Sistem Pengenalan Ekspresi Wajah Secara Real Time Menggunakan Metode Viola-Jones

Proposal Tugas Akhir

Kelas TA 1

La Ode Muhamad Sofyan Syarafa
NIM: 1301150785

Program Studi Sarjana Teknik Informatika

Fakultas Informatika

Universitas Telkom

Bandung

2018

Lembar Persetujuan

Sistem Pengenalan Ekspresi Wajah Secara Real Time Menggunakan Metode Viola-Jones

Real Time Face Expression Recognition System Using the Viola-Jones Method

La Ode Muhamad Sofyan Syarafa
NIM: 1301150785

Proposal ini diajukan sebagai usulan pembuatan tugas akhir pada
Program Studi Sarjana Teknik Informatika
Fakultas Informatika Universitas Telkom

Bandung, 22 Agustus 2018
Menyetujui

Calon Pembimbing 1

Dr. Putu Harry Gunawan
NIP: 16860043

Abstrak

Penelitian ini bertujuan untuk membuat suatu sistem pengenalan wajah beserta ekspresinya secara *real time*. Untuk itu, sistem tersebut dapat menunjukkan suatu *detector* yang membedakan objek yang dimaksud dengan gambar latar belakang. Sistem tersebut akan mengambil citra melalui *webcam* perangkat untuk dapat diklasifikasikan oleh sistem. Sistem yang dikerjakan dengan menerapkan Metode *Viola-Jones* dimulai dari proses penerapan fitur *Haar-Like*, *Integral Image*, *Adaptive Boosting*, hingga menerapkan skema *Cascading* untuk pengklasifikasiannya. Pengerjaan sistem dilakukan dengan menggunakan bahasa pemrograman python serta menggunakan Pustaka OpenCV. Hasil keluaran dari sistem akan menampilkan pengklasifikasian wajah beserta ekspresinya dan latar belakang yang dianggap sebagai citra non-objek. Hasil klasifikasi tersebut ditandai dengan munculnya *rectangle* sebagai detektor visual untuk menujukkan objek yang diharapkan.

Kata Kunci: viola-jones, haar-like features, integral image, adaptive boosting, cascade, python, opencv

Daftar Isi

Bab I

Pendahuluan

1.1 Latar Belakang

Pemrosesan wajah (*face processing*) merupakan salah satu teknologi dalam bidang *computer vision*yang paling berkembang saat ini. Proses ini banyak diaplikasikan dalam sistem pengenalan biometrik [13]yang dapat dijadikan sebagai suatu alternatif untuk pengamanan sistem [3], sistem lokalisasi wajah (*face localization*), penjejakan wajah (*face tracking*), serta sistem pengenalan expresi wajah (*facial expression recognition*) [10]. Dalam pengenalan ekspresi wajah, suatu komputer diharapkan dapat menjadi alat untuk dapat mengenali emosi-emosi seseorang yang diungkapkan melalui wajahnya. Secara alami, manusia dapat mengenali ekspresi-ekspresi tersebut terlepasi dari perbedaan bahasa dan budaya. Umumnya, manusia sepakat bahwa ekspresi mencerminkan pengalaman fundamental [8]. Emosi fundamental ini mencakup kemarahan, penghinaan, kejijikan, ketakutan, kebahagiaan, kesedihan, dan kejutan [2][6]. Proses pengenalan ekspresi ini diilustrasikan melalui Gambar 1.1

Gambar 1.1: Ilustrasi proses pengenalan ekspresi pada wajah

Saat ini, beberapa peneliti memanfaatkan pengenalan wajah dan pengenalan ekspresi wajah oleh komputer sebagai objek penelitiannya [[9] [14] [15] [8] [11]]. Papageorgiou [9] menjelaskan kerangka umum untuk mengenali wajah dengan berbagai *classification threshold*. Paul Viola [14] menjelaskan kerangka

kerja pendeteksian objek dengan sangat cepat. Wang [15] melakukan analisis terhadap algoritma *Viola-Jones*. Kemudian ada James Pao dan Wasim Ahmed Syed [8] [11] melakukan pendeteksian emosi melalui pengenalan fitur wajah. Untuk itu, penelitian ini dimaksudkan untuk membuat sistem pendeteksian dan pengenalan objek wajah serta objek-objek lain yang ada pada wajah dengan mengacu pada penelitian-penelitian tersebut.

Algoritma Viola-Jones merupakan algoritma yang paling populer[1] untuk memproses suatu masukan citra sehingga dapat dikenali oleh komputer. Pada pemrosesan wajah, algoritma ini telah menjadi standar *defacto*. Secara umum, algoritma ini digunakan untuk untuk mendeteksi objek dengan pemrosesan yang sangat cepat dan tingkat akurasi yang sangat tinggi dengan pendekatan *machine learning* [14]. Di samping itu, metode atau kerangka kerja ini telah dikembangkan untuk menentukan lokasi objek [7]].

Tulisan ini digunakan sebagai proposal untuk menunjang pengerjaan tugas akhir yang berjudul *Sistem Pengenalan Ekspresi Wajah Secara Real Time Menggunakan Metode Viola-Jones* . Oleh karena itu, tulisan ini memuat tahapan-tahapan yang akan dilakukan untuk mengerjakan program komputer yang berkaitan dengan judul tersebut. Dimulai dari melakukan studi literatur, mengimplemetasikan hasil studi tersebut ke dalam program komputer, hingga mengamati dan melakukan analisis terhadap hasil program tersebut.

1.2 Perumusan Masalah

Berikut rumusan masalah yang ingin saya angkat adalah

1. Apakah sistem dapat mendeteksi wajah manusia dan ekspresinya secara *real time*?

2. Bagaimana cara program menunjukkan wajah serta ekspresinya dari sebuah citra *real time*?

1.3 Tujuan

Berikut adalah tujuan yang ingin dicapai pada penulisan proposal/TA.

1. Mengetahui proses pengenalan wajah dan ekspresinya secara *real time* beserta hasilnya;

2. Mengetahui cara kerja *detector* selama melakukan proses pengenalan wajah beserta ekspresinya;

1.4 Batasan Masalah

Batasan masalah dari tulisan ini adalah

1. Sistem hanya akan bekerja melalui masukan citra video dari *webcam*, tidak melalui masukan file citra *.jpg, .png,* dan sejenisnya;

2. Proses ini hanya dapat digunakan jika citra masukan tegak lurus dengan posisi *webcam*;

3. Ekspresi wajah yang dapat dikenali sistem dibatasi untuk mengenali ekspresi senyum pada wajah yang menggambarkan kebahagiaan;

4. Data latih yang digunakan didapatkan dari berbagai sumber.

1.5 Rencana Kegiatan

Rencana kegitana yang akan saya lakukan adalah sebagia berikut:

- Studi literatur
- Menganalisis metode dan perancangan sistem
- Implementasi sistem
- Memeriksa dan menganalisis hasil
- Penulisan laporan tugas akhir

1.6 Jadwal Kegiatan

Laporan proposal ini akan dijadwalkan sesuai dengan tabel di bawah ini.

Tabel 1.1: Jadwal kegiatan proposal tugas akhir

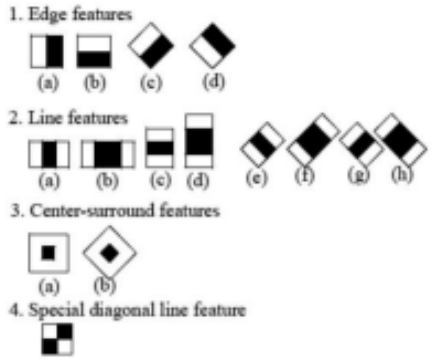
No	**Kegiatan**	**Bulan ke-**																					
		1				**2**				**3**				**4**				**5**				**6**			
1	Studi Literatur																						
2	Pengumpulan Data																						
3	Analisis dan Perancangan Sistem																						
4	Implementasi Sistem																						
5	Analisa Hasil Implementasi																						
6	Penulisan Laporan																						

Bab II

Kajian Pustaka

2.1 Haar-Like Features

Haar-like features dapat diartikan sebagai produk skalar antara gambar asli dan beberapa dengan beberapa pola dasar atau fitur *Haar-Like* [15]. Fitur *Haar-Like* yang dimaksud dapat ditunjukkan pada Gambar 2.1



Gambar 2.1: Pola Dasar *Haar-Like Features*

Ada beberapa alasan Metode Viola-Jones menggunakan Haar-Like Features daripada menggunakan pixel sebuah gambar secara langsung [14]. Pertama, fitur dapat disandingkan dengan gambar asli untuk memudahkan proses pembelajaran pada data latih. Kedua, penggunaan fitur *Haar-Like* akan lebih cepat dan efektif dibandingkan dengan penggunaan pixel gambar secara langsung.

Setiap fitur *Haar-Like* memiliki suatu nilai yang mewakili intensitas warna fitur. Nilai-nilai tersebut didapatkan melalui persamaan 2.1 [12].

$$F_{(\text{Haar})} = \Sigma F_{\text{White}} - \Sigma F_{\text{Black}} \tag{2.1}$$

$\text{F}_{(\text{Haar})}$ menyatakan nilai total dari fitur, ΣF_{White} menyatakan nilai fitur pada area terang, sedangkan ΣF_{Black} menyatakan nilai fitur pada area gelap.

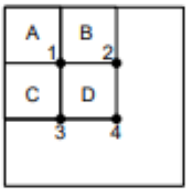
2.2 Metode Viola-Jones

Secara umum, metode Viola-Jones yang dikembangkan oleh Paul Viola dan Michael Jones digunakan dalam pengolahan citra digital untuk mendeteksi dan mengenali objek secara *real time* dengan waktu pemrosesan yang sangat cepat dan lebih akurat. Metode ini memiliki tingkat keakuratan 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* [12]. Metode ini mengandung tiga kontribusi utama [14], yaitu *integral image*, *Adaptive Boosting*, dan *Cascade of Classifier*.

2.2.1 Integral Image

Metode Viola-Jones mengaplikasikan fitur *Haar-Like* kepada citra masukan dengan dua alasan yang telah dikemukakan sebelumnya serta masing-masing fiturnya akan dihitung nilai-nilainya. Sebagai contoh, suatu citra masukkan yang hanya berukuran 24x24 pixel dapat memiliki lebih dari 180,000 fitur *Haar-Like*. Hal tersebut tentunya dapat mengakibatkan biaya komputasinya menjadi lebih besar[14]. Oleh karena itu, *integral image* digunakan untuk mempercepat perhitungan sehingga dapat mengurangi biaya komputasi.

Gambar 2.2: *Integral Image* pada titik *x,y*



Gambar 2.3: Perhitungan Nilai fitur *Haar-Like*

Berdasarkan Gambar 2.2, perhitungan *integral image* pada titik tersebut dapat dilakukan dengan persamaan 2.4 [12].

$$ii(x,y) = \sum_{x' \leqslant x, y \leqslant y} i(x',y') \tag{2.2}$$

di mana *ii(x,y)* adalah *integral image* dan *i(x,y)* adalah citra masukan yang asli dengan kondisi

$$ii(x,y) = i(x,y) + ii(x-1,y) + ii(x,y-1) - ii(x-1,y-1) \tag{2.3}$$

Sebagai contoh, Gambar 2.3 dapat dihitung dengan menggunakan teknik *integral image*. Nilai *integral image* pada titik 1 merupakan jumlah setiap pixel dari A. Sehingga pada titik 2 nilainya adalah A + B dan pada titik 3 adalah A + C, serta pada titik 4 adalah A + B + C + D. Oleh karena itu, nilai fitur D dapat dihitung dengan cara 4 + 1 - (2 + 3).

2.2.2 Adaptive Boosting

AdaBoost (*Adaptive Boost*) merupakan suatu algoritma pembelajaran yang digunakan untuk meningkatkan kinerja pengklasifikasian atau regresi. Disebut adaptif karena digunakan untuk menyesuaikan beberapa *weak classifier* sehingga menjadi satu *strong classifier* secara iteratif. Pada pengaplikasiannnya pada pemrosesan wajah, *AdaBoost* bekerja dengan menerapkan suatu *threshold* sehingga dapat memisahkan wajah dari objek non-wajah pada suatu citra masukan. Algoritma ini mengaplikasikan *weak classifier* pada setiap fitur yang ada pada data latih [5]. Suatu *weak classifier* dinyatakan dengan persamaan:

$$h_{\mathrm{j}}(x) = \begin{cases} 1 & untuk \;\; p_{\mathrm{j}} f_{\mathrm{j}}(x) < p\Theta_{\mathrm{j}} \\ -1 & untuk \;\; yang \;\; lainnya \end{cases} \tag{2.4}$$

di mana $h_j(x)$ menyatakan *weak classifier* yang memiliki fitur $f_j(x)$ dengan *threshold* Θ_j dan paritas p_j. *Weak classifier* bernilai 1 apabila diklasifikasikan sebagai objek wajah dan bernilai -1 apabila diklasifikasikan sebagai objek non-wajah. *AdaBoost* akan menyeleksi fitur-fitur yang akan digunakan dengan cara memberi bobot kemiripan. Fitur yang memiliki nilai *error* yang minimal akan memiliki bobot lebih besar.

Setelah *Weak classifier* telah diindentifikasi, selanjunya *weak classifier* digunakan untuk menyusun suatu *stron classfier*. Langkah-langkah untuk menyusun *strong classifier* ditunjukan dengan algoritma lengkap *AdaBoost* berikut [14] yang ditunjukkan pada Algoritma 1.

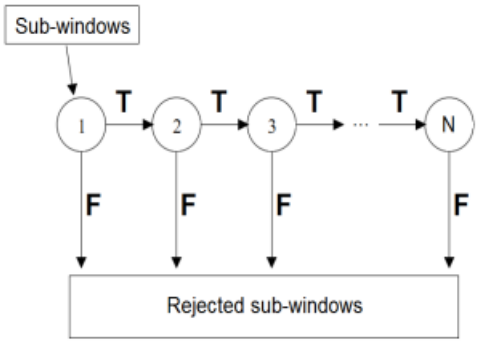
2.2.3 Cascade of Classifier

Metode *Viola-Jones* menggunakan teknik pengklasifikasian objek menggunakan fitur *haar-like* dengan menerapkannya pada seluruh gambar di citra masukan [7]. Pada metode ini, pengklasifikasian objek terdiri dari beberapa tingkatan dengan menggunakan skema *Cascade*. Skema *cascade* digunakan sebagai teknik untuk menyusun dan menggunakan beberapa *classifier* pada suatu sistem. Dengan menggunakan teknik ini, kecepatan pendeteksian objek akan meningkat karena hanya menfokuskan pada daerah citra yang berpeluang saja [12].

Setiap tingkatan penyeleksian *classifier* didapatkan dengan melakukan pembelajaran data latih menggunakan algoritma *AdaBoost*. Dengan menggunakan *AdaBoost*, *threshold* suatu fitur akan ditentukan dan digunakan untuk meminimalkan angka *false negative*. Secara umum, nilai *threshold* yang rendah akan

Algorithm 1 Algoritma *AdaBoost*

1: Diberikan suatu gambar (x_1,y_1), ..., (x_n,y_n) di mana $y_i = 0$ untuk contoh positif dan $y_1 = 1$ untuk contoh negatif
2: Inisialisasi bobot $w_{1,\mathrm{i}} = \frac{1}{2m}, \frac{1}{2l}$ di mana m dan l adalah nilai negatif dan positif
3: **For** $t = 1$ **to** T **do**
4: Normalisasi bobot sehingga w_t menjadi distribusi probabilitas,
5: $$w_{\mathrm{t,i}} \leftarrow \frac{w_{\mathrm{t,i}}}{\sum_{j=1}^{n} w_{\mathrm{t,j}}}$$
6: Untuk setiap fitur j, melatih *classifier* h_j untuk setiap fitur tunggal.
7: *Error* (ϵ_j) dievaluasi dengan bobot w_t,
8: $$\epsilon_\mathrm{j} = \sum_i w_\mathrm{i} |h_\mathrm{j}(x_\mathrm{j}) - y_\mathrm{i}|.$$
9: Pilih classifier h_t dengan nilai kesalahan (ϵ_t) paling sedikit.
10: Memperbarui bobot,
11: $$w_{\mathrm{t+1,i}} = w_{\mathrm{t,i}} {\beta_\mathrm{t}}^{1-\mathrm{e_i}}$$
12: di mana $e_\mathrm{i} = 0$ jika x_i diklasifikasikan dengan benar, $e_\mathrm{i} = 1$ lainnya
13: dan $\beta_\mathrm{t} = \frac{\epsilon}{1-\epsilon_\mathrm{t}}$
14: **Endfor**
15: Hasil *strong classifier* adalah
16: $$h(x) = \begin{cases} 1, & \sum_{t=1}^{T} \alpha_\mathrm{t} h_\mathrm{t}(x) \geq \frac{1}{2} \sum_{t=1}^{T} \alpha_\mathrm{t} \\ 0, & \text{lainnya} \end{cases}$$. di mana $\alpha_\mathrm{t} = log \frac{1}{\beta_\mathrm{t}}$



Gambar 2.4: Skema *Cascade* pendeteksian

menghasilkan tingkat deteksi yang lebih tinggi dan tingkat *false positive* yang lebih tinggi [14].

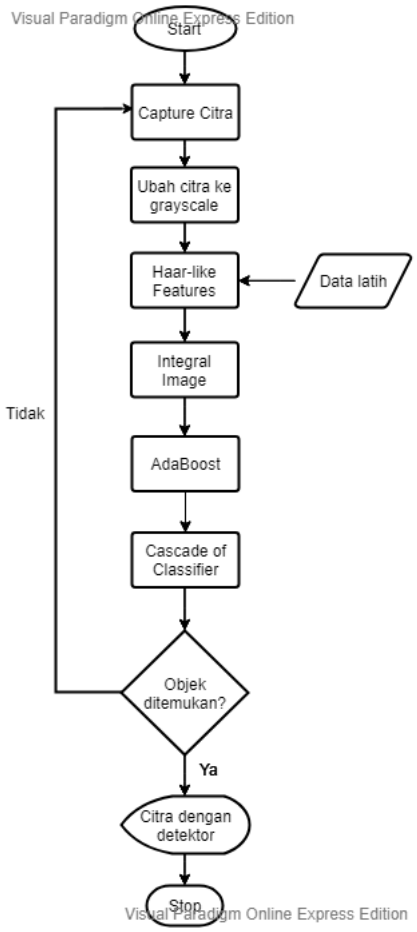
Gambar 2.4 merupakan skema pendeteksian objek menggunakan teknik *cascading*. Gambar tersebut menunjukkan tahapan pendeteksian dengan skematik dengan N tahapan [7]. Suatu *sub-window* diambil dari citra masukan dan kemudian akan diseleksi dengan menggunakan suatu *classifier*. Hasil positif

classifier pada *sub-window* akan memicu munculnya evaluasi *classifier* kedua dan seterusnya hingga mendapatkan objek yang diinginkan. Hasil negatif penyeleksian *classifier* akan ditolak dan tidak akan dilibatkan untuk penyeleksian *classifier* berikutnya [14] serta diklasifikasikan sebagai bentuk non-objek.

Bab III

Metodologi dan Desain Sistem

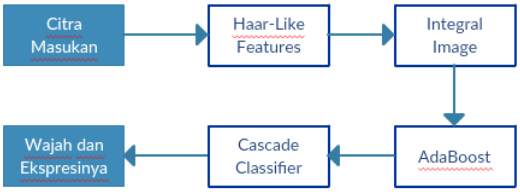
3.1 Flowchart sistem



Gambar 3.1: Flowchart sistem

3.2 Skema Umum

Sistem yang akan dikerjakan umumnya didesan dengan tujuan dapat menerapkan prinsip kerja Metode *Viola-Jones* dalam pengolahan citra wajah. Secara khusus, sistem yang dimaksud diharapkan dapat menerapkan prinsip kerja tersebut dalam mendeteksi serta mengenali wajah secara *real time* beserta ekspresinya menggunakan *webcam*. *Webcam* akan digunakan sebagai alat untuk dapat menangkap citra *real time* agar dapat diolah. Berikut adalah skema umum pendeteksian menggunakan Metode *Viola-Jones*.



Gambar 3.2: Skema proses pendeteksian *Viola-Jones*

Dalam pengerjaannya, penulis akan menggunakan *Python* sebagai bahasa pemrograman dan pustaka Open Computer Vision (OpenCV). Menurut Li [4] dalam papernya yang berjudul *An improved algorithm on Viola-Jones object detector*, OpenCv adalah suatu pustaka yang merupakan bentuk implementasi dari Metode *Viola-Jones*. Oleh karena itu, pengerjaan sistem dilakukan dengan menggunakan beberapa fungsi dan *class* yang telah tersedia pada OpenCV serta membentuk suatu *rectangle* sebagai detektor visual. Sebagai batasan pengerjaan, penulis akan menggunakan data latih yang diambil bebas dari Internet serta hanya akan mendeteksi ekspresi senyum pada wajah.

Pengerjaan sistem dimulai dengan proses penangkapan citra *real time* menggunakan *webcam*. Kemudian akan dilakukan pembacaan fitur *Haar-Like* menggunakan pustaka OpenCV. Pembacaan fitur ini dilakukan dengan mengimpor data latih ke dalam *class CascadeClassifier* pada modul OpenCV. Langkah selanjutnya, akan dibuat suatu fungsi untuk melakukan pendeteksian. Fungsi tersebut memiliki parameter *gray* dan *frame*. Parameter *gray* digunakan untuk menkonversi tangkapan citra berwarna menjadi citra *greyscale*. Citra *grayscale* digunakan dalam pendeteksian untuk mempermudah pembacaan fitur dengan tingkat kecepataan yang lebih tinggi. Parameter *frame* merupakan citra asli berwarna yang akan digunakan kembali untuk menampilkan hasil olahan citra dalam bentuk citra berwarna. Di dalam fungsi tersebut akan dilakukan pendeteksian objek yang dinginkan dan akan didesain suatu *rectangle* yang digunakan sebagai detektor visual objek pada citra. Pada proses pendeteksian objek, akan digunakan fungsi detectMultiScale() dari pustaka OpenCV. Fungsi tersebut memiliki parameter *gray*, *scaleFactor*, dan *minNeighbors*. *Gray* merupakan citra *grayscale*, *scaleFactor* merupakan parameter

untuk menentukan rasio seberapa banyak ukuran gambar berkurang pada setiap skala gambar. *minNeighbors* merupakan jumlah minimal persegi di sekitar yang akan membentuk suatu objek. Sebagai langkah terakhir, *rectangle* akan dibentuk untuk menentukan daerah objek yang diinginkan. Hasil keluaran dari program tersebut akan mengembalikan *frame* dengan adanya detektor visual berupa *rectangle* jika objek yang dinginkan terdeteksi.

3.3 Algoritma

Atau dalam bentuk algoritma seperti contoh pada Algoritma 2 berikut ini:

Algorithm 2

```
1: Start
2: Import OpenCv
3: Define faceCascade = cv2.CascadeClassifier "frotal face cascade"
4: Define smileCascade = cv2.CascadeClassifier "smile cascade"
5: function DETECT(gray, frame)
6:     Define faces = faceCascade.detectMultiScale()
7:     For x, y, w, h in faces  do
8:         cv2.rectangle(frame, (x, y), (x + w, y + h), RGB, lineWidth)
9:         Define roiGray = gray[y : y + h, x : x + w]
10:        Define roiColor = frame[y : y + h, x : x + w]
11:        Define smile = smileCascade.detectMultiScale()
12:            For sx, sy, sw, sh in smile do
13:                cv2.rectangle(roiColor, (sx, y), (sx + sw, sy + sh), RGB, lineWidth)
14:            EndFor
15:    EndFor
16:    return frame
17:    Define videoCapture = cv2.VideoCapture()
18:    While true:
19:        Define frame = video_capture.read()
20:        Define gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
21:        Define canvas = detect(gray, frame)
22:        If cv2.waitKey(1)0xFF == ord('q') : then
23:            break
24:    End
25:
```

Bab IV

Preliminary

4.1 Capaian Sekarang

Saat ini, Sistem Pengenalan Ekspresi Wajah Secara Real Time Menggunakan Metode Viola-Jones telah dapat mendeteksi dan mengenali wajah secara *real time* melalui *webcam* bawaan dari perangkat seperti pada Gambar 4.2.

Gambar 4.1: Pengenalan Wajah oleh Sistem

Dari gambar di atas, terlihat bahwa sistem telah dapat mengenali wajah dengan baik. Wajah dapat dikenali dengan ditandai dengan adanya *rectangle* putih. *Rectangle* tersebut akat terus menandai wajah selama berada di area cakupan kamera.

4.2 Capaian Selanjutnya

Hal yang akan dicapai selanjutnya adalah sistem yang dikerjakan akan dikembangkan sehingga sistem dapat mendeteksi dan mengenali ekspresi wajah secara *real time* dari citra yang diberikan. Ekspresi wajah yang akan dikenali dibatasi dengan pengenalan ekspresi senyum pada wajah yang menandakan emosi kebahagiaan seseorang.

4.3 Pelaporan Progres

4.3.1 Penambahan Label pada *Rectangle*

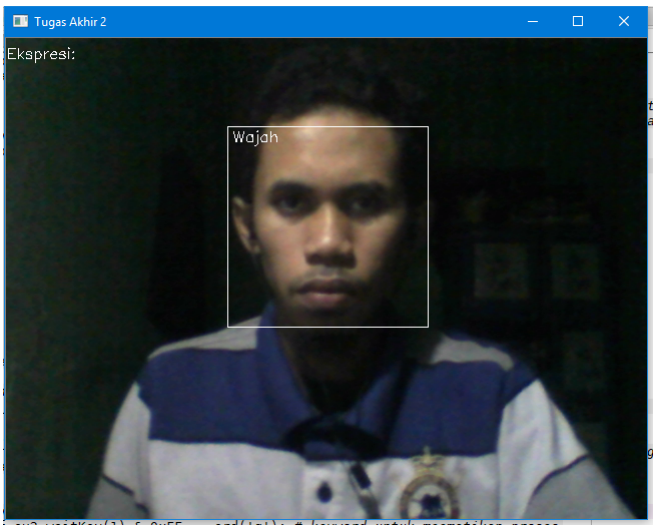
Label telah ditambahkan pada jendela *output* tepat pada sudut kiri atas dari *rectangle* dan label pengenalan ekspresi pada sudut kiri atas jendela *output*. Untuk label pada *rectangle*, ditambahkan *class putText()* dari Pustaka OpenCv sebagai berikut.

- font = cv2.FONTHERSHEYSIMPLEX
- cv2.putText(frame, 'Wajah', (x+5,y+15), font, 0.5, (255,255,255))

Variabel *font* adalah variabel yang digunakan untuk menginisialisasi penggunaan jenis huruf pada label Penjelasan parameter pada *class putText()*:

- frame : Gambar berwarna pada hasil *output*
- 'Wajah' : Teks yang ditampilkan sebagai label pada jendela *output*
- (x+5, y+15) : Tuple untuk menjelaskan lokasi label
- font : variabel untuk penggunaan jenis huruf pada OpenCV
- 0.5 : Ketebalan dari teks label yang ditampilkan
- (255,255,255) : Tuple untuk memberi warna pada teks label (RGB)

Hasil dari pemberian label pada jendela *output*:



Gambar 4.2: Pemberian Label

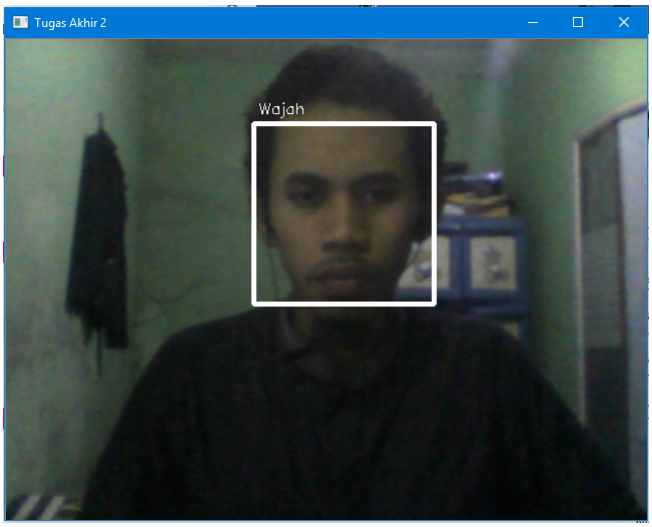
Note: Data latih yang digunakan masih menggunakan data latih diambil dari salah satu repository pada github.com

4.3.2 Perancangan Data Training

1. Data mentah berupa gambar potongan wajah *greyscale* berdimensi 320 x 243 pixel didapatkan dari *database* wajah Yale University
2. Semua *file* diubah menjadi ekstensi .jpg
3. Gambar negatif dan gambar positif dipisahkan
    - Untuk wajah keseluruhan
    - Untuk ekspresi pada wajah
4. Sampel data latih dibuat, dan dikonversi ke ekstensi *.vec*
5. Melakukan *training* pada *classifier* yang dibentuk
6. Hasil dari poin sebelumnya, akan menghasilkan file *.xml* sebagai hasil akhir perancangan data latih

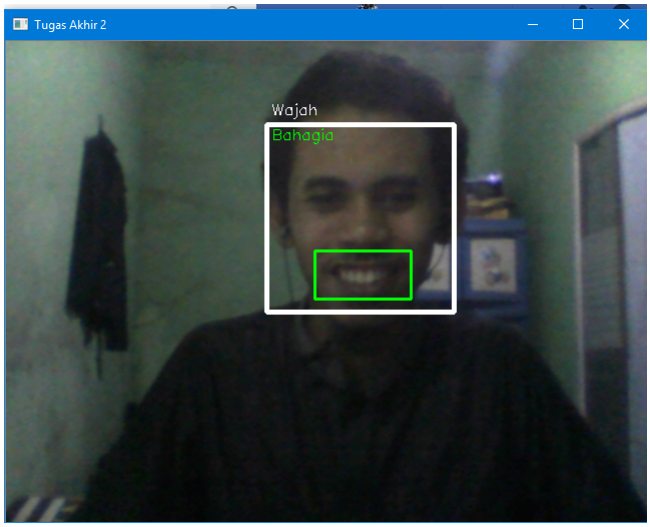
Note: Proses perancangan masih mengalami masalah pada saat membuat sampel data latih

4.3.3 Pengealan Ekspresi



Gambar 4.3: Hasil akhir: kondisi normal

Gambar 4.3 menunjukkan hasil akhir pendeteksian dengan kondisi wajah normal dan tanpa ekspresi. Pada gambar, hanya menampilkan *rectangle* sebagai alat pendeteksian visual untuk menujukkan adanya objek wajah di depan kamera *webcam* beserta labelnya. Sedangkan gambar 4.4, merupakan hasil akhir yang diharapkan dari perancangan sistem ini. Ekspresi senyum dapat



Gambar 4.4: Hasil akhir: dengan ekspresi

terdeteksi dengan baik dan akurat oleh sistem yang dibangun. *Rectangle* berwarna hijau menujukkan objek mulut yang sedang senyum di depan *webcam*, serta label berwarna hijau menunjukkan penamaan ekspresi yang terdeteksi oleh sistem.

Langkah pendeteksiannya adalah sebagai berikut.

1. Data latih $haarcascade_smile.xml$ diproses ke dalam kelas $detectMultiScale()$ pada OpenCV

2. *Looping* ditambahkan pada baris kode untuk memproses pendeteksian ekspresi.

Note: Data latih yang digunakan masih menggunakan data latih diambil darisalah satu repository pada github.com

Daftar Pustaka

[1] AD Egorov. Algorithm for optimization of viola–jones object detection framework parameters. In *Journal of Physics: Conference Series*, volume 945, page 012032. IOP Publishing, 2018.

[2] Paul Ekman and Dacher Keltner. Universal facial expressions of emotion: An old controversy and new findings. 1997.

[3] S Heranurweni. Pengenalan wajah menggunakan learning vector quantization (lvq). *Prosiding SNST Fakultas Teknik*, 1(1), 2010.

[4] Qian Li, Usman Niaz, and Bernard Merialdo. An improved algorithm on viola-jones object detector. In *Content-Based Multimedia Indexing (CBMI), 2012 10th International Workshop on*, pages 1–6. IEEE, 2012.

[5] Hamed Masnadi-Shirazi. Adaboost face detection.

[6] David Matsumoto and Cenita Kupperbusch. Idiocentric and allocentric differences in emotional expression, experience, and the coherence between expression and experience. *Asian Journal of Social Psychology*, 4(2):113–131, 2001.

[7] R Padilla, CFF Costa Filho, and MGF Costa. Evaluation of haar cascade classifiers designed for face detection. *World Academy of Science, Engineering and Technology*, 64:362–365, 2012.

[8] James Pao. Emotion detection through facial feature recognition.

[9] Constantine P Papageorgiou, Michael Oren, and Tomaso Poggio. A general framework for object detection. In *Computer vision, 1998. sixth international conference on*, pages 555–562. IEEE, 1998.

[10] M Dwisnanto Putro, Teguh Bharata Adji, and Bondhan Winduratna. Sistem deteksi wajah dengan menggunakan metode viola-jones. 2012.

[11] Wasim Ahmed Syed, Arun Kumar Uppalapti, Amruth Vadakattu, Akhil Vasireddy, and Purna Chandra Reddy. Emotion detection through facial feature recognition. *Emotion*, 5(02), 2018.

[12] Andrianus Hendro Triatmoko, Sholeh Hadi Pramono, and Harry S Dachlan. Penggunaan metode viola-jones dan algoritma eigen eyes dalam sistem kehadiran pegawai. *Jurnal EECCIS*, 8(1):41–46, 2014.

[13] Esty Vidyaningrum and Prihandoko Prihandoko. Human face detection by using eigenface method for various pose of human face, 2010.

[14] Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pages I–I. IEEE, 2001.

[15] Yi-Qing Wang. An analysis of the viola-jones face detection algorithm. *Image Processing On Line*, 4:128–148, 2014.

Lampiran