

# APLIKASI PENDETEKSI OBYEK PADA CITRA DIGITAL DENGAN MENGGUNAKAN METODE GLOBAL CONTOUR SHAPE (GCS)

Eko Suwono – Rully Soelaiman, S.Kom, M.Kom

Fakultas Teknologi Informasi,  
Institut Teknologi Sepuluh Nopember (ITS), Surabaya, 60111, Indonesia  
E-mail: ekoes@cs.its.ac.id

**Abstraksi--** Penggunaan metode global contour shape untuk mendeteksi obyek pada citra digital terbukti lebih sesuai terhadap beragam skala dan rotasi. Diawali dengan segmentasi pada citra, boundary pada setiap region diambil sebagai input untuk proses selanjutnya. Proses segmentasi dilakukan menggunakan metode statistical region merging (SRM). Untuk mendeskripsikan obyek pada citra digunakan sekuen sudut tangen yang diambil dari vertek-vertik pada boundary region hasil segmentasi. Sekuen sudut tangen pada template obyek dan sekuen sudut tangen pada citra digunakan untuk mencari distance antar keduanya. Perhitungan distance antar dua shape obyek tersebut digunakan algoritma yang efisien yaitu dynamic programming. Kemudian untuk dapat mendeteksi obyek inputan yang memiliki similaritas yang tinggi digunakan algoritma multi-start gradient descent. Algoritma ini digunakan untuk menghitung distance dari region-region pada citra inputan dan juga distance dari kombinasi antara region-region tersebut. Kandidat obyek yang memiliki similaritas yang rendah dihapus. Kandidat yang memiliki similaritas yang tinggi dengan obyek template akan dilakukan pengecekan overlapping object. Kandidat yang memiliki nilai similaritas lebih rendah dan nilai overlap lebih tinggi dari overlap threshold akan dihapus. Dengan demikian, hanya kandidat yang memiliki similaritas yang tinggi dan tidak saling overlap yang akan dideteksi.

**Kata kunci:** deteksi obyek, contour matching, statistical region merging, dynamic programming, probabilistic shape distance, region grouping

## I. PENDAHULUAN

Pendeteksian dan pengenalan obyek merupakan bagian dari pengenalan pola yang akhir-akhir ini mendapatkan banyak perhatian dalam *computer vision*. Pada beberapa tahun terakhir, pendekatan dalam melakukan pengenalan obyek kebanyakan menggunakan *local appearance*. Suatu kelas obyek direpresentasikan oleh kumpulan *visual stimuli* yang lebih kecil, yang masing-masing dihubungkan oleh model konfigurasi ("*part-based models*"), atau tanpa menggunakan informasi posisi relatif ("*bag-of-features models*"). Hasil yang menakjubkan telah dicapai, sebagian berhubungan dengan metode baru yang mendeskripsikan tentang penggunaan *local appearance*.

*Local appearance* bukanlah satu-satunya kunci dalam mendeteksi obyek. Faktanya beberapa kelas obyek *local appearance*-nya berisi sedikit sekali informasi tentang kelas tersebut, tetapi kelas obyek tersebut dapat dengan mudah dikenali melalui bentuk konturnya. Pendeteksian menggunakan bentuk telah dilakukan pada penelitian sebelumnya. Ide dasar untuk semua metode adalah untuk mendefinisikan jarak pengukuran antar bentuk. Kemudian mencoba untuk menemukan minima dari jarak tersebut. Pada metode klasik (*chamfer matching*), jarak didefinisikan sebagai jarak rata-rata dari titik pada *template shape* ke titik terdekat pada *image shape*.

Bagaimanapun juga, metode tersebut (*chamfer matching*) tidak dapat menangani dengan baik masalah *clutter* dan *shape deformations*. sekalipun sebuah hirarki dari banyak *template* telah digunakan untuk menangani masalah *deformations*, *rate of false positive*-nya masih lebih tinggi (pada umumnya  $> 1$  *false positive per image*, FPPI). metode yang lebih canggih membolehkan *shape template* untuk diubah, jadi dapat disesuaikan dengan isi citra, termasuk metode seperti *spline-based shape matching*, *diffusion snake*, dan *active shape models*. jarak pada kasus ini adalah energi deformasi (penambahan panjang) pada *template shape*. semua metode yang telah disebutkan (*deformable matching techniques*) adalah *local optimiser*, dan semua itu memerlukan initial posisi yang baik atau gambar yang bersih untuk menghindari *local minima*. pada paper ini, diajukan suatu strategi pencarian, yang mengurangi masalah kesalahan *local minima*, pada penetapan biaya untuk merevaluasi jarak bentuk pada masing-masing langkah pencarian. *Probabilistically motivated shape distance* yang dapat dihitung dengan efisien.

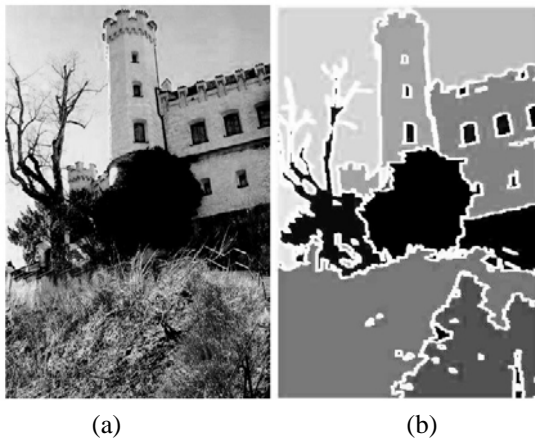
Baru-baru ini, peneliti telah beralih dari model asli (*original model*), yang mana mendeskripsikan seluruh bentuk dari obyek dalam satu kesatuan utuh yang disebut *Global shape models*[1]. Sebuah obyek yang terdiri dari beberapa bagian lokal dari kontur dalam suatu konfigurasi tertentu. Metode dengan menggunakan *fragment contour*

mempunyai keuntungan yaitu juga memodelkan posisi relatif mereka karena *short boundary fragments* tidak terlalu spesifik untuk di gunakan pada *bag-of-features* model, maka bentuk dari contour ini hanya berguna apabila di lihat secara utuh dalam bentuk obyek, sementara *local shape of fragments* hanya menyediakan sedikit informasi. Observasi terakhir ini menghasilkan keinginan baru untuk mencocokkan bagian-bagian contour yang besar. Disini, kami akan memberi contoh dari pendekatan dari sebuah obyek yang di deteksi menggunakan *global shape*, yang berdasarkan *elastic matching of contours*. Sebuah *edge map* dengan *closed edge chains* di peroleh dengan cara segmentasi menjadi *super-pixels*. Pertama-tama Sebuah *probabilistic* yang mengukur kecocokan antara dua kontur diperoleh, dan dikombinasikan dengan skema optimasi untuk mencari *closed contours* pada citra, yang mempunyai tingkat kemiripan yang tinggi dengan *template shape*. Metode ini hanya membutuhkan sebuah *template* obyek, yang dalam kasus Tugas Akhir ini adalah sebuah sketsa buatan tangan.

## II. STATISTICAL REGION MERGING

Segmentasi adalah proses membagi citra menjadi daerah-daerah yang terpisah, dimana piksel dalam satu region memiliki nilai yang homogen[2]. Metode segmentasi terbaru yang lebih cepat dan mengurangi biaya komputasi, yang dikembangkan untuk segmentasi citra berwarna yaitu Statistical Region Merging (SRM). Metode ini sejenis dengan teknik region growing and mergin. Pada region growing, region adalah kumpulan piksel yang homogen dan secara iterasi merupakan hasil gabungan dari region yang lebih kecil.

Notasi  $| \cdot |$  menandakan kardinalitas. Citra yang diobservasi  $I$ , berisi  $|I|$  piksel, yang masing-masing berisi nilai *Red-Green-Blue* (RGB), masing-masing dari tiga nilai tersebut memiliki nilai  $\{1,2,3,...,g\}$ , pada prakteknya  $g$  bernilai 256.



Gambar 1. (a) Citra Sebelum dan (b) Sesudah Segmentasi

$I$  adalah observasi dari scene utuh  $I^*$  yang belum diketahui piksel mana yang sebenarnya direpresentasikan oleh anggota distribusi, dari masing-masing channel warna yang dicoba. Pada

masing-masing piksel  $I^*$ , setiap *color-channel* diganti nilainya dengan nilai random sejumlah  $Q$ . Nilai random tersebut merupakan nilai positif yang diambil pada domain dengan batas  $g/Q$ , yang termasuk dalam  $\{1,2,...,g\}$ . Dalam *statistical region* bisa saja terdapat nilai piksel yang berbeda dengan syarat piksel tersebut masih memenuhi *homogeneity property*. Biasanya dipilih  $Q=1$  tetapi tidak fleksibel untuk *region* yang kecil. Karena  $Q$  setara dengan kompleksitas segmentasi yang dihasilkan maka  $Q$  dapat digunakan untuk mengatur kompleksitas segmentasi.

*Homogeneity Property* :

- *Inside a region, the statistical pixels have the same expectation for every color channel.*
- *Different regions have different expectations for at least one color channel.*

Dari *Homogeneity Property* tersebut Nielsen dan Nock [3] mendapatkan *merging predicate* berikut :

$$P(R, R') = \begin{cases} \text{true} & \text{if } |\bar{R}_a' - \bar{R}_a| \leq \sqrt{b^2(R) + b^2(R')} \\ \text{false} & \text{otherwise} \end{cases} \quad (1)$$

$$b(R) = g \sqrt{(1/2Q |R|) \ln(|R| / \delta)} \quad (2)$$

$\bar{R}_a$  menandakan nilai rata-rata channel warna  $a$  dalam *region*  $R$  sedangkan  $R_{|l|}$  jumlah *region* yang memiliki  $l$  piksel.

Urutan dimana tes penggabungan dilakukan mengikuti invarian sederhana  $A$  :

- Ketika ada dua *true region* yang didapatkan, itu berarti semua tes di dalam masing-masing region sebelumnya sudah dilakukan

Pada uji coba,  $A$  diwakili oleh algoritma sederhana yang berbasis pada gradien piksel yang bertetangga. Dalam keterangannya Nielsen dan Nock menggunakan fungsi  $f$  yang didefinisikan sebagai berikut :

$$f(p, p') = \max_{a \in R, G, B} fa(p, p') \quad (3)$$

Fungsi sederhana untuk  $fa$  adalah berikut ini :

$$fa(p, p') = |p_a - p'_a| \quad (4)$$

Fungsi yang lebih kompleks dengan menambahkan deteksi tepi klasik kernel konvolusi bisa digunakan untuk mendefinisikan  $fa$ .

Berikut ini pseudo-code yang sesuai dengan algoritma Nielsen dan Nock

```

INPUT: an image I
Let SI the set of the 4-connexity couples of adjacent
pixels
in image I
S0I = orderIncreasing(SI, f);
for i = 1 to |S0I| do
if (R(pi) 6= R(p0i)) and P(R(pi), R(p0i)) == true) then
merge(R(pi), R(p0i));

```

Himpunan pasangan piksel bertetangga ( $S_i$ ) diurutkan sesuai dengan nilai pada persamaan (3). Setelah itu algoritma tersebut mengunjungi setiap pasang piksel ( $p, po$ ) pada  $S_i$  dan jika *region* yang termasuk ( $R(p)$  dan  $R(po)$ ) bukan merupakan *region* yang sama dan memenuhi *merging predicate* (1), maka kedua *region* tersebut di-merge.

Algoritma SRM memberikan himpunan piksel pada masing-masing *region*. Dengan adanya *region* tersebut didapatkan tepi setiap *region* dengan melakukan pencari tepi untuk masing-masing *region*. Pencarian dilakukan dengan melakukan pengecekan ke delapan-arah mata angin.

### III. PEMODELAN BENTUK

Pemodelan bentuk digunakan untuk mendeskripsikan suatu obyek. bentuk  $X$  dimisalkan dengan poligon tertutup dengan jumlah vertek  $N$  yang memiliki jarak yang sama diantaranya. Karena jumlah titik sama dengan jumlah spasi diantaranya, maka urutan titik dapat diparameterkan dengan panjang busurnya :  $X = \{x(u), u=0, \dots, N-1\}$ . Vertek terakhir sama dengan vertek pertama  $x(N) = x(0)$ .

Sebagai shape descriptor  $X'$  dari bentuk  $X$ , digunakan sudut tangen yang berurutan. Sudut tangen dinormalisasikan dengan meluruskan vektor tangen pertama dengan axis  $x$ . Vektor tangen pada titik  $x(u)$  dinotasikan dengan  $\mathbf{x}(u) = [x(u), y(u)]$ , kemudian

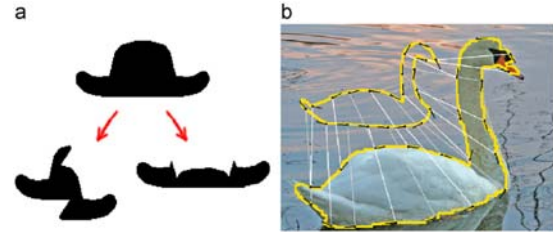
$$X'(u) = \{x'(u), u = 0, \dots, N-1\},$$

dimana

$$x'(u) = \arctan \frac{y(u)}{x(u)} - \arctan \frac{y(0)}{x(0)} \quad (5)$$

Sudut tangen lebih dipilih dari pada curvatures (kadang disebut “*turning angles*”) karena sudut tangen lebih tahan terhadap perubahan bentuk daripada curvatures. Sebagai gambaran akan ditunjukkan pada gambar 2.

Jika sudah ada citra yang telah tersegmentasi menjadi super-pixels, maka akan mudah untuk mendapatkan shape descriptor. Beberapa kelompok super-pixels yang bertetangga tanpa lubang memiliki kontur tertutup yang dapat diekstrak dengan menggunakan boundary-tracer, dan mengambil sejumlah  $N$  titik. Pada eksperimen kali ini,  $N$  ditentukan bernilai 100. karena dari hasil percobaan jumlah tersebut sudah mewakili bentuk keseluruhan.



Gambar 2. Representasi dan Matching Shape (a) *curvature* yang tidak cocok sebagai pemodelan bentuk. (b) *non linear elastic matching*, hanya 100 sudut tangen per kontur yang diplot

### IV. PENCOCOKAN BENTUK

Diberikan dua bentuk yaitu  $X$  dan  $Y$ , salah satu digunakan sebagai *shape template* dan yang lainnya sebagai kandidat kontur yang didapatkan dari citra uji-coba. Pencocokan antara kedua bentuk adalah sebuah fungsi, yang menghubungkan kumpulan titik  $\{x(u)\}$  dan  $\{y(v)\}$  (keduanya adalah parameter dari panjang busur masing-masing), masing-masing titik pada tiap kurva memiliki sedikitnya satu titik yang berkorespondensi pada kurva yang lain. Titik yang sama pada  $X$  dapat memiliki banyak kecocokan pada  $Y$  selama sekuen masing-masing masih terjaga. Pencocokan didapatkan dari sekuen panjang busur pada titik kontur pencocokan.

Misal diketahui satu pasang titik yang cocok pada dua kontur, misalkan panjang busur pada titik sekarang adalah  $(u=v=0)$ . Kemudian kecocokan dari dua kontur didapat dengan menggunakan rumus berikut:

$$V(X, Y) = \{(u_i \Leftrightarrow v_i), i = 1, \dots, K_v\} \\ = \{(0 \Leftrightarrow 0), \dots, (u_i \Leftrightarrow v_i), \dots, (N \Leftrightarrow N)\} \quad (6)$$

Kemungkinan bahwa dua titik cocok dapat dibagi dalam dua komponen, satu untuk smiliaritasnya dan satu lagi untuk *stretch* pada kurva:

$$P(u_i \Leftrightarrow v_i) = P_D(u_i \Leftrightarrow v_i) P_S(u_i \Leftrightarrow v_i)$$

penilaian antara dua titik dianggap lebih memiliki kesamaan jika sudut tangen lokal dari dua kurva sama, dan jika tidak ada *local stretching* yang diperlukan untuk membuat dua titik cocok. Untuk menghitung *stretching* dan untuk memastikan pencocokan yang lengkap dan terurut, didefinisikan

$$P_S(u_i \Leftrightarrow v_i) = \frac{1}{1 + e^{-E}} e^{-s(u_i, v_i)} \quad (7)$$

Dimana

$$S(u_i \Leftrightarrow v_i) = \begin{cases} 0 & \text{if } (u_{i-1} = u_i - 1, v_{i-1} = v_i - 1) \\ E & \text{if } (u_{i-1} = u_i, v_{i-1} = v_i - 1), \\ & \text{or } (u_{i-1} = u_i - 1, v_{i-1} = v_i) \\ \infty & \text{else} \end{cases}$$

(8)

dengan  $E$  non-negatif konstanta. Dua kasus yang pertama memastikan bahwa pencocokan dua titik tanpa *stretching* kurva memiliki probabilitas yang lebih tinggi daripada dengan *stretching*. Kasus yang ketiga memastikan hanya *valid assignment* yang dibuat, dengan melarang *assignment* yang menloncati  $u_i$  atau  $v_i$  dan yang tidak mengikuti urutan-urutan.

Kemudian akan dicari  $P_D(u_i \Leftrightarrow v_i)$  dengan kata lain harus ditentukan bagaimana mengukur *dissimilarity*  $D(u, v)$  antara dua titik. *Dissimilarity* ada fungsi tidak negative dari perbedaan sudut tangen, dengan interval  $[-\pi, \pi]$ . Rumus fungsi *dissimilarity*

$$P_D(u_i \Leftrightarrow v_i) = \frac{1}{H(B)} e^{-\frac{1}{B} D(u_i, v_i)}, \quad (9)$$

diberikan bentuk fungsional dari eksponen,  $H(B)$  konstan, konstanta  $B$  secara kualitatif tidak akan mengubah *cost function*, tetapi hanya mempengaruhi penalti pada *stretching contour*.  $B$  diberi nilai 1, dan digunakan distribusi *laplace*, sehingga  $H = 2(1 - e^{-\pi})$  dan  $D(u_i, v_i) = |x'(u) - y'(v)|$ . Total probabilitas matching  $V$ , mencapai minimum  $K_v = N$  jika tidak ada *stretching*, dan mencapai maksimum  $K_v = 2N$  jika semua *assignment* menghasilkan *local stretching*

$$P(V) = \prod_{i=1}^{K_v} P_D(u_i \Leftrightarrow v_i) \prod_{i=1}^{K_v} P_S(u_i \Leftrightarrow v_i) \quad (10)$$

Pada kombinasi kumpulan matching yang mungkin antara dua kontur, akan diambil satu yang disebut *best matching*. *Best matching* adalah satu-satunya yang mengecilkan *negative log-likelihood*

$$C(V) = \frac{1}{B} \sum_{i=1}^{K_v} D(u_i, v_i) + \sum_{i=1}^{K_v} (Q + S(u_i, v_i)), V.$$

Dimana,

$$Q = -\log(H(1 + e^{-E})) \quad (11)$$

Jika ingin meyakinkan bahwa hanya *valid matching* saja yang dibandingkan, maka kejadian  $S = \infty$  tidak akan pernah terjadi. Selanjutnya akan ada  $2(K_v - N)$  titik yang *match* dengan  $S = E$ , dan  $(2N - K_v)$  *match* dengan  $S = 0$ . Dengan menggunakan persamaan tersebut, dapat dikembangkan penjumlahan kedua persamaan 3.7 dengan mengurangi konstanta  $NQ$  dari *cost*, dan dikalikan dengan  $B$ , untuk mendapatkan *best matching* (*distance* antara dua bentuk).

$$\tilde{C}(X, Y) = \min_v [\sum_{i=1}^{K_v} D(u_i, v_i) + B(K_v - N)(Q + 2E)]$$

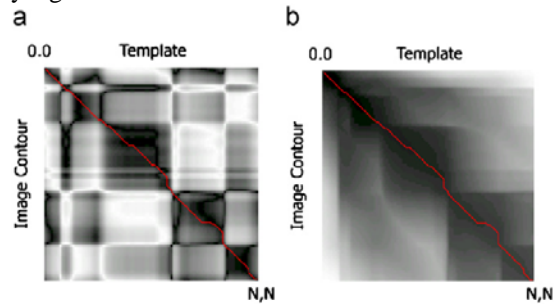
$$\text{s.t } u_1 = v_1 = 0, u_K = v_K = N,$$

$$\forall i : u_{i+1} - u_i \in \{0, 1\},$$

$$v_{i+1} - v_i \in \{0, 1\} \quad (12)$$

*Cost function* adalah bentuk yang efisiensi dengan menemukan *best matching* yang memberikan pasangan titik awal menggunakan *dynamic programming*[4]. Dengan kompleksitas  $O(N^2)$ , pendefinisian jarak dengan cara ini dikenal dalam pengenalan pola sebagai “*nonlinear elastic matching*” dengan *stretch cost*  $R$ , disingkat  $NEM_R$ , dimana *stretch cost*  $R = \frac{1}{2} B(Q + 2E)$ .

Sejauh ini masih dibutuhkan untuk mengetahui pasangan *matching point* yang dipilih sebagai titik awal  $x(0)$  dan  $y(0)$ . Masalah ini dapat diselesaikan dengan mudah dengan melakukan tes rotasi relatif yang berbeda secara berulang-ulang. Pada waktu yang sama dihasilkan beragam *cost* pada setiap rotasi. Agar dapat berotasi  $X$  relatif terhadap kontur  $Y$ , sudut tangen  $x'(u)$  digeser dengan cara melingkar, kemudian dinormalisasi seperti  $x'(0) = 0$  untuk titik awal yang baru.



Gambar 3 (a) matrik *dissimilarity*, (b) matrik *cumulative cost*

## MENGHITUNG SHAPE DISTANCE

Pencarian *matching*  $\tilde{C}(X, Y)$  antara dua kelompok yang berurutan yang dihasilkan dalam *minimum cost* adalah aplikasi klasik dari *dynamic programming*. Pada umumnya, masalah pencocokan diubah menjadi *path-search problem* pada *directed graph*, dimana masing node merepresentasikan pasangan titik yang berkorespondensi.  $\{u_i \Leftrightarrow v_i\}$ . Sekarang akan diberikan penjelasan singkat secara intuitif daripada perumusan *graph-theoretic* secara mendetail yang tidak praktis. Sekali lagi, misalkan bahwa satu pasang *matching point*  $\{u = 0, v = 0\}$  diketahui.

Langkah pertama untuk menghitung  $(N \times N)$  *dissimilarity* matrik  $D$ , dengan *dissimilaritas*  $d_{ij} = D(u_i, v_j)$  antara semua pasang vertek yang mungkin  $(u_i \Leftrightarrow v_i)$ . Diketahui bahwa titik yang

pertama harus sama dengan titik terakhir:  $(0 \Leftrightarrow 0), (N \Leftrightarrow N)$ . Dengan demikian tugas sekarang adalah menemukan path dari  $d_{00}$  sampai  $d_{nn}$  yang memiliki harga kumulatif lebih rendah untuk *dissimilarity* dan *stretching*. Selanjutnya, setiap vertex pada masing-masing kontur harus memiliki sedikitnya satu *matching*. Dengan syarat urutan vertek harus tetap terjaga. Jadi pada titik lain  $\{i, j\}$  sepanjang kontur ini ada tiga kemungkinan kelanjutan:

- (1) mencocokkan pasangan vertek berikutnya tanpa stretching kontur, dengan bergerak ke titik  $\{i + 1, j + 1\}$ ;
- (2) melakukan *stretch* X: mencocokkan vertek berikutnya ke vertek pada sumbu Y, dengan bergerak ke titik  $\{i + 1, j\}$ ;
- (3) melakukan *stretch* Y: mencocokkan vertek berikutnya ke vertek pada sumbu X, dengan bergerak ke titik  $\{i, j + 1\}$ ;

Ada tiga langkah yang mungkin yang secara langsung mendefinisikan rekursi untuk matrik *cumulative matching cost*  $\mathcal{P}$  lihat algoritma 1. Distance antara dua kontur adalah total matching cost  $r_{NN}$ , yang telah diakumulasi ketika kembali pada titik awal. Titik yang diambil dalam perhitungan tidak dibutuhkan untuk diketahui tetapi dapat dengan mudah ditemukan dengan melakukan back-tracking dari  $r_{NN}$  ke  $r_{11}$ .

**Algoritma 1.** dynamic programming rekursi untuk menemukan shape distance  $\tilde{C}(X, Y)$ .

**Input** : Shape descriptors  $\{x'(u)$  dan  $\{y'(v)\}$

**Output** : ShapeDistance  $\mathcal{P}$

```

Allocate DNxN, RNxN;
//compute vertex dissimilarities
for i=0 to N do
    for j=0 to N do
        di,j ← D(ui,vj);
    end
end
//compute cumulative matching cost
r0,0 ← d0,0;
for i=1 to N do
    ri,0 ← ri-1,0 + di,0 + R;
    r0,i ← r0,i-1 + d0,i + R;
end
for i=1 to N do
    for j=1 to N do
        h10 ← ri-1,j + R;
        h01 ← ri,j-1 + R;
        h11 ← ri-1,j-1;
        rij ← min(h10,h01,h11)+dij;
    end
end
shapeDistane ← rN,N;
```

## VI. DETEKSI OBYEK

*Nonlinear elastic matching distance*[5] adalah pengukuran similaritas antara dua kontur. Yang masih kurang pada metode tersebut adalah optimalisasi dalam mencari kontur pada *edge map* yang memiliki similaritas yang tinggi dengan *template* obyek. Beberapa kumpulan *superpixel* yang bertetangga berbentuk sebuah kontur yang tertutup, dan kombinasi himpunan dari semua kontur adalah ruang pencarian dalam mendeteksi obyek. Permasalahannya, *nonlinear elastic matching distance* semua set hanyalah perkiraan, dan tidak cocok dengan beberapa prosedur optimisasi sederhana.

Oleh karena itu, digunakan *Greedy multi-start gradient descent*. Masing-masing *super-pixel* yang mendapat giliran dipilih sebagai *seed region*, metode tersebut berusaha mengurangi nilai *elastic matching* sebanyak mungkin dengan menggabungkan suatu *region* dengan *region* tetangga. Hal ini dilakukan berulang hingga *local minimum* didapatkan, dan tidak ada lagi kemungkinan untuk meningkatkan. *Local minima* untuk semua *seed region* memiliki potensial untuk dideteksi. *Detection threshold* digunakan untuk membuang kontur yang memiliki similaritas yang rendah.

Karena beberapa *starting region* bisa meliputi pada *local minima* yang sama atau mirip, *non-minima suppression* digunakan pada langkah terakhir. Masing-masing kandidat dikunjungi mulai dari kandidat yang memiliki similaritas terbesar ke kandidat yang memiliki similaritas lebih kecil. Semua kandidat yang *overlap* dengan kandidat saat ini yaitu lebih dari *overlap threshold* maka kandidat tersebut dibuang. Berikut algoritma multi-start gradient descent :

**Algoritma 2.** Multi-start gradient descent untuk deteksi obyek yang mirip dengan template

**Input:** object template, list of super-pixels, detection threshold, overlap threshold

**Output** : detected object

```

foreach SuperPixel do
    //greedily maximize similarity
    set object candidate: Candidate SuperPixel;
    set Distance NEMR(Candidate,Template);
    while Distance decrease do
        set Neighbours adjacent Super-pixels of Cnddte;
        foreach Neighbours do
            compute NEMR(Candidate Neigh ,Tmp);
        end
        find Neighbouri which most decreases the distance;
        set Candidate Candidate Neighbouri;
    end
    //only keep candidate if similarity is high
    if NEMR(Candidate,Template)>DetectionThreshold
    then
        delete Candidate;
    end
end
//non-minima suppression in candidate list
Sort Candidates in descending order of Distance;
for i=1 to #Candidates do
    for j=i+1 to #Candidates do
        if Overlap(Candidatei,Candidatej) > OvrTrshld
```

```

then
    Remove Candidatej
end
end
end
//return detected objects
detections Candidates;

```

## VII. UJI COBA

Uji coba dilakukan untuk mengetahui kinerja dari system. Pada uji coba segmentasi citra, ada dua uji coba yang dilakukan. Yang pertama adalah uji coba pengaruh parameter Q terhadap hasil segmentasi. Sedangkan yang kedua adalah uji coba pengaruh average filter terhadap segmentasi yang dihasilkan

Uji coba pengaruh segmentasi parameter Q terhadap hasil uji coba dilakukan dengan mencoba parameter Q dari nilai yang terkecil sampai yang terbesar. Pada Penjelasan SRM sudah dijelaskan mengenai parameter Q. Pemilihan nilai Q pada percobaan ini diharapkan dapat mewakili nilai-nilai Q pada rentang antara 1 sampai 256. Hasil uji coba parameter Q dapat dilihat pada gambar 4 sedangkan pengaruh filtering terhadap hasil segmentasi dapat dilihat pada gambar 5.

Uji Coba pendeteksian dilakukan dengan melakukan segmentasi terlebih dahulu. Segmentasi citra dilakukan dengan parameter  $Q = 32$ . Hasil segmentasi akan diproses lagi dengan mengambil salah satu fiturnya yaitu kontur. Kontur tersebut akan dicocokkan dengan kontur pada obyek *template*. Pencocokan dilakukan menggunakan metode *non linear elastic matching dengan cost R* yang disingkat NEMR. Dari pencocokan kedua kontur tersebut diperoleh suatu nilai yang disebut *shape distance*. Semakin kecil nilai *shape distance* maka semakin besar kemiripan kedua obyek tersebut. Demikian sebaliknya, semakin besar nilai *shape distance* maka semakin tidak mirip. Perhitungan nilai *shape distance* dilakukan dengan menggunakan *dynamic programming*. Agar dapat melakukan pencocokan untuk kombinasi dari setiap *region* yang ada pada citra inputan, maka digunakan algoritma *mutli-start gradient descent* telah dijelaskan pada Deteksi Obyek. Citra hasil deteksi menggunakan NEMR dapat dilihat pada gambar 12. Sedangkan grafik untuk *shape distance* masing-masing obyek dapat dilihat pada gambar 8 sampai 11.

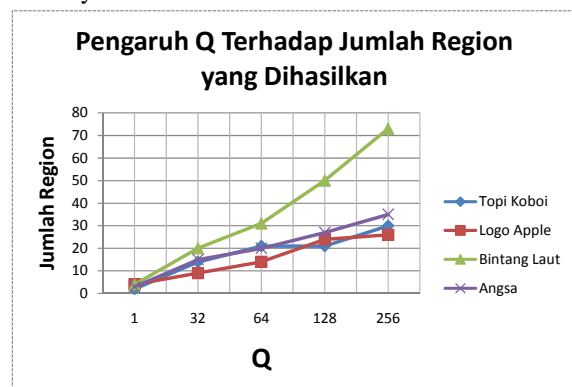
Salah satu kelebihan dari metode *global contour shape* adalah dapat mendeteksi obyek meskipun obyek tersebut dirotasi dengan berbagai sudut. Untuk itu dilakukan percobaan untuk menguji hasil pendeteksian pada obyek jika obyek tersebut dirotasi. Rotasi pada percobaan kali ini dilakukan searah jarum jam dengan kenaikan sudut rotasi  $30^\circ$  untuk setiap percobaan. Percobaan dilakukan sebanyak 12 kali. Ada dua kelas obyek yang diujicoba yaitu lingkaran yang terpotong seperempat bagian dan juga ikan. Grafik hasil uji coba dapat dilihat pada gambar 6.

Selain tahan terhadap beragam rotasi pendeteksian obyek menggunakan metode *global contour shape* juga dapat tahan terhadap beragam skala. Untuk membuktikan hal tersebut, maka dilakukan uji coba pendeteksian dengan obyek yang diperbesar dan diperkecil. Pembesaran dilakukan sebanyak 12 kali dengan skala 5% dari ukuran awal. Demikian juga dengan Pengecilan yang dilakukan sebanyak 12 kali dengan skala 5%. Grafik hasil uji coba pendeteksian terhadap perubahan skala dapat dilihat pada gambar 7.

## VIII. KESIMPULAN

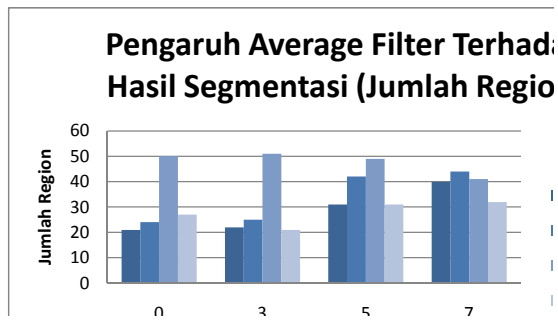
Dari ujicoba dan analisis yang telah dilakukan dapat diambil kesimpulan sebagai berikut :

1. Aplikasi Pendeteksian Obyek menggunakan metode *global contour shape* ini dapat menangani permasalahan pendeteksian obyek yang di dasarkan pada kontur. Dalam hal ini *shape matching* dihitung dengan menggunakan algoritma *dynamic programming*.
2. Algoritma *Statistical Region Merging* dapat melakukan segmentasi dengan baik. Dengan parameter Q yang digunakan untuk mengatur kompleksitas segmentasi.
3. Proses *blurring* dengan *average filter 3x3* dapat mengoptimalkan hasil segmentasi pada beberapa obyek.
4. Metode *global contour shape* terbukti dapat mendeteksi berbagai macam obyek yang memiliki bentuk yang beragam. Dalam hal ini toleransi pendeteksian dapat diatur pada nilai *threshold*-nya.
5. Metode *global contour shape* terbukti dapat mendeteksi obyek walaupun obyek tersebut diputar dari  $0^\circ$  sampai  $360^\circ$ .
6. Metode *global contour shape* terbukti dapat mendeteksi obyek walaupun obyek tersebut diperbesar atau diperkecil dengan skala tertentu. Pada uji coba, perbesaran dilakukan hingga mencapai 160% dan pengecilan mencapai 45%.
7. Nilai *shape distance* mencapai minimum pada uji coba rotasi dengan sudut  $180^\circ$ .
8. Hasil segmentasi yang baik berpengaruh terhadap akurasi sistem dalam mendeteksi obyek.

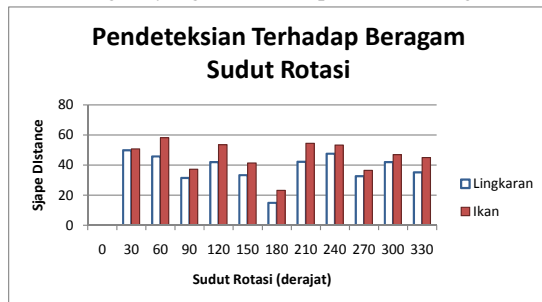


Gambar 4. Grafik Pengaruh Q Terhadap Jumlah Region yang Dihasilkan pada Proses Segmentasi

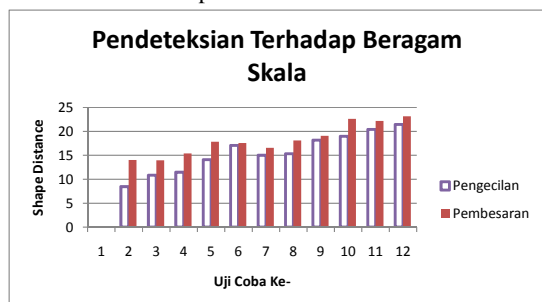




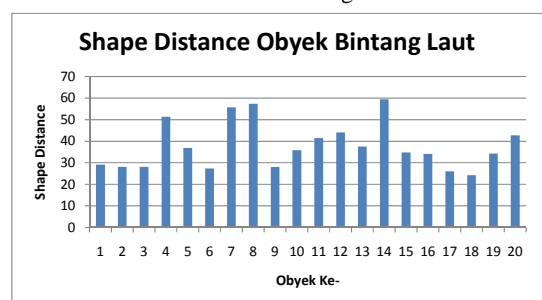
Gambar 5. Grafik Pengaruh Average Filter Terhadap Jumlah Region yang Dihasilkan pada Proses Segmentasi



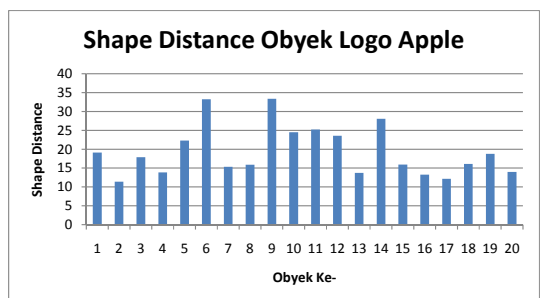
Gambar 6. Grafik Pengaruh Rotasi Terhadap Shape Distance pada Proses Pendeteksian



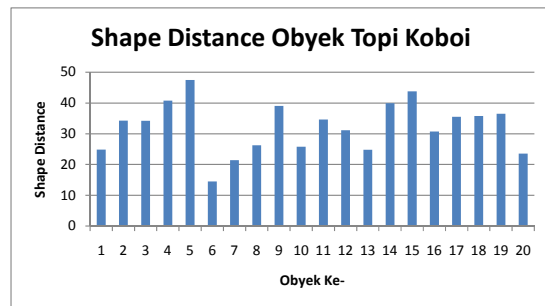
Gambar 7. Grafik Perbandingan Shape Distance pada Pembesaran dan Pengecilan



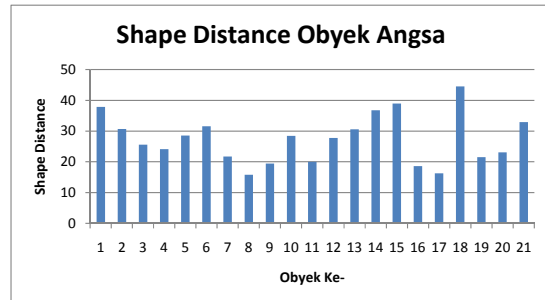
Gambar 8. Grafik Shape Distance Pada Kelas Bintang Laut



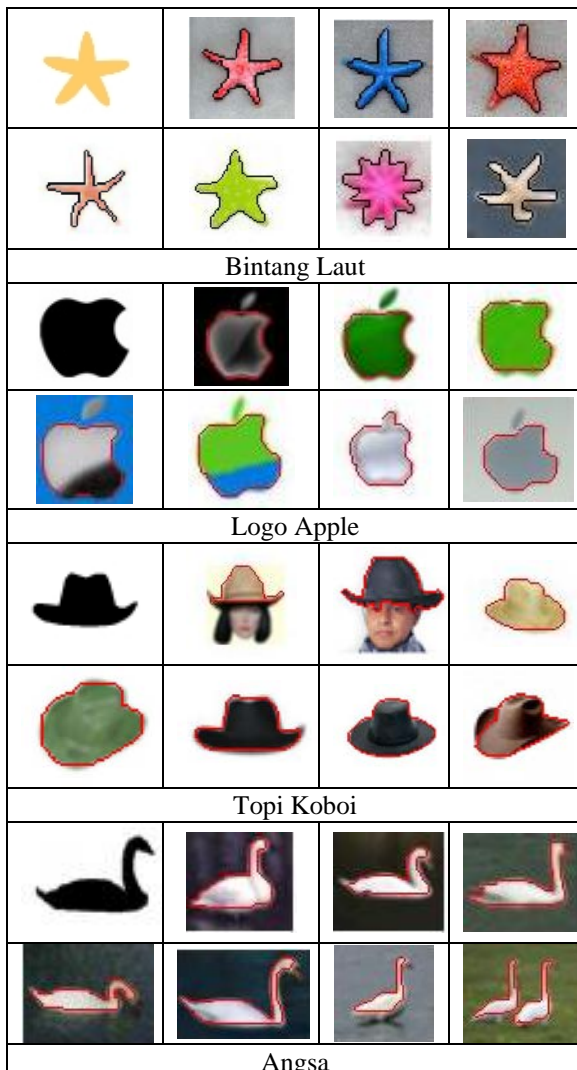
Gambar 9. Grafik Shape Distance Pada Kelas Logo Apple



Gambar 10. Grafik Shape Distance Pada Kelas Topi Kobi



Gambar 11. Grafik Shape Distance Pada Kelas Angsa



Gambar 12. Hasil Deteksi Menggunakan Elastic Matching (Obyek Pertama pada setiap kelas adalah obyek *template*)

## IX. DAFTAR PUSTAKA

- [1] Konrad Schindler, David Suter.(2008).**Object Detection by Global Contour Shape**. ScienceDirect, Pattern Recognition 41,3736-3748.
- [2] S. Battiato, G. M. Farinella, and G. Puglisi.(2006). **SVG Vectorization by Statistical Region Merging**,Eurographics Italian Chapter Conference. Dipartimento di Matematica e Informatica, University of Catania, Italy.
- [3] Richard Nock, Frank Nielsen.(2004).**Statistical Region Merging**. IEEE Transactions On Pattern Analysis And Machine Intelligence. Vol. 26, No. 11.
- [4] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein.(2001). **Introduction to Algorithms Second Edition**. McGraw – Hill Book Company. The MIT Press, Cambridge, Massachusett, London, England.
- [5] R.C. Veltkamp. (2001). **Shape Matching: Similarity Measures And Algorithms**, Technical Report UU-CS-2001-03, Institute of Information and Computing Sciences, Utrecht University.