

# Metode *Convex Hull* dan *Convexity Defects* untuk Pengenalan Isyarat Tangan

Nina Yuliana<sup>#1</sup>, Ken Ratri Retno Wardani<sup>#2</sup>

<sup>#</sup>Departemen Teknik Informatika - Institut Teknologi Harapan Bangsa

Jl. Dipatiukur no. 83 – 84 Bandung

<sup>1</sup>yuliana.nyna@gmail.com

<sup>2</sup>ken\_ratri@ithb.ac.i

**Abstract**— In this study apply hand gesture recognition to interact with the computer. Hand gesture to be recognized consists of eight cues is a combination of the thumb, forefinger and little finger to represent the basic tetikus operations. Each finger in a stretched position, does not fit, and the hands are used in this study are left hand. Background models and samples of skin color is required as an input. Image processing is used to obtain the hand segmentation using Freeman chain code algorithms and the results will be used in the process of feature extraction. The feature extraction process consists of algorithms Convex hull to find points as fingertips, convexity defects lookout point defects as a description of the fingers, and palms central point detection using maximum Inscribed circle (MIC). The results showed that the method of Convex Hull and Convexity Defects managed to recognize hand signals with an accuracy value of more than 90% with sufficient lighting conditions.

**Keywords**— Human computer interaction, hand gesture recognition, Convex Hull, Convexity Defects, Freeman Chain Code, and Maximum Inscribed Circle.

**Abstrak**— Teknik Interaksi Manusia dan Komputer (IMK) mengalami perkembangan pesat. Dibuktikan dengan banyak teknik baru dengan antarmuka yang alami dan mudah digunakan tanpa peralatan eksternal yang khusus. Pada penelitian ini diterapkan pengenalan isyarat tangan pada IMK agar dapat berinteraksi dengan komputer tanpa dibatasi oleh penggunaan tetikus. Isyarat tangan yang akan dikenali terdiri atas 8 isyarat yang merupakan kombinasi dari ibu jari, jari telunjuk, dan jari kelingking karena mewakili operasi dasar yang sering digunakan pada tetikus. Setiap jari dalam posisi direntangkan dan tidak berhimpitan. Tangan yang digunakan pada penelitian ini adalah tangan kiri. Background model dan sampel warna kulit diperlukan sebagai masukan. Pengolahan citra digunakan untuk mendapatkan segmentasi tangan dengan menggunakan algoritma Freeman Chain Code. Hasilnya digunakan pada proses ekstraksi fitur. Proses ekstraksi fitur terdiri atas algoritma: Convex Hull untuk mencari titik Hull sebagai ujung jari, Convexity Defects mencari titik defects sebagai deskripsi jari, dan Maximum Inscribed Circle (MIC) untuk deteksi titik pusat telapak tangan. Hasil penelitian berhasil mengenali isyarat tangan dengan nilai akurasi lebih dari 90% dengan kondisi pencahayaan kurang dan cukup.

**Kata kunci**— Interaksi manusia komputer, pengenalan isyarat tangan, Convex Hull, Convexity Defects, Freeman Chain Code, Maximum Inscribed Circle.

## I. PENDAHULUAN

Teknologi komputer telah berkembang pesat selama satu dekade terakhir. Seiring kemajuan teknologi mengakibatkan terjadi perubahan teknik Interaksi Manusia dan Komputer (IMK). Saat ini keyboard dan tetikus merupakan perangkat yang paling umum digunakan untuk berinteraksi. Paradigma yang ada di dalam pengembangan IMK, yaitu interaksi dalam bentuk isyarat gerak bersama dengan teknik pengenalan wajah, pengenalan suara, pelacakan gerakan mata, dan bibir. Teknik interaksi tersebut dikenal dengan istilah *Perceptual user interface* (Pui) [1]. Dalam [1] menjelaskan tujuan dari Pui adalah untuk meningkatkan efisiensi dan kemudahan penggunaan untuk berinteraksi.

Dalam komputasi personal, aksi adalah hal yang paling sering dilakukan terkait dengan perintah masukan. Untuk penggunaan di lingkungan khusus, misalnya di pabrik atau pengguna dengan disabilitas, gerakan dapat diterapkan untuk kemudahan berinteraksi karena interaksi lebih alami. Isyarat gerak (*gesture recognition*) adalah interpretasi matematis gerakan manusia dengan perangkat komputasi [1].

Penelitian ini membahas teknik pengenalan isyarat tangan menggunakan metode *Convex Hull* dan *Convexity Defects*. Kedua metode tersebut adalah suatu teknik pengenalan fitur dengan menemukan titik *Hull* dan *defect* pada kontur tangan [5]. Fitur tersebut berguna untuk membantu menemukan jari tangan, di mana setiap isyarat diklasifikasikan berdasarkan perhitungan gerakan jari yang diangkat atau dinaikkan. *Convexity defects* diwakili dengan fitur yang terdiri atas titik awal dan titik akhir sebagai ujung jari dan titik kedalaman, yaitu ruang antar jari. *Convex Hull* merupakan deskripsi bentuk tangan yang terdiri atas set obyek lengkungan yang mengandung tepi. Masukan adalah citra kontur tangan dari hasil prapemrosesan. Citra kontur tangan akan identifikasi untuk menghasilkan keluaran berupa perintah dari isyarat tangan.

## II. PERANCANGAN

Isyarat tangan yang akan dikenali dalam penelitian ini terdiri atas 8 isyarat tangan (lihat Tabel I) yang dipilih berdasarkan aksi dasar yang paling sering digunakan ketika bekerja dengan menggunakan tetikus. Isyarat tangan dilaku-

kan dengan berbagai kombinasi jari, yaitu ibu jari, jari telunjuk dan jari kelingking, sebagai operasi dasar pada tetikus. Masing-masing jari dalam posisi direntangkan dan tidak berhimpitan. Tangan yang digunakan pada penelitian ini adalah tangan kiri. Gambar tangan kiri pada Tabel I nampak seperti tangan kanan karena tangan kiri yang ditangkap oleh *webcam* diatur dengan program *mirror*.

Sistem pengenalan isyarat tangan terdiri atas beberapa tahapan yang harus dilakukan. Pertama, sistem meminta pengguna untuk memasukkan *background* model dan sampel warna tangan. Kedua, pengolahan citra dilakukan untuk mendapatkan segmentasi tangan sebagai masukan untuk proses ekstraksi fitur. Ketiga, proses ekstraksi fitur dilakukan dengan mengambil fitur penting dari hasil segmentasi tangan untuk pengolahan isyarat tangan lebih lanjut. Proses ini dilak-

kukan dengan *Freeman Chain Code* untuk mendeteksi kontur tangan, *Convex Hull* untuk mencari titik *Hull* sebagai ujung jari, *Convexity Defects* untuk mencari titik sebagai deskripsi jari, dan *Maximum Inscribed Circle* untuk mencari titik pusat telapak tangan. Keempat, hasil ekstraksi fitur dimasukkan ke dalam proses identifikasi berdasarkan karakteristik ciri yang didapatkan.

Keluaran sistem menghasilkan perintah dari isyarat tangan sesuai gerakan yang diberikan *user* secara kontinyu. *Flowchart* sistem pengenalan isyarat tangan menggunakan metode *Convex Hull* dan *Convexity Defects* ditunjukkan pada Gambar 1.

A. *Input Frame*








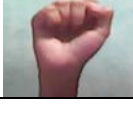
Citra isyarat tangan diperoleh dengan menggunakan *webcam*. *Frame 1* diproses mulai dari pengolahan citra hingga proses pengenalan *gesture*. Proses yang sama diulang untuk *frame 2*, *frame 3*, sampai *frame* terakhir dengan kecepatan 10 fps. *Frame* yang dimaksud adalah citra diam berukuran 640 x 480 piksel dengan ruang warna RGB dan jarak tangan dengan *webcam* antara 30-40 cm agar isyarat tangan dapat dikenali dengan baik.

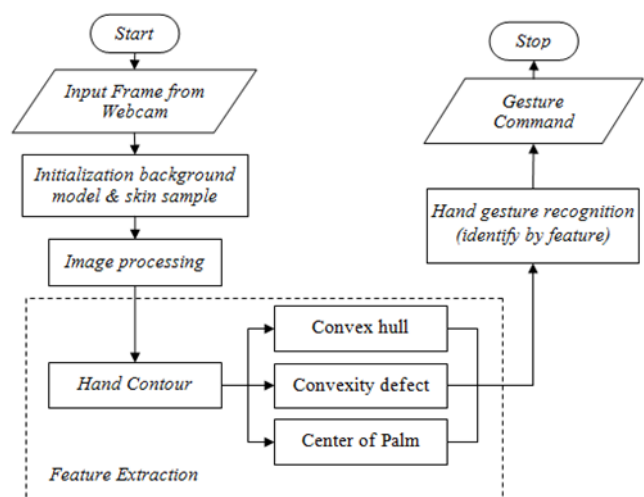
B. *Inisialisasi Background Model dan Sampel Kulit*

Proses inisialisasi *background* model adalah mengambil obyek latar belakang sebagai acuan menggunakan *webcam* dengan ukuran yang sama, yaitu 640 x 480 piksel dengan ruang warna RGB. Untuk menghapus latar belakang dari obyek yang akan dikenali menggunakan metode *background subtraction* sesuai dengan teori [3].

Kedua, lakukan inisialisasi sampel area warna kulit yang akan digunakan sebagai acuan untuk mendeteksi area tangan. Pada implementasinya sampel warna kulit pengguna direkam dengan menempatkan telapak tangan untuk mendapatkan area kulit pada kotak hitam yang tersedia pada layar dengan ukuran

TABEL I  
KOMBINASI JARI SEBAGAI POLA ISYARAT TANGAN

No	Ilustrasi	Isyarat Tangan	Karakteristik jari
G1		<i>Start</i>	Kelima jari tangan yang direntangkan
G2		<i>Pointer</i>	Hanya jari telunjuk yang direntangkan
G3		<i>Left click</i>	Ibu jari dan jari telunjuk direntangkan
G4		<i>Right click</i>	Jari telunjuk dan jari kelingking direntangkan
G5		<i>Double click</i>	Ibu jari, jari telunjuk dan jari kelingking direntangkan
G6		<i>Scroll up</i>	Hanya jari kelingking yang direntangkan
G7		<i>Scroll down</i>	Hanya ibu jari yang direntangkan
G8		<i>Stop</i>	Tidak ada jari yang direntangkan, telapak tangan mengempal



Gambar 1. *Flowchart* pengenalan isyarat tangan

50 x 50 piksel. Area tersebut digunakan sebagai sampel warna kulit yang akan digunakan pada proses deteksi tangan.

C. *Pengolahan Citra Isyarat Tangan*

Tahap prapemrosesan citra bertujuan untuk mengoreksi citra dan menentukan segmentasi obyek tangan (*foreground*) sebelum dilakukan proses ekstraksi fitur (lihat Gambar 2).

Langkah pertama yang diterapkan adalah proses *background subtraction* ditentukan dengan nilai *threshold* global 50. Jika nilai  $R \leq 50$ , maka piksel diubah menjadi hijau. *Background subtraction* dirumuskan sebagai berikut [3]:

$$R = |I_r - B_r| + |I_g - B_g| + |I_b - B_b| \quad (1)$$

$I$  adalah *frame* saat ini,  $B$  adalah *background* model, intensitas warna RGB diwakili variabel  $r = red$ ,  $g = green$ , dan  $b = blue$ .

RGB dikonversi ke ruang warna HSV. Warna HSV dipilih dalam penelitian ini karena model warna ini lebih sensitif terhadap perubahan pencahayaan [6] dan sesuai dengan persepsi penglihatan manusia [8].

Histogram *back projection* dihitung untuk mendeteksi warna kulit dan nonkulit untuk memperoleh hasil segmentasi tangan berdasarkan warna kulit manusia pada citra [7].

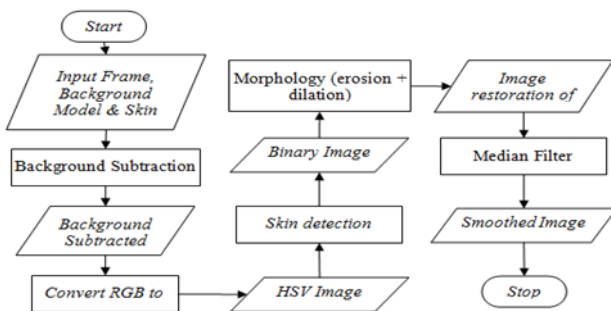
Masukan adalah citra tangan dari *webcam* dan model sampel kulit dengan warna HSV. Keluaran dari proses ini berupa citra biner dengan nilai *bin* 30, 9 dan 16 yang digunakan pada masing-masing histogram.

Berikut ini adalah algoritma histogram *back projection* yang dibuat berdasarkan teori [7]:

1. Hitung histogram warna dari kedua citra masukan, yaitu sampel warna kulit ( $M$ ) dan citra *input*, citra yang akan dilakukan pencarian ( $I$ ) dengan persamaan:

$$\text{hist}_{(x,y)} = \text{rounddown} \left( \frac{c * k}{L} \right) \quad (2)$$

$c$  adalah warna (nilai *hue* dan *saturation*),  $k$  jumlah bin dan  $L$  adalah nilai kuantisasi derajat keabuan.



Gambar 2 *Flowchart* prapemrosesan citra

2. Hitung rasio  $R$  dengan persamaan:

$$R_i = \min \left( \frac{M_i}{I_i}, 1 \right) \quad (3)$$

3. Gunakan  $R$  sebagai *palette* dan buat citra baru dengan probabilitas tiap piksel dengan rumus:

$$B_{(x,y)} = R[\text{hist}_{(x,y)}] \quad (4)$$

4. Ubah nilai *hue* dan *saturation* sesuai *threshold*. Jika probabilitasnya lebih besar dari 0, maka atur piksel biner sama dengan 1. Jika lebih kecil dari 0, atur piksel biner sama dengan 0.

Proses selanjutnya adalah morfologi citra dengan menerapkan erosi kemudia dilasi. Hal ini bertujuan untuk memperhalus batas-batas obyek, memisahkan obyek yang berdempetan, dan menghilangkan obyek-obyek yang memiliki ukuran lebih kecil dari struktur elemen [9]. Struktur elemen yang digunakan adalah seperti yang ditunjukkan pada Gambar 3, yaitu *cross* 3 x 3 dengan *neighbors-4*.

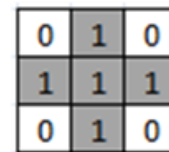
Langkah terakhir adalah melakukan proses *median filter* untuk membuang *noise* pada citra [9].

D. *Ekstraksi Fitur*

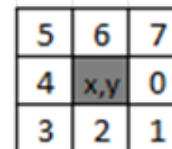
Ekstraksi fitur adalah proses pengambilan informasi yang terdapat dalam suatu citra. Informasi tersebut merupakan masukan pada proses pengenalan yang menghasilkan keluaran berupa perintah isyarat tangan yang diberikan. Proses ekstraksi fitur yang akan dilakukan terdiri dari tahapan sebagai berikut.

1) *Deteksi kontur tangan*

Pada tahap ini, proses deteksi kontur tangan dilakukan untuk menemukan kontur citra dengan asumsi bahwa kontur tangan merupakan kontur terbesar, sehingga mengabaikan semua kontur lain pada citra. Proses deteksi kontur dilakukan menggunakan metode *Freeman Chain Code* [2] dengan kode 8-arah (lihat Gambar 4). Kontur obyek tangan digunakan untuk proses pengenalan isyarat tangan.



Gambar 3 *Structuring* elemen 3x3 dengan *neighbors-4*



Gambar 4 *Mask* persegi 3x3 dengan kode 8-arah

Berikut algoritma *Freeman chain code* dibuat berdasarkan teori [2].

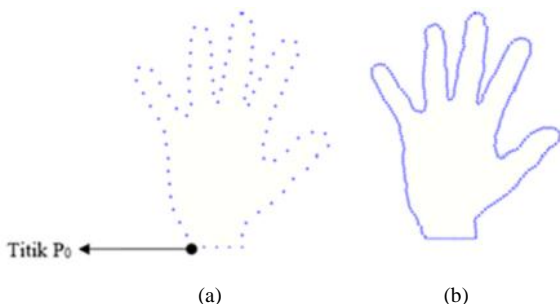
1. Dimulai dari piksel [0][0] di dalam citra, temukan titik awal  $s_1$ , yaitu piksel warna putih. Piksel sebelumnya adalah hitam dan piksel tersebut belum diberi *flag* 1.
2. Posisikan titik origin sesuai kode 8-arah pada titik awal  $x,y$ .
3. Lakukan pengecekan berdasarkan urutan kode 8-arah, jika arah menunjukkan ke piksel putih, dan arah selanjutnya (*next*) merupakan piksel hitam, maka titik origin pindah ke piksel putih (*current*). Beri *flag* 1 sebagai daerah yang sudah ditelusuri dan simpan lintasan ke *temp\_list*.
4. Lakukan pengulangan sampai titik awal ditemukan kembali atau ke arah lain tidak ditemukan (buntu) untuk mengakiri proses.

Hasil dari deteksi kontur *Freeman Chain Code* ditunjukkan pada Gambar 5. Lintasan ditelusuri dan kontur disimpan dalam daftar sesuai dengan urutan pencarian lintasan. Titik korespondensi merupakan representasi dari kontur tangan dengan merenggangkan jari tangan.

Titik korespondensi yang dibutuhkan penelitian ini terdiri atas 2 titik, yaitu representasi titik kontur dengan jarak 20 dan 5 (lihat Gambar 6). Jarak tersebut dipilih berdasarkan analisis bahwa jarak 20 ideal digunakan pada deteksi *Convex Hull* dan *Convexity Defects* karena paling baik untuk merepresentasikan ujung jari. Jika jarak terlalu kecil, maka kumpulan titik pada ujung jari menjadi rapat. Untuk proses pencarian titik pusat telapak tangan adalah sebaliknya. Dibutuhkan titik korespondensi yang rapat agar pencarian dapat dilakukan dengan tepat di pusat telapak tangan.



Gambar 5 Hasil *Freeman chain code*



Gambar 6 Titik korespondensi (a) Titik korespondensi 20 (b) Titik korespondensi 5

Masukan dari algoritma titik korespondensi adalah daftar kontur hasil deteksi *Freeman Chain Code* dan keluarannya, yaitu titik korespondensi dengan titik pertama dengan koordinat  $y$  terkecil.

Berikut ini adalah algoritma titik korespondensi yang dibuat.

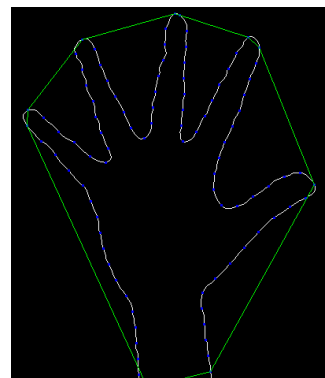
1. Atur reset lokasi = 0.
2. Lakukan pengulangan terhadap daftar kontur dengan melakukan pengecekan: Jika koordinat  $x = height - 1$ , maka simpan lokasi sesuai indeks dan hentikan pencarian.
3. Lakukan pengulangan terhadap daftar (*list*) kontur sebanyak panjang *list* dengan melakukan pengecekan untuk melakukan penyortiran *list* kontur.
  - Jika lokasi + indeks < lokasi *list* kontur indeks terakhir, maka lokasi\_baru = lokasi + indeks
  - Jika tidak, maka lokasi\_baru = reset lokasi dan reset lokasi = reset lokasi + 1
4. Lakukan pengulangan terhadap *list* kontur dengan melakukan pengecekan: Jika lokasi mod jarak = 0, maka simpan lokasi ( $x,y$ ) ke dalam *list* titik korespondensi.

## 2) *Convex Hull*

Proses *Honvex Hull* diterapkan untuk menghubungkan titik-titik membentuk *polygon* yang merupakan simpul dari kontur tangan dengan menggunakan algoritma *Graham's scan* [4]. Hasilnya diperlihatkan pada Gambar 7. Simpul poligon ditentukan dengan menghitung tiga titik silang  $P_1 = (x_1,y_1)$ ,  $P_2 = (x_2,y_2)$  dan  $P_3 = (x_3,y_3)$ .

Berikut ini adalah algoritma *Graham's scan*:

1. Inisialisasikan *list hull*.
2. Atur *counter* = 2.
3. Simpan array  $p[0]$  dan  $p[1]$  ke dalam *list hull*.
4. Hitung arah titik  $p[0]$ ,  $p[1]$ , dan  $p[2]$  dengan rumus:
 
$$CH = (x_2 - x_1)(y_3 - y_1) - (y_2 - y_1)(x_3 - x_1) \quad (5)$$
5. Jika  $CH > 0$ , maka  $p[1]$  diterima dan  $p[2]$  dimasukkan ke dalam *list Hull*. Tambahkan *counter* + 1.
6. Jika  $CH = 0$ , maka pada garis lurus tambahkan *counter* + 1.
7. Jika tidak memenuhi kriteria sebelumnya, hapus titik terakhir pada *list Hull*.



Gambar 7 Hasil simpul kontur tangan dengan *Graham's scan*

3) *Convexity defects*

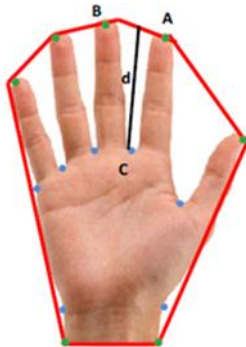
Proses *Convexity Defects* dilakukan untuk menemukan *defect* antara *Convex Hull* dan kontur [4]. *Defect* berguna untuk menemukan fitur penting pada kontur tangan dan untuk menemukan poligon 4 elemen, seperti *start point*, *end point*, *depth point* dan *depth* yang disimpan dalam *array* untuk diolah lebih lanjut. Pada Gambar 8, titik hijau merupakan titik *Convex Hull* dan titik biru merupakan titik *convexity defect*. Gambar 9 menampilkan hasil dari *convexity defects*. Dari sejumlah set titik *Hull* yang ditemukan dilakukan pengecekan titik kedalaman (*depth point*) dan panjang kedalaman (*depth*). Jika memenuhi kriteria, maka *start point* pada *convexity defect* diidentifikasi sebagai jari tangan.

Pada masing-masing *tuple* antara titik A dan titik B, titik C merupakan jarak maksimum dari garis yang menghubungkan A dan B. Jarak antara C(x<sub>0</sub>,y<sub>0</sub>) dengan garis A(x<sub>1</sub>,y<sub>1</sub>) dan B(x<sub>2</sub>,y<sub>2</sub>) dihitung menggunakan persamaan berikut [9].

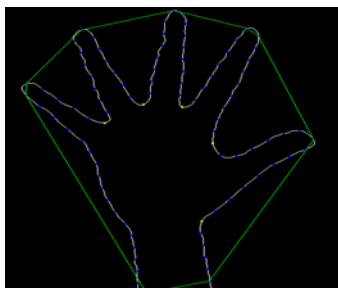
$$d = \frac{|(x_2 - x_1)(y_2 - y_1) - (x_1 - x_0)(y_2 - y_1)|}{\sqrt{(x_2 - x_1)^2 - (y_2 - y_1)^2}} \quad (6)$$

4) *Deteksi pusat telapak tangan*

Deteksi titik pusat telapak tangan pada kontur ditentukan menggunakan metode *Maximum Inscribed Circle* (MIC) [10]. Untuk menghitung titik pusat telapak tangan dihitung dari lingkaran terbesar pada kontur.



Gambar 8 *Convex hull* dan *convexity defects*



Gambar 9 Hasil *convexity defects*

Langkah pertama, yaitu menentukan 3 titik (P<sub>1</sub>,P<sub>2</sub>,P<sub>3</sub>) pada segitiga lancip.

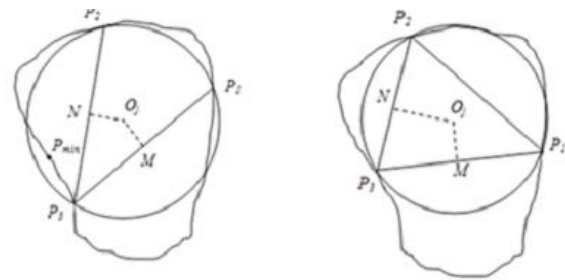
$$\text{Acute Triangle} = P_1 < 90^\circ \text{ AND } P_2 < 90^\circ \text{ AND } P_3 < 90^\circ \quad (7)$$

Untuk menghitung titik pusat O<sub>j</sub> pada segitiga (Gambar 10) menggunakan persamaan 8 dan 9 berikut ini. Variable X dan Y menunjukkan posisi koordinat dari titik yang terpilih. Nilai X<sub>m</sub>,Y<sub>m</sub> adalah koordinat titik tengah antara titik P<sub>1</sub> dan P<sub>3</sub>. X<sub>n</sub>,Y<sub>n</sub> adalah titik tengah antara titik P<sub>1</sub> dan P<sub>2</sub>. Nilai X<sub>j</sub>,Y<sub>j</sub> adalah koordinat titik pusat.

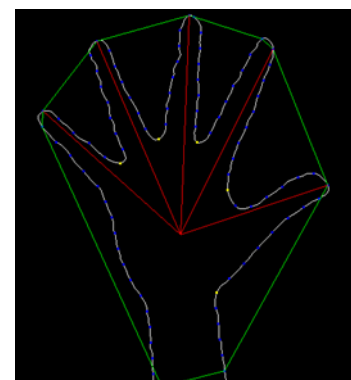
Gambar 11 menunjukkan hasil ekstraksi fitur dengan titik pusat telapak tangan yang terdeteksi.

$$\begin{aligned} X_M &= \frac{X_1 + X_3}{2} & X_N &= \frac{X_2 + X_3}{2} \\ Y_M &= \frac{Y_1 + Y_3}{2} & Y_N &= \frac{Y_2 + Y_3}{2} \end{aligned} \quad (8)$$

$$\begin{aligned} X_j &= \frac{(Y_M - Y_N)(Y_3 - Y_1)(Y_3 - Y_2) + X_M(X_3 - X_1)(Y_3 - Y_2) - X_N(X_3 - X_2)(Y_3 - Y_1)}{(X_3 - X_1)(Y_3 - Y_2) - (X_3 - X_2)(Y_3 - Y_1)} \\ Y_j &= \frac{(X_M - X_N)(X_3 - X_1)(X_3 - X_2) - Y_M(Y_3 - Y_1)(X_3 - X_2) + Y_N(Y_3 - Y_2)(X_3 - X_1)}{(X_3 - X_1)(Y_3 - Y_2) - (X_3 - X_2)(Y_3 - Y_1)} \end{aligned} \quad (9)$$



Gambar 10 *Maximum inscribed circle* (a) Referensi lingkaran awal (b) Lingkaran mendekati MIC



Gambar 11 Hasil *maximum inscribed circle*



### E. Pengenalan Isyarat Tangan

Pada bagian ini, telah dilakukan ekstraksi fitur penting dari isyarat tangan. Implementasi pengenalan isyarat tangan terdiri atas tahapan sebagai berikut:

- Identifikasi jari tangan sebagai pola referensi. Pada tahap ini, implementasi identifikasi jari tangan sebagai pola referensi diperoleh dengan cara pengguna memilih tombol “*Start Recognition*” lalu sistem menyimpan pola isyarat tangan sebagai referensi untuk acuan pengenalan isyarat tangan. Pola isyarat tangan terdiri atas referensi sudut masing-masing jari [10] untuk dicocokkan pada pola untuk proses pengenalan isyarat tangan.
- Identifikasi jari tangan untuk mengidentifikasi jari yang diperoleh dari algoritma *Convex Hull* dan titik pusat telapak tangan berdasarkan referensi sudut masing-masing jari disimpan di dalam daftar. Asumsinya adalah jika jari dianggap valid (jari ditinggikan atau direntangkan), maka:
  - koordinat  $y$  dari titik *Hull* (ujung jari) lebih besar dari koordinat  $y$  titik pusat telapak tangan.
  - jarak jari dari pusat telapak tangan dengan titik  $Hull > avg_{jari}$  yang diperoleh dari pola referensi.
- Bagian terakhir dilakukan identifikasi isyarat tangan. Pada proses ini dilakukan pengenalan isyarat dari identifikasi jari tangan pada proses sebelumnya. Identifikasi isyarat tangan berdasarkan kombinasi jari yang ditinggikan atau direntangkan untuk pola isyarat tangan yang ditentukan adalah jika hasil identifikasi lulus, maka isyarat tangan dapat dikenali oleh sistem.

### III. IMPLEMENTASI

Spesifikasi perangkat lunak yang diperlukan untuk mengimplementasikan aplikasi terdiri atas:

- Sistem operasi: *Windows 7 Ultimate 64-bit Service Pack 1*.
- *Development Tools: Java Development Kit 1.7.0 32-bit, Netbeans IDE 7.3.*

### IV. PENGUJIAN

Tujuan pengujian isyarat tangan adalah untuk mengetahui akurasi dan kelemahan dari metode pemrosesan yang ada. Masing-masing pengujian dilakukan sebanyak 10 kali yang meliputi:

- Pengaruh kondisi pencahayaan terhadap hasil pengenalan kontur tangan.
- Proses pengolahan citra tujuan untuk mendapatkan hasil segmentasi terbaik.
- Pengujian gestur (sesuai Tabel I).

#### A. Hasil Pengujian Pencahayaan

Pengujian dilakukan terhadap pencahayaan cukup, kurang, dan sangat terang (terpapar sinar matahari). Jarak *webcam* dengan tangan adalah sekitar 30-40cm dengan lingkungan pengujian sebagai berikut:

- Pencahayaan cukup dengan kondisi di dalam ruangan berukuran 5x4m dengan lampu ruangan menyala (*Philips™ 24w-63 lumen/watt*) dan kondisi tidak ada cahaya matahari pada malam hari. Semua kontur berhasil dikenali dengan baik, sehingga proses pengenalan pola berjalan dengan baik.
- Pencahayaan kurang dengan kondisi di dalam ruangan berukuran 5x4m dengan lampu ruangan padam dan sangat minim cahaya matahari pada jam 6 sore (pantulan cahaya hanya dari layar laptop). Semua kontur berhasil dikenali dengan baik, sehingga proses pengenalan pola berjalan dengan baik.
- Pencahayaan sangat terang dengan kondisi di dalam ruangan 5x4m dengan lampu ruangan menyala (*Philips™ 24w-63 lumen/watt*) dan ada cahaya matahari pada jam 9 pagi. Kontur berhasil dikenali meskipun tidak sempurna. Jari tangan tidak maksimal dikenali karena ada bagian yang terpotong-potong. Paparan sinar matahari menyebabkan segmentasi tangan terganggu, sehingga kontur yang ditemukan tidak stabil.

#### B. Pengujian Prapemrosesan

Pengujian ini bertujuan untuk mendapatkan hasil prapemrosesan citra terbaik. Pengujian melibatkan delapan kombinasi metode pengolahan citra (lihat Tabel II).


Hasil pengolahan citra yang memberikan hasil terbaik adalah pengujian 5 *Background Subtraction* dan erosi. Keluaran yang diberikan dapat mengenali kontur dengan baik, sehingga isyarat dapat dikenali. Proses median filter 5 x 5 dapat menghilangkan *noise* yang hilang tetapi tidak dapat mengenali kontur.


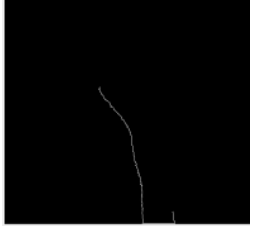
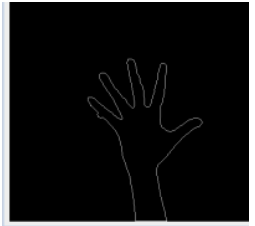


#### C. Pengujian Gestur



Pengujian *gesture* ditunjukkan pada Tabel 1, yaitu: isyarat *start, pointer, left click, right click, double click, scroll up, scroll down, dan stop*. Pengujian dilakukan dengan pencahayaan yang cukup dan kurang serta akurasi 100% yang berarti hasil pengenalan dapat sesuai. Jika pencahayaan berlebihan, maka proses segmentasi tidak berhasil. Gerakan posisi tangan terhadap cahaya mempengaruhi hasil segmentasi yang tidak stabil, sehingga hasil pengenalan isyarat tidak dikenali oleh aplikasi (lihat Tabel III).

TABEL II

PENGUJIAN PRA- PEMROSESAN





No	Skenario uji dan keluaran	Analisis
1	<i>Background subtraction</i> dan morfologi: 	Hasil morfologi minim <i>noise</i> , kontur dapat dideteksi dengan baik, dan isyarat tangan dapat dikenali.





No	Skenario uji dan keluaran	Analisis
2	Morfologi tanpa <i>background subtraction</i> : 	Terdapat <i>noise</i> , kontur yang dihasilkan dapat dideteksi isyarat tangan, dan isyarat tangan dapat dikenali.
3	<i>Background subtraction</i> tanpa morfologi: 	Kontur tidak dapat dideteksi karena nilai <i>threshold</i> . Deteksi kontur tidak dapat diproses dengan baik, sehingga hasil pengenalan tidak dikenali.
4	<i>Background subtraction</i> dan <i>median filter</i> : 	Isyarat tangan dan kontur berhasil dikenali dengan baik, meskipun kontur tangan menjadi lebih tipis.
5	<i>Background subtraction</i> dan erosi: 	Isyarat tangan dan kontur berhasil dikenali dengan baik. Hasil kontur lebih baik dari pengujian no 4.
6	<i>Background subtraction</i> dan dilasi: 	Kontur yang dihasilkan lebih tipis dari pengujian 4, sehingga tepi jari rentan terputus dari kontur. Hasil pengenalan tidak sesuai dengan yang diharapkan.

No	Skenario uji dan keluaran	Analisis
7	<i>Median filter 3x3</i> : 	<i>Noise</i> dapat dihilangkan dengan proses yang lebih cepat dibandingkan <i>median 5x5</i> . Isyarat tangan dan kontur berhasil dikenali dengan baik.
8	<i>Median filter 5x5</i> : 	<i>Noise</i> yang dihasilkan lebih minim, namun pengolahan citra menjadi lebih lambat (pengujian <i>real time</i> ). Kontur tidak dapat dideteksi. Pengenalan tidak sesuai dengan yang diharapkan.

TABEL III

PENGUJIAN GESTUR PENCAHAYAAN CUKUP

No	Skenario uji	Hasil analisis
1.	Isyarat Tangan “ <i>Start</i> ”: 	Gestur 1 berhasil dikenali dengan baik sesuai dengan pola referensinya.
2.	Isyarat Tangan “ <i>Pointer</i> ”: 	Gestur 2 berhasil dikenali dengan baik.
3.	Isyarat Tangan “ <i>Left Click</i> ”: 	Gestur 3 berhasil dikenali dengan baik.
4.	Isyarat Tangan “ <i>Right Click</i> ”: 	Gestur 4 berhasil dikenali dengan baik.

No	Skenario uji	Hasil analisis
5.	Isyarat Tangan “ <i>Double Click</i> ”: 	Gestur 5 berhasil dikenali dengan baik.
6.	Isyarat Tangan “ <i>Scroll Up</i> ”: 	Gestur 6 berhasil dikenali dengan baik.
7.	Isyarat Tangan “ <i>Scroll Down</i> ”: 	Gestur 7 berhasil dikenali dengan baik.
8.	Isyarat Tangan “ <i>Stop</i> ”: 	Gestur 8 berhasil dikenali dengan baik.

## V. KESIMPULAN

Kesimpulan di dalam penelitian ini adalah metode *Convex Hull* dan *Convexity Defects* mampu mengenali dan membedakan 8 isyarat dengan nilai akurasi lebih dari 90% dengan kondisi pencahayaan yang cukup dan kurang.

Ekstraksi fitur *Convex Hull*, *Convexity Defects*, dan titik pusat telapak tangan cukup untuk merepresentasikan fitur tangan untuk dijadikan sistem pengenalan isyarat tangan.

Kesalahan identifikasi isyarat tangan disebabkan oleh pencahayaan yang berlebihan, sehingga hasil segmentasi tangan menjadi tidak stabil dan posisi ibu jari yang terlalu renggang menyebabkan tidak dianggap sebagai jari yang valid.

Algoritma perhitungan titik pusat telapak tangan dapat dioptimalisasi dengan menentukan 3 titik uji di 3 area yang tepat, sehingga tidak perlu melakukan banyak pengulangan.

## DAFTAR REFERENSI

- [1] Achintya K. Bhowmik, *Interactive Displays: Natural Human-Interface Technologies*, John Wiley & Sons, Ltd., 2015.
- [2] Andrzej Dziech, *Contour Extraction and Compression- Selected Topics*, 2007.
- [3] Birgi Tamersoy, *Background Subtraction*, courses University of Texas 2009.
- [4] Ghassem Tofighi, Nasser Ali Afarin, Kamraan Raahemifar, dan Anastasios N. Venetsanopoulos, “Hand Pointing Detection Using Live Histogram Template of Forehead Skin”, 2014.
- [5] Hazem Khaled, Samir Gaber, El Sayed Mostafa dan Hossam Ali, “Hand gesture recognition using average background and logical heuristic equations”, *International Journal of Computers and Technology* vol 11, no 5 pp 2635 - 2640, Oct 2013.
- [6] Joyeeta Singha dan Karen Das, *Hand Gesture Recognition Based on Karhunen-Loeve Transform*, 2013.
- [7] Michael J. Swain dan Dana H. Ballard, *Color Indexing*, 1991.
- [8] Priyanto Hidayatullah dan Hubert Konik, *CAMSHIFT Improvement on Multi-Hue and Multi-Object Tracking*, 2011.
- [9] R.C. Gonzales, R.E. Woods, *Digital Image Processing*, Third Edition, Pearson Prentice-Hall, 2010.
- [10] Zhengwei Yao, Zhigeng Pan, Shuchang Xu, “Wrist Recognition and the Center of the Palm Estimation Based on Depth Camera”, 2013.

**Nina Yulian**, mahasiswi jurusan Teknik Informatika Institut Teknologi Harapan Bangsa Bandung lulus tahun 2015. Minat penelitian pada pengolahan citra.

**Ken Ratri Retno Wardani**, menerima gelar Sarjana Teknik Informatika dari Sekolah Tinggi Sains dan Teknologi pada tahun 1993. Menyelesaikan studi Magister di Institut Teknologi Bandung jurusan Teknologi Informasi dan aktif sebagai pengajar di Departement Teknik Informatika, Institut Teknologi Harapan Bangsa di Bandung. Bidang penelitian adalah pengolahan citra dan interaksi manusia komputer.