ISSN: 2085-6989

ALGORITMA IMAGE THINNING

Oleh
Zurnawita dan Zulharbi Suar
Staf Pengajar Teknik Elektro Politeknik Negeri Padang

ABSTRACT

Using image thinning algorithm, various example of application is processing image to become more and more variate and helpfully for recognition process an image such as were studied by example. First, this handout will be explained some term which concerning image thinning in digital image processing, then definition from image thinning and algorithm to process image thinning at one particular digital image will be explained. In the next will be given an image thinning application which used for Extraction of Filarial Worms.

Keywords: ***Image thinning, Algor***

PENDAHULUAN

Pada saat sekarang ini teknologi komputer telah berkembang pesat. Teknologi komputer ini pada mulanya hanya berkembang dalam teknologi pengolahan data saja. Namun seiring perkembangan zaman dan teknologi di bidang lain, maka teknologi komputer saat ini pun ikut menyesuaikan diri. Salah satu yang sedang pesat berkembang adalah teknologi komputer dalam bidang multimedia, dimana bidang multimedia saat ini sedang menjadi pusat perhatian. Teknologi komputer multimedia ini meliputi pengolahan gambar (*image*), audio, video, dan lain sebagainya.

Sejak kemunculan pertama gambar digital tahun 1920, terutama munculnya aplikasi dalam teknologi komputer dalam memproses gambar digital tahun 1960 pemoresaan sebuah gambar dalam komputer menjadi lebih mudah. Teknologi pencitraan komputer yang berkembang dengan cepat dan mencapai hasil yang hebat dalam menggantikan, bahkan melampaui kehebatan penglihatan manusia.

Teknologi pemrosesan citra oleh komputer banyak digunakan pada berbagai aplikasi, baik kedokteran, biologi, geografis, dan banyak lainnya. Dalam teknik pengolahan gambar/citra (*image*) ini dikenal suatu teknik pengenalan pola atau *pattern recognition*. Salah satu proses awal dalam *pattern recognition* adalah penipisan gambar, yang lebih dikenal dengan *image thinning*. Dalam sebuah aplikasi biasanya image thinning dikombinasikan dengan berbagai fungsi lain, seperti *skeletoning*, *region filling*, *hit or miss*, dan sebagainya.

Melihat manfaat *thinning* sebagai *preprocessing operation* untuk proses pengolahan gambar atau citra selanjutnya, maka pada makalah kali ini akan dijelaskan mengenai *thinning*, termasuk beberapa algoritma yang dapat digunakan untuk melakukan proses *thinning* tersebut.

STRUKTURE IMAGE THINNING

Definisi Beberapa Terminologi

Thinning merupakan salah satu algoritma perubahan citra dalam *morphological operation*. Berdasarkan etimologi bahasa, *morphology* berasal dari kosakata bahasa Jerman yaitu *morphologie* yang terdiri dari morph+logie/logy. *Morph* adalah suatu kata kerja yang berupa singkatan dari *metamorphose* yang artinya perubahan bentuk atau karakter akibat perubahan pada struktur atau komposisi (transformasi). Sesuai dengan arti dasarnya, konsep *Morphological Operation* pada pengolahan citra adalah operasi-operasi perubahan bentuk pada *binary image* atau *grayscale image*. Berbagai jenis *Morphological Operation* antara lain : *Erosion, dilation, opening, closing, thinning* dan *thickness.*

Binary image adalah citra yang direpresentasikan sebagai himpunan dari *pixel-pixel foreground* (obyek depan). *Binary image* merupakan citra yang hanya mempunyai dua informasi intensitas warna, yaitu hitam dan putih. Konvensi untuk menentukan intensitas warna pada *binary image* bisa berbeda-beda, namun pada umumnya digunakan konvensi yang mendeskripsikan nilai 0 sebagai warna hitam dan nilai 1 atau 255 sebagai warna putih. Warna putih biasanya juga digunakan untuk warna *foreground*, sedangkan warna hitam adalah warna *background*, namun sekali lagi ketentuan ini bukan merupakan keharusan tergantung dari *binary image* yang dihasilkan. *Binary image* seringkali dihasilkan dari proses *thresholding* suatu *grayscale image* (citra yang memiliki informasi tingkat keabuan) dan *color image* (citra yang memiliki informasi intensitas warna yang bervariasi).

Pada setiap operasi *morphology* selalu diperlukan adanya *structuring element*/SE atau yang biasa disebut dengan *kernel*, yaitu basis atau matriks yang menentukan detil-detil tertentu akibat pengaruh suatu operator pada citra. SE/*Kernel* mempunyai bermacam-macam ukuran/dimensi/orde matriks, masing-masing mempunyai satu *origin* yang bisa terletak dimana saja, namun pada umumnya *origin* terletak di *entry* tengah pada matriks. SE juga memiliki satu atau lebih entry kosong (*empty spots*) yang dianggap *don't care*.

Gambar 1. *structuring element/ kernel*

Definisi, Tujuan dan Manfaat Thinning

Definisi *image thinning* adalah proses *morphology image* yang merubah bentuk asli *binary image* menjadi *image* yang menampilkan batas-batas obyek/*foreground* hanya setebal satu *pixel*. Algoritma *thinning* secara iteratif 'menghapus' *pixel-pixel* pada *binary image*, dimana transisi dari 0 ke 1 (atau dari 1 ke 0 pada konvensi lain) terjadi sampai dengan terpenuhi suatu keadaan dimana satu himpunan dari lebar per unit (satu *pixel*) terhubung menjadi suatu garis. *Morphological thinning* dari *binary image* oleh SE juga melibatkan proses *hit-or-miss transform*.

Sepintas, *image thinning* mempunyai kemiripan dengan *edge detection* dalam hal output dari citra yang dihasilkan, kedua proses tersebut sama-sama menampilkan batas obyek pada citra. Namun, tetap saja ada perbedaaan antara *Image Thinning* dengan *Edge Detection* dari segi prinsip kerjanya, yaitu:

- *Edge detection* : merubah *graylevel image* atau *color image* menjadi *image* yang menampilkan batas-

batas/*boundaries* obyek berdasarkan kekontrasan warna antar *pixel*.

- *Image Thinning* : mereduksi *pixel-pixel* pada obyek *binary image* menjadi *pixel* yang benilai sama dengan nilai *pixel* pada *background*. Menghasilkan *binary image* dengan informasi berupa batas-batas obyek berdasarkan *pixel* dengan ketebalan satu *pixel*.

Tujuan *image thinning* adalah untuk menghilangkan *pixel-pixel* yang berada didalam obyek depan (*foreground object*) pada *binary images*.

Manfaat image thinning adalah sebagai berikut:

- biasanya diterapkan pada proses *skeletonisasi.*
- Berguna untuk merapikan/menyem-purnakan hasil output proses *edge detection* dengan cara mengurangi lebar sisi/batas/*edge*.

Algoritma Strukture Image Thinning

Thinning merupakan metode yang digunakan untuk *skeletonizing* yang salah satu penggunaanya adalah dalam aplikasi *pattern recognition*. Terdapat cukup banyak algoritma untuk *image thinning* dengan tingkat kompleksitas, efisiensi dan akurasi yang berbeda-beda. Citra yang digunakan adalah citra biner, jika citra itu merupakan suatu citra *grayscale*, biasanya dilakukan *thresholding* terlebih dahulu sedemikian rupa sehingga citra tersebut menjadi citra biner. Citra biner adalah citra yang hanya memiliki 2 kemungkinan nilai pada setiap piksel-pikselnya, yaitu 0 atau 1. Nilai 0 adalah *background points*, biasanya bukan merupakan bagian dari citra sesungguhnya. Sedangkan nilai 1 adalah *region points*, yaitu bagian dari citra sebenarnya (bukan latar belakang). Citra hasil dari algoritma *thinning* biasanya disebut dengan *skeleton*.

Umumnya suatu algoritma *thinning* yang dilakukan terhadap citra biner seharusnya memenuhi kriteria-kriteria sebagai berikut:

a. *Skeleton* dari citra kira-kira berada di bagian tengah dari citra awal sebelum dilakukan *thinning*.
b. Citra hasil dari algoritma *thinning* harus tetap menjaga struktur keterhubungan yang sama dengan citra awal.
c. Suatu *skeleton* seharusnya memiliki bentuk yang hampir mirip dengan citra awal.
d. Suatu skeleton seharusnya mengandung jumlah pixel yang seminimal mungkin namun tetap memenuhi kriteria-kriteria sebelumnya.

Algoritma Dan Proses Thinning

Ada beberapa algoritma yang digunakan dalam proses *thinning*. Diantaranya adalah metode Stentiford dan metode Zhan-Suen.

Stentiford *method* menggunakan *template-based mark-and-delete thinning algorithm*. Algoritma ini menggunakan *template matching*, dimana jika bagian dari gambar sesuai dengan *template*, hapus *pixel* yang di tengah.

Zhan-Suen *method* menggunakan metode iterasi, yang berarti nilai yang baru didapat dari proses sebelumnya. Cara ini mudah untuk diimplementasikan.

Thinning dengan Algoritma Zhang Suen

Algoritma ini adalah salah satu algoritma *thinning* yang cukup populer dan telah digunakan sebagai suatu basis perbandingan untuk *thinning*. Algoritma ini cepat dan mudah diimplementasikan. Setiap iterasi dari metode ini terdiri dari dua sub-iterasi yang berurutan yang dilakukan terhadap *contour points* dari wilayah citra. *Contour point* adalah

setiap pixel dengan nilai 1 dan memiliki setidaknya satu *8-neighbor* yang memiliki nilai 0.
Dengan informasi ini, langkah pertama adalah menandai *contour point* p untuk dihapus jika semua kondisi ini dipenuhi:

(a) $2 \le N(p_1) \le 6$;
(b) $S(p_1) = 1$;
(c) $p_2 . p_4 . p_6 = 0$;
(d) $p_4 . p_6 . p_8 = 0$;

dimana $N(p_1)$ adalah jumlah tetangga dari p_1 yang tidak 0; yaitu,
$N(p_1) = p_2 + p_3 + ... + p_8 + p_9$

p_9	p_2	p_3
p_8	p_1	p_4
p_7	p_6	p_5

dan $S(p_1)$ adalah jumlah dari transisi 0-1 pada urutan $p_2, p_3, ..., p_8, p_9$.

Dan pada langkah kedua, kondisi (a) dan (b) sama dengan langkah pertama, sedangkan kondisi (c) dan (d) diubah menjadi:

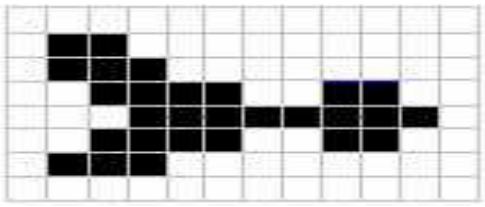
(c') $p_2 . p_4 . p_8 = 0$;
(d') $p_2 . p_6 . p_8 = 0$;

Langkah pertama dilakukan terhadap semua *border pixel* di citra. Jika salah satu dari keempat kondisi di atas tidak dipenuhi atau dilanggar maka nilai piksel yang bersangkutan tidak diubah. Sebaliknya jika semua kondisi tersebut dipenuhi maka piksel tersebut ditandai untuk penghapusan.

Piksel yang telah ditandai tidak akan dihapus sebelum semua *border points* selesai diproses. Hal ini berguna untuk mencegah perubahan struktur data. Setelah langkah 1 selesai dilakukan untuk semua *border points* maka dilakukan penghapusan untuk titik yang telah ditandai (diubah menjadi 0). Setelah itu dilakukan langkah 2 pada data hasil dari langkah 1 dengan cara yang sama dengan langkah 1 sehingga, dalam satu kali iterasi urutan algoritmanya terdiri dari:

1. Menjalankan langkah 1 untuk menandai *border points* yang akan dihapus,
2. Hapus titik-titik yang ditandai dengan menggantinya menjadi angka 0,
3. Menjalankan langkah 2 pada sisa *border points* yang pada langkah 1 belum dihapus lalu yang sesuai dengan semua kondisi yang seharusnya dipenuhi pada langkah 2 kemudian ditandai untuk dihapus,
4. Hapus titik- titik yang ditandai Dengan menggantinya menjadi angka 0.

Prosedur ini dilakukan secara iteratif sampai tidak ada lagi titik yang dapat dihapus, pada saat algoritma ini selesai maka akan dihasilkan *skeleton* dari citra awal
Contoh proses thinning dengan Algoritma Zhang Suen



Gambar 2. Sebelum proses *thinning*

Langkah-langkahnya:

a. Beri tanda semua piksel 8-tetangga yang memenuhi kondisi (1) sampai dengan (4).
b. Hapus piksel tengahnya.
c. Beri tanda semua piksel 4-tetangga yang memenuhi kondisi (5) sampai dengan (8).
d. Hapus piksel tengahnya.

Lakukan langkah a sampai d berulang kali, sampai tidak ada perubahan.

Kondisi:

(1) $2 \le N(p_1) \le 6$
(2) $S(p_1) = 1$
(3) $p_2 \bullet p_4 \bullet p_6 = 0$
(4) $p_4 \bullet p_6 \bullet p_8 = 0$
(5) $2 \le N(p_1) \le 6$
(6) $S(p_1) = 1$
(7) $p_2 \bullet p_4 \bullet p_8 = 0$
(8) $p_2 \bullet p_6 \bullet p_8 = 0$

Dimana:

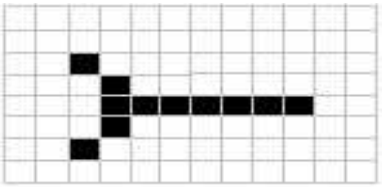
$N(p_1)$ = jumlah dari tetangga p1 yang tidak nol

$S(p_1)$ = jumlah transisi 0 – 1 dalam urutan p_2, p_3, ...

Penamaan piksel :

p9	p2	p3
p8	p1	p4
p7	p6	p5

Jika pada Gambar 2. dilakukan proses *thinning*, maka gambar akan menjadi:

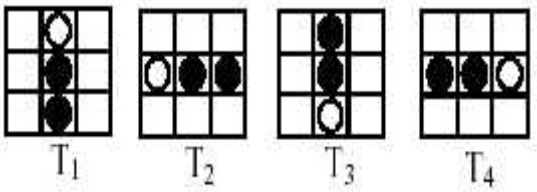


Gambar 3. Setelah proses *thinning*

Thinning dengan Algoritma Stentiford

Metode ini adalah algoritma *thinning* dengan menggunakan teknik *template-based mark-and-delete*. Metode ini cukup terkenal karena *reliable* dan kefektifannya. Metode *thinning* jenis ini menggunakan *template* untuk dicocokkan dengan citra yang akan di-*thinning*. Algoritma ini bersifat iteratif yang berguna untuk mengikis lapisan *pixel* terluar sampai tidak ada lapisan lagi yang dapat dihilangkan.

Template yang dipakai adalah 4 buah *template* 3 x 3 yaitu,



Gambar 4. Template 3 x 3

Berikut ini akan dijelaskan langkah–langkah algoritma *Stentiford* :

1. Tandai *pixel* tersebut untuk kemudian dihilangkan (*remove*).
2. ***Endpoint pixel*** : cari *pixel* pada lokasi (i,j) dimana *pixel-pixel* pada image cocok dengan *template* T_1. Dengan *template* ini, maka semua *pixel* di bagian atas dari image akan dihilangkan (*remove*). Pencocokkan *template* ini bergerak dari kiri ke kanan dan dari atas ke bawah.

   Bila *pixel* tengah bukan merupakan *endpoint* dan mempunyai jumlah konektivitas (*connectivity number*) 1, merupakan batas akhir dan hanya terhubung dengan 1 *pixel* saja. Contoh: suatu *pixel* hitam hanya mempunyai satu tetangga saja yang hitam juga dari kemungkinan delapan tetangga.

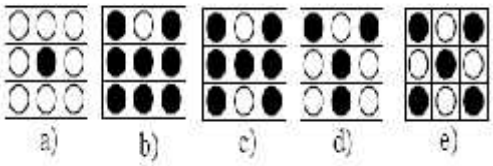
   Connectivity number : merupakan suatu ukuran berapa banyak obyek yang terhubung dengan *pixel* tertentu (dihitung berdasarkan rumus di bawah).

$$C_n = \sum_{k \in S} N_k - (N_k \cdot N_{k+1} \cdot N_{k+2})$$

   N_k merupakan nilai dari 8 tetangga di sekitar *pixel* yang akan dianalisa (*central pixel*) dan nilai S = {1,3,5,7}. N_0 adalah nilai dari *pixel* tengah (*central pixel*). N_1 adalah nilai dari *pixel* pada sebelah kanan *central pixel* dan sisanya diberi nomor berurutan dengan arah berlawanan jarum jam.

4	3	2
5	0	1
6	7	8

Contoh :



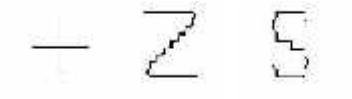
Bagian a) menjelaskan tentang jumlah konektivitas (*connectivity number*) = 0.
Bagian b) menjelaskan tentang jumlah konektivitas (*connectivity number*) = 1.
Bagian c) menjelaskan tentang jumlah konektivitas (*connectivity number*) = 2.
Bagian d) menjelaskan tentang jumlah konektivitas (*connectivity number*) = 3.
Bagian e) menjelaskan tentang jumlah konektivitas (*connectivity number*) = 3

3. Ulangi langkah 1 dan 2 untuk semua *pixel* yang cocok dengan *template* T_1.
4. Ulangi langkah 1–3 untuk *template* T_2, T_3 dan T_4.
   Pencocokkan *template* T_2 akan dilakukan pada sisi kiri dari obyek dengan arah dari bawah ke atas dan dari kiri ke kanan.
   Pencocokkan *template* T_3 akan dilakukan pada sisi bawah dari obyek dengan arah dari kanan ke kiri dan dari bawah ke atas.
   Pencocokkan *template* T_4 akan dilakukan pada sisi kanan dari obyek dengan arah dari atas ke bawah dan dari kanan ke kiri.
5. *Pixel-pixel* yang ditandai untuk dihilangkan (*remove*) dibuat sama dengan *background* (di-set 0 untuk *binary image*).

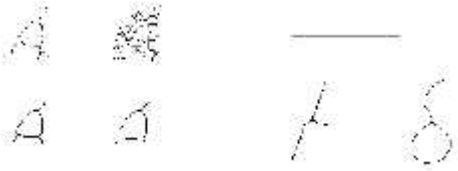
Contoh Hasil Thinning
Gambar Asli
Gambar 1

Gambar 2

Dengan algoritma Zhang-Suen
Citra setelah di-*thinning* :
Gambar 1:

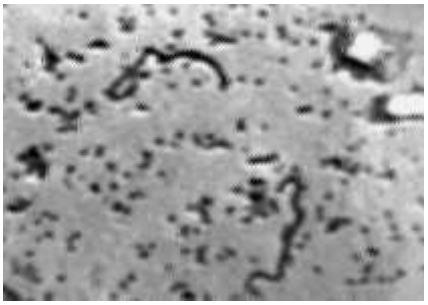


Gambar 2:

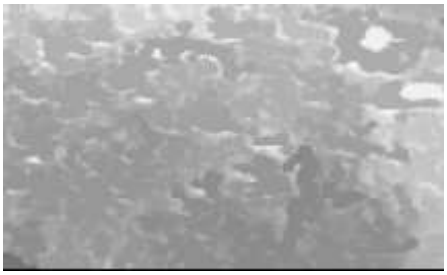


Aplikasi Strukture Image Thinning

Pada contoh berikut ini, *image thinning* digunakan untuk mengekstraksi citra cacing filaria dari suatu citrayang didapat. http://www.dca.fee.unicamp.br/projects/khoros/mmach/tutor/application/biological/fila/fila.html. Berikut adalah langkah-langkahnya:

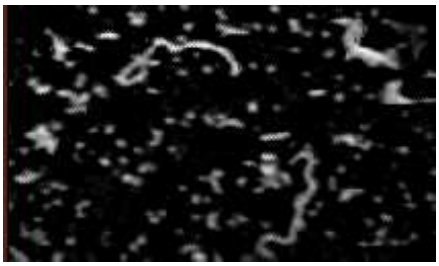


Gambar 5. Citra mikroskopik dari transmiter cacing filaria.



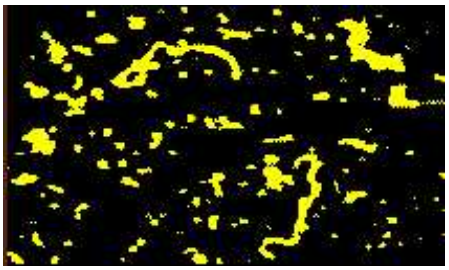
Gambar 6 *Closing by reconstruction*

Pada citra tersebut dilakukan filter morfologis untuk memisahkan bagian-bagiannya yang terlihat berlubang. *Filtering* ini dilakukan dengan *closing by reconstruction*.



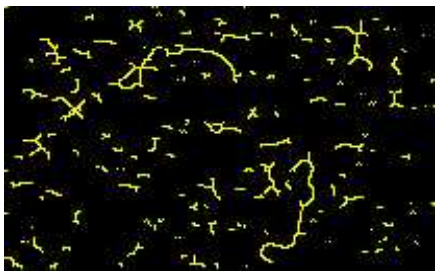
Gambar 7. *Subtraction*

Selanjutnyadilakukan *morphological subtraction* Gambar 5 dari Gambar 6 yang menghasilkan Gambar 7.



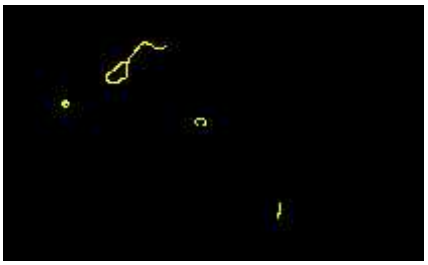
Gambar 8. *Thresholding*

Berikutnya dilakukan *thresholding* pada gambar, yang menghasilkan Gambar 8.



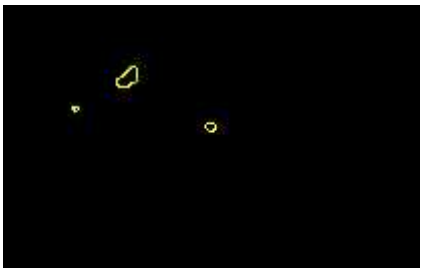
Gambar 9. *Skeletonization*

Sekarang tujuannya adalah untuk membedakan cacing filaria dari obyek yang lain. Hal itu dapat dilakukan dengan mengingat fakta bahwa cacing filaria lebih panjang daripada obyek lain. Sehingga tinggal mencari tanda untuk obyek terpanjang dari *binary image*. Aplikasikan *skeleton* dengan cara *thinning* pada Gambar 9.



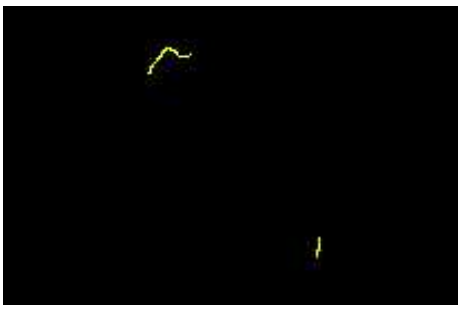
Gambar 10. N-*Thinning*

Dilanjutkan oleh N-*Thinning* pada Gambar 10.



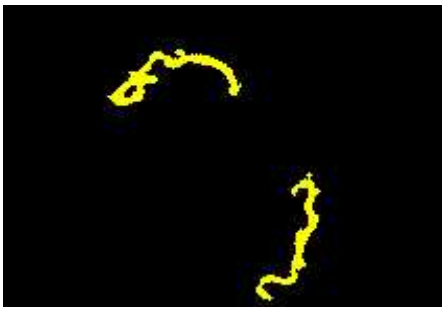
Gambar 11. *Skeletonization*

Tail dari citra di atas bisa menjadi tanda yangdiinginkan. Untuk mendapatkannya, kita bisa mengaplikasikan *skeleton* yang akan menghabiskan citra pada Gambar 10.



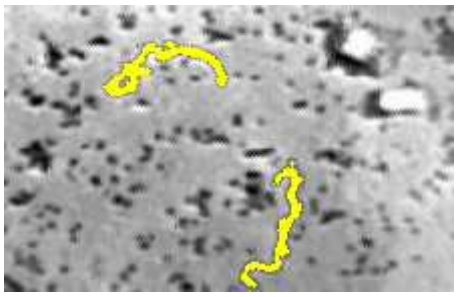
Gambar 12. *Subtraction*

Dan kurangkan citra Gambar 11 dari citra Gambar 10 yang akan menghasilkan Gambar 12.



Gambar 13. *Opening by reconstruction*

Untuk menyelesaikannya, kita hanya harus merekonstruksikan gambar cacing filaria dari *tail*. Ini dilakukan dengan *opening by reconstruction* yang hasilnya adalah Gambar 13.



Gambar 14. Citra kombinasi

Gambar 14. adalah gambar kombinasi antara citra cacing filaria yang terekstraksi dan citra asli.

KESIMPULAN

Dari berbagai pembahasan yang telah dilakukan di atas, dapat disimpulkan sebagai berikut:

1. Struktur *image thinning* seringkali berhubungan dengan proses pengolahan citra yang lain, seperti *subtraction*, *reconstruction*, *skele-toning*, dan sebagainya.
2. Untuk melakukan *thinning* dapat memilih beberapa algoritma dari banyak algoritma yang tersedia, diantaranya adalah metode Stentiford dan metode Zhan-Suen.
3. Algoritma Zhang-Suen memiliki efisiensi yang cukup baik dan implementasi yang cukup mudah namun kualitasnya termasuk relatif kurang baik dibandingkan dengan algoritma Stentiford.
4. Algoritma Stentiford memiliki efisiensi yang kurang dan implementasinya cukup sulit jika dibandingkan dengan algoritma Zhang-Suen.

DAFTAR PUSTAKA

Extraction of Filarial Worms [*on-line*]. Didapat dari http://www.dca.fee.unicamp.br/projects/khoros/mmach/tutor/application/biological/fila/fila.html; Internet; diakses tanggal 30 April 2007.

Mathematical Morpohology [*on-line*]. Didapat dari www.imm.dtu.dk/~jmc/02501/lectures/02501_morphology.pdf; Internet; diakses tanggal 30 April 2007.

Selected Morphological Algorithm [*on-line*]. Didapat dari http://documents.wolfram.com/applications/digitalimage/UsersGuide/

6.5.html Internet; diakses tanggal 30 April 2007.

Thining [*on-line*]. Didapat dari http://vision.cse.psu.edu/resources/hipr/html/thin.htm; Internet; diakses tanggal 30 April 2007.

Cheng, Ya-Lin dan Cherng-Min Ma.*A Topology Preserving 4-subiteration Gray- Level Thinning Algorithm* [on-line]. Tersedia di: http://www.csie.mcu.edu.tw/~yklee/CVGIP03/CD/Paper/IP/IP-06.pdf; Internet; Diakses 30 April 2007.

Gonzalez, Rafael C and Richard E.Woods. 1992.***Digital Image Processing***. Addison- Wesley Publishing Company.

A. Jain,1989 *Fundamentals of Digital Image Processing*, Prentice-Hall.