

## BAB II

### LANDASAN TEORI

#### 2.1. Webcam

*Web camera* atau yang biasa dikenal dengan *webcam*, adalah kamera yang gambarnya bisa di akses menggunakan *world wide web* (www), program *instant messaging*, atau aplikasi komunikasi dengan tampilan *video* pada PC. *Webcam* juga digambarkan sebagai kamera *video* digital yang sengaja didesain sebagai kamera dengan resolusi rendah. *webcam* dapat digunakan untuk sistem keamanan. Pada beberapa *webcam*, ada yang di lengkapi dengan *software* yang mampu mendeteksi pergerakan dan suara. Dengan *software* tersebut, memungkinkan PC yang terhubung ke kamera untuk mengamati pergerakan dan suara, serta merekamnya ketika terdeteksi. Hasil rekaman ini bisa disimpan pada komputer, e-mail atau di *upload* ke internet (Wibowo, 2010).

*Webcam* sangat bermanfaat dalam bidang telekomunikasi, bidang keamanan dan bidang industri. Sebagai contoh *webcam* digunakan untuk *videocall chatting*, *surveillance camera*, dan sebagai *video conference* oleh beberapa *user*.

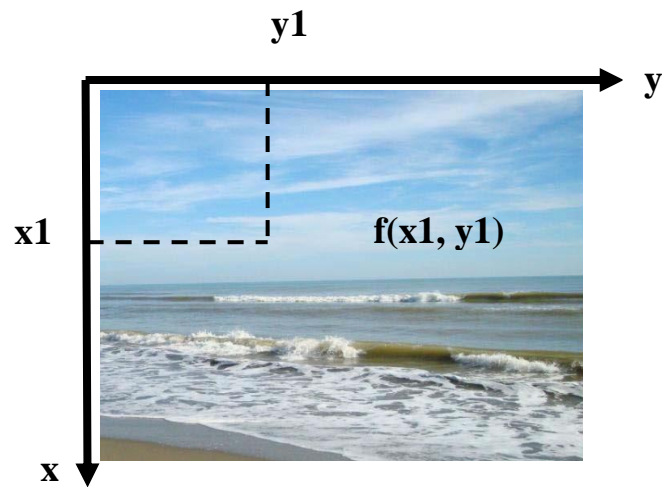


Gambar 2.1 Webcam (M.Syarif , 2011)

## 2.2. Citra Digital

Citra digital adalah citra dua dimensi yang dapat ditampilkan pada layar monitor komputer sebagai himpunan berhingga (diskrit) nilai digital yang disebut *pixel* (*picture elements*). *Pixel* adalah elemen citra yang memiliki nilai yang menunjukkan intensitas warna. Berdasarkan cara penyimpanan atau pembentukannya, citra digital dapat dibagi menjadi dua jenis. Jenis pertama adalah citra digital yang dibentuk oleh kumpulan *pixel* dalam array dua dimensi. Citra jenis ini disebut citra *bitmap* atau citra *raster*. Jenis citra yang kedua adalah citra yang dibentuk oleh fungsi-fungsi geometri dan matematika. Jenis citra ini disebut grafik vektor. Citra digital (diskrit) dihasilkan dari citra analog (*kontinu*) melalui digitalisasi. Digitalisasi citra analog terdiri *sampling* dan *quantization*. *Sampling* adalah pembagian citra ke dalam elemen-elemen diskrit (*pixel*), sedangkan *quantization* adalah pemberian nilai intensitas warna pada setiap *pixel* dengan nilai yang berupa bilangan bulat (G.W. Awcock, 1996).

Citra digital dapat didefinisikan sebagai fungsi dua variabel,  $f(x,y)$ , dimana  $x$  dan  $y$  adalah koordinat spasial dan nilai  $f(x,y)$  adalah *brightness level* dari citra pada koordinat tersebut, hal tersebut diilustrasikan pada Gambar 2.2



Gambar 2.2 Citra Digital (Nixon dan Aguado, 2002)

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).

### 2.3. Pengolahan Citra Digital

Pengolahan citra merupakan teknik manipulasi citra secara digital yang khususnya menggunakan komputer, menjadi citra lain yang sesuai untuk digunakan dalam aplikasi tertentu. Agar mudah diinterpretasi oleh manusia atau komputer, pengolahan citra harus dilakukan dengan berbagai macam metode untuk mencapai citra sesuai yang diinginkan.

Operasi pengolahan citra digital umumnya dilakukan dengan tujuan memperbaiki kualitas suatu gambar sehingga dapat dengan mudah diinterpretasikan oleh mata manusia dan untuk mengolah informasi yang ada pada suatu gambar untuk kebutuhan identifikasi objek secara otomatis.

Operasi-operasi yang dilakukan di dalam pengolahan citra banyak jenisnya. Secara umum, operasi pengolahan citra dapat diklasifikasikan dalam beberapa jenis sebagai berikut:

1. *Image enhancement.*

Jenis operasi ini bertujuan untuk memperbaiki kualitas citra dengan cara memanipulasi parameter-parameter citra.

Dengan operasi ini, ciri-ciri khusus yang terdapat di dalam citra lebih ditonjolkan. Contoh-contoh operasi perbaikan citra:

- a. perbaikan kontras gelap/terang
- b. perbaikan tepian objek (*edge enhancement*)
- c. penajaman (*sharpening*)
- d. *grayscale*
- e. *noise filtering*

2. *Image restoration.*

Operasi ini bertujuan menghilangkan/ meminimumkan cacat pada citra. Tujuan pemugaran citra hampir sama dengan operasi perbaikan citra. Bedanya, pada pemugaran citra penyebab degradasi gambar diketahui. Contoh-contoh operasi pemugaran citra:

- a. penghilangan kesamaran (*deblurring*)
- b. penghilangan *noise*

3. *Image compression.*

Jenis operasi ini dilakukan agar citra dapat direpresentasikan dalam bentuk yang lebih kompak sehingga memerlukan memori yang lebih sedikit. Hal penting yang harus diperhatikan dalam pemampatan adalah citra yang telah dimampatkan

harus tetap mempunyai kualitas gambar yang bagus. Contoh metode pemampatan citra adalah metode JPEG.

#### 4. *Image segmentation.*

Jenis operasi ini bertujuan untuk memecah suatu citra ke dalam beberapa segmen dengan suatu kriteria tertentu. Jenis operasi ini berkaitan erat dengan pengenalan pola.

#### 5. *Image analysis.*

Jenis operasi ini bertujuan menghitung besaran kuantitatif dari citra untuk menghasilkan deskripsinya. Teknik pengorakan citra mengekstraksi ciri- ciri tertentu yang membantu dalam identifikasi objek. Proses segmentasi kadangkala diperlukan untuk melokalisasi objek yang diinginkan dari sekelilingnya. Contoh-contoh operasi pengorakan citra:

- a. Pendeteksian tepi objek (*edge detection*)
- b. Ekstraksi batas (*boundary*)
- c. Representasi daerah (*region*)

#### 6. *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. (Nixon dan Aguado, 2002)

### 2.3.1. *Gray-Scaling*

Proses awal yang banyak dilakukan dalam *image processing* adalah mengubah citra berwarna menjadi citra *gray-scale*, hal ini digunakan untuk

menyederhanakan model citra. Seperti telah dijelaskan di depan, citra berwarna terdiri dari 3 layer matrik yaitu R-layer, G-layer dan B-layer. Sehingga untuk melakukan proses-proses selanjutnya tetap diperhatikan tiga layer di atas. Bila setiap proses perhitungan dilakukan menggunakan tiga layer, berarti dilakukan tiga perhitungan yang sama. Sehingga konsep itu diubah dengan mengubah 3 layer di atas menjadi 1 layer matrik gray-scale dan hasilnya adalah citra *gray-scale*. Dalam citra ini tidak ada lagi warna, yang ada adalah derajat keabuan.

Untuk mengubah citra berwarna yang mempunyai nilai matrik masing-masing  $r$ ,  $g$  dan  $b$  menjadi citra *gray-scale* dengan nilai  $s$ , maka konversi dapat dilakukan dengan mengambil rata-rata dari nilai  $r$ ,  $g$  dan  $b$  sehingga dapat dituliskan menjadi:

$$I(x,y) = \frac{R(x,y) + G(x,y) + B(x,y)}{3} \dots\dots\dots (2.1)$$

Keterangan:

$I(x,y)$  = nilai intensitas citra grayscale

$R(x,y)$  = nilai intensitas warna merah dari citra asal

$G(x,y)$  = nilai intensitas warna hijau dari citra asal

$B(x,y)$  = nilai intensitas warna biru dari citra asal

Atau dapat menggunakan persamaan:

$$I = (0,299 \times R(x,y)) + (0,587 \times G(x,y)) + (0,144 \times B(x,y)) \dots\dots\dots (2.2)$$

(Taylor dan Francis Group, 2007)

Berikut adalah contoh gambar hasil grayscaleing



Gambar 2.3 Hasil *Grayscaleing*( Nata S, 2010 )

### 2.3.2. *Thresholding*

*Thresholding* merupakan konversi citra berwarna ke citra biner yang dilakukan dengan cara mengelompokkan nilai derajat keabuan setiap *pixel* kedalam 2 kelas, hitam dan putih. Pada citra hitam putih terdapat 256 level, artinya mempunyai skala “0” sampai “255” atau [0,255], dalam hal ini nilai intensitas 0 menyatakan hitam, dan nilai intensitas 255 menyatakan putih, dan nilai antara 0 sampai 255 menyatakan warna keabuan yang terletak antara hitam dan putih (Munir, 2004).

Pada operasi pengambangan, nilai intensitas *pixel* dipetakan ke salah satu dari dua nilai,  $\alpha_1$  atau  $\alpha_2$ , berdasarkan nilai ambang (*threshold*)  $T$  dapat ditunjukkan seperti pada persamaan 2.1.

$$f(x, y)' = \begin{cases} \alpha_1, & f(x, y) < T \\ \alpha_2, & f(x, y) \geq T \end{cases} \dots\dots\dots(2.3)$$

Jika  $\alpha_1 = 0$  dan  $\alpha_2 = 1$ , maka operasi pengambangan mentransformasikan citra hitam-putih ke citra biner. Dengan kata lain, nilai intensitas *pixel* semula dipetakan ke dua nilai saja: hitam dan putih. Nilai ambang yang dipakai dapat

berlaku untuk keseluruhan *pixel* atau untuk wilayah tertentu saja (berdasarkan penyebaran nilai intensitas pada wilayah tersebut) (Munir, 2004).

*Thresholding* merupakan salah satu teknik segmentasi yang baik digunakan untuk citra dengan perbedaan nilai intensitas yang signifikan antara latar belakang dan objek utama. Dalam pelaksanaannya *thresholding* membutuhkan suatu nilai yang digunakan sebagai nilai pembatas antara objek utama dengan latar belakang, dan nilai tersebut dinamakan dengan *threshold*. Sementara pada kasus segmentasi objek yang membutuh dua nilai T atau lebih, maka disebut dengan *Multiple Threshold* (Gonzales dan Woods, 2002).

### 2.3.3. *Background Subtraction*

Subtraksi latar mempunyai peranan penting dalam visi komputer yaitu dalam hal sistem pemantauan. Peranan yang dilakukan dalam subtraksi latar adalah mengetahui atau membedakan bagian latar dan objek yang ada pada sebuah citra. Untuk mengetahui adanya objek di dalam citra maka hal yang harus dilakukan adalah dengan mempelajari atau mengetahui model dari latar. Ide dasar dari subtraksi latar adalah

$$| \text{background}(r) - \text{image}(r) | = \text{frame}(r) \dots \dots \dots (2.4)$$

$$| \text{background}(g) - \text{image}(g) | = \text{frame}(g) \dots \dots \dots (2.5)$$

$$| \text{background}(b) - \text{image}(b) | = \text{frame}(b) \dots \dots \dots (2.6)$$

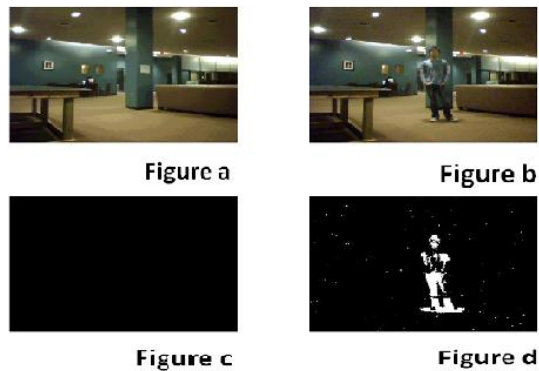
$$\text{frame}(r) > \text{threshold} \dots \dots \dots (2.7)$$

$$\text{frame}(g) > \text{threshold} \dots \dots \dots (2.8)$$

$$\text{frame}(b) > \text{threshold} \dots \dots \dots (2.9)$$



Bila *pixel* ke-*i* memenuhi persamaan tersebut, maka *pixel* tersebut digolongkan ke dalam kelompok *pixel* objek dan selain itu adalah latar. Berikut adalah contoh hasil subtraksi latar.

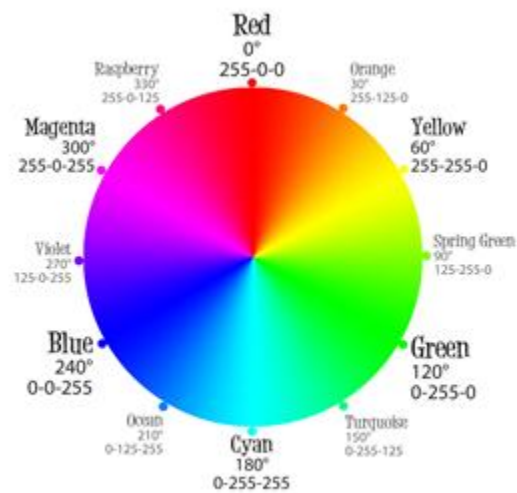


Gambar 2.4 Subtraksi Latar (Picardi dan Jan, 2003)

## 2.4. Color Space

### 2.4.1. RGB

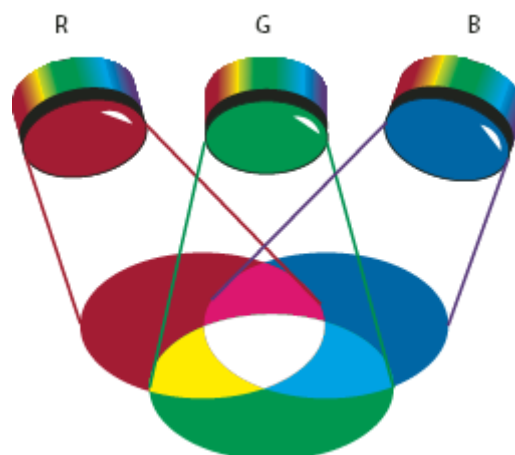
Model warna RGB adalah model warna berdasarkan konsep penambahan kuat cahaya primer yaitu *Red*, *Green* dan *Blue*. Dalam suatu ruang yang sama sekali tidak ada cahaya, maka ruangan tersebut adalah gelap total. Tidak ada signal gelombang cahaya yang diserap oleh mata kita atau RGB (0,0,0). Ruang warna RGB.dapat dilihat pada Gambar 2.5.



Gambar 2.5 RGB *Colorwheels* (Syuhadi,2010)

Apabila kita menambahkan cahaya merah pada ruangan tersebut, maka ruangan akan berubah warna menjadi merah misalnya RGB (255,0,0), semua benda dalam ruangan tersebut hanya dapat terlihat berwarna merah. Demikian apabila cahaya kita ganti dengan hijau atau biru (Permana,2010). Pencerahan serta pengaruh intensitas cahaya dalam ruang warna RGB dapat dilihat pada Gambar

2.6.



Gambar 2.6 RGB *Lightness* (Permana,2010)

## 2.5. *Computer Vision*

*Computer Vision* adalah pencitraan komputer dimana aplikasi tidak melibatkan manusia dalam proses pengulangan visual. Dengan kata lain, gambar yang diperiksa dan di olah oleh komputer. Meskipun orang yang terlibat dalam pengembangan sistem aplikasi, akhirnya membutuhkan komputer untuk mengambil informasi visual secara langsung (Umbaugh, 1998).

*Computer vision* merupakan sebuah proses otomatis yang mengintegrasikan sejumlah besar proses persepsi visual, seperti pengolahan citra, klasifikasi citra, pengenalan citra dan akuisisi citra. *Computer vision* didefinisikan sebagai salah satu cabang ilmu pengetahuan yang mempelajari bagaimana komputer dapat mengenali obyek yang diamati atau diobservasi. Cabang ilmu ini bersama kecerdasan buatan (*Artificial Intelligence*) akan mampu menghasilkan sistem kecerdasan visual (*Visual Intelligence System*) (Munir, 2004).

$$Vision = Geometri + Measurement + Interpretatio.....(2.10)$$

Proses-proses dalam *computer vision* dapat dibagi menjadi tiga aktivitas:

- a. Memperoleh atau mengakuisisi citra digital.
- b. Melakukan teknik komputasi untuk memproses atau memodifikasi data citra.
- c. Menganalisis dan menginterpretasi citra dan menggunakan hasil pemrosesan untuk tujuan tertentu, misalnya memandu robot, mengontrol peralatan, memantau proses manufaktur, dan lain-lain.

## 2.6. OpenCV

OpenCV (*Open Computer Vision*) adalah sebuah API (*Application Programming Interface*) library yang sudah sangat familiar pada pengolahan citra *computer vision*. *Computer vision* itu sendiri adalah salah satu cabang dari bidang ilmu pengolahan citra (*Image Processing*) yang memungkinkan komputer dapat melihat seperti manusia. Dengan *computer vision* tersebut komputer dapat mengambil keputusan, melakukan aksi, dan mengenali terhadap suatu objek. Beberapa pengimplementasian dari *computer vision* adalah *face recognition*, *face detection*, *face/object tracking*, *road tracking*, dll.

OpenCV adalah *library open source* untuk *computer vision* untuk C/C++, OpenCV didesain untuk aplikasi *real-time*, memiliki fungsi-fungsi akuisisi yang baik untuk *image/video*. OpenCV juga menyediakan *interface* ke *Integrated Performance Primitives* (IPP) Intel sehingga jika anda bisa mengoptimasi aplikasi *computer vision* anda jika menggunakan prosesor Intel (Syafi'i, 2011).

Fitur yang dimiliki OpenCV antara lain :

1. Manipulasi data citra (*allocation, copying, setting, convert*).
2. Citra dan video I/O (*file dan kamera based input, image/video file output*).
3. Manipulasi Matriks dan Vektor beserta rutin-rutin aljabar linear (*products, solvers, eigenvalues, SVD*).
4. Data struktur dinamis (*lists, queues, sets, trees, graphs*).
5. Pemroses citra fundamental (*filtering, edge detection, corner detection, sampling and interpolation, color conversion, morphological operations, histograms, image pyramids*).

6. Analisis struktur(*connected components, contour processing, distance Transform, various moments, template matching, Hough Transform, polygonal approximation, line fitting, ellipse fitting, Delaunay triangulation*).
7. Kalibrasi kamera (*calibration patterns, estimasi fundamental matrix, estimasi homography, stereo correspondence*).
8. Analisis gerakan (*optical flow, segmentation, tracking*).
9. Pengenalan obyek (*eigen-methods, HMM*).
10. Graphical User Interface (*display image/video, penanganan keyboard dan mouse handling, scroll-bars*).

OpenCV terdiri dari 3 *library*, yaitu:

1. CV : Untuk algoritma *Image Processing* dan *Vision*
2. Highgui : Untuk GUI, *Image* dan *Video I/O*
3. CXCORE : Untuk struktur data, *support XML* dan fungsi-fungsi grafis.

