

BAB 2

LANDASAN TEORI

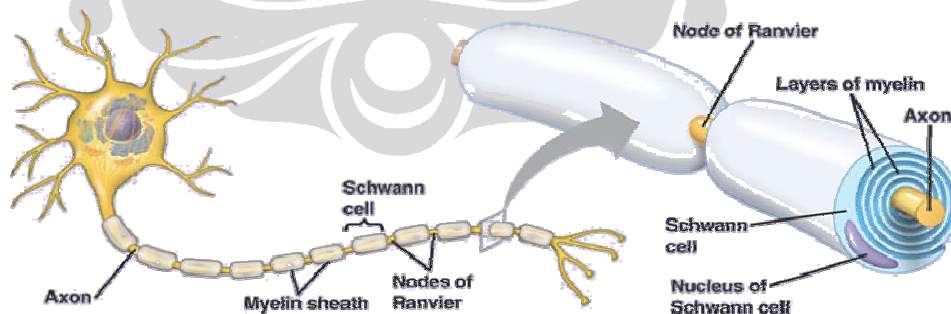
Bab ini membahas hasil yang didapatkan penulis dari studi literatur tentang jaringan saraf, penelitian jaringan saraf dengan isoform Na_v 1.8, citra digital, pemrosesan citra digital, dan penjelasan tentang proses segmentasi secara umum.

2.1 Jaringan Saraf

Bagian ini akan menjelaskan tentang jaringan saraf secara umum dan penelitian jaringan saraf dengan menggunakan Na_v 1.8.

2.1.1 Deskripsi Jaringan Saraf

Jaringan saraf terdiri dari neuroglia dan sel *schwann* (sel-sel penyokong), serta neuron (sel-sel saraf). Keduanya memiliki kaitan yang erat dan terintegrasi satu sama lainnya sehingga menjadi suatu unit fungsional. Bagian-bagian dari sel saraf ditunjukkan pada Gambar 2.1.



Gambar 2.1 Bagian-bagian Jaringan Saraf¹

Neuroglia (*nerve glue*) mengandung berbagai macam sel yang berfungsi untuk melindungi, menunjang, dan menjadi sumber nutrisi sel saraf pada otak dan *medula*

¹ Sumber: <http://fig.cox.miami.edu/~cmallery/>

spinalis. Sel *schwann* merupakan penyokong dan pelindung neuron dengan cara membentuk myelin maupun *neurolemma* pada saraf tepi.

Myelin merupakan suatu kompleks protein lemak berwarna putih yang mengisolasi serabut saraf. Walaupun begitu, selubung yang dibentuk myelin tidaklah mulus, sehingga terdapat celah-celah yang dinamakan dengan *nodus ranvier*. Selain itu, tidak semua serabut saraf ditutupi oleh selubung myelin. Serabut saraf yang bermyelin dinamakan dinamakan massa putih (*substansia alba*), sedangkan yang tidak bermyelin dinamakan massa kelabu (*substansia grisea*).

Transmisi impuls saraf di sepanjang serabut bermyelin lebih cepat dari transmisi di sepanjang serabut tak bermyelin, karena impuls berjalan dengan cara meloncat dari nodus ke nodus yang lain di sepanjang selubung myelin. Cara transmisi seperti ini dinamakan konduksi saltatorik.

2.1.2 Penelitian Jaringan Saraf dengan Isoform Na_v 1.8

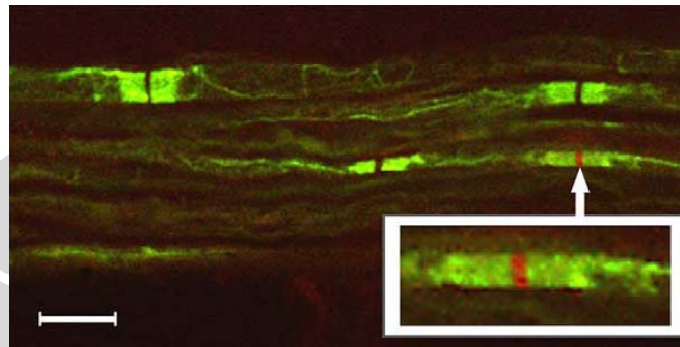
Penggunaan gigi sebagai model sistem untuk mempelajari mekanisme sensor dan sakit sudah *well-established* karena contoh gigi manusia sangat banyak tersedia. Hal ini disebabkan ekstraksi gigi, baik gigi normal maupun berpenyakit, dilakukan secara rutin dan merupakan hal yang lumrah.

Untuk meneliti jaringan saraf pada gigi, serabut saraf gigi diekstrak dan diberi perlakuan khusus. Salah satunya adalah dengan diberikan *fluorophore* merah (IR Na_v 1.8) dan *fluorophore* hijau (IR *caspr*). Serabut tersebut kemudian difoto dengan menggunakan mikroskop *fluorescent*. Terhadap suatu sampel saraf dilakukan dua kali pengambilan gambar, yaitu dengan menggunakan filter merah dan setelah itu dengan memakai filter hijau.

Filter hijau digunakan untuk melihat struktur saraf yang dapat mengambil zat warna (*fluorophore*) hijau, sedangkan filter merah digunakan untuk melihat struktur saraf yang dapat mengambil *fluorophore* merah. Struktur saraf yang menyerap *fluorophore* hijau adalah daerah paranodus (bagian tepi dari *nodus ranvier*), sedangkan struktur saraf yang mengikat *fluorophore* merah adalah celah antar bagian paranodus. Untuk

lebih jelasnya dapat dilihat pada Gambar 2.2. Kotak pada kanan bawah di gambar tersebut menunjukkan nodus dengan IR $\text{Na}_v1.8$ pada skala yang lebih besar.

Pada hasil pengambilan gambar dengan menggunakan filter hijau, ada kemungkinan bahwa yang berwarna hijau bukan hanya bagian paranodus. Akan tetapi, tingkat kehijauan bagian paranodus tetap relatif lebih tinggi dibandingkan dengan bagian lainnya.



Gambar 2.2 Gambar Paranodus dan Celah Antara Paranodus pada Jaringan Saraf²

Beberapa penelitian menunjukkan bahwa serabut saraf, khususnya yang bermyelin, memiliki konsentrasi IR $\text{Na}_v 1.8$ yang signifikan pada *nodus ranvier*-nya. Selain itu, serabut dengan konsentrasi IR $\text{Na}_v 1.8$ memediasi konduksi saltatorik secara cepat dan juga mendukung propagasi aksi potensial pada frekuensi tinggi. Dengan begitu, dapat dikatakan bahwa semakin tinggi konsentrasi IR $\text{Na}_v 1.8$ pada jaringan saraf, maka semakin sensitif jaringan tersebut terhadap rasa sakit atau nyeri (Henry, et al., 2005).

2.2 Citra Dijital dan Pemrosesannya

Citra dapat didefinisikan sebagai fungsi dua dimensi $f(x,y)$, di mana x dan y merupakan koordinat spasial. Besar nilai f pada titik yang berupa pasangan koordinat (x,y) , merepresentasikan intensitas tingkat keabuan citra pada titik tersebut. Citra yang memiliki nilai x , y , dan besaran f yang berhingga dan diskrit dinamakan sebagai citra digital (Gonzalez, et al., 2003).

²Sumber: (Henry, et al., 2005)

Jika $f(x,y)$ adalah suatu citra digital yang memiliki M baris dan N kolom, sehingga dapat dikatakan bahwa ukuran citra adalah $M \times N$. Representasi fungsi citra dalam bentuk matriks dapat dituliskan sebagai berikut:

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

Bagian kanan dari persamaan tersebut merupakan citra digital sesuai dengan definisi. Setiap elemen dari *array* ini dinamakan elemen citra, elemen gambar, piksel, atau pel. Istilah citra dan piksel akan digunakan pada penelitian ini untuk menandakan citra digital dan elemennya.

2.2.1 Jenis-Jenis Citra

Pada bagian ini akan dijelaskan jenis-jenis citra yang digunakan oleh penulis dalam penelitian ini.

2.2.1.1 Citra Biner

Citra biner, yang juga dapat disebut dengan *bi-level* or *two-level*, adalah citra yang setiap elemennya memiliki dua kemungkinan nilai (bernilai *boolean*), yaitu 0 atau 1. Pada penelitian ini, citra biner yang komponennya bernilai 0 memiliki warna hitam, dan yang bernilai 1 memiliki warna putih (Gonzalez, et al., 2003).

Untuk memperoleh citra biner dari suatu citra non-biner, biasanya dilakukan operasi *thresholding* terhadap citra non-biner tersebut. Hal ini perlu dilakukan karena terdapat beberapa operasi pemrosesan yang biasanya diaplikasikan pada citra yang berjenis biner, salah satunya adalah operasi morfologi.

2.2.1.2 Citra Grayscale

Pada citra *grayscale*, setiap piksel pada koordinat (x,y) memiliki suatu nilai yang bersifat tunggal, yang berarti nilai tersebut merupakan satu-satunya informasi intensitas yang dikandung oleh piksel tempatnya berada (Gonzalez, et al., 2002).

Intensitas dari sebuah piksel diekspresikan sebagai jangkauan antara nilai minimum dan maksimum yang inklusif. Pada citra *grayscale*, jumlah kemungkinan intensitas yang dimiliki oleh masing-masing piksel bergantung pada ukuran yang diinginkan. Umumnya, intensitas tersebut diakomodasi dengan nilai sebesar 8 bit walau tidak tertutup kemungkinan digunakan nilai yang lain seperti 16 bit.

Jika nilai 8 bit yang digunakan, maka berarti terdapat 256 kemungkinan intensitas tingkat keabuan yang berbeda untuk suatu pikselnya. Nilai ini menjangkau mulai dari intensitas terendah 0 (*total absence*, hitam) sampai tertinggi 255 (*total presence*, putih).

2.2.1.3 Citra Berwarna

Secara sederhana, citra berwarna dapat didefinisikan sebagai citra yang mengandung informasi warna pada tiap pikselnya. Untuk penggunaan pemrosesan citra, diperlukan tiga saluran warna untuk tiap piksel, yang akan diinterpretasikan sebagai koordinat pada ruang warna. Contohnya adalah pada ruang warna RGB yang sering digunakan untuk tampilan di komputer.

Untuk representasi pada matriks, ruang warna dapat dianggap sebagai *array* berukuran $M \times N \times 3$, dengan M adalah jumlah baris, N adalah jumlah kolom, dan 3 menandakan jumlah saluran warna. Secara sederhana, dapat dikatakan bahwa tiap saluran dapat dianggap sebagai suatu komponen *grayscale*, sehingga dapat dianalogikan bahwa suatu citra berwarna terdiri atas tiga buah komponen *grayscale* (Gonzalez, et al., 2003).

2.2.2 Konsep *Connected Component*

Konektivitas antara piksel-piksel adalah konsep fundamental yang menyederhanakan definisi dari berbagai macam konsep citra digital, seperti batas dan sisi. Untuk mengatakan bahwa dua buah piksel tersambung, maka harus ditetapkan bahwa piksel-piksel tersebut bertetangga dan tingkat keabuannya memenuhi kriteria yang dispesifikasikan. Contohnya, pada citra biner dengan nilai 0 dan 1, dua piksel bisa berada dalam *4-neighbours*, tetapi mereka dikatakan terkoneksi hanya jika mereka memiliki nilai yang sama.

Misalkan V adalah himpunan dari nilai tingkat keabuan yang digunakan untuk mendefinisikan *adjacency*. Pada citra biner, $V = \{1\}$ jika kita merujuk pada *adjacency* pada piksel-piksel dengan nilai 1. Pada citra *grayscale*, idenya adalah sama, tetapi himpunan V biasanya mengandung lebih banyak elemen. Sebagai contoh, pada *adjacency* piksel-piksel dengan kemungkinan nilai tingkat keabuan 0 sampai 255, himpunan V adalah *subset* dari 256 nilai tersebut.

Terdapat dua tipe dari *adjacency*:

- a. *4-adjacency*: dua piksel p dan q dengan nilai dari V adalah *adjacent* jika q berada dalam set $N_4(p)$
- b. *8-adjacency*: dua piksel p dan q dengan nilai dari V adalah *adjacent* jika q berada dalam set $N_8(p)$

Misalkan S merepresentasikan sebuah *subset* dari sebuah citra. Dua piksel p dan q dikatakan terhubung dalam S jika terdapat sebuah *path* antara mereka yang mengandung semua piksel dalam S . Untuk suatu piksel p dalam S , himpunan piksel-piksel yang terkoneksi kepada p dinamakan sebagai *connected component* dari S . Jika hanya terdapat satu *connected component*, maka S dinamakan sebagai sebuah *connected set*.

2.2.3 Operasi Morfologi

Morfologi adalah alat untuk mengekstraksi komponen citra yang berguna dalam membantu meningkatkan representasi dan deskripsi dari bentuk wilayah, seperti wilayah batas, kerangka, dan *convex hull*. Operasi ini biasanya dilakukan untuk *pre-processing* atau *post-processing*, seperti *filtering*, *thinning*, dan *pruning*.

Salah satu pemrosesan yang disarankan pada citra biner adalah operasi morfologi. Salah satu keuntungan dari operasi ini dibandingkan dengan pada pendekatan frekuensi adalah metode morfologi memberikan hasil langsung berupa citra biner. Sedangkan jika pendekatan frekuensi digunakan pada citra biner, akan diperoleh hasil sementara yang berupa citra *grayscale*, dan memerlukan fungsi *thresholding* untuk

mengembalikannya kepada bentuk biner. Hal ini tentunya merupakan sebuah biaya tersendiri dalam melakukan pemrosesan citra.

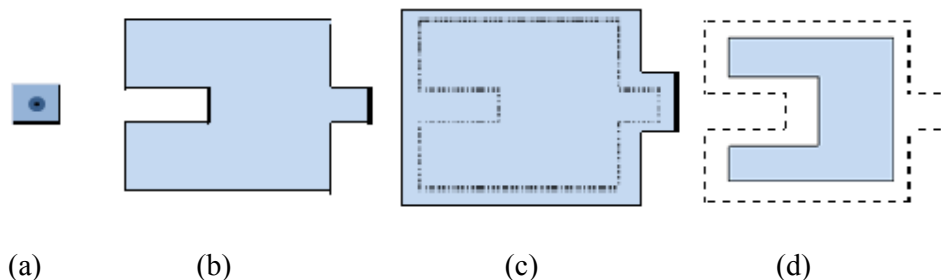
Pada operasi morfologi, dibutuhkan minimal sebuah citra dan sebuah *structuring element*. Citra dan *structuring element* ini akan dianggap sebagai himpunan dalam pengoperasiannya. Dua operasi yang fundamental untuk pemrosesan morfologi adalah operasi dilasi dan erosi. Pada saat dilakukan operasi dilasi atau erosi pada sebuah citra A dengan *structural element* B, keduanya akan dianggap sebagai himpunan di koordinat kartesius berdimensi dua. Citra A yang didilasi dengan *structural element* B dinamakan dilasi A oleh B, dan ditandai dengan notasi $A \oplus B$. Citra A yang dierosi dengan *structural element* B dinamakan erosi A oleh B, dan ditandai dengan notasi $A \ominus B$.

Opening pada citra A oleh *structuring element* B, ditandai dengan $A \circ B$, merupakan operasi dilasi yang diterapkan pada himpunan hasil erosi A oleh B. Persamaan untuk operasi *opening* dapat dilihat pada persamaan 2.1

$$A \circ B = (A \ominus B) \oplus B \quad (2.1)$$

Operasi *closing* merupakan urutan operasinya berkebalikan dengan *opening*. *Closing* pada citra A oleh *structuring element* B, ditandai dengan $A \bullet B$, merupakan operasi erosi yang diterapkan pada himpunan hasil dilasi A oleh B. Persamaan untuk operasi *closing* dapat dilihat pada persamaan 2.2.

$$A \bullet B = (A \oplus B) \ominus B \quad (2.2)$$



Gambar 2.4 Operasi Morfologi (a) Structural element (b) Citra awal (c) Citra pasca dilasi (d) Citra pasca erosi³

³ Sumber: (Gonzalez, et al., 2002)

Gambar 2.4 menunjukkan contoh dari operasi morfologi erosi dan dilasi. Gambar 2.4(a) merupakan *structural element* yang digunakan untuk melakukan operasi morfologi. Gambar 2.4(b) menunjukkan citra yang ingin dilakukan operasi morfologi terhadapnya. Gambar 2.4(c) adalah citra 2.4(b) yang telah mengalami operasi dilasi menggunakan *structural element* 2.4(a). Gambar 2.4(d) menunjukkan citra 2.4(b) yang telah mengalami operasi erosi menggunakan *structural element* 2.4(a).

2.2.4 Median Filter

Smoothing filter sering digunakan untuk melakukan *blurring* dan mengurangi *noise*. *Blurring* umum digunakan pada tahap *preprocessing*, seperti penghilangan detil kecil dari sebuah citra sebelum ekstraksi obyek yang lebih besar, dan menjembatani jurang kecil dari garis atau kurva. Pengurangan *noise* dapat dicapai oleh *blurring* dengan sebuah *linear filtering* dan juga dengan *nonlinear filtering*.

Median filter bekerja dengan cara mengganti nilai sebuah piksel dengan nilai median dari tingkat keabuan dalam ketetangaan piksel tersebut (nilai asli dari piksel tersebut termasuk dalam komputasi). *Median filter* merupakan metode yang cukup populer karena untuk tipe *noise* acak tertentu menyediakan kemampuan pengurangan *noise* yang sangat baik dengan *blurring* yang lebih sedikit dibandingkan *linear smoothing filter* pada ukuran yang sama. *Median filter* efektif dalam kasus *impulsive noise*, yang dinamakan *salt-and-pepper noise* karena penampilannya seperti titik-titik hitam dan putih yang ada pada citra.

Median, ξ , dari himpunan nilai-nilai adalah sedemikian sehingga separuh dari nilai-nilai pada himpunan tersebut adalah lebih kecil atau sama dengan ξ , dan setengahnya lagi lebih besar atau sama dengan ξ . Untuk melakukan *median filtering* pada sebuah titik pada citra, kita pertama-tama mengurutkan nilai piksel pada titik tersebut dan tetangganya, menentukan mediannya, dan meng-assign-nya kepada piksel tersebut. Sebagai contoh, median pada konsep 3 x 3 ketetangaan adalah nilai terbesar kelima, pada konsep 5 x 5 ketetangaan adalah nilai terbesar ke-13, dan seterusnya.

Prinsip dari *median filter* adalah untuk menjadikan titik dari piksel-piksel dengan *gray level* yang berbeda agar sama dengan tetangganya. Penghitungan dari *median filter* dapat dilihat pada persamaan 2.3

$$\hat{f}(x, y) = \mathbf{median}_{(s,t) \in \mathcal{S}_{x,y}} \{g(s, t)\} \quad (2.3)$$

2.2.5 Deteksi Sisi (*Edge Detection*)

Secara intuisi, sebuah sisi adalah sebuah himpunan dari piksel-piksel berhubungan yang terletak pada batas antara dua buah wilayah. Akan tetapi, definisi dari sisi yang sesungguhnya terkait dengan konsep pengukuran transisi tingkat keabuan.

Magnitude dari turunan pertama pada citra dapat digunakan untuk mendeteksi kehadiran dari sebuah sisi pada sebuah titik pada citra (misalnya untuk menentukan apakah sebuah titik terdapat pada sebuah lembah). Tanda dari turunan kedua dapat digunakan untuk menentukan apakah sebuah piksel sisi terletak pada sisi gelap atau terang dari sebuah sisi.

Untuk melakukan operasi fungsi turunan secara matematis tidaklah mudah. Oleh karena itu, untuk mendapatkan nilai turunan biasanya digunakan operasi yang menggunakan jendela ketetanggaan seperti yang ditunjukkan oleh Gambar 2.5. Titik Z_5 sebagai titik pusat merupakan piksel yang ingin diketahui nilai turunannya.

Z_1	Z_2	Z_3
Z_4	Z_5	Z_6
Z_7	Z_8	Z_9

Gambar 2.5 Jendela Ketetanggaan Piksel

2.2.5.1 Sobel Edge Detector

Sobel edge detector menggunakan *masks* pada Gambar 2.6(a) untuk mendekati turunan pertama G_x dan G_y secara digital. G_x merupakan turunan dilihat dari absis, dan G_y merupakan turunan dilihat dari ordinat. Dengan kata lain, gradien dari pusat titik dalam sebuah ketetanggaan dengan detektor Sobel dapat dihitung sebagai berikut:

$$g = [G_x^2 + G_y^2]^{\frac{1}{2}}$$

$$= \{ [(Z_7 + 2Z_8 + Z_9) - (Z_1 + 2Z_2 + Z_3)]^2 + [(Z_3 + 2Z_6 + Z_9) - (Z_1 + 2Z_6 + Z_7)]^2 \}^{1/2}$$

Dapat dikatakan bahwa sebuah piksel pada lokasi (x,y) adalah sebuah piksel sisi jika $g \geq T$ pada lokasi tersebut, dengan T adalah *threshold* yang dispesifikasikan.

2.2.5.2 Prewitt Edge Detector

Parameter dari fungsi ini adalah identik dengan dengan parameter detektor Sobel. Detektor Prewitt sedikit lebih sederhana untuk diimplementasikan secara komputasional dibandingkan dengan detektor Sobel, tetapi cenderung memproduksi hasil yang lebih kasar. Hal ini disebabkan nilai koefisien 2 pada detektor Sobel yang menghasilkan efek *smoothing*. Pada detektor Prewitt, digunakan *masks* pada Gambar 2.6(b) untuk mendekati turunan pertama G_x dan G_y secara digital. Gradien dari pusat titik dalam sebuah ketetanggaan dengan detektor Prewitt dapat dihitung sebagai berikut :

$$g = [G_x^2 + G_y^2]^{\frac{1}{2}}$$

$$= \{ [(Z_7 + Z_8 + Z_9) - (Z_1 + Z_2 + Z_3)]^2 + [(Z_3 + Z_6 + Z_9) - (Z_1 + Z_6 + Z_7)]^2 \}^{1/2}$$

-1	-2	-1	-1	0	1	-1	-1	-1	-1	0	1
0	0	0	-2	0	2	0	0	0	-1	0	1
1	2	1	-1	0	1	1	1	1	-1	0	1
G_x			G_y			G_x			G_y		
(a)						(b)					

Gambar 2.6 Masks untuk Deteksi Sisi (a) Masks Deteksi Sobel (b) Masks Deteksi Prewitt

2.2.5.3 Canny Edge Detector

Detektor Canny memiliki algoritma sebagai berikut:

1. Citra dihaluskan menggunakan *gaussian filter* dengan standar deviasi σ untuk mengurangi *noise*.

2. Gradien lokal, $g(x,y) = [G_x^2 + G_y^2]^{1/2}$, dan arah sisi, $\alpha(x,y) = \tan^{-1}(G_x/G_y)$, dikomputasi pada tiap-tiap titik. Salah satu dari detektor Sobel, Prewitt, atau Roberts dapat digunakan untuk menghitung G_x dan G_y . Sebuah piksel sisi didefinisikan sebagai titik yang kekuatannya adalah maksimum lokal dalam arah dari gradien.
3. Titik sisi yang ditentukan dalam (2) menghasilkan kurva yang menaik pada citra *gradient magnitude*. Algoritma ini lalu menelusuri bagian batas dari kurva ini dan menset semua piksel yang bukan merupakan puncak kurva, sehingga memberikan garis tipis pada *output*. Oleh karena itu proses ini dinamakan *nonmaximal suppression*. Piksel-piksel pada kurva kemudian di-*threshold* menggunakan dua *threshold*, T_1 dan T_2 , dengan $T_1 < T_2$. Piksel-piksel pada kurva dengan nilai lebih besar dari T_2 dinamakan piksel sisi kuat, dan ruang di antaranya dinamakan piksel sisi lemah.
4. Pada tahap akhir diterapkan algoritma *edge linking* dengan menyatukan piksel lemah yang *8-connected* dengan piksel kuat.

2.2.6 Fungsi Transformasi Intensitas

Seperti yang telah dijelaskan sebelumnya, suatu citra terdiri dari piksel-piksel yang memiliki suatu intensitas tertentu. Intensitas yang dimiliki oleh piksel-piksel tersebut tentunya beragam antara satu dengan yang lainnya, dan dapat dibuat suatu diagram persebaran intensitas citra yang dinamakan histogram intensitas. Pada citra *grayscale*, histogram intensitas ini dinamakan dengan histogram tingkat keabuan.

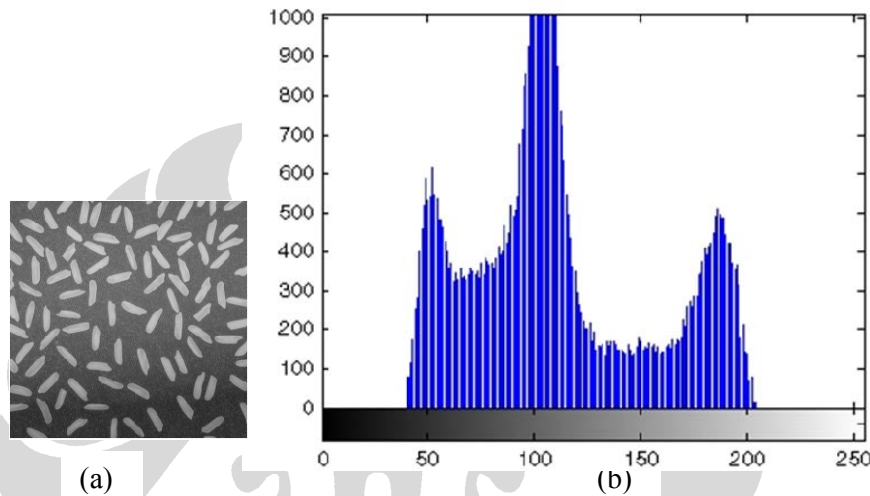
Dengan menggunakan histogram ini, penilaian terhadap karakteristik citra akan menjadi lebih mudah. Misalnya, kita dapat mengetahui apakah suatu citra memiliki lebih banyak piksel berintensitas tinggi (terlihat terang), lebih banyak piksel berintensitas rendah (terlihat gelap), memiliki kontras yang tinggi (persebaran intensitas merata), ataupun memiliki kontras yang rendah (persebaran intensitas mengumpul pada suatu bagian).

Pada intensitas tersebut dapat dilakukan suatu proses transformasi yang bertujuan untuk mengubah citra sehingga memiliki karakteristik yang diinginkan. Proses

transformasi intensitas T terhadap suatu intensitas s pada citra awal dan menghasilkan intensitas r pada citra hasil dapat diekspresikan sebagai berikut:

$$s = T(r)$$

Berhubung fungsi transformasi ini hanya bergantung pada nilai intensitas dan tidak pada posisi koordinatnya, maka korespondensi intensitas pada titik (x,y) dalam citra tidak dituliskan pada persamaan tersebut.



Gambar 2.7 Fungsi Transformasi Intensitas (a) Contoh Citra *Grayscale* (b) Histogram Intensitas dari Citra *Grayscale*⁴

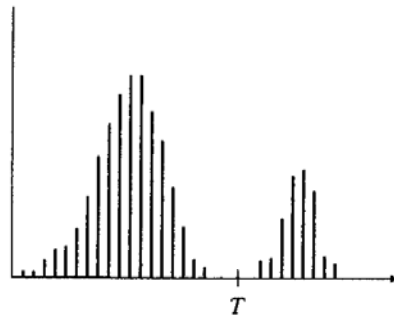
Contoh dari histogram dapat dilihat pada Gambar 2.7(b), yang merupakan representasi intensitas bentuk diagram dari citra yang ditunjukkan Gambar 2.7(a). Sumbu absis menunjukkan tingkat intensitas, yaitu dari 0 sampai dengan 255, sumbu ordinat menunjukkan jumlah piksel. Terlihat pada histogram tersebut bahwa banyak piksel pada citra yang memiliki intensitas keabuan di sekitar nilai 100.

2.2.7 Thresholding

Thresholding merupakan salah satu metode segmentasi yang merupakan bagian dari teknik *intensity slicing* (atau terkadang dinamakan *density slicing*) (Csetverikov, 2004). Misalkan histogram tingkat keabuan yang ditunjukkan pada Gambar 2.8 berkorespondensi dengan sebuah citra $f(x,y)$, yang dibentuk dari objek terang pada

⁴ Sumber: MATLAB® *Help Menu*

latar gelap, sedemikian sehingga piksel obyek dan latar memiliki tingkat keabuan yang dikelompokkan menjadi dua mode dominan. Salah satu cara yang jelas untuk mengekstrak obyek dari latar adalah dengan memilih *threshold* T yang memisahkan mode ini. Titik (x,y) dikatakan sebagai titik obyek jika memiliki $f(x,y) > T$, dan dinamakan latar jika nilai $f(x,y) \leq T$.



Gambar 2.8 Histogram Intensitas dan Nilai *Threshold*⁵

Thresholding dapat dilihat sebagai operasi yang melibatkan pengecekan terhadap fungsi T dengan bentuk

$$T = T[x, y, p(x,y), f(x,y)]$$

Notasi $f(x,y)$ menandakan tingkat keabuan dari titik (x,y) , dan $p(x,y)$ menandakan beberapa properti lokal dari titik tersebut (misalnya rata-rata tingkat keabuan dari ketetanggaan yang berpusat pada (x,y)). Ketika T bergantung hanya pada $f(x,y)$, *threshold* dinamakan global. Jika T bergantung pada baik $f(x,y)$ dan $p(x,y)$, maka *threshold* dinamakan lokal. Jika T bergantung pada koordinat spasial x dan y , *threshold* dinamakan *dynamic* atau *adaptive*.

Penentuan nilai T bisa dilakukan secara manual, ataupun melalui metode terotomatisasi dengan cara iterasi. Salah satu metode terotomatisasi yang sering digunakan adalah dengan menggunakan algoritma Otsu (Gonzalez, et al., 2003). Algoritma ini mencari *threshold* dengan cara memaksimalkan variansi antara kelas pada histogram tingkat keabuan citra masukan.

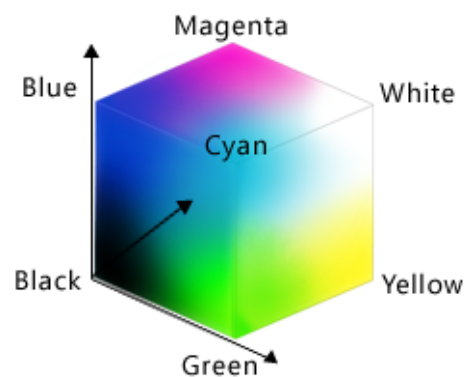
⁵ Sumber: (Gonzalez, et al., 2002)

Metode yang sering digunakan dalam pemrosesan citra adalah *threshold* global. Akan tetapi, metode ini memiliki kelemahan jika tingkat iluminasi dari latar adalah tidak seragam. Pada situasi ini, tindakan yang biasanya diambil adalah dengan melakukan *threshold* lokal, contohnya dengan *pre-processing* pada citra untuk menanggulangi masalah iluminasi, setelah itu barulah melakukan *threshold* global pada citra tersebut.

2.3 Ruang Warna RGB

Ruang warna RGB adalah ruang warna yang komponennya berupa triplet yang masing-masing berkorespondensi dengan komponen merah, hijau, dan biru dari sebuah citra RGB pada lokasi spasial yang spesifik.

Sebuah citra RGB dapat dilihat sebagai suatu *stack* dari tiga buah citra *grayscale*, yang masing-masing *stack* berisi sebuah komponen warna. Jadi, suatu piksel pada citra RGB dapat memiliki tiga buah komponen, yang jika diterjemahkan ke dalam bentuk matriks adalah berupa matriks tiga dimensi. Gambar 2.9 menunjukkan sebuah kubus RGB, yang dapat digunakan untuk memahami konsep dari ruang warna RGB. Terlihat bahwa jika seluruh komponen warna bernilai minimal, maka piksel akan berwarna hitam. Sebaliknya, jika seluruh komponen warna bernilai maksimal, maka piksel akan berwarna putih.



Gambar 2.9 Kubus Ruang Warna RGB⁶

⁶ Sumber: <http://msdn.microsoft.com/en-us/library/>

2.4 Segmentasi Citra

2.4.1 Segmentasi Citra Digital pada Bidang Biomedis

Segmentasi membagi sebuah citra menjadi objek-objek atau wilayah-wilayah yang menjadi ketertarikan. Level pembagian yang dilakukan sangat bergantung kepada jenis masalah yang ingin diselesaikan. Proses segmentasi sebaiknya berhenti ketika objek ketertarikan dalam sebuah aplikasi sudah ditemukan. Kesuksesan atau kegagalan dari prosedur analisis segmentasi dapat dilihat dari akurasi yang didapatkan.

Algoritma segmentasi citra secara umum berdasarkan pada satu atau dua properti awal dari nilai intensitas: diskontinuitas dan kesamaan. Untuk kategori pertama, pendekatannya adalah untuk membagi citra berdasarkan perubahan yang tiba-tiba pada intensitas, seperti sisi pada sebuah citra. Prinsip pendekatan pada kategori yang kedua adalah berdasarkan partisi sebuah citra menjadi wilayah-wilayah yang mirip berdasarkan himpunan kriteria yang telah didefinisikan.

Hingga saat ini telah terdapat berbagai macam jenis metode segmentasi yang dikembangkan. Secara umum, metode-metode segmentasi dapat diklasifikasikan menjadi empat macam yaitu, (1) metode-metode klasik seperti *thresholding*, *region-growing*, dan *edge-based technique*, (2) metode-metode berbasis statistik seperti *maximum-likelihood-classifier* (MLC), (3) metode-metode *neural network* seperti SOPCA dan SOFM, dan yang terakhir adalah (4) metode-metode *clustering*, seperti *hard k-means clustering* dan *fuzzy c-means clustering* (FCM) (Rajapakse, 1997).

Proses segmentasi pada citra biomedis merupakan suatu bagian dari bidang pengolahan citra yang mendapatkan banyak perhatian. Hal ini disebabkan citra biomedis memiliki variasi citra yang sangat banyak dan hasil dari proses segmentasi memberikan manfaat bagi perkembangan dunia kesehatan. Banyaknya jenis variasi citra pada bidang biomedis tidak hanya disebabkan oleh perbedaan organ yang menjadi fokus perhatian, melainkan juga disebabkan perbedaan cara pengakuisisian citra. Beberapa contoh cara pengakuisisian citra tersebut yaitu ultrasonografi, *magnetic resonance imaging* (MRI), dan mikroskopik. Setiap cara pengakuisisian citra tersebut dapat terbagi-bagi lagi menjadi cara yang lebih spesifik.

Secara khusus pada bidang biomedis, Pham pada laporan surveinya membagi metode segmentasi menjadi delapan kategori yaitu: pendekatan *thresholding*, *region growing*, *classifiers*, pendekatan *clustering*, *artificial neural networks*, *deformable models*, dan pendekatan *atlas guided* (Pham, et al., 1998).

2.4.2 Segmentasi Citra Jaringan Saraf

Segmentasi jaringan saraf berbasiskan komputer pada tingkat sel masih belum banyak dilakukan. Salah satu contoh dari segmentasi yang ada yaitu pengenalan neuron pada jaringan otak manusia dengan menggunakan metode berbasiskan statistik fisik yang melibatkan *image preprocessing*, segmentasi citra paralel, dan seleksi *cluster* berdasarkan basis ukuran, kepadatan optik, dan ukuran (Peng, et al., 2002).

Penelitian segmentasi otomatis pada citra yang berkaitan dengan jaringan saraf, lebih banyak dilakukan pada tingkat organ yang lebih besar, seperti segmentasi citra otak manusia seperti yang telah dilakukan oleh (Rajapakse, 1997) (Tolba, et al.).

2.4.3 Segmentasi Citra *Fluorescent*

Beberapa segmentasi yang dilakukan pada citra *fluorescent* sebagai berikut:

1. Segmentasi otomatis inti sel hasil tangkapan mikroskop konfokal.

Segmentasi terotomatisasi pada inti sel yang diberikan *fluorophore* pada citra mikroskop konfokal menggunakan metode 3D *watershed* (Lin, et al., 2003). Pengolahan secara 3D dimungkinkan karena data hasil tangkapan mikroskop konfokal adalah bersifat tiga dimensi. Penggunaan 3D *watershed* dimaksudkan untuk mengurangi kesalahan segmentasi pada inti sel yang disebabkan oleh *clustering* yang ketat dari inti sel.

Pemilihan algoritma di atas berdasarkan analisis terhadap metode segmentasi lain yang sudah pernah diterapkan sebelumnya. Algoritma berbasiskan deteksi sisi yang diterapkan pada citra masukan terbukti cenderung memiliki banyak *error* seperti identifikasi dari *noisy edges* dan *boundaries* yang diskontinu serta membutuhkan *postprocessing* yang kompleks. Pendekatan *region based*

seperti *thresholding* dan *labeling* hanya cocok untuk citra yang mengandung objek yang terisolasi dengan baik. Teknik lainnya seperti *splitting* dan *merging*, *simple region growing*, *multiple thresholding*, dan *direct morphology* tidak memproduksi hasil yang baik. Terutama ketika citra mengandung inti sel dengan jumlah yang banyak, dan tiap sel memiliki variasi yang signifikan dari ukuran, bentuk, dan intensitas.

2. Segmentasi pada *micrograph fluorescent*

Micrograph adalah sebuah reproduksi grafis dari citra yang didapatkan dengan menggunakan mikroskop. Terdapat beberapa metode segmentasi yang pernah dilakukan dalam segmentasi *micrograph fluorescent*.

Pada metode *neural network*, Sebuah citra diproses dalam dua langkah: Pada langkah pertama, posisi sel dideteksi dengan menggunakan *pre-trained neural network*, pada langkah kedua sel yang terdeteksi digunakan sebagai “*focus point*” untuk memandu sistem kepada sel-sel di citra. Metode ini memanfaatkan konsep *Local Linear Map* (LLM) untuk mendeteksi posisi sel, yang juga menggunakan prosedur *thresholding* yang dikombinasikan dengan *local maximum search*. Metode ini juga memanfaatkan konsep *Competitive Layer Model* (CLM) untuk mengekstraksi kontur (Nattkemper, et al., 2000).

Metode lainnya untuk proses segmentasi pada *micrograph fluorescent* adalah dengan mengkombinasikan operator morfologi dan *thresholding* nilai keabuan. Nilai *threshold* seringkali didapatkan dengan menganalisis histogram tingkat keabuan citra yang dilakukan secara otomatis (Nattkemper, 2004).

Beberapa metode yang jarang digunakan pada dalam segmentasi *micrograph fluorescent* adalah metode *Support Vector Machine* (SVM) dan *machine learning*. Penggunaan SVM jarang dilakukan pada segmentasi *micrograph fluorescent* karena memiliki masalah pada normalisasi nilai *output* pada kasus *training sets* yang tidak seimbang. Pendekatan berbasis *machine learning* juga tidak banyak dilakukan mengingat dibutuhkannya *training sets* dengan jumlah yang memadai. Hal ini tentunya selain membutuhkan citra yang

banyak, juga memerlukan kerja yang tidak sedikit dari para ahli (Nattkemper, 2004).

3. Segmentasi pada citra mikroorganism

Analisis citra *fluorescent* pada mikroorganism biasanya terdiri dari tiga buah tahapan utama, yaitu representasi obyek, menandai pembedaan dengan latar dengan menset nilai *threshold* dan kuantifikasi, enumerasi obyek atau intensitas sinyal (O'Mahonya, et al., 2005).

Salah satu metode segmentasi adalah dengan cara melibatkan *alpha channel* pada citra yang sudah mengalami proses *Laplacian* dan *Gaussian filtering* serta proses *thresholding* (Shopov, et al., 2000). Metode lainnya adalah dengan cara semi-otomatis, yaitu pengguna memasukkan rentang intensitas terhadap beberapa ROI yang memiliki intensitas keabuan yang berbeda (Reddy, et al., 2001).

