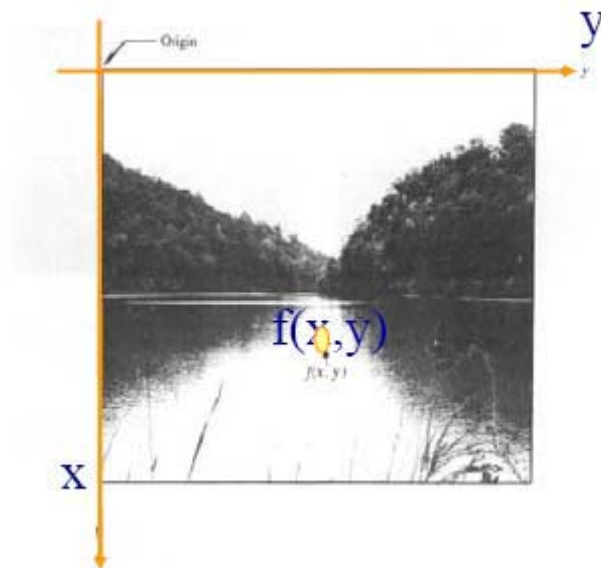


BAB 2

LANDASAN TEORI

2.1 Citra Digital

Citra digital dapat didefinisikan sebagai fungsi dua variabel, $f(x,y)$, dimana x dan y adalah koordinat spasial dan nilai $f(x,y)$ yang merupakan intensitas citra pada koordinat tersebut. Teknologi dasar untuk menciptakan dan menampilkan warna pada citra digital berdasarkan pada penelitian bahwa sebuah warna merupakan kombinasi dari tiga warna dasar, yaitu merah, hijau, dan biru (*Red, Green, Blue - RGB*).[3] Sistem kordinat pada sebuah citra digital dapat dilihat pada Gambar 2.1.

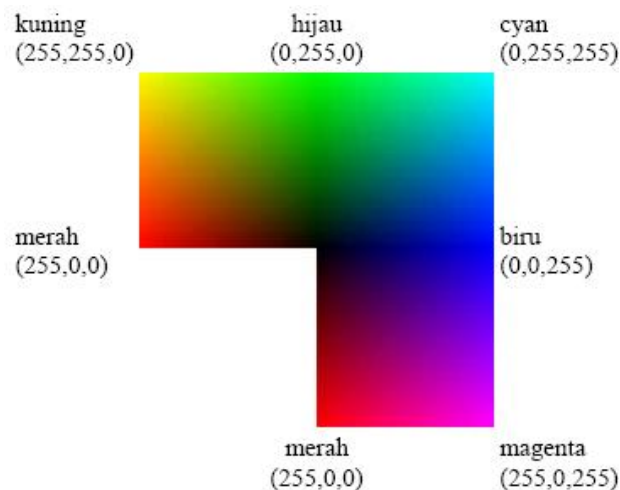


Gambar 2.1 Sistem Kordinat Citra Digital

RGB adalah suatu model warna yang terdiri dari merah, hijau, dan biru, digabungkan dalam membentuk suatu susunan warna yang luas. Setiap warna dasar, misalnya merah, dapat diberi rentang nilai. Untuk monitor komputer, nilai rentangnya paling kecil = 0 dan paling besar = 255. Pilihan skala 256 ini didasarkan pada cara mengungkap 8 digit bilangan biner yang digunakan oleh

mesin komputer. Dengan cara ini, akan diperoleh warna campuran sebanyak $256 \times 256 \times 256 = 16777216$ jenis warna. Sebuah jenis warna, dapat dibayangkan sebagai sebuah vektor di ruang dimensi 3 yang biasanya dipakai dalam matematika, koordinatnya dinyatakan dalam bentuk tiga bilangan, yaitu komponen- x , komponen- y dan komponen- z . Misalkan sebuah vektor dituliskan sebagai $r = (x, y, z)$. Untuk warna, komponen-komponen tersebut digantikan oleh komponen R(ed), G(reen), B(lue). Jadi, sebuah jenis warna dapat dituliskan sebagai berikut: warna = RGB(30, 75, 255). Putih = RGB (255,255,255), sedangkan untuk hitam= RGB(0,0,0).[6]

Bentuk Representasi warna dari sebuah citra digital dapat dilihat pada Gambar 2.2.



Gambar 2.2 Representasi Warna RGB Pada Citra Digital

Misalnya terdapat gambar berukuran 100 *pixel* x 100 *pixel* dengan *color encoding* 24 bit dengan R = 8 bit, G = 8 bit, B = 8 bit, maka *color encoding* akan mampu mewakili 0 ... 16.777.215 (mewakili 16 juta warna), dan ruang *disk* yang dibutuhkan = $100 \times 100 \times 3$ bit (karena RGB) = 30.000 bit = 30 KB atau $100 \times 100 \times 24$ bit = 240.000 bit.[6]

2.2 Format Citra Digital

Citra Digital memiliki beberapa format yang memiliki karakteristik tersendiri. Format pada citra digital ini umumnya berdasarkan tipe dan cara kompresi yang digunakan pada citra digital tersebut.

Ada dua format citra digital yang sering dijumpai, antara lain [5]:

1. *Bitmap* (BMP)

Merupakan format gambar yang paling umum dan merupakan format *standard windows*. Ukuran *filenya* sangat besar karena bisa mencapai ukuran *megabyte*. *File* ini merupakan format yang belum terkompresi dan menggunakan sistem warna RGB (*Red, Green, Blue*) di mana masing-masing warna *pixel*-nya terdiri dari 3 komponen R, G, dan B yang dicampur menjadi satu. *File* BMP dapat dibuka dengan berbagai macam *software* pembuka gambar seperti *ACDSee, Paint, Irvan View* dan lain-lain. *File* BMP tidak bisa (sangat jarang) digunakan di *web (internet)* karena ukurannya yang besar. Detail gambar BMP dapat dilihat pada Tabel 2.1 dan Tabel 2.2.

Tabel 2.1 *Bitmap Info Header*

Nama Field	<i>Size in Bytes</i>	Keterangan
bfType	2	Mengandung karakter “BM” yang mengidentifikasikan tipe file
bfSize	4	Memori file
bfReserved1	2	Tidak dipergunakan
bfReserved1	2	Tidak dipergunakan
bfOffBits	4	Offset untuk memulai data <i>pixel</i>

Tabel 2.2 *Bitmap Core Header*

Field Name	Size in Bytes	Keterangan
bcSize	4	Memori <i>Header</i>
bcWidth	2	Lebar Gambar
bcHeight	2	Tinggi Gambar
bcPlanes	2	Harus 1
bcBitCount	2	<i>Bits per pixels</i> – 1,4,8 atau 24

2. *Joint Photographic Expert Group (JPEG/JPG)*

Format JPEG merupakan format yang paling terkenal sampai sekarang ini. Hal ini karena sifatnya yang berukuran kecil (hanya puluhan/ratusan KB saja), dan bersifat *portable*. Format *file* ini sering digunakan pada bidang fotografi untuk menyimpan *file* foto hasil perekaman *analog to digital converter* (ADC). Karena ukurannya kecil maka *file* ini banyak digunakan di web (*internet*).

2.3 *Citra Grayscale*

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*. [4]

Citra *grayscale* disimpan dalam format 8 bit untuk setiap sample pixel, yang memungkinkan sebanyak 256 intensitas. Format ini sangat membantu dalam pemrograman karena manipulasi bit yang tidak terlalu banyak. Untuk mengubah citra berwarna yang mempunyai nilai matrik masing-masing R, G dan B menjadi citra *grayscale* dengan nilai X, maka konversi dapat dilakukan dengan mengambil rata-rata dari nilai R, G dan B. [5]

2.4 *Citra Threshold*

Citra *threshold* dilakukan dengan mempertegas citra dengan cara mengubah citra hasil yang memiliki derajat keabuan 255 (8 bit), menjadi hanya dua buah yaitu hitam dan putih. Hal yang perlu diperhatikan pada proses *threshold* adalah memilih sebuah nilai *threshold* (T) dimana piksel yang bernilai dibawah nilai

threshold akan diset menjadi hitam dan piksel yang bernilai diatas nilai *threshold* akan diset menjadi putih. Umumnya nilai T dihitung dengan menggunakan persamaan:

$$T = \frac{f_{maks} + f_{Min}}{2} \dots\dots\dots (2.1)$$

Dimana *f_{maks}* adalah nilai intensitas maksimum pada citra dan *f_{min}* adalah nilai intensitas minimum pada citra.[6]

Jika *f(x,y)* adalah nilai intensitas *pixel* pada posisi (*x,y*) maka *pixel* tersebut diganti putih atau hitam tergantung kondisi berikut.[1]

f(x,y) = 255, jika *f(x,y)* ≥ T

f(x,y) = 0, jika *f(x,y)* < T

Sebagai contoh misalnya diketahui citra *grayscale* 4x4 *pixel* dengan kedalaman 8 bit seperti Gambar 2.3.

200	230	150	75
240	50	170	92
210	100	120	80
100	90	200	230

Gambar 2.3 Citra Grayscale 4x4 Pixel

Dengan motede ini, nilai *threshold* T adalah:

$$T = \frac{f_{maks} + f_{Min}}{2} = T = \frac{240 + 50}{2} = 145$$

Bila nilai T = 145 diterapkan untuk citra pada Gambar 2.8 di atas maka diperoleh citra seperti pada Gambar 2.4.

255	255	255	0
255	0	255	0
255	0	0	0
0	0	255	255

Gambar 2.4 Citra Hasil Threshold

2.5 Citra Biner

Citra biner adalah citra dimana piksel-pikselnya hanya memiliki dua buah nilai intensitas yaitu bernilai 0 dan 1 dimana 0 menyatakan warna latar belakang (*background*) dan 1 menyatakan warna tinta/objek (*foreground*) atau dalam bentuk angka 0 untuk warna hitam dan angka 255 untuk warna putih. Citra biner diperoleh dari nilai citra *threshold* sebelumnya.[4]

Gradasi citra biner dapat dilihat pada Gambar 2.5.



Gambar 2.5 Gradasi Citra Biner

Contoh sebuah citra biner seperti pada Gambar 2.6



Gambar 2.6 Gradasi Citra Biner

2.6 Deteksi Tepi

Teknik *image enhancement* digunakan untuk meningkatkan kualitas suatu citra digital, baik dalam tujuan untuk menonjolkan suatu ciri tertentu dalam citra tersebut, maupun untuk memperbaiki aspek tampilan. Proses ini biasanya didasarkan pada prosedur yang bersifat eksperimental, subjektif, dan amat bergantung pada tujuan yang hendak dicapai.[5]

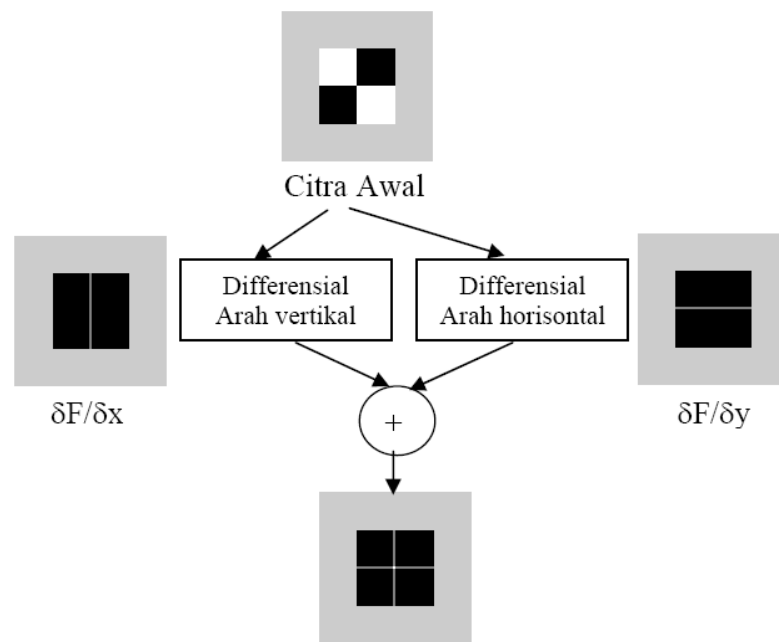
Proses peningkatan mutu citra bertujuan untuk memperoleh citra yang dapat memberikan informasi sesuai dengan tujuan/kepentingan pengolahan citra. Proses peningkatan mutu citra ini termasuk memperbaiki citra yang ketika proses

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.

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*.

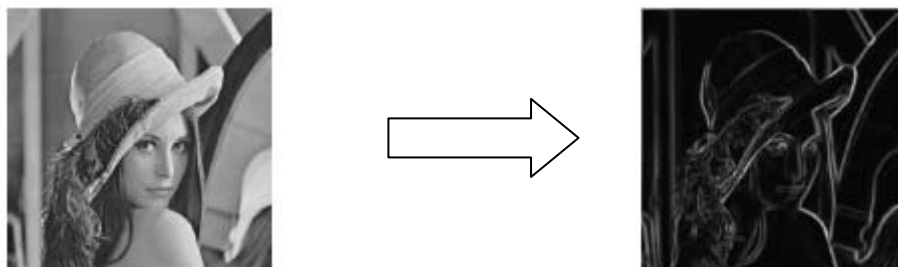
Tepi (*edge*) adalah perubahan nilai intensitas derajat keabuan yang cepat/tiba-tiba (besar) dalam jarak yang singkat. Sedangkan deteksi tepi (*edge detection*) pada suatu citra adalah suatu proses yang menghasilkan tepi-tepi dari obyek-obyek citra, tujuannya adalah untuk menandai bagian yang menjadi detail citra, untuk memperbaiki detail dari citra yang kabur, yang terjadi karena *error* atau adanya efek dari proses akuisisi citra. Suatu titik (x,y) dikatakan sebagai tepi (*edge*) dari suatu citra bila titik tersebut mempunyai perbedaan yang tinggi dengan tetangganya.[4]

Pada Gambar 2.7 di bawah ini dapat dilihat proses yang dilakukan untuk memperoleh tepi gambar dari suatu citra yang ada.



Gambar 2.7. Proses Deteksi Tepi Citra

Pada Gambar 2.8 terlihat bahwa hasil deteksi tepi berupa tepi-tepi dari suatu gambar. Bila diperhatikan bahwa tepi suatu gambar terletak pada titik-titik yang memiliki perbedaan tinggi.



Gambar 2.8. Hasil Deteksi Tepi

Proses deteksi tepi (*edge detection*) sendiri masing dapat dikelompokkan berdasarkan operator atau metode yang digunakan dalam proses pendeteksian tepi suatu citra untuk memperoleh citra hasil.

2.6.1 Metode Robert

Metode robert adalah nama lain dari teknik differensial pada arah horizontal dan differensial pada arah vertikal, dengan ditambahkan proses konversi biner setelah dilakukan differensial. Teknik konversi biner yang disarankan adalah konversi biner dengan meratakan distribusi warna hitam dan putih.. Metode robert ini juga disamakan dengan teknik DPCM (*Differential Pulse Code Modulation*).[2]

Operator robert menggunakan operator *gradient* berukuran 2 x 2 :

1	1
-1	-1

Gradient magnitude dari operator robert adalah sebagai berikut :

$$G[f(i,j)] = [f(i,j) - f(i+1,j+1)] + [f(i+1,j) - f(i,j+1)] \dots\dots\dots (2.2)$$

Karena operator robert hanya menggunakan *convolution mask* berukuran 2 x 2, maka operator robert sangat sensitive terhadap *noise*.

2.6.2 Metode Prewitt

Metode prewitt merupakan pengembangan metode robert dengan menggunakan filter HPF yang diberi satu angka nol penyangga. Metode ini mengambil prinsip dari fungsi laplacian yang dikenal sebagai fungsi untuk membangkitkan HPF.[2]

Operator prewitt menggunakan empat buah kernel operator gradien :

1	1	1
0	0	0
-1	-1	-1

(a)

0	1	1
-1	0	1
-1	-1	0

(b)

-1	0	1
-1	0	1
-1	0	1

(c)

-1	-1	0
-1	0	1
0	1	1

(d)

2.6.3 Metode Sobel

Metode sobel merupakan pengembangan metode robert dengan menggunakan filter HPF yang diberi satu angka nol penyangga. Metode ini mengambil prinsip dari fungsi laplacian dan gaussian yang dikenal sebagai fungsi untuk membangkitkan HPF. Kelebihan dari metode sobel ini adalah kemampuan untuk mengurangi *noise* sebelum melakukan perhitungan deteksi tepi.[2]

Operator sobel menggunakan kernel operator *gradient* 3 x 3 :

1	2	1
0	0	0
-1	-2	-1

(a)

1	0	-1
2	0	-2
1	0	-1

(b)

Operator sobel melakukan deteksi tepi dengan memperhatikan tepi vertikal dan horizontal. *Gradient magnitude* dari operator sobel adalah sebagai berikut :

$$G_x = [f(i-1, j-1) + 2f(i-1, j) + f(i-1, j+1)] - [f(i+1, j-1) + 2f(i+1, j) + f(i+1, j+1)]$$

$$G_y = [f(i-1, j-1) + 2f(i, j-1) + f(i+1, j-1)] - [f(i-1, j+1) + 2f(i, j+1) + f(i+1, j+1)]$$

$$G[f(x, y)] = \sqrt{G_x^2 + G_y^2} \dots\dots\dots (2.3)$$

2.7 Operasi Berbasis Bingkai

Operasi ini bisa melibatkan dua buah citra atau lebih dan menghasilkan sebuah citra keluaran sebagai hasil operasinya [6].

2.7.1 Operasi Penjumlahan Citra

Penggabungan citra dilakukan dengan cara menjumlahkan sebuah citra dengan citra yang lain. Penggabungan ini bisa dilakukan juga bila jumlah citra yang digabungkan lebih dari dua buah. Secara sistematis rumus penggabungan citra adalah :

$$f_o(x,y) = w_1.A_1(x,y) + w_2.A_2(x,y) + w_3.A_3(x,y) + \dots w_n.A_n(x,y) \dots\dots\dots(2.4)$$

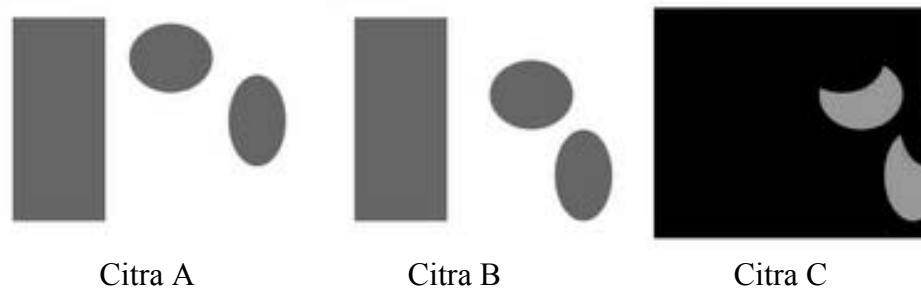
dimana $w_1, w_2, w_3, \dots, w_n$ masing-masing adalah bobot untuk citra $A_1, A_2, A_3, \dots, A_n$. Dengan catatan bahwa : $w_1 + w_2 + w_3 + \dots + w_n = 1$.

2.7.2 Operasi Pengurangan Citra (Deteksi Gerakan)

Deteksi gerak sebuah benda dapat dilakukan dengan operasi pengurangan. Untuk mencari benda yang bergerak pada sebuah citra, yang harus dilakukan adalah mengambil citra pertama dan citra kedua, kemudian salah satu citra dikurangkan terhadap citra lain. Secara sistematis dapat ditulis sebagai berikut.

$$A(x,y) - B(x,y) = C(x,y) \dots\dots\dots (2.5)$$

Pada operasi ini, bagian yang tidak bergerak dalam citra akan menghasilkan nilai 0, sedangkan untuk bagian yang bergerak akan memberikan nilai yang tidak nol. Dengan operasi ini dapat diketahui apakah pada citra terdapat objek yang bergerak. Gambar 2.9 adalah contoh untuk mendeteksi gerak objek dalam sebuah citra.



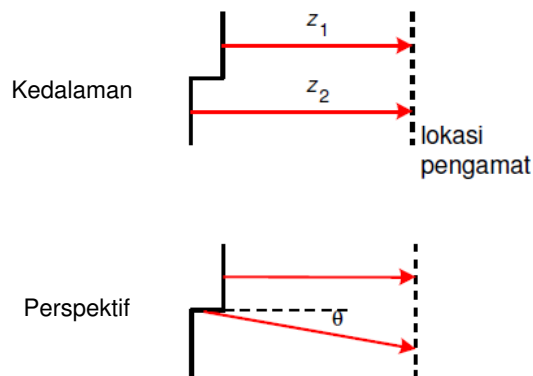
Gambar 2.9 Operasi Pengurangan untuk Mendeteksi Objek yang Bergerak

Berdasarkan citra A, B, dan C terlihat bahwa objek oval atas bergeser ke bawah, sedangkan objek oval yang bawah bergeser ke kanan bawah.

2.8 Citra 3 Dimensi

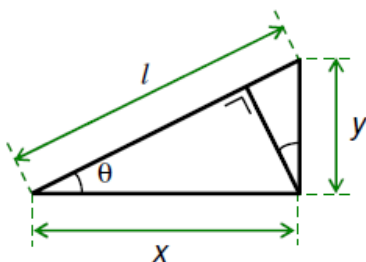
Citra 3 Dimensi adalah proses *morphology image* yang merubah bentuk asli *binary image* menjadi *image* yang menampilkan batas-batas objek/*foreground* hanya setebal satu *pixel*. Algoritma ini secara iteratif menghapus *pixel-pixel* pada *binary image*, dimana transisi dari 0 ke 1 (dari 1 ke 0 pada konvensi lain) terjadi sampai dengan terpenuhi suatu keadaan dimana satu himpunan dari lebar per unit (satu *pixel*) terhubung menjadi suatu garis.[8]

Pengertian citra 3 dimensi adalah citra yang seolah-olah obyeknya hadir secara utuh memperlihatkan kedalaman dan pandangan perspektif. Kedalaman ditentukan oleh jarak perambatan gelombang, sedangkan perspektif ditentukan oleh arah perambatan gelombang seperti diperlihatkan pada Gambar 2.10.



Gambar 2.10 Kedalaman dan Perspektif Citra 3 Dimensi.

Fungsi gelombang yang dipakai pada citra tiga dimensi adalah :

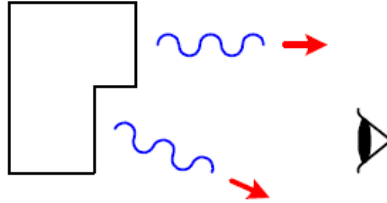


$$\bar{U} = A \exp (j\varphi) = A \exp [j(2\pi/\lambda)l]$$

$$l = x \cos \theta + y \sin \theta$$

$$\bar{U} = A \exp [j(2\pi/\lambda).(x \cos \theta + y \sin \theta)]$$

Fungsi gelombang diatas digunakan untuk membawa informasi mengenai kedalaman objek dan perspektif pandangan kedalam mata manusia seperti pada Gambar 2.11.



Gambar 2.11 Gelombang Informasi Kedalaman Objek dan Perspektif.

2.9 *Virtual Reality* (VR)

Virtual Reality (VR) atau Realitas Maya adalah teknologi yang dibuat sehingga pengguna dapat berinteraksi dengan suatu lingkungan yang disimulasikan oleh komputer (*computer-simulated environment*), suatu lingkungan sebenarnya yang ditiru atau benar-benar suatu lingkungan yang hanya ada dalam imajinasi. Secara sederhana, *virtual reality* adalah pemunculan gambar-gambar 3 dimensi yang di bangkitkan komputer, yang terlihat nyata dengan bantuan sejumlah peralatan tertentu. Ciri terpentingnya adalah dengan menggunakan perangkat yang dirancang untuk tujuan tertentu, teknologi ini mampu menjadikan orang yang merasakan dunia maya tersebut terkecoh dan yakin bahwa yang dialaminya adalah nyata.[16]

Istilah realitas maya tidak pasti asalnya. Pengembang realitas maya, Jaron Lanier mengakui bahwa ia menggunakan istilah itu pertama kali dan ada istilah yang terkait digunakan oleh Myron Krueger adalah “kenyataan tiruan” telah digunakan sejak 1970.

Virtual reality sering digunakan untuk menggambarkan berbagai aplikasi, umumnya terkait dengan mendalam, sangat visual, 3D lingkungan. CAD pengembangan perangkat lunak, akselerasi perangkat keras grafik, kepala-*mount*

display, sarung tangan *database* dan miniaturisasi telah membantu mempopulerkan gagasan. Dalam buku *The Metaphysics of Virtual Reality*, Michael R. Heim mengidentifikasi tujuh konsep yang berbeda *virtual reality* yaitu [16] :

1. Simulasi
2. Interaksi
3. Kepalsuan
4. Imersi
5. *Tele Presence*
6. Seluruh Tubuh Imersi
7. Jaringan Komunikasi

Untuk memasuki *virtual reality*, pengguna mengenakan sarung tangan khusus, *earphone*, dan kacamata khusus yang terhubung dengan komputer dan sistem yang di dalamnya. Melalui cara ini, setidaknya tiga indera tubuh kita terkontrol oleh komputer. Untuk hasil yang lebih baik, biasanya piranti *virtual reality* ini juga memonitor apa yang dilakukan *user*. Misalnya kacamata yang mengontrol pergerakan bola mata pengguna dan meresponnya dengan mengirim masukkan video yang baru. *Virtual reality* kadang digunakan untuk menyebut dunia *virtual* yang disajikan ke dalam komputer, seperti pada berbagai macam game permainan komputer yang kini marak perkembangannya, meskipun hanya berbasis representasi teks, suara dan grafis.

Sekarang, istilah *virtual reality* mulai tergantikan oleh istilah *virtual environment* (VE) oleh para ahli komputer. Konsepnya tetap sama, yaitu mensimulasikan lingkungan 3 dimensi yang bisa dijelajahi oleh pengguna seolah-olah benar-benar bisa dirasakan lewat indera.

2(dua) syarat yang harus ada dalam VR/ VE adalah:

1. Gambar/ grafis/ penglihatan 3-D yang nyata menurut perspektif penglihatan pengguna.

2. Kemampuan untuk mendeteksi gerakan-gerakan pengguna, seperti gerakan kepala dan arah bola mata, untuk menyesuaikan grafis yang dihasilkan supaya menyesuaikan perubahan dunia 3 dimensi-nya.

Saat berada dalam VR, pengguna akan merasa melebur seolah menyatu dengan dunianya, dan bisa berinteraksi dengan objek-objek yang ada di sana. Hal ini disebut dengan *telepresence*. [16]

Menurut *Jonathan Stauer*, ada dua komponen dalam perasaan melebur ini, yang disebut:

1. *Depth of information*, merupakan banyak dan kualitas data yang ditransfer demi menciptakan lingkungan VR, seperti resolusi, ketajaman gambar, dll.
2. *Breadth of information*, yaitu seberapa besar indera pengguna dimanipulasi, yang biasanya terbatas pada penglihatan dan pendengaran. Namun saat ini sedang dikembangkan VR yang bisa memanipulasi indera sentuhan dan pembau.

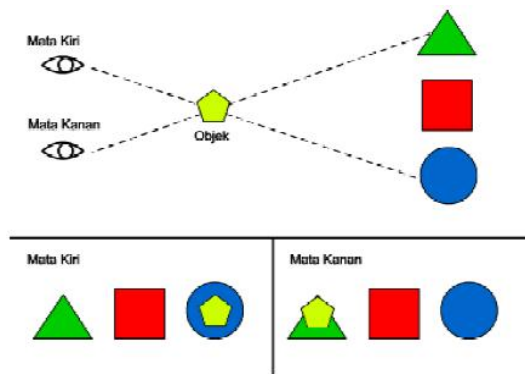
Salah satu contoh aplikasi *virtual reality* yang digunakan pada saat ini yaitu dalam bidang militer. *Virtual reality* dipakai untuk melakukan simulasi latihan perang, simulasi latihan terjun payung, dan sebagainya. Dimana dengan pemakaian teknologi ini bisa lebih menghemat biaya dan waktu dibandingkan dengan cara konvensional.

2.10 Citra Stereoscopy

Apabila berhubungan dengan dunia pemodelan, selalu yang dibahas adalah bagaimana menampilkan keadaan objek yang sebenarnya pada layar tampilan. Dengan demikian akan selalu berhubungan dengan objek 3 dimensi. Dalam aplikasi grafika yaitu *virtual reality* digunakan untuk menampilkan pemandangan yang senyata mungkin. Kesan realistis yang paling bagus adalah menampilkan tingkat kedalaman pada objek yang dilihat. Oleh karena itu bermunculan metode-metode untuk menampilkan tingkat kedalaman objek tiga dimensi, dan salah satunya adalah metode *stereoscopy*. *Stereoscopy* adalah teknik penyampaian informasi visual 3 dimensi melalui media citra 2 dimensi, sehingga penikmat akan merasakan adanya ilusi kedalaman pada citra tersebut. [8]

Stereoscopy merupakan teknik untuk membuat rekaman (foto) yang memperlihatkan kedalaman (pembedaan obyek yang jauh dan dekat). Film tiga dimensi sebenarnya tidak menghasilkan citra 3 dimensi tetapi citra *stereoscopy*. [10]

Citra *stereoscopy* mempunyai ilusi kedalaman. Efek *stereoscopy* dapat tercipta apabila mata melihat suatu *scene* pada citra seperti mata melihat *scene* tersebut pada dunia nyata. Pada dunia nyata kedua mata masing-masing melihat gambar yang sekilas sama namun sebenarnya berbeda dikarenakan perbedaan orientasi objek atau perbedaan sudut pandang tiap mata (*parallax*). [17] Perbedaan *parallax* diperlihatkan pada Gambar 2.12.



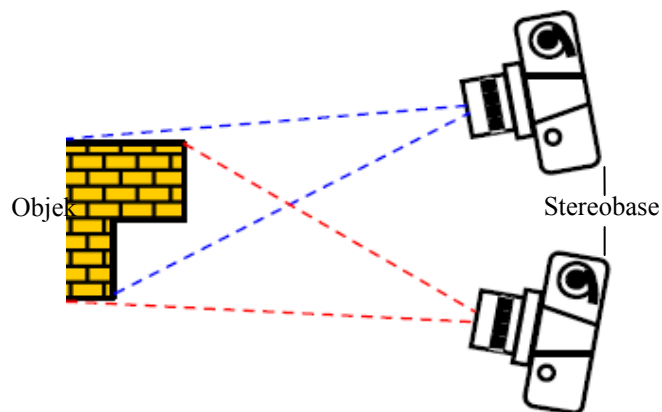
Gambar 2.12 Perbedaan Orientasi Objek Sudut Pandang Tiap Mata

Objek akan mempunyai posisi yang berbeda terhadap latar dikarenakan perbedaan sudut pandang dan jarak antara objek dan latar. Untuk dapat melihat efek *stereoscopy* dibutuhkan dua buah gambar, satu gambar mata kiri dan satu gambar mata kanan, yang identik, baik dalam segi *scene / point of interest*, resolusi, maupun kualitas, tetapi berbeda sudut pengambilan gambarnya (sudut pengambilan mewakili mata kiri dan mata kanan). Gambar *stereoscopy* biasa diambil dengan menggunakan dua kamera untuk kesamaan *scene*. Membuat foto 3D juga mudah, yaitu dengan meniru cara pandang dari mata manusia. Dengan menggunakan sepasang kamera yang identik, akan mendapatkan 2 gambar yang kelihatan sama, tetapi memiliki informasi sudut pandang yang berbeda. [17]



Gambar 2.13 Gambar Mata Kiri Kanan yang Berbeda Sudut Pandang

Menurut Hart dua buah citra *stereoscopy* yang ditumpuk, akan timbul yang disebut gambar disparitas (*disparity image*) yaitu gambar rangkap dua yang terpisah karena adanya beda *parallax*. [14]. *Disparity image* ini tercipta karena adanya *stereobase*, yaitu jarak antara kedua kamera dengan sumbu lensa seperti yang ditunjukkan pada Gambar 2.14.



Gambar 2.14 Stereobase untuk 2 Buah Kamera

Adapun objek yang digunakan untuk menentukan besar *stereobase* dalam citra *stereoscopy* adalah :

1. Citra Normal Stereo
2. Citra Hypostereo

3. Citra *Hyperstereo*

2.10.1 Citra Normal Stereo

Citra normal *stereo* adalah citra dengan jarak pandang normal. Artinya tidak perlu menggunakan alat bantu apapun untuk melihat suatu objek dengan jelas. pada umumnya citra normal *stereo* dipersepsikan untuk benda yang berjarak 2-10 meter dari kamera.[18]

Jarak antara kedua kamera, dapat dianalogikan pada jarak kedua mata yang besarnya 65 mm. Dengan acuan ini, *stereobase* dapat diperhitungkan dengan menggunakan rumus dasar 1/30. Dimana persamaan ini menjelaskan bahwa untuk setiap foto *stereoscopy*, besarnya jarak terdekat benda dengan kamera dibagi 30. Aturan 1/30 diperoleh berdasarkan persamaan dibawah ini sekaligus sebagai pedoman untuk menentukan *stereo-base* 2 buah kamera :[19]

$$B = N \times I/V \dots\dots\dots (2.6)$$

Dimana :

I = Jarak *interocular* (jarak antara kedua mata, biasanya sekitar 65mm)

V = Jarak sebenarnya ke obyek (Suatu citra dengan jarak normal dan yang memberikan hasil terbaik adalah pada jarak 2m)

N = Jarak ke jendela *stereo*, yaitu jarak dengan objek terdekat.

B = *Stereo-base* atau jarak antara kedua sumbu kamera paralel.

Sehingga persamaan menjadi :

$$\begin{aligned} B &= N \times 65\text{mm} / 2000\text{mm} \\ &= N \times 1 / 30,7692 \\ &= N \times 1 / 30 \dots\dots\dots (2.7) \end{aligned}$$

2.10.2 Citra *Hypostereo*

Adapun yang dimaksud dengan citra *hypostereo* yaitu mengacu pada gambar stereo yang menggunakan *lebih kecil* dari *stereobase* normal (jarak antara lensa kiri dan kanan). Dimana jarak kedua mata secara normal adalah 65 mm maka *stereobase* yang digunakan harus lebih kecil dari 65 mm.[17]

Besarnya *stereobase* pada citra *hypostereo* diukur menggunakan persamaan Di Marzio Frank dalam *stereophotography*. Secara sistematis persamaan Di Marzio Frank adalah :

$$B = D_n / 15 \dots\dots\dots (2.8)$$

Dimana :

B = *Stereobase* /jarak antar kamera

D_n = Jarak kamera ke objek terdekat

Persamaan ini hanya berlaku jika $D_f < 2D_n$, dimana D_n adalah jarak kamera ke objek terdekat dan D_f adalah jarak kamera dengan objek terjauh ($D_n +$ ketebalan objek).[14]

2.10.3 Citra *Hyperstereo*

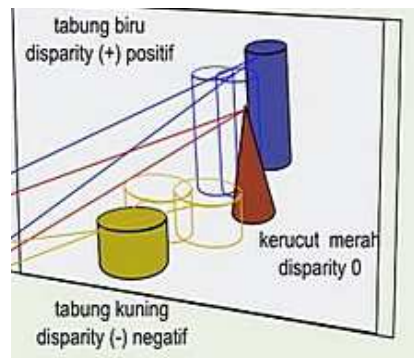
Citra *hyperstereo* mengacu pada gambar *stereo* yang menggunakan lebih besar dari *stereobase* normal (jarak antara lensa kiri dan kanan). Maka untuk citra *hyperstereo*, *stereobase* harus lebih besar dari 65 mm.[17]

Citra *hyperstereo* biasanya diperoleh dengan menggunakan dua kamera yang dipisahkan oleh jarak yang lebih besar dari jarak normal, atau menggunakan dua buah kamera yang identik dan dihubungkan dengan kabel sinkronisasi. Cara lain untuk memperoleh citra *hyperstereo* dengan menggunakan kamera tunggal, dengan membuat dua gambar secara terpisah. Foto pertama diambil, kemudian ambil foto kedua yang dipisahkan oleh jarak yang lebih besar dari *stereobase* normal. Tergantung pada subjek, *stereobase* dapat puluhan atau bahkan ratusan meter jaraknya. Ini memberikan efek *stereo* untuk objek yang cukup jauh yang biasanya tidak ada persepsi kedalaman.

Besar kecilnya *stereobase* akan mempengaruhi *disparity image*. *Disparity image* ini lah yang akan menunjukkan efek 3 dimensi, yang membuatnya berbeda dengan gambar-gambar 2 dimensi lainnya.[12]

Karena adanya *stereobase*, maka timbullah *disparity* atau bayang-bayang jika kedua gambar *stereoscopy* ditumpuk, bayang-bayang tersebut menyatakan bahwa kedua gambar tidaklah sama. Bayang-bayang itulah yang disebut *disparity image* yang akan memberikan efek *virtual* dan sensasi kedalaman.

Disparity image terdiri atas *disparity* negatif, *disparity* positif, dan *disparity* nol. *Disparity* negatif adalah bayangan yang akan tampil di depan kertas foto. sedangkan yang terlihat dibelakang adalah *disparity* positif. Sedangkan objek yang berhimpit atau tidak memili bayangan disebut *disparity* nol seperti yang ditunjukkan pada Gambar 2.15.



Gambar 2.15 Disparity Image

Sebuah citra *stereoscopy* memiliki *virtual reality* yang optimum jika memiliki 3 aspek seni foto 3 dimensi, yaitu :[12]

- a. semua object sebaiknya ke "dalam" atau disparity positif
- b. jika terdapat object yang keluar bingkai (pop up) atau disparity negatif maka gambar/object harus tidak terpotong frame gambar
- c. besar kecil disparity tergantung pada besar foto yang akan ditampilkan, apakah diatas kertas foto, di layar PC, di Blackberry, di Ipad atau dilayar bioskop. Untuk dikertas foto stereoscopy harus dicetak dalam ukuran 10R atau A4. sedangkan untuk di layer PC maka range ukuran monitor antara 11-22 inch.

2.10.4 Melihat Gambar *Stereoscopy*

Berikut adalah beberapa cara mudah untuk dapat melihat efek foto/gambar *stereoscopy* :[12]

2.10.4.1Langsung (Tanpa Alat Bantuan)

Untuk melihat gambar *stereoscopy* secara langsung, dikenal dua cara yaitu paralel view dan cross view. Pada paralel view, mata kiri hanya melihat foto kiri dan sebaliknya mata kanan hanya melihat foto sebelah kanan dari dua foto kiri dan kanan yang diletakkan berdampingan. Jangan fokus pada satu titik di gambar tersebut, buyarkanlah pandangan anda seolah-olah pandangan mengambang, lalu satukan keduanya.

Kekurangan cara ini adalah keterbatasan ukuran gambar (lebar). Lebar gambar tidak boleh melebihi jarak antara kedua pupil manusia. Bila terlalu lebar, gambar kiri dan kanan tidak akan ‘bergabung’ oleh pandangan pengamat.

Cara lain adalah cross view (melihat menyilang), perbandingan cara melihat tanpa alat dapat dilihat pada Gambar 2.16.



Gambar 2.16 Cara Melihat Gambar *Stereoscopy* Tanpa Alat Bantuan

2.10.4.2Dengan Alat Pengamat

Melihat *stereoscopy* dapat dilakukan dengan menggunakan sebuah cermin. Dengan menggunakan cermin, mata kiri melihat langsung ke foto kiri, dan mata kanan melihat langsung melihat gambar kanan melalui bayangan di cermin.

Dengan demikian masing-masing mata melihat masing-masing gambar (kiri/kanan). Ilusi 3 dimensi akan terjadi di sebelah kiri pengamat. Cara ini hanya

membutuhkan satu cermin dan salah satu foto yang dicetak terbalik (flip horizontal) dan ditunjukkan pada Gambar 2.17

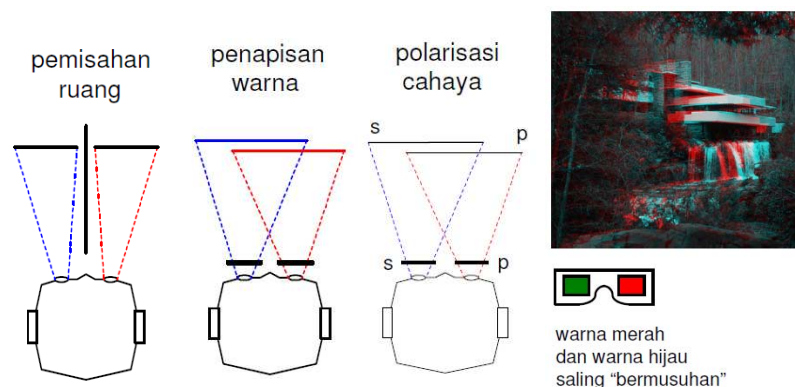


Gambar 2.17 Melihat Stereoscopy dengan Menggunakan Cermin

2.10.4.3 Dengan Kacamata Anaglyph

Untuk dapat melihat efek *stereoscopy*, salah satu cara adalah dengan teknik *anaglyph* penikmat diharuskan untuk mengenakan kacamata *anaglyph* yaitu kacamata berfilter warna sesuai dengan *color channel* citra *anaglyph* dimana masing-masing filter warna pada kacamata akan mem-block *color channel* yang berlawanan.[9]

Pada citra *stereoscopy* citra kanan dan citra kiri dipisahkan dengan cara pemisahan ruang, penapisan warna, dan polarisasi cahaya seperti pada Gambar 2.18.



Gambar 2.18 Pemisahan Citra Stereoscopy

Melihat melalui kacamata *anaglyph* (filter merah-biru) pada masing-masing mata, akan memisah kembali gambar/foto yang telah menyatu pada gambar *stereoscopy anaglyph*. Ketika mata kanan melihat foto kanan, foto kiri di blok. Dan mata kiri melihat foto kiri , foto kanan di blok. Kemudian kedua informasi foto dari kedua mata dibawa ke otak untuk diproses sehingga kita melihat sensasi kedalaman yang ditampilkan pada gambar.

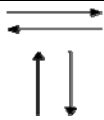

Kekurangan cara ini adalah foto berwarna menjadi terdistorsi (sangat baik untuk foto 3D hitam-putih) dan bila filter tidak sempurna, akan muncull bayangan yang tidak diinginkan (*ghost*).[12]





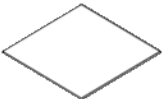


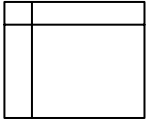
2.11 *Flowchart* (Diagram Alur)




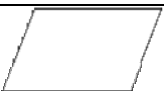





Flowchart adalah penyajian yang sistematis tentang proses dan logika dari kegiatan penanganan informasi atau penggambaran secara grafik dari langkah-langkah dan urutan-prosedur dari suatu program. *Flowchart* menolong analis dan programmer untuk memecahkan masalah kedalam segmen-segmen yang lebih kecil dan menolong dalam menganalisis alternatif-alternatif lain dalam pengoperasian.[11]

Sistem *flowchart* adalah urutan proses dalam sistem dengan menunjukkan alat media input, output serta jenis media penyimpanan dalam proses pengolahan data. Program *flowchart* adalah suatu bagan dengan simbol-simbol tertentu yang menggambarkan urutan proses secara mendetail dan hubungan antara suatu proses (instruksi) dengan proses lainnya dalam suatu program. Simbol-simbol yang digunakan dalam *flowchart* dapat dilihat pada Tabel 2.3.[11]

Tabel 2.3 Simbol-Simbol *Flowchart*

Simbol	Nama	Fungsi
	Garis Alir (<i>Flow Line</i>)	Arah aliran program
	<i>Communication</i>	Menyatakan transmisi data dari satu

	<i>Link</i>	lokasi ke lokasi lain
	<i>Connector</i>	Menyatakan sambungan dari proses ke proses lainnya dalam halaman yang sama
	<i>Offline Connector</i>	Menyatakan sambungan dari proses ke proses lainnya dalam halaman yang berbeda
	Proses	Menyatakan suatu tindakan (proses) yang dilakukan oleh komputer
	Manual	Menyatakan suatu tindakan (proses) yang tidak dilakukan oleh komputer
	<i>Decision</i>	Menujukkan suatu kondisi tertentu yang akan menghasilkan dua kemungkinan jawaban : ya / tidak
	<i>Predefined Process</i>	Menyatakan penyediaan tempat penyimpanan suatu pengolahan untuk memberi harga awal
	Terminal	Menyatakan permulaan atau akhir suatu program
	<i>Internal Storage</i>	Menyatakan penyimpanan yang disimpan sementara.

	<i>Keying Operation</i>	Menyatakan segala jenis operasi yang diproses dengan menggunakan suatu mesin yang mempunyai keyboard
	<i>Offline-storage</i>	Menunjukkan bahwa data dalam simbol ini akan disimpan ke suatu media tertentu
	Manual Input	Memasukkan data secara manual dengan menggunakan online keyboard
	<i>Input/Output</i>	Menyatakan proses input atau output tanpa tergantung jenis peralatannya
	<i>Punched Card</i>	Menyatakan input berasal dari kartu atau output ditulis ke kartu
	<i>Magnetic Tape</i>	Menyatakan input berasal dari pita magnetis atau output disimpan ke pita magnetis
	<i>Disk Storage</i>	Menyatakan input berasal dari disk atau output disimpan ke disk
	<i>Document</i>	Mencetak keluaran dalam bentuk dokumen (melalui printer)
	<i>Display</i>	Mencetak keluaran dalam layar monitor

Flowchart selalu diawali dan diakhiri oleh bagan terminator. Aliran selalu dari atas ke bawah, satu demi satu langkah. Tidak ada proses yang dikerjakan bersamaan, semua dikerjakan satu persatu.

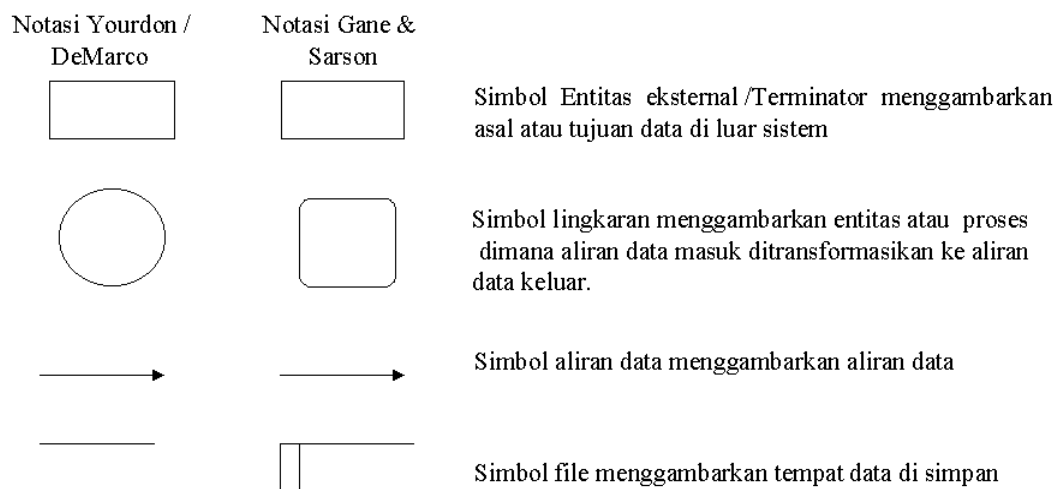
2.12 Data Flow Diagram (DFD)

Data Flow Diagram (DFD) adalah alat pembuatan model yang memungkinkan profesional sistem untuk menggambarkan sistem sebagai suatu jaringan proses

fungsional yang dihubungkan satu sama lain dengan alur data, baik secara manual maupun komputerisasi. DFD ini sering disebut juga dengan nama Bubble chart, Bubble diagram, model proses, diagram alur kerja, atau model fungsi.[13]

DFD ini adalah salah satu alat pembuatan model yang sering digunakan, khususnya bila fungsi-fungsi sistem merupakan bagian yang lebih penting dan kompleks dari pada data yang dimanipulasi oleh sistem. Dengan kata lain, DFD adalah alat pembuatan model yang memberikan penekanan hanya pada fungsi sistem.

DFD ini merupakan alat perancangan sistem yang berorientasi pada alur data dengan konsep dekomposisi dapat digunakan untuk penggambaran analisa maupun rancangan sistem yang mudah dikomunikasikan oleh profesional sistem kepada pemakai maupun pembuat program. Komponen yang digunakan dalam DFD diperlihatkan pada Gambar 2.17.



Gambar 2.19 Komponen DFD