

## **BAB II**

### **LANDASAN TEORI**

#### **2.1 Definisi Citra**

Indrawati (2013), Citra atau bisa disebut dengan gambar merupakan salah satu komponen dari multimedia yang memegang peranan penting karena mengandung informasi dalam bentuk visual. Citra memiliki lebih banyak informasi yang dapat disampaikan dibandingkan dalam bentuk teks. Sedangkan menurut Fauzi (2007), Citra merupakan kumpulan elemen-elemen gambar (*pixel*) yang secara keseluruhan merekam suatu adegan (*scene*) melalui pengindera visual (kamera). Untuk kebutuhan pengolahan dengan bantuan komputer, citra disajikan dalam bentuk diskrit yang disebut citra digital.

Sutoyo et. al. (2009), menyatakan “Citra adalah suatu representasi (gambaran), kemiripan, atau imitasi dari suatu objek”. Citra sebagai keluaran suatu sistem perekaman data dapat bersifat optic berupa foto, bersifat analog berupa sinyal-sinyal video seperti gambar pada monitor televisi, atau bersifat digital yang dapat langsung disimpan pada suatu media penyimpanan. Perekaman data citra dapat dibagi menjadi dua yaitu:

##### **a. Citra Analog**

Citra analog yaitu terdiri dari sinyal-sinyal elektromagnetik yang tidak dapat dibedakan sehingga pada umumnya tidak dapat ditentukan ukurannya. Citra analog mempunyai fungsi yang kontinu. Hasil perekaman citra analog dapat bersifat optik yakni berupa foto (film foto konvensional) dan bersifat sinyal video seperti gambar pada monitor televisi.

##### **b. Citra Digital**

Citra digital terdiri dari sinyal-sinyal yang dapat dibedakan dan mempunyai fungsi yang tidak kontinu yakni berupa titik-titik warna pembentuk citra. Hasil perekaman citra digital dapat disimpan pada suatu media penyimpanan.

Citra merupakan istilah lain dari gambar yang merupakan komponen multimedia yang memegang peranan sangat penting sebagai bentuk informasi visual. Citra mempunyai karakteristik yang tidak dimiliki oleh data teks, yaitu kaya akan informasi. Citra digital adalah citra hasil digitalisasi citra kontinu (analog). Tujuan dibuatnya citra digital adalah agar citra tersebut dapat diolah menggunakan komputer atau piranti digital.

Citra merupakan salah satu komponen multimedia yang memegang peranan penting sebagai bentuk informasi visual, karena karakteristiknya yang kaya dengan informasi. Secara harfiah, citra adalah gambar pada bidang dwimatra (dua dimensi). Secara matematis citra adalah gambar pada bidang dwimatra (dua dimensi) yang dihasilkan dari gambar analog dua dimensi yang kontinu menjadi gambar diskret melalui proses sampling. Gambar analog dibagi menjadi  $N$  baris dan  $M$  kolom sehingga menjadi gambar diskret. Persilangan antara baris dan kolom tertentu disebut dengan piksel. Contohnya adalah gambar/titik diskret pada baris  $n$  dan kolom  $m$  disebut dengan piksel  $[n,m]$ .

## 2.2 Pengertian Citra Digital

Munir (2004), Citra ada dua macam yaitu citra kontinu dan citra diskrit. Citra kontinu dihasilkan dari sistem optik yang menerima sinyal analog, contohnya mata manusia, kamera analog. Citra diskrit dihasilkan dari proses digitalisasi terhadap citra kontinu contohnya kamera digital, scanner.

Putra (2010), secara umum pengolahan citra digital menunjuk pada pemrosesan gambar 2 dimensi menggunakan komputer. Dalam konteks yang lebih luas, pengolahan citra digital mengacu pada setiap data 2 dimensi. Citra digital merupakan sebuah larik (array) yang berisi nilai-nilai real maupun kompleks yang dipresentasikan dengan deretan bit tertentu.

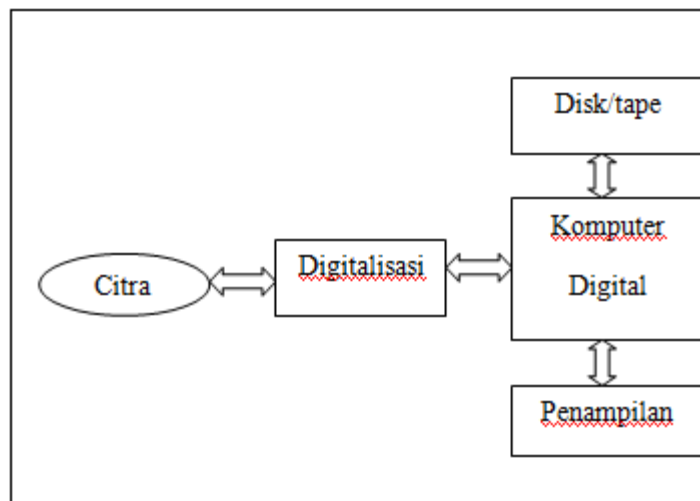
Yudistira (2010), pengolahan citra adalah pemrosesan citra, khususnya dengan menggunakan komputer, menjadi citra yang kualitasnya lebih baik.

Tujuan utama pengolahan citra adalah agar citra yang mengalami gangguan mudah diinterpretasi oleh manusia maupun mesin (komputer). Teknik pengolahan citra digital adalah mentransformasikan citra dua dimensi menjadi citra lain dengan menggunakan komputer. Proses ini mempunyai ciri data masukan dan informasi keluaran yang berbentuk citra. Jadi masukannya berupa citra dan keluarannya juga berbentuk citra, dengan kualitas yang lebih baik dari citra masukan. Beberapa contoh operasi pengolahan citra adalah pengubahan kontras citra, penghilangan derau (noise) dengan operasi penapisan (filtering), penajaman (sharpening), pemberian warna semu (pseudocoloring), dan sebagainya. Operasi-operasi tersebut akan diterapkan pada pengolahan citra apabila:

1. Digunakan untuk meningkatkan kualitas penampakan atau menonjolkan beberapa aspek informasi yang terkandung dalam citra dengan perbaikan atau modifikasi citra;
2. Perlu pengelompokkan, pencocokan atau pengukuran elemen pada citra;
3. Perlu penggabungan sebagian citra dengan bagian citra lainnya.

### 2.3 Pengolahan Citra Digital

Berikut ini adalah blok sistem pengolahan citra digital:



**Gambar 2.1** Diagram blok sistem pengolahan citra digital

Wijaya dan Prijono (2007), Digitalisasi citra yaitu mengubah citra masukan menjadi sinyal listrik dan kemudian mencuplikan sinyal tersebut dengan menggunakan *A/D Converter (Analog to Digital Converter)*. Digitalisasi ini dapat berupa scanner atau kamera digital yang mengubah citra kontinue kedalam suatu representasi numerik, sehingga citra ini dapat diproses oleh komputer digital. Jadi digitalisasi citra adalah proses mengubah citra analog menjadi citra digital.

Proses pengolahan data dapat dilakukan oleh komputer, baik berupa mikrokomputer sederhana (*microprocessor based computer*) atau komputer besar (*mainframe computer*), tergantung jumlah data dan jenis pengolahan. Proses penampil data merupakan salah satu segi yang penting karena bagaimanapun juga citra digital hasil olahan harus dapat dinilai oleh mata manusia melalui suatu penampil (*display*). Penampil yang digunakan biasanya berupa suatu *graphic* monitor atau suatu *graphic* printer/plotter.

Hendriyani (2012), *Image processing* adalah suatu teknologi yang berfungsi untuk menyelesaikan masalah mengenai pengolahan citra. Tujuannya yaitu untuk mengolah citra sedemikian rupa, sehingga citra tersebut lebih mudah untuk diproses lebih lanjut atau diedit.

## 2.4 Jenis – jenis Citra Digital

Sutoyo et. al. (2009), menjelaskan jenis-jenis citra digital adalah sebagai berikut:

### 1. Citra Biner (Monokrom)

Citra biner adalah citra digital yang hanya memiliki dua kemungkinan nilai piksel yaitu hitam (0) dan putih (1). Citra biner juga disebut sebagai citra bw (Black and White) atau citra monokrom.

### 2. Citra *Grayscale* (skala keabuan)

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 menunjukkan tingkat intensitas. Warna

yang dimiliki citra *grayscale* adalah warna keabuan dengan berbagai tingkatan dari hitam hingga putih. Citra *grayscale* dapat diperoleh dari citra RGB. Nilai intensitas citra *grayscale* (keabuan) dihitung dengan nilai intensitas citra RGB dengan menggunakan persamaan:

$$\text{Nilai keabuan} = 0.2989 * R + 0.5870 * G + 0.1140 * B$$

Dimana,

R : Nilai intensitas warna merah

G : Nilai intensitas warna hijau

B : Nilai intensitas warna biru

### 3. Citra Warna (True Color)

Citra warna sering disebut juga citra RGB atau citra *true color* karena dapat mempresentasikan warna objek menyerupai warna aslinya dengan mengkombinasikan ketiga warna dasar yaitu merah/*red* (R), hijau/*green* (G), dan biru/*blue* (B). Tiap piksel memiliki tiga nilai kanal yang mewakili tiap komponen dasar citra.

## 2.5 Elemen – elemen Dasar Citra

Sutoyo et. al. (2009), menjelaskan elemen-elemen dasar citra, yaitu:

### a. Kecerahan (*Brightness*)

Yang dimaksud dengan kecerahan (*brightness*) adalah intensitas cahaya yang dipancarkan oleh piksel dari citra yang dapat ditangkap oleh *system* penglihatan. Kecerahan pada sebuah titik (piksel) didalam citra merupakan intensitas rata – rata dari suatu area yang melingkupinya.

### b. Kontras (*contrast*)

Kontras (*contrast*) menyatakan sebaran terang dan gelap dalam sebuah citra. Pada citra yang baik, komposisi gelap dan terang tersebar secara merata.

### c. Kontur (*contour*)

Yang dimaksud dengan kontur (*contour*) adalah keadaan yang ditimbulkan oleh perubahan intensitas pada piksel – piksel yang bertetangga. Karena adanya perubahan intensitas inilah mata mampu mendeteksi tepi – tepi objek dialam citra.

d. Warna (*color*)

Warna (*color*) adalah persepsi yang ditangkap sistem visual terhadap perubahan panjang gelombang cahaya yang dipantulkan oleh objek. Setiap warna memiliki panjang gelombang yang berbeda-beda. Warna merah memiliki panjang gelombang ( $\lambda$ ) yang paling tinggi, sedangkan warna violet mempunyai panjang gelombang ( $\lambda$ ) yang paling rendah.

e. Bentuk (*shape*)

Bentuk adalah properti intrinsik dari objek 3 dimensi, dengan pengertian bahwa bentuk merupakan properti intrinsik utama untuk sistem visual manusia.

f. Tekstur (*texture*)

Tekstur (*texture*) dicirikan sebagai distribusi spasial dari derajat keabuan didalam sekumpulan piksel-piksel bertetangga. Tekstur adalah sifat – sifat atau karakteristik yang dimiliki oleh suatu daerah yang cukup besar sehingga secara alami sifat – sifat tadi dapat berulang dalam daerah tersebut. Tekstur adalah keteraturan pola – pola tertentu yang terbentuk dari susunan piksel – piksel dalam citra digital. Informasi tekstur dapat digunakan untuk membedakan sifat– sifat permukaan suatu benda dalam citra yang berhubungan dengan kasar dan halus, juga sifat – sifat spesifik dari kekasaran dan kehalusan permukaan tadi, yang sama sekali terlepas dari warna permukaan tersebut.

## 2.6 Operasi Pengolahan Citra

Wijaya dan Prijono (2007), menjelaskan operasi-operasi yang dilakukan dalam pengolahan citra banyak ragamnya. Namun, secara umum pada pengolahan citra terdapat enam jenis operasi pengolahan, yaitu:

1. Peningkatan kualitas citra (*image enhancement*)

Jenis operasi ini bertujuan untuk memperbaiki kualitas citra dengan cara memanipulasi parameter-parameter citra. Dengan operasi-operasi ini, ciri-ciri khusus yang terdapat didalam citra lebih ditonjolkan. Contoh – contoh operasi peningkatan kualitas citra:

a. Perbaiki kontras gelap/terang

- b. Perbaikan tepian objek (*Edge enhancement*)
- c. Penajaman (*Sharpening*)
- d. Pemberian warna semu (*Pseudocoloring*)
- e. Penapisan derau (*Noise filtering*)

## 2. Restorasi citra (*image restoration*)

Operasi ini bertujuan menghilangkan / meminimumkan cacat pada citra. Tujuan restorasi citra hampir sama dengan operasi peningkatan kualitas citra. Bedanya, pada restorasi citra penyebab degradasi gambar diketahui. Contoh – contoh operasi restorasi citra:

- a. Penghilangan derau (*noise*)
- b. Penghilangan kesamaran (*deblurring*)

## 3. Kompresi 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 kompresi citra adalah citra yang telah dikompresikan harus tetap mempunyai kualitas gambar yang bagus. Contoh metode kompresi citra adalah metode JPEG. Misalkan ada citra kapal yang berukuran 258 KB. Hasil kompresi citra dengan metode JPEG dapat mereduksi ukuran citra semula sehingga menjadi 49KB saja.

## 4. Segmentasi citra (*image segmentation*)

Operasi ini adalah suatu tahap pada proses analisis citra yang bertujuan untuk memperoleh informasi yang ada dalam citra tersebut dengan membagi citra kedalam daerah-daerah terpisah dimana setiap daerah adalah homogen dan mengacu pada sebuah kriteria keseragaman yang jelas. Segmentasi yang dilakukan pada citra harus tepat agar informasi yang terkandung didalamnya dapat diterjemahkan dengan baik.

## 5. Analisis citra (*image analysis*)

Jenis operasi ini bertujuan menghitung besaran kuantitatif dari citra untuk menghasilkan deskripsinya. Teknik analisis citra mengekstraksi ciri-ciri tertentu yang membantu dalam identifikasi objek. Proses segmentasi

kadangkala diperlukan untuk melokalisasi objek yang diinginkan dari sekelilingnya. Contoh – contoh operasi analisis 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.

## 2.7 Peningkatan Mutu Citra

Wijayanto, (2001), peningkatan mutu citra dilakukan untuk memperoleh keindahan citra yang akan digunakan untuk kepentingan analisis citra. Proses peningkatan mutu citra bertujuan untuk memperoleh citra yang dapat memberikan informasi sesuai dengan tujuan/kepentingan pengolahan citra. Proses peningkatan mutu citra ini termasuk memperbaiki citra yang ketika proses akuisisi mengalami gangguan yang signifikan seperti noise, gangguan geometris, radiometrik dan beberapa gangguan faktor alam lainnya. Secara umum domain dalam peningkatan mutu citra ini dapat dilakukan secara spatial dan frekuensi. Domain Spatial melakukan manipulasi nilai pixel secara langsung dengan dipengaruhi oleh nilai pixel lainnya secara spatial sedangkan domain frekuensi berdasarkan frekuensi spektrum citra.

Domain Spasial merupakan teknik peningkatan mutu citra yang melakukan manipulasi langsung pixel (x,y) suatu citra dengan menggunakan fungsi transformasi:  $g(x, y) = T[f(x, y)]$ , dimana  $f(x, y)$  sebagai citra input,  $g(x, y)$  hasil citra yang sudah diproses dan T adalah operator pada f yang didefinisikan berdasarkan beberapa lingkungan di (x, y). Teknik ini ditunjukkan pada gambar 5.2 Masking/Filter suatu pixel (x,y) ditentukan berdasarkan pixel tetangganya yang didefinisikan sebagai bentuk bujur sangkar (sering digunakan) ataupun circular sebagai sub-citra yang berpusat



di titik  $(x,y)$  dengan ukuran lebih dari  $1 \times 1$  (gambar 5.3 menggunakan masking  $3 \times 3$ ). Pusat sub-citra berpindah dari satu pixel ke pixel lainnya dimulai dari pojok atas. Nilai koefisien masking ditentukan berdasarkan prosesnya. Teknik masking digunakan untuk penajaman citra dan penghalusan citra.

Edge detection adalah pendekatan yang paling umum digunakan untuk mendeteksi diskontinuitas graylevel. Hal ini disebabkan karena titik ataupun garis yang terisolasi tidak terlalu sering dijumpai dalam aplikasi praktis. Suatu edge adalah batas antara dua region yang memiliki graylevel yang relatif berbeda. Pada dasarnya ide yang ada di balik sebagian besar teknik edge-detection adalah menggunakan perhitungan local derivative operator. Gradien dari suatu citra  $f(x,y)$  pada lokasi  $(x,y)$  adalah vector.

## 2.8 Citra Biner

Citra biner adalah citra yang hanya mempunyai dua nilai derajat keabuan yaitu, hitam dan putih. Piksel-piksel objek bernilai '1' dan piksel-piksel latar belakang bernilai '0'. Pada saat menampilkan gambar, '0' adalah putih dan '1' adalah hitam. Meskipun saat ini citra berwarna lebih disukai karena memberi kesan yang lebih kaya daripada citra biner, namun tidak membuat citra biner mati. Pada beberapa aplikasi citra biner masih tetap dibutuhkan, misalnya citra logo instansi, citra kode batang (bar code) yang tertera pada label barang, citra hasil pemindai dokumen teks, dan sebagainya.

Adapun alasan penggunaan citra biner adalah karena citra biner masih memiliki keunggulan sebagai berikut:

1. Kebutuhan memori kecil karena nilai derajat keabuan hanya membutuhkan representasi 1 bit.
2. Waktu pemrosesan lebih cepat dibandingkan dengan citra aras keabuan (greyscale).

## 2.9 Pengertian Ciri (*Feature*) dan Pola (*Pattern*)

Ciri adalah segala jenis aspek pembeda, kualitas atau karakteristik. Ciri bisa berwujud simbolik (misalnya warna) atau numerik (misalnya tinggi). Ciri yang bagus adalah ciri yang memiliki daya pembeda yang tinggi, sehingga pengelompokan pola berdasarkan ciri yang dimiliki dapat dilakukan dengan keakuratan yang tinggi.

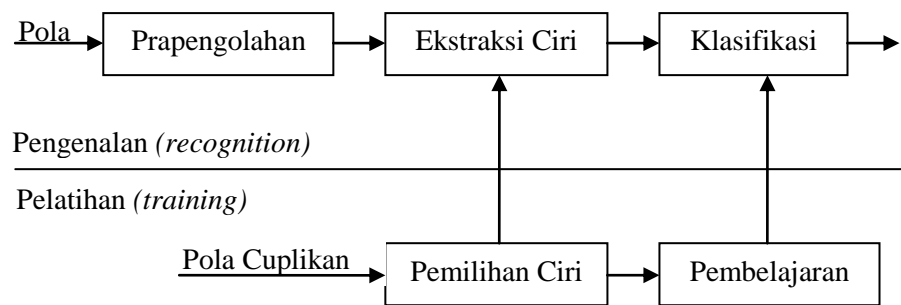
Sebagai contoh, segitiga yang memiliki ciri yaitu: memiliki tiga buah titik sudut, atau lingkaran yang memiliki ciri yaitu: jari-jari yang besarnya konstan. Citra pada suatu pola diperoleh dari hasil pengukuran terhadap objek uji. Khusus pada pola yang didapat di dalam citra, ciri-ciri yang diperoleh berasal dari:

- a. Spasial: intensitas piksel, histogram dan sebagainya.
- b. Tepi: Arah, kekuatan dan sebagainya.
- c. Kontur: garis, elips, lingkaran dan sebagainya.
- d. Wilayah/ bentuk: keliling, luas dan sebagainya.
- e. Hasil Transformasi Fourier: frekuensi dan sebagainya

## 2.10 Sistem Pengenalan Pola dengan Pendekatan Statistik

Pengenalan pola bertujuan untuk menentukan kelompok atau kategori pola berdasarkan ciri-ciri yang dimiliki oleh pola tersebut. Dengan kata lain, pengenalan pola membedakan suatu objek dengan objek lain.

Sistem pengenalan pola dengan pendekatan statistik menggunakan teori-teori ilmu peluang. Ciri-ciri yang dimiliki oleh suatu pola ditentukan distribusi statistiknya, dimana pola yang berbeda memiliki distribusi yang berbeda pula. Dengan menggunakan teori keputusan di dalam statistik, kita menggunakan distribusi ciri untuk mengklasifikasikan pola. Adapun diagram kotaknya dapat dilihat pada Gambar 2.2



**Gambar 2.2** Sistem pengenalan pola dengan pendekatan statistik

Ada dua fase dalam sistem pengenalan pola, yaitu fase pelatihan dan fase pengenalan. Pada fase pelatihan, beberapa contoh citra dipelajari untuk dapat menentukan ciri yang akan digunakan dalam proses pengenalan serta prosedur klasifikasinya. Pada fase pengenalan, citra diambil cirinya, kemudian ditentukan kelas kelompoknya. Adapun penjelasan dari diagram kotak pada Gambar 2.2 adalah:

1. Prapengolahan

Proses awal yang dilakukan untuk memperbaiki kualitas citra (*image enhancement*) dengan menggunakan teknik-teknik yang ada.

2. Ekstraksi Ciri

Proses mengambil ciri-ciri yang terdapat pada objek di dalam citra. Pada proses ini objek di dalam citra perlu dideteksi seluruh tepinya, lalu menghitung properti-properti objek yang berkaitan sebagai ciri.

3. Klasifikasi

Proses pengelompokan objek ke dalam kelas yang sesuai.

4. Pemilihan Ciri

Proses memilih ciri pada suatu objek agar diperoleh ciri yang optimum, yaitu ciri yang dapat membedakan suatu objek dengan objek lainnya

5. Pembelajaran

Proses belajar membuat aturan klasifikasi sehingga jumlah kelas yang tumpang tindih dibuat sekecil mungkin.

## 2.11 Jadwal Penelitian

Jadwal pengembangan aplikasi yang meliputi persiapan pelaksanaan dan penyusunan penelitian disajikan dalam bentuk Tabel 2.1 dibawah ini :

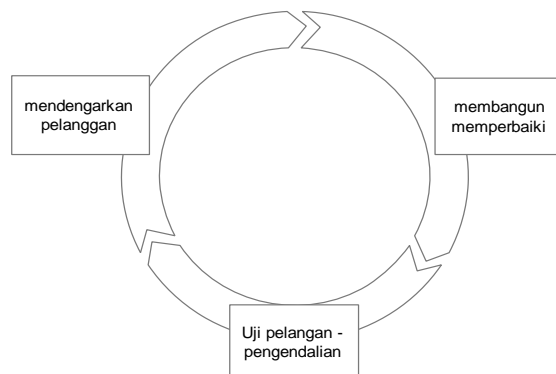
**Tabel 2.1** Penjadwalan

No	Kegiatan	November				Desember				Januari				Februari	
		1	2	3	4	1	2	3	4	1	2	3	4	1	2
1	<b>Pra-Penelitian</b>														
	a. Identifikasi Masalah														
	b. Studi Pustaka														
	c. Konsultasi Bimbingan														
2	<b>Penelitian</b>														
	a. Pembuatan Desain Program														
	b. Pembuatan Program														
	c. Test program														
3	<b>Kesimpulan</b>														
	Penulisan Skripsi														

## 2.12 Metode *Prototype Model*

Pressman (2002), menyatakan bahwa *prototype model* merupakan metode yang efektif dalam merancang perangkat lunak. *Prototype model* dimulai dengan mengumpulkan kebutuhan. Pengembang dan pelanggan bertemu dan mendefinisikan *object* keseluruhan dari perangkat lunak, mengidentifikasi segala kebutuhan yang diketahui dan kemudian melakukan “perancangan kilat”. Perancangan kilat berfokus pada penyajian dari aspek-aspek perangkat lunak tersebut yang akan nampak bagi pelanggan atau pemakai (contohnya pendekatan input dan format output). Perancangan kilat membawa kepada konstruksi sebuah *prototype*. *Prototype* tersebut dievaluasi

oleh pelanggan dan dipakai untuk menyaring kebutuhan pengembangan perangkat lunak.



**Gambar 2.3** *Prototipe Paradigm*

*Prototype model* juga dapat didefinisikan sebagai proses pengembangan suatu prototipe secara cepat untuk digunakan terlebih dahulu dan ditingkatkan terus menerus sampai didapatkan sistem yang utuh. *Prototype model* merupakan proses yang digunakan untuk membantu pengembang perangkat lunak dalam membentuk *prototype* dari perangkat lunak yang harus dibuat. Proses pada model *prototyping* dapat dijelaskan sebagai berikut:

- 1) Pengumpulan kebutuhan : *Developer* dan klien bertemu dan menentukan tujuan umum, kebutuhan yang diketahui dan gambaran bagian-bagian yang akan dibutuhkan berikutnya.
- 2) Perancangan : Perancangan dilakukan cepat dan rancangan mewakili semua aspek perangkat lunak yang diketahui, dan rancangan ini menjadi dasar pembuatan *prototype*.
- 3) Evaluasi *prototype* : Klien mengevaluasi *prototype* yang dibuat dan digunakan untuk memperjelas kebutuhan perangkat lunak.

Perulangan ketiga proses ini terus berlangsung hingga semua kebutuhan terpenuhi. *Prototype-prototype* dibuat untuk memuaskan kebutuhan klien dan untuk membangun perangkat lunak lebih cepat, namun tidak semua *prototype* bisa dimanfaatkan. Demi kebutuhan klien lebih baik *prototype* yang dibuat diusahakan dapat dimanfaatkan.

### 2.13 UML (*Unified Markup Language*)

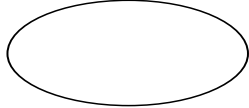

Rosa dan Shalahuddin (2014) menyatakan UML adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak. UML merupakan metodologi dalam mengembangkan sistem berorientasi object dan juga merupakan alat untuk mendukung pengembangan sistem. UML saat ini banyak dipergunakan dalam dunia industri yang merupakan standar bahasa pemodelan umum pada dunia industri perangkat lunak dan pengembangan software.

Alat bantu yang dipergunakan dalam perancangan ber orientasi objek berbasis UML adalah sebagai berikut :



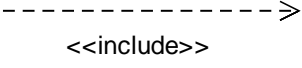
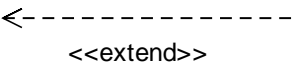
#### 2.13.1 *Use Case Diagram*

*Use case diagram* merupakan pemodelan untuk kelakuan (*behavior*) sistem yang dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih actor dengan sistem yang akan dibuat. Dapat dikatakan *use case* digunakan untuk mengetahui fungsi apa saja yang dibuat dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Berikut simbol yang digunakan pada *use case diagram* :

**Tabel 2.2** Simbol *Use Case Diagram*

Gambar	Keterangan
	<p><i>Use case</i> menggambarkan fungsionalitas yang disediakan <i>system</i> sebagai unit-unit yang bertukar pesan antar unit dengan <i>actor</i>, biasanya dinyatakan dengan menggunakan kata kerja diawal nama <i>use case</i>.</p>
 <p data-bbox="517 1892 587 1921">Actor</p>	<p><i>Actor</i> adalah orang lain atau <i>system</i> lain yang mengaktifkan fungsi dari target <i>system</i>. Untuk mengidentifikasi <i>actor</i>, harus ditentukan pembagian kerja dan tugas-tugas yang berkaitan dengan peran pada target <i>system</i>. Orang atau <i>system</i> bias muncul dalam beberapa peran.</p>





**Tabel 2.2** (lanjutan)

	Ososiasi antara <i>actor</i> dan <i>use case</i> yang menggunakan panah terbuka untuk mengindikasikan bila <i>actor</i> berinteraksi secara pasif dengan <i>system</i> .
	Ososiasi antara <i>actor</i> dan <i>use case</i> yang menggunakan garis untuk mengindikasikan siapa atau apa yang meminta interaksi langsung bukan mengindikasikan aliran data.
	<i>Include</i> merupakan pemanggilan <i>use case</i> oleh <i>use case</i> lainnya , contohnya pemanggilan sebuah fungsi program.
	Perluasan dari <i>use case</i> lain jika kondisi atau syarat terpenuhi.


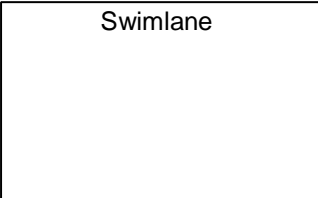
### 2.13.2 Activity Diagram

Secara grafis digunakan untuk menggambarkan rangkaian aliran aktivitas baik proses bisnis maupun *use case*. *Activity diagram* dapat juga digunakan untuk memodelkan *action* yang akan dilakukan saat sebuah operasi dieksekusi, dan memodelkan hasil dari *action* tersebut. Simbol-simbol yang digunakan dalam *activity diagram* yaitu :

**Tabel 2.3** Simbol *Activity Diagram*

Gambar	Keterangan
	Mengambarkan suatu proses/kegiatan sistem
	Menunjukkan eksekusi dari suatu aksi
	Start Point merupakan awal dari suatu aktivitas
	End point, akhir aktivitas





**Tabel 2.3** (lanjutan)

	Fork/Rake Node Satu aliran atau beberapa aliran yang pada tahap tertentu berubah menjadi satu atau beberapa aliran.
	Swimlane, pembagian activity diagram untuk menunjukkan siapa melakukan apa

### 2.13.3 Sequence Diagram

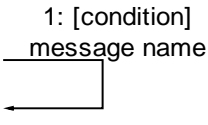
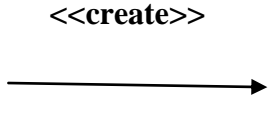
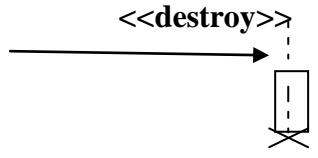

*Sequence diagram* menggambarkan kelakuan objek pada *use case* dengan mendeksripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek. Simbol-simbol yang digunakan dalam *sequence diagram*, yaitu :

**Tabel 2.4** Simbol *Sequence Diagram*

Gambar	Keterangan
	Orang, proses atau <i>system</i> lain yang berinteraksi dengan <i>system</i> .
<p>1: Masukan</p> 	Menyatakan bahwa suatu objek mengirimkan data atau masukan atau informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim.
	Activation, activation mewakili sebuah eksekusi operasi dari objek.
<p>1: Keluaran</p> 	<i>Message return</i> , simbol membalas pesan



Tabel 2.4 (lanjutan)

	<i>Recursive</i> , menggambarkan pengiriman pesan untuk dirinya sendiri.
	Menyatakan suatu objek membuat objek yang lain, arah panah mengarah objek yang di buat.
	Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada <i>create</i> maka ada <i>destroy</i> .
<p style="text-align: center;"><b>Garis Hidup / Lifeline</b></p> 	Menyatakan kehidupan suatu objek

## 2.14 Pengertian MATLAB

Matlab adalah singkatan dari MATrix LABoratory, merupakan bahasa pemrograman yang dikembangkan oleh The Mathwork Inc. yang hadir dengan fungsi dan karakteristik yang berbeda dengan bahasa pemrograman lain yang sudah ada lebih dahulu seperti Delphi, Basic maupun C++. Matlab merupakan bahasa pemrograman level tinggi yang dikhususkan untuk kebutuhan komputasi teknis, *visualisasi* dan pemrograman seperti komputasi matematik, analisis data, pengembangan algoritma, simulasi dan pemodelan dan grafik-grafik perhitungan. Pada awalnya Matlab dibuat untuk memberikan kemudahan mengakses data matrik pada proyek LINPACK dan EISPACK. Saat ini matlab memiliki ratusan fungsi yang dapat digunakan sebagai *problem solver* baik permasalahan yang mudah maupun masalah-masalah yang kompleks dari berbagai disiplin ilmu.

Beberapa kelebihan Matlab jika dibandingkan dengan program lain seperti Fortran, dan Basic adalah :

- a. Mudah dalam memanipulasi struktur matriks dan perhitungan berbagai operasi matriks yang meliputi penjumlahan, pengurangan, perkalian, invers dan fungsi matriks lainnya.
- b. Menyediakan fasilitas untuk memplot struktur gambar (kekuatan fasilitas grafik tiga dimensi yang sangat memadai).
- c. *Script* program yang dapat diubah sesuai dengan keinginan *user*.
- d. Jumlah *routine-routine powerful* yang berlimpah dan terus berkembang.
- e. Kemampuan *interface* (misal dengan bahasa C, *word* dan *mathematica*).
- f. Dilengkapi dengan *toolbox*, *simulink*, *stateflow* dan sebagainya, serta mulai melimpahnya *source code* di internet yang dibuat dalam matlab (contoh *toolbox* misalnya : *signal processing*, *control system*, *neural networks* dan sebagainya).