BAB 2
LANDASAN TEORI

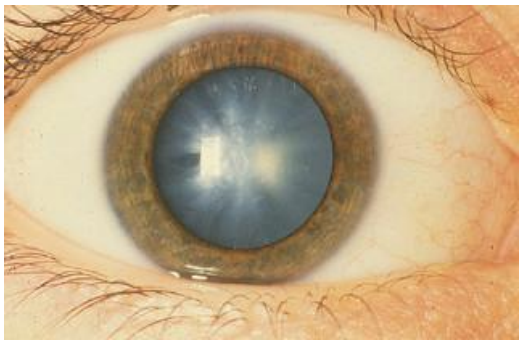
2.1 Mata

Mata merupakan alat indra yang terdapat pada manusia. Secara konstan mata menyesuaikan jumlah cahaya yang masuk, memusatkan perhatian pada objek yang dekat dan jauh serta menghasilkan gambaran yang kontinyu yang dengan segera dihantarkan ke otak. Ketika cahaya mengenai mata sinyal saraf terbentuk dan dikirimkan ke otak, untuk memberikan pesan tentang keberadaan cahaya dan kekuatan cahaya [15].

Mata manusia terdiri dari banyak bagian, semuanya bekerja sama untuk mengubah sinar cahaya yang masuk melalui pupil mata menjadi informasi yang kemudian akan diterjemahkan oleh otak. Sinar cahaya menyinari benda dan terpental dari benda sebelum mereka sampai ke pupil mata, proses itulah yang menentukan warna dan kecerahan yang otak persepsikan. Berkat proses itu kita melihat tanda tipis dari spektrum elektromagnetik yang kita sebut cahaya yang tampak dan menyadari apa yang ada di sekeliling kita.

Pada mata normal, cahaya yang masuk akan dibiaskan oleh kornea, melalui pupil, dan difokuskan oleh lensa tepat pada retina. Retina terdiri dari *nerve cells* dinamakan *rods and cones*. Sel-sel tersebut merubah cahaya menjadi impuls listrik yang dikirim oleh syaraf optik ke otak kita [15].

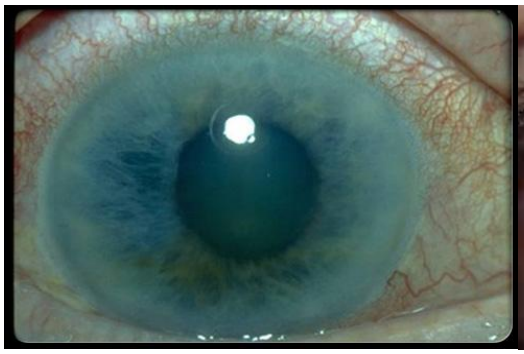
Jika mata tidak berfungsi dengan baik, dengan sendirinya akan berpengaruh terhadap penglihatan kita. Tidak berfungsinya mata sebagaimana mestinya disebabkan adanya kelainan atau penyakit. Penyakit pada mata itu bermacam-macam, seperti *conjunctivitis*, *cataract*, *glaucoma*, dan *pterygium* dan beberapa penyakit lainnya, berikut beberapa contoh kasus penyakit mata.



(a) *Cataract*



(b) *Conjunctivitis*



(c) *Glaucoma*



(d) *Pterygium*

2.1 Gambar beberapa penyakit mata

Cataract adalah pengkabutan dari lensa-lensa dalam mata yang mempengaruhi penglihatan. *Cataract* seringkali terjadi pada orang yang sudah usia lanjut namun tidak menutup kemungkinan terjadi pada anak kecil dan usia remaja. *Cataract* dapat terjadi pada satu atau kedua mata, tetapi tidak menular.

Conjunctivitis (mata merah) merupakan peradangan atau infeksi pada membran transparan (*conjunctiva*) yang membatasi kelopak mata dengan bola mata. Peradangan ini menyebabkan pembuluh darah di *conjunctiva* menjadi lebih mencolok, yang menyebabkan perubahan warna pink atau merah pada bagian putih mata. *Conjunctivitis* umumnya disebabkan oleh bakteri atau infeksi virus, reaksi alergi atau pada bayi yang baru lahir pembukaan saluran air mata yang tidak sempurna.

Glaucoma merupakan penyakit mata dimana tekanan cairan dalam bola mata menjadi terlalu tinggi, sehingga merusak serat lembut saraf optik yang membawa sinyal penglihatan dari mata ke otak. Kerusakan ini tidak dapat disembuhkan dan dapat menyebabkan kebutaan pada tahapan yang parah.

Pterygium adalah cedera mata yang biasa terjadi pada orang yang banyak menghabiskan waktu diluar rumah. Karena ini banyak menyerang peselancar (*surfer*), kondisi ini disebut juga dengan istilah mata peselancar (*surfer's eye*). Pterygium bisa menyerang siapa saja, baik anak kecil maupun orang dewasa. Pterygium ditandai dengan munculnya selaput berwarna merah di bagian putih mata, umumnya di ujung dalam mata di dekat hidung. Pada kasus yang parah, *pterygium* bisa tumbuh menutupi pupil hingga mengganggu penglihatan

2.2 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 [9].

2.3 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 pixel dan histogram.
2. Tepi: arah dan kekuatan.
3. Kontur: garis, elips dan lingkaran.
4. Wilayah/bentuk: keliling, luas dan pusat massa.
5. Hasil transformasi Fourier: frekuensi.

2.4 Pengenalan Pola

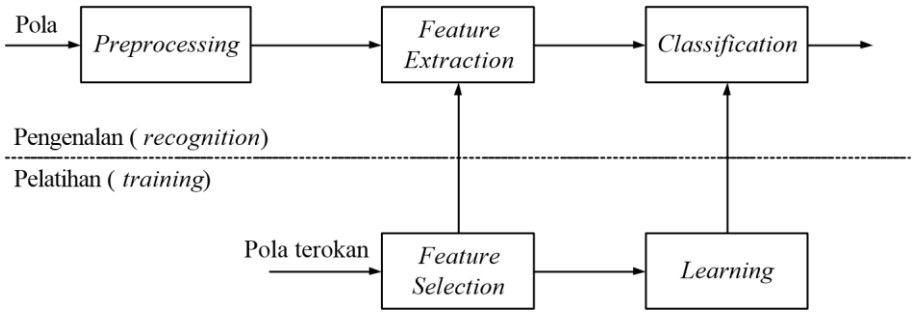
Pengenalan Pola adalah mengelompokkan data numerik dan simbolik (termasuk citra) secara otomatis oleh mesin (dalam hal ini komputer). Pengenalan pola bertujuan menentukan kelompok atau kategori pola berdasarkan ciri-ciri yang

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 struktural [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 ditunjukan oleh diagram pada gambar 2.2.



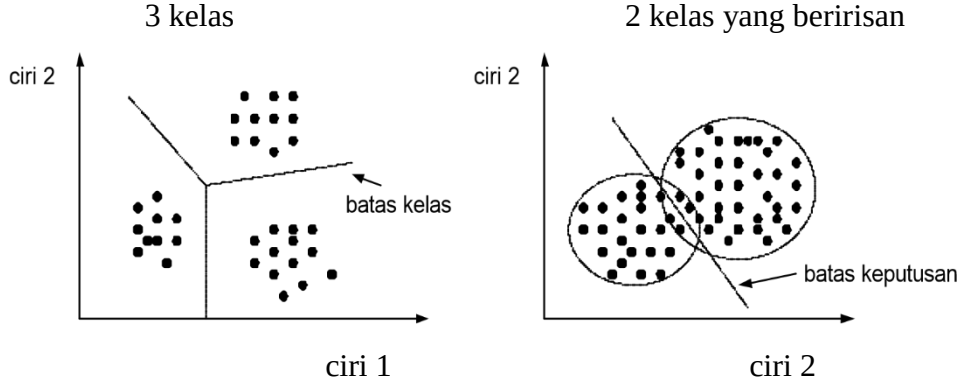
Gambar 2.2 Sistem pengenalan pola dengan pendekatan statistik [6].

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.

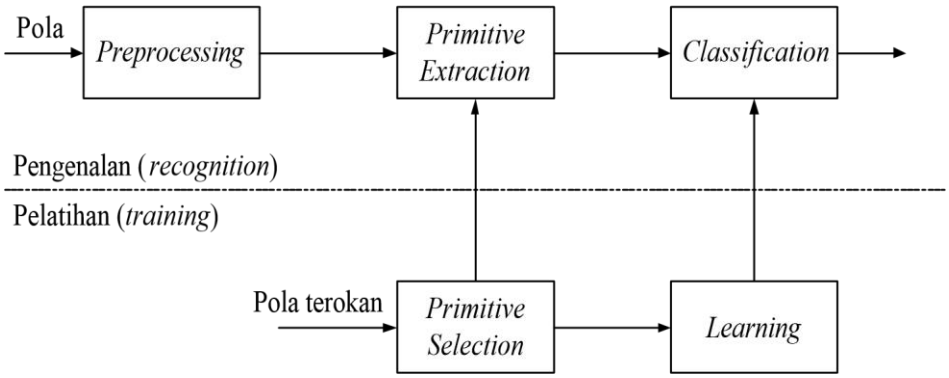
Kumpulan ciri dari suatu pola dinyatakan sebagai vektor ciri dalam ruang bahumatra (multi dimensi). Jadi, setiap pola dinyatakan sebagai sebuah titik dalam ruang bahumatra. Ruang bahumatra dibagi menjadi sejumlah uparuang (sub-ruang). Tiap uparuang dibentuk berdasarkan pola-pola yang sudah dikenali kategori dan ciri-cirinya (melalui fase pelatihan) [6]. Lihat Gambar (2.3).



Gambar 2.3 Contoh pembagian kelas pola [6]

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 2.4 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 ditunjukan oleh diagram pada gambar 2.4.



Gambar 2.4 Sistem pengenalan pola dengan pendekatan sintaktik [6].

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].

2.5 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 dari pada citra masukan [6].

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 [6]:

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-contoh operasi perbaikan citra:

   a. Perbaikan kontras gelap/terang
   b. Perbaikan tepian objek (*edge enhancement*)
   c. Penajaman (*sharpening*)
   d. Pemberian warna semu (*pseudocoloring*)
   e. 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-contoh operasi pemugaran citra:

   a. Penghilangan kesamaran (*deblurring*).
   b. 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-contoh operasi pengorakan citra:

   a. Pendeteksian tepi objek (*edge detection*)
   b. Ekstraksi batas (*boundary*)
   c. 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 [9]. 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 :

$$X = 0.21 \cdot R + 0.72 \cdot G + 0.07 \cdot B \quad (2.1)$$

Dimana : R = nilai red * 15 / 255

G = nilai green * 15 / 255

B = nilai blue * 15 / 255

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 [9].

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 2.5 berikut.



Gambar 2.5 Contoh Citra Tekstur [9]

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 [9].

2.7.1 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 [9].

2.8 Gray Level Aura Matrix

Pada gray level aura matrix atau biasa disingkat dengan GLAM, peristiwa dari menghitung set pixel dengan cara mencari pixel terkait yang

bersebelahan yang didefinisikan dengan struktur elemen tertentu [2]. Maka akan dihasilkan sebuah kemungkinan dari kumpulan variable distribusi.
Jika terdapat derajat keabuan m, ”probability density function” adalah vektor berdimensi m yang mana komponen ke 1 adalah probabilitas bahwa (i, j) akan memiliki nilai i. Data- data yang diambil meliputi semua features yang telah ditentukan. Fitur-fitur dalam menentukan data adalah Contrast, Uniformity, Entropy, Inverse Difference Moment (IDM) dan Dissimilarity.

1. Kontras (*Contrast*)

Menunjukkan ukuran penyebaran (momen inersia) elemen-elemen matriks citra. Jika letaknya jauh dari diagonal utama, nilai kekontrasan besar. Secara visual, nilai kekontrasan adalah ukuran variasi antar derajat keabuan suatu daerah citra. Hasil perhitungan kontras berkaitan dengan jumlah keberagaman intensitas keabuan dalam citra.

Rumus $Kontras = \frac{1}{N^2} \sum_{i,j=0}^{n-1} (Pi, j).(i - j)^2$ (2.2)

Dimana:
i dan j adalah sifat keabuan dari resolusi 2 piksel yang berdekatan
p (i,j) adalah *Probabilitas kolom(i,j)*

2. Uniformity

Uniformity yang menunjukkan kesamaan citra berderajat keabuan sejenis. Citra akan memiliki harga kesamaan yang besar.

Rumus Uniformity = $\sum_{i,j=0}^{n-1} P(i, j) / [1 + (i - j)^2]$ (2.3)

Dimana:
i dan j adalah sifat keabuan dari resolusi 2 piksel yang berdekatan
p (i,j) adalah *Probabilitas kolom(i,j)*

3. Entropi (*Entropy*)

Entropi dapat menunjukkan ketidakteraturan ukuran bentuk, jika nilai Entropinya besar untuk citra dengan transisi derajat keabuan merata dan bernilai kecil jika struktur citra tidak teratur (bervariasi).

Rumus Entropy $= -\sum_{i,j=0}^{n-1} P(i,j) \ \log(P(i,j))$ (2.4)

Dimana:

i dan j adalah sifat keabuan dari resolusi 2 piksel yang berdekatan

p (i,j) adalah *Probabilitas kolom(i,j)*

4. IDM

Idm menyatakan ukuran konsentrasi pasangan dengan intensitas keabuan tertentu pada matriks, dimana (i, j) menyatakan nilai pada baris i dan kolom j pada matriks kookuren

Rumus IDM = $\sum_{i,j=0}^{n-1} P(i,j)^2$ (2.5)

Dimana:

i dan j adalah sifat keabuan dari resolusi 2 piksel yang berdekatan

p (i,j) adalah *Probabilitas kolom(i,j)*

5. Dissimilarity

Dissimilarity menyatakan ukuran ketidaksamaan linear derajat keabuan citra sehingga dapat memberikan petunjuk adanya struktur linear dalam citra.

Rumus Dissimilarity = $\sum_{i,j=0}^{n-1} P(i,j).|i-j|$ (2.6)

Dimana:

i dan j adalah sifat keabuan dari resolusi 2 piksel yang berdekatan

p (i,j) adalah *Probabilitas kolom(i,j)*

2.9 Klasifikasi Naïve Bayes

Naïve bayes adalah teknik prediksi berbasis probabilistik sederhana yang berdasar pada penerapan teorema Bayes [14].

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 :

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

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.

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

   Dimana:

   μ = mean

   n = banyaknya data

   $\sum x_i$ = jumlah nilai data

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

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

   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 μ = mean

   π = 3,14 σ = standar deviasi

3. Setelah didapatkan nilai densitas probabilitasnya, selanjutnya menghitung posterior masing-masing kelas dengan menggunakan persamaan 2.11.

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

Atau bisa ditulis

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

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

2.10 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 [12].

2.10.1 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 [12].

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.

2.11 OOP (Object Oriented Progamming)

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 [10].

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.

2.11.1 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.

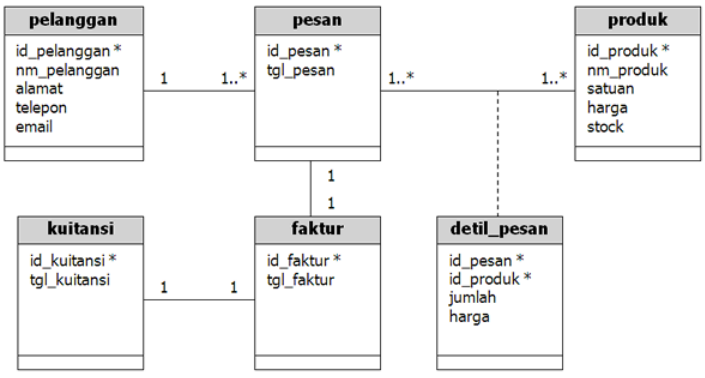
2.12 UML (*Unified Modelling Language*)

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 [12]:

A. Struktur Diagram, yaitu kumpulan diagram yang digunakan untuk menggambarkan suatu struktur statis dari sistem yang dimodelkan. Pada Struktur Diagram dibagi menjadi 6 bagian [11] :

1. Diagram 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.



Gambar 2.6 Contoh Diagram kelas

2. Diagram Objek

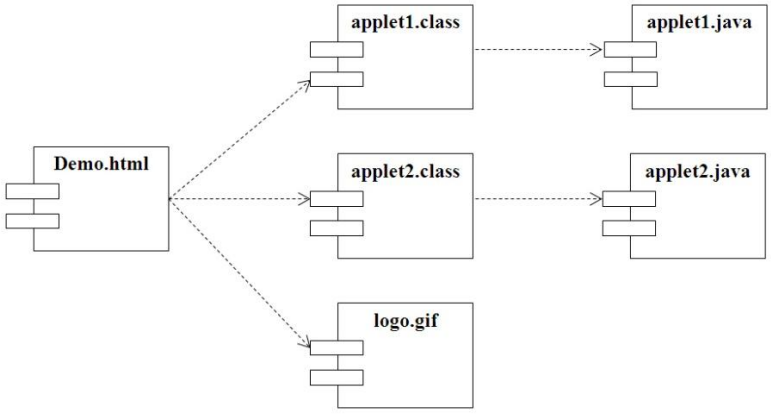
   Digram objek menggambarkan struktur sistem dari segi penamaan objek dan jalannya objek dalam sistem.



Gambar 2.7 Contoh Diagram Objek

3. Diagram Komponen

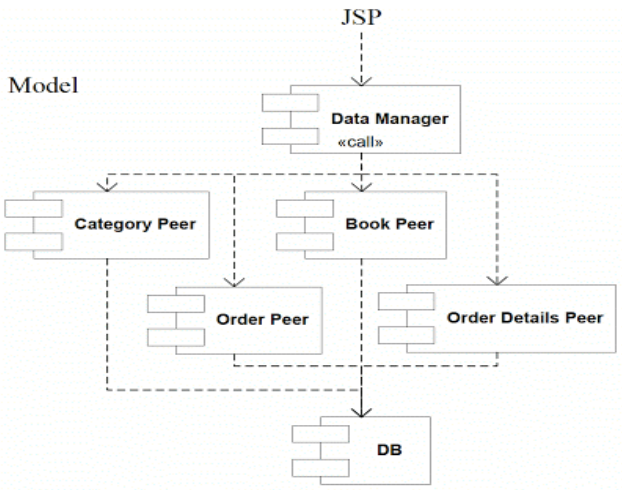
   Diagram komponen dibuat untuk menunjukan organisasi dan ketergantungan diantara kumpulan komponen dalam sebuah sistem.



Gambar 2.8 Contoh Diagram Komponen

4. *Composite Structure Diagram*

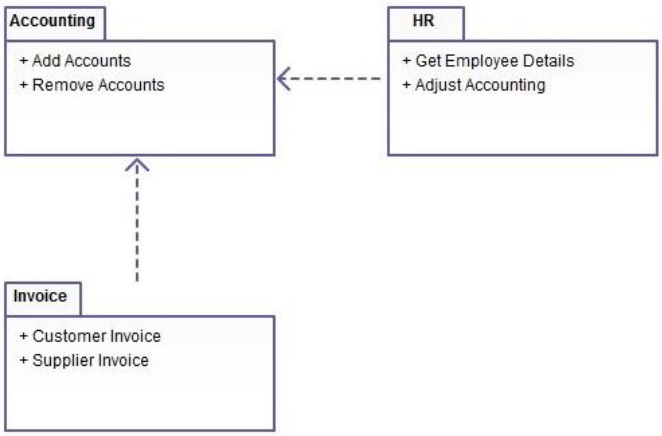
   Composite Structure Diagram baru mulai ada pada UML versi 2.0. diagram ini dapat digunakan untuk menggambarkan struktur dari bagian-bagian yang saling terhubung maupun mendeskripsikan struktur pada saat berjalan (*runtime*) .



Gambar 2.9 Contoh Composite Structure Diagram

5. *Package Diagram*

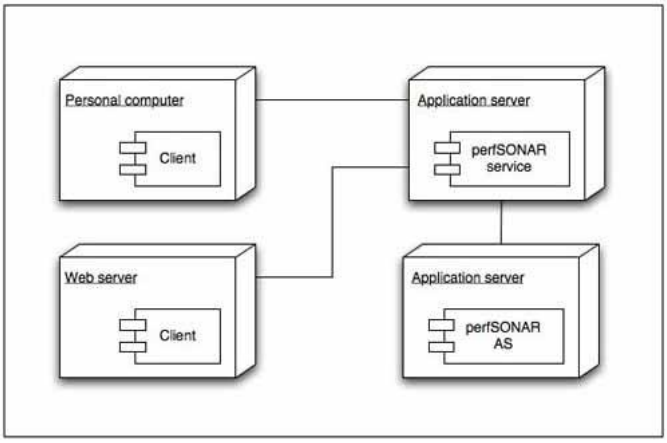
   Package diagram menyediakan cara mengumpulkan elemen-elemen yang saling terkait dalam diagram UML. Hampir semua diagram dalam UML dapat dikelompokan menggunakan *Package Diagram*.



Gambar 2.10 Contoh Package Diagram

6. *Deployment Diagram*

   Deployment diagram menunjukan konfigurasi komponen dalam proses eksekusi aplikasi.

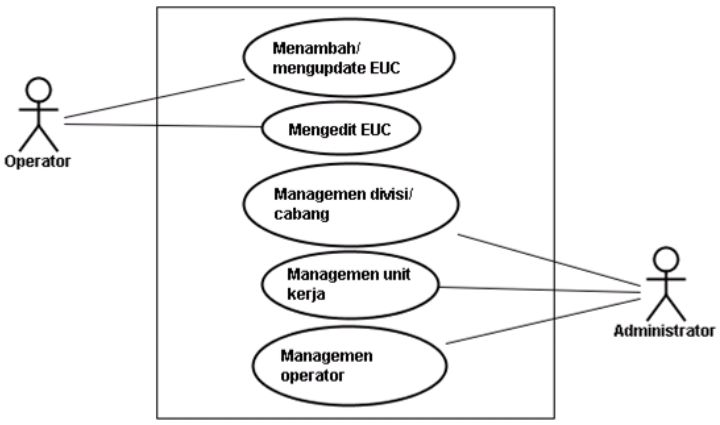


Gambar 2.11 Contoh Deployment Diagram

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*

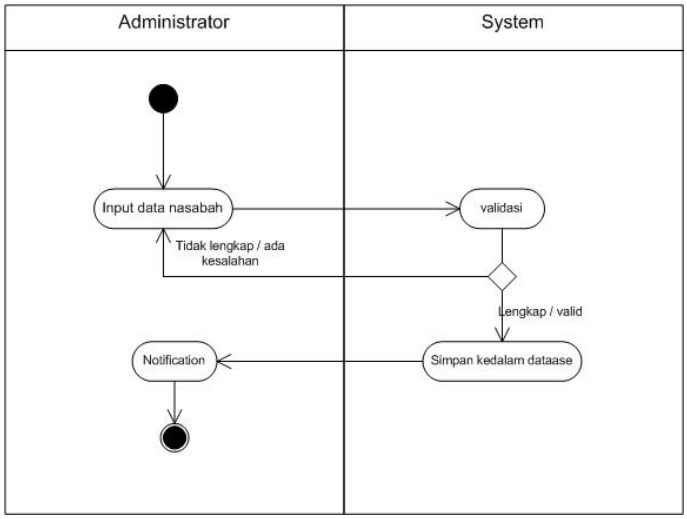
   Use case diagram merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat,. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat.



Gambar 2.12 Contoh Use Case Diagram

2. *Activity Diagram*

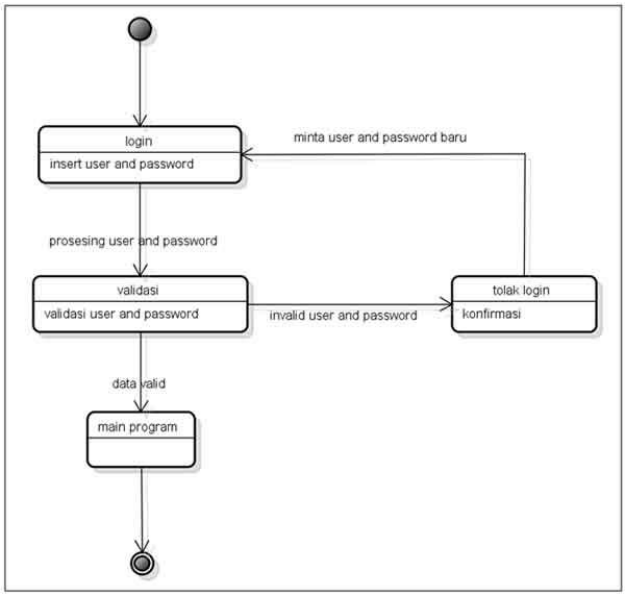
Activity diagram menggambarkan *workflow* atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak.



Gambar 2.13 Contoh Activity Diagram

3. *State Machine Diagram*

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

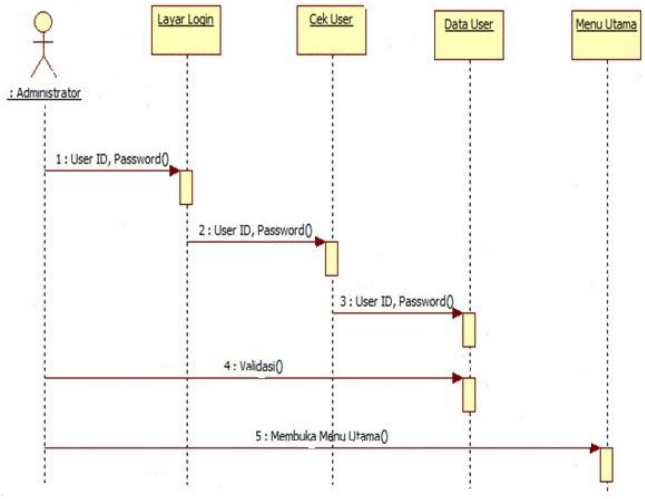


Gambar 2.14 Contoh State Machine Diagram

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*

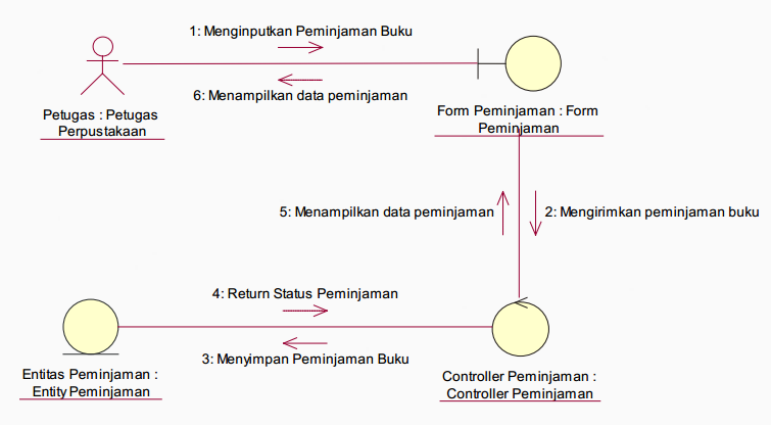
   Sequence diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan message yang dikirimkan dan diterima antar objek.



Gambar 2.15 Contoh Sequence Diagram

2. *Communication Diagram*

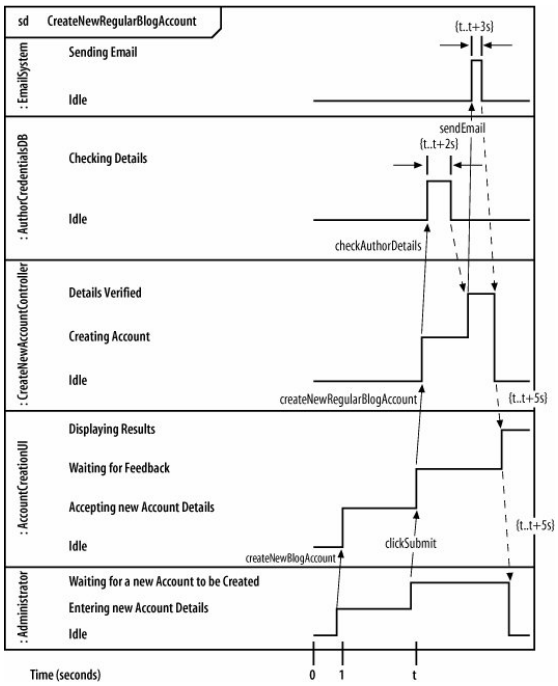
   Communication diagram menggambarkan interaksi antar bojek/bagian dalam bentuk urutan pengiriman pesan. Diagram komunikasi merepresentasikan informasi yang diperoleh dari diagram kelas, diagram *sequence*, dan diagram *use case* untuk mendeskripsikan gabungan antara struktur statis dan tingkah laku dinamis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



Gambar 2.16 Contoh Communication Diagram

3. *Timing Diagram*

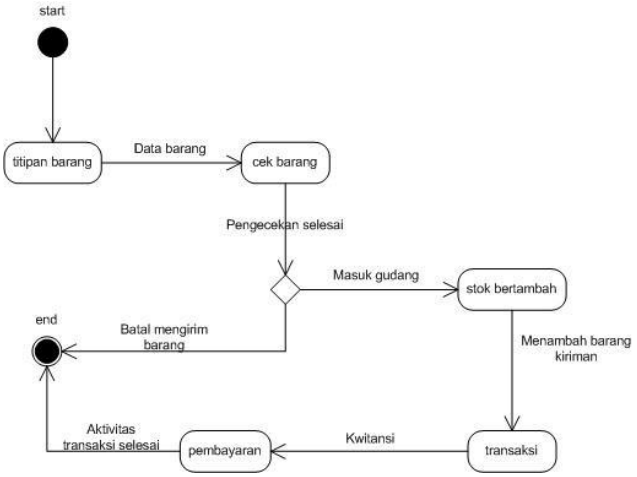
Timing diagram merupakan diagram yang fokus pada penggambaran terkait batas waktu.



Gambar 2.17 Contoh Timing Diagram

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.



Gambar 2.18 Contoh Interaction Overview Diagram

2.13 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 [13].

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 [13].