

SISTEM PENGATURAN OTOMATIS BUKA TUTUP PINTU BERDASARKAN DETEKSI WAJAH MENGGUNAKAN TEKNIK INTERPOLASI

Budi Setiadi
Staf Pengajar Jurusan Teknik Elektro Politeknik Negeri Bandung
E-mail: budi_kendali2003@yahoo.com

Abstrak

Sistem pengaturan otomatis buka tutup pintu yang ada saat ini, kebanyakan masih menggunakan perangkat masukan (sensor) infra-red, sonar, dan sensor sejenis yang lainnya. Penggunaan sensor sejenis ini memiliki keterbatasan dalam hal penentuan jenis objek. Dalam penelitian ini, penggunaan sensor tersebut digantikan menggunakan kamera (WebCam). Sehingga pintu akan terbuka pada saat kamera mendeteksi objek wajah manusia. Selanjutnya objek wajah manusia tersebut akan diolah oleh mikrokontroler (miniPC Raspberry Pi) dan dikeluarkan dalam bentuk posisi. Dimana posisi tersebut menunjukkan letak dari objek wajah manusia yang berada diarea gambar dengan ukuran piksel 640x480. Data posisi tersebut selanjutnya dilakukan proses interpolasi terhadap lebar bukaan pintu (disimulasikan menggunakan motor servo standar). Hasilnya, sisi pintu kanan maupun kiri dapat terbuka ketika kamera mendeteksi objek wajah manusia.

Kata kunci : Kamera, Piksel , Raspberry Pi, Interpolasi, Servo, PWM

1. Pendahuluan

Teknologi *embedded system* yang sangat cepat berkembang, khususnya mikrokontroler atau Mini PC yang semula tidak memungkinkan ditanam sistem operasi sampai dengan dapat ditanam sistem operasi Linux (Android, Raspbian, Achhilux). Berimbang pada berkembangnya penggunaan teknologi tersebut pada bidang otomasi gedung (pintu), otomotif (econo), dan industri (PLC). Salah satu bentuk nyata yang dapat dilihat pada sistem gedung adalah penggunaan pengaturan otomatis buka tutup pintu.

Pengaturan otomatis buka tutup pintu yang ada saat ini masih menggunakan masukan (sensor) infra-red, sonar atau sensor sejenisnya. Penggunaan sensor tersebut masih memiliki keterbatasan dalam hal pendeteksian jenis objek. Sehingga ketika ada objek padat, cair, atau gumpalan angin (debu) sensor tersebut dapat mendeteksi dan pintu akan terbuka.

Pada penelitian ini, penggunaan sensor tersebut digantikan menggunakan kamera (WebCam) yang biasa digunakan pada komputer (PC). Sehingga

pintu akan terbuka pada saat kamera mendeteksi objek wajah manusia. Selanjutnya objek wajah manusia tersebut akan diolah oleh mikrokontroler (miniPC Raspberry Pi) dan dikeluarkan dalam bentuk posisi. Dimana posisi tersebut menunjukkan letak dari objek wajah manusia yang berada diarea gambar dengan ukuran piksel 640x480. Data posisi tersebut selanjutnya dilakukan proses **interpolasi linier** terhadap lebar bukaan pintu. Sehingga kedua sisi pintu, baik pintu kanan maupun kiri dapat terbuka ketika kamera mendeteksi objek wajah manusia.

Hasil dari penggunaan sensor kamera ini, pintu dapat terbuka dengan lebar bukaan bergantung dari posisi objek wajah manusia. Serta pintu tidak dapat terbuka pada saat objek mendeteksi objek selain wajah manusia.

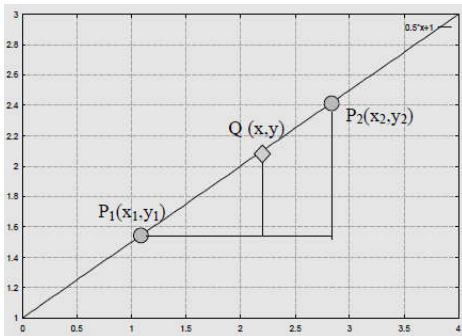
2. Landasan Teori

Pada sub-bab ini akan dijelaskan mengenai definisi, pengertian, istilah dan teori yang mendukung penelitian ini.

2.1 Interpolasi Linier

Interpolasi adalah suatu teknik yang digunakan untuk mencari titik nilai yang terletak diantara n buah titik dengan menggunakan pendekatan fungsi tertentu. Teknik interpolasi secara umum terbagi menjadi interpolasi linier, kuadratik, polinomial, dan lagrange.

Interpolasi linier digunakan untuk menentukan titik-titik diantara 2 buah titik dengan menggunakan garis lurus. Seperti terlihat pada Gambar 2.1 dibawah.



Gambar 2.1 Kurva Interpolasi Linier

Dari Gambar 2.1 diatas dapat dicari nilai titik Q yang terletak diantara titik P1(x1,y1) dan P2(x2,y2) dengan menggunakan persamaan garis lurus, sebagai berikut

$$\frac{y-y_1}{y_2-y_1} = \frac{x-x_1}{x_2-x_1} \dots\dots\dots \text{pers(1)}$$

Sehingga dari persamaan (1) diatas dapat diperoleh persamaan baru, sebagai berikut.

$$y = \frac{y_2-y_1}{x_2-x_1} * (x - x_1) + y_1 \dots\dots\dots \text{pers(2)}$$

2.2 Kamera

Secara umum kamera merupakan perangkat elektronik yang memiliki fungsi untuk mengabadikan suatu objek menjadi sebuah file gambar. Adapun Teknik pengambilan data file dari hasil proses mengabadikan suatu objek berupa serial atau paralel. Dan data file berupa piksel. Berikut bentuk fisik kamera USB Webcam seperti pada Gambar 2.2 dibawah.



Gambar 2.2 USB WebCam

2.3 Piksel

Pixel adalah blok matrik untuk gambar digital. Sebuah piksel yang kita sebut warna atau nilai-nilai cahaya yang menempati tempat tertentu dalam gambar. Gambar matrik, dengan masing-masing persegi di grid mengandung satu warna. Grid ini disebut bitmap. Sebuah gambar dengan resolusi 1024 x 768 adalah grid dengan 1.024 kolom dan baris 768, yang kemudian mengandung 1.024 * 768 = 786.432 piksel.

Banyaknya piksel dalam gambar tidak ada kaitan dengan dimensi fisik dari gambar. Karena piksel bukanlah unit ukuran. Jadi 1 piksel tidak sama dengan 1 milimeter, 1 mikrometer, atau 1 nanometer. Sebaliknya, "besar" piksel akan tergantung pada piks/inci (ppi) pengaturan untuk gambar itu.

Setiap pixel diwakili oleh nomor atau serangkaian angka dan kisaran angka ini disebut kedalaman warna atau bit kedalaman. Dengan kata lain, kedalaman warna menunjukkan jumlah maksimum warna potensial yang dapat digunakan dalam gambar.

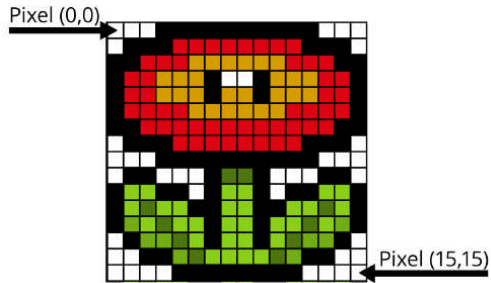
Warna 8-bit kedalaman menggunakan nomor 0-255 (8-bit) untuk setiap kanal warna dalam sebuah piksel. Ini berarti 1024 x 768 gambar dengan satu saluran (hitam dan putih) kedalaman warna 8-bit akan membuat 768 kB gambar.

Setiap piksel dapat direpresentasikan menjadi grayscale dan warna. Piksel grayscale hanya memiliki nilai tunggal yang mewakili nilai, nol menjadi hitam

dan 255 menjadi putih. sedangkan Kebanyakan piksel warna memiliki tiga nilai yang mewakili warna merah, hijau, dan biru(RGB). Setiap warna diwakili oleh satu Byte, atau nilai dari 0 sampai 255, yang menunjukkan jumlah warna yang diberikan. Dalam bahasa pemrograman penulisan RGB digabung menjadi format (merah, hijau, biru). Misalnya, (125, 0, 125) berarti bahwa piksel memiliki beberapa merah, ada hijau, dan beberapa biru, sehingga nilai tersebut mewakili warna ungu.

Berikut contoh warna yang diwakili oleh RGB.

- Merah: (255, 0, 0)
- Hijau: (0, 255, 0)
- Biru: (0, 0, 255)
- Kuning: (255, 255, 0)
- Brown: (165, 42, 42)
- Oranye: (255, 165, 0)
- Hitam: (0, 0, 0)
- Putih: (255, 255, 255)



Gambar 2.3 Pikel dan Koordinat

2.4 Gambar

Gambar merupakan gabungan antara bitmaps dan piksel. Gambar tersusun dalam bentuk array dua dimensi. Dua dimensi array seperti sepotong kertas grafik, yang terdiri atas sejumlah set unit vertikal dan set unit horizontal. Setiap persegi diindeks oleh satu set dua angka. Angka pertama mewakili baris horizontal untuk persegi itu, dan angka kedua adalah kolom vertikal. Pendekatan ini disebut "Cartesian koordinat", (x,y).

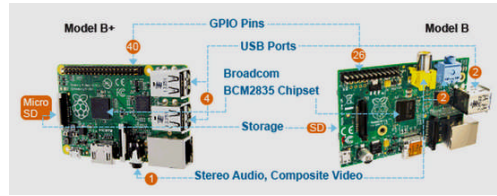


Gambar 2.4 Pikel 120, 150

Informasi untuk piksel individu dapat diekstraksi dari gambar. Berikut hasil mengekstrak piksel di (120, 150) dari Gambar 2.4 diatas. RGB untuk piksel di (120, 150) akan menghasilkan warna dalam format (242,0, 222,0, 204,0), titik merah.

2.5 Raspberry Pi

Raspberry pertama kali dikembangkan di Inggris pada tahun 2012. Raspberry merupakan modul perangkat keras yang tersusun dari komponen mikrokontroler pabrikan samsung, dan merupakan pengembangan dari ARM-cortex4. Arsitektur raspberry dibuat menjadi sistem mikroprosesor (mini PC), setara cortex 8. Dalam mini PC ini ditambahkan memori eksternal DCard/MiniSDCard yang berfungsi sebagai memori ROM pengganti HardDisk. Memori eksternal ini merupakan tempat untuk ditanamkannya sistem operasi dan program aplikasi. Berikut bentuk modul raspberry seperti pada Gambar 2.5 dibawah.



Gambar 2.5 Modul Raspberry Pi

Prinsip kerja modul raspberry mirip dengan komputer biasa. Bahkan memiliki fitur tambahan berupa GPIO (*General Purpose Input Output*). GPIO ini berfungsi untuk proses antarmuka dengan perangkat digital lainnya, dalam bentuk format data serial, paralel, dan bit. Berikut susunan GPIO modul raspberry seperti pada Gambar 2.6 dibawah.

Rev 1 P1 GPIO Header		Rev 2 P1 GPIO Header		B+ J8 GPIO Header	
Pin No.	3.3V 5V	Pin No.	3.3V 5V	Pin No.	3.3V 5V
GPIO0	3 4	GPIO2	3 4	GPIO2	3 4
GPIO1	5 6	GPIO3	5 6	GPIO3	5 6
GPIO4	7 8	GPIO4	7 8	GPIO4	7 8
GND	9 10	GPIO15	9 10	GPIO15	9 10
GPIO17	11 12	GPIO18	11 12	GPIO18	11 12
GPIO21	13 14	GND	13 14	GPIO27	13 14
GPIO22	15 16	GPIO23	15 16	GPIO23	15 16
3.3V	17 18	GPIO24	17 18	GPIO24	17 18
GPIO10	19 20	GND	19 20	GPIO10	19 20
GPIO8	21 22	GPIO25	21 22	GPIO25	21 22
GPIO11	23 24	GPIO8	23 24	GPIO8	23 24
GND	25 26	GPIO7	25 26	GPIO7	25 26
				DNC	27 28
				DNC	29 30
				GPIO6	31 32
				GPIO13	33 34
				GPIO19	35 36
				GPIO26	37 38
				GND	39 40
				GPIO21	

Gambar 2.6 Konfigurasi GPIO

2.5.1 Python

Python masuk kedalam kategori bahasa pemrograman tingkat tinggi. Bahasa pemrograman python pertama kali diperkenalkan pada tahun 1985 di Belanda. Adapun struktur bahasa pemrograman python mirip dengan struktur bahasa pemrograman tingkat tinggi lainnya, seperti C, Java, Oracle dan lainnya. Perbedaan mendasar didalam bahasa pemrograman python adalah dalam hal penulisan variabel. Penulisan variabel tidak perlu dideklarasikan. Serta penulisan variabel tidak boleh sama dengan nama register, library, dan intruksi. Berikut intruksi dan fungsi yang dikenali oleh python, seperti tabel 2.1 dibawah.

Tabel 2.1 Intrukri dan Fungsi di Python

Conditionals	Loops	Built-in Functions	Classes, Modules, Functions	Error Handling
if	for	print	class	try
else	in	pass	def	except
elif	while	del	global	finally
not	break		lambda	raise
or	as		nonlocal	assert
and	continue		yield	with
is			import	
True			return	
False			from	
None				

Intruksi dibawah merupakan contoh penulisan bahasa pemrograman python. From merupakan intruksi untuk memanggil fungsi utama, serta import berfungsi untuk mengambil bagian fungsi dari fungsi utama. While merupakan intruksi percabangan, selama bernilai true maka program utama akan berulang. Now merupakan variabel, menerima data string berupa waktu dari RTC raspberry. Print merupakan intruksi untuk menampilkan data di variabel now ke layar. Sleep (1) berfungsi untuk menahan kondisi tampilan data waktu dilayar.

```
from datetime import datetime
from time import sleep
while True:
    now = str(datetime.now())
    print(now)
    sleep(1)
```

2.5.2 SimpleCV

SimpleCV dan OpenCV merupakan program *opensource* yang biasa digunakan dibidang *computer vision*. Perbedaan antara SimpleCV dan OpenCV terdapat pada fitur librari yang tersedia beserta ukuran file-ya. SimpleCV berisi fitur librari pengenalan wajah, hidung, bagian tengah badan, dan bagian bawah badan. Penggunaan SimpleCV di modul Raspberry Pi perlu di instal terlebih dahulu. Adapun tatacara peginstalann-nya sebagai berikut.

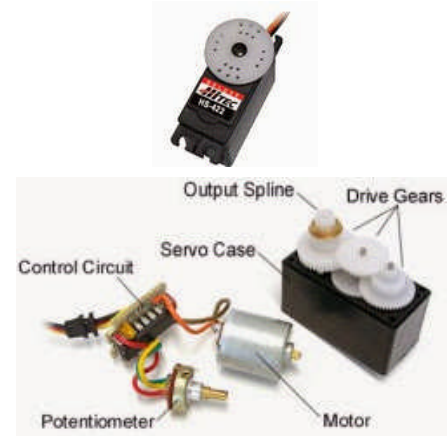
- Buka editor text di Raspberry Pi (LX Terminal).
- Ketikan perintah:
pi@raspberrypi ~ \$ **sudo apt-get install python-opencv python-scipy pythonnumpy python-pip**
- Tekan tombol enter, tunggu proses download dan install selesai.
- Ketikan perintah:
pi@raspberrypi ~ \$ **sudo pip install https://github.com/ingenuitas/SimpleCV/zipball/master**
- Tekan tombol enter, tunggu proses download dan install selesai.
- Ketikan perintah:
pi@raspberrypi ~ \$ **python**
- Selanjutnya akan tampil
Python 2.7.3rc2 (default, May 6 2012, 20:02:25)
[GCC 4.6.3] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> import SimpleCV
>>>

- Proses download dan install SimpleCV telah selesai.

2.6 Motor Servo

Motor servo adalah sebuah perangkat atau aktuator putar (motor) yang dirancang dengan sistem kontrol umpan balik loop tertutup (servo), sehingga dapat di set-up atau di atur untuk menentukan dan memastikan posisi sudut dari poros output motor. Motor servo merupakan perangkat yang terdiri dari motor DC, serangkaian gear, rangkaian kontrol dan potensiometer. Serangkaian gear yang melekat pada poros motor DC akan memperlambat putaran poros dan meningkatkan torsi motor servo, sedangkan potensiometer dengan perubahan resistansinya saat motor berputar berfungsi sebagai penentu batas posisi putaran poros motor servo.

Penggunaan sistem kontrol loop tertutup pada motor servo berguna untuk mengontrol gerakan dan posisi akhir dari poros motor servo. Berikut bentuk fisik dari motor servo seperti terlihat pada Gambar 2.7 di bawah.



Gambar 2.7 Bentuk Fisik Motor Servo

Dilihat dari jenis sumbernya motor servo terdiri atas motor servo AC dan DC. Motor servo AC lebih dapat menangani arus yang tinggi atau beban berat, sehingga sering diaplikasikan pada mesin-mesin industri. Sedangkan motor servo DC biasanya lebih cocok untuk digunakan pada aplikasi-aplikasi yang lebih kecil.

Dilihat dari jenis rotasinya, motor servo terdiri atas motor servo standar dan kontinu.

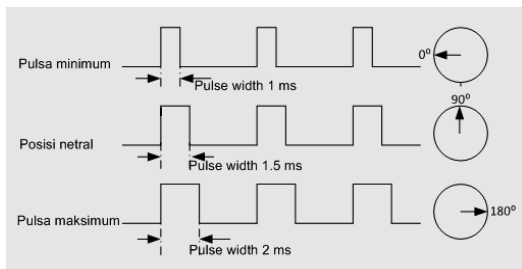
- Motor servo standar (servo rotation 180°) adalah jenis yang paling umum dari motor servo, dimana putaran poros outputnya terbatas hanya dan -90° kearah kiri dan +90° kearah kanan. Dengan kata

lain total putarannya hanya setengah lingkaran atau 0° s/d 180° .

- Motor servo rotation continuous merupakan jenis motor servo yang sebenarnya sama dengan jenis servo standard, hanya saja perputaran porosnya tanpa batasan atau dengan kata lain dapat berputar terus, baik ke arah kanan maupun kiri.

Prinsip kerja motor servo motor servo dikendalikan dengan memberikan sinyal modulasi lebar pulsa (Pulse Wide Modulation / PWM) melalui kabel kontrol. Lebar pulsa sinyal kontrol yang diberikan akan menentukan posisi sudut putaran dari poros (motor servo standar) dan menentukan arah serta kecepatan (motor servo kontinu).

Contoh PWM motor servo standar, lebar pulsa dengan waktu 1 ms (mili detik) akan memutar poros motor servo ke posisi sudut 0° . Bila pulsa lebih 1 ms dan tepat 1,5 ms maka akan berputar ke arah posisi 90° , sedangkan bila pulsa yang diberikan lebih lama dari 1,5 ms dan tepat 2 ms maka poros motor servo akan berputar ke arah posisi 180° . Berikut bentuk pulsa PWM untuk motor servo seperti terlihat pada Gambar 2.8 dibawah.



Gambar 2.8 Sudut PWM Motor Servo

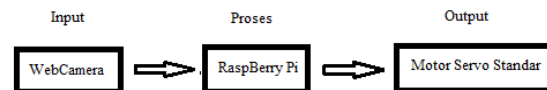
Contoh PWM motor servo kontinu, lebar pulsa dengan waktu 1,5 ms (mili detik) akan membuat poros motor servo berhenti (stop). Bila pulsa kurang 1,5 ms motor servo akan berputar kekiri, semakin kecil nilai pulsa (mendekati 1 ms) maka semakin cepat putaran poros servo. Sebaliknya, apabila pulsa lebih 1,5 ms motor servo akan berputar kekanan, semakin besar nilai pulsa (mendekati 2 ms) maka semakin cepat putaran poros servo.

3 Metodologi/Perancangan

Metode penelitian yang pertama sekali dilakukan adalah studi pustaka. Penelitian ini sangat menunjang kesuksesan sistem yang akan dirancang agar dapat disusun sesuai dengan teori yang ada. Penelitian ini bisa berasal dari buku maupun jurnal ilmiah dari sebuah proyek yang

mirip dengan yang akan penulis rancang. Dengan penelitian ini, ilmu untuk merancang sistem dengan baik, serta kendala yang dapat dihindari di depan semuanya dapat dikumpulkan dan dianalisis. Setelah membaca referensi yang dibutuhkan, dilakukan perancangan alat.

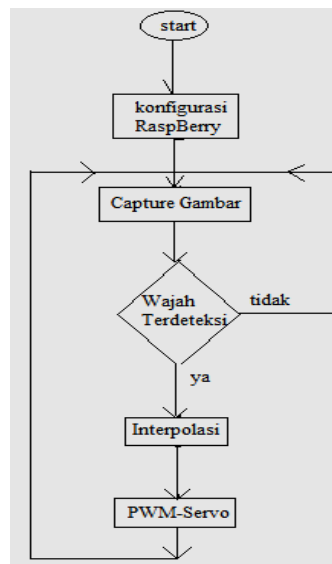
Pada penelitian ini, ketiga elemen alat diintegrasikan menjadi sebuah sistem, yang terdiri dari input, proses, dan output. Sebagai inputnya, menggunakan kamera USB WebCam 1,5 Mega piksel. Bagian proses pengolahan data hasil file gambar dari kamera diolah oleh Raspberry Pi menjadi, untuk selanjutnya dikeluarkan dalam bentuk pulsa PWM. Dan outputnya berupa motor servo standar (prototipe Pintu), akan terbuka dan menutup berdasarkan pulsa PWM yang diterima dari modul Raspberry Pi. Berikut gambar blok diagram perancangan sistem kontrol kamera seperti terlihat pada Gambar 2.9 dibawah.



Gambar 2.9 Diagram Blok Sistem Kontrol Kamera

3.1 Perancangan Perangkat Lunak

Perancangan perangkat lunak terdiri atas bagian proses pengambilan gambar, pendeteksian wajah, dan posisi piksel wajah. Selanjutnya data posisi piksel digunakan sebagai data masukan/perbandingan untuk proses interpolasi. Keluaran proses interpolasi diubah menjadi bentuk PWM dan disalurkan ke motor servo standar.



Gambar 2.10 Diagram Alir Sistem

3.1.1 Psecode Pengambilan Gambar

Berikut merupakan potongan program untuk proses pengambilan file data gambar dari kamera WebCam.

```
from SimpleCV import Camera, Display
import time
myCamera = Camera(prop_set={'width': 640,
'height': 480})
myDisplay = Display(resolution=(640,480))
while True:
    myCamera.getImage()
    myCamera.save(myDisplay)
    time.sleep(.1)
```

3.1.2 Psecode Pendeteksian Wajah dan Posisi Piksel

Berikut merupakan potongan program untuk proses pendeteksian wajah dan posisi piksel wajah, dengan resolusi 640x480

```
from SimpleCV import Camera, Display
import time
myCamera = Camera(prop_set={'width':640,
'height': 480})
myDisplay = Display(resolution=(640, 480))
while True:
    frame = myCamera.getImage()
    faces = frame.findHaarFeatures('face')
    if faces:
        for face in faces:
            print "Posisi Wajah: "
                +
                str(face.coordinates())
            posisi=raw_input (
str(face.coordinates()))
    else:
        print "Wajah Tidak Terdeteksi."
        frame.save(myDisplay)
        time.sleep(.1)
```

3.1.3 Psecode Interpolasi Posisi Wajah dan sudut Servo

Berikut merupakan potongan program untuk proses interpolasi antara posisi wajah (piksel) dengan sudut servo (pintu).

```
from SimpleCV import Camera, Display
import time
x1=0
x2=640
y1=0
y2=180
myCamera = Camera(prop_set={'width':640,
'height': 480})
```

```
myDisplay = Display(resolution=(640, 480))
while True:
    frame = myCamera.getImage()
    faces = frame.findHaarFeatures('face')
    if faces:
        for face in faces:
            print "Posisi Wajah: "
                + str(face.coordinates())
            posisi=raw_input (str(face.coordinates()))
            y=(((y2-y1)/(x2-x1))*(posisi[0]-x1))+y1
    else:
        print "Wajah Tidak Terdeteksi."
        frame.save(myDisplay)
        time.sleep(.1)
```

3.1.4 Psecode Interpolasi sudut Servo dan PWM

Berikut merupakan potongan program interpolasi antara sudut servo dan keluaran PWM (pulsa_pwm 1000 s/d 2000).

```
from SimpleCV import Camera, Display
import time
from RPIO import PWM
servo=PWM.Servo()
servo.set_servo(23,1000)
servo.set_servo(24,1000)
x1=0
x2=640
y1=0
y2=180
pwm1=1000
pwm2=2000
myCamera = Camera(prop_set={'width':640, 'height':
480})
myDisplay = Display(resolution=(640, 480))
while True:
    frame = myCamera.getImage()
    faces = frame.findHaarFeatures('face')
    if faces:
        for face in faces:
            print "Posisi Wajah:"
                + str(face.coordinates())

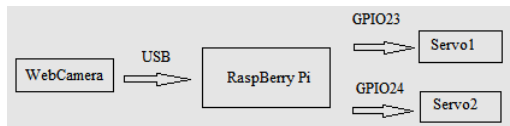
            posisi=raw_input ( str(face.coordinates()))
            pulsa_pwm=(((pwm2-pwm1)/
(y2-y1))*(posisi[0]-y1))
                +pwm2
            pulsa_pwm=pulsa_pwm+100
            pulsa_pwm=float(pulsa_pwm)
            servo.set_servo(23,pulsa_pwm)
            servo.set_servo(24,pulsa_pwm)
```

else:

```
print "Wajah Tidak Terdeteksi"
frame.save(myDisplay)
time.sleep(.1)
```

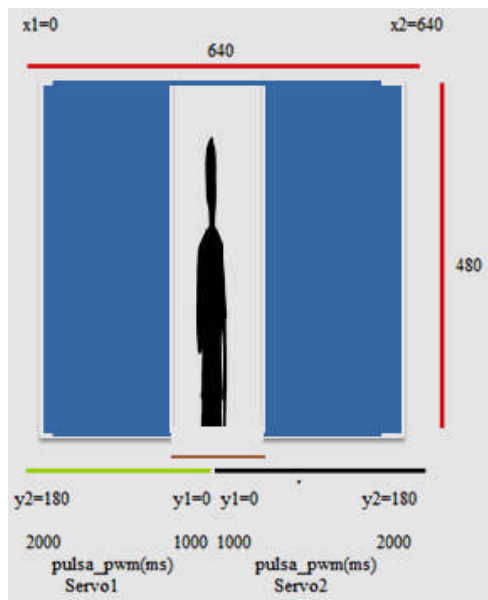
3.2 Perancangan Perangkat Keras

Pada penelitian ini, sebagai masukan sistem digunakan WebCam kamera USB. Sedangkan keluaran prototipe pintu menggunakan 2 buah penggerak, yaitu servo1 untuk pintu sisi kiri dan servo2 untuk sisi kanan. Berikut gambar blok diagram perangkat keras seperti terlihat pada Gambar 3.1 dibawah.



Gambar 3.1 Blok Diagram Perangkat Keras

Servo1 dan servo2 akan bergeser berdasarkan perubahan nilai pulsa_pwm. Pulsa_pwm servo1 dan servo2 bernilai 1000, menunjukkan kedua buah pintu tertutup, pada pembacaan tidak mendeteksi wajah. Pulsa_pwm servo1 dan servo2 bernilai 1500, menunjukkan kedua buah pintu terbuka sebagian, pada pendeteksian wajah di piksel masukan kamera 320. Berikut gambar ilustrasi pintu terbuka seperti Gambar dibawah.



Gambar 3.2 Ilustrasi Sistem

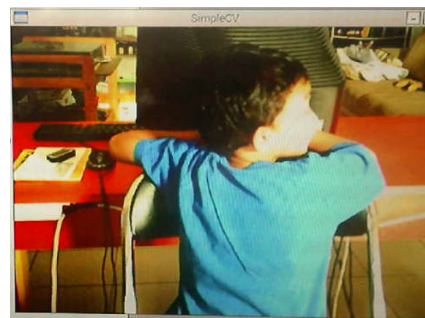
Teknik interpolasi sangat berperan dalam penentuan keputusan berapa besar sudut bukaan motor servo1 dan servo2 terbuka.

4 Pengujian Sistem

Pengujian ini dilakukan untuk mengetahui seberapa besar tingkat keberhasilan pembacaan posisi wajah terhadap keluaran PWM untuk pergerakan posisi servo (pintu). Pada tahapan pengujian ini dilakukan 2 jenis pengujian, yaitu pengujian pada saat wajah tidak terdeteksi, dan wajah terdeteksi.

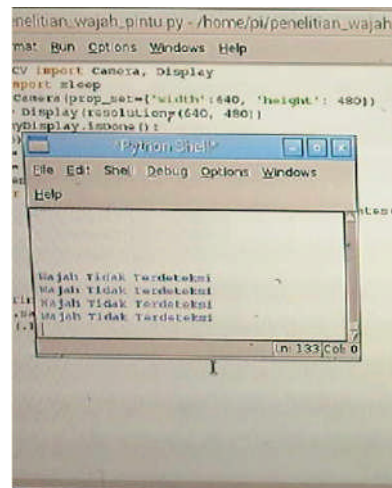
4.1 Wajah Tidak Terdeteksi

Pada tahapan pengujian ini, dilakukan pengujian pengambilan gambar menyamping, seperti terlihat pada Gambar 4.1 dibawah.



Gambar 4.1 Wajah Tidak Terdeteksi

Sehingga algoritma/pseudocode pendeteksian wajah yang tertanam di Raspberry tidak mengenali wajah. Selanjutnya Raspberry akan mengirimkan data ke servo1 dan servo2 berupa PWM 1000 (menandakan pintu tertutup). Adapun data hasil pengolahan Raspberry seperti terlihat pada Gambar 4.2 dibawah.



Gambar 4.2 Hasil Pengolahan DataWajah Tidak Terdeteksi

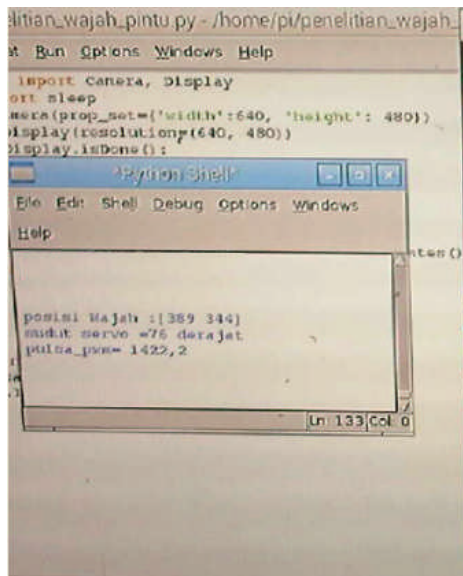
4.2 Wajah Terdeteksi

Pada tahapan pengujian ini, dilakukan pengujian pengambilan gambar dari depan, seperti terlihat pada Gambar 4.3 dibawah.



Gambar 4.3 Wajah Terdeteksi

Sehingga algoritma/psecode pendeteksi wajah yang tertanam di Raspberry mengenali wajah. Selanjutnya Raspberry akan menampilkan data posisi wajah (x=389), perhitungan sudut servo (y=76), serta keluaran pulsa_pwm (1422,2). Data tersebut menandakan menandakan pintu terbuka sekitar 76°, dengan rentang maksimum bukaan setiap pintu 180°. Adapun data hasil pengolahan Raspberry seperti terlihat pada Gambar 4.4 dibawah.



Gambar 4.4 Hasil Pengolahan DataWajah Terdeteksi

4.3 Data Pengujian

Data pengujian ini merupakan hasil sampling dari 10 posisi pengujian wajah tidak terdeteksi dan 10

posisi Pengujian wajah terdeteksi. Berikut tabel data hasil pengujian seperti terlihat pada tabel 4.1, 4.2 dibawah.

Tabel 4.1 Pengujian Wajah Tidak Terdeteksi

Ke	Posisi Piksel (x)	Sudut Servo ° (y)	Pulsa_pwm (ms)	Pengukuran Langsung Sudut Servo °
1	0	0	0	0
2	0	0	0	0
3	0	0	0	0
4	0	0	0	0
5	0	0	0	0
6	0	0	0	0
7	0	0	0	0
8	0	0	0	0
9	0	0	0	0
10	0	0	0	0

Tabel 4.2 Pengujian Wajah Terdeteksi

Ke	Posisi Piksel (x)	Sudut Servo ° (y)	Pulsa_pwm (ms)	Pengukuran Langsung Sudut Servo °
1	120	33,75	1187,5	33,66
2	156	43,87	1243,72	43,2
3	193	54,28	1301,55	54
4	240	67,5	1375	66,6
5	300	84,37	1468,72	82,8
6	360	101,25	1562,5	100,8
7	450	126,56	1703,11	126
8	500	140,62	1781,22	140,4
9	535	150,46	1835,88	149,4
10	600	168,75	1937,5	167,4

Perbandingan antara sudut, pulsa_pwm pengolahan data di Raspberry dengan pengukuran sudut servo secara langsung, menghasilkan keakuratan 86,675 %. Dengan rata-rata sudut pengolahan Raspberry=111,203 dan rata-rata sudut pengukuran langsung=96,386

Kesimpulan

Dari hasil pengujian dan analisa yang telah dilakukan pada perancangan sistem pengontrol kecepatan pada penelitian ini, dapat diambil beberapa kesimpulan sebagai berikut :

1. Pengambilan file data gambar harus mencakup kelengkapan wajah (2 mata dan alis, hidung, mulut dan kening). Pengambilan data gambar yang tidak mewakili dari ciri tersebut dianggap wajah tidak terdeteksi;
2. Pintu dapat terbuka maupun menutup berdasarkan masukan sensor kamera dengan tingkat keakuratan 86,675 % terhadap pengukuran sudut secara langsung.

Daftar Pustaka

- [1].Anton H, 1995. Aljabar Linear Elementer. Drexel University
- [2].Demag K, Oliver A, Oostendorp N, Scott K. 2012. *Practical Computer Vision Use Simple CV*. OREILLY: Beijing, Cambridge, Tokyo
- [3].Hartanto, Thomas Wahyu Dwi dan Y.Wahyu Agung Prasetyo.2003. *Analisis dan Desain Sistem Kontrol dengan Matlab*. Yogyakarta: Andi
- [4].Kusumadewi, sri. 2003. *Artificial Intelligence (Teknik dan Aplikasinya)*. Yogyakarta: Penerbit Graha Ilmu.
- [5].Ogata, katsuhiko. 1997. *Teknik Kontrol Automatik*. Jakarta: Erlangga.