

# HAAR CASCADE CLASSIFIER DAN ALGORITMA ADABOOST UNTUK DETEKSI BANYAK WAJAH DALAM RUANG KELAS

Hadi Santoso<sup>1</sup>, Agus Harjoko<sup>2</sup>

<sup>1</sup>Program Studi Sistem Informasi STIMIK Atma luhur

(Mahasiswa Program Studi S3 Ilmu Komputer Universitas Gadjah Mada  
Jl. Raya Sungaliat Selindung Baru – Pangkalpinang Telp (0717) 4333506  
E-Mail : hadimkom@gmail.com, hadisantoso,atmaluhur@gmail.com

<sup>2</sup>Program Studi Elektronika dan Instrumentasi, FMIPA Universitas Gadjah Mada  
Sekip Utara Yogyakarta

E-mail : aharjoko@ugm.ac.id, aharjoko@gmail.com

## ABSTRACT

Face detection is one of the important technological developments in computer vision field such as security systems, control systems, including the presence or attendance system. The proposed system leads to automatic attendance system but only to the extent that there is face detection student in the class based on the position of the face is straight ahead position, rotation parallel to right 15°, 15° rotation aligned to the left, lift the chin 15°, 15° head up and down and based on three distance face objects, in example 100 cm, 150 cm and 200 cm, and calculate the number of students that are in class. Based on this background we proposed multiple-face detection system in the classroom. The method used to detect faces using AdaBoost algorithms and applications written in the programming language C# in Visual Studio 2008. Face image taken from a webcam using EmguCV capture library functions. Tests carried out in the normal light intensity, in addition, the accuracy is also measured in terms of the real face are detected directly from the webcam. From the testing that has been done seven facial image results obtained with the three position and distance of the object can be detected with a good face of 8 face images in the classroom due to obstructed by other students.

*Keywords: AdaBoost algorithm, EmguCV, haar cascade classifier, multiple face detection*

## INTISARI

Deteksi wajah merupakan salah satu perkembangan teknologi yang penting dalam bidang computer vision seperti sistem sekuriti, sistem kontrol, termasuk sistem kehadiran atau presensi. Sistem yang diusulkan ini mengarah kepada sistem absensi secara otomatis namun hanya sebatas deteksi wajah mahasiswa yang ada di kelas berdasarkan posisi wajah yaitu posisi lurus ke depan, rotasi sejajar 15° kekanan, rotasi sejajar 15° kekiri, mengangkat dagu 15° keatas dan menunduk kepala 15° dan berdasarkan tiga jarak objek wajah, yaitu 100 cm, 150 cm dan 200 cm serta menghitung banyaknya mahasiswa yang ada didalam kelas. Berdasarkan latarbelakang tersebut diusulkan suatu sistem pendeteksi banyak wajah dalam ruang kelas. Metode yang digunakan untuk mendeteksi wajah tersebut menggunakan algoritma AdaBoost dan aplikasi ditulis dengan bahasa pemrograman C# pada Visual Studio 2008. Citra wajah di ambil dari webcam dengan menggunakan fungsi capture library EmguCV. Ujicoba dilakukan dalam intensitas cahaya normal, selain itu akurasi juga diukur dalam hal jumlah wajah yang sebenarnya terdeteksi langsung dari webcam. Dari pengujian yang telah dilakukan didapatkan hasil 7 citra wajah dengan posisi dan tiga jarak objek wajah dapat dideteksi dengan baik dari 8 citra wajah di dalam ruang kelas dikarenakan terhalangi oleh mahasiswa yang lain.

Kata Kunci : algoritma AdaBoost, EmguCV, haar cascade classifier, deteksi banyak wajah

## PENDAHULUAN

Perkembangan teknologi yang terkait dengan pemrosesan menggunakan komputer sudah semakin pesat, dimana pemrosesan *image* (citra) sudah menggunakan teknologi digital. Deteksi wajah manusia merupakan salah satu domain dalam aplikasi *computer vision*.

Deteksi wajah didasarkan pada identifikasi dan menemukan lokasi citra wajah manusia dalam gambar terlepas dari ukuran, posisi, dan kondisi (Padmaja & Prabakar, 2012). Pengenalan wajah termasuk kedalam perkembangan teknologi yang menjadi penting dalam beberapa bidang seperti

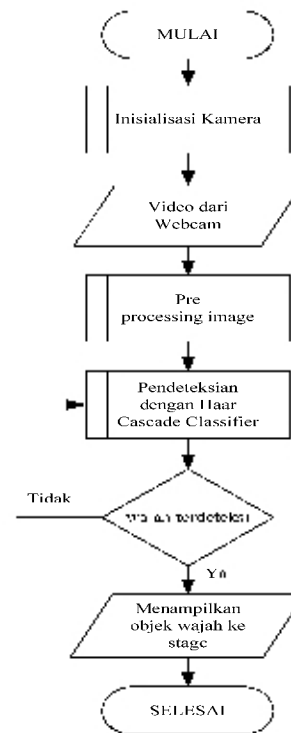
sistem sekuriti, sistem kontrol, termasuk sistem kehadiran atau presensi.

Kehadiran mahasiswa dalam perkuliahan sangat diperlukan karena termasuk dalam sistem penilaian di suatu institusi yang menyelenggarakan pendidikan. Setiap Institusi Perguruan Tinggi punya cara tersendiri dalam menerapkan metode penilaian kehadiran seorang mahasiswa. Permasalahan yang sering terjadi dalam pendeteksian wajah adalah mengenai oklusi wajah, pose dan faktor pencahayaan. Banyak sistem biometrik dapat diterapkan pada sistem absensi, tapi semuanya kebanyakan menggunakan teknik autentifikasi yang sama (Zhang et al. 2009). Setiap sistem biometrik khususnya pada proses absensi bersifat unik dimana data secara pribadi disimpan kedalam basis data dan terdapat proses identifikasi dan verifikasi. Artinya ada dua proses perbandingan ciri biometrik dari seseorang dengan *template* basis data sebelumnya yang diambil pada saat proses training. *Template* biometrik dapat berupa sidik jari, iris mata, wajah, alur tangan, tanda tangan dan suara. Viola-Jones mengusulkan kerangka kerja deteksi wajah berdasarkan algoritma pembelajaran AdaBoost menggunakan Haar feature (Viola & Jones, 2001). (Lienhart & Maydt, 2002) melakukan penelitian serupa dengan meningkatkan kemampuan untuk melakukan scaling sehingga dapat mendeteksi adanya wajah yang berukuran lebih besar ataupun lebih kecil dari gambar pada *classifier*. Di dalam teknik ini terdapat metode *cascade of classifier*, *haar-like feature classifier* serta algoritma *adaboost*. Metode yang digunakan untuk mendeteksi wajah tersebut menggunakan Algoritma AdaBoost (Schneiderman & Kanade, 1998) serta (Viola & Jones, 2004).

Berdasarkan latar belakang tersebut diusulkan sebuah sistem yang mengarah pada sistem absensi secara otomatis, namun hanya sebatas deteksi wajah mahasiswa yang ada di kelas berdasarkan hasil ukuran citra dan menghitung banyaknya mahasiswa yang ada didalam kelas, dengan tujuan menghitung dan mengidentifikasi mahasiswa di ruang kelas secara real time dengan menggunakan webcam.

#### METODE YANG DIUSULKAN

Diagram dari metode yang digunakan untuk melakukan deteksi citra banyak wajah manusia dalam ruang kelas dapat dilihat pada Gambar 1.



Gambar 1. Flowchart Proses Training

#### Inisialisasi Kamera

Inisialisasi adalah proses penentuan awal semua hal yang diperlukan untuk menjalankan proses selanjutnya. Pada inisialisasi ini, GUI diinisialisasikan terlebih dahulu untuk mendapatkan tampilan elemen-elemen interface, sehingga memudahkan penggunaan bagi user. Inisialisasi dari kamera mulai dari menginisialisasi video masukan sampai pemrosesan image. Face detector akan mengatur hal-hal yang berkaitan dengan pendeteksian wajah. Pada proses ini citra wajah ditangkap dari webcam. Dimana resolusi yang digunakan adalah 320 x 240 piksel. Citra wajah yang diambil untuk citra latih (*training*) berdasarkan posisi wajah yaitu posisi tegak lurus, rotasi sejajar 15° kekanan, rotasi sejajar 15° kekiri, mengangkat dagu 15° keatas dan menunduk kepala 15° serta berdasarkan tiga jarak objek wajah, yaitu 100 cm, 150 cm dan 200 cm.

#### Pre-Processing Image

Pada tahap pre-processing image dilakukan proses scaling, grayscale, dan thresholding untuk menjadi inputan selanjutnya dalam metode Haar Cascade.

#### Tahap Scaling

Scaling merupakan proses mengubah ukuran citra digital, hal ini perlu dilakukan

agar semua citra digital memiliki ukuran yang sama. Untuk melakukan scaling di sistem ini dapat memanfaatkan package display object. Citra digital tersebut diperkecil dengan menggunakan metode interpolasi. Metode ini menggunakan rata-rata suatu region untuk mewakili region tersebut. Berikut ini adalah citra hasil interpolasi:



Gambar 2. Penskalaan citra

### Tahap Grayscale

Citra digital yang telah melalui proses penskalaan kemudian diubah menjadi citra dua warna dengan proses *grayscale*. Proses perubahan citra RGB menjadi citra *grayscale* adalah sebagai berikut :

Misalkan suatu citra wajah memiliki nilai :

R = 152  
G = 132  
B = 133

Maka nilai *grayscale* dari citra tersebut dapat dihitung seperti di bawah ini :

$$W = 0,2989R + 0,5870G + 0,1140B$$

$$(1)$$

$$w = (0,2989 \times 152) + (0,5870 \times 132) + (0,1140 \times 133) = 138,0788$$

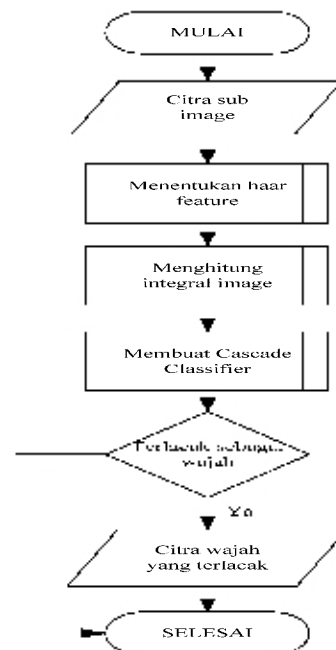
Berikut ini adalah citra hasil *grayscale* :



Gambar 3. Perubahan citra RGB menjadi citra *grayscale*

### Pendeteksian dengan Metode Haar Cascade Classifier

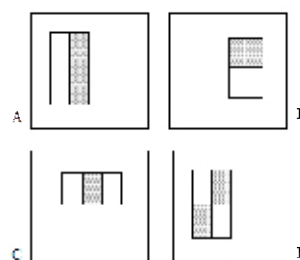
Wajah merupakan salah satu bagian dari manusia yang memiliki ciri berbeda untuk setiap manusia. Pada penelitian ini digunakan metode haar cascade classifier sebagai metode untuk pengenalan pola wajah. Berikut ini alur proses metode haar cascade classifier :



Gambar 4. Flowchart Metode Haar Cascades Classifier

### Proses Menentukan Haar Feature

Metode ini menggunakan haar-like features dimana perlu dilakukan training terlebih dahulu untuk mendapatkan suatu pohon keputusan dengan nama cascade classifier sebagai penentu apakah ada obyek atau tidak dalam tiap frame yang di proses. Terdapat tiga jenis fitur berdasarkan jumlah persegi panjang yang terdapat di dalamnya (Krishna & Srinivasulu, 2012), seperti yang dapat dilihat pada gambar di bawah ini :



Gambar 5. Haar-like feature

Setelah melakukan penskalaan pada sebelumnya kemudian dilakukan pencarian posisi wajah yaitu dengan cara mencari fitur-fitur yang memiliki tingkat pembeda yang tinggi. Hal ini dilakukan dengan mengevaluasi setiap fitur terhadap data latih dengan menggunakan nilai dari fitur tersebut. Fitur yang memiliki batas terbesar antara wajah dan bukan wajah dianggap sebagai fitur terbaik.



Gambar 6. Contoh pencarian dengan haar-like feature

#### Proses Menghitung *Integral Image*

Pada gambar 5. dapat dilihat bahwa fitur A dan B terdiri dari dua persegi panjang. Cara menghitung nilai dari fitur ini adalah mengurangkan nilai piksel pada area hitam dengan piksel pada area putih. Jika nilai perbedaannya itu di atas nilai ambang (*threshold*), maka dapat dikatakan bahwa fitur tersebut ada. Untuk menentukan ada atau tidaknya dari ratusan fitur *haar* pada sebuah citra dan pada skala yang berbeda secara efisien, digunakan satu teknik yang disebut *integral image*.

Citra integral (*integral image*) adalah sebuah citra yang nilai tiap pikselnya merupakan akumulasi dari nilai piksel atas dan kirinya. Sebagai contoh, piksel (a,b) memiliki nilai akumulatif untuk semua piksel (x,y) dimana  $x \leq a$  dan  $y \leq b$ . Salah satu contoh seperti perhitungan berikut :



Gambar 7. Sampel perhitungan *integral image*

1	2	3	1	2	3	1	1+2	1+2+3
2	3	4	2	3	4	1+2	1+2+2	1+2+3
4	5	1	4	5	1	+3	+2+3+	4
						1+2	1+2+2	1+2+3
						+4	+3+4+	+2+3+
						5	4+4+5	

Maka hasil perhitungan citra integralnya adalah :

1	3	6
3	8	15
7	17	25

Nilai fitur = |(total piksel hitam)– (total piksel putih)|

$$= |[6+0-(0+0)]+[25+3-(15+0)]- [15+1-(6+0)]|$$

$$= |6 + 13 - 10|$$

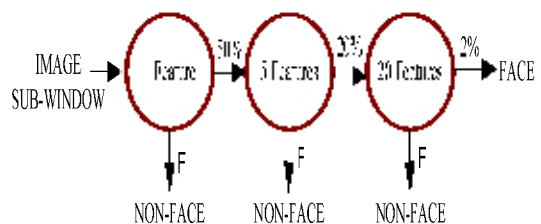
$$= 9$$

Nilai haar feature yang didapatkan di atas yaitu 9 adalah nilai perbedaan antar kotak (fitur putih dan fitur hitam) yang biasa disebut dengan *threshold*. *Threshold* ini digunakan sebagai parameter klasifikasi objek yang terdeteksi sebagai wajah atau tidak.

#### Proses Membuat *Cascade Classifier*

Algoritma pembelajaran AdaBoost digunakan sebagai tahap awal untuk deteksi wajah menggunakan data citra wajah (Choi et al. 2011). *Haar-like feature* mempunyai sifat *learner* dan *classifier* yang lemah. Jika ingin mendapatkan hasil yang lebih akurat maka harus dilakukan proses *haar-like feature* secara massal, semakin banyak proses *haar-like feature* yang dilakukan maka akan semakin akurat hasil yang dicapai. Oleh karena itu pemrosesan *haar-like feature* yang banyak tersebut diorganisir di dalam *cascade classifier*.

Di bawah ini adalah alur kerja dari klasifikasi bertingkat.



Gambar 8. Proses *Cascade Classifier*

Pada klasifikasi filter pertama, tiap subcitra akan diklasifikasi menggunakan satu fitur. Jika hasil nilai fitur dari filter tidak memenuhi kriteria yang diinginkan, hasil ditolak.

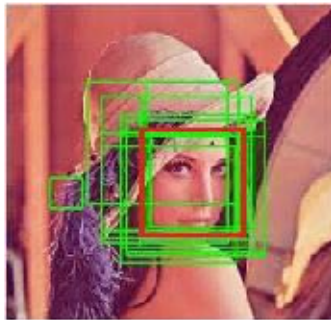
Algoritma kemudian bergerak ke sub window selanjutnya dan menghitung nilai fitur kembali. Jika didapat hasil sesuai dengan *threshold* yang diinginkan, maka dilanjutkan

ke tahap filter selanjutnya. Hingga jumlah sub window yang lolos klasifikasi pun akan berkurang hingga mendekati citra yang ada pada sampel.

Seperti pada gambar 8 di atas, dapat dijelaskan secara keseluruhan proses *cascade classifier*-nya sebagai berikut :

- 1) Filter pertama (*first classifier*)
  - a) 1 fitur
  - b) *Detection rate* : 100%, *false positive rate* : 50%
- 2) Filter kedua (*second classifier*)
  - a) 5 fitur
  - b) *Detection rate* : 100%, *false positive rate* : 80%
- 3) Filter ketiga dan keempat (*third and fourth classifiers*)
  - a) 20 fitur

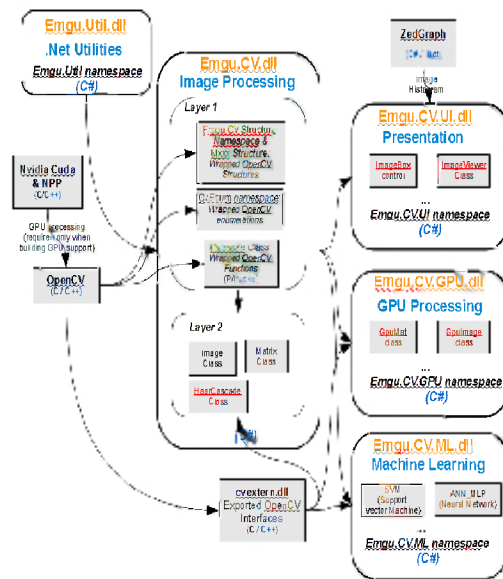
Sedangkan hasil dari filter pada proses *cascade classifier* dapat dilihat pada gambar 9



Gambar 9. Hasil deteksi wajah

### EmguCV

Menurut Bradski & Kaehler (2008) OpenCV adalah *open source Computer Vision Library*. EmguCV adalah lintas platform .Net yang dipaketkan kedalam OpenCV pada proses pengolahan citra. Dengan EmguCV, fungsi-fungsi pada OpenCV dapat dipanggil dengan bahasa pemrograman yang kompatibel dengan .NET seperti C#, VB, VC++, IronPython dan lainnya. Paket dari OpenCV dapat dikompilasi dengan perintah command dan berjalan pada Sistem Operasi Windows, Linux, Mac OS X, iPhone, aplikasi pada iPad dan Android. EmguCV terdiri dari 2 paket layer yang terlihat pada Gambar 10 berikut :



Gambar 10. Paket Layer EmguCV

Keuntungan menggunakan EmguCV yang paling utama adalah *library* ini sepenuhnya ditulis dengan C# yang mana tentunya lebih *safe* karena pembuatan *object* atau pun *reference* di-*manage* oleh *garbage collector*. Dengan Emgu CV, kita dapat membuat aplikasi apapun seperti layaknya menggunakan OpenCV. Namun paling tidak ada dua konsep penting yang perlu diketahui terlebih dahulu sebelum menggunakan Emgu CV. Yang pertama mengenai layer pada Emgu CV. Emgu CV terdiri dari 2 layer, yaitu *basic layer* dan *second layer*. *Basic layer* mengandung fungsi, struktur, dan enumerasi yang secara langsung merefleksikan apa yang ada di OpenCV. Dengan adanya layer inilah kita bisa memanggil fungsi-fungsi pada OpenCV dengan bahasa pemrograman C#. Sedangkan *second layer* mengandung kelas-kelas yang memanfaatkan keunggulan teknologi .NET. Konsep kedua yang perlu dipahami yaitu mengenai *code mapping*, yaitu bagaimana kode-kode dalam OpenCV dipetakan ke dalam Emgu CV. Ada 3 tipe *mapping* kode, yaitu *function mapping*, *structure mapping*, dan *enumeration mapping*.

### Ujicoba dan Evaluasi

Sebelum pengujian deteksi wajah dilakukan secara *real-time* yaitu memperoleh citra latih dari setiap mahasiswa yang akan diuji menggunakan *capture* dari webcam yang dihubungkan ke Laptop atau PC. Jumlah mahasiswa yang akan diuji berjumlah 8 orang dan setiap orangnya memiliki 8 citra

wajah. Proses pengambilan citra latih dari setiap mahasiswa dilakukan pada setiap mahasiswa, pada kondisi lokasi sama yang akan dijadikan lokasi pengujian. Proses pengambilan citra latih dilakukan pada kondisi ruang kelas sama dengan kondisi ruang kelas pada saat pengujian berlangsung. Waktu pengambilan citra latih disamakan dengan waktu pada saat proses pengujian, yaitu pagi hari.

Pengujian dilakukan dengan 2 tahap, tahap pertama yaitu pengujian deteksi wajah berdasarkan hasil ukuran citra dan tahap yang kedua pengujian deteksi wajah berdasarkan posisi wajah. Pengujian deteksi wajah berdasarkan hasil ukuran citra adalah pengujian deteksi wajah untuk mengetahui ukuran citra yang dihasilkan. Pengujian ini

dilakukan untuk mengetahui ukuran citra wajah ideal yang dibutuhkan untuk dijadikan ukuran citra latih. Pengujian deteksi wajah ini dilakukan terhadap tiga jarak objek wajah, yaitu 100 cm, 150 cm dan 200 cm. Pengujian deteksi wajah berdasarkan posisi wajah dilakukan untuk mengetahui posisi wajah yang mampu dideteksi oleh sistem, dan menjadi data dasar untuk dijadikan ketentuan dalam proses mendapatkan citra latih dari setiap mahasiswa. Pengujian posisi wajah yang diuji adalah posisi tegak lurus, rotasi sejajar 15° kekanan, rotasi sejajar 15° kekiri, mengangkat dagu 15° keatas dan menunduk kepala 15°. Pengujian ini dilakukan dengan kondisi cahaya normal. Berikut gambar hasil deteksi wajah berdasarkan posisi wajah dan ukuran citra wajah :



Gambar 11. Hasil deteksi citra wajah pada jarak 100 cm



Gambar 12. Hasil deteksi citra wajah pada jarak 150 cm



Gambar 13. Hasil deteksi citra wajah pada jarak 200 cm

Tabel 1. Pengujian deteksi wajah berdasarkan hasil ukuran citra

Jarak objek dari kamera (cm)	Ukuran citra wajah
100	418 piksel x 465 piksel
150	373 piksel x 387 piksel
200	244 piksel x 249 piksel

Dari tabel diatas ada perbedaan ukuran pada pengujian ini dikarenakan perbedaan jarak objek (wajah) dari kamera, semakin jauh objek dari kamera, maka semakin kecil pula citra wajah yang ditangkap oleh kamera. Dari data ini maka ukuran citra yang digunakan sebagai ukuran citra latih adalah 100 x 100 piksel. Hal ini dilakukan untuk memperoleh keakurasian tinggi saat proses pengujian pengenalan wajah atau proses absensi berlangsung.

Dari percobaan yang telah dilakukan untuk mengetahui posisi wajah yang dapat dideteksi sistem, diperoleh hasil yang diperlihatkan pada Tabel 2.

Tabel 2. Pengujian deteksi wajah berdasarkan posisi citra wajah

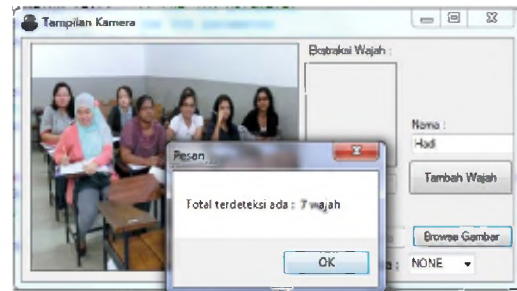
Posisi wajah	Hasil Deteksi
Tegak Lurus	Terdeteksi
Rotasi sejajar 15° kekanan	Terdeteksi
Rotasi sejajar 15° kekiri	Terdeteksi
Mengangkat dagu 15°	Terdeteksi
Menundukkan kepala 15°	Terdeteksi

Pada tabel diatas diperoleh hasil bahwa terdapat lima posisi wajah yang mampu dideteksi oleh sistem, yaitu posisi tegak, posisi rotasi sejajar 15° kekanan, rotasi sejajar 15° kekiri, posisi mengangkat dagu 15° dan posisi menundukkan kepala 15°. Berdasarkan hasil pengujian pada Tabel 2 ditetapkan ketentuan posisi wajah yang digunakan sebagai posisi dalam mendapatkan citra latih adalah tegak lurus, rotasi sejajar 15° kekanan dan rotasi sejajar 15° kekiri.

Setelah proses pengambilan citra latih selesai, maka citra yang telah dipilih dilakukan pelatihan. Pelatihan citra ini berfungsi mencari nilai bobot setiap citra yang akan dijadikan dasar dalam proses pengenalan nantinya. Proses awal pelatihan citra wajah adalah normalisasi citra, yaitu merubah citra menjadi ukuran 100 piksel x 100 piksel dalam dalam format grayscale seperti terlihat pada Gambar 9.



Gambar 14. Citra Latih pada jarak 100 cm



Gambar 15. Menghitung jumlah wajah yang terdeteksi



Gambar 16. Deteksi wajah dalam ruang kelas



Gambar 17. Deteksi wajah dalam ruang kelas dengan posisi 15° kekanan



Gambar 18. Deteksi wajah dalam ruang kelas dengan posisi 15° menundukkan kepala



Gambar 19. Deteksi wajah dalam ruang kelas dengan posisi 15° ke kiri

## KESIMPULAN

Berdasarkan pengujian yang dilakukan yaitu deteksi banyak wajah dalam ruang kelas dengan metode *haar cascade classifier* dan algoritma Adaboost dapat diambil kesimpulan bahwa:

- Metode *haar cascade classifier* sangat ideal digunakan untuk deteksi banyak wajah dalam ruang kelas secara *real time*;
- Berdasarkan indikasi dari deteksi banyak wajah secara *real time*, maka jumlah orang akan diketahui;
- Pada pengujian secara *real time* ini jika citra wajah terhalang oleh objek lain maka citra wajah tersebut tidak akan terdeteksi seperti diperlihatkan pada Gambar 15.

## DAFTAR PUSTAKA

Bradski, G. & Kaehler, A., *Learning OpenCV*, 2008, O'Reilly Media, 1005 Gravenstein Highway North, Sebastopol, CA 95472  
 Choi, J., Han, D. & Yoo, S.J., 2011. Real-Time Face-Detection Engine for Robustness to Variable Illumination and Rotated Faces. *2011 First ACIS/JNU International Conference on*

*Computers, Networks, Systems and Industrial Engineering*, pp.53–58. Available at: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5954277> [Accessed September 25, 2013].

Krishna, M.G. & Srinivasulu, A., 2012. Face Detection System On AdaBoost Algorithm Using Haar Classifiers. , 2, pp.3556–3560.

Lienhart, R. & Maydt, J., 2002. An Extended Set of Haar-like Features for Rapid Object Detection. , pp.900–903.

Padmaja, K. & T.N Prabakar, FPGA Based Real Time Multi View Face Detection Using Adaptive Boosting And Histogram Equalization. , 06(6), pp.17–23.

Schneiderman, H. & Kanade, T., Probabilistic modeling of local appearance and spatial relationships for object recognition. *Proceedings. 1998 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No.98CB36231)*, pp.45–51. Available at: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=698586>.

Viola, P. & Jones, M., 2001. Robust Real-time Object Detection. *Second International Workshop On Statistical And Computational Theories OF Vision – Modeling, Learning, Computing, And Sampling Vancouver, Canada, July 13, 2001*.

Viola, P. & Jones, M.J., 2004. Robust Real-Time Face Detection. , 57(2), pp.137–154.

Zhang, T. et al., 2009. Using Gradientfaces. , 18(11), pp.2599–2606.