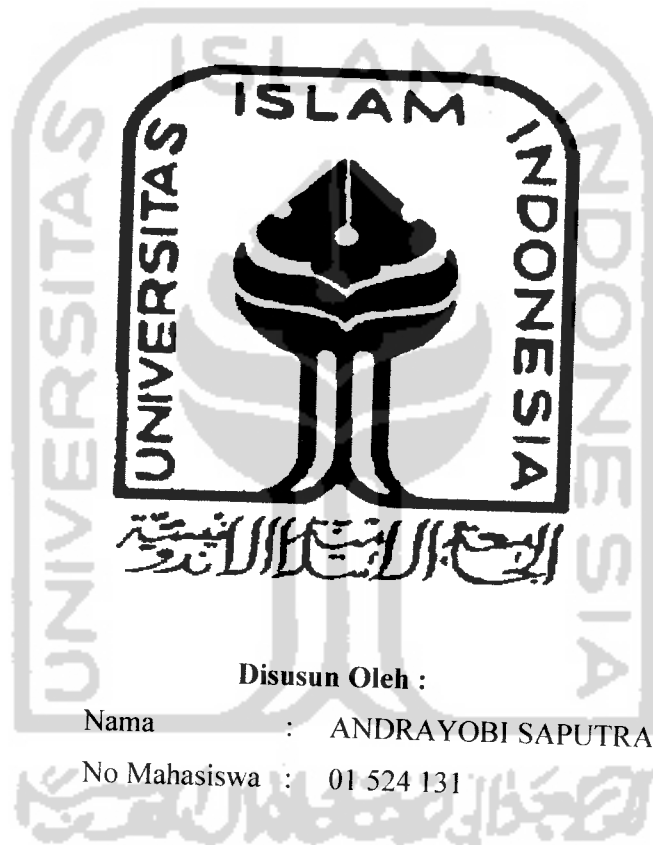


**PENGENDALI LENGAN ROBOT DENGAN ASK  
(AMPLITUDO SHIFT KEYING)**

**TUGAS AKHIR**

Diajukan sebagai Salah Satu Syarat Untuk Memperoleh Gelar Sarjana  
Teknik Elektro - Fakultas Teknologi Industri  
Universitas Islam Indonesia



**Disusun Oleh :**

Nama : ANDRAYOBI SAPUTRA

No Mahasiswa : 01 524 131

**TEKNIK ELEKTRO  
FAKULTAS TEKNOLOGI INDUSTRI  
UNIVERSITAS ISLAM INDONESIA  
YOGYAKARTA**

**2007**

**LEMBAR PENGESAHAN DOSEN PENGUJI**  
**PENGENDALI LINGKUP ROBOT DENGAN ASK**  
**(AMPLITUDO SHIFT KEYING)**

**TUGAS AKHIR**

**Disusun oleh :**

**Nama : ANDRA YOBI SAPUTRA**

**No. Mhs : 01 524 131**

**Telah Dipertahankan Di Depan Sidang Penguji, Sebagai Salah Satu**  
**Syarat Untuk Memperoleh Gelar Sarjana Teknik Elektro,**  
**Fakultas Teknologi Industri, Universitas Islam Indonesia**



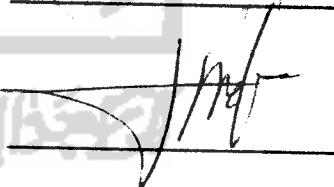
**Yogyakarta, 28 September 2007**

**Tim Penguji**


**Tito Yuwono, ST. M.Sc**  
**Ketua**

**Yusuf Aziz Amrullah, ST**  
**Anggota I**

**Hendra Setiawan, ST. MT**  
**Anggota II**

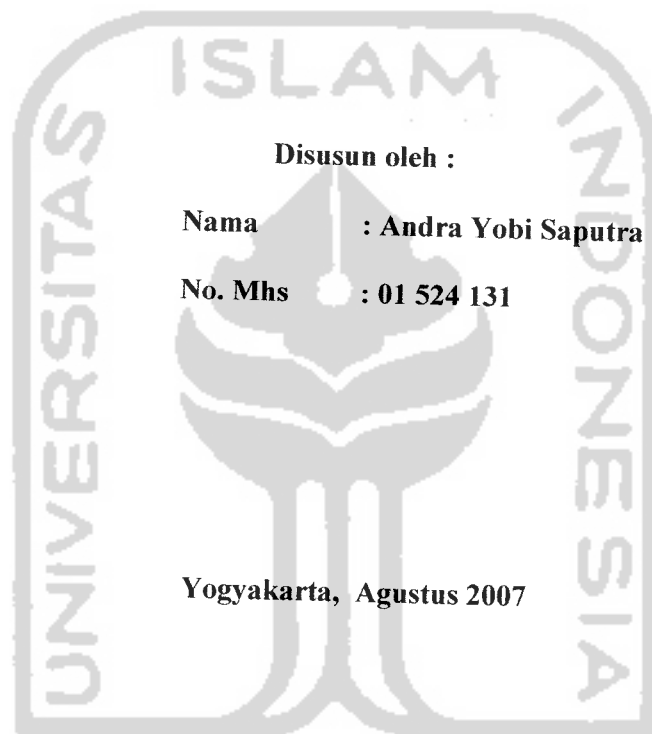
  
\_\_\_\_\_  
  
\_\_\_\_\_  
  
\_\_\_\_\_

**Mengetahui,**  
**Tim Penguji Sarjana Teknik Elektro**  
**Universitas Islam Indonesia**

  
**(Tito Yuwono, ST. M.Sc)**

**LEMBAR PENGESAHAN PEMBIMBING**  
**TUGAS AKHIR**

Pengendali Lengan Robot dengan ASK  
(Amplitudo Shift Keying )



Dosen Pembimbing I

(TITO YUWONO, ST., M.Sc.)

Dosen Pembimbing II

(YUSUF AZIZ. A., ST)

## HALAMAN PERSEMBAHAN

*Dengan penuh hormat, cinta, dan kasih sayang skripsi ini dipersembahkan sebagai ungkapan terima kasih yang tulus kepada mereka yang sangat berarti dalam hidupku:*

*Ayahanda dan Ibunda tercinta terima kasih atas limpahan cinta dan kasih sayang, doa, dukungan serta kepercayaan yang diberikan kepada ananda.*

*Kakanda tercinta terima kasih atas doa, dukungan serta yang selalu memberikan semangat kepada ananda.*

*serta*

*Orang-orang yang sangat dan selalu ananda sayangi.*

*Semoga semuanya akan menjadi kenangan yang terindah dan tak terlupakan.*

*Amien Ya Robbal 'Aalamien.....*

## MOTTO

*"Hai orang-orang yang beriman, mintalah pertolongan dari Allah dengan kesabaran dan salat. Sungguh Allah bersama orang-*

*orang yang sabar."*

*(QS. Al-Baqarah:153)*

*"Sungguh bersama kesukaran pasti ada kemudahan"*

*(QS. Asy Syarh : 5)*

*"Tiada suatu kepayahan, kesakitan, kesedihan, kesusahan, penderitaan dan kesukaran sampaipun duri yang menyakitkan itu menimpa kepada seorang mu'min melainkan dengan itu semua*

*Allah akan menutupi dosa-dosanya"*

*(HR. Bukhari dan Muslim)*

## KATA PENGANTAR



*Assalamu 'alaikum Wr. Wb.*

Puji dan syukur kepada Allah SWT yang telah memberikan rahmat dan hidayah-Nya kepada penulis, sehingga penulis dapat menyelesaikan Tugas Akhir ini dengan baik. Dan juga tidak lupa penulis menghaturkan shalawat serta salam kepada junjungan besar Nabi Muhammad SAW, keluarga, para sahabat, dan semua para pengikut-pengikutnya hingga akhir jaman. Semoga kita semua termasuk didalamnya. Tugas Akhir ini berjudul “PENGENDALI LENGAN ROBOT DENGAN ASK (AMPLITUDO SHIFT KEYING)”, disusun untuk memenuhi salah satu syarat dalam menyelesaikan program studi demi meraih gelar sarjana S1 pada jurusan Teknik Elektro Universitas Islam Indonesia.

Dengan berakhirnya tugas akhir ini penulis juga ingin mengucapkan banyak terima kasih kepada pihak-pihak yang telah banyak membantu penulis semasa kuliah dan selama penyusunan tugas akhir ini, yaitu kepada :

1. Bapak Fathul Wahid ST, M Sc selaku Dekan Fakultas Teknologi Industri Universitas Islam Indonesia.
2. Bapak Tito Yuwono, ST, M.Sc selaku Ketua Jurusan Teknik Elektro Fakultas Teknologi Industri Universitas Islam Indonesia, Sekaligus Pemimbing I.

3. Bapak Yusuf Aziz, ST selaku dosen pembimbing II yang telah banyak membimbing serta memberikan arahan kepada penulis dalam menyelesaikan tugas akhir ini.
4. Segenap Dosen Fakultas Teknologi Industri Universitas Islam Indonesia pada umumnya serta Dosen dan Karyawan Jurusan Teknik Elektro pada khususnya yang telah banyak memberikan ilmunya.
5. Ayahanda, Ibunda dan Adinda tercinta yang telah memberikan segala dukungan, pengorbanan, kasih sayang serta do'anya demi kelancaran tugas akhir ini.
6. Sahabat dan teman-teman kost serta kontrakan yang selalu bersama baik senang maupun susah.
7. Sahabat dan teman-teman main yang senantiasa menemani selama kurang lebih 6 tahun penulis hidup di Yogyakarta.
8. Sahabat dan teman-teman di kampus yang banyak membantu baik kuliah maupun pembuatan tugas akhir ini.
9. Pihak-pihak yang tidak mungkin penulis sebutkan satu persatu.

Dalam penulisan Tugas Akhir ini penulis menyadari bahwa masih jauh dari sempurna dan terdapat kekurangan, dengan demikian penulis mengharapkan kritikan yang bersifat membangun serta saran-saran kepada penulis sehingga dalam penulisan berikutnya akan lebih baik.

Apabila ada kesalahan, semata-mata kekhilafan Penulis, sedangkan kebenaran semuanya hanyalah milik ALLAH SWT.

Akhir kata besar harapan penulis laporan ini bermanfaat bagi pengembangan ilmu pengetahuan dan teknologi untuk masa yang akan datang.

*Wassalamu'alaikum Wr. Wb.*

Yogyakarta, Agustus 2007

Penulis





## ABSTRAK

Dalam pergerakan lengan robot, digunakan motor servo sebagai penggerakannya. Dalam lengan Robot ini digunakan 4 buah jenis motor servo yang digunakan untuk menggerakkan 4 buah *joint*. Badan robot dibuat dengan menggunakan bahan *acrylic*. *joint* 1 digunakan Hitec HS-311 yang merupakan standar servo, sehingga robot hanya dapat bergerak dalam jangkauan 180°. *Joint* 2 merupakan *joint* yang memerlukan torsi paling besar. Pada *Joint* 2 digunakan dua buah motor servo. Motor yang digunakan parallax px-900-00008pf yang merupakan *continous* motor servo. Motor yang lain adalah hitec HS-5645MG. motor ini adalah jenis *high torque* yang mampu mengangkat 13,1 kg pada tegangan 6 volt. Untuk *joint* 3 digunakan HS-322 yang juga merupakan servo standar. sedangkan *joint* 4 dikontrol oleh HS-311. Dalam pemrograman mikrokontroler, digunakan program Delphi. Bahasa ini mudah untuk dipahami. Dalam pengontrolannya digunakan PC (*Personal Computer*) dengan ASK (*Amplitude Shift Keying*) sebagai pengirim dan penerima data masukan, menggunakan mikrokontroler ATtiny2313. Berdasarkan percobaan yang dilakukan, didapatkan data bahwa kesalahan derajat pergerakan pada lengan robot masih sangat banyak.



## DAFTAR ISI

HALAMAN JUDUL.....	i
LEMBAR PENGESAHAN PEMBIMBING.....	ii
LEMBAR PENGESAHAN PENGUJI.....	iii
HALAMAN PERSEMBAHAN.....	iv
MOTTO.....	v
KATA PENGANTAR.....	vi
ABSTRAK.....	ix
DAFTAR ISI.....	x
DAFTAR GAMBAR.....	xiv
DAFTAR TABEL.....	xv
<b>BABI PENDAHULUAN</b>	
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Batasan Masalah.....	2
1.4 Tujuan Penelitian.....	3
1.5 Manfaat.....	3
1.6 Metodologi Penulisan.....	3
1.7 Sistematika Penulisan.....	4

## BAB II DASAR TEORI

2.1 Robot.....	6
2.1.1 Definisi Robot.....	6
2.1.2 Struktur Robot.....	6
2.1.3 Perangkat Robot.....	7
2.2 Dioda.....	8
2.2.1 Teori Dioda.....	8
2.3 Mikrokontroler AVR ATtiny2313.....	11
2.3.1 Arsitektur ATtiny 2313.....	11
2.3.2 Blok Diagram.....	14
2.3.3 Hardware IC.....	16
2.3.4 Organisasi Memori.....	18
2.4 Sinyal Port Serial dan Fungsi Pin.....	24
2.4.1 Perangkat lunak <i>Borland Delphi</i> .....	26
A. Form <i>Borland Delphi</i> .....	26
B. Unit <i>Borland Delphi</i> .....	27
2.4.2 Komponen <i>Borland Delphi</i> .....	28
2.4.3 <i>Object Inspector</i> .....	30
2.4.4 Komponen serial <i>Delphi</i> .....	30
2.5 Motor Servo.....	30
2.6 <i>Amplitude Shift Keying (ASK)</i> .....	31
2.7 <i>Binary ASK</i> .....	33

2.8 Kinerja <i>Binary</i> ASK.....	36
------------------------------------	----

### **BAB III PERANCANGAN SISTEM**

3.1 Diagram Blok.....	34
3.2 Perancangan Perangkat Keras.....	34
3.2.1 Rangkaian Pengirim Data.....	34
3.2.2 Rangkaian Penerima Data.....	35
3.2.3 Power Supply.....	36
3.2.4 Mikrokontroler.....	37
3.3 Kerangka Lengan Robot.....	38
3.4 Pengendalian Robot.....	43
3.4.1 <i>Forward Kinematic</i> .....	43
3.4.2 <i>Invers Kinematic</i> .....	44
3.5 Pergerakan Robot.....	45

### **BAB IV ANALISA DAN PEMBAHASAN**

4.1 Pengujian.....	50
4.1.1. Pengukuran PWM pada motor 1.....	50
4.1.2. Pengukuran PWM pada motor 2.....	51
4.1.3. Pengukuran PWM pada motor 3.....	52
4.1.4. Pengukuran PWM pada motor 4.....	53
4.1.5. Pengujian Data Error pada Lengan Robot pada Joint 1.....	54

4.1.6 Pengujian Data Error pada Lengan Robot pada Joint 2 .....	55
4.1.7. Pengujian Data Error pada Lengan Robot pada Joint 3 .....	55
4.1.8. Pengujian Data Error pada Lengan Robot pada Joint 4 .....	56
4.2 Perhitungan Baudret .....	57
4.3 Bentuk Gelombang PWM ( <i>Pulse Width Modulation</i> ) .....	59
4.4 Pengujian <i>Amplitude Shift Keying</i> (ASK) .....	60
4.5 Pengukuran Jarak <i>Amplitude Shift Keying</i> .....	60

## **BAB V PENUTUP**

5.1 Kesimpulan.....	62
5.2 Saran.....	63

## **DAFTAR PUSTAKA**

## **LAMPIRAN**



## DAFTAR TABEL

Tabel 2.1. Pin Port Serial dan Tipe Sinyal.....	25
Tabel 4.1. Pengujian nilai PWM pada <i>motor 1</i> .....	50
Tabel 4.2. Pengujian nilai PWM pada <i>motor 2</i> .....	51
Tabel 4.3. Pengujian nilai PWM pada <i>motor 3</i> .....	52
Tabel 4.4. Pengujian nilai PWM pada <i>motor 4</i> .....	53
Tabel 4.5. Pengujian <i>End Effector</i> pada <i>Joint 1</i> .....	54
Tabel 4.6. Pengujian <i>End Effector</i> pada <i>Joint 2</i> .....	55
Tabel 4.7. Pengujian <i>End Effector</i> pada <i>Joint 3</i> .....	56
Tabel 4.8. Pengujian <i>End Effector</i> pada <i>Joint 4</i> .....	56
Tabel 4.9. Pengujian Lebar Pulsa pada <i>Joint 1</i> .....	58
Tabel 4.10 Pengujian Lebar Pulsa pada <i>Joint 2</i> .....	58
Tabel 4.11 Pengujian Lebar Pulsa pada <i>Joint 3</i> .....	58
Tabel 4.12 Pengujian Lebar Pulsa pada <i>Joint 4</i> .....	59
Tabel 4.13 Pengamatan sinyal ASK.....	60
Tabel 4.14 Pengukuran Jarak Sinyal ASK dalam satu Ruangan.....	60
Tabel 4.15 Pengukuran Jarak Sinyal ASK dalam Ruang Bersekat.....	61
Tabel 4.16 Pengukuran Jarak Sinyal ASK dalam Ruang Beda Lantai.....	61

## DAFTAR GAMBAR

Gambar 2.1. Dioda dengan Penjar Maju dan Mundur .....	8
Gambar 2.2. Karakteristik Dioda .....	9
Gambar 2.3. Simbol Dioda Penyearah .....	10
Gambar 2.4. Simbol Photodiode .....	10
Gambar 2.5. Simbol LED .....	10
Gambar 2.6. Arsitektur ATtiny2313 .....	14
Gambar 2.7 IC ATtiny2313. ....	16
Gambar 2.8. Hubungan antar memori .....	18
Gambar 2.9. Peta Memori Program .....	19
Gambar 2.10. <i>General Purpose Register</i> .....	21
Gambar 2.11. Register <i>Pointer X, Y, Z</i> .....	21
Gambar 2.12 Peta Memori Data .....	22
Gambar 2.13. Skematik dari DB-9 ( <i>male</i> ) .....	24
Gambar 2.14. Motor Servo .....	31
Gambar 2.15. Gelombang Sinyal ASK .....	32
Gambar 2.16. Bentuk ASL, TLP433 .....	33
Gambar 2.17. Bentuk Gelombang ASK dengan Indeks Modulasi $m = 0$ .....	34
Gambar 2.18. Bentuk Gelombang ASK dengan Indeks Modulasi $m = \frac{1}{2}$ .....	34
Gambar 2.19. Bentuk Gelombang ASK dengan Indeks Modulasi $m = 1$ .....	35
Gambar 3.1. Diagram Blok Pengendali Lengan Robot .....	37
Gambar 3.2. Rangkaian pemancar ASK .....	38

Gambar 3.3. Rangkaian Penerima ASK.....	39
Gambar 3.4. Rangkaian catu daya.....	40
Gambar 3.5. Arsitektur Lengan Robot.....	42
Gambar 3.6 Bagian A.....	42
Gambar 3.7. Bagian B.....	43
Gambar 3.8. Bagian C, D, E, dan F.....	43
Gambar 3.9. Bagian G dan H.....	44
Gambar 3.10 Bagian I dan J.....	44
Gambar 3.11 Bagian K dan L.....	45
Gambar 3.12 <i>Forward Kinematic</i> .....	47
Gambar 3.13 <i>Flowchart</i> Delphi.....	48
Gambar 3.14 <i>Flowchart</i> Mikrokontroler.....	49
Gambar 4.1. Grafik Percobaan Motor 1.....	51
Gambar 4.2. Grafik Percobaan Motor 2.....	52
Gambar 4.3. Grafik Percobaan Motor 3.....	53
Gambar 4.4. Grafik Percobaan Motor 4.....	54
Gambar 4.5. Bentuk Gelombang PWM.....	59
Gambar 4.6. Bentuk Gelombang Data Serial.....	59



# BAB I

## PENDAHULUAN

### 1.1. Latar Belakang

Kebanyakan dari kita mengetahui komputer/PC hanya digunakan sebagai alat bantu mengetik, menonton film dan bermain game. Di negara-negara maju PC sudah banyak dipergunakan sebagai alat pengontrol/pengendali peralatan-peralatan disekitar kita atau sebagai alat bantu pendeteksi sesuatu benda/barang. Salah satu metode yang dapat digunakan pada tranmisi data dengan “*Amplitude Shift Keying*” apabila teknologi robot dan komputer/PC digabungkan , maka akan menjadikan teknologi yang tepat guna dan bermanfaat. Misal seperti robot yang digunakan sebagai penjinak bahan peledak yang dikendalikan dari jarak jauh melalui sebuah komputer/PC. Dikarenakan manusia sebagai user-nya akan lebih mudah untuk mengontrol robot itu sesuai dengan keinginan dan user lebih aman.

Perkembangan teknologi robot saat ini sangat berkaitan dengan kepresisian sebuah robot yaitu posisi *end effector* dan kecermatan robot dalam bergerak untuk melaksanakan pekerjaannya sesuai yang diperintahkan pengguna (*user*). Melalui PC (Personal Computer) yang sudah dilengkapi dengan program *User Interface* sebuah lengan robot dapat dikendalikan. User dapat mempelajari persamaan-persamaan *Forward* dan *Invers Kinematic* untuk mengetahui cara mencari posisi *end effector* lengan robot. Dengan menggunakan teknologi robot kita dapat mempercepat proses produksi, lebih presisi, mengefisienkan waktu dan masih banyak manfaat lainnya yang didapat dari penggunaan teknologi robot. Di

Indonesia sendiri teknologi robotika masih jauh tertinggal jika dibandingkan dengan negara-negara tetangga dan negara maju lainnya. Begitu juga dengan sebuah komputer atau yang lebih sering disebut PC (Personal Computer), negara kita sangat jauh tertinggal dalam hal pemanfaatan teknologi ini.

Proses-proses di industri banyak melibatkan sistem otomatisasi yang lebih dikenal dengan robot industri. Robot industri yang banyak digunakan saat ini adalah sistem lengan robot. Ada berbagai macam sistem lengan robot dengan berbagai macam motor sebagai penggerakannya. Sistem ini dirancang agar dapat bekerja pada tempat-tempat yang tidak dapat dikerjakan manusia, sehingga tangan robot ini perlu ketelitian yang bagus. Ketelitian ini terutama dibangun oleh penggerak joint-joint yang ada pada robot itu sendiri.

### **1.2. Rumusan Masalah**

Dari latar belakang di atas dapat diambil suatu rumusan masalah, bagaimana mengendalikan robot berbasis PC menggunakan *Amplitudo Shift Keying* (ASK) sebagai pengirim dan penerima data, dengan memanfaatkan mikrokontroler ATtiny2313 sebagai pengendali sistem.

### **1.3. Batasan masalah**

Pembatasan masalah dalam hal ini meliputi:

1. Lengan robot dikendalikan menggunakan PC (personal computer) dengan ASK (*Amplitude shift keying*) sebagai pengirim dan penerima data masukan,

menggunakan mikrokontroler ATtiny2313 Sebagai pembangkit pulsa motor lengan robot.

2. Untuk menentukan posisi *end effector* dari lengan robot digunakan *Forward kinematic* dan *invers kinematic* manipulator.
3. Besarnya sudut yang dapat dijangkau.

#### **1.4. Tujuan Penelitian**

Tujuan penelitian pada tugas akhir ini adalah mengendalikan lengan robot melalui komputer dengan ASK (*Amplitude Shift Keying*).

#### **1.5. Manfaat**

Memberikan kemudahan bagi manusia untuk mengendalikan lengan robot dari jarak jauh.

#### **1.6. Metodologi Penulisan**

Keberhasilan suatu pengembangan sistem akan tergantung pada teknik bagaimana menggunakan metode pengembangan sistem yang tepat yaitu bila metodologi tersebut mencakup seluruh aspek perancangan mulai dari tahap perumusan masalah yang dapat menentukan kebutuhan untuk merancang suatu sistem.

Adapun metodologi penelitian yang diterapkan sebagai berikut:

1. Studi Pustaka.

Pengumpulan data ini digunakan untuk mendapatkan informasi-informasi yang berkaitan dengan materi tugas akhir, sehingga dapat digunakan sebagai acuan.

2. Perancangan *software* dan *hardware*.

3. Perakitan atau pembuatan *software* dan *hardware*.

4. Pengujian *software* dan *hardware*.

5. Analisa.

### 1.7. Sistematika Penulisan

Sistematika penulisan dari laporan akhir penelitian ini dibagi dalam lima bab, dengan isi masing masing bab diuraikan sebagai berikut:

#### **BAB 1      Pendahuluan**

Berisi tentang latar belakang masalah, rumusan masalah, batasan masalah, tujuan penelitian, metodologi penelitian dan sistematika penulisan.

#### **BAB II     Landasan Teori**

Menguraikan Tentang teori-teori yang menjadi acuan dalam pembuatan tugas akhir, diantaranya teori tentang sistem pengendalian lengan robot dengan ASK, teori tentang mikrokontroler ATtiny 2313 sebagai pengendali.

### **BAB III Perancangan Sistem**

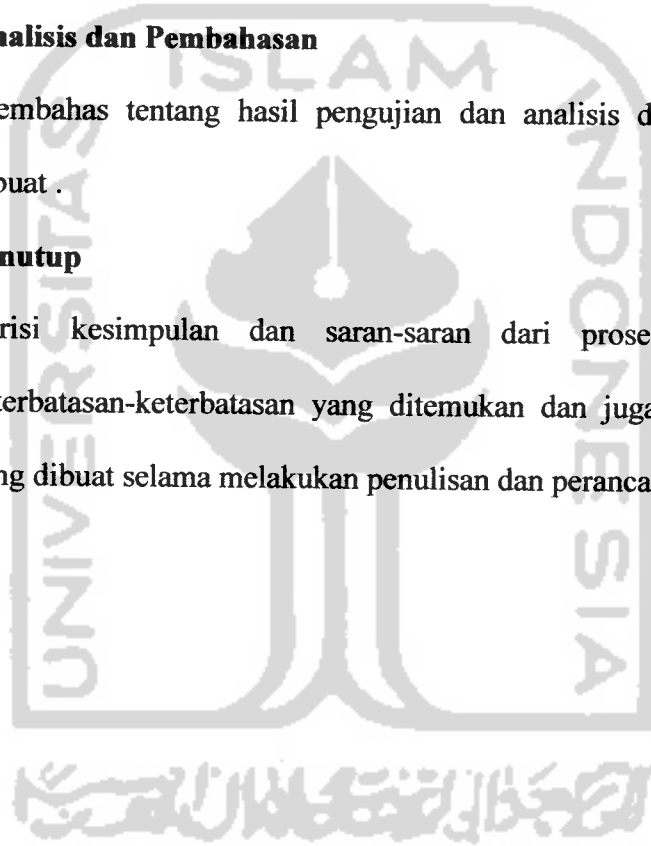
Berisi tentang perancangan dan pembuatan sistem pengendalian lengan robot mulai dari diagram blok, prinsip kerja, serta pembahasan mengenai software sebagai otak sistem pengendalinya.

### **BAB IV Analisis dan Pembahasan**

Membahas tentang hasil pengujian dan analisis dari sistem yang dibuat .

### **BAB V Penutup**

Berisi kesimpulan dan saran-saran dari proses perancangan, keterbatasan-keterbatasan yang ditemukan dan juga asumsi-asumsi yang dibuat selama melakukan penulisan dan perancangan hardware.



## **BAB II**

### **DASAR TEORI**

#### **2.2.1 Robot**

Robot menurut Karel Capek berasal dari kata Robota yang berarti budak/pekerja. Dengan menggunakan teknologi robot, dapat mempercepat proses produksi, mengefisienkan waktu dan masih banyak kegunaan lainnya dengan menggunakan teknologi robot.

##### **2.1.1 Definisi robot.**

Definisi Robot secara umum adalah suatu alat multi fungsi otomatis yang dapat diprogram untuk melakukan pekerjaan tertentu.

Unsur utama dalam kata "robot" adalah:

1. Seperangkat peralatan mesin/device.
2. Dapat diprogram
3. Bekerja/bergerak secara otomatis
4. Mampu melaksanakan tugas tertentu sesuai program.

##### **2.1.2 Struktur Robot**

Pada umumnya sebuah robot memiliki struktur sebagai berikut:

1. Manipulator
  - Badan (body)
  - Pergelangan (wrist)

- Lengan (arm )
- 2. *End Effector*  
Penjepit (gripper), bor.
- 3. Penggerak (Driver)  
Motor, hidrolik, pneumatik
- 4. Sensor  
Posisi, panas, kecepatan.
- 5. Pengendali (Controler)  
PC/Komputer, mikrokontroler, mikroprosesor.

### 2.1.3. Perangkat Robot

Macam-macam tenaga penggerak pada robot, yaitu :

1. Elektris  
Tenaga penggeraknya menggunakan jenis motor *stepper* dan motor *servo*
2. Hidrolis  
Penggerak pada jenis ini menggunakan semacam cairan untuk menggerakkan. Tenaganya diperoleh dari pompa hidrolis dan penggeraknya diatur oleh *solenoid valve* yang kemudian diteruskan ke bagian aktuator melalui piston.
3. Pneumatis  
Sama seperti penggerak hidrolis tetapi medium yang dipakai untuk menyalurkan tenaga berupa gas/udara.

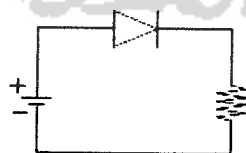
#### 4. Mekanis

Dihubungkan ke *Arithmetic Logic Unit* (ALU) melalui dua register independen, sehingga dapat diakses dalam satu instruksi yaitu satu siklus clock. Hasil dari arsitektur ini lebih efisien sepuluh kali dari mikrokontroler jenis CISC seperti AT89S2051.

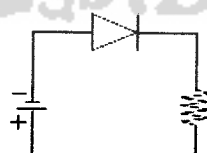
### 2.3 Dioda

#### 2.2.1 Teori Dioda

Dioda merupakan komponen semikonduktor yang mempunyai dua kaki yaitu anoda dan katoda dan digunakan sebagai penyearah karena hanya dapat dialiri arus dalam satu arah saja. Dioda merupakan salah satu komponen yang tidak tahan terhadap arus yang berlebihan dan apabila terjadi, maka dioda menjadi rusak. Dioda dibuat dari bahan jenis N yang mempunyai kelebihan elektron, disambung dengan bahan jenis P yang mempunyai kekurangan elektron. Bahan jenis N dan P dihasilkan dengan cara “*doping*” pada bahan semikonduktor germanium atau silikon dengan proses kimia, yaitu dengan proses pemanasan agar berfungsi, dan tersambung menjadi susunan kristal tunggal.



(a) Dioda dengan tegangan maju

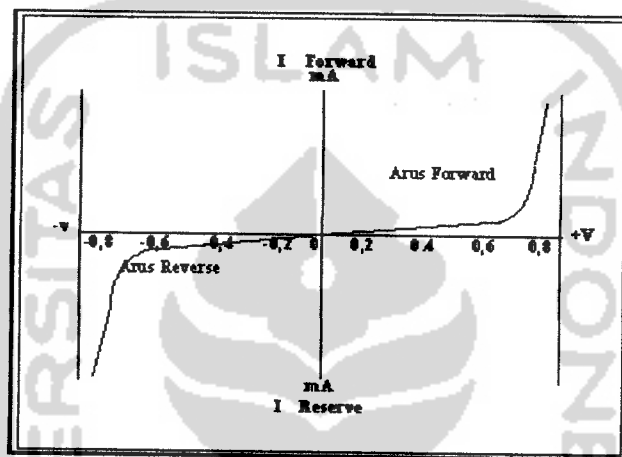


(b) Dioda dengan tegangan balik

**Gambar 2.1** Dioda dengan Panjar Maju dan Mundur



Karakteristik dari dioda adalah grafik arus anoda ( $I$ ) sebagai fungsi dari tegangan anoda-katoda ( $V$ ). Karakteristik dari dioda dapat dibedakan atas karakteristik arah maju (*forward*) dan arah mundur (*reverse*) yang diperlihatkan pada gambar 2.2. Pada gambar tersebut lonjakan terjadi pada tegangan yang relatif rendah, yaitu antara 0,6 Volt sampai 0,8 untuk bahan silikon dan lonjakan antara 0,2 Volt sampai 0,3 Volt untuk bahan germanium.



Gambar 2.2 Karakteristik Dioda.

Sifat dioda ideal adalah dioda akan mengalirkan arus jika anoda mendapatkan tegangan lebih positif dari pada katoda dan tidak mengalirkan arus jika anoda mendapatkan tegangan lebih negatif dari pada katoda. apabila anoda lebih positif dari pada katoda maka resistansi dioda sangat kecil (hampir sama dengan nol), dalam keadaan ini dioda dikatakan mendapat tegangan maju.

Berdasar pada kegunaan dan jenis, dioda terbagi menjadi beberapa macam yaitu :

##### 5. Dioda Penyearah (*rectifier*).

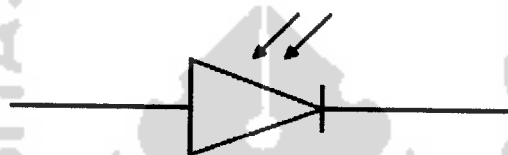
Merupakan dioda yang digunakan untuk menyearahkan tegangan bolak-balik menjadi searah.



**Gambar 2.3.** Simbol Dioda Penyearah.

c. Photodioda.

Merupakan dioda yang jika mendapatkan sinyal cahaya akan menghantarkan arus. Sinyal yang diterima adalah gelombang cahaya pada spektrum infra merah. Dioda ini dapat dipakai untuk pengubahan energi cahaya menjadi listrik.



**Gambar 2.4.** Simbol Photodioda.

c. Dioda Pemancar Cahaya (LED, *Light Emitting Diode*).

Merupakan dioda yang dapat mengeluarkan cahaya atau sinar, digunakan sebagai indikator. Untuk penggunaan pengendali jarak jauh (*Remote Control*) di pakai dioda emisi infra merah. Dioda ini ini dapat memancarkan gelombang cahaya pada spektrum infra merah. Led infra merah memiliki lensa yang berkualitas tinggi untuk memfokuskan cahaya infra merah sehingga membentuk berkas yang sempit.



**Gambar 2.5.** Simbol LED.

### 2.3 Mikrokontroler AVR ATtiny2313

Perkembangan teknologi dalam dunia elektronika dewasa ini sangat pesat khususnya perkembangan dalam dunia mikroelektronika. Mikrokontroler adalah suatu komponen semi konduktor yang didalamnya sudah terdapat suatu sistem mikroprosesor seperti ALU, ROM, RAM, dan Port I/O. Berdasarkan arsitekturnya mikrokontroler dapat dibedakan menjadi dua jenis yaitu mikrokontroler dengan arsitektur CISC dan mikrokontroler dengan arsitektur RISC. Perbedaan mendasar dari kedua jenis mikrokontroler ini adalah kecepatan dalam eksekusi program. Mikrokontroler dengan arsitektur CISC untuk mengeksekusi 1 cycle perintah membutuhkan 12 siklus clock sedangkan mikrokontroler RISC semua instruksi dikemas dalam kode 16-bit (*word*) sehingga sebagian besar instruksi dieksekusi dalam 1 siklus clock. Contoh mikrokontroler CISC adalah mikrokontroler keluarga ASM51 (AT89Cxx, AT89Sxx) sedangkan mikrokontroler RISC contohnya adalah mikrokontroler keluarga AVR (AT90Sxx, ATtiny, ATmega).

#### 2.3.1 Arsitektur ATtiny 2313

ATtiny2313 adalah mikrokontroler 8-bit berteknologi CMOS yang berdaya rendah berbasis AVR dengan arsitektur RISC. Mikrokontroler ATtiny2313 memiliki fitur-fitur utama antara lain sebagai berikut:

1. Sistem mikroprosesor RISC- 8bit dengan kecepatan maksimum 20Mhz
2. 120 macam instruksi,

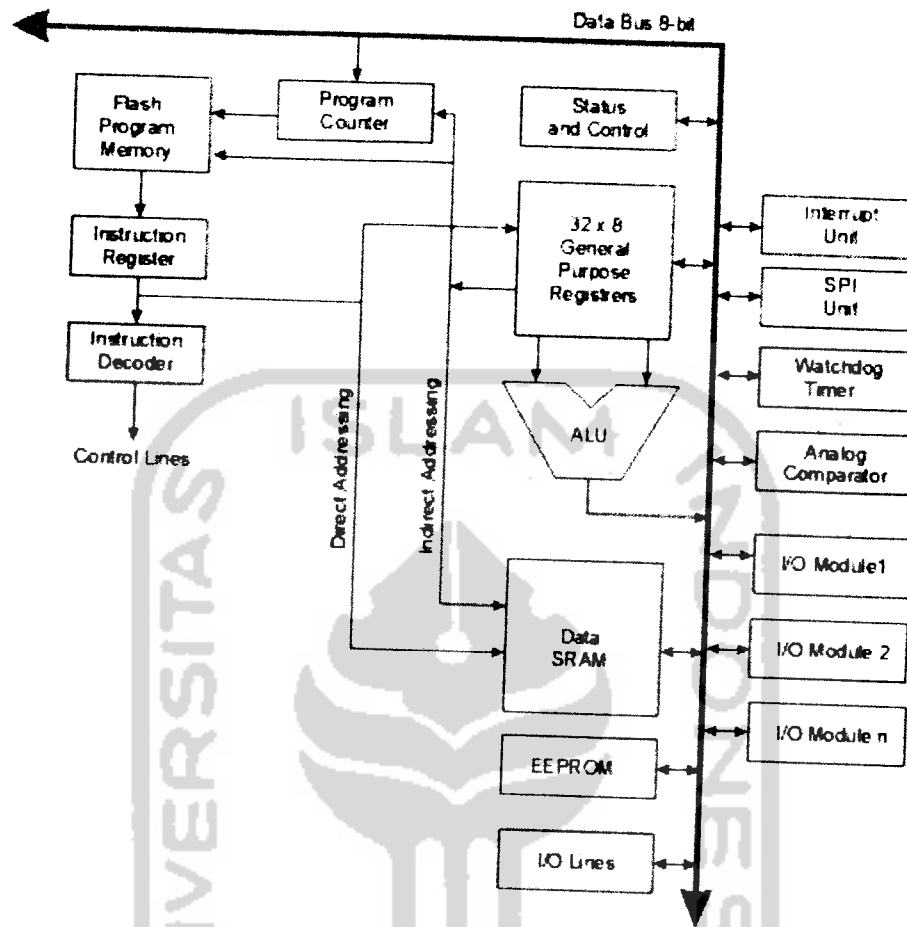
3. 32 x 8bit *General Purpose Register*,
4. 2 Kbyte memori program,
5. 128 byte memori SRAM,
6. 128 byte memori EEPROM,
7. Timer/Counter 8 bit dan 16 bit,
8. 18 pin jalur I/O,
9. 4 channel PWM (*Pulse Width Modulation*),
10. Komparator analog,
11. Komunikasi serial USART (*Universal Synchronous Asynchronous Receiver and Trasnmitter*),
12. Port SPI (*Serial Peripheral Interface*),
13. Internal Osilator dan Internal Power-on-reset.

Mikrokontroler AVR memiliki model arsitektur *harvard*, dimana memori untuk data dan program terpisah, bus untuk program dan bus untuk data juga terpisah. Dalam arsitektur AVR, seluruh GPR (*General Purphose Register* [32-register]) terhubung langsung ke ALU (*Arithmetic Logic Unit*) / prosesor, sehingga eksekusi instruksi lebih cepat. Dalam satu siklus clock terdapat dua register independen yang dapat diakses oleh satu instruksi. Teknik yang digunakan adalah *fetch during execution* atau memegang sambil mengerjakan. Hal ini berarti, dua operan dibaca dari satu register, dilakukan eksekusi operasi dan

hasilnya disimpan kembali kedalam satu register, semuanya dilakukan hanya dalam satu siklus clock.

Dari 32 register terdapat enam buah register yang dapat digunakan untuk pengalamatan tidak langsung 16-bit sebagai register pointer (penunjuk). ALU mendukung operasi bit, fungsi aritmetika dan logika antara register dengan register atau register dengan nilai konstan, atau hanya operasi satu register. Untuk kontrol aliran program disediakan instruksi lompatan bersyarat dan tak bersyarat, instruksi CALL (panggil), dapat ditempatkan di seluruh ruangan program. Kebanyakan instruksi AVR mempunyai format 16-bit word. Setiap alamat memori program mengandung sebuah instruksi 16- atau 32-bit.

Selama interupsi dan pemanggilan subrutin, alamat program counter (PC) disimpan kedalam *Stack*. *Stack* akan efektif diletakkan di SRAM, dan konsekuensinya ukuran *stack* dibatasi oleh ukuran SRAM dan penggunaan SRAM. Setiap pengguna harus menginisialisai SP dalam rutin RESET (sebelum subrutin atau interupsi dieksekusi). Stack pointer dapat ditulis dan dibaca dalam ruangan I/O. SRAM dapat diakses melalui lima mode pengalamatan dalam arsitektur AVR.



Gambar 2.6 Arsitektur ATtiny2313

### 2.3.2 Blok Diagram

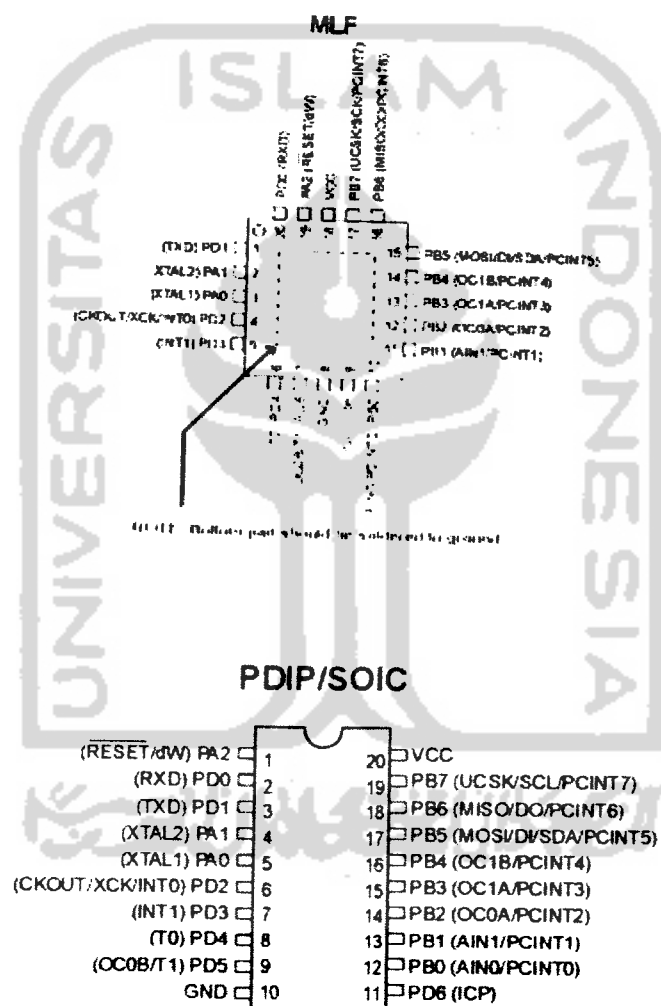
Inti kombinasi AVR adalah kombinasi instruksi yang beraneka ragam dengan 32 register (32 *general purpose working register*). Ke-32 register tersebut langsung dihubungkan ke *Arithmetic Logic Unit* (ALU) melalui dua register independen, sehingga dapat diakses dalam satu instruksi yaitu satu siklus clock. Hasil dari arsitektur ini lebih efisien sepuluh kali dari mikrokontroler jenis CISC seperti AT89Sxxx.

ATtiny 2313 memiliki 2K byte *In-System Programmable Flash*, 128 byte EEPROM, 128 byte SRAM, 18 saluran *general purpose I/O line*, 32 *general purpose working register*, *interface single-wire* untuk *On-chip debugging*, dua *flexible* Timer/Counter dengan mode pembandingan, interupsi internal dan eksternal, sebuah *programmable serial* USART, USI (*Universal Serial Interface*) dengan detektor kondisi awal, *Programmable Watchdog Timer* dengan osilator internal, dan tiga software *Selectable Mode Power-Saving*. Mode Idle berhenti sementara CPU membiarkan SRAM, Timer/Counter, SPI port dan sistem interupsi untuk berfungsi atau bekerja. Mode *Power-down* menyimpan isi register tapi membekukan osilator, *men-disable* semua fungsi IC sampai interupsi eksternal berikutnya atau hardware reset. Mode *standby* osilator tetap bekerja ketika IC dalam mode Sleep.

IC ini dibuat menggunakan teknologi memori *non-volatile Atmel's high-density*. *On-chip In-System Programmable Flash* mengizinkan memori program untuk diprogram ulang dalam sistem melalui sebuah SPI serial interface, melalui programmer konvensional *non-volatile memory programmer*. Dengan mengkombinasikan sebuah RISC 8-bit CPU dengan *In System Self-Programmable Flash* pada sebuah *monolithic chip*, Atmel ATtiny2313 sangat mendukung untuk alat yang memerlukan fleksibilitas yang tinggi dan efektif yang banyak diterapkan pada aplikasi control. ATtiny2313 AVR mendukung beberapa program dan *tools development system* termasuk: *C compilers*, *macro assemblers*, program debugger/simulator, *In Circuit Emulators* dan kit evaluasi.

### 2.3.3 Hardware IC

Hardware IC mikrokontroler AVR ATtiny2313 terdapat 3 buah port utama, vcc, ground, reset, dan 2 buah kristal yang berfungsi sebagai input dan output inverting osilator. Lebih jelasnya pada Gambar 2.3 di bawah ini:



b. PDIP/SOIC

Gambar 2.7 IC ATtiny2313



- VCC : Supply tegangan
- GND : Ground
- Port A (PA2..PA0) : Port A mempunyai 3-bit *bi-directional I/O port*. Pin port dapat diberi resistor pull-up internal secara individual. buffer output port A mempunyai karakteristik penggerak dengan kemampuan mencatu (source) dan menyerap (sink). Ketika pin PA0 sampai PA2 digunakan sebagai input dan di-pull low, maka akan menghasilkan sumber arus jika resistor internal *pull-up* diaktifkan. Pin port A mempunyai tiga keadaan ketika kondisi reset bahkan ketika clock tidak aktif.
- Port B (PB7..PB0) : Port B mempunyai 8-bit *bi-directional I/O port*. Pin port dapat diberi resistor *pull-up* internal secara individual. Buffer ouput port B mempunyai karakteristik penggerak dengan kemampuan mencatu (source) dan menyerap (sink). Ketika pin PB0 sampai PB7 digunakan sebagai input dan di-pull low, maka akan menghasilkan sumber arus jika resistor internal *pull-up* diaktifkan. Pin port B mempunyai tiga keadaan ketika kondisi reset bahkan ketika clock tidak aktif.
- Port D (PD6.PD0) : Port D mempunyai 7-bit *bi-directional I/O port*. Pin port dapat diberi resistor *pull-up* internal secara individual. Buffer output Port D mempunyai karakteristik penggerak

dengan kemampuan mencatu (source) dan menyerap (sink).

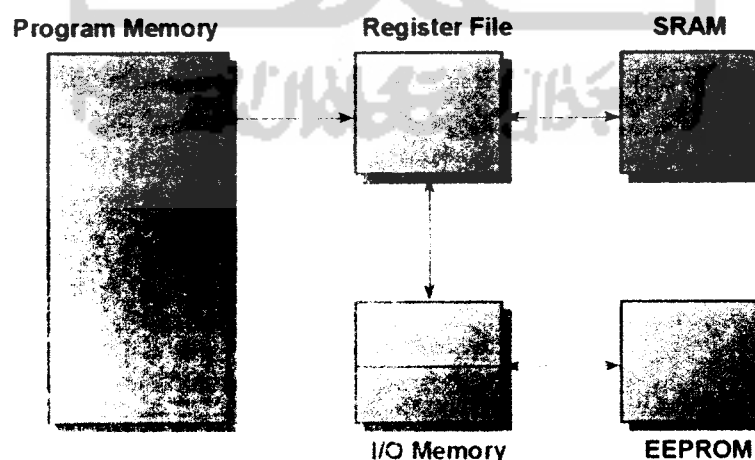
Ketika pin PD0 sampai PD7 digunakan sebagai input dan di-pull low, maka akan menghasilkan sumber arus jika resistor internal *pull-up* diaktifkan. Pin port D mempunyai tiga keadaan ketika kondisi reset bahkan ketika clock tidak aktif.

**RESET** : Input reset, ketika terjadi kondisi rendah pada pin ini akan membuat mikrokontroler masuk dalam kondisi reset walaupun clock mati/tidak berdetak.

**XTAL1** : Input bagi inverting osilator amplifier dan input bagi operasi rangkaian clock internal.

**XTAL2** : Output inverting oscillator amplifier.

#### 2.3.4 Organisasi Memori



Gambar 2.8 Hubungan antar memori

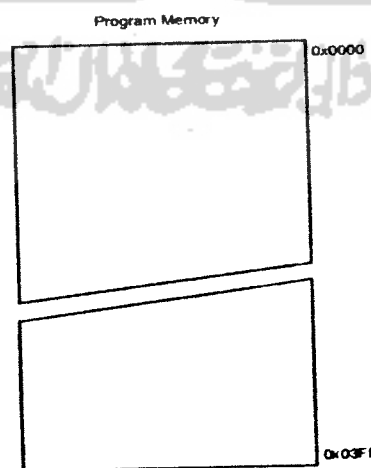
Seperti yang terlihat pada Gambar 2.4 program memori hanya dapat dibaca, seperti ditunjuk oleh tanda panah. Artinya program memori bertugas sebagai penyimpan kode program yang akan dieksekusi oleh ALU. Register file atau GPR (General Purpose Register) berhubungan secara langsung program memori, SRAM, dan I/O memori. Artinya kedudukan GPR sangat vital, karena hal inilah kenapa setiap instruksi harus melalui GPR. Untuk dapat berhubungan dengan memori EEPROM harus melalui I/O memori baru ke GPR.

Secara garis besar ATtiny2313 mempunyai dua memori utama, yaitu : memori data dan memori program. Memori EEPROM digunakan untuk memori data program *nonvolatile* (tidak akan hilang bila daya tidak ada).

A. Memori Data :

- 32 register keperluan umum (GPR) untuk seluruh operasi instruksi
- Register I/O untuk mengontrol mikrokontroler
- 128 x 8 SRAM untuk keperluan penyimpanan data sementara

B. Memori Program



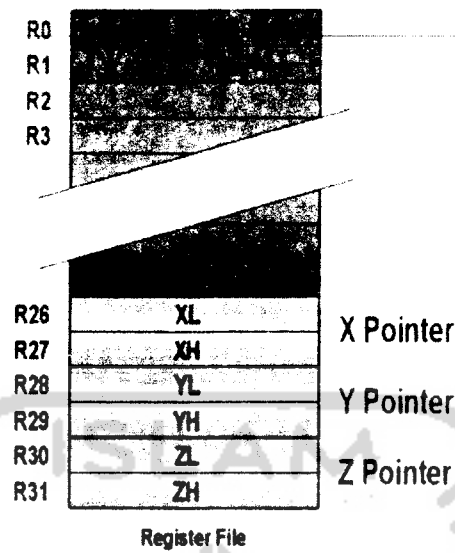
Gambar 2.9 Peta memori program

ATtiny2313 mengandung Flash memori 2kbyte On-chip untuk menyimpan program. Semenjak semua instruksi AVR mempunyai format 16 atau 32 bit, Flash dibentuk menjadi 1K x 16. Memori flash program sedikitnya mempunyai 10.000 siklus tulis/hapus. Lebar data PC (program counter) adalah 10 bit yang digunakan untuk mengamati 1Kx16 memori program. Tabel konstan dapat dialokasikan pada program memori.

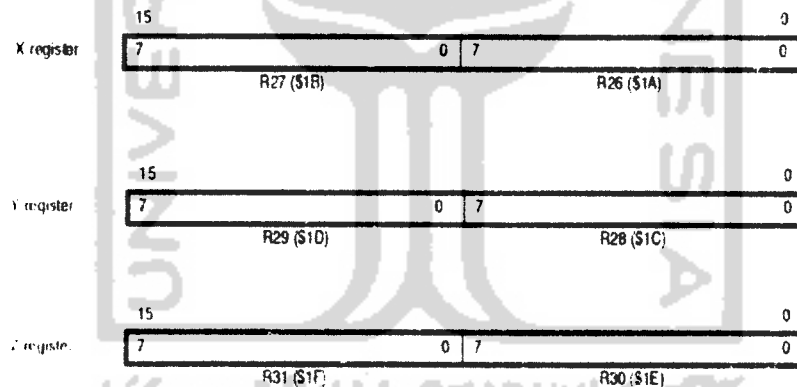
### C. *General Purpose Register (GPR)*

Seluruh set instruksi operasi register memiliki akses langsung dalam satu siklus ke semua register. Kecuali untuk lima instruksi aritmatika-logika yang mengoperasikan register dengan konstanta (SBCI, SUBI, CPI, ANDI, dan ORI) termasuk LDI yang mengoperasikan pemuatan data konstan imidiet, instruksi-instruksi tersebut dioperasikan hanya pada separo lokasi register terakhir GPR (R16.....R31). Instruksi untuk operasi umum seperti SBC, SUB, CP, AND, OR, dan operasi lainnya yang mengoperasikan dua register atau satu register dapat melakukan akses langsung terhadap seluruh register GPR.

Untuk pengalamatan tidak langsung atau pengalamatan ber-index disediakan enam register terakhir yaitu R26 sampai R31 dengan nama khusus yaitu register penunjuk X (R27:R26), register penunjuk Y(R29:R28), dan register penunjuk Z(R31:R30).



Gambar 2.10 General Purpose Register



Gambar 2.11 Register Pointer X, Y, Z

#### D. SRAM

ATtiny2313 mempunyai 224 (0x0DF) alamat lokasi memori. Kapasitas SRAM (Statik RAM) ATtiny2313 adalah 128 byte lokasi memori SRAM dalam peta memori AVR ATtiny2313 ditunjukkan seperti Gambar 2.8. Lokasi memori

terbawah adalah GPR, I/O memori kemudian baru SRAM. Lokasi memori pertama adalah GPR dan I/O memori, dan 128 lokasi berikutnya internal SRAM.

### Data Memory



Gambar 2.12 Peta memori data

Ada enam pengalamatan untuk memori data yaitu :

1. Mode langsung (*direct addressing*).
2. Mode tak langsung (*indirect addressing*).
3. Mode tak langsung dengan penggeseran (*indirect with Dis-palcement*).
4. Mode tak langsung dengan pengurangan-sebelum (*indirect with Pre-decrement*).
5. Mode tak langsung dengan penambahan-sesudah (*indirect with Post-increment*).

#### E. Register I/O

Register I/O menempati lokasi memori alamat \$20 hingga \$5F. Register ini berguna untuk mengatur dan melakukan kontrol terhadap peripheral mikrokontroler.

Lokasi I/O diakses dengan instruksi transfer data “IN” dan “OUT” antara 32 GPR dan ruang I/O. Register I/O menempati alamat \$00 - \$1F yang dapat diakses langsung secara bit menggunakan instruksi SBI dan CBI. Dalam register I/O, nilai bit (single) suatu register dapat dicek dengan instruksi SBIS dan SBIC. Ketika menggunakan perintah IN dan OUT, maka alamat I/O yang digunakan harus dalam ruang \$00 - \$3F. Ketika pengaksesan menggunakan SRAM, nilai \$20 harus kita tambahkan pada alamat ini. Diantara register I/O, register yang paling banyak digunakan diantaranya SREG dan SP (Stack Pointer).

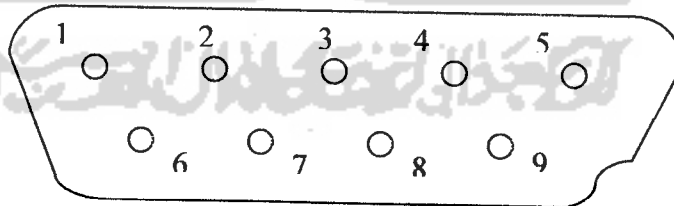
SREG (Status Register) ini berisi status yang dihasilkan setiap operasi yang dilakukan oleh ALU ketika suatu instruksi dieksekusi. Register ini biasanya sering dipakai dalam instruksi percabangan bersyarat.

Sedangkan Stack Pointer (SP) adalah sebuah register 8-bit pada I/O address \$3E(\$5E) – \$3D(\$5D). 8-bit digunakan untuk memasukkan alamat awal SRAM yang akan dijadikan stack. Stack pointer menunjuk pada area data stack SRAM dimana Subrutin dan Stack Interupsi ditempatkan. Area stack pada data SRAM harus didefinisikan oleh program sebelum memanggil beberapa subrutin dieksekusi atau meng-*enable* interupsi. Stack Pointer harus diset pada alamat SRAM diatas \$60. Stack Pointer di-*decrement* (dikurangi) satu ketika data di-push (didorong masuk) kedalam stack dengan instruksi “PUSH”, dan dikurangi (decrement) dua ketika alamat di-push kedalam stack dengan subrutin panggilan (subrutin calls) dan interupsi. Stack Pointer di-incremen (ditambah) satu ketika data di-pop (diambil) dari stack dengan instruksi “POP”, dan ditambah (incremen)

dua ketika address/alamat di-pop dari stack dengan selesainya subrutin (akhir subrutin "RET") atau selesainya interupsi RETI.

#### 2.4 Sinyal Port Serial dan Fungsi Pin

Port serial terdiri atas dua tipe sinyal, yaitu data sinyal dan *control* sinyal, dan port serial juga mempunyai dua macam konektor yaitu konektor DB-25 dan DB-9, keduanya dalam bentuk jantan (*male*) pada personal computer. Walaupun demikian sebagian besar PC menggunakan konektor DB-9 karena lebih mudah untuk mengaplikasikannya dibanding menggunakan konektor DB-25. Apabila menggunakan konektor DB-25 maka akan banyak pin yang tidak dipakai/kosong. Pada kenyataannya hanya 3 pin yang dapat digunakan sebagai serial komunikasi, satu untuk menerima data, satu untuk mengirim data, dan yang satunya sebagai sinyal *ground* seperti yang ditunjukkan pada Gambar 2.9 Skematik dari DB-9 (*male*).



Gambar 2.13 Skematik dari DB-9 (*male*)



Tabel 2.1 Pin port serial dan tipe sinyal

Pin	Label	Nama sinyal	Tipe sinyal
1	CD	Carrier Detect	Control
2	RD	Received Data	Data
3	TD	Transmitted Data	Data
4	DTR	Data Terminal Ready	Control
5	GND	Signal Ground	Ground
6	DSR	Data Ser Ready	Control
7	RTS	Request to Send	Control
8	CTS	Clear to Send	Control
9	RI	Ring Indikator	Control

Dari 9 pin atau kaki port serial di atas, masing-masing mempunyai fungsi antara lain:

1. TD untuk pengiriman data serial (TXD).
2. RD untuk penerimaan data serial (RXD).
3. CTS digunakan untuk memberitahukan bahwa modem siap untuk melakukan pertukaran data.
4. DCD saat modem mendeteksi suatu 'carrier' dari modem lain (di tempat lain) maka sinyal ini akan diaktifkan.
5. DSR memberitahukan UART bahwa modem siap melakukan hubungan komunikasi (link).
6. DTR kebalikan dari DSR, untuk memberitahukan bahwa UART siap melakukan hubungan komunikasi.
7. RTS sinyal untuk menginformasikan modem bahwa UART siap melakukan pertukaran data.

8. RI akan aktif jika modem mendeteksi adanya sinyal dering dari saluran telepon.

#### 2.4.1 Perangkat lunak *Borland Delphi*

Perkembangan bahasa pemrograman berlangsung dengan pesat, mulai dari bahasa tingkat rendah (yang lebih dekat dengan *hardware*) seperti bahasa *assembly*, ataupun bahasa tingkat tinggi, seperti bahasa C dan Pascal. Pada bahasa ini agak susah mengembangkannya dan membuat tampilannya menjadi menarik, apalagi bila menggunakan bahasa pemrograman ini dalam bentuk grafik. Perkembangan bahasa pemrograman terus berlanjut dengan dikeluarkannya Windows oleh pihak Microsoft. Pada Windows memperkenalkan model OOP (*Object Oriented Programming*) yaitu yang lebih menampilkan kemudahan dan tampilan yang menarik (seperti windows-windows pada aplikasi yang sering digunakan pada *MS Word*). Delphi merupakan salah satu program yang berbasis pada OOP, jadi dengan bahasa Delphi dapat dibuat program-program yang menarik untuk dilihat dan fleksibel serta *user friendly*.

##### A. Form *Borland Delphi*

Berbeda dengan Pascal, pada Delphi dikenal OOP (*Object oriented programming*), Jadi bila pada Pascal tampilannya menjemukan, pada Delphi tampilannya dapat diatur semenarik mungkin pada *form* yang digunakan. Untuk mengatur tampilan *form* caranya cukup mudah, hanya dengan menaruh komponen-komponen yang diinginkan pada *form* tersebut, dan mengfungsikan masing-masing komponen sesuai dengan yang diinginkan.

## B. Unit *borland delphi*

Setiap perubahan pada *form* akan berakibat perubahan pada unit. Untuk pindah dari *form* ke unit, dapat dilakukan dengan menekan tombol F12. Berikut ini adalah bentuk unit yang diberikan Delphi saat pertama kali membuka sebuah *form* :

```

Unit Unit1;

Interface

Uses
  Windows, Messages, Sys Utils, Classes, Graphics, Controls, Forms,
  dialogs;

Type

TForm1 = class(tForm)

Private
  { private declarations }

Public
  { Public declarations }

end;

var

Form1 : TForm1;

Implementation

{SR * .DFM}

end.

```

### 2.4.2 Komponen *Borland delphi*

Dalam membuat program, Delphi telah menyediakan banyak kemudahan, yaitu dengan disediakannya komponen-komponen. Komponen ini merupakan sebuah prosedur/program yang sudah dikompilasi dan langsung dapat digunakan, sesuai dengan fungsinya masing-masing. Untuk menggunakan komponen ini dengan meng-klik komponen yang diinginkan, kemudian klik di *form*, maka komponen tersebut akan muncul di *form*.

Kegunaan dari beberapa komponen adalah sebagai berikut:

a. *Button / Bitbtn*

Komponen ini biasa digunakan sebagai tombol kendali. Perbedaan antara *bitbtn* dengan *btn* adalah : *bitbtn* dapat menyisipkan warna pada tombol dan *icon* tertentu, lain halnya dengan menggunakan *btn*.

b. *Panel*

Panel berfungsi untuk mengelompokkan komponen-komponen didalamnya.

c. *Label*

User dapat menamakan atau memberi keterangan pada program.

d. *Edit*

*Edit* berfungsi sebagai masukan data (*input*) dalam bentuk *string*, dari bentuk *string* dapat diubah menjadi bentuk *integer* atau bentuk lainnya. Yang kemudian dapat digunakan untuk operasi selanjutnya.

e. *Chart*

Dengan *chart* data-data yang telah dianalisa, dapat ditampilkan kedalam grafik, sehingga memudahkan untuk menganalisanya.

f. *Stringgrid*

*Stringgrid* berguna untuk menaruh data *string* kedalam bentuk kolom tabel, seperti pada Excel.

g. *Popup Menu*

*Popup menu* berfungsi sebagai perintah yang aktif bila meng-klik kanan *mouse*, Untuk mengaktifkannya harus mengaktifkan *popup menu* pada komponen yang diinginkan, caranya : ubah pada *object inspector*.

h. *Main menu*

Contoh *main menu* adalah *option* pada tiap aplikasi program, dengan komponen ini, fungsi-fungsi program dapat ditaruh seperti pada aplikasi umumnya.

i. *Combo Box*

*Combo Box* berfungsi sebagai petunjuk untuk pemilihan berbagai masukan.

j. *Check Box*

Bila komponen ini di *check* maka ada aplikasi yang bisa disetting untuk bekerja dibawahnya.

k. *Radio Button*

Prinsip kerjanya hampir sama dengan *check box*, Cuma tampilannya saja yang berbeda.

### 2.4.3 Object inspector

*Object inspector* berguna sebagai menu pilihan dari masing-masing komponen. Dengan *object inspector* komponen-komponen yang digunakan dapat dimanipulasi.

### 2.4.4 Komponen serial delphi

Ada beberapa metode yang digunakan untuk mengakses data melalui port serial menggunakan Delphi salah satunya adalah dengan menggunakan komponen ComPort. Pada menu komponen standar yang digunakan pada Delphi 7 tidak terdapat komponen ComPort sehingga untuk dapat menggunakan komponen tersebut terlebih dahulu komponen ComPort diinstallkan ke Delphi 7. Cara menginstall komponen ComPort ke Delphi 7 adalah sebagai berikut:

1. Buat nama folder baru didalam folder Delphi 7, kemudian ekstrak file ComPort. Zip ke folder baru. File ComPort.Zip dapat di download pada [www.sourceforge.net/projects/comport](http://www.sourceforge.net/projects/comport)
2. Klik menu komponen kemudian klik install packages kemudian buka file CportLib7.dpk, kemudian klik tombol "compile".
3. Kemudian buka file DsgnCPort7.dpk, kemudian klik tombol "compile" setelah dicompile kemudian klik tombol install.

## 2.5 Motor Servo

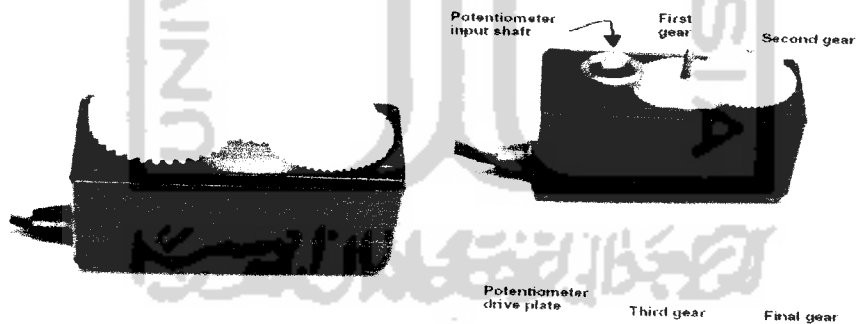
Jenis motor ini banyak digunakan para penggemar mainan Radio control/RC, sreperti pesawat terbang model, perahu, mobil dan juga sering

## 2.8 Kinerja binary ASK

Kinerja pada sistem komunikasi banyak ditentukan oleh kemampuan bagian penerima dalam mengolah kembali sinyal yang diterima dari pengirim. Bagaimanapun kecanggihan sistem pada pengirim tidak akan memberi arti apabila penerima tidak mampu menterjemahkan pesan yang disampaikan melalui gelombang *carrier*. Sehingga kemampuan penerima untuk mengolah informasi yang terdapat pada gelombang *carrier* akan menentukan sejauh mana kemampuan sistem komunikasi tersebut. Dari seluruh pesan dalam hal ini bit atau bisa saja simbol yang dikirim pemancar berapa bagian yang diterjemahkan secara benar oleh penerima dan berapa bagian yang salah. Bagian yang salah dalam satu satuan waktu (detik) disebut sebagai *prosen error* per detik atau lebih dikenal sebagai *bit error rate* (BER). Seberapa besar bagian yang salah dari seluruh bit yang diterima disebut sebagai *prosen error*. Kemungkinan kesalahan yang terjadi dari sejumlah bit yang dikirim lebih dikenal sebagai *probability of error* atau disingkat  $P_e$ . Nilai  $P_e$  ini selanjutnya digunakan sebagai salah satu kriteria dari kinerja sistem komunikasi digital. Semakin kecil nilai  $P_e$  maka semakin baik kinerja yang dimiliki sistem komunikasi tersebut.

digunakan pada robot. Dengan menggunakan motor servo pengendalian posisi mainan tersebut lebih presisi sehingga memudahkan pengguna dalam memakainya.

Motor servo adalah motor khusus yang disertai dengan rangkaian tambahan atau feedback posisi untuk menentukan posisi sumbu motor servo yang diinginkan. Rangkaian feedback posisi tersebut berupa potensio meter yang dapat berputar sesuai dengan putaran gir utama motor servo. Rangkaian feedback akan terus membandingkan antara posisi motor servo yang diperoleh dari perubahan nilai hambatan pada potensio meter dengan pulsa masukan. Pada motor servo terdapat tiga buah kabel berwarna, kabel warna merah digunakan untuk catu daya (tegangan 4.8 - 6 Vdc), kabel warna hitam untuk ground dan warna putih untuk data.



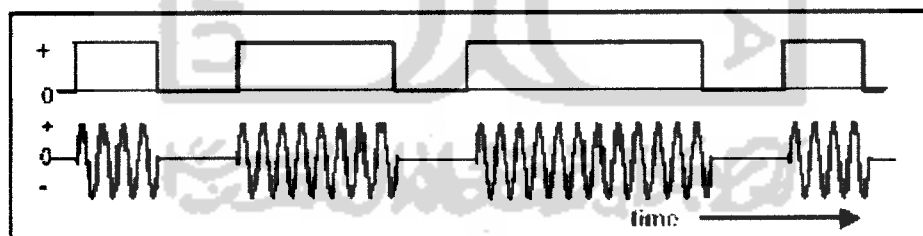
Gambar 2.14 Motor servo

## 2.6 *Amplitude Shift Keying (ASK)*

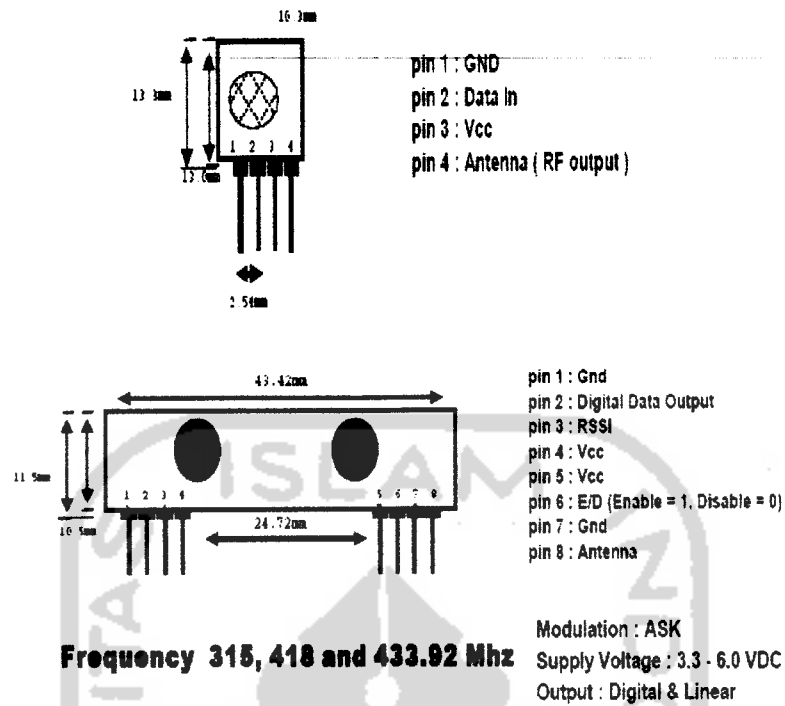
*Amplitude shift keying (ASK)* adalah sebuah komunikasi wireless (tanpa kabel) yang beroperasi dalam pita frekuensi 433 MHz dengan jarak jangkauan



$\pm 100\text{m}$ . ASK terdiri dari ASK *transmitter* (pengirim) dan ASK *receiver* (penerima), untuk penerimanya sendiri menggunakan ASK jenis RLP433 dan untuk pengirimnya menggunakan ASK jenis TLP433. ASK merupakan teknik pembangkitan gelombang AM yang dilakukan dengan dua pendekatan berbeda, yang pertama adalah dengan membangkitkan sinyal AM secara langsung tanpa harus dengan membentuk sinyal *base band* yang secara langsung menyiratkan arti sebuah terminologi yang menggambarkan suatu teknik modulasi digital. Yang kedua dengan menggunakan sinyal *base band* untuk memodulasi amplitudo suatu sinyal carier yang dalam hal ini merupakan sinyal sinusoida (baik *cos* atau *sinus*), seringkali ini dikenal sebagai AM analog dengan informasi dalam bentuk digital. Kedua teknik ini merupakan pembangkitan gelombang AM untuk mentransmisi informasi digital, untuk selanjutnya keduanya dikenal sebagai dua bentuk pembangkitan ASK atau lebih dikenal sebagai AM digital.



Gambar 2.15 Sinyal ASK



Gambar 2.16 Bentuk ASL TLP433

## 2.7 Binary ASK

Sinyal *base band* yang ditransmisikan memiliki dua kemungkinan nilai informasi yaitu antara nol (0) dan satu (1). Karena kemungkinan nilai informasinya tersusun dari dua keadan tersebut maka selanjutnya sistem ini dikenal dengan binary ASK atau BASK.

Bentuk sinyal termulasi dalam hal ini dapat didekati dengan sebuah persamaan berikut:

$$V(t) = V_c / 2 [1 + m v_m(t)] \cos(2\omega_c t) \dots\dots\dots (2.1)$$

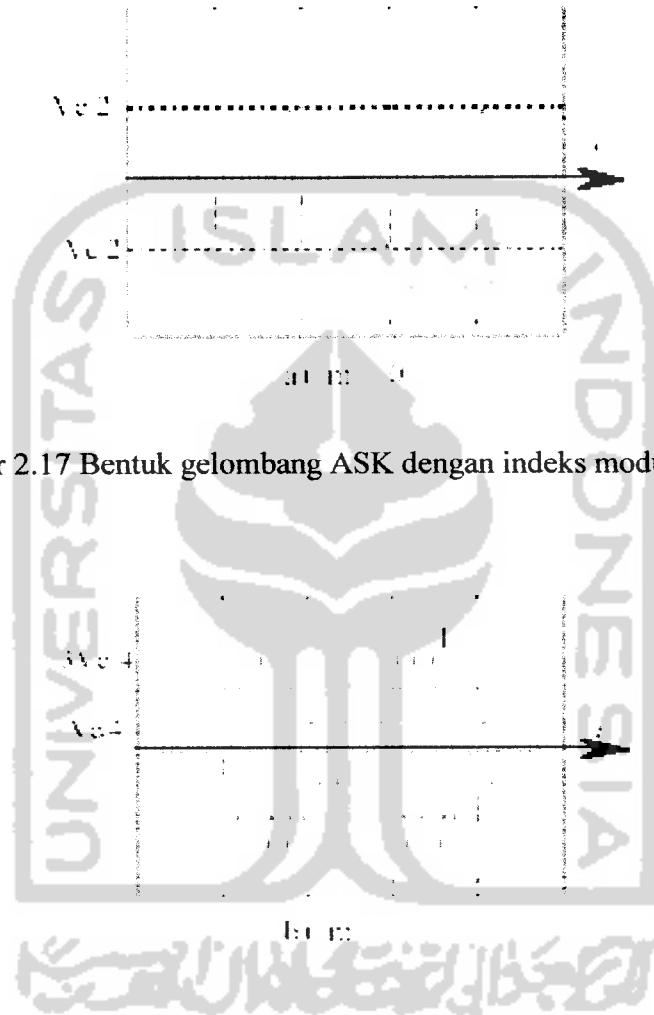
dimana:

$V_c$  = amplitude sinyal carier

$V_m$  = sinyal pemodulasi yang bernilai 1 atau 0

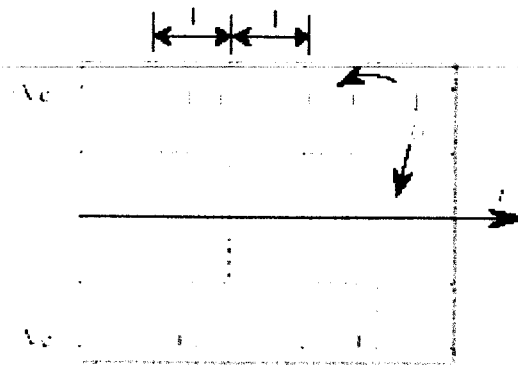
$m$  = indek modulasi

$\omega_c = 2\pi f_c$  = frekuensi carrier dalam nilai gradian



Gambar 2.17 Bentuk gelombang ASK dengan indeks modulasi  $m = 0$

Gambar 2.18 Bentuk gelombang ASK dengan indeks modulasi  $m = \frac{1}{2}$



Gambar 2.19

Gambar 2.19 Bentuk gelombang ASK dengan indeks modulasi  $m = 1$

Dua bentuk sinyal akan dapat dihasilkan, dengan nilai  $V_m(t) = 0$  atau 1 untuk mengirimkan nilai informasi biner nol (0) atau satu (1). Dalam hal ini  $V_m(t)$  bisa juga bernilai 1 atau -1, sehingga dapat dipertimbangkan sebagai data bipolar ternormalisasi. Indeks modulasi ( $m$ ) dapat bernilai  $0 \leq m \leq 1$ . Untuk indeks modulasi  $m = 0$  maka akan dikirimkan sebuah sinusoida murni seperti pada Gambar 2.13. Jika  $m$  dipilih dengan nilai  $\frac{1}{2}$  maka akan dikirimkan sebuah sinusoida yang memiliki dua nilai berbeda, amplitudo  $V_c/4$  untuk nilai informasi 0 dan amplitudo  $3V_c/4$  untuk nilai informasi 1. Untuk lebih jelasnya dapat dilihat pada Gambar 2.14.

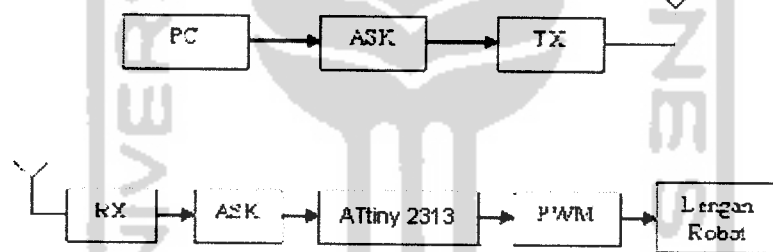
Untuk kasus  $m = 1$  dimana merupakan indeks modulasi yang sering digunakan. Dengan indeks modulasi ini maka akan dikirimkan sinyal beramplitudo nol untuk nilai biner nol (0) dan sinyal beramplitudo  $V_c$  untuk nilai biner satu (1), hal ini dikenal sebagai *on-off keying* (OOK) dan diperlihatkan pada Gambar 2.15.

## BAB III

### PERANCANGAN SISTEM

#### 3.1 Perancangan Sistem

Dalam perancangan sistem pengendalian sebuah lengan robot, maka dapat dirancang sebuah sistem pengendalian dengan jarak jauh menggunakan komputer (PC), dengan memanfaatkan ASK sebagai sistem pemancarnya menggunakan mikrokontroler ATtiny2313 sebagai pembangkit pulsa.



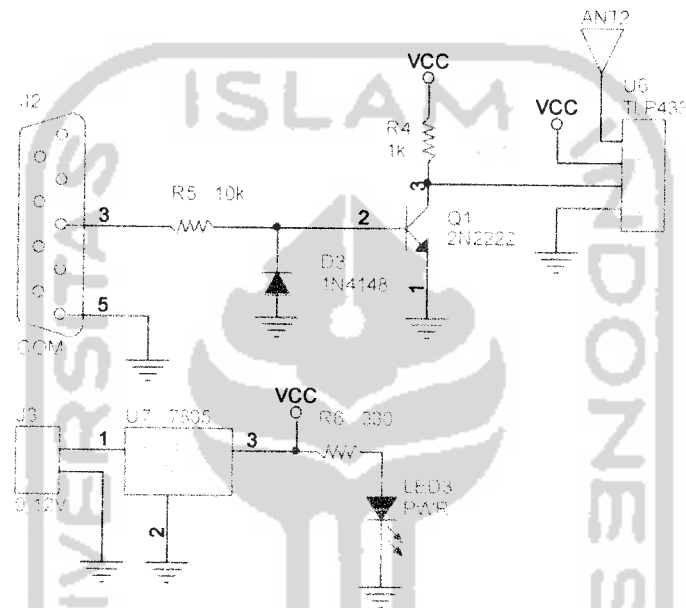
Gambar 3.1. Diagram blok pengendali lengan robot

#### 3.2 Perancangan Perangkat Keras

##### 3.2.1 Rangkaian Pemancar Data

PC (Personal Computer) yaitu blok ini berfungsi sebagai interface (antar muka) antar pengguna computer/PC sebagai pengendali. Disini semua proses komputasi dieksekusi, terutama perhitungan *forward* dan *invers kinematik manipulator*. Setelah semua data yang diperlukan diproses, maka computer/PC

akan mengirimkan /mentrasfer ke ASK pengiriman data dilakukan dengan jalan komunikasi serial. Amplitude Shift Keying (ASK) yaitu berupa dua frekuensi sinyal pembawa biasanya salah satu dari amplitudo adalah nol. Transmitter (TX) yaitu mengirimkan sinyal yang telah selesai diproses oleh PC melalui sinyal ASK.

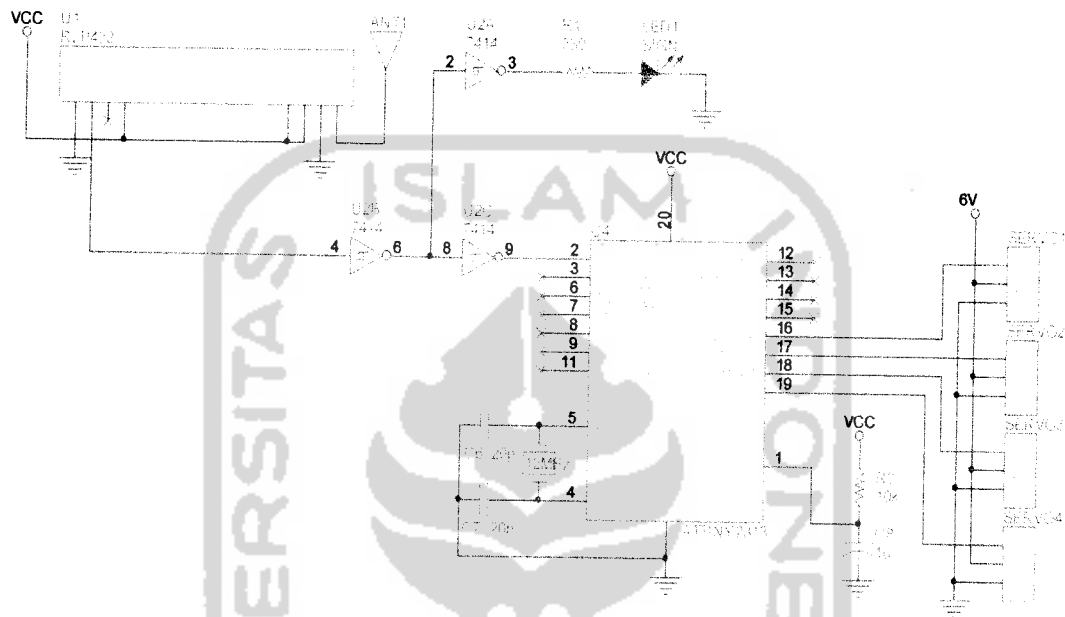


Gambar 3.2 Rangkaian Pemancar ASK

### 3.2.2 Rangkaian Penerima data

*Recivier* (RX) menerima sinyal sesuai dengan perintah yang dikirimkan oleh rangkaian pemancar ASK. *Driver* digunakan untuk menggerakkan motor servo pada lengan robot, dengan memanfaatkan mikrikontroler ATtiny2313 yang digunakan sebagai pengubah data serial yang diterima dari komputer /PC menjadi data paralel dan sekaligus sebagai pemroses data komputer agar bisa diterima

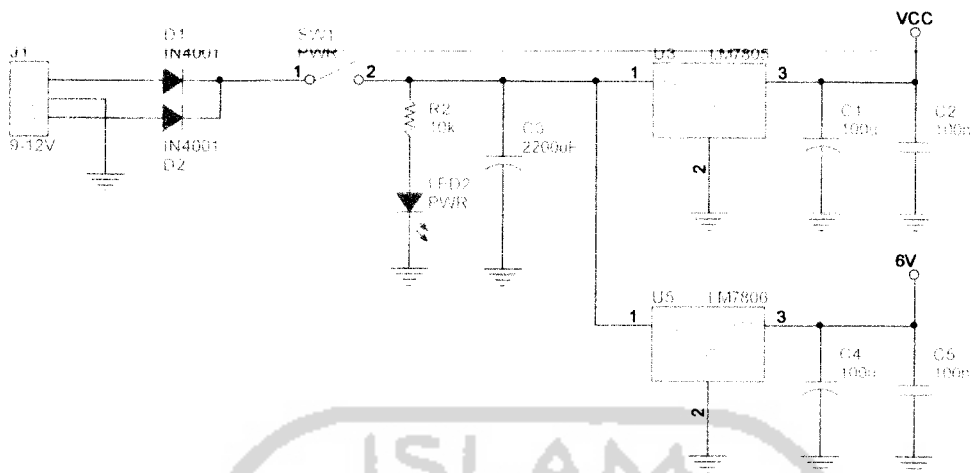
oleh motor servo. Banyak port output pada mikrokontroler tergantung pada banyaknya motor servo yang digunakan pada lengan robot.



Gambar 3.3 Rangkaian penerima ASK

### 3.2.3 Power Supply

Motor servo memerlukan sumber tegangan yang sangat stabil. Dalam hal ini digunakan *switching power supply* yang berasal dari *power supply* komputer. *Power supply* ini mempunyai beberapa *range* tegangan antara lain 5Vdc dan 12Vdc. Motor servo memerlukan tegangan antara 4,8Vdc sampai 6Vdc, sehingga yang dapat langsung digunakan adalah 5Vdc, walaupun hal tersebut tidak bisa memberikan *torsi* yang maksimal pada servo. *Torsi* maksimal akan didapatkan dengan memberikan tegangan input 6V.



Gambar 3.4 Rangkaian catu daya

### 3.2.4 Mikrokontroler

Mikrokontroler adalah suatu terobosan teknologi mikroprosesor dan mikrokomputer, yang mana teknologi ini adalah teknologi semikonduktor dengan kandungan transistor yang lebih banyak, namun hanya membutuhkan ruang yang kecil serta dapat diproduksi secara massal (dalam jumlah banyak) sehingga harganya menjadi lebih murah.

Mikrokontroler ini kemampuan digitalnya menirukan fungsi otak manusia, sehingga meliputi fungsi atau instruksi aritmatika (berhitung), logika (mempertimbangkan suatu kondisi), dan memori. Mikrokontroler ini berbeda halnya dengan mikroprosesor yang hanya pemrosesannya terdiri dari Central Processing Unit (CPU) dan register-register, tanpa memori, tanpa I/O, dan priferal yang dibutuhkan oleh suatu sistem supaya dapat bekerja. Namun apabila mikroprosesor ini dikombinasikan dengan I/O dan memori (RAM dan ROM) akan



menghasilkan sebuah mikrokontroler, yang mana kombinasi dari komponen-komponen ini sudah terdapat dalam satu *chip Integrated Circuit* (IC).

### 3.3 Kerangka Lengan Robot

#### A. Bahan utama Robot

Akrilik dengan ketebalan 3 mm

#### B. Dimensi lengan robot :

- Banyak Joint : 4 buah
- Banyak lengan : 4 buah
- Panjang lengan pertama : 13 cm
- Panjang lengan ke dua : 10 cm
- Panjang lengan ke tiga : 10 cm
- Panjang lengan ke empat : 9 cm

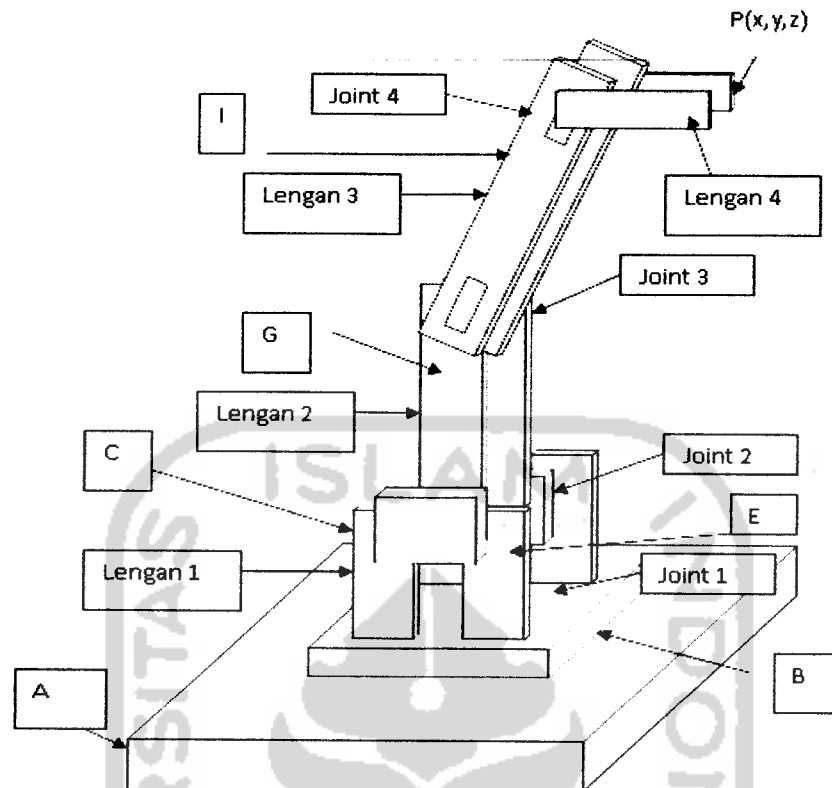
Mekanik robot disusun oleh beberapa bagian:

#### 1 Bagian A

Bagian ini merupakan landasan dasar robot.

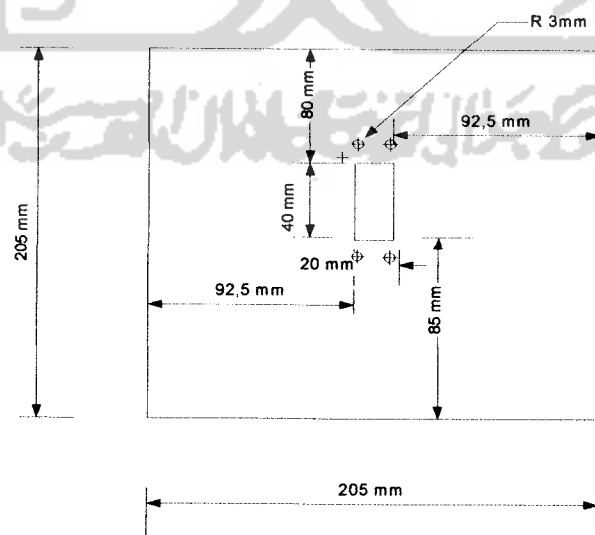
#### 2. Bagian B

Dijadikan dudukan untuk bagian kerangka tangan Robot yang lain (bagian C, D, E, F) yang merupakan bagian dari lengan pertama. Joint 1 terusan dan bagian A dan B yang terhubung.

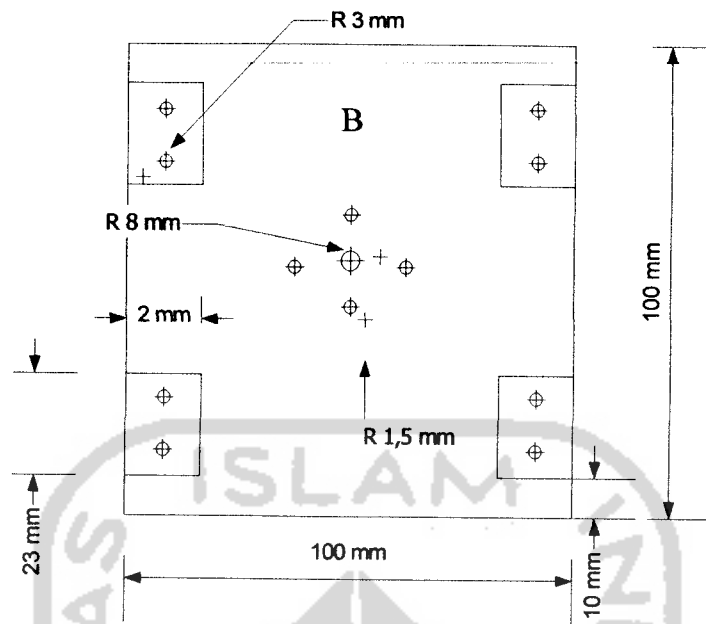


Gambar 3.5 Arsitektur lengan robot

Ukuran dimensi kerangka tangan robot :

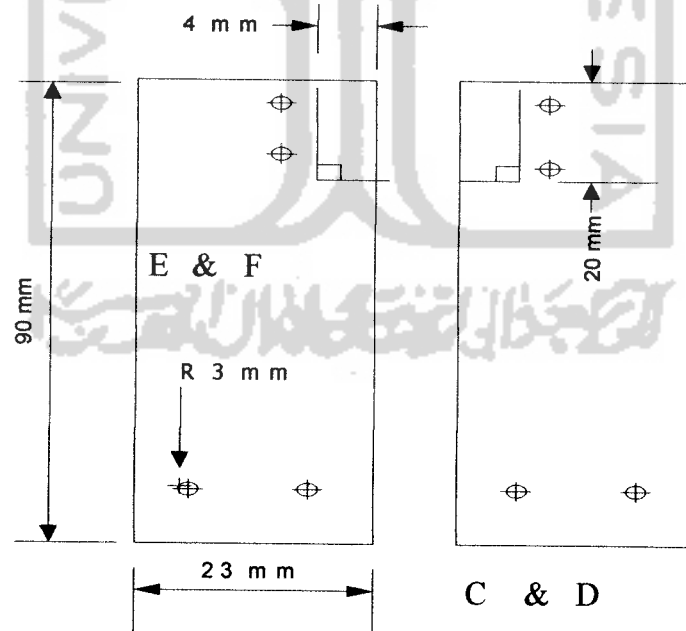


Gambar 3.6 Bagian A



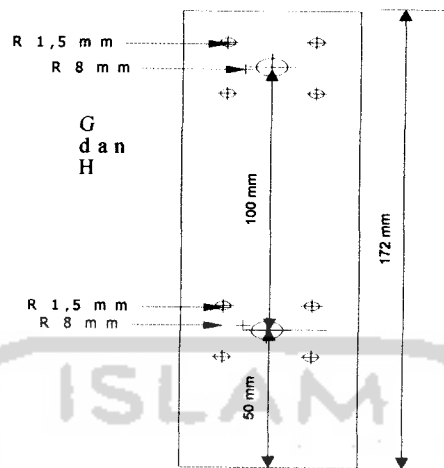
Gambar 3.7. Bagian B

## 3. Bagian C, D, E, dan F



Gambar 3.8 Bagian C,D,E, dan F

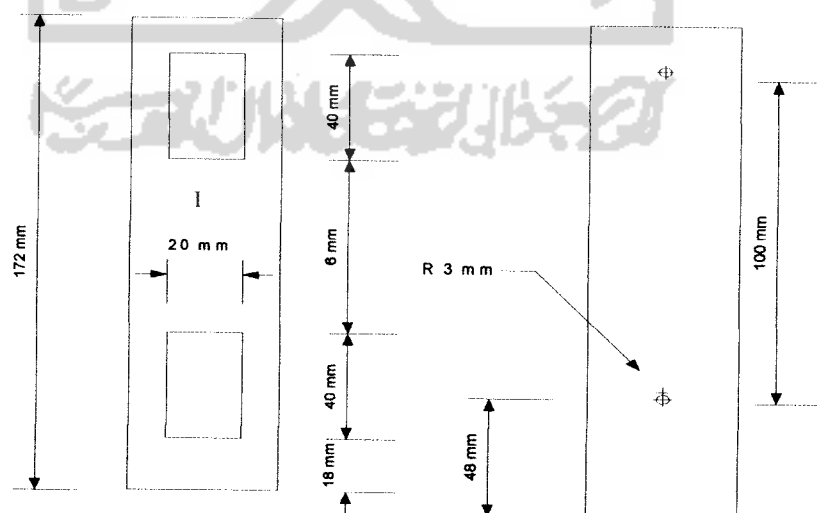
#### 4. Bagian G dan H



Gambar 3.9 Bagian G, dan H

#### 5. Bagian I dan J

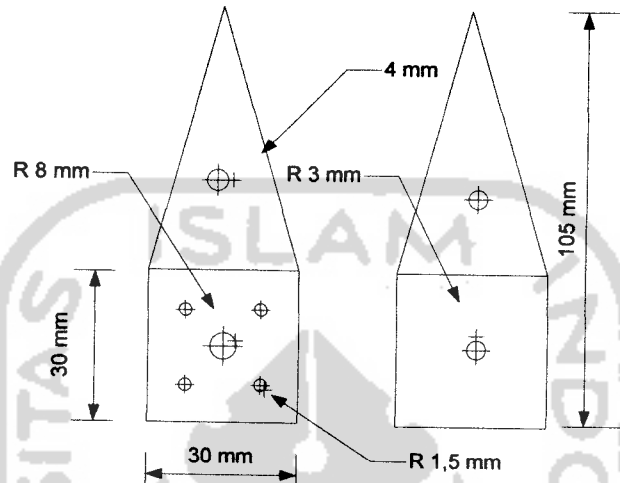
Bagian G, H, I, dan J mempunyai panjang yang sama, namun berbeda penggunaannya, pada bagian I tangan dilubangi agar bisa digunakan untuk masukan bodi servo kebagian 1. Hal ini dapat membuat lengan 4 lebih panjang.



Gambar 3.10 Bagian I, dan J

## 6. Bagian K, dan L

Bagian ini merupakan ujung *manipulator* robot



Gambar 3.11 Bagian K dan L

Lengan robot adalah salah satu jenis robot yang hanya terdiri dari sebuah lengan robot atau lebih. Penggunaan dari lengan robot ini sangat banyak dan beragam, baik dalam dunia pendidikan maupun dunia industri, terutama untuk mengerjakan pekerjaan yang tidak mungkin dikerjakan manusia. Fungsi utama lengan robot ini untuk menggantikan tangan manusia di beberapa tempat yang tidak mungkin dijangkau manusia, daerah berbahaya ataupun memerlukan kondisi tertentu, misalnya lingkungan yang panas, ada zat beracun ataupun kegiatan perlu daya angkat besar, sehingga lengan robot ini harus fleksibel dan dapat berfungsi seperti tangan manusia.

### 3.4 Pengendalian Robot

Pengendalian robot umumnya dilakukan pada joint-jointnya (*joint space*). Sedangkan menggerakkan robot adalah bagaimana memindahkan posisi *end effector* menuju suatu posisi mengikuti pada lintasan/trayektori. Data acuan tersebut memanfaatkan untuk membangkitkan sinyal control sudut joint (trayektori joint), melalui mekanisme invers kinematik, yaitu mekanisme untuk mendapatkan nilai-nilai sudut joint dari posisi dan orientasi *end effector* yang telah diketahui. Penyelesaian ini akan mentransformasikan gerakan manipulator dari *operational space* menjadi *joint space*. Invers kinematik dapat diselesaikan dengan banyak cara seperti invers *transform*, *screw algebra*, *dual matriks*, *dual quaternion*, *intractive*, dan pendekatan *geometri*.

#### 3.4.1 Forward Kinematic

Untuk mencari *Forward kinematic* digunakan persamaan Geometri, dan persamaan matriks transformasi. Analisis secara matematika geometri *forward kinematic* yaitu :

$$\text{Sumbu } x = \cos \theta_1 \{L_2 \cos (\theta_2) + L_3 \cos (\theta_2 + \theta_3)\} \dots\dots\dots (3.1)$$

$$\text{Sumbu } y = \sin \theta_1 \{L_2 \cos (\theta_2) + L_3 \cos (\theta_2 + \theta_3)\} \dots\dots\dots (3.2)$$

$$\text{Sumbu } z = L_1 + \{L_2 \sin (\theta_2)\} + \{L_3 \sin (\theta_2 + \theta_3)\} \dots\dots\dots (3.3)$$

Keterangan :

$L_1$  = Panjang lengan pertama

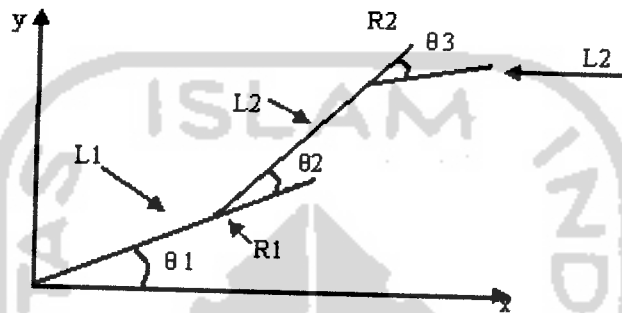
$L_2$  = Panjang lengan ke dua

$L_3$  = Panjang lengan ke tiga

$\theta_1$  = Besar sudut yang dihasilkan dari putaran sumbu z pada lengan pertama.

$\theta_2$  = Besar sudut yang dihasilkan antara lengan ke dua dengan sumbu x positif.

$\theta_3$  = Besar sudut yang dihasilkan antara lengan 3 dan lengan 2



Gambar 3.12 *Forward kinematic*

### 3.4.2 *Invers Kinematic*

Yaitu masukan yang berupa posisi sudut dan orientasi menggerakkan joint sehingga membentuk sudut tertentu. Dalam pemakaiannya invers kinematic lebih sulit, dikarenakan harus menyelesaikan persamaan non linear simulation.

$$\theta_2 = \cos^{-1} \frac{X^2 + Y^2 - 2L_3(X \cos \theta_1 - Y \sin \theta_1) + L_1^2 + L_2^2 + L_3^2}{2L_1L_2} \dots\dots\dots (3.4)$$

$$\theta_1 = \text{tg}^{-1} \frac{(YL_3 \sin \theta_2)(L_1 + L_2 \cos \theta_2) - (X - L_3 \cos \theta_2)L_2 \sin \theta_2}{(X - L_3 \cos \theta_2)(L_1 + L_2 \cos \theta_2) + (Y - L_3 \sin \theta_2)L_2 \sin \theta_2} \dots\dots\dots (3.5)$$

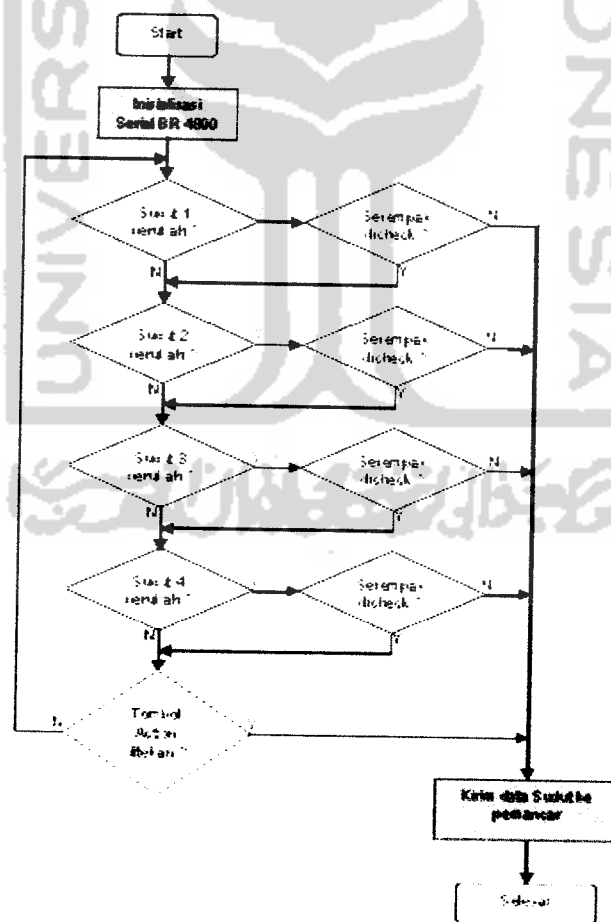
$$\theta_3 = \psi - (\theta_1 + \theta_2) \dots\dots\dots (3.6)$$

$$\psi = \text{tg}^{-1} \frac{Y}{X} \dots\dots\dots (3.7)$$

### 3.5. Pergerakan robot.

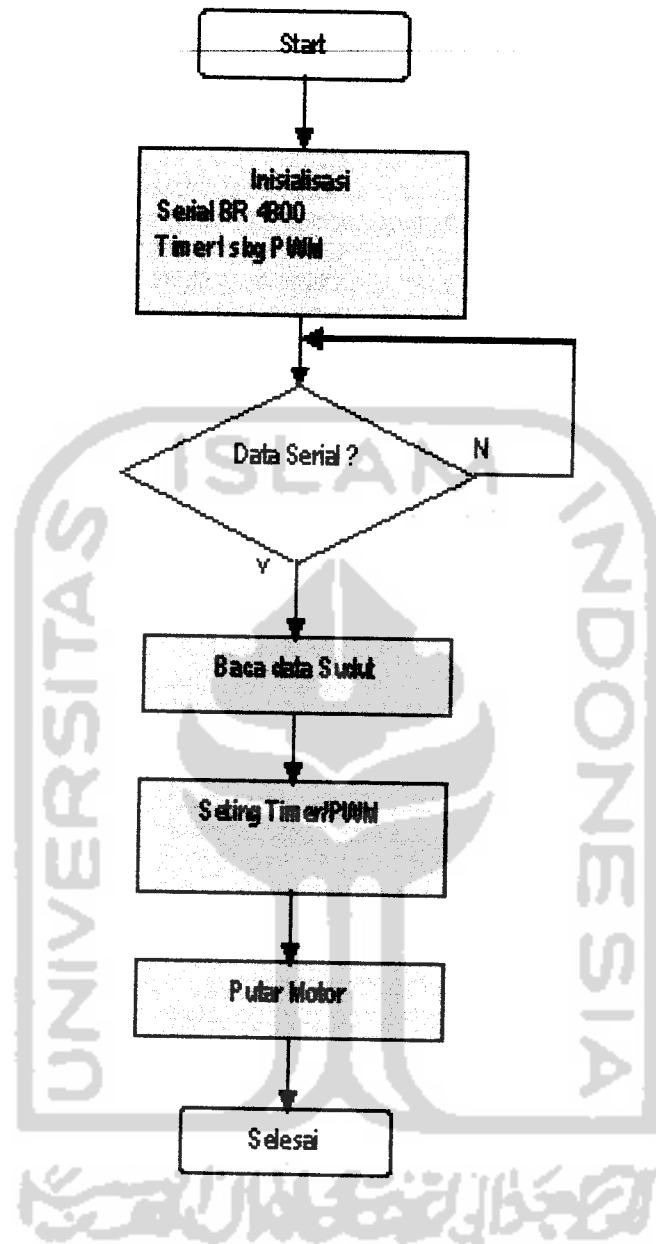
Masukan dibatasi untuk tiap-tiap joint. Pada joint 1 dibatasi  $0^\circ - 180^\circ$ , joint 2 antara  $0^\circ - 110^\circ$ , joint 3 antara  $-90^\circ - 100^\circ$  dan joint 4 antara  $-90^\circ - 100^\circ$ . Robot mempunyai *zero positioning*. Agar tidak jatuh, maka PWM harus selalu diberikan ke servo, untuk servo HS 564 MG, cukup diberikan tegangan 0 dan +5V, servo akan mempunyai daya (torsi) tahan yang cukup untuk menahan beban, namun untuk servo yang lainnya, PWM harus terus diumpankan ke servo agar servo mempunyai torsi tahan yang cukup untuk menahan beban yang diberikan.

Jalannya program pada mikrokontroler akan berlangsung secara terus-menerus sampai proses dihentikan seperti pada Gambar 3.13 dan Gambar 3.14.



Gambar 3.13 Flowchart delphi





Gambar 3.14 *Flowchart* mikrokontroler

## BAB IV

### ANALISA DAN PEMBAHASAN

#### 4.1 Pengujian

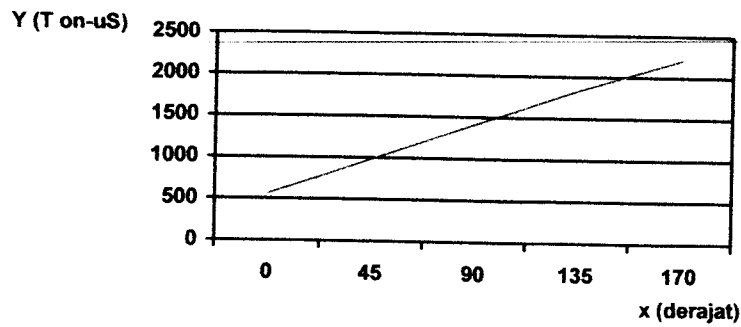
Pengujian alat meliputi pengujian kemampuan lengan robot untuk bergerak, baik untuk perangkat keras ataupun perangkat lunak.

##### 4.1.1 Pengukuran PWM pada Motor 1

Hasil pengujian pengendalian *end effector* lengan robot dengan mikrokontroler pada *joint* 1 ditunjukkan pada Tabel 4.1 berikut ini:

Tabel 4.1 Pengujian nilai PWM motor 1

No	Sudut (Derajat)	PWM (uS)	Keterangan
1.	0°	560	Putar Kiri
2.	45°	980	Putar Kiri
3.	90°	1409	Lurus depan
4.	135°	1831	Putar Kanan
5.	170°	2200	Putar Kanan



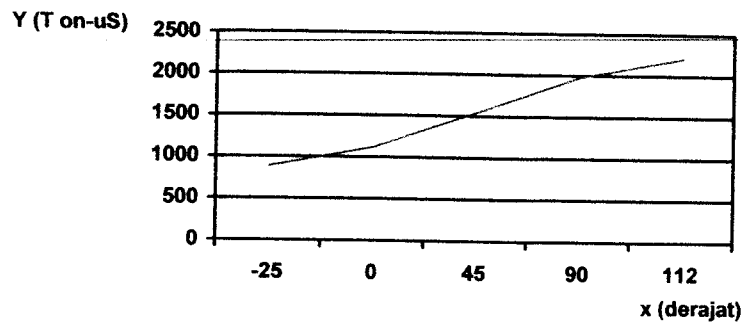
Gambar 4.1 Grafik hasil pengujian motor 1

#### 4.1.2 Pengukuran PWM pada Motor 2

Hasil pengujian pengendalian *end effector* lengan robot dengan mikrokontroler pada *joint* 1 ditunjukkan pada Tabel 4.1 berikut ini:

Tabel 4.2 Pengujian nilai PWM motor 2

No	Sudut (Derajat)	PWM (uS)	Keterangan
1.	0°	1113	Lurus depan
2.	45°	1536	Atas
3.	90°	1972	Atas
4.	112°	2200	Atas
5.	-25°	881	Bawah



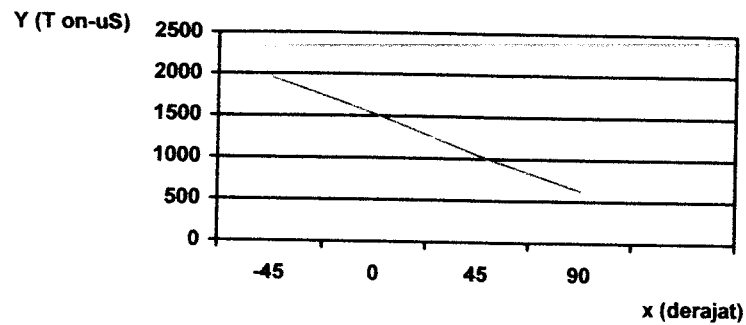
Gambar 4.2 Grafik hasil pengujian motor 2

#### 4.1.3 Pengukuran PWM pada Motor 3

Hasil pengujian pengendalian *end effector* lengan robot dengan mikrokontroler pada *joint 1* ditunjukkan pada Tabel 4.3.

Tabel 4.3 Pengujian nilai PWM motor 3

No	Sudut (Derajat)	PWM (uS)	Keterangan
1.	0°	1515	Luru depan
2.	45°	1037	Atas
3.	90°	623	Atas
4.	-45°	1952	Bawah



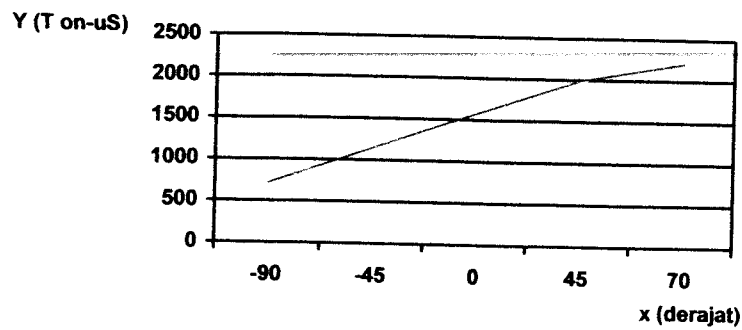
Gambar 4.3 Grafik hasil pengujian motor 3

#### 4.1.4 Pengukuran PWM pada Motor 4

Hasil pengujian pengendalian *end effector* lengan robot dengan mikrokontroler pada *joint* 1 ditunjukkan pada Tabel 4.4.

Tabel 4.4 Pengujian nilai PWM motor 4

No	Sudut (Derajat)	PWM (uS)	Keterangan
1.	0°	1570	Lurus depan
2.	45°	1984	Atas
3.	70°	2200	Atas
4.	-45°	1143	Bawah
5.	-90°	712	bawah



Gambar 4.4 Grafik hasil pengujian motor 4

#### 4.1.5 Pengujian Data Error pada Lengan Robot pada Joint 1

Pada pengujian joint 1 ini dengan mengabaikan joint 2, joint 3, joint 4. Hasil pengujian dapat dilihat pada Tabel 4.5.

Tabel 4.5 Hasil pengujian pengendalian *end effector* lengan robot dengan mikrokontroler pada *joint 1*

No	Masukan Sudut (°)	Sudut <i>joint</i> terukur			Kesalahan $\frac{\sum \text{error}}{\text{masukan sudut}} \times 100\%$
		Coba1 /error	Coba2 /error	Coba3 /error	
1.	0/180	30	31	32	82,7
2.	45	63	60	60	35,5
3.	90	95	95	95	5,55
4.	100	110	110	110	10

Dari hasil yang diperoleh terdapat kesalahan pada join 1. sudut masukan yang diberikan dari data tabel dengan sudut yang terukur, sehingga  $\text{error} = 33,44\%$

#### 4.1.6 Pengujian Data Error pada Lengan Robot pada Joint 2

Pengujian pada *joint 2* ini dengan mengabaikan *joint 1*, *joint 3*, dan *joint 4*.

Hasil pengujian dapat dilihat pada Tabel 4.6.

Tabel 4.6 Hasil pengujian pengendalian *end effector* lengan robot dengan mikrokontroler pada *joint 2*

No	Masukan Sudut ( °)	Sudut <i>joint</i> terukur			Kesalahan $\frac{\sum error}{masukan\ sudut} \times 100\%$
		Coba1 /error	Coba2 /error	Coba3 /error	
1.	0/180	-20	-20	-20	88,8
2.	45	20	20	20	55,5
3.	90	65	66	70	25,5
4.	100	80	80	80	20

Dari hasil yang diperoleh terdapat kesalahan pada *joint 2*. Sudut masukan yang diberikan dari data tabel dengan sudut yang terukur, sehingga  $error = 47,45\%$

#### 4.1.7 Pengujian Data Error pada Lengan Robot pada Joint 3

Pengujian pada *joint 3* ini dengan mengabaikan *joint 1*, *joint 2*, dan *joint 4*.

Hasil pengujian dapat dilihat pada Tabel 4.7.

Tabel 4.7 Hasil pengujian pengendalian *end effector* lengan robot dengan mikrokontroler pada *joint 3*

No	Masukan Sudut ( ° )	Sudut <i>joint</i> terukur			Kesalahan $\frac{\sum error}{masukan\ sudut} \times 100\%$
		Coba1/ error	Coba2/ error	Coba3/ error	
1.	0	0	0	0	0
2.	45	45	45	45	0
3.	90	90	90	90	0

Dari hasil yang diperoleh tidak terdapat kesalahan pada *joint 3*. Sudut masukan yang diberikan dari data tabel dengan sudut yang terukur, sehingga  $error = 0$ .

#### 4.1.8 Pengujian Data Error pada Lengan Robot pada Joint 4

Pengujian pada *joint 4* ini dengan mengabaikan *joint 1*, *joint 2*, dan *joint 3*. Hasil pengujian dapat dilihat pada Tabel 4.8.

Tabel 4.8 Hasil pengujian pengendalian *end effector* lengan robot dengan mikrokontroler pada *joint 4*

No	Masukan Sudut ( ° )	Sudut <i>joint</i> terukur			Kesalahan $\frac{\sum error}{masukan\ sudut} \times 100\%$
		Coba1 /error	Coba2 /error	Coba3 /error	
1.	0/180	10	10	10	94,4
2.	45	20	20	20	55,5
3.	90	80	80	80	11,1



Dari hasil yang diperoleh tidak terdapat kesalahan pada *joint* 4. Sudut masukan yang diberikan dari Data tabel di atas, *error* = 53,67%.

Pada pengujian masing-masing *joint* diketahui besarnya pergerakan dari tiap-tiap *joint*. Dari data yang dihasilkan oleh *joint* 2, *joint* 3, dan *joint* 4 ketika diberi masukan yang sama dan pulsa yang sama hasilnya tidak mengalami suatu kesalahan atau *error*. Dalam melakukan percobaan masih banyak terjadi *error*, kemungkinan karena perbedaan setingan hambatan yang ada di dalam motor servo itu sendiri.

Sebenarnya keempat motor servo tersebut memiliki torsi yang berbeda-beda, motor servo Hitec HS564MG mempunyai respon yang lebih lambat dari motor servo lainnya sedangkan motor servo tersebut mempunyai torsi yang besar pula. Motor servo HS-311 dan HS-322 menggunakan PCB yang sama yaitu HS-322, namun dengan perbedaan besaran komponen yang digunakan maka dapat memberikan torsi yang berbeda-beda.

#### 4.2 Perhitungan Baudret

Dengan menggunakan kristal 11,0592 MHz, maka pewaktuan dalam mikrokontroler ditunjukkan pada persamaan 4.1 sebagai berikut:

$$\begin{aligned}
 UBRRL &= \frac{fosc}{16 \cdot Boudrate} - 1 \dots\dots\dots(4.1) \\
 &= \frac{110592}{16 \cdot 4800} - 1 \\
 &= 143
 \end{aligned}$$

Tabel 4.9 Hasil pengujian lebar pulsa pada *joint 1*

No	PWM ( $\mu$ S)	Sudut (derajat)	Lebar Pulsa (mS)
1	560	0	590
2	980	45	1021
3	1409	90	1417
4	1831	135	1863
5	2200	170	2197

Tabel 4.10 Hasil pengujian lebar pulsa pada *joint 2*

No	PWM ( $\mu$ S)	Sudut (derajat)	Lebar Pulsa (mS)
1	1113	0	1120
2	1536	45	1540
3	1972	90	1991
4	2200	112	2197
5	881	-25	925

Tabel 4.11 Hasil pengujian lebar pulsa pada *joint 3*

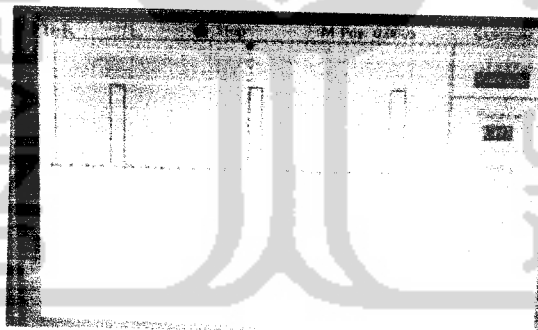
No	PWM ( $\mu$ S)	Sudut (derajat)	Lebar Pulsa (mS)
1	1515	0	1520
2	1037	45	1040
3	623	90	637
4	1915	-45	1925

Tabel 4.12 Hasil pengujian lebar pulsa pada *joint 4*

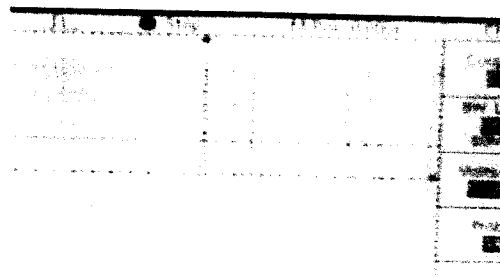
No	PWM ( $\mu S$ )	Sudut (derajat)	Lebar Pulsa (mS)
1	1570	0	1572
2	1984	45	1996
3	2200	70	2194
4	1143	-45	1160
5	712	-90	709

### 4.3 Bentuk Gelombang *PWM* (*Pulse Width Modulation*)

Gelombang *PWM* yang dikirim kemotor servo memiliki bentuk yang berbeda-beda tergantung masukan sudut yang diberikan oleh *client*. Gambar 4.5 menunjukkan bentuk gelombang *PWM* ketika *joint 1* diberi masukan sudut 90°.



Gambar 4.5 Bentuk gelombang PWM



Gambar 4.6 Bentuk gelombang data serial

#### 4.4 Pengujian Amplitude Sift keying (ASK)

Setelah dilakukan percobaan, dengan kondisi berbeda ruangan maka dapat dilihat hasil pengamatan, dapat dilihat dari Tabel 4.9 di bawah ini :

Tabel 4.13 Pengamatan sinyal ASK

No.	Kondisi Ruangan	Kondisi sinyal ASK
1.	Satu Ruangan	√
2.	Ruang Bersekat	√
3.	Beda Lantai	√

#### 4.5 Pengukuran Jarak Amplitude Shift Keying (ASK)

Dalam melakukan percobaan ini jarak pemancaran sinyal ASK di batasi jarak maksimalnya 100 m, dapat dilihat dari Tabel 4.14.

Tabel 4.14 Pengukuran Jarak Sinyal ASK dalam Satu Ruangan

No.	Jarak (meter)	Kondisi sinyal ASK
1.	10	√
2.	20	√
3.	30	√
4.	40	√
5.	50	√
6.	60	√
7.	70	√
8.	80	√
9.	90	√
10.	100	√

Tabel 4.15 Pengukuran Jarak Sinyal ASK dalam Ruang Bersekat

No.	Jarak (meter)	Kondisi sinyal ASK
1.	10	√
2.	20	√
3.	30	√
4.	40	√
5.	50	√
6.	60	√
7.	70	√
8.	80	√
9.	90	√
10.	100	√

Tabel 4.16 Pengukuran Jarak Sinyal ASK beda Lantai

No.	Jarak (meter)	Kondisi sinyal ASK
1.	10	√
2.	20	√
3.	30	√
4.	40	√
5.	50	√
6.	60	√
7.	70	√
8.	80	√
9.	90	√
10.	100	√

## BAB V

### PENUTUP

#### 5.1 Kesimpulan

Setelah selesainya uraian pada tahap-tahap pembuatan, pengujian dan pembahasan sistem pada tugas akhir ini, maka dapat dibuat kesimpulan sebagai berikut:

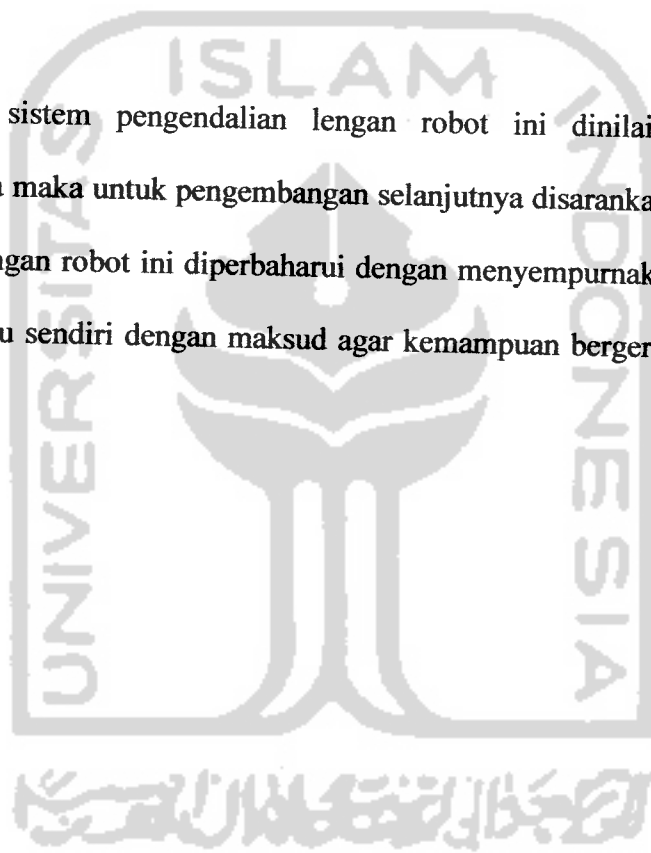
1. Pengendalian Robot Dengan PC (personal computer) menggunakan ASK sebagai transmitter dan receiver data yang dikirimkan ke robot Berbasis mikro kontroler menggunakan ATtiny 2313 Sebagai pembangkit pulsa, dengan perhitungan forward Maka lengan robot bergerak sesuai dengan intruksi yang di berikan,
2. Dalam proses penggerakan lengan Robot menggunakan program Delphi,
3. Pada pengujian joint 1 dengan mengabaikan joint 2, joint 3, joint 4. Dari hasil yang diperoleh terdapat kesalahan pada joint 1. sudut masukan yang diberikan dari data tabel dengan sudut yang terukur, sehingga  $error = 33,44\%$
4. Pengujian pada joint 2 ini dengan mengabaikan joint 1, joint 3, dan joint 4. Dari hasil yang diperoleh terdapat kesalahan pada joint 2. Sudut masukan yang diberikan dari data tabel dengan sudut yang terukur, sehingga  $error = 47,45\%$
5. Pengujian pada joint 3 ini dengan mengabaikan joint 1, joint 2, dan joint 4. Dari hasil yang diperoleh tidak terdapat kesalahan pada joint 3. Sudut

masukan yang diberikan dari data tabel dengan sudut yang terukur, sehingga  $error = 0$ .

6. Hasil pengujian pada *joint 4* dibawah ini dengan mengabaikan *joint 1*, *joint 2*, dan *joint 3*. Dari hasil yang diperoleh tidak terdapat kesalahan pada *joint 4*. Sudut masukan yang diberikan dari Data tabel di atas,  $error = 53,67\%$ .

## 5.2 Saran

Karena sistem pengendalian lengan robot ini dinilai masih banyak kekurangannya maka untuk pengembangan selanjutnya disarankan untuk kedepan hendaknya lengan robot ini diperbaharui dengan menyempurnakan mekanik dari lengan robot itu sendiri dengan maksud agar kemampuan Bergeraknya bisa lebih sempurna lagi.

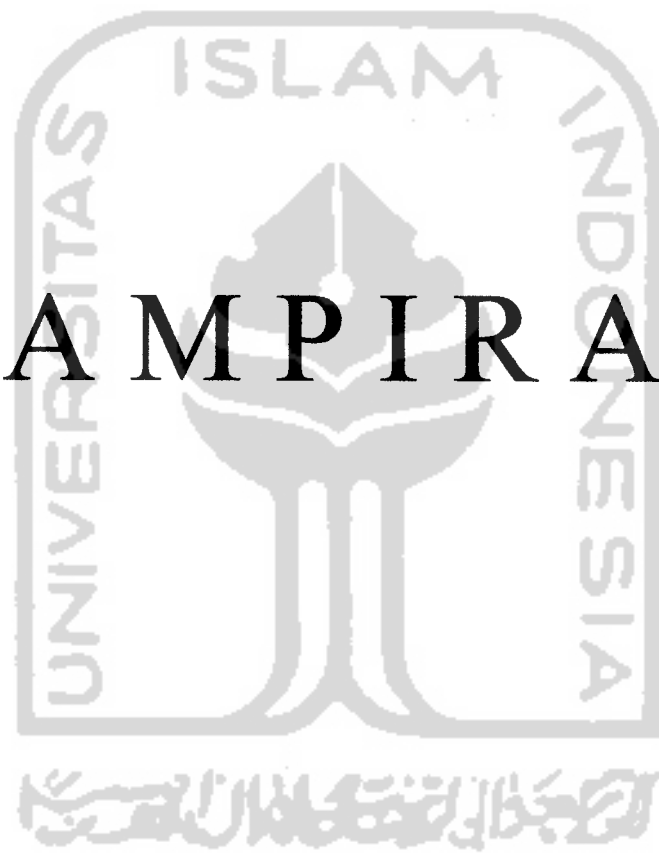


## DAFTAR PUSTAKA

- Amrullah, Yusuf, Aziz, ST**, 2004, *Bahan Kuliah Sistem Berbasis Mikroprocessor*, Yogyakarta.
- ATMEL Corporation**, ....., *8-Bit Microcontroller with 4 Kbytes Flash*,  
[www.ATMEL.com](http://www.ATMEL.com)
- Cahyadi, Indra**, 2004, "*Memfaatkan Htp Sebagai Kendali Posisi Motor Servo Dengan Feedback Kamera*", Skripsi, Teknik Elektro, Fti, UII, Yogyakarta
- Jazidie, Achmad dan Prasetyo, Dwi, Danang**, 2002, "*Pembangkitan Trayektori Joint Robot Dengan Metode Adaptive Non-Linear Least Square*", *Proceeding of Seminar on Intelligent Technology and Its Applications (SITIA 2002)* Institut Teknologi Sepuluh Nopember, Surabaya
- Ogata, Katsuhiko** 1997, "*Teknik Kontrol Auromatik*", Jilid 1 Edisi Kedua, Penerbit Erlangga.
- Setiawan, Hendra, ST**, 2004, *Bahan Kuliah Robotika*, Yogyakarta
- Winoto, Ardi.**, 2006. *Belajar Mikrokontroler Atmel AVR ATtiny2313 Step by step*, Jogjakarta: Gava Media.  
<http://www.alldatasheet.com>  
<http://www.enesc.stj.ca/reference/data-sheets.html>  
<http://www.fairchildsemi.com>



# LAMPPIRAN



Chip type : ATTiny2313  
Clock frequency : 11.059200 MHz  
Memory model : Tiny  
External SRAM size : 0  
Data Stack size : 32

\*\*\*\*\*/

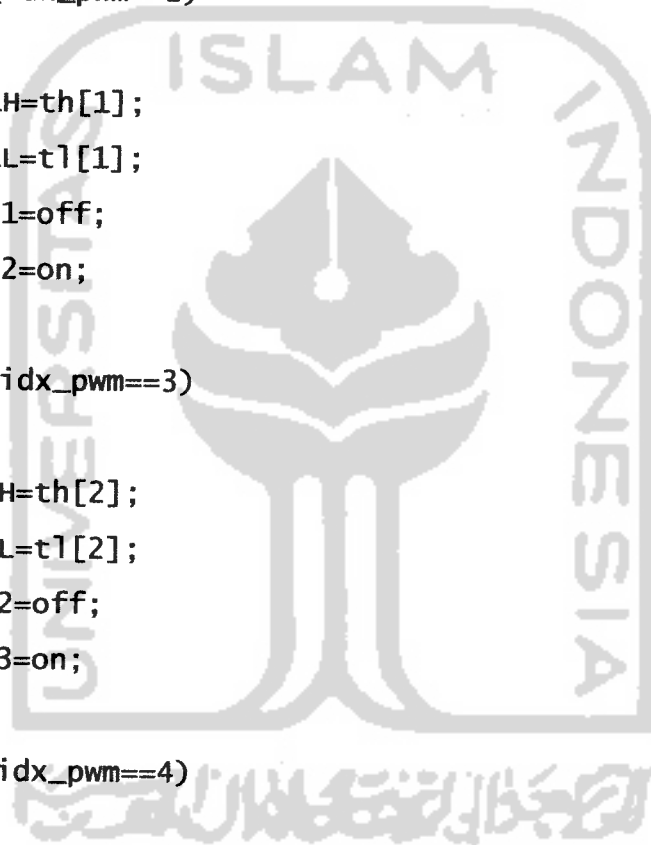
```
#include <tiny2313.h>
#include <stdio.h>
#include <delay.h>
#define motor1 PORTB.4
#define motor2 PORTB.5
#define motor3 PORTB.6
#define motor4 PORTB.7
#define on 1
#define off 0
int sudut[4];
unsigned char th[4]={0xf8,0xf8,0xf8,0xf8};
unsigned char tl[4]={0x70,0x70,0x70,0x70};

interrupt [TIM1_OVF] void timer1_ovf_isr(void)
{
    static unsigned char idx_pwm;
    if(idx_pwm==0)
    {
        TCNT1H=0xe5;
        TCNT1L=0x00;
        motor1=off;
        motor2=off;
        motor3=off;
        motor4=off;
    }
}
```

```

    }
    else if(idx_pwm==1)
    {
        TCNT1H=th[0];
        TCNT1L=tl[0];
        motor1=on;
    }
    else if(idx_pwm==2)
    {
        TCNT1H=th[1];
        TCNT1L=tl[1];
        motor1=off;
        motor2=on;
    }
    else if(idx_pwm==3)
    {
        TCNT1H=th[2];
        TCNT1L=tl[2];
        motor2=off;
        motor3=on;
    }
    else if(idx_pwm==4)
    {
        TCNT1H=th[3];
        TCNT1L=tl[3];
        motor3=off;
        motor4=on;
    }
    idx_pwm++;if(idx_pwm>4)idx_pwm=0;
}

```



```
void hitung_timer(char n,int val)
```

```
{  
    unsigned int nilai_timer;  
    nilai_timer=65536-(val*1.38);  
    th[n]=(unsigned char)(nilai_timer / 256);  
    tl[n]=(unsigned char)(nilai_timer & 0xff);  
}
```

```
void main(void) // Fungsi Utama
```

```
{  
  
    CLKPR=0x80;  
    CLKPR=0x00;  
    PORTB=0x00;  
    DDRB=0xFF;  
  
    PORTD=0x00;  
    DDRD=0x04;  
  
    TCCR1A=0x00;  
    TCCR1B=0x02;  
    TCNT1H=0x00;  
    TCNT1L=0x00;
```

```
    TIMSK=0x80;
```

```
    UCSRA=0x00;
```

```
    UCSRB=0x18;
```

```
    UCSRC=0x06;
```

```
    UBRRH=0x00;
```

```
UBRRL=0x8F;
```

```
#asm("sei")
```

```
while (1)
```

```
{
```

```
if(gets()=="robot")
```

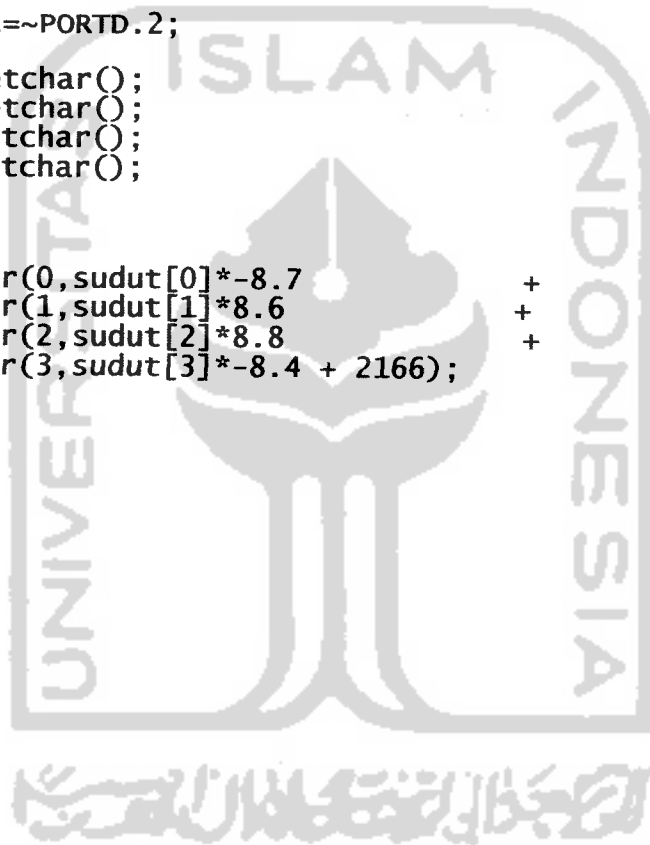
```
{
```

```
PORTD.2=~PORTD.2;
```

```
sudut[0]=getchar();  
sudut[1]=getchar();  
sudut[2]=getchar();  
sudut[3]=getchar();
```

```
hitung_timer(0,sudut[0]*-8.7 +  
hitung_timer(1,sudut[1]*8.6 +  
hitung_timer(2,sudut[2]*8.8 +  
hitung_timer(3,sudut[3]*-8.4 + 2166);
```

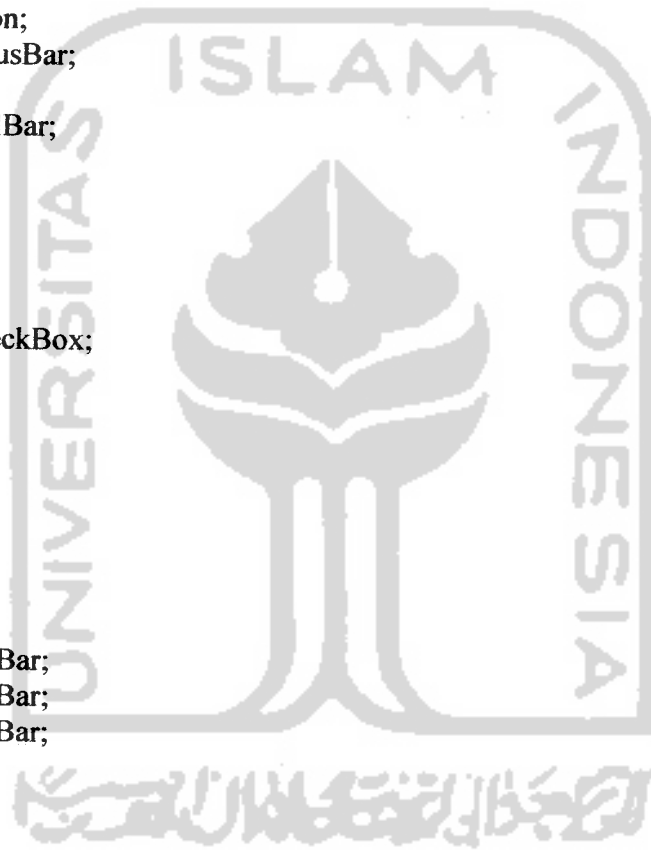
```
2078);  
1014);  
601);
```



## Listing Delphi

```
unit Unit1;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, CPort, ComCtrls, XPMAN, ExtCtrls, math;

type
  TForm1 = class(TForm)
    ComPort: TComPort;
    cbPort: TComboBox;
    btHubung: TButton;
    StatusBar1: TStatusBar;
    Panel1: TPanel;
    sbMotor1: TScrollBar;
    lbmtr1: TLabel;
    Button1: TButton;
    Timer1: TTimer;
    Shape1: TShape;
    Button2: TButton;
    CheckBox1: TCheckBox;
    Panel5: TPanel;
    Label3: TLabel;
    Label4: TLabel;
    Label5: TLabel;
    Image1: TImage;
    Panel6: TPanel;
    Image2: TImage;
    Button3: TButton;
    sbMotor2: TScrollBar;
    sbMotor3: TScrollBar;
    sbMotor4: TScrollBar;
    lbmtr2: TLabel;
    lbmtr3: TLabel;
    lbmtr4: TLabel;
    edx: TEdit;
    Label1: TLabel;
    edy: TEdit;
    Label2: TLabel;
    Button4: TButton;
    Label7: TLabel;
    edz: TEdit;
    procedure sbMotor2Change(Sender: TObject);
    procedure sbMotor3Change(Sender: TObject);
    procedure sbMotor4Change(Sender: TObject);
```



```

procedure cbPortChange(Sender: TObject);
procedure btHubungClick(Sender: TObject);
procedure sbMotor1Change(Sender: TObject);
procedure FormClose(Sender: TObject; var Action: TCloseAction);
procedure Button1Click(Sender: TObject);
procedure Timer1Timer(Sender: TObject);
procedure CheckBox1Click(Sender: TObject);
procedure Button2Click(Sender: TObject);
procedure FormCreate(Sender: TObject);
procedure Button3Click(Sender: TObject);
procedure Button4Click(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;

var
  Form1: TForm1;
  sudut1,sudut2,sudut3,sudut4:byte;
  sdt1,sdt2,sdt3,sdt4:byte;
  lx,L1, L2, L3, L4: extended;
  h,w: Integer;

implementation

{$R *.dfm}

function sbx(x:extended):integer;
begin
  Result:=Round(w div 2+x);
end;

function sby(y:extended):integer;
begin
  Result:=Round(h-y);
end;

function Gbr(sd2, sd3, sd4: extended;Image2:TImage): integer;
var x1,x2,x3,x4,y1,y2,y3,y4:extended;
    s2, s3, s4: extended;
    label Andra;
begin
  Andra:
  lx:=6;

```

```
s2:=sd2*pi/180;  
s3:=sd3*pi/180;  
s4:=sd4*pi/180;
```

```
x1:=0;  
x2:=x1+L2*lx*cos(s2);  
x3:=x2+L3*lx*cos(s2+s3);  
x4:=x3+L4*lx*cos(s2+s3+s4);
```

```
y1:=L1*lx;  
y2:=y1+L2*lx*sin(s2);  
y3:=y2+L3*lx*sin(s2+s3);  
y4:=y3+L4*lx*sin(s2+s3+s4);
```

```
if (y4<=5) then  
begin  
sd4:=sd4-1;  
goto Andra;  
end;  
if (y3<=10) then  
begin  
sd3:=sd3-1;  
goto Andra;  
end;
```

```
with Image2 do  
begin
```

```
Canvas.Pen.Color:=clBlue;  
Canvas.Brush.Color:=clWhite;  
Canvas.Pen.Width:=5;  
Canvas.Rectangle(0,0,w,h);  
Canvas.Pen.Color:=clRed;  
Canvas.Rectangle(w div 2-8,h-7,w div 2+8,h);  
Canvas.MoveTo(sbx(0), sby(0));
```

```
Canvas.LineTo(sbx(x1), sby(y1));  
Canvas.LineTo(sbx(x2), sby(y2));  
Canvas.LineTo(sbx(x3), sby(y3));  
Canvas.LineTo(sbx(x4), sby(y4));  
Canvas.Ellipse(sbx(x1-3), sby(y1+3),sbx(x1+3),sby(y1-3));  
Canvas.Ellipse(sbx(x2-3), sby(y2+3),sbx(x2+3),sby(y2-3));  
Canvas.Ellipse(sbx(x3-3), sby(y3+3),sbx(x3+3),sby(y3-3));  
Canvas.TextOut(sbx(x4+5), sby(y4+10),'(' +  
+inttostr(round(x4) div 6)  
+','+inttostr(round(y4) div 6)+')');  
Canvas.Pen.Color:=clBlue;
```





```
end;  
Result:=1;  
end;
```

```
procedure TForm1.sbMotor1Change(Sender: TObject);  
begin  
  lbmtr1.Caption:='MOTOR 1 = '+inttostr(sbMotor1.Position)+'°';  
  Button3.Click;  
end;
```

```
procedure TForm1.sbMotor2Change(Sender: TObject);  
begin  
  lbmtr2.Caption:='MOTOR 2 = '+inttostr(sbMotor2.Position)+'°';  
  Button3.Click;  
end;
```

```
procedure TForm1.sbMotor3Change(Sender: TObject);  
begin  
  lbmtr3.Caption:='MOTOR 3 = '+inttostr(sbMotor3.Position)+'°';  
  Button3.Click;  
end;
```

```
procedure TForm1.sbMotor4Change(Sender: TObject);  
begin  
  lbmtr4.Caption:='MOTOR 4 = '+inttostr(sbMotor4.Position)+'°';  
  Button3.Click;  
end;
```

```
procedure TForm1.Timer1Timer(Sender: TObject);  
begin  
  sudut1:=sbMotor1.Position;  
  sudut2:=sbMotor2.Position;  
  sudut3:=sbMotor3.Position;  
  sudut4:=sbMotor4.Position;
```

```
if checkbox1.Checked=false then  
begin  
  sdt1:=sudut1;  
  sdt2:=sudut2;  
  sdt3:=sudut3+90;  
  sdt4:=sudut4+90;  
end;
```

```
sdt3:=90-sudut3;  
sdt4:=90-sudut4;
```

```

if comport.Connected then
begin
  Comport.WriteStr('  '+#13+#10+'robot');
  Comport.Write(sdt1,1);
  Comport.Write(sdt2,1);
  Comport.Write(sdt3,1);
  Comport.Write(sdt4,1);

  if shape1.Brush.Color=citime then shape1.Brush.Color:=clbtnface
  else shape1.Brush.Color:=citime;
end
else shape1.Brush.Color:=clbtnface;
end;

procedure TForm1.cbPortChange(Sender: TObject);
begin
  ComPort.Port:=cbPort.Text;
end;

procedure TForm1.btHubungClick(Sender: TObject);
begin
  if ComPort.Connected then
  begin
    Comport.Connected:=False;
    btHubung.Caption:='Connect';
    cbPort.Enabled:=True;
  end
  else
  begin
    Comport.Connected:=True;
    btHubung.Caption:='Disconnect';
    cbPort.Enabled:=False;
    sbMotor1.Position:=0;
    sbMotor2.Position:=0;
    sbMotor3.Position:=0;
    sbMotor4.Position:=0;
  end;
end;

procedure TForm1.FormCreate(Sender: TObject);
begin
  h:=300;
  w:=470;

  L1:=13;
  L2:=10;

```

```
L3:=10;  
L4:=9;
```

```
Canvas.Brush.Color:=clWhite;  
Canvas.Pen.Color:=clBlue;  
Canvas.Pen.Width:=10;  
Canvas.Rectangle(0,0,w,h);  
Gbr(0,0,0,Image2);  
end;
```

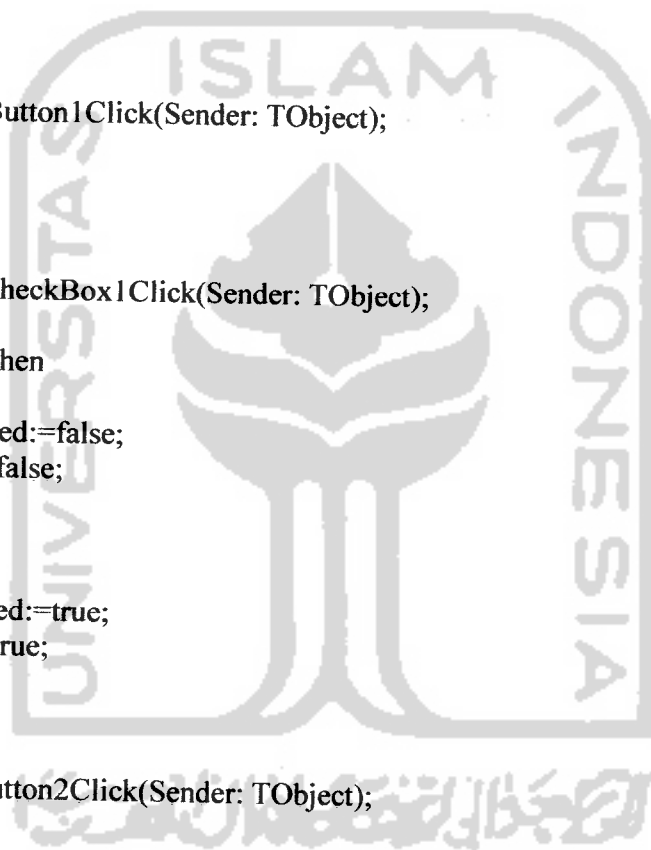
```
procedure TForm1.FormClose(Sender: TObject; var Action: TCloseAction);  
begin  
  Comport.Close;  
end;
```

```
procedure TForm1.Button1Click(Sender: TObject);  
begin  
  Form1.Close;  
end;
```

```
procedure TForm1.CheckBox1Click(Sender: TObject);  
begin  
  if button2.Enabled then  
  begin  
    CheckBox1.Checked:=false;  
    button2.Enabled:=false;  
  end  
  else  
  begin  
    CheckBox1.Checked:=true;  
    button2.Enabled:=true;  
  end;  
end;
```

```
procedure TForm1.Button2Click(Sender: TObject);  
begin  
  sdt1:=sudut1;  
  sdt2:=sudut2;  
  sdt3:=sudut3;  
  sdt4:=sudut4;  
end;
```

```
procedure TForm1.Button3Click(Sender: TObject);  
begin  
  Gbr(sbMotor2.Position,  
    sbMotor3.Position,
```



```
sbMotor4.Position,  
Image2);  
end;
```

```
procedure TForm1.Button4Click(Sender: TObject);
```

```
var
```

```
r3,theta1,theta2,theta3,theta4,x3,z3,x4,y4,z4,gama,beta : single;
```

```
label andra1,andra2;
```

```
begin
```

```
  x4:=strtofloat(edx.Text)+1;
```

```
  y4:=strtofloat(edy.Text);
```

```
  z4:=strtofloat(edz.Text)-L1;
```

```
  theta4:=-89;
```

```
  andra1:
```

```
  theta4:=theta4+1;
```

```
  if theta4=91 then goto andra2;
```

```
  x3:=x4-(L4*cos(theta4*pi/180));
```

```
  z3:=z4-(L4*sin(theta4*pi/180));
```

```
  r3:=sqrt(sqr(x3)+sqr(z3));
```

```
  if r3>(L2+L3) then goto andra1;
```

```
  theta3:=arccos((sqr(r3)-sqr(L2)-sqr(L3))/(2*L2*L3));
```

```
  gama:=arccos((sqr(r3)+sqr(L2)-sqr(L3))/(2*L2*r3));
```

```
  beta:=round(arctan(z3/x3)*180/pi)+1;
```

```
  theta2:=beta-gama;
```

```
  theta4:=theta4-theta2-theta3;
```

```
  Gbr(theta2,theta3,theta4,Image2);
```

```
    sbmotor2.Position:=round(theta2);
```

```
    sbmotor3.Position:=round(theta3);
```

```
    sbmotor4.Position:=round(theta4);
```

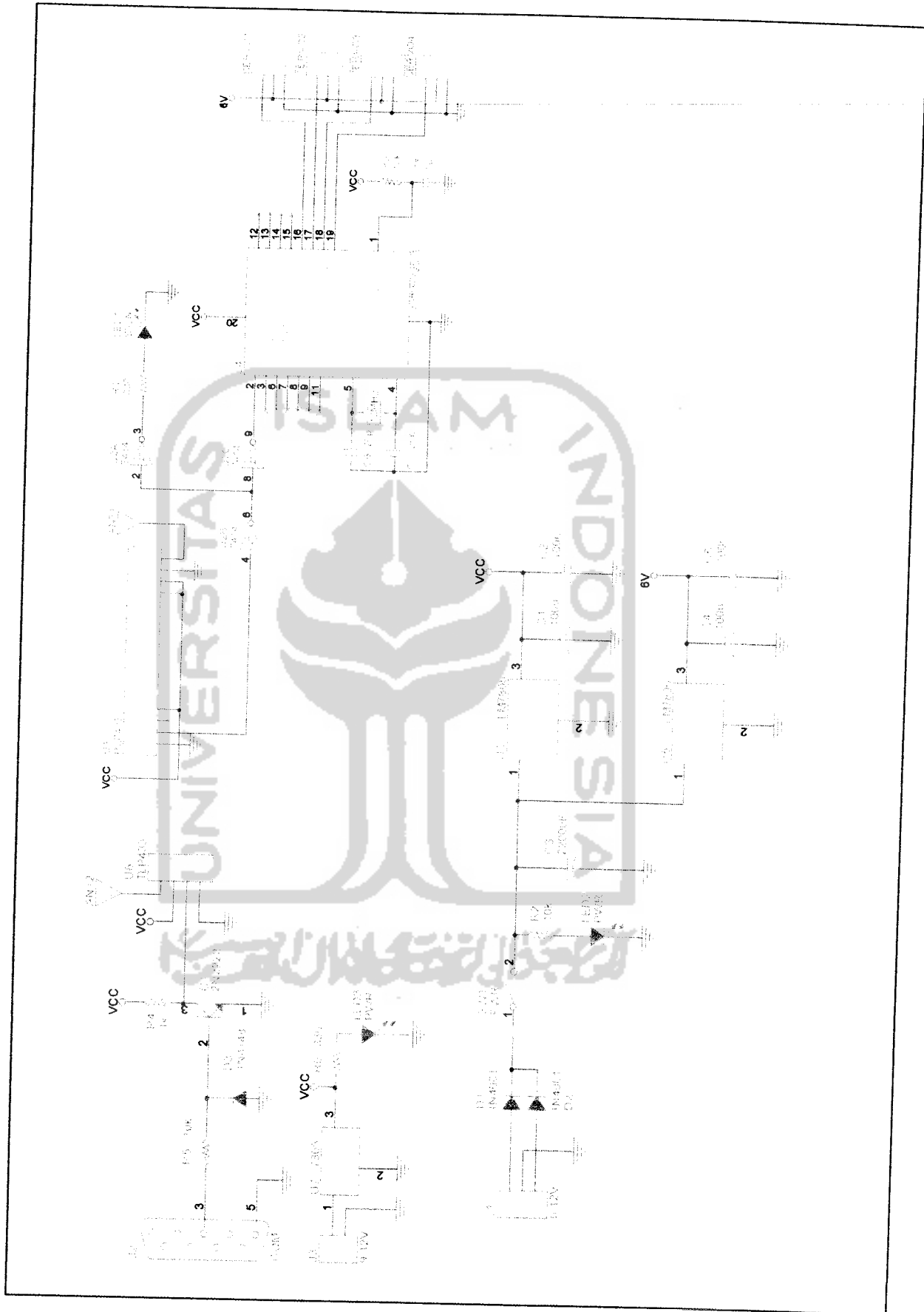
```
  andra2:
```

```
end;
```

```
end.
```



# SKEMATIK RANGKAIAN PENGENDALI LENGAN ROBOT



**Tabel Register I/O atau Port I/O**

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
\$3F (\$5F)	SREG	I	T	H	S	V	N	Z	C
\$3D (\$5D)	SPL	SP7	SP6	SP5	SP4	SP3	SP2	SP1	SP0
\$3C (\$5C)	OCR0B	Register pembanding Timer/Counter0 B 8-bit							
\$3B (\$5B)	GIMSK	INT1	INT0	PCIE					
\$3A (\$5A)	EIFR	INTF1	INTF0	PCIF					
\$39 (\$59)	TIMSK	TOIE1	OCIE1A	OCIE1B		TICIE1	OCIE0B	TOIE0	OCIE0A
\$38 (\$58)	TIFR	TOV1	OCF1A	OCF1B		ICF1	OCF0B	TOV0	OCF0A
\$37 (\$57)	SPMCR				CTPB	RFLB	PGWRT	PGRES	SELFPERGN
\$36 (\$56)	OCR0A	Register pembanding Timer/Counter0 A 8-bit							
\$35 (\$55)	MCUCR	PUD	SM01	SE	SM1	ISC11	ISC10	ISC01	ISC00
\$34 (\$54)	MCUSR					WDRF	BCRF	EXTRF	PORF
\$33 (\$53)	TCCR0B	FOC0A	FOC0B			WGM02	CS02	CS01	CS00
\$32 (\$52)	TCNT0	Register pencacah Timer/Counter 0 8-bit							
\$31 (\$51)	OSCAL		CAL6	CAL5	CAL4	CAL3	CAL2	CAL1	CAL0
\$30 (\$50)	TCCR0A	COM0A1	COM0A0	COM0B1	COM0B0			WGM01	WGM00
\$2F (\$4F)	TCCR1A	COM1A1	COM1A0	COM1B1	COM1B0			WGM11	WGM10
\$2E (\$4E)	TCCR1B	CNC1	ICES1		WGM13	WGM12	CS12	CS11	CS10
\$2D (\$4D)	TCNT1H	Register pencacah Timer/Counter 1 Byte atas							
\$2C (\$4C)	TCNT1L	Register pencacah Timer/Counter 1 Byte bawah							

Address	Nama	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
\$2B (\$4B)	OCR1AH	Register pembanding Timer/Counter 1 A Byte atas							
\$2A (\$4A)	OCR1AL	Register pembanding Timer/Counter 1 A Byte bawah							
\$29 (\$49)	OCR1BH	Register pembanding Timer/Counter 1 B Byte atas							
\$28 (\$48)	OCR1BL	Register pembanding Timer/Counter 1 B Byte bawah							
\$27 (\$46)	CLKPR	CLKPCE				CLKPS3	CLKPS2	CLKPS1	CLKPS0
\$25 (\$45)	ICR1H	Register penangkap input Timer/Counter 1 Byte atas							
\$24 (\$44)	ICR1L	Register penangkap input Timer/Counter 1 Byte bawah							
\$23 (\$43)	GTCR								PSR10
\$22 (\$42)	TCCR1C	FOC1A	FOC1B						
\$21 (\$41)	WDTCR	WDFR	WDIE	WDP3	WDTCE	WDE	WDP2	WDP1	WDPC
\$20 (\$40)	PCMSK	PCINT7	PCINT6	PCINT5	PCINT4	PCINT3	PCINT2	PCINT1	PCINT0
\$1F (\$3E)	EEAR	Register alamat EEPROM							
\$1D (\$3D)	EEDR	Register data EEPROM							
\$1C (\$3C)	EEDR			EEDR1	EEDR0	EEDR7	EEDR6	EEDR5	EEDR4
\$1A (\$38)	PORTA					PORTA2	PORTA1	PORTA0	
\$14 (\$3A)	DDRA					DDRA2	DDRA1	DDRA0	
\$10 (\$39)	PINA					PINA2	PINA1	PINA0	
\$10 (\$38)	PORTB	PORTB7	PORTB6	PORTB5	PORTB4	PORTB3	PORTB2	PORTB1	PORTB0
\$11 (\$37)	DDRB	DDRB7	DDRB6	DDRB5	DDRB4	DDRB3	DDRB2	DDRB1	DDRB0
\$10 (\$36)	PINB	PINB7	PINB6	PINB5	PINB4	PINB3	PINB2	PINB1	PINB0
\$10 (\$35)	GPIOR2	General Purpose IO Register 2							
\$11 (\$34)	GPIOR1	General Purpose IO Register 1							
\$10 (\$33)	GPIOR0	General Purpose IO Register 0							
\$10 (\$32)	PORTD	PORTD7	PORTD6	PORTD5	PORTD4	PORTD3	PORTD2	PORTD1	PORTD0
\$10 (\$31)	DDRD	DDRD7	DDRD6	DDRD5	DDRD4	DDRD3	DDRD2	DDRD1	DDRD0
\$10 (\$30)	PIND	PIND7	PIND6	PIND5	PIND4	PIND3	PIND2	PIND1	PIND0
\$10 (\$2F)	USISR	Register data USI							
\$10 (\$2E)	USISR	USISIF	USIOIF	USIFF	USIDC	USICNT3	USICNT2	USICNT1	USICNT0
\$10 (\$2D)	SPCR	SPIE	SPE	DOR	STR	CPOL	CPH	SPR1	SPR0
\$10 (\$2C)	UDR	Register data IO UART							
\$10 (\$2B)	UCSRA	RXC	TXC	UDRE	FE	DOR	UPE	U2X	MPCM
\$10 (\$2A)	UCSPR	RXCIE	TXCIE	UDRIE	RXN	TXN	UCSZ2	RXB8	TXB8
\$10 (\$29)	UBRR1L	Register generator baud rate UART (7..0)							
\$10 (\$28)	ACSR	ACD	ACBG	ACC	ACI	ACIE	ACIC	ACIS1	ACIS0
\$10 (\$27)	UCSR0L		UMSEL	UPM1	UPM0	USBS	UCSZ1	UCSZ0	UCPOL
\$10 (\$26)	UCSR0H								
\$10 (\$25)	UCSR0C								
\$10 (\$24)	UCSR0B								
\$10 (\$23)	UCSR0A								
\$10 (\$22)	UCSR0D							AIN1D	AIN0D