

## RANCANG BANGUN CAMERA FACE TRACKER DENGAN MENGUNAKAN METODE HAAR-LIKE FEATURE DAN PID

RD. Kusumanto, S.T., M.M<sup>1</sup>, Wahyu Setyo Pambudi, ST., MT<sup>2</sup>, Alan Novi Tompunu, ST., MT<sup>1</sup>, Bon Maria Nurintan Simorangkir, ST<sup>2</sup>

1. Jurusan Teknik Komputer, Politeknik Negeri Sriwijaya, Palembang 30139

2. Jurusan Teknik Elektro Fakultas Teknologi Industri Universitas Internasional Batam

Email : [manto\\_6611@yahoo.co.id](mailto:manto_6611@yahoo.co.id); [yoe2fa@yahoo.com](mailto:yoe2fa@yahoo.com); [alan\\_polsri@yahoo.com](mailto:alan_polsri@yahoo.com) ;

### Abstract

Computer Vision is one branch of the Field Science Image Processing which allows computers to see like humans, recognize objects and make decisions one example application is face detection, which later developed as the face tracker.

Haar-like feature is a feature based on the Haar wavelet. Haar wavelet is a single square wave (high and low). For two dimensions, one light and one dark. The workings of haar-like features in a face detection technique is to use a sliding window, the whole picture and try to see if there is a part of the image that is shaped like a face or not.

Controller Proportional-Integral-Derivative (PID) closed loop control model is a widely used in industrial. This control model will calculate the error as the value of the difference between the measured results with the desired set point value. Controller PID will minimize the error by adjusting the process control inputs.

In this study utilizing Haar-like feature and PID to perform the face tracker system, starting with the webcam detects a face, using EmguCV will determine the required set-point motor driver to drive the motor with a PID controller, thus motor will follow the movement of the face to the left and or right.

Based on the test results in this study, the system can recognize the faces with optimal response when the value of  $K_p = 0.8$ ,  $K_i = 0001$  and  $K_d = 0.05$  and an average error of 2%.

**Keyword :** Computer Vision, Haar-like feature, PID Controller, Face detection, Face tracker.

### Abstrak

Computer Vision adalah salah satu cabang dari Bidang Ilmu Pengolahan Citra (Image Processing) yang memungkinkan komputer dapat melihat seperti manusia, seperti mengenali terhadap suatu obyek dan mengambil keputusan, bentuk aplikasinya adalah face detection yang dapat dikembangkan menjadi face tracker.

Haar-like features adalah feature yang didasarkan pada haar wavelet. Haar wavelet adalah sebuah gelombang kotak (high dan low) untuk dua dimensi maka menjadi gelap dan terang. Prinsip kerja haar-like features pada teknik deteksi wajah adalah menggunakan sliding window, semua gambar akan dicari ada atau tidaknya bagian seperti wajah.

Kontrol Proportional-Integral-Derivative (PID) adalah sistem kontrol yang aplikasinya banyak digunakan secara luas pada industri. Kontrol ini akan menghitung nilai error yang dihasilkan dari perbedaan antara hasil pengukuran dengan nilai setpoint. Kontrol PID ini akan meminimalisasi error dengan mengatur proses inputnya.

Pada penelitian ini yang menggabungkan antara haar-like feature dan PID untuk mendapatkan sistem face tracker, dimulai dengan mendeteksi wajah menggunakan webcam menggunakan EmguCV akan menentukan setpoint PID untuk mengatur motor, kemudian motor akan mengikuti pergerakan wajah ke kiri dan kanan. Berdasarkan pengujian pada penelitian ini didapatkan nilai bahwa respon optimal pada saat  $K_p=0.8$ ,  $K_i=0.001$  dan  $K_d=0.05$  dengan error rata-rata 2%.

**Keyword :** *Computer Vision, Haar-like feature, Kontrol PID, Face detection, Face tracker.*

## 1. Pendahuluan

Perkembangan teknologi yang terkait dengan pemrosesan menggunakan *computer* demikian pesatnya, dimana pemrosesan *image* (citra) yang awalnya menggunakan teknologi analog mulai bergeser menggunakan teknologi digital. Disiplin ilmu yang mempelajari tentang teknik-teknik mengolah citra secara digital disebut sebagai Pengolahan citra digital (*Digital Image Processing*). Proses digital disini mempunyai maksud bahwa pengolahan citra dilakukan secara digital menggunakan komputer [1]. Dimana *OpenCV* dan *EmguCV* adalah sebuah API (*Application Programming Interface*) *Library* yang sudah sangat familiar pada pengolahan citra.

*Computer Vision* adalah salah satu cabang dari Bidang Ilmu Pengolahan Citra (*Image Processing*) yang memungkinkan komputer dapat melihat seperti manusia [2]. Dengan *computer vision* tersebut komputer dapat mengambil keputusan, melakukan aksi, dan mengenali terhadap suatu Objek. Beberapa implementasi dari *Computer Vision* adalah *Face Recognition, face detection, face/object tracking, road tracking, etc* [3]. Sejalan dengan waktu, bahwa *face detection* dapat dikembangkan sebagai *face tracker*, dimana prinsipnya adalah suatu objek wajah yang tertangkap oleh kamera akan diikuti pergerakannya ke kiri dan ke kanan.

Pada desain *facetracker* ini akan menggunakan *webcam*, serta pendeteksi wajah menggunakan sistem *haar-like features* dan pergerakannya menggunakan *PID (Proportional Integral Derivative)*. *System Haar-Like features* pertama kali diteliti oleh Viola dan Jones kemudian dikembangkan oleh Lienhart [4]. Metode ini menghitung perbedaan jumlah setiap piksel pada daerah persegi panjang (*rectangular*) yang berdekatan pada lokasi tertentu dalam jendela deteksi. Berdasarkan hal tersebut maka penelitian ini akan dikembangkan tentang model simulasi untuk *face tracker*, dimana pergerakan wajah secara horizontal akan diikuti oleh *webcam*.

Pada desain ini sistem yang dibangun terdiri dari *haar-like features* untuk mendeteksi wajah sedangkan *control* pergerakan motor menggunakan sistem *PID*. Berdasarkan ke-dua metode tersebut maka penelitian ini akan mengembangkan *face tracker* dengan pendekatan terhadap pola gerakan dari wajah. Desain *hardware*, gerak mekanis dan sistem kendali *PID* disesuaikan dengan motor DC yang digunakan, sehingga dapat melakukan gerakan yang lebih optimal untuk mencapai posisi tertentu. Posisi awal citra dari *webcam* dengan adanya masukan data posisi obyek dari *PC (Personal Computer)*, *PC* akan mengirimkan informasi citra melalui *webcam* yang akan di proses *PC* akan mengirimkan informasi citra melalui *webcam* yang akan di proses dengan sistem *Haar-like features*, untuk mendeteksi posisi dengan menghitung error yang terjadi yang menggunakan sistem *PID*, hasil citra akan dikirim ke *driver* motor oleh mikrokontroler jenis *ATMEGA*

8535, kemudian hasil proses konversi akan dikomunikasikan secara serial menuju *PC (Personal Computer)*.

## 2.1 LITERATURE REVIEW

### 2.1.1. Metode Haar Cascade Classifier

Proses deteksi adanya citra mata dalam sebuah gambar pada *OpenCV*, menggunakan sebuah metode yang dipublikasikan oleh Paul Viola dan Michael Jones tahun 2001. Umumnya disebut metode *Haar Classifier*, metode ini merupakan metode yang menggunakan *statistical model (classifier)* [3]. Pendekatan untuk mendeteksi objek dalam gambar menggabungkan empat konsep utama :

1. Training data.
2. Fitur segi empat sederhana yang disebut fitur Haar.
3. Integral image untuk pendeteksian fitur secara cepat.
4. Pengklasifikasi bertingkat (*Cascade classifier*) untuk menghubungkan banyak fitur secara efisien,

### 2.1.2 Training data pada Haar

Metode ini memerlukan 2 tipe gambar objek dalam proses *training* yang dilakukan, yaitu :

#### a. Positive samples

Berisi gambar obyek yang ingin di deteksi, apabila ingin mendeteksi mata maka *positive samples* ini berisi gambar mata, begitu juga obyek lain yang ingin dikenali.

#### b. Negative samples

Berisi gambar obyek selain obyek yang ingin dikenali, umumnya berupa gambar *background* (tembok, pemandangan, lantai, dan gambar lainnya). Resolusi untuk sampel negatif disarankan untuk memiliki resolusi yang sama dengan resolusi kamera.

*Training* dari Haar menggunakan dua tipe sampel diatas. Informasi dari hasil *training* ini lalu dikonversi menjadi sebuah parameter model statistik.

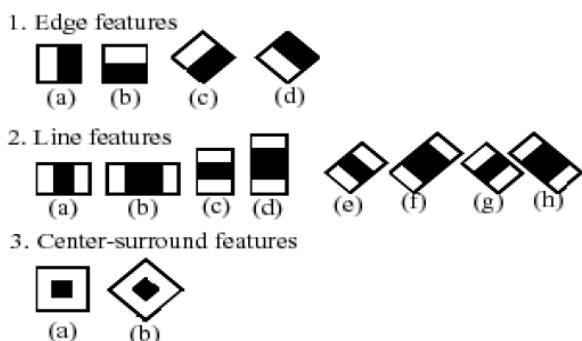
### 2.1.3 Sistem Kerja Algoritma Haar Cascade Classifier

Algoritma Haar menggunakan metode statistik dalam melakukan pengenalan mata. Metode ini menggunakan *simple haar-like features* dan juga *a cascade of boosted tree classifier* [4]. Classifier ini menggunakan gambar berukuran tetap. Cara kerja dari haar dalam mendeteksi mata adalah menggunakan teknik *sliding window* pada keseluruhan gambar dan mencari apakah terdapat bagian dari gambar yang berbentuk seperti mata atau tidak. Haar juga memiliki kemampuan untuk melakukan *scaling* sehingga dapat mendeteksi adanya mata yang berukuran lebih besar ataupun lebih kecil dari gambar pada *classifier*.

Tiap *feature* dari *haar-like feature* didefinisikan pada bentuk dari *feature*, diantaranya koordinat dari *feature* dan juga ukuran dari *feature* tersebut.

### 2.1.4 Haar Feature

*Haar Feature* adalah fitur yang digunakan oleh Viola dan Jones didasarkan pada *Wavelet Haar*. *Wavelet Haar* adalah gelombang tunggal bujur sangkar (satu interval tinggi dan satu interval rendah) [3]. Untuk dua dimensi, satu terang dan satu gelap. Selanjutnya kombinasi-kombinasi kotak yang digunakan untuk pendeteksian objek visual yang lebih baik. Setiap *Haar-like feature* terdiri dari gabungan kotak - kotak hitam dan putih [3,4].



**Gambar 2.1 Haar Feature**

(Sumber : Viola, Paul and Michael Jones , 2001).

3 tipe kotak(rectangular) feature:

Tipe *two-rectangle feature* (horizontal/vertikal)

Tipe *three-rectangle feature*

Tipe *four-rectangle feature*

Adanya fitur Haar ditentukan dengan cara mengurangi rata-rata piksel pada daerah gelap dari rata-rata piksel pada daerah terang. Jika nilai perbedaannya itu diatas nilai ambang atau treshold, maka dapat dikatakan bahwa fitur tersebut ada.

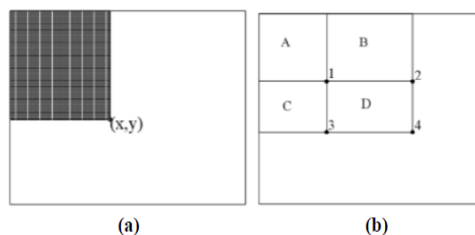
Nilai dari *Haar-like feature* adalah perbedaan antara jumlah nilai-nilai piksel *gray level* dalam daerah kotak hitam dan daerah kotak putih:

$$f(x) = \text{Sumblack rectangle} - \text{Sumwhite rectangle} \dots\dots\dots(2.1)$$

Kotak Haar-like feature dapat dihitung secara cepat menggunakan “*integral image*”

### 2.1.5 Integral Image

*Integral Image* digunakan untuk menentukan ada atau tidaknya dari ratusan fitur Haar pada sebuah gambar dan pada skala yang berbeda secara efisien. Pada umumnya, pengintegrasian tersebut berarti menambahkan unit-unit kecil secara bersamaan. Dalam hal ini unit-unit kecil tersebut adalah nilai-nilai piksel. Nilai integral untuk masing-masing piksel adalah jumlah dari semua piksel-piksel dari atas sampai bawah. Dimulai dari kiri atas sampai kanan bawah, keseluruhan gambar itu dapat dijumlahkan dengan beberapa operasi bilangan bulat per piksel.



**Gambar 2.2 Integral Image**

(Sumber : McAndrew Alasdair, 2004)

### 2.1.6. Proportional Integral Derivative (PID) Control.

Kontroler *proporsional-integral-derivatif* (PID controller) adalah model *control close loop* yang banyak digunakan secara luas pada industrial sistem kontrol. Model kontrol ini akan menghitung *error* sebagai nilai perbedaan antara hasil proses terukur dengan nilai *setpoint* yang diinginkan. Kontroler ini akan mencoba untuk meminimalkan kesalahan dengan menyesuaikan *input* proses control [6].

Sebuah kontroler PID dapat dianggap sebagai bentuk pengendali dengan kompensator fasa *lead-lag* dari nilai *setpoint* titik awal sampai tak terhingga. Demikian pula dengan kontroler sejenis, yaitu PI dan PD dapat juga dianggap sebagai bentuk pengendali dengan kompensator dari fasa *lag* dan fasa *lead* secara berurutan [7]. Sebuah kontroler PID standar juga dikenal sebagai pengendali *three-term* atau tiga kondisi, yang fungsi transfer umumnya ditulis dalam bentuk paralel seperti persamaan 2.7.

$$G(s) = K_p + \frac{K_i}{s} + K_d s \dots\dots\dots(2.2)$$

dimana nilai  $K_p$  untuk *proportional gain*,  $K_i$  untuk *integral gain* dan  $K_d$  untuk *derivative gain*.

Pengendali tiga kondisi tersebut mempunyai fungsi masing-masing, diantaranya adalah :

- Proportional term* - memberikan aksi kontrol secara proporsional terhadap respon sinyal error melalui *all-pass gain factor*.
- Integral term* - mengurangi *error steady-state* melalui kompensasi frekwensi rendah dengan sebuah *integrator*.
- Derivative term* - meningkatkan respon transien melalui kompensasi frekwensi tinggi dengan sebuah *differentiator*.

Controller PID dan Analisa Respon Transien Efek dari perubahan nilai untuk masing-masing kondisi tersebut, terdapat pada **tabel 2.1** berikut ini :

**Tabel 2.1. Efek kontrol terhadap perubahan tiap-tiap kondisi**

| Closed-Loop Response | Rise Time      | Overshoot | Settling Time  | Steady-State Error | Stability |
|----------------------|----------------|-----------|----------------|--------------------|-----------|
| Increasing $K_p$     | Decrease       | Increase  | Small Increase | Decrease           | Degrade   |
| Increasing $K_i$     | Small Decrease | Increase  | Increase       | Large Decrease     | Degrade   |
| Increasing $K_d$     | Small Decrease | Decrease  | Decrease       | Minor Change       | Improve   |

Sumber : Ang. K. H. dkk, 2005

Adanya perkembangan computer dan embedded system, maka metode kendali PID ini dapat dijalankan secara digital atau dengan menggunakan program. Agar dapat mengimplementasikan sistem kendali PID secara digital, maka persamaan PID harus diubah ke dalam persamaandiskrit [6,7].

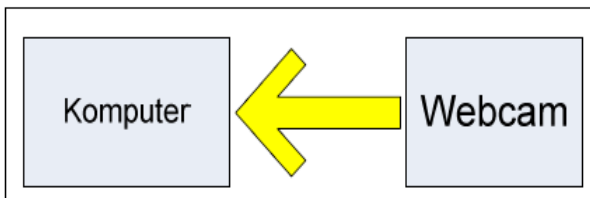
$$V_o = K_p + K_i \int edt + K_d \frac{de}{dt}$$

### 3.1 Metodologi Penelitian.

### 3.2. Perancangan Sistem

Perancangan perangkat keras dari sistem ini terdiri atas beberapa bagian utama. Secara garis besar dapat dikategorikan sebagai berikut:

1. Komputer, digunakan untuk menampilkan wajah.
2. Webcam untuk mengenal objek wajah.



**Gambar 3.1. Blok Perangkat Keras**

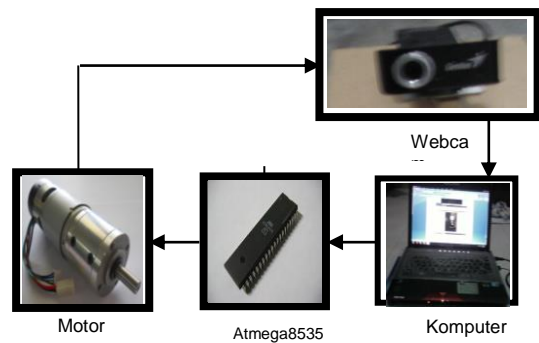
### 3.2.1 Analisis Perancangan sistem

Perancangan perangkat keras dari sistem ini terdiri atas beberapa bagian utama. Secara garis besar dapat dikategorikan sebagai berikut:

1. Perangkat pemroses utama (minimum sistem mikrokontroler ATmega8535)

### Blok Diagram Sistem

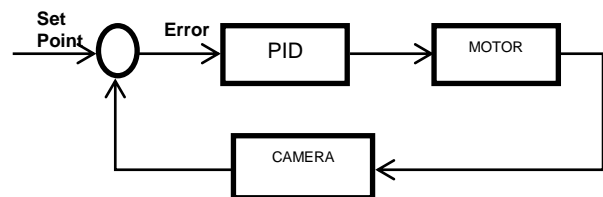
Perancangan sistem dapat digambarkan dalam bentuk blok diagram seperti yang diperlihatkan pada Gambar 3.2. Sistem terdiri atas 4 bagian utama yaitu sensor vision (Webcam), PC *embedded*/Laptop, mikrokontroler ATmega8535 dan motor DC yang dilengkapi dengan Diver Motor. Sensor vision yang ditempatkan pada motor DC sebagai pengolahan citra.



**Gambar 3.2. Diagram Blok Sistem Kamera**

Sumber : Diolah dari Data Primer (4 Jun 2012)

Sensor ini akan memberikan informasi citra yang terdapat di tangkap sensor ke komputer agar melakukan proses PID dan menghitung *error* yang terjadi melalui software Microsoft Visual Studio, lalu hasil bacaan sensor tersebut akan dikirim ke driver motor melalui mikrokontroler, dimana driver motor akan diatur oleh mikrokontroler untuk mengatur pergerakan motor DC, agar bergerak sesuai informasi dari sensor vision dan sensor vision dapat menangkap citra tepat di posisi tengah sesuai tampilan di komputer seperti yang diperlihatkan pada Gambar 3.3.

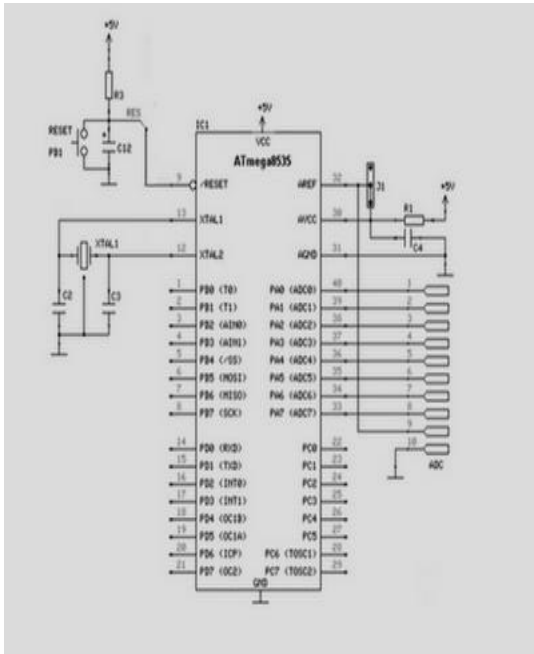


**Gambar 3.3. Blok Diagram**

### 3.2.2. Perancangan Minimum System

*Minimum system* adalah rangkaian paling dasar dimana chip mikrokontroler dapat bekerja. Mikrokontroler adalah suatu IC penyimpan dimana terdapat mikroprosesor dan memori program (ROM) serta memori RAM. Mikrokontroler AVR (*Alf and Vegard's Risc processor*) memiliki arsitektur RISC 8 bit, dimana semua instruksi dikemas dalam kode 16-bit dan sebagian besar instruksi dalam 1 siklus *clock*, berbeda dengan instruksi MCS 51 yang membutuhkan 12 siklus *clock*. *Minimum system* ini kemudian dihubungkan dengan rangkaian lain untuk menjalankan fungsi tertentu. Pada penelitian ini digunakan ATmega8535 sebagai IC mikrokontrollernya. Gambar di bawah merupakan diagram skematik *minimum system*.





Gambar 3.4. Diagram Skematik Minimum System

Untuk membuat sistem mikrokontroler ATmega8535 minimum yang digunakan ada beberapa pin, yaitu:

1. Serial RXD (PD0/RXD) pada port 14.
2. Serial TXD (PD1 / TXD) pada port 15.
3. Dir di PD7/OC2 di port 21 dan PD6/ICP1 di port 20.
4. PWM di PD4/OC 1B di port 18 dan PD5/OC 1A di port 19.

### 3.3 Motor DC.

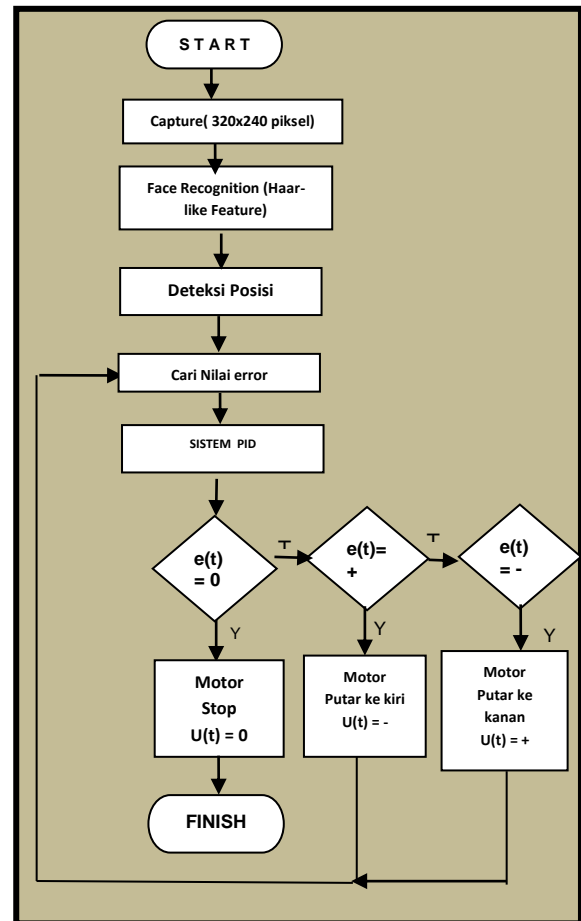
Dalam penelitian yang dilakukan ini, *actuator* yang digunakan berupa 3 buah motor DC yang telah dilengkapi sensor putaran. Motor DC yang digunakan menggunakan tipe GM42. GM42 ini merupakan motor DC dengan tegangan kerja sampai 24V dan dilengkapi reduksi gearbox 49:1, yang cocok untuk digunakan pada robot. Selain itu GM42 ini juga dilengkapi dengan Sensor putaran atau *rotary encoder*. Sensor ini menggunakan prinsip *hall effect sensor*. Keluaran dari *rotary encoder* ini terdiri dari sinyal A dan B, yang masing-masing mengeluarkan *pulse signal*. Sinyal A dan B tersebut akan berubah fasanya apabila terdapat perubahan arah putaran motor.



Gambar 3.5. Motor DC GM42

### 3.4 Flowchart Diagram

Adapun diagram blok dari sistem alat yang dirancang adalah sebagai berikut:



Gambar 3.6 Flow chart sistem Pengenalan Wajah berdasarkan Haar-like feature & PID

Cara kerja dari sistem dapat dilihat pada gambar diagram blok diatas. **Haar-Like Feature** digunakan dalam peng-capture obyek wajah. Face Tracker yang digunakan sebagai modul visi, memberikan data ke program pengendalian utama. Program kontrol robot meng-capture wajah. Dalam hal ini, 'pergerakan' objek wajah, yang tertangkap kamera sensor vision. Pergerakan tersebut dimana di *capture* dengan ukuran frame 320x240 pixel, dan akan dideteksi/face recognition dengan metode haar like features. Batas wajah di tengah visual lapangan dengan jarak  $\pm 2$  meter. Jika ukuran jelas dari wajah yang terdeteksi di titik tengah frame dengan nilai  $u(t) = 0$ , motor akan berhenti, tetapi jika sebaliknya yang terdeteksi lebih kecil (wajah lebih jauh) atau wajah yang terdeteksi tidak terpusat di tengah bidang pandang visual, sinyal kontrol dikirim ke program PID untuk mengurangi kesalahan (perbedaan antara nilai aktual dan nilai target). Sinyal kontrol yang dihasilkan dengan menambahkan output dari dua pengendali proporsional integrative dan derivatif. Pengendali PID kontroler proporsional dengan konstanta proporsionalitas untuk

istilah integratif dan derivatif (Ki dan Kd) ditetapkan ke 0.

### 3.5 Perancangan dan Pembuatan *Face Tracker*

#### 3.5.1 Kamera *Face Tracker*

Gambar di bawah ini merupakan gambar tampak depan, di mana kamera tersebut dibuat dengan posisi duduk dengan menggunakan Tupperware sebagai pondasinya untuk memudahkan meng-*capture* objek yang tegak lurus di hadapan kamera tersebut.



Gambar 3.7. Kamera *Face Tracker*

#### 3.5.2. Mesin Kamera *Face Tracker* Tampak keseluruhan

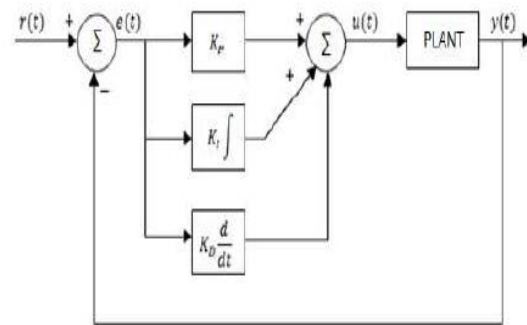
Mesin kamera *face tracker* Bagian keseluruhan *system* yang merupakan tempat untuk meletakkan rangkaian minimum sistem serta rangkaian untuk motor.



Gambar 3.8. Mesin Kamera *Face Tracker* Tampak Keseluruhan

### 3.6 Perancangan Perangkat Lunak

Desain dari perangkat lunak ini mengikuti diagram blok standar PID, dimana sinyal  $r(t)$  atau *setpoint/target* didapatkan dari posisi obyek berupa wajah. Sinyal  $e(t)$  atau *error* ke  $t$  diperoleh dari selisih antara posisi obyek  $r(t)$  dengan posisi penanda/*desired*  $y(t)$ .



Gambar 3.9. Blok Perangkat Lunak

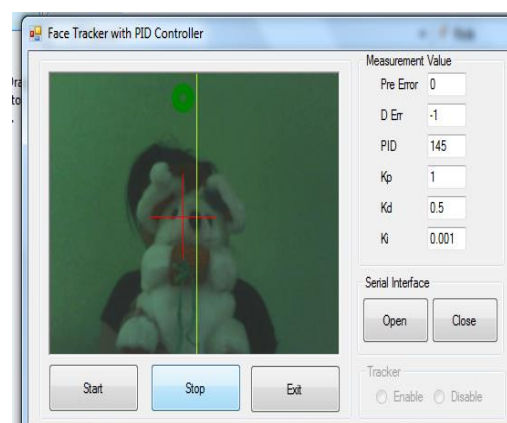
### 4.1. PENGUJIAN DAN ANALISA DATA

#### 4.2. Pengujian kamera *Face Tracker*

Faktor pencahayaan terhadap proses pelacakan wajah menunjukkan bahwa faktor pencahayaan tidak begitu berpengaruh pada proses pelacakan wajah karena hasil menunjukkan tidak ada yang gagal dalam pelacakannya, dalam arti 100% berhasil, akan tetapi ini tetap tergantung pada penangkapan oleh kamera secara kualitasnya, jika kamera tidak menangkap citra wajah maka proses pelacakan juga tidak dapat dilaksanakan. Tingkat keberhasilan pelacakan ini juga didukung oleh faktor pelatihannya yang tidak memperhatikan warna dan pengembangan citra.

- a. Pengujian dengan sampel wajah boneka.

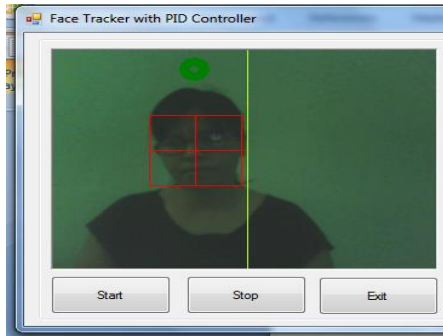
Pada sampel boneka dengan wajah yang ditutup dengan kain, dimana proses pelacakan wajah tidak terdeteksi. seperti halnya yang bisa dilihat pada gambar 4.1.



Gambar 4.1. wajah tidak terdeteksi

- b. Pengujian dengan wajah menggunakan kacamata.

Pada sampel yang menggunakan kacamata dan wajah yang bergerak ke-kanan dan ke-kiri dimana proses pelacakan wajah tetap terdeteksi, seperti halnya yang bisa dilihat pada gambar 4.2.



Gambar 4.2. wajah terdeteksi

### 4.3. Implementasi Posisi pada Kendali PID.

Pada implementasi dengan menggunakan *system* kendali PID dengan nilai  $K_p$ ,  $K_d$ , dan  $K_i$  yang di ubah-ubah dapat dilihat penampilan grafikya seperti pada gambar 4.15 – gambar 4.22, sampai menentukan nilai  $K_p$ ,  $K_d$ , dan  $K_i$  yang paling optimal dengan set point yang mendapatkan nilai error lebih kecil.



Gambar 4.3. Respon Kontrol *Facetracker* pada  $K_p=1$ ,  $K_d=1$  dan  $K_i=0.1$ .

Tabel 4.1. Nilai  $K_p = 1$ ,  $K_d = 1$ .  $K_i = 0.1$

| No | Set Point (Posisi x) | Actual (Posisi x) | Nilai Error Steady State |
|----|----------------------|-------------------|--------------------------|
| 1  | 0                    | 0                 | 0                        |
| 2  | 0                    | 0                 | 0                        |
| 3  | 0                    | 0                 | 0                        |
| 4  | 0                    | 0                 | 0                        |
| 5  | 160                  | 48                | 112                      |
| 6  | 226                  | 160               | 66                       |
| 7  | 226                  | 160               | 66                       |
| 8  | 323                  | 159               | 164                      |
| 9  | 240                  | 157               | 83                       |
| 10 | 240                  | 157               | 83                       |
| 11 | 240                  | 157               | 83                       |
| 12 | 240                  | 157               | 83                       |

|                 |     |     |       |
|-----------------|-----|-----|-------|
| 13              | 240 | 157 | 83    |
| 14              | 240 | 157 | 83    |
| 15              | 240 | 157 | 83    |
| 16              | 240 | 157 | 83    |
| 17              | 240 | 177 | 63    |
| 18              | 180 | 92  | 88    |
| 19              | 181 | 66  | 115   |
| 20              | 181 | 177 | 4     |
| 21              | 181 | 177 | 4     |
| 22              | 181 | 177 | 4     |
| 23              | 184 | 177 | 7     |
| 24              | 289 | 183 | 106   |
| 25              | 281 | 181 | 100   |
| 26              | 183 | 174 | 9     |
| 27              | 183 | 95  | 88    |
| 28              | 180 | 128 | 52    |
| 29              | 222 | 179 | 43    |
| 30              | 266 | 180 | 86    |
| 31              | 215 | 179 | 36    |
| 32              | 179 | 135 | 44    |
| 33              | 179 | 117 | 62    |
| 34              | 179 | 172 | 7     |
| 35              | 231 | 180 | 51    |
| 36              | 229 | 180 | 49    |
| 37              | 180 | 175 | 5     |
| 38              | 181 | 133 | 48    |
| 39              | 182 | 148 | 34    |
| 40              | 198 | 182 | 16    |
| 41              | 224 | 189 | 35    |
| 42              | 201 | 201 | 0     |
| 43              | 181 | 161 | 20    |
| 44              | 183 | 150 | 33    |
| 45              | 182 | 177 | 5     |
| 46              | 208 | 208 | 0     |
| 47              | 208 | 208 | 0     |
| 48              | 183 | 180 | 3     |
| 49              | 179 | 159 | 20    |
| 50              | 179 | 166 | 13    |
| Total Error     |     |     | 2322  |
| Rata-rata Error |     |     | 46.44 |

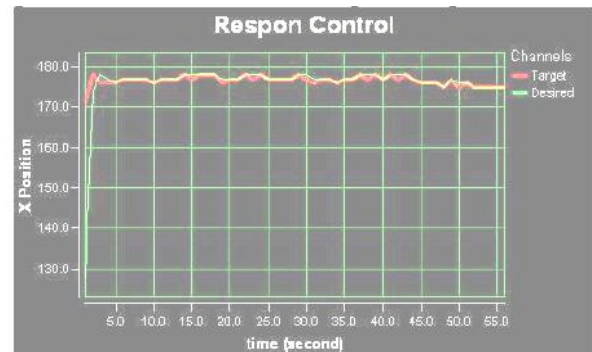


Gambar 4.4. Respon Kontrol *Facetracker* pada  $K_p=1$ ,  $K_d=0.05$  dan  $K_i=0.001$

Tabel 4.2. Nilai  $K_p=1$ ,  $K_d=0.05$ ,  $K_i=0.001$

| No | Set Point (Posisi x) | Actual (Posisi x) | Nilai Error Steady State |
|----|----------------------|-------------------|--------------------------|
| 1  | 159                  | 151               | 8                        |
| 2  | 165                  | 172               | 7                        |
| 3  | 165                  | 166               | 1                        |
| 4  | 165                  | 165               | 0                        |
| 5  | 165                  | 165               | 0                        |
| 6  | 165                  | 164               | 1                        |
| 7  | 164                  | 163               | 1                        |
| 8  | 163                  | 165               | 2                        |
| 9  | 165                  | 163               | 2                        |
| 10 | 163                  | 163               | 0                        |
| 11 | 163                  | 164               | 1                        |
| 12 | 163                  | 164               | 1                        |
| 13 | 164                  | 164               | 0                        |
| 14 | 164                  | 162               | 2                        |
| 15 | 164                  | 163               | 1                        |
| 16 | 162                  | 162               | 0                        |
| 17 | 163                  | 163               | 0                        |
| 18 | 62                   | 163               | 101                      |
| 19 | 163                  | 163               | 0                        |
| 20 | 163                  | 162               | 1                        |
| 21 | 163                  | 162               | 1                        |
| 22 | 162                  | 164               | 2                        |
| 23 | 162                  | 162               | 0                        |
| 24 | 162                  | 161               | 1                        |
| 25 | 164                  | 163               | 1                        |
| 26 | 162                  | 162               | 0                        |
| 27 | 161                  | 163               | 2                        |
| 28 | 163                  | 162               | 1                        |
| 29 | 162                  | 162               | 0                        |
| 30 | 163                  | 162               | 1                        |
| 31 | 162                  | 163               | 1                        |
| 32 | 162                  | 161               | 1                        |
| 33 | 162                  | 163               | 1                        |
| 34 | 163                  | 162               | 1                        |
| 35 | 163                  | 162               | 1                        |
| 36 | 161                  | 162               | 1                        |
| 37 | 162                  | 162               | 0                        |
| 38 | 162                  | 162               | 0                        |
| 39 | 163                  | 162               | 1                        |
| 40 | 163                  | 162               | 1                        |

|                 |     |     |      |
|-----------------|-----|-----|------|
| 41              | 163 | 162 | 1    |
| 42              | 163 | 162 | 1    |
| 43              | 163 | 162 | 1    |
| 44              | 163 | 162 | 1    |
| 45              | 163 | 162 | 1    |
| 46              | 163 | 162 | 1    |
| 47              | 163 | 162 | 1    |
| 48              | 163 | 162 | 1    |
| 49              | 163 | 162 | 1    |
| 50              | 163 | 162 | 1    |
| Total Error     |     |     | 84   |
| Rata-rata Error |     |     | 1.68 |



Gambar 4.5. Respon Kontrol *Facetracker* pada  $K_p=0.8$ ,  $K_d=0.05$  dan  $K_i=0.001$

Tabel 4.3. Nilai  $K_p=0.8$ ,  $K_d=0.05$ ,  $K_i=0.001$

| No | Set Point (Posisi x) | Actual (Posisi x) | Nilai Error Steady State |
|----|----------------------|-------------------|--------------------------|
| 1  | 170                  | 161               | 9                        |
| 2  | 168                  | 167               | 1                        |
| 3  | 168                  | 162               | 6                        |
| 4  | 167                  | 168               | 1                        |
| 5  | 167                  | 167               | 0                        |
| 6  | 168                  | 168               | 0                        |
| 7  | 167                  | 168               | 1                        |
| 8  | 166                  | 167               | 1                        |
| 9  | 166                  | 166               | 0                        |
| 10 | 168                  | 168               | 0                        |
| 11 | 167                  | 167               | 0                        |
| 12 | 167                  | 167               | 0                        |
| 13 | 167                  | 167               | 0                        |
| 14 | 167                  | 167               | 0                        |
| 15 | 167                  | 167               | 0                        |
| 16 | 167                  | 167               | 0                        |
| 17 | 168                  | 168               | 0                        |
| 18 | 169                  | 169               | 0                        |
| 19 | 168                  | 169               | 1                        |
| 20 | 168                  | 168               | 0                        |
| 21 | 168                  | 168               | 0                        |
| 22 | 168                  | 169               | 1                        |
| 23 | 169                  | 169               | 0                        |
| 24 | 168                  | 168               | 0                        |
| 25 | 168                  | 169               | 1                        |
| 26 | 170                  | 170               | 0                        |
| 27 | 170                  | 170               | 0                        |



|                 |     |     |      |
|-----------------|-----|-----|------|
| 28              | 170 | 170 | 0    |
| 29              | 170 | 170 | 0    |
| 30              | 170 | 170 | 0    |
| 31              | 169 | 169 | 0    |
| 32              | 172 | 171 | 1    |
| 33              | 172 | 172 | 0    |
| 34              | 172 | 172 | 0    |
| 35              | 172 | 172 | 0    |
| 36              | 173 | 173 | 0    |
| 37              | 175 | 175 | 0    |
| 38              | 176 | 176 | 0    |
| 39              | 176 | 176 | 0    |
| 40              | 176 | 176 | 0    |
| 41              | 176 | 176 | 0    |
| 42              | 176 | 176 | 0    |
| 43              | 176 | 176 | 0    |
| 44              | 176 | 176 | 0    |
| 45              | 176 | 176 | 0    |
| 46              | 176 | 176 | 0    |
| 47              | 176 | 176 | 0    |
| 48              | 176 | 176 | 0    |
| 49              | 176 | 176 | 0    |
| 50              | 176 | 176 | 0    |
| Total Error     |     |     | 11   |
| Rata-rata Error |     |     | 0.22 |

Setelah dilakukan pengambilan data *respon control* terhadap *set point* nilai masing-masing Kd & Ki yang diubah-ubah, Penentuan nilai Kp, Ki dan Kd pada kendali PID akan sangat menentukan *respon control* terhadap *set point*, dimana pada penelitian ini yang paling optimal adalah pada saat Kp=0.8, Ki=0.001 dan Kd=0.05 dengan error rata-rata 2%, Penambahan kendali nilai pada PID untuk motor DC ini terbukti dapat memberikan *respon* untuk selalu mendekati nilai *set point*.

### 5.1 Kesimpulan.

Berdasarkan hasil analisa yang telah dikerjakan dapat disimpulkan, sebagai berikut :

1. Penentuan nilai Kp, Ki dan Kd pada kendali PID akan sangat menentukan *respon control* terhadap *set point*, dimana pada penelitian ini nilai optimal adalah pada saat Kp=0.8, Ki=0.001 dan Kd=0.05 dengan *error* rata-rata 2%.
2. Sistem *Haar-like feature* mampu mengenali wajah.
3. Penangkapan obyek dengan menggunakan model *Haar-like feature* bisa dilakukan.
4. Dari hasil eksperimental, dapat disimpulkan bahwa PID controller dapat mengontrol pergerakan *webcam* sesuai dengan pergerakan wajah.
5. Dapat mengimplementasikan *Face Tracker* dengan sistem *Haar-like feature* dan PID.

### 5.2 Saran.

Salah satu kelemahan dari penggunaan kontroler PID dan *Haar-like Feature* yang dikembangkan pada

penelitian ini adalah di dalam *setting* parameter Kp, Ki, dan Kd terjadi perubahan kondisi sistem (perubahan kecepatan), maka konstanta (*gain*) harus disesuaikan kembali.

Upaya untuk mengatasi kelemahan ini, maka sistem kontroler ini perlu dikembangkan lagi dengan menggunakan metode kontroler lain yang lebih handal. Penggabungan konsep logika *fuzzy* dan PID, dimana dengan metode pembelajaran aturan *fuzzy* dan PID, pengaturan kecepatan motor dapat mengatasi kelemahan kontroler PID ini.

- [1] Sutoyo. T, Mulyanto Edy, Suhartono. Vincent, Dwi Nurhayati Oky, Wijanarto, Andi Yogyakarta dan UDINUS Semarang “*Teori Pengolahan Citra Digital*”, (2009).
- [2] Bradski Gary , Kaehler Adrian,”*Learning OpenCV* “,O’Reilly Media, 1005 Gravenstein Highway North, Sebastopol, CA 95472, September (2008).
- [3] Viola, Paul and Michael Jones. “*Rapid object detection using boosted cascade of simple features*”. In: Proceedings IEEE Conf. on Computer Vision and Pattern Recognition, (2001).
- [4] Lienhart, Rainer and Jochen Maydt. “*An extended set of haar-like features for rapid object detection*”. In: IEEE ICIP, Vol.1, pp 900-903 (2002).
- [5] RD. Kusumanto, Alan Novi Tompunu, “*Pengolahan Citra Digital untuk Mendeteksi Obyek Menggunakan Pengolahan Warna Model Normalisasi RGB Computer Vision*, Binus Univesity, (2002).
- [6] Sutoyo. T, Mulyanto. Edy, Suhartono. Vincent, Dwi Nurhayati Oky, Wijanarto, ROBOTIKA: Teori PID *Cloos Loop*, Andi Yogyakarta dan UDINUS Semarang (2009).
- [7] Ang. K. H, Chong. G, Li Yun, *PID Control System Analysis, Design, and Technology*, IEEE Transactions On Control Systems Technology, Vol. 13, No. 4, July (2005), 559.
- [8] Teori M, Friday Magnetics – Effects electric-current, “ *Sistem Kendali Pada Pengaturan Kecepatan Motor DC* ”, Makara, Teknologi, Vol. 8, No. 1, April 2004: 25-34. Manafeddin Namazov, Onur Basturk (2010).
- [9] <http://www.integralprojection.com/colorpixel.html>
- [10] Skema Hubungan visi *computer* dengan bidang lain Anvil K I , (2000)
- [11] [www.tutorvista.com/content/science/science-ii/magnetic-effects-electric](http://www.tutorvista.com/content/science/science-ii/magnetic-effects-electric)