BAB 3

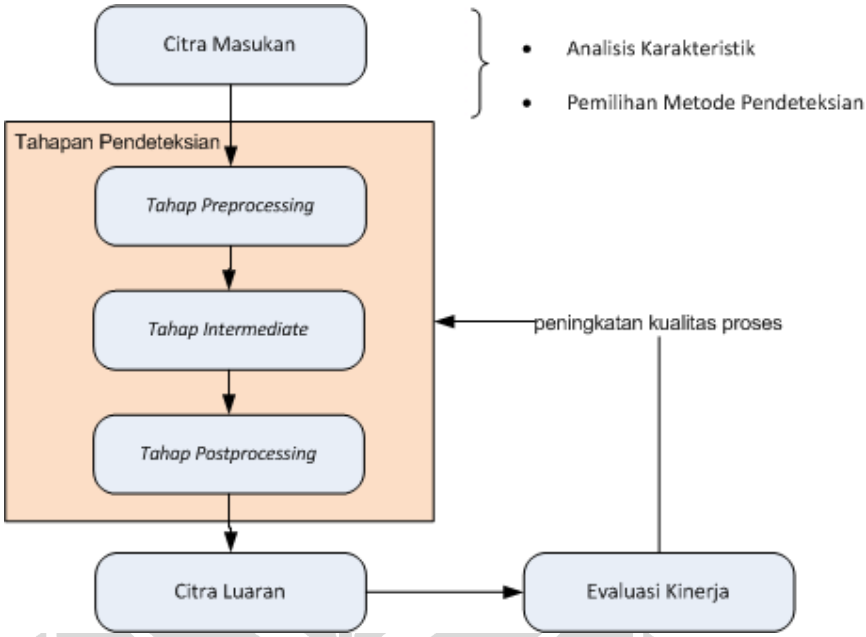
IMPLEMENTASI SISTEM

Bab ini akan membahas mengenai proses implementasi dari metode pendeteksian paranodus yang digunakan dalam penelitian ini. Bab ini terbagai menjadi empat bagian, bagian 3.1 menjelaskan tahapan pemrosesan citra yang diperoleh, bagian 3.2 akan membahas karakteristik citra masukan, bagian 3.3 menjelaskan proses pemilihan algoritma pendeteksian, dan bagian 3.4 berisi tentang penjelasan langkah-langkah yang diperlukan dalam pendeteksian paranodus.

3.1 Tahapan Pemrosesan Citra Masukan

Pada Gambar 3.1 ditampilkan bagan dari tahapan pemrosesan citra masukan. Secara umum, algoritma proses pendeteksian terbagi ke dalam tiga tahapan, yakni tahap *preprocessing*, tahap *intermediate*, dan tahap *postprocessing*. Hasil dari satu tahapan akan menjadi masukan bagi tahap selanjutnya, hingga akhirnya mendapatkan citra yang paranodusnya telah terdeteksi. Selain ketiga tahap tersebut, pada penelitian ini juga dilakukan kegiatan seperti analisis karakteristik citra masukan, pemilihan algoritma pendeteksian, serta evaluasi kinerja untuk melihat adanya kemungkinan peningkatan kualitas proses yang dilakukan.

Analisis karakteristik citra masukan merupakan kegiatan yang dilakukan pertama kali sebelum memulai proses pendeteksian. Analisis ini diperlukan untuk mengetahui dan mendefinisikan ROI pada citra masukan. Selain itu, hasil dari analisis ini juga berguna dalam pemilihan algoritma pendeteksian. Penjelasan lengkap mengenai analisis karakteristik ini terdapat pada subbab 3.2.



Gambar 3.1 Tahapan Pemrosesan Citra Masukan

Kemudian dilakukan pemilihan algoritma pendeteksian paranodus yang akan diimplementasikan. Berdasarkan kajian terhadap kemungkinan adaptasi metode pendeteksian yang ada serta eksplorasi penggunaan deteksi sisi dan *thresholding,* penulis memutuskan untuk menggunakan metode *thresholding* pada pendeteksian paranodus. Paparan mengenai proses pemilihan algoritma ini terdapat pada subbab 3.3.

Seperti yang terlihat pada bagan, tahapan pendeteksian dapat dibagi menjadi *preprocessing*, *intermediate*, dan *postprocessing*. Proses-proses dalam tahap *preprocessing* merupakan proses pendahuluan yang bertujuan menyesuaikan karakteristik citra masukan dan meningkatkan kualitasnya sebelum citra tersebut memasuki tahap berikutnya. Tahap *intermediate* merupakan tahap identifikasi awal terhadap citra yang menghasilkan komponen-komponen kandidat paranodus. Pada tahap *postprocessing*, kandidat paranodus yang diperoleh dari tahap sebelumnya dievaluasi menggunakan kriteria-kriteria yang ditetapkan untuk mendapatkan paranodus yang diinginkan. Penjelasan mengenai tahapan pendeteksian ini terdapat pada subbab 3.4.1.

Setelah ketiga tahapan tersebut diimplementasikan, evaluasi kinerja dilakukan dan kemudian dilihat apakah terdapat kemungkinan untuk meningkatkan kualitas proses pendeteksian paranodus yang sudah diimplementasikan. Pada penelitian ini metode pendeteksian yang sudah diperbaiki tersebut dinamakan dengan metode *local enhancement*, sehingga metode sebelumnya dapat dinamakan dengan metode *non-local enhancement* (NLE). Metode *local enhancement* dapat dilakukan secara manual (*Manual Local Enhancement*/MLE) maupun otomatis (*Automated Local Enhancement*/ALE). Berhubung sebagian besar proses yang terdapat dalam metode MLE dan ALE adalah sama, maka pembahasan mengenai kedua metode tersebut tidak dibagi secara khusus. Penjelasan mengenai metode *local enhancement* dapat dilihat pada subbab 3.4.2.

3.2 Karakteristik Citra Masukan

Untuk menentukan metode segmentasi yang akan digunakan dalam pendeteksian paranodus, maka penulis melakukan analisis karakteristik citra masukan terlebih dahulu. Hal ini perlu dilakukan mengingat segmentasi adalah proses yang sangat bergantung pada jenis citra yang menjadi masukan.

Penulis melakukan analisis terhadap citra masukan yang telah ditandai bagian-bagian mana saja yang menjadi ROI-nya. Analisis yang dilakukan berupa pengidentifikasian karakteristik citra, terutama karakteristik bagian yang menjadi perhatian di citra tersebut.

Karakteristik ROI berdasarkan *paper* awal menurut Kazarinoya-Noyes dan ditunjukkan pada Gambar 3.2 adalah:

1. Suatu ROI terdiri dari dua buah paranodus yang letaknya berdampingan dan dipisahkan oleh suatu celah kecil.
2. Suatu paranodus merupakan suatu wilayah berwarna hijau dengan bentuk menyerupai kotak.
3. Tidak ada suatu paranodus yang berdiri sendiri, karena letaknya selalu berdampingan dengan paranodus lainnya.

4. Intensitas tingkat kehijauan paranodus lebih tinggi dibandingkan dengan intensitas tingkat kehijauan wilayah lain pada citra.

Gambar 3.2 Citra pada *Paper* Awal[7]

Pada penelitian ini, penulis mendapatkan beberapa tipe citra dengan perbedaan karakteristik antara satu dengan yang lainnya, dan sebagian besar dari tipe-tipe tersebut sangat berbeda dengan karakteristik citra pada *paper* acuan. Untuk penelitian ini, dipilih tipe citra masukan yang karakteristiknya mendekati citra pada *paper* acuan. Walaupun begitu, perlu dilakukan analisis lanjut terhadap karakteristik ROI mengingat belum ada deskripsi matematis yang tersedia.

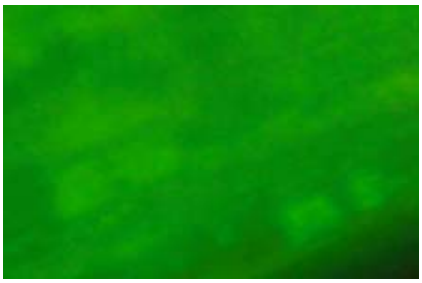
Menurut drg. Didi Santosa sebagai narasumber penelitian, berikut ini adalah karakteristik pada citra masukan yang dapat digunakan untuk membedakan antara ROI dengan wilayah lainnya pada citra:

1. Suatu ROI terdiri dari dua buah paranodus yang letaknya berdampingan dan dipisahkan oleh suatu celah kecil.
2. Suatu paranodus merupakan suatu wilayah berwarna hijau dengan bentuk menyerupai kotak dan memiliki rentang ukuran tertentu.
3. Tidak ada suatu paranodus yang berdiri sendiri, karena letaknya selalu berdampingan dengan paranodus lainnya.
4. Intensitas tingkat kehijauan paranodus lebih tinggi dibandingkan dengan intensitas tingkat kehijauan wilayah tetangganya, bukan wilayah secara global.
5. Pasangan paranodus pada suatu ROI memiliki tingkat kemiringan tertentu.

Contoh citra masukan ditunjukkan pada Gambar 3.3. Kotak kuning yang terdapat pada sudut kanan bawah menunjukkan ROI yang terdapat pada masukan. Berhubung citra yang digunakan pada penelitian ini memiliki resolusi yang sangat besar (2560 x

[7] Sumber: (Kazarinova-Noyes, et al., 2001)

1920 piksel), maka citra dengan resolusi asli disisipkan dalam lampiran. Bentuk citra secara utuh dapat dilihat pada Lampiran A.



Gambar 3.3 Citra Masukan dengan Pembesaran 150%

3.3 Pemilihan Algoritma Pendeteksian Paranodus

Pemilihan algoritma menjadi suatu hal yang penting pada penelitian ini. Hal ini disebabkan segmentasi, yang merupakan bagian dari deteksi paranodus, merupakan suatu proses yang sangat bergantung kepada jenis permasalahan yang dihadapi. Oleh karena itu, sebelum menggunakan metode segmentasi terpilih, penulis melakukan beberapa eksplorasi sederhana terhadap kemungkinan algoritma yang dirasa sesuai terhadap masalah yang ingin diselesaikan.

3.3.1 Analisis Pemilihan Algoritma Pendeteksian Paranodus

Selama ini pendeteksian paranodus pada jaringan saraf dilakukan secara manual dengan menggunakan tenaga ahli. Seperti yang sudah dipaparkan pada bagian 2.4.2, sebagian besar segmentasi terkait jaringan saraf melibatkan organ yang besar seperti otak sehingga tidak mungkin diterapkan pada pendeteksian paranodus ini. Adapun segmentasi pada jaringan saraf lain seperti pengenalan neuron pada jaringan saraf otak adalah berbasiskan statistik yang membutuhkan jumlah data memadai.

Berhubung citra jaringan saraf yang digunakan pada penelitian ini merupakan citra *fluorescent*, penulis juga mencoba melihat kemungkinan untuk mengadaptasi segmentasi yang dilakukan pada citra *fluorescent*. Hal ini karena citra *fluorescent* tentunya memiliki beberapa persamaan karakteristik, seperti jumlah jenis warna pada citra yang biasanya relatif sedikit dan intensitas warna pada latar yang tidak seragam.

Beberapa metode segmentasi yang melibatkan citra *fluorescent* disampaikan pada bagian 2.4.3.

Berdasarkan penjelasan pada bagian 2.4.3 dapat dilihat bahwa metode *thresholding* merupakan metode yang sering digunakan atau dilibatkan pada segmentasi citra *fluorescent* walau memiliki beberapa kelemahan. Selain itu, bersama-sama dengan deteksi sisi dan *region growing*, metode *thresholding* ini merupakan metode yang sifatnya sederhana (Rajapakse, 1997), sehingga penulis menganggapnya cukup cocok untuk digunakan dalam proses segmentasi terhadap pendeteksian citra *fluorescent*.

Dua metode klasik yang dieksplorasi penulis untuk penelitian ini adalah metode deteksi sisi dan *thresholding*. Walaupun metode deteksi sisi jarang diimplementasikan pada citra *fluorescent*, penulis tetap mencoba melakukan eksplorasi untuk mencoba kemungkinan penggunaan deteksi sisi pada segmentasi citra jaringan saraf. Alasannya adalah selain metode tersebut tergolong sederhana, metode deteksi sisi juga merupakan salah satu metode yang paling banyak digunakan dalam melakukan segmentasi (Pham, et al., 1998).

3.3.2 Eksplorasi Penggunaan Deteksi Sisi pada Citra Masukan

Bila dilihat secara sekilas, karakteristik ROI pada citra masukan cukup dapat dideskripsikan dalam bentuk *region* dan memiliki bentuk spesifik yang membedakannya dengan wilayah lain pada citra. Oleh karena itu, penulis mencoba untuk melakukan proses deteksi sisi pada citra masukan.

Deteksi sisi dilakukan pada *layer* hijau citra masukan dengan menggunakan fungsi yang sudah tersedia pada aplikasi MATLAB®. Berikut ini adalah analisis dari hasil deteksi sisi yang dilakukan pada citra masukan:

a. *Sobel Detection*

   Deteksi sisi menggunakan operator Sobel pada citra masukan menghasilkan luaran yang tidak memuaskan. Gambar 3.4 merupakan contoh penggunaan deteksi sisi Sobel pada *layer* hijau citra masukan (Gambar 3.3).

Gambar 3.4 Penggunaan Deteksi Sobel pada Citra Masukan

b. *Prewitt Detection*

Deteksi sisi menggunakan operator Prewitt pada citra masukan, ternyata juga menghasilkan luaran yang tidak memuaskan. Gambar 3.5 merupakan contoh penggunaan deteksi sisi Prewitt pada *layer* hijau citra masukan (Gambar 3.3).

Gambar 3.5 Penggunaan Deteksi Prewitt pada Citra Masukan

c. *Canny Detection*

Deteksi Sobel dan Prewitt, yang menggunakan konsep turunan pertama, ternyata tidak memberikan hasil yang memuaskan. Oleh karena itu, penulis mencoba menggunakan deteksi sisi yang menggunakan konsep turunan kedua, yang salah satunya adalah deteksi sisi Canny. Gambar 3.6 merupakan contoh penggunaan deteksi sisi Canny pada *layer* hijau citra masukan (Gambar 3.3).

Gambar 3.6 Penggunaan Deteksi Canny pada Citra Masukan

Hasil yang diberikan memang terlihat lebih baik dibandingkan dengan dua metode deteksi sebelumnya. Walaupun begitu, pendeteksian *region* tetap sulit dilakukan terhadap luaran yang diberikan karena sisi yang dihasilkan terputus-putus (dikontinu) dan banyak *noise* yang berbentuk *suspicious edge*.

Walaupun begitu, arah pendeteksian yang digunakan pada deteksi sisi Sobel, Prewitt, dan Canny pada penelitian ini masih dilakukan terhadap arah horizontal dan vertikal. Melihat karakteristik citra, pendeteksian yang dilakukan secara diagonal dapat dicoba untuk dieksplorasi lebih lanjut pada penelitian ke depannya.

d. ImageJ[8]

ImageJ merupakan suatu perangkat lunak *open source* yang sering digunakan untuk pengolahan citra biomedik. Salah satu fitur yang dimilikinya adalah fitur deteksi sisi. Hasil yang diberikan memang lebih baik dibandingkan dengan metode lain yang dicoba, hal ini terlihat dengan beberapa sisi yang terlihat lebih jelas. Walaupun begitu, *noise* yang dideteksi juga terlihat lebih banyak sehingga menyulitkan pendeteksian. Hasil deteksi sisi dengan menggunakan ImageJ dapat dilihat pada Gambar 3.7.

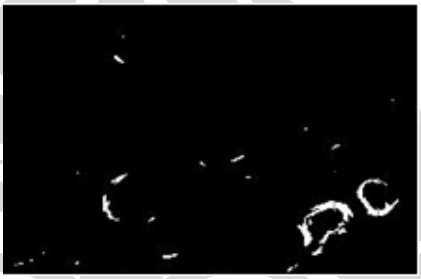
Gambar 3.7 Penggunaan Deteksi Sisi dengan Menggunakan ImageJ

Buruknya hasil deteksi sisi yang didapatkan membuat penulis mencoba melakukan beberapa eksperimen untuk mendapatkan hasil yang lebih memuaskan. Salah satunya adalah dengan cara melakukan *tophat filtering*

[8] Situs ImageJ: http://rsbweb.nih.gov/ij/

terhadap *layer* hijau dari citra masukan, yang dilanjutkan dengan melakukan *median filter.*

Hasil yang didapatkan terlihat jauh lebih memuaskan dibandingkan hasil deteksi sisi yang lain. Akan tetapi, hasil ini belum memenuhi syarat untuk dilakukannya proses pendeteksian ROI terhadap citra karena bentuk wilayah yang ingin diamati masih kasar. Selain itu, banyak ROI yang masih belum dikenali bentuknya. Hasil dari proses deteksi sisi ini dapat dilihat pada Gambar 3.8.



Gambar 3.8 Penggunaan *Improved Edge Detection* Menggunakan ImageJ

Selain deteksi sisi, metode segmentasi lain yang umum digunakan adalah *thresholding*. Algoritma *thresholding* secara umum sudah dijelaskan pada subbab 2.2.7. Pada penelitian ini, penggunaan metode *thresholding* lebih tepat karena wilayah yang ingin dideteksi memiliki intensitas tingkat kehijauan tertentu. Selain itu, jumlah *noise* yang banyak tidak terlalu mempengaruhi wilayah yang ingin dideteksi. Pada Gambar 3.9 tampak wilayah yang ingin dideteksi memiliki bentuk dan ukuran yang jelas berbeda dibandingkan dengan wilayah yang lain.

Gambar 3.9 Penggunaan *Threshold* pada Citra Masukan

Berdasarkan hasil analisis dan eksplorasi di atas, maka metode segmentasi yang digunakan pada penelitian ini adalah *thresholding*.

3.4 Langkah-Langkah Pendeteksian Paranodus

Pendeteksian paranodus yang dilakukan dapat dibagi menjadi dua metode, yaitu metode *non-local enhancement* (NLE) dan perbaikannya yang dinamakan metode *local enhancement*.

3.4.1 Metode *Non-Local Enhancement*

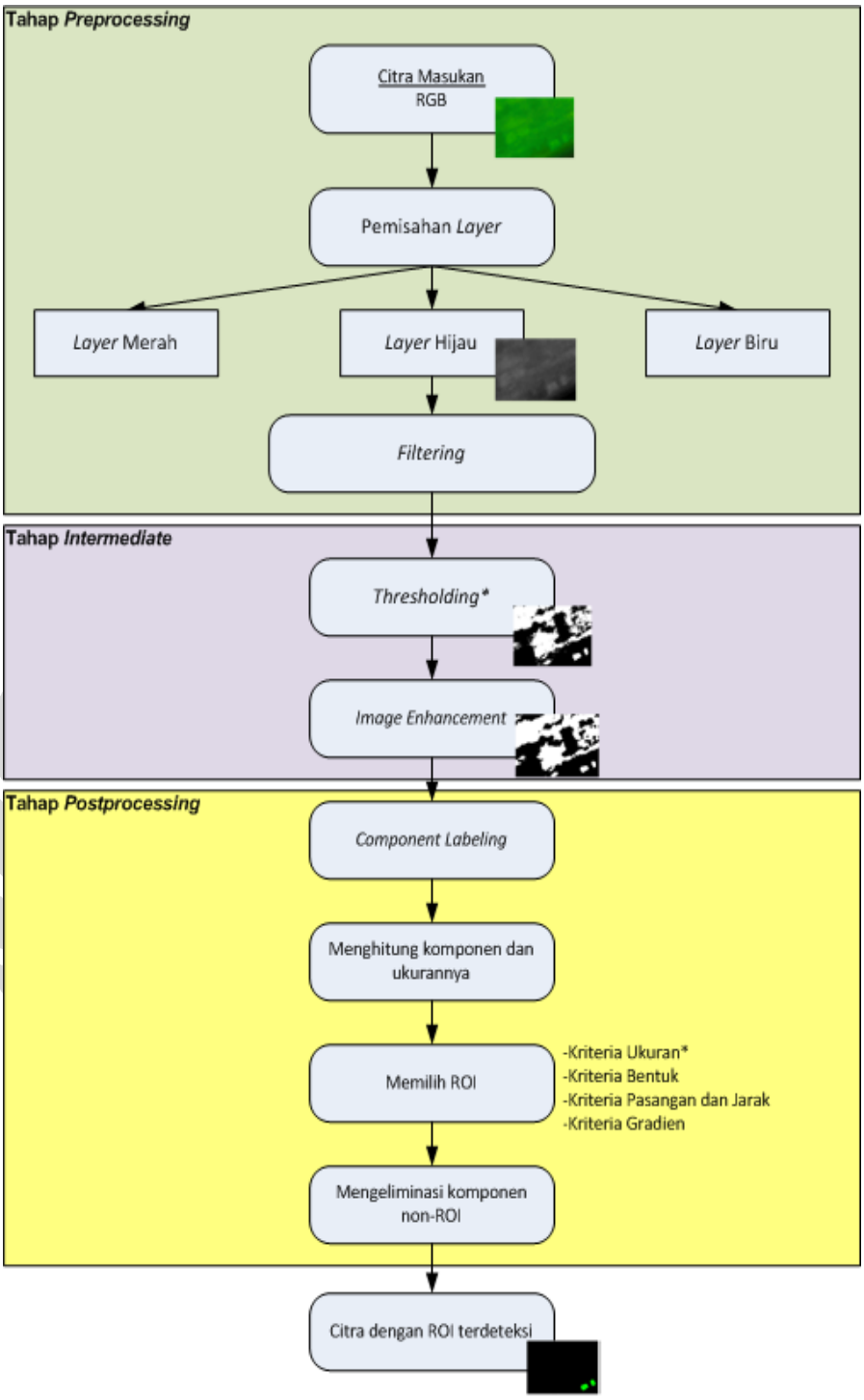
Metode ini merupakan metode awal yang digunakan untuk mendeteksi paranodus dan belum mendapat perbaikan. Tahapan yang dilakukan pada metode ini dapat dilihat pada Gambar 3.10. Terlihat pada bagan bahwa sistem memerlukan bantuan dari pengguna untuk masukan pada dua bagian, yaitu spesifikasi nilai *thresholding* serta spesifikasi ukuran paranodus.

3.4.1.1 Tahap *Preprocessing*

Pada tahap *preprocessing* ini terdapat dua buah tahapan, yaitu pemisahan *layer* RGB dan pengimplementasian *median filter*.

Pemisahan antar *layer*

Pemisahan antar *layer* sesuai dengan kebutuhan dari tujuan proses segmentasi yang dilakukan, yaitu untuk mendeteksi letak paranodus pada citra jaringan saraf. Informasi mengenai paranodus hanya terdapat pada *layer* hijau, sedangkan pada *layer* merah terdapat informasi mengenai kandungan Na_v 1.8, adapun *layer* biru tidak memiliki informasi yang digunakan pada penelitian ini. Oleh karena itu, pendeteksian paranodus hanya diterapkan pada citra *layer* hijau. Berhubung citra *layer* hijau merupakan citra *grayscale*, maka istilah *intensitas kehijauan* dan istilah *intensitas keabuan pada citra layer hijau* yang digunakan pada penelitian ini merujuk pada hal yang sama.



Gambar 3.10 Diagram Tahapan Pendeteksian Paranodus Pada Jaringan Saraf Gigi Manusia dengan Metode Non-*Local Enhancement*

Median Filter

Pada citra *layer* hijau, dilakukan *median filter* dengan ukuran jendela 3 x 3. *Median filte*r ini dilakukan dengan tujuan untuk memperbaiki kualitas citra (untuk mengurangi *noise* yang ada). Intensitas warna yang ada pada tiap paranodus menjadi lebih seragam dibandingkan sebelumnya, sehingga proses *thresholding* yang dilakukan pada tahap berikutnya dapat menghasilkan luaran yang lebih baik.

3.4.1.2 Tahap *Intermediate*

Pada tahap *intermediate* ini terdapat dua buah tahapan, yaitu *thresholding* dan *image enhancement*.

Thresholding

Proses pemilihan batas atas dan batas bawah terhadap intensitas kehijauan pada citra *layer* hijau, hal ini dinamakan dengan *thresholding*. Wilayah yang berada dalam rentang intensitas tingkat kehijauan dipetakan menjadi warna putih dan merupakan kandidat bagi ROI yang akan dideteksi pada tahap berikutnya, sedangkan wilayah yang di luar rentang intensitas tingkat kehijauan akan dipetakan ke warna hitam.

Rentang intensitas warna ROI antar citra ternyata tidak sama sehingga penentuan batas atas bawah intensitas warna dalam metode ini cukup sulit. Tidak dapat ditentukannya nilai *threshold* dari distribusi histogram pada citra secara global juga merupakan penyebab algoritma otomatisasi nilai *threshold*, seperti algoritma Otsu, tidak dapat diimplementasikan. Pada kasus yang dihadapi, intensitas ROI tidak dapat ditentukan hanya dengan analisis terhadap histogram.

Penulis sudah mencoba mencari kesamaan karakteristik dari batas atas dan batas bawah dari tiap citra, seperti besar rentang dan persentase rentang dibandingkan tingkat kehijauan secara keseluruhan. Akan tetapi, kesimpulan dari analisis adalah belum ditemukannya adanya kesamaan karakteristik antar tingkat kehijauan ROI antar citra.

(a) (b) (c)

Gambar 3.11 Ilustrasi Proses yang Terjadi pada Tahap *Preprocessing* (a) Citra masukan yang bertipe RGB (b) *Single layer* komponen *green* dari Citra RGB (c) Hasil *thresholding*

Gambar 3.11(a) menunjukkan citra masukan yang bertipe RGB. Gambar 3.10(b) menunjukkan *layer* hijau yang merupakan hasil pemisahan *layer* dari citra RGB pada Gambar 3.11(a). Gambar 3.11(c) menunjukkan citra biner hasil *thresholding* citra *layer* hijau pada 3.11(b).

Image Enhancement

Setelah *thresholding*, dilakukan *image enhancement* atau peningkatan kualitas dari citra biner hasil tahap sebelumnya yang berupa operasi morfologi erosi serta *median filter*. Sebagaimana yang sudah dijelaskan pada bab 2.2.2, operasi erosi berfungsi untuk mengecilkan ukuran komponen dan membagi komponen yang belum terpisah dengan baik. Pada penelitian ini digunakan *structural element* berbentuk kotak dengan ukuran 2x2 untuk melakukan operasi morfologi erosi.

Perlunya dilakukan operasi *median filter* disebabkan citra hasil operasi *thresholding* pada tahap sebelumnya memiliki gangguan atau *noise* yang banyak. Sebenarnya, akurasi pendeteksian ROI tidak begitu terkendala dengan adanya *noise* tersebut. Akan tetapi *noise* tersebut, yang nantinya akan dideteksi sebagai suatu komponen, dapat memperlambat jalannya proses yang akan berlangsung di tahap berikutnya. Hal ini disebabkan komponen *noise* tersebut akan dikenali dan diperlakukan sama dengan komponen lainnya pada pemrosesan tahap lanjut.

Gambar 3.12 Citra Hasil *Median Filter*

Ukuran dari jendela pada *median filter* yang digunakan bergantung pada karakteristik citra masukan. Akan tetapi, berdasarkan citra masukan yang digunakan pada penelitian ini, ukuran yang digunakan adalah 3 x 3 piksel. Ukuran jendela yang terlalu besar dapat menyebabkan rusaknya ukuran ataupun struktur dari komponen yang merupakan bagian dari ROI. Rusaknya struktur tersebut dapat menyebabkan komponen menjadi tidak dapat dikenali sebagai komponen ROI karena komponen menjadi terlampau kecil/hilang, ataupun tersambung menjadi bagian dari komponen lainnya. Gambar 3.12 merupakan contoh dari penggunaan *median filter* pada citra Gambar 3.11(c) yang merupakan hasil *thresholding*. Terlihat bahwa bentuk citra menjadi lebih berkualitas dengan jumlah *noise* yang berkurang.

3.4.1.3 Tahap *Postprocessing*

Citra yang sudah dikurangi *noise*-nya, menjalani tahap selanjutnya dari proses segmentasi, yaitu tahap *postprocessing*.. Pada tahap ini terdapat dua buah tahapan, yaitu *component labeling* dan penyeleksian ROI.

Component Labeling

Pada penelitian ini, latar dari citra adalah warna hitam dan yang menjadi objek perhatian berarti bagian citra yang berwarna putih. Gambar 3.13 mengilustrasikan konsep *component labeling* yang berlandaskan konsep *connected-component* (subbab 2.2.2). Gambar 3.13(a) menunjukkan sebuah citra biner, dengan angka 1 merepresentasikan piksel yang berwarna putih dan angka 0 merepresentasikan piksel yang berwarna hitam. Gambar 3.13(b) menunjukkan citra biner A yang mengalami proses *labeling* dengan konsep 8-*adjacency*.

Citra biner diberikan label untuk setiap komponen yang berhubungan, dan suatu komponen diberikan nilai bilangan bulat *i*, dengan *i* = 1,2,3, ..., *n*. *n* adalah jumlah seluruh komponen yang terdapat pada citra tersebut. Pada Gambar 3.13(b) terdapat dua buah label komponen, yaitu 1 dan 2. Penggunaan *median filter* pada tahap sebelumnya berfungsi untuk mengurangi jumlah *n* pada citra biner.

$A =$

1	1	1	0	1	1	0
1	1	1	0	1	1	0
1	1	1	0	0	0	1
1	1	1	0	0	0	1
1	1	1	0	0	0	0

(a)

Label (A) =

1	1	1	0	2	2	0
1	1	1	0	2	2	0
1	1	1	0	0	0	2
1	1	1	0	0	0	2
1	1	1	0	0	0	0

(b)

Gambar 3.13Konsep *Labeling* dengan 8-*adjacency*(a) Citra biner A dalam bentuk matriks (b) Hasil dari operasi *labeling*

Citra hasil *component labeling* lalu diproses lebih lanjut, yaitu dengan menghitung ukuran masing-masing komponen. Selain itu, masing-masing piksel pada setiap komponen dicatat lokasinya, dengan cara menyimpan koordinat absis dan ordinatnya.

Penyeleksian ROI

Pendeteksian paranodus dilakukan dengan cara menguji beberapa kriteria pada komponen. Terdapat empat buah kriteria, yaitu kriteria ukuran, bentuk, pasangan dan jarak, serta gradien. Adapun kriteria intensitas kehijauan tidak dicantumkan karena kriteria ini sudah diakomodasi ketika dilakukan proses *thresholding* pada tahap *intermediate*. Penjelasan mengenai kriteria yang digunakan untuk menguji kandidat ROI adalah sebagai berikut:

1. **Kriteria ukuran**

   Ukuran komponen yang menjadi paranodus berada dalam suatu rentang tertentu. Komponen yang memenuhi kriteria ukuran akan dipilih dan komponen yang lainnya diabaikan. Berdasarkan citra masukan yang penulis dapat, ukuran paranodus antar citra yang berbeda memiliki sedikit perbedaan meskipun tidak signifikan. Tidak hanya perbedaan ukuran parodus pada citra yang berbeda, ukuran antar paranodus dalam sebuah citra pun dapat berbeda

pula. Adapun ukuran suatu paranodus yang terdapat pada citra penelitian ini berkisar antara 230 sampai dengan 600 piksel.

2. Kriteria bentuk

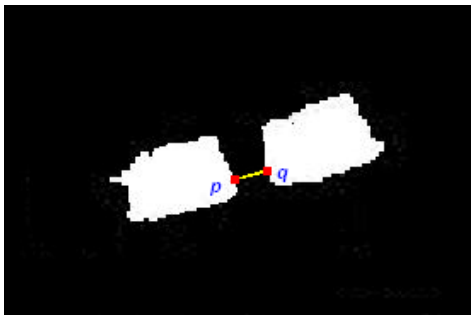
Salah satu karakteristik paranodus untuk pendeteksian secara manual menggunakan mata manusia adalah bentuknya yang menyerupai segi empat. Akan tetapi, metode yang dirancang ini tidak melakukan proses pendeteksian bentuk segi empat terhadap kandidat paranodus. Hal yang menyulitkan dalam pengenalan bentuk ini adalah tidak mulusnya bentuk segi empat yang ingin dideteksi, yang merupakan konsekuensi dari metode *thresholding* yang digunakan. Salah satu proses pengenalan bentuk yang dilakukan adalah dengan pengukuran rasio antara lebar dan tinggi (atau sebaliknya) yang tidak melebihi 2. Jika rasio antara tinggi dan lebar lebih dari 2, maka komponen tersebut bukan termasuk kandidat paranodus. Selain itu, juga dilakukan pengukuran densitas dari komponen yang ingin diuji. Jika komponen memiliki densitas yang terlalu rendah, maka komponen tersebut tidak memenuhi syarat.

3. Kriteria pasangan dan jarak

Setelah mendapatkan komponen yang ukuran dan bentuknya sesuai dengan keinginan, perlu dilakukan pengukuran jarak antar komponen. Hal ini disebabkan ketentuan bahwa suatu ROI terdiri dari dua buah paranodus yang dipisahkan oleh celah kecil. Pengukuran jarak pada penelitian ini menggunakan *euclidean distance*. Konsep mengenai *euclidean distance* dapat dilihat pada persamaan 3.1. Untuk dua buah titik p dan q, $p = (x_1 , y_1)$, $q = (x_2 , y_2)$, jarak antara kedua titik tersebut dihitung dengan persamaan berikut:

$$\sqrt{(p_x - q_x)^2 + (p_y - q_y)^2} \qquad (3.1)$$

Pada penelitian ini, pengukuran jarak yang dilakukan adalah pengukuran jarak antar komponen, bukan pengukuran jarak antara titik atau piksel. Umumnya, jarak antar komponen dihitung dengan cara menghitung jarak masing-masing elemen pada suatu komponen dengan masing-masing elemen pada komponen yang lainnya. Setelah itu, jarak terendahlah yang diambil sebagai jarak antar komponen.



Gambar 3.14 Hubungan Pasangan Komponen dalam Suatu ROI

Gambar 3.14 menunjukkan suatu citra biner yang memiliki suatu ROI yang terdiri dari dua buah komponen berwarna putih. Titik *p* dan *q* adalah dua buah titik pada masing-masing komponen sehingga jarak antara dua komponen menjadi minimal. Jarak antara *p* dan *q* ditandai oleh sebuah garis berwarna kuning.

Pengukuran jarak antara masing-masing elemen untuk setiap komponen tentunya memakan sumber daya yang tidak sedikit, apalagi *euclidean distance* membutuhkan operasi akar kuadrat yang dalam komputasinya tidak mudah. Oleh karena itu, pada penelitian ini komponen-komponen tersebut dijadikan ke dalam bentuk *boundaries* terlebih dahulu sebelum dilakukan perhitungan jarak antar komponen. *Boundaries* adalah elemen terluar dari tiap komponen, sehingga jarak terdekat antar komponen dapat direpresentasikan dengan jarak terdekat antar *boundaries* komponen tersebut.

Penghitungan jarak antar *boundaries* komponen membutuhkan sumber daya yang lebih sedikit karena pereduksian jumlah elemen yang menjadi masukan operasi penghitungan jarak antar komponen. Selain itu, operasi untuk menjadikan komponen dalam bentuk *boundaries* tidak terlalu rumit dibandingkan dengan operasi *euclidean distance*. Terdapat bermacam-macam cara untuk menjadikan komponen ke dalam bentuk *boundaries*, seperti *edge detection* dan operasi konvolusi. Adapun pada penelitian ini digunakan fungsi `boundaries` yang disediakan oleh MATLAB.

Konsep *boundaries* ditunjukkan pada Gambar 3.15. Gambar 3.15(a) menunjukkan representasi sebuah citra biner dalam bentuk matriks. Piksel-piksel yang terletak pada bagian dalam dari suatu komponen diset nilainya menjadi 0 (menjadi daerah latar) dan bagian komponen yang berada di daerah luar nilainya adalah tetap. Gambar 3.15(b) merupakan *boundaries* dari citra biner yang ditunjukkan Gambar 3.15(a).

1	1	1
1	1	1
1	1	1
1	1	1
1	1	1

(a)

1	1	1
1	0	1
1	0	1
1	0	1
1	1	1

(b)

Gambar 3.15 Konsep *Boundaries* (a) Citra biner dalam bentuk matriks (b) Hasil operasi *boundaries*

Setelah jarak antar komponen didapatkan, dilakukan pencarian pasangan komponen yang jarak antara keduanya berada dalam batas toleransi. Jika jaraknya melebihi jarak toleransi yang diperbolehkan, maka berarti pasangan komponen tersebut gagal menjadi kandidat pasangan paranodus. Adapun pada penelitian ini, batas toleransi antara dua buah pasangan paranodus berkisar antara 5 sampai dengan 25 piksel.

4. Kriteria gradien

Pasangan komponen yang sudah memenuhi tiga buah kriteria sebelumnya belum dapat dianggap sebagai pasangan paranodus. Pasangan komponen tersebut harus memenuhi kriteria gradien terlebih dahulu, yaitu nilai tingkat kemiringan atau gradien antar pasangan komponen yang tidak lebih dari 2. Oleh karena itu, pada proses pendeteksian paranodus ini dibutuhkan penghitungan gradien antara dua buah komponen. Konsep gradien antar komponen diilustrasikan pada Gambar 3.14, gradien antara dua buah komponen adalah gradien garis kuning.

Penghitungan gradien antara dua buah titik ditunjukkan oleh persamaan 3.2. Untuk dua buah titik p dan q, $p = (x_1 , y_1)$ dari komponen satu, $q = (x_2 , y_2)$ dari komponen dua, jarak antara mereka dihitung dengan :

$$Gradien\ (p,q) = \frac{y_2 - y_1}{x_2 - x_1} \tag{3.2}$$

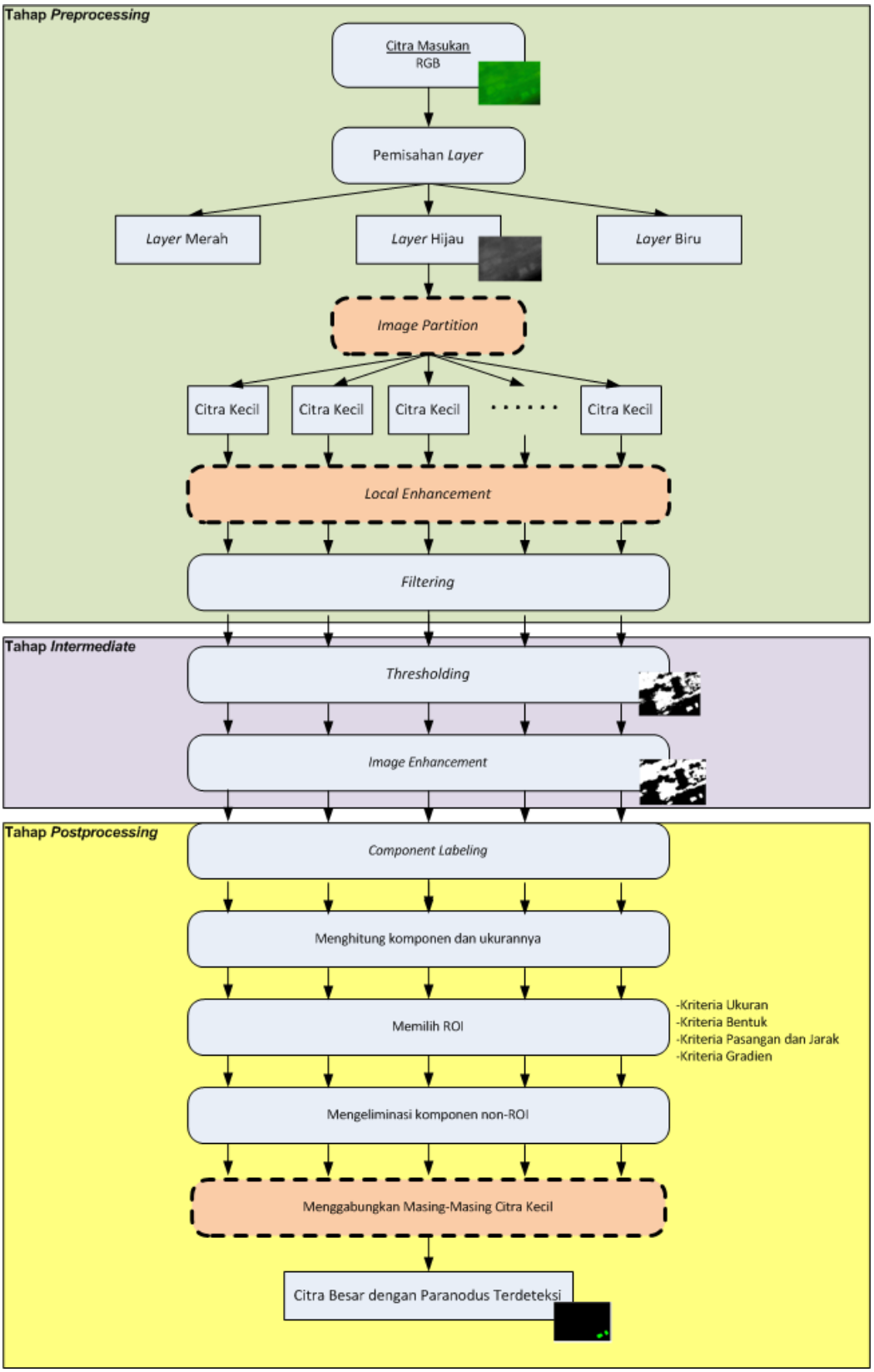
Apabila gradien antara dua buah komponen lebih besar dari 2, maka dua buah komponen tersebut bukan merupakan pasangan paranodus dalam sebuah ROI. Jika terdapat alternatif komponen lain yang jaraknya memenuhi syarat terhadap salah satu dari pasangan komponen yang sedang dievaluasi, maka dilakukan pemeriksaan gradien pula. Jika syarat gradien dan jarak memenuhi, maka pasangan alternatif tersebut diidentifikasi sebagai pasangan paranodus pada ROI menggantikan pasangan komponen sebelumnya.

Pengujian terhadap kriteria dilakukan secara berurutan, dimulai dari kriteria ukuran sampai kepada kriteria gradien. Komponen-komponen yang memenuhi empat kriteria di atas merupakan ROI yang terdeteksi dan posisinya disimpan. Komponen yang tidak termasuk dalam ROI dieliminasi dan tidak ditampilkan dalam citra luaran. Jadi, citra luaran hanya menampilkan paranodus-paranodus pada ROI yang terdeteksi.

3.4.2 Metode *Local Enhancement*

Pendeteksian dengan menggunakan metode *local enhancement* dilakukan setelah melihat kekurangan pada metode *non-local enhancement* (NLE). *Local enhancement* pada penelitian ini dapat didefinisikan sebagai peningkatan kualitas citra secara lokal dengan melakukan fungsi transformasi intensitas terhadap histogram citra.

Kekurangan metode NLE disebabkan oleh kelemahan *thresholding* yang tidak dapat mendeteksi paranodus dengan intensitas warna berbeda dibandingkan paranodus lainnya. Hal ini dapat dilihat dari kenyataan bahwa intensitas warna paranodus ternyata bukan intensitas yang tertinggi pada seluruh wilayah citra (global), tetapi hanya lebih tinggi dibandingkan dengan wilayah tetangga paranodus tersebut (lokal). Pendeteksian pada tingkat lokal diharapkan dapat menambah persentase keberhasilan pendeteksian paranodus.



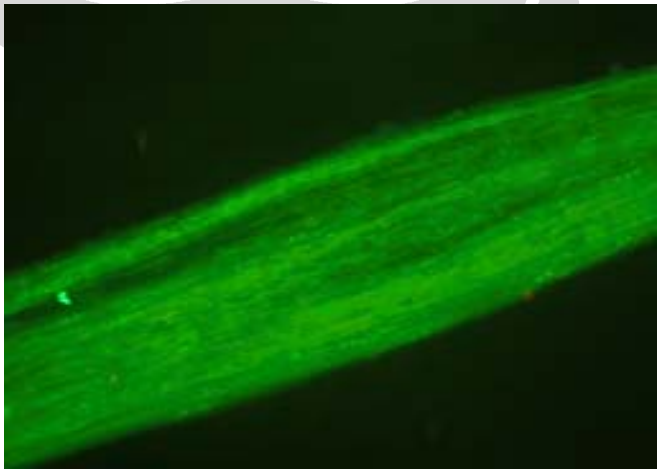
Gambar 3.16 Diagram Tahapan Pendeteksian Paranodus dengan Menggunakan Metode *Local Enhancement*

Secara umum, sebagian besar proses yang dilakukan metode ini adalah sama dengan yang dilakukan pada metode NLE. Gambar 3.16 menunjukkan diagram tahapan pendeteksian paranodus dengan menggunakan *local enhancement*. Kotak yang bergaris putus-putus menggambarkan tambahan tahapan yang diperlukan untuk menjalankan metode ini.

3.4.1.1 Perubahan pada Tahap *Preprocessing*

Citra masukan mengalami proses pemotongan terlebih dahulu sebelum proses pemisahan *layer* RGB dilakukan. Hal ini dilakukan dengna cara membagi citra masukan menjadi citra-citra yang lebih kecil dengan ukuran masing-masing sebesar 160 x 128 piksel. Pemilihan ukuran ini didasarkan oleh ukuran citra masukan serta ukuran ROI. Jadi, gambar besar dapat terbagi dengan sempurna karena ukuran citra kecil merupakan faktor pembagi dari ukuran citra besar. Selain itu, ukuran citra kecil yang lebih besar dari ukuran ROI dimaksudkan agar tujuan pendeteksian tetap terlaksana.

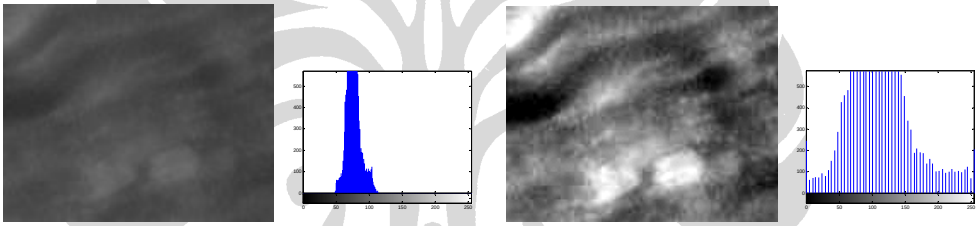
Gambar 3.17 menunjukkan citra masukan yang mengalami proses pemotongan menjadi 240 buah citra kecil. Proses pendeteksian paranodus pada tahap *intermediate* dan *postprocessing* dilakukan kepada masing-masing citra kecil ini dan bukan kepada citra masukan yang berukuran besar.



Gambar 3.17 Pemotongan Citra

Pada masing-masing citra kecil tersebut kemudian dilakukan fungsi transformasi intensitas seperti yang telah dijelaskan pada subbab 2.2.6. Hal ini bertujuan untuk meningkatkan kontras dari citra tersebut, sehingga wilayah ROI akan terlihat lebih jelas dan lebih terang dibandingkan wilayah yang berada di sekitarnya.

Gambar 3.18(a.1) menunjukkan citra kecil yang belum mengalami proses transformasi intensitas, dengan histogram intensitasnya ditunjukkan oleh Gambar 3.18 (a.2). Terlihat bahwa paranodus tidak terlihat jelas, dan intensitas pada citra mengumpul pada rentang yang sangat kecil. Jika transformasi intensitas sudah dilakukan, maka rentang histogram menjadi lebar (Gambar 3.18 (b.2)) serta paranodus terlihat lebih terang dibandingkan dengan daerah sekitarnya (Gambar 3.18 (b.1)).



(a.1) (a.2) (b.1) (b.2)

Gambar 3.18 Transformasi Intensitas pada Citra Kecil (a.1) Gambar citra awal (a.2) Histogram intensitas citra awal (b.1) Citra pasca transformasi intensitas (b.2) Histogram citra pasca transformasi intensitas

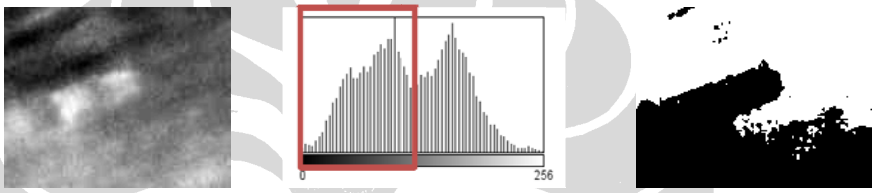
3.4.2.2 Perubahan pada Tahap *Intermediate*

Secara umum, konsep *image enhancement* yang berupa operasi morfologi dan *filtering* adalah sama dengan yang dilakukan pada metode NLE. Namun, terdapat sedikit perbedaan penerapan pada *thresholding* di metode ini. Penggunaan metode *local enhancement* di tahap *preprocessing* menimbulkan keuntungan pada proses pemilihan nilai batas atas dan batas bawah *threshold*, karena citra yang mengalami proses *local enhancement* cenderung memiliki nilai *threshold* yang sama.

Pendefinisian nilai batas atas dan batas bawah untuk *thresholding* pada metode ini dapat dilakukan secara manual ataupun otomatis. Metode *local enhancement* yang memerlukan pendefinisian nilai *threshold* secara manual dinamakan *Manual Local*

Enhancement (MLE), sedangkan jika pendefinisian secara otomatis dinamakan *Automated Local Enhancement* (ALE). Selain karena nilai *threshold* yang cenderung sama, implementasi secara otomatis juga didorong oleh memuaskannya hasil yang ditunjukkan metode manual.

Adapun penetapan nilai *threshold* belum bisa menggunakan algoritma Otsu, yang merupakan salah algoritma penghitungan nilai *threshold* secara otomatis. Hal ini disebabkan bentuk histogram tidak selalu bersifat unimodal (memiliki dua puncak dan satu lembah). Seandainya histogram tersebut unimodal sekalipun, nilai *threshold* yang dikehendaki juga belum tentu sesuai dengan hasil perhitungan algoritma Otsu.



(a.1) (a.2) (b)

Gambar 3.19 Kegagalan Penggunaan Algoritma Otsu untuk Penetapan Nilai *Threshold* (a.1) Gambar citra (a.2) Histogram intensitas citra. Daerah yang berada dalam kotak jingga merupakan bagian latar menurut algoritma Otsu (b) Citra hasil *thresholding* menggunakan algoritma Otsu

3.4.2.2 Perubahan pada Tahap *Postprocessing*

Perbedaan pada tahap *postprocessing* hanya terletak pada akhir tahap, yaitu penggabungan citra-citra kecil menjadi sebuah citra yang berukuran sama dengan citra awal. Selebihnya, proses yang dilakukan adalah sama dengan yang diimplementasikan pada metode *non-local enhancement*. Pada citra akhir tersebut, hanya tampak paranodus-paranodus karena komponen selain itu sudah dieliminasi.