BAB 2
LANDASAN TEORI

2.1 Pengolahan Citra (*Image Processing*)

Pengolahan citra merupakan proses pengolahan suatu citra (gambar) yang dilakukan secara komputerisasi, untuk menghasilkan suatu citra lain. Pengolahan suatu citra mempunyai dua tujuan utama, diantaranya :

1. Memperbaiki kualitas citra agar mudah diinterprestasi oleh manusia atau komputer, dimana citra yang dihasilkan menampilkan informasi yang jelas. Dan manusia berperan sebagai pengolah informasi.
2. Mengektraksi informasi ciri yang menonjol pada suatu citra yang akan menghasilkan informasi dari citra itu sendiri berupa numerik atau dengan kata lain komputer (mesin) melalui besaran-besaran data yang dapat dibedakan secara jelas.

Image Processing dan *Computer Vision* merupakan perkembangan komputer yang digunakan sebagai pengganti mata manusia dengan perangkat input *Image capture* seperti kamera dan *scanner* sebagai mata dan komputer berperan sebagai otak yang mengolah informasi. Oleh sebab itu muncul berupa pecahan bidang dalam *Computer Vision* antara lain :

1. *Pattern recognition* (pengenalan pola).
2. *Biometric* (pengenalan identifikasi berdasarkan ciri biologis pada manusia).
3. *Content based Image and video retrieval*.
4. *Video editing* dan lain lain.

2.2 Citra Digital

Secara umum.Pengolahan citra digital menunjukan pada pemrosesan gambar dua dimensi menggunakan *Computer*. Dalam konteks yang lebih luas, pengolahan citra digital mengacu pada pemrosesan setiap data 2 dimensi. Citra digital merupakan sebuah larik (*array*) yang berisi nilai-nilai *real* maupun

kompleks yang direpresentasikan dengan deretan bit tertentu.

Citra secara harafiah, adalah gambar pada bidang dua dimensi (dwimatra). Ditinjau dari sudut pandang matematis, citra merupakan fungsi menerus (*continue*) dari intensitas cahaya pada bidang dwimatra. Sedangkan citra digital adalah citra yang dapat diolah oleh *Computer*[3].

Suatu citra dapat didefinisikan sebagai fungsi f(x,y) berukuran M baris dan N kolom, dengan x dan y adalah koordinat spasial, dan *amplitude* f di titik koordinat (x,y) dinamakan intensitas atau tingkat keabuan dari citra pada titik tersebut. Apabila nilai x,y, dan nilai amplitudo f secara keseluruhan berhingga (*finite*) dan bernilai diskrit maka dapat dikatan bahwa citra tersebut citra digital. Sebuah citra dapat dibentuk dalam bentuk persamaan 2.1.

$$f(x,y) = \begin{bmatrix} f(0,0) & f(0,0) & \cdots & f(0,N-1) \\ f(1,0) & f(1,1) & \cdots & f(1,N-1) \\ \vdots & \vdots & & \vdots \\ f(M-1,0) & f(M-1,1) & \cdots & f(M-1,N-1) \end{bmatrix} \quad \text{.....(2.1)}$$

Nilai pada suatu irisan antara baris dan kolom (pada posisi x,y) disebut dengan *picture elements*, *Image elements, pels, atau pixel*. Istilah terakhir (*pixel*) paling sering digunakan pada citra digital.

2.2.1. Karakteristik Citra Digital

Setiap citra digital memiliki beberapa karakteristik, antara lain yaitu :

1. Ukuran citra digital

Ukuran citra (*Image size*) menyatakan ukuran banyaknya *pixel* penyusun citra raster yang dinyatakan dalam matrik 2 dimensi, yaitu (X × Y) *Pixel*, dimana X menyatakan ukuran banyaknya *pixel* perbaris pada arah horizontal sedangkan Y menyatakan ukuran banyaknya *pixel* perkolom pada arah *vertikal*. Sebagai contoh, Citra digital berukuran 800 × 600 *pixel*, terdiri dari 800 × 600 *pixel* = 480.000 *pixel*, dengan susunan 800 *pixel* setiap baris pada arah *horizontal* dan 600 *pixel* setiap kolom pada arah *vertikal*.

2. Resolusi

Atribut citra digital yang tidak kalah pentingnya adalah resolusi (*resolution*), yang didefinisikan sebagai banyaknya *pixel* dalam setiap satuan panjang. Umumnya, resolusi dinyatakan dalam satuan dpi (*dot per inch*). Sebagai contoh, citra digital yang memiliki resolusi 75 dpi, berarti terdiri dari 75 dot (titik) pada setiap inchi. Semakin tinggi resolusi suatu citra digital, maka kualitasnya akan semakin baik.

2.2.2. Tipe Citra Digital

Berdasarkan format penyimpanan nilai warnanya citra digital terbagi dalam tiga tipe yaitu :

1. Citra Biner

Citra yang hanya memiliki dua nilai yaitu 1 dan 0. Dinyatakan dalam persamaan 2.2

$$f(x,y) \Sigma \{0,1\} \text{..... } (2.2)$$

2. Citra Gray-scale

Citra yang terdiri dari satu layer warna dengan derajat keabuan tertentu. Untuk kebanyakan citra digital 8 bit, maka sistem Gray-scale diukur berdasarkan skala intensitas kecerahan, yang bernilai 0 – 255, dimana yang hitam pekat adalah 0 dan yang terputih adalah 255. Dinyatakan dalam persamaan 2.3 :

$$f(x,y) \Sigma [0...255] \text{ ..... } (2.3)$$

3. Citra Berwarna

Citra standar (Bitmap) dengan ukuran 512 × 512 dan mempunyai kedalaman warna 24 bit atau dikenal dengan istilah true color, dalam hal ini kedalaman warna yang dimaksud yaitu jumlah bit yang dipakai untuk merepresentasikan tiap titik pada gambar tersebut dinyatakan dalam bit per *pixel* . komputer mempunyai tiga kom*purchase order*nen warna yaitu, merah, hijau, dan biru. Ketiga kom*purchase order*nen dasar ini dikenal dengan istilah warna RGB

(*RedGreenBlue*). Untuk citra digital dengan 8 bit per *pixel* mempunyai 256 warna sedangkan pada citra dengan 24 bit per *pixel* mempunyai 16 juta warna, tiap *pixel* dinyatakan dengan :

Bit ke-0 sampai ke-7 untuk warna merah

Bit ke-8 sampai ke-15 untuk warna hijau

Bit ke-16 sampai ke-23 untuk warna biru

Sehingga kombinasi warnanya adalah $256^3 + 256^2 + 256^1 = 16.843.008$. Warna-warna ini merupakan warna dasar penyusun setiap *pixel* pada citra digital. Representasi dalam citra digital dinyatakan dalam persamaan 2.4, 2.5,dan 2.6:

$$fR(x,y)\ \Sigma\ [0...255]..... \quad (2.4)$$

$$fG(x,y)\ \Sigma\ [0...255]..... \quad (2.5)$$

$$fB(x,y)\ \Sigma\ [0...255]..... \quad (2.6)$$

2.2.1 2.2.3. Format File Citra Digital

Beberapa contoh format umum, pada citra digital yaitu Bitmap (BMP), *Joint Photographic Group Experts* (JPEG), *Graphics Interchange Format* (GIF), dan *Purchase orderrtable Network Graphics* (PNG).

1. Bitmap

Bitmap merupakan format baku citra pada sistem operasi *windows* dan IBM OS/2. Citra berformat BMP merupakan citra yang tidak terkompresi, sehingga pada umumnya citra berformat BMP mempunyai ukuran yang relatif lebih besar dibandingkan dengan forman citra lainnya. Intensitas *pixel* dari citra berformat BMP dipetakan ke sejumlah bit tertentu. Panjang setiap *pixel* pada bitmap yaitu 4 bit, 8 bit, sampai 24 bit yang merepresentasikan nilai intensitas *pixel*. Dengan demikian ada sebanyak 28 = 256 derajat keabuan, mulai dari 0 sampai 255[4].

2. *Joint Photographic Group Experts* (JPEG)

Joint Photographic Group Experts (JPEG) merupakan standar kompresi file yang dikembangkan oleh *Group Joint Photographic Experts* menggunakan kombinasi DCT dan pengkodean Huffman untuk mengkompresikan citra. Citra JPEG merupakan citra terkompresi yang bersifat lossy, artinya citra tidak bisa dikembalikan ke bentuk aslinya.Citra ini memiliki ukuran yang relatif lebih kecil dibandingkan dengan citraberformat BMP karena telah terkompresi.

3. *Graphics Interchange Format* (GIF)

Graphics Interchange Format (GIF) merupakan format citra terkompresi. Ukuran dari citra berformat GIF merupakan yang terkecil dari semua format citra digital. Kombinasi warna yg tersedia sebanyak 256 warna. Citra jenis ini banyak dugunakan untuk keperluan website, untuk membuat sebuah icon, gambar dan keperluan lainnya yang mengharuskan menggunakan citra dengan ukuran yang kecil. Citra jenis ini juga dapat dibuat animasi.

4. *Portable Network Graphics* (PNG)

Portable Network Graphics (PNG) adalah salah satu format penyimpanan citra yang menggunakan metode kompresi yang tidak menghilangkan bagian dari citra tersebut (*lossless compression*). Citra berformat PNG merupakan salah satu format yang baik untuk digunakan pengolahan citra, karena format ini selain tidak menghilangkan bagian dari citra yang sedang diolah.

2.3 Citra *Grayscale*

Citra digital *Grayscale* atau greyscale adalah suatu citra dimana nilai dari setiap *pixel* merupakan sample tunggal [5]. Citra yang ditampilkan dari citra jenis ini terdiri atas warna abu-abu, bervariasi pada warna hitam pada bagian yang intensitas terlemah dan warna putih pada intensitas terkuat. Citra *Grayscale* berbeda dengan citra ”hitam-putih”, dimana pada konteks komputer, citra hitam putih hanya terdiri atas 2 warna saja yaitu ”hitam” dan ”putih” saja. Pada citra *Grayscale* warna bervariasi antara hitam dan putih, tetapi variasi warna diantaranya sangat banyak. Citra *Grayscale* seringkali merupakan perhitungan

dari intensitas cahaya pada setiap *pixel* pada spektrum elektromagnetik single band.

Citra *Grayscale* disimpan dalam format 8 bit untuk setiap sample *pixel*, yang memungkinkan sebanyak 256 intensitas. Format ini sangat membantu dalam pemrograman karena manupulasi bit yang tidak terlalu banyak. Pada aplikasi lain seperti pada aplikasi *medical imaging* dan *remote sensing* biasa juga digunakan format 10,12 maupun 16 bit.

Untuk mendapatkan nilai *Grayscale* dari sebuah citra, dapat dilakukan dengan persamaan 2.7 seperti berikut :

$$grayscale = \frac{(R+G+B)}{3} \text{ ..... (2.7)}$$

Dengan :

R = nilai *pixel* matriks R.

G = nilai *pixel* matrik G.

B = nilai *pixel* matrik B.

Misal terdapat suatu pixel dengan nilai R=100, G = 100 dan B = 100. Untuk mendapatkan nilai *Grayscale* pada pixel ini menggunakan rumus (2.7) maka caranya adalah sebagai berikut.

$$grayscale = \frac{(100+100+100)}{3}$$

$$= 100$$

.Nilai ini dipakai sebagai nilai *Grayscale* pada lokasi pixel tertentu.

2.4 Deteksi Tepi (*Edge Detection*)

Deteksi tepi (*Edge Detection*) merupakan salah satu proses pengolahan yang sering dibutuhkan pada analisa citra digital. Proses ini bertujuan untuk meningkatkan penampakan garis tepi pada suatu citra digital. Informasi tentang tepi merupakan informasi yang penting dalam pengenalan citra sehingga deteksi tepi merupakan proses yang sering dilakukan dalam pengolahan citra digital.

Definisi tepi dibagi menjadi dua, yaitu :

1. Citra kontinu adalah perubahan intensitas yang tajam antara titik yang bertetanggaan.
2. Citra diskrit adalah lokasi titik (*pixel*) dimana terjadi perubahan gray level secara mendadak, misalnya suatu titik hitam dikatakan tepi bila atau paling tidak mempunyai tetangga putih dan sebaliknya.

Edge Detection adalah penelusuran gambar, dalam hal ini adalah *pixel* secara vertikal dan horisontal, sambil melihat terjadinya perubahan warna yang mendadak yang melebihi harga sensitifitas antara dua titik *pixel* yang berdekatan.Tujuan *Edge Detection* adalah untuk mengidentifikasi suatu daerah pada citra digital dimana terdapat perubahan yang besar dalam intensitas. Dengan deteksi tepi pada suatu sistem pemrosesan citra (berbasis komputer) akan menemukan tanda-tanda permukaan atau garis bentuk yang timbul dari suatu objek.

2.4.1. Canny

Edge detection canny merupakan algoritma yang digunakan untuk menganalisis suatu citra tepi. Pengembang algoritma *Edge detection canny* adalah John F. Canny pada tahun 1986 dan menggunakan algoritma multi-tahap untuk mendeteksi berbagai tepi pada gambar.

Algoritma Canny berjalan dalam 5 langkah yang terpisah yaitu :

Tabel 2-1 Step Canny

a)	Smoothing	Menggabungkan gambar untuk menghilangkan *noise*.
b)	Finding gradien	Tepian harus ditandai pada gambar memiliki gradien yang besar.
c)	Non-maksimum-suppresion	Hanya maxima lokal yang harus ditandai egde
d)	Double thresholding	Tepian yang berpotensi ditentukan oleh thresholding.
e)	Edge tracking by	Tepian final ditentukan dengan menekan semua sisis

	hysteresis	yang tidak terhubung dengan tepian yang sangat kuat

Tidak dapat dipungkiri bahwa semua gambar yang diambil akan berisi sejumlah *noise*. Untuk mencegah *noise* salah dideteksi sebagai tepian, maka *noise* harus dikurangi. Oleh karena itu pada langkah pertama gambar harus diperhalus dengan menggunakan Gaussian filter. Inti dari *Gaussian filter* adalah standar deviasi dengan σ = 1,4 ditentukan pada persamaan (2.8) di bawah ini.

$$\frac{1}{115}\begin{bmatrix}2 & 4 & 5 & 4 & 2\\ 4 & 9 & 12 & 9 & 4\\ 5 & 12 & 15 & 12 & 5\\ 4 & 9 & 12 & 9 & 4\\ 2 & 4 & 5 & 4 & 2\end{bmatrix} \text{x } \textit{Grayscale}\text{..... (2.8)}$$

Pada persamaan diatas merupakan matriks *Gaussian filter* yang digunakan untuk menghilangkan untuk menghilangkan *noise*.

Algoritma Canny pada dasarnya menemukan titik tepi pada gambar *grayscale* dengan perubahan nilai intensitas yang paling besar, daerah ini ditemukan dengan menentukan gradien gambar. Gradien pada setiap piksel gambar yang telah diperhalus ditentukan dengan menerapkan operator Sobel. Langkah Kedua adalah memperkirakan gradien pada arah x dan y. Hal tersebut ditunjukkan dalam Persamaan (2.9).

$$M_x = \begin{bmatrix}-1 & 0 & 1\\ -2 & 0 & 2\\ -1 & 0 & 1\end{bmatrix} \; M_y = \begin{bmatrix}1 & 2 & 1\\ 0 & 0 & 0\\ -1 & -2 & -1\end{bmatrix} \text{..... (2.9)}$$

Magnitudo gradien (juga dikenal sebagai kekuatan tepi) dapat ditentukan sebagai jarak *Euclidean* yang diukur mengukur dengan menerapkan hukum Pythagoras seperti yang ditunjukkan dalam Persamaan (2.10). Yang terkadang disederhanakan dengan menerapkan ukuran jarak Manhattan seperti yang ditunjukkan dalam Persamaan (2.11) untuk mengurangi kompleksitas komputasi.

$$|G| = \sqrt{G_x^3} + \sqrt{G_y^2} \ldots. \quad (2.10)$$

$$|G| = |G_x| + |G_y| \ldots. \quad (2.11)$$

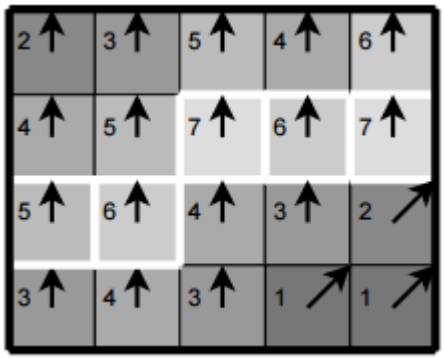
dimana: Gx dan Gy adalah gradien pada masing-masing arah x dan y. Hal ini tampak jelas bahwa gambar dengan gradien yang besar sering menunjukkan tepian yang cukup jelas. Namun, tepian biasanya luas dan dengan demikian tidak dapat menunjukkan persis di mana tepian yang sebenarnya. Untuk menentukan tepian yang sebenarnya ini, arah tepian harus ditentukan dan disimpan seperti ditunjukkan dalam Persamaan (2.12).

$$\theta = arctan\left(\frac{|G_x|}{|G_y|}\right) \ldots. \quad (2.12)$$

Pada langkah ketiga bertujuan untuk mengkorversikan tepian yang masih *blurred* pada gambar hasil magnitude gradien hingga menghasilkan tepian yang tajam. Pada dasarnya hal ini dilakukan dengan mempertahankan semua maxima lokal dalam gambar gradien dan menghapus segala sesuatu yang lain. Algoritma adalah untuk setiap piksel pada gambar gradien adalah sebagai berikut:

1. Putar arah gradien θ ke arah 45˚ terdekat, kemudian hubungkan dengan 8 titik tetangga yang terhubung dengannya.
2. Bandingkan nilai piksel tepian saat ini dengan nilai piksel tepian dalam arah positif dan negatif gradien. Jika arah gradien adalah utara (θ = 90◦), bandingkan dengan piksel ke utara dan selatan.
3. Jika nilai piksel tepian saat ini adalah yang terbesar, maka simpan nilai tepian tersebut, namun jika bukan, hapus nilai tersebut.

Sebuah contoh sederhana dari penghapusan non maksimum ditunjukkan pada Gambar 2.1. Hampir semua piksel yang memiliki arah gradien yang menunjukkan arah utara, oleh karena itu mereka dibandingkan dengan piksel atas dan bawah. Piksel yang berubah menjadi maksimal dalam perbandingan ini ditandai dengan warna putih pada perbatasan, sisanya dihapus.



Gambar 2.1 Illustrasi Penghapusan Non Maksimum

Piksel tepian yang tersisa setelah dilakukan penghapusan non maksimum ditandai dengan nilai piksel per piksel yang kuat. Kebanyakan dari titik ini adalah tapian yang nyata pada gambar, akan tetapi beberapa kemungkinan disebabkan oleh *noise* atau variasi wana karena permukaan yang kasar. Cara paling sederhana untuk membedakannya adalah menggunakan nilai threshold (ambang batas) sehingga hanya tepian dengan nilai yang kuar yang akan dipertahankan. Disini pada algoritma Canny menggunakan sistem *thresholding* ganda yang dimana *thresholding* tersebut akan dibandingkan dengan *thresolding* yang memiliki nilai yang lebih besar, apabila nilai lebih besar maka nilai thresolding tersebut akan digunakan sebagai acuan menentukan tepian dan nilai thresolding yang kecil akan dihilangkan.

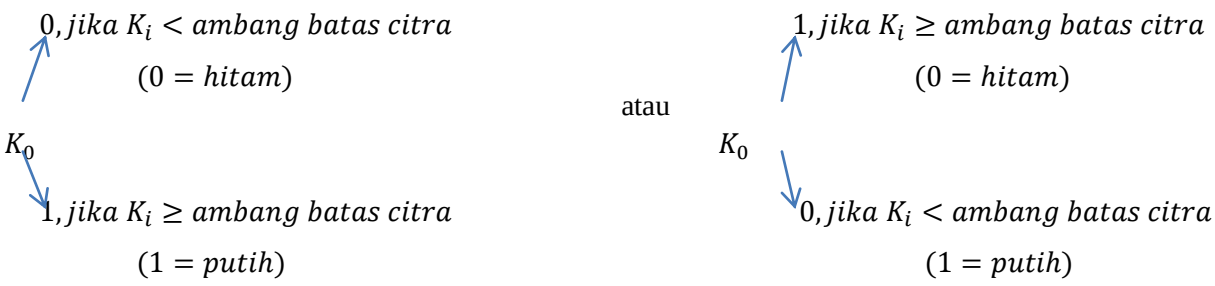
2.4.2. Normalisasi

Normalisasi merupakan proses merubah dimensi region tiap karakter. Pada proses ini menggunakan proses *Thresholding,* merupakan salah satu operasi yang termasuk ke dalam operasi titik dalam pengolahan citra digital. Operasi ini digunakan untuk mengubah citra dengan format skala keabuan yang dimiliki kemungkinan nilai lebih dari 2 ke citra biner yang memiliki 2 buah nilai (0 dan 1).

Thresholding terdiri dari dua jenis, yaitu:

a. Thresholding Tunggal

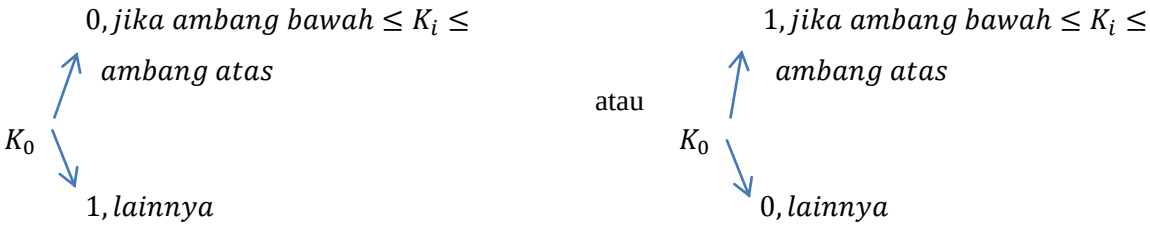
   Thresholding Tunggal merupakan proses *thresholding* yang hanya memiliki sebuah nilai ambang batas atau batasan suatu citra.



Gambar 2.2 Fungsi Thresholding Tunggal

b. Thresholding Ganda

Memiliki ambang batas atas dan batas bawah. Dilakukan untuk menampilkan titik-titik yang memiliki rentang nilai skala keabuan tertentu.



Gambar 2.3 Fungsi Thresholding Ganda

2.5 Proses Pencocokkan (*Matching Process*)

Matching Process digunakan untuk menghitung jarak antar tepi (*Edge*) .Dimana dalam tahapan ini menentukan tingkat kemiripan dari tepi citra uji dan tepi citra latih.Semakin kecil jaraknya maka citra tersebut semakin mirip dengan citra uji. Pada penelitian ini, proses pencocokkan menggunakan perhitungan tepi. Perhitungan ini diimplementasikan dengan persamaan euclidean sebagai berikut (2.13)

$$d = \sqrt{(j_1 - j_2)^2 + (i_1 - i_2)^2} \text{.....(2.13)}$$

Dengan :

d =Jarak dari dua citra *Grayscale* dengan perhitungan *Chamfer*.

j_1= data koordinat ke-j_1 pada tepi (*Edge*) citra uji.

j_2 = data koordinat ke-j_2 pada tepi (*Edge*) citra yang ada di latih.

i_1 = data koordinat ke-i_1 pada tepi (*Edge*) citra uji.

i_2 = data koordinat ke-i_2 pada tepi (*Edge*) citra yang ada di latih.

2.5.1 Chamfer Matching

|"*ChamferMatching* adalah metode yang dilakukan untuk mencari jarak terdekat dari titik tepi dari citra"[6]. Perhitungan ini diimplementasikan dengan persamaan sebagai berikut (2.14) :

$$n \sum_{i=1} \min(\|p_i,q_j\|,\|p_i,q_j\|,\cdots\|p_i,q_m\|) \qquad (2.14)$$

Dengan :

p_1= titik yang ada pada citra uji.

q_j = titik yang ada pada citra latih.

2.6 Unified Modelling Language (UML)

Unified Modelling Language (UML) adalah sebuah bahasa pemodelan standar yang memiliki sintaks dan semantic.Pemodelan ini sangat cocok digunakan untuk merancang dan memodelkan sistem berorientasi objek [6].Diagram pada UML dibagi menjadi dua bagian yaitu structural diagram dan behavior diagram.Structural diagram digunakan untuk mendeskripsikan relasi antar kelas.Tools yang digunakan pada bagian ini yaitu *Class* diagram. Sedangkan behavior diagram digunakan untuk mendeskripsikan interaksi antara aktor dan sebuah *Use Case* (bagaimana seorang aktor menggunakan sistem). Tools yang digunakan pada bagian ini yaitu *Use Case* diagram, *Sequence* diagram, dan *Activity* diagram.

2.6.1 Diagram Use Case

Diagram *Use Case* digunakan untuk mendeskripsikan kejadian-kejadian apa saja yang dapat dilakukan oleh user/aktor dan fungsionalitas-fungsionalitas apa saja yang diharapkan dari sistem yang akan dibangun, tanpa mendeskripsikan bagaimana sistem menyelesaikannya [7]. Sebuah *Use Case* menggambarkan suatu urutan interaksi antara satu atau lebih aktor dan sistem. Dalam fase requirements, model *Use Case* mengambarkan sistem sebagai sebuah kotak hitam dan interaksi

antara aktor dan sistem dalam suatu bentuk naratif, yang terdiri dari input user dan res*purchase order*n-res*purchase order*n sistem.

Setiap *Use Case* menggambarkan perilaku sejumlah aspek sistem, tanpa mengurangi struktur internalnya. Selama pembuatan model *Use Case* secara pararel juga harus ditetapkan obyek-obyek yang terlibat dalam setiap *Use Case*.

2.6.2 Diagram Sequence

Diagram Sequence menggambarkan interaksi antar objek di dalam dan di sekitar sistem yang menekankan pada pengiriman pesan dalam suatu waktu tertentu [7].*Sequence* diagram biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai res*purchase order*ns dari sebuah *event* untuk menghasilkan output tertentu. Diawali dari apa yang men-trigger aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan output apa yang dihasilkan.

2.6.3 Diagram Activity

Diagram Activity menggambarkan berbagai alur aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, decision yang mungkin terjadi, dan bagaimana mereka berakhir [7]. *Activity* diagram merupakan *state* diagram khusus, di mana sebagian besar *state* adalah *action* dan sebagian besar transisi di-*trigger* oleh selesainya *state* sebelumnya (*internal Processing*). Oleh karena itu *Activity* diagram tidak menggambarkan behaviour internal sebuah sistem (dan interaksi antar subsistem) secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum.

2.6.4 Class Diagram

Diagram Class digunakan untuk menggambarkan keadaan suatu sistem dengan menjelaskan keterhubungan antara suatu *Class* dengan *Class* yang lain yang terdapat pada system[7]. Sebuah *Class* terdiri dari nama, atribut dan *method*. Atribut dan *method* dari sebuah kelas mempunyai *visibility*, Ada tiga jenis *visibility* yang digunakan yaitu *private, public, dan protected*. Setiap *Class* pada *Class* diagram mempunyai hubungan dengan *Class* lainnya, ada beberapa jenis

hubungan *Class*, yaitu *dependency*, asosiasi, agregasi, kom*purchase order*sisi dan generalisasi.

1. Dependency, Dependency merupakan hubungan terlemah antar Class. Dependency bermakna satu Class menggunakan atau memiliki pengetahuan terhadap Class lain, namun hubungannya hanya sementara dan tidak ada batas waktu yang jelas.
2. Asosiasi, Asosiasi mempunyai hubungan yang lebih kuat dari hubungan dependency, dimana suatu Class tetap berhubungan dengan Class lain seterusnya. Hubungan asosiasi dibagi menjadi dua jenis yaitu directional dan bidirectional.
3. Agregasi, Agregasi merupakan bentuk hubungan yang mengimplikasikan kepemilikan suatu Class. Agregasi juga merupakan bentuk yang lebih kuat dari asosiasi. Hubungan agregasi dinyatakan dengan simbol diamond pada pemilik kelas dan garis utuh berpanah kekelas yang dimiliki.
4. Kompurchase ordersisi, Kompurchase ordersisi merupakan bentuk hubungan antar Class yang paling kuat. Kompurchase ordersisi digunakan untuk mengambil seluruh bagian dari Class yang berhubungan. Aturan dari kompurchase ordersisi yaitu hanya boleh ada satu kompurchase ordersisi dalam satu waktu.
5. Generalisasi, Generalisasi merupakan bentuk hubungan antar Class, dari Class yang umum dengan Class yang lebih khusus. Contoh Class hewan memiliki hubungan generalisasi dengan Class kucing, karena kucing merupakan Class khusus dari Class hewan yang lebih umum.