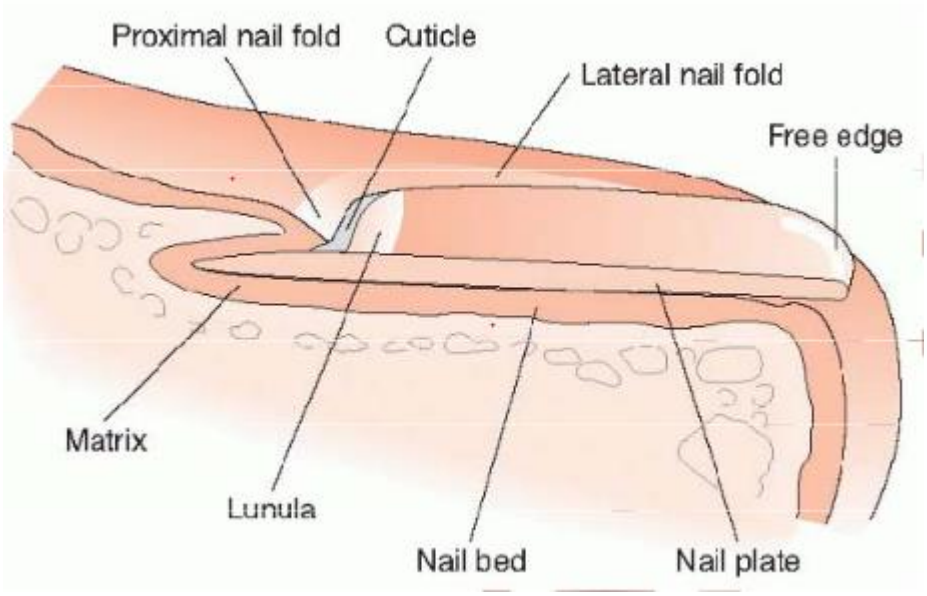
2. Matriks kuku (nail matrix)

Terdiri dari multilayer epithelium yang ditandai dengan keratinisasi mendadak tanpa

Epitel dasar kukuinterposisi granula keratohialin. Bagian matriks kuku merupakan tempat awal pertumbuhan kuku. epithelium matriks mengandung melanosit dan sel-sel Langerhans. Sel-sel melanosit bersifat dorman dan tidak memproduksi pigmen. Tetapi, pada masyarakat kulit hitam, pita-pita berpigmen dapat terlihat pada kuku [17].

3. Dasar kuku (nail bed)

Keratinisasi pada bagian dasar kuku terjadi mendadak. Stratum granulosum hanya muncul kembali pada hyponchium yang merupakan bagian yang mengalami penebalan. Melanosit jarang ditemukan pada bagian dasar kuku. kuku melekat erat pada periosteum tulang jari dengan ikaten kolagen [17].



Gambar 2.14 Anatomi kuku

2.7.1 Kelainan Pada Kuku

Kuku adalah jendela untuk melihat kedalam tubuh seseorang.Sebab melalui kuku kita bisa mendeteksi penyakit apa saja yang sedang berproses dalam tubuh manusia. Suatu penyakit dapat memberikan berbagai bentuk lesi pada kuku yang berbeda, dan sebaliknya satu kelainan kuku dapat merupakan ekspresi dari berbagai macam penyakit dengan etiologi, perjalanan penyakit dan prognosis yang berbeda. Untuk memahami kelainan pada kuku dapat dilihat melalui :

2.7.1.2 Warna Kuku

Studi Dermatologi mengatakan bahwa berbagai masalah kesehatan bisa dianalisis dengan perubahan warna kuku jari. Penyakit yang paling umum dapat didiagnosis dengan warna kuku alah satunya Penyakit Liver atau penyakit pada hati, kemudian kuku dengan warna setengah merah muda dan setengah kuku putih, kuku kuning, kuku putih, kuku merah, kuku biru, kuku hitam, diantaranya kondisi jantung dan diabetes.

2.7.1.3 Tekstur

perubahan tekstur di kuku dapat disebabkan oleh berbagai penyakit. Umumnya kuku yang sehat lembut, terlihat mengkilap dan halus sementara kuku yang berpenyakit atau terdapat kelainan diantaranya rapuh, kasar, diadu, garis vertikal pada permukaan kuku. Selain itu Kelainan pada kuku dapat dibedakan menjadi tiga bagian besar bersadarkan anatomi kuku, yaitu kelainan pada lipatan kuku, kelainan pada dasar kuku, dan kelainan pada lempeng kuku..

2.7.2 Jenis – Jenis Kelainan

2.7.2.1 *Clubbing*/Jari Tabuh

jari tabuh atau *clubbing finger* adalah kelainan bentuk jari dan kuku tangan yang menjadikan jari tangan dan kaki membulat yang berkaitan dengan penyakit jantung dan paru-paru. Penyebab terjadinya *clubbing finger* adalah penambahan jaringan ikat yang terjadi pada bagian jaringan lunak di dasar kuku yang berkaitan dengan kekurangan oksigen kronik/hipoksia kronik.Jari-jari tabuh terjadi karena adanya sianosis jangka

lama. Sianosis menujukkan bahwa kurang kadar oksigen. Hal ini terjadi di jari-jari sebab terdapat pembuluh darah *perifer*. Kurangnya kadar oksigen di *perifer* khususnya di jari-jari merangsang otak untuk mendilatasikan pembuluh darah di jari-jari. Dilatasi pembuluh darah ini bersifat permanen yang mengakibatkan jari-jari tabuh



Gambar 2.15 *Clubbing*/Jari Tabuh

Penyakit yang berkaitan erat dengan *clubbing finger* :

- Paru-paru, seperti kanker paru primer dan *metastatic*, *bronkiektasis*, abses paru, kristik *fribosis* serta *mesotelioma*. Jantung, seperti kelainan jantung sianotik bawaan lahir, atrial *myxoma*, *endokarditis infeksiosa*. *Gastrointestinal*, yang mencakup enteritis regional, *colitis ulseratif kronik* dan *sirosis hepatic*. Lainnya, seperti *hipertiroid*, sakit liver, malabsorbsi.

2.7.2.2 Koilonikia atau disebut juga *spoon nails*

yaitu bentuk normal bagian tengah Lempeng Kuku menjadi datar atau cekung dengan pinggir lateral dan distal menghadap ke atas, sehingga bentuk kuku seperti sendok. Lempeng Kuku dapat menebal atau menipis. Koilonikia dapat merupakan kelainan yang didapat atau merupakan kelainan yang diturunkan. Pada anak usia 1 atau 2 tahun pertama dapat dijumpai bentuk Lempeng Kuku seperti ini, yang pada beberapa kasus menetap sampai dewasa tanpa adanya kelainan familial. Tipisnya kuku dihubungkan dengan status nutrisi yang buruk (anemia defesiensi besi) dan kurangnya asam amino yang mengandung sulfur.



Gambar 2.16 koilonikia/*spoon nails*

Penyakit yang berkaitan erat dengan koilonikia :

- Diabetes, anemi defisiensi Fe, pajanan asam kuat, hipotiroid, nail patella syndrome, Penyakit Raynaud (sebuah kondisi umum yang mempengaruhi pasokan darah ke jari tangan dan kaki), hemochromatosis, SLE (lupus), trauma parah pada matriks kuku, sindrom kuku-patela

2.7.2.3 *Onychomycosis*

Onychomycosis adalah infeksi jamur superfisial yang ditemukan di seluruh dunia. Di negara maju (industri) didapatkan angkat insiden *Onychomycosis* hingga 30% dari seluruh kejadian infeksi jamur superfisial, 40% dari seluruh penyakit kuku dan diperkirakan terjadi 2 hingga 18% pada populasi. *Onychomycosis* lebih sering terjadi pada kuku jari kaki dari pada kuku jari tangan. *Onychomycosis* merupakan *dermatomikosis superfisial* yang sebagian besar penyebabnya adalah golongan *de*rmatofita, berarti jamur yang keratinolitik, dimana di dalam hidupnya membutuhkan keratin. Jamur akan mengambil keratin disekitarnya untuk hidupnya. Kuku tersusun dari keratin. Karena keratin diambil oleh jamur maka lambat laun kuku menjadi rapuh dan akhirnya rusak.



Gambar 2.17 *Onychomycosis*

Penyakit yang berkaitan erat dengan *Onychomycosis* :

distres intrauterin berat, diabetes, trauma matriks, radiodermatitis, inflamasi lipat kuku proksimal, dan juga akibat efek samping obat asam valpoat yang di gunakan sebagai obat epilepsi, *nekrolisis epidermal toksik*, *stevevens jhonson syndrome*, kelainan Periode iskemia (Gagal jantung, iskemia perifer, Himalaya wandeling), Viral infeksi (*sindroma Kawasaki* , campak,penyakit tangan kuku dan mulut)dan Penyakit dengan demam tinggi atau penyakit parah (Malaria, *drug fever*, *sepsis*, demam berdarah)

2.7.2.4 *Terry Nails* (Kuku Terry)

kondisi kuku dimana sebagian besar kuku tampak putih kecuali di bagian ujung kuku yang masih tampak merah muda. Kondisi kuku terry kadang merupakan efek penuaan. Tetapi pada kasus lain, kuku Terry bisa jadi pertanda penyakit.

Penyakit yang berkaitan erat dengan Terry Nails berbahaya,seperti penyakit liver, gagal jantung kongestif, gagal ginjal, atau diabetes.

2.8 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 denga 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 [18].

2.9 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 [18].

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.

2.10 OOP(Object Oriented Programing)

Pemrograman 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 [19].

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. Kecepata 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 salah satu contoh bahasa pemograman yang mendukung pemrograman berorinentasi objek adalah C#

C# adalah bahasa pemrograman yang diciptakan oleh Microsoft (dikembangkan dibawah kepemimpinan Anders Hejlsberg yang notabene juga telah menciptakan berbagai macam bahasa pemrograman termasuk Borland Turbo C++ dan Borland Delphi). Bahasa C# juga telah di standarisasi secara internasional oleh ECMA.

Seperti halnya bahasa pemrograman yang lain, C# bisa digunakan untuk membangun berbagai macam jenis aplikasi, seperti aplikasi berbasis windows (desktop) dan aplikasi berbasis web serta aplikasi berbasis web services.

Kelebihan menggunakan c# diantaranya :

1. Sederhana (simple)

   C# menghilangkan beberapa hal yang bersifat kompleks yang terdapat dalam beberapa macam bahasa pemrograman seperti Java dan C++, termasuk

diantaranya mengilangkan macro, templates, multiple inheritance dan virtual base classes. Hal hal tersebut yang dapat menyebabkan kebingungan pada saat menggunakannya, dan juga berpotensial dapat menjadi masalah bagi para programmer C++. Jika anda pertama kali belajar bahasa C# sebagai bahasa pemrograman, maka hal-hal tersebut diatas tidak akan membuat waktu anda terbuang terlalu banyak untuk mempelajarinya. C# bersifat sederhana, karena bahasa ini didasarkan kepada bahasa C dan C++. Jika anda familiar dengan C dan C++ atau bahkan Java, anda akan menemukan aspekaspek yang begitu familiar, seperti statements, expression, operators, dan beberapa fungsi yang diadopsi langsung dari Java, C dan C++, tetapi dengan berbagai perbaikan yang membuat bahasanya menjadi lebih sederhana.

2. Modern

beberapa fitur seperti exception handling, garbage collection, extensible data types, dan code security (keamanan kode/bahasa pemrograman). Dengan adanya fitur-fitur tersebut, menjadikan bahasa C# sebagai bahasa pemrograman yang modern.

3. Object-Oriented Language

Kunci dari bahasa pemrograman yang bersifat Object Oriented adalah encapsulation, inheritance, dan polymorphism. Secara sederhana, istilah-istilah tersebut bisa didefinisikan sebagai berikut (definisi dan penjelasan lebih lanjut akan di uraikan pada bab-bab selanjutnya). encapsulation, dimana semua fungsi ditempatkan dalam satu paket (single package). inheritance, adalah suatu cara yang terstruktur dari suatu kode-kode pemrograman dan fungsi untuk menjadi sebuat program baru dan berbentuk suatu paket. Polymorphism, adalah kemampuan untuk mengadaptasi apa yang diperlukan untuk dikerjakan. Sifat-sifat tersebut di atas, telah di miliki oleh C# sehingga bahasa C# merupakan bahasa yang bersifat Object Oriented.

4. Powerfull dan fleksibel

C# bisa digunakan untuk membuat berbagai macam aplikasi, seperti aplikasi pengolah kata, grafik, spreadsheets, atau bahkan membuat kompiler untuk sebuah bahasa permrograman.

5. Efisien

C# adalah bahasa pemrograman yang menggunakan jumlah kata-kata yang tidak terlalu banyak. C# hanya berisi kata-kata yang biasa disebut dengan keywords. Keywords ini digunakan untuk menjelaskan berbagai macam informasi. Jika anda berpikiran bahwa bahasa pemrograman yang menggunakan sangat banyak kata-kata (keywords) akan lebih powerfull, maka jawabannya adalah "pemikiran itu tidak selalu benar", karena hal itu justru bisa menambah kerumitan para developer pada saat membuat suatu aplikasi.

6. Modular

Kode C# ditulis dengan pembagian masing Class-Class (classes) yang terdiri dari beberapa routines yang disebut sebagai member methods. Class-Class dan metodemetode ini dapat digunakan kembali oleh program atau aplikasi lain. Hanya dengan memberikan informasi yang dibutuhkan oleh Class dan metode yang dimaksud, maka kita akan dapat membuat suata kode yang dapat digunakan oleh satu atau beberapa aplikasi dan program (reusable code).

2.10.1 Konsep Dasar 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.

2.10.2 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 [20].

A. Struktur Diagram, yaitu kumpulan diagram yang digunakan untuk menggambarkan suatu struktur statis dari sistem yang dimodelkan. Pada Struktur Diagram dibagi menjadi 6 bagian[20].

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

   Digram 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 bru muali 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.