

BAB II

LANDASAN TEORI

2.1 Forensik Odontologi

Forensik odontologi adalah cabang ilmu kedokteran gigi yang mempelajari cara penanganan dan pemeriksaan benda bukti gigi serta cara evaluasi dan presentasi temuan gigi untuk kepentingan peradilan. Salah satu bukti yang diambil dari gigi dapat digunakan untuk pengidentifikasian seseorang yang memiliki gigi tersebut. Hal ini dilakukan dengan menggunakan data rekaman gigi atau foto gigi (Lukman, 2006). Seorang ahli forensik membutuhkan minimal satu gigi *molar* untuk dapat melakukan pengidentifikasian (Abdel-Mottaleb dkk, 2004).

Pengidentifikasian forensik terbagi dua yaitu (Lukman, 2006):

1. Pengidentifikasian *ante-mortem* (AM) adalah pengidentifikasian forensik yang dilakukan sebelum kematian seseorang.
2. Pengidentifikasian *post-mortem* (PM) adalah pengidentifikasian forensik yang dilakukan setelah kematian seseorang.

Beberapa alasan dapat dikemukakan mengapa gigi dapat dipakai sebagai sarana identifikasi adalah sebagai berikut (Lukman, 2006):

1. Gigi merupakan jaringan keras yang resisten terhadap pembusukan dan pengaruh lingkungan yang ekstrim
2. Bentuk gigi manusia di dunia ini tidak sama, karena berdasarkan penelitian bahwa gigi manusia kemungkinan sama satu banding dua miliar.
3. Gigi geligi tahan panas sampai suhu kira-kira 400°C.
4. Gigi geligi tahan terhadap asam keras.

2.2 *Dental Radiographs*

Dental radiographs (umumnya merujuk pada film *X-ray*) adalah gambar dari gigi, tulang, dan bagian lunak yang berada di sekitarnya untuk melindungi dan membantu mengidentifikasi masalah dengan gigi, mulut, dan rahang (Haring, 2000). *Dental radiographs* dapat dikategorikan menjadi *intraoral radiograph* dan *extraoral radiograph* (Brocklebank, 1997).

1. *Intraoral Radiograph*

Citra *intraoral radiograph* diperoleh dengan menempatkan film radiografis atau sensor di dalam mulut untuk menghasilkan sebuah *radiograph* (Brocklebank, 1997). Terdapat tiga tipe *intraoral radiograph* yaitu:

a. *Periapical Radiograph*

Periapical radiograph merupakan foto *rontgen* standar yang sering digunakan pada dalam bidang kedokteran gigi. *Periapical radiograph* dapat memperlihatkan gambaran gigi beserta tulang disekelilingnya secara keseluruhan.

b. *Sefalometrik Radiograph*

Foto *rontgen* ini jarang dipakai di bidang forensik karena memerlukan alat *rontgen* khusus yang disebut *cephalostat* dan jenazah yang diidentifikasi harus berupa kerangka.

c. *Bitewing Radiograph*

Bitewing radiograph adalah foto *rontgen* yang dapat memperlihatkan gambaran karies dan restorasi-restorasi pada gigi posterior.

2. *Extraoral Radiograph*

Citra *extraoral radiograph* diperoleh dengan menempatkan film radiografis atau sensor di luar mulut, pada sisi yang berseberangan dengan kepala dari sumber x-ray (Brocklebank, 1997).

Panoramic radiograph adalah salah satu foto *rontgen ekstraoral* yang telah digunakan secara umum di bidang kedokteran untuk mendapatkan gambaran utuh dari keseluruhan maksilofasial (daerah wajah, rahang, leher, hidung dan gigi) (Dannewitz dkk, 2002). Foto panoramik sangat

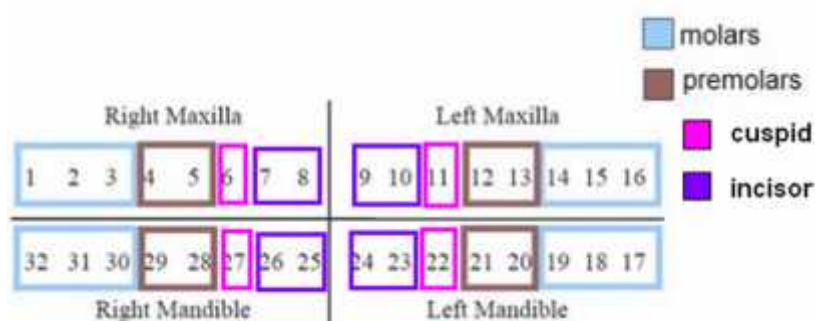
populer di bidang kedokteran karena teknik yang sederhana. Gambaran *panoramic radiograph* mencakup seluruh gigi dan rahang dengan tingkat radiasi yang rendah (Whites, 2007).



Gambar 2.1 Citra *panoramic radiograph*

2.3 Susunan Gigi

Orang dewasa memiliki tiga puluh dua gigi. Tiap rahang memiliki enam belas gigi. Terdapat dua rahang yaitu rahang atas (*maxilla*) dan rahang bawah (*mandible*), Dua rahang tersebut dibagi dalam empat kuadran yang sama. Dimana satu kuadran terdiri dari delapan gigi yaitu dua gigi seri (*incisor*), satu gigi taring (*cuspid*), dua gigi geraham depan (*premolar*), tiga gigi geraham belakang (*molar*) (Mahoor, 2005 dikutip dari Yofiyanto, 2010).



Gambar 2.2 Susunan gigi orang dewasa
(Mahoor, 2005 dikutip dari Yofiyanto, 2010)

2.4 Citra

Citra (*image*) adalah gambar pada bidang dua dimensi. Citra terbagi dua yaitu citra kontinu dan citra diskrit. Citra kontinu dihasilkan dari sistem optik

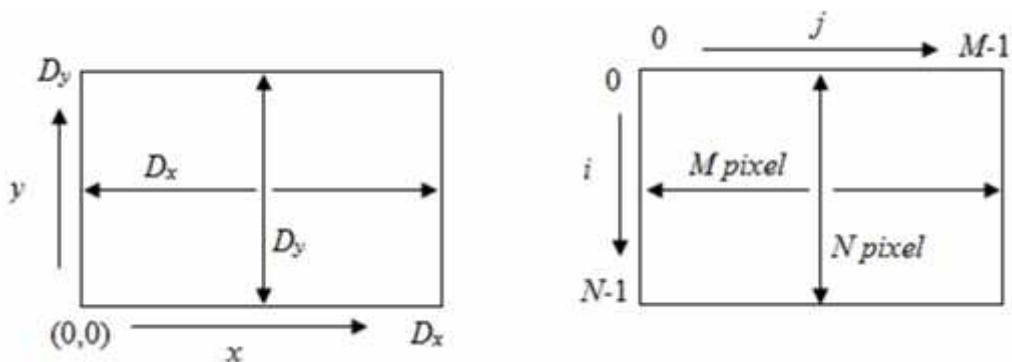
yang menerima sinyal analog, misalnya mata manusia dan kamera analog. Citra diskrit dihasilkan melalui proses digitalisasi terhadap citra kontinu. Beberapa sistem optik yang menghasilkan citra diskrit, misalnya kamera *digital* dan *scanner*. Citra diskrit disebut dengan citra digital (Munir, 2004).

Citra *digital* merupakan citra yang dapat diolah oleh komputer *digital* karena citra *digital* direpresentasikan secara numerik dengan nilai-nilai diskrit. Citra *digital* yang berukuran $N \times M$ dapat direpresentasikan dengan matriks berukuran N baris dan M kolom sebagai berikut (Munir, 2004):

$$f(x,y) = \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0,M) \\ f(1,0) & f(1,1) & \dots & f(1,M) \\ \vdots & \vdots & \vdots & \vdots \\ f(N-1,0) & f(N-1,1) & \dots & f(N-1,M-1) \end{bmatrix} \quad (2.1)$$

Masing-masing elemen pada citra digital (elemen matriks) disebut dengan *pixel*. Citra yang berukuran $N \times M$ mempunyai NM *pixel*. Contoh, sebuah citra berukuran 256×256 *pixel* direpresentasikan secara numerik dengan matriks sehingga 256 baris (dimulai dari 0 sampai 255) dan 256 kolom (dimulai dari 0 sampai 255) (Munir, 2004).

Terdapat perbedaan antara koordinat gambar dan koordinat matriks (hasil digitalisasi). Titik asal (0,0) pada gambar dan elemen (0,0) pada matriks tidak sama. Koordinat x dan y pada gambar dimulai dari sudut kiri bawah, sedangkan penomoran piksel pada matriks dimulai dari sudut kiri atas, dapat dilihat pada gambar 2.3 (Galbiati, 1990 dikutip dari Munir, 2004):



(a) (b)

**Gambar 2.3 (a) Koordinat gambar dan (b) Koordinat matriks
(Galbiati, 1990 dikutip dari Munir, 2004)**

2.4.1 Jenis Citra

Pixel memiliki nilai jangkauan yang berbeda-beda tergantung dari jenis warnanya. Secara umum, nilai jangkauan *pixel* yaitu 0-255. Berikut ini adalah jenis citra berdasarkan nilai *pixel* yaitu:

1. Citra Warna

Pada citra warna, setiap titik warna yang spesifik merupakan kombinasi dari tiga warna dasar, yaitu: merah, hijau dan biru. Citra ini sering disebut citra RGB (*Red, Green, Blue*). Setiap warna dasar memiliki intensitas sendiri dengan nilai maksimum 255 (Putra, 2010).

2. Citra *grayscale*

Citra *grayscale* adalah citra digital yang memiliki satu nilai kanal pada setiap pikselnya. Nilai kanal digunakan untuk menunjukkan tingkat intensitas. Warna yang dimiliki pada citra *grayscale* adalah warna hitam, keabuan dan putih. Citra *grayscale* memiliki kedalaman warna 8 bit (256 kombinasi warna keabuan). Citra *grayscale* digunakan untuk menyederhanakan model citra. (Putra, 2010).

3. Citra Biner

Citra biner adalah citra yang mempunyai dua nilai derajat keabuan yaitu hitam dan putih. *Pixel-pixel* objek bernilai 1 dan *pixel-pixel* latar belakang bernilai 0. Pada saat citra ditampilkan, 0 adalah putih dan 1 adalah hitam. Citra biner membutuhkan 1 bit untuk merepresentasikan nilai *pixel* (Munir, 2004).

2.4.2 Format Citra

Format citra digunakan untuk menyimpan citra dalam sebuah *file*. Setiap format memiliki karakteristik masing-masing. Berikut ini adalah penjelasan beberapa format citra yaitu (Putra, 2010):

1. Format Bitmap

Bitmap adalah format penyimpanan standar tanpa pemampatan (kompresi) yang umum digunakan untuk menyimpan citra biner hingga citra warna

(Putra, 2010). Format BMP disebut dengan bitmap atau format DIB (*Device Independent Bitmap*). Format BMP digunakan terutama pada sistem operasi *Microsoft Windows* atau OS/2 (Munir, 2004).

Format BMP memiliki ukuran berkas yang relatif lebih besar daripada format JPG maupun GIF. Namun format BMP mempunyai kelebihan dari segi kualitas gambar. Citra dalam format .bmp lebih bagus daripada citra dalam format yang lainnya, karena citra pada format BMP tidak dimampatkan (dikompresi) sehingga tidak ada informasi yang hilang (Munir, 2004).

2. Format-format standar kompresi pada citra digital

a. Format JPEG (*Join Photographic Expert Group*)

Format JPEG mampu mengkompresi objek dengan tingkat kualitas sesuai dengan pilihan yang disediakan. Format JPEG sering dimanfaatkan untuk menyimpan gambar yang akan digunakan untuk keperluan halaman web, multimedia dan lainnya. Format JPEG dapat menyimpan gambar dalam mode warna *grayscale*, CMYK dan RGB.

b. Format GIF (*Graphic Interchange Format*)

Format GIF merupakan format citra yang umum digunakan pada aplikasi web yang mendukung kedalaman warna sampai palet 8 bit. Kualitas yang rendah menyebabkan format ini tidak populer di kalangan peneliti pengolahan citra.

c. Format PNG (*Portable Network Graphics*)

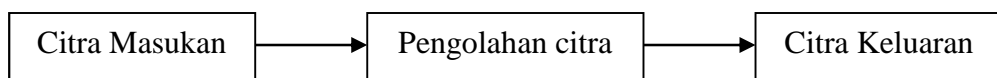
Format PNG merupakan format penyimpanan citra terkompresi yang digunakan pada citra *grayscale*, citra dengan palet warna dan citra berwarna. Format PNG dapat menyimpan informasi sebesar 1 hingga 16 bit per kanal.

d. Format TIF (*Tagged Image Format*)

Format TIF merupakan format penyimpanan citra yang dapat digunakan untuk menyimpan citra yang tidak terkompresi dan terkompresi. Format TIF mampu menyimpan gambar dengan kualitas hingga 32 bit.

2.5 Pengolahan Citra

Pengolahan citra adalah pemrosesan citra yang bertujuan untuk memperbaiki kualitas citra agar mudah diinterpretasi oleh manusia atau mesin (komputer). Operasi pengolahan citra dapat mentransformasikan citra menjadi citra lain, seperti: pada gambar 2.4. Citra masukan diolah menggunakan operasi-operasi pada pengolahan citra, sehingga menghasilkan citra keluaran yang memiliki kualitas yang baik (Munir, 2004).



Gambar 2.4 Operasi pengolahan citra (Munir, 2004)

Operasi-operasi yang dilakukan di dalam pengolahan citra banyak ragamnya. Secara umum, operasi pengolahan citra dapat diklasifikasikan dalam beberapa jenis yaitu (Munir,2004):

1. Perbaikan kualitas citra (*image enhancement*)
2. Pemugaran citra (*image restoration*)
3. Pemampatan citra (*image compression*)
4. Segmentasi citra (*image segmentation*)
5. Pengorakan citra (*image analysis*)
6. Rekonstruksi citra (*image reconstruction*)

Umumnya, operasi-operasi pada pengolahan citra diterapkan pada citra bila (Munir, 2004):

1. Perbaikan citra perlu dilakukan untuk meningkatkan kualitas penampakan dan menonjolkan beberapa aspek informasi yang terkandung dalam citra.
2. Elemen di dalam citra perlu dikelompokkan, dicocok atau diukur.
3. Sebagian citra perlu digabung dengan citra lain.

2.5.1 *Preprocessing*

Preprocessing merupakan tingkat awal pengolahan citra. *Preprocessing* bertujuan untuk mengubah intensitas *pixel* gambar agar mudah digunakan pada proses selanjutnya. Tahapan *preprocessing* yang dilakukan yaitu:

1. *Grayscale*

Grayscale disebut juga dengan operasi skala keabuan karena pada umumnya warna yang digunakan adalah warna keabuan (Sigit dkk, 2006). Proses mengubah citra warna yang memiliki nilai matrik masing-masing r, g, dan b menjadi citra *grayscale* dapat dilakukan dengan persamaan 2.2

$$grayscale = 0.2989 * R + 0.5870 * G + 0.1140 * B \quad (2.2)$$

R merupakan nilai warna merah (*red*) pada citra

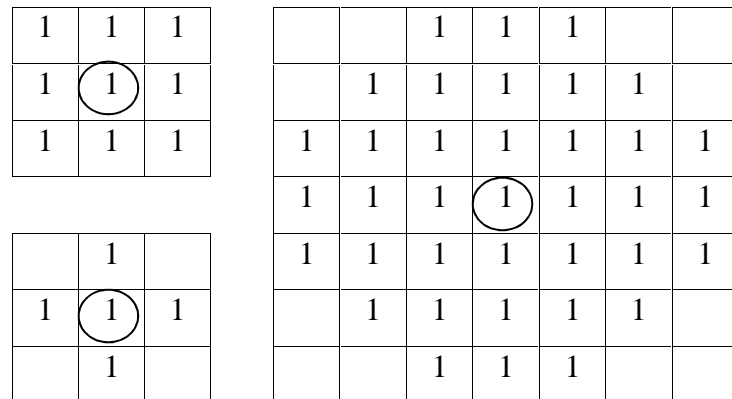
G merupakan nilai warna hijau (*green*) pada citra

B merupakan nilai warna biru (*blue*) pada citra

2. *Morfologi*

Kata morfologi secara sederhana dapat diartikan sebagai bentuk dan struktur objek (Putra, 2010). Operasi morfologi merupakan operasi untuk mengubah struktur objek yang terkandung dalam citra.

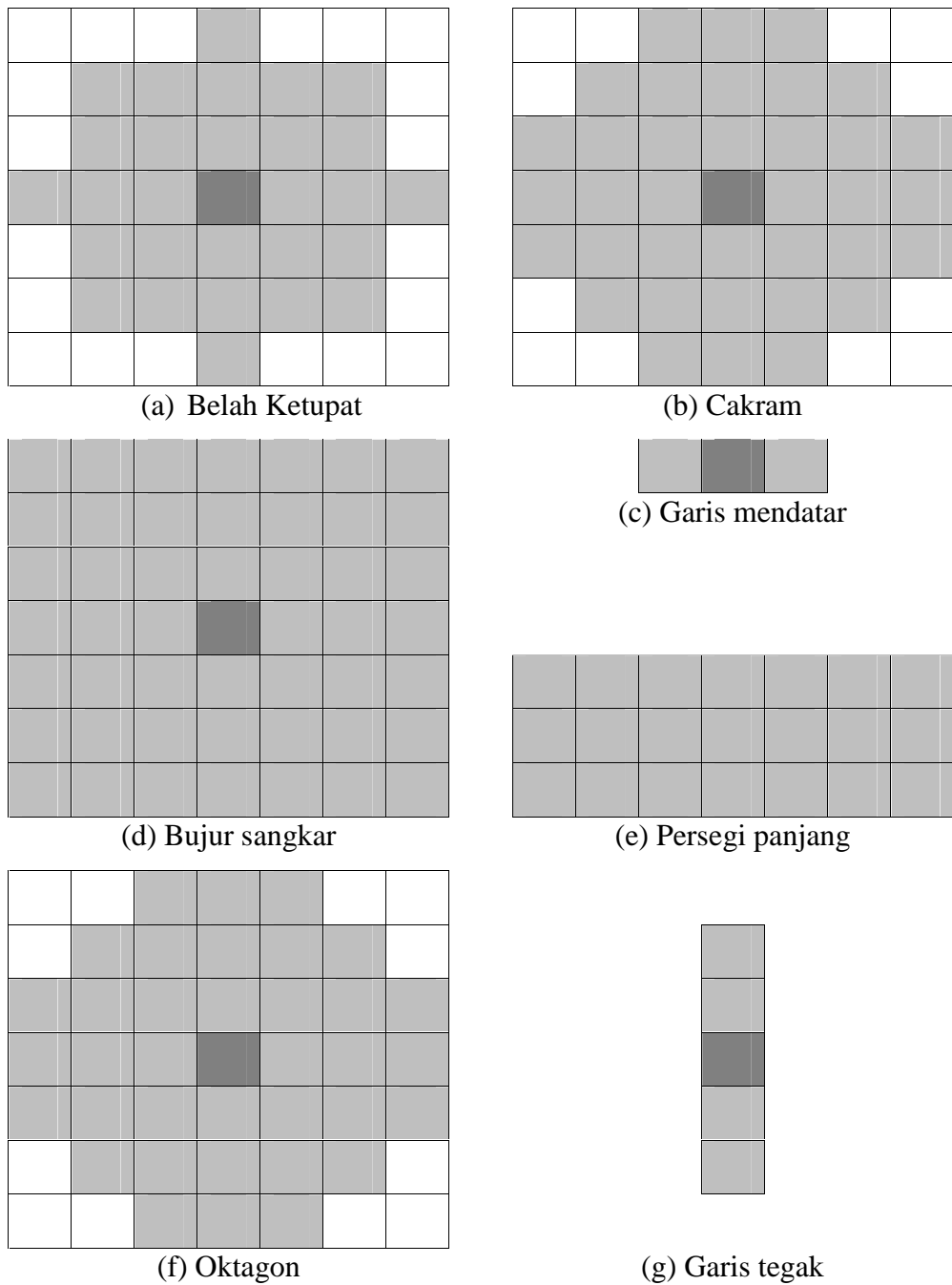
Operasi morfologi menggunakan dua input yaitu citra dan kernel. Pada morfologi, istilah kernel disebut dengan *structuring elements* (struktur elemen). Struktur elemen (SE) berupa matrik dan pada umumnya berukuran kecil (Putra, 2010).



Gambar 2.5 Beberapa contoh SE (Putra, 2010)

Posisi yang dilingkari pada gambar 2.5 menyatakan pusat koordinat.

Bentuk umum yang digunakan pada operasi morfologi adalah cakram dan lingkaran. Efek yang diberikan merata pada segala arah. Bentuk struktur elemen yang lain adalah belah ketupat, garis, persegi panjang, bujur sangkar dan oktagon.



Gambar 2.6 Bentuk struktur elemen (Gonzalez dan Woods, 2002)

Terdapat dua operasi dasar morfologi yaitu dilasi dan erosi. Kedua operasi ini merupakan dasar untuk membuat berbagai operasi morfologi seperti: *opening, closing*.

a. Dilasi

Operasi dilasi menghasilkan objek yang lebih luas, karena piksel yang memiliki intensitas keabuan yang rendah diubah menjadi tinggi (Putra, 2010). Operasi dilasi pada citra akan mencari nilai piksel yang bernilai maksimum di dalam lingkungan struktur elemen. Dilasi pada citra *grayscale* menggunakan struktur elemen 5x5 didefinisikan sebagai berikut (Pratt, 2007):

$$G(j, k) = \text{MAX}\{ F(j, k), F(j, k + 1), F(j, k + 2), F(j + 1, k), F(j + 1, k + 1), F(j + 1, k + 2), F(j + 2, k), F(j + 2, k + 1), F(j + 2, k + 2), F(j, k-1), F(j + 1, k-1), F(j + 2, k-1), F(j, k-2), F(j + 1, k-2), F(j + 2, k-2), F(j-1, k), F(j-1, k + 1), F(j-1, k + 2), F(j-1, k-1), F(j-1, j-2), F(j-2, k), F(j-2, k + 1), F(j-2, k + 2), F(j-2, k-1), F(j-2, k-2) \} \quad (2.3)$$

Persamaan (2.3) digambarkan dalam bentuk matriks:

$(j-2, k-2)$	$(j-2, k-1)$	$(j-2, k)$	$(j-2, k+1)$	$(j-2, k+2)$
$(j-1, k-2)$	$(j-1, k-1)$	$(j-1, k)$	$(j-1, k+1)$	$(j-1, k+2)$
$(j, k-2)$	$(j, k-1)$	(j, k)	$(j, k+1)$	$(j, k+2)$
$(j+1, k-2)$	$(j+1, k-1)$	$(j+1, k)$	$(j+1, k+1)$	$(j+1, k+2)$
$(j+2, k-2)$	$(j+2, k-1)$	$(j+2, k)$	$(j+2, k+1)$	$(j+2, k+2)$

b. Erosi

Operasi erosi menghasilkan objek yang menyempit (mengecil), karena piksel yang memiliki intensitas keabuan yang tinggi diubah menjadi rendah (Putra, 2010). Operasi erosi pada citra akan mencari nilai piksel yang bernilai minimum di dalam lingkungan struktur elemen. Erosi pada citra *grayscale* menggunakan struktur elemen 5x5 didefinisikan sebagai berikut (Pratt, 2007):

$$G(j, k) = \text{MIN}\{ F(j, k), F(j, k + 1), F(j, k + 2), F(j + 1, k), F(j + 1, k + 1), F(j + 1, k + 2), F(j + 2, k), F(j + 2, k + 1), F(j + 2, k + 2), F(j, k - 1), F(j + 1, k - 1), F(j + 2, k - 1), F(j, k - 2), \}$$

$$\begin{aligned}
&F(j+1, k-2), F(j+2, k-2), F(j-1, k), F(j-1, k+1), \\
&F(j-1, k+2), F(j-1, k-1), F(j-1, j-2), F(j-2, k), \\
&F(j-2, k+1), F(j-2, k+2), F(j-2, k-1), \\
&F(j-2, k-2)\} \tag{2.4}
\end{aligned}$$

c. *Opening*

Operasi *opening* adalah operasi erosi yang diikuti dengan dilasi menggunakan struktur elemen yang sama. Operasi ini berguna untuk menghaluskan kontur objek dan menghilangkan piksel di area yang terlalu kecil untuk ditempati oleh struktur elemen. Semua struktur elemen akan tereleminasi oleh erosi dan kemudian penghalusan akan dilakukan melalui dilasi (Gonzalez dan Woods, 2002).

Defenisi operasi opening seperti berikut (Fauzi dan Memi, 2011):

$$\begin{aligned}
(f \circ b)(j, k) = \text{MAK}\{ &\text{MIN}[F(j, k), F(j, k+1), F(j, k+2), F(j+1, k), \\
&F(j+1, k+1), F(j+1, k+2), F(j+2, k), F(j+2, k+1), \\
&F(j+2, k+2), F(j, k-1), F(j+1, k-1), F(j+2, k-1), \\
&F(j, k-2), F(j+1, k-2), F(j+2, k-2), F(j-1, k), \\
&F(j-1, k+1), F(j-1, k+2), F(j-1, k-1), F(j-1, j-2), \\
&F(j-2, k), F(j-2, k+1), F(j-2, k+2), F(j-2, k-1), \\
&F(j-2, k-2)]\} \tag{2.5}
\end{aligned}$$

d. *Closing*

Operasi *closing* berguna untuk menghaluskan kontur dan menghilangkan lubang-lubang kecil (Gonzalez dan Woods, 2002).

Defenisinya sebagai berikut (Fauzi dan Memi, 2011):

$$\begin{aligned}
(f \bullet b)(j, k) = \text{MIN}\{ &\text{MAX}[F(j, k), F(j, k+1), F(j, k+2), F(j+1, k), \\
&F(j+1, k+1), F(j+1, k+2), F(j+2, k), F(j+2, k+1), \\
&F(j+2, k+2), F(j, k-1), F(j+1, k-1), F(j+2, k-1), \\
&F(j, k-2), F(j+1, k-2), F(j+2, k-2), F(j-1, k), \\
&F(j-1, k+1), F(j-1, k+2), F(j-1, k-1), F(j-1, j-2), \\
&F(j-2, k), F(j-2, k+1), F(j-2, k+2), F(j-2, k-1), \\
&F(j-2, k-2)]\} \tag{2.6}
\end{aligned}$$

Jadi, operasi *closing* dilaksanakan dengan melakukan operasi dilasi terlebih dahulu dan kemudian diikuti dengan operasi erosi.

e. Transformasi *Top Hat*

Transformasi *top hat* merupakan perbedaan antara citra masukan (citra *grayscale*) dan citra setelah mengalami operasi *opening* (Solomon dan Breckon, 2011). Secara matematis seperti berikut (Gonzalez dan Woods, 2002):

$$T_{hat}(f) = f - (f \circ b) \quad (2.7)$$

Pada rumus diatas, f adalah citra masukan (citra *grayscale*) dan $(f \circ b)$ adalah citra hasil operasi *opening*. Transformasi ini berguna untuk mendapatkan bentuk global suatu objek yang mempunyai intensitas yang bervariasi.

f. Transformasi *Bottom Hat*

Transformasi *bottom hat* didefinisikan sebagai berikut (Gonzalez dan Woods, 2002):

$$B_{hat} = (f \bullet b) - f \quad (2.8)$$

Transformasi *bottom hat* merupakan perbedaan antara citra setelah mengalami operasi *closing* dan citra masukan (citra *grayscale*). Pada rumus diatas, f adalah citra masukan (citra *grayscale*) dan $(f \bullet b)$ adalah citra hasil operasi *closing*.

g. *Top-bottom hat*

Pada MATLAB, *top hat* dan *bottom hat* dapat digunakan bersama-sama untuk meningkatkan kontras dalam foto.

Syntax:

```
J=imsubtract(imadd(I,tophat),bottomhat);
```

Maksud *syntax* diatas citra masukan ditambah dengan citra hasil transformasi *top hat*, kemudian dikurangi dengan citra hasil transformasi *bottom hat*.

I merupakan citra masukan

se merupakan struktur elemen

`imadd` digunakan untuk menambah dua gambar atau konstan gambar.

`Imsubtract` digunakan untuk mengurangi satu gambar atau konstan gambar.

3. Histogram *Equalization* (Perataan Histogram)

Histogram citra memberikan informasi tentang penyebaran intensitas piksel-piksel di dalam citra. Misalnya, citra yang terlalu terang atau terlalu gelap memiliki histogram yang sempit (Munir, 2004).

Citra yang baik dapat diperoleh dengan mengubah penyebaran nilai intensitas piksel-piksel citra. Teknik yang sering dipakai yaitu perataan histogram (*histogram equalization*). Tujuan dari perataan histogram adalah untuk memperoleh penyebaran histogram yang merata, sehingga setiap derajat keabuan memiliki jumlah piksel yang relatif sama.

Pemerataan histogram dilakukan sebagai berikut:

➤ Tentukan frekuensi tiap *gray level*.

➤ Hitung PMF (*Probability Mass Function*).

PMF singkatan dari fungsi massa probabilitas. PMF digunakan untuk menghitung probabilitas tiap *gray level*. PMF dihitung dengan cara membagi nilai frekuensi tiap *gray level* dengan jumlah total frekuensi *gray level*.

➤ Hitung CDF (*Cumulative Distributive Function*).

CDF singkatan dari fungsi distributif kumulatif. CDF digunakan untuk menghitung jumlah kumulatif dari semua nilai-nilai yang dihitung pada PMF. CDF dihitung dengan cara:

- Nilai CDF pertama adalah nilai PMF pertama.
- Nilai CDF kedua adalah hasil penjumlahan nilai PMF pertama dan kedua.
- Nilai CDF ketiga adalah hasil penjumlahan nilai PMF pertama sampai nilai PMF ketiga. Dan seterusnya.

➤ Tentukan *gray level* baru dengan cara mengalikan tiap nilai CDF dengan jumlah *level* pada citra dikurang satu.

4. Filter Spasial

Pada proses *filter* spasial, nilai piksel baru umumnya dihitung berdasarkan piksel tetangga. Proses *filter* spasial tidak dapat terlepas dari teori kernel (*mask*) dan konvolusi (Putra, 2010).

Kernel (*mask*)

Kernel adalah matrik berukuran kecil dengan elemen berupa bilangan. Kernel digunakan pada proses konvolusi. Kernel juga disebut dengan *convolution window* (jendela konvolusi). Ukuran kernel dapat berbeda-beda seperti: 3 x 3, 5 x 5, dan sebagainya. Elemen kernel disebut dengan bobot (*weight*) yaitu bilangan yang membentuk pola tertentu (Putra, 2010).

1	0
0	1

(a)

1	-1	1
-1	4	-1
1	-1	1

(b)

Gambar 2.7 Contoh kernel (*mask*) (a) kernel berukuran 2 x 2 dan (b) kernel berukuran 3 x 3 (Putra, 2010)

Konvolusi (*convolution*)

Proses konvolusi yaitu mengalikan setiap nilai piksel tetangga pada citra *input* dengan bobot pada kernel (*mask*) kemudian menjumlahkan hasil perkalian tersebut sehingga menghasilkan nilai piksel baru (Putra, 2010).

Contoh citra *input* dan kernel (*mask*) pada gambar 2.8

U _{1,1}	U _{1,2}	U _{1,3}	U _{1,4}	U _{1,5}	U _{1,6}	U _{1,7}	U _{1,8}
U _{2,1}	U _{2,2}	U _{2,3}	U _{2,4}	U _{2,5}	U _{2,6}	U _{2,7}	U _{2,8}
U _{3,1}	U _{3,2}	U _{3,3}	U _{3,4}	U _{3,5}	U _{3,6}	U _{3,7}	U _{3,8}
U _{4,1}	U _{4,2}	U _{4,3}	U _{4,4}	U _{4,5}	U _{4,6}	U _{4,7}	U _{4,8}
U _{5,1}	U _{5,2}	U _{5,3}	U _{5,4}	U _{5,5}	U _{5,6}	U _{5,7}	U _{5,8}
U _{6,1}	U _{6,2}	U _{6,3}	U _{6,4}	U _{6,5}	U _{6,6}	U _{6,7}	U _{6,8}
U _{7,1}	U _{7,2}	U _{7,3}	U _{7,4}	U _{7,5}	U _{7,6}	U _{7,7}	U _{7,8}
U _{8,1}	U _{8,2}	U _{8,3}	U _{8,4}	U _{8,5}	U _{8,6}	U _{8,7}	U _{8,8}

(a)

K _{1,1}	K _{1,2}	K _{1,3}
K _{2,1}	K _{2,1}	K _{2,1}

(b)

Gambar 2.8 (a) citra *input* (b) kernel (*mask*) (Putra, 2010)

Secara matematika proses konvolusi menggunakan tanda (*) dapat dinyatakan sebagai berikut:

$$O(i, j) = \sum_{k=1}^m \sum_{l=1}^n U(i+k-1, j+l-1)K(k, l) \quad (2.12)$$

$i = 1 \dots M - m + 1$ dan $j = 1 \dots N - n + 1$. Dimana M dan N menyatakan ukuran baris dan kolom dari citra *input*, sedangkan m dan n menyatakan ukuran baris dan kolom dari kernel. Ukuran citra hasil konvolusi adalah $(M - m + 1)$ baris dan $(N - n + 1)$ kolom, berarti ukuran citra hasil konvolusi lebih kecil dari ukuran citra awal.

Proses konvolusi di luar batas ukuran citra tetap dilakukan dengan menambahkan nilai 0 (*zero*) untuk piksel di luar batas. Proses konvolusi tetap dapat dilakukan. Namun, nilai piksel konvolusi di luar batas ukuran citra tidak menunjukkan nilai piksel sebenarnya (Putra, 2010).

Gaussian Filtering

Gaussian filtering digunakan untuk proses penghalusan (*smoothing*), menghilangkan detil, dan menghilangkan derau (*noise*) (Putra, 2010).

Fungsi *Gaussian* dimensi 1 (1-D)

$$G(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp(-x^2/2\sigma^2) \quad (2.13)$$

σ menyatakan standar deviasi dari distribusi. $\pi = 3,14$. Fungsi diatas diasumsikan memiliki mean 0 (pusat distribusi pada garis $x = 0$). Semakin besar nilai σ maka kurva distribusi *Gaussian* semakin melebar dan puncaknya menurun.

Berikut ini nilai kernel *Gaussian* 1-D untuk $\sigma = 1$ (Putra, 2010).

Gaussian 1-D untuk $\sigma = 1$

0.000	0.004	0.054	0.242	0.399	0.342	0.054	0.004	0.000
-------	-------	-------	-------	-------	-------	-------	-------	-------

Nilai 0,399 pada *Gaussian* 1-D untuk $\sigma = 1$ diperoleh dari:

$$G(0) = \frac{1}{\sqrt{(2)(3.14)(1^2)}} \exp(-0^2/(2)(1^2)) = 0,3989$$

Fungsi *Gaussian* dimensi 2 (2-D):

$$G(x, y) = \frac{1}{2\pi\sigma^2} \exp(-(x^2 + y^2)/2\sigma^2) \quad (2.14)$$

Berikut ini nilai kernel *Gaussian 2-D* untuk $\sigma = 1$ dan ukuran kernel 5×5

0.0029	0.0131	0.0215	0.0131	0.0029
0.0131	0.0585	0.0965	0.0585	0.0131
0.0215	0.0965	0.1592	0.0965	0.0215
0.0131	0.0585	0.0965	0.0585	0.0131
0.0029	0.0131	0.0215	0.0131	0.0029

$$G(0,0) = \frac{1}{(2)(3.14)(1^2)} \exp(-(0^2 + 0^2)/(2)(1^2)) = 0.1592$$

Proses penghalusan citra dapat dilakukan dengan proses konvolusi citra *input* dengan kernel *Gaussian*. Tingkat atau derajat kehalusan citra hasil *Gaussian filtering* dapat diatur dengan mengubah nilai σ .

2.5.2 Segmentasi Citra

Segmentasi citra merupakan proses yang ditujukan untuk mendapatkan objek-objek yang terkandung di dalam citra atau membagi citra ke dalam beberapa daerah dengan setiap objek atau daerah yang memiliki kemiripan atribut (Putra, 2010).

Teknik segmentasi citra didasarkan pada dua properti dasar nilai keabuan: ketidaksinambungan dan kesamaan antar piksel. Pertama, pemisahan citra didasarkan pada perubahan mendadak pada keabuan, contohnya menggunakan pendekatan seperti deteksi tepi dan deteksi garis. Kedua, didasarkan pada kesamaan antar piksel dalam suatu area, yang termasuk cara kedua ini adalah pengembangan berdasarkan histogram, pertumbuhan area, pemisahan dan penggabungan area, pengelompokan atau pengklasifikasian, pendekatan teori graf, dan pendekatan berbasis aturan (Acharya dan Ray, 2005).

2.5.3 Pengembangan (*Thresholding*)

Pengembangan mengelompokkan nilai derajat keabuan setiap *pixel* ke dalam dua kelas yaitu hitam dan putih. Terdapat dua pendekatan yang digunakan dalam pengembangan yaitu (Munir, 2004):

1. Pengembangan secara global

Pada pengembangan secara global, seluruh *pixel* pada citra dikonversikan menjadi hitam atau putih dengan satu nilai ambang (T).

2. Pengembangan secara lokal.

Pengembangan secara lokal dilakukan terhadap daerah-daerah di dalam citra. Dalam hal ini, citra dibagi menjadi beberapa bagian kecil, kemudian pengembangan dilakukan secara lokal. Nilai ambang untuk setiap bagian belum tentu sama dengan bagian lain.

2.5.4 Metode Otsu

Menghitung nilai ambang (T) pada metode *Otsu* dilakukan secara otomatis berdasarkan gambar masukan dengan cara membagi histogram citra *gray level* kedalam dua daerah yang berbeda. Pendekatan yang dilakukan pada metode *Otsu* adalah analisis diskriminan. Analisis diskriminan adalah menentukan variabel yang dapat membedakan antara dua atau lebih kelompok yang muncul secara alami. Analisis Diskriminan akan memaksimumkan variabel tersebut agar dapat membagi objek latar depan (*foreground*) dan latar belakang (*background*) (Otsu, 1979).

Algoritma metode *Otsu* (Otsu, 1979):

Nilai ambang (T) yang akan dicari dari suatu citra *gray level* dinyatakan dengan k. Nilai k berkisar antara 1 sampai dengan L.

1. Hitung probabilitas distribusi (normalisasi histogram *gray level*) dari citra.

$$P_i = \frac{n_i}{N} \quad (2.15)$$

Dengan P_i menyatakan probabilitas pada *gray level*, dimana
 $i = 1, 2, \dots, L$

n_i menyatakan frekuensi tiap nilai *gray level*

N menyatakan total jumlah frekuensi *gray level*, dimana ($N = n_1 + n_2 + \dots + n_L$)

2. Histogram dibagi menjadi dua kelas yaitu latar belakang dan objek.

Jumlah kumulatif latar belakang dan objek dinyatakan dengan persamaan:

$$\omega_0(k) = \sum_{i=1}^k P_i \quad (2.16)$$

$$\omega_1(k) = \sum_{i=k+1}^L P_i \quad (2.17)$$

$\omega_0(k)$ adalah jumlah kumulatif latar belakang.

$\omega_1(k)$ adalah jumlah kumulatif objek.

k adalah indeks nilai ambang ($k = 1, 2, \dots, L$)

3. Hitung rerata kumulatif latar belakang dan objek menggunakan persamaan:

$$\mu_0(k) = \sum_{i=1}^k \frac{iP_i}{\omega_0(k)} \quad (2.18)$$

$$\mu_1(k) = \sum_{i=k+1}^L \frac{iP_i}{\omega_1(k)} \quad (2.19)$$

$\mu_0(k)$ adalah rerata kumulatif latar belakang.

$\mu_1(k)$ adalah rerata kumulatif objek.

4. Hitung total rerata (μ_T) menggunakan persamaan:

$$\mu_T = \sum_{i=1}^L iP_i \quad (2.20)$$

5. Hitung *between class variance* menggunakan persamaan:

$$\sigma_B^2(k) = \omega_0(k) \cdot (\mu_0(k) - \mu_T)^2 + \omega_1(k) \cdot (\mu_1(k) - \mu_T)^2 \quad (2.21)$$

6. Pilih nilai *threshold* k^* yang merupakan indeks dimana nilai varians antar kelas maksimum ($\sigma_B^2(k) \rightarrow \max$).

2.5.5 Pengambangan berbasis Entropi (*Entropy Based Thresholding*)

Menurut Shannon, entropi adalah ukuran informasi yang terkandung di dalam citra (Shanon dan Weaver, 1949). Entropi efektif digunakan untuk ambang batas dalam segmentasi citra. Dalam *entropy based thresholding*, entropi objek dan latar belakang digunakan untuk seleksi optimal nilai ambang batas (Sahoo dkk, 1988). Entropi digunakan dalam pengambangan untuk memperoleh nilai ambang yang optimal (Kapur dkk, 1985).

Metode pengambangan berbasis entropi yang dikemukakan oleh Kapur dkk sebagai berikut:

$$P(i) = \frac{f_i}{N} \quad (2.22)$$

$P(i)$ menyatakan probabilitas pada *gray level* (i), dimana $i = 1, 2, \dots, L$

N menyatakan total jumlah frekuensi pada citra

f_i menyatakan frekuensi tiap nilai *gray level*

Jika t adalah nilai ambang, entropi latar belakang dinyatakan sebagai berikut ini:

$$H_b(t) = - \sum_{i=1}^t \frac{P(i)}{P_b(t)} \log \frac{P(i)}{P_b(t)} \quad (2.23)$$

$$P_b(t) = \sum_{i=1}^t P(i) \quad (2.24)$$

$P_b(t)$ menyatakan probabilitas pada latar belakang citra.

t adalah indeks nilai ambang, dimana $t = 1, 2, \dots, L$

dan entropi objek dinyatakan sebagai berikut:

$$H_o(t) = - \sum_{i=t+1}^L \frac{P(i)}{P_o(t)} \ln \frac{P(i)}{P_o(t)} \quad (2.25)$$

$$P_o(t) = \sum_{i=t+1}^L P(i) \quad (2.26)$$

$P_o(t)$ menyatakan probabilitas pada objek citra.

Jumlah H_b dan H_o dinyatakan sebagai berikut:

$$\psi(t) = \left(- \sum_{i=1}^t \frac{P(i)}{P_b(t)} \log \frac{P(i)}{P_b(t)} \right) + \left(- \sum_{i=t+1}^L \frac{P(i)}{P_o(t)} \ln \frac{P(i)}{P_o(t)} \right) \quad (2.27)$$

Optimalisasi dilakukan dengan memaksimalkan fungsi $\psi(t)$.

2.5.6 Metode Pengukuran Kuantitatif

Pengukuran kuantitatif adalah mengukur seberapa banyak perbedaan nilai piksel demi piksel antara citra asli dengan citra yang diuji. Dalam citra digital terdapat suatu standar pengukuran kuantitatif citra. Pada tugas akhir ini, pengukuran kuantitatif terbagi dua yaitu pengukuran kuantitatif untuk pemisahan objek dan latar belakang dan pengukuran kuantitatif untuk kualitas citra.

2.5.6.1 Pengukuran Kuantitatif untuk Pemisahan Objek dan Latar Belakang

Pengukuran kuantitatif untuk pemisahan objek dan latar belakang menggunakan *misclassification error* (ME) dan *relative area foreground error* (RAE). Perhitungan pengukuran kuantitatif ini menggunakan citra *ground truth* sebagai pembanding untuk citra hasil segmentasi. Citra *ground truth* merupakan citra referensi untuk mengukur kinerja dari metode pengukuran ME dan RAE. Citra *ground truth* di segmentasi secara manual berdasarkan citra aslinya (Zainal dan Asano, 2006).

1. *Misclassification Error* (ME)

Misclassification Error (ME) digunakan untuk mengetahui kesalahan klasifikasi rasio piksel latar belakang yang dinyatakan sebagai latar depan dan sebaliknya.

Menghitung *Misclassification Error* (ME) dapat dilakukan sebagai berikut (Sankur dan Sezgin, 2004):

$$ME = 1 - \frac{|E_o \cap B_T| + |F_o \cap F_T|}{|E_o| + |F_o|} \quad (2.28)$$

B_o menyatakan latar belakang dari citra *ground truth*.

F_o menyatakan latar depan dari citra *ground truth*.

B_T menyatakan latar belakang dari citra yang diuji (hasil segmentasi)

F_T menyatakan latar depan dari citra yang diuji (hasil segmentasi)

$||$ menyatakan kardinalitas dari himpunan (jumlah elemen pada himpunan)

Jika hasil ME bernilai rendah, maka gambar sempurna diklasifikasi.

Jika hasil ME bernilai tinggi, maka terdapat kesalahan dalam binerisasi.

2. *Relative Foreground Area Error* (RAE)

Relative Foreground Area Error (RAE) digunakan untuk membandingkan daerah latar depan antara citra *ground truth* dan citra yang diuji (hasil segmentasi).

Relative Foreground Area Error (RAE) dapat dilakukan sebagai berikut (Sankur dan Sezgin, 2004):

$$RAE = \begin{cases} \frac{A_o - A_T}{A_o} & \text{if } A_T < A_o \\ \frac{A_T - A_o}{A_T} & \text{if } A_T \geq A_o \end{cases} \quad (2.29)$$

A_o menyatakan daerah latar depan dari citra *ground truth*.

A_T menyatakan daerah latar depan dari citra yang diuji (hasil segmentasi).

Jika hasil RAE bernilai rendah, maka daerah latar depan (*foreground*) tersegmentasi dengan baik.

2.5.6.2 Pengukuran Kuantitatif untuk Kualitas Citra

Pengukuran kuantitatif untuk kualitas citra menggunakan *peak signal to noise ratio* (PSNR). Perhitungan pengukuran kuantitatif ini menggunakan citra *grayscale* sebagai pembanding untuk citra hasil segmentasi. Nilai *peak signal to noise ratio* (PSNR) ditentukan oleh besar atau kecil nilai dari MSE. Hubungan

antara nilai PSNR dengan nilai MSE adalah semakin besar nilai PSNR, maka akan semakin kecil nilai MSE.

1. *Mean Square Error* (MSE)

Mean Square Error (MSE) adalah kesalahan kuadran rata-rata. *Mean Square Error* (MSE) dapat dilakukan sebagai berikut (Mahjud, 2010):

$$MSE = \frac{1}{N \times M} \sum_{i=1}^N \sum_{j=1}^M [I(i,j) - I'(i,j)]^2 \quad (2.30)$$

N menyatakan baris piksel pada citra.

M menyatakan kolom piksel pada citra.

$I(i,j)$ menyatakan citra grayscale.

$I'(i,j)$ menyatakan citra hasil segmentasi

Jika hasil MSE bernilai besar, maka tampilan pada citra hasil segmentasi akan semakin buruk .

Jika hasil MSE bernilai kecil, maka tampilan pada citra hasil segmentasi

2. *Peak Signal to Noise Ratio* (PSNR)

Peak Signal to Noise Ratio (PSNR) merupakan nilai rasio yang menunjukkan tingkat *noise* tertentu terhadap banyaknya *noise* pada sinyal citra. Noise adalah kerusakan sinyal pada bagian tertentu. Satuan untuk PSNR adalah desibel (dB). *Peak Signal to Noise Ratio* (PSNR) dapat dilakukan sebagai berikut (Mahjud, 2010):

$$PSNR = 10 \log_{10} \left[\frac{(255)^2}{MSE} \right] \quad (2.31)$$

Jika hasil PSNR bernilai besar, maka kualitas citra hasil segmentasi baik.

Jika hasil PSNR bernilai kecil, maka kualitas citra hasil segmentasi buruk.