

Naskah Publikasi

PROYEK TUGAS AKHIR

**SISTEM DETEKSI CITRA WAJAH
MENGUNAKAN ALGORITMA VIOLA JONES
DAN ADABOOST CASCADE CLASSIFIER**



Disusun oleh:
ANDIKA MAULANA
5150411222

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI DAN ELEKTRO
UNIVERSITAS TEKNOLOGI YOGYAKARTA
2019**

Naskah Publikasi

PROYEK TUGAS AKHIR

**SISTEM DETEKSI CITRA WAJAH
MENGUNAKAN ALGORITMA VIOLA JONES
DAN ADABOOST CASCADE CLASSIFIER**

Disusun oleh:
ANDIKA MAULANA
5150411222

Telah disetujui oleh pembimbing

Pembimbing:

Donny Avianto, S.T.,M.T.

Tanggal:

SISTEM DETEKSI CITRA WAJAH MENGUNAKAN ALGORITMA VIOLA JONES DAN ADABOOST CASCADE CLASSIFIER

Andika Maulana

Program Studi Teknik Informatika, Fakultas Teknologi Informasi & Elektro
Universitas Teknologi Yogyakarta
Jl. Ringroad Utara Jombor Sleman Yogyakarta
Email: andik4maulana@gmail.com

ABSTRAK

Penggunaan teknologi komputer dewasa ini dapat dirasakan pada berbagai macam bidang. Penggunaan komputer berperan penting sebagai alat pendukung dalam mengembangkan suatu teknologi. Penggunaan komputer sebagai alat untuk membantu pekerjaan manusia dirancang untuk mengikuti pola pikir manusia. Segala macam bentuk interaksi manusia dapat ditiru dengan teknologi komputer, salah satunya adalah proses pendeteksian wajah. Deteksi wajah merupakan salah satu topik yang banyak dipelajari pada bidang *Computer Vision*. Deteksi wajah merupakan langkah awal dalam proses interaksi manusia, *image retrieval*, *image tracking*, pengenalan wajah dan lain-lain. Algoritma *Viola Jones* merupakan algoritma standar untuk deteksi wajah. Algoritma ini terdiri dari tahap *integral image*, *haar like feature*, *adaboost classifier*, dan *cascade classifier*. Hasil dari penelitian ini berupa sebuah sistem yang dapat melatih dataset, hasil dari pelatihan data yang berupa *classifier* dapat disimpan dan digunakan untuk pengujian menggunakan citra yang dapat diimport dari direktori komputer. Akurasi pada sistem memiliki rata-rata akurasi sebesar 83.7% dan rata-rata waktu pengujian selama 1 menit 13.2 detik.

Kata kunci: Deteksi wajah, *Viola Jones*, Citra, *Computer Vision*..

I. Pendahuluan

1.1 Latar Belakang

Teknologi biometrik merupakan suatu metode untuk mengidentifikasi seseorang berdasarkan ciri khusus pada diri atau karakteristik perilaku seseorang. Pada diri manusia terdapat beberapa bagian tubuh yang bisa membedakan dengan orang lain meskipun dengan saudara kembarnya. Ciri khusus tersebut bisa berupa wajah, retina mata, sidik jari, suara maupun bagian tubuh lain yang menjadi identitas pribadi yang bisa diinterpretasikan menjadi suatu parameter atau data yang mempunyai nilai kuantitatif. Identitas tersebut bisa dideteksi dan dikenali dengan memanfaatkan teknologi yang berkembang sekarang ini salah satunya adalah deteksi wajah.[1].

Salah satu contoh pemanfaatan teknologi biometrik yaitu KTP Elektronik (e-KTP). Kartu Tanda Penduduk (KTP) semula hanya menyimpan biodata diri saja, namun kini dengan adanya perkembangan teknologi, KTP dikembangkan menjadi e-KTP. Dengan adanya chip pada e-KTP, data yang dimuat di dalam e-KTP dapat berupa

biodata diri, foto wajah, citra tanda tangan dan dua sidik jari telunjuk kanan dan kiri. Pendeteksian wajah dari sebuah citra merupakan topik yang penting dan menarik saat ini. Dewasa ini teknologi pengenalan wajah semakin banyak diaplikasikan, antara lain untuk sistem presensi, fitur *face unlock* pada *smartphone*, hingga fitur *auto-tagging* pada layanan media sosial Facebook. Kepopuleran teknologi pengenalan wajah ini muncul karena wajah merupakan alat pengenalan seseorang yang mudah dilihat secara visual. Dalam kehidupan sehari-hari, manusia lebih mudah mengenali seseorang berdasarkan wajahnya dibandingkan sidik jari maupun iris matanya.

Menurut [2] dalam bidang penelitian pemrosesan wajah, pendeteksian wajah merupakan salah satu tahap awal yang sangat penting di dalam proses pengenalan wajah. Proses deteksi keberadaan wajah ini menjadi dasar dari proses pengenalan wajah. Tujuan dari proses deteksi wajah adalah untuk mengetahui apakah terdapat wajah di dalam suatu citra, yang kemudian menemukan letak keberadaan suatu citra wajah. Secara umum,

metode deteksi wajah awalnya dikembangkan berdasarkan pengetahuan, karakteristik wajah dan template matching. Namun, kelemahan metode tersebut adalah sensitif terhadap cahaya, akurasi rendah dan diperlukan transformasi ukuran wajah yang dideteksi.

Saat ini, penelitian mengenai pengenalan wajah dengan cepat berkembang. Aplikasi komersial tentang ini telah banyak diimplementasikan namun pada dasarnya teknologi ini belum sempurna. Penelitian perlu terus dikembangkan untuk memperoleh hasil yang diinginkan. Satu hal yang dapat ditambah atau diimplementasikan dalam pengembangan teknologi pengenalan wajah yaitu dengan menambah tingkat kecepatan dan akurasi dalam pendeteksian wajah. Banyak dari sistem pendeteksian tersebut menggunakan metode *Viola-Jones* sebagai metode pendeteksi objek. Metode *Viola-Jones* dikenal memiliki kecepatan dan keakuratan yang cukup tinggi karena menggabungkan beberapa konsep menjadi sebuah metode utama untuk mendeteksi objek [3].

Penelitian lain yang dilakukan oleh Kirana, C. dan Isnanto, B., (2018) yang berjudul *Face Identification For Presence Applications Using Viola-Jones and Eigenface Algorithm*. Penelitian tersebut membahas bagaimana algoritma *Viola-Jones* dan algoritma *Eigenface* digunakan untuk proses identifikasi wajah dalam aplikasi presensi. Algoritma *Eigenface* digunakan untuk melakukan identifikasi citra wajah yang terdeteksi dari suatu citra wajah dengan menggunakan Principal Component Analysis (PCA). Setelah dilakukan pengujian, aplikasi presensi wajah yang diusulkan mampu menghasilkan tingkat akurasi sebesar 90,90%. Berdasarkan uraian di atas, penelitian ini menggunakan algoritma *Viola Jones*. dan *Adaboost Cascade Classifier* untuk mendeteksi citra wajah yang diimplementasikan pada sistem deteksi wajah.

II. Tinjauan Pustaka

2.1. Landasan Teori

a. Citra

Menurut [5] citra (*image*) adalah gambar pada bidang dwimatra (dua dimensi). Ditinjau dari sudut pandang matematis, citra merupakan suatu fungsi menerus (*continue*) dari intensitas cahaya pada bidang dwimatra. Sumber cahaya menerangi objek, objek memantulkan kembali sebagian dari berkas cahaya tersebut. Pantulan cahaya ini di tangkap oleh alat-alat optik, misalnya mata pada

manusia, kamera, pemindai (*scanner*), dan sebagainya, sehingga bayangan objek yang disebut citra tersebut terekam.

b. Wajah

Menurut [6] wajah merupakan bagian dari tubuh manusia yang menjadi fokus perhatian di dalam interaksi sosial, wajah memainkan peranan vital dengan menunjukkan identitas dan emosi. Kemampuan manusia untuk mengetahui seseorang dari wajahnya sangat luar biasa. Kita dapat mengenali ribuan karena frekuensi interaksi yang sangat sering ataupun hanya sekilas bahkan dalam rentang waktu yang sangat lama. Bahkan kita mampu mengenali seseorang walaupun terjadi perubahan pada orang tersebut karena bertambahnya usia atau pemakaian kacamata atau perubahan gaya rambut. Oleh karena itu wajah digunakan sebagai organ dari tubuh manusia yang dijadikan indikasi pengenalan seseorang atau *face recognition*.

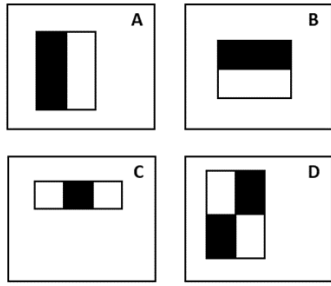
c. Viola Jones

Prosedur deteksi wajah *Viola-Jones* mengklasifikasikan gambar berdasarkan pada nilai fitur sederhana. Terdapat banyak alasan untuk menggunakan fitur daripada piksel secara langsung. Alasan yang paling umum adalah bahwa fitur dapat digunakan untuk mengkodekan pengetahuan *domain ad-hoc* yang sulit dalam pembelajaran terhadap data latih yang terbatas jumlahnya. Alasan penting kedua untuk menggunakan fitur adalah sistem fitur berbasis operasi jauh lebih cepat daripada sistem berbasis pixel [3].

Pada proses pendeteksian wajah dengan menggunakan metode *Viola-Jones*, terdapat beberapa proses yang dilakukan sebelum menghasilkan sebuah *output* wajah yang terdeteksi pada sebuah citra, proses-proses tersebut meliputi *Haar-Like Feature*, *Integral Image*, *Adaboost (Adaptive Boosting)*, dan *Cascade Classifier*. Adapun detail dari setiap tahap yang dilalui dari sebuah citra saat proses pendeteksian wajah menggunakan algoritma *Viola-Jones* sebagai berikut:

d. Haar-like feature

Klasifikasi gambar dilakukan berdasarkan nilai dari sebuah fitur. Penggunaan fitur dilakukan karena pemrosesan fitur berlangsung lebih cepat dibandingkan pemrosesan citra per-piksel. Terdapat tiga jenis fitur berdasarkan jumlah persegi panjang yang terdapat di dalamnya, seperti yang dapat dilihat pada Gambar 2.1.



Gambar 2.1 Haar Like Feature [3]

e. Integral Image

Proses *integral image* digunakan untuk menghitung hasil penjumlahan nilai piksel pada daerah yang dideteksi oleh fitur *haar*. Nilai-nilai piksel yang akan dihitung merupakan nilai-nilai piksel dari sebuah citra masukan yang dilalui oleh fitur *haar* pada saat pencarian fitur pada citra wajah.

f. Adaboost (Adaptive Boosting)

Setelah sebelumnya dilakukan pemilihan fitur *haar*, pada proses selanjutnya yaitu deteksi kembali fitur pada citra menggunakan *Adaboost*. Teknik ini merupakan teknik yang digunakan untuk mengkombinasikan banyak *classifier* lemah untuk membentuk suatu gabungan *classifier* yang lebih baik. Proses *Adaboost* akan menghasilkan sebuah *classifier* yang kuat dari *classifier* dasar. Tujuan dari proses ini untuk mengetahui apakah ada fitur wajah pada daerah dengan klasifikasi fitur yang lemah. Pada *classifier* lemah akan dilakukan penghitungan dan perbandingan dengan *classifier* lainnya secara acak. Selanjutnya dilakukan kombinasi pada *classifier* lemah untuk membentuk suatu kombinasi yang linier.

g. Cascade Classifier

Pada proses *Cascade Classifier*, hasil dari fitur-fitur yang diperoleh dari proses *Adaboost* akan dilakukan proses organisasi dengan bentuk klasifikasi bertingkat.

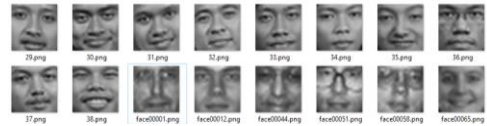
III. Pembahasan

3.1. Objek Penelitian

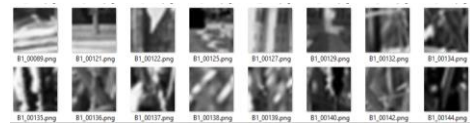
Objek yang digunakan dalam penelitian ini adalah wajah. Menggunakan data primer yaitu wajah mahasiswa Universitas Teknologi Yogyakarta khususnya jurusan Teknik Informatika kelas D angkatan 2015 dan *dataset* berupa wajah dan bukan wajah yang bersumber dari *The Center for Biological & Computational Learning* di *Massachusetts Institute of Technology (CBCL MIT)* yang diunduh dari situs

<https://github.com/INVASIS/VIOLA-Jones/tree/master/data>.

Kedua jenis *dataset* tersebut memiliki ukuran 81 x 81 *pixel* dengan format *png*. Total *dataset* yang digunakan adalah 275 dimana 80% dari *dataset* akan digunakan untuk pelatihan dan 20% dari *dataset* akan digunakan sebagai pengujian. Sampel Citra wajah dapat dilihat pada Gambar 3.1. Sedangkan sampel citra bukan wajah dapat dilihat pada Gambar 3.2.



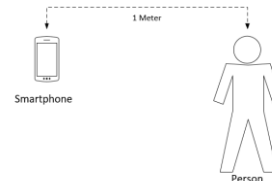
Gambar 3.1 Sampel Citra Wajah



Gambar 3.2 Sampel Citra Bukan Wajah

3.2. Metode Pengumpulan Data

Metode penelitian yang dilakukan penulis dalam melakukan penelitian ini adalah studi pustaka. Penulis menghimpun informasi yang terkait dengan pendeteksi wajah, dan algoritma *Viola-Jones* dari jurnal, karya ilmiah, seminar, dan artikel. Cara memperoleh data primer adalah dengan mengambil citra wajah dari mahasiswa Universitas Teknologi Yogyakarta khususnya jurusan Teknik Informatika kelas D angkatan 2015 menggunakan kamera *smartphone* dengan jarak objek yaitu 1 meter dari kamera *smartphone*.

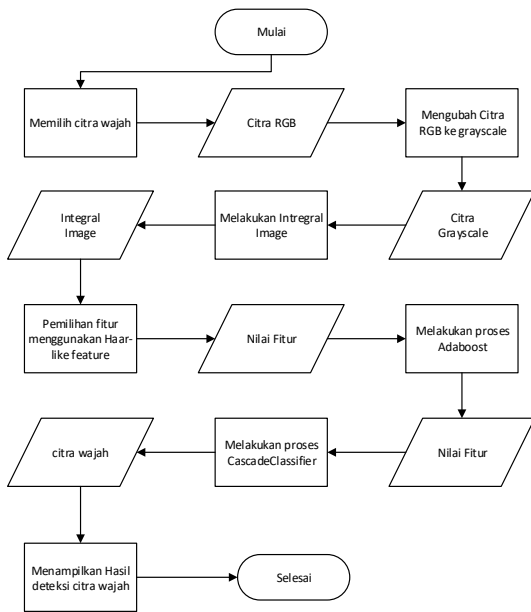


Gambar 3.3 Pengambilan Citra

Setelah dilakukan proses pengambilan citra selanjutnya data citra wajah diedit dengan mengambil *Region of Interest (ROI)* dengan cara *cropping* pada posisi wajah dan dilakukan proses *resize* dengan ukuran 81 x 81 *pixel*. Proses *cropping* dan *resize* dilakukan menggunakan aplikasi Adobe Photoshop CS6.

3.3. Proses algoritma Viola Jones

Skema deteksi wajah pada algoritma *Viola Jones* dapat dilihat pada Gambar 3.4.



Gambar 3.4 Penggambaran Sistem

a. Pemilihan Fitur Haar

Proses pemilihan fitur pada citra *grayscale* yang digunakan untuk proses deteksi citra wajah dapat dilihat pada Gambar 3.5.



Gambar 3.5 Pemilihan Fitur

Sebuah citra masukan yang dilalui oleh fitur haar dapat dilihat pada Gambar 3.6.



Gambar 3.6 Proses Haar Like Feature

Nilai *pixel* yang diperoleh dari *haar like feature* pada Gambar 3.6 dapat dilihat pada Gambar 3.7.

20	18	22	16
15	16	17	20
10	12	9	10
8	6	10	12

Gambar 3.7 Nilai Pixel Haar Like Feature

b. Menghitung Nilai Integral Image

Contoh penghitungan *integral image* dengan citra masukan dari Gambar 3.6 akan dijelaskan pada Tabel 3.1.

Tabel 3.1 Perhitungan *Integral Image*

Posisi	Nilai <i>pixel</i>	Keterangan				
(1,1)	<table border="1"> <tr><td>$i(x,y)=20$</td><td>0</td></tr> <tr><td>0</td><td>0</td></tr> </table>	$i(x,y)=20$	0	0	0	Nilai intensitas <i>pixel</i> adalah 20 atau $i(x,y) = 20$ $s(x-1, y) = 0$ (diluar batas matriks) $s(x, y-1) = 0$ (diluar batas matriks) $s(x-1, y-1) = 0$ (diluar batas matriks) $s(x, y) = i(x,y) + s(x,y-1) + s(x-1,y) - s(x-1,y-1)$ $s(x, y) = 20 + 0 + 0 - 0 = 20$
$i(x,y)=20$	0					
0	0					
(1,2)	<table border="1"> <tr><td>$s(x,y)=20$</td><td>$i(x,y)=18$</td></tr> <tr><td>0</td><td>0</td></tr> </table>	$s(x,y)=20$	$i(x,y)=18$	0	0	$i(x,y) = 18$ $s(x-1, y) = 20$ $s(x, y-1) = 0$ (diluar batas matriks) $s(x-1, y-1) = 0$ (diluar batas matriks) $s(x, y) = i(x,y) + s(x,y-1) + s(x-1,y) - s(x-1,y-1)$ $s(x, y) = 18 + 20 + 0 - 0 = 38$
$s(x,y)=20$	$i(x,y)=18$					
0	0					
(2,1)	<table border="1"> <tr><td>$s(x,y)=20$</td><td>$s(x,y)=38$</td></tr> <tr><td>$i(x,y)=15$</td><td>0</td></tr> </table>	$s(x,y)=20$	$s(x,y)=38$	$i(x,y)=15$	0	$i(x,y) = 15$ $s(x-1, y) = 0$ (diluar batas matriks) $s(x, y-1) = 20$ $s(x-1, y-1) = 0$ (diluar batas matriks) $s(x, y) = i(x,y) + s(x,y-1) + s(x-1,y) - s(x-1,y-1)$ $s(x, y) = 15 + 0 + 20 - 0 = 35$
$s(x,y)=20$	$s(x,y)=38$					
$i(x,y)=15$	0					
(2,2)	<table border="1"> <tr><td>$s(x,y)=20$</td><td>$s(x,y)=38$</td></tr> <tr><td>$s(x,y)=35$</td><td>$i(x,y)=16$</td></tr> </table>	$s(x,y)=20$	$s(x,y)=38$	$s(x,y)=35$	$i(x,y)=16$	$i(x,y) = 16$ $s(x-1, y) = 35$ $s(x, y-1) = 38$ $s(x-1, y-1) = 20$ $s(x, y) = i(x,y) + s(x,y-1) + s(x-1,y) - s(x-1,y-1)$ $s(x, y) = 16 + 38 + 35 - 20 = 69$
$s(x,y)=20$	$s(x,y)=38$					
$s(x,y)=35$	$i(x,y)=16$					

Apabila telah dilakukan perhitungan untuk semua *pixel* yang terdapat dalam kotak-kotak fitur pada Gambar 3.7 akan didapat hasil penghitungan dari *integral image*, hasil perhitungan tersebut dapat dilihat pada Gambar 3.8.

20	18	22	16	20	38	60	76
15	16	17	20	35	69	108	144
10	12	9	10	45	91	139	185
8	6	10	12	53	105	163	221

Citra asli Citra Integral

Gambar 3.8 Hasil Perhitungan *Integral Image*

Setelah didapat hasil dari perhitungan *integral image*, selanjutnya dilakukan penghitungan luas untuk area *Haar Like Feature*.

20	38	60	76
35	69	108	144
45	91	139	185
53	105	163	221

Citra Integral

Gambar 3.9 Menghitung Luas Pada Area *Haar*

Untuk menghitung luas *pixel* pada area D pada Gambar 3.9, maka menggunakan persamaan 3.1.

$$D = L1 + L4 - (L2 + L3) \quad (0.1)$$

Contoh:

$$L1 = 69, L2 = 144, L3 = 105, L4 = 221$$

maka luas *pixel* pada area D adalah:

$$69 + 221 - (144 + 105) = 41$$

Apabila sudah didapat nilai *integral image* dari sebuah citra masukan dan nilai nilai jumlah *pixel* pada area tertentu, maka hasil tersebut akan dihitung antara nilai *pixel* pada area terang dikurangi nilai *pixel* pada area gelap. Jika hasil perhitungan diatas nilai *threshold* maka area tersebut dinyatakan memiliki fitur.

c. Pelatihan Data

Setelah melalui tahap *Haar Like Feature* nilai *pixel* dari citra akan dilatih menggunakan algoritma *Adaptive Boosting (Adaboost)*. Berikut adalah contoh pelatihan data menggunakan algoritma *Adaboost*.

1. Diberikan data latih sebagai berikut:

x	0	1	2	3	4	5	6	7	8	9
y	1	1	1	-1	-1	-1	1	1	1	-1

Keterangan:

X(i): merupakan nilai fitur.

Y(i): -1 untuk citra negatif, +1 untuk citra positif.

m: 10

2. Inisialisasi bobot awal.

$$w(i) = 1/m$$

w	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1
---	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

3. Misal diberikan nilai *threshold* 2.5

$$h_1 = I(x < 2.5)$$

$$x(i) = \begin{cases} 1, & x(i) < 2.5 \\ -1, & x(i) \geq 2.5 \end{cases}$$

x	0	1	2	3	4	5	6	7	8	9
y	1	1	1	-1	-1	-1	1	1	1	-1
w	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1

Untuk $t=2.5$ diperoleh 3 data yang diklasifikasikan salah, yaitu pada nilai citra 6, 7, dan 8.

4. Menghitung nilai error

$$\epsilon_1 = w(6) + w(7) + w(8)$$

$$\epsilon_1 = 0.1 + 0.1 + 0.1$$

$$\epsilon_1 = 0.3$$

5. Menghitung nilai alpha

$$\alpha_1 = \frac{1}{2} \ln \left(\frac{1 - 0.3}{0.3} \right) = 0.423649$$

6. Menghitung nilai c(x)

$$c(x) = \begin{cases} e^{-0.423649} = 0.654654, & \text{untuk nilai True} \\ e^{0.423649} = 1.52753, & \text{untuk nilai False} \end{cases}$$

7. Pre normalisasi

w	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1	0,1
P	0,6	0,6	0,6	0,6	0,6	0,6	0,6	0,6	0,6	0,6

8. Cari nilai Z

$$Z_1 = 0,65 + 0,65 + 0,65 + 0,65 + 0,65 + 0,65 + 0,15 + 0,15 + 0,15 + 0,65$$

$$Z_1 = 0,916515$$

9. Perbarui bobot

$$W_i = P_i / Z_1$$

P	0,6	0,6	0,6	0,6	0,6	0,6	0,6	0,6	0,6	0,6
W	0,7	0,7	0,7	0,7	0,7	0,7	0,7	0,7	0,7	0,7

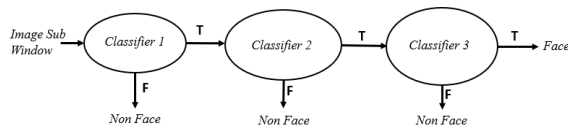
Maka diperoleh *weak classifier* sebagai berikut:

$$f_1(x) = 0.423649 (x < 2.5), 3 \text{ error}$$

Ulangi kembali proses klasifikasi mulai dari tahap ke 3 dengan bobot baru hingga tidak ditemukan error.

d. Klasifikasi Bertingkat

Pada 3.10 merupakan proses rangkaian filter yang dilalui oleh setiap *classifier*.



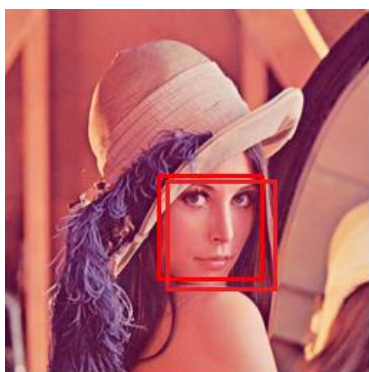
Gambar 3.10 Cascade Classifier

Pada klasifikasi filter pertama, tiap subcitra akan diklasifikasi menggunakan satu fitur. Jika hasil nilai fitur dari filter tidak sesuai dengan nilai *threshold*, maka hasil tersebut ditolak.

Algoritma kemudian bergerak ke filter selanjutnya dan menghitung nilai fitur kembali. Jika didapat hasil sesuai dengan *threshold* yang diinginkan, maka dilanjutkan ke tahap filter selanjutnya hingga jumlah filter yang lolos klasifikasi akan berkurang hingga mendekati citra yang dideteksi.

Pada filter pertama dipilih 4 fitur *classifier* dengan menerima minimal 2 fitur. Pada filter kedua dipilih 6 buah fitur *classifier* dengan menerima 3 fitur. Pada tahap ketiga dipilih 12 fitur *classifier* dengan menerima minimal 6 fitur. Pada tahap keempat dipilih 18 fitur *classifier* dengan menerima minimal 10 fitur, sehingga total minimal fitur yang diperlukan sebuah citra untuk dapat terdeteksi sebagai wajah harus memiliki minimal 21 fitur. Setelah dilakukan proses *Cascade Classifier* akan didapatkan hasil pendeteksian berupa wajah atau bukan wajah.

Setelah dilakukan serangkaian proses, maka akan didapatkan sebuah hasil pendeteksian. Hasil deteksi dapat berupa wajah atau bukan wajah. Pada proses klasifikasi bertingkat, pada citra yang dideteksi sebagai wajah akan ditandai dengan sebuah *landmark* berbentuk persegi Panjang. Pada Gambar 3.11 merupakan contoh hasil pendeteksian dari algoritma *Viola-Jones*.



Gambar 3.11 Hasil Pendeteksian

Setelah proses pendeteksian terdapat *landmark* yang *overlap* pada posisi wajah yang sama maka dilakukan proses penghilangan *overlap* agar menghasilkan hanya 1 *landmark* saja yang ada pada posisi wajah.

e. Penghilangan Landmark Yang Overlap

Algoritma untuk proses penghilangan *landmark* dapat ditunjukkan pada Tabel 3.2:

Tabel 3.2 *Overlapping*

```

for i from 1 to N do
  for j from i to N do
    overlapArea =  $S_i \cap S_j$       (3.2)
    totalArea =  $S_i \cup S_j$       (3.3)
    overlapRate[i][j]
    = overlapArea /
    totalArea      (3.4)
  end for
end for
reduced = [1, ..., i, ..., N]
for i from 1 to N do
  for j from i+1 to N do
    if( overlapRate [i] [j] >
    overlapThreshold )
      reduced[j] = reduced[i]
    end for
  end for
return reduced

```

Keterangan:

- S = nilai *landmark* ke i
- overlapArea = area irisan *landmark*
- totalArea = area gabungan *landmark*
- overlapRate = nilai tingkat *overlap*
- reduced = nilai *landmark* setelah dihilangkan *overlap* nya
- overlapThreshold = nilai ambang batas *overlap* (0.1)

Sebagai contoh, misalnya hasil pada pendeteksian pada Gambar diketahui memiliki 2 *landmark* yang diasumsikan sebagai *landmark* A dan *landmark* B dengan nilai piksel pada masing masing *landmark* yang dapat dilihat pada Tabel 3.3.

Tabel 3.3 Nilai Piksel Pada *Landmark*

Landmark A			Landmark B		
50	51	50	55	58	20
53	55	58	53	21	23
30	80	21	10	24	22

Maka untuk menghitung area *overlap* dari *landmark* A dan *landmark* B dengan persamaan 3.2 yaitu:

$A = \{50, 51, 50, 53, 55, 58, 30, 80, 21\}$
 $B = \{55, 58, 20, 53, 21, 23, 10, 24, 22\}$
 Maka $A \cap B = \{21, 58, 53, 55\}$

Sedangkan untuk menghitung total area dari *landmark* A dan *landmark* B dapat menggunakan persamaan 3.3 yaitu:

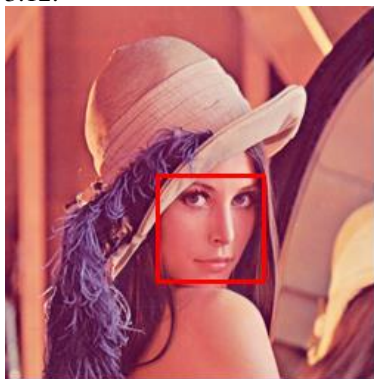
$A = \{50, 51, 50, 53, 55, 58, 30, 80, 21\}$
 $B = \{55, 58, 20, 53, 21, 23, 10, 24, 22\}$
 Maka $A \cup B = \{10, 80, 50, 51, 20, 53, 21, 55, 23, 24, 58, 22, 30\}$

Untuk menghitung *overlap rate* maka masing-masing hasil dari tiap-tiap nilai area

overlap dan total area dicari nilai luasnya yaitu dengan cara menjumlahkan nilainya kemudian menghitung *overlap rate* dengan Persamaan 3.4 yaitu:

$$\begin{aligned} \text{overlapArea}(i) &= 21 + 58 + 53 + 55 \\ \text{overlapArea}(i) &= 187 \\ \text{totalArea}(i) &= 10+80+50+51+20+53+ 21+55+23+ \\ &24+58+22+30 \\ \text{totalArea}(i) &= 497 \\ \text{overlapRate}(i) &= 187/497 \\ \text{overlapRate}(i) &= 0.3762 \end{aligned}$$

Karena nilai *overlapRate*(0.3762) lebih besar dari *overlapThreshold* (0.1) maka *landmark A* dinyatakan *overlap* dengan *landmark B*, maka nilai dari *landmark A* di ganti dengan *landmark B*. Sehingga hasil deteksi citra wajah setelah melalui proses penghilangan *overlap* akan menghasilkan 1 *landmark* pada citra yang dapat dilihat pada Gambar 3.12.

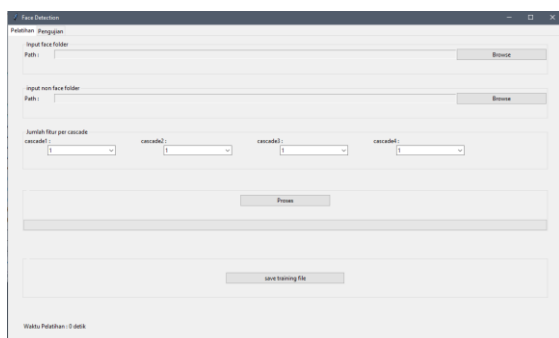


Gambar 3.12 Hasil Penghilangan *Landmark*

IV. Implementasi

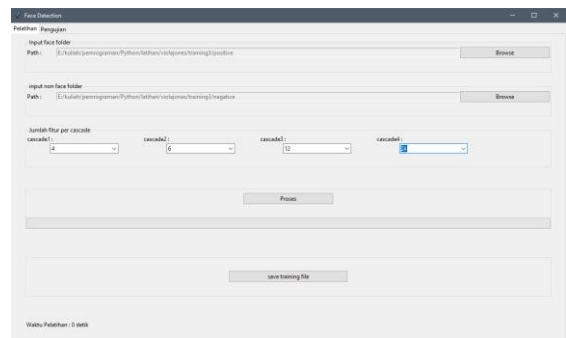
Sistem yang dibuat berbasis desktop menggunakan bahasa pemrograman *Python*, *framework OpenCV* dan *Tkinter*. *Framework OpenCV* digunakan untuk mengubah gambar ke citra *grayscale*, yang selanjutnya akan digunakan untuk proses deteksi wajah, sedangkan *Framework Tkinter* digunakan untuk membangun antar muka pengguna.

Pada Gambar 4.1 merupakan tampilan bagian pelatihan data sistem deteksi citra wajah menggunakan algoritma *viola jones*.



Gambar 4.1 Halaman Pelatihan Data

Untuk pelatihan data pengguna harus menambahkan lokasi folder yang menyimpan citra wajah dan citra bukan wajah. Pelatihan data dilakukan menggunakan fitur 4 pada *cascade* pertama, 6 fitur pada *cascade* ke dua, 12 fitur pada *cascade* ke tiga, dan 18 fitur pada *cascade* ke empat. Pada Gambar 4.2 adalah tampilan sistem ketika sudah di tambahkan folder yang terdapat citra wajah, citra bukan wajah.

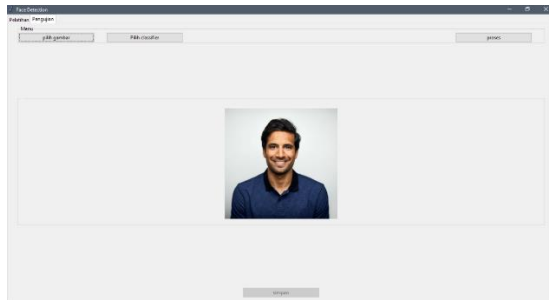


Gambar 4.2 Input Pelatihan Data

Setelah menekan tombol proses maka sistem akan memulai pelatihan data. Status pelatihan data dapat di lihat pada *pogresbar* yang terletak pada bawah tombol proses. Ketika sudah selesai pelatihan data akan ditampilkan informasi bahwa pelatihan data telah berhasil.

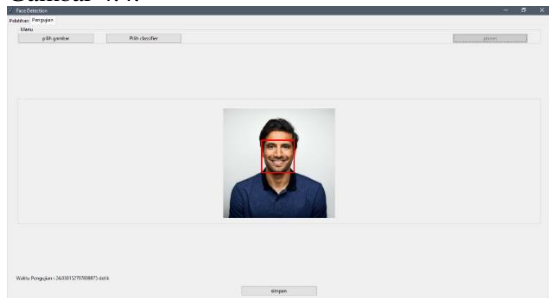
Hasil pelatihan data dengan menggunakan 80% dari *dataset* yaitu sebanyak 90 citra wajah dan 130 citra bukan wajah dengan jumlah 4 fitur pada *cascade* pertama, 6 fitur pada *cascade* ke dua, 12 fitur pada *cascade* ke tiga, dan 18 fitur pada *cascade* ke empat ini memakan waktu selama 1 jam lebih 23,3 menit. Hasil pelatihan berupa *classifier* kuat disimpan dengan mekan tombol *save training file* yang dalam direktori komputer. Hasil pelatihan tersebut memiliki format *file csv*.

Proses pengujian data dari sistem deteksi citra wajah menggunakan algoritma *viola jones* terletak pada *tab* pengujian. Untuk melakukan proses pengujian dari citra wajah yang akan diuji, pengguna diharuskan memilih citra wajah dari direktori komputer dengan menekan tombol pilih gambar yang kemudian citra yang dipilih akan di tampilkan. Kemudian memilih hasil pelatihan data yang ada di direktori dengan menekan tombol pilih *classifier*. Untuk memulai proses pengujian, pengguna menekan tombol *process*. Tampilan halaman pengujian data dapat dilihat pada Gambar 4.3.



Gambar 4.3 Halaman Pengujian Data

Hasil pengujian tersebut dapat dilihat pada Gambar 4.4.



Gambar 4.4 Hasil pengujian

Setelah dilakukan pengujian data maka akan dihasilkan *landmark* berbentuk persegi berwarna merah pada lokasi citra yang dideteksi sebagai wajah. Untuk menyimpan gambar hasil deteksi, pengguna dapat menekan tombol simpan yang terletak pada bagian bawah tab pengujian.

4.1 Pengujian

Proses pengujian data dilakukan untuk mengetahui hasil dari penelitian sistem deteksi wajah menggunakan algoritma *Viola Jones* dan *Adaboost Cascade Classifier* yang dibuat berhasil atau tidak. Pengujian dilakukan sebanyak 2 kali. Pengujian pertama menggunakan data wajah yang sudah melalui proses *cropping*, sedangkan pada pengujian kedua menggunakan data asli, yaitu data wajah yang belum melalui proses *cropping*. Pengujian pertama menggunakan 20% data dari *dataset* yang berjumlah 55 data yang terdiri dari 30 citra wajah dan 25 citra bukan wajah yang memiliki ukuran 81x81 piksel. Pengujian kedua menggunakan sebanyak 40 data yang diunduh dari <http://gettyimages.com> dengan ukuran dan jumlah wajah yang berbeda pada masing-masing *file* citra. Pengujian pertama dilakukan menggunakan *dataset* dapat dilihat pada Tabel 4.1 dan Tabel 4.2.

Tabel 4.1 Pengujian *Dataset*

Jumlah data	Waktu pengujian	Akurasi
55	0.23 detik	76.36%

Tabel 4.2 *Confusion Matrix Dataset*

		Predicted Values	
		Positive	Negative
Actual Values	Positive	TP = 20	FP = 10
	Negative	FN = 3	TN = 22
Akurasi		76.36%	
Precision		66.67%	
Recall		86.96%	

Pengujian kedua dilakukan menggunakan data asli yang ditunjukkan pada Tabel 4.3 dan Tabel 4.4.

Tabel 4.3 Pengujian Data Asli

Jumlah data	Waktu pengujian	Akurasi
40	1 menit 13,2 detik	91.04%

Tabel 4.4 *Confusion Matrix Data Asli*

		Predicted Values	
		Positive	Negative
Actual Values	Positive	TP = 55	FP = 0
	Negative	FN = 6	TN = 6
Akurasi		91.04%	
Precision		100%	
Recall		90.16%	

Berdasarkan hasil dua pengujian diatas, dapat diringkas kedalam tabel yang ditunjukkan pada Tabel 4.5 sebagai berikut:

Tabel 4.5 Hasil Pengujian Data

Jenis data	Jumlah data	Waktu pengujian	Akurasi	Recall	Precision
Dataset	55	0.23 detik	76.36 %	86.96 %	66.67 %
Data asli	40	1 menit 13,2 detik	91.04 %	90.16 %	100%

V. Penutup

5.1. Kesimpulan

Berdasarkan implementasi dan pengujian sistem yang telah dilakukan pada sistem deteksi wajah menggunakan algoritma *Viola Jones* dan *Adaboost Cascade Classifier*, diperoleh kesimpulan sebagai berikut:

1. Algoritma *Viola Jones* dan *Adaboost Cascade Classifier* merupakan algoritma yang dapat digunakan untuk mendeteksi citra wajah.
2. Lama proses pendeteksian wajah bergantung pada ukuran *pixel* pada *file* citra. Semakin besar ukuran *pixel* pada *file* citra maka akan semakin lama proses deteksi *file* citra.

3. Waktu pelatihan data menggunakan 90 citra wajah dan 130 citra bukan wajah dengan jumlah 4 fitur pada *cascade* pertama, 6 fitur pada *cascade* ke dua, 12 fitur pada *cascade* ke tiga, dan 18 fitur pada *cascade* ke empat memerlukan waktu pelatihan selama 1 jam lebih 23,3 menit.
4. Hasil pengujian menghasilkan rata-rata akurasi sebesar 76.36% pada citra yang sudah melalui proses *cropping* dengan rata-rata waktu pengujian selama 0.23 detik. Sedangkan pada citra asli menghasilkan rata-rata akurasi sebesar 91.04% dengan rata-rata waktu pengujian selama 1 menit 13.2 detik.

Yogyakarta, 19 Juni 2010, vol. Vol 1, no. Snati, hal. Pengembangan Perangkat Lunak Pengenalan Citra Waja, 2010.

5.2. Saran

Dari semua uraian yang telah dibahas maka dapat diberikan saran yaitu:

1. Untuk mendapatkan hasil akurasi yang lebih baik maka dibutuhkan data latih yang lebih banyak.
2. Waktu proses pelatihan data dan pengujian data memakan waktu yang cukup lama. Maka diperlukan optimasi *software* dan *hardware* untuk mempercepat dua proses tersebut.

DAFTAR PUSTAKA

- [1] S. P. Shulur, "Perancangan Aplikasi Deteksi Wajah Menggunakan Algoritma Viola-Jones," Universitas Pasundan Bandung, 2015.
- [2] A. Chen, L. Pan, Y. Tong, dan N. Ning, "Face detection technology based on skin color segmentation and template matching," in *2nd International Workshop on Education Technology and Computer Science, ETCS 2010*, 2010.
- [3] P. Viola dan M. Jones, "Rapid Object Detection using a Boosted Cascade of Simple Features," *Comput. Vis. Pattern Recognit.*, vol. 18, no. 10, 2001.
- [4] C. Kirana dan B. Isnanto, "Face Identification For Presence Applications Using Violajones and Eigenface Algorithm," *J. Sisfokom (Sistem Inf. dan Komputer)*, 2018.
- [5] R. Munir, *Pengolahan Citra Digital Dengan pendekatan Algoritmik*. Bandung: Informatika Bandung, 2004.
- [6] N. W. Marti, "Pemanfaatan GUI Dalam Pengembangan Perangkat Lunak Pengenalan Citra Wajah Manusia Menggunakan Metode Eigenface," *Semin. Nas. Apl. Teknol. Inf. 2010 (SNATI 2010)*