

# Temu Kembali Citra Busana Muslimah Berdasarkan Bentuk Menggunakan *Curvature Scale Space* (CSS)

Hayatun Maghfirah<sup>1</sup>, Fitri Arnia<sup>2</sup>, Khairul Munadi<sup>3</sup>

**Abstract**— Nowadays in Indonesia, Islamic woman's clothing has been popular and follows the latest trend. Clothing with various color, texture, and shape are available. Furthermore, the online clothing trading system is becoming more attractive, which facilitates the users the apparel images through the websites. These images can be retrieved by querying a text to the retrieval system. However, the users face difficulties in describing the clothes precisely. Thus, a retrieval method based on content, which is known as content-based image retrieval (CBIR), is developed. Here, the content is represented by color, texture, and shape. This paper aims to present and discuss an application of *Curvature Scale Space* (CSS) as a shape feature for Islamic woman's clothing retrieval system. The performance of retrieval results of three clothing categories is analyzed: blouse-pant, long dress, and tunic, and used different feature length. The simulations run with as many as 300 images from the three categories, 100 images from each. Performance is measured in recall and precision. The results are compared by applying another shape feature; that is the histogram of gradient (HOG). The blouse-pant group achieves the highest performance, followed by tunic and long dress categories. The different feature length affects the retrieval performance; the longer the features, the lower *recall* and *precision* values. The feature of length 4 achieves the highest performance. The CSS is applied as the feature in CBIR of Islamic clothing, results in higher performance than the HOG.

**Intisari**—Di Indonesia, model busana muslim wanita berkembang pesat dan semakin populer, sehingga muncul busana dengan warna, tekstur dan bentuk (*shape*) yang variatif. Seiring dengan itu, sistem penjualan busana muslimah melalui media daring semakin menarik karena sistem penjualan tersebut memfasilitasi pengguna dengan citra-citra busana pada situs penjualannya. Citra busana ini dapat dicari menggunakan kata kunci berupa teks. Tetapi, pengguna seringkali dihadapkan pada kesulitan untuk mendeskripsikan busana yang diinginkan secara tepat. Karena itu muncul suatu teknik yang dikenal dengan nama temu kembali citra berdasarkan isi (TKCI). Isi yang dimaksud adalah warna, tekstur, dan bentuk. Makalah ini bertujuan membahas penerapan *Curvature Scale Space* (CSS) sebagai metode ekstraksi ciri bentuk dari busana muslim wanita pada suatu sistem TKCI. Unjuk kerja temu kembali dievaluasi pada tiga kategori busana berupa blus celana, gamis, dan tunik, dan dengan panjang ciri CSS yang berbeda. Sebanyak 300 citra busana muslimah yang terdiri atas 100 citra untuk masing-masing kategori digunakan pada simulasi. Unjuk kerja diukur

dengan parameter *recall* dan *precision*, dan metode ekstraksi ciri pembandingan yang digunakan adalah *histogram of gradient* (HOG). Hasil temu kembali terbaik diperoleh pada kategori blus celana, diikuti oleh tunik, dan gamis. Pengambilan panjang ciri CSS yang berbeda berpengaruh pada hasil TKCI, yaitu semakin panjang ciri, semakin turun unjuk kerja temu kembali, dengan panjang ciri 4 menghasilkan unjuk kerja terbaik. Penggunaan CSS sebagai ciri menghasilkan unjuk kerja temu kembali yang lebih baik dibandingkan dengan penggunaan HOG.

**Kata Kunci**— TKCI busana, bentuk, CSS, HOG, *recall* & *precision*.

## I. PENDAHULUAN

Beberapa dekade yang lalu, busana muslim wanita sudah ada tetapi belum populer. Sekarang, busana muslimah dengan cepat mengikuti perkembangan dunia mode, sehingga muncul model busana muslimah dengan warna, tekstur, dan bentuk (*shape*) yang variatif. Sama halnya dengan penjualan busana jenis umum, penjualan busana muslimah banyak yang dilakukan secara daring. Sampai saat ini, untuk mencari busana pada suatu sistem penjualan daring, digunakan kata kunci, berupa teks sebagai *query*.

Penelitian menunjukkan bahwa *querying* menggunakan teks sering menghasilkan citra-citra yang tidak sesuai dengan keinginan pengguna. Setidaknya ada dua penyebab, yaitu manusia mengalami kesulitan mendeskripsikan sebuah objek menggunakan kata-kata, baik itu dari segi warna, tekstur, maupun bentuk, atau sistem tidak dapat memfasilitasi variasi kata kunci (teks) yang diberikan manusia sebagai *query* ke sistem. Kondisi ini dapat diselesaikan jika *query* ke sistem adalah citra itu sendiri. Proses *querying* dengan *query* berupa citra dikenal dengan nama temu kembali citra berdasarkan isi (TKCI). Isi yang dimaksud adalah warna, tekstur, dan bentuk objek. Pada TKCI, isi objek citra diekstrak dan disimpan dalam wadah yang disebut ciri (*feature*).

Penelitian menunjukkan bahwa bentuk sebagai ciri visual sulit atau hampir tidak mungkin dideskripsikan dengan teks [1]. Selanjutnya, dikemukakan bahwa penerapan TKCI berdasarkan bentuk lebih sulit jika dibandingkan dengan TKCI berdasarkan warna atau tekstur [2]. Banyak penelitian telah dilakukan untuk mengekstrak ciri bentuk dari suatu objek citra [3] - [5]. *Curvature Scale Space* (CSS) adalah ciri bentuk standar yang didefinisikan pada *Motion Picture Experts Group-7* (MPEG-7), selain beberapa ciri lain seperti *chain-code* dan deskriptor *wavelet*. CSS telah digunakan untuk mencari kemiripan bentuk binatang laut [3]. Penelitian ini menunjukkan bahwa metode CSS lebih unggul daripada deskriptor Fourier dan *moment invariants*. Selanjutnya, telah digunakan juga metode CSS pada TKCI merek dagang [4]. Penelitian tersebut menunjukkan bahwa citra yang

<sup>1</sup> Mahasiswa, Magister Teknik Elektro Program Pascasarjana, Universitas Syiah Kuala, Jln. Tgk. Syech Abdurrauf No. 7, Darussalam, Banda Aceh, INDONESIA (e-mail: hayatunmaghfirah@gmail.com)

<sup>2,3</sup> Dosen, <sup>2</sup>Corresponding Author, Jurusan Teknik Elektro dan Komputer, Fakultas Teknik, Universitas Syiah Kuala, Jln. Tgk. Syech Abdurrauf No. 7, Darussalam, Banda Aceh, INDONESIA (e-mail: f.arnia@unsyiah.ac.id, khairul.munadi@unsyiah.ac.id)

direpresentasikan dengan CSS tidak terpengaruh oleh adanya derau citra, perubahan skala, dan orientasi citra.

Penerapan TKCI pada busana merupakan ranah riset baru yang aktif diteliti beberapa tahun ini. Sebuah penelitian membahas TKCI dengan membandingkan penggunaan lima deskriptor, yaitu deskriptor warna dan arah tepi, MPEG-7, *correlogram* warna, histogram warna, dan tekstur Tamura [6]. Semua deskriptor ini digunakan dalam eksperimen dengan serangkaian citra busana dengan berbagai merek dan gaya yang ada dalam lima tahun. Hasil eksperimen menunjukkan bahwa deskriptor tekstur Tamura melebihi deskriptor-deskriptor lain dalam hal *precision* dan *recall*. Penelitian lain mengusulkan metode *sparse-coding* untuk pencarian citra busana berbasis isi, dengan ciri berupa *style*, warna, dan tekstur [7].

Sebuah penelitian membahas cara menerapkan teknik “*shape contex*” untuk hasil yang lebih baik dalam pencarian citra busana [8]. Metode ini mencapai tingkat temu kembali rata-rata sebesar 32,73%. Pada penelitian lain, diusulkan metode temu kembali berdasarkan “*shape contex*” [9]. Artikel ini mengungkapkan bahwa bentuk adalah hal yang penting dalam TKCI busana. Selanjutnya disebutkan bahwa sistem belanja secara daring sudah selayaknya mempertimbangkan barang yang diperjualbelikan berupa busana.

Ciri bentuk lain yang banyak dipakai sebagai *descriptor* bentuk pada TKCI busana adalah *histogram of gradient* (HOG) [5]. Pada penelitian tersebut, diukur unjuk kerja penggunaan HOG dan beberapa ciri lain untuk mendeteksi bagian-bagian dari busana. Dibandingkan dengan ciri yang lain, ditemukan bahwa penggunaan HOG menghasilkan unjuk kerja terbaik dengan nilai presisi tertinggi sebesar 0.6.

Studi menunjukkan bahwa penerapan CSS pada TKCI citra-citra umum lebih baik dibandingkan dengan ciri bentuk lain yang didefinisikan pada MPEG-7 [3]. Selanjutnya, ciri bentuk lain, yaitu HOG, yang banyak telah dicoba untuk diterapkan pada TKCI busana masih menghasilkan nilai unjuk kerja yang rendah [5], [8]. Makalah ini memiliki kontribusi sebagai berikut.

1. Membahas penerapan CSS sebagai metode ekstraksi ciri bentuk untuk busana muslim wanita. Penerapan suatu ciri yang sama pada domain yang berbeda akan menghasilkan unjuk kerja yang berbeda-beda. Studi penggunaan CSS pada TKCI busana muslimah perlu dilakukan, mengingat keandalannya pada TKCI citra umum dan perkembangan pasar busana muslimah yang semakin besar dan cenderung akan terus membesar dalam tahun-tahun mendatang [10].
2. Dalam eksperimen, ditampilkan unjuk kerja penggunaan panjang ciri CSS yang berbeda-beda. Hal seperti ini tidak pernah ada pada penelitian-penelitian sebelumnya [3], [11].
3. Menunjukkan bahwa penggunaan CSS menghasilkan unjuk kerja yang lebih baik daripada HOG, berdasarkan parameter *recall* dan *precision*.

## II. CURVATURE SCALE SPACE (CSS)

CSS merupakan metode ekstraksi ciri bentuk kontur tertutup. Metode berbasis kontur ini awalnya dikembangkan oleh Mokhtarian dan selanjutnya disempurnakan untuk

digunakan dalam standar MPEG-7. Manusia memetakan kontur dalam bentuk cekung dan cembung [4]. CSS juga menggunakan teknik yang sama, menyegmentasikan kontur menjadi bagian cekung dan cembung dengan menghitung titik pada kontur. Kontur kemudian dihaluskan menggunakan filter Gaussian pada masing-masing titik perubahan dari cekung ke cembung dan sebaliknya (titik infleksi). Proses filter dilakukan pada semua titik infleksi di sepanjang kontur. Jumlah semua titik yang mengalami proses filter dicatat dan filter berhenti ketika seluruh kontur menjadi cembung (tidak ada lagi titik infleksi). Sebelum menerapkan CSS, diperlukan tahap pengolahan awal dan ekstraksi kontur.

### A. Pengolahan Awal

Pegolahan awal yang dilakukan adalah normalisasi ukuran citra, mengubah format RGB ke *grey-scale*, dilanjutkan dengan proses *binarization* untuk menghasilkan citra biner.

### B. Ekstraksi Kontur

Kontur bentuk dihitung dari citra biner. Algoritme pelacakan kontur yang paling umum digunakan adalah *square tracing*. Algoritme ini mempunyai kelebihan dalam penerapan, prosedurnya sederhana, dan karena itu sering digunakan untuk melacak kontur suatu pola. Ide algoritme *square Tracing* dijelaskan sebagai berikut [12].

#### Algoritme *square tracing*:

Masukan adalah susunan segi empat (*square tessellation*)

**T**, yang terdiri atas **P** komponen sel hitam yang terhubung  
Keluaran **B** (**b**<sub>1</sub>, **b**<sub>2</sub>, ...**b**<sub>k</sub>) adalah piksel *boundary* (kontur)

Mulai

1. Atur **B** kosong
2. Dari bawah ke atas dan kiri ke kanan *scan* sel-sel **T** sampai ditemukan piksel hitam, **s**, dari **P**.
3. Masukkan **s** ke dalam **B**.
4. Atur piksel saat ini, **p**, menjadi piksel awal, **s**.
5. Belok kiri, kunjungi piksel di sebelah kiri **p**.
6. Update **p**, dengan mengaturnya menjadi piksel saat ini.
7. While **p** tidak sama dengan **s** do

If piksel saat ini **p** adalah piksel hitam

- Masukkan **p** ke **B** dan belok kiri (kunjungi piksel di sebelah kiri **p**).
- Update **p**, yaitu mengaturnya menjadi piksel saat ini.

Else

- Belok kanan (kunjungi piksel di sebelah kanan **p**)
- Update **p**, yaitu mengaturnya menjadi piksel saat ini.

end

End

### C. Ekstraksi Ciri CSS

Setelah melakukan ekstraksi kontur, langkah selanjutnya adalah menentukan koordinat kontur dan normalisasi panjang kontur. Diasumsikan **X** adalah suatu kontur tertutup yang memenuhi syarat

$$X = \{(x_1, y_1), (x_2, y_2) \dots (x_n, y_n)\}$$

dengan  $(x_n, y_n)$  adalah koordinat titik yang terdapat pada kontur. Titik-titik ini kemudian digunakan untuk membangun fungsi  $x(u)$  dan  $y(u)$ , dengan  $u$  adalah panjang busur (*arc-length*) kontur yang dinormalisasi. Normalisasi panjang busur dilakukan dengan mencuplik ulang kontur untuk mendapatkan jumlah titik yang sama pada semua kontur.

Proses selanjutnya adalah penghitungan peta kontur CSS. Peta kontur CSS adalah organisasi multi-skala dari titik-titik infleksi (titik-titik *zero-crossing* dari kurva), yang kelengkungannya dihitung dengan (1). Selanjutnya, kontur dikonvolusikan dengan fungsi Gaussian sesuai dengan (2), dengan nilai  $\sigma$  yang bervariasi. Ketika nilai  $\sigma$  naik, maka bentuk objek menjadi lebih halus dan lengkungannya akan berkurang. Seiring dengan itu, jumlah titik *zero-crossing* akan semakin sedikit. Fungsi Gaussian didefinisikan pada (3).

$$k(u) = \frac{\dot{x}(u) \cdot \ddot{y}(u) - \ddot{x}(u) \cdot \dot{y}(u)}{(\dot{x}^2(u) + \dot{y}^2(u))^{3/2}} \quad (1)$$

$$x'(u) = x(u) * g(u, \sigma), \quad y'(u) = y(u) * g(u, \sigma) \quad (2)$$

$$g(u, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{u^2}{2\sigma^2}} \quad (3)$$

dengan  $u$  = parameter fungsi  $x(u)$  dan  $y(u)$ , dan  $c$  = konstanta yang bernilai 1.

Algoritme untuk menghitung peta kontur CSS adalah sebagai berikut [13] – [15].

#### Algoritme peta kontur CSS:

1. Normalisasi bentuk sehingga titik pada *boundary* berjumlah tetap;
2. Buat sebuah array  $ZC [ ] [ ]$  untuk mencatat titik *zero crossing* kelengkungan pada setiap skala;
3. Hitung kelengkungan dari setiap posisi  $u$  pada skala  $\sigma$  saat ini menurut (1);
4. Catat setiap titik *zero crossing* kelengkungan pada skala  $\sigma$  saat ini, ke variabel  $ZC [\sigma] [u]$ ;
5. Haluskan kontur berdasarkan (2);
6. Ulangi langkah 3-5 sampai tidak ditemukan lagi titik *zero crossing*;
7. Plot semua titik *zero crossing* kelengkungan yang tercatat pada  $ZC [ ] [ ]$  ke bidang Cartesian untuk membuat peta kontur CSS.

Langkah terakhir dari ekstraksi ciri CSS adalah ekstraksi puncak CSS. Representasi dari peta kontur CSS ditentukan berdasarkan lokasi nilai-nilai tertinggi pada peta kontur CSS. Nilai-nilai ini akan digunakan sebagai ciri vektor dari citra. Berikut adalah algoritme ekstraksi puncak kontur CSS [13].

#### Algoritme ekstraksi puncak kontur CSS:

1. *Scan* peta kontur CSS dari baris atas;
2. Jika titik *zero-crossing* ditemukan di lokasi  $(i, j)$ , periksa titik tetangga di atasnya, yaitu pada lokasi  $(i-1, j-1)$ ,  $(i-1, j)$ , dan  $(i-1, j+1)$ . Jika tiga titik tetangga di atas bukan titik *zero-crossing*, maka lokasi  $(i, j)$  merupakan kandidat puncak; cari semua kandidat puncak pada baris  $i$ .

3. Untuk masing-masing kandidat puncak  $(i, j)$  pada baris  $i$ , periksa kandidat puncak tetangga, jika kandidat puncak tetangga  $(i, k)$  ditemukan dengan jarak lebih dari 5 titik, maka  $(i, j)$  adalah puncak. Jika kandidat tetangga ditemukan dengan jarak  $\leq 5$  titik, maka ada puncak di tengahnya, yaitu pada  $(i, (j+k)/2)$
4. Ulangi langkah 2 dan 3 untuk setiap baris, sampai semua puncak CSS ditemukan.

#### D. Ukuran Kemiripan Citra

Tingkat kemiripan antara dua citra diukur dengan menghitung jarak Euclidian antara vektor ciri CSS dari kedua citra tersebut. Rumus dari jarak Euclidian didefinisikan di (4).

$$d_{ij} = \sqrt{\sum_{k=1}^n (x_{ik} - x_{jk})^2} \quad (4)$$

dengan  $d_{ij}$  merupakan jarak vektor  $i$  dan  $j$ ,  $x_{ik}$  adalah elemen ke  $k$  dari vektor  $x_i$ , dan  $x_{jk}$  merupakan elemen ke  $k$  dari vektor  $x_j$ .

#### E. Pengukuran Unjuk Kerja Temu Kembali

Unjuk kerja temu kembali diukur menggunakan parameter *precision* dan *recall*, yang dijelaskan sebagai berikut.

*Precision* adalah perbandingan jumlah citra yang relevan terhadap *query* dengan jumlah citra yang terambil dari hasil pencarian, rumus dari *precision* diberikan di (5) [7].

$$Precision = \frac{RI}{TR} \quad (5)$$

dengan  $RI$  adalah jumlah citra hasil temu kembali yang relevan dan  $TR$  merupakan total jumlah temu kembali yang relevan.

*Recall* adalah perbandingan jumlah citra relevan yang terambil sesuai dengan *query* yang diberikan dengan total kumpulan citra dalam basis data yang relevan dengan *query*. Rumus dari *recall* diberikan di (6) [7].

$$Recall = \frac{RI}{TD} \quad (6)$$

dengan  $RI$  adalah variabel jumlah citra hasil temu kembali yang relevan dan  $TD$  mewakili total jumlah citra relevan dalam basis data.

### III. METODOLOGI

Makalah ini membahas tentang simulasi temu kembali citra busana menggunakan ciri CSS. Bahan simulasi adalah 300 citra busana muslimah, terdiri atas tiga kategori, yaitu blus celana, gamis, dan tunik, yang dikumpulkan dari berbagai situs daring penjualan busana yang populer. Ukuran citra busana dinormalisasi menjadi 160x249 piksel. Ukuran ini dianggap tidak terlalu besar, sehingga proses ekstraksi ciri dapat dilakukan dengan cepat, tetapi juga tidak terlalu kecil, sehingga ciri CSS dapat diekstrak dengan baik. Gbr. 1 menampilkan beberapa contoh citra busana dari setiap kategori.



Gbr. 1 Contoh busana muslimah dari kategori blus celana (baris atas), tunik (baris tengah), dan gamis (baris bawah).

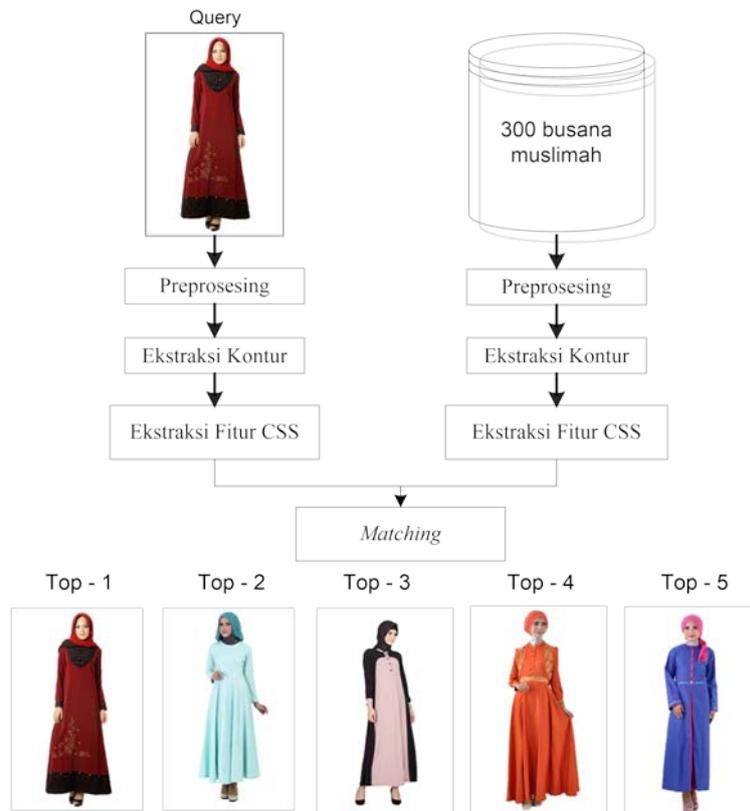
Gbr. 2 menunjukkan proses sistem temu kembali citra busana muslimah menggunakan ekstraksi ciri CSS, yang terdiri atas sisi *querying* dan sisi basis data citra. Dari sisi *query*,

pertama citra busana dinormalisasi ukurannya menjadi 160x249 piksel, kemudian format citra diubah menjadi *grey-scale* dan kontur bentuk busana diekstrak. Selanjutnya, sistem akan menghitung peta kontur CSS dan mengekstrak puncak-puncak pada peta kontur tersebut. Posisi puncak-puncak tersebut merupakan vektor ciri dari citra *query*. Pada sisi basis data citra, hal yang sama dilakukan. Selanjutnya, kedua vektor ciri tersebut dihitung jaraknya menggunakan (4). Untuk melihat pengaruh penggunaan jumlah puncak terhadap unjuk kerja temu kembali, empat jumlah puncak yang berbeda diekstrak dari peta kontur CSS.

Dalam simulasi, dari setiap kategori busana dipilih empat citra *query* secara acak, sehingga total citra *query* adalah 12. Setiap citra *query* dihitung jaraknya dengan 300 citra pada basis data, sehingga total proses *querying* adalah 3600. Untuk setiap hasil *query* pada masing-masing kategori, dihitung nilai *precision* dan *recall*-nya.

IV. HASIL DAN PEMBAHASAN

Pada bagian ini ditunjukkan contoh proses perhitungan ciri CSS pada citra busana muslimah, mulai dari tahap pengolahan awal, ekstraksi kontur, dan ekstraksi fitur CSS. Selanjutnya, dibahas hasil temu kembali citra busana dari setiap kategori, dilanjutkan dengan pembahasan pengaruh panjang vektor ciri CSS terhadap unjuk kerja temu kembali. Unjuk kerja TKCI berbasis CSS dibandingkan dengan unjuk kerja TKCI yang menggunakan HOG sebagai metode ekstraksi cirinya.



Gbr. 2 Diagram alir TKCK busana muslim wanita.

A. Proses Perhitungan Ciri CSS

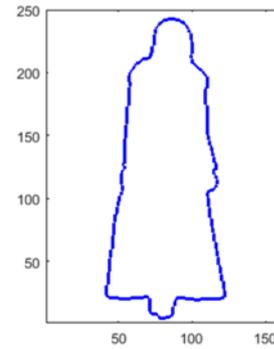
1) *Pengolahan Awal*: Pada Gbr. 3 ditunjukkan proses pengolahan awal citra gamis, mulai dari perubahan format RGB ke *grey-scale*, sampai dengan proses *binarization*. Nilai ambang yang digunakan pada proses *binarization* adalah 230. Jika suatu piksel bernilai lebih besar daripada 230, maka piksel tersebut diubah menjadi hitam, dan sebaliknya. Nilai 230 adalah nilai ambang terbaik yang diperoleh melalui proses *trial and error*.

2) *Ekstraksi kontur*: Kontur busana muslimah diperoleh menggunakan algoritme *square tracing*. Hasil penerapan algoritme tersebut pada hasil *binarization* busana gamis ditunjukkan pada Gbr. 4.

3) *Ekstraksi Ciri CSS*: Untuk kebutuhan ekstraksi ciri, *boundary* kontur dinormalisasi sehingga titik pada *boundary* berjumlah 200. Kemudian diterapkan algoritme peta kontur CSS. Gbr. 5 menunjukkan hasil tahap penghalusan dari algoritme peta kontur CSS, langkah 3 sampai dengan 5. Gbr. 6 menunjukkan peta kontur CSS pada busana gamis. Selanjutnya, dari peta kontur, diekstraksi empat jenis panjang vektor ciri yang ditunjukkan pada Tabel I. Sebagai contoh, pada kolom ‘maksimum 4’, diambil dua nilai puncak maksimum dengan koordinat puncak CSS (35, 116) dan (112, 138). Nilai-nilai ini disusun dalam vektor 1-dimensi dengan urutan 116, 112, 35, 138.

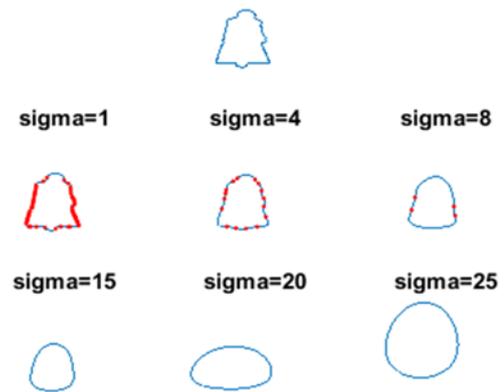


Gbr. 3 Tahap pengolahan awal, (a) RGB, (b) *Greyscale*, (c) Biner.



Gbr. 4 Kontur gamis dari Gbr. 3(c).

square tracking



Gbr. 5 Hasil tahap penghalusan kontur gamis dari Gbr. 4.

TABEL I  
EMPAT MACAM PANJANG FITUR CSS

No	Maksimum 4		Maksimum 10		Maksimum 20		Maksimum 30	
	$\sigma$	$u$	$\sigma$	$u$	$\sigma$	$u$	$\sigma$	$u$
1	116	35	116	35	116	35	116	35
2	112	138	112	138	112	138	112	138
3			66	116	66	116	66	116
4			50	175	50	175	50	175
5			48	192	48	192	48	192
6					41	72	41	72
7					39	52	39	52
8					39	96	39	96
9					28	17	28	17
10					22	36	22	36
11							20	111
12							18	156
13							15	9
14							14	54
15							9	47

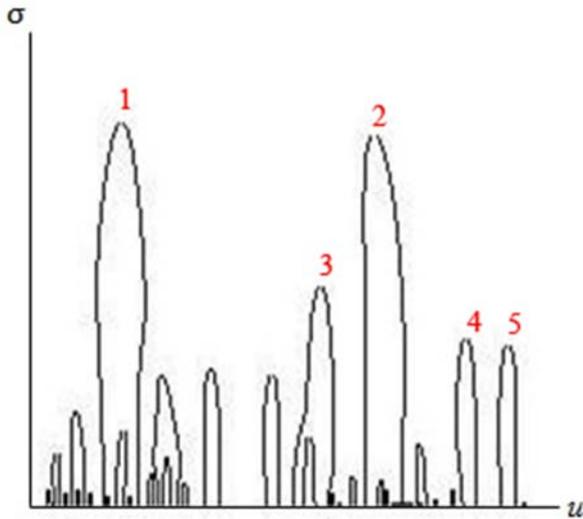
B. Hasil TKCI untuk Setiap Kategori Busana

Gbr. 7 sampai dengan Gbr. 9 menunjukkan contoh hasil temu kembali ketiga kategori busana. Pada setiap gambar, ditunjukkan 20 urutan busana yang jarak vektor cirinya paling dekat dengan citra *query*. Kotak merah digunakan untuk menandai kejadian *false positive*.

Gbr. 7 adalah hasil temu kembali citra blus celana menggunakan panjang vektor ciri 30 (maksimum 30 pada Tabel I). Terdapat tiga citra busana yang merupakan *false positive*, yaitu citra-citra dengan nomor urut 12, 17, dan 19.

Ketiganya merupakan citra busana gamis. Gbr. 8 menunjukkan hasil temu kembali busana kategori tunik dengan panjang ciri 10. Tiga citra busana gamis yang berada pada posisi 10, 16, dan 19 merupakan citra *false positive*. Selanjutnya, Gbr. 9 menunjukkan hasil temu kembali busana kategori gamis dengan panjang ciri 4. Pada urutan ke-12 terdapat *false positive* berupa citra tunik.

Pada Gbr. 10 ditunjukkan unjuk kerja penggunaan CSS berupa nilai *recall* dan *precision*. Di sini, HOG sebagai metode ekstraksi ciri digunakan sebagai pembanding. Pada setiap grafik terdapat tiga *point*. *Point* paling kiri sampai *point* paling kanan masing-masing adalah titik perpotongan nilai *recall* dan *precision* untuk hasil temu kembali sebanyak 10, 20, dan 30 citra. Untuk semua kategori, CSS menghasilkan unjuk kerja yang lebih baik dibandingkan dengan HOG.



Gbr. 6 Peta kontur CSS gamis dari Gbr. 3.

Pada CSS sendiri, kategori blus celana menghasilkan unjuk kerja temu kembali terbaik, dengan nilai *recall* dan *precision* mencapai 100%, diikuti kategori tunik, dengan nilai *recall* dan *precision* di atas 75%, dan diakhiri dengan kategori gamis dengan nilai *recall* dan *precision* di atas 70%.

C. Penggunaan Vektor Ciri CSS dengan Panjang yang Berbeda

Pada Gbr. 11 ditunjukkan grafik pengaruh penggunaan panjang ciri CSS yang berbeda, yaitu 4, 10, 20, dan 30, untuk semua kategori busana. Penggunaan panjang ciri 4 menghasilkan kinerja paling baik, seperti ditunjukkan pada Gbr. 11(a). Secara umum, semakin panjang ciri yang digunakan, hasil temu kembalinya semakin buruk.

Untuk kategori blus celana, hasil temu kembali dari 10 urutan pertama (titik paling kiri pada setiap grafik), menghasilkan nilai *precision* yang berkisar antara 90% - 100%. Untuk kategori tunik, diperoleh nilai *precision* antara 70% - 94%, sedangkan untuk kategori gamis, diperoleh nilai *precision* antara 68% - 82%.



Gbr. 7 Contoh hasil temu kembali busana kategori blus celana menggunakan ciri CSS dengan panjang 30.



Gbr. 8 Contoh hasil temu kembali busana kategori tunik menggunakan fitur CSS dengan panjang 10.

#### D. Pembahasan

Hasil yang ditampilkan pada Gbr. 10 dan Gbr. 11 menunjukkan bahwa busana kategori blus celana selalu menempati posisi terbaik, dalam arti jumlah citra sekilas yang ditemukan pada proses temu kembalinya adalah yang paling tinggi. Jika dilihat dari bentuk objek, blus celana memiliki bentuk yang berbeda jika dibandingkan dengan tunik dan gamis. Blus celana memiliki lengkungan cekung yang dalam, khususnya pada bagian celana, yang menyebabkan puncak CSS yang mewakili daerah lengkungan jauh lebih tinggi dibandingkan puncak-puncak yang lain. Semakin dalam lengkungan cekung sebuah objek, maka semakin tinggi nilai puncak CSS-nya. Lain halnya dengan gamis yang memiliki lengkungan yang kurang dalam, yaitu semua titik infleksi pada konturnya kurang lebih memiliki nilai perubahan yang sama, sehingga nilai puncak tertinggi CSS-nya tidak jauh berbeda dibandingkan dengan puncak lain.

Gbr. 11 menunjukkan bahwa penggunaan panjang ciri yang berbeda memiliki pengaruh terhadap unjuk kerja temu

kembali. Semakin panjang ciri yang digunakan, semakin rendah unjuk kerja penggunaan CSS pada semua kategori busana. Hal ini disebabkan karena ciri yang panjang, banyak puncak-puncak yang diambil dari peta kontur CSS, termasuk puncak-puncak yang rendah, seperti tampak pada Gbr. 6, yang dapat bertindak sebagai derau.

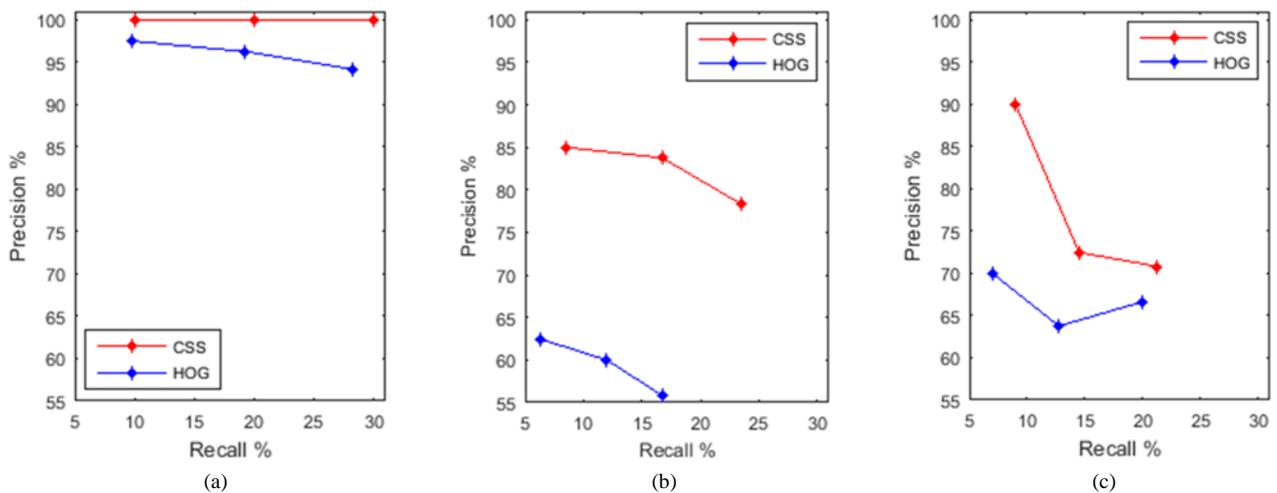
Hal ini dapat dijelaskan dengan bantuan Gbr. 5 dan Gbr. 6. Pada Gbr. 5, titik-titik merah yang terlihat pada kontur gamis adalah titik-titik infleksi. Ketika nilai  $\sigma = 1$ , titik infleksi masih sangat banyak. Jumlah titik infleksi berkurang seiring dengan bertambahnya besar  $\sigma$ . Pada Gbr. 6, dua titik infleksi terbesar dari kontur gamis diwakili oleh dua puncak yang paling tinggi, yaitu puncak tertinggi pertama dan kedua. Penggunaan banyak titik puncak pada Gbr. 6 berarti penggunaan ciri CSS yang lebih panjang pada Tabel I. Jika dihubungkan dengan Gbr. 5, puncak-puncak ini adalah titik-titik merah pada kontur. Tampak untuk nilai  $\sigma = 1$ , jumlah titik merah terlalu banyak, sehingga sebagian besar dari titik titik tersebut tidak memberikan informasi berarti tentang titik infleksi (bentuk kontur), dan karenanya berperan

sebagai derau. Derau menyamakan karakteristik ciri yang seharusnya hanya perlu diwakili oleh beberapa nilai puncak

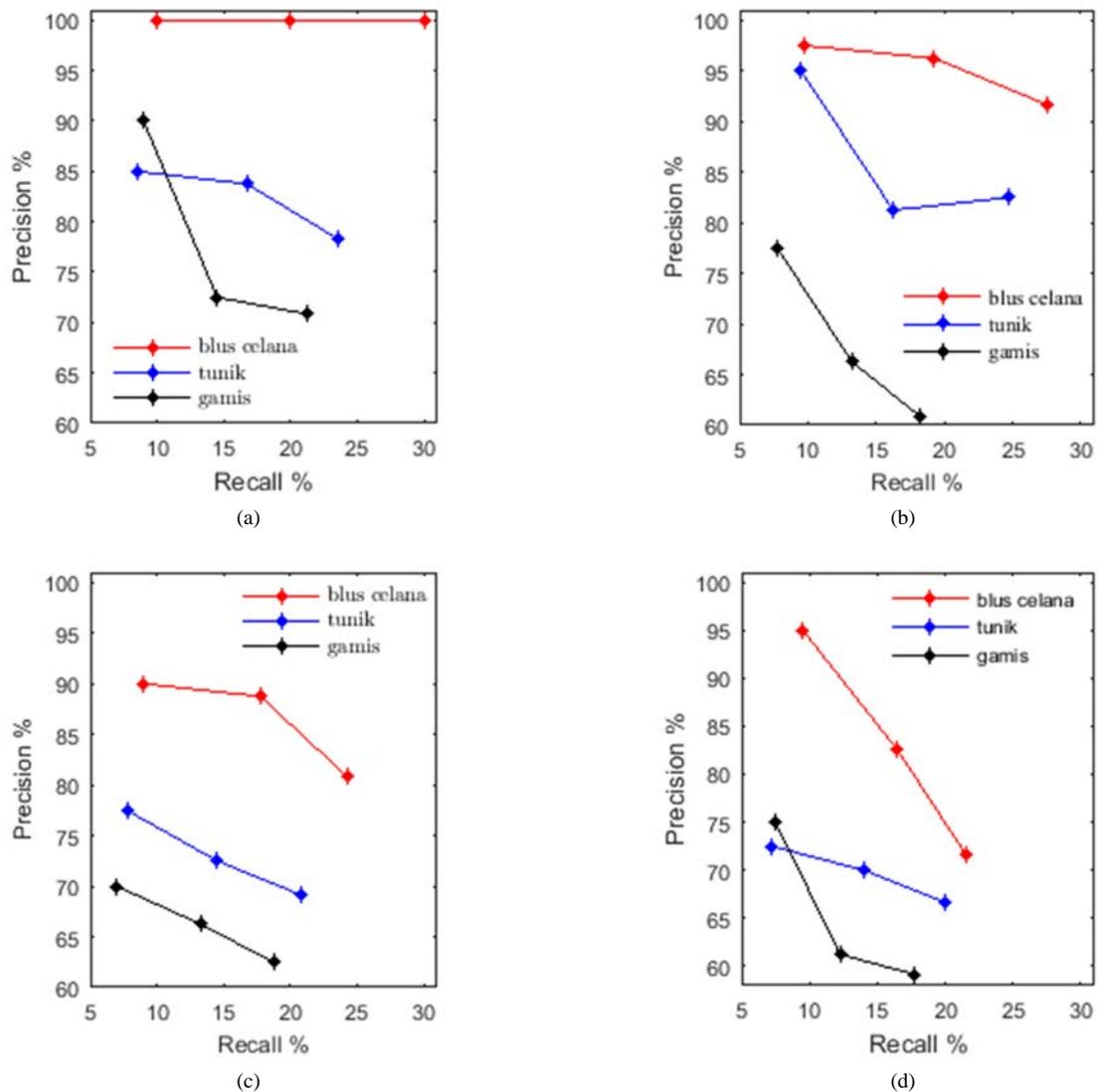
tertinggi. Hal serupa pernah disebutkan penelitian lain, walaupun tidak disertai dengan data-data eksplisit [11].



Gbr. 9 Contoh hasil temu kembali busana kategori gamis menggunakan ciri CSS dengan panjang 4.



Gbr. 10 Nilai recall dan precision dari CSS dan HOG untuk tiga kategori busana, (a) kategori blus celana, (b) kategori blus celana, (c) kategori gamis.



Gbr. 11 Nilai *recall* dan *precision* untuk empat macam jumlah panjang ciri CSS, (a) ciri CSS = 4, (b) ciri CSS = 10, (c) ciri CSS = 20, (d) ciri CSS = 30.

## V. KESIMPULAN

Metode CSS berhasil diterapkan untuk ekstraksi ciri bentuk pada busana muslim wanita, dengan unjuk kerja yang lebih baik dibandingkan ciri HOG. Sebanyak 300 citra busana muslimah dalam basis data, terdiri atas 100 citra blus celana, 100 citra gamis, dan 100 citra tunik, digunakan dalam simulasi. Hasil temu kembali terbaik diperoleh untuk kategori busana blus celana, diikuti oleh tunik, dan kemudian gamis. Hal ini dicapai karena busana kategori blus celana memiliki ciri CSS yang lebih unik dibandingkan dengan gamis ataupun tunik. Pengambilan panjang ciri CSS yang berbeda berpengaruh pada hasil TKCI. Semakin panjang ciri, semakin rendah unjuk kerja temu kembali, karena ciri yang panjang memuat puncak-puncak kecil dari peta kontur CSS yang berperan sebagai derau. Dalam hal ini, panjang ciri 4 memberikan hasil terbaik.

## UCAPAN TERIMA KASIH

Penelitian ini didanai oleh Direktorat Jenderal Penguatan Riset dan Pengembangan, Kementerian Riset, Teknologi, dan Pendidikan Tinggi melalui Hibah Penelitian Tim Pascasarjana tahun 2016.

## REFERENSI

- [1] Y.A. Aslandogan dan C.T. Yu, "Techniques and Systems for Image and Video Retrieval", *IEEE Transactions on Knowledge and Data Engineering*, vol.11, no.1, pp. 56-63, Feb, 1999.
- [2] F. Mokhtarian, S. Abbasi dan J. Kittler, "Robust and Ecient Shape Indexing through Curvature Scale Space", *In Proceedings of British Machine Vision Conference (BMVC)*, Sep, 1996.
- [3] S. Abbasi, F. Mokhtarian dan J. Kittler, "Curvature Scale Space image in shape similarity retrieval", *Journal Multimedia Systems*, vol. 7, no.6, pp. 467 – 476, Nov. 1999.
- [4] D. Jamkhandikar dan V.D. Mytri, "CSS Based Trademark Retrieval System", *In Electronic Systems, Signal Processing and Computing Technologies (ICESC)*, Jan. 2014.

- [5] S. Liu, et.al., "Street-to-shop: Cross Scenario Clothing Retrieval via Parts Alignment and Auxiliary Set", *Proceedings of the 20th ACM international conference on Multimedia*, pp. 1335-1336, Nov. 2012.
- [6] K. Kawinakrathiti dan S. Phimoltares, "A Comparative Study of CBIR Descriptors on Innovative Application of Fashion Image", *In Digital Information and Communication Technology and it's Applications (DICTAP)*, May. 2014.
- [7] C.M. Huang, S. Chen, et al, "A Sparse-Coding Based Approach to Clothing Image Retrieval", *In Intelligent Signal Processing and Communications Systems (ISPACS)*, Nov. 2012.
- [8] C. H. Tseng, Chin-Hsien, et al, "An Efficient Garment Visual Search Based on Shape Context", *Journal WSEAS Transactions on Computers*, vol. 8, no. 7, pp. 1195-1204, Jul. 2009.
- [9] J.-J. Tsay, et al, "Visual Clothing Search by Shape and Style", *Journal Information Technology*, vol.13, no. 6, pp.1001-1013, 2014.
- [10] Thomson Reuters and D. Standard, "Global Islamic," 2015.
- [11] K. Durga Sreenivas, et.al., "Contour Aproximation of Image Revognition by using Curvature Scale-space and Invariant Moment-Based Method", *International Journal of Advances in Engineering & Technology*, vol.7, No. 1, pp. 359-371, Mar. 2014.
- [12] P.Rajashekar Reddy, V.Amarnadh dan M. Bhaskar, "Evaluation of Stopping Criterion in Contour Tracing Algorithms", *Journal Computer Science and Information Technologies (CSIT)*, vol.3, no. 3, pp. 3888-3894, Jan. 2012.
- [13] D. Zhang dan G. Lu, "Evaluation of MPEG-7 Shape Descriptors Against Other Shape Descriptors", *Journal Multimedia Systems*, vol. 9, no.1, pp. 15-30, Jul. 2003.
- [14] W. Bret, "An Exploration of MPEG-7 Shape Descriptors", Thesis, Rochester Institute of Technology, New York, 2003.
- [15] G. Çiftçi, "Shape Analysis Using Contour-Based and Region-Based Approaches", Disertasi, Middle East Technical University, 2003.