

## Implementasi Perhitungan Face Detection Dengan Metode Haar Cascade Classifier

Chaerullah Fadli<sup>1</sup>, Desmulyati<sup>2</sup>

<sup>1</sup>Universitas Nusa Mandiri

Jl. Jatiwaringin No. 2, Cipinang Melayu, Makasar, Jakarta Timur - 13620, Indonesia

<sup>2</sup>Universitas Bina Sarana Informatika

Jl. Kramat Raya No.98, RW.9, Kwitang, Kec. Senen, Kota Jakarta Pusat, Daerah Khusus Ibukota Jakarta  
10450, Indonesia

Corresponding author's e-mail: <sup>1</sup>chaerullah.fadli01@gmail.com, <sup>2</sup>desmulyati.dmy@bsi.ac.id

*Abstrak*— *Face detection* saat ini banyak diimplementasikan di perusahaan, untuk absensi, keamanan, dan lain-lain. Begitupun pada SMK Teratai Putih yang masih melakukan absensi secara manual dengan memanggil nama persiswa, sedang mencoba mengimplementasikan *Face Detection* di salah satu ruangan yaitu, laboratorium komputer untuk keperluan absensi siswa, yang menggunakan foto-foto siswa sebagai datasetnya. Metode *Haar Cascade Classifier* untuk *face detection* menggunakan pemrograman python dan opencv adalah salah satu yang digunakan, dikarenakan yang tingkat akurasi dalam mengukur atau mendeteksi wajah manusia mendapatkan hasil yang baik. Dengan diterapkan *face detection* harapan adalah dapat mengurangi waktu absensi siswa yang dilakukan oleh guru, dengan memanggil nama siswa satu-persatu. Penulis melihat dari hasil penelitian yang dilakukan di SMK Teratai Putih Jakarta, adalah hasil yang cukup baik dengan tingkat akurasi diatas 50% (>50%). Dengan syarat *file size* fotonya dibawah 1 MB dan *dimensions* 1000 pixel dan diujicoba dimulai dari nilai *Haar Cascade Classifier* terkecil. Untuk mendeteksi wajah siswa dilaboratorium komputer yang dijadikan absensi secara manual tidak diperlukan lagi dan waktunya bisa digunakan untuk belajar yang lebih ke mata pelajaran-pelajaran yang diajarkan oleh guru-guru.

**Kata Kunci:** Face Detection, Haar Cascade Classifier, Opencv, Python

*Abstract- Face detection is currently widely applied in companies, for attendance, security, and others. Likewise at SMK Teratai Putih, which still performs attendance manually by calling a fixed name, tries to apply Face Detection in one of the rooms, namely the computer laboratory for student attendance purposes, which uses student photos as datasets. The Haar Cascade Classifier method for face detection using python and opencv programming is one that is used, because the level of accuracy in measuring or detecting human faces gets good results. With the implementation of face detection, it is hoped that the teacher can reduce student attendance by calling the names of students one by one. From the results of the research that the author did at SMK Teratai Putih Jakarta, the results were quite good with an accuracy rate above 50% (>50%). With the condition that the photo file size is below 1 MB and the dimensions are 1000 pixels and tested starting from the smallest Haar Cascade Classifier value. To detect students' faces in computer labs that are used as attendance manually is no longer needed and time can be used to learn more about the subjects taught by the teacher.*

**Keywords:** Face Detection, Haar Cascade Classifier, Opencv, Python

### 1. Pendahuluan

Perkembangan teknologi yang begitu cepat memudahkan kita dalam mendapatkan semua informasi dengan praktis, efisiensi dari sisi biaya, dan waktu.

Saat ini telah banyak perkembangan sistem yang memanfaatkan deteksi wajah baik menggunakan foto atau video yang tujuannya untuk sistem akses keamanan maupun sistem kontrol. Salah satu deteksi wajah tersebut adalah *Haar Cascade Classifier* yaitu algoritma pendeteksian objek pembelajaran mesin yang digunakan untuk mengidentifikasi objek dalam gambar atau video dan berdasarkan konsep fitur yang diusulkan oleh Paul Viola dan Michael Jones dalam makalah mereka "*Rapid Object Detection using a Boosted Cascade of Simple Features*" di 2001. Metode penyimpanan secara berkelompok berdasarkan data atau citra wajah yang akan diteliti [1]

"Pendeteksian wajah atau *face detection* merupakan suatu kegiatan yang umum dilakukan manusia dalam kesehariannya" [2]. *Face detection* atau pengenalan wajah sudah diterapkan di berbagai perusahaan-perusahaan kecil atau besar. Mendeteksi wajah juga dilakukan dengan perangkat PC [2]. Contohnya adalah *Face Fingerprint* teknologi yang digunakan untuk absensi kehadiran pegawai di perusahaan-perusahaan.

Pada SMK Teratai Putih Jakarta masih dilakukannya absensi siswa-siswi secara manual dengan memanggil

satu persatu. Untuk menghilangkan absensi secara manual penulis melakukan penelitian di sekolah SMK Teratai Putih dengan mengambil salah satu contoh foto siswa-siswi di ruang laboratorium komputer. Dengan tujuan untuk menerapkan *Face detection* di ruang laboratorium.

*Face detection* yang dilakukan SMK Teratai Putih Jakarta adalah menfoto siswa lalu di proses menggunakan metode *Haar Cascade Classifier* pada opencv dengan bahasa pemrograman python. Foto-foto tersebut di deteksi apakah foto tersebut terdeteksi wajah dan ada berapa dalam fotonya.

## 2. Tinjauan Pustaka

### 2.1. Artificial Intelligence

Menurut Jhon McCarthy salah satu pakar AI, menjelaskan AI adalah disiplin ilmu yang membuat mesin menjadi cerdas [3]

### 2.2. Image Processing

Berarti memproses gambar menggunakan komputer, dengan cara mengubah suatu data gambar menjadi lebih baik atau bermutu [4]

### 2.3. Face detection

Di dalam teknologi pengenalan wajah, *face detection* merupakan tahap awal pemrosesan untuk mengenali wajah seseorang, dimana *face detection* menentukan dimana bagian wajah yang muncul pada citra masukan. [5]

### 2.4. Haar Cascade Classifier

*Haar like feature* atau yang dikenal sebagai *Haar Cascade Classifier* merupakan *rectangular* (persegi) *feature*, yang memberikan indikasi secara spesifik pada sebuah gambar atau *image*. *Haar cascade classifier* berasal dari gagasan Paul Viola dan Michael Jhon, karena itu dinamakan metode Viola & Jhon. Ide dari *Haar like feature* adalah mengenali obyek berdasarkan nilai sederhana dari fitur tetapi bukan merupakan nilai piksel dari image obyek tersebut. *Cascade Classifier* merupakan metode klasifikasi bertingkat yang bertugas untuk menolak area gambar yang tidak terdeteksi wajah dengan menggunakan *classifier* yang telah dilatih oleh algoritma *AdaBoost* pada tiap tingkatan klasifikasinya”[6]

### 2.5. Metode Algorithma

Metode pengelompokkan *image* atau video sesuai data yang dibentuknya. Termasuk metode *Haar Cascade Classifier* ini yang melakukan penyesuaian dari data yang dinilai [7]

### 2.6. Dataset

Berupa image atau foto wajah yang dipilih dari para siswa. [8]

### 2.7. Konsep Pemrograman

Terdiri dari aplikasi *Python*, yaitu aplikasi yang digunakan untuk merancang program berbasis obyek [9]. *PyCharm. Software* hasil perluasan yang dikembangkan dari aplikasi *python* [10]. Dan *OpenCV, Software* untuk membantu mengolah gambar [11]

## 3. Metode Penelitian

Metode penelitian.berupa tahapan-tahapan yang dilakukan dalam pengembangan *software* dengan metode *waterfall*, sebagai berikut:

#### 1) Analisa Kebutuhan

*Face Detection* dengan metode *Haar Cascade Classifier* dirancang untuk mempermudah dan mempercepat proses absensi melalui pengenalan wajah, khususnya untuk absensi di laboratorium. Guru hanya perlu memfoto siswa untuk di absensi, tanpa harus memanggil satu-persatu.

#### 2) Desain

Alur desain sistem yang dimana guru menfoto siswa, lalu foto tersebut di masukan ke *software PyCharm* yang akan di kelolah menggunakan program python, setelah itu akan muncul hasilnya.

#### 3) Coding

*Face Detection* dengan metode *Haar Cascade Classifier* menggunakan program *python* dan library *openCV*.

#### 4) Testing

terdiri dari data test: Menggunakan dataset foto yang sudah ada, di eksekusi software dengan program *python*, Input test: Foto siswa, ProsesTest dan output yang dihasilkan.

#### 5) Support

Berupa dukungan perangkat keras/*hardware* yang digunakan dalam penelitian. Kemudian dukungan *software*-nya berupa *PyCharm*.

#### 4. Hasil dan Pembahasan

Penulis akan menggunakan dua foto yang berbeda, lalu hasil pengujian perhitungan jumlah wajah ini akan dibahas dari *Size*, *Dimenssion*, Nilai *Haar cascade*, dan *Code Program* untuk mendapatkan hasil yang optimal.

##### 1. *Face Detection* Foto Kesatu








Menggunakan foto yang *size* 233 KB, dan *pixels* 648x486.



Gambar 1. Hasil Akhir Foto Kesatu

Tabel hasil pengujian foto kesatu:

Tabel 1. Hasil *Face Detection* Foto Kesatu

No	Nilai Haar Cascade	Jumlah Wajah Terdeteksi	Citra	Hasil	Akurasi (%)	Error (%)
1	1.1, 1	11 Orang		Berhasil	93,72	13,4
2	1.1, 2	10 Orang		Berhasil	51,54	7,36
3	1.1, 3	8 Orang		Berhasil	42,954	6,136
4	1.1, 4	7 Orang		Berhasil	36,81	5,26
5	1.1, 5	7 Orang		Berhasil	29,45	4,20
6	1.1, 6	7 Orang		Berhasil	24,54	3,50
7	1.1, 7	6 Orang		Berhasil	24,54	3,50

8	1.1, 8	6 Orang		Berhasil	21,477	3,06
9	1.1, 9	6 Orang		Berhasil	19,09	2,72
10	1.1, 10	6 Orang		Berhasil	17,18	2,45
Rata-rata					36,1	5,1

Dari tabel hasil pengujian diatas menggunakan Metode *Haar Cascade Classifier* untuk menghitung jumlah wajah pada suatu foto dapat mendeteksi wajah manusia dengan Nilai *Haar Cascade* 1.1, 1 sampai 1.1, 10 adalah sebagai parameter *Haarcase*. Yang dimana 1,1 adalah sebagai *Scale factor* merupakan parameter yang menentukan seberapa besar ukuran gambar diturunkan nilainya pada tiap skala gambar, sedangkan 1 adalah nilai *min neighbors* merupakan jumlah minimal persegi di sekitar yang akan membentuk suatu objek. Jadi dari tabel di atas dapat diperoleh rata-rata akurasi 36%, dan *error* 5,1% dari 10 kali percobaan. Jumlah akurasi 36% dikarenakan terlalu banyak objek dalam satu foto, jadi untuk tingkat akurasi dibawah 50%.

Rumus perhitungan nilai akurasi dan *error* adalah:

A. Akurasi

$(\text{dimensi foto} + \text{dimensi foto}) / (\text{para meter haarcase}) / \text{jumlah wajah yang terdeteksi} * 100\%$

Contoh foto no.1 pada tabel diatas:

$$(648 + 486) / (1.1 \times 1) / 11 * 100\%$$

$$= 1134 / 1,1 / 11 * 100\%$$

$$= 1030,90 / 11 * 100\%$$

$$= 93,72\%$$

B. Error

$(\text{Hasil Akurasi} / \text{Jumlah Foto Wajah Real}) * 100\%$

Contoh foto no.1 pada tabel diatas:

$$(93,72 / 7) * 100\%$$

$$= 13,388 * 100\%$$

$$= 13,388\%$$

2. *Face Detection* foto kedua











*Face detection* foto kedua menggunakan foto yang *size* 231 KB, dan *pixel* 486X648.



Gambar 2. Hasil Akhir Foto Kedua

Tabel hasil pengujian:

Tabel 2. Hasil *Face Detection* Foto Kedua

No	Nilai Haar cascade	Jumlah Wajah Terdeteksi	Citra	Hasil	Akurasi (%)	Error (%)
1	1.1, 1	3 Orang		Berhasil	16,2	2,7
2	1.1, 2	3 Orang		Berhasil	35,1	5,85
3	1.1, 3	3 Orang		Berhasil	48,6	8,1
4	1.1, 4	2 Orang		Berhasil	43,2	10,8
5	1.1, 5	2 Orang		Berhasil	58,57	14,64
6	1.1, 6	2 Orang		Berhasil	64,8	16,2
7	1.1, 7	2 Orang		Berhasil	75,6	18
8	1.1, 8	2 Orang		Berhasil	86,4	21,6
9	1.1, 9	2 Orang		Berhasil	97,33	24,33
10	1.1, 10	2 Orang		Berhasil	97,33	24,33
Rata-rata					62,31	14,65

Dari tabel hasil pengujian diatas menggunakan Metode *Haar Cascade Classifier* untuk menghitung jumlah wajah pada suatu foto dapat mendeteksi wajah manusia dengan Nilai *Haar Cascade* 1.1, 1 sampai 1.1, 10 adalah sebagai parameter *Haarcase*. Yang dimana 1,1 adalah sebagai *Scale factor* merupakan parameter yang menentukan seberapa besar ukuran gambar diturunkan nilainya pada tiap skala gambar, sedangkan 1 adalah nilai *min neighbors* merupakan jumlah minimal persegi di sekitar yang akan membentuk suatu objek. Jadi dari tabel di atas dapat diperoleh rata-rata akurasi 62,31%, dan *error* 14,65%. Jumlah akurasi 62,31% dipatakan objeknya yang sedikit dalam satu foto, jadi untuk tingkat akurasinya diatas 50%. Rumus perhitungan nilai akurasi dan *error* adalah:



A. Akurasi

$(\text{Dimensi foto} + \text{dimensi foto}) / (\text{size foto} / (\text{para meter haarcase})) * \text{jumlah wajah yang terdeteksi} * 100\%$

Contoh foto no.1 pada tabel diatas:

$$\begin{aligned} & (648 + 486) / (231 / (1.1 \times 1)) * 3 * 100\% \\ & = (1134) / (231 / 1,1) * 3 * 100\% \\ & = (1134 / 210) * 3 * 100\% \\ & = 5,4 * 3 * 100\% \\ & = 16,2 * 100\% \\ & = 16,2\% \end{aligned}$$

B. Error

Contoh foto no.1 pada tabel diatas:

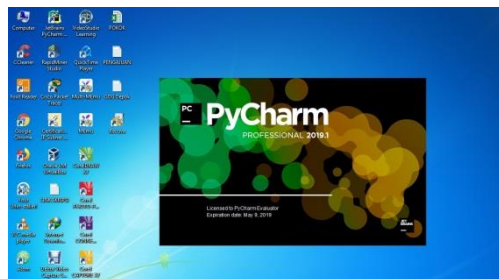
$$\begin{aligned} & (5,4 / 2) * 100\% \\ & = 2,7 * 100\% \\ & = 2,7\% \end{aligned}$$

C. Cara Kerja Aplikasi

Cara kerja aplikasi *Face Detection* yang dilakukan oleh penulis dengan mengumpulkan foto siswa sebagai model awal untuk dicoba. Dibawah ini beberapa cara kerja aplikasi *Face Detection* di SMK Teratai Putih Jakarta:

1) *Install Pycharm*

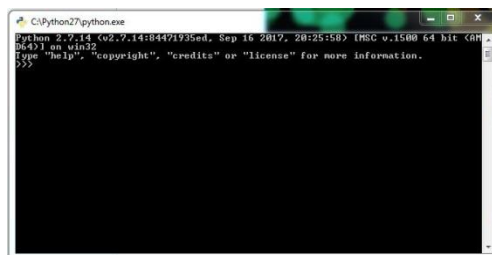
Menginstall *software texteditor* untuk digunakan bahasa pemrograman *python*.



Gambar.3 Software Pycharm

2) *Install Python 2.7*

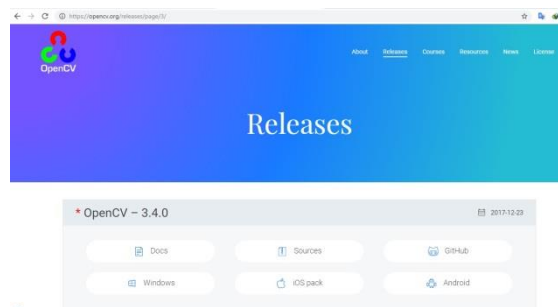
Menginstall *python 2.7* untuk pemrograman *python* yang akan digunakan *Face Detection*.



Gambar.4 Software Python

3) *Download OpenCV*

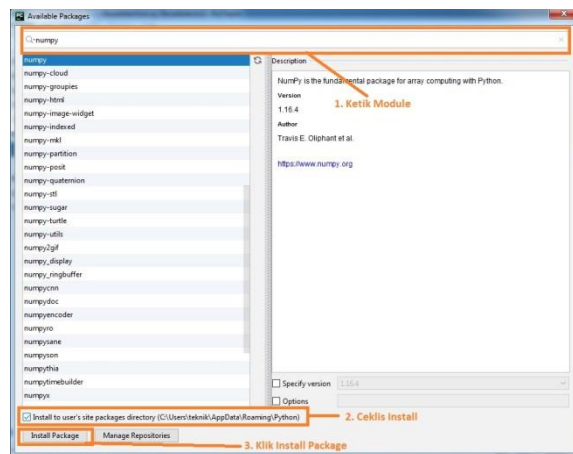
*Opencv* merupakan module utama yang digunakan untuk melakukan pemrosesan terhadap citra, baik gambar maupun foto. *Opencv* dapat melakukan banyak tugas, mulai dari membaca *file* gambar, melakukan *editing* warna, hingga mendeteksi wajah secara efektif.



Gambar.5. OpenCV

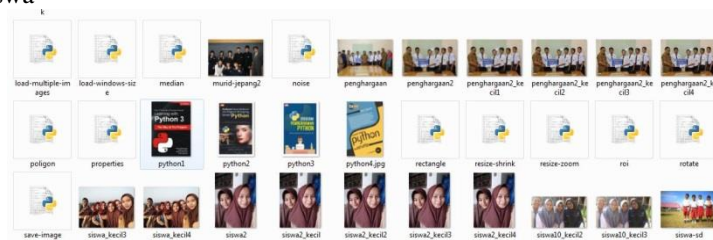
4) *Download Module Numpy*

Fungsi numpy adalah *module* yang dibutuhkan agar Opencv dapat bekerja dengan baik. *Numpy* sendiri singkatan dari *Numerical Python*, adalah *library* yang menyediakan berbagai routine untuk mengolah array multidimensi. Dengan kata lain, numpy adalah library python yang untuk membantu komputasi ilmiah dengan bahasa pemrograman tersebut.



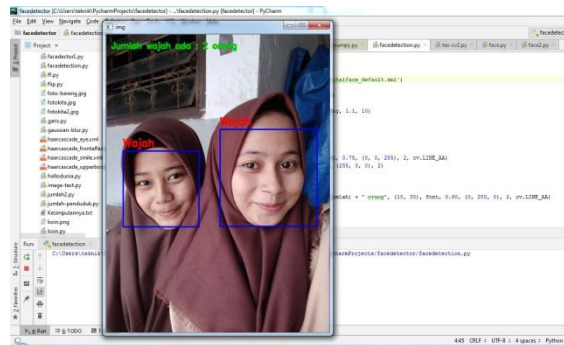
Gambar 5. *Download Numpy*

5) *Dataset Foto Siswa*



Gambar 6. *Dataset Siswa*

## 6) Menjalankan Aplikasi



Gambar 7. Hasil *Face Detection*

*Face detection* menggunakan foto dengan *size* 231 KB, dan *pixel* 486 x 648.

## 5. Kesimpulan

Setelah melakukan penelitian *Face Detection* dengan metode *Haar Cascade Classifier* di SMK Teratai Putih Jakarta, *face detection* mendapatkan hasil yang cukup bagus, dilihat dari tabel hasil foto *face detection* didapatkan pengujian dengan akurasi di atas 50% (>50%). Dengan syarat *file size* fotonya dibawah 1 MB dan *dimensions* 1000 pixel dan diujicoba dimulai dari nilai *Haar Cascade Classifier* terkecil.

## Daftar Pustaka

- [1] M. Al-gaufiqy, S. Rasmana, and I. Puspasari, "Otomasi Sistem Greenhouse Berbasis Mikrokontroler," vol. 6, no. 1, pp. 73–86, 2017.
- [2] M. Syarif and Wijanarto, "Deteksi Kedipan Mata Dengan Haar Cascade Classifier Dan Contour Untuk Password Login," vol. 14, no. 4, pp. 242–249, 2015.
- [3] R. Primartha, *Belajar Machine Learning: Teori dan Praktik*. 2018.
- [4] I. P. Sinaga, I. Prasetya, D. Wibawa, and E. Kuniawan, "Background Substraction Dan Haar Cascade People Counter And Face Identification System With Background," vol. 4, no. 2, pp. 1544–1551, 2017.
- [5] L. W. Alexander, S. R. Sentinuwo, and A. M. Sambul, "Implementasi Algoritma Pengenalan Wajah Untuk Mendeteksi Visual Hacking," vol. 11, no. 1, 2017.
- [6] D. I. S. Saputra, W. S. Anjar, and K. A. N. Ramadhan, "Pelacakan dan deteksi wajah menggunakan video langsung pada webcam," vol. 10, no. 1, pp. 50–59, 2017.
- [7] K. O. Sanjaya, G. Indrawan, K. Yota, and E. Aryanto, "Pendeteksian Objek Rokok Pada Video Berbasis Haar Cascade Classifier," vol. 1, pp. 92–99, 2017.
- [8] M. Prawiro, "PENGERTIAN DATA," 2019. [Online]. Available: <https://www.maxmanroe.com/vid/teknologi/pengertian-data.html>.
- [9] "The Python Tutorial," 2019. [Online]. Available: <https://docs.python.org/3/tutorial>.
- [10] JetBrains, "PyCharm," 2017. [Online]. Available: <https://www.jetbrains.com/pycharm/guide/>.
- [11] S. Purnamawati, R. F. Rahmat, M. Santana, V. Jones, and M. Haar, "Untuk Menghitung Jumlah Manusia," *Lentera Vol. 15.Juli 2015*, vol. 15, pp. 73–80, 2015.