

Implementasi Algoritma Hard Melokalisasi Fitur Wajah Pada Citra Warna

Ujang Juhardi¹, Marhalim²

^{1,2}Jurusan Teknik Informatika, Fakultas Teknik, Universitas Muhammadiyah Bengkulu

Email: ujangjuhardi@umb.ac.id¹, marhalim.harmin@gmail.com²

Abstrak

Pendeteksian wajah (*face detection*) adalah salah satu tahap awal yang sangat penting dalam sistem pengenalan wajah (*face recognition*) yang digunakan dalam identifikasi biometrik. Sejauh ini, kendala utama yang dihadapi dalam sistem pendeteksian wajah berkisar pada masalah ukuran (resolusi citra), Resolusi citra merupakan tingkat detailnya suatu citra. Semakin tinggi resolusinya semakin tinggi pula tingkat detail dari citra tersebut. Haar like Feature merupakan metode yang lazim digunakan dalam pendeteksian obyek khususnya pendeteksian wajah dan fitur-fiturnya. Prinsip Haar-like features adalah mengenali obyek berdasarkan nilai sederhana dari fitur tetapi bukan merupakan nilai piksel dari image obyek tersebut. Metode ini memiliki kelebihan yaitu komputasinya sangat cepat, karena hanya bergantung pada jumlah piksel dalam persegi bukan setiap nilai piksel dari sebuah image. Untuk mengimplementasikan dan menganalisis kecepatan hasil algoritma haar dalam melokalisasi fitur wajah penelitian ini menggunakan software MATLAB R2012b agar dapat mengetahui bagaimana cara menganalisis dan mengimplementasikan serta mendapatkan hasil menganalisis pengaruh resolusi citra dari algoritma haar dalam melokalisasi fitur wajah (mata, hidung, dan mulut). Penelitian ini dilaksanakan secara mandiri baik pengambilan data sekunder maupun proses pengolahan datanya, untuk metode pengumpulan data pada penelitian ini penulis menggunakan metode studi pustaka dan studi laboratorium. Disarankan dengan adanya penelitian ini, penulis berharap dapat memberikan kontribusi kepada peneliti yang lain untuk meneliti pengaruh-pengaruh lain yang mempengaruhi keberhasilan algoritma haar, sehingga algoritma haar dapat dikembangkan lebih baik lagi.

Kata kunci: Pendeteksian wajah, resolusi citra, haar like feature

Abstract

Face detection is one of the early stage is very important in a facial recognition system (*face recognition*) used in biometric identification. So far, the main obstacle in the face detection system revolves around the issue size (image resolution), the image resolution is the level of detail of an image. The higher the resolution the higher the level of detail of that image. Haar like Feature is a method commonly used in the detection of objects particularly the face detection and its features. Principle Haar-like features are simple to recognize objects based on the value of the feature but not the pixel values of the image of that object. This method has the advantage that the computation is very fast, because it depends on the number of pixels in a square instead of each pixel value of an image. To implement and analyze speed haar algorithm results in a localized facial features of this research using MATLAB R2012b software in order to know how to analyze and implement and get the results to analyze the effect of image resolution algorithm localizes haar in the facial features (eyes, nose, and mouth). This research was carried out independently both secondary data collection and processing of data, for the data collection method in this study the authors use the method of literature and laboratory studies. Suggested the presence of this study, the authors hope to contribute to fellow-researcher to examine other influences which affect the success haar algorithms, so the algorithm can be developed haar better.

Keywords: face detection, image resolution, haar like feature

1. PENDAHULUAN

Computer Vision merupakan pengembangan dari pengolahan citra digital yang menambahkan kemampuan untuk dapat memahami dan mengerti sebuah citra. Telah sering dilakukannya penelitian dalam bidang ini diantaranya menggunakan mata manusia, gerakan tangan, gerak isyarat, gerak tubuh, gerak jari, dan sebagainya[13].

Pendeteksian wajah (*face detection*) adalah salah satu tahap awal yang sangat penting dalam sistem pengenalan wajah (*face recognition*) yang digunakan dalam identifikasi biometrik. Deteksi wajah juga dapat digunakan untuk pencarian atau pengindeksan data wajah dari citra yang berisi wajah dengan berbagai ukuran, posisi, dan latar belakang dari wajah tersebut. Pendekatan untuk melakukan pendeteksian wajah adalah Resolusi citra. Resolusi citra merupakan tingkat detailnya suatu citra. Semakin tinggi resolusinya semakin tinggi pula tingkat detail dari citra tersebut. Berdasarkan penelitian dapat diprediksikan bahwa kecepatan dalam pendeteksian wajah dan fitur-fitur wajah terkait juga dengan resolusi citra, dengan resolusi citra yang semakin besar maka perhitungan komputasi oleh komputer juga akan semakin besar sehingga pada akhirnya akan memperlambat waktu eksekusi (kecepatan deteksi). Sebaliknya semakin kecil resolusi citra, nilai digital citra tersebut semakin kecil, maka perhitungan komputasi oleh komputer juga akan semakin kecil dan akhirnya kecepatan pendeteksian akan meningkat[1].

Haar-like features merupakan *rectangular features* (fungsi persegi), yang memberikan indikasi secara spesifik pada sebuah gambar atau *image*. Prinsip *Haar-like features* adalah mengenali obyek berdasarkan nilai sederhana dari fitur tetapi bukan merupakan nilai piksel dari *image* obyek tersebut. Metode ini memiliki kelebihan yaitu komputasinya sangat cepat, karena hanya bergantung pada jumlah piksel dalam persegi bukan setiap nilai piksel dari sebuah *image* [2].

2. LANDASARN TEORI

2.1 Citra Digital (*Image*) dan Citra Berwarna

Citra adalah suatu representasi (gambaran), kemiripan, atau imitasi dari suatu objek. Citra sebagai suatu keluaran dari sistem perekaman data yang dapat bersifat optik, bersifat analog ataupun bersifat digital[3].

2.2 Citra Digital

Citra digital merupakan citra yang dapat diolah oleh komputer. Sebuah citra digital dapat mewakili oleh sebuah matriks yang terdiri dari M kolom N baris, dimana perpotongan antara kolom dan baris disebut piksel (piksel = picture element), yaitu elemen terkecil dari sebuah citra[4].

Piksel mempunyai dua parameter, yaitu koordinat dan intensitas atau warna. Nilai yang terdapat pada koordinat (x,y) adalah f(x,y), yaitu besar intensitas atau warna dari piksel di titik itu. Oleh sebab itu, sebuah citra digital dapat ditulis dalam bentuk matriks 2.1 berikut :

$$f_{x,y} = \begin{matrix} f(0,0) & \dots & f(0,M-1) \\ \dots & \dots & f(1,M-1) \dots \dots \dots (2) \\ f(N-1,0) & f(N-1,1) & f(N-1,M-1) \end{matrix}$$

Berdasarkan gambaran tersebut, secara matematis citra digital dapat dituliskan sebagai fungsi intensitas f(x,y), dimana harga x (baris) dan y (kolom) merupakan koordinat posisi dan f(x,y) adalah nilai fungsi pada setiap titik (x,y) yang menyatakan besar intensitas citra atau tingkat keabuan atau warna dari piksel di titik tersebut. Pada proses digitalisasi (sampling dan kuantitas) diperoleh besar baris M dan kolom N hingga citra membentuk matriks M x N dan jumlah tingkat keabuan piksel G[5].

2.3 Citra Berwarna

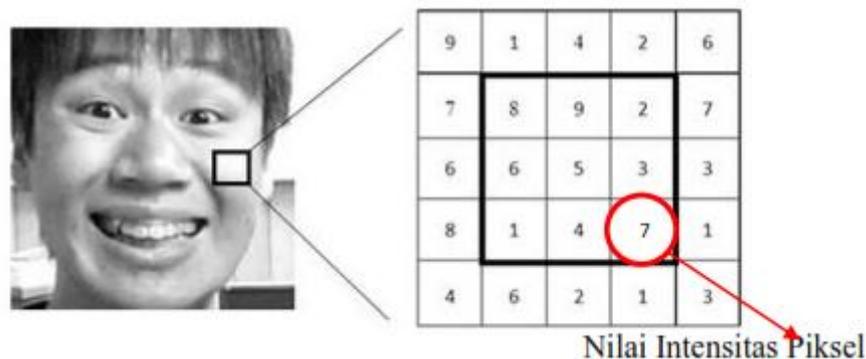
Format data citra digital berhubungan erat dengan warna. Setiap piksel pada citra warna mewakili warna yang merupakan kombinasi dari tiga warna dasar (RGB = Red

Green Blue). Setiap warna dasar menggunakan penyimpanan 8 bit = 1 byte, yang berarti mempunyai gradasi sebanyak 255. Penyimpanan citra true color didalam memori berbeda dengan citra grayscale. Setiap piksel dari citra grayscale 256 gradasi warna diwakili oleh 1 byte. Sedangkan 1 piksel citra true color diwakili oleh 3 byte yang masing- masing byte merepresentasikan warna merah (*Red*), hijau (*Green*), biru (*Blue*)[5].

2.4 Piksel

Piksel berasal dari akronim bahasa Inggris *Picture Element* yang disingkat menjadi pixel. Piksel adalah unsur gambar atau representasi sebuah titik terkecil dalam sebuah gambar grafis yang dihitung per inchi. Piksel juga disebut titik-titik cahaya yang membentuk sebuah objek. Makin banyak jumlah piksel dalam sebuah citra, makin besar resolusi spasial citra tersebut sehingga citra terlihat makin tajam [6].

Setiap piksel mewakili tidak hanya satu titik dalam sebuah citra melainkan sebuah bagian berupa kotak yang merupakan bagian terkecil. Berikut ini pada Gambar 2.2 nilai piksel dari citra objek manusia.



Gambar 2.2. Nilai piksel dari citra objek manusia[7].

2.5 Resolusi Citra

Resolusi citra merupakan tingkat detailnya suatu citra. Semakin tinggi resolusinya semakin tinggi pula tingkat detail dari citra tersebut[1]. Ada dua jenis resolusi yang perlu diketahui, yaitu [5]:

1. Resolusi Spasial

Resolusi spasial ini merupakan ukuran halus atau kasarnya pembagian kisi-kisi baris dan kolom pada saat sampling. Resolusi ini dipakai untuk menentukan jumlah pixel per satuan panjang. Biasanya satuan resolusi ini adalah dpi (*dot per inchi*). Resolusi ini sangat berpengaruh pada detail dan perhitungan gambar.

2. Resolusi kecemerlangan

Resolusi kecemerlangan (intensitas/ brightness) atau biasanya disebut dengan kedalaman bit/ kedalaman warna (*Bit Depth*) adalah ukuran halus kasarnya pembagian tingkat gradasi warna saat dilakukan kuantisasi. *Bit Depth* menentukan berapa banyak informasi warna yang tersedia untuk ditampilkan dalam setiap piksel. Semakin besar nilainya, semakin bagus kualitas gambar yang dihasilkan dan tentu ukuran juga semakin besar. Resolusi citra berkaitan erat dengan kualitas resolusi citra, Kualitas dapat didefinisikan sebagai suatu kondisi dinamis yang berhubungan dengan produk. jasa. Inanusia. proses dan lingkungan yang memenuhi atau melebihi harapan[8].

2.6 Pendeteksian Wajah dan Lokalisasi Fitur Wajah

“Pendeteksian wajah (*face detection*) adalah salah satu tahap awal yang sangat penting sebelum dilakukan proses pengenalan wajah (*face recognition*)” [9]. Pada tahap ini citra wajah yang akan dideteksi dibagi kedalam dua bagian yaitu wilayah wajah dan wilayah bukan wajah (latar belakang citra). “Deteksi wajah dapat dipandang sebagai masalah klasifikasi pola dimana inputnya adalah citra masukan dan akan ditentukan output yang berupa label kelas dari citra tersebut. Dalam hal ini terdapat dua label kelas, yaitu wajah dan nonwajah”[10].

2.7 Algoritma

Algoritma adalah prosedur komputasi yang mengambil beberapa nilai atau kumpulan nilai sebagai input kemudian di proses sebagai output sehingga algoritma merupakan urutan langkah komputasi yang mengubah input menjadi output[11].

Kriteria Algoritma yang baik :

1. Tepat, benar, sederhana, standar dan efektif.
2. Logis, terstruktur dan sistematis.
3. Semua operasi terdefinisi.
4. Semua proses harus berakhir setelah sejumlah langkah dilakukan.
5. Ditulis dengan bahasa yang standar dengan format pemrograman agar mudah untuk diimplementasikan dan tidak menimbulkan arti ganda.

2.8 Algoritma Haar

Algoritma *Haar* merupakan Algoritma pendeteksian obyek yang memiliki tingkat keakuratan yang cukup tinggi yaitu sekitar 93,7 % dengan kecepatan 15 kali lebih cepat daripada detektor Rowley Baluja-Kanade dan kurang lebih 600 kali lebih cepat daripada detektor Schneiderman-Kanade. Algoritma ini, diusulkan oleh Paul Viola dan Michael Jones pada tahun 2001[2]. Algoritma *Viola-Jones* menggabungkan empat kunci utama yaitu *Haar Like Feature*, *Integral Image*, *Adaboost learning* dan *Cascade classifier*. *Haar Like Feature* yaitu selisih dari jumlah piksel dari daerah di dalam persegi panjang. Contoh *Haar Like Feature* disajikan dalam Gambar 2.6.



Gambar 2.6. Contoh *Haar Like Feature* [2]

Nilai *Haar Like Feature* diperoleh dari selisih jumlah nilai piksel daerah gelap dengan jumlah nilai piksel daerah terang:

$$F_{Harr} = \sum F_{white} - \sum F_{Black} \quad (2.2)$$

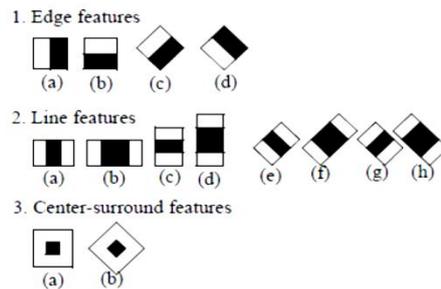
$$F_{Harr} = \text{Nilai fitur total}$$

$$\sum F_{white} = \text{Nilai fitur pada daerah terang}$$

$$\sum F_{Black} = \text{Nilai fitur pada daerah gelap}$$

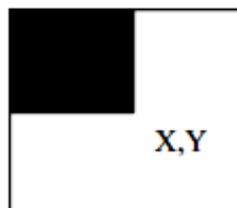
Setiap Haar-Like Feature terdiri dari gabungan kotak-kotak hitam dan putih. Ada tiga tipe kotak feature dalam Haar:

- a. Tipe *two-rectangle feature* (horizontal, vertikal)
- b. Tipe *three-rectangle feature*
- c. Tipe *four-rectangle feature*



Gambar 2.7. Variasi Fitur pada Haar[12]

Integral Image yaitu suatu teknik untuk menghitung nilai fitur secara cepat dengan mengubah nilai dari setiap piksel menjadi suatu representasi citra baru, sebagaimana disajikan dalam Gambar 2.8



Gambar 2.8. *Integral image* (x,y)[2]

Berdasarkan Gambar 2.8, citra integral pada titik (x,y) ($ii(x,y)$) dapat dicari menggunakan persamaan (2.3):

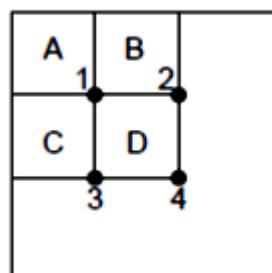
$$ii(x,y) = \sum_{x' \leq x, y' \leq y} i(x',y') \quad (2.3)$$

Keterangan

$ii(x,y)$ = Citra integral pada lokasi x,y

$i(x',y')$ = nilai piksel pada citra asli

Perhitungan nilai dari suatu fitur dapat dilakukan secara cepat dengan menghitung nilai citra integral pada empat buah titik sebagaimana disajikan dalam Gambar 2.9



Gambar 2.9. Perhitungan Nilai Fitur[2]

Jika nilai *integral image* titik 1 adalah A, titik 2 adalah A+B, titik 3 adalah A+C, dan di titik 4 adalah A+B+C+D, maka jumlah piksel di daerah D dapat diketahui dengan cara $4+1 - (2+3)$.

Algoritma *Adaboost learning*, digunakan untuk meningkatkan kinerja klasifikasi dengan pembelajaran sederhana untuk menggabungkan banyak *classifier* lemah menjadi satu *classifier* kuat. *Classifier* lemah adalah suatu jawaban benar dengan tingkat kebenaran yang kurang akurat. Sebuah *classifier* lemah dinyatakan:

$$H_j(x) = \begin{cases} 1, & \text{jika } p_j f_j(x) < p_j q_j(x) \\ 0, & \text{lainnya} \end{cases} \quad (2.4)$$

Keterangan

- $H_j(x)$ = adalah klasifikasi lemah
 p_j = adalah *parity* ke j
 q_j = adalah *threshold* ke j
 x = adalah dimensi *sub image* misalnya 24x24

Langkah-langkah untuk mendapatkan sebuah *classifier* kuat dinyatakan dalam suatu algoritma sebagai berikut :

1. Diberikan contoh gambar $(x_1, y_1), \dots, (x_n, y_n)$ dimana $y_i = 0$ untuk contoh positif dan $y_i = 1$ untuk contoh negatif
2. Inisialisasi bobot $w_{i,1} = \frac{1}{m'} \frac{1}{2l}$; m dan l adalah jumlah negatif dan positif
3. Untuk $t=1, \dots, T$ Menormalkan bobot sehingga w_t adalah distribusi probabilitas

$$w_{t,i} \leftarrow \frac{w_{t,i}}{\sum_{j=1}^n w_{t,i}} \quad (2.5)$$

Untuk setiap fitur, j melatih classifier h_j , untuk setiap fitur tunggal Kesalahan (e_j) dievaluasi dengan bobot w_t

$$e_j = \sum_i w_i |h_j(x_i) - y_i| \quad (2.6)$$

Pilih classifier h_t dengan eror terkecil dimana $e_i = 0$ untuk x_i adalah klasifikasi benar, dan $e_i = 1$ untuk yang lain.

Perbaharui bobot :

$$w_{t+1,i} = w_{t,i} \beta_t^{1-e_i} \quad (2.7)$$

$$\text{Dimana } \beta_t = \frac{\epsilon t}{1 - \epsilon t} \quad (2.8)$$

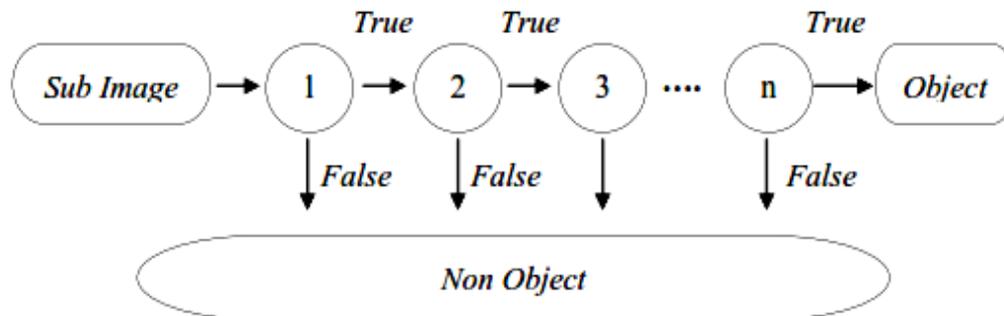
Didapatkan sebuah *Classifier* kuat yaitu

$$H_j(x) = \begin{cases} 1, & \sum_{t=1}^T \alpha_t h_t(x) \geq \frac{1}{2} \sum_{t=1}^T \alpha_t \\ \text{lainnya} & \end{cases} \quad (2.9)$$

dimana :

$$\alpha_t = \log \frac{1}{\beta_t} \quad (2.10)$$

Cascade classifier adalah sebuah metode untuk mengkombinasikan *classifier* yang kompleks dalam sebuah struktur bertingkat yang dapat meningkatkan kecepatan pendeteksian obyek dengan memfokuskan pada daerah citra yang berpeluang saja. Struktur *cascade classifier* disajikan Gambar 2.10.



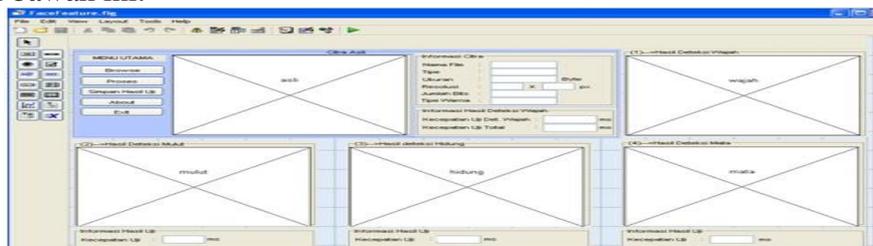
Gambar 2.10. Cascade Clasifier [2]

Gambar 2.10 menjelaskan proses penyeleksian keberadaan obyek. Di asumsikan suatu *sub image* di evaluasi oleh *classifier* pertama dan berhasil melewati *classifier* tersebut, hal ini mengindikasikan *sub image* berpotensi terkandung obyek dan dilanjutkan pada *classifier* ke dua sampai dengan ke-n, jika berhasil melewati keseluruhan *classifier*, maka disimpulkan terdapat obyek yang dideteksi. Jika tidak, proses evaluasi tidak dilanjutkan ke *classifier* berikutnya dan disimpulkan tidak terdapat obyek.

4. HASIL DAN ANALISIS

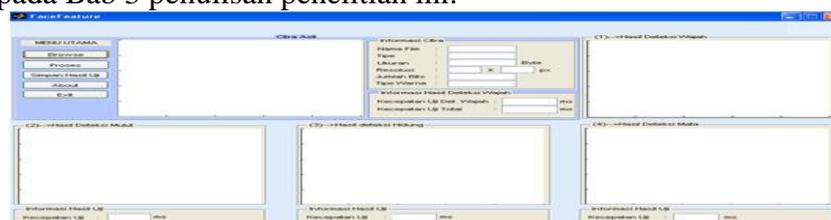
4.1. HASIL

Untuk membuat tampilan GUI (Graphical User Interface) aplikasi, software matlab R2012b sudah menyediakan toolbox-toolbox yang bisa langsung digunakan dengan cara drag and drop pada halaman utama design. Seperti ditunjukkan pada gambar di bawah ini.



Gambar 4.2 Tampilan Toolbox Design Aplikasi Matlab R2012b

Halaman utama aplikasi, hasil dari design dan penulisan kode program pada software matlab yang telah berhasil tanpa ada error seperti di tunjukkan pada gambar 4.3 Tampilan utama ini digunakan untuk melakukan pengujian pendeteksian wajah dan fitur wajah terhadap citra input yang resolusi dan ukuran dimensi yang berbeda seperti dijelaskan pada Bab 3 penulisan penelitian ini.



Gambar 4.3 Tampilan Utama Aplikasi

Agar tampilan utama ini dapat digunakan, maka pada editor penulisan kode program terlebih dahulu harus mengklik tombol Run (lihat gambar 4.6)

Ada 5 menu pada tampilan utama aplikasi ini yaitu

1. Browse, merupakan tombol yang digunakan untuk memilih citra input yang akan diuji. Setelah diklik maka, jendela open dialog pemilihan citra akan muncul seperti terlihat pada gambar 4.5. di bawah ini.



Gambar 4.5 Jendela pilih citra input

Pada jendela input, user harus memilih salah satu dari input citra yang akan diuji, lalu klik tombol open. Aplikasi ini akan menampilkan citra dan informasi citra tersebut ke dalam tampilan utama, seperti terlihat pada gambar 4.6 di bawah ini.



Gambar 4.6 Tampilan Input citra uji dan informasinya

2. Proses, merupakan tombol untuk memulai melakukan perhitungan dan pendeteksian wajah dan fitur nya (mulut, hidung dan mata). Perhitungan pada proses ini meliputi perhitungan waktu eksekusi pendeteksian baik pendeteksian wajah dan pendeteksian fitur serta total waktu eksekusi keseluruhan. Tampilan proses seperti ditunjukkan pada gambar 4.7



Gambar 4.7 Tampilan Proses Aplikasi

Wajah dan fitur wajah (mulut, hidung, mata) yang berhasil dideteksi oleh aplikasi akan ditampilkan ke dalam citra hasil deteksi seperti ditunjukkan pada gambar 4.7 berikut juga dengan total waktu eksekusi masing-masing. Aplikasi akan menampilkan windows informasi bahwa proses selesai.

3. Simpan Hasil Uji, merupakan tombol yang berguna untuk menyimpan hasil citra hasil uji deteksi wajah, mulut, mata dan hidung. Aplikasi akan menyimpan secara otomatis ke dalam folder citra input yang di masukkan. Lihat gambar 4.8



Gambar 4.8 Tampilan proses tombol Simpan Hasil Uji

Aplikasi akan menampilkan dialog informasi bahwa proses penyimpanan citra hasil uji berhasil, seperti ditunjukkan pada gambar 4.8 di atas, Hasil penyimpanan citra uji berupa 4 buah citra hasil uji seperti ditunjukkan pada gambar di bawah ini.



Gambar 4.9 Hasil penyimpanan citra hasil deteksi wajah dan fitur nya (mulut, hidung, mata)

4.2. ANALISIS

- 4.2.1. Pembahasan Perhitungan Pengaruh Resolusi terhadap kecepatan deteksi wajah dan fitur nya (mulut, hidung, mata).

Sebelum melakukan proses pengujian dan perhitungan kecepatan (waktu eksekusi), penulis akan menerangkan input yang akan digunakan dalam pengujian ini. Input yang penulis gunakan pada penelitian ini berupa file citra berformat jpg sebanyak 10 buah citra yang penulis ambil secara acak di internet lewat akun teman-teman penulis di facebook. 10 buah citra input tersebut memiliki latar belakang (background), posisi dan ekspresi yang berbeda untuk lebih jelasnya dapat di lihat pada gambar 4.12 di bawah ini



Gambar 4.12 Sepuluh (10) buah citra input yang akan diuji

Informasi dimensi (ukuran p x l) dari citra input dan resolusinya dalam satuan PPI (pixel per inch) seperti di jelaskan pada tabel berikut:

Tabel 4. 2 Informasi Ukuran dan Resolusi Citra Input

No	Nama Citra	Ukuran Resolusi (px)			Resolusi(PPI)	ukuran (kb)
1	1.jpg	960	x	960	72	191
2	2.jpg	202	x	305	96	15
3	3.jpg	720	x	720	72	29
4	4.jpg	720	x	720	72	46
5	5.jpg	391	x	391	72	43
6	6.jpg	360	x	360	72	25
7	7.jpg	480	x	640	72	46
8	8.jpg	720	x	720	72	124
9	9.jpg	960	x	960	72	97
10	10.jpg	320	x	426	72	30

Untuk mendapatkan hasil pengaruh resolusi citra (Ukuran dan resolusi) terhadap kecepatan algoritma haar dalam mendeteksi wajah dan fitur wajah, maka penulis merancang citra input agar menjadi citra input yang berbeda yaitu dengan meningkatkan ukuran resolusi (PPI) yang secara tidak langsung akan merubah ukuran dimensi (p x l) pixel citra tersebut. Peningkatan citra uji tersebut penulis rancang untuk meningkat sebanyak 25%,50%,75% dan 100% (2x lipat) dari ukuran citra input masing-masing terhadap 10 citra input tersebut. Tabel peningkatan citra input tersebut seperti pada tabel di bawah ini.

Tabel 4. 3 Citra input setelah resolusi ditingkatkan

No	Nama Citra	resolusi 1 (+25%)			PPI (+25%)	Resolusi 2(+50%)			PPI (+50%)	Resolusi 3(+75%)			PPI (+75%)	Resolusi 4(+100%)			PPI (+100%)
			x				x			x			x		x		
1	1.jpg	1200	x	1200	90	1440	x	1440	108	1680	x	1680	126	1920	x	1920	144
2	2.jpg	252.5	x	381.25	120	303	x	457.5	144	353.5	x	533.75	168	404	x	610	192
3	3.jpg	900	x	900	90	1080	x	1080	108	1260	x	1260	126	1440	x	1440	144
4	4.jpg	900	x	900	90	1080	x	1080	108	1260	x	1260	126	1440	x	1440	144
5	5.jpg	488.75	x	488.75	90	586.5	x	586.5	108	684.25	x	684.25	126	782	x	782	144
6	6.jpg	450	x	450	90	540	x	540	108	630	x	630	126	720	x	720	144
7	7.jpg	600	x	800	90	720	x	960	108	840	x	1120	126	960	x	1280	144
8	8.jpg	900	x	900	90	1080	x	1080	108	1260	x	1260	126	1440	x	1440	144
9	9.jpg	1200	x	1200	90	1440	x	1440	108	1680	x	1680	126	1920	x	1920	144
10	10.jpg	400	x	532.5	90	480	x	639	108	560	x	745.5	126	640	x	852	144

Untuk mendapatkan citra input yang sama namun dengan resolusi yang berbeda, penulis menggunakan software adobe photoshop cs 3. Seperti ditunjukkan pada file photoshop berikut.

Name	Size	Type
1.psd	11.208 KB	PSD File
2.psd	2.489 KB	PSD File
3.psd	7.633 KB	PSD File
4.psd	10.783 KB	PSD File
5.psd	4.695 KB	PSD File
6.psd	3.713 KB	PSD File
7.psd	7.368 KB	PSD File
8.psd	9.526 KB	PSD File
9.psd	22.504 KB	PSD File
10.psd	4.252 KB	PSD File

Gambar 4.13 File Photoshop untuk mengubah resolusi citra

Hasil perhitungan waktu eksekusi (kecepatan) terhadap 10 buah citra input dengan resolusi dan ukurannya sesuai dengan resolusi asli dan ukuran asli citra inputnya, menghasilkan perhitungan waktu eksekusi sebagai berikut:

Tabel 4.4 Hasil uji citra input dengan dengan resolusi asli

No	Nama Citra	Waktu Resolusi Asli (ms)				
		wajah	mulut	hidung	mata	total
1	1.jpg	2,21632	2,14974	3,46088	0,87491	8,70185
2	2.jpg	0,48748	0,53639	0,92143	0,34936	2,29466
3	3.jpg	1,13577	1,15693	2,08557	0,57263	4,95090
4	4.jpg	1,29842	1,23277	2,11974	0,53647	5,18740
5	5.jpg	0,6419	0,67175	1,11659	0,40766	2,83790
6	6.jpg	0,58504	0,66288	1,10788	0,38223	2,73803
7	7.jpg	0,95131	0,97938	1,64418	0,45541	4,03028
8	8.jpg	1,18106	1,40287	2,05473	0,58756	5,22622
9	9.jpg	2,14008	1,67223	3,6737	0,72679	8,21280
10	10.jpg	0,60431	0,62515	1,01643	0,383261	2,62915
Rata-Rata		1,124169	1,109009	1,920113	0,5276281	4,6809191

Dari tabel di atas menunjukkan bahwa rata-rata waktu eksekusi(kecepatan) algoritma haar dalam mendeteksi wajah adalah sebesar 1,124169 ms,waktu rata-rata mendeteksi mulut 1,109009 ms , waktu rata-rata mendeteksi hidung 1,920113 ms dan waktu rata-rata mendeteksi mata 0,5276281 ms. Untuk waktu eksekusi keseluruhan adalah 4,6809191 ms. Waktu eksekusi tercepat adalah dalam mendeteksi mata sedangkan waktu eksekusi paling lama ada mendeteksi hidung.

Perbandingan perhitungan waktu eksekusi (kecepatan) terhadap 40 buah citra yang telah ditingkatkan resolusinya terhadap perhitungan waktu eksekusi 10 buah citra asli sebelum ditingkatkankan resolusinya dalam mendeteksi wajah,mulut,hidung dan mata dapat di lihat pada tabel berikut.

Tabel 4.5 Tabel Statistik Perbedaan Kecepatan Waktu Eksekusi

No	Nama Citra	Waktu Eksekusi Wajah (ms)					Rata-Rata
		Asli	25%	50%	75%	100%	
1	1.jpg	2,21632	3,2975	4,79227	7,27566	8,31222	5,17879
2	2.jpg	0,48748	0,55249	0,630405	0,734019	0,870727	0,65502
3	3.jpg	1,13577	1,62305	2,19161	3,14302	3,62143	2,34298
4	4.jpg	1,29842	1,86263	2,59261	3,42865	4,31754	2,69997
5	5.jpg	0,6419	0,83478	1,05628	1,31705	1,64458	1,09892
6	6.jpg	0,58504	0,713233	0,876612	1,07816	1,30664	0,91194
7	7.jpg	0,95131	1,28762	1,72203	2,24537	2,88785	1,81884
8	8.jpg	1,18106	1,6744	2,30919	3,01401	3,84317	2,40437
9	9.jpg	2,14008	3,45988	5,25255	6,9993	9,13399	5,39716

10	10.jpg	0,60431	0,775449	0,993001	1,21235	1,49131	1,01528
Rata-Rata		1,124169	1,6081032	2,2416558	3,0447589	3,7429457	2,35233

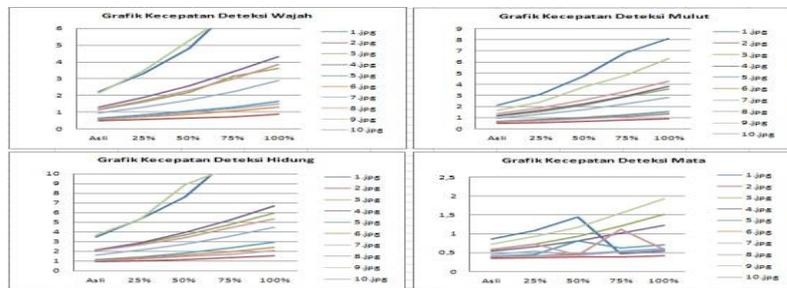
No	Nama Citra	Waktu Eksekusi Mulut (ms)					Rata-Rata
		Asli	25%	50%	75%	100%	
1	1.jpg	2,14974	3,1129	4,75543	6,82404	8,11965	4,99235
2	2.jpg	0,53639	0,60346	0,698342	0,800331	0,937642	0,71523
3	3.jpg	1,15693	1,56929	2,131	2,89879	3,60126	2,27145
4	4.jpg	1,23277	1,68674	2,25603	2,96053	3,82799	2,39281
5	5.jpg	0,67175	0,82558	1,02256	1,27885	1,56934	1,07362
6	6.jpg	0,66288	0,7985	0,963787	1,16369	1,38086	0,99394
7	7.jpg	0,97938	1,30147	1,71302	2,20817	2,80552	1,80151
8	8.jpg	1,40287	1,94038	2,59039	3,37759	4,29752	2,72175
9	9.jpg	1,67223	2,43063	3,77981	4,8109	6,30984	3,80068
10	10.jpg	0,62515	0,739197	0,954038	1,09371	1,34329	0,95108
Rata-Rata		1,109009	1,5008147	2,0864407	2,7416601	3,4192912	2,17144

No	Nama Citra	Waktu Eksekusi Hidung (ms)					Rata-Rata
		Asli	25%	50%	75%	100%	
1	1.jpg	3,46088	5,3351	7,60488	11,5194	13,6115	8,30635
2	2.jpg	0,92143	1,0229	1,16324	1,34064	1,56297	1,20224
3	3.jpg	2,08557	2,7864	3,6753	4,77265	5,98145	3,86027
4	4.jpg	2,11974	2,8879	3,92518	5,22692	6,6602	4,16399
5	5.jpg	1,11659	1,438	1,83994	2,33655	2,96899	1,94001
6	6.jpg	1,10788	1,34078	1,61114	1,95204	2,42122	1,68661
7	7.jpg	1,64418	2,13453	2,75442	3,53639	4,49838	2,91358
8	8.jpg	2,05473	2,67504	3,4204	4,43855	5,38595	3,59493
9	9.jpg	3,6737	5,38407	8,91815	10,6157	13,8683	8,49198
10	10.jpg	1,01643	1,18542	1,46796	1,71093	2,06397	1,48894
Rata-Rata		1,920113	2,619014	3,638061	4,744977	5,902293	3,76489

No	Nama Citra	Waktu Eksekusi Mata (ms)					Rata-Rata
		Asli	25%	50%	75%	100%	
1	1.jpg	0,87491	1,091	1,45447	0,46912	0,537522	0,88540
2	2.jpg	0,34936	0,36152	0,382788	0,386886	0,415616	0,37923
3	3.jpg	0,57263	0,7227	0,9383	1,21443	1,52213	0,99404
4	4.jpg	0,53647	0,66104	0,821253	1,01553	1,22839	0,85254
5	5.jpg	0,40766	0,46557	0,821253	0,625336	0,725171	0,60900
6	6.jpg	0,38223	0,42205	0,435898	0,527503	0,585108	0,47056
7	7.jpg	0,45541	0,53971	0,435898	0,527503	0,585108	0,50873
8	8.jpg	0,58756	0,73539	0,435898	1,12363	0,585108	0,69352
9	9.jpg	0,72679	0,936219	1,17593	1,5565	1,93362	1,26581
10	10.jpg	0,383261	0,419031	0,469128	0,537522	0,603646	0,48252
Rata-Rata		0,52763	0,635423	0,7370816	0,798396	0,8721419	0,71413

Dari hasil uji statistik pada tabel di atas, menunjukkan bahwa terjadi peningkatan kecepatan (Lebih lama) dari setiap peningkatan resolusi dari mulai 25%,50%,75% dan 100% dalam mendeteksi wajah,mulut,hidung dan mata. Untuk

melihat lebih jelas perubahan waktu eksekusi (kecepatan) tersebut penulis menyajikan grafiknya seperti pada grafik di bawah ini.



Gambar 4.14 Grafik Statistik Kecepatan Deteksi

Secara Keseluruhan untuk masing-masing deteksi baik berupa wajah,mulut,hidung dan mata, dapat dengan jelas di lihat pada grafiki di bawah ini.



Gambar 4.15 Grafik Rata-rata kecepatan deteksi secara keseluruhan

Dari Tabel di atas, peningkatan resolusi citra akan membuat pendeteksian wajah dan fiturnya pada algoritma haar menjadi semakin lama.

5. KESIMPULAN

Dari hasil pembahasan dan pengujian yang telah penulis jelaskan pada Bab IV, maka penulis menarik kesimpulan sebagai berikut:

1. Algoritma haar untuk mendeteksi wajah dan fitur wajah (mulut, hidung dan mata) telah berhasil penulis buat dan berhasil dalam mendeteksi wajah dan fiturnya dengan menggunakan software Matlab R2012b.
2. Berdasarkan uji statistik yang telah penulis jelaskan pada Bab IV, penulis menarik kesimpulan bahwa Peningkatan resolusi citra terhadap citra uji yang diinputkan mempengaruhi kecepatan algoritma haar dalam melakukan deteksi wajah dan fiturnya. Semakin tinggi resolusi citra tersebut, semakin lama pula waktu yang dibutuhkan untuk melakukan deteksi wajah dan fiturnya.

6. DAFTAR PUSTAKA

- [1] Putra, D. 2010. *Pengolahan Citra Digital*.Yogyakarta. Penerbit: ANDI.
- [2] Viola, P dan Jones, M. 2004. *Robust Real-Time Detection*. *International Journal of Computer Vision* 57(2):137-154
- [3] Aniati Murni A dan Suryana Setiawan.1992. *Pengantar Pengolahan Citra Digital*.PT Elex Media Komputindo.
- [4] Kusumanto, RD dan Tomponu Novi Alan. 2011. *Pengolahan Citra Digital Untuk Mendeteksi Obyek Menggunakan Pengolahan Warna Model Normalisasi Rgb Jurusan Teknik Komputer, Politeknik Negeri Sriwijaya, Palembang*.

- [5] Sutoyo, T., 2009, Teori Pengolahan Citra Digital. Yogyakarta. Penerbit ANDI.
- [6] Usman, A. 2005. Pengolahan Citra Digital dan Teknik Pemrogramannya. Yogyakarta: Graha Ilmu
- [7] Robin, 2015. *Utechzone Machine Vision Prize* (Online) <http://robinhsieh.com/?p=156> (22 Juni 2015).
- [8] Afnisyah Taurisna. 2010. Analisis Pengaruh Kualitas Resolusi Citra Terhadap Kinerja Metode Pendeteksi Tepi. Universitas Sumatera Utara.
- [9] Yang, M.H., et al. 2002. "Detecting Faces in Images: A Survey". *IEEE Trans. Pattern Analysis and Machine Intelligence*. 24 (1), 34-58.
- [10] Sung, K.K. 1996. Learning and Example Selection for Object and Pattern Detection. AITR 1572: Massachusetts Institute of Technology AI Lab.
- [11] Munir, R. 2007. *Strategi Algoritma* . Teknik Informatika: Bandung.
- [12] Lienhart, R dan Maydt, J. 2002. An Extended Set of Haar-Like Features for Rapid.IEEE ICIP (1):900-903.
- [13] Reswan, Y., & Prabowo, D. A. (2018). Implementasi Kalman Filter Dalam Teknik Hand Tracking Sebagai Kontrol Pointer Mouse Komputer. *Jurnal Sistem Informasi*, 10(1).