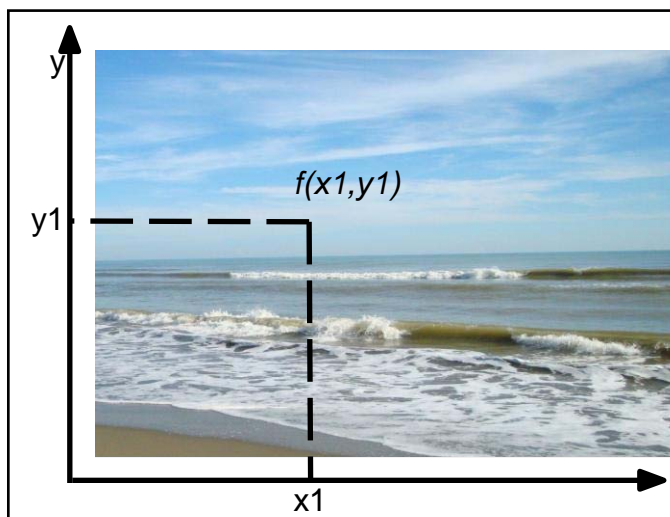


BAB 2

LANDASAN TEORI

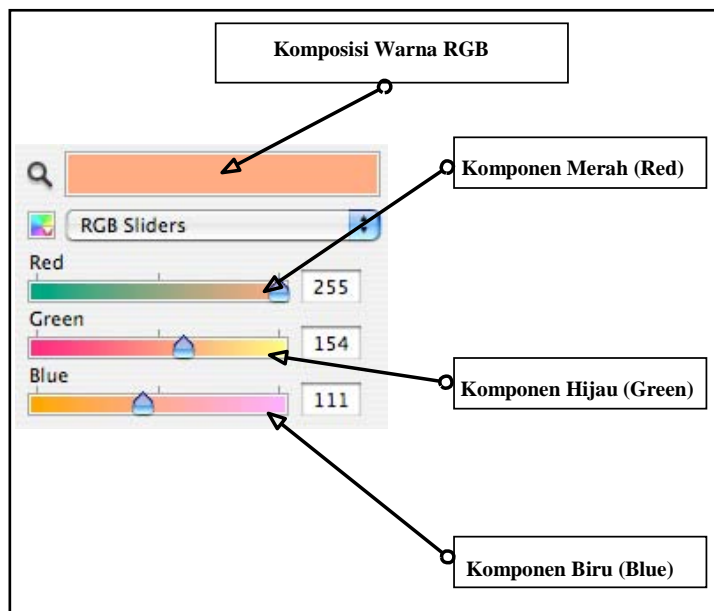
2.1 Pengolahan Citra Digital

Citra digital dapat didefinisikan sebagai fungsi dua variabel, $f(x,y)$, dimana x dan y adalah koordinat spasial dan nilai $f(x,y)$ adalah intensitas citra pada koordinat tersebut, hal tersebut diilustrasikan pada Gambar 2.1. 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). Komposisi warna RGB tersebut dapat dijelaskan pada Gambar 2.2.



Gambar 2.1 Citra Digital

Sebuah citra diubah ke bentuk digital agar dapat disimpan dalam memori komputer atau media lain. Proses mengubah citra ke bentuk digital bisa dilakukan dengan beberapa perangkat, misalnya *scanner*, kamera digital dan *handycam*. Ketika sebuah citra sudah diubah ke dalam bentuk digital (selanjutnya disebut citra digital), bermacam-macam proses pengolahan citra dapat diperlakukan terhadap citra tersebut.



Gambar 2.2 Komposisi Warna RGB

2.2 Operasi Pengolahan Citra Digital

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

1. Perbaikan kualitas citra (*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. pemberian warna semu (*pseudocoloring*)
- e. penapisan derau (*noise filtering*)

2. Pemugaran citra (*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 derau (*noise*)

3. Pemampatan citra (*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. Segmentasi citra (*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. Pengorakan citra (*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. Rekonstruksi citra (*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.

2.3 Ekstraksi Jalan

Ekstraksi adalah proses pemisahan suatu zat berdasarkan perbedaan kelarutannya (wikipedia.org). Ekstraksi jalan dari foto udara adalah proses pemisahan jalan dari foto udara dengan objek-objek lain yang terdapat dalam foto udara berdasarkan perbedaan warna atau nilai intensitasnya.

Proses ekstraksi unsur jalan pada citra inderaja dikembangkan berdasarkan beberapa karakteristik fotometrik dan geometrik yang diantaranya adalah [Zhao et al, 2002]:

1. Batas jalan nampak dengan jelas
2. Lebar jalan berubah perlahan;
3. Jalan membentuk garis lurus yang sejajar;
4. Arah jalan berubah perlahan;
5. Jalan nampak sebagai area homogen memanjang.

Menurut [Baumgartner, et. Al., 1999], karakteristik di atas adalah sudut pandang terhadap jalan dilihat dari level *real world*. Karakteristik tersebut kemudian dijadikan model jalan yang beberapa definisinya pada level citra adalah piksel yang memiliki karakteristik sebagai berikut:

1. Kelompok piksel homogen yang memiliki perbedaan cukup tajam dengan objek disekitarnya;
2. Perubahan jarak antara titik piksel pusat jalan dengan piksel tepi jalan kecil;
3. Garis tepi tidak bercabang;
4. Untaian piksel membentuk kurva dengan perubahan arah perlahan;
5. Untaian piksel jalan cenderung memanjang.

2.4 Threshold

Thresholding merupakan salah satu teknik segmentasi yang baik digunakan untuk citra dengan perbedaan nilai intensitas yang signifikan antara latar belakang dan objek utama (Katz, 2000). Dalam pelaksanaannya *Thresholding* membutuhkan suatu nilai yang digunakan sebagai nilai pembatas antara objek utama dengan latar belakang, dan nilai tersebut dinamakan dengan *threshold*.

Thresholding digunakan untuk mempartisi citra dengan mengatur nilai intensitas semua piksel yang lebih besar dari nilai *threshold* T sebagai latar depan dan yang lebih kecil dari nilai *threshold* T sebagai latar belakang. Biasanya pengaturan nilai *threshold* dilakukan berdasarkan histogram *grayscale* (Gonzales dan Woods, 2002; Fisher, dkk, 2003; Xiaoyi dan Mojon, 2003).

Dalam menentukan nilai *threshold* T secara adaptif, Fisher dkk (2003) menyarankan untuk menggunakan atribut-atribut statistik seperti yang terdapat pada persamaan (2.1), (2.2) dan (2.3). Atribut-atribut statistik tersebut diaplikasikan terhadap distribusi nilai intensitas.

Fungsi untuk distribusi nilai rata-rata,

(2.1)

nilai tengah,

(2.2)

atau rata-rata dari nilai minimum dan maksimum,

(2.3)

Untuk kasus yang dihadapi oleh Fisher, dkk (2003) hanya butuh satu nilai T . Sementara pada kasus segmentasi objek yang butuh dua nilai T atau lebih, maka disebut dengan *Multiple Threshold* (Gonzales dan Woods, 2002).

2.5 Grayscale

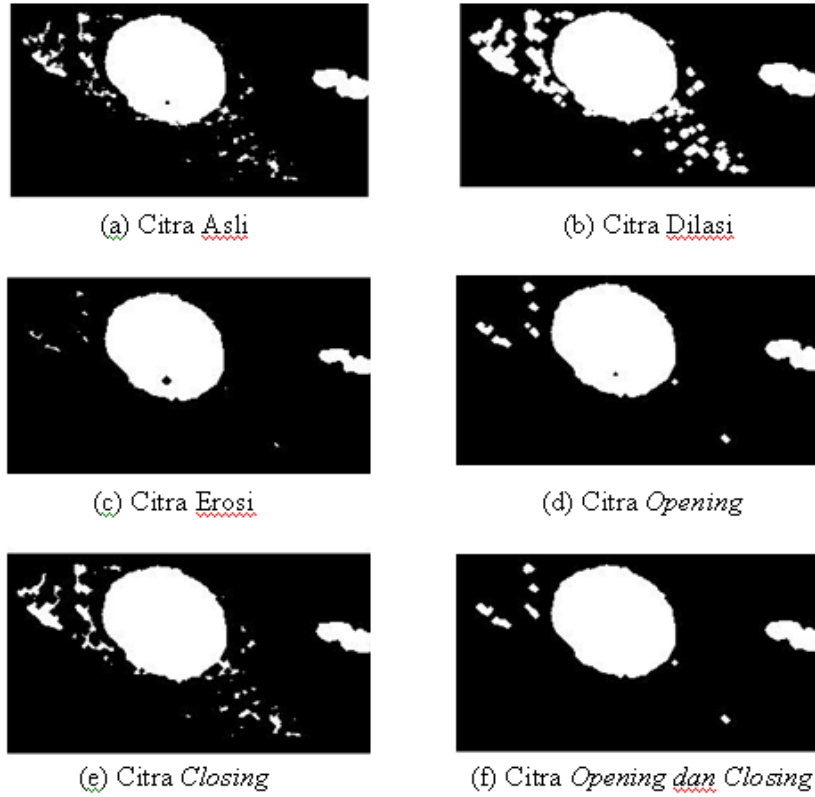
Citra *grayscale* adalah citra yang hanya menggunakan warna pada tingkatan warna abu-abu. Warna abu-abu adalah satu-satunya warna pada ruang RGB dengan komponen merah, hijau dan biru mempunyai intensitas yang sama. Pada citra beraras keabuan hanya perlu menyatakan nilai intensitas untuk tiap piksel sebagai nilai tunggal, sedangkan pada citra berwarna perlu tiga nilai intensitas untuk tiap pikselnya.

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

$$s = \frac{r + g + b}{3} \quad (2.4)$$

2.6 Morfology

Morfology mempunyai dua operator dasar, yaitu Dilasi (*dilation*) dan Erosi (*erosion*) yang biasa digunakan untuk mengekstrak komponen yang diinginkan dalam sebuah citra. Berdasarkan dua operator tersebut, dapat diturunkan dua operator lainnya yang berguna untuk menghaluskan batas subinterval komponen yang telah diekstrak, yaitu Pembukaan (*opening*) dan Penutupan (*closing*) (Gonzales dan Woods, 2002, Wang dan Jin, 2005). Contoh citra hasil penerapan mathematical morfology dapat dilihat pada Gambar 2.3.



Gambar 2.3 Contoh Citra Hasil Penerapan Mathematical Morfology

Sumber : Gonzales dan Woods, 2002

Dalam penggunaannya, Morfology selalu melibatkan sebuah citra dengan komponen I dan elemen penyusun E . Operator-operator Morfology tersebut adalah sebagai berikut (Gonzales dan Woods, 2002):

$$\text{Dilasi} \quad : \quad I \oplus E = \{z \mid (\hat{E})_z \cap I \neq \emptyset\} \quad (2.5)$$

$$\text{Erosi} \quad : \quad I \ominus E = \{z \mid (E)_z \subseteq I\} \quad (2.6)$$

$$\text{Opening} \quad : \quad I \circ E = (I \ominus E) \oplus E \quad (2.7)$$

$$\text{Closing} \quad : \quad I \bullet E = (I \oplus E) \ominus E \quad (2.8)$$

$(E)_z$ merupakan translasi dari komponen I terhadap titik z , sedangkan $(\hat{E})_z$ adalah refleksinya. Dilasi digunakan untuk memperbesar komponen yang diinginkan dengan cara menambahkan seluruh tepinya dengan elemen penyusun E . Erosi digunakan untuk mengikis komponen yang diinginkan dengan cara mengurangi seluruh tepinya dengan elemen penyusun E .

2.7 Deteksi Tepi

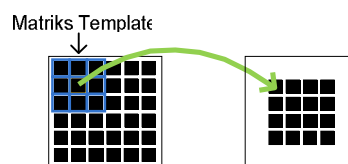
2.7.1 Deteksi Tepi Unsur pada Citra

Keberadaan tepi unsur ditandai dengan tingginya perubahan nilai piksel atau kontras [Nixon dan Aguado, 2002]. Tepi unsur pada citra dideteksi dengan operator deteksi (detektor) tepi diantaranya berupa matriks *template* berukuran tertentu, seperti pendeteksi tepi *Robert* (terdiri dari matriks detektor tepi diagonal M^- dan M^+) serta pendeteksi tepi *Prewitt* dan *Sobel* (keduanya terdiri dari matriks template vertikal, M_x dan horisontal, M_y) seperti ditunjukkan Gambar 2.4.

M^-	M^+	M_x	M_y	M_x	M_y																																												
<table><tr><td>1</td><td>0</td></tr><tr><td>0</td><td>-1</td></tr></table>	1	0	0	-1	<table><tr><td>0</td><td>1</td></tr><tr><td>-1</td><td>0</td></tr></table>	0	1	-1	0	<table><tr><td>1</td><td>0</td><td>-1</td></tr><tr><td>1</td><td>0</td><td>-1</td></tr><tr><td>1</td><td>0</td><td>-1</td></tr></table>	1	0	-1	1	0	-1	1	0	-1	<table><tr><td>1</td><td>1</td><td>1</td></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>-1</td><td>-1</td><td>-1</td></tr></table>	1	1	1	0	0	0	-1	-1	-1	<table><tr><td>1</td><td>0</td><td>-1</td></tr><tr><td>2</td><td>0</td><td>-2</td></tr><tr><td>1</td><td>0</td><td>-1</td></tr></table>	1	0	-1	2	0	-2	1	0	-1	<table><tr><td>1</td><td>2</td><td>1</td></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>-1</td><td>-2</td><td>-1</td></tr></table>	1	2	1	0	0	0	-1	-2	-1
1	0																																																
0	-1																																																
0	1																																																
-1	0																																																
1	0	-1																																															
1	0	-1																																															
1	0	-1																																															
1	1	1																																															
0	0	0																																															
-1	-1	-1																																															
1	0	-1																																															
2	0	-2																																															
1	0	-1																																															
1	2	1																																															
0	0	0																																															
-1	-2	-1																																															
Robert		Prewitt		Sobel																																													

Gambar 2.4 Matriks Template Pendeteksi Tepi
(Sumber : Nixon dan Aguado, 2002)

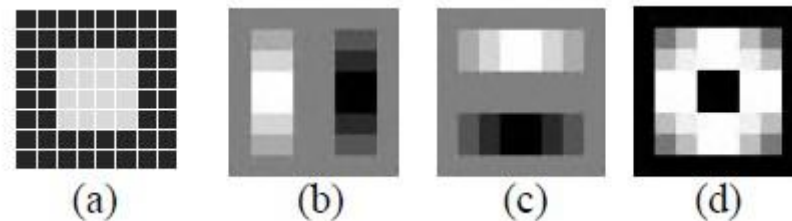
Dalam mendeteksi tepi, dilakukan perkalian elementer antara matriks template pada Gambar 2.4 di atas dengan kelompok piksel pada citra *input*. Pada hasil perkalian yang diperoleh, nilai setiap elemen dijumlahkan dan dijadikan nilai dari elemen matriks hasil konvolusi [Nixon dan Aguado, 2002]. Lihat Gambar 2.5.



Gambar 2.5 Gambar proses konvolusi
(Sumber : Nixon dan Aguado, 2002)

Deteksi tepi dengan metode *Prewitt* pada suatu citra biner berukuran 8x8 piksel (Gambar 2.6(a)), akan menghasilkan citra konvolusi vertikal (Gambar 2.6(b)) dan

citra konvolusi horizontal (Gambar 2.6(c)), citra hasil deteksi tepi keseluruhan seperti ditunjukkan Gambar 2.6(d).



Gambar 2.6 Contoh Hasil Deteksi Tepi *Prewitt*

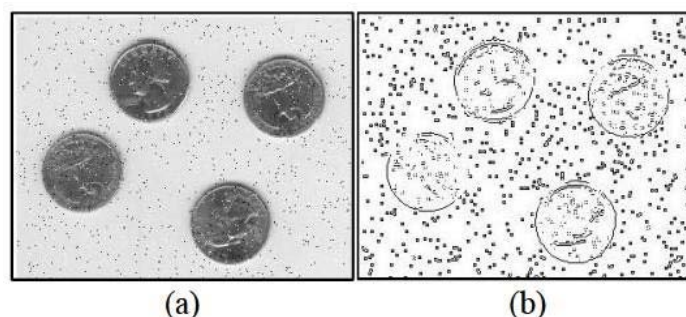
(Sumber : Nixon dan Aguado, 2002)

Keterangan Gambar:

- | | |
|---|---|
| (a) Citra Asli | (b) Hasil deteksi tepi horizontal (M_y) |
| (c) Hasil deteksi tepi vertikal (M_x) | (d) gabungan (c) dan (d) (M) |

2.7.2 Deteksi Tepi *Canny*

Deteksi tepi dengan operator-operator pada 2.7.1 masih memiliki kelemahan, diantaranya pada satu perubahan nilai piksel akan diperoleh banyak piksel tepi seperti ditunjukkan pada Gambar 2.6(d). Disamping itu, derau dan tekstur yang memiliki variasi nilai piksel yang cukup tinggi (tekstur kasar) pada suatu citra dapat juga terdeteksi sebagai tepi. Sebagai contoh, pada deteksi tepi dengan metode *Prewitt*, derau pada Gambar 2.7(a) terdeteksi sebagai piksel tepi. Lihat Gambar 2.7(b)



Gambar 2.7 Citra dengan derau dan tekstur dan hasil deteksi tepi *Prewitt*nya

Upaya perbaikan terhadap hasil deteksi tepi telah banyak diteliti orang dengan mengembangkan berbagai metode. Salah satu metode yang dikenal secara luas adalah deteksi tepi metode *Canny* yang memiliki kriteria sebagai berikut [Nixon dan Aguado, 2002]:

- Deteksi tepi optimal tanpa adanya kesalahan deteksi;
- Lokalisasi yang baik dengan jarak minimal antara tepi terdeteksi dengan posisi tepi sebenarnya;
- Respon tunggal terhadap tepi unsur.

Untuk mengakomodasi kriteria-kriteria tersebut di atas, [Canny,1986] menambahkan pula prosedur-prosedur perbaikan sebelum dan sesudah pendeteksian tepi (*pre dan post processing*) agar hasil deteksi tepi yang diperoleh menjadi lebih baik. *Pre* dan *post processing* yang dilakukan pada deteksi tepi metode *Canny* menyangkut [Nixon dan Aguado, 2002]:

- *Smoothing (preprocessing)*;
- *Non maximum suppression (post-processing)*;
- *Hysteresis thresholding (post-processing)*.

Proses *smoothing* dilakukan untuk menghilangkan derau dan menurunkan pengaruh tekstur pada citra sehingga diperoleh hasil deteksi yang lebih baik. Pada metode *Canny*, digunakan *filter gaussian* dalam bentuk matriks *template* yang merupakan bobot (weight) dalam perhitungan nilai rata-rata suatu kelompok piksel pada citra input yang diantaranya berukuran 3x3. Lihat Gambar 2.8.

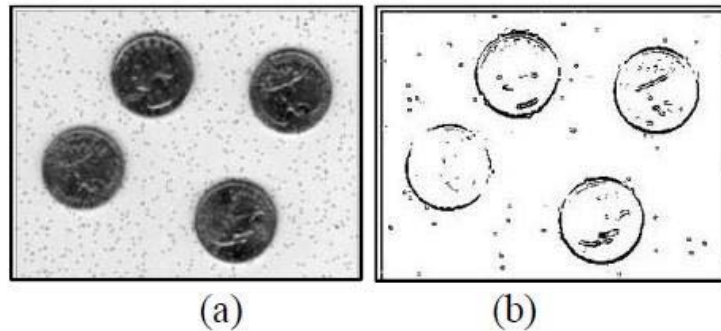
0.37	0.61	0.37
0.61	1	0.61
0.37	0.61	0.37

Gambar 2.8 Matriks template dari *filter Gaussian*

Nilai matriks *template* pada Gambar 2.8 tersebut di atas diperoleh dari persamaan (2.8) yang merupakan fungsi sebaran normal *gauss* g pada koordinat x , y dimana besarnya nilai elemen matriks *template* ditentukan oleh nilai σ^2 [Nixon dan aguado, 2002].

$$g(x, y) = e^{-\left(\frac{x^2+y^2}{2\sigma^2}\right)} \quad (2.9)$$

Sebagai contoh, penerapan *filtering Gaussian* pada Gambar 2.7(a) akan menghasilkan citra terfilter yang ditunjukkan oleh Gambar 2.9(a). Pada hasil deteksi tepi dengan metode *Prewitt* pada citra terfilter tersebut (Gambar 2.9(b)), nampak bahwa sebagian derau yang terdeteksi sebagai tepi berkurang.



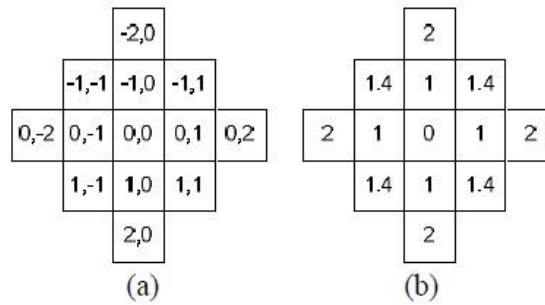
Gambar 2.9 Hasil deteksi tepi pada citra terfilter

Proses Non Maximum Suppression yang mirip dengan proses *thinning* (perampingan) dilakukan untuk menentukan piksel tepi dengan posisi paling mendekati lokasi terjadinya perubahan nilai piksel diantara banyaknya piksel tepi yang terdeteksi. Dimana pada umumnya, perubahan nilai piksel berada pada pusat kumpulan piksel tepi [Nixon dan Aguado, 2002]. Sebagai contoh nampak pada Gambar 2.6(d), sebagian diantara piksel-piksel tepi perlu dihilangkan karena perubahan nilai piksel pada citra input (Gambar 2.6(a)), hanya pada batas antara piksel hitam dan piksel putih saja.

Penentuan pusat kumpulan piksel tepi diantaranya dengan penghitungan jarak *Euclides* antara setiap piksel tepi $p(x,y)$ ke piksel bukan tepi $q(s,t)$ yang memiliki bentuk persamaan (2.9).

$$D = ([x - s]^2 + [y - t]^2)^{1/2} \quad (2.10)$$

Dimana piksel pada pusat suatu kumpulan piksel akan memiliki jarak ke piksel tepi terjauh. Disamping itu, jarak *Euclides* pada sumbu vertikal dan horisontal memiliki nilai yang sama dengan selisih koordinat pikselnya. Lihat Gambar 2.10

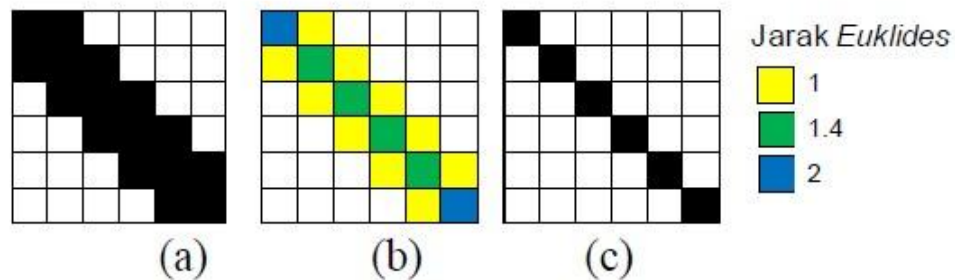


Gambar 2.10 Jarak *Euclides* pada satu piksel dengan piksel sekitarnya

Keterangan Gambar :

- (a) Jarak elementer (baris, kolom) dari pusat kelompok piksel
- (b) Jarak *Euclides* dari pusat kelompok piksel

Sebagai contoh, jarak *Euclides* antara piksel-piksel tepi (hitam) ke piksel bukan tepi (putih) pada suatu citra tepi biner (Gambar 2.11(a)) akan menghasilkan besaran jarak-jarak *Euclides* (Gambar 2.11(b)). Pada proses *thinning*, piksel dengan jarak *Euclides* lebih besar dari suatu nilai *threshold* $T = 1.4$ sehingga pada akhirnya tetap memiliki nilai 0 (hitam). Lihat Gambar 2.11(c).



Gambar 2.11 Contoh proses *thinning*

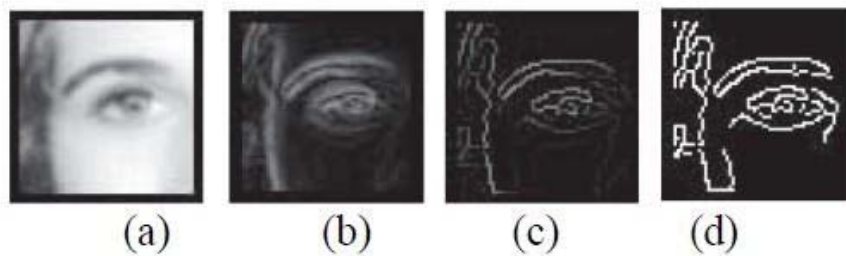
Keterangan :

- a. Hasil deteksi tepi
- b. Hasil penghitungan jarak *euclides*
- c. Hasil *thinning*

Berbeda dengan metode *thinning*, pada proses *Non Maximum suppression*, pengubahan menjadi citra biner tersebut menggunakan dua nilai *threshold* T_1 dan T_2 dimana $T_1 > T_2$ yang sering disebut juga *hysteresis thresholding* [Nixon dan Aguado,2002]. Setiap piksel tepi dengan nilai lebih besar dari T_1 dipertahankan

sebagai piksel tepi. Piksel tepi di sekitar piksel tepi yang nilainya lebih besar dari nilai *threshold* T_1 di atas juga dipertahankan sebagai piksel tepi jika nilainya masih lebih besar dari T_2 . Dari sudut pandang *hysteresis thresholding*, contoh pada Gambar 2.11(c) dapat diperoleh dengan nilai-nilai *threshold* $T_1 = 2$ dan $T_2 = 1.4$.

Hasil dari rangkaian proses deteksi tepi dengan metode *Canny* pada suatu adalah citra biner yang terdiri dari piksel-piksel tepi tunggal seperti pada Gambar 2.12.



Gambar 2.12 Contoh proses deteksi tepi Canny

Keterangan :

(a) Citra terfilter

(b) Hasil deteksi tepi Sobel

(c) Hasil *Non maximum suppression*

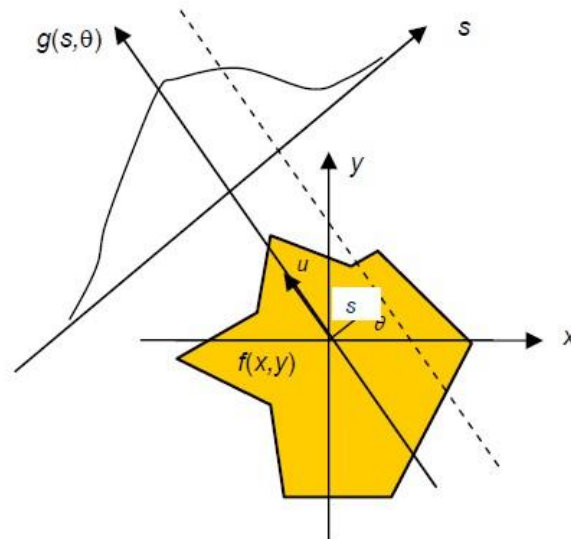
(d) Hasil *Hysteresis thresholding*

2.8 Transformasi

2.8.1 Transformasi Radon

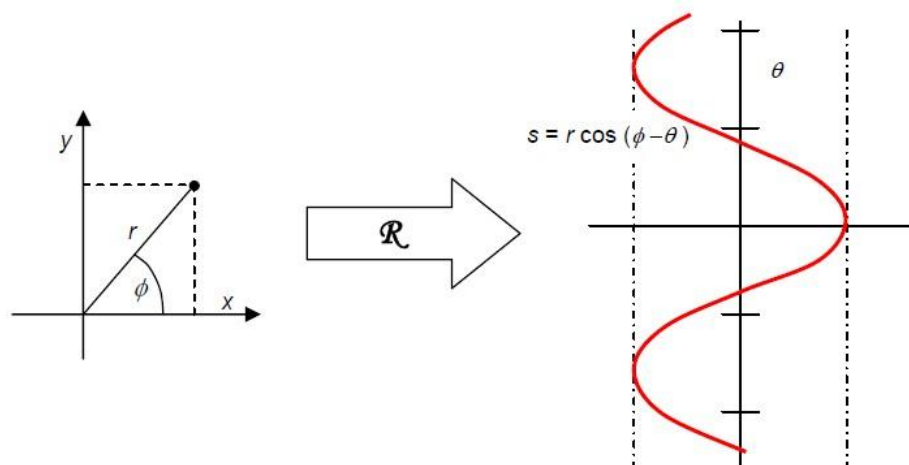
Transformasi Radon merepresentasikan citra sebagai kumpulan sinyal 1D hasil proyeksi citra tersebut pada berbagai sudut orientasi. Transformasi Radon dari suatu citra pada suatu sumbu proyeksi dinyatakan sebagai integral garis dari fungsi representasi citra $f(x,y)$ pada arah yang tegak lurus dengan sumbu proyeksinya. Hasil transformasi Radon dalam domain transformasi dinyatakan sebagai sinyal $g(s,\theta)$, dimana s merupakan jarak titik yang ditransformasikan terhadap titik pusat koordinat citra asal, dan θ merupakan sudut orientasi proyeksi. Lihat Gambar 2.13.

(2.11)



Gambar 2.13 Ilustrasi transformasi Radon dalam koordinat asal kartesian

Apabila citra dinyatakan dalam format koordinat asal polar $f(r, \theta)$, integral garis transformasi Radon sepanjang r menghasilkan kurva sinusoidal $s = r \cos(\phi - \theta)$. Bentuk sinusoidal ini menghasilkan istilah umum “sinogram” untuk kumpulan hasil transformasi Radon pada berbagai orientasi proyeksi. Lihat Gambar 2.14.



Gambar 2.14 Ilustrasi Transformasi Radon dalam koordinat asal polar

Dalam *image processing toolbox* Matlab, transformasi Radon dapat dilakukan dengan memanggil fungsi berikut:

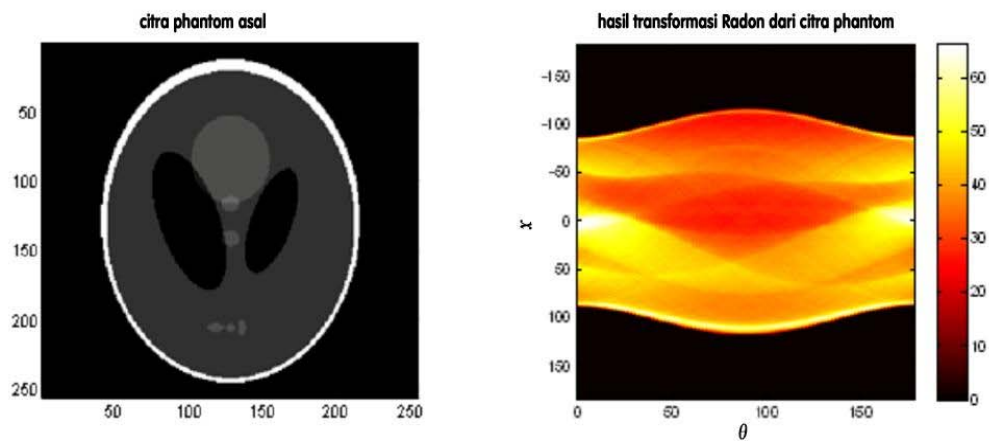
$$[R, xp] = \text{radon}(I, \theta);$$

dimana R merupakan hasil transformasi Radon terhadap citra asal I , pada jangkauan orientasi arah θ dan jangkauan lebar daerah proyeksi pada koordinat transformasi xp .

Berikut adalah *code* lengkap dari simulasi transformasi Radon terhadap gambar *phantom* irisan lintang kepala (Gambar 2.15) menggunakan Matlab:

```
P=phantom(256);
figure(1);imagesc(P);colormap(gray);title('citra phantom asal');

%Transformasi Radon citra phantom theta=0:2:180;
[R,xp]=radon(P,theta);
figure(2);imagesc(theta,xp,R);colormap(hot);
colorbar;xlabel('\theta');ylabel('x\prime'); title('hasil
transformasi Radon dari citra phantom');
```



Gambar 2.15 Hasil transformasi radon pada citra irisan kepala

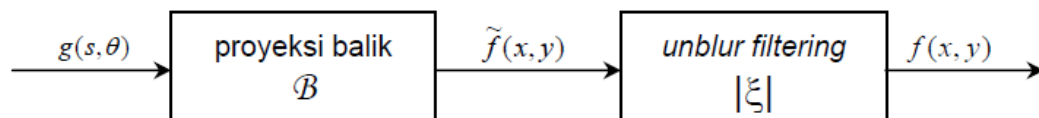
2.8.2 Transformasi Radon Invers

Invers transformasi Radon dari proyeksi $g(s, \theta) \equiv R(f)$, $-\infty < s < \infty$, $0 \leq \theta < \pi$ dirumuskan sebagai berikut:

$$f(x, y) = \frac{1}{2\pi^2} \int_0^\pi \int_{-\infty}^\infty - \frac{[(\partial g / \partial s)(s, \theta)]}{x \cos \theta + y \sin \theta - s} ds d\theta \quad (2.12)$$

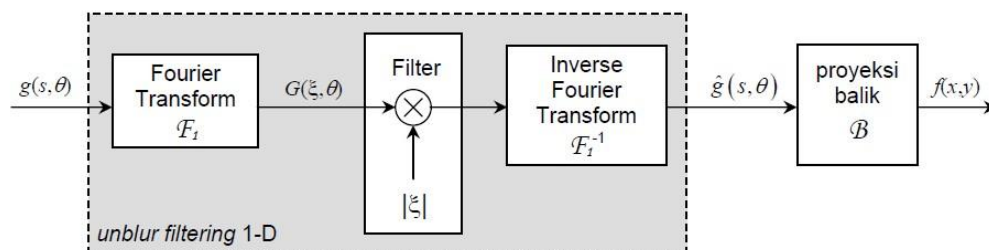
Transformasi Radon invers (Gambar 2.16) terdiri dari dua buah tahapan, yaitu:

- Proyeksi balik (*backprojection*)
- Unblur filtering*



Gambar 2.16 Skema transformasi Radon Invers

Agar menghasilkan rekonstruksi sempurna, proses *unblur filtering* membutuhkan filter *unblur* 2 dimensi. Akan tetapi pemfilteran 2 dimensi dengan kernel *unblur* merupakan operasi yang sangat *exhaustive* secara komputasional, terutama untuk resolusi citra yang besar. Strategi yang biasa digunakan dalam aplikasi praktis adalah melalui *unblur filtering* pada sinyal proyeksi 1 dimensi, dan dilanjutkan dengan operasi *backprojection*. Strategi ini diilustrasikan pada Gambar 2.17.

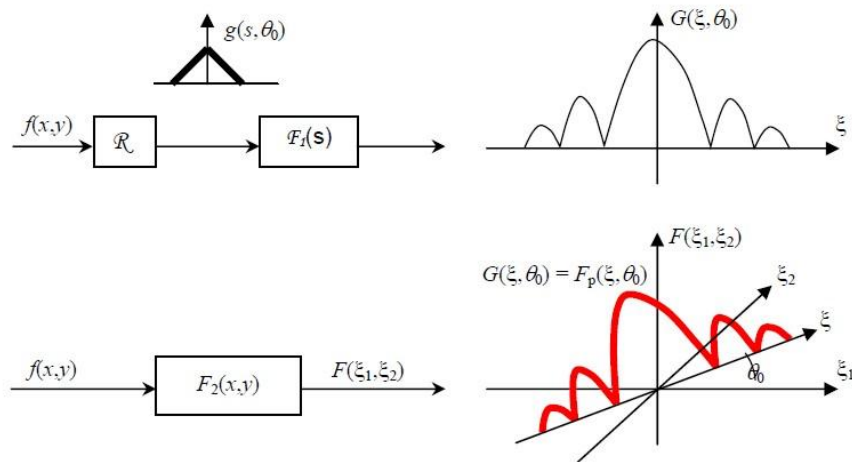


Gambar 2.17 Skema transformasi Radon invers, apabila *unblur filtering* dilakukan terhadap sinyal satu dimensi

Penukaran posisi blok *unblur filtering* dan blok proyeksi balik (antara Gambar 2.16 dan Gambar 2.17) diperbolehkan oleh adanya teorema (iris) – proyeksi. Dengan berpegang pada teorema ini, proses *unblur filtering* dapat dikerjakan pada domain frekuensi, sehingga akan mengurangi beban komputasi. Lihat Gambar 2.18

Teorema (iris)
proyeksi

Transformasi Fourier 1-dimensi dari proyeksi $g(s, \theta)$ yang dilakukan terhadap variabel s adalah sama dengan irisan sentral pada sudut θ dari transformasi Fourier 2-dimensi dari citra asal $f(x, y)$.



Gambar 2.18 Teorema (Irisan) – Proyeksi

2.8.2.1 Operasi *Unblur Filtering*

Mekanisme *unblur filtering* dapat dilakukan dengan menggunakan berbagai skema filter *unblur* yang dapat dipilih. Filter *unblur* yang umum dipergunakan adalah filter *Shepp-Logan* dengan properti sebagai berikut:

$$\text{Respons frekuensi} \quad : \quad H(\xi) = |\xi| \operatorname{sinc}(\xi d) \operatorname{rect}(\xi d) \quad (2.13)$$

$$\text{Respons impuls} \quad : \quad h(s) = 2(1 + \sin 2\pi \xi_0 s) / \pi^2 (d^2 - 4s^2) \quad (2.14)$$

$$\text{Respons impuls diskrit} \quad : \quad h(m) = 2 / (\pi^2 d (1 - 4m^2)) \quad (2.15)$$

2.8.2.2 Operasi Proyeksi Balik

Operasi *backprojection* dapat didefinisikan sebagai berikut:

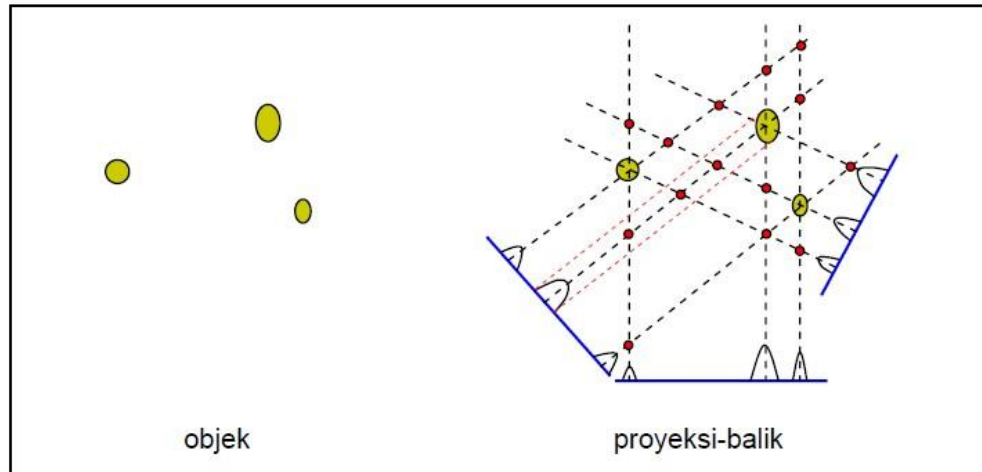
Koordinat cartesian :

$$b(x, y) \equiv \mathcal{B}(g) = \int_0^\pi g(x \cos \theta + y \sin \theta, \theta) d\theta \quad (2.16)$$

Koordinat polar :

$$b(x, y) = b_p(r, \phi) = \int_0^\pi g(r \cos(\theta - \phi), \theta) d\theta \quad (2.17)$$

Hasil operasi proyeksi balik menunjukkan jumlah berkas proyeksi yang melewati suatu titik (x,y) dalam citra, sebagaimana diilustrasikan dalam Gambar 2.19.



Gambar 2.19 Ilustrasi operasi *backprojektion*

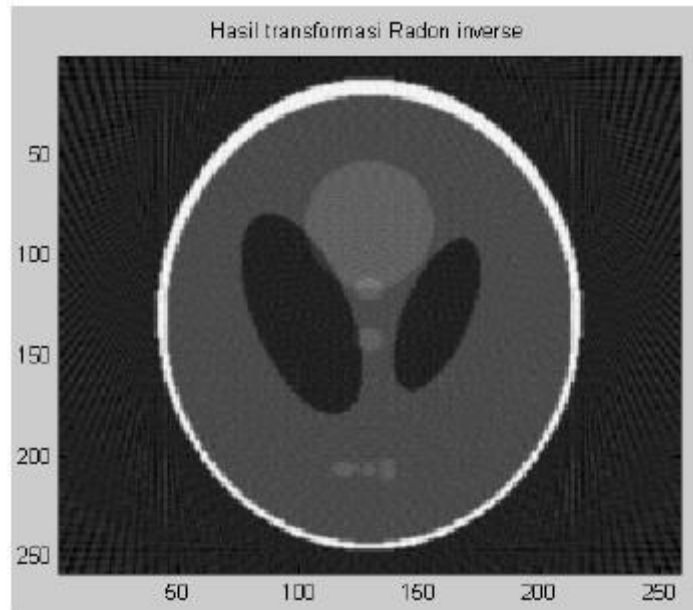
Hasil operasi *backprojection* dapat dinyatakan sebagai citra asal yang dikaburkan dengan fungsi *blurring* dengan *point spread function* berupa $(x^2 + y^2)^{-1/2}$.

$$\bar{f}(x, y) \equiv \mathcal{B}(g) = \mathcal{B}(\mathcal{R}(f)) = f(x, y) \otimes (x^2 + y^2)^{-1/2} = f_p(r, \phi) \otimes \frac{1}{|r|} \quad (2.18)$$

Dalam *code* Matlab, operasi transformasi Radon invers dilakukan dengan memanggil fungsi sebagai berikut: (ilustrasi *code* sebagai lanjutan dari *code* transformasi Radon sebelumnya).

```
%Inverse-Radon tranform  
I=iradon(R,2); figure(3);imagesc(I);colormap(gray);  
title('Hasil transformasi Radon inverse');
```

Fungsi ini sekaligus menjalankan operasi *unblurring* dan *backprojection*. Matlab menyediakan berbagai skema *unblurring*, yang dapat dipilih. Hasil transformasi Radon invers pada citra irisan kepala dapat dilihat pada Gambar 2.20.



Gambar 2.20 Hasil transformasi Radon invers (citra irisan kepala)