

# Deteksi Wajah Menggunakan Segmentasi Warna Kulit dan *Template Matching* Menggunakan Metode *Modified Chamfer Matching Algorithm*

Vikki Aria Dinata<sup>1</sup>, Saparudin<sup>2</sup>, Julian Supardi<sup>2</sup>

<sup>1</sup> Hitachi eBworx, Petaling Jaya, Malaysia

<sup>2</sup> Fakultas Ilmu Komputer, Universitas Sriwijaya, Palembang, Indonesia

Email: julian@unsri.ac.id

**Abstrak**—Pendeteksian wajah merupakan suatu proses menentukan lokasi wajah pada citra. Pendeteksian wajah biasanya digunakan untuk tahapan awal dari proses pengenalan wajah. Banyak metode yang telah dikembangkan untuk proses pendeteksian wajah tersebut, salah satunya menggunakan metode *template matching*. Pada penelitian ini digunakan metode *template matching* menggunakan *modified chamfer matching algorithm* yang menggunakan warna YCbCr untuk proses segmentasi warna kulit. Proses yang dilakukan pertama kali adalah melakukan segmentasi warna kulit untuk mendapatkan kandidat wajah, selanjutnya kandidat tersebut akan digunakan untuk proses *template matching*. Dari 30 citra yang diujikan pada perangkat lunak, metode yang digunakan ini menghasilkan tingkat akurasi 30% untuk wajah yang terdeteksi keseluruhan, 23,33 % untuk wajah yang terdeteksi keseluruhan tapi terdapat objek lain selain wajah yang terdeteksi, 40 % untuk wajah yang tidak terdeteksi keseluruhan, dan 6,66 % untuk tidak ada wajah yang terdeteksi

**Kata Kunci**—*Deteksi wajah, YCbCr, Modified Chamfer Matching Algorithm, Template Matching, Segmentasi warna kulit*

## I. PENDAHULUAN

Wajah merupakan salah satu identitas dari manusia dikarenakan sifat wajah yang unik. Melalui wajah, seseorang dapat dikenali. Manusia dapat mengenali wajah dengan mudah yakni dengan cara mengingat wajah seseorang yang telah dikenal sebelumnya, selanjutnya ketika bertemu lagi dengan orang tersebut maka manusia langsung dapat mengenalinya, lain halnya bagi komputer yang mana hal tersebut merupakan hal yang rumit. Untuk dapat melakukan pengenalan terhadap wajah, komputer memerlukan mekanisme tersendiri. Mekanisme yang dilakukan komputer dalam proses pengenalan wajah mendeteksi wajah, ekstraksi ciri, pencocokan ciri, dan identifikasi wajah.

Pendeteksian wajah merupakan hal khusus dari pendeteksian pola (*pattern detection*) dan biasanya sebagai tahap awal dalam pengenalan wajah (*face recognition*). Pendeteksian wajah pada suatu citra dilakukan dengan penentuan lokasi (posisi) dan ukuran dari setiap wajah (jika ada) dalam suatu citra. Banyak metode yang telah dikembangkan berkaitan dengan pendeteksian wajah. Menurut [1] metode deteksi wajah dapat dikelompokkan menjadi empat kategori, yaitu (1) metode berdasarkan pengetahuan; (2) metode fitur pendekatan invarian; (3) metode pencocokan *template (template matching)*; (4) metode berdasarkan penampilan.

Metode *template matching* merupakan salah satu metode yang telah banyak dikembangkan. Terdapat beberapa algoritma yang digunakan untuk perhitungan jarak antara model gambar dan *template* pada metode *template matching*, seperti *euclidean distance*, *hausdorff distance* dan *chamfer*

*distance*. *Template matching* yang menggunakan *chamfer distance* disebut *chamfer matching*. *Chamfer matching* pertama kali diperkenalkan pada tahun 1977 oleh Barrow et al. Ide asli dari *chamfer matching* memiliki beberapa sifat yang bagus seperti kemampuan untuk menangani data yang tidak sempurna, akan tetapi masih terdapat kesalahan dalam proses pencocokan. Sebagaimana masalah pencocokan telah diperhitungkan secara intensif, munculah beberapa penyelesaian yang merupakan pengembangan *chamfer matching* itu sendiri seperti *hierarchical chamfer matching algorithm (HCMA)* dan *modified chamfer matching algorithm (MCMA)*.

Wajah memiliki warna yang berbeda dibandingkan dengan objek lain yang ada di lingkungan. Wajah memiliki warna seperti anggota tubuh lainnya yaitu warna kulit. Warna kulit sama seperti warna lainnya sehingga bisa didapat rentang nilai yang menyatakan warna tersebut merupakan warna kulit, akan tetapi suatu warna dapat berubah bergantung pada tingkat kecerahan atau pengaruh dari cahaya sehingga rentang warna kulit menjadi berubah-ubah tergantung pada kondisi kecerahan. Warna YCbCr dapat memisahkan nilai kecerahan yang disebut dengan *luminance* dari nilai warna asli. Warna yang ditampilkan oleh warna YCbCr merupakan warna asli tanpa pengaruh *luminance*, sehingga rentang warna kulit dapat ditentukan. Dengan memanfaatkan YCbCr objek didalam suatu citra dapat dibagi menjadi kulit dan non kulit. Objek yang akan dicocokkan pada proses *template matching* hanya objek kulit sehingga hal tersebut akan menghemat waktu proses dan mengurangi resiko kesalahan pencocokan.

Wajah memiliki warna kulit, bentuk dan ukuran tertentu. Warna dari kulit dapat dideteksi dengan menentukan rentang nilai yang menyatakan suatu warna tergolong warna kulit, hal tersebut dapat dilakukan dengan memanfaatkan transformasi warna YCbCr. Setelah melalui proses segmentasi warna kulit, wajah bukanlah satu-satunya objek yang terdeteksi memiliki warna kulit, terdapat objek lain seperti anggota tubuh. Untuk dapat membedakan wajah dengan objek yang juga memiliki warna kulit adalah melalui bentuk dari wajah tersebut. Bentuk dari wajah dapat dikenali dengan menggunakan metode tertentu, salah satu metode tersebut adalah metode *template matching*

## II. STUDI PUSTAKA

### A. Representasi Citra Digital

Citra digital merupakan fungsi intensitas cahaya  $f(x,y)$ , dimana harga  $x$  dan  $y$  merupakan koordinat spasial dan harga fungsi tersebut pada setiap titik  $(x,y)$  merupakan tingkat kecermerlangan citra pada titik tersebut. Citra digital adalah citra  $f(x,y)$  dimana dilakukan diskritisasi koordinat spasial

(sampling) dan diskritisasi tingkat kecerlangannya/keabuan (kuantisasi). Citra digital merupakan suatu matriks dimana indeks baris dan kolomnya menyatakan suatu titik pada citra tersebut dan elemennya matriksnya (yang disebut sebagai elemen gambar / piksel / *pixel* / *picture element* / *pels*) menyatakan tingkat keabuan pada titik tersebut. Citra dengan skala keabuan berformat 8-bit memiliki 256 intensitas warna yang berkisar pada nilai 0 sampai 255. Nilai 0 menunjukkan tingkat paling gelap (hitam) dan 255 menunjukkan nilai paling cerah (putih) [2].

#### B. Deteksi Wajah

Deteksi wajah dapat dipandang sebagai masalah klasifikasi pola dimana masukannya adalah citra masukan dan ditentukan oleh keluaran yang berupa label kelas dari citra tersebut. Dalam hal ini terdapat dua label kelas, yaitu wajah dan non- wajah [3].

Teknik-teknik pengenalan wajah yang dilakukan selama ini banyak yang menggunakan asumsi bahwa data wajah yang tersedia memiliki ukuran yang sama dan latar belakang yang seragam. Di dunia nyata, asumsi ini tidak selalu berlaku karena wajah dapat muncul dengan berbagai ukuran dan posisi di dalam citra dan dengan latar belakang yang bervariasi [4].

Bidang – bidang penelitian yang terkait dengan pemrosesan wajah adalah [1]:

1. Pengenalan wajah (*face recognition*) yaitu membandingkan citra wajah masukan dengan suatu database wajah dan menemukan wajah yang paling cocok dengan citra masukan tersebut.
2. Autentikasi wajah (*face authentication*) yaitu menguji keaslian/kesamaan suatu wajah dengan data wajah yang telah dimasukkan sebelumnya.
3. Lokalisasi wajah (*face localization*) yaitu pendeteksian wajah namun dengan asumsi hanya ada satu wajah di dalam citra.
4. Penjejukan wajah (*face tracking*) yaitu memperkirakan lokasi suatu wajah di dalam video secara *real time*.
5. Pengenalan ekspresi wajah (*facial expression recognition*) untuk mengenali kondisi emosi manusia.

Permasalahan dalam mendeteksi wajah yaitu [1]:

1. Posisi wajah. Posisi wajah di dalam citra dapat bervariasi karena posisinya bisa tegak, miring, menoleh, atau dilihat dari samping.
2. Komponen-komponen pada wajah yang bisa ada atau tidak ada, misalnya kumis, jenggot, dan kacamata.
3. Ekspresi wajah. Penampilan wajah sangat dipengaruhi oleh ekspresi wajah seseorang, misalnya tersenyum, tertawa, sedih, berbicara, dan sebagainya.
4. Terhalang objek lain. Citra wajah dapat terhalangi sebagian oleh objek atau wajah lain, misalnya pada citra berisi sekelompok orang.
5. Kondisi pengambilan citra. Citra yang diperoleh sangat dipengaruhi oleh faktor-faktor seperti intensitas cahaya

ruangan, arah sumber cahaya, dan karakteristik sensor dan lensa kamera.

Metode deteksi wajah menjadi empat kategori [1], yaitu:

1. Metode berdasarkan pengetahuan. Metode ini kebanyakan digunakan untuk lokalisasi wajah.
2. Fitur pendekatan invarian. Metode ini kebanyakan digunakan untuk lokalisasi wajah.
3. Metode pencocokan template. Metode ini digunakan untuk lokalisasi wajah maupun deteksi wajah.
4. Metode berdasarkan penampilan. Metode ini kebanyakan digunakan untuk deteksi wajah.

#### C. Warna YCbCr

Warna yang diterima oleh mata dari sebuah objek ditentukan oleh warna sinar yang dipantulkan oleh objek tersebut. Sebagai contoh, suatu objek berwarna hijau karena objek tersebut memantulkan sinar biru dengan panjang gelombang 450 sampai 490 nanometer (nm). Warna sinar yang direspon oleh mata adalah sinar tampak (*visible spectrum*) dengan panjang gelombang berkisar dari 400 (biru) sampai 700 nm (merah). Warna-warna yang diterima oleh mata manusia merupakan hasil kombinasi cahaya dengan panjang gelombang berbeda. Penelitian memperlihatkan bahwa kombinasi warna yang memberikan rentang warna yang paling lebar adalah *red* (R), *green* (G), *blue* (B). Ketiga warna tersebut diberi nama warna pokok (primaries), dan sering disingkat sebagai warna dasar RGB. Warna-warna lain dapat diperoleh dengan mencampurkan ketiga warna pokok tersebut dengan perbandingan tertentu (meskipun sepenuhnya benar, karena tidak semua kemungkinan warna dapat dihasilkan dengan kombinasi RGB saja), sesuai dengan teori Young yang menyatakan bahwa sembarang warna dapat dihasilkan dari pencampuran warna-warna pokok C1, C2, C3 dengan presentasi tertentu :

$$C = a C_1 + b C_2 + c C_3 \quad (1)$$

Bila citra warna digitalisasi, maka tiga buah *filter* digunakan untuk mengekstraksi intensitas warna merah, hijau, dan biru, dan bila ketiganya dikombinasikan akan memperoleh persepsi warna. Selain RGB, warna juga dapat dimodelkan berdasarkan atribut warnanya. Setiap warna memiliki 3 buah atribut, yaitu *intensity* (I), *hue* (H), dan *saturation* (S).

Ruang warna YCbCr disebut juga ruang warna CCIR 601 (*International Radio Consultative Committee*). Model warna ini dikembangkan untuk mengantisipasi perkembangan informasi berbasis video, sehingga model ini banyak digunakan pada video digital. Secara umum dapat dikatakan bahwa model warna ini merupakan bagian dari ruang warna transmisi video dan televisi. Model warna lain yang mirip dengan YCbCr adalah YUV dan YIQ, perbedaannya terletak pada bahwa YCbCr adalah sistem warna digital sedangkan yang lainnya adalah sistem warna analog. Model warna YCbCr memisahkan nilai RGB menjadi informasi luminance dan chrominance yang berguna untuk aplikasi kompresi. Transformasi RGB ke YCbCr dilakukan dengan formulasi pada persamaan (2) sebagai berikut :

$$\begin{aligned} Y &= 0.299900R + 0.58700G + 0.11400B \\ C_B &= -0.16874R - 0.33126G + 0.50000B \\ C_R &= 0.50000R - 0.41869G - 0.08131B \end{aligned} \quad (2)$$

$$\begin{aligned} R &= 1.00000Y + 1.40200C_R \\ G &= 1.00000Y - 0.34414C_B - 0.71414C_R \\ B &= 1.00000Y + 1.77200C_B \end{aligned} \quad (3)$$

Formulasi konversi RGB-YCbCr dalam format lain ditunjukkan oleh sebagai berikut:

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} + \begin{bmatrix} 64.481 & -37.757 & 112 \\ 128.553 & -74.203 & -93.786 \\ 24.996 & 112 & -18.214 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (4)$$

Transformasi warna dari basis *CIE RGB* ke *CIE XYZ* dapat dilakukan sebagai berikut, diberikan triplet RGB ( $R_i, G_i, B_i$ ) untuk *pixel i*, maka triplet *XYZ* ( $X_i, Y_i, Z_i$ ) dihitung dengan

$$\begin{bmatrix} X_i \\ Y_i \\ Z_i \end{bmatrix} = \begin{bmatrix} 0.490 & 0.310 & 0.200 \\ 0.177 & 0.813 & 0.011 \\ 0.000 & 0.010 & 0.99 \end{bmatrix} \begin{bmatrix} R_i \\ G_i \\ B_i \end{bmatrix} \quad (5)$$

Transformasi sebaliknya dari *CIE XYZ* ke *CIE RGB* dapat dilakukan dengan persamaan

$$\begin{bmatrix} R_i \\ G_i \\ B_i \end{bmatrix} = \begin{bmatrix} 2.365 & -0.3896 & -0.468 \\ -0.515 & 1.425 & 0.088 \\ 0.005 & -0.014 & 1.009 \end{bmatrix} \begin{bmatrix} X_i \\ Y_i \\ Z_i \end{bmatrix} \quad (6)$$

Model warna *RGB* dan *XYZ* yang dikemukakan diatas adalah baku dari *CIE*. Ada juga model warna yang diusulkan untuk platform perangkat keras tertentu untuk menampilkan gambar. Misalnya *Nation Television Systems Committee (NTSC)* menggunakan model warna *RGB* khusus untuk menampilkan citra berwarna pada layar *CRT*. Format *NTSC* digunakan pada televisi di Amerika Serikat. Salah satu keuntungan utama dari format ini adalah data greyscale dipisahkan dari data warnanya, sehingga sinyal yang sama dapat digunakan baik untuk layar berwarna maupun layar hitam putih [5].

#### D. Dateksi Tepi

Tepi (*Edge*) adalah perubahan nilai intensitas derajat keabuan yang mendadak (besar) dalam jarak yang singkat. Terdapat tiga macam tepi di dalam citra digital, yaitu [6]:

##### a. Tepi Curam

Tepi curam adalah tepi dengan perubahan intensitas yang tajam. Arah tepi berkisar 90°.

##### b. Tepi Landai

Tepi landai yaitu tepi dengan sudut arah yang kecil. Tepi landai dapat dianggap terdiri dari sejumlah tepi-tepi lokal yang lokasinya berdekatan.

##### c. Tepi yang mengandung derau

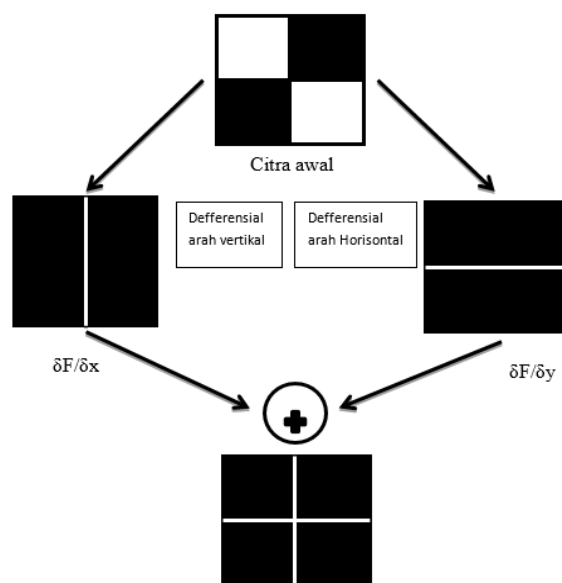
Umumnya tepi yang terdapat pada aplikasi visi komputer mengandung derau. Operasi peningkatan kualitas citra dapat dilakukan terlebih dahulu sebelum pendeteksian tepi.

Deteksi tepi (*Edge detection*) adalah operasi yang dijalankan untuk mendeteksi garistepi (*edges*) yang

membatasi dua wilayah citra homogen yang memiliki tingkatkecerahan yang berbeda. Deteksi tepi pada suatu citra adalah suatu proses yang menghasilkan tepi-tepi dari obyek-obyek citra, tujuannya adalah:

- Untuk menandai bagian yang menjadi detail citra
- Untuk memperbaiki detail dari citra yang kabur, yang terjadi karena error atau adanya efek dari proses akuisisi citra.
- Serta untuk mengubah citra 2D menjadi bentuk kurva

Suatu titik ( $x,y$ ) dikatakan sebagai tepi (*edge*) dari suatu citra bila titik tersebut mempunyai perbedaan yang tinggi dengan tetangganya. Gambar 1 berikut ini menggambarkan bagaimana tepi suatu gambar di peroleh.



Gambar 1. Gambar1. Proses Deteksi Tepi [7]

Berdasarkan prinsip-prinsip filter pada citra maka tepi suatu gambar dapat diperoleh menggunakan *High Pass Filter (HPF)*, yang mempunyai karakteristik

$$\sum_y \sum_x H(x, y) = 0 \quad (7)$$

Operator yang sering digunakan dalam proses deteksi tepi biasanya adalah operator Robert cross, operator prewitt, operator sobel dan operator laplace (Cahyo, 2010).

Dalam penelitian ini akan digunakan operator sobel untuk proses deteksi tepi. Proses yang digunakan oleh operator sobel merupakan proses dari sebuah konvolusi yang telah ditetapkan terhadap citra yang terdeteksi. Dalam operator sobel digunakan matrik konvolusi 3 X 3 dan susunan piksel-pikselnya di sekitar pixel ( $x, y$ ) seperti Gambar 2 berikut:

$$\begin{bmatrix} P1 & P2 & P3 \\ P8 & (x, y) & P4 \\ P7 & P6 & P5 \end{bmatrix}$$

Gambar 2. Susunan Pixel [7]

Operator sobel merupakan pengembangan Operator robert dengan menggunakan filter *High Pass Filter (HPF)*

yang diberi satu angka nol penyangga. Operator ini mengambil prinsip dari fungsi laplacian dan gaussian yang dikenal sebagai fungsi untuk membangkitkan HPF.

Kelebihan dari Operator sobel ini adalah kemampuan untuk mengurangi noise sebelum melakukan perhitungan deteksi tepi.

Sehingga besar gradient dapat di hitung dengan menggunakan persamaan:

$$\begin{aligned} S_x &= (p_3 + c_p4 + p_5) - (p_1 + c_p8 + p_7) \\ S_y &= (p_1 + c_p2 + p_3) - (p_7 + c_p6 + p_5) \\ G &= \sqrt{S_x^2 + S_y^2} \end{aligned} \quad (8)$$

Dengan nilai C konstanta bernilai dua, sehingga terbentuk matrik operator sobel pada Gambar 3.

$$S_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad S_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

Gambar 3. Matrik Operator Sobel [7]

Biasanya operator sobel menempatkan penekanan atau pembobotan pada piksel-piksel yang lebih dekat dengan titik pusat jendela, sehingga pengaruh piksel-piksel tetangga akan berbeda sesuai dengan letaknya terhadap titik di mana gradien dihitung. Dari susunan nilai-nilai pembobotan pada jendela juga terlihat bahwa perhitungan terhadap gradien juga merupakan gabungan dari posisi mendatar dan posisi vertikal [7].

#### E. Binerisasi

Citra biner adalah citra yang memiliki dua nilai tingkat keabuan yaitu hitam dan putih. Secara umum proses binerisasi citra *gray scale* untuk menghasilkan citra biner adalah sebagai berikut :

$$g(x,y) = \begin{cases} 1, & \text{jika } f(x,y) \geq T \\ 0, & \text{jika } f(x,y) < T \end{cases} \quad (9)$$

dengan  $g(x,y)$  adalah citra biner dari citra keabuan  $f(x,y)$ , dan T menyatakan nilai ambang.

Nilai T dapat ditentukan dengan salah satu dari 3 cara berikut [8]:

1. Nilai Ambang Global (*Global Threshold*)  
 $T = T\{f(x,y)\}$   
 Dengan T tergantung pada nilai *gray level* dari *pixel* pada posisi  $x,y$ .
2. Nilai Ambang Lokal (*Local Threshold*)  
 $T = T\{A(x,y), f(x,y)\}$   
 Dengan T tergantung pada properti *pixel* tetangga.  $A(x,y)$  menyatakan nilai *pixel* tetangga.
3. Nilai Ambang dinamis (*Dynamic Threshold*)  
 $T = T\{x, y, A(x, y), f(x, y)\}$   
 Dengan T tergantung pada koordinat-koordinat *pixel*.

Pada penelitian ini, nilai ambang didapat berdasarkan rentang nilai warna YCbCr yang menyatakan warna kulit. Adapun rentang nilai yang menyatakan warna tersebut merupakan warna kulit adalah sebagai berikut [9]:

$$Y > 80 \quad 85 \leq C_b \leq 135 \quad 135 \leq C_r \leq 180 \quad (10)$$

Berdasarkan nilai tersebut, jika nilai pixel memenuhi nilai ambang maka pixel tersebut bernilai satu sebaliknya akan bernilai nol.

#### F. Chamfer Matching

*Chamfer matching* pertama kali diperkenalkan oleh Barrow et al pada tahun 1977 dengan tujuan untuk mencocokkan dua model koleksi potongan garis. Sampai sekarang *Chamfer matching* dipergunakan secara luas dalam deteksi objek dan klasifikasi pekerjaan dikarenakan toleransinya pada ketidaksesuaian dalam posisi, skala dan perputaran. *Chamfer matching* merupakan teknik yang digunakan untuk menemukan kesesuaian yang terbaik antara titik sudut dari dua gambar yang berbeda, dengan cara memperkecil jarak yang umum diantara keduanya. Titik sudut dari suatu gambar diubah oleh sekumpulan persamaan *parametric transformation*, hal ini menjelaskan bagaimana gambar dapat menyimpang secara geometri dalam hubungan antara satu dan lainnya [10]. Versi asli dari algoritma *chamfer matching* digunakan hanya dalam jumlah yang terbatas pada beberapa aplikasi. Metode ini merupakan metode yang bagus, yang membutuhkan permulaan hipotesis yang bagus pada transformasi yang dapat menjadikan suatu tepian menjadi suatu kesesuaian [11].

Kemiripan diantara dua bentuk dapat diukur menggunakan *chamfer distance*. Diberikan dua kumpulan titik  $U = \{u_i\}_{i=1}^n$  dan  $V = \{v_j\}_{j=1}^m$ , fungsi *chamfer distance* merupakan rata-rata dari jarak antara masing-masing titik,  $u_i \in U$  dan titik terdekat pada  $V$

$$d_{cham}(U, V) = \frac{1}{n} \sum_{u_i \in U} \min_{v_j \in V} \|u_i - v_j\| \quad (11)$$

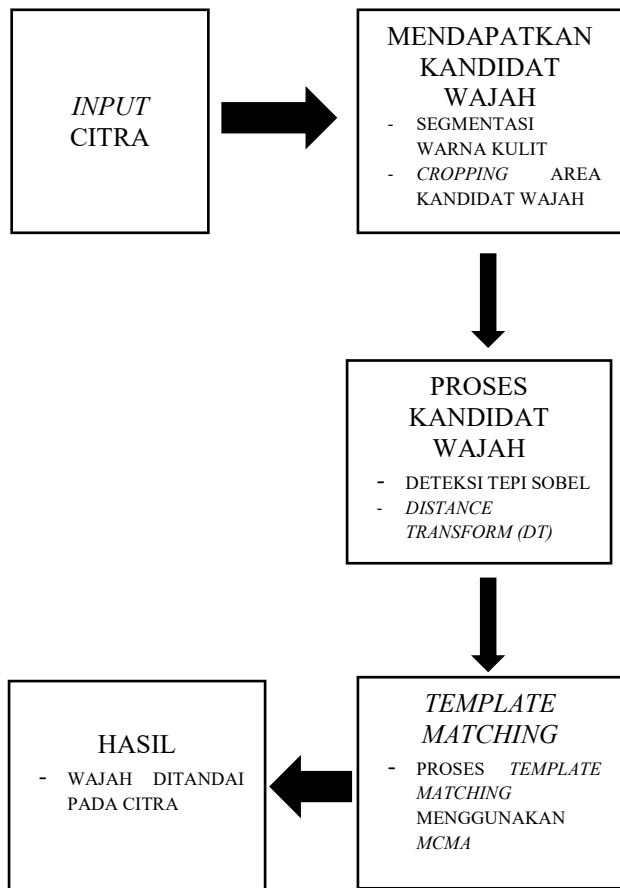
*Chamfer distance* yang simetris diperoleh dengan menambahkan  $d_{cham}(V, U)$ . *Chamfer distance* antara dua bentuk dapat dihitung dengan lebih efisien menggunakan *distance transform(DT)*. Transformasi ini mengambil ciri binari gambar sebagai masukan, dan menetapkan jarak masing-masing pixel pada gambar ke ciri terdekatnya. Jarak antara *template* dan peta tepi dapat dihitung sebagai rata-rata dari nilai DT pada titik kordinat *template* [12].

### III. METODOLOGI

#### A. Analisis Masalah

Permasalahan pada penelitian ini adalah bagaimana mendeteksi wajah pada suatu citra dengan memanfaatkan warna YCbCr dan *Modified Chamfer Matching Algorithm (MCMA)*. Pada subbab ini akan dibahas mengenai analisis penggunaan warna YCbCr untuk proses pendeteksian warna kulit manusia dan penerepan *MCMA* pada proses *template*

matching. Diagram sistem proses deteksi wajah yang dilakukan terdapat pada Gambar 4.



Gambar 4. Diagram Sistem Proses Deteksi Wajah

### 1) Analisis warna wajah

Setelah *user* melakukan pemilihan citra yang akan diuji, citra *input* tersebut pertama kali akan melalui proses segmentasi warna kulit. Proses segmentasi warna kulit akan dilakukan menggunakan warna YCbCr dimana citra *input* akan dikonversi terlebih dahulu dari warna RGB menjadi warna YCbCr menggunakan persamaan konversi RGB ke YCbCr.

Proses konversi warna dari RGB ke warna YCbCr akan menggunakan persamaan (2). Untuk melakukan konversi pertama kali citra akan ditampung kedalam kelas *BufferedImage* pada java dan kemudian akan digunakan kelas *Color* untuk mendapatkan nilai RGB dari citra tersebut. Nilai RGB ini terdiri dari *red*, *green* dan *blue*. Nilai ini kemudian akan digunakan pada persamaan (2) untuk mendapatkan warna YCbCr.

Proses konversi warna RGB ke YCbCr dapat dilakukan seperti gambar III-2 berikut. Diketahui nilai *Red (R)* adalah 3, nilai *Green (G)* adalah 2 dan nilai *Blue (B)* adalah 3, maka nilai YCbCr dari piksel tersebut adalah (Gambar 6):

$$\begin{aligned}
 Y &= 0.299900R + 0.58700G + 0.11400B \\
 &= 0.299900*3 + 0.58700*2 + 0.11400*3 \\
 &= 0.8997 + 1.174 + 0.342 \\
 &= 2.4157 \\
 Cb &= -0.16874R - 0.33126G + 0.50000B
 \end{aligned}$$

$$\begin{aligned}
 &= -0.16874*3 - 0.33126*2 + 0.50000*3 \\
 &= -0.50622 - 0.66252 + 1.5 \\
 &= 0.33126
 \end{aligned}$$

$$\begin{aligned}
 Cr &= 0.50000R - 0.41869G - 0.08131B \\
 &= 0.50000*3 - 0.41869*2 - 0.08131*3 \\
 &= 1.5 - 0.83738 - 0.24393 \\
 &= 0.41869
 \end{aligned}$$

Gambar 5. Konversi Warna RGB ke YCbCr

Berdasarkan perhitungan pada gambar III-2 telah didapat warna Y adalah 2.4157, Cb adalah 0.33126 dan Cr adalah 0.41869. Rentang warna pada suatu piksel adalah 0 – 255 maka akan dilakukan pembulatan pada warna diatas sehingga warna Y menjadi 2, Cb menjadi 0 dan Cr menjadi 0. Selanjutnya warna Y, Cb dan Cr ini lah yang menjadi warna pada piksel.

### 2) Analisis Kandidat Wajah

Setelah melalui proses konversi warna dari RGB ke YCbCr maka selanjutnya akan dilakukan proses untuk mendapatkan kandidat wajah. Objek pada citra akan dikelompokkan menjadi dua yaitu kulit dan non kulit. Untuk menentukan objek berupa kulit dapat digunakan rentang nilai warna YCbCr yang menyatakan suatu warna dapat digolongkan sebagai kulit. Adapun rentang nilai yang digunakan untuk menentukan warna tergolong warna kulit terdapat pada persamaan (10).

Proses perbandingan dapat dilakukan melalui langkah berikut, pertama – tama nilai YCbCr pada piksel yang akan diperiksa ditampung terlebih dahulu pada suatu variabel. Selanjutnya nilai Y, Cb dan Cr tersebut masing – masing akan dibandingkan dengan rentang nilai kulit yaitu untuk Y lebih besar dari 80, Cb terletak diantara 85 dan 135, Cr terletak diantara 135 dan 180. Jika nilai Y, Cb dan Cr memenuhi persyaratan tersebut maka warna pada piksel tersebut akan diubah menjadi warna putih, sebaliknya jika warna pada piksel tidak memenuhi persyaratan tersebut maka warna pada piksel tersebut akan diubah menjadi hitam.

Selanjutnya akan dilakukan proses *cropping* untuk mendapatkan kandidat wajah. Citra terlebih dahulu akan diproses menggunakan metode *connected component labelling (ccl)* untuk menentukan area - area yang memiliki piksel putih. Penentuan area dilakukan *ccl* dengan memberikan label pada area-area yang memiliki piksel putih. Selanjutnya area – area tersebut akan dihitung luas areanya, luas area didapat dengan menghitung berapa banyak piksel putih yang terdapat pada area tersebut, jika area mengandung lebih dari 100 piksel maka area tersebut akan dijadikan kandidat wajah. Area yang dijadikan kandidat wajah akan dilakukan proses *cropping*, dimana area yang menjadi kandidat wajah akan ditampung ke dalam kelas *BufferedImage* pada java. Setelah kandidat wajah tersebut ditampung selanjutnya kandidat wajah tersebut akan diubah ukurannya menjadi 80 x 119 piksel sesuai dengan ukuran dari *template* wajah.

Pada tahap ini kandidat wajah telah didapat, sehingga proses selanjutnya dapat dilakukan.

## B. Analisis Proses Kandidat Wajah

### 1) Analisis Deteksi Tepi Sobel

Setelah kandidat wajah didapat maka langkah selanjutnya adalah melakukan proses deteksi tepi terhadap kandidat wajah. Proses yang dilakukan pertama kali adalah mengambil nilai piksel dari kandidat wajah tersebut, selanjutnya nilai piksel akan digunakan untuk mencari nilai  $S_x$  dan  $S_y$  pada persamaan sobel. Setelah didapat nilai  $S_x$  dan  $S_y$ , nilai tersebut akan digunakan untuk mencari nilai gradien.

Setelah nilai gradien didapat, nilai tersebut akan diperiksa kembali jika nilai kurang dari 0 maka nilai akan dijadikan 0 dan jika nilai lebih dari 255 maka nilai akan dijadikan 255. Perubahan nilai tersebut dilakukan karena rentang nilai dari suatu warna pada piksel adalah 0 – 255. Nilai gradien selanjutnya akan digunakan sebagai nilai pada piksel yang diproses dan deteksi tepi telah selesai dilakukan.

### 2) Analisis Distance Transform (DT)

Setelah dilakukan proses deteksi tepi tahap selanjutnya adalah melakukan proses *distance transform* ( $dt$ ). Untuk menjadikan kandidat wajah menjadi *distance transform*, kandidat wajah pertama kali akan ditampung kedalam kelas *BufferedImage* pada java dan kemudian akan digunakan kelas *Color* untuk mendapatkan nilai piksel dari kandidat wajah tersebut. Setelah nilai piksel diperoleh maka selanjutnya nilai tersebut akan ditampung kedalam *array*. *Array* yang menampung nilai dari piksel ini selanjutnya akan diproses dan dicari nilai *distance transform* menggunakan persamaan (12) dan (13) pada bab 2. Setelah *array* tersebut diproses dan *distance transform* didapat, nilai tersebut akan digunakan sebagai nilai piksel pada kandidat wajah dan kandidat wajah sekarang telah berada dalam bentuk *distance transform*.

## C. Analisis Template Matching

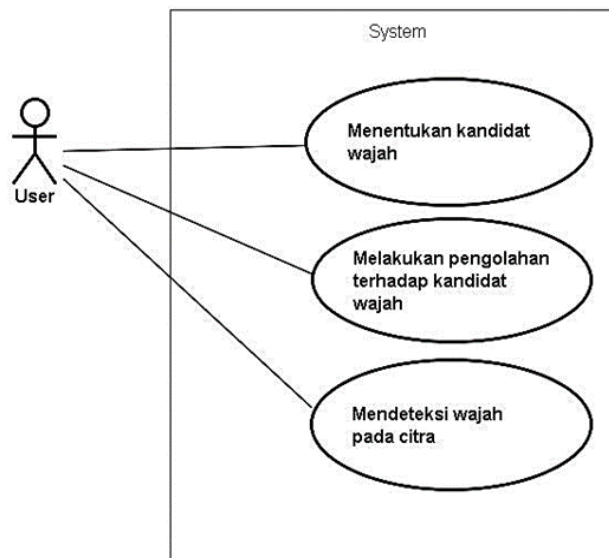
Pada tahap ini akan dilakukan proses terakhir yaitu proses *template matching* antara kandidat wajah dengan *template* wajah. Proses ini bertujuan untuk menentukan apakah kandidat wajah merupakan wajah atau bukan, jika hasil dari *template matching* sesuai maka kandidat wajah akan dinyatakan sebagai wajah dan akan diberi tanda dengan kotak warna merah. Kandidat wajah akan dinyatakan sebagai wajah jika hasil dari perhitungan jarak antara kandidat wajah dengan *template* wajah kurang dari 2 %. Untuk melakukan proses perhitungan jarak akan digunakan persamaan (11) pada bab 2, persamaan tersebut merupakan persamaan untuk menghitung *chamfer distance* yang digunakan pada metode *chamfer matching*.

Proses yang dilakukan pertama kali adalah menampung masing – masing nilai piksel dari kandidat wajah dan *template* wajah pada suatu *array*. Selanjutnya nilai ini akan digunakan untuk perhitungan pada persamaan (11) tersebut dimana *array* kandidat wajah akan mewakili *variable*  $u$  dan *array template* wajah akan mewakili *variable*  $v$ . Setelah hasil perhitungan didapat maka selanjutnya hasil perhitungan akan diperiksa, jika hasil yang didapat kurang dari 2 % maka kandidat tersebut akan dinyatakan sebagai wajah dan

selanjutnya akan dibuat kotak warna merah pada posisi kandidat wajah didalam citra yang diujikan.

## D. Analisis Perangkat Lunak

Perangkat lunak yang dikembangkan dalam penelitian ini adalah perangkat lunak yang digunakan untuk mendeteksi wajah manusia, yang menggunakan segmentasi warna kulit untuk mendapatkan kandidat wajah dan untuk proses penentuan wajah akan digunakan metode *template matching* menggunakan *Modified Chamfer Matching Algorithm (MCMA)*. Perangkat lunak ini menerima masukan berupa citra RGB yang memiliki format JPEG. *Functionality system* yang dikembangkan terdapat pada Gambar 6.



Gambar 6. *Functionality system*

## IV. HASIL DAN PEMBAHASAN

Aplikasi yang di hasilkan dilakukan pengujian untuk melihat kesesuaian perancangan perangkat lunak dengan implementasi perangkat lunak maka dapat disimpulkan bahwa unit dan antar muka yang diujikan berjalan dengan baik.

Pengujian pada Gambar 7 menunjukkan bahwa kandidat wajah telah didapat, kandidat – kandidat tersebut ditandai oleh kotak merah. Kandidat – kandidat tersebut lah kemudian yang akan diproses dan digunakan untuk proses *template matching* untuk diidentifikasi apakah kandidat tersebut wajah atau bukan.

Pengujian pada Gambar 8 menunjukkan tidak ada kandidat wajah yang didapat, hal ini dapat dilihat pada citra yang tidak adanya kotak merah yang menandai kandidat wajah. Kandidat wajah tidak didapat dikarenakan citra berupa *greyscale*.

Pengujian pada Gambar 9 menunjukkan bahwa proses pengolahan kandidat wajah telah selesai dilakukan, hal tersebut ditunjukkan oleh citra kandidat wajah yang telah

berupa *distance transform*. Citra tersebut selanjutnya akan digunakan untuk proses *template matching*.



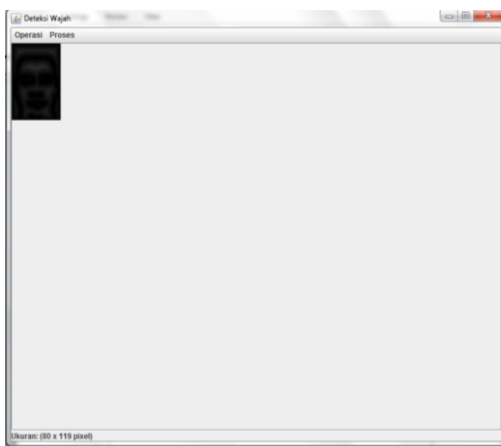
Gambar 7. Pengujian Menentukan Kandidat Wajah



Gambar 10. Pengujian Mendeteksi Wajah Pada Citra



Gambar 8. Pengujian Menentukan Kandidat Wajah (*Greyscale*)



Gambar 9. Pengujian Melakukan Pengolahan Terhadap Kandidat Wajah

Pengujian pada Gambar 10 menunjukkan bahwa proses pendeteksian wajah telah selesai. Wajah yang terdapat pada citra telah tandai oleh kotak merah, hal ini menunjukkan bahwa hasil *template matching* antara kandidat wajah dengan *template* wajah sesuai, tetapi masih terdapat objek yang bukan wajah ditandai oleh kotak merah, hal ini menunjukkan bahwa masih terdapat kesalahan dalam *matching* antara *template* wajah dengan kandidat wajah, hasil tersebut tidak sesuai dengan keluaran yang diharapkan.

Tabel 1 Hasil Pendeteksian Wajah

No	Nama File	Hasil Pendeteksian Wajah			
		Terdeteksi seluruh wajah	Terdapat yang tidak terdeteksi	Terdeteksi selain objek wajah	Tidak terdeteksi
1	S1	✓			
2	S2	✓			
3	S3	✓			
4	S4	✓		✓	
5	S5	✓		✓	
6	S6	✓			
7	S7	✓		✓	
8	S8	✓			
9	S9	✓			
10	S10	✓		✓	
11	S11		✓	✓	
12	S12		✓	✓	
13	S13	✓		✓	
14	S14		✓	✓	
15	S15		✓	✓	
16	S16		✓	✓	
17	S17		✓	✓	
18	S18		✓	✓	
19	S19		✓	✓	
20	S20		✓	✓	
21	S21	✓			
22	S22	✓		✓	
23	S23		✓	✓	
24	S24		✓	✓	
25	S25			✓	✓
26	S26			✓	✓
27	S27	✓		✓	
28	S28		✓	✓	
29	S29	✓			
30	S30	✓			

Hasil pengujian yang telah dilakukan di tampilkan pada Tabell. Pengujian dilakukan terhadap 4 faktor yaitu terdeteksi seluruh wajah, terdapat bagian wajah yang tidak terdeteksi, terdeteksi objek selain wajah, wajah tidak terdeteksi. Hasil pendeteksian diatas dapat dihitung dengan persamaan 4.1 sebagai berikut :

$$\text{Persentase (\%)} = \frac{\text{Jumlah citra kategori}}{\text{Jumlah seluruh citra uji}} \times 100\% \quad (12)$$

Setelah melakukan perhitungan dengan Persamaan 12 diatas maka diperoleh hasil sebagai berikut :

1. Citra uji yang berhasil terdeteksi seluruh wajah dan tidak terdapat objek lain selain wajah yang terdeteksi sebesar 30 %
2. Citra uji yang berhasil terdeteksi seluruh wajah dan terdapat objek lain selain wajah yang terdeteksi sebesar 23,33 %
3. Citra uji yang terdapat wajah yang tidak terdeteksi dan terdapat objek lain selain wajah yang terdeteksi sebesar 40 %
4. Citra uji yang tidak berhasil terdeteksi wajah sebesar 6,66 %

## V. KESIMPULAN

Kesimpulan yang diperoleh pada penelitian ini adalah sebagai berikut :

1. Warna YCbCr dapat digunakan untuk segmentasi warna kulit dengan memanfaatkan rentang nilai warna kulit pada YCbCr yaitu  $Y > 80, 85 \leq C_b \leq 135$  dan  $135 \leq C_r \leq 180$ .
2. *Modified Chamfer Matching Algorithm* dapat digunakan sebagai algoritma pada proses *template matching*.
3. Keakuratan dari *Modified Chamfer Matching Algorithm* dalam menentukan wajah adalah 30 % untuk terdeteksi keseluruhan wajah, 23,33 % untuk terdeteksi keseluruhan wajah tapi masih terdapat objek lain yang terdeteksi, 40 % untuk masih terdapat beberapa wajah yang tidak terdeteksi dan 6,66 % untuk wajah yang tidak terdeteksi sama sekali.

4. *Modified Chamfer Matching Algorithm* masih banyak memiliki kesalahan dalam pencocokan antara *template wajah* dengan kandidat wajah.
5. Perangkat lunak ini mampu digunakan untuk mendeteksi wajah pada citra dengan kondisi citra yang diujikan bebas dari *noise*.
6. Kinerja dari perangkat lunak bergantung pada banyaknya lingkungan dan besar citra yang diuji, semakin banyak lingkungan dan semakin besar citra maka semakin lama waktu yang dibutuhkan untuk proses pendeteksian.
7. *Noise* pada wajah akan mempengaruhi hasil *template matching*.

## DAFTAR PUSTAKA

- [1] M. H. Yang, D. Kriegman, and N. Ahuja, "Detecting faces in images: A survey. IEEE Transactions on Pattern Analysis & Machine Intelligence," 2002.
- [2] S. Saparudin and E. Rasywir, "Pengenalan potensi anak melalui sidik jari Menggunakan algoritma voting feature intervals 5 (vfi5)," *J. Res. Comput. Sci. Appl.*, vol. 1, no. 1, 2012.
- [3] K.-K. Sung, "Learning and example selection for object and pattern recognition," *PhD Thesis, MIT AI Lab*, 1996.
- [4] E. Hjelmas and B. K. Low, "Face detection: A survey," *Comput. Vis. image Underst.*, vol. 83, no. 3, pp. 236–274, 2001.
- [5] E. A. Murinto and W. Nurhidayati, "Deteksi Jenis Warna Kulit Wajah Untuk Klasifikasi Ras Manusia Menggunakan Transformasi Warna," *Univ. Ahmad Dahlan, Yogyakarta*, 2008.
- [6] B. Niam, A. Hidayatno, and R. R. Isnanto, "Analisis Deteksi Tepi Pada Citra Berdasarkan Perbaikan Kualitas Citra," *Jurusan Teknik Elektro Fakultas Teknik*, 2011.
- [7] S. D. Cahyo, "Analisis Perbandingan Beberapa Metode Deteksi Tepi Menggunakan Delphi 7," *Skripsi Progr. Stud. Tek. Inform.*, 2009.
- [8] D. Putra, "Binerisasi citra tangan dengan metode otsu," *Tekno. Elektro*, vol. 3, no. 2, pp. 11–13, 2004.
- [9] G. Kukharev and A. Nowosielski, "Visitor identification-elaborating real time face recognition system," 2004.
- [10] A. Ghafoor, R. N. Iqbal, and S. Khan, "Image matching using distance transform," in *Scandinavian Conference on Image Analysis*, 2003, pp. 654–660.
- [11] G. Borgefors, "Hierarchical chamfer matching: A parametric edge matching algorithm," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 10, no. 6, pp. 849–865, 1988.
- [12] A. Thayananthan, B. Stenger, P. H. S. Torr, and R. Cipolla, "Shape context and chamfer matching in cluttered scenes," in *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings.*, 2003, vol. 1, pp. I–I.