

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/228408578>

SEGMENTASI CITRA MEDIS DENGAN ALGORITMA DETEKSI TEPI KONTUR BERBASIS PELACAKAN TARGET SECARA DINAMIS

Article · January 2010

CITATIONS

0

READS

7,719

3 authors, including:



Agus Zainal Arifin

Institut Teknologi Sepuluh Nopember

173 PUBLICATIONS 1,072 CITATIONS

[SEE PROFILE](#)



Anny Yuniarti

Institut Teknologi Sepuluh Nopember

69 PUBLICATIONS 383 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



serious games [View project](#)



Detection of Malaria Parasite Grom Thickblood Smear Microscopist Image [View project](#)

SEGMENTASI CITRA MEDIS DENGAN ALGORITMA DETEKSI TEPI KONTUR BERBASIS PELACAKAN TARGET SECARA DINAMIS

Puruhito Bagus Prakosa¹, Agus Zainal Arifin², Anny Yuniarti³

Teknik Informatika, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember

email : puruhito.prakosa@yahoo.com¹, agusza@cs.its.ac.id², anny@if.its.ac.id³

ABSTRAK

Pada citra medis, identifikasi dari tepi objek atau daerah yang diinginkan dapat menyediakan informasi yang berguna untuk diagnosis dan perawatan penyakit. Proses segmentasi yang terlibat dapat dilakukan secara manual, tetapi sangat bergantung pada keahlian pengguna. Tingkat akurasi segmentasi yang dihasilkan metode manual juga rendah karena pengaruh intra- dan inter-observer. Oleh karena itu, dibutuhkan metode segmentasi secara otomatis.

Dalam makalah ini diimplementasikan sebuah metode segmentasi citra medis dengan deteksi tepi kontur berbasis pelacakan target secara dinamis. Jalur pada tepi objek dihasilkan melalui deteksi tepi dengan menerapkan sembilan operator arah. Selanjutnya, target-target di sepanjang jalur tepi objek dalam citra dilacak dengan menggunakan teknik *modified edge-following*.

Dari percobaan yang dilakukan terhadap citra *Magnetic Resonance* dan *Dental Panoramic Radiograph*, metode ini mampu melakukan segmentasi objek dengan cukup baik. Metode ini juga memiliki keunggulan dimana keseluruhan proses segmentasi dilakukan secara otomatis.

Kata kunci : Segmentasi citra, deteksi tepi, operator arah, pelacakan target, teknik *edge-following*

1 PENDAHULUAN

Segmentasi citra merupakan topik penting dalam bidang pengolahan citra digital dan dapat ditemukan dalam berbagai bidang riset citra. Sebagai contoh adalah citra medis dan video digital yang memiliki pendekatan berbeda dalam penerapan proses segmentasi. Meskipun memiliki pendekatan berbeda, tujuan yang ingin dicapai adalah sama yaitu menentukan objek yang spesifik atau sebuah area dalam citra untuk tujuan pencocokan atau identifikasi. Dalam citra medis, identifikasi dari objek atau area yang diinginkan dapat menyediakan informasi yang berguna untuk diagnosis dan perawatan penyakit. Namun, seringkali proses segmentasi yang terlibat dilakukan secara manual serta bergantung pada keterlibatan pengguna sehingga sangat memakan waktu dan menyulitkan pengguna. Oleh karena itu, proses segmentasi secara otomatis lebih digemari [1].

Segmentasi citra secara otomatis merupakan area riset yang telah menghasilkan beberapa metode. Salah satunya adalah deteksi tepi dimana teknik *edge tracing*

atau *edge following* termasuk di dalamnya. Adapun kekurangan yang biasa ditemukan dalam teknik *edge following* adalah keahliannya yang rendah. Hal ini disebabkan oleh proses penentuan tepi objek yang dilakukan sebelum proses segmentasi seringkali menunjukkan hasil yang kurang baik sehingga mempengaruhi hasil segmentasi. Meskipun demikian, identifikasi dari tepi dengan menghubungkan titik tepi tetangga menyediakan informasi berguna yang tidak ditemukan di metode lain.

Dalam teknik *edge following*, penentuan nilai - *threshold* memiliki peran yang sangat penting. Apabila nilai *threshold* yang digunakan tidak tepat maka akan mempengaruhi akurasi dari hasil segmentasi. Proses penentuan nilai *threshold* sering membingungkan pengguna, karena pengguna harus mengetahui karakteristik dari citra itu sendiri untuk mendapatkan nilai *threshold* yang tepat.

Makalah ini membahas tentang segmentasi citra medis dengan algoritma deteksi tepi kontur berbasis pelacakan target secara dinamis, sebuah pendekatan baru yang diharapkan dapat mengatasi permasalahan-permasalahan seputar segmentasi citra yang ada, seperti proses segmentasi manual yang merepotkan pengguna, hasil deteksi tepi objek yang kurang baik, dan penentuan nilai *threshold* secara manual. Proses pelacakan target menggunakan teknik *modified edge-following* dimana sebuah target atau titik dengan berbekal jalur hasil deteksi tepi, informasi karakteristik citra dan arah dapat ditentukan pergerakannya. Dari informasi yang dimiliki, penentuan target awal, nilai *threshold* dan arah pergerakan target sampai membentuk kontur tertutup dapat dilakukan secara otomatis..

2 DETEKSI TEPI

Sebelum dilakukan proses segmentasi pada citra masukan, dilakukan proses deteksi tepi untuk menyediakan jalur target di sepanjang tepi objek citra. Metode deteksi tepi yang digunakan adalah *Canny edge detector*.

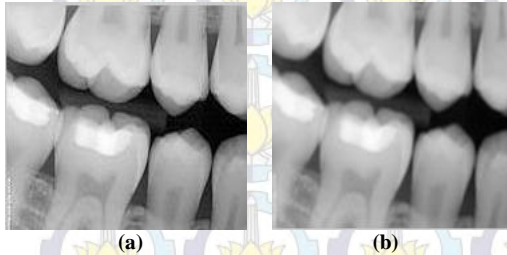
2.1 Penghalusan Citra

Langkah pertama yang dilakukan adalah melakukan filter terhadap *noise* pada citra sebelum melakukan pendeteksian tepi. Hal tersebut dapat dilakukan dengan operasi penghalusan citra. Metode yang digunakan adalah *gaussian filter*. *Gaussian filter* dikonvolusi dengan citra masukan sehingga menghasilkan *gaussian filtered image*. Proses ditunjukkan oleh Persamaan (1) dan (2).

$$g(x, y) = G_{\sigma}(x, y) * f(x, y) \quad (1)$$

$$\text{Dimana } G_{\sigma} = \frac{1}{\sqrt{2\pi}\sigma} \exp\left[-\frac{x^2+y^2}{2\sigma^2}\right] \quad (2)$$

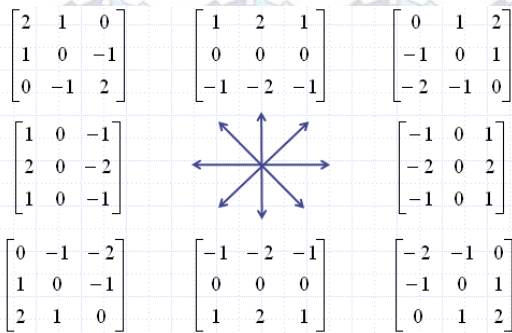
$g(x, y)$ adalah *gaussian filtered image*, $G_{\sigma}(x, y)$ adalah gaussian filter dan $f(x, y)$ adalah citra masukan. Efek penghalusan citra dengan *gaussian filter* ditunjukkan oleh Gambar 1.



Gambar 1. (a) Citra asli dan (b) Citra yang telah dihaluskan

2.2 Mencari Jarak dan Arah Gradien

Deteksi tepi Canny mencari tepi dimana intensitas derajat keabuan citra berubah secara drastis. Lokasi tersebut didapatkan dengan menentukan gradien dari citra. Gradien pada tiap *pixel* dari citra yang telah dihaluskan ditentukan dengan operator Sobel [11]. Langkah pertama adalah untuk memperkirakan gradien sesuai jumlah operator arah dengan mengaplikasikan kernel yang ditunjukkan oleh Gambar 2.



Gambar 2. Sembilan operator arah

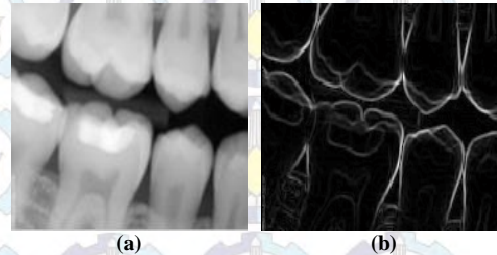
Jarak gradien (atau yang biasa disebut *edge strengths*) dapat kemudian ditentukan sebagai *Euclidean distance measure* dengan mengaplikasikan hukum Pythagoras seperti yang ditunjukkan pada Persamaan (3). Kadang aplikasi *Euclidean distance measure* disederhanakan dengan mengaplikasikan *Manhattan distance measure* seperti yang ditunjukkan pada Persamaan (4) untuk mengurangi kompleksitas komputasi. Operasi tersebut kemudian dilakukan terhadap citra. Gambar 3 menunjukkan citra hasil penghitungan jarak gradien.

$$|G| = \sqrt{G_1^2 + G_2^2 + \dots + G_n^2} \quad (3)$$

$$|G| = |G_1| + |G_2| + \dots + |G_n| \quad (4)$$

Dimana G_1, G_2, \dots, G_n adalah gradien sesuai jumlah arah operator.

Nampak jelas dari Gambar 3 bahwa jarak gradien dapat menunjukkan tepi objek secara jelas. Namun, tepi nampak lebih tebal dan tidak menandakan dimana lokasi tepi sebenarnya.



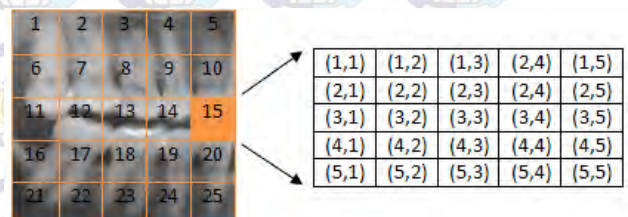
Gambar 3. (a) Citra yang telah dihaluskan dan (b) Hasil jarak gradien pada citra

3 PENENTUAN TARGET PADA TEPI OBJEK PADA CITRA

Setelah citra masukan diproses deteksi tepi dengan operator arah, langkah berikutnya adalah menentukan target disepanjang jalur tepi objek citra. Dalam penentuan target, terdapat beberapa proses yang terlibat yaitu partisi citra, operasi blok hasil partisi citra, penentuan target kandidat dan penentuan target pada tepi objek citra.

3.1 Partisi Citra

Citra dipartisi menjadi beberapa blok dimana masing-masing blok mempunyai ukuran $B \times B$ *pixel*. Tujuan dilakukan partisi citra menjadi beberapa blok adalah untuk melakukan segmen beberapa objek. Ukuran tiap blok dapat disesuaikan, sebagai contoh: 5×5 *pixel* atau 10×10 *pixel*. Tiap penentuan ukuran blok memiliki efek tersendiri. Bila blok mempunyai ukuran cukup kecil, misalnya 5×5 *pixel*, maka objek kecil dan objek dengan perbedaan kontras kecil dapat ditentukan titik-titik targetnya. Namun, semakin kecil ukuran blok maka akan semakin banyak blok yang akan diproses. Sehingga waktu komputasi juga akan semakin bertambah. Contoh ilustrasi penerapan partisi citra menjadi beberapa blok dan ukuran *pixel* masing-masing blok ditunjukkan oleh Gambar 4.



Gambar 4. Ilustrasi partisi citra menjadi beberapa blok

Gambar 4 memberikan ilustrasi partisi citra. Dalam gambar menunjukkan citra gigi yang dipartisi menjadi 25 blok dimana masing-masing blok terdiri dari 5 x 5 *pixel*. *Pixel* pertama dalam blok memiliki koordinat (1,1) dan *pixel* terakhir dalam blok memiliki koordinat (5,5). Sistem koordinat tersebut berlaku untuk semua blok.

Ada kalanya saat proses partisi citra, citra tidak dapat dipartisi secara utuh. Hal ini dikarenakan ukuran citra yang tidak habis dibagi dengan ukuran blok. Misal sebuah citra memiliki ukuran 256 x 256 *pixel* dan ingin dipartisi dengan blok dengan ukuran 5 x 5 *pixel*. Maka terdapat 1 *pixel* terakhir pada sisi baris dan kolom yang tidak memperoleh alamat blok. Hal ini dapat ditangani dengan teknik *padding*. Teknik *padding* dilakukan dengan melakukan modulus baris dan kolom citra dengan baris dan kolom blok. Misal ukuran baris citra adalah 256 dan ukuran baris blok adalah 5, maka $256 \bmod 5 = 1$. Nilai ukuran baris blok, 5 dikurangi hasil modulus tersebut adalah 4 maka 4 nilai *pixel* inilah yang akan ditambahkan ke sisi baris citra. Hal tersebut juga berlaku pada sisi kolom citra. Sehingga hasil akhirnya ukuran citra menjadi 260 x 260 dan dapat dipartisi secara penuh.

3.2 Operasi Blok Hasil Partisi Citra

Pada proses ini dilakukan operasi pada tiap *pixel* dalam blok untuk semua blok. Sehingga setiap blok melakukan operasi yang sama. Operasi yang dimaksud adalah mencari titik-titik tetangga dari koordinat *pixel* (x,y). Di sini (x,y) merepresentasikan koordinat dari setiap *pixel* dalam blok. Gambar 5 menunjukkan ilustrasi delapan tetangga yang harus dicari untuk tiap koordinat *pixel*. Dalam ilustrasi Gambar 5, (x,y) adalah koordinat titik yang dicari tetangga-tetangganya, dan koordinat yang diberi warna adalah tetangga-tetangga yang didapatkan.

(x-2,y-2)	(x-2,y-1)	(x-2,y)	(x-2,y+1)	(x-2,y+2)
(x-1,y-2)	(x-1,y-1)	(x-1,y)	(x-1,y+1)	(x-1,y+2)
(x,y-2)	(x,y-1)	(x,y)	(x,y+1)	(x,y+2)
(x+1,y-2)	(x+1,y-1)	(x+1,y)	(x+1,y+1)	(x+1,y+2)
(x+2,y-2)	(x+2,y-1)	(x+2,y)	(x+2,y+1)	(x+2,y+2)

Gambar 5. Ilustrasi pencarian koordinat tetangga (x,y)

Namun, delapan tetangga yang harus dicari untuk tiap koordinat *pixel* juga disesuaikan dengan letak koordinat dalam blok. Hal ini disebabkan karena operasi pencarian tetangga suatu koordinat *pixel* hanya berlaku pada blok tempat *pixel* tersebut berada, tidak mempengaruhi blok lainnya.

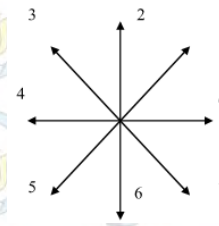
Tetangga-tetangga suatu koordinat *pixel* dapat ditentukan oleh Persamaan (5).

$$r = (x - \text{round}(\sin(d \times \frac{\pi}{4})), y + \text{round}(\cos(d \times \frac{\pi}{4}))) \quad \dots(5)$$

$$l = (x - \text{round}(\sin((d+4) \times \frac{\pi}{4})), y + \text{round}(\cos((d+4) \times \frac{\pi}{4})))$$

Dimana r dan l adalah koordinat tetangga dari (x,y) pada arah d dan $(d+4)$. Rentang nilai d adalah 0 sampai 3 yang

merepresentasikan delapan arah seperti yang ditunjukkan oleh Gambar 6.



Gambar 6. Delapan arah yang direpresentasikan oleh d

3.3 Penentuan Target Kandidat Tiap Blok

Setelah titik-titik tetangga dari semua koordinat dalam blok didapatkan, langkah berikutnya adalah mencari perbedaan nilai *gray level* untuk setiap d . Perbedaan nilai *gray level* ditunjukkan dengan Persamaan (6).

$$D_{mn}^d(x, y) = G_{mn}^d(r) - G_{mn}^{d+4}(l) \quad \dots\dots\dots(6)$$

Dimana d adalah arah, r dan l adalah tetangga suatu koordinat dan mn adalah lokasi suatu blok. Terdapat empat nilai selisih *gray level* sesuai dengan d atau arah yang digunakan. Sebagai ilustrasi dijelaskan dalam Gambar 7 Tetangga suatu koordinat yang ternyata berada di luar lingkup blok, yang dalam implementasi ini diabaikan, diberi nilai 0. Setelah keempat nilai selisih *gray level* didapatkan, nilai-nilai tersebut kemudian dijumlahkan. Penjumlahan dari selisih nilai *gray level* dari tiap *pixel* dalam empat nilai d ditunjukkan oleh Persamaan (7).

$$D_{mn}(x, y) = \sum_{d=0}^3 |D_{mn}^d(x, y)| \quad \dots\dots\dots(7)$$

20	30	40
50	(x,y)	60
60	40	50

$d = 0, D_{mn}^0(x, y) = 60 - 50 = 10$

20	30	40
50	(x,y)	60
60	40	50

$d = 1, D_{mn}^1(x, y) = 40 - 60 = -20$

20	30	40
50	(x,y)	60
60	40	50

$d = 2, D_{mn}^2(x, y) = 30 - 40 = -10$

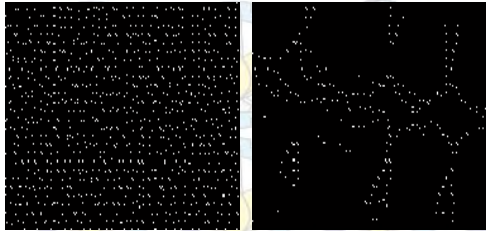
20	30	40
50	(x,y)	60
60	40	50

$d = 3, D_{mn}^3(x, y) = 20 - 50 = -30$

Gambar 7. Selisih *gray level* sesuai dengan arah yang digunakan

Setiap koordinat titik dalam suatu blok mempunyai satu nilai $D_{mn}(x, y)$. Kemudian, pada tiap blok nilai $D_{mn}(x, y)$ diranking dan nilai terbesar dipilih dan didefinisikan sebagai T_{mn} . Setiap blok memiliki satu nilai T_{mn} . T_{mn} inilah kandidat target dalam tiap blok. T_{mn} merupakan representasi koordinat titik yang memiliki selisih nilai *gray level* terbesar terhadap tetangga-tetangganya.

Kemudian, sebuah histogram dibuat dari nilai-nilai T_{mn} dari tiap blok. Dari histogram tersebut nilai *threshold global*, didefinisikan sebagai T_g , dapat ditentukan. Sekitar seperempat dari jumlah total blok dipilih sebagai target pada jalur tepi objek [3]. *Threshold* ini bertujuan untuk mendapatkan target pada jalur tepi yang mana lebih mendekati bentuk kontur. Gambar 8 menunjukkan hasil *thresholding* dari target-target kandidat menjadi target-target pada jalur tepi.



Gambar 8. Hasil thresholding dari target kandidat

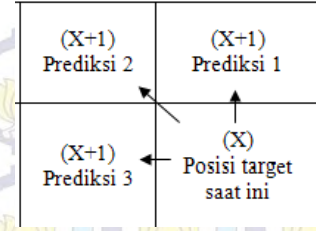
Gambar tersebut menunjukkan bahwa target-target yang didapat lebih mendekati bentuk kontur objek. Adapun proses *thresholding* yang dilakukan sebagai berikut:

- Jika nilai T_{mn} suatu blok lebih besar dari T_g , $D_{mn}(x, y)$ diranking pada empat arah untuk mencari nilai terbesar dan arah yang bersangkutan. Karena arah pada tepi objek tegak lurus dengan arah dari nilai terbesar $D_{mn}(x, y)$, arah dari target pada blok tersebut didefinisikan sebagai $(d+2)$ [12]. Koordinat target dan arahnya ditampung pada sebuah variabel W .
- Jika nilai T_{mn} suatu blok lebih kecil dari T_g . Maka blok tersebut tidak memiliki target pada tepi objek.

4 PELACAKAN TARGET

Setelah target-target di sepanjang jalur tepi didapatkan, langkah berikutnya adalah melacak pergerakan setiap target untuk membentuk suatu kontur tertutup. Sebuah variabel W menampung koordinat dan arah target-target di sepanjang jalur tepi. W dapat direpresentasikan sebagai $W = ((x, y), d)$, dimana (x, y) adalah koordinat target dan d adalah arah pergerakannya. Proses pembentukan kontur dilakukan pada setiap target. Setiap target adalah representasi titik kontur awal (initial contour point) sehingga satu target dapat menghasilkan beberapa titik kontur.

Dimulai dari satu target dengan arahnya, dapat diprediksi pergerakan ke titik kontur berikutnya. Diasumsikan bahwa target tidak memiliki transisi atau pembelokan yang tajam (*sharp transition*) sehingga pencarian titik kontur berikutnya dilakukan terhadap tiga arah seperti yang ditunjukkan oleh Gambar 9. Tiga arah yang diprediksi adalah $[(8+d-1) \bmod 8]$, d , $[(8+d+1) \bmod 8]$.

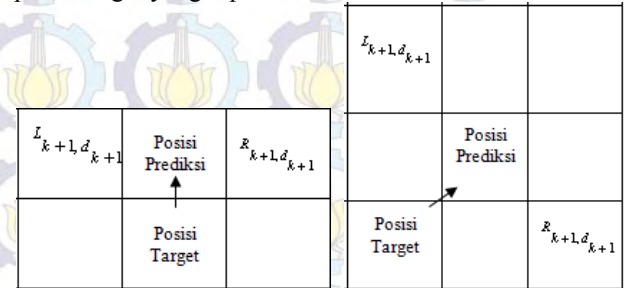


Gambar 9. Contoh prediksi pergerakan target dalam 3 arah ketika nilai $d=3$

Dalam tiga arah tersebut, dapat ditentukan sisi kanan dan sisi kiri dari posisi pergerakan target yang diprediksi. Penentuan sisi kanan dan sisi kiri didapatkan dengan Persamaan (8):

$$\begin{aligned} R_{k+1, d_{k+1}} &= (x_k - \text{round}(2 \sin(d_{k+1} - 1 \times \frac{\pi}{4})), y_k + \text{round}(2 \cos(d_{k+1} - 1 \times \frac{\pi}{4}))) \dots (8) \\ L_{k+1, d_{k+1}} &= (x_k - \text{round}(2 \sin(d_{k+1} + 1 \times \frac{\pi}{4})), y_k + \text{round}(2 \cos(d_{k+1} + 1 \times \frac{\pi}{4}))) \end{aligned}$$

Dimana R dan L adalah sisi kanan dan kiri dari prediksi posisi target selanjutnya, k adalah titik kontur ke- k dan d_k adalah arah dari target pada titik kontur ke- k . Gambar 10 menunjukkan bahwa R dan L selalu tegak lurus dengan posisi target yang diprediksi.



Gambar 10. Contoh posisi prediksi target dengan sisi kiri dan kanannya

Penentuan R dan L dari posisi target berikutnya yang diprediksi bertujuan untuk memeriksa nilai rata-rata *gray level* pada sisi kanan dan kiri untuk memperoleh titik kontur yang paling tepat dan sesuai dengan jalur tepi objek. Pemeriksaan nilai rata-rata *gray level* pada sisi kanan dan kiri dilakukan dengan Persamaan (9), (10) dan (11).

$$\bar{L} = \frac{1}{k} \sum_{p=0}^k L_{k-p, d_{k-p}} \dots (9)$$

$$\bar{R} = \frac{1}{k} \sum_{p=0}^k R_{k-p, d_{k-p}} \dots (10)$$

$$\Delta_{k+1}(i) = \left| I(L_{k+1, d_{k+j+i}}) - I(R_{k+1, d_{k+j+i}}) \right| - \left| I(L_{k+1, d_{k+j+i}}) - \bar{L} \right| - \left| I(R_{k+1, d_{k+j+i}}) - \bar{R} \right| \dots (11)$$

Dimana $i=1, 2, \dots, q$ dan $j=(q+1)/2$. Pada awalnya q bernilai 3 sebagai representasi tiga arah yang diprediksi. Nilai rata-rata *gray level* sisi kiri dan kanan digunakan untuk mencegah gangguan *noise*. Bagian kedua dan ketiga pada Persamaan (11) dapat berfungsi untuk

mencegah persamaan mencari prediksi kontur yang salah akibat gangguan *noise*.

Setelah didapat tiga nilai $\Delta_{k+1}(i)$, tiga nilai tersebut diurutkan untuk memilih nilai terbesar dan mendefinisikannya sebagai $\max(\Delta_{k+1}(i))$.

- Jika nilai $\max(\Delta_{k+1}(i))$ lebih besar dari nilai *local threshold* (TI), maka arah yang benar telah ditemukan. Dalam hal ini karena *global threshold* (Tg) adalah penjumlahan selisih nilai *gray level* dalam empat arah, maka nilai *local threshold* (TI) untuk selisih nilai *gray level* dalam satu arah didefinisikan sebagai $Tg/4$.
- Jika nilai $\max(\Delta_{k+1}(i))$ lebih kecil dari nilai *local threshold* (TI), maka kemungkinan pencarian arah sebelumnya telah menyimpang dari arah yang benar. Sehingga pencarian prediksi posisi target kembali dilakukan dari awal namun dengan tujuh arah.

Arah dari nilai $\max(\Delta_{k+1}(i))$ yang memenuhi *local threshold* (TI) digunakan untuk mendapatkan posisi target berikutnya (koordinat titik kontur) yang sebenarnya. Koordinat dan arah tersebut digunakan untuk mendapatkan koordinat dan arah titik kontur berikutnya. Untuk mendapatkan arah dan posisi sebenarnya digunakan Persamaan (12):

$$d_{k+1} = d_k + j + i \quad (12)$$

$$W_{k+1,d_{k+1}} = (x_k - \text{round}(\sin(d_{k+1} \times \frac{\pi}{4}), (y_k + \text{round}(\cos(d_{k+1} \times \frac{\pi}{4}))))$$

Jika titik kontur yang didapatkan memiliki posisi yang sama dengan titik kontur yang telah didapatkan sebelumnya atau telah melewati ukuran citra, maka proses pencarian titik kontur dari satu target pada jalur tepi objek telah selesai. Ubah d menjadi $d+4$ untuk melakukan proses pencarian titik kontur pada arah yang berlawanan. Kemudian, proses berlanjut pada target-target yang lainnya sampai semua target di sepanjang tepi objek telah diproses. Namun, jika kedua kondisi tersebut tidak terpenuhi, maka proses berlanjut untuk mencari titik kontur berikutnya.

5 METODE

Langkah-langkah yang diterapkan dalam algoritma deteksi tepi kontur berbasis pelacakan target secara dinamis ditunjukkan oleh Gambar 11. Gambar 11 menunjukkan perancangan proses sistem secara umum. Sistem akan menerima data masukan berupa citra grayscale dan penentuan jumlah operator arah untuk proses deteksi tepi. Kemudian akan dilakukan proses deteksi tepi dengan deteksi tepi Canny untuk menentukan tepi objek yang akan menjadi jalur pencarian target di sepanjang tepi objek. Setelah proses deteksi tepi, proses pelacakan dan penentuan target akan dilakukan di sepanjang tepi objek sampai membentuk kontur tertutup.

6 UJI COBA DAN PEMBAHASAN

6.1 Uji Coba Terhadap Citra Sintetis dengan Ground Truth

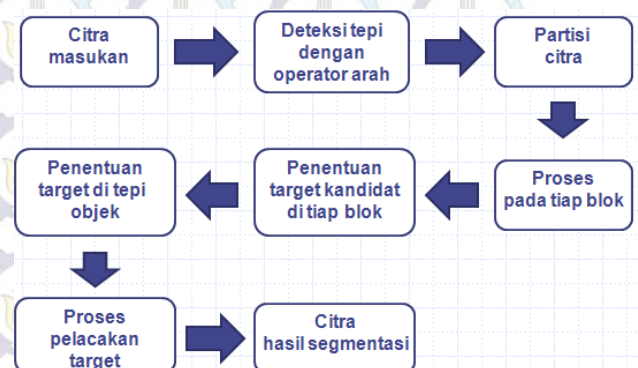
Pada uji coba ini proses segmentasi dilakukan terhadap citra sintetis dimana citra memiliki *ground truth* sebagai acuan hasil segmentasi. Pada pengujian digunakan beberapa operator arah pada proses deteksi tepi untuk dilihat hasilnya pada citra keluaran segmentasi. Operator arah yang digunakan adalah 1 operator arah dengan interval sudut 180° , 2 operator arah dengan interval sudut 90° , 3 operator arah dengan interval sudut 60° , 6 operator arah dengan interval sudut 30° dan 9 operator arah dengan interval sudut 20° .

Citra hasil segmentasi akan dihitung akurasi terhadap citra *ground truth* sesuai dengan Persamaan 14.

$$\text{Akurasi} = \frac{TP + TN}{TP + TN + FP + FN} \dots \dots \dots (14)$$

Dimana:

- TP: Pada C1 dianggap batas objek, pada C2 dianggap batas objek
- TN: Pada C1 dianggap bukan batas objek, pada C2 dianggap bukan batas objek
- FP: Pada C1 dianggap batas objek, pada C2 dianggap bukan batas objek
- FN: Pada C1 dianggap bukan batas objek, pada C2 dianggap batas objek

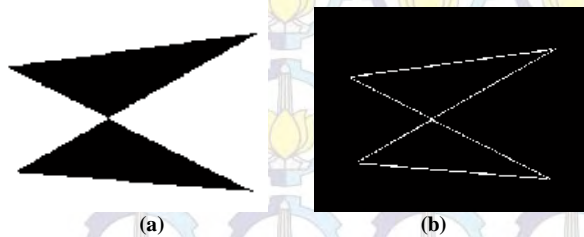


Gambar 11. Diagram Proses Segmentasi Citra dengan Deteksi Tepi Kontur Berbasis Pelacakan Target Secara Dinamis

Cara pengukuran tingkat akurasi yaitu dengan menentukan kesamaan *perpixel* C1 dan C2. Selain itu citra hasil segmentasi dilakukan pengamatan secara visual untuk melihat jumlah objek dalam citra yang berhasil disegmen. Citra keluaran hasil percobaan pada citra sintetis dapat dilihat pada Gambar 12. Sedangkan data hasil segmentasi pada citra sintetis ditunjukkan oleh Tabel 1.

Hasil uji coba menunjukkan bahwa algoritma yang diterapkan dapat menghasilkan tingkat akurasi segmentasi yang cukup tinggi. Dari citra sintetis yang diuji, objek pada citra berhasil disegmen dengan baik sehingga tidak ada objek citra yang terlewatkan.

Pengubahan parameter operator arah pada deteksi tepi juga mempengaruhi hasil akurasi segmentasi.



Gambar 12. Citra Sintetis 1 (a) Citra Masukan; dan (b) Citra Hasil Segmentasi

Dari Tabel 1 dapat dilihat bahwa terjadi penurunan tingkat akurasi segmentasi pada penggunaan 3, 6 dan 9 operator arah. Hal ini disebabkan hasil deteksi tepi objek pada ketiga operator arah tersebut menghasilkan garis yang lebih tebal di sekitar objek sehingga dapat menghasilkan target di sepanjang tepi objek yang lebih banyak. Akibatnya, proses pelacakan target menghasilkan titik-titik kontur yang mempengaruhi penambahan jumlah pixel tepi pada citra hasil segmentasi sehingga dapat mengurangi tingkat akurasi.

Tabel 1 Hasil Segmentasi Citra Sintetis

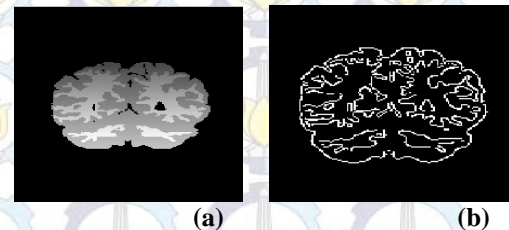
Nama Citra	Jumlah Operator Arah	Jumlah Objek pada Citra	Jumlah Objek Tersegmen	Tingkat Akurasi (%)
Citra Sintetis 1	1	2	2	97.4
	2			97.8
	3			97,1
	6			97,1
	9			97.2
Citra Sintetis 2	1	1	1	97.9
	2			97.9
	3			97.2
	6			97.2
	9			97.9

6.2 Uji Coba Terhadap Citra Tes MRI

Pada uji coba ini proses segmentasi dilakukan terhadap citra tes MRI. Pada pengujian juga digunakan beberapa operator arah pada proses deteksi tepi untuk dilihat hasilnya pada citra keluaran segmentasi. Karena citra tes MRI tidak terdapat *ground truth*, maka pengenalan jumlah objek dalam citra dan jumlah objek hasil segmentasi dilakukan secara visual. Citra keluaran hasil percobaan pada citra tes MRI dapat dilihat pada Gambar 13. Sedangkan data hasil segmentasi pada citra tes MRI ditunjukkan oleh Tabel 2.

Hasil uji coba menunjukkan bahwa jumlah operator arah pada deteksi tepi kembali mempengaruhi hasil segmentasi. Hal tersebut dapat dilihat pada Citra Tes MRI, deteksi tepi Canny dengan 1 operator arah tidak mampu melakukan segmentasi objek citra dan gagal membentuk kontur tertutup. Sama halnya dengan Citra

Tes MRI, deteksi tepi Canny dengan 1 dan 2 operator arah juga tidak mampu melakukan segmentasi objek citra dan gagal membentuk kontur tertutup.



Gambar 13. Citra Tes MRI 1 (a) Citra Masukan; dan (b) Citra Hasil Segmentasi

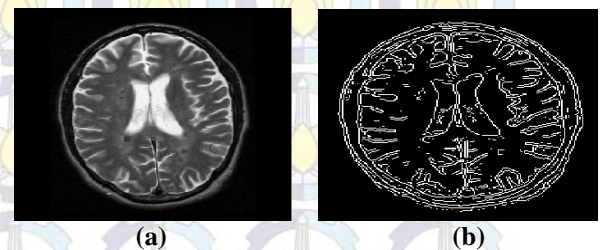
Secara keseluruhan, algoritma pelacakan target tidak mampu untuk melakukan segmentasi otak bagian dalam. Hal itu dikarenakan otak bagian dalam banyak memiliki *sharp transition* sehingga proses pelacakan target sering menyimpang dan gagal membentuk kontur tertutup.

Tabel 2 Hasil Segmentasi Citra Tes MRI

Nama Citra	Jumlah Operator Arah	Jumlah Objek pada Citra	Jumlah Objek Tersegmen	Kontur Tertutup
Citra Tes MRI 1	1	5	0	Tidak
	2		1	Ya
	3		1	Ya
	6		1	Ya
	9		1	Ya
Citra Tes MRI 2	1	6	0	Tidak
	2		0	Tidak
	3		1	Ya
	6		1	Ya
	9		1	Ya

6.3 Uji Coba Terhadap Citra MRI

Pada uji coba ini proses segmentasi dilakukan terhadap citra MRI asli. Berbeda dengan citra tes MRI yang merupakan citra simulasi atau citra sintetis, citra MRI asli memiliki perubahan gradien derajat keabuan yang halus. Pada citra MRI tidak terdapat *ground truth*. Oleh karena itu, pengenalan jumlah objek dalam citra dan jumlah objek hasil segmentasi dilakukan secara visual. Citra keluaran hasil percobaan pada citra MRI dapat dilihat pada Gambar 14.

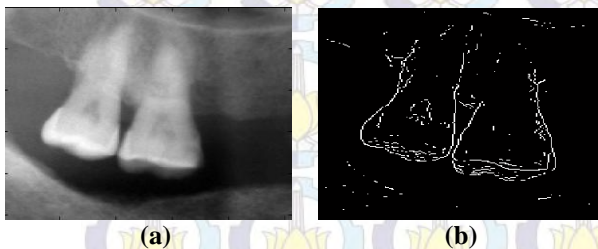


Gambar 14. Citra MRI 1 (a) Citra Masukan; dan (b) Citra Hasil Segmentasi

Pada uji coba ini algoritma pelacakan target tidak mampu melakukan segmentasi objek dari citra MRI dan membentuk kontur tertutup. Meskipun terjadi peningkatan kualitas visual citra seiring dengan bertambahnya jumlah operator arah pada deteksi tepi Canny, perbedaan derajat keabuan yang halus dapat mempengaruhi penentuan nilai *threshold* yang digunakan dalam proses pencarian titik-titik kontur yang baru. Sehingga prediksi arah yang didapatkan dan digunakan dalam pelacakan target tidak mampu membuat bentuk kontur yang tertutup.

6.4 Uji Coba Terhadap Citra Gigi

Pada uji coba ini proses segmentasi dilakukan terhadap citra gigi. Mirip dengan karakteristik citra MRI, citra gigi memiliki perubahan gradien derajat keabuan yang halus. Pada citra gigi tidak terdapat *ground truth*. Oleh karena itu, pengenalan jumlah objek dalam citra dan jumlah objek hasil segmentasi dilakukan secara visual. Citra keluaran hasil percobaan pada citra gigi dapat dilihat pada Gambar 15.



Gambar 15. Citra Gigi 1 (a) Citra Masukan; dan (b) Citra Hasil Segmentasi

Sama dengan pengujian terhadap citra MRI, pada uji coba ini kembali menunjukkan bahwa algoritma pelacakan target tidak mampu melakukan segmentasi objek dari citra gigi dan membentuk kontur tertutup. Terjadi peningkatan kualitas visual citra seiring dengan bertambahnya jumlah operator arah pada deteksi tepi Canny. Namun, karakteristik citra yang dimiliki yaitu perbedaan nilai derajat keabuan antara objek (*foreground*) dan bagian latar (*background*) yang sedikit dapat mempengaruhi penentuan nilai *threshold* yang digunakan dalam proses pencarian titik-titik kontur yang baru. Sehingga prediksi arah digunakan dalam pelacakan target tidak dapat membuat bentuk kontur yang tertutup.

Dari keempat percobaan yang dilakukan, mulai dari uji coba terhadap citra sintetis dengan *ground truth* sampai dengan uji coba terhadap citra gigi, terdapat semacam kontradiksi antara hasil akurasi segmentasi dengan hasil visual. Dari uji coba didapatkan sebuah kesimpulan bahwa semakin banyak jumlah operator arah pada deteksi tepi, semakin rendah tingkat akurasi segmentasi. Hal ini disebabkan dengan bertambahnya jumlah operator arah, proses deteksi tepi dapat menghasilkan garis yang lebih tebal di sekitar objek sehingga dapat menghasilkan target di sepanjang tepi objek yang lebih banyak. Akibatnya, proses pelacakan

target menghasilkan titik-titik kontur yang mempengaruhi penambahan jumlah *pixel* tepi pada citra hasil segmentasi sehingga dapat mengurangi tingkat akurasi. Namun, pada saat yang bersamaan semakin banyak jumlah operator arah, semakin baik kualitas visual hasil segmentasi. Akibat kontradiksi inilah dilakukan skenario uji coba tambahan. Pada deteksi tepi dilakukan operasi *thinning* sehingga hasil deteksi tepi berupa *local maxima* dan tepi objek merupakan tepi sesungguhnya (*true edges*). Kemudian, hasil deteksi tepi akan dibandingkan dengan citra *ground truth* untuk dihitung akurasinya. Ketebalan garis tepi objek citra *ground truth* sendiri dibuat menjadi dua jenis yaitu ketebalan 1 *pixel* dan 2 *pixel*.

6.5 Uji Coba Deteksi Tepi dengan Thinning Terhadap Citra Sintetis dengan Ground Truth

Pada uji coba ini dilakukan proses *thinning* pada deteksi tepi sehingga hasil deteksi tepi berupa *local maxima* dan tepi objek merupakan tepi sesungguhnya (*true edges*). Hasil deteksi tepi kemudian akan dibandingkan dengan citra *ground truth* untuk dihitung akurasinya. Pada percobaan pertama, dilakukan deteksi tepi dengan *thinning* terhadap dua citra sintetis. Perbandingan hasil akan dilakukan terhadap citra *ground truth* dengan ketebalan tepi objek 1 *pixel*. data hasil deteksi tepi pada citra sintetis ditunjukkan oleh Tabel 3.

Tabel 3 Hasil Uji Coba Deteksi Tepi Terhadap Ground Truth

Nama Citra	Jumlah Operator Arah	Tingkat Akurasi (%)
Citra Sintetis 1	1	92.5
	2	95.8
	3	96.0
	6	96.2
	9	96.2
Citra Sintetis 2	1	93.3
	2	96.4
	3	96.8
	6	97.2
	9	97.3

Dari hasil uji coba yang didapatkan terhadap kedua citra sintetis, tingkat akurasi terhadap *ground truth* meningkat seiring dengan bertambahnya jumlah operator arah pada deteksi tepi. Hal ini sesuai dengan peningkatan kualitas citra hasil deteksi tepi yang semakin baik seiring dengan bertambahnya jumlah operator arah. Pada penggunaan 1, 2, 3 dan 6 operator arah terjadi peningkatan tingkat akurasi yang cukup signifikan. Namun, pada peralihan penggunaan dari 6 ke 9 operator arah peningkatan akurasi yang dihasilkan tidak signifikan, hanya berkisar 0.06% saja.

6.6 Pembahasan

Dari data percobaan dengan menggunakan beberapa citra uji untuk proses segmentasi, terlihat bahwa algoritma pelacakan target mampu melakukan segmentasi

objek dengan baik dengan tingkat akurasi yang cukup tinggi pada citra sintetis. Namun, ketika algoritma diterapkan pada citra uji berupa citra medis *real image*, algoritma pelacakan target tidak mampu mengulangi kehandalan performa seperti yang ditunjukkan pada citra sintetis. Karakteristik citra medis yang memiliki nilai derajat keabuan yang tidak variatif (halus) dapat mempengaruhi proses awal segmentasi yaitu deteksi tepi. Peran deteksi tepi pada algoritma pelacakan target yang berbasis teknik *edge-following* cukup vital karena deteksi tepi menyediakan jalur untuk target kandidat di sepanjang jalur tepi. Apabila deteksi tepi yang didapatkan kurang baik, maka jumlah target kandidat di sepanjang jalur tepi akan tereduksi sehingga proses pelacakan target tidak dapat membentuk suatu kontur tertutup.

Karakteristik derajat keabuan yang dimiliki citra medis juga dapat mempengaruhi penentuan nilai *thresholding* untuk menentukan target kandidat dan proses pencarian titik-titik kontur sehingga pada proses pencarian titik kontur akan didapatkan arah prediksi titik kontur berikutnya yang tidak sesuai. Hal tersebut dapat menyebabkan ditemukannya titik-titik kontur yang menyimpang dari jalur tepi objek.

Dari pengujian juga terlihat bahwa peningkatan jumlah operator arah pada proses deteksi tepi dapat meningkatkan kualitas visual pada proses segmentasi. Hal ini disebabkan dengan bertambahnya jumlah operator arah, proses deteksi tepi dapat menghasilkan garis yang lebih tebal di sekitar objek sehingga dapat menghasilkan target di sepanjang tepi objek yang lebih banyak. Hal ini didukung dengan uji coba tambahan berupa deteksi tepi citra dengan *thinning* terhadap citra sintetis. Pada uji coba tambahan, tingkat akurasi deteksi tepi terhadap *ground truth* meningkat seiring dengan bertambahnya jumlah operator arah pada deteksi tepi. Hal ini sesuai dengan peningkatan kualitas citra hasil deteksi tepi yang semakin baik seiring dengan bertambahnya jumlah operator arah.

Meskipun algoritma pelacakan target yang diimplementasikan dalam makalah ini belum dapat melakukan segmentasi citra medis secara optimal, algoritma ini memiliki kelebihan yaitu seluruh proses segmentasi dilakukan secara otomatis. Keterlibatan pengguna tidak dibutuhkan dalam penentuan titik awal segmentasi, penentuan *thresholding* dan proses pelacakan target.

7 KESIMPULAN

Dari uji coba yang telah dilakukan dan setelah melakukan analisis hasil pengujian terhadap implementasi segmentasi citra medis dengan algoritma deteksi tepi kontur berbasis pelacakan target secara dinamis ini dapat diambil beberapa kesimpulan, antara lain:

1. Algoritma deteksi tepi kontur berbasis pelacakan target secara dinamis dapat melakukan seluruh proses segmentasi citra secara otomatis. Hal ini dibuktikan dengan tidak dibutuhkannya keterlibatan pengguna selama proses segmentasi.

2. Peningkatan jumlah operator arah pada proses deteksi tepi dapat memperbaiki kualitas visual pada hasil segmentasi. Hal ini disebabkan dengan bertambahnya jumlah operator arah, proses deteksi tepi dapat menghasilkan garis yang lebih tebal di sekitar objek sehingga dapat menghasilkan target di sepanjang tepi objek yang lebih banyak.
3. Tingkat akurasi deteksi tepi terhadap *ground truth* meningkat seiring dengan bertambahnya jumlah operator arah pada deteksi tepi. Hal ini sesuai dengan peningkatan kualitas citra hasil deteksi tepi yang semakin baik seiring dengan bertambahnya jumlah operator arah.
4. Algoritma pelacakan target yang dikembangkan pada makalah ini belum dapat melakukan segmentasi citra medis secara optimal. Proses *thresholding* yang digunakan untuk pencarian dan pembentukan kontur tertutup kurang handal pada citra yang memiliki karakteristik nilai derajat keabuan yang tidak variatif.

8 SARAN

Adapun saran yang disampaikan penulis untuk pengembangan lebih lanjut dari makalah ini antara lain:

1. Perlu dilakukan pengembangan metode pelacakan target yang lebih efektif, terutama dalam proses penentuan nilai *thresholding* sehingga dapat menangani citra dengan karakteristik nilai derajat keabuan yang tidak variatif.
2. Perlu dilakukan skenario uji coba yang lebih variatif dan mendalam untuk menguji kehandalan dan tingkat akurasi algoritma pelacakan target ini.

REFERENSI

- [1] Fu, K. S. and Mui, J. K. [1981]. *A Survey on Image Segmentation*. Pattern Recognition 3-16.
- [2] Teaching Module from Laboratory of Computer Vision and Media Technology [2001]. *Canny Edge Detector*. Aalborg University, Denmark.
- [3] Chen, Y. B. and Chen, Oscar T.-C. [2003]. *Robust Fully Automatic Segmentation based on Modified Edge-Following Technique*. Proc. of ICASSP, vol. 3, pp. 333-336.