

# SISTEM PENGENALAN WAJAH MENGGUNAKAN METODE *TEMPLATE MATCHING*

<sup>1</sup>Yunifa Miftachul Arif, <sup>2</sup>Achmad Sabar

<sup>1</sup>Jurusan Teknik Informatika, Fakultas Saintek, UIN Maulana Malik Ibrahim Malang

<sup>2</sup>Jurusan Sistem Komputer, STMIK Asia Malang

**Abstrak-***Dewasa ini perhatian peneliti dalam memanfaatkan teknologi biometrik pada kehidupan untuk mengidentifikasi dan mengenal karakteristik manusia sudah banyak. Teknologi ini mengidentifikasi bagian tubuh manusia yang unik dan tetap seperti sidik jari, mata dan wajah manusia. Hal khusus di bidang identifikasi dan pengenalan wajah manusia memanfaatkan pengolahan dan analisis citra wajah, seperti menentukan daerah komponen wajah manusia dan karakteristiknya, yang akan membentuk suatu semantik wajah yang membantu mengungkapkan bagaimana komponen individu berperan dalam pengenalan wajah. Pada penelitian ini dikembangkan sistem yang memisahkan citra wajah ke dalam komponen wajah, kemudian mengekstraksinya ke dalam fitur mata dan batas wajah pada citra diam tunggal yang diambil dari posisi tampak depan. Antara tiap komponen diukur jaraknya, kemudian dikombinasikan dengan fitur lainnya untuk membentuk semantik wajah. Melalui tahapan deteksi wajah, berdasarkan model warna kulit dan normalisasi daerah wajah serta ekstraksi fitur mata, maka jarak masing-masing fitur dapat ditentukan.*

**Kata kunci:** *Template Matching, Face Recognition, Pengenalan Wajah*

## 1. PENDAHULUAN

Pengenalan wajah manusia merupakan salah satu bidang penelitian yang penting. Penelitian tersebut telah banyak dilakukan dengan kelebihan dan kekurangan tertentu. Dari sebuah wajah, banyak informasi yang didapat baik secara statis maupun dinamis, misalnya saja warna kulit, struktur tulang wajah, dan ekspresi wajah.

Dengan perangkat lunak *face recognition*, penegak hukum dapat mencari dan mengidentifikasi wajah seorang kriminal. Jika semula polisi hanya berpedoman pada ilustrasi gambar wajah dan sidik jari penjahat secara manual, kini teknologi komputer dapat melakukan tugas tersebut dengan lebih cepat dan akurat, serta mendapatkan informasi tentang orang tersebut.

Tujuan utama penelitian ini adalah untuk merancang aplikasi yang dapat mengenali, mencari, dan membandingkan wajah seseorang dengan inputan foto wajah. Sehingga tahapan permasalahan yang perlu diselesaikan adalah sebagai berikut:

1. Deteksi warna kulit atau *skin detection*
2. *Face Location*, pencarian lokasi wajah

3. Pencarian lokasi fitur dari wajah
4. Ekstraksi ciri fitur dari wajah
5. Perbandingan fitur foto yang dicari dengan fitur foto pada database

## 2. TINJAUAN PUSTAKA

### Deteksi Warna Kulit

Pendeteksian wajah serta menentukan lokasi wajah secara otomatis adalah salah satu masalah yang sangat kompleks dan masih menjadi bahan penelitian hingga saat ini. Hal ini dikarenakan banyaknya aplikasi yang menggunakan sistem berbasis *face detection*. Misalnya untuk aplikasi pengidentifikasi seseorang pada alat-alat keamanan, aplikasi pengenalan jenis kelamin, dan pengidentifikasi ekspresi wajah. Semua aplikasi tersebut bertujuan sama agar hubungan antara mesin dan manusia bisa berjalan lebih baik.

Pada tahun-tahun terakhir, penelitian terhadap pendeteksian wajah menjadi lebih kompleks lagi. Hal ini dikarenakan penelitian dilakukan berdasarkan warna kulit. Warna sangat cocok untuk pendeteksian wajah dalam sebuah gambar yang kompleks karena proses segmentasi pada gambar

Created with

berwarna lebih cepat perhitungannya dan lebih presisi, terutama untuk mengetahui *iluminasi*, *shading*, dan *background* yang kompleks dibandingkan dengan proses segmentasi pada gambar *grayscale*. Kepresisian yang tinggi dapat dicapai apabila dalam pengambilan *colorspace* ada pemisahan antara *chrominance* dan *luminance* dari gambar aslinya dan penentuan distribusi *chrominance* contoh warna kulit manusia sebagai nilai *threshold* diperhitungkan secara tepat.

*Colorspace* yang bisa digunakan sangat beraneka ragam dan tentunya dengan kekurangan dan kelebihan masing-masing. Berikut ini adalah macam-macam *colorspace* yang bisa diterapkan untuk memodelkan warna kulit dengan memisahkan faktor iluminasi: HSV(atau HIS), YIQ, YCrCb yang ampuh untuk identifikasi kulit orang-orang asia. Pemilihan *chrominance space* yang tepat harus menjadi perhatian utama bila menginginkan proses pengidentifikasian warna kulit berjalan sempurna, karena distribusi dari *chrominance* kulit bergantung pada *chrominance space*.

Pemodelan warna kulit dibuat dengan cara memadukan berbagai macam warna dan jenis kulit manusia yang sudah ada dalam database. Biasanya, para ahli mempunyai gambar-gambar wajah dari minimum 40 orang model. Gambar-gambar tersebut dikumpulkan dari berbagai macam ras, suku, umur, dan jenis kelamin dan tentu saja dengan berbagai macam situasi pencahayaan. Pandangan umum yang harus diperhatikan benar-benar adalah bahwa setiap orang mempunyai warna kulit yang berbeda antara satu dengan yang lainnya meskipun masih tergolong dalam satu ras yang sama (Ujang Fajar Awaludin, 2006: 19).

Deteksi tepi

Deteksi tepi adalah proses untuk menemukan perubahan intensitas yang berbeda nyata dalam sebuah bidang

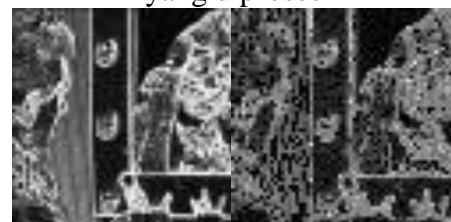
citra. Sebuah operator deteksi tepi merupakan operasi bertetangga, yaitu sebuah operasi yang memodifikasi nilai keabuan sebuah titik berdasarkan nilai-nilai keabuan dari titik-titik yang ada di sekitarnya (tetangganya) yang masing-masing mempunyai bobot tersendiri. Bobot-bobot itu sendiri nilainya tergantung pada operasi yang akan dilakukan, sedangkan banyaknya titik tetangga yang terlibat biasanya 2x2, 3x3, 3x4, 7x7, dan sebagainya.

Biasanya operator yang digunakan untuk mendeteksi tepi yang pertama adalah operator berbasis gradient (turunan pertama), yaitu operator robert, operator sobel, dan operator prewitt. Yang kedua adalah operator berbasis turunan kedua, yaitu operator laplician dan operator laplacian Gaussian. (T. Sutoyo dkk, 2009: 227)

Dalam penelitian ini digunakan operator sobel yang memerlukan proses perhitungan yang lebih lama dibandingkan dengan operator *Robert Cross*. Akan tetapi, bila nantinya dalam proses digunakan kernel berukuran besar, maka akan memperhalus gambar input sehingga nantinya akan mereduksi *noise*. Keunggulan lain dari operator Sobel dibandingkan dengan *Robert Cross* adalah nilai-nilai output dari proses konvolusi dengan Sobel lebih akurat dan presisi (Ujang Fajar Awaludin, 2006: 21). Hal ini dapat dibuktikan pada gambar dibawah ini:



**Gambar 1** Salah satu input gambar yang diproses



Created with

**Gambar 2** Gambar kiri adalah hasil dari operator Sobel, sebelah kanan menggunakan operator *Robert Cross*

### *Integral Projection*

*Integral projection* adalah suatu metode yang digunakan untuk mencari daerah atau lokasi dari objek. Metode ini dapat digunakan untuk mendeteksi batas dari daerah gambar yang berbeda, sehingga kita bisa mencari daerah lokasi wajah dan fitur-fiturnya. Metode ini juga bisa disebut dengan integral baris dan kolom dari piksel, karena integral ini menjumlahkan piksel per baris dan piksel per kolom. Dari metode ini kita akan dengan mudah untuk menemukan daerah lokasi objek yang kita perlukan.

Dari metode ini akan kita dapatkan hasil penjumlahan baris dan kolom yang nantinya akan kita olah atau proses lebih lanjut. Metode ini akan memudahkan kita saat kita akan mencari lokasi wajah, sebelum kita gunakan metode ini, kita cari nilai clusternya kemudian setelah kita dapatkan, kita jumlahkan tiap pikselnya. Tiap piksel citra atau gambar yang kita proses memiliki nilai cluster, jadi dalam satu gambar semua piksel memiliki nilai cluster dengan intensitas cluster yang berbeda, jika suatu daerah memiliki kemiripan warna dengan nilai cluster, berarti pikselnya memiliki nilai yang tinggi dan jika kita jumlahkan maka daerah ini akan memiliki jumlah yang lebih besar dari pada daerah lain yang hanya memiliki nilai yang kecil (intensitas kecil). Kita jumlahkan piksel ke arah sumbu  $y$  selebar  $x$  dan kita jumlah piksel  $x$  sepanjang atau setinggi  $y$ , jadi kita akan mengetahui tinggi dan lebar dari image yang memiliki nilai tinggi, dari sini kita sudah mendapatkan lokasi dari objek yang kita cari. Untuk mendapatkan lokasi yang lebih tepat maka kita akan memproses lebih lanjut. Jadi untuk menjumlahkan piksel baris dan kolom tergantung pada kita, parameter apa yang kita gunakan. Jika

kita gunakan parameter kulit maka kita asumsikan semua piksel gambar memiliki nilai cluster, tapi nilainya tergantung pada intensitas tertentu. Jika suatu daerah piksel memiliki warna yang sesuai dengan parameter yang kita gunakan (misal: kulit), maka daerah tersebut memiliki intensitas yang tinggi, jika kita jumlahkan maka pada daerah ini nilainya akan tinggi baik dijumlahkan ke arah  $x$  atau  $y$ .

### *Template Matching*

Beberapa objek memiliki karakteristik yang menampilkan rata-rata lebar dari tampilan sesuai petunjuk dan kondisi. Sedangkan *template matching* merupakan sebuah strategi pengenalan objek yang menemukan objek dengan mencocokkan gambar bagian-bagian kecil dengan contoh *template*.

Aplikasi alamiah dari *template matching* digunakan untuk membangun gambar *template* yang diminta untuk membagi kategori semantik. *Template* ini dapat dibangun *on-line*, dan digunakan untuk menyederhanakan *query* dengan menggunakan suatu *template* yang ada dan bukan menyusun suatu *query*. Menemukan wajah merupakan bagian *template matching* dengan tampilan frontal dari wajah, kemudian mencari bagian yang gelap yang khas dari mata, hidung dan mulut. Jika wajah dapat ditemukan, maka tidak terikat pada identitas dari seseorang tertentu yakni hanya dengan mencari polanya.

Fungsi *MatchTemplate* mengimplementasikan satu bagian metode untuk menemukan bagian gambar yang sama pada *template* yang diberikan.

Pemberian gambar dengan piksel  $W \times H$  dan *template* dengan piksel  $w \times h$ , dihasilkan gambar dengan piksel  $Ww+1 \times Hh+1$ , dan nilai piksel pada tiap lokasi  $(x,y)$  yang memiliki karakteristik kesamaan antara *template* dan gambar

dengan sudut atas kiri pada (x,y) dan sudut kanan bawah pada (x+w-1,y+h-1).

Persamaannya dapat dirumuskan sebagai berikut :

$$R(x, y) = \sum_{y'=0}^{h-1} \sum_{x'=0}^{w-1} T'(x', y') I'(x + x', y + y')$$

dimana  $T=template$  dan  $I=image$  sedangkan  $T'$  dan  $I'$  merupakan nilai masing-masing piksel dikurangi dengan rata-rata pada satu blok

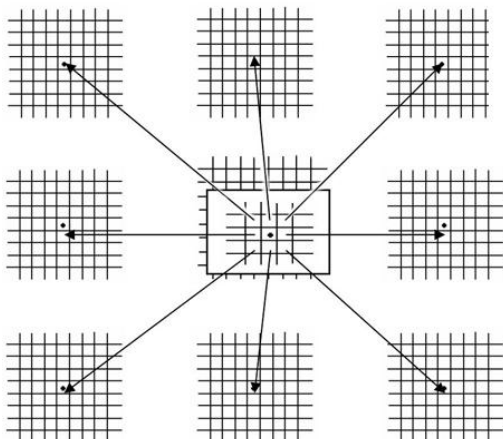
$$T'(x', y') = T(x', y') - \bar{T}$$

$$I'(x + x', y + y') = I(x + x', y + y') - \bar{I}(x, y)$$

Untuk memperoleh nilai korelasinya , maka digunakan rumus seperti dibawah ini :

$$Corr(x, y) = \frac{R(x, y)}{\sqrt{\sum_{y'=0}^{h-1} \sum_{x'=0}^{w-1} T'(x', y')^2 \sum_{y'=0}^{h-1} \sum_{x'=0}^{w-1} I'(x + x', y + y')^2}}$$

Untuk menentukan kandidat titik dari tracking sebelumnya diperoleh dari hasil korelasi tiap blok yang dimana terdiri dari 9 blok dengan mengambil nilai korelasi yang paling besar. Dibawah adalah gambar ilustrasi proses daerah yang dipakai untuk mencari nilai korelasinya, dimana sebuah koordinat yang mengambil piksel tetangganya untuk memperoleh posisi yang dimaksudkan.



**Gambar 3** Daerah piksel yang dihitung korelasinya

Dari hasil perhitungan 9 bloks data dengan informasi tiap data terdiri dari piksel 7x7 diperoleh nilai korelasi yang lebih besar dengan nilai *template* yang telah ada , kemudian dipilih untuk dijadikan kandidat *template* pada

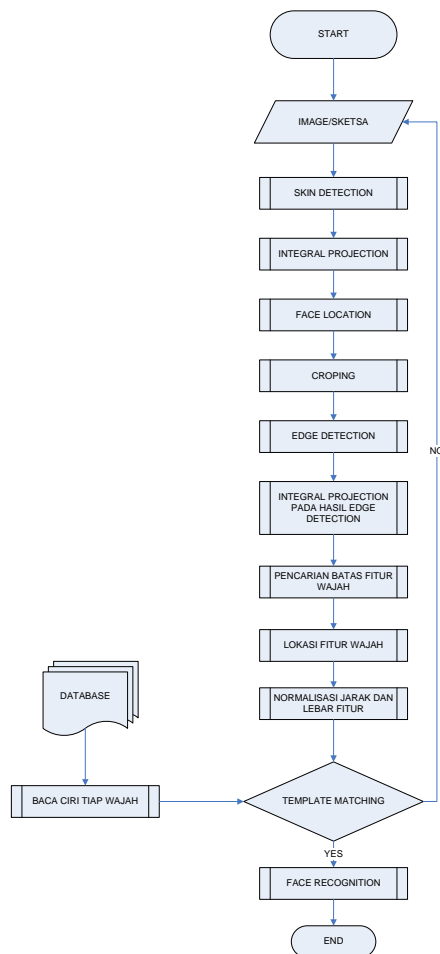
pencarian korelasi maksimum dengan ukuran piksel 3x3, selanjutnya nilai akhirnya merupakan nilai kandidat posisi yang dicari. (Ujang Fajar Awaludin, 2006: 25)

### 3. RANCANGAN SISTEM DAN HASIL PENELITIAN

Secara umum sistem pengenalan wajah ditunjukkan pada gambar 3.1. Gambar diambil melalui sebuah kamera dengan posisi wajah frontal menghadap kamera. Untuk mendeteksi wajah digunakan permodelan warna kulit dan integral projection untuk menentukan dimana letak wajah. Setelah diketahui dengan pasti posisi dari wajah, langkah selanjutnya adalah mencari fitur-fitur yang terdapat didalam wajah. Dari masing-masing fitur yang ada ditentukan titik-titik fiturnya.

Titik fitur digunakan pada membandingkan antara foto inputan dengan foto pada database. *Template matching* adalah salah satu metode yang dapat dipakai pada saat proses membandingkan titik fitur.

*Flowchart system* dari penelitian ini adalah sebagai berikut:



**Gambar 4** Flowchart system *Skin Detection*

Pendeteksian warna kulit (*Skin Color Detection*) adalah tahap dimana kita harus mencari daerah yang memiliki warna kulit. Untuk mendapatkan daerah ini kita harus melakukan proses untuk mencari daerah warna kulit dan memproses sample untuk warna kulit. Untuk pengolahan warna kulit ini dilakukan dengan melalui beberapa tahap. Dalam skin detection ini kami menggunakan metode YCrCb untuk pemodelan warna kulit, karena model warna YCrCb dinilai cukup stabil untuk mereduksi efek lighting dari suatu gambar.

### YcrCb Color Space

Gambar yang diperoleh dari kamera memiliki struktur warna menggunakan sistem Red-Green-Blue (RGB). Untuk memodelkan warna kulit

sistem warna RGB ini ditransformasikan ke sistem warna YCrCb untuk memisahkan intensitas Y dengan chromaticity yang dinyatakan dalam dua variabel Cr dan Cb. Harga Cr membedakan warna antara jarak intensitas terhadap unsur warna merah sedangkan Cb menyatakan jarak intensitas terhadap unsure warna merah. Dalam memodelkan warna kulit hanya informasi Cr dan Cb yang dipakai, sehingga pengaruh perubahan intensitas dapat dihilangkan. Pada daerah saturasi dari cahaya yang tertangkap kamera, harga Cr dan Cb sangat stabil, sehingga nilai Cr dan Cb merupakan informasi handal untuk proses klasifikasi warna. Untuk mengubah citra berwarna yang mempunyai nilai matrik masing-masing r, g dan b menjadi citra warna YCrCb dapat dilakukan dengan menghitung seperti pada persamaan di bawah ini:

$$Y = 0.59G + 0.31R + 0.11B$$

$$Cr = 0.713 * (R - Y)$$

$$Cb = 0.564 * (B - Y)$$

### Pengambilan sample Warna Kulit

Tahap kedua dalam *skin detection* adalah pengambilan sample warna kulit dan yang bukan warna kulit. Rumus persamaan warna kulit yang digunakan dalam penelitian ini adalah sebagai berikut:

$$Y > 60 \text{ And } Y > (1.5 * cr) \text{ And } cr > 17 \\ \text{And } cb < 1.2 * cr \text{ And } cb > 5.$$

Hasil persamaan tersebut cukup stabil untuk standart foto dengan lighting yang normal. Setelah didapatkan persamaan seperti diatas maka dilakukan threshold dengan menggunakan persamaan tersebut. Dari hasil threshold diperoleh segmentasi warna kulit dan objek yang mengandung warna kulit. Gambar 6 berikut ini adalah contoh hasil proses *skin detection*.





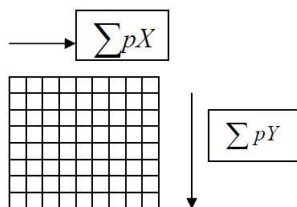
**Gambar 6** Hasil *skin detection*

Lokasi Wajah dengan *Integral Projection*

Pendeteksian lokasi wajah (*Face Localization*) adalah tahap dimana kita akan melakukan proses untuk mencari lokasi objek yaitu wajah. Lokasi wajah didapatkan dari hasil *skin detection*, *skin detection* menghasilkan segmentasi object wajah yang didapat dari theshold warna kulit.

Untuk proses pencarian lokasi wajah ini kita lakukan dengan metode *Integral Projection*. Setelah kita temukan daerah warna kulit, maka untuk mencari lebar dan tingginya daerah tersebut kita lakukan dengan *integral projection*.

Seperti yang telah dijelaskan di bab sebelumnya, metode ini akan sangat membantu saat kita mencari lokasi object yang kita perlukan. Metode ini akan menjumlahkan pixel per baris dan per kolomnya, seperti gambaran berikut:



**Gambar 7** Proses *Integral Projection*

Keterangan:

$\sum pX$  = jumlah pixel X per baris

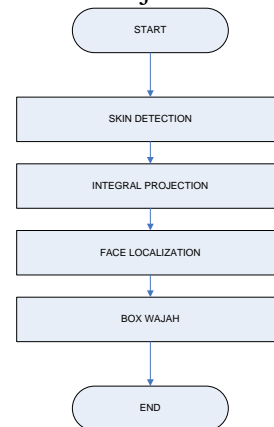
$\sum pY$  = jumlah pixel Y per kolom

Jadi tiap pixel x kita jumlahkan sepanjang width dan sebanyak high-nya, begitu juga dengan penjumlahan pixel y-nya.

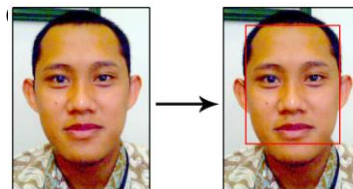
Dalam pencarian lokasi wajah ini, kita akan mencari batas-batas kulit. Tahapannya adalah sebagai berikut, pertama kita integrasikan setiap pixel ke arah sumbu y sepanjang sumbu x (lebar gambar), untuk pixel yang lebih kecil dari *threshol*d maka termasuk kulit jika lebih besar maka bukan kulit. Hal ini

dapat dijadikan sebagai batasan antara kulit atau bukan kulit

Berikut adalah *Flowchart* untuk proses pencarian lokasi wajah:



**Gambar 8** Pencarian Lokasi Wajah

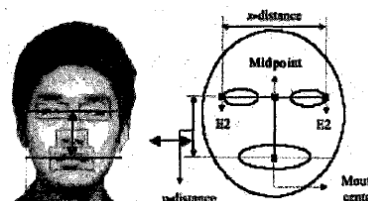


**Gambar 9** *Facelocalization*

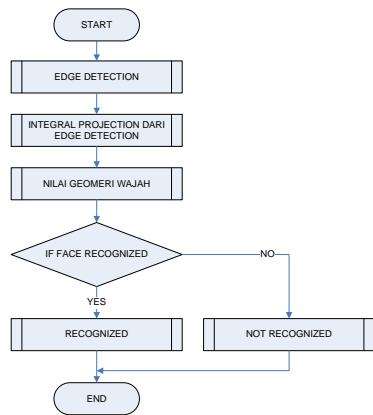
Deteksi Fitur-Fitur Wajah

Setelah didapatkan lokasi wajah, maka kita akan lakukan proses untuk mencari lokasi fitur wajah. Untuk mendapatkan lokasi fitur wajah ini juga kita lakukan *integral projection* didalam lokasi daerah warna kulit yang telah kita temukan tadi. *Integral projection* kita lakukan pada gambar hasil *edge detection* yang memiliki daerah warna kulit.

Untuk mempermudah pencarian lokasi fitur wajah, digunakan *geometry face*, yaitu pembagian lokasi wajah yang dijelaskan pada gambar berikut sebagai berikut:



**Gambar 10** *Geometry face*



**Gambar 11** Pencarian Fitur Wajah

Dalam mencari fitur-fitur wajah menggunakan *Edge detection*. Tetapi sebelumnya gambar diubah kedalam *greyscale*. Setelah kita dapat lokasi wajah dari tahap sebelumnya, maka kita proses gambar *Edge detection* pada daerah-daerah yang sudah ditentukan oleh *geometry face*.

Setelah dilakukan proses *edge detection*, kita jumlahkan nilai per baris dan kolomnya (*Integral Projection*) hanya pada lokasi wajah yang telah ditentukan sebelumnya. Dalam artian penjumlahan baris dan kolom hanya dilakukan pada daerah wajah yang sebelumnya telah dibatasi dengan menggunakan sebuah *box*.

Setelah dilakukan operasi *edge detection*, untuk mempermudah pencarian fitur-fitur wajah dalam box wajah maka kami menerapkan metode *geometry face*. Dengan metode ini pencarian hanya dilakukan dalam lokasi-lokasi tertentu saja.

Berikut adalah ketentuan pencarian lokasi fitur pada wajah :

**Tabel 3.1** *Geometry Face*

Fitur Wajah	Lokasi pada box (x)
Alis	Diatas batas mata
Mata	x/4

Dari hasil *integral projection* pada bagian mata dan alis berdasarkan *geometry face* maka akan didapatkan

lokasi masing masing fitur mata dan alis, seperti yang langkah-langkah dilakukan pada *face localization*, kita ambil batas-batas pada masing-masing fitur baik mata atau alis. Setelah didapatkan batas-batas tersebut kemudian dibuat box fitur.









Hasil *edge detection* dengan menggunakan filter sobel diatas relatif kurang stabil karena masih banyak mengandung noise, sehingga terkadang pengambilan batas untuk masing-masing fitur tidak presisi.

Pada proses *fitur localization* ini penentuan ketepatan dalam mendapatkan lokasi fitur sangat di pengaruhi oleh hasil dari *face localization*. Jika hasil *face localization* tidak presisi maka kemungkinan besar lokasi fitur yang didapatkan pun tidak presisi.

Berikut ini pengujian *fitur detection* untuk 10 gambar

No	Gambar asal	Hasil <i>fitur detection</i>	Analisa
1			Hasil <i>fitur detection</i> baik
2			Hasil <i>fitur detection</i> baik
3			Hasil <i>fitur detection</i> baik
4			Hasil <i>fitur detection</i> baik
5			Hasil <i>fitur detection</i> baik
6			Hasil <i>fitur detection</i> kurang baik

Created with

7			Hasil fitur detection kurang baik
8			Hasil fitur detection baik
9			Hasil fitur detection baik
10			Hasil fitur detection baik

Dari hasil yang didapat, ada beberapa gambar yang kurang baik proses fitur detection nya, hal ini terjadi karena kadang daerah yang bukan fitur masih ikut terdeteksi. Untuk mendapatkan hasil yang lumayan maka wajah kita harus lurus. Metode geometri wajah tidaklah sempurna. Karena metode ini menerapkan batasan bahwa fitur tertentu terletak di posisi tertentu. Untuk itu proses deteksi wajah harus dilakukan secara benar terutama pembuatan box wajahnya. Apabila box wajah tidak benar otomatis pencarian fitur wajah juga akan terganggu.

Kesalahan menentukan posisi wajah bisa disebabkan kesalahan pada pembuatan sampel kulit. Bisa juga karena background gambar berwarna menyerupai kulit karena pencahayaan yang buruk. Sebenarnya efek pencahayaan harus diperhitungkan. Akan tetapi seperti yang telah dijelaskan sebelumnya, kami hanya memperhatikan nilai-nilai tint dan saturasi saja. Sedangkan nilai light yang memegang peranan dalam penghitungan cahaya tidak diikutsertakan dalam proses.

#### Template Matching

Beberapa objek memiliki karakteristik yang menampilkan rata-rata lebar dari tampilan sesuai petunjuk

dan kondisi. Sedangkan *template matching* merupakan sebuah strategi pengenalan objek yang menemukan objek dengan mencocokkan gambar bagian-bagian kecil dengan contoh template.

Aplikasi alamiah dari *template matching* digunakan untuk membangun gambar template yang diminta untuk membagi kategori semantik. Template ini dapat dibangun *on-line*, dan digunakan untuk menyederhanakan query dengan menggunakan suatu template yang ada dan bukan menyusun suatu query. Menemukan wajah merupakan bagian *template matching* dengan tampilan frontal dari wajah, kemudian mencari bagian yang gelap yang khas dari mata. Jika wajah dapat ditemukan, maka tidak terikat pada identitas dari seseorang tertentu yakni hanya dengan mencari polanya.



Gambar 12 Hasil Template Matching 1



Gambar 13 Hasil Template Matching 2





**Gambar 14** Hasil *Template Matching* 3

#### 4.KESIMPULAN

Dalam penelitian ini, hal yang sangat berpengaruh dalam pengolahan image adalah memodelkan warna kulit dan mencari rata-rata kulit pada umumnya. Hasil ini akan berpengaruh untuk hasil proses selanjutnya dan hasilnya sangat menentukan baik buruknya kandidat wajah dan akan berpengaruh pada rangkaian sistem selanjutnya.

Berikut adalah beberapa kesimpulan yang dapat diambil dari percobaan dan pengujian:

1. Proses pengambilan objek gambar harus dilakukan berkali-kali dan harus dalam kondisi yang berbeda-beda baik lokasi atau pencahayaan agar nantinya saat diolah bisa memproses dalam keadaan yang berbeda-beda.
2. Penggunaan metode *integral projection* akan memudahkan kita untuk mencari lokasi objek, tapi kita harus lebih teliti dalam pengoperasiannya. Dengan menggunakan *integral projection* ini memudahkan kita untuk mencari koordinat batas objek.
3. Dalam proses pengambilan titik dari masing-masing fitur wajah perlu dilakukan beberapa normalisasi terhadap jarak fitur dan kemiringannya.
4. Proses pengambilan titik sangat dipengaruhi oleh hasil *edge detection*, semakin sedikit *noise* dalam *edge*

*detection* maka hasil titik semakin presisi dan sebaliknya.

#### 5. DAFTAR PUSTAKA

Awaludin Ujang Fajar, 2006, Mendeteksi Wajah Menggunakan Sketsa, ITS

Basuki Ahmad and Palandi F. Jozua and fatchurrochman, 2005, Pengolahan Citra Digital menggunakan VisualBasic, Graha Ilmu

Fadlisyah, 2007, Computer Vision dan Pengolahan Citra, ANDI

Kurniawan Rulianto, 2010, PHP dan MySQL Untuk Orang Awam Edisi Ke 2, Maxikom

Maziyah Millatul dan Noortjahja Andy, 2007, Implementasi VB.06 pada Face Detection Berbasis Image Processing untuk Sistem Identifikasi, UNESA

Sutoyo T, Mulyanto Edy, dkk, 2009, Teori Pengolahan Citra Digital, ANDI

Created with

 **nitro**<sup>PDF</sup> professional

download the free trial online at [nitropdf.com/professional](https://nitropdf.com/professional)