BAB II
TINJAUAN PUSTAKA DAN DASAR TEORI

2.1. Tinjauan Pustaka

Identifikasi sel goblet sebagai produsen utama musin dapat dilakukan dengan memberi warna pada musin melalui teknik pewarnaan *Alcian Blue-Periodic Acid Schiff* (AB-PAS). Teknik pewarnaan AB-PAS membuat musin berwarna biru, sehingga menimbulkan perbedaan warna antara sel goblet dan jaringan lainnya (Osho dkk., 2017), dan menjadikan perbedaan warna tersebut dapat digunakan dalam pengolahan citra untuk identifikasi sel goblet berdasarkan ciri warna (Van De Sande dkk., 2010). Identifikasi sel goblet berdasarkan ciri warna dapat dilakukan dengan ekstraksi ciri komponen *hue, saturation* dan *value* yang memilki keunggulan pada ruang warna merah, hijau dan biru (Zou dan Kim, 2007). Ekstraksi pada komponen *hue* sendiri jelas memperlihatkan perbedaan objek dan latar belakangnya (Adi, dkk., 2013).

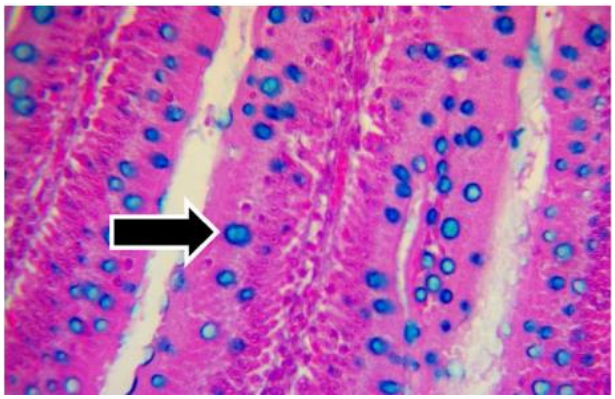
Objek yang telah diekstraksi dapat dipisahkan dengan latar belakangnya dengan cara segmentasi. Segmentasi dapat digunakan pada identifikasi sel dengan cara memisahkan suatu bagian sel dengan jaringan lainnya (Pamungkas dkk., 2012) (Ji dkk., 2016). Segmentasi dapat dilakukan dengan *multilevel thresholding*, seperti yang dilakukan dalam penelitian identifikasi ikan berdasarkan ciri dengan memisahkan objek mata ikan dengan bagian lainnya (Kumaseh dkk., 2013), sedangkan untuk mendapatkan hasil pemisahan objek yang baik diperlukan pemilihan nilai ambang yang tepat (Kotte dkk., 2016). Sel yang telah dipisahkan dari jaringannya seringkali ditemukan terjadi penumpukan sel atau adanya noise di sekitar sel sehingga perlu dilakukan operasi morfologi. Operasi morfologi dengan operasi erosi dan dilasi dapat membuat sel yang menumpuk menjadi terpisah dan menghilangkan noise di sekitar sel (Mirnasari dan Adi, 2013). Sel yang telah dilakukan operasi morfologi merupakan obyek yang diidentifikasi dengan jelas, sehingga dapat dapat dilakukan proses penghitungan jumlah sel.

Penghitungan jumlah sel dapat dilakukan melaui metode *multilevel thresholding* dan pelabelan (Pamungkas dkk., 2012). Proses segmentasi dengan

multilevel thresholding dan pelabelan juga digunakan sebagai metode penghitungan sel dengan akurasi tinggi yang dapat menghitung sel secara *real time* dengan tingkat akurasi sampai dengan 93 %, serta memeperlihatkan dengan jelas batas-batas antar sel (Ji. dkk., 2016). Identifikasi sel dengan ektraksi ciri, segmentasi, operasi morfologi, dan pelabelan untuk menghitung jumlah sel dapat dilakukan menggunakan bahasa pemrograman *Matlab.* Bahasa pemrograman *matlab* dapat digunakan untuk mengembangkan metode peghitungan sel dan membuat laporan terdokumentasi dari perhitungan yang dilakukan (Piccinini dkk., 2014).

2.2 Sel Goblet

Sel goblet merupakan produsen utama musin, terletak di dalam epitel usus halus dan memiliki peran utama untuk pencernaan dan penyerapan nutrisi (Contreras-Ruiz dkk., 2013). Jumlah sel goblet meningkat karena kontaminasi aflatoksin makanan (Jahanian dkk., 2016). Beberapa teknik preparasi telah ditetapkan untuk penghitungan sel goblet salah satunya teknik pewarnaan AB-PAS. Teknik pewarnaan AB-PAS yaitu metode dual-tujuan untuk mendeteksi sel-sel goblet baik netral maupun asam mucins (Spicer, 1960). Fotomikrograf yang menunjukkan sel goblet dengan AB-PAS terdapat pada Gambar 2.1.



Gambar 2.1. Fotomikrograf yang menunjukkan sel goblet (Osho dkk., 2017).

Prosedur pewarnaan AB-PAS meliputi jaringan usus yang telah dipotong diwarnai dengan *reagen Alcian Blue* dan *Periodic Acid-Schiff* (Luna, 1968) menggunakan larutan biru *Alcian* (1 g Alcian blue, pH 2,5, 3 mL / L asam asetat, dan 97 mL air suling) selama 30 menit. Kemudian jaringan usus tersebut direndam

menggunakan air keran selama 10 menit, selanjutnya dilakukan oksidasi dalam asam periodik (5 g / L) selama 5 menit, dan direndam lagi menggunakan air keran hangat selama 10 menit, selanjutnya dilakukan pewarnaan menggunakan reagen Colbert's Schiff sebagai noda counter selama 10 menit.

2.3 Citra Digital

Citra adalah gambar pada bidang dua dimensi f(x,y), dimana x dan y merupakan koordinat spasial, dan amplitudo dari f yang terdapat pada koordinat (x,y) merupakan intensitas citra pada titik tersebut. Citra digital terbentuk ketika x dan y serta nilai intensitas dari f seluruhnya terbatas (Prasetyo, 2011). Bentuk matriks citra digital adalah sebagai berikut:

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

Dengan:

M = jumlah baris, nilai x mulai dari 0 sampai M-1

N = jumlah kolom, nilai y mulai dari 0 sampai N-1

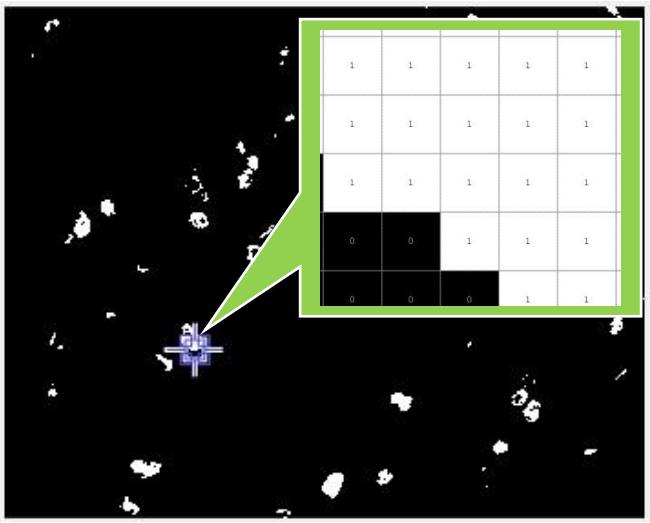
2.3.1 Citra Biner

Citra biner adalah citra dimana setiap piksel hanya dinyatakan dengan sebuah nilai dari dua buah kemungkinan yaitu nilai 0 dan 1 (Kadir dan Susanto, 2012). Nilai 0 merupakan warna hitam dan nilai 1 merupakan warna putih. Citra biner banyak dipakai dalam pemrosesan citra, misalnya untuk kepentingan memperoleh tepi bentuk maupun identifikasi suatu objek. Contoh citra bertipe biner ditunjukkan pada Gambar 2.2.

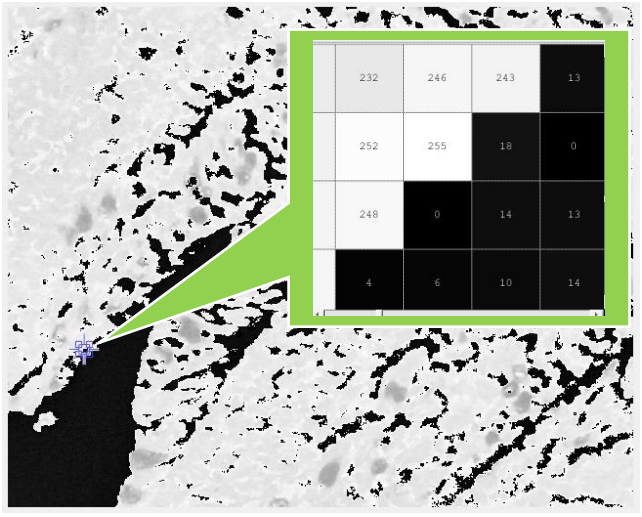
2.3.2 Citra *Grayscale*

Citra *grayscale* atau disebut dengan citra berskala keabuan merupakan citra dimana setiap piksel mewakili derajat keabuan dengan kisaran nilai antara 0 (hitam) sampai 255 (putih) (Kadir dan Susanto, 2012). Nilai pada kisaran nilai 0 sampai

255 memiliki arti bahwa setiap piksel berukuran 8 bit atau 1 byte. Citra bertipe *grayscale* ditunjukkan pada contoh Gambar 2.3.



Gambar 2.2 Citra bertipe biner.



Gambar 2.3 Citra bertipe *grayscale*

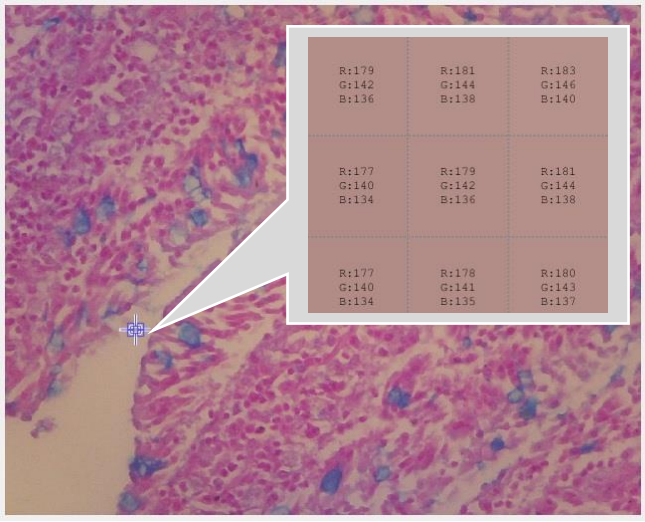
2.3.3 Citra RGB

Citra RGB merupakan citra dimana setiap pikselnya memiliki 3 komponen warna yaitu R (Red/merah), G (Green/hijau), dan B (Blue/biru). Setiap komponen warna menggunakan 8 bit (nilainya berkisar antara 0 sampai dengan 255). Hal ini akan memberikan kemungkinan total warna hingga 255 x 255 x 255 atau

16.581.375 warna (Kadir dan Susanto., 2012). Sebuah warna tidak hanya memiliki komposisi R, G, dan B tunggal. Pada Tabel 1 terlihat bahwa warna biru mempunyai R = 0, G = 0, dan B = 255. Namun, komposisi R = 1, G = 1, B = 255 juga berwarna biru. Sedangkan keadaan suatu citra RGB dan representasi warnanya disajikan pada Gambar 2.4.

Tabel 2.1 Warna dan Nilai Penyusunnya berkisar 0 sampai 255.

Warna	Komponen Red	Komponen Green	Komponen Blue
Putih	255	255	255
Merah	255	0	0
Hijau	0	255	0
Biru	0	0	255
Hitam	0	0	0

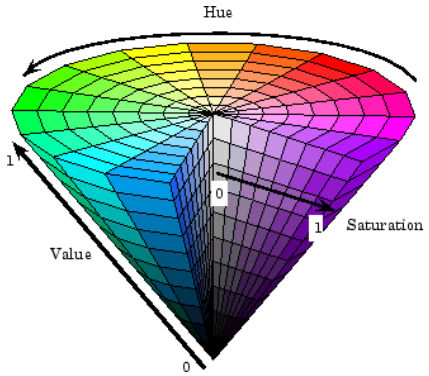


Gambar 2.4. Citra bertipe RGB

2.3.4 Citra HSV

Citra HSV merupakan citra warna dalam istilah *Hue, Saturation* dan *Value*. HSV memiliki karakteristik dimana terdapat warna-warna yang sama dengan yang ditangkap oleh indra manusia. Hue memiliki arti warna sebenarnya, seperti merah, violet, dan kuning dan digunakan menentukan kemerahan (*redness*), kehijauan (*greeness*) dan sebagainya. *Saturation* sering disebut kroma, yang merupakan

kemurnian atau kekuatan warna. Sedangkan *value* merupakan kecerahan yang dimiliki oleh warna. Nilai kecerahan tersebut berkisar antara 0-100 %. Apabila memiliki nillai 0 maka warnanya akan menjadi hitam, semakin besar nilai maka semakin cerah dan muncul variasi-variasi baru dari warna tersebut. Model HSV ditunjukkan pada Gambar 2.5.



Gambar 2.5 Model HSV (Smith, 1978).

2.3.5 Konversi Warna RGB ke HSV

Konversi warna RGB ke HSV adalah suatu proses perubahan dari citra bertipe RGB ke citra bertipe HSV (Kadir dan Susanto., 2012). Konversi warna RGB ke HSV dirumuskan pada persamaan 2.2 sebagai berikut:

$$r = \frac{R}{(R+G+B)}, g = \frac{G}{(R+G+B)}, b = \frac{B}{(R+G+B} \quad (2.2)$$

$$V = \max(r, g, b) \quad (2.3)$$

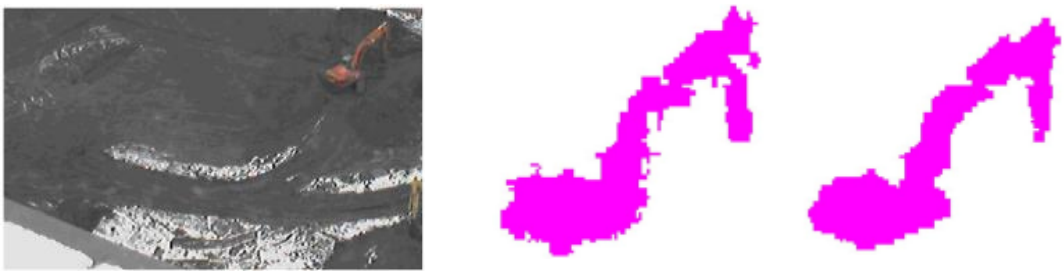
$$S = \begin{cases} 0, & jika\ V = 0 \\ 1 - \frac{\min(r,g,b)}{V}, & V > 0 \end{cases} \quad (2.4)$$

$$H = \begin{cases} 0, & jika\ S = 0 \\ \frac{60*(g-b)}{S*V}, & jika\ V = r \\ 60 * \left[2 + \frac{b-r}{S*V}\right], & jika\ V = g \\ 60 * \left[4 + \frac{r-g}{S*V}\right], & jika\ V = b \end{cases} \quad (2.5)$$

2.3.6 Ekstraksi Komponen *Hue, Saturation* dan *Value*

Ekstraksi Komponen *Hue*, *Saturation* dan *Value* merupakan suatu cara yang dapat digunakan dalam Teknik pengolahan citra guna menentukan suatu objek

berdasarkan ciri warna. Komponen *Hue*, *Saturation* dan *Value* memiliki keunggulan pada ruang warna merah, hijau dan biru dalam mendeteksi sebuah obyek (Zou dan Kim, 2007). Kelas data yang dihasilkan merupakan kelas data *double*. Gambar 2.6 berikut ini merupakan contoh ekstraksi Komponen *Hue* dan *Saturation.*



a. b. c.

Gambar 2.6 Contoh ekstraksi Komponen *Hue* dan *Saturation.* a. Citra Asli, b. Ekstraksi *hue,* c. Ekstraksi *saturation.*

2.4 Kelas Data *Matlab*

Pada *matlab* nilai piksel menggunakan koordinat *integer*, tetapi nilai piksel itu sendiri tidak dibatasi pada *integer* saja. Kelas data *numeric* terdiri dari *double, uint8, uint16, uint32, int8, int16, int32* dan *single*. Sedangkan kelas *char* dan *logical* adalah *char* dan *logical* (Prasetyo, 2011). Daftar kelas data matlab disajikan dalam Table 2.

Tabel 2.2 Kelas data pada *Matlab*

Nama	Penjelasan
Double	Presisi ganda, angka floating-point berkisar -10^{308} sampai 10^{308}
uint8	Unsigned 8-bit berkisar [0, 255] (1 byte per elemen)
uint16	Unsigned 16-bit berkisar [0, 65535] (2 byte per elemen)
uint32	Unsigned 32-bit berkisar [0, 4294967295] (4 byte per elemen)
int8	Signed 8-bit berkisar [-128, 127] (1 byte per elemen)
int16	Signed 8-bit berkisar [-32768, 32767] (2 byte per elemen)
int32	Signed 8-bit berkisar [-2147483648, 2147483647] (4 byte per elemen)
single	Presisi tunggal, angka floating-point berkisar -1038 sampai 1038 (4 byte per elemen)
char	Karakter (2 bit per elemen)
logical	Hanya nilai 0 atau 1 (1 byte per elemen)

Kelas *double* digunakan untuk semua komputasi numerik dalam matlab. Sedangkan kelas *uint8* sering ditemukan pada citra dari pembacaan perangkat penyimpanan sebagai citra 8-bit. Pada sebuah angka, 8 byte dibutuhkan untuk kelas

data *double*, 1 byte untuk kelas data *uint8* dan *int8*, 2 byte untuk kelas *uint16* dan *int16*, 8 byte untuk kelas *uint32*, *int32* dan *single*. Representasi *unicode* untuk kelas *char*, sedangkan nilai 0 dan 1 untuk kelas *logical* dengan 1 byte perelemen.

Kelas data matlab dapat dilakukan konversi antar kelas. Konversi kelas data *double* ke *uint8* sebagai berikut, jika nilai input kurang dari 0 maka dikonversi menjadi 0, jika nilai input lebih dari 1 maka dikonversi menjadi 255, sedangkan jika nilai input diantara 0 dan 1 maka konversinya dikalikan dengan 255.

2.5 Segmentasi Citra

Segmentasi citra adalah proses pengolahan citra untuk memisahkan objek yang dikehendaki dengan latar belakang (Destyningtias, dkk., 2010). Hasil yang diperleh dari segmentasi citra merupakan citra biner di mana objek yang dikehendaki memiliki warna putih atau bernilai 1, sedangkan latar belakang yang ingin dihilangkan memiliki warna hitam atau bernilai 0. Metode segmentasi citra antara lain deteksi diskontinuitas, *thresholding*, segmentasi berbasis *region*, segmentasi berbasis algoritma *watershed* dan deteksi garis menggunakan transformasi *hough* (Prasetyo, 2011).

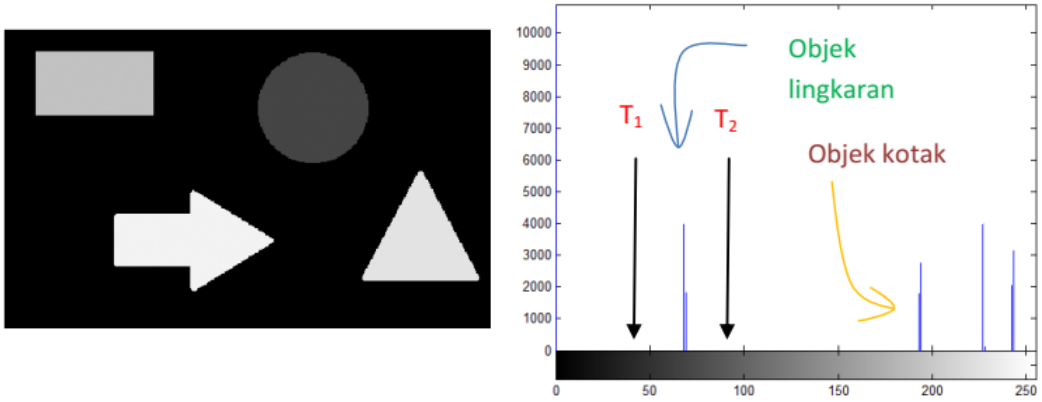
2.6 *Multilevel Thresholding*

Multilevel thresholding merupakan metode segmentasi citra dengan pengambangan level banyak. Citra dibagi menjadi beberapa bagian dengan menggunakan beberapa nilai ambang, hal ini dilakukan jika pada histogram terdapat beberapa puncak yang membedakan antara satu objek dengan objek yang lain, dimana nilai intensitas objek diketahui berada diantara T1 dan T2 dengan citra masukan merupakan kelas *uint8* (Putra, 2010). Sebagai contoh diperlihatkan pada Gambar 2.7.

Formula segmentasi dengan dua nilai ambang sebagai berikut:

$$f_B(i,j) = \begin{cases} 1, & T_1 \leq f_g(i,j) \leq T_2 \\ 0, & lainnya \end{cases} \qquad (2.3)$$

dengan $f_B(i,j)$ adalah citra biner, $f_g(i,j)$ adalah citra *grayscale*, T1 adalah nilai ambang minimal dan T2 adalah nilai ambang maksimal.



a. b.

Gambar 2.7 Contoh citra dengan beberapa puncak dan lembah yang terpisah. a. Citra "benda.png", b. Histogram.

2.7 Morfologi Citra

Morfologi citra merupakan alat yang digunakan untuk mengekstrak komponen citra dengan tujuan representasi dan deskripsi bentuk daerah, antara lain *boundaries, skeleton* dan *convex hull*. Kegunaan lain morfologi citra sebagai alat untuk *preprocessing* maupun *postprosessing* antara lain *filtering, thining* dan *pruning*. Dua jenis operasi morfologi citra dasar yaitu dilasi dan erosi (Presetyo, 2011). Prinsip operasi morfologi hanya bergantung pada urutan kemunculan dari piksel sedangkan nilai numerik dari piksel tidak berpengaruh. Hal tersebut menyebabkan teknik morfologi tepat digunakan dalam melakukan pengolahan citra biner dan citra berskala keabuan (Nelly dan Adi, 2013).

2.7.1 Erosi

Erosi merupakan operasi morfologi yang dapat digunakan untuk mengurangi piksel pada batas antar objek dalam suatu citra digital (Presetyo, 2011). Erosi antara A dan B dinyatakan A⊖B dan didefinisikan:

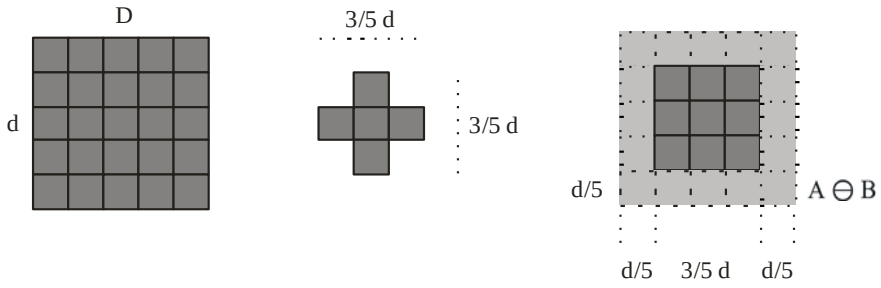
$$A \ominus B = \{z \mid (B)_z \subseteq A) \qquad (2.4)$$

Erosi A oleh B adalah kumpulan semua titik di mana B ditranslasikan oleh z di dalam isi A. B di sini adalah structuring elemen. Persamaan 2.4 tersebut dapat ditulis ulang dengan bentuk ekivalennya.

$$A \ominus B = \{z \mid (B)_z \cap A^c \neq \emptyset)) \qquad (2.5)$$

di mana A^c adalah komplemen A dan Ø adalah himpunan kosong. Operasi ini menggunakan aturan sebagai berikut: "Untuk citra *grayscale* maka nilai hasil operasi (*output pixel*) adalah nilai minimal yang diperoleh dari himpunan piksel tetangganya. Dalam binary image, jika ada piksel tetangga yang bernilai 0 maka output pixel akan diset menjadi 0".

Proses erosi dalam citra biner merupakan proses "pengecilan" atau "penipisan" objek. Erosi juga disebut sebagai *"morphological filtering"* dimana detil citra dengan ukuran lebih kecil dari strel menjadi hilang dari citra. Gambar 2.8 berikut ini merupakan gambaran proses erosi pada himpunan objek. Gambar 2.8 (c) menunjukan bahwa boundary yang solid merupakan batas penyeberangan *displacement* origin B yang menyebabkan *strel* berhenti menyeberang di dalam A.



Gambar 2.8 Proses erosi himpunan objek. a. Himpunan objek, b. Strel +, c. Erosi a oleh b.

2.7.2 Dilasi

Dilasi merupakan operasi morfologi yang akan menambahkan piksel pada batas antar objek dalam suatu citra digital (Presetyo, 2011). Dilasi antara A dan B dinyatakan A⊕B dan dirumuskan dengan persamaan.

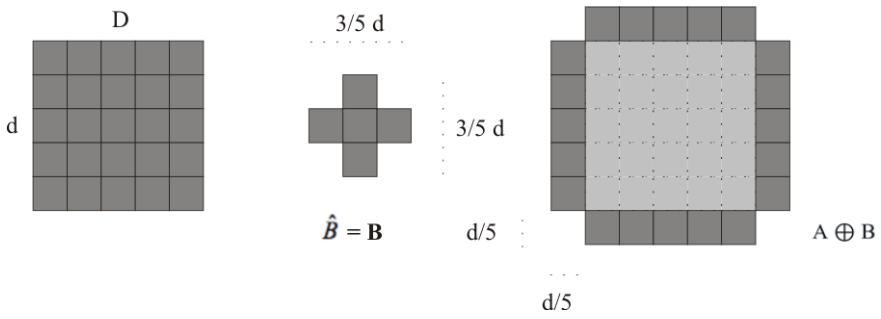
$$A \oplus B = \{z \mid (B)_z \cap A \neq \emptyset) \quad (2.6)$$

Persamaan ini didasarkan pada perefleksian B terhadap aslinya, dan pergeseran refleksi oleh Z. Dilasi A oleh B merupakan himpunan semua displacement Z, sebagaimana B dan A overlap paling sedikit satu elemen. Berdasarkan pada interpretasi tersebut, Persamaan 2.6 dapat ditulis kembali secara ekivalen sebagai.

$$A \oplus B = \{z \mid [(B)_z \cap A] \subseteq A) \quad (2.7)$$

Himpunan B berupa strel, sedangkan A merupakan himpunan objek citra yang terdilasi. Prinsip operasi dilasi sebagai berikut: “Untuk citra *grayscale* maka nilai hasil operasi (*output pixel*) adalah nilai maksimal yang diperoleh dari himpunan piksel tetangganya. Dalam binary image, jika ada piksel tetangga yang bernilai 1 maka output pixel akan diset menjadi 1”.

Proses dilasi dalam citra biner merupakan proses “penumbuhan” atau “penebalan” sesuai dengan bentuk strel yang digunakan. Dilasi dapat digunakan dalam perbaikan citra pada objek-objek yang teputus atau perbaikan pada kerusakan bentuk objek yang disebabkan resolusi yang kurang bagus atau noise yang terjadi saat pengambilan citra. Gambar 2.9 berikut ini merupakan gambaran proses dilasi pada himpunan objek.



Gambar 2.9 Proses dilasi himpunan objek. a. Himpunan objek, b. Strel +, c. Dilasi a oleh b.

Garis putus-putus seperti yang terlihat pada Gambar 2.8 (c) menunjukkan objek asli dan garis solid batas penyeberangan (pendilasian) yang merupakan *displacement* origin $\hat{B}$ oleh z. *Displacement* tersebut menyebabkan irisan antara $\hat{B}$ dan A menjadi kosong, sehingga semua titik pada dan di dalam boundary tersebut merupakan dilasi A oleh B.

2.7.3 *Strel* tipe *Disk*

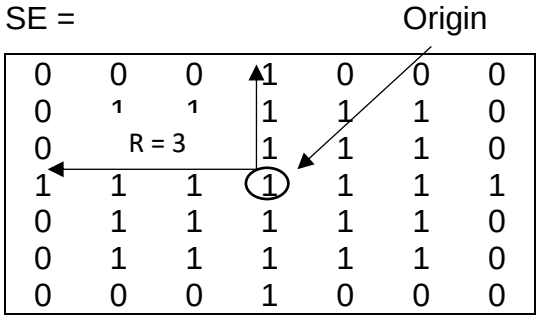
Strel (structuring element) merupakan refleksi dan translasi yang digunakan untuk memformulasikan operasi morfologi berupa himpunan kecil atau sub-image

yang digunakan untuk meneliti citra dalam pembelajaran propertinya (Presetyo, 2011). Tipe dan parameter fungsi *strel* dalam *toolbok* MATLAB disajikan pada Tabel 2.3 berikut ini.

Table 2.3 Tipe dan parameter fungsi *strel*

Tipe	Format Fungsi
Disk	SE = strel('disk',R,N)
Diamond	SE = strel('diamond', R)
Octagon	SE = strel('octagon',R)
Line	SE = strel('line',LEN,DEG)
Periodicline	SE = strel('periodicline',P,V)
Pair	SE = strel('pair', OFFSET)
Square	SE = strel('square',W)
Rectangle	SE = strel('rectangle',MN)
Arbitary	SE = strel('arbitary', NHOOD)

Strel tipe *Disk* merupakan *strel* berbentuk *disk flat*, dimana R menetapkan radius dan bernilai integer non-negatif. Nilai N berupa 0, 2, 4, atau 8. Jika N lebih besar dari 0, *strel* tipe *disk* diperkirakan dengan *strel* urutan *periodic-line* N, namun jika N = 0, tidak ada perkiraan yang digunakan dan anggota *strel* merupakan semua piksel dimana pusatnya tidak lebih dari *origin.* Nilai N menjadi 4, jika nilai N tidak dimasukkan. Pixel tengah elemen penataan disebut *origin* digunakan untuk mengidentifikasi piksel dalam gambar yang sedang diproses. Gambar 2.10 berikut ini merupakan ilutrasi *strel* tipe *disk.*



Gambar 2.10 Ilutrasi *strel* tipe *disk*

2.8 Pelabelan

Pelabelan dalam sebuah citra merupakan teknik untuk memberikan label dalam bentuk nomor pada objek yang berbeda di setiap objek yang ada dalam citra.

Pemberian label objek pada umumnya dioperasikan pada citra biner. Objek dalam citra merupakan sekumpulan pixel dengan intensitas warna yang sama dan saling terhubung. Pada citra biner, sekumpulan pixel dengan intensitas warna 1 dan saling terhubung dapat dianggap sebagai sebuah objek sedangkan intensitas warna 0 sebagai latar belakangnya. Sehingga dapat ditentukan nilai 0 merupakan pixel latar belakang, nilai 1 merupakan latar depan dan nilai 2,3, ... n adalah nomer objek, ilustrasi pada Gambar 2.11 merupakan matrix data citra biner yang terdiri dari 3 objek. (Sujito dan Yunus, 2016).

Gambar 2.11 Operasi pelabelan objek pada citra biner