

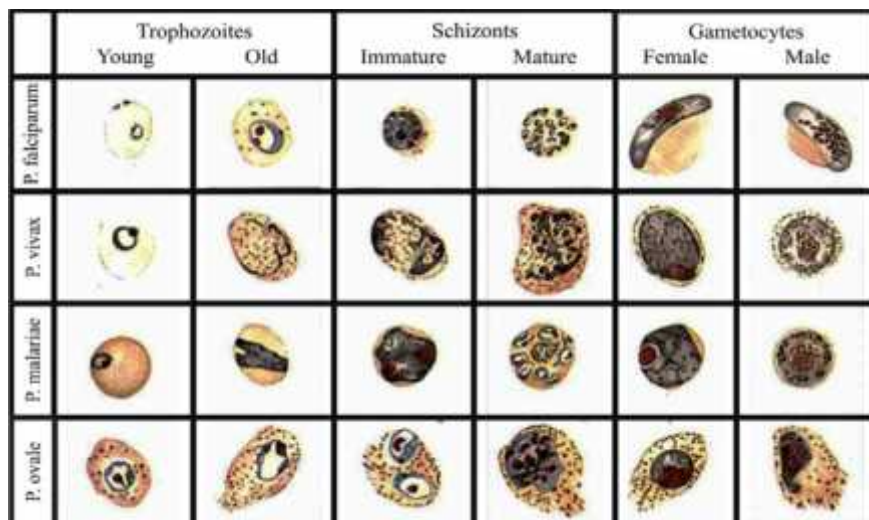
BAB II

LANDASAN TEORI

2.1. Malaria

Malaria adalah penyakit infeksi menular yang disebabkan oleh parasit dari genus *plasmodium*. Malaria ditularkan melalui gigitan nyamuk *Anopheles* dengan gambaran penyakit berupa demam periodik, *anemia*, pembesaran limpa dan berbagai kumpulan gejala penyakit lainnya. Parasit penyebab malaria terdiri dari empat jenis, yaitu: *plasmodium falciparum*, *plasmodium vivax*, *plasmodium malariae*, dan *plasmodium ovale*. Dari keempat jenis *plasmodium* tersebut, *plasmodium falciparum* dan *plasmodium vivax* merupakan jenis *plasmodium* yang paling sering dijumpai pada kasus penyebaran penyakit malaria pada sel darah merah (Adi, 2012).

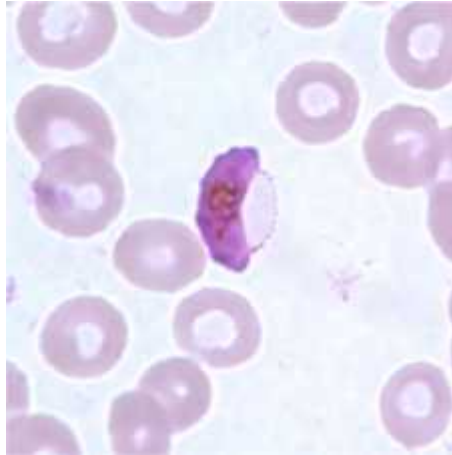
Teknik Morfologi *plasmodium* dibagi menjadi tiga fase yaitu *trophozoite*, *schizont* dan *gametocyte*. Fase *trophozoite* merupakan fase ketika parasit dalam proses pertumbuhan, fase *shizont* yaitu parasit dalam proses pembiakan, dan fase *gametocyte* adalah parasit dalam proses pembentukan kelamin. Bentuk *plasmodium* penyebab penyakit malaria berdasarkan fase perkembangannya ditunjukkan pada gambar 2.1.



Gambar 2.1 Fase perkembangan *plasmodium* penyebab malaria (Adi, 2012)

2.1.1 *Plasmodium Falciparum*

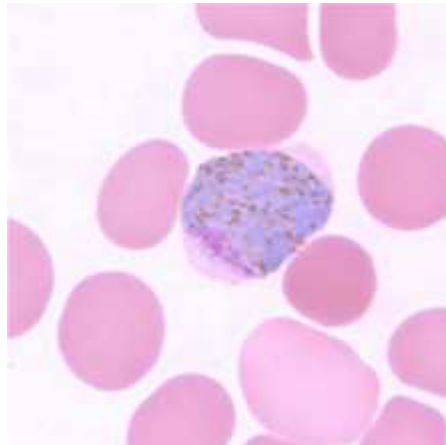
Menyebabkan malaria falciparum atau malaria tertiana yang maligna (ganas) atau dikenal dengan nama lain sebagai malaria tropika yang menyebabkan demam setiap hari. Citra mikroskopis sel darah merah yang terserang *plasmodium falciparum* dapat di lihat pada gambar 2.2



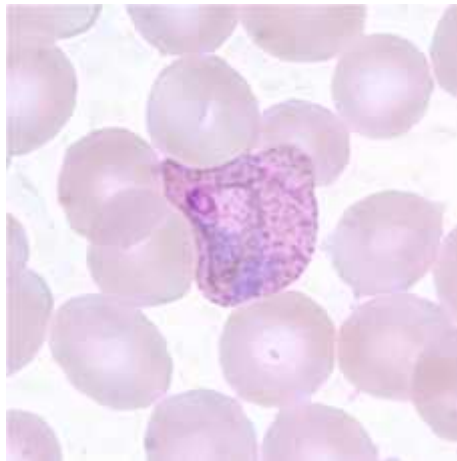
Gambar 2.2 *Plasmodium falciparum* fase gametocyte (Clifton, 2010)

2.1.2 *Plasmodium Vivax*

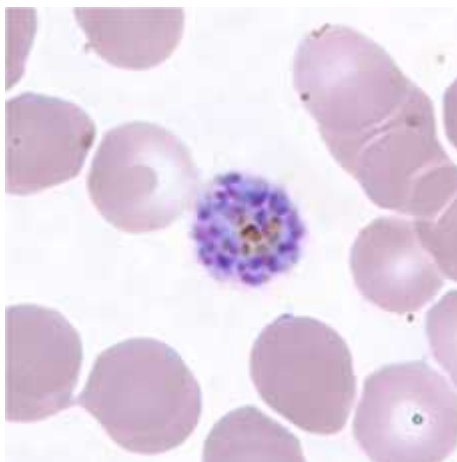
Menyebabkan malaria vivax atau disebut juga malaria tertiana benigna (jinak). Masa inkubasinya (masa dari penggigitan di tubuh manusia hingga menimbulkan peyakit) adalah sekitar 8-13 hari. Infeksi parasit ini bisa sampai ke bagian limpa. Parasit tipe ini bias bersembunyi di dalam hati dan kembali lagi setelah kondisi memungkinkan. Citra mikroskopis sel darah merah yang terserang *plasmodium vivax* fase gametocyte dapat di lihat pada gambar 2.3, fase thropozoite dapat di lihat pada gambar 2.4, dan fase schizont dapat dilihat pada gambar 2.5.



Gambar 2.3 *Plasmodium vivax* fase gametocyte (Clifton, 2010)



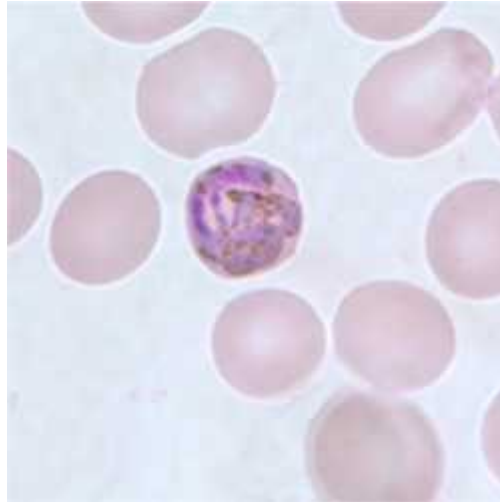
Gambar 2.4 *Plasmodium vivax* fase thropozoite (Clifton, 2010)



Gambar 2.5 *Plasmodium vivax* fase schizont (Clifton, 2010)

2.1.3 *Plasmodium Malariae*

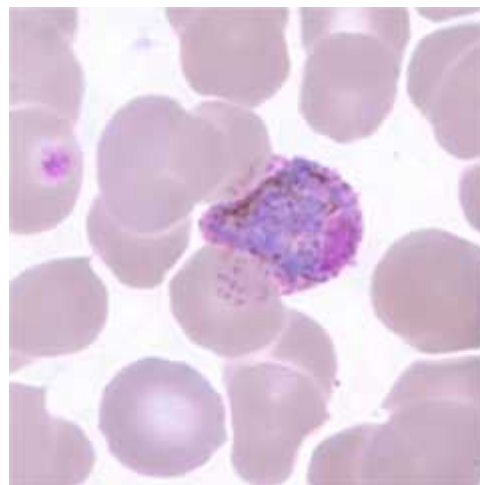
Menyebabkan malaria kuartana atau malaria malariae. Plasmodium ini masa inkubasinya 2 sampai 4 minggu. Jika tidak diobati, infeksi bisa bertahan dalam waktu tahunan. Citra mikroskopis sel darah merah yang terserang *plasmodium malariae* dapat di lihat pada gambar 2.4



Gambar 2.6 *Plasmodium malariae* fase gametocyte (Clifton, 2010)

2.1.4 *Plasmodium Ovale*

Jenis *plasmodium* ini jarang sekali dijumpai di Indonesia, karena umumnya banyak kasusnya terjadi di Afrika dan Pasifik Barat. Masa inkubasiya adalah selama 8 sampai 17 hari. Citra mikroskopis sel darah merah yang terserang *plasmodium ovale* dapat di lihat pada gambar 2.5



Gambar 2.7 *Plasmodium ovale* fase gametocyte (Clifton, 2010)

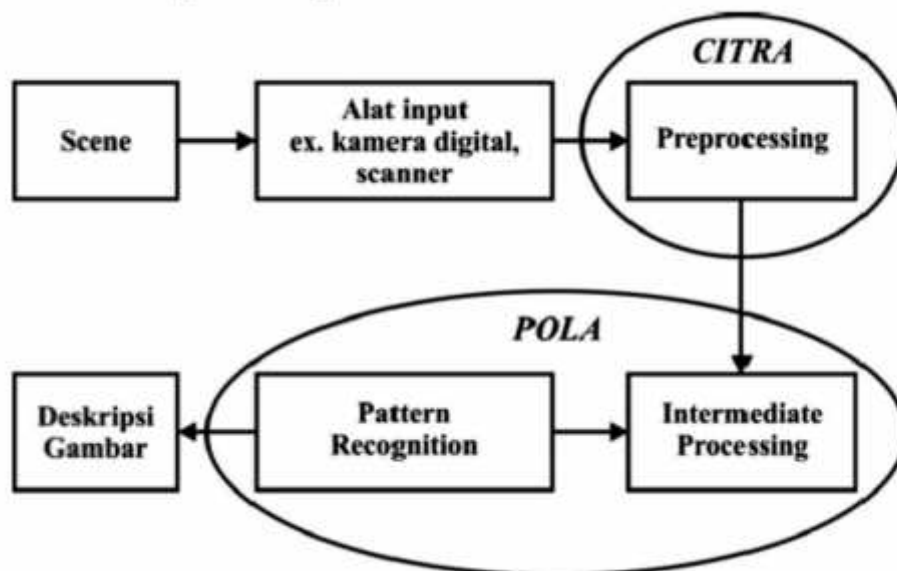
2.2. Citra Digital

Citra merupakan kumpulan elemen gambar yang secara keseluruhan merekam suatu adegan melalui media indra visual (Wirayuda, 2006). Citra digital adalah citra dua dimensi yang dapat direpresentasikan dengan sebuah fungsi intensitas cahaya di mana x dan y menyatakan koordinat spasial. Elemen terkecil dari sebuah citra digital disebut dengan *image element*, *picture element*, *pels*, atau *pixels*. Manipulasi citra digital dilakukan dengan memanipulasi nilai dari matrik dua dimensi yang merepresentasikan citra tersebut.

2.3. Pengolahan Citra Digital

Pengolahan citra digital adalah pemrosesan gambar berdimensi-dua melalui computer digital (Jain, 1989). Menurut Efford (2000), pengolahan citra adalah istilah umum untuk berbagai teknik yang keberadaannya untuk memanipulasi dan memodifikasi citra dengan berbagai cara.

Secara garis besar proses yang dapat dilakukan terhadap citra dapat dilihat pada gambar 2.2



Gambar 2.8 Proses pada suatu citra (Wirayuda, 2006)

2.4 Jenis – Jenis Citra Digital

Menurut wahana komputer (2013) citra digital dapat dikelompokkan menjadi 4 kategori, yaitu :

2.4.1 Citra Biner

Pixel terdiri dari warna hitam dan putih. Karena itulah kemungkinan nilai yang ada hanya dua yaitu hitam atau putih, oleh karena itu hanya diperlukan satu bit per *pixel*. Citra biner memiliki kelebihan irit penyimpanan data. Citra biner cocok untuk digunakan pada citra yang menggambarkan *text*, citra arsitektural, dan *fingerprint*.

2.4.2 Citra Abu-Abu

Citra yang terdiri dari pixel berisi warna abu-abu yang terdiri dari nilai 0 (hitam) sampai 255 (putih). Dengan kata lain, jarak pada tiap *pixel* adalah bernilai 8 bit atau 1 byte. Citra abu-abu atau citra *grayscale* biasa digunakan untuk kepentingan medis (*x-ray*) dan citra yang berkaitan pekerjaan cetak.

2.4.3 Citra RGB

Pada citra RGB, setiap *pixel* memiliki sebuah warna tertentu. Warna dideskripsikan oleh campuran warna merah (*red*), hijau (*green*), dan biru (*blue*). Jika setiap komponen (biru, hijau, merah) memiliki range antara 0-255, maka range keseluruhannya yang digunakan oleh citra RGB adalah 155 pangkat 3 atau 16.77.216 kombinasi warna. Karena citra RGB terdiri dari 3 matrix yang mewakili merah, hijau, dan biru, maka setiap *pixel* memiliki tiga buah nilai.

2.4.4 Indexed

Pada citra ini setiap *pixel* tidak berisikan nilai warna, tetapi nilai yang tertera pada gambar adalah index yang merujuk pada map. Cara tersebut sangat efektif apabila sebuah gambar memiliki kedalaman warna 256, karena index hanya memerlukan satu byte.

2.5 Ukuran File Citra

Ukuran sebuah citra bisa jadi berukuran sangat besar. Karena itu perlu diketahui seberapa besar informasi yang terdapat file tersebut. Contohnya, jika ada sebuah citra biner dengan dimensi ukuran 512 x 512 maka jumlah bit yang

dimiliki oleh citra tersebut adalah $512 \times 512 \times 1 = 262144 \text{ bit} = 32801,75 \text{ bytes}$ atau 32.0329 KB atau setara dengan $0,031 \text{ MB}$.

Perhitungan tersebut akan berbeda jika citra tersebut adalah citra RGB. Maka perhitungan ukuran citra tersebut menjadi $512 \times 512 \times 3 = 786432 \text{ bit} = 98304 \text{ KB} = 0,751 \text{ MB}$.

2.6 Thresholding

Menurut Kumaseh, Latumakulita, dan Nainggolan (2013) yang mengutip dari Purba dan Darma (2010) bahwa proses *thresholding* adalah mengubah citra *grayscale* menjadi suatu citra biner, secara matematis dapat ditulis sebagai berikut.

$$g(x,y) = \begin{cases} 1 & \text{if } f(x,y) \geq T \\ 0 & \text{if } f(x,y) < T \end{cases} \quad (2.1)$$

Dengan $g(x,y)$ adalah citra biner dari citra grayscale $f(x,y)$ dan T adalah nilai *threshold*. Nilai T dapat ditentukan menggunakan 2 cara yaitu *thresholding global* dan *thresholding local*.

2.6.1 Thresholding Global

Thresholding global adalah metode dengan seluruh *pixel* pada citra dikonversi menjadi menjadi hitam putih (citra biner) dengan suatu nilai *thresholding*. Metode otsu adalah contoh metode yang banyak digunakan dalam *thresholding global*. Citra $g(x,y)$ yang merupakan hasil Global threshold terhadap citra $f(x,y)$ adalah

$$g(x,y) = \begin{cases} 1 & \text{if } f(x,y) \geq T \\ 0 & \text{sebaliknya} \end{cases} \quad (2.2)$$

2.6.2 Thresholding Local

Thresholding local memiliki perbedaan dengan *thresholding global*. Pada *thresholding local*, nilai T tidak bergantung pada citra $f(x,y)$ saja, tetapi juga bergantung pada $p(x,y)$, dimana $p(x,y)$ adalah cirri khusus pada tiap *pixel* citra tersebut. Bentuk umum *thresholding local* adalah

$$g(x,y) = \begin{cases} 1 & \text{if } f(x,y) \leq T(x,y) \\ 0 & \text{sebaliknya} \end{cases} \quad (2.3)$$

2.7 Operasi Morfologi

Morfologi adalah teknik pengolahan citra digital dengan menggunakan bentuk objek sebagai pedoman dalam pengolahan (Wirayuda, 2006). Nilai dari setiap piksel dalam citra digital hasil diperoleh melalui proses perbandingan antara piksel yang bersesuaian pada citra digital masukan dengan piksel tetangganya.

Operasi morfologi bergantung pada urutan kemunculan dari piksel, tidak memperhatikan nilai *numeric* dari piksel sehingga teknik morfologi sesuai apabila digunakan untuk melakukan pengolahan *binary image* dan *grayscale image*.

Operasi morfologi standar yang biasa dilakukan adalah proses erosi dan dilasi. Dilasi adalah proses penambahan piksel pada batas dari suatu objek pada citra digital masukan, sedangkan erosi adalah proses pengurangan piksel pada batas dari suatu objek. Jumlah piksel yang ditambahkan atau yang dikurangkan dari batas objek pada citra digital masukan tergantung pada ukuran dan bentuk dari *structuring element* yang digunakan

2.7.1 Operasi Dilasi

Operasi dilasi biasa dipakai untuk mendapatkan efek pelebaran terhadap piksel yang bernilai 1. Operasi ini dirumuskan seperti berikut (Gonzales & Woods, 2002):

$$A \oplus B = \{z \mid [(B)_z \cap A] \subseteq A\} \quad (2.4)$$

Dalam hal ini,

- a) $\hat{B} = \{w \mid w = -b, \text{ untuk } b \in B\}$
- b) $(B)_z = \{c \mid c = a + z, \text{ untuk } a \in A\}$
- c) $z = (z_1, z_2)$

Burger & Burge (2008) mendefinisikan operasi dilasi sebagai berikut:

$$A \oplus B = \{z | z = a + b, \text{ dengan } a \in A \text{ dan } b \in B\} \quad (2.5)$$

Keterangan:

A = Himpunan *pixel* citra A

B = Himpunan *pixel* penstruktur B

Hasil dilasi berupa penjumlahan seluruh pasangan koordinat dari I dan H.

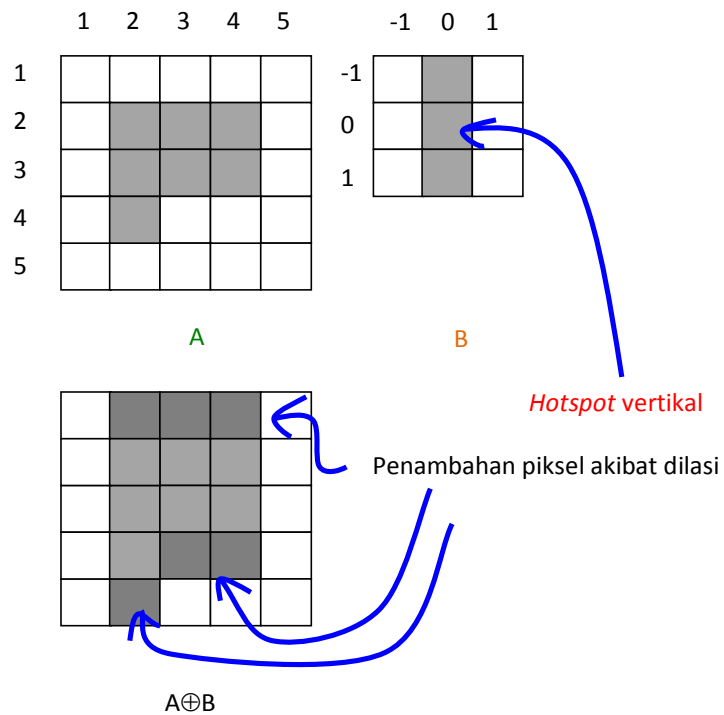
Contoh operasi dilasi dapat dilihat pada Gambar 2.9. Pada contoh tersebut,

$$A = \{ (2,2), (2,3), (2,4), (3,2), (3,3), (3,4), (4,3) \}$$

$$B = \{ (-1, 0), (0,0), (1,0) \}$$

Dengan demikian,

$$\begin{aligned} A \oplus B &= \{ (2,2) + (-1, 0), (2,2) + (0, 0) + (2,2) + (1, 0), \\ &\quad (2,3) + (-1, 0), (2,3) + (0, 0) + (2,3) + (1, 0), \\ &\quad (2,4) + (-1, 0), (2,4) + (0, 0) + (2,4) + (1, 0), \\ &\quad (3,2) + (-1, 0), (3,2) + (0, 0) + (3,2) + (1, 0), \\ &\quad (3,3) + (-1, 0), (3,3) + (0, 0) + (3,3) + (1, 0), \\ &\quad (3,4) + (-1, 0), (3,4) + (0, 0) + (3,4) + (1, 0), \\ &\quad (4,3) + (-1, 0), (4,3) + (0, 0) + (4,3) + (1, 0) \} \\ &= \{ (1,2), (2,2), (3,2), (1,3), (2,3), (3,3), \\ &\quad (1,4), (2,4), (3,3), (2,2), (3,2), (4,2), \\ &\quad (2,3), (3,3), (4,3), (2,4), (3,4), (4,4), \\ &\quad (3,3), (4,3), (5,3) \} \\ &= \{ (1,2), (1,3), (1,4), (2,2), (2,3), (2,4), \\ &\quad (3,2), (3,3), (3,4), (4,2), (4,3), (4,4), (5,3) \} \end{aligned}$$



Gambar 2.9 Efek dilasi dengan hotspot vertical (Kadir & Susanto, 2013)

2.7.2 Operasi Erosi

Operasi erosi mempunyai efek memperkecil struktur citra. Operasi ini dirumuskan seperti berikut (Gonzalez & Woods, 2002).

$$A \ominus B = \{z | (B)_z \subseteq A\} \quad (2.6)$$

Adapun Burger & Burge (2008) mendefinisikan erosi sebagai berikut:

$$A \ominus B = \{p \in Z^2 | (a + b) \in I, \text{ untuk setiap } b \in B\} \quad (2.7)$$

Keterangan:

A = Himpunan *pixel* citra A

B = Himpunan *pixel* penstruktur B

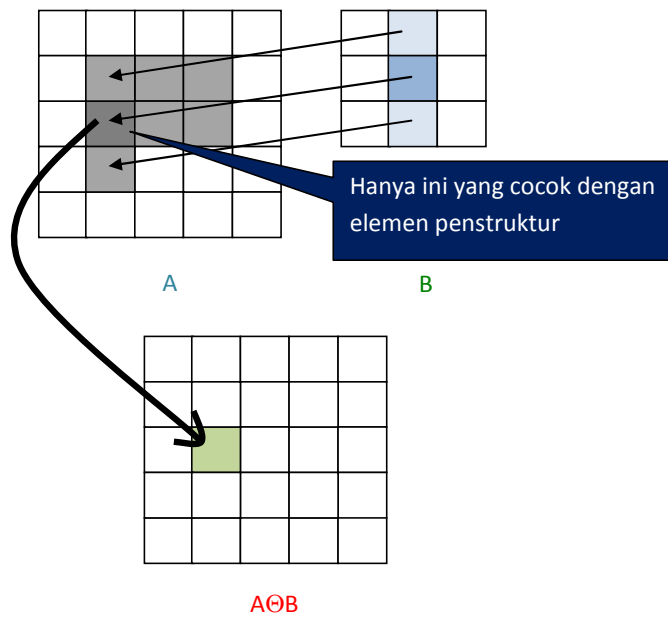
Contoh penggunaan fungsi erosi dapat dilihat berikut ini.

```
>> F = [ 0 0 0 0 0
0 1 1 1 0
0 1 1 1 0
0 1 0 0 0
0 0 0 0 0];
>> H = [0 1 0
0 1 0
0 1 0];
```

```
>> G = erosi(F, H)
G =
0 0 0 0 0
0 0 0 0 0
0 1 0 0 0
0 0 0 0 0
0 0 0 0 0
```

>>

Gambar berikut memperlihatkan bentuk visual untuk contoh di atas.



Gambar 2.10 Contoh visualisasi operasi erosi (Kadir & Susanto, 2013)

2.7.3 Operasi Closing

Operasi *closing* berguna untuk menghaluskan kontur dan menghilangkan lubang-lubang kecil. Definisinya seperti berikut:

$$A \bullet B = (A \oplus B) \ominus B \quad (2.8)$$

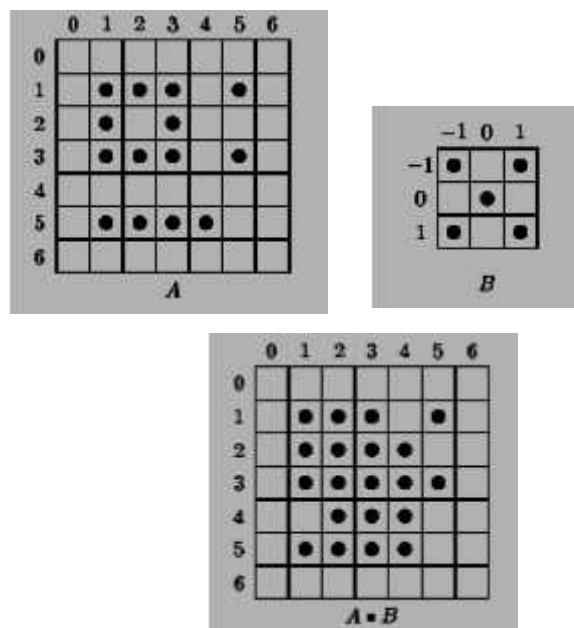
Keterangan:

A = Pixel citra A

B = Elemen penstruktur citra B

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

Contoh penggunaan operasi *closing* dapat dilihat pada gambar berikut ini :



Gambar 2.11 Contoh visualisasi operasi *closing* (Indah, 2009)

2.7.4 Operasi Opening

Operasi *opening* adalah operasi erosi yang diikuti dengan dilasi dengan menggunakan elemen penstruktur yang sama (Kadir & Susanto, 2013). Operasi ini memiliki fungsi untuk memberikan penghalusan permukaan objek dan mengeleminasi semua piksel di area yang terlalu sempit dan kecil untuk diisi oleh

elemen. Sehingga akhirnya seluruh area yang berukuran lebih kecil dari elemen penstruktur, akan dihilangkan menggunakan operasi erosi lalu operasi dilasi berperan untuk proses penghalusan.

Operasi *opening* membutuhkan fungsi elemen penstruktur khusus dalam pengoperasiannya. Fungsi tersebut adalah fungsi *strel*. Pada bahasa pemrograman MATLAB, *strel* memiliki berbagai macam pilihan dalam membuat elemen penstruktur. Table 4.1 menampilkan berbagai bentuk *strel*.

Tabel 2.1 Contoh *strel* untuk membuat berbagai bentuk elemen penstruktur

Penentu Bentuk	Contoh
'disk' (berbentuk cakram)	<code>strel('disk', 4)</code> → radius 4
'diamond' (berbentuk belah ketupat)	<code>strel('diamond', 4)</code> → radius 4
'line' (berbentuk garis)	<code>strel('line', 3, 0)</code> → panjang 3 dan sudut 0 derajat (datar) <code>strel('line', 3, 45)</code> → panjang 3 dan sudut 45 derajat (datar)
'octagon' (berbentuk segi delapan)	<code>strel('octagon', 6)</code> Argumen kedua harus kelipatan 3
'rectangle' (berbentuk persegi panjang)	<code>strel('rectangle', [4 2])</code> → 4 baris 2 kolom
'square' (berbentuk bujur sangkar)	<code>strel('square', 4)</code> → bujur sangkar 4 x 4

Berikut contoh proses operasi *opening* yang diaplikasikan pada citra struktur gambar 2.12. Dan hasil proses *opening* dapat dilihat pada gambar 2.12.



Gambar 2.13 Citra struktur (Kadir & Susanto, 2013)



Gambar 4.14 Operasi *Opening* dengan *strel* ('disk', 5)
(Kadir & Susanto, 2013)

Definisi operasi *opening* seperti berikut:

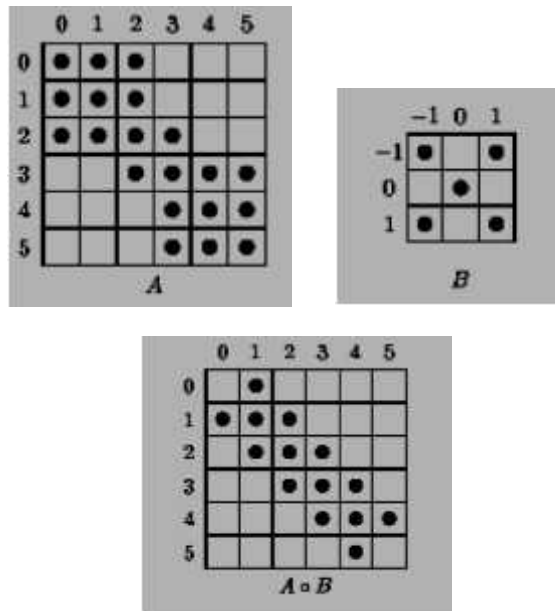
$$A \circ B = (A \ominus B) \oplus B \quad (2.9)$$

Keterangan:

A = *pixel* citra A

$B = \text{pixel}$ elemen penstruktur B

Contoh penggunaan operasi *closing* dapat dilihat pada gambar berikut ini :



Gambar 2.15 Contoh visualisasi operasi *opening* (Indah, 2009)

2.7.5 Operasi Region Filling

Operasi *region filling* adalah operasi yang berfungsi menutup lubang kecil pada suatu bagian citra. Operasi ini akan mengisi bagian dari piksel yang memiliki nilai 1 menjadi 0. Operasi ini mengacu pada piksel tetangga pada sebuah piksel.

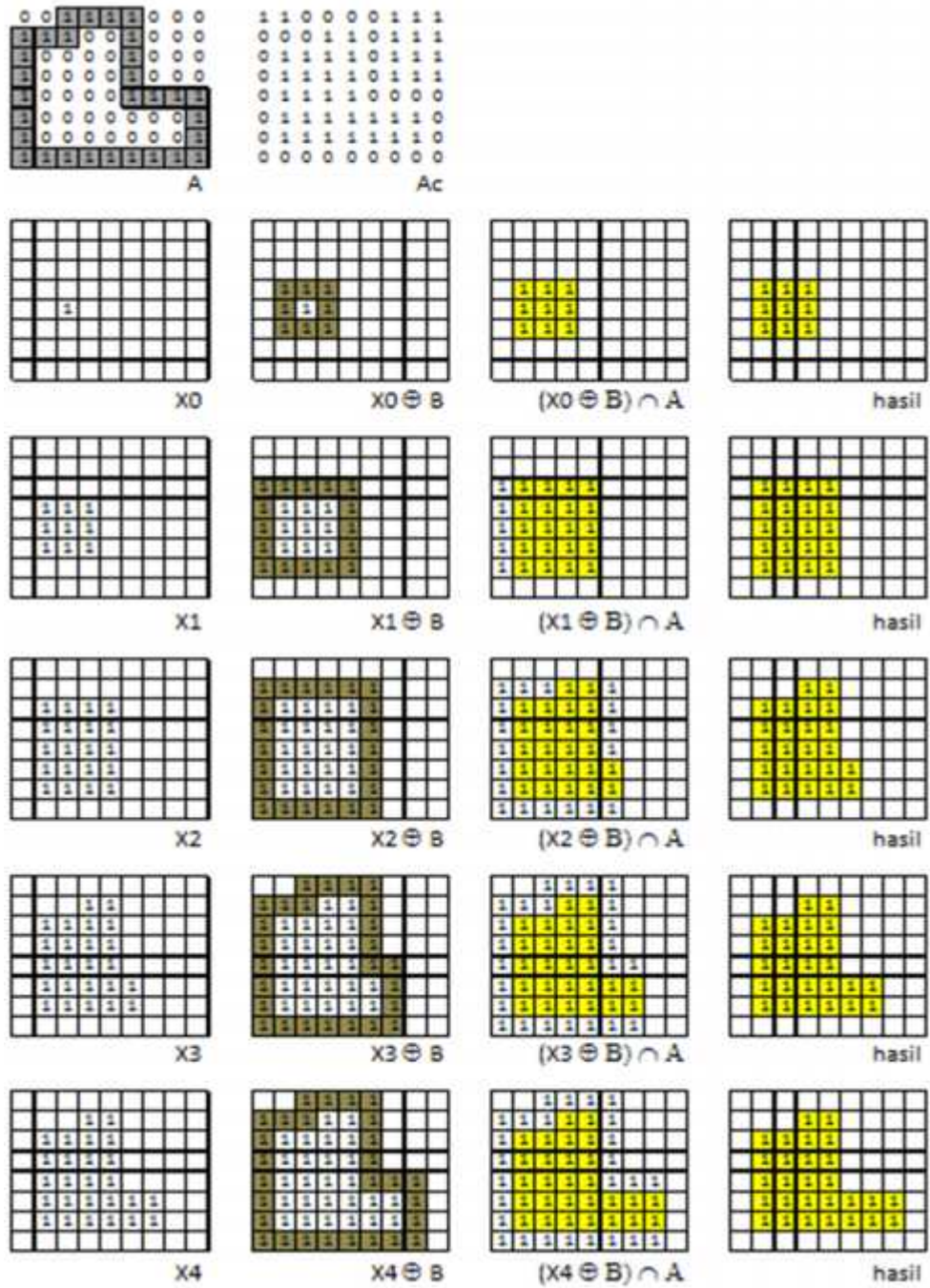
Berikut langkah-langkah dalam operasi *region filling* (Prasetyo, 2011) :

1. Dimulai dengan sebuah titik p di dalam boundary, tujuannya adalah untuk mengisi semua region dengan nilai 1.
2. Jika mengikuti konvensi bahwa semua titik non-boundary (background) diberi nilai 0, maka p harus diberi nilai 1 untuk memulai.
3. Prosedur di bawah ini adalah untuk mengisi region dengan nilai 1:

$$X_k = (X_{k-1} \oplus B) \cap A^c \quad k = 1, 2, 3, \dots$$

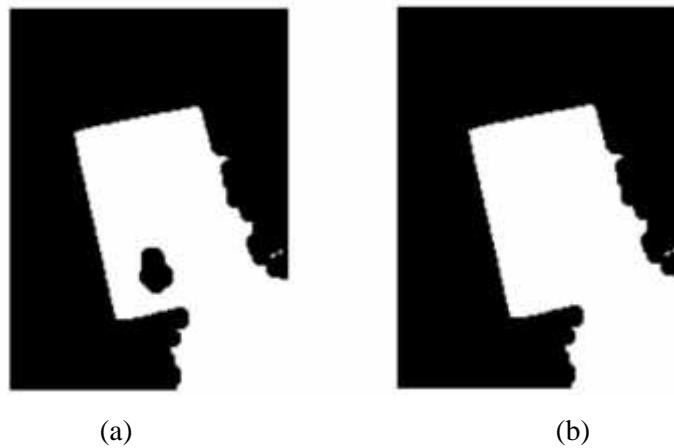
di mana $X_0 = p$ dan B adalah strel simetrik. Algoritma berhenti pada iterasi langkah ke k jika $X_k = X_{k-1}$.

Himpunan union dari X dan A mengisi daerah isian dan boundary-nya.



Gambar 2.16 Contoh visualisasi operasi *region filling* (Prasetyo, 2009)

Berikut contoh implementasi operasi *region filling* pada citra dapat dilihat pada gambar 2.17



Gambar 2.17 (a) Citra awal (b) Citra hasil *region filling* (Kadir & Susanto, 2013)

2.8 Ciri Gambar

Ciri merupakan suatu tanda yang khas, yang membedakan antara satu dengan yang lain. Gambar juga memiliki ciri yang dapat membedakannya dengan gambar yang lain. Masing-masing ciri gambar didapatkan dari proses ekstraksi ciri. Ciri – ciri dasar dari gambar yang dikutip oleh Fauzi (2013) dari Andrinato (2006) adalah sebagai berikut:

1. Warna
 - a. Ciri warna suatu gambar dapat dinyatakan dalam bentuk histogram dari gambar tersebut yang dituliskan dengan: $H(r,g,b)$, dimana $H(r,g,b)$ adalah jumlah munculnya pasangan warna r (red), g (green) dan b (blue) tertentu.
2. Bentuk
 - a. Ciri bentuk suatu gambar dapat ditentukan oleh tepi (sketsa), atau besaran moment dari suatu gambar. Pemakaian besaran moment pada ciri bentuk ini banyak digunakan orang dengan memanfaatkan nilai-nilai transformasi *fourier* dari gambar.

- b. Proses yang dapat digunakan untuk menentukan ciri bentuk adalah deteksi tepi, *threshold*, segmentasi dan perhitungan moment seperti (*mean*, *median* dan standard deviasi dari setiap lokal gambar).
3. Tekstur
 - a. Tekstur merupakan karakteristik intrinsik dari suatu citra yang terkait dengan tingkat kekasaran (*roughness*), granularitas (*granulation*), dan keteraturan (*regularity*) susunan struktural piksel. Aspek tekstural dari sebuah citra dapat dimanfaatkan sebagai dasar dari segmentasi, klasifikasi, maupun interpretasi citra.

2.9 Tekstur

Tekstur dapat digunakan untuk beberapa kepentingan. Secara umum, aplikasi tekstur dapat dibagi menjadi dua kategori (Kadir & Susanto, 2013). Pertama untuk kepentingan segmentasi, untuk segmentasi tekstur berperan untuk memisahkan antara objek yang satu dengan yang lainnya. Kedua adalah untuk klasifikasi, yang menggunakan fitur-fitur tekstur untuk klasifikasi objek. Beberapa contoh aplikasi tekstur dapat dilihat dibawah ini.

1. Inspeksi secara otomatis pada industri tekstil, pengecatan mobil, pemakaian karpet.
2. Analisis citra medis. Sebagai contoh, tekstur digunakan untuk klasifikasi penyakit paru-paru, diagnosis leukemia, dan pembedaan tipe-tipe sel darah putih.
3. Analisis penginderaan jarak jauh. Sebagai contoh, tekstur dipakai untuk kepentingan klasifikasi area tanah.

2.9.1 Kategori Tekstur

Berdasarkan keteraturan pengulangan pola pada objek, tekstur dikategorikan dalam dua bentuk: tekstur teratur dan tekstur tidak teratur. Tekstur buatan manusia cenderung tergolong pada katagori tekstur teratur, sedangkan tekstur alamiah bersifat tidak teratur.

Metode yang digunakan untuk memperoleh fitur tekstur dapat dibedakan menjadi tiga kategori: metode statistis, metode structural, dan metode spektral

(Kadir & Susanto, 2013). Metode statistis menggunakan perhitungan statistika dalam membentuk tekstur. Beberapa contoh yang termasuk dalam metode statistis adalah GLCM dan Tamura. Metode structural menjabarkan susunan elemen kedalam tekstur. Contoh metode structural adalah *Shape Grammar*. Metode spektral adalah metode yang didasarkan pada domain frekuensi-spasial. Contoh metode spectral adalah distribusi energi domain *Fourier, Gabor*, dan filter *Laws*.

2.9.2 Tekstur Berbasis Histogram

Untuk mendapatkan tekstur dengan cara sederhana adalah dengan mendasarkan pada histogram. Berikut fitur-fitur yang dapat dikenal secara statistic melalui histogram.

Fitur pertama yang dihitung secara statistis adalah rerata intensitas. Komponen fitur ini **dihitung** berdasarkan persamaan

$$m = \sum_{i=0}^{L-1} i \cdot p(i) \quad (2.10)$$

Dalam hal ini i adalah aras keabuan pada citra f dan $p(i)$ menyatakan probabilitaskemunculan i dan L menyatakan nilai aras keabuan tertinggi. Rumus diatas menghasilkan rerata kecerahan objek.

Fitur kedua berupa deviasi standar. Perhitungannya sebagai berikut:

$$\sigma = \sqrt{\sum_{i=1}^{L-1} (i - m)^2 p(i)} \quad (2.11)$$

Dalam hal ini, σ^2 dinamakan varians atau momen orde dua ternormalisasi karena $p(i)$ merupakan fungsi peluang. Fitur ini memberikan ukuran kekontrasan.

Fitur *skewness* merupakan ukuran ketidaksimetrisan terhadap rerata intensitas. Definisinya :

$$skewness = \sum_{i=1}^{L-1} (i - m)^3 p(i) \quad (2.12)$$

Skewness sering disebut sebagai momen orde tiga ternormalisasi. Nilai negatif menyatakan bahwa distribusi kecerahan condong ke kiri terhadap rerata dan nilai positif menyatakan bahwa distribusi kecerahan condong ke kanan terhadap rerata. Dalam praktik, nilai *skewness* dibagi dengan $(L-1)^2$ supaya ternormalisasi.

Deskriptor energi adalah ukuran yang menyatakan distribusi intensitas piksel terhadap jangkauan aras keabuan. Definisinya sebagai berikut:

$$energi = \sum_{i=0}^{L-1} [p(i)]^2 \quad (2.13)$$

Citra yang seragam dengan satu nilai aras keabuan akan memiliki nilai energi yang maksimum, yaitu sebesar 1. Secara umum, citra dengan sedikit aras keabuan akan memiliki energi yang lebih tinggi daripada yang memiliki banyak nilai aras keabuan. Energi sering disebut sebagai keseragaman.

Entropi mengindikasikan kompleksitas citra. Perhitungannya sebagai berikut:

$$entropi = - \sum_{i=0}^{L-1} p(i) \log_2(p(i)) \quad (2.14)$$

Semakin tinggi nilai entropi, semakin kompleks citra tersebut. Perlu diketahui, entropi dan energi berkecenderungan berkebalikan. Entropi juga merepresentasikan jumlah informasi yang terkandung di dalam sebaran data.

Properti kehalusan biasa disertakan untuk mengukur tingkat kehalusan/kekasaran intensitas pada citra. Definisinya sebagai berikut:

$$R = 1 - \frac{1}{1+\sigma^2} \quad (2.15)$$

Pada rumus di atas, σ adalah deviasi standar. Berdasarkan rumus di atas, *Smoothness* untuk mengukur kehalusan relatif dari intensitas citra. Nilai $R = 0$ untuk citra dengan intensitas yang konstan, sedangkan nilai R yang mendekati 1 untuk citra dengan intensitas yang bertebaran.

2.10 Labeling

Labeling adalah suatu proses pemberian label yang sama pada sekumpulan piksel pembentuk objek yang saling berdekatan pada suatu citra (Mozef, 2003). Objek yang berbeda memiliki label yang berbeda pula. Labeling termasuk pemrosesan citra tahap *intermediate level*. Labeling memiliki peran yang sangat penting pada pengolahan citra untuk mempermudah proses analisa bentuk dan pengenalan pola tahap *high level*.

2.11 Validasi Model

Validasi merupakan proses pengujian tingkat keakuratan data keluaran dari model yang dirancang dibandingkan dengan data lain yang lebih valid. Validasi dinyatakan dalam persamaan berikut ini (Sandra, 2006):

$$\text{Validasi (\%)} = \frac{A}{B} \times 100\% \quad (2.16)$$

Keterangan:

A = data keluaran model

B = data target