



**TUGAS AKHIR - TE090362**

**PERANCANGAN SISTEM TRACKING CAMERA PENGAWAS  
BERDASARKAN EVALUASI DATA GAMBAR**

**Ari Firman Andriyanto  
NRP 2211030047  
Welly Julian  
NRP 2211030109**

**Dosen Pembimbing  
Ir. Rusdhianto Effendi AK., MT.  
NIP. 19570424 198502 1 001**

**PROGRAM STUDI D3 TEKNIK ELEKTRO  
Fakultas Teknologi Industri  
Institut Teknologi Sepuluh Nopember  
Surabaya 2014**



*FINAL PROJECT - TE090362*

***DESIGN SYSTEM OF TRACKING CAMERA SUPERINTENDENT IN  
DATA PICTURE EVALUATION***

**Ari Firman Andriyanto**  
NRP 2211030047  
**Welly Julian**  
NRP 2211030109

*Supervisor*  
**Ir. Rusdhianto Effendi AK., MT.**  
NIP. 19570424 198502 1 001

*Electrical Engineering D3 Program*  
*Industrial Technology Faculty*  
*Institut Teknologi Sepuluh Nopember*  
*Surabaya 2014*



**PERANCANGAN SISTEM TRACKING CAMERA PENGAWAS  
BERDASARKAN EVALUASI DATA GAMBAR**

**TUGAS AKHIR**

**Diajukan Guna Memenuhi Sebagian Persyaratan  
Untuk Memperoleh Gelar Ahli Madya  
Pada**

**Bidang Studi Komputer Kontrol  
Program Studi D3 Teknik Elektro  
Fakultas Teknologi Industri  
Institut Teknologi Sepuluh Nopember**

**Menyetujui  
Dosen Pembimbing.**

**Ir. Rusdhianto Effendi AK., MT.  
NIP. 19570424 198502 1 001**

**SURABAYA  
JULI, 2014**

PERANCANGAN SISTEM *TRACKING CAMERA* PENGAWAS  
BERDASARKAN EVALUASI DATA GAMBAR

**TUGAS AKHIR**

Diajukan Guna Memenuhi Sebagian Persyaratan  
Memperoleh Gelar Ahli Madya Teknik  
Pada  
Bidang Studi Komputer Kontrol  
Program Studi D3 Teknik Elektro  
Fakultas Teknologi Industri  
Institut Teknologi Sepuluh Nopember

Menyetujui :  
Dosen Pembimbing 1

Ir. Rusdhianto Effendi AK.,MT.  
NIP. 19570424 198502 1 001

**SURABAYA**  
**JULI, 2014**

*Halaman ini sengaja dikosongkan*

# PERANCANGAN SISTEM *TRACKING CAMERA* PENGAWAS BERDASARKAN EVALUASI DATA GAMBAR

**Nama Mahasiswa** : Ari Firman Andriyanto  
**NRP** : 2210 030 047  
**Nama Mahasiswa** : Welly Julian  
**NRP** : 2210 030 109  
**Dosen Pembimbing** : Ir. Rusdhianto Effendi AK.,MT.

## Abstrak

Perkembangan teknologi saat ini sangatlah pesat, baik pada bidang perangkat keras maupun perangkat lunak. Salah satu alasan perancangan sistem *tracking camera* berdasarkan evaluasi data gambar yang dapat mengikuti objek bergerak terutama untuk mendeteksi postur tubuh dan sebagai kamera pengawas yang menentukan bagaimana cara mengatur pergerakan kamera agar tetap memantau objek postur tubuh yang terdeteksi.

Pada perancangannya, sebuah *tracking camera* yang terintegrasi memiliki kemampuan mencari, mendeteksi, dan menangkap adanya gerakan dari suatu objek. Selain itu, dilengkapi juga dengan dua buah motor servo. PC sebagai otak dan rangkaian kontrol yang akan mengendalikan kemana posisi mekanik akan bergerak.

Dalam Tugas Akhir ini diperoleh dari beberapa pengujian sistem hasil dari pengolahan citra berupa deteksi gerakan postur tubuh yang bertugas memerintahkan ke rangkaian kontrol untuk menggerakkan mekanik sesuai dengan kondisi target secara *realtime*. Dan tingkat keakuratan pada saat mendeteksi postur tubuh berpengaruh pada intensitas cahaya. dibuktikan dengan pengujian diluar ruangan pada siang hari memiliki keakuratan 100 %.

**Kata kunci** : *Tracking camera, computer vision, image processing, webcam, Servo motor.*





## **DESIGN SYSTEM OF TRACKING CAMERA SUPERINTENDENT IN DATA PICTURE EVALUATION**

**Name** : Ari Firman Andriyanto  
**NRP** : 2211 030 047  
**Name** : Welly Julian  
**NRP** : 2211 030 109  
**Consultative Lecturer** : Ir. Rusdhianto Effendi AK.,MT.

### **Abstract**

*The development of technology is currently very rapidly, both in the field of hardware and software. One reason for camera tracking system design based on the evaluation of the image data that can follow a moving object especially to detect posture and as a surveillance camera that determines how to set the camera to keep track of the movement of objects detected posture.*

*In its design, an integrated camera tracking have the ability to search for, detect, and capture the motion of an object. In addition, also equipped with two servo motors. PC as a brain and control circuit that will control where the mechanical position will move.*

*In this final obtained from multiple test results of image processing systems such as motion detection posture ordered to charge control circuit to drive the mechanical line with the target condition in realtime. And accuracy when detecting posture effect on the intensity of light. proved by testing outdoors during the day have 100% accuracy.*

**Keyword** : *Tracking camera, computer vision, image processing, webcam, Servo motor.*



## KATA PENGANTAR

Segala puji atas kehadiran Allah SWT yang telah melimpahkan rahmat dan hidayah-Nya kepada kami semua dalam usaha dan kerja kami sehingga kami dapat menyelesaikan Tugas Akhir dengan judul :“Perancangan Sistem *Tracking Camera* Pengawas Berdasarkan Evaluasi Data Gambar”.

Tugas Akhir ini disusun untuk memenuhi persyaratan dalam menyelesaikan studi di Bidang Studi Komputer Kontrol, Program Studi D3 Teknik Elektro, Jurusan Teknik Elektro, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember, Surabaya.

Dalam penyusunan laporan Tugas Akhir ini, kami banyak mendapatkan bantuan dan dukungan dari berbagai pihak. Oleh karena itu, penulis dengan segala kerendahan hati ingin menyampaikan banyak terima kasih kepada:

1. Orang tua dan juga keluarga yang selalu mendukung sehingga buku Tugas Akhir ini dapat terselesaikan
2. Bapak Ir. Rusdhianto Effendi AK.,MT selaku dosen pembimbing Tugas Akhir kami, atas segala kesabaran dan kesediaannya meluangkan waktu untuk membimbing serta memberikan dukungan sehingga Tugas Akhir ini dapat terselesaikan.
3. Bapak Eko Setijadi, ST., MT., Ph.D. selaku Kaprodi D3 Teknik Elektro, Komputer Kontrol FTI – ITS
4. Dan semua pihak yang tidak dapat kami sebutkan satu per satu yang telah memberi dorongan dan bantuan dalam menyelesaikan Tugas Akhir ini baik secara langsung maupun tidak langsung.

Kesalahan juga merupakan bagian tak terpisahkan dari jalan kehidupan manusia. Sehingga hanya pintu maaf yang kami harapkan atas kesalahan-kesalahan kami. Dengan itu kami berharap apa yang ada dalam buku Tugas Akhir ini dapat bermanfaat dan berguna sebagai bagi kita semua dalam memajukan pembangunan bangsa dan negara.

Surabaya, Juli 2014

Penyusun



## DAFTAR ISI

<b>JUDUL</b> .....	i
<b>PENGESAHAN</b> .....	iii
<b>ABSTRAK – ABSTRACT</b> .....	v
<b>KATA PENGANTAR</b> .....	ix
<b>DAFTAR ISI</b> .....	xi
<b>DAFTAR GAMBAR</b> .....	xiii
<b>DAFTAR TABEL</b> .....	xv
<b>BAB I PENDAHULUAN</b>	
1.1. Latar Belakang.....	1
1.2. Perumusan Masalah.....	1
1.3. Batasan Masalah.....	2
1.4. Maksud dan Tujuan.....	2
1.5. Sistematika Laporan.....	2
1.6. Relevansi.....	3
<b>BAB II TEORI PENUNJANG</b>	
2.1. <i>Metode Viola Jones</i> .....	5
2.2. Pengolahan Citra Digital .....	8
2.2.1 Citra Digital .....	8
2.2.2 Resolusi Citra .....	9
2.2.3 Jenis Citra .....	9
2.2.4 Format Warna RGB .....	10
2.3. <i>Open Cv Library</i> .....	11
2.4. Kamera .....	12
2.5. <i>Personal Computer (PC)</i> .....	14
2.6. Motor Servo .....	14
2.6.1 Pengertian Motor Servo.....	14
2.6.2 Prinsip Dasar Cara Kerja .....	15
2.6.3 Prinsip Arah Putaran Motor .....	15
2.7. Mikrokontroler ATmega 16 .....	16
2.8. <i>Power Supply</i> .....	18
2.9. LCD 16 x 2 .....	19
2.10. Metode Komunikasi .....	20
2.10.1 Komunikasi Antara Kamera Dengan PC.....	20
2.10.2 Komunikasi Antara PC Dengan Kontroler .....	21

### **BAB III PERANCANGAN DAN PEMBUATAN ALAT**

3.1. Perancangan Perangkat Keras ( <i>Hardware</i> ) .....	25
3.2. Perancangan Perangkat Lunak ( <i>Software</i> ).....	30
3.3. Perancangan Mekanik.....	34

### **BAB IV PENGUJIAN DAN ANALISA DATA**

4.1. Pengujian Rangkaian <i>Power Supply</i> .....	36
4.2. Pengujian dan Analisa Mikrokontroler .....	37
4.3. Pengujian dan Analisa Modul LCD .....	39
4.4. Pengujian dan Analisa Sistem Komunikasi.....	40
4.5. Pengujian Motor Servo.....	42
4.6. Pengujian dan Analisa Program .....	45
4.6.1 Pengujian dan Analisa pada Program PC .....	45
4.6.2 Pengujian dan Analisa pada Program Kontroler.....	47
4.7. Pengujian dan Analisa Alat Secara Keseluruhan .....	48

### **BAB V PENUTUP**

5.1. Kesimpulan.....	56
5.2. Saran.....	56

<b>DAFTAR PUSTAKA</b> .....	58
-----------------------------	----

<b>LAMPIRAN I LISTING PROGRAM</b> .....	A-1
---	-----

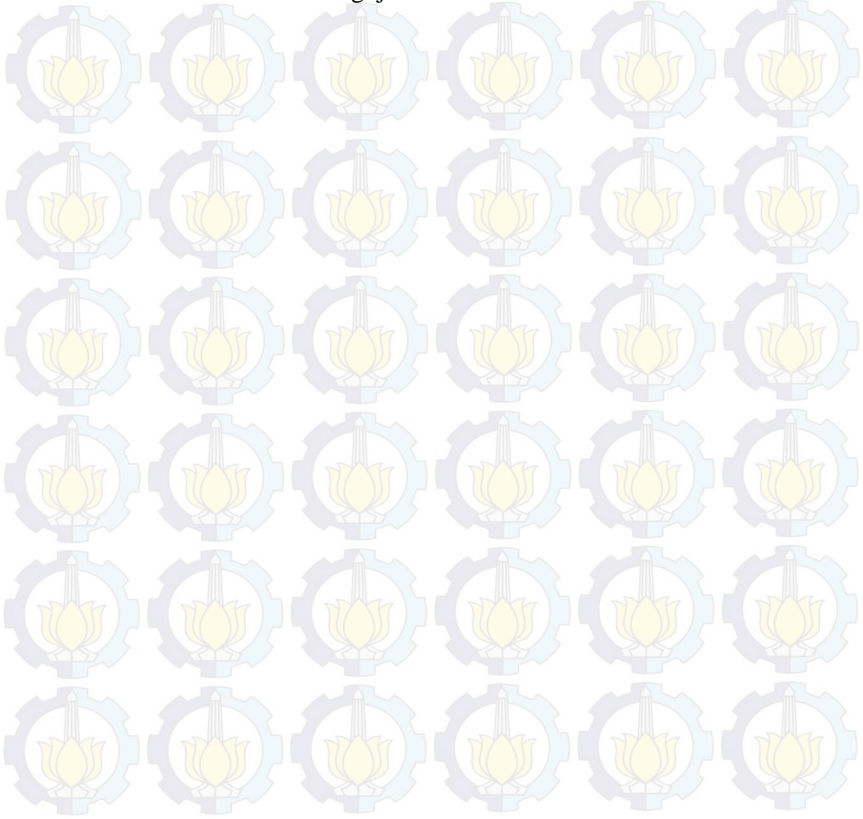
<b>LAMPIRAN II Datasheet</b> .....	B-1
------------------------------------	-----

<b>DAFTAR RIWAYAT HIDUP</b> .....	C-1
-----------------------------------	-----

## DAFTAR GAMBAR

<b>Gambar 2.1</b>	Contoh Fitur yang Digunakan Viola-Jones.....	5
<b>Gambar 2.2</b>	Fitur Haar .....	6
<b>Gambar 2.3</b>	Gambar Integral .....	6
<b>Gambar 2.4</b>	<i>Cascade Classifier</i> .....	7
<b>Gambar 2.5</b>	Citra Gray-Scale .....	9
<b>Gambar 2.6</b>	Citra Biner .....	9
<b>Gambar 2.7</b>	Citra Berwarna .....	10
<b>Gambar 2.8</b>	Warna RGB .....	10
<b>Gambar 2.9</b>	Gabungan Citra RGB .....	10
<b>Gambar 2.10</b>	<i>Webcam A4tech</i> .....	12
<b>Gambar 2.11</b>	Webcam M-Tech 1,3 MP .....	13
<b>Gambar 2.12</b>	Laptop (PC) .....	13
<b>Gambar 2.13</b>	Bagian-Bagian Motor Servo.....	14
<b>Gambar 2.14</b>	Arah Putar Motor Servo.....	15
<b>Gambar 2.15</b>	Pin ATmega 16 .....	16
<b>Gambar 2.16</b>	<i>Power Supply</i> .....	18
<b>Gambar 2.17</b>	LCD 16 x 2 .....	18
<b>Gambar 2.18</b>	Jenis Konektor USB .....	20
<b>Gambar 2.19</b>	Penetapan Kaki Konektor.....	20
<b>Gambar 2.20</b>	Port DB9 .....	21
<b>Gambar 3.1</b>	Blok Diagram Sistem.....	23
<b>Gambar 3.2</b>	Blok Fungsional ATmega 16.....	25
<b>Gambar 3.3</b>	<i>Schematic Rangkaian ATmega 16</i> .....	26
<b>Gambar 3.4</b>	<i>Schematic Rangkaian LCD 16x2</i> .....	26
<b>Gambar 3.5</b>	Motor Servo HS-325HB .....	27
<b>Gambar 3.6</b>	<i>Schematic Rangkaian Power Supply</i> .....	28
<b>Gambar 3.7</b>	<i>Schematic Rangkaian Driver MAX232</i> .....	29
<b>Gambar 3.8</b>	Alur Pembuatan Sistem .....	30
<b>Gambar 3.9</b>	Flowchart Deteksi Bentuk Tubuh.....	31
<b>Gambar 3.10</b>	Flowchat Pelacakan Objek .....	32
<b>Gambar 3.11</b>	Gambar <i>Box Untuk Controller</i> .....	33
<b>Gambar 3.12</b>	Sketsa Mekanik <i>Tracking Camera</i> .....	33
<b>Gambar 4.1</b>	Rangkaian Mikrokontroler .....	36
<b>Gambar 4.2</b>	Sistem Minimum ATmega 16 .....	37
<b>Gambar 4.3</b>	Tampilan LCD .....	39
<b>Gambar 4.4</b>	Rangkaian Serial RS 232 .....	39
<b>Gambar 4.5</b>	Tampilan Serial pada LCD.....	40
<b>Gambar 4.6</b>	Sinyal PWM Dengan Pulsa <i>High 2.1 ms</i> .....	41

<b>Gambar 4.7</b>	Arah Putar Motor Servo Dengan Sudut $180^\circ$ .....	41
<b>Gambar 4.8</b>	Sinyal PWM Dengan Pulsa High 1.5 ms.....	42
<b>Gambar 4.9</b>	Arah Putar Motor Servo Dengan Sudut $90^\circ$ .....	42
<b>Gambar 4.10</b>	Sinyal PWM Dengan Pulsa High 0.9 ms.....	43
<b>Gambar 4.11</b>	Arah Putar Motor Servo Dengan Sudut $0^\circ$ .....	43
<b>Gambar 4.12</b>	Simulasi Tampilan Frame.....	45
<b>Gambar 4.13</b>	Tampilan Frame.....	45
<b>Gambar 4.14</b>	Aplikasi Konsol.....	46
<b>Gambar 4.15</b>	Tampilan pada <i>Hyperterminal</i> .....	47
<b>Gambar 4.16</b>	Alat Keseluruhan .....	48
<b>Gambar 4.17</b>	Sistem Pengujian .....	48





## DAFTAR TABEL

<b>Tabel 2.1</b> Spesifikasi Kamera <i>Webcam A4tech</i> .....	12
<b>Tabel 2.2</b> Fungsi Pin ATmega 16.....	17
<b>Tabel 2.3</b> Fungsi Pin LCD .....	19
<b>Tabel 2.4</b> Jenis pin RS 232 .....	21
<b>Tabel 4.1</b> Data Pengukuran Rangkaian <i>Power Supply</i> .....	35
<b>Tabel 4.2</b> Hasil Pengukuran Tegangan Setiap <i>PORT</i> pada Sistem Minimum ATmega16.....	37
<b>Tabel 4.3</b> Tabel Sinyal PWM Menurut Perhitungan dan Aktual.....	44
<b>Tabel 4.4</b> Tabel Pengujian Siang Hari di Dalam Ruangan.....	49
<b>Tabel 4.5</b> Tabel Pengujian Malam Hari di Dalam Ruangan.....	50
<b>Tabel 4.6</b> Tabel Pengujian Siang Hari di Luar Ruangan.....	51
<b>Tabel 4.7</b> Tabel Pengujian Malam Hari di Luar Ruangan.....	52



*Halaman ini sengaja dikosongkan*

# BAB I

## PENDAHULUAN

### 1.1. Latar Belakang

Perkembangan teknologi saat ini sangatlah pesat baik pada bidang perangkat keras maupun perangkat lunak. Salah satu alasan merancang sistem *tracking camera* yang dapat mengikuti objek bergerak secara otomatis adalah karena dapat diterapkan untuk pengawasan ataupun keamanan guna diaplikasikan dalam lingkungan museum benda-benda langka, perusahaan, perkantoran atau perumahan sebagai pemantauan aktifitas karyawan, mesin-mesin produksi, keamanan dan lain-lain. Dengan adanya aplikasi dan sistem ini, untuk itu diperlukan sistem pengawas yang dapat menyimpan aktifitas – aktifitas yang dilihatnya. Dan diharapkan pihak-pihak yang bertanggung jawab dapat memantau suatu lokasi secara terus menerus hanya dengan menggunakan komputer.

Dalam Tugas Akhir ini kami akan membuat sebuah sistem pengaman yang memanfaatkan ilmu robotika dari bidang *Computer Vision* yang memakai metode *image processing*. Dengan menggabungkan kedua ilmu tersebut, akan dibangun sebuah *tracking camera* yang mampu melakukan *scanning* pada sebuah ruangan. Sistem ini dilengkapi dengan sebuah kamera yang selalu aktif mengolah citra yang didapat sehingga dapat mendeteksi adanya gerakan dari sebuah objek tertentu. Selain itu, sistem ini dapat digunakan sebagai alat yang dapat mengenali dan mencari benda tertentu karena ketelitian tinggi yang dimiliki oleh sistem.

### 1.2. Perumusan Masalah

Adapun permasalahan yang dihadapi dalam penyelesaian Tugas Akhir ini adalah sebagai berikut :

- Bagaimana cara mendeteksi objek postur tubuh?
- Bagaimana cara mengatur pergerakan kamera agar tetap memantau objek postur tubuh yang terdeteksi?
- Bagaimana cara melakukan komunikasi antara kontroler dengan PC?
- Bagaimana cara melakukan *tracking camera* dengan real time dan ketika ada objek postur tubuh maka akan terdeteksi?

### 1.3. Batasan Masalah

Dari perumusan masalah di atas, maka batasan masalah dari Tugas Akhir ini adalah:

- Deteksi objek postur tubuh menggunakan 2 *webcam* satu sebagai pendeteksi objek yang dikombinasikan dengan program pada PC dengan menggunakan *opencv* dan 1 sebagai kamera *tracking*.
- Objek yang dideteksi berupa postur tubuh yang terlihat utuh dari depan dengan jarak dari kamera < 6 meter.
- Aktuator utama berupa 2 buah motor servo yang dikendalikan oleh mikrokontroler.
- PC memberikan perintah kepada kontroler melalui komunikasi serial.

### 1.4. Maksud dan Tujuan

Tujuan dari pembuatan alat dalam Tugas Akhir ini adalah

- Membuat *tracking* kamera yang dapat mendeteksi postur tubuh dengan menggunakan *haar training* pada *OpenCV*.
- Membuat *tracking* kamera yang dapat mengikuti objek berdasarkan evaluasi gambar.

### 1.5. Sistematika Laporan

Sistematika pembahasan Tugas Akhir ini terdiri dari lima bab, yaitu pendahuluan, teori penunjang, perencanaan dan pembuatan alat, pengujian dan analisa alat, serta penutup.

#### BAB I : PENDAHULUAN

Membahas tentang latar belakang, permasalahan, batasan masalah, maksud dan tujuan, sistematika laporan, serta relevansi.

#### BAB II : TEORI PENUNJANG

Berisi tentang teori penunjang yang mendukung dalam perencanaan dan pembuatan alat.

#### BAB III : PERENCANAAN DAN PEMBUATAN ALAT

Membahas tentang perencanaan dan pembuatan perangkat keras yang meliputi rangkaian-rangkaian, desain miniatur rancang bangun, dan perangkat lunak yang meliputi program yang akan digunakan untuk mengaktifkan alat tersebut.

## BAB IV : PENGUJIAN DAN ANALISA ALAT

Membahas tentang pengukuran, pengujian, dan penganalisaan terhadap hasil alat yang telah kami buat.

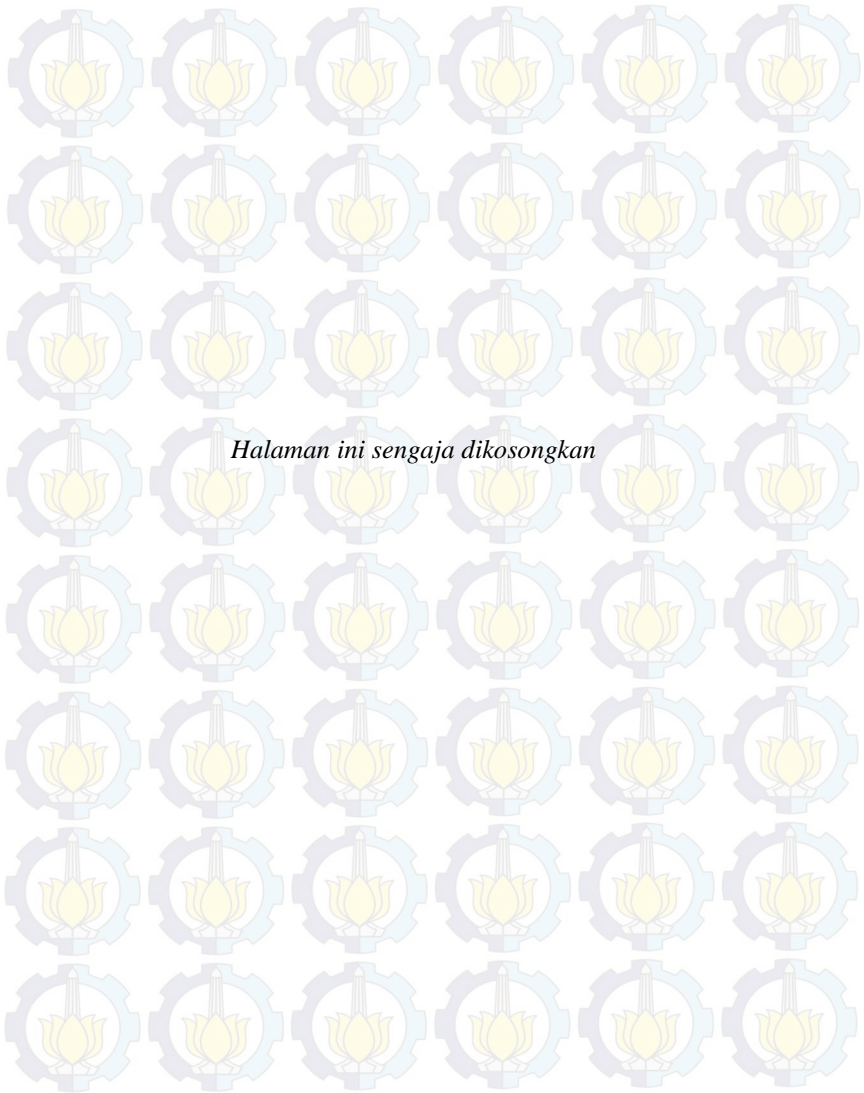
## BAB V : PENUTUP

Menjelaskan tentang kesimpulan dari Tugas Akhir ini dan saran-saran untuk pengembangan alat ini lebih lanjut.

### 1.6. Relevansi

Manfaat dari pembuatan Perancangan Sistem *Tracking Camera* Pengawas Berdasarkan Evaluasi Data Gambar dalam Tugas Akhir ini antara lain :

1. Dapat digunakan untuk sistem pengawas dengan kamera dan sistem mekanik *tracking camera* dapat mendeteksi keberadaan objek bergerak.
2. Dapat digunakan di perumahan, museum, pemantauan aktifitas karyawan, mesin-mesin produksi,dll.



## BAB II

### TEORI PENUNJANG

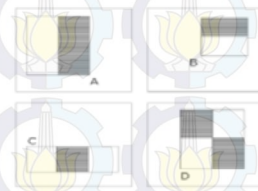
Dalam bab ini akan dijelaskan mengenai teori penunjang yang terdiri teori-teori penunjang yang berhubungan dengan Tugas Akhir yang dibuat.

#### 2.1. Metode Viola-jones

Proses deteksi adanya citra postur tubuh dalam sebuah gambar pada *OpenCV*, menggunakan sebuah metode yang dipublikasikan oleh Paul Viola dan Michael Jones tahun 2001. Umumnya disebut metode Viola-Jones. Pendekatan untuk mendeteksi objek dalam gambar menggabungkan empat konsep utama :

- Fitur segi empat sederhana yang disebut fitur *Haar*.
- Integral image untuk pendeteksian fitur secara cepat.
- Metode *machine learning AdaBoost*.
- Pengklasifikasi bertingkat (*Cascade classifier*)

Contoh fitur pada viola-jones terdapat pada Gambar 2.1 di bawah ini.

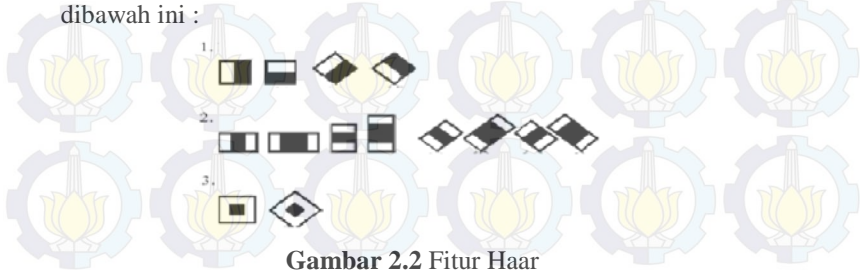


**Gambar 2.1** Contoh Fitur yang Digunakan Viola-Jones

Untuk menghubungkan banyak fitur secara efisien. 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 ). Untuk dua dimensi, satu terang dan satu gelap. Selanjutnya kombinasi-kombinasi kotak yang digunakan untuk pendeteksian objek visual yang lebih baik disebut fitur *Haar*, atau fitur *Haarlike*, fitur yang digunakan dalam pendeteksian citra tubuh oleh viola dan jones.

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 *threshold*, maka dapat dikatakan bahwa fitur tersebut ada. Untuk menentukan ada atau tidaknya dari ratusan fitur *Haar* pada sebuah gambar dan pada skala yang berbeda secara efisien, Viola dan Jones menggunakan satu teknik yang disebut *Integral Image*.

Prosedur pendeteksian obyek ini menggolongkan gambar berdasarkan pada nilai dari fitur sederhana seperti pada Gambar 2.2 dibawah ini :



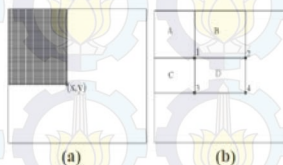
**Gambar 2.2** Fitur Haar

(1. Fitur dua kotak, 2. Fitur tiga kotak, 3. Fitur kotak di tengah).

Fitur yang digunakan terdiri dari 3 macam fitur, yaitu :

- Fitur dengan dua kotak yaitu putih dan hitam yang bersebelahan) adalah perbedaan antara penjumlahan dari piksel-piksel di dalam dua daerah rectangular. Daerah daerah dari fitur mempunyai bentuk dan ukuran sama baik secara horisontal ataupun tegak lurus (bersebelahan putih dan hitam).
- fitur dengan tiga kotak ialah pengambilan nilai dilakukan dengan cara menghitung jumlah dalam dua kotak terluar dikurangi dengan jumlah dari kotak yang berada di dalam.
- Fitur kotak di tengah dilakukan dengan menghitung perbedaan tiap kotak yang diagonal.

Pada umumnya, pengintegrasian 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 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 2.3** Gambar Integral

Seperti yang ditunjukkan oleh gambar 2.3(a) di atas setelah pengintegrasian, nilai pada lokasi piksel (x,y) berisi jumlah dari semua

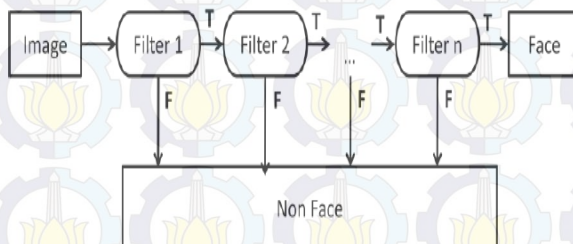


piksel di dalam daerah segiempat dari kiri atas sampai pada lokasi (x,y) atau daerah yang diarsir. Untuk menentukan nilai rata-rata piksel pada area segiempat (daerah yang diarsir) ini dapat dilakukan hanya dengan membagi nilai pada (x,y) oleh area segiempat.

Untuk mengetahui nilai piksel untuk beberapa segiempat yang lain misal, seperti segiempat D pada gambar 2.3(b) di atas dapat dilakukan dengan cara menggabungkan jumlah piksel pada area segiempat  $A+B+C+D$ , dikurangi jumlah dalam segiempat  $A+B$  dan  $A+C$ , ditambah jumlah piksel di dalam  $A$ . Dengan,  $A+B+C+D$  adalah nilai dari *integral image* pada lokasi 4,  $A+B$  adalah nilai pada lokasi 2,  $A+C$  adalah nilai pada lokasi 3, dan  $A$  pada lokasi 1. Sehingga hasil dari  $D$  dapat dikomputasikan  $D = (A+B+C+D)-(A+B)-(A+C)+A$ .

Untuk memilih fitur *Haar* yang spesifik yang akan digunakan dan untuk mengatur nilai ambangnya (*threshold*), Viola dan Jones menggunakan sebuah metode *machine learning* yang disebut *AdaBoost*. *AdaBoost* menggabungkan banyak *classifier* lemah untuk membuat sebuah *classifier* kuat. Lemah disini berarti urutan filter pada *classifier* hanya mendapatkan jawaban benar lebih sedikit. Jika keseluruhan *classifier* lemah digabungkan maka akan menjadi *classifier* yang lebih kuat. *AdaBoost* memilih sejumlah *classifier* lemah untuk disatukan dan menambahkan bobot pada setiap *classifier*, sehingga akan menjadi *classifier* yang kuat.

Viola Jones menggabungkan beberapa *AdaBoost classifier* sebagai rangkaian filter yang cukup efisien untuk menggolongkan daerah *image*. Masing-masing filter adalah satu *AdaBoost classifier* terpisah yang terdiri *classifier* lemah atau satu filter *Haar*, terdapat pada Gambar 2.4 dibawah ini:



**Gambar 2.4** Cascade classifier

*Threshold* yang dapat diterima untuk masing-masing level filter di set rendah. Selama proses pemfilteran, bila ada salah satu filter gagal untuk melewati sebuah daerah gambar, maka daerah itu langsung digolongkan sebagai bukan wajah. Namun ketika filter melewati sebuah daerah gambar dan sampai melewati semua proses filter yang ada dalam rangkaian filter, maka daerah gambar tersebut digolongkan sebagai tubuh. Viola dan Jones memberi sebutan *cascade*. Urutan filter pada *cascade* ditentukan oleh bobot yang diberikan AdaBoost. Filter dengan bobot paling besar diletakkan pada proses pertama kali, bertujuan untuk menghapus daerah gambar bukan tubuh secepat mungkin.

## 2.2. Pengolahan Citra Digital

Pengolahan citra merupakan suatu metode yang digunakan untuk mengolah citra sehingga menghasilkan citra lain sesuai keinginan dengan kualitas yang lebih baik. Citra dalam perwujudannya dapat bermacam-macam, mulai dari gambar hitam putih dalam sebuah foto (yang tidak bergerak) sampai pada gambar berwarna yang bergerak.

Proses transformasi dari bentuk tiga dimensi ke bentuk dua dimensi untuk menghasilkan citra akan dipengaruhi oleh bermacam-macam faktor yang mengakibatkan penampilan citra suatu benda tidak sama persis seperti bentuk fisik nyatanya. Faktor-faktor tersebut merupakan efek degradasi atau penurunan kualitas yang dapat berupa rentang kontras benda yang terlalu sempit atau terlalu lebar, distorsi geometri (*geometric distortion*), kekaburan (*blur*), kekaburan akibat obyek citra yang bergerak (*motion blur*), gangguan (*noise*) yang disebabkan interferensi peralatan pembuat citra, baik itu berupa peralatan elektronik ataupun peralatan *optic*. Pengolahan citra ini dapat diaplikasikan ke dalam berbagai bidang antara lain dalam bidang robotik, biomedik, penginderaan jarak jauh, hiburan, geologi maupun hukum.

Adapun beberapa proses pengolahan citra antara lain :

1. Penjumlahan Warna
2. Perubahan Kecerahan Gambar (*Image Brightness*)
3. Penskalaan Citra
4. Pengenalan Pola

### 2.2.1. Citra Digital

Citra digital merupakan fungsi intensitas cahaya  $f(x,y)$ , dimana harga  $x$  dan  $y$  merupakan koordinat spasial dan harga fungsi tersebut pada setiap titik  $(x,y)$  merupakan tingkat kecermerlangan citra pada titik tersebut. Citra digital adalah citra  $f(x,y)$  dimana dilakukan diskritisasi koordinat spasial (*sampling*) dan diskritisasi tingkat kecermerlangannya/keabuan (*kwantisasi*).

Citra digital merupakan suatu matriks dimana indeks baris dan kolomnya menyatakan suatu titik pada citra tersebut dan elemen matriksnya (yang disebut sebagai elemen gambar / *pixel* / *picture element* / *pels*) menyatakan tingkat keabuan pada titik tersebut.

### 2.2.2 Resolusi Citra

Dikenal: resolusi spasial dan resolusi kecermerlangan, berpengaruh pada besarnya informasi citra yang hilang.

- a. Resolusi spasial: halus / kasarnya pembagian kisi-kisi baris dan kolom. Transformasi citra kontinue ke citra digital disebut dijitisasi (*sampling*). Hasil digitasi dengan jumlah baris 256 dan jumlah kolom 256 - resolusi spasial  $256 \times 256$ .
- b. Resolusi kecermerlangan (intensitas / *brightness*): halus / kasarnya pembagian tingkat kecermerlangan. Transformasi data analog yang bersifat kontinue ke daerah intensitas diskrit disebut kwantisasi. Bila intensitas piksel berkisar antara 0 dan 255 - resolusi kecermerlangan citra adalah 256.

### 2.2.3 Jenis Citra

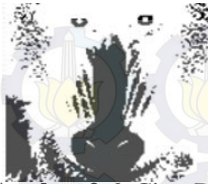
- a. Citra Grey-Scale (hitam putih)  $I(x,y) \in [0...255]$



**Gambar 2.5** Citra Gray-Scale.

Pada Gambar 2.5 merupakan contoh dari citra *grey scale* yang mana, mengubah atau mengolah data gambar menjadi hitam putih.

**b. Citra Biner  $I(x,y) \in \{0,1\}$**



**Gambar 2.6** Citra Biner

Pada Gambar 2.6 merupakan contoh dari citra biner, yang mana mengubah atau mengolah data gambar menjadi data 0 atau 1. Sehingga dapat dikatakan x dan y antara 0 dan 1.

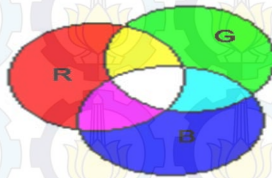
**c. Citra Berwarna  $I_R(x,y) I_G(x,y) I_B(x,y)$**



**Gambar 2.7** Citra Berwarna

Pada Gambar 2.7 merupakan contoh dari citra berwarna, yang mana mengubah atau mengolah data gambar menjadi data yang memiliki warna yang banyak. Sehingga didapat seperti nilai aslinya.

**2.2.4 Format Warna RGB**



**Gambar 2.8** Warna RGB

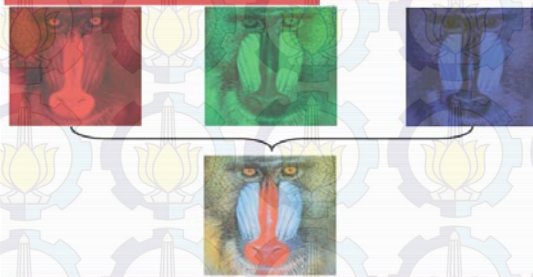
Pada Gambar 2.8 berupa format warna 24 BIT dinyatakan dengan

**11001001** Red (R) (8 bit)

**01011001** Green (G) (8 bit)

**00001011** Blue (B) (8 bit)

Masing-masing komponen warna RGB mempunyai nilai 0 s/d 255 (8 bit) derajat kecerahan (derajat keabuan) terdapat pada Gambar 2.9.



**Gambar 2.9** Gabungan Citra RGB.

### 2.3. *Open Cv Library*

*Open CV* adalah singkatan dari *Open Computer Vision*, yaitu *library-library open source* yang di khususkan untuk melakukan *image processing*. Tujuannya adalah agar komputer mempunyai kemampuan yang mirip dengan cara pengolahan visual pada manusia. *Library* ini dibuat untuk bahasa C/C++ sebagai optimasi *realtime* aplikasi, mempunyai API (*Aplication Programming Interface*) untuk *High level* maupun *Low level*, terdapat fungsi-fungsi yang siap pakai untuk *loading*, *saving*, *acucision* gambar dan video. Pada *library OpenCV* ini mempunyai *feature* sebagai berikut :

- Manipulasi data gambar (alokasi memori, melepaskan memori, kopi gambar, setting serta konversi gambar)
- *Image/Video I/O* (Bisa menggunakan camera yang sudah didukung oleh *library* ini)
- Manipulasi matrix dan vektor serta terdapat juga *routines linear algebra (products, solvers, eigenvalues, SVD)*
- *Image processing* dasar (*filtering, edge detection*, pendeteksian tepi, *sampling* dan *interpolasi*, konversi warna, operasi morfologi, *histograms, image pyramids*).
- Analisis struktural
- kalibrasi kamera
- Pendeteksian gerak
- Pengenalan objek

- *Basic GUI (Display gambar/video, mouse/keyboard kontrol, scrollbar)*
- *Image Labelling (line, conic, polygon, text drawing)*

#### 2.4. Kamera

*Webcam* singkatan dari *web* dan *cam* adalah kamera video sederhana berukuran relatif kecil, dan sering digunakan untuk konferensi video jarak jauh atau sebagai kamera pemantau. *Webcam* pada umumnya tidak membutuhkan kaset dan tempat penyimpanan data, data hasil perekaman yang didapat langsung ditransfer ke komputer.

Kamera *webcam* juga disebut periferal berupa kamera sebagai pengambil citra/ gambar dan mikropon (*optional*) sebagai pengambil suara/audio yang dikendalikan oleh sebuah komputer atau oleh jaringan. Gambar yang diambil oleh *webcam* ditampilkan ke layar monitor, karena dikendalikan oleh komputer maka ada *interface* atau *port* yang digunakan untuk menghubungkan *webcam* dengan komputer atau jaringan.

Pada sistem ini, kamera yang digunakan adalah kamera *web*, atau yang sering disebut *webcam*. Untuk pemilihan kamera, sebaiknya kita memilih *webcam* yang memiliki kecepatan serta resolusi yang bagus dan juga harus memiliki kemampuan yang baik dalam adaptasi terhadap cahaya di sekelilingnya. Pada Gambar 2.10 merupakan kamera yang digunakan:



Gambar 2.10 *Webcam A4tech*

Kamera *webcam* yang digunakan pada Tugas Akhir ini adalah tipe A4tech. Spesifikasi dari kamera web A4tech terdapat pada Tabel 2.1.

**Tabel 2.1** Spesifikasi kamera *webcam* A4tech

<i>Model</i>	<i>PK-710G</i>
<i>Computer interface</i>	<i>USB 2.0</i>
<i>Still image</i>	<i>Up to 16 Megapixel, 4608x3456</i>
<i>Image sensor</i>	<i>1/6 CMOS, 640 x 480 pixels</i>
<i>Frame rate</i>	<i>30 fps</i>
<i>Lens</i>	<i>F=2.5, f=3.5 mm</i>
<i>Viewing angle</i>	<i>60 degrees</i>
<i>Focus range</i>	<i>Automatic focus 10 cm to infinit</i>
<i>Exposure control</i>	<i>Automatic</i>
<i>White balance</i>	<i>Automatic</i>

Dan pada Gambar 2.11 merupakan kamera yang digunakan untuk tracking object.



**Gambar 2.11** *Webcam M-Tech 1,3 MP*

Spesifikasi dari kamera web A4tech terdapat pada Tabel 2.2.

**Tabel 2.2** Spesifikasi kamera *webcam M-Tech 1,3MP*

<i>Model</i>	<i>M-Tech</i>
<i>Computer interface</i>	<i>USB 2.0</i>
<i>Still image</i>	<i>1,3 Megapixel</i>
<i>Frame rate</i>	<i>30 fps</i>

## 2.5. *Personal Computer (PC)*

*PC* atau *Personal Computer* adalah seperangkat unit pemroses data yang terdiri dari *monitor, keyboard, motherboard, processor,* dan lain-lain. Fungsi *PC* pada sistem ini adalah sebagai "otak" yang harus mengambil keputusan untuk langkah berikut yang harus diambil. *PC* ini menerima informasi dari *webcam* yang berbentuk image yang merepresentasikan situasi di sekeliling sistem. Data ini dikirim oleh *webcam* melalui jalur *USB* yang dimiliki oleh *PC*. Terdapat pada Gambar 2.12 merupakan *PC* yang digunakan:



**Gambar 2.12** Laptop (*PC*)

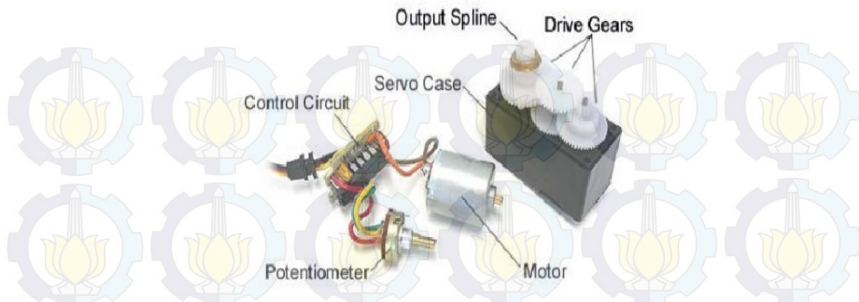
## 2.6. **Motor Servo**

Dalam subab motor servo ini akan dijelaskan beberapa hal yaitu pengertian motor servo, prinsip dasar cara kerja, dan prinsip arah putaran motor.

### 2.6.1 **Pengertian Motor Servo**

Motor listrik merupakan perangkat elektromagnetis yang mengubah energi listrik menjadi energi mekanik. Motor servo adalah sebuah motor dengan sistem *closed feedback* di mana posisi dari motor akan diinformasikan kembali ke rangkaian kontrol yang ada di dalam motor servo. Seperti pada Gambar 2.13 merupakan gambar bagian-bagian dari motor servo.





**Gambar 2.13** Bagian-Bagian Motor Servo

Motor ini terdiri dari sebuah motor DC, serangkaian *gear*, potensiometer, dan rangkaian kontrol. Potensiometer berfungsi untuk menentukan batas sudut dari putaran servo. Sedangkan sudut dari sumbu motor servo diatur berdasarkan lebar pulsa yang dikirim melalui kaki sinyal dari kabel motor. Motor servo terdiri dari dua jenis, yaitu:

1. Motor Servo *Standart*

Motor servo yang mampu bergerak CW (*clockwise*) dan CCW (*counter clockwise*) dengan sudut perasi tertentu, misalnya 60°, 90°, atau 180°.

2. Motor Servo *Continuous*

Motor servo yang mampu bergerak CW dan CCW tanpa batasan sudut operasi atau berputar 360°.

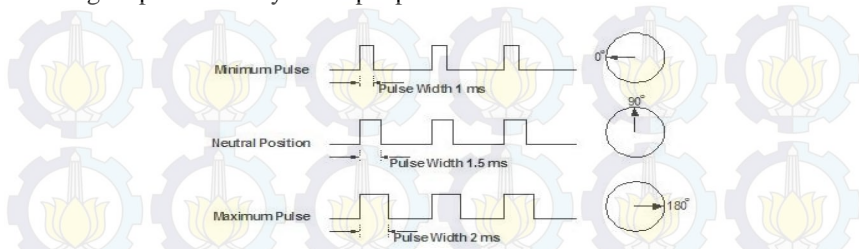
### 2.6.2 Prinsip Dasar Cara Kerja

Motor servo biasanya menggunakan tegangan supply 4.8 volt atau 7.2 volt. Pada dasarnya motor servo dikendalikan dengan cara mengirimkan sebuah lebar pulsa tertentu. Hanya salah satu perbedaannya yaitu pada sudut putarnya. Untuk servo standard sudut putarnya adalah 180°. Pulsa tersebut akan dimasukkan melalui kabel kontrol motor servo. Lebar pulsa tersebut yang akan mengakibatkan perubahan posisi pada motor servo.

### 2.6.3 Prinsip Arah Putaran Motor

Motor servo *standart* adalah motor yang mampu bekerja dua arah yaitu CW (*clockwise*) atau CCW (*counter clockwise*) di mana arah dan sudut pergerakan rotornya dapat dikendalikan dengan

memberikan pengaturan lebar pulsa (*duty cycle*) sinyal PWM pada bagian pin kontrolnya terdapat pada Gambar 2.14.



**Gambar 2.14** Arah Putaran Motor Servo

Operasional dari servo motor dikendalikan oleh sebuah pulsa selebar 20 ms. Motor Servo akan bekerja secara baik jika pada bagian pin kontrolnya diberikan sinyal PWM dengan frekuensi 50Hz. Adapun posisi dan pembentukan pulsa pada motor servo, yaitu:

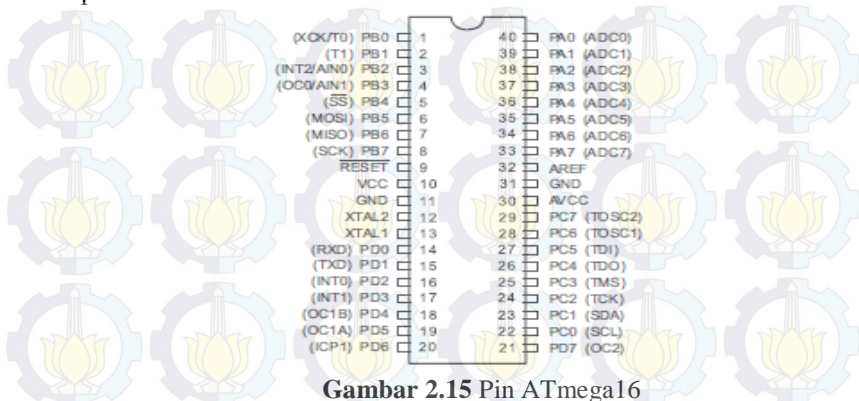
1. Saat sinyal dengan frekuensi 50Hz tersebut dicapai pada kondisi lebar pulsa 1.5ms, maka rotor dari motor akan berhenti tepat di tengah-tengah (sudut  $0^\circ$  atau netral).
2. Pada saat lebar pulsa dari sinyal yang diberikan kurang dari 1.5ms, maka rotor akan berputar ke arah kiri dengan membentuk sudut yang besarnya linier terhadap besarnya lebar pulsa dan akan bertahan diposisi tersebut.
3. Dan sebaliknya, jika lebar pulsa dari sinyal yang diberikan lebih dari 1.5ms, maka rotor akan berputar ke arah kanan dengan membentuk sudut yang linier pula terhadap besarnya lebar pulsa dan bertahan diposisi tersebut.

## 2.7. Mikrokontroler ATmega16

Mikrokontroler adalah sebuah sistem komputer lengkap dalam satu *chip*. Mikrokontroler lebih dari sekedar sebuah mikroprosesor karena sudah berisi ROM (*Read-Only Memory*), RAM (*Read-Write Memory*), beberapa *port* masukan maupun keluaran, dan beberapa *peripheral* seperti pencacah/pewaktu, ADC (*Analog to Digital converter*), DAC (*Digital to Analog converter*), dan serial komunikasi.

Mikrokontroler AVR ATmega16 Memiliki memori flash 16k dan 32 jalur input output, serta dilengkapi dengan ADC 8 kanal dengan resolusi 10-bit dan 4 kanal PWM. Sebuah *chip* dengan fitur cukup lengkap untuk mendukung beragam aplikasi. Berikut

merupakan gambar pin mikrokontroler ATmega16 yang ditunjukkan pada Gambar 2.15.



**Gambar 2.15** Pin ATmega16

Untuk memaksimalkan dalam penggunaan mikrokontroler ATmega16 maka perlu diperhatikan bahwa pin-pin ATmega16 baik dalam penggunaan *port A*, *port B*, *port C*, *port D*, maupun penggunaan ADC agar disesuaikan dengan kebutuhan dan *hardware* yang digunakan.

Mikrokontroler ATmega16 terdiri dari 40 pin dan 4 buah *Port* , yaitu *port A*, *port B*, *port C*, dan *port D*. Keempat *port* tersebut memiliki fungsi sebagai masukan atau keluaran dari mikrokontroler dan juga memiliki fungsi-fungsi khusus yang dapat dijelaskan pada Tabel 2.3 bawah ini.

**Tabel 2.3** Fungsi Pin ATmega16

Pin	Nama Pin	Fungsi
Pin 1-Pin 8	<i>Port B</i>	<i>Port</i> paralel 8 bit dua arah ( <i>input/output</i> ) dan pin fungsi khusus
Pin 9	<i>Reset</i>	Pin yang digunakan untuk mereset mikrokontroler
Pin 10	Vcc	Pin masukan catu daya
Pin 11 dan Pin 31	GND	Pin <i>ground</i>
Pin 12 pin 13	XTAL2 dan XTAL1	Pin masukan <i>clock external</i>
Pin 14-Pin 21	<i>Port D</i>	<i>Port</i> paralel 8 bit dua arah ( <i>Input/Output</i> ) dan pin fungsi

Pin	Nama Pin	Fungsi
		khusus
Pin 22-Pin 29	<i>Port C</i>	<i>Port</i> paralel 8 bit dua arah ( <i>Input/Output</i> ) dan pin fungsi khusus
Pin 30	AVcc	Pin masukan tegangan untuk ADC
Pin 32	AREF	Pin masukan tegangan untuk referensi ADC
Pin 33-Pin 40	<i>Port A</i>	Pin <i>input/output</i> dua arah dan pin masukan ADC

### 2.8. *Power Supply*

*Power supply* adalah rangkaian elektronika yang berfungsi sebagai pencatu daya ke komponen lain pada perangkat elektronika. Semua komponen elektronika yang ada dalam suatu perangkat elektronika akan memperoleh masukan daya dari *power supply* tersebut. *Power supply* mempunyai peranan yang penting dalam suatu perangkat elektronika. Adapun tegangan yang umum disediakan oleh *power supply* adalah +5V dan +12V. Besar tegangan keluaran dari *power supply* ini juga harus disesuaikan dengan kebutuhan tegangan beban atau perangkat elektronika yang digunakan.

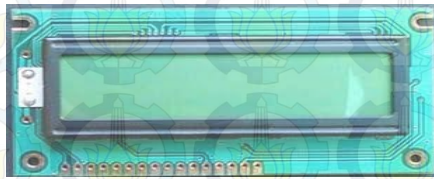
Dalam rangkaian *power supply* terdapat *rectifier* yang berfungsi sebagai penyearah, filter untuk membuat sinyal keluaran dari rangkaian penyearah agar lebih lurus dan halus, dan regulator yang dapat meregulasi tegangan yang diinginkan, ditunjukkan pada Gambar 2.16.



**Gambar 2.16** *Power Supply*

## 2.9. LCD 16 x 2

LCD (*Lyquid Cristal Display*) merupakan suatu bentuk kristal cair yang akan menyala apabila diberi tegangan. LCD berfungsi sebagai ouput yang dapat menampilkan suatu data berupa karakter, huruf, ataupun angka. Didalam LCD terdapat sebuah mikroprosesor yang mengendalikan tampilan. Untuk menjalankan LCD perlu dibuat sebuah program untuk berkomunikasi. Modul LCD terdiri dari sejumlah memori yang digunakan sebagai *display*. Semua teks yang kita tuliskan ke modul LCD akan disimpan dalam memori ini dan modul LCD secara berurutan membaca memori ini untuk menampilkan teks sesuai dengan program yang telah dibuat, ditunjukkan pada Gambar 2.17 di bawah ini.



**Gambar 2.17** Bentuk LCD 16x2

LCD yang dipergunakan mempunyai spesifikasi sebagai berikut :

- a. Terdiri dari 16 karakter yang tersusun dalam dua baris (masing-masing 8 karakter) dengan *display dot matrik 5x7*.
- b. Karakter generator ROM dengan 192 tipe karakter.
- c. Karakter generator RAM dengan 8 tipe karakter.
- d. 80 x 8 *bit display data RAM*.
- e. Catu daya + 5 V.
- f. *Reset* pada saat *power on*.

Di tunjukkan pada Tabel 2.4 di bawah ini:

**Tabel 2.4** Fungsi Pin LCD

Pin	Nama Pin	Fungsi
Pin 16	V – BL	Sebagai <i>ground</i> dari <i>backlight</i>
Pin 15	V + BL	Sebagai kutub positif dari <i>backlight</i>
Pin 7-Pin 14	DB0– DB7	Merupakan saluran data, berisi perintah dan data yang akan ditampilkan.
Pin 6	E	Sinyal operasi awal, sinyal ini mengaktifkan data tulis atau baca
Pin 5	R/W	Sinyal seleksi tulis atau baca 0 = tulis 1 = baca
Pin 4	RS	Sinyal pemilih <i>register</i> 0 = instruksi <i>register</i> (tulis) 1 = data <i>register</i> (tulis dan baca)
Pin 3	Vlc	Untuk mengendalikan kecerahan LCD dengan merubah Vlc
Pin 2	Vcc	Tegangan catu + 5 V
Pin 1	Vss	Terminal <i>Ground</i>

## 2.10. Metode Komunikasi

Beragamnya metode komunikasi antarmuka memungkinkan untuk menggunakan metode komunikasi yang tepat dan sesuai dengan sistem yang akan dibangun. Salah satu metode komunikasi yang digunakan dan yang akan dijelaskan berikut ini adalah USB 2.0 dan RS232.

### 2.10.1 Komunikasi Antara Kamera Dengan PC

Metode komunikasi yang akan diterapkan pada antarmuka PC-Kamera adalah USB 2.0 sebagai standar bus berseri untuk perangkat penghubung. Dalam perkembangan teknologi piranti komunikasi, USB adalah suatu teknologi komunikasi data serial yang bersifat multi guna dengan topologi BUS. USB 1.0 memiliki kecepatan sekitar 12 Mbps, sedangkan USB 2.0 memiliki kecepatan sekitar 480 Mbps.

Sistem USB mempunyai desain yang asimetris, yang terdiri dari pengontrol *host* dan beberapa peralatan terhubung yang berbentuk pohon dengan menggunakan peralatan hub yang khusus, jenis konektor di tunjukkan pada Gambar 2.18 dibawah ini:

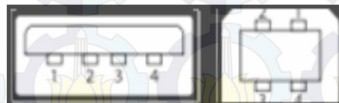


**Gambar 2.18** Jenis Konektor USB

Konektor USB memiliki 4 lempeng tembaga atau emas yang masing-masing lempeng/ kaki memiliki karakteristik berbeda-beda, yaitu;

1. Tegangan Bus yang memiliki range antara 4.75V-5.25V
2. Transfer data negatif (D -)
3. Transfer data positif (D +)
4. Lempeng Pentanahan (GND)

Penetapan kaki konektor di tunjukkan pada Gambar 2.19 dibawah ini:



**Gambar 2.19** Penetapan Kaki Konektor

### 2.10.2 Komunikasi Antara PC Dengan Kontroler

Pada prinsipnya, komunikasi serial ialah komunikasi dimana pengiriman data dilakukan per bit, sehingga lebih lambat dibandingkan komunikasi *parallel* seperti pada port printer yang mampu mengirim 8 bit sekaligus dalam sekali detak. Beberapa contoh komunikasi serial ialah *mouse*, *scanner* dan *system acuisition* data yang terhubung ke *port* COM1/COM2.

Peralatan pada komunikasi serial *port* dibagi menjadi 2 (dua) kelompok yaitu *Data Communication Equipment (DCE)* dan *Data Terminal Equipment (DTE)*. Contoh dari DCE ialah

modem, plotter, scanner dan lain lain sedangkan contoh dari DTE ialah terminal di komputer. Spesifikasi elektronik dari serial port merujuk pada

*Electronic Industry Association (EIA):*

1. “Space” (logika 0) ialah tegangan antara + 3 hingga +25 V.
2. “Mark” (logika 1) ialah tegangan antara –3 hingga –25 V.
3. Daerah antara + 3V hingga –3V tidak didefinisikan /tidak terpakai
4. Tegangan open circuit tidak boleh melebihi 25 V.
5. Arus hubungan singkat tidak boleh melebihi 500mA.

Komunikasi serial membutuhkan port sebagai saluran data. dimana *port serial DB9* yang umum digunakan sebagai *port serial*, Port DB 9 di tunjukkan pada Gambar 2.20 di bawah ini :



**Gambar 2.20** Port DB 9

Konektor *port serial* terdiri dari 2 jenis, yaitu konektor 25 pin (DB25) dan 9 pin (DB9) yang berpasangan (jantan dan betina). Bentuk dari konektor DB-25 sama persis dengan *port paralel*. Umumnya COM1 berada dialamat 3F8H, sedangkan COM2 dialamat 2F8H. Jenis pin rs 232 ditunjukkan pada Tabel 2.5 di bawah ini:

**Tabel 2.5** Jenis Pin RS 232

Nama Sinyal	Arah Sinyal	Nomor Kaki Konektor	
		DB9	DB25
Signal Common	-	5	7
Transmitted Data (TD)	Ke DCE	3	2
Received Data (RD)	Dari DCE	2	3
Request to Send (RTS)	Ke DCE	7	4
Clear to Send (CTS)	Dari DCE	8	5
DCE Ready (DSR)	Dari DCE	6	6
DTE Ready (DTR)	Ke DCE	4	20
Ring Indicator (RI)	Dari DCE	9	22
Data Carrier Detect (DCD)	Dari DCE	1	8

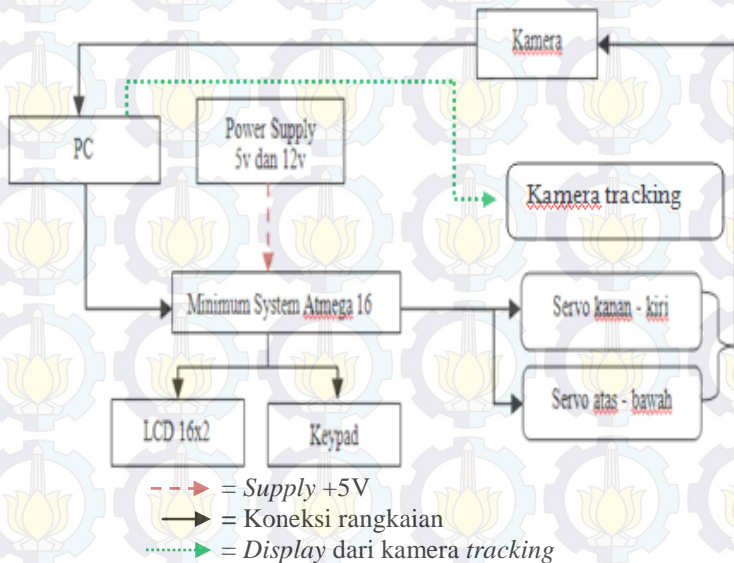




### BAB III

## PERANCANGAN DAN PEMBUATAN ALAT

Pada tahap perencanaan ini akan dibahas mengenai pembuatan dan penggunaan dari tahap perencanaan yaitu perancangan rangkaian elektronika dan perancangan mekanik. Rangkaian elektronika terdiri dari perangkat keras (*hardware*) dan perancangan perangkat lunak terdiri dari program (*software*). Perancangan perangkat keras meliputi perancangan *power supply*, mikrokontroler sistem minimum ATmega16, rangkaian RS 232, perancangan LCD 16x2, dan motor servo. Untuk perancangan perangkat lunak meliputi pembuatan program AVR yang akan diunduh ke mikrokontroler ATmega16 untuk menerima masukan dari perintah kamera yang telah di serial ke laptop kemudian menggerakkan motor servo yang mengikuti objek yang bergerak dan koordinat pada objek yang terbaca kamera ditampilkan ke LCD 16x2. Sedangkan perancangan mekanik merupakan pembuatan *prototype* mekanik kamera dan servo yang dapat di tempatkan pada dinding dan dapat di tempatkan pada aero modeling terdapat sistem *tracking* sebagai media peraga sistem. Adapun blok diagram sistem pada Gambar 3.1.



**Gambar 3.1** Blok Diagram Sistem

Konsep keseluruhan dari alat ini adalah kamera akan mendeteksi objek manusia yang bergerak pada titik terdekat dengan kamera dan di tampilkan pada computer yang mana sistem kamera yang menggerakkan kamera tracking menggunakan 2 motor servo akan mengikuti objek bergerak tersebut dan kamera tracking akan melakukan scanning apabila tidak objek yang terdeteksi.

Perancangan rangkaian elektronika sistem kontrol dibagi menjadi tiga bagian, yaitu perancangan perangkat keras (*hardware*) dan perancangan perangkat lunak (*software*), dan perancangan mekanik.

### 3.1. Perancangan Perangkat Keras (*hardware*)

Perancangan perangkat keras dirancang sesuai dengan kebutuhan sistem yakni :

#### 1. Power Supply

*Power supply* ini sebagai penyedia *supply* tegangan sebesar 5 volt dan 12 volt. *Supply* tegangan sebesar 5 volt akan digunakan untuk *Minimum system*, LCD 16x2, dan servo.

#### 2. Mikrokontroler ATmega16

Mikrokontroler berfungsi untuk memproses seluruh program dan algoritma sistem kontrol. Masukan dari program dari PC yaitu dengan pemrograman *Opencv* akan ditampilkan di LCD 16x2 dengan keluaran respon dari motor servo.

#### 3. Rangkaian Serial

Rangkaian ini berfungsi untuk membantu Mikrokontroler untuk berkomunikasi serial dengan PC dan peralatan yang terhubung dengan minimum system. Komponen inti penyusun rangkaian ini adalah sebuah IC *MAX232*. IC ini dirangkai dengan beberapa resistor sebagai penganan. Jalur komunikasi ini memiliki 3 kabel, yaitu *Tx*, *Rx* dan *Gnd*. Rangkaian ini terhubung dengan mikrokontroler pada *port* yang memiliki fungsi sebagai *Tx*, *Rx*, dan pada port *Ground* dari mikrokontroler.

#### 4. LCD 16x2

LCD ini sebagai *display* data x dan y yang digunakan untuk menampilkan perintah komputer pada mikrokontroler untuk menggerakkan motor servo.

## 5. Motor Servo

Motor servo ini digunakan sebagai pengarah pada sistem *tracking camera*. sehingga sistem *tracking* dapat mengikuti objek sesuai perintah pada program di PC dan disinkronkan pada pemrograman pada sistem minimum.

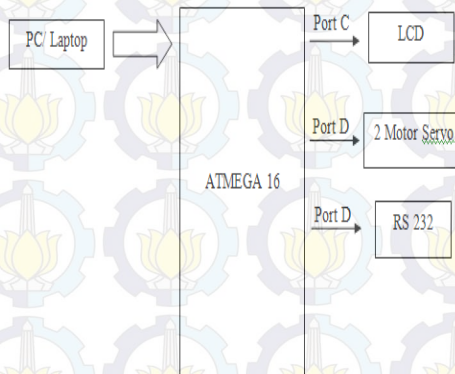
Adapun Perancangan rangkaian sistem ini yang terdiri dari sistem minimum dari mikrokontroler ATmega16, LCD 16x2, 2 buah motor servo dan *power supply*. Penggunaan masing-masing *Port* input dan output mikrokontroler ATmega16 yang terdapat pada sistem kontrol sebagai berikut :

- a. *Port* C.7 – C.0 : sebagai keluaran LCD 16x2
- b. *Port* D.0 : sebagai keluaran *rx*
- c. *Port* D.1 : sebagai keluaran *tx*
- d. *Port* D.4 - D.5 : sebagai keluaran 2 motor servo

Berikut penjelasan dari rancangan rangkaian sistem perangkat keras.

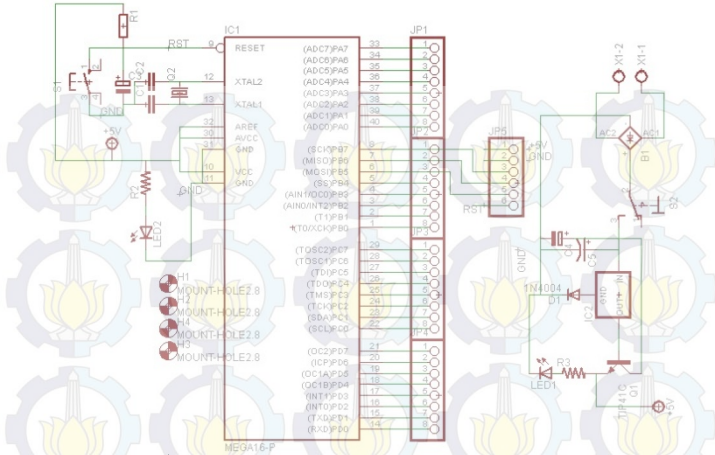
### 1. Perancangan Sistem Minimum ATmega16

Blok fungsional *port* yang digunakan pada sistem minimum mikrokontroler di tunjukkan pada Gambar 3.2 di bawah ini.



**Gambar 3.2** Blok Fungsional ATmega16

Secara lengkap rangkaian mikrokontroler ATmega16 di tunjukkan seperti pada Gambar 3.3 di bawah ini:

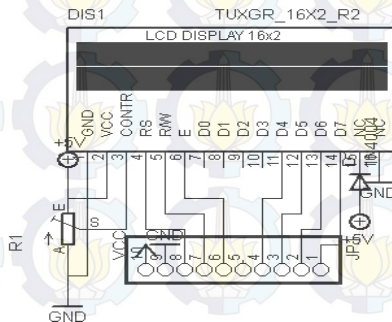


**Gambar 3.3** Schematic Rangkaian ATmega16

Minimum sistem dari *microcontroller* AVR ATmega16 ini digunakan sebagai pengatur dari semua proses yang ada. Fungsi utama dari *microcontroller* ini adalah menerima data serial program *opencv* lalu akan diterima oleh mikro melalui rangkaian *rs 232* yang akan menjalankan perintah motor servo kemana akan bergerak. Disamping itu juga menampilkan data yang masuk ke LCD.

## 2. LCD 16x2

Pembuatan LCD ini bertujuan untuk menampilkan posisi objek yang telah terdeteksi. Secara lengkap rangkaiannya terdapat pada Gambar 3.4 di bawah ini:



**Gambar 3.4** Schematic Rangkaian LCD 16x2

### 3. Motor Servo

Motor servo yang digunakan adalah motor servo standart dari Hitech tipe HS-325HB. Motor servo ini bergerak dari sudut  $0^\circ$  hingga  $180^\circ$  dengan kontrol lebar pulsa sebesar 0.9 ms hingga 2.1 ms. Motor servo ini akan mengarahkan sistem kamera *tracking* sesuai perintah kamera dari open cv dengan perhitungan titik tengah frame (0,0) dengan menyamakan nilai antara titik objek dan nilai tersebut akan dikirim ke servo untuk memerintahkan kemana servo akan bergerak. Jenis servo yang di gunakan terdapat pada Gambar 3.5 di bawah ini:



**Gambar 3.5** Motor Servo HS-325HB

Servo standart dari Hitech tipe HS-325HB ini memiliki 3 pin yaitu:

- a) Kabel warna hitam: *Ground*
- b) Kabel warna merah: *VCC*
- c) Kabel warna kuning: *Data/signal*

### 4. Perancangan *Power Supply*

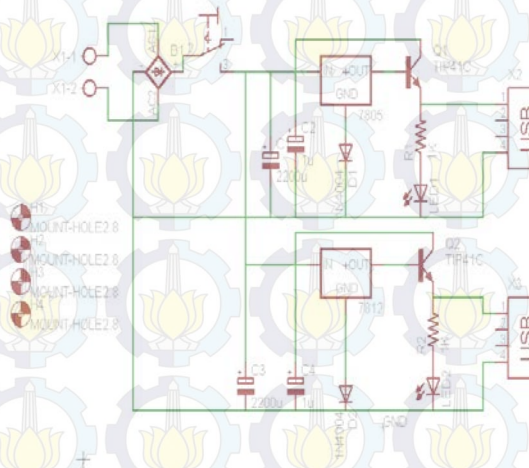
*Power supply* adalah sebuah alat atau sistem yang berfungsi untuk menyalurkan listrik atau bentuk energi jenis apapun pada beban atau sekelompok beban. *Power supply* sering digunakan untuk menyalurkan energi listrik dan beberapa digunakan pada mesin-mesin listrik. *Power supply* dapat digunakan sebagai pengganti sumber tenaga listrik sebagai sumber utama atau cadangan seperti:

- Mengubah bentuk listrik dari sumber ke bentuk tegangan yang diinginkan. Biasanya digunakan untuk mengubah sumber AC 120 atau 240 volt ke tegangan DC

yang lebih rendah untuk digunakan pada peralatan elektronik.

- Pengganti baterai
- *Generator* atau *alternator*

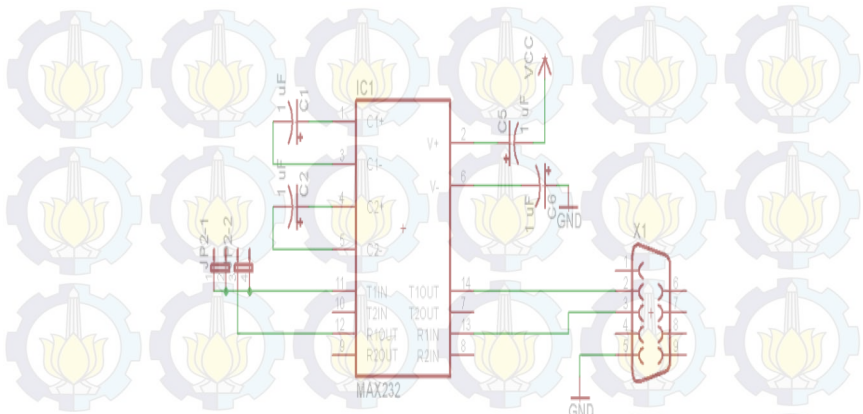
Gambar rangkaian power supply di tunjukkan pada Gambar 3.6 di bawah ini :



**Gambar 3.6** Schematic Rangkaian Power Supply

### 5. Perancangan Rangkaian *Driver* MAX232

MAX232 merupakan salah satu jenis IC rangkaian antar muka *dual RS-232 transmitter/receiver* yang memenuhi semua spesifikasi standar EIA-232-E. IC MAX232 hanya membutuhkan *power supply* 5 Volt (*single power supply*) sebagai catu. IC MAX232 di sini berfungsi untuk merubah level tegangan pada COM menjadi level tegangan TTL/CMOS. IC MAX232 terdiri atas tiga bagian yaitu *dual charge-pump voltage converter*, *driver RS232*, dan *receiver RS232*. Gambar 3.5 adalah rangkaian *driver* MAX232 di bawah ini:



**Gambar 3.7** Schematic Rangkaian Driver MAX232

### 3.2. Perancangan Perangkat Lunak ( Software )

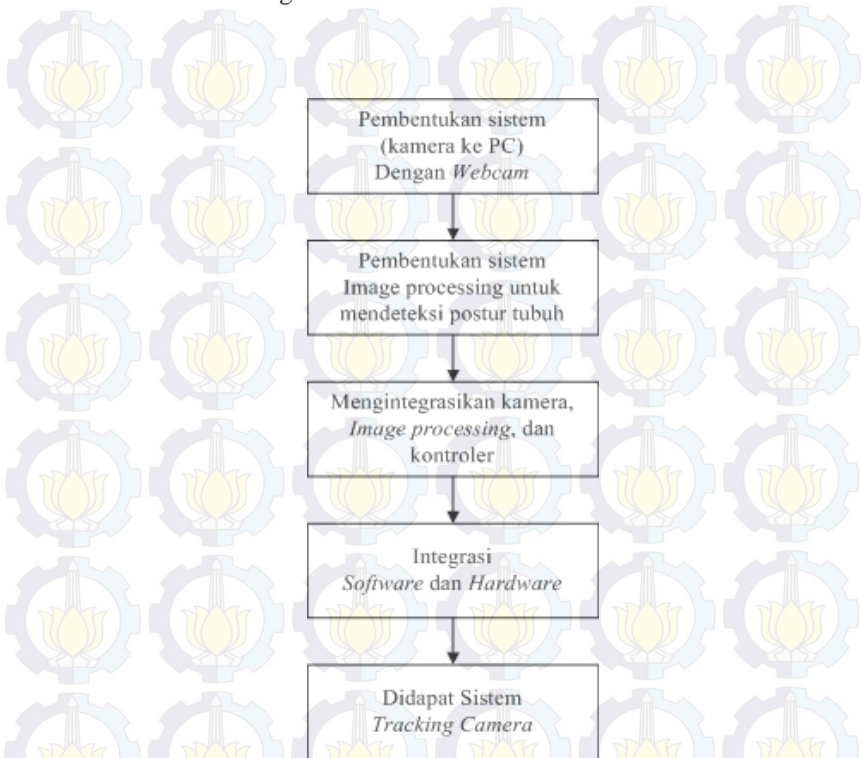
Berikut ini akan dibahas mengenai perancangan perangkat lunak (*software*) pada mikrokontroler untuk memproses algoritma dengan menggunakan *software codevison*.

Kemudian membuat Program pada PC merupakan aplikasi konsol deteksi postur tubuh yang dibuat dengan software MS Visual Studio ditambah library OpenCV dan menggunakan metode Haarcascade.

Kamera pada sistem ini berfungsi untuk melihat daerah sekeliling dan melaporkan hasil deteksi kamera ke PC. Selain kamera, pada sistem terdapat PC, PC sebagai pengambil keputusan. Dari laporan yang diberikan kamera, PC akan mempertimbangkan langkah berikutnya yang akan diambil. Keputusan sepenuhnya berada di tangan PC yang juga disebut “otak”. PC mengendalikan peralatan lain melalui rangkaian mikrokontroler. PC hanya perlu memberikan perintah melalui kabel serial yaitu dengan rangkaian RS 232. Tugas rangkaian mikrokontroller pada sistem adalah sebagai pelaksana perintah yang diberikan oleh PC. Untuk kemudian, rangkain mikrokontroler dapat mengendalikan peralatan lain yaitu motor servo. Fungsi motor servo pada sistem ini adalah sebagai penggerak kamera, sehingga sistem dapat terus menerus memantau



target. Adapun alur dari pembuatan sistem ini di tunjukkan pada Gambar 3.8 sebagai berikut:



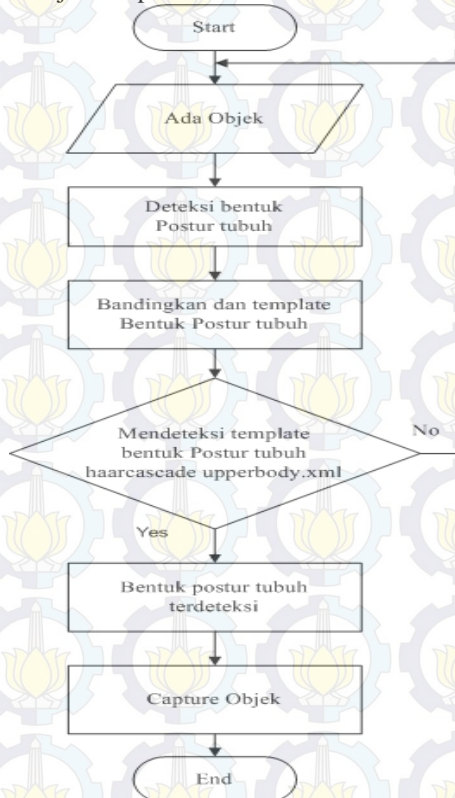
**Gambar 3.8** Alur Pembuatan Sistem

Penjelasan pada alur pembuatan sistem diatas yaitu :

1. Pembentukan sistem, dimana menghubungkan koneksi antara kamera dengan PC. Kamera yang digunakan yaitu *webcam*.
2. Pembentukan sistem *image processing* untuk mendeteksi tubuh, yaitu menggunakan metode *haarcascade training* pada library *opencv*.
3. Mengintegrasikan kamera, *image processing*, dan kontroler, yaitu membuat suatu flowchart algoritma pada image processing dengan open cv maupun kontroler dengan *codevision avr*.
4. Integrasi *software* dan *hardware*, dimana membuat dan menyatukan program di PC dengan kontroler yang digunakan.

## 5. Didapat sistem *tracking camera*

Dengan pembuatan alur pembuatan sistem yang telah direncanakan, maka dapat dibuat flowchart untuk deteksi postur tubuh dan flowchart pelacak objek di tunjukkan pada Gambar 3.9 di bawah ini :

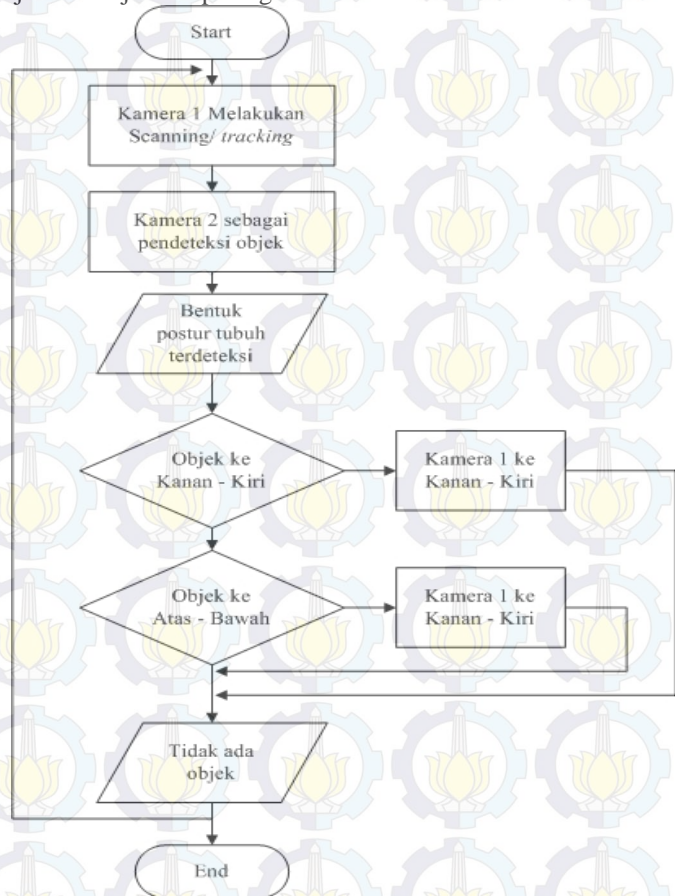


**Gambar 3.9** Flowchart Deteksi Bentuk Tubuh

Dari flowchart gambar 3.9 yaitu ketika ada objek maka sistem akan deteksi objek dari bentuk postur tubuh, kemudian dibandingkan dengan template bentuk postur tubuh yang sudah di buat dengan lebih dari ribuan bentuk postur tubuh yang diambil dari *haar training* pada *library opencv*. Setelah dibandingkan dengan sample bentuk postur tubuh, maka akan di deteksi sebagai manusia. Lalu di beri tanda bulat biru terhadap obyek yang dideteksi sebagai manusia dengan parameter

bentuk postur tubuh. Kemudian obyek yang terdeteksi akan di capture secara *realtime*.

Setelah mendapatkan program deteksi postur tubuh pada program *opencv*,selanjutnya mengintegrasikan dan menggabungkan program sebagai pelaksana perintah yang diberikan, yaitu dengan program *codevision avr*. Pelacakan object di tunjukkan pada gambar 3.10 di bawah ini :



**Gambar 3.10** Flowchart Pelacakan Objek

Penjelasan dari flowchart pelacakan obyek adalah Ketika postur tubuh obyek sudah terdeteksi oleh kamera 1 maka akan

mengerakkan kamera tracking mengikuti pergerakan obyek jika ke kanan maka akan ke kanan begitu sebaliknya, ketika jejak obyek menghilang maka kamera *tracking* akan melakukan scanning kekanan – kekiri, jika tidak ditemukan obyek maka camera akan berhenti jika ditemukan maka akan mendeteksi postur tubuh obyek kembali.

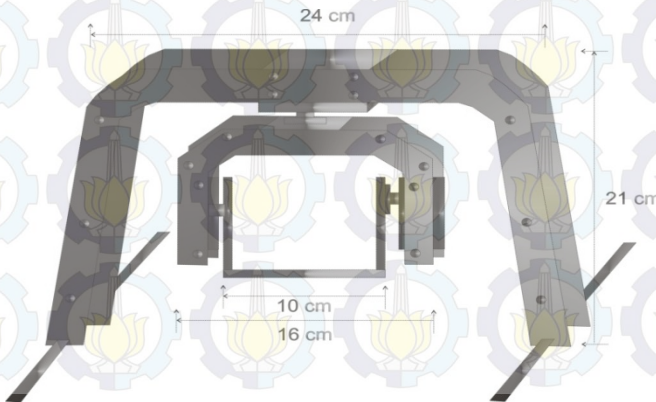
### 3.3. Perancangan Mekanik

Perancangan mekanik berupa pembuatan *box* yang terbuat dari akrilik. Yang didalamnya terdapat sistim minimum, supply, LCD, downloader, dan RS 232 di tunjukkan pada Gambar 3.9 di bawah ini:



**Gambar 3.9** Gambar *Box* Untuk Kontroler

Dalam menjalankan *system tracking* ini membutuhkan alat peraga berupa mekanik kamera *tracking*, miniatur ini terdiri dari kamera dan 2 servo. Miniatur ini dibuat berbalaskan acrylic dengan ketebalan 3mm yang memiliki spesifikasi ukuran dimensi panjang 24 cm dan lebar 21cm di tunjukkan pada Gambar 3.10 di bawah ini :



**Gambar 3.10** Sketsa Mekanik *Tracking Camera*



## BAB IV

### PENGUJIAN DAN ANALISA DATA

Pada bab ini akan dibahas mengenai pengujian dan analisa alat maupun sistem secara keseluruhan untuk mengetahui apakah tujuan-tujuan dari pembuatan alat ini telah terlaksana atau tidak terhadap alat yang dibuat.

#### 4.1. Pengujian dan Analisa Rangkaian *Power Supply*

Rangkaian *power supply* digunakan untuk menyediakan *supply* sebesar +5 Volt dan +12 Volt. Pengujian dilakukan dengan menggunakan AVOMeter *digital* dan pengujian terhadap rangkaian ini menghasilkan data seperti pada Tabel 4.1 berikut.

**Tabel 4.1** Data Pengukuran Rangkaian *Power Supply*

Pengukuran ke-	+5 V	+12 V
	Vout 5V (V)	Vout 12V (V)
1	5.5	12.5
2	5.5	12.5
3	5.5	12.5
4	5.5	12.5
<b>Σ Rata-rata</b>	<b>5.5</b>	<b>12.5</b>

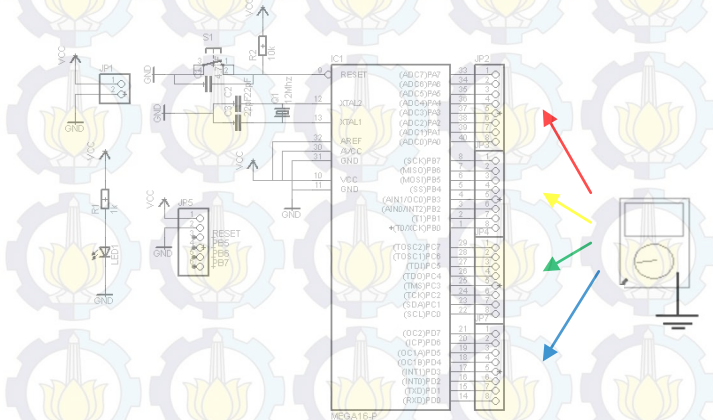
$$\begin{aligned} \% \text{ Error untuk supply +5 Volt} &= [(5-5,5)/5] \times 100 \% \\ &= 10 \% \end{aligned}$$

$$\begin{aligned} \% \text{ Error untuk supply +12 Volt} &= [(12-12,5)/12] \times 100 \% \\ &= 4,16 \% \end{aligned}$$

Didapatkan % *Error* sebesar 10% untuk *supply* +5 Volt dan 4,16% untuk *supply* +12 Volt. Dari hasil yang didapat masih tergolong kecil sehingga tegangan sebesar 5,5 Volt dan 12,5 Volt dapat digunakan untuk menyuplai rangkaian lainnya. Yaitu 5,5 Volt untuk menyuplai rangkaian sistem minimum 16, sumber servo motor dan sumber serial RS 232.

## 4.2. Pengujian dan Analisa Mikrokontroler

Pengujian rangkaian Sistem Minimum Mikrokontroler ATmega16 dengan menggunakan rangkaian pada Gambar 4.1 telah dilakukan dengan menerapkan program sederhana pada mikrokontroler tersebut, di mana aplikasinya adalah untuk menyalakan dan mematikan 8 indikator *LED* pada tiap *port* yang dimiliki oleh Sistem Minimum ATmega16 dan melakukan pengukuran tegangan pada tiap *port*nya saat diberi logika 1 (*high*) maupun logika 0 (*low*) di tunjukkan pada Gambar 4.1 di bawah ini.



**Gambar 4.1** Rangkaian Mikrokontroler

1. Pengukuran tegangan keluaran di *PORT A* ditunjukkan pada tanda panah berwarna merah dengan tegangan dari *power supply* sebesar 5.5 Volt. Output *PORT A* dihubungkan dengan kaki positif pada AVOMeter digital dan kaki negatif dihubungkan pada *ground*.
2. Pengukuran tegangan keluaran di *PORT B* ditunjukkan pada tanda panah berwarna kuning dengan tegangan dari *power supply* sebesar 5.5 Volt. Output *PORT B* dihubungkan dengan kaki positif pada AVOMeter digital dan kaki negatif dihubungkan pada *ground*.
3. Pengukuran tegangan keluaran di *PORT C* ditunjukkan pada tanda panah berwarna hijau dengan tegangan dari *power supply* sebesar 5.5 Volt.

Output *PORT C* dihubungkan dengan kaki positif pada AVOMeter *digital* dan kaki negatif dihubungkan pada *ground*.

4. Pengukuran tegangan keluaran di *PORT D* ditunjukkan pada tanda panah berwarna biru dengan tegangan dari *power supply* sebesar 5.5 Volt.

Output *PORT D* dihubungkan dengan kaki positif pada AVOMeter *digital* dan kaki negatif dihubungkan pada *ground*.

*Wiring* pada sistem minimum yang di tunjukkan pada Gambar 4.2 di bawah ini.



**Gambar 4.2** Sistem Minimum ATmega 16

Hasil dari pengukuran mikrokontroler dengan logika 1 (*high*) dan logika 0 (*low*) dengan tegangan yang dihasilkan *power supply* sebesar 5.5 Volt dituliskan dalam Tabel 4.2.

**Tabel 4.2** Hasil Pengukuran Tegangan Setiap *PORT* pada Sistem Minimum ATmega16

PIN	PORT							
	A		B		C		D	
	High (V)	Low (V)	High (V)	Low (V)	High (V)	Low (V)	High (V)	Low (V)
0	5.5	0	5.5	0	5.5	0	5.5	0
1	5.5	0	5.5	0	5.5	0	5.5	0
2	5.5	0	5.5	0	5.5	0	5.5	0
3	5.5	0	5.5	0	5.5	0	5.5	0



PIN	PORT							
	A		B		C		D	
	High (V)	Low (V)	High (V)	Low (V)	High (V)	Low (V)	High (V)	Low (V)
4	5.5	0	5.5	0	5.5	0	5.5	0
5	5.5	0	5.5	0	5.5	0	5.5	0
6	5.5	0	5.5	0	5.5	0	5.5	0
7	5.5	0	5.5	0	5.5	0	5.5	0

Dari hasil pengujian di setiap *PORT* pada Sistem Minimum Mikrokontroler ATmega16, maka didapatkan hasil bahwa pada *PORT A*, *PORT B*, *PORT C*, dan *PORT D* dapat bekerja ketika diberikan logika 1 (high) maupun logika 0 (low).

#### 4.3. Pengujian dan Analisa Modul LCD

Pada pengujian tampilan LCD menggunakan program untuk menampilkan tulisan yang bisa berupa informasi penting ataupun status – status yang ada. Cara pengujian terhadap modul LCD ini adalah dengan cara mencoba menampilkan tulisan pada modul LCD. Pertama kali yang dilakukan adalah memasukkan program pengujian modul LCD ke atmega 16 dengan bahasa pemrograman *codevison avr*.

Program untuk menampilkan tulisan pada LCD adalah dengan meng include kan *lcd.h* pada *codevison avr*. Berikut ini adalah kode inti dari program pengujian modul LCD.

```

#include <mega16.h>
// Alphanumeric LCD Module functions
#include <alcd.h>

void main(void)
{
PORTA=0x00;
DDRA=0xFF;

// Alphanumeric LCD initialization
// Connections specified in the
// Project|Configure|C Compiler|Libraries|Alphanumeric LCD menu:
// R5 - PORTA Bit 0
// RD - PORTA Bit 1
// EN - PORTA Bit 2
// D4 - PORTA Bit 4
// D5 - PORTA Bit 5
// D6 - PORTA Bit 6
// D7 - PORTA Bit 7
// Characters/line: 16
lcd_init(16);
lcd_gotoxy(1,0);
lcd_putsf("#Tracking Cam#");
lcd_gotoxy(4,1);
lcd_putsf("C_TRACK");
}

```

Pada saat program diatas didownload ke Sistem Minimum ATmega16. Berikut tampilan pada Gambar 4.3 dibawah ini:



**Gambar 4.3** Tampilan LCD

Hasil pengujian terlihat bahwa LCD dapat menampilkan kalimat sesuai dengan program yang kita buat. Apabila rangkaian modul LCD belum benar maka tidak akan menampilkan kalimat. Sedangkan, apabila rangkaian modul LCD sudah benar maka akan muncul kalimat sesuai dengan program diatas.

#### 4.4. Pengujian dan Analisa Sistem Komunikasi

Pada pengujian sistem komunikasi serial RS 232, pada dasarnya rangkaian ini terdiri dari IC *Max 232* yaitu IC *serial* yang bertugas meneruskan data serial yang masuk, gambar RS 232 di tunjukkan pada Gambar 4.4.

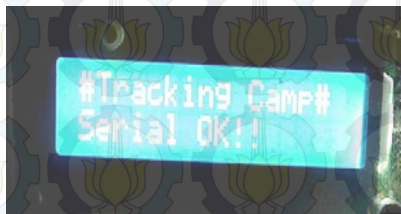


**Gambar 4.4** Rangkaian Serial RS 232

Pengujian pada rangkaian ini dilakukan dengan cara mencoba mengirim data serial dari PC dan menampilkan pada LCD. Berikut ini adalah program yang menjalankan tugas tersebut.

```
// USART initialization
// Communication Parameters: 8 Data, 1 Stop, No Parity
// USART Receiver: On
// USART Transmitter: On
// USART Mode: Asynchronous
// USART Baud Rate: 9600
UCSRA=0x00;
UCSRB=0x18;
UCSRC=0x86;
UBRRH=0x00;
UBRRL=0x47;
lcd_init(16);
lcd_gotoxy(0,0);
lcd_putsf("#Tracking Camp#");
while (1)
{
    for(i=0; i<16; i++)
    {
        data[i]=getchar();
    }
    lcd_gotoxy(0,1);
    lcd_puts(data);
}
```

Apabila pengujian berhasil maka akan muncul tulisan pada modul LCD di tunjukan pada Gambar 4.5 di bawah ini :



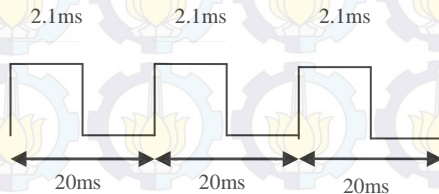
**Gambar 4.5** Tampilan Serial pada LCD

#### 4.5. Pengujian dan Analisa Motor Servo

Pengujian motor servo dilakukan dengan menggunakan program untuk melihat arah putaran dari motor servo. Pengujian awal yang dilakukan adalah membuat sinyal PWM dengan menggunakan program agar motor servo dapat bergerak dengan sudut  $0^\circ$  hingga  $180^\circ$ . Sinyal tersebut yang nantinya akan menentukan arah putaran dari motor servo. Gambar sinyal PWM dapat dilihat seperti pada gambar hasil pengujian.

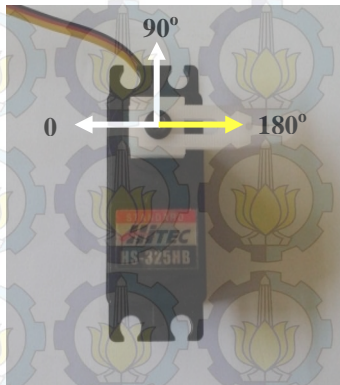
1. Arah putar motor servo dengan sudut  $180^\circ$

Motor servo standart HS-325HB memerlukan sinyal PWM dengan pulsa *high* sebesar 2.1 ms dengan periode 20 ms untuk bergerak ke arah sudut  $180^\circ$ . Tampilan sinyal PWM dapat dilihat pada Gambar 4.6 di bawah ini :



**Gambar 4.6** Sinyal PWM dengan Pulsa *High* 2.1 ms

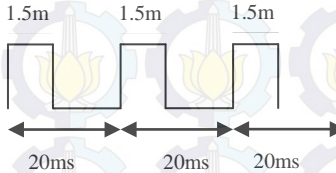
Dengan sinyal PWM seperti gambar 4.6 menghasilkan arah putaran motor servo dengan sudut  $180^\circ$  yang ditampilkan pada Gambar 4.7.



**Gambar 4.7** Arah Putar Motor Servo dengan Sudut  $180^\circ$

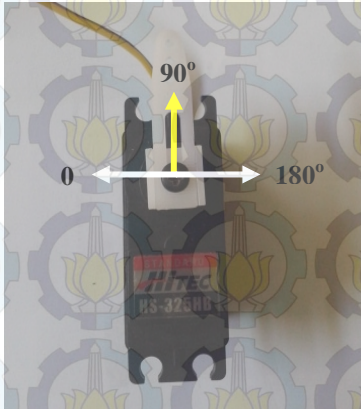
2. Arah putar motor servo dengan sudut  $90^\circ$

Motor servo standart HS-325HB memerlukan sinyal PWM dengan pulsa *high* sebesar 1.5 ms dengan periode 20 ms untuk bergerak ke arah sudut  $0^\circ$ . Tampilan sinyal PWM dapat dilihat pada Gambar 4.8 di bawah ini :



**Gambar 4.8** Sinyal PWM dengan Pulsa *High* 1.5 ms

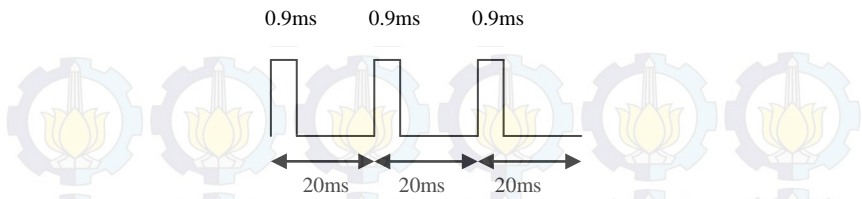
Dengan sinyal PWM seperti gambar 4.8 menghasilkan arah putaran motor servo dengan sudut  $90^\circ$  yang ditampilkan pada Gambar 4.9 di bawah ini :



**Gambar 4.9** Arah Putar Motor Servo dengan Sudut  $90^\circ$

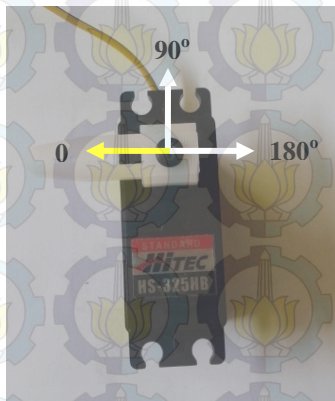
1. Arah putar motor servo dengan sudut  $0^\circ$

Motor servo standart HS-325HB memerlukan sinyal PWM dengan pulsa *high* sebesar 2.1 ms dengan periode 20 ms untuk bergerak ke arah sudut  $180^\circ$ . Tampilan sinyal PWM dapat dilihat pada Gambar 4.10.



**Gambar 4.10** Sinyal PWM dengan Pulsa High 0.9ms

Dengan sinyal PWM seperti gambar 4.10 menghasilkan arah putaran motor servo dengan sudut  $0^\circ$  yang ditampilkan pada Gambar 4.11 di bawah ini :



**Gambar 4.11** Arah Putar Motor Servo dengan Sudut  $0^\circ$

Dan menganalisa program menurut perhitungan dan hasil yang didapat secara aktual:

Jadi untuk mencari sinyal PWM pada motor servo kita harus mengetahui referensi motor servo HS 325HB (pulsa *high* 0,9 - 2,1 dengan periode 20 ms) :

$$fc = n/t$$

$$n = fc \cdot t$$

$$n = 30.000$$

$$t = 1/f = 1/50 \text{ hz}$$

$$n = \text{nilai top}$$

$$fc = \text{frekuensi crystal pada avr}$$

$$\text{duty cycle} = 0,9/20 \times 100 \% = 4,5 \% \quad (0^\circ)$$

$$\text{duty cycle} = 1,5/20 \times 100 \% = 7,5 \% \quad (90^\circ)$$

$$\text{duty cycle} = 2,1/20 \times 100 \% = 10,5 \% \quad (180^\circ)$$

/////////sinyal PWM

$$\text{OCR1x} = \text{duty cycle} \times n = \dots?$$

$$\text{OCR1x} = 4,5 \% \times 30.000 = 1350 \quad (0^\circ)$$

$$\text{OCR1x} = 7,5 \% \times 30.000 = 2250 \quad (90^\circ)$$

$$\text{OCR1x} = 4,5 \% \times 30.000 = 3150 \quad (180^\circ)$$

Terdapat perhitungan dan aktual pada sinyal pwm yang ditunjukkan pada Tabel 4.3 di bawah ini.

**Tabel 4.3** Tabel Sinyal PWM Menurut Perhitungan dan Aktual

Sudut	OCR1x Perhitungan	OCR1x aktual	Selisih
0°	1350	1000	350
90°	2250	2300	50
180°	3150	3400	250

Jadi, dapat disimpulkan bahwa setiap motor servo memiliki spesifikasi berbeda-beda dan sinyal untuk mencari PWM dalam perhitungan maupun aktual juga berbeda.

## 4.6 Pengujian dan Analisa Program

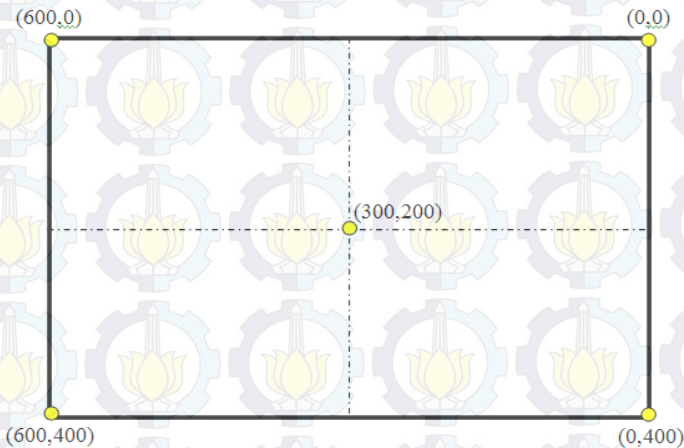
### 4.6.1. Pengujian dan Analisa pada Program PC

Pada pengujian dan analisa pada program PC ini yaitu dengan pemrograman C++ (*OpenCV*), Program yang dimaksud terdiri dari, bagian komunikasi serial dan bagian deteksi wajah.

Bagian komunikasi serial bertugas pada bagian transmisi data. Bagian ini bertugas mulai dari inialisasi, menyiapkan dan mengirim data serial pada program PC. Pada proses mempersiapkan data, program ini sangat bergantung pada perhitungan di program deteksi postur tubuh. Hal itu dikarenakan data yang akan dikirim tersebut hasil keluaran x dan y pada bagian program *open cv* deteksi postur tubuh.

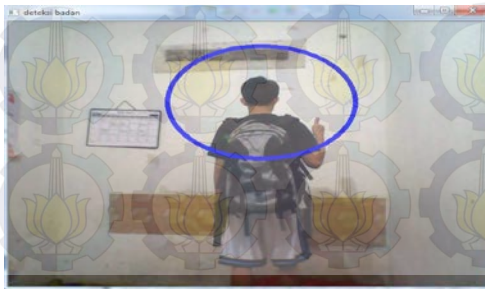
Bagian deteksi wajah tersebut berupa aplikasi konsol, di dalamnya terdapat beberapa perhitungan mengenai *frame* pada

kamera, titik pusat objek dan nilai x dan y ketika terdeteksi objek berupa postur tubuh. Sehingga akan menentukan pada proses pengiriman data ke bagian mikro. Berikut adalah simulasi tampilan *frame* kamera yang digunakan di tunjukkan pada Gambar 4.12 di bawah ini.



**Gambar 4.12** Simulasi Tampilan *Frame*

Dari gambar diatas, *frame* yang digunakan 600 x 400 *pixel* yang mana pada *pixel* tersebut merupakan data apabila ada objek maka akan mengirimkan data x dan y. Misalkan apabila ada objek di titik (150,100) maka data x dan y akan dikirimkan ke mikro dan pada program mikro akan menentukan arah yang telah dihitung. Tampilan *frame* di tunjukkan pada Gambar 4.13 di bawah ini :



**Gambar 4.13** Tampilan *Frame*



Berikut ini adalah *screen shoot* dari aplikasi konsol yang menjelaskan kondisi objek yang terdeteksi, koordinat pusat objek pada *frame* dan komunikasi serial yang terjadi. Maka akan di tampilkan pada konsol yang di tunjukkan pada Gambar 4.14 di bawah ini :

```

koordinat x = 328      frame = 600
koordinat y = 184      frame = 400
send --> data=1      [328184]
koordinat x = 328      frame = 600
koordinat y = 184      frame = 400
send --> data=1      [328184]
koordinat x = 308      frame = 600
koordinat y = 183      frame = 400
send --> data=1      [308183]
koordinat x = 308      frame = 600
koordinat y = 183      frame = 400
send --> data=1      [308183]
koordinat x = 293      frame = 600
koordinat y = 179      frame = 400
send --> data=1      [293179]
koordinat x = 293      frame = 600
koordinat y = 179      frame = 400
send --> data=1      [293179]

```

**Gambar 4.14** Aplikasi Konsol

Dari gambar diatas, terdapat tulisan koordinat x dan koordinat y itu menunjukkan koordinat pusat target pada *frame* (600,400). Untuk tulisan data menunjukan bahwa ada tidaknya objek. Ketika data = 1 berarti ada objek dan jika data = 0 maka tidak terdeteksi objek. Tulisan send menunjukan bahwa data x dan y yang diubah menjadi karakter yang akan di kirimkan ke mikro, misalnya x=293 dan y=179 maka akan mengirim berupa karakter [293179].

#### 4.6.2. Pengujian dan Analisa pada Program Kontroler

Pada pengujian dan analisa program kontroler yang bekerja pada rangkaian mikrokontroler. Dimana program kontroler yang bertugas menerima perintah dari PC dan menjalankan perintah tersebut. Terdapat beberapa fungsi dari program, yaitu membaca data serial pada program PC, mengambil keputusan berdasarkan data serial yang diterima dan mengendalikan peralatan luar.

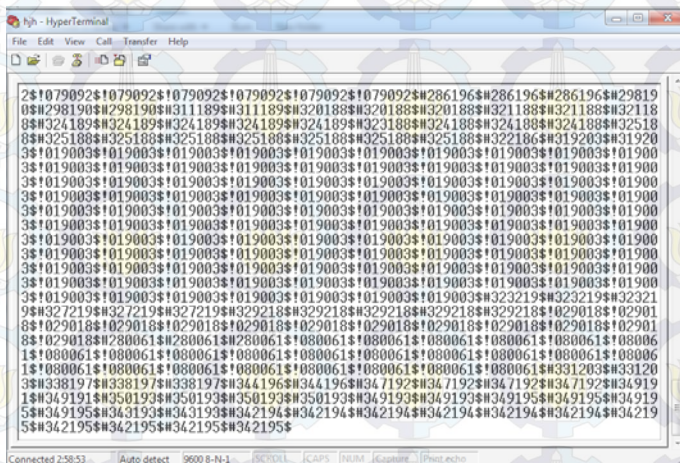
Fungsi pertama adalah membaca data serial. Berikut ini adalah contoh data yang dikirim pada program PC dan diterima oleh program kontroler.

{“#”, “karakter x”, “karakter y”, “\$”}

Karakter pertama dari data tersebut berfungsi sebagai penanda apabila ada objek maka akan mengirim “#” dan karakter berikutnya merupakan data penting dan “\$” data *enter* untuk eksekusi program.

Dan apabila tidak terdeteksi objek maka pengiriman dan penerimaan berupa karakter “!” dan menandakan program kontroler untuk perintah lain. Perintah lain tersebut yaitu perintah untuk melakukan *scanning* / melacak objek.

Berikut tampilan pada percobaan di *hyperterminal* untuk penerimaan data dari program PC dan akan dikirim ke kontroler di tunjukkan pada Gambar 4.15 di bawah ini:



Gambar 4.15 Tampilan pada *Hyperterminal*

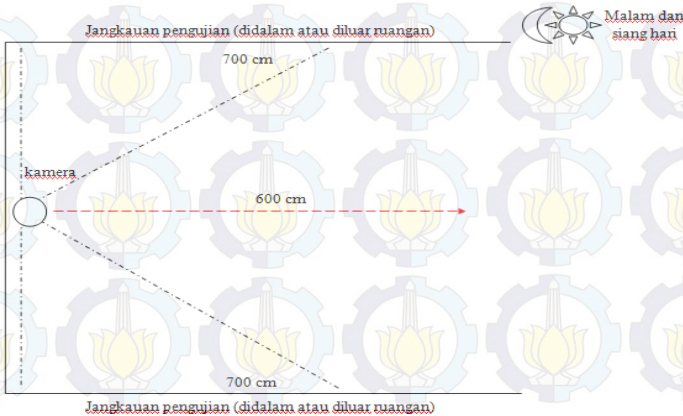
#### 4.7 Pengujian dan Analisa Alat Secara Keseluruhan

Pengujian Keseluruhan ini dilakukan yaitu menganalisa sistem perancangan sistem *tracking camera* yang telah dibuat. Dan seberapa besar pengaruh intensitas cahaya dalam mendeteksi objek dengan *Haarcascade\_upperbody.xml* sehingga dapat diketahui batas maksimal objek yang terdeteksi berupa postur tubuh manusia bagian atas.

Pada pengujian ini diambil 4 keadaan dimana ketika pengujian dilakukan pada malam hari didalam ruangan maupun diluar ruangan. Dan pada kondisi siang hari dan malam hari. Sehingga didapat seberapa besar pengaruh intensitas cahaya pada percobaan yang dilakukan. Hal ini di tunjukkan pada Gambar 4.16 di bawah ini :



**Gambar 4.16** Alat Keseluruhan



**Gambar 4.17** Sistem Pengujian

Penjelasan sistem pengujian :

1. Pada pengujian *tracking* objek dilakukan dengan memperhitungkan jangkauan pengujian, peletakan kamera dan posisi objek yang akan terdeteksi.
2. Pada pengujian ini mendeteksi objek postur tubuh < 600 cm dari kamera dan objek berjalan dengan kecepatan 7 km/jam.

Adapun hasil pengujian dapat dilihat pada tabel berikut:

- I. Pengujian *tracking object (haarcascade\_upperbody.xml)* pada siang hari di dalam ruangan di tunjukkan pada Tabel 4.4 di bawah ini :

**Tabel 4.4** Tabel Pengujian Siang Hari didalam Ruangan

No	Jarak	Hasil		Respon deteksi	
		(1)Menghadap kamera	(2)Membelakangi Kamera	(1)detik	(2)detik
1	600 cm	Terdeteksi	Tidak	02.19	-
2	560 cm	Terdeteksi	Terdeteksi	01.08	01.79
3	520 cm	Terdeteksi	Terdeteksi	02.72	03.45
4	480 cm	Terdeteksi	Terdeteksi	02.35	03.15
5	440 cm	Terdeteksi	Terdeteksi	05.58	02.51
6	400 cm	Terdeteksi	Terdeteksi	02.13	02.44
7	360 cm	Terdeteksi	Terdeteksi	02.43	02.68
8	320 cm	Terdeteksi	Terdeteksi	05.53	02.81
9	280 cm	Terdeteksi	Terdeteksi	01.55	01.82
10	240 cm	Terdeteksi	Terdeteksi	02.48	02.50
11	200 cm	Terdeteksi	Terdeteksi	04.35	02.12
12	160 cm	Terdeteksi	Terdeteksi	02.62	02.28
13	120 cm	Tidak	Terdeteksi	-	04.55
Total & Keberhasilan (%)		92,3 %	92,3 %		

Pada Hasil Pengujian Sistem pada siang hari dengan kondisi di dalam ruangan ini, terlihat bahwa tingkat keberhasilan menghadap kamera mencapai 92,3% dan membelakangi kamera 92,3 %.

II. Pengujian *tracking object (haarcascade\_upperbody.xml)* pada malam hari di dalam ruangan di tunjukkan pada Tabel 4.5 di bawah ini.

**Tabel 4.5** Tabel Pengujian Malam Hari didalam Ruangan

No	Jarak	Hasil		Respon deteksi	
		(1)Menghadap kamera	(2)Membelakangi kamera	(1)detik	(2)detik
1	600 cm	Tidak	Tidak	-	-
2	560 cm	Terdeteksi	Terdeteksi	01.68	02.02
3	520 cm	Terdeteksi	Terdeteksi	03.13	02.15
4	480 cm	Terdeteksi	Terdeteksi	02.09	02.17
5	440 cm	Tidak	Terdeteksi	-	02.54
6	400 cm	Tidak	Terdeteksi	-	02.97
7	360 cm	Tidak	Terdeteksi	-	03.09
8	320 cm	Tidak	Terdeteksi	-	03.15
9	280 cm	Terdeteksi	Terdeteksi	01.55	02.95
10	240 cm	Tidak	Terdeteksi	-	02.10
11	200 cm	Terdeteksi	Terdeteksi	04.35	02.20
12	160 cm	Tidak	Terdeteksi	-	02.00
13	120 cm	Tidak	Terdeteksi	-	02.40
Total & Keberhasilan (%)		38,5 %	92,3 %		

Pada Hasil Pengujian Sistem pada malam hari dengan kondisi di dalam ruangan ini, terlihat bahwa tingkat keberhasilan menghadap kamera mencapai 38,5 % dan membelakangi kamera 92,3 %.

III. Pengujian *tracking object (haarcascade\_upperbody.xml)* pada siang hari di luar ruangan di tunjukkan pada Tabel 4.6 di bawah ini.

**Tabel 4.6** Tabel Pengujian Siang Hari diluar Ruangan

No	Jarak	Hasil		Respon deteksi	
		(1)Menghadap kamera	(2)Membelakangi Kamera	(1)detik	(2)detik
1	600 cm	Terdeteksi	Terdeteksi	02.19	02.50
2	560 cm	Terdeteksi	Terdeteksi	01.08	01.79
3	520 cm	Terdeteksi	Terdeteksi	02.72	03.45
4	480 cm	Terdeteksi	Terdeteksi	02.35	03.15
5	440 cm	Terdeteksi	Terdeteksi	05.58	02.51
6	400 cm	Terdeteksi	Terdeteksi	02.13	02.44
7	360 cm	Terdeteksi	Terdeteksi	02.43	02.68
8	320 cm	Terdeteksi	Terdeteksi	05.53	02.81
9	280 cm	Terdeteksi	Terdeteksi	01.55	01.82
10	240 cm	Terdeteksi	Terdeteksi	02.48	02.50
11	200 cm	Terdeteksi	Terdeteksi	04.35	02.12
12	160 cm	Terdeteksi	Terdeteksi	02.62	02.28
13	120 cm	Terdeteksi	Terdeteksi	02.98	04.55
Total & Keberhasilan (%)		100 %	100 %		

Pada Hasil Pengujian Sistem pada siang hari dengan kondisi di luar ruangan ini, terlihat bahwa tingkat keberhasilan menghadap kamera mencapai 100 % dan membelakangi kamera 100 %.

IV. Pengujian *tracking object (haarcascade\_upperbody.xml)* pada malam hari di luar ruangan di tunjukkan pada Tabel 4.7 di bawah ini.

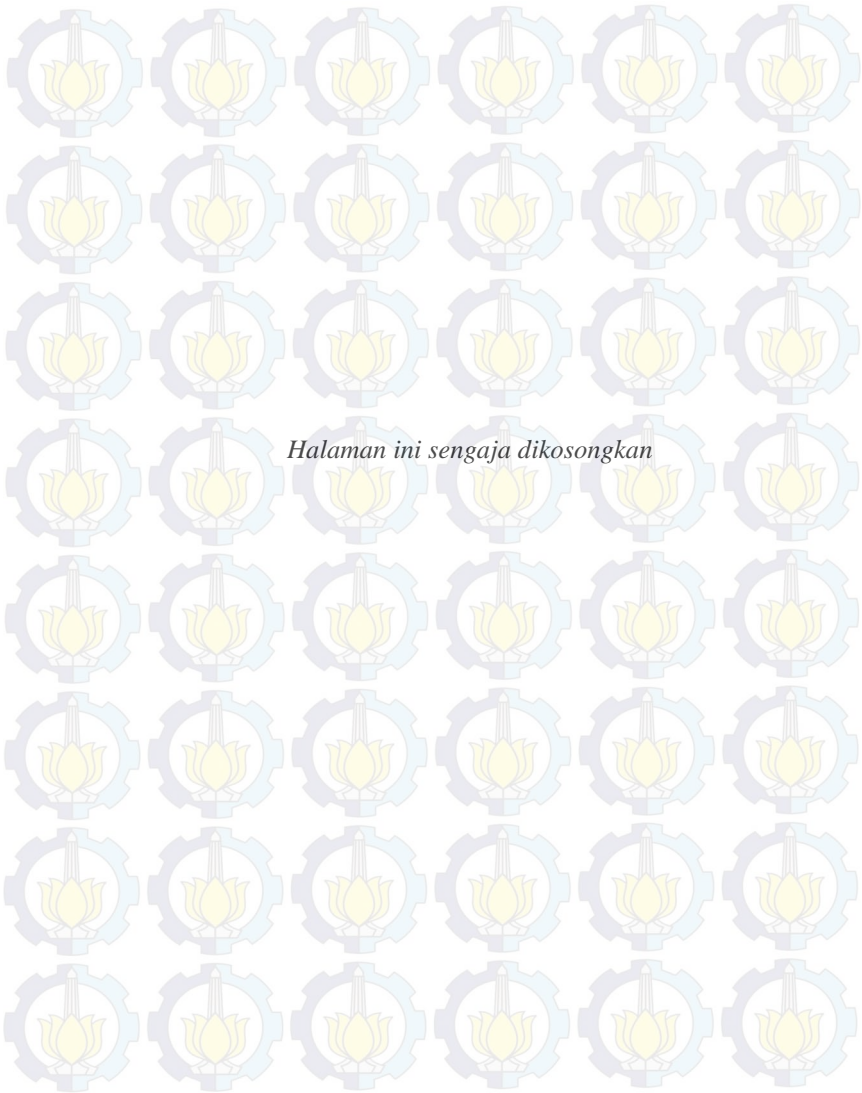
**Tabel 4.7** Tabel Pengujian Malam Hari diluar Ruangan

No	Jarak	Hasil		Respon deteksi	
		(1)Menghadap kamera	(2)Membelakangi Kamera	(1)detik	(2)detik
1	600 cm	Terdeteksi	Terdeteksi	02.16	03.43
2	560 cm	Terdeteksi	Terdeteksi	04.08	02.98
3	520 cm	Terdeteksi	Terdeteksi	04.03	03.20
4	480 cm	Terdeteksi	Terdeteksi	04.56	03.60
5	440 cm	Tidak	Terdeteksi	-	03.27
6	400 cm	Tidak	Terdeteksi	-	03.53
7	360 cm	Terdeteksi	Terdeteksi	04.60	03.40
8	320 cm	Terdeteksi	Terdeteksi	02.62	02.52
9	280 cm	Terdeteksi	Terdeteksi	02.80	02.45
10	240 cm	Tidak	Terdeteksi	-	02.00
11	200 cm	Tidak	Tidak	-	-
12	160 cm	Tidak	Tidak	-	-
13	120 cm	Tidak	Tidak	-	-
Total & Keberhasilan (%)		53,9 %	76,92 %		

Pada Hasil Pengujian Sistem pada malam hari dengan kondisi di luar ruangan ini, terlihat bahwa tingkat keberhasilan menghadap kamera mencapai 100 % dan membelakangi kamera 100 %.

Analisa pengujian pada 4 percobaan diatas yaitu pada siang hari di luar maupun di dalam ruangan dan malam hari di luar maupun di dalam ruangan. Pada proses *tracking camera* ini terlihat bahwa kondisi terbaik adalah pada saat kondisi siang hari di luar ruangan menghadap kamera maupun membelakangi kamera pada saat pengujian berlangsung, hal ini disebabkan oleh karenanya, pencahayaan dari sinar matahari juga stabil, sedangkan pada saat malam hari didalam ruangan menghadap kamera persen keberhasilannya sangat jelek dikarenakan pencahayaan kurang dan sama pada saat malam hari di luar ruangan menghadap kamera. Hal ini dikarenakan, pencahayaan yang kurang, sehingga dimensi objek yang dimaksud kan di dalam program, tidak dapat didefinisikan dengan baik.





## **BAB V**

### **PENUTUP**

Bab penutup ini berisi tentang kesimpulan-kesimpulan yang diperoleh selama pembuatan Tugas Akhir ini dan saran-saran untuk perbaikan dan pengembangannya.

#### **5.1. Kesimpulan**

Hasil dari pengujian serta analisa data dari Perancangan Sistem *Tracking Camera* Pengawas Berdasarkan Evaluasi Data Gambar, dapat diambil kesimpulan sebagai berikut:

1. Sistem ini bisa mendeteksi postur tubuh bagian atas seperti, dengan beberapa postur badan menghadap dan berbalik badan.
2. Penentuan posisi webcam dengan objek dan pencahayaan memiliki pengaruh besar dalam pengenalan objek, untuk itu dibutuhkan pengaturan kamera dan pencahayaan yang cukup.
3. Dari sistem ini dapat diketahui kecepatan minimum ketika objek terdeteksi berjalan biasa dengan kecepatan 5km/jam , jika pada objek berjalan melebihi kecepatan 8km/jam tidak dapat mendeteksi objek.
4. Pada analisa objek berlari di dapatkan tidak dapat terdeteksi dikarenakan melebihi kecepatan minimum bahkan kamera yang kita gunakan respon pada objek bergerak cepat sangat kurang.
5. Penggunaan kamera webcam tidak cocok untuk kamera tracking seharusnya dalam *system tracking* menggunakan kamera dengan spesifikasi yang lebih bagus
6. Hasil analisa didapatkan hasil ruangan diluar dengan intensitas cahay lebih banyak mendapatkan hasil deteksi lebih fokus sedangkan di dalam ruangan dengan intensitas cahaya lebih sedikit di dapatkan tidak focus pada saat deteksi gambar
7. Dari hasil pengujian keberhasilan dalam mendeteksi postur tubuh bagian atas terlihat sistem ini mampu mendeteksi dengan jarak maksimal 6 meter.
8. Servo dapat bergerak ke arah objek sesuai perintah yang diberikan pada program *opencv*.

## 5.2. Saran

Untuk pengembangan dan penyempurnaan dari Perancangan Sistem *Tracking Camera* Pengawas Berdasarkan Evaluasi Data Gambar ini, maka diberikan beberapa saran sebagai berikut :

1. Untuk pengembangan Perancangan Sistem *Tracking Camera* Pengawas Berdasarkan Evaluasi Data Gambar. Dengan membuat *image processing*/ deteksi tubuh dengan respon cepat. Sehingga respon mengikuti objek yang terdeteksi lebih stabil.
2. Dalam penggunaan kamera tracking di sarankan untuk menggunakan kamera scanning atau kamera lebih bagus.
3. Dalam sistem ini di sarankan dengan penggunaan 2 kamera yakni 1 kamera berfungsi sebagai pengawas pada tracking dan 1 kamera sebagai pengolah deteksi gambar.

## DAFTAR PUSTAKA

- [1] Gupta, K., Kulkarni, A. V., *“Implementation of an Automated Single Camera Object Tracking System Using Frame Differencing and Dynamic Template Matching”*, Indian Institute of Technology, Kanpur, India, 2008.
- [2] Ahmad, Usman, *Pengolahan Citra Digital dan Teknik Pemrogramannya*, Pemrogramannya. Yogyakarta: Graha Ilmu, 2005.
- [3] Paul Viola and Michael J. Jones, *Rapid Object Detection using a Boosted cascade of Simple Features*, *IEEE CVPR*, 2001.
- [5] Arihutomo, Mukhlis, *Rancang Bangun Sistem Penjejakan Objek Menggunakan Metode Viola Jones Untuk Aplikasi Eyebot*, Tugas akhir, Surabaya : PENS, 2010.
- [6] Martono, *Rancang Bangun Robot Penggambar Berdasarkan Visualisasi Kamera*, Tugas akhir, Surabaya : Institut Teknologi Sepuluh Nopember, 2008.
- [7] Andrianto, Heri, *Pemrograman Mikrokontroler AVR Atmega16 Menggunakan Bahasa (Code Vision AVR)*, Informatika, Bandung, 2010.
- [8] Winoto, Ardi, *Mikrokontroler AVR ATmega8/32/16/8535 dan Pemrogramannya dengan Bahasa C pada WinAVR*, Informatika, Bandung, 2008.
- [9] ..., *Hitech HS-325HB - Standard Ball Bearing Servo*, <http://www.servodatabase.com/servo/hitec/hs-325hb>, Diambil pada 25 Juni 2014.



*Halaman ini sengaja dikosongkan*

## LAMPIRAN I Program pada AVR

### *LISTING PROGRAM*

/\*\*\*\*\*\*  
/\*\*\*\*\*

This program was produced by the  
CodeWizardAVR V2.05.3 Standard

Automatic Program Generator

© Copyright 1998-2011 Pavel Haiduc, HP InfoTech s.r.l.

<http://www.hpinfotech.com>

Project :

Version :

Date : 6/2/2014

Author : Pujangga Tugas Akhir

Company :

Comments:

Chip type : ATmega16

Program type : Application

AVR Core Clock frequency: 12.000000 MHz

Memory model : Small

External RAM size : 0

Data Stack size : 256

\*\*\*\*\*  
/

```
#include <mega16.h>
```

```
#include <delay.h>
```

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
// Alphanumeric LCD functions
```

```
#include <alcd.h>
```

```
#define nilaim90x 2600
```

```
#define nilaim90y 2600
```

```
//
```

```
#define deltaocrx 1000
```

```
#define deltaocry 1200
```

```
#define framey 400.0 //satuan pixel
```

```
#define framex 600.0
```

```
#define x OCR1A //PORTRD.5
```

```
#define y OCR1B //PORTRD.4
```

```
int i=0,j=300,error_x=300,error_y=200;
```

```
float dy,dx;
```

```
char data[100];
```

```
unsigned int datax,datay://,datayh,dataxh,;
```

```
unsigned int rx,px,sx,ry,py,sy;
```

```
unsigned char par[7];;
```

```
#ifndef RXB8
```

```
#define RXB8 1
```

```
#endif
```

```
#ifndef TXB8
```

```
#define TXB8 0
```

```
#endif
```

```
#ifndef UPE
```

```
#define UPE 2
```

```
#endif
```

```
#ifndef DOR
```

```
#define DOR 3
```

```
#endif
```

```
#ifndef FE
```

```
#define FE 4
```

```
#endif
```

```
#ifndef UDRE
```

```

#define UDRE 5
#endif

#ifndef RXC
#define RXC 7
#endif

#define FRAMING_ERROR (1<<FE)
#define PARITY_ERROR (1<<UPE)
#define DATA_OVERRUN (1<<DOR)
#define DATA_REGISTER_EMPTY (1<<UDRE)
#define RX_COMPLETE (1<<RXC)

// USART Receiver buffer
#define RX_BUFFER_SIZE 8
char rx_buffer[RX_BUFFER_SIZE];

#if RX_BUFFER_SIZE <= 256
unsigned char rx_wr_index,rx_rd_index,rx_counter;
#else
unsigned int rx_wr_index,rx_rd_index,rx_counter;
#endif

// This flag is set on USART Receiver buffer overflow
bit rx_buffer_overflow;

// USART Receiver interrupt service routine
interrupt [USART_RXC] void usart_rx_isr(void)
{
char status,data;
status=UCSRA;
data=UDR;
if(data!='')
{
rx_wr_index = 0;
for(i=0;i<=7;i++)
rx_buffer[i]=0;
}
}

```



```

    if ((status & (FRAMING_ERROR | PARITY_ERROR |
DATA_OVERRUN))==0)
    {
        rx_buffer[rx_wr_index++]=data;
        #if RX_BUFFER_SIZE == 256
        // special case for receiver buffer size=256
        if (++rx_counter == 0) rx_buffer_overflow=1;
        #else
        if (rx_wr_index == RX_BUFFER_SIZE) rx_wr_index=0;
        if (++rx_counter == RX_BUFFER_SIZE)
        {
            rx_counter=0;
            rx_buffer_overflow=1;
        }
        #endif
    }
}

#ifndef _DEBUG_TERMINAL_IO_
// Get a character from the USART Receiver buffer
#define _ALTERNATE_GETCHAR_
#pragma used+
char getchar(void)
{
    char data;
    while (rx_counter==0);
    data=rx_buffer[rx_rd_index++];
    #if RX_BUFFER_SIZE != 256
    if (rx_rd_index == RX_BUFFER_SIZE) rx_rd_index=0;
    #endif
    #asm("cli")
    --rx_counter;
    #asm("sei")
    return data;
}
#pragma used-
#endif

```

```

// Standard Input/Output functions
#include <stdio.h>

// Declare your global variables here
void hitung()
{
    dx=nilaim90x-(error_x/framex*deltaocrx);
    dy=nilaim90y-(error_y/framex*deltaocry);
    delay_us(100);
    x=dx;
    y=dy;

    delay_ms(10);
}
void tampil()
{
    lcd_gotoxy(0,0);
    sprintf(data,"x=%d ",error_x);
    lcd_puts(data);
    lcd_gotoxy(0,1);
    sprintf(data,"y=%d ",error_y);
    lcd_puts(data);
}
void parsing()
{
    if(rx_buffer[0]=='#')
    {
        for(i=1;i<=6;i++)
        {
            if(rx_buffer[i]=='0')
                par[i]=0;
            else
                if(rx_buffer[i]=='1')
                    par[i]=1;
                else
                    if(rx_buffer[i]=='2')
                        par[i]=2;
        }
    }
}

```

```

else
if(rx_buffer[i]=='3')
par[i]=3;
else
if(rx_buffer[i]=='4')
par[i]=4;
else
if(rx_buffer[i]=='5')
par[i]=5;
else
if(rx_buffer[i]=='6')
par[i]=6;
else
if(rx_buffer[i]=='7')
par[i]=7;
else
if(rx_buffer[i]=='8')
par[i]=8;
else
if(rx_buffer[i]=='9')
par[i]=9;
}
rx=100*par[1];
px=10*par[2];
sx=par[3];
ry=100*par[4];
py=10*par[5];
sy=par[6];

datax=rx+px+sx;
datay=ry+py+sy;

lcd_gotoxy(10,0);
sprintf(data,"X=%d ",datax);
lcd_puts(data);
lcd_gotoxy(10,1);
sprintf(data,"Y=%d ",datay);
lcd_puts(data);

```

```

if(rx_buffer[7]=='$')
{
    lcd_clear();
    error_x=datax;
    error_y=datay;
    lcd_gotoxy(0,0);
    sprintf(data,"x=%d",error_x);
    lcd_puts(data);
    lcd_gotoxy(0,1);
    sprintf(data,"y=%d",error_y);
    lcd_puts(data);
    hitung();
}
}
else
if(rx_buffer[0]=='!'||rx_buffer[7]=='!')
{
    while(1)
    {
        error_y=200;
        lcd_clear();
        for (i=j ; i<=600 ; i++) //CEK JALAN
        {
            if(rx_buffer[0]=='#'||rx_buffer[7]=='$')
                parsing();
            else
            {
                error_x=i;
                tampil();
                hitung();
            }
        }
        for (i=600 ; i>=0 ; i--) //CEK JALAN
        {
            if(rx_buffer[0]=='#'||rx_buffer[7]=='$')
                parsing();
            else
            {
                error_x=i;
                tampil();
                hitung();
            }
        }
        j=0;
    }
}
}

```

```
    }  
  }  
}  
void track()  
{  
  for (i=j ; i<=600 ; i++) //CEK JALAN  
  { if(rx_buffer[0]=='#'||rx_buffer[7]=='$')  
    parsing();  
    else  
    error_x=i;  
    tampil();  
    hitung();  
  }  
  for (i=600 ; i>=0 ; i--) //CEK JALAN  
  {  
    if(rx_buffer[0]=='#'||rx_buffer[7]=='$')  
    parsing();  
    else  
    error_x=i;  
    tampil();  
    hitung();  
    j=0;  
  }  
}
```

```
void main(void)  
{  
  PORTA=0x00;  
  DDRA=0x00;  
  PORTB=0x00;  
  DDRB=0x00;  
  PORTC=0x00;  
  DDRC=0x00;  
  PORTD=0x00;
```

DDRD=0x30;

// Timer/Counter 0 initialization

TCCR0=0x00;

TCNT0=0x00;

OOCR0=0x00;

// Timer/Counter 1 initialization

// Clock source: System Clock

// Clock value: 1500.000 kHz

// Mode: Fast PWM top=ICR1

// OC1A output: Non-Inv.

// OC1B output: Non-Inv.

// Noise Canceler: Off

// Input Capture on Falling Edge

// Timer1 Overflow Interrupt: Off

// Input Capture Interrupt: Off

// Compare A Match Interrupt: Off

// Compare B Match Interrupt: Off

TCCR1A=0xA2;

TCCR1B=0x1A;

TCNT1H=0x00;

TCNT1L=0x00;

ICR1H=0x75;

ICR1L=0x2F;

// Timer/Counter 2 initialization

ASSR=0x00;

TCCR2=0x00;

TCNT2=0x00;

OOCR2=0x00;

// External Interrupt(s) initialization

// INT0: Off

// INT1: Off

// INT2: Off

MCUCR=0x00;

MCUCSR=0x00;

```
// Timer(s)/Counter(s) Interrupt(s) initialization
TIMSK=0x00;

// USART initialization
// Communication Parameters: 8 Data, 1 Stop, No Parity
// USART Receiver: On
// USART Transmitter: On
// USART Mode: Asynchronous
// USART Baud Rate: 9600
UCSRA=0x00;
UCSRB=0x98;
UCSRC=0x86;
UBRRH=0x00;
UBRRL=0x4D;

// Analog Comparator initialization
// Analog Comparator: Off
// Analog Comparator Input Capture by Timer/Counter 1: Off
ACSR=0x80;
SFIOR=0x00;

// ADC initialization
// ADC disabled
ADCSRA=0x00;

// SPI initialization
// SPI disabled
SPCR=0x00;

// TWI initialization
// TWI disabled
TWCR=0x00;

// Alphanumeric LCD initialization
// Connections are specified in the
// Project|Configure|C Compiler|Libraries|Alphanumeric LCD menu:
// RS - PORTC Bit 7
// RD - PORTC Bit 6
// EN - PORTC Bit 5
```

```
// D4 - PORTC Bit 3
// D5 - PORTC Bit 2
// D6 - PORTC Bit 1
// D7 - PORTC Bit 0
// Characters/line: 16
lcd_init(16);
```

```
// Global enable interrupts
#asm("sei")
```

```
while (1)
```

```
{
```

```
    track();
```

```
}
```

```
}
```



### Program pada *Visual Studio C++ OpenCV*

```
//Perancangan sistem tracking camera pengawas
berdasarkan evaluasi data gambar
[Final Project] by Welly dan Firman
//debuging >> command argument = --cascade="C:/Program
Files/OpenCV/data/haarcascades/upperbody.xml"
//D3 Teknik Elektro ITS - 2011 [bismillah = Wisuda
110]*amin - [=====C_TRACK=====]-
//=====
=====
#include "cvaux.h"
#include "highgui.h"
#include "cv.h"
#include "SerialPort.h"
#include <stdio.h>
#include <string.h>
#include <iostream>
#include <stdlib.h>
#include <time.h>
#include <windows.h>
//=====
=====
using namespace std;
using namespace cv;
//=====
=====
/// Fungsi void deteksi dan display ( Mat frame ) ///
String deteksibadanload = "haarcascade_upperbody.xml";
CascadeClassifier deteksibadan;
string windowload = "deteksi badan";
string window = " thresholded";

/// Deklarasi serial port ///
CSerialPort::CSerialPort(void)
{
}

CSerialPort::~CSerialPort(void)
{
}
```

```

}

/// Setup a serial COM port connection ///
void CSerialPort::OpenSerial( void ) {
    /// Deklarasi serial port ///
    SerialPort = CreateFile(TEXT("\\\\.\\COM1"),
        GENERIC_WRITE, NULL, NULL, OPEN_EXISTING, NULL, NULL);
    /// Inisialiasasi port serial ///
    if (SerialPort==INVALID_HANDLE_VALUE) {
        Error=GetLastError();
        printf("\nError opening serial
port!\nError code: %i\n\n", Error);
        system("PAUSE");
    }
    else
        printf("\nSerial port was successfully
established!\n\n");
    /// Verify port settings ///
    DCB dcbConfig;
    if(GetCommState(SerialPort, &dcbConfig)) {
        dcbConfig.BaudRate = 9600;
        dcbConfig.ByteSize = 8;
        dcbConfig.Parity = NOPARITY;
        dcbConfig.StopBits = ONESTOPBIT;
        dcbConfig.fBinary = TRUE;
        dcbConfig.fParity = TRUE;
        SetCommState(SerialPort, &dcbConfig);
    }
}

/// Send data out the COM port ///
void CSerialPort::Send( char *data ) {
    if( WriteFile( SerialPort, (LPVOID)data, 1,
        &dwNumberOfBytesWritten, NULL) == 0 ) {

```

```

        Error=GetLastError();
        printf("\nWrite error!\nError code:
%i\n\n", Error);
    }
}

void CSerialPort::Send1( int data ) {
    if( WriteFile( SerialPort, (LPVOID)data, 1,
&dwNumberOfBytesWritten, NULL) == 0 ) {
        Error=GetLastError();
        printf("\nWrite error!\nError code:
%i\n\n", Error);
    }
}

void CSerialPort::CloseSerial( void ) {
    if(SerialPort != INVALID_HANDLE_VALUE)
        CloseHandle(SerialPort);
}
//=====
// Fungsi utama //
int main( int argc, const char** argv )
{
    CvCapture* capture; /// capture video dari webcam
    ///
    Mat frame;
    CSerialPort serial;
    serial.OpenSerial();
    /// MeLoad HaarClassifierCascade ///
    if( !deteksibadan.load( deteksibadanload ) ){
    printf("--(!)Error loading\n"); return -1; };
    capture = cvCaptureFromCAM( 0 );
    if( capture )
    {
        while( true )
        {

```

```

frame = cvQueryFrame( capture );
if( !frame.empty() )
    { std::vector<Rect> deteksi;
    Mat frame_gray;
    int x, y, dataxy1, dataxy2, dataxy3, dataxy4,
    dataxy5, dataxy6, S;

    /// Mendeteksi obyek dan menyimpannya dalam urutan ///
    cvtColor( frame, frame_gray, CV_BGR2GRAY );
    equalizeHist( frame_gray, frame_gray );
    deteksi badan.detectMultiScale( frame_gray, deteksi,
    1.1 , 1, 0|CV_HAAR_SCALE_IMAGE, Size(100,100) );
    //=====
    =====
    //=====
    =====
    /// Looping deteksi translasi badan dengan (Metode
    viola jones) ///
    for( size_t i = 0; i < deteksi.size(); i++ )
    {
        Point center( deteksi[i].x + deteksi[i].width*0.5,
        deteksi[i].y + deteksi[i].height*0.5 );
        ellipse( frame, center, Size( deteksi[i].width*0.5,
        deteksi[i].height*0.5), 0, 0, 360,
        Scalar( 255, 0, 0 ), 4, 8, 0 );
        x = deteksi[i].x;
        y = deteksi[i].y;
    }

    /// keluaran data pada aplikasi konsol ///
    cout << " koordinat x = " << x << " " << " " << " "
    << "frame = 600" << endl;
    cout << " koordinat y = " << y << " " << " " << " "
    << "frame = 400" << endl;
    //=====
    =====
    //=====
    =====

```

```

/// deklarasi pengiriman serial ke kontroler ///
S = deteksi.size();
if(S>0)
serial.Send("#");
else
serial.Send("!");

/// pengiriman data x ///
dataxy1=(int)x/100;
if (dataxy1==0) serial.Send("0");
if (dataxy1==1) serial.Send("1");
if (dataxy1==2) serial.Send("2");
if (dataxy1==3) serial.Send("3");
if (dataxy1==4) serial.Send("4");
if (dataxy1==5) serial.Send("5");
if (dataxy1==6) serial.Send("6");
if (dataxy1==7) serial.Send("7");
if (dataxy1==8) serial.Send("8");
if (dataxy1==9) serial.Send("9");

x=(int)x-100*dataxy1;
dataxy2=(int)x/10;
if (dataxy2==0) serial.Send("0");
if (dataxy2==1) serial.Send("1");
if (dataxy2==2) serial.Send("2");
if (dataxy2==3) serial.Send("3");
if (dataxy2==4) serial.Send("4");
if (dataxy2==5) serial.Send("5");
if (dataxy2==6) serial.Send("6");
if (dataxy2==7) serial.Send("7");
if (dataxy2==8) serial.Send("8");
if (dataxy2==9) serial.Send("9");

dataxy3=(int)x - dataxy2*10;
if (dataxy3==0) serial.Send("0");
if (dataxy3==1) serial.Send("1");
if (dataxy3==2) serial.Send("2");
if (dataxy3==3) serial.Send("3");
if (dataxy3==4) serial.Send("4");
if (dataxy3==5) serial.Send("5");

```

```
    if (dataxy3==6) serial.Send("6");
    if (dataxy3==7) serial.Send("7");
    if (dataxy3==8) serial.Send("8");
    if (dataxy3==9) serial.Send("9");

/// Pengiriman data y ///
    dataxy4=(int)y/100;
    if (dataxy4==0) serial.Send("0");
    if (dataxy4==1) serial.Send("1");
    if (dataxy4==2) serial.Send("2");
    if (dataxy4==3) serial.Send("3");
    if (dataxy4==4) serial.Send("4");
    if (dataxy4==5) serial.Send("5");
    if (dataxy4==6) serial.Send("6");
    if (dataxy4==7) serial.Send("7");
    if (dataxy4==8) serial.Send("8");
    if (dataxy4==9) serial.Send("9");

    y=(int)y-100*dataxy4;
    dataxy5=(int)y/10;
    if (dataxy5==0) serial.Send("0");
    if (dataxy5==1) serial.Send("1");
    if (dataxy5==2) serial.Send("2");
    if (dataxy5==3) serial.Send("3");
    if (dataxy5==4) serial.Send("4");
    if (dataxy5==5) serial.Send("5");
    if (dataxy5==6) serial.Send("6");
    if (dataxy5==7) serial.Send("7");
    if (dataxy5==8) serial.Send("8");
    if (dataxy5==9) serial.Send("9");

    dataxy6=(int)y - dataxy5*10;
    if (dataxy6==0) serial.Send("0");
    if (dataxy6==1) serial.Send("1");
    if (dataxy6==2) serial.Send("2");
    if (dataxy6==3) serial.Send("3");
    if (dataxy6==4) serial.Send("4");
    if (dataxy6==5) serial.Send("5");
    if (dataxy6==6) serial.Send("6");
    if (dataxy6==7) serial.Send("7");
```

```

        if (dataxy6==8) serial.Send("8");
        if (dataxy6==9) serial.Send("9");

        serial.Send("$");
        cout << " " << "send --> data=" <<
deteksi.size() << " " << "[" << dataxy1 << dataxy2 <<
dataxy3 << dataxy4 << dataxy5 << dataxy6 << "]" <<endl;
//=====
//=====
//=====
        imshow( windowload, frame );
        imshow (window , frame_gray); }
        else
        { printf(" --(!) No captured frame -- Break!");
break; }

        int c = waitKey(10);
        if( (char)c == 'c' ) { break; }
    }
    serial.CloseSerial();
    return 0;
}
//=====
//=====

```

# LAMPIRAN II

## 1. Datasheet Atmega16

### Features

- High-performance, Low-power AVR® 8-bit Microcontroller
- Advanced RISC Architecture
  - 131 Powerful Instructions – Most Single-clock Cycle Execution
  - 32 x 8 General Purpose Working Registers
  - Fully Static Operation
  - Up to 16 MIPS Throughput at 16 MHz
  - On-chip 2-cycle Multiplier
- High Endurance Non-volatile Memory segments
  - 16K Bytes of In-System Self-programmable Flash program memory
  - 512 Bytes EEPROM
  - 1K Byte Internal SRAM
  - Write/Erase Cycles: 10,000 Flash/100,000 EEPROM
  - Data retention: 20 years at 85°C/100 years at 25°C<sup>(1)</sup>
  - Optional Boot Code Section with Independent Lock Bits
    - In-System Programming by On-chip Boot Program
    - True Read-While-Write Operation
  - Programming Lock for Software Security
- JTAG (IEEE std. 1149.1 Compliant) Interface
  - Boundary-scan Capabilities According to the JTAG Standard
  - Extensive On-chip Debug Support
  - Programming of Flash, EEPROM, Fuses, and Lock Bits through the JTAG Interface
- Peripheral Features
  - Two 8-bit Timer/Counters with Separate Prescalers and Compare Modes
  - One 16-bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture Mode
  - Real Time Counter with Separate Oscillator
  - Four PWM Channels
  - 8-channel, 10-bit ADC
    - 8 Single-ended Channels
    - 7 Differential Channels in TQFP Package Only
    - 2 Differential Channels with Programmable Gain at 1x, 10x, or 200x
  - Byte-oriented Two-wire Serial Interface
  - Programmable Serial USART
  - Master/Slave SPI Serial Interface
  - Programmable Watchdog Timer with Separate On-chip Oscillator
  - On-chip Analog Comparator
- Special Microcontroller Features
  - Power-on Reset and Programmable Brown-out Detection
  - Internal Calibrated RC Oscillator
  - External and Internal Interrupt Sources
  - Six Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, Standby and Extended Standby
- I/O and Packages
  - 32 Programmable I/O Lines
  - 40-pin PDIP, 44-lead TQFP, and 44-pad QFN/MLF
- Operating Voltages
  - 2.7 - 5.5V for ATmega16A
- Speed Grades
  - 0 - 16 MHz for ATmega16A
- Power Consumption @ 1 MHz, 3V, and 25°C for ATmega16A
  - Active: 0.6 mA
  - Idle Mode: 0.2 mA
  - Power-down Mode: < 1µA



8-bit AVR<sup>®</sup>  
Microcontroller  
with 16K Bytes  
In-System  
Programmable  
Flash

ATmega16A

8154B-AVR-07/09

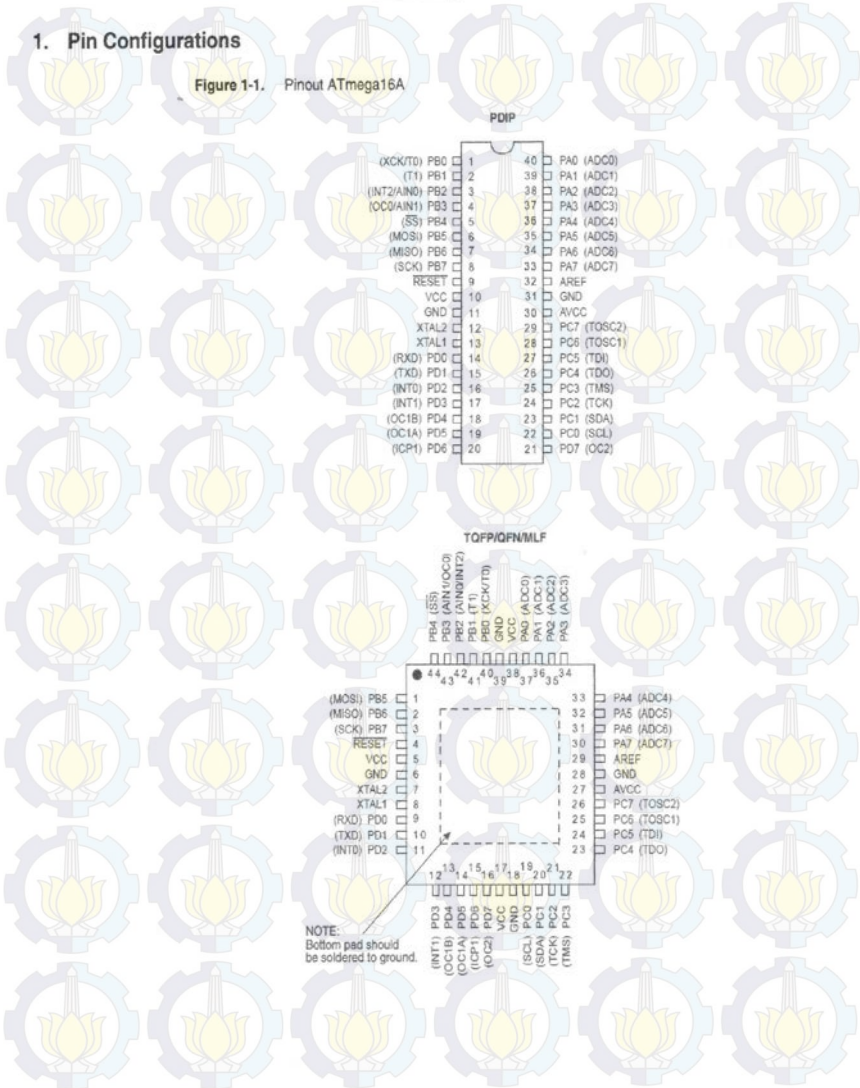






# 1. Pin Configurations

Figure 1-1. Pinout ATmega16A



## 2. Overview

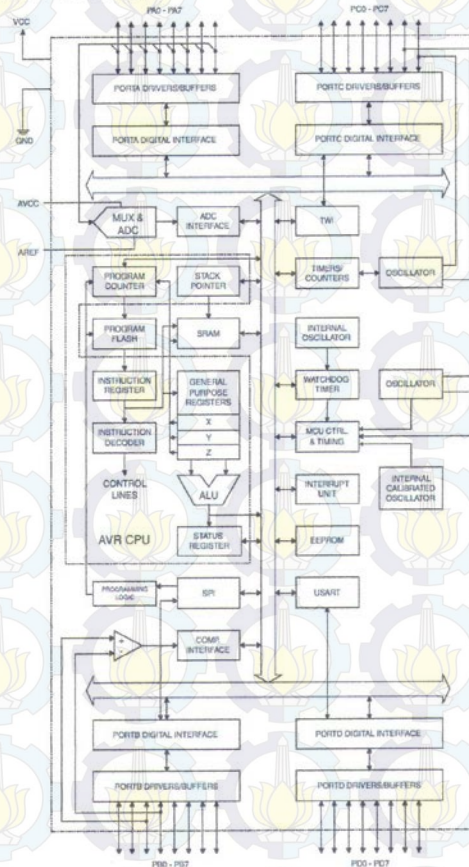
The ATmega16A is a low-power CMOS 8-bit microcontroller based on the AVR enhanced RISC architecture. By executing powerful instructions in a single clock cycle, the ATmega16A achieves throughputs approaching 1 MIPS per MHz allowing the system designer to optimize power consumption versus processing speed.





## 2.1 Block Diagram

Figure 2-1. Block Diagram



The AVR core combines a rich instruction set with 32 general purpose working registers. All the 32 registers are directly connected to the Arithmetic Logic Unit (ALU), allowing two independent registers to be accessed in one single instruction executed in one clock cycle. The resulting architecture is more code efficient while achieving throughputs up to ten times faster than conventional CISC microcontrollers.

The ATmega16A provides the following features: 16K bytes of In-System Programmable Flash Program memory with Read-While-Write capabilities, 512 bytes EEPROM, 1K byte SRAM, 32 general purpose I/O lines, 32 general purpose working registers, a JTAG interface for Boundary-scan, On-chip Debugging support and programming, three flexible Timer/Counters with compare modes, Internal and External Interrupts, a serial programmable USART, a byte oriented Two-wire Serial Interface, an 8-channel, 10-bit ADC with optional differential input stage with programmable gain (TQFP package only), a programmable Watchdog Timer with Internal Oscillator, an SPI serial port, and six software selectable power saving modes. The Idle mode stops the CPU while allowing the USART, Two-wire interface, A/D Converter, SRAM, Timer/Counters, SPI port, and interrupt system to continue functioning. The Power-down mode saves the register contents but freezes the Oscillator, disabling all other chip functions until the next External Interrupt or Hardware Reset. In Power-save mode, the Asynchronous Timer continues to run, allowing the user to maintain a timer base while the rest of the device is sleeping. The ADC Noise Reduction mode stops the CPU and all I/O modules except Asynchronous Timer and ADC, to minimize switching noise during ADC conversions. In Standby mode, the crystal/resonator Oscillator is running while the rest of the device is sleeping. This allows very fast start-up combined with low-power consumption. In Extended Standby mode, both the main Oscillator and the Asynchronous Timer continue to run.

The device is manufactured using Atmel's high density nonvolatile memory technology. The On-chip ISP Flash allows the program memory to be reprogrammed in-system through an SPI serial interface, by a conventional nonvolatile memory programmer, or by an On-chip Boot program running on the AVR core. The boot program can use any interface to download the application program in the Application Flash memory. Software in the Boot Flash section will continue to run while the Application Flash section is updated, providing true Read-While-Write operation. By combining an 8-bit RISC CPU with In-System Self-Programmable Flash on a monolithic chip, the Atmel ATmega16A is a powerful microcontroller that provides a highly-flexible and cost-effective solution to many embedded control applications.

The ATmega16A AVR is supported with a full suite of program and system development tools including: C compilers, macro assemblers, program debugger/simulators, in-circuit emulators, and evaluation kits.

## 2.2 Pin Descriptions

### 2.2.1 VCC

Digital supply voltage.

### 2.2.2 GND

Ground.

### 2.2.3 Port A (PA7:PA0)

Port A serves as the analog inputs to the A/D Converter.

Port A also serves as an 8-bit bi-directional I/O port, if the A/D Converter is not used. Port pins can provide internal pull-up resistors (selected for each bit). The Port A output buffers have symmetrical drive characteristics with both high sink and source capability. When pins PA0 to PA7 are used as inputs and are externally pulled low, they will source current if the internal pull-up resistors are activated. The Port A pins are tri-stated when a reset condition becomes active, even if the clock is not running.

### 2.2.4 Port B (PB7:PB0)

Port B is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port B output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port B pins that are externally pulled low will source current if the pull-up resistors are activated. The Port B pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Port B also serves the functions of various special features of the ATmega16A as listed on page 57.

### 2.2.5 Port C (PC7:PC0)

Port C is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port C output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port C pins that are externally pulled low will source current if the pull-up resistors are activated. The Port C pins are tri-stated when a reset condition becomes active, even if the clock is not running. If the JTAG interface is enabled, the pull-up resistors on pins PC5(TDI), PC3(TMS) and PC2(TCK) will be activated even if a reset occurs.

Port C also serves the functions of the JTAG interface and other special features of the ATmega16A as listed on page 60.

### 2.2.6 Port D (PD7:PD0)

Port D is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port D output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port D pins that are externally pulled low will source current if the pull-up resistors are activated. The Port D pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Port D also serves the functions of various special features of the ATmega16A as listed on page 62.

**2.2.7 RESET**

Reset input. A low level on this pin for longer than the minimum pulse length will generate a reset, even if the clock is not running. The minimum pulse length is given in Table 27-2 on page 296. Shorter pulses are not guaranteed to generate a reset.

**2.2.8 XTAL1**

Input to the inverting Oscillator amplifier and input to the internal clock operating circuit.

**2.2.9 XTAL2**

Output from the inverting Oscillator amplifier.

**2.2.10 AVCC**

AVCC is the supply voltage pin for Port A and the A/D Converter. It should be externally connected to  $V_{CC}$ , even if the ADC is not used. If the ADC is used, it should be connected to  $V_{CC}$  through a low-pass filter.

**2.2.11 AREF**

AREF is the analog reference pin for the A/D Converter.

**3. Resources**

A comprehensive set of development tools, application notes and datasheets are available for download on <http://www.atmel.com/avr>.

**4. Data Retention**

Reliability Qualification results show that the projected data retention failure rate is much less than 1 PPM over 20 years at 85°C or 100 years at 25°C.

**5. About Code Examples**

This documentation contains simple code examples that briefly show how to use various parts of the device. These code examples assume that the part specific header file is included before compilation. Be aware that not all C Compiler vendors include bit definitions in the header files and interrupt handling in C is compiler dependent. Please confirm with the C Compiler documentation for more details.

## 2. Datasheet IC Max 232

### MAX232, MAX232I

#### DUAL EIA-232 DRIVERS/RECEIVERS

SLLS047L - FEBRUARY 1989 - REVISED MARCH 2004

- Meets or Exceeds TIA/EIA-232-F and ITU Recommendation V.28
- Operates From a Single 5-V Power Supply With 1.0- $\mu$ F Charge-Pump Capacitors
- Operates Up To 120 kbit/s
- Two Drivers and Two Receivers
- $\pm 30$ -V Input Levels
- Low Supply Current ... 8 mA Typical
- ESD Protection Exceeds JESD 22 - 2000-V Human-Body Model (A114-A)
- Upgrade With Improved ESD (15-kV HBM) and 0.1- $\mu$ F Charge-Pump Capacitors is Available With the MAX202
- Applications
  - TIA/EIA-232-F, Battery-Powered Systems, Terminals, Modems, and Computers


**description/ordering information**

The MAX232 is a dual driver/receiver that includes a capacitive voltage generator to supply TIA/EIA-232-F voltage levels from a single 5-V supply. Each receiver converts TIA/EIA-232-F inputs to 5-V TTL/CMOS levels. These receivers have a typical hysteresis of 1.3 V, a typical hysteresis of 0.5 V, and can accept  $\pm 30$ -V inputs. Each driver converts TTL/CMOS input levels into TIA/EIA-232-F levels. The driver, receiver, and voltage-generator functions are available as cells in the Texas Instruments LinASIC™ library.

**ORDERING INFORMATION**

TA	PACKAGET	ORDERABLE PART NUMBER	TOP-SIDE MARKING
0°C to 70°C	PDIP (N)	Tube of 25 MAX232N	MAX232N
	SOIC (D)	Tube of 40 MAX232D	MAX232
		Reel of 2500 MAX232DR	MAX232
		SOIC (DW)	Tube of 40 MAX232DW
	SOP (NS)	Reel of 2000 MAX232NSR	MAX232
-40°C to 85°C	PDIP (N)	Tube of 25 MAX232IN	MAX232IN
	SOIC (D)	Tube of 40 MAX232ID	MAX232I
		Reel of 2500 MAX232IDR	MAX232I
		SOIC (DW)	Tube of 40 MAX232IDW
	Reel of 2000 MAX232IDWR	MAX232I	


† Package drawings, standard packing quantities, thermal data, symbolization, and PCB design guidelines are available at [www.ti.com/iso/package](http://www.ti.com/iso/package).



LinASIC is a trademark of Texas Instruments.

Please be aware that an important notice concerning availability, standard warranty, and use in critical applications of Texas Instruments semiconductor products and disclaimers thereto appears at the end of this data sheet.

Copyright © 2004, Texas Instruments Incorporated



**MAX232, MAX232I**  
**DUAL EIA-232 DRIVERS/RECEIVERS**

BL15047L - FEBRUARY 1989 - REVISED MARCH 2004

**Function Tables**

**EACH DRIVER**

INPUT TIN	OUTPUT TOUT
L	H
H	L

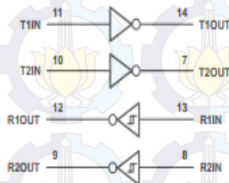
H = high level, L = low level

**EACH RECEIVER**

INPUT RIN	OUTPUT ROUT
L	H
H	L

H = high level, L = low level

logic diagram (positive logic)





## MAX232, MAX2321 DUAL EIA-232 DRIVERS/RECEIVERS

SLS047L – FEBRUARY 1989 – REVISED MARCH 2004

### absolute maximum ratings over operating free-air temperature range (unless otherwise noted)<sup>†</sup>

Input supply voltage range, $V_{CC}$ (see Note 1)	-0.3 V to 6 V
Positive output supply voltage range, $V_{S+}$	$V_{CC} - 0.3$ V to 15 V
Negative output supply voltage range, $V_{S-}$	-0.3 V to -15 V
Input voltage range, $V_I$ : Driver	-0.3 V to $V_{CC} + 0.3$ V
Receiver	$\pm 30$ V
Output voltage range, $V_O$ : T1OUT, T2OUT	$V_{S-} - 0.3$ V to $V_{S+} + 0.3$ V
R1OUT, R2OUT	-0.3 V to $V_{CC} + 0.3$ V
Short-circuit duration: T1OUT, T2OUT	Unlimited
Package thermal impedance, $\theta_{JA}$ (see Notes 2 and 3): D package	73°C/W
DW package	57°C/W
N package	67°C/W
NS package	64°C/W
Operating virtual junction temperature, $T_J$	150°C
Storage temperature range, $T_{stg}$	-65°C to 150°C

<sup>†</sup> Stresses beyond those listed under "absolute maximum ratings" may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated under "recommended operating conditions" is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

NOTES: 1. All voltages are with respect to network GND.

2. Maximum power dissipation is a function of  $T_J(\max)$ ,  $\theta_{JA}$ , and  $T_A$ . The maximum allowable power dissipation at any allowable ambient temperature is  $P_D = (T_J(\max) - T_A)/\theta_{JA}$ . Operating at the absolute maximum  $T_J$  of 150°C can affect reliability.

3. The package thermal impedance is calculated in accordance with JESD 51-7.

### recommended operating conditions

		MIN	NOM	MAX	UNIT
$V_{CC}$	Supply voltage	4.5	5	5.5	V
$V_{IH}$	High-level input voltage (T1IN, T2IN)	2			V
$V_{IL}$	Low-level input voltage (T1IN, T2IN)			0.8	V
R1IN, R2IN	Receiver input voltage			$\pm 30$	V
$T_A$	Operating free-air temperature	MAX232	0	70	°C
		MAX2321	-40	85	

### electrical characteristics over recommended ranges of supply voltage and operating free-air temperature (unless otherwise noted) (see Note 4 and Figure 4)

PARAMETER	TEST CONDITIONS	MIN	TYP <sup>‡</sup>	MAX	UNIT
$I_{CC}$	Supply current $V_{CC} = 5.5$ V, $T_A = 25^\circ\text{C}$ , All outputs open,		8	10	mA

<sup>‡</sup> All typical values are at  $V_{CC} = 5$  V and  $T_A = 25^\circ\text{C}$ .

NOTE 4: Test conditions are C1-C4 = 1  $\mu\text{F}$  at  $V_{CC} = 5 \text{ V} \pm 0.5 \text{ V}$ .

## MAX232, MAX231 DUAL EIA-232 DRIVERS/RECEIVERS

SLLS047L - FEBRUARY 1989 - REVISED MARCH 2004

### DRIVER SECTION

electrical characteristics over recommended ranges of supply voltage and operating free-air temperature range (see Note 4)

PARAMETER	TEST CONDITIONS	MIN	TYP†	MAX	UNIT
$V_{OH}$ High-level output voltage	T1OUT, T2OUT $R_L = 3\text{ k}\Omega$ to GND	5	7		V
$V_{OL}$ Low-level output voltage‡	T1OUT, T2OUT $R_L = 3\text{ k}\Omega$ to GND		-7	-5	V
$r_o$ Output resistance	T1OUT, T2OUT $V_S = V_S - 0$ , $V_O = \pm 2\text{ V}$	300			$\Omega$
$I_{OS}^{\S}$ Short-circuit output current	T1OUT, T2OUT $V_{CC} = 5.5\text{ V}$ , $V_O = 0$		$\pm 10$		mA
$I_{IS}$ Short-circuit input current	T1IN, T2IN $V_I = 0$			200	$\mu\text{A}$

† All typical values are at  $V_{CC} = 5\text{ V}$ ,  $T_A = 25^\circ\text{C}$ .

‡ The algebraic convention, in which the least-positive (most negative) value is designated minimum, is used in this data sheet for logic voltage levels only.

§ Not more than one output should be shorted at a time.

NOTE 4: Test conditions are C1-C4 = 1  $\mu\text{F}$  at  $V_{CC} = 5\text{ V} \pm 0.5\text{ V}$ .

switching characteristics,  $V_{CC} = 5\text{ V}$ ,  $T_A = 25^\circ\text{C}$  (see Note 4)

PARAMETER	TEST CONDITIONS	MIN	TYP	MAX	UNIT
SR Driver slew rate	$R_L = 3\text{ k}\Omega$ to 7 k $\Omega$ , See Figure 2			30	V/ $\mu\text{s}$
SR(t) Driver transition region slew rate	See Figure 3		3		V/ $\mu\text{s}$
Data rate	One TOUT switching		120		kBWs

NOTE 4: Test conditions are C1-C4 = 1  $\mu\text{F}$  at  $V_{CC} = 5\text{ V} \pm 0.5\text{ V}$ .

### RECEIVER SECTION

electrical characteristics over recommended ranges of supply voltage and operating free-air temperature range (see Note 4)

PARAMETER	TEST CONDITIONS	MIN	TYP†	MAX	UNIT
$V_{OH}$ High-level output voltage	R1OUT, R2OUT $I_{OH} = -1\text{ mA}$	3.5			V
$V_{OL}$ Low-level output voltage‡	R1OUT, R2OUT $I_{OL} = 3.2\text{ mA}$			0.4	V
$V_{IT+}$ Receiver positive-going input threshold voltage	R1IN, R2IN $V_{CC} = 0\text{ V}$ , $T_A = 25^\circ\text{C}$		1.7	2.4	V
$V_{IT-}$ Receiver negative-going input threshold voltage	R1IN, R2IN $V_{CC} = 5\text{ V}$ , $T_A = 25^\circ\text{C}$	0.8	1.2		V
$V_{thys}$ Input hysteresis voltage	R1IN, R2IN $V_{CC} = 5\text{ V}$	0.2	0.5	1	V
$\eta$ Receiver input resistance	R1IN, R2IN $V_{CC} = 5$ , $T_A = 25^\circ\text{C}$	3	5	7	k $\Omega$

† All typical values are at  $V_{CC} = 5\text{ V}$ ,  $T_A = 25^\circ\text{C}$ .

‡ The algebraic convention, in which the least-positive (most negative) value is designated minimum, is used in this data sheet for logic voltage levels only.

NOTE 4: Test conditions are C1-C4 = 1  $\mu\text{F}$  at  $V_{CC} = 5\text{ V} \pm 0.5\text{ V}$ .

switching characteristics,  $V_{CC} = 5\text{ V}$ ,  $T_A = 25^\circ\text{C}$  (see Note 4 and Figure 1)

PARAMETER	TYP	UNIT
$t_{PLH(R)}$ Receiver propagation delay time, low-to-high-level output	500	ns
$t_{PHL(R)}$ Receiver propagation delay time, high-to-low-level output	500	ns

NOTE 4: Test conditions are C1-C4 = 1  $\mu\text{F}$  at  $V_{CC} = 5\text{ V} \pm 0.5\text{ V}$ .

### 3. Datasheet Motor Servo HS-325HB

PREPARED BY JUN HEE LEE  
UPDATE: APR 01, 2002

## ANNOUNCED SPECIFICATION OF HS-325HB STANDARD BALL BEARING SERVO

<p>1. TECHNICAL VALUE</p> <p>CONTROL SYSTEM</p> <p>OPERATING VOLTAGE RANGE</p> <p>TEST VOLTAGE</p> <p>OPERATING SPEED</p> <p>STALL TORQUE</p> <p>IDLE CURRENT</p> <p>RUNNING CURRENT</p> <p>STALL CURRENT</p> <p>DEAD BAND WIDTH</p> <p>OPERATING TRAVEL</p> <p>DIRECTION</p> <p>MOTOR TYPE</p> <p>POTENTIOMETER TYPE</p> <p>AMPLIFIER TYPE</p> <p>DIMENSIONS</p> <p>WEIGHT</p> <p>BALL BEARING</p> <p>GEAR MATERIAL</p> <p>HORN GEAR SPLINE</p> <p>SPLINED HORNS</p> <p>CONNECTOR WIRE LENGTH</p> <p>CONNECTOR WIRE STRAND COUNTER</p> <p>CONNECTOR WIRE GAUGE</p>	<p>~PULSE WIDTH CONTROL 1500usec NEUTRAL</p> <p>~4.8V TO 6.0V</p> <p>~AT 4.8V</p> <p>~0.19sec/60°/MT NO LOAD</p> <p>~3kg.cm(41.66oz.in)</p> <p>~7.4mA AT STOPPED</p> <p>~160mA/60°/MT NO LOAD</p> <p>~700mA</p> <p>~5usec</p> <p>~40°/NONE SIDE PULSE TRAVELING 400usec</p> <p>~CLOCK WISE/PULSE TRAVELING 1500 TO 1900usec</p> <p>~CORED METAL BRUSH</p> <p>~4 SLIDER/DIRECT DRIVE</p> <p>~ANALOG CONTROLLER &amp; TRANSISTOR DRIVER</p> <p>~40x20x36.5mm(1.57"x0.78"x1.43in)</p> <p>~49g(1.51oz)</p> <p>~TOP BALL BEARING</p> <p>~2 HEAVY DUTY RESIN</p> <p>~24 SEGMENTS/ON/76</p> <p>~RESULAR/R-C,R-D,R-1,R-O,R-X,SUPER/R-XA</p> <p>~300mm(11.81in)</p> <p>~40EA</p> <p>~25AWG</p>
---	---

<p>2. FEATURES</p> <p>LONG LIFE POTENTIOMETER, TOP BALL BEARING, 2 HEAVY DUTY RESIN GEARS</p>	<p>3. APPLICATIONS</p> <p>AIRCRAFT 20-40 SIZE, STEERING AND THROTTLE SERVO FOR CARS, TRUCK AND BOATS</p>
---	--

<p>4. ACCESSORY &amp; OPTION</p> <p>CASE SET/</p> <p>HS322T:1EA</p> <p>HS322M:1EA</p> <p>HS322L:1EA</p> <p>PH-VT-2, 2x30 NI:4EA</p>	<p>GEAR SET/</p> <p>HS3220:1EA</p> <p>HS325G2:1EA</p> <p>HS325G3:1EA</p> <p>HS3220:1EA</p> <p>HS300RB:1EA</p>	<p>BALL BEARING SET/</p> <p>MR106x1EA</p>	<p>HORN SET/</p> <p>R-C:1EA</p> <p>R-D:1EA</p> <p>R-1:1EA</p> <p>R-O:1EA</p> <p>R-X:1EA</p> <p>R-XA:1EA</p> <p>WH/W 2.1x15 NI:4EA</p> <p>BST 3x5.5:4EA</p> <p>NBR 9x6.5x6:4EA</p>
---	---	---	---

**HITEC RCD KOREA INC.**

#### 4. Spesifikasi webcam PK-710G Anti-glare dan M-Tech 1,3 MP

PK-710G Anti-glare WebCam



Model : PK-710G  
Computer interface : USB 2.0  
Microphone : Built-in  
Still image : Up to 16 Megapixel, 4608x3456  
(software interpolated)  
Image sensor : 1/6" CMOS, 640x480 pixels  
Frame rate : 30fps  
Lens : F=2.4, F=3.5 mm  
Viewing angle : 60 degrees  
Focus range : Automatic focus, 10 cm to infinity  
Exposure control : Automatic  
White balance : Automatic

Webcam M-Tech 1.3MP



Speck :  
- 1.3 MP  
- Clear audio input and output  
- High Quality Video up to 30 fps  
- Plug and play  
- Koneksi USB  
- Bisa untuk YM, SKYPE, QQ, AQL, Windows Live Messenger



## DAFTAR RIWAYAT HIDUP



### Biodata

Nama : Ari Firman Andriyanto  
Jenis Kelamin : Laki  
Tempat Lahir : Surabaya  
Tanggal Lahir : 03 Januari 1993  
Agama : Islam  
Kebangsaan : Indonesia  
Alamat : Jl Zainal Abidin  
59 B Waru Sidoarjo  
HP : 081357395412  
E-mail : arifirman@yahoo.com

### Riwayat Pendidikan

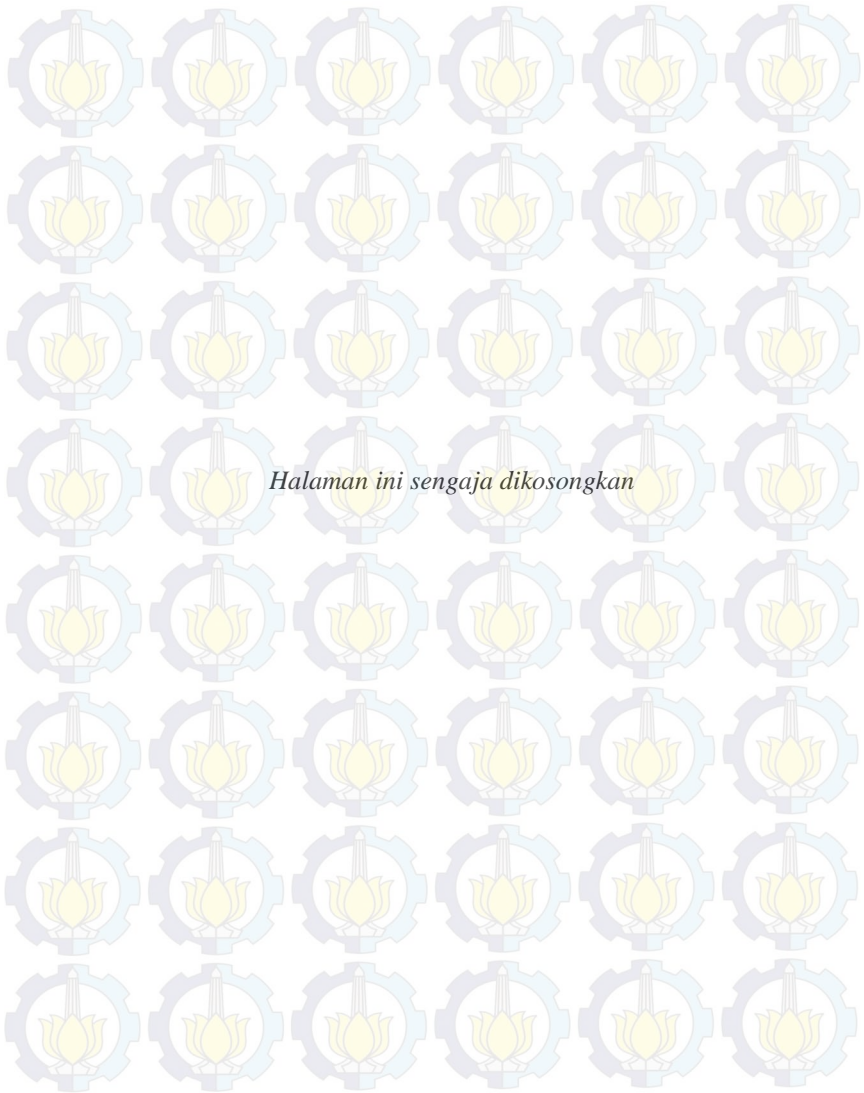
2011 - sekarang : Program Studi Diploma 3 Teknik Elektro – Bidang Studi Komputer Kontrol, Jurusan Teknik Elektro – Institut Teknologi Sepuluh Nopember (ITS) Surabaya  
2008 - 2011 : SMA NEGERI 1 WARU  
2005 - 2008 : SMP NEGERI 1 WARU  
1999 - 2005 : SDN TAMBAK REJO 1

### Pengalaman Kerja

- Kerja Praktek di PT. Astra otopart, Jakarta (Juli - Agustus 2013)

### Pengalaman Organisasi

- Staff Department Dalam Negeri HIMAD3TEKTRO 2012/2013
- Koordinator Komunitas Pencita Alam D3 Teknik Elektro ITS
- Sie Akomodasi Industrial Automation and Robotic Competition (IARC) 2012 dan 2013





### **Biodata**

Nama : Welly Julian  
Jenis Kelamin : Laki-laki  
Tempat Lahir : Jakarta  
Tanggal Lahir : 19 Juli 1993  
Agama : Islam  
Kebangsaan : Indonesia  
Alamat : Jl Keputih gg 1 No 5  
Surabaya  
HP : 087782340443  
E-mail : wellyjulian@yahoo.com

### **Riwayat Pendidikan**

2011 - sekarang : Program Studi Diploma 3 Teknik Elektro – Bidang  
Studi Komputer Kontrol, Jurusan Teknik Elektro –  
Institut Teknologi Sepuluh Nopember (ITS) Surabaya  
2008 - 2011 : SMA NEGERI 9 SURABAYA  
2005 - 2008 : SMP NEGERI 27 SURABAYA  
1999 - 2005 : SD NEGERI HARAPAN JAYA 1 BEKASI UTARA

### **Pengalaman Kerja**

- Kerja Praktek di PJB UP Brantas, PLTA Sengguruh, Malang (Juli - Agustus 2013)

### **Pengalaman Organisasi**

- Staff Department Dalam Negeri HIMAD3TEKTRO 2012/2013
- Sie Dana Industrial Automation and Robotic Competition (IARC) 2012 dan 2013
- Trainer Pelatihan MOS Guru Se-Jatim 2013 dan 2014
- Coordinator Department Litbang di Komunitas Hati Bening Progressive dan Happy and Happy Corporation.



