

2. TEORI PENUNJANG

Pada bab ini yang akan dibahas meliputi teori-teori yang diperlukan dalam mewujudkan tiap metode yang ada. Teori-teori yang akan diperlukan dalam mewujudkan tiap metode adalah *computer vision* dan *image processing*, sistem warna, *open source computer vision library*, *intel® image processing library*, segmentasi berdasarkan warna, image enhancement, frame proses dan bagaimana *men-tracking* posisi obyek dengan titik berat.

2.1. Computer Vision Dan Image Processing

Pada tugas akhir ini *Computer Vision* merupakan suatu hal yang sangat besar perannya. Peran dari *Computer Vision* adalah sebagai satu-satunya penyedia data input bagi komputer untuk mengerti keadaan di sekelilingnya sehingga komputer dapat memberi respon.

Berikut adalah beberapa permasalahan dalam *computer vision* yang merupakan fokus utama:

1. *Sensing*

Bagaimana sensor memperoleh gambar dari dunia luar ? Bagaimana gambar tersebut men-*encode* properti dari dunia seperti *material*, bentuk, iluminasi dan hubungan *spatial* dan bahkan dunia 3D, termasuk geometri, tekstur, *motion*, dan identitas dari obyek didalamnya sehingga dapat digunakan oleh komputer.

2. *Decoded Information*

Bagaimana caranya untuk membuka dan mengambil setiap informasi yang ada di dalam gambar itu sehingga komputer dapat mendapat semua informasi selengkap-lengkapnyanya dan dapat bereaksi dengan setepat-tepatnya.

3. *Using the information*

Memilih informasi apa saja yang benar-benar sangat dibutuhkan dan harus diprioritaskan lebih dari pada yang lainnya. Juga harus dipilih informasi apa dalam gambar itu yang justru harus dibuang karena dapat mengganggu jalannya system.

4. *Algorithms*

Metode/algoritma apa yang dibutuhkan untuk memproses informasi dari gambar dan bagaimana memanfaatkannya.

Untuk masalah *sensing* tentunya kita akan memanfaatkan adanya kamera digital yang dapat mengambil gambar dari luar. Namun kita harus juga selektif dalam memilih kamera sehingga hasil gambar yang didapat dapat dipertanggungjawabkan. Selain itu juga kita harus menentukan berapa banyak kamera yang kita butuhkan untuk mendapatkan informasi selengkap-lengkapnyanya. Misalnya untuk mendapatkan informasi dalam dunia 3D yang akurat mungkin kita akan membutuhkan 6 kamera yang dipakai untuk mengawasi obyek dari 6 posisi yang berbeda. Atau mungkin bila bendanya dianggap cukup simetris dan bila kita mengambil kurang dari 6 posisi saja dianggap sudah cukup untuk mewakili keseluruhan bentuk obyek maka itu pun dapat dilakukan untuk menghemat biaya.

Sedangkan untuk men-*decode* seluruh informasi gambar yang ada juga untuk memilih informasi mana yang sangat diperlukan atau pun sama sekali tidak diperlukan dan memang harus dibuang dipakai *Image Processing*. Pengertian sederhana dari *image processing* adalah manipulasi dan analisis suatu informasi gambar oleh komputer. Yang dimaksud dengan informasi gambar di sini adalah gambar visual dalam dua dimensi. Segala operasi untuk memperbaiki, menganalisa, atau mengubah suatu gambar disebut *image processing*. Konsep dasar dari *image processing* diambil dari kemampuan indera penglihatan manusia yang selanjutnya dihubungkan dengan kemampuan otak manusia. Dalam sejarahnya, *image processing* telah diaplikasikan dalam berbagai bentuk, dengan tingkat kesuksesan yang cukup besar. Seperti berbagai cabang ilmu lainnya, *image processing* menyangkut pula berbagai gabungan cabang-cabang ilmu. Seperti diantaranya optik, elektronik, matematika, fotografi, dan teknologi komputer.

Beberapa faktor menyebabkan perkembangan sistem *image processing* menjadi lebih berkembang lebih pesat saat ini. Salah satu yang utama adalah penurunan biaya peralatan komputer yang dibutuhkan. Peralatan *unit processing* dan *bulk storage* menjadi semakin murah dari tahun ke tahun. Faktor lain adalah peningkatan tersedianya peralatan untuk proses digital dan tampilan gambar.

Berbagai bidang telah banyak menggunakan aplikasi dari *image processing* baik di bidang komersial, industri, dan medik. Bahkan bidang militer telah menggunakan perkembangan dunia *digital image processing* ini.

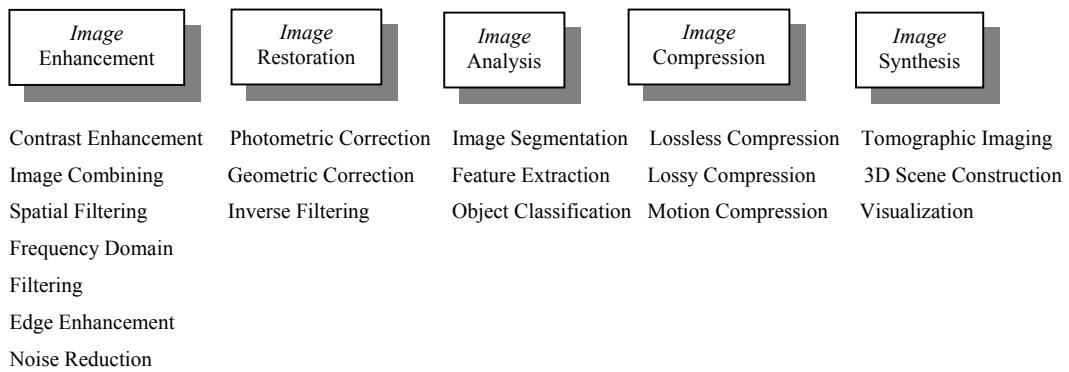
Pada umumnya, obyektif dari *image processing* adalah mentransformasikan atau menganalisis suatu gambar sehingga informasi baru tentang gambar dibuat lebih jelas. Ada banyak cara yang dapat diaplikasikan dalam suatu operasi *image processing*.

Hampir sebagian besar dalam bentuk optikal. Gambar optikal dikonversikan menjadi sinyal elektrik dengan menggunakan kamera video atau peralatan lain sejenisnya. Konversi ini merubah representasi gambar dari suatu cahaya optik menjadi sinyal elektrik kontinyu. Sinyal elektrik ini disebut sinyal *analog*. Lebih lanjut, gambar *analog* dapat di-*digital*-kan dan berubah menjadi data *digital*. Operasi pada sistem *image processing* dapat diaplikasikan pada suatu gambar dengan bentuk optikal, *analog*, atau *digital*

Digital image processing dapat dibagi menjadi lima kelas fundamental yaitu :

- *Image Enhancement*
- *Image Restoration*
- *Image Analysis*
- *Image Compression*
- *Image Synthesis*

Setiap kelas terdiri atas operasi-operasi yang spesifik, seperti yang terlihat pada gambar 2.1.



Gambar 2. 1 Lima Kelas Fundamental *Digital Image Processing*

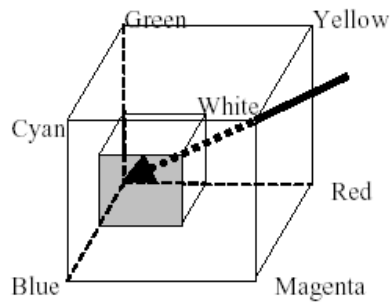
2.2. Sistem Warna

Warna dari suatu obyek merupakan suatu fungsi dari panjang gelombang yang tak dapat diserap dari cahaya yang dipantulkan dari obyek tersebut. Teori tentang cahaya ini dapat diperoleh dari teori tentang optik dari Isaac Newton, yang menyatakan bahwa sensasi dari warna dapat terpancar karena perbedaan panjang gelombang sehingga menghasilkan perbedaan efek visual yang dimunculkan oleh suatu obyek diakibatkan oleh panjang gelombang selektif yang dipantulkan dari transmisi cahaya tampak. Newton menyatakan bahwa terdapat 7 warna utama (merah, oranye, kuning, hijau, biru, dan violet) dan dipakai untuk menghitung warna dari spektrum panjang gelombang yang lainnya.

2.2.1. Sistem Warna RGB

Pada kenyataannya ditemukan bahwa semua warna dapat diperoleh dengan mengatur proporsi dari tiga komponen warna saja. Ketiga komponen warna ini terdiri dari merah, hijau, dan biru. Dan disebut dengan warna primer. Misalnya dengan mencampur warna merah dan hijau dengan perbandingan sama maka akan diperoleh warna kuning.

Sistem warna RGB adalah kombinasi aditif dari ketiga warna utama merah, hijau, dan biru. Ketiga warna ini dicampur bersamaan maka akan diperoleh warna putih atau warna hitam tergantung nilai dari RGB dari *pixel-pixel* tersebut. Sistem warna RGB merupakan sistem warna standar yang dipakai.



Gambar 2. 2 Sistem Warna RGB

2.2.2. Sistem Warna HSV

Sistem warna HSV (*Hue, Saturation, Value*) adalah warna pelengkap yang dihasilkan dari pengaturan warna utama (RGB). Sistem warna HSV merupakan hasil proyeksi sistem warna standar RGB ke diagonal utama yaitu dari putih ke hitam (lihat gambar 2.2). Ini akan menghasilkan *hexcone* pada gambar 2.3.

Brightness atau *value* menyatakan jumlah cahaya yang diterima oleh mata dari warna yang bersangkutan dari batas paling redup sampai paling terang. *Hue* menyatakan spektrum warna dominan dalam cahaya. *Saturation* mengindikasikan kejernihan spektrum dari warna dalam cahaya dan menyatakan bagaimana warna itu terkonsentrasi.

Setiap komponen pada sistem warna HSV dapat dijabarkan dalam suatu perhitungan matematis. Berikut perhitungan *hue*, *saturation*, dan *value*.

Hue :

$$\text{Redh} = \text{red} - \min(\text{red}, \text{green}, \text{blue})$$

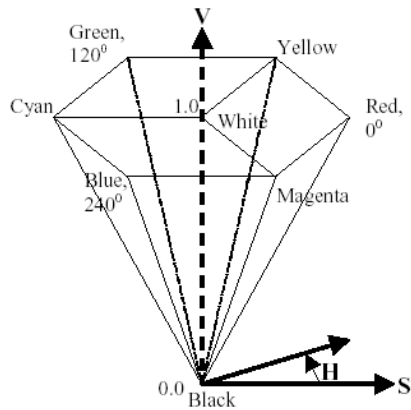
$$\text{Greenh} = \text{green} - \min(\text{red}, \text{green}, \text{blue})$$

$$\text{Blueh} = \text{blue} - \min(\text{red}, \text{green}, \text{blue})$$

$$\text{Saturation} = \frac{\max(\text{red}, \text{green}, \text{blue}) - \min(\text{red}, \text{green}, \text{blue})}{\max(\text{red}, \text{green}, \text{blue})}$$

$$\text{Value} = \max(\text{red}, \text{green}, \text{blue})$$

Setelah dilakukan perhitungan maka dilakukan proses pengambilan nilai dari *lookup* tabel. *Procedure* ini dipanggil melalui *Open source computer vision library*.



Gambar 2. 3 Sistem Warna HSV

2.3. *Open Source Computer Vision Library*

Open source computer vision library mulai dikembangkan tiga tahun yang lalu oleh *Visual Interactivity Group* di dalam *INTEL's Microprocessor Research Lab*. Proyek ini dibuat dengan tujuan untuk mendirikan sebuah komunitas *open source vision* dan menyediakan sebuah situs dimana usaha terdistribusi dari komunitas dapat dikonsolidasikan dan *performance*-nya dapat dioptimalkan. *Library* ini ditujukan untuk digunakan oleh peneliti dan pengembang *software* komersial. *Open source computer vision library* dibuat berdasarkan fungsi-fungsi dasar yang terdapat pada *INTEL Performance Library*. Keunggulan *library* ini adalah semua fungsi-fungsinya telah dioptimasi untuk prosesor INTEL sehingga dapat berjalan jauh lebih cepat.

Open source computer vision library Committee terdiri dari beberapa orang antara lain Dr. Gary Bradski, Prof. Trevor Darrell, Prof. Irfan Essa, Prof. Jitendra Malik, Prof. Pietro Perona, Prof. Stan Sclaroff, dan Prof. Carlo Tomasi. Berikut adalah beberapa area dari fungsi umum yang dapat didukung oleh *Open source computer vision library Committee* :

- *Geometric Methods*
- *Recognition*
- *Measures*
- *Segmentation*

- *Utilities*
- *Features*
- *Image Pyramids*
- *Camera Calibration*
- *Tracking*
- *Fitting*
- *Matriks*
- *Image Processing*

Open source computer vision library dibuat berdasarkan fungsi-fungsi dasar dan library dari INTEL[®] *Image Processing Library*. *Library* ini menyediakan sekumpulan fungsi-fungsi C yang mengimplementasikan fungsi-fungsi *image processing* pada *processor* berarsitektur INTEL

2.4. Intel[®] Image Processing Library

Dalam INTEL[®] *Image Processing Library* tersedia sekumpulan fungsi-fungsi *low-level* yang digunakan untuk memanipulasi suatu gambar atau *image* dalam standar DLL (*Dynamic Link Library*) dan format *library* yang *static*. Fungsi-fungsi ini telah dioptimasi untuk *processor* dengan arsitektur INTEL[®], dan sangat efektif karena menggunakan keunggulan dari teknologi MMX[™] (dipergunakan untuk multimedia, grafik), *Streaming SIMD Extensions* (SSE) untuk operasi aritmatik dan SSE-2. Pada saat ini versi-versinya telah dikembangkan untuk *processor* INTEL486[™] kompatibel, *processor* Pentium[®], *processor* Pentium Pro, *processor* Pentium dengan teknologi MMX, hingga *processor* Pentium 4. Untuk setiap masing-masing *processor* telah disediakan DDL yang berbeda-beda.

Library ini terdiri dari fungsi-fungsi untuk melakukan:

- Proses *filtering*.
- Proses *thresholding*.
- Proses transformasi (FFT, DCT dan geometri)
- Operasi-operasi aritmatika
- Proses *morphology*.

INTEL[®] *Image Processing Library* ini menggunakan format gambar yang fleksibel, mendukung gambar dengan 1, 8 (delapan) dan 16 (enam belas) *bit* dan *pixel* bisa berupa *integer* atau *floating point 32-bit*. Pada setiap gambar dapat mempunyai sejumlah *channel* yang berbeda. Mendukung format gambar Windows DIB (*device independent bitmap*), dan konversi antara gambar berwarna dan gray-scale juga disediakan.

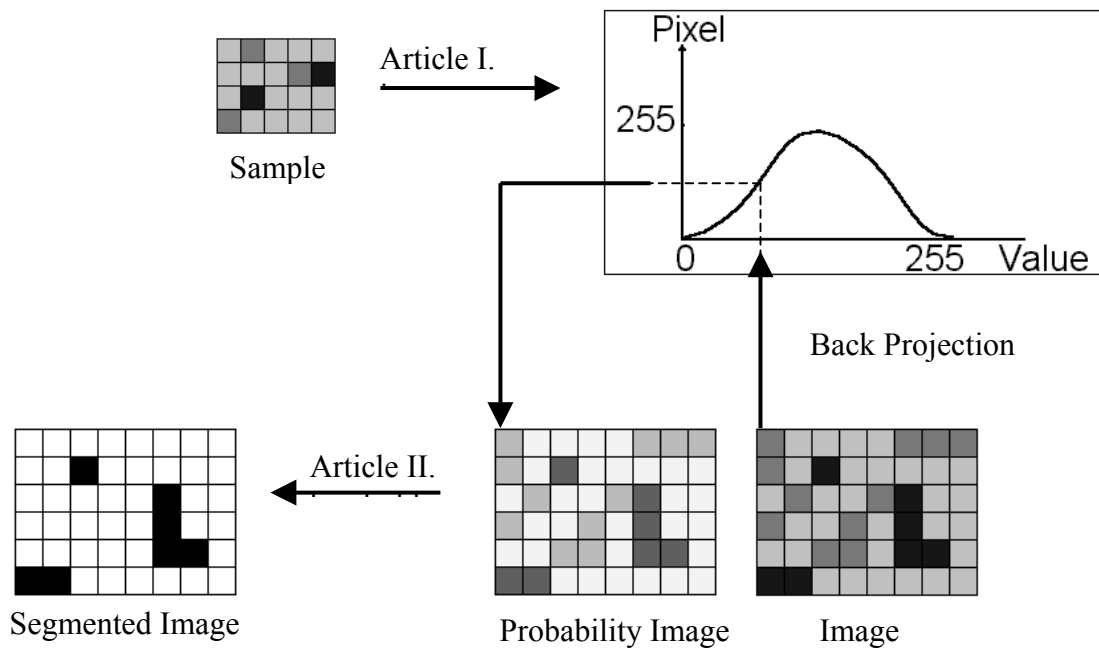
Versi terbaru dari INTEL[®] *Image Processing Library* adalah versi 2.5. Dalam penggunaan INTEL[®] *Image Processing Library* mempunyai satu kelemahan yaitu tidak *open-source*, sehingga *software programmer* agak terbatas bila ingin melakukan perubahan-perubahan pada fungsi *library* ini.

2.5. Segmentasi Berdasarkan Warna

Untuk dapat membedakan antara gambar tangan/benda yang akan ditrack dengan latar digunakan metode segmentasi berdasarkan warna, metode ini membagi *image* menjadi beberapa bagian berdasarkan kesamaan warna, dalam tugas akhir ini segmentasi digunakan untuk mengubah *color image* yang berasal dari kamera yang terdiri dari gambar tangan/obyek lain dan latar sekitarnya menjadi *binary image* yang hanya terdiri dari warna putih yang mewakili tangan/obyek lainnya dan warna hitam yang mewakili latar.

Untuk memperoleh *binary image* tersebut maka pertama-tama sampel tangan/obyek lain dihitung histogramnya, kemudian gambar hasil dari kamera di-*back project* terhadap histogram sampel tersebut, sehingga hasil yang diperoleh berupa *grayscale image* yang mana bagian yang paling terang mewakili warna yang paling mirip dengan sampel yang digunakan, dalam hal ini warna sampel yang digunakan adalah warna kulit tangan/obyek lainnya sehingga pada *grayscale image* bagian yang paling terang akan mewakili latar.

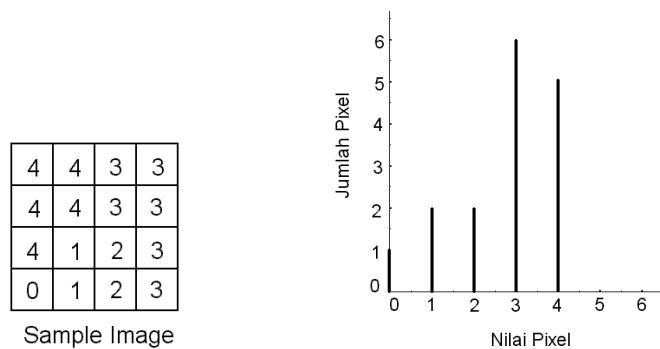
Grayscale image tersebut kemudian di-*threshold* untuk membuatnya menjadi *binary image* yang hanya terdiri dari warna putih dan hitam (warna putih mewakili kulit tangan/obyek lainnya dan warna hitam mewakili di latar. Gambar proses segmentasi ditunjukkan pada gambar 2.4.



Gambar 2. 4 Blok Diagram Segmentasi Berdasarkan Warna

2.5.1. Histogram

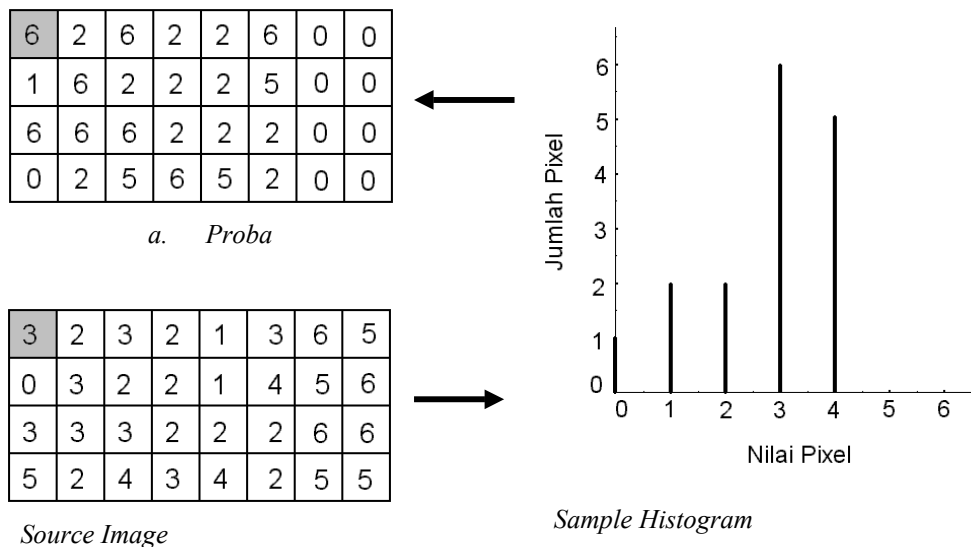
Histogram merupakan merupakan suatu *tool* yang digunakan untuk melihat komposisi dari suatu *image*. Histogram memberikan informasi berupa kontras dan distribusi intensitas dari suatu *image*. Contoh suatu histogram ditunjukkan gambar 2.5.



Gambar 2. 5 Histogram Dari Suatu Image

2.5.2. Back Projection

Back Projection merupakan suatu teknik untuk mencari bagian dari suatu *image* yang mirip dengan sampel, dengan menggunakan histogram dari sampel tersebut. *Pixel-pixel* dari *source image* di-set dengan nilai antara 0-255 sesuai dengan tingkat kemiripannya terhadap sampel, semakin mirip dengan sampel nilai yang di-set semakin tinggi. Blok diagram dari back projection dapat dilihat pada gambar 2.6.



Gambar 2. 6 Blok Diagram Back Projection

2.5.3. ROI

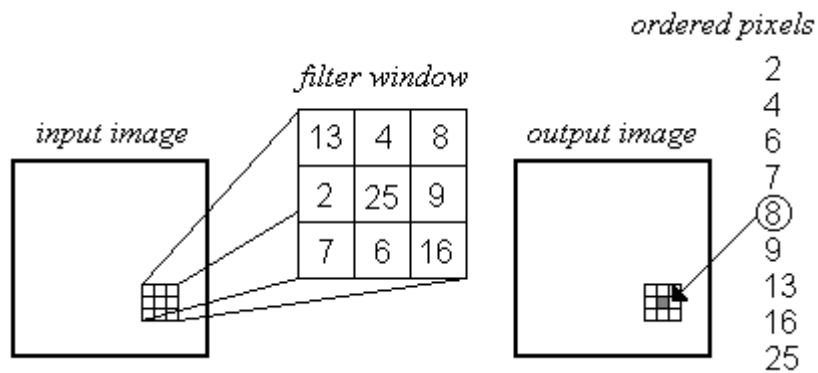
Yang dimaksud dengan ROI (Region Of Interest) disini adalah suatu daerah dalam image yang disorot atau dipilih untuk menjadi pusat aktivitas. Dengan adanya ROI ini maka bagian image yang lainnya tidak akan berubah oleh suatu aktivitas. Jadi tujuan dari ROI ini adalah untuk menghindari kerusakan dibagian lain dari image oleh adanya suatu aktivitas yang hanya ditujukan pada suatu bagian tertentu saja dari image.

2.6. Image Enhancement

Untuk memperbaiki kualitas *suatu image*, dapat dilakukan dengan berbagai cara diantaranya adalah dengan *filtering* dan *morphology*.

2.6.1. Median Filter

Median filter biasanya digunakan untuk menghilangkan *impluse noise* (*pixel-pixel* yang memiliki intensitas yang sangat berbeda terhadap *pixel-pixel* disekitarnya, misalnya 0 atau 255). Pada *median filter*, *pixel-pixel* pada suatu *image* yang tercakup dalam suatu *kernel* (*window*) diurutkan sesuai nilainya dan dicari nilai tengahnya (*median*), nilai tersebut digunakan untuk men-*set pixel* pada *output image*. Gambar 2.7 memperlihatkan Cara kerja dari *median filter*.



Gambar 2. 7 Cara Kerja Median Filter

2.6.2. Sharpen Filter

Sharpen filter berfungsi untuk mempertajam suatu *image* dengan menonjolkan detail (bagian *high frequency*) dari suatu *image*, *sharpen filter* juga meningkatkan kontras dari suatu *image*. *Sharpen filter* menggunakan kernel dengan koefisien positif ditengahnya dan negatif di sekelilingnya. Kernel sharpen filter yang digunakan:

$$(1/8) * \begin{pmatrix} -1 & -1 & -1 \\ -1 & 16 & -1 \\ -1 & -1 & -1 \end{pmatrix}$$

2.6.3. Morphology

Proses *morphology* yang terdiri dari dua macam operasi, yaitu: erosi dan dilasi. Erosi merupakan suatu proses menipiskan tepi dari suatu obyek sedangkan Dilasi merupakan suatu proses menebalkan tepi dari suatu obyek, kedua operasi ini jika dilakukan secara bergantian beberapa kali dapat menghilangkan bintik-bintik kecil pada suatu *image*.

2.7. Frame Proses

- *Frame process* merupakan operasi yang melibatkan dua atau lebih *image* untuk menghasilkan *image* baru, beberapa macam *frame process* adalah:
- *Add*, proses ini menghasilkan suatu *image* baru yang merupakan hasil penjumlahan dari *pixel-pixel* bersesuaian pada *input image*.
- *Subtract*, proses ini menghasilkan suatu *image* baru yang merupakan hasil pengurangan dari *pixel-pixel* bersesuaian pada *input image*.
- *AND/OR*, proses ini menghasilkan suatu *image* baru yang merupakan hasil operasi AND (*output*=1 bila kedua *input*=1, selain itu maka *output*=0) atau OR (*output*=1 bila salah satu *input*=1, jika kedua *input*=0 maka *output*=0) dari *pixel-pixel* bersesuaian pada *input image*.
- *Average*, proses ini menghasilkan suatu *image* baru yang merupakan hasil perhitungan nilai rata-rata dari *pixel-pixel* bersesuaian pada *input image*.

2.8. Tracking Posisi Dengan Mencari Titik Berat

Pada tugas akhir kali ini *tracking* posisi amat diperlukan untuk menentukan koordinat yang tepat untuk cursor mouse berdasarkan posisi tangan/obyek lain yang di-*tracking*. Untuk itu dilakukan dengan cara mencari titik berat dari benda yang di-*tracking* yang dalam hal ini adalah tangan/obyek lainnya. Untuk mencari titik berat dilakukan dengan rumus berikut:

$$W_x = \frac{\sum_{x=0}^k x \times p_x}{jp}$$

$$W_y = \frac{\sum_{y=0}^b y \times p_y}{jp}$$

dimana W_x, W_y = koordinat x,y titik berat

x, y = koordinat x,y yang sedang dihitung

p_x, p_y = jumlah pixel pada x,y

jp = jumlah pixel pada image

Dari rumus itu maka akan diketahui posisi tangan/obyek lainnya pada gambar yang diambil dari kamera. Hasil posisi ini kemudian dikonversi dengan perbandingan resolusi antara layar dan kamera.