BAB II

LANDASAN TEORI

2.1. Teori Mata Manusia

Melihat suatu benda, mata kanan dan mata kiri manusia melihat dengan prespektif yang berbeda. Hal tersebut terjadi karena jarak antara kedua pupil setiap mata berkisar antara 2-3 inchi. Selain itu, hidung juga menghalangi pandangan setiap mata ke sisi yang berlawanan. Oleh karena itu, ketika melihat sebuah obyek, setiap mata menangkap obyek dengan prespektifnya masing-masing, kemudian akan dikirimkan ke otak untuk diproses. (Martini, Timmons & Mckinley, 2000).

Perbedaan presfektif yang ditangkap oleh mata ini, yang digunakan dalam melihat *anaglyph image*. Dengan menggunakan *anaglyph glasses*, setiap mata akan melihat prespektif yang berbeda pada suatu citra. Ketika mata mengirimkan obyek yang dilihat ke otak, otak akan memproses obyek itu sehingga, obyek dalam citra tersebut akan tampak dalam 3D.

2.2. Sistem Warna RGB (Gonzales & Woods, 2001)

Image processing adalah suatu cara yang digunakan untuk melakukan suatu proses manipulasi sebuah citra. Sedangkan, pada *digital image processing* citra yang diproses, merupakan citra yang memiliki nilai *integer* pada setiap nilai Warna RGB-nya.

Warna ada di sekeliling manusia. Sensasi dari warna memberikan kegembiraan dan emosi pada kehidupan manusia. Semua obyek yang ada di bumi ini pasti memiliki warna. Tanpa warna, bumi ini pasti memiliki kekurangan keindahannya. Sensasi dari warna dapat menambahkan kegembiraan dalam kehidupan manusia.

Untuk dapat mengenali warna, mata manusia membutuhkan cahaya. Tanpa cahaya, tidak akan ada warna dan dunia RGB-*colour*. Cahaya dibuat dari gelombang energi yang digolongkan menjadi beberapa golongan. Golongan tersebut dinamakan dengan *spectrum*. Cahaya yang tampak di mata manusia memang berwarna putih. Namun, sebenarnya terdiri dari beberapa warna. *Spectrum-spectrum* warna tersebut tidak berwarna, tetapi *spectrum-spectrum* tersebut dapat menimbulkan sensasi warna.

Sistem warna RGB terdiri dari tiga bagian warna yang dominan, yaitu merah (*Red*), hijau (*Green*), dan biru (*Blue*). Tiga warna ini dianggap sebagai tiga warna utama pada *spectrum* cahaya.

Pada umumnya, setiap warna *red*, *green*, dan *blue* terdiri dari delapan (8) bit. Oleh karena itu, setiap *primary colors* memiliki nilai sebanyak 256 nilai. Variasi nilai yang dibentuk berkisar antara 0 hingga 255. Jika nilai-nilai setiap *primary colors* dikombinasikan, maka jenis warna yang dapat dibentuk adalah 16.777.216 (256^3 atau 2^{24}) warna. Meskipun demikian, nilai setiap warna *red*, *green*, dan *blue* tidak selalu terdiri dari delapan (8) bit saja. Namun, bisa terdiri dari lima (5) bit, enam belas (16) bit dan beberapa jenis lainnya.

2.3. *Anaglyph Image*

2.3.1. Sejarah *anaglyph*

Anaglyph image diperkenalkan pada tahun 1853 oleh W.Rollman. Prinsip *anaglyph image* ini merupakan citra dengan latar belakang hitam dengan garis biru dan merah. Citra ini dilihat dengan menggunakan *anaglyph glasses* dengan warna merah-biru. Jadi, pada awalnya prinsip *anaglyph* hanya digunakan untuk garis-garis saja.

Namun, pada tahun 1858, Joseph D'Almeida mulai memproyeksikan citra yang berbentuk lentera sihir 3D dengan warna merah dan hijau. Citra ini dilihat dengan menggunakan *anaglyph glasses* dengan warna merah dan biru. Pada waktu inilah, pertama kali dikenalkan *anaglyph image.*

Pada tahun 1881, Louis Ducas du Hauron mulai mencetak *anaglyph image-anaglyph image* yang pertama. Cara yang digunakan Hauron untuk mencetak *anaglyph image* dengan menggunakan dua negatif film stereoskopic untuk kertas yang sama, dengan satu film berwarna merah dan satu film berwarna biru atau hijau.

Pada tahun 1889, William Friese-Green mulai memproduksi *anaglyph image* dalam citra-citra yang bergerak. Kemudian pada tahun 1893, film ini mulai diluncurkan pada publik. Namun, film 3D ini mulai terkenal baru pada tahun 1920-an. Film-film semacam ini mulai meraup keuntungan secara besar-besaran pada tahun 1950. Kemudian pada tahun 1954, film 3D ‘‘*The Creature from the Black Lagoon*’’ meraih kesuksesan yang sangat besar.

Pada tahun 1975, *anaglyph movie* kembali diluncurkan dengan judul *"The Groove Room 3D"*. pada saat ini, masih ada beberapa film yang menggunakan *anaglyph* ini. Seperti pada tahun 2001, berjudul *"Haunted Castle"*. Kemudian pada tahun 2003, muncul kembali film dengan judul *"Spy Kid 3-D Game Over"*. Pada tahun 2005, film dengan judul *"Barbie and the Magic of Pegasus"*, *"Shark Boy and Lava Girl 3D"* dan *"Santa vs Snowman 3D"*. *"Santa vs Snowman 3D"* ini diluncurkan pada tahun 2002. Namun, versi 3D-nya baru diluncurkan pada tahun 2005.

Dalam media cetak, pada tahun 1953, *anaglyph* mulai dimuat dalam majalah, Koran, buku komik dan berbagai media cetak lainnya. Komik 3D ini merupakan pencetakan *anaglyph image* yang paling menarik dibandingkan dengan media cetak yang memuat *anaglyph image* yang lainnya ("History of Anaglyph", n.d.).

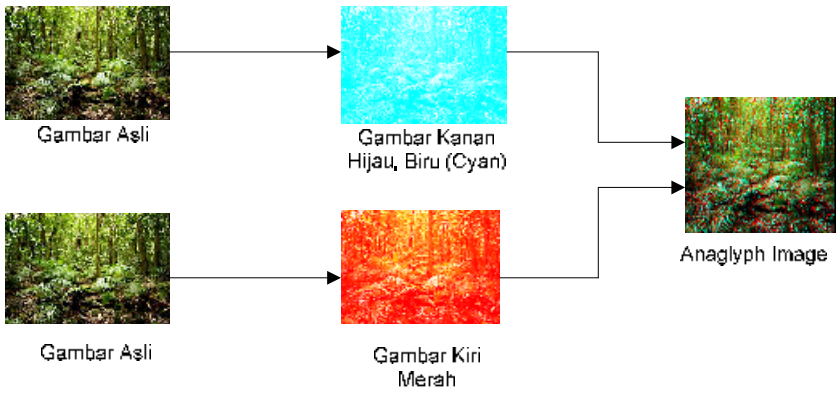
2.3.2. Pengertian *Anaglyph*

Anaglyph image adalah sebuah citra dimana ketika citra tersebut dilihat dengan mata kanan dan mata kiri akan saling bertumpukan dengan warna yang berbeda. Sedangkan, setiap mata hanya memfilter warna yang diterimanya dan mengirimkannya ke otak. Kemudian, otak akan memprosesnya dan menangkapnya sebagai citra dalam 3D. Dalam pengertian yang lain, *anaglyph image* merupakan sebuah citra yang dibentuk dari pengintegrasian dua citra.

Anaglyph image juga merupakan bagian dari grafik 3D yang meningkatkan persepsi kedalaman suatu benda 3D. citra yang dihasilkan tidak

hanya tampak sebagai benda datar pada layar, tetapi akan tampak seolah-olah muncul dari layar.

Secara fisik, *anaglyph image* merupakan citra dalam bentuk dua dimensi yang terlihat tidak jelas karena ada pergeseran *channel* warna R (*red*) pada citra tersebut. Jadi, ketika mata manusia melihat tanpa bantuan apapun, otak akan menangkap citra tersebut sebagai obyek yang tidak jelas karena ada bayangan pada citra tersebut. Oleh karena itu, mata manusia membutuhkan kacamata untuk membantu dalam melihat *anaglyph image* ini. Kacamata ini dinamakan dengan *anaglyph glasses*. Penjelasan pembuatan *anaglyph image* dapat dilihat pada Gambar 2.1.



Gambar 2.1. Pembuatan *Anaglyph*

Pada Gambar 2.1, terdapat dua citra, yaitu citra kiri dan citra kanan. Dari citra kiri diambil *channel* warna merah dan dari citra yang kanan diambil *channel*

warna hijau dan biru. Kemudian, kedua citra ini disatukan menjadi *anaglyph image*.

Keuntungan dari penggunaan *anaglyph image,* antara lain :

a. Tidak membutuhkan *optical gymnastics* seperti, menyilangkan mata.
b. Setiap orang yang memiliki kedua mata yang normal dapat melihat obyek yang tampak 3D pada citra ini.
c. Tampilan citra akan memenuhi seluruh komputer, tidak hanya setengah layar saja. Pemecahan resolusi dan *stereoskopic* akan tampak lebih baik dengan pasangannya.
d. *Anaglyph glasses* mudah untuk didapatkan. User dapat membuatnya sendiri. Dengan mika plastik warna merah untuk mata kiri dan mika plastic warna hijau untuk mata kanan, user telah mendapatkan *anaglyph glasses*.

Selain keuntungan dari pemakaian *anaglyph image*, *anaglyph image* juga memiliki tiga masalah utama dalam pembuatannya, antara lain :

a. *Ghosting*

   Jika warna citra yang ditujukan untuk mata kiri ditangkap oleh mata kanan, maka otak akan melihat bayangan obyek secara sayup-sayup.

b. *Retinal rivalry*

   Jika mata menerima kecerahan citra yang tidak sama, akan menimbulkan efek-efek yang tidak menyenangkan. Selain itu, akan

muncul kilauan-kilauan yang tidak normal sehingga tampak berkelap-kelip.

c. *Wrong colours*

Ada kemungkinan otak akan menerima warna yang berbeda dari warna aslinya karena setiap mata hanya menangkap sebagian dari keseluruhan RGB. Para ahli telah berusaha untuk mengatasi masalah ini. Namun sampai saat ini masalah ini belum dapat diselesaikan dengan sempurna.

(Wattie, 2007)

2.3.3. Jenis-jenis *Anaglyph Image*

Pada saat ini, ada beberapa jenis *anaglyph image* yang sedang dikembangkan, antara lain :

1. *Monochrome anaglyph*

Monochrome anaglyph disebut juga dengan *chromatic anaglyph*. *Anaglyph image* jenis ini menggunakan *anaglyph glasses* dengan warna *filter* merah untuk kiri dan hijau untuk kanan. Namun, *monochrome anaglyph* ini dapat juga dilihat dengan warna *filter* merah untuk kiri dan *cyan* untuk kanan.

Jika kedua warna tersebut dibalik, hasil anaglyph image tersebut dengan *pseudo-stereoscopy*, atau disebut juga dengan *reverse stereopsis.*

2. *Colour anaglyph*

Colour anaglyph pada umumnya dilihat dengan menggunakan *anaglyph glasses* dengan warna *filter* merah untuk kiri dan *cyan* untuk kanan. Namun ada

beberapa yang menggunakan *anaglyph glasses* dengan warna *filter* merah untuk kiri dan biru untuk kanan.

(Wattie, 2007)

3. *Polarized anaglyph*

Polarized anaglyph dilihat dengan *polarized glasses*. Ilusi 3D yang terjadi karena *polarized glasses* membatasi cahaya yang akan mencapai setiap mata. *Filter* atau kacamata yang dipakai tidak mahal, tetapi harus mengandung *filter orthogonally polarize. Filter* ini akan menimbulkan efek ketika setiap mata hanya akan melihat salah satu gambar saja.

2.4. *Anaglyph Glasses*

Anaglyph glasses terdiri dari dua warna yang berbeda untuk setiap sisi matanya. Pada saat ini, ada beberapa model kacamata dengan variasi *filter* warna yang beraneka ragam. Jenis-jenis kacamata *anaglyph image* berdasarkan *filter* warna yang digunakan, antara lain :

1. Warna merah untuk mata kiri dan tidak berwarna atau bening untuk mata kanan
2. Warna merah untuk mata kiri dan warna hijau untuk mata kanan
3. Warna merah untuk mata kiri dan warna biru untuk mata kanan
4. Warna merah untuk mata kiri dan warna *cyan* (kombinasi hijau dan biru) untuk mata kanan
5. Warna merah tua untuk mata kiri dan warna *cyan* dengan sedikit campuran dari warna merah untuk mata kanan

6. Warna coklat untuk mata kiri dan warna biru tua untuk mata kanan

Dari jenis-jenis *filter* warna yang telah dijabarkan diatas, pada tugas akhir ini hanya menggunakan jenis *filter* yang kedua, yaitu warna merah untuk mata kiri dan warna hijau untuk mata kanan. Pada pemakaian *anaglyph glasses* pada umumnya, jenis ini juga merupakan jenis yang paling sering digunakan.

Untuk dapat mendapatkan *anaglyph glasses* tersebut, *user* dapat membuatnya sendiri. *User* hanya perlu menyediakan mika plastik warna merah dan warna hijau. Kemudian, *user* dapat membentuk kedua mika plastik tersebut dengan kreasi sendiri. Ada hal yang paling penting yang perlu diperhatikan dalam pembuatan *anaglyph glasses* adalah kadar ketebalan antara kedua mika plastik warna merah dan hijau harus sama. Namun, jika terjadi perbedaan kadar ketebalan di antara kedua mika plastik tersebut, perbedaannya tidak boleh terlalu jauh. Model dari *anaglyph glasses* ini dapat dilihat pada Gambar 2.2.



Gambar 2.2. Gambar *Anaglyph Glasses*

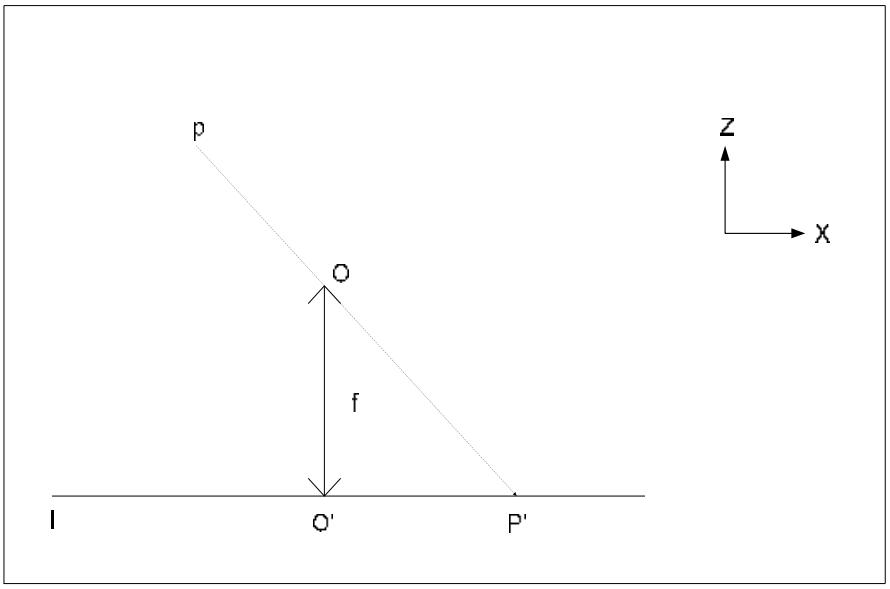
2.5. Pergeseran Citra

Untuk dapat mempelajari prespektif mata lebih dalam, dapat dilihat pada sistem dari sebuah kamera. Ilustrasi dari sistem kamera ini dapat dilihat pada Gambar 2.3. Pada Gambar 2.3. ini, lensa kamera terletak pada titik o (*focal point*

atau *center)*. Garis I merupakan layar untuk menampilkan gambar. Panjang f (*focal length)* merupakan panjang titik o dengan prespektifnya (o') pada layar. Sedangkan, p merupakan obyek yang akan ditampilkan dan p' merupakan prespektif yang ditarik ke layar I melalui titik o. Diasumsikan titik p merupakan titik dengan lokasi 3D yaitu (x, y, z) yang akan digambarkan pada layar I dengan dimensi 2D. Maka, rumus yang dihasilkan sebagai berikut :

$$\frac{x'}{f} = \frac{x}{z} \tag{2.1}$$

$$\frac{y'}{f} = \frac{y}{z} \tag{2.2}$$



Gambar 2.3. Prespektif Suatu Obyek Pada Sebuah kamera

Sumber : Cornell University

Keterangan Rumus dan Gambar :

x : nilai koordinat x pada titik p(*x*, *y*, *z*)

z : nilai koordinat z pada titik p(*x*, *y*, *z*)

f : jarak antara kamera dan layar

x’ : nilai koordinat x pada titik *p’*(*x’*, *y’*)

y : nilai koordinat y pada titik *p*(*x*, *y*, *z*)

y’ : nilai koordinat y pada titik *p’*(*x’*, *y’*)

p : titik dengan koordinat 3D (*x*, *y*, *z*)

o : titik letak kamera

f : jarak antara kamera dan layar

garis *I* : layar

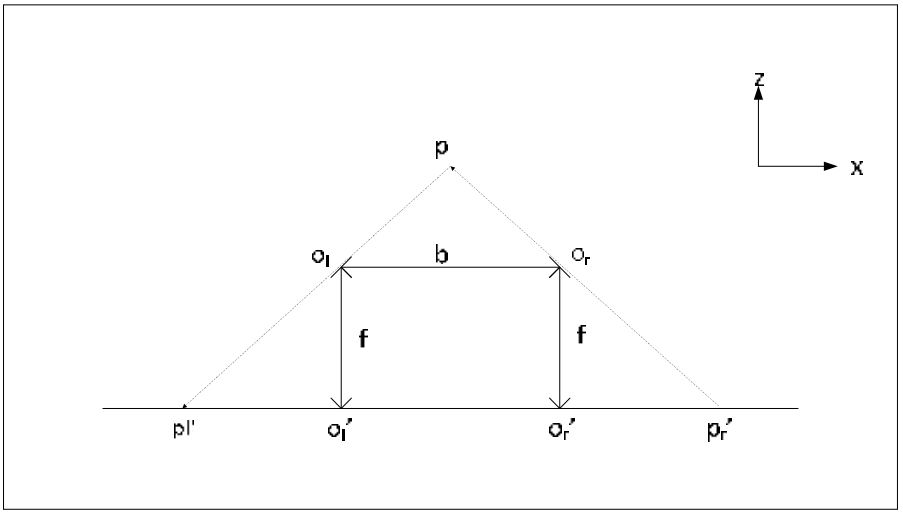
p' : hasil prespektif titik *p* ke layar *I* melalui titik *o*

Pada *anaglyph*, maka kamera yang dibutuhkan ada 2 posisi seperti yang dapat dilihat pada Gambar 2.4. Sehingga ada 2 titik pada lensa yaitu titik o_l untuk lensa sebelah kiri dan titik o_r untuk lensa sebelah kanan. Jarak antara kedua titik tersebut adalah *b*. Kemudian, diasumsikan jarak titik o_l dan titik o_r ke layar sama besar dengan jarak *f*. Kemudian titik *p* dengan lokasi 3D (*x*, *y*, *z*) diproyeksikan ke layar *I* melalui kedua titik lensa o_l dan lensa o_r menjadi dua lokasi 2D, yaitu titik *pl'* pada lokasi (*xl'*, *yl'*) dan titik *pr'* pada lokasi (*xr'*, *yr'*). Maka rumus yang dihasilkan sebagai berikut :

$$\frac{x_1'}{f} = \frac{x - b/z}{z} \tag{2.3}$$

$$\frac{x_r'}{f} = \frac{x - b/2}{z} \tag{2.4}$$

$$\frac{y_1'}{f} - \frac{y_r'}{z} = \frac{y}{z} \tag{2.5}$$



Gambar 2.4. Prespektif Suatu Obyek pada Dua Buah Kamera

Sumber: Cornell University

Keterangan Rumus dan Gambar:

x : nilai koordinat *x* pada titik *p*(*x*,*y*,*z*)

z : nilai koordinat *z* pada titik *p*(*x*,*y*,*z*)

f : jarak antara kamera dengan layar

b : jarak antara titik o_r dan titik o_l

x_r' : nilai koordinat x pada titik p_r'(x_r',y_r')

x_l' : nilai koordinat x pada titik p_l'(x_l',y_l')

y : nilai koordinat y pada titik *p*(*x*,*y*,*z*)

y_r' : nilai koordinat y pada titik p_r'(x_r',y_r')

y_l' : nilai koordinat y pada titik p_l'(x_l',y_l')

p : titik dengan koordinat 3D (*x*,*y*,*z*)

o_r : titik letak kamera yang kanan

o_l : titik letak kamera yang kiri

f : jarak antara kamera dengan layar

garis I : layar

p_r' : hasil prespektif titik p ke layar I melalui titik o_r

p_l' : hasil prespektif titik p ke layar I melalui titik o_l

Dari persamaan (2.2), maka dapat diketahui jarak pergeseran prespektif pada suatu obyek pada layar, yaitu antara titik p_l dan titik p_r. Jarak tersebut didapatkan dengan mengkurangkan titik x_l' dengan titik x_r'. Maka, rumus yang dihasilkan antara lain :

$$\frac{x_1' - x_r}{f} = \frac{b}{z} \tag{2.6}$$

Keterangan Rumus:

z : nilai koordinat z pada titik $p(x,y,z)$

f : jarak antara kamera dengan layar

b : jarak antara titik o_r dan titik o_l

x_r' : nilai koordinat x pada titik p_r'(x_r', y_r')

x_l' : nilai koordinat x pada titik p_l'(x_l', y_l')

Dari persamaan (2.3), dapat diketahui bahwa, semakin besar jarak *z* (titik *p*) dengan layar, maka jarak pergeseran hasil prespektif akan menjadi semakin kecil. Oleh karena itu, prinsip ini digunakan pula untuk melakukan pergeseran citra pada *anagyph image*. Dalam pembuatan *anaglyph image*, semakin dalam suatu obyek pada citra, maka pergeseran citra tersebut semakin kecil. Demikian pula sebaliknya.

(Huttenlocher,1992-1993)

Dalam pengimplementasian rumus tersebut kedalam aplikasi, variabel *f*, *b*, dan *z* di-*input*-kan dalam satuan sentimeter (cm). Sedangkan, aplikasi melakukan pergeseran dalam satuan *pixel*. Oleh karena itu, satuan sentimeter diubah ke *pixel* dengan ukuran 1cm = 37.795275591 *pixel*. (Anvica Software Development, 2002 2007).

2.6. Bitmap (BMP)

BMP atau *DIB* (*device independent bitmap*), adalah sebuah format grafik yang digunakan secara internal oleh microsoft windows dan subsistem OS/2, dan sering digunakan sebagai format file grafik sederhana pada *flatform-flatform* tersebut.

Gambar secara umum disajikan dalam ketajaman warna 2 (1-*bit*), 16 (4-*bit*), 256 (8-*bit*), 65.536 (16-*bit*), 16.7 juta (24-*bit*) warna (*bit-bit* ini mempresentasi *bit-bit* per-*pixel*). Sebuah gambar 8-*bit* juga dapat dirubah ke warna *grayscale* disamping warna indeks. Sebuah *channel alpha* (untuk warna transparan) boleh disajikan dalam *file* terpisah, dimana sama dengan gambar

grayscale. Versi 32-*bit* dengan *channel alpha* terintegrasi telah diperkenalkan oleh Windows XP dan digunakan untuk sistem *login* dan *theme*.

Umumnya file *BMP* menggunakan model warna *RGB.* Pada model ini sebuah warna terbentuk dari campuran intensitas yang berbeda (bervariasi dari 0 sampai 255), warna merah (R), hijau (G), dan biru (B). Dengan kata lain sebuah warna akan didefinisikan menggunakan 3 nilai, yaitu R, G dan B. Blok dari *bit* mendeskripsikan gambar secara *pixel* per *pixel*. *Pixel* disajikan mulai dari sudut kiri bawah berjalan dari kiri ke kanan dan kemudian baris per baris dari bawah ke atas. Setiap *pixel* dideskripsikan menggunakan satu atau lebih *bit*.

2.7. Video

Video merupakan sebuah film atau gambar hidup yang dihasilkan dengan rekaman dari orang dan benda (termasuk fantasi dan *figure* palsu) dengan menggunakan kamera, dan memiliki fungsi dua dimensi yang terbentuk dari penglihatan dalam suatu tempat (*scene)* yang merupakan basis dari pembentukan video. Secara umum video dibagi menjadi dua macam, yaitu :

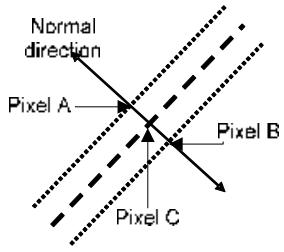
1. Analog, yaitu video hasil tangkapan lensa kamera terhadap tempat (*scene)* yang di-*scene* secara vertikal dan horizontal oleh *video* kamera.
2. *Digital*, yaitu video yang direpresentasikan sebagai sebuah matriks yang masing-masing elemennya merepresentasikan nilai intensitas

Video *digital* pada dasarnya tersusun atas serangkaian *frame*. Rangkaian *frame* tersebut ditampilkan pada layar dengan kecepatan tertentu , tergantung pada

frame rate yang diberikan (dalam *frame/second*). Jika *frame rate* cukup tinggi, mata manusia tidak dapat menangkap gambar atau *frame*, melainkan menangkapnya sebagai rangkaian yang kontinu/berlanjut (video).

Masing-masing *frame* merupakan citra *digital*. Suatu citra *digital* direpresentasikan dengan sebuah matriks yang masing-masing elemennya merepresentasikan nilai intensitas. Jika I adalah matriks dua dimensi, I(x,y) adalah nilai intensitas yang sesuai pada posisi baris x dan kolom y pada matriks tersebut. Titik-titik ditempatkan *image* di *sampling* disebut *picture elements*, atau sering dikenal sebagai piksel. *Pixel* atau piksel (*picture element /* unsur gambar*)* adalah tititk–titik kecil. Gambar apapun yang tampak di layar komputer sebenarnya tersusun dari titik-titik kecil.

Jika beberapa piksel diletakkan berderet maka yang tampak adalah sebuah garis. Jadi semua garis, sehalus apapun tampaknya dilayar komputer, sebenarnya adalah deretan piksel. Sebuah piksel memang bisa dianggap sebagai sebuah titik, namun dalam kenyataannya, piksel-piksel lebih mirip dengan persegi panjang kecil yang tingginya tidak sebanding dengan lebarnya.



Gambar 2.5. Piksel A, piksel B dan piksel C

2.8. Pengertian *Movie*

Movie terdiri dari beberapa citra yang mengalami perubahan sedikit demi sedikit. *Movie* dapat dibuat dari kamera *video* atau dengan membuat gambar yang dibuat beranimasi. *Movie* sendiri terdiri dari sekumpulan citra yang disebut dengan *frame*. Ketika citra-citra tersebut dirangkai dan dijalankan dengan cepat, maka animasi tersebut dapat terlihat.

Pada saat ini, ada beberapa standar untuk banyak *frame* setiap detiknya, antara lain :

a. 60i (*intelaced*; 50i = 50 *frame* di Eropa dan Australia) adalah standar *video field* untuk setiap detiknya (60 *interlaced fields* = 29.97 *frames*). Standar ini telah digunakan untuk pertelevisian selama beberapa tahun. Selain itu, standar ini juga digunakan untuk DVD dan *home camcorder*.

b. 30p (30 *frame progressive* adalah format yang *noninterlaced*) menghasilkan *video* dengan 30 *frame* tiap detiknya. Biasanya standar ini digunakan untuk perindustrian *music video* sejak 1980.

c. 24p (24 *frame progressive* adalah format yang *noninterlaced*) sekarang digunakan secara luas untuk melakukan transfer signal *video* ke film. Hal ini dikarenakan kamera untuk merekam film adalah *35 mm movie cameras* yang menggunakan standar 24 *frame* tiap detik.

d. 25p adalah sebuah format *video* yang menjalankan 25 *frame* tiap detik. Standar ini diambil berdasarkan PAL (standar dari televisi) yang mempunyai 50 *interlaced frame* tiap detik. Sementara 25p mengambil

separuh dari pergerakan gambar pada 50i PAL *register* yang normal. Hal ini dilakukan untuk memperoleh hasil gambar yang lebih baik karena *progressive-scan*. Standar ini cocok digunakan pada *output* yang berupa LCD, monitor komputer, dan proyektor. (Wikipedia, 2007)

2.9. AVI (Audio Video Interleave)

Audio Video Interleave atau dikenal dengan AVI adalah format multimedia yang dikenalkan oleh *Microsoft* pada bulan November 1992, bagian dari teknologi video untuk *windows*, file AVI dapat berisi data video dan audio, sebuah standarisasi yang berisikan sinkronisasi antara audio dan video, seperti DVD file AVI mendukung multi *streaming* audio dan video, disamping digunakan secara tersendiri, file AVI juga banyak digunakan sebagai format extensi file yang dikembangkan oleh *Matrox Open DML group* pada February 1996.

Pengembangan file AVI yaitu AVI 2.0 digunakan khusus untuk kasus *Resource interchange File Format* (RIFF), sebuah file AVI dapat dikompres kedalam bentuk format kompresi multimedia seperti *Motion JPEG*, *VDOWave*, *ClearVideo/RealVideo*, dan lain-lain serta dapat melakukan *share file* dan melakukan proses *rendering* seperti *capture, edit, playback.* Pada file AVI terdapat beberapa karakteristik dari AVI antara lain :

1. File AVI memiliki masalah dalam ukuran , karena AVI dapat mendukung file yang besar tetapi dalam aplikasinya ada beberapa permasalahan yang terjadi

contohnya *player-player* yang mendukung MPEG tidak mendukung file AVI berukuran lebih dari 2 GB.

2. Spesifikasi dari AVI dan kelebihan yang dikembangkan oleh *Matrox Open DML group*, AVI dapat menangani 256 *stream*s baik itu audio ataupun video.
3. AVI dapat menangani subtitle *text*, karena AVI memiliki *AVI-Mux GUI* dan *DVobSub/VSFilter*.
4. *Subtitle* dalam AVI sangat mudah diimplementasikan baik melakukan pemotongan atau penggabungan *subtitle*, blok data dari *subtitle* dapat dilakukan dengan pemrosesan, penampilan, pembuatan, dan perbaikan.
5. AVI tidak mendukung *Bidirectional coding* (*B-frame* pada MPEG), karena AVI tidak dirancang untuk menangani prediksi dari pergerakan *frame* (P-*frame*) dan arah gerak *frame* (*B-frame*) seperti format MPEG.

Untuk lebih jelas mengenai isi dari *file* AVI dapat dilihat pada Gambar 2.5.

Headers
Picture Frame #1
Audio Frame #1
Picture Frame #2
Audio Frame #2
Picture Frame #*n*
Audio Frame #*n*

Gambar 2.6. Struktur *File* AVI

Sedangkan isi dari *header file* AVI mengacu pada format RIFF (*Resource Interchange File Format*). Di dalam *header* AVI terdapat data-data, antara lain:

FOURCC fcc;

DWORD cb;

DWORD dwMicroSecPerFrame;

DWORD dwMaxBytesPerSec;

DWORD dwPaddingGranularity;

DWORD dwFlags;

DWORD dwTotalFrames;

DWORD dwInitialFrames;

DWORD dwStreams

DWORD dwSuggestedBufferSize;

DWORD dwWidth;

DWORD dwHeight;

DWORD dwReserved[4];

2.9.1. Komponen *AviLoader*

Komponen *AviLoader* digunakan untuk membuka *file* yang bertipe AVI, memainkannya, serta membagi file tersebut ke dalam *frame-frame* yang berbentuk citra. Komponen ini menggunakan *file* avifil32.dll yang terdapat dalam *folder* system32 pada Windows. Jika *file* ini tidak terdapat dalam *folder* tersebut, maka

program tersebut tidak dapat menjalankan fungsi dari komponen *AviLoader*. Komponen ini dibuat oleh HydroS Co pada tahun 1995-2000.

Untuk dapat menggunakan komponen *AviLoader*, hanya perlu menghubungkan *unit* a_vfw dengan aplikasi yang membutuhkan. Kemudian, aplikasi hanya perlu memanggil fungsi-fungsi yang disediakan oleh *unit* tersebut. Fungsi-fungsi dan yang disediakan oleh komponen ini dapat dilihat pada Tabel 2.1.

Tabel 2.1 Tabel Fungsi-fungsi yang Disediakan oleh Komponen *AviLoader*

Nama Fungsi yang Disediakan	Digunakan untuk
AVIFileInit	Menginisialisasikan *file* AVI yang akan di-*load*
AVIFileOpen	Membuka *file* AVI
AVIFileInfo	Mengetahui data-data *header* dari *file* AVI

2.9.2 Komponen *AviWriter_2*

Komponen *AviWriter_2* digunakan untuk menghasilkan *file* yang bertipe AVI. Seperti komponen *AviLoader*, komponen *AviWriter_2* juga menggunakan *file* avifil32.dll. Demikian pula, jika *file* tersebut tidak terdapat dalam *folder* systems32 pada Windows, komponen *AviWriter_2* ini tidak dapat dijalankan. Komponen ini dibuat oleh Renate Schaaf.

Di dalam komponen ini terdapat fungsi-fungsi untuk menambahkan *frameframe* yang berupa citra pada *file* AVI. Selain itu, komponen ini juga bisa digunakan untuk menambahkan *file audio*. Untuk dapat mengerti fungsi-fungsi yang disediakan oleh AviWriter dengan lebih jelas, dapat dilihat pada Tabel 2.2.

Tabel 2.2. Tabel Fungsi-fungsi yang Disediakan oleh Komponen *AviWriter_2*

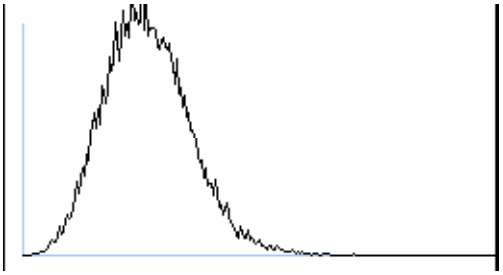
Nama Fungsi-fungsi yang Disediakan	Digunakan untuk
InitVideo	Menginisialisasi *file* AVI yang akan dibuat
AddFrame(*file* BMP)	Menambahkan *frame* baru
FinalizeVideo	Menandakan bahwa penambahan *frame* telah selesai
WriteAvi	Meng-*input*-kan *frame-frame* yang telah ditambahkan pada fungsi *AddFrame* ke dalam *file* AVI

2.10. Histogram

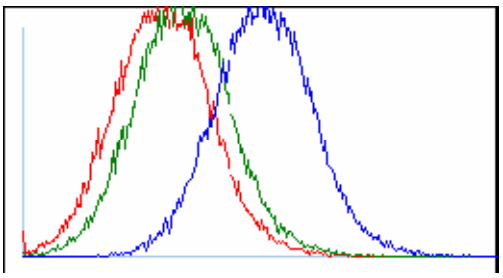
Salah satu alat bantu yang paling sederhana dan sangat berguna dalam pengolahan citra digital adalah histogram tingkat keabuan (gray-level histogram). Informasi suatu citra seringkali dapat diwakili oleh histogram ini.

Histogram tingkat keabuan adalah suatu fungsi yang menunjukkan jumlah titik yang ada di dalam suatu citra untuk setiap tingkat keabuan. Absis (sumbu x)-nya adalah tingkat keabuan, dan ordinat (sumbu y)-nya adalah frekuensi

kemunculan atau banyaknya titik dengan nilai keabuan tertentu. Gambar 2.6 menunjukkan contoh histogram dari sebuah citra kain yang berformat citra skala keabuan 8 bit. Sedangkan Gambar 2.7 menunjukkan contoh citra warna, warna dasar histogram dibuat masing-masing: merah, hijau, dan biru.



Gambar 2.7. Contoh histogram sebuah citra skala keabuan



Gambar 2.8. Contoh histogram sebuah citra warna

2.10.1. Histogram Citra

Informasi yang dikandung oleh sebaran frekuensi lebih mudah ditangkap bila disajikan secara grafik. Pada kebanyakan orang, gambar visual sangat membantu dalam memahami ciri-ciri penting pada suatu sebaran frekuensi. Salah satu sajian grafik yang biasa digunakan adalah dalam bentuk histogram.

Histogram citra adalah grafik yang menggambarkan penyebaran nilai-nilai intensitas pixel dari suatu citra atau bagian tertentu di dalam citra. Dari sebuah

histogram dapat diketahui frekuensi kemunculan nisbi (relative) dari intensitas pada citra tersebut. Histogram juga memberikan gambaran tentang komposisi citra, informasi tentang kontras, dan distribusi intensitas citra secara keseluruhan.

1. Membuat histogram

   Langkah ini dilakukan dengan menghitung jumlah (frekuensi) setiap nilai pixel dalam citra. Pembuatan histogram ini diawali dengan persiapan sebuah array yang berisi nol. Sebuah citra yang memiliki jumlah warna sebanyak 8 bit, memerlukan array berukuran 256 (0 sampai 255). Kemudian proses dilanjutkan dengan mengisi array tersebut dengan jumlah masing-masing nilai pixel pada citra. Misalkan citra digital memiliki L derajat keabuan, yaitu dari nilai 0 sampai L – 1 (pada citra dengan kuantisasi derajat keabuan 8-bit, nilai derajat keabuan dari 0 sampai 255). Secara matematis histogram citra dihitung dengan rumus :

$$hi = \frac{ni}{n} \qquad , i = 0,1,2 \ldots . , L - 1$$

   yang dalam hal ini,

   L = derajat keabuan

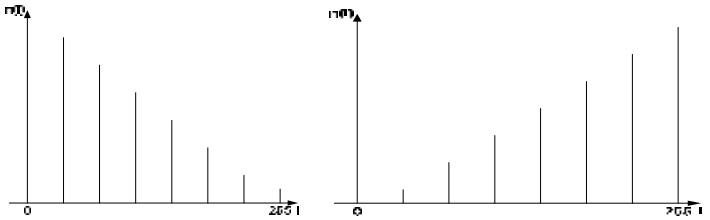
   ni = jumlah pixel yang memiliki derajat keabuan i

   n = jumlah seluruh pixel di dalam citra

   Nilai h i berada di dalam selang 0 sampai 1

   Plot h i versus f i dinamakan histogram.

Pada gambar 2.11 di gambarkankan histogram derajat keabuan pada sebuah citra (a) histogram dengan keutamaan titik gelap (b) histogram dengan keutamaan titik terang.



(a) (b)

Gambar 2.9. (a) Histogram citra dengan dominasi titik gelap. (b) Histogram citra dengan dominasi titik terang

Distribusi p(i) atau n(i) dapat memberikan informasi tentang keadaan dari suatu citra, sebuah citra mempunyai distribusi yang hampir sama dengan gambar di atas.