

SISTEM KENDALI JARI TANGAN ROBOT

TUGAS AKHIR

**Diajukan Sebagai Salah Satu Syarat
Untuk Memperoleh Gelar Sarjana Teknik Mesin**



Disusun oleh :

Nama : Dwi Setiyanto

NIM : 02 525 029

**JURUSAN TEKNIK MESIN
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA
YOGYAKARTA**

2011

LEMBAR PENGESAHAN PEMBIMBING

SISTEM KENDALI JARI TANGAN ROBOT

TUGAS AKHIR



Disusun oleh:

Nama : Dwi Setiyanto

No. Mahasiswa : 02 525 029

Yogyakarta, Mei 2011

Menyetujui,

Pembimbing I

(Muhammad Ridwan, S.T., M.T)

Pembimbing II

(Agung Nugroho Adi, S.T.,M.T)

LEMBAR PENGESAHAN DOSEN PENGUJI

SISTIM KENDALI JARI TANGAN ROBOT

TUGAS AKHIR

Disusun oleh:

Nama : Dwi Setiyanto

No. Mahasiswa : 02 525 029

Telah Dipertahankan di Depan Sidang Penguji Sebagai Salah Satu Syarat
Untuk Memperoleh Gelar Sarjana Teknik Mesin
Fakultas Teknologi Industri Universitas Islam Indonesia

Yogyakarta, Mei 2011

Tim Penguji

(Muhammad Ridlwan, S.T., M.T.)

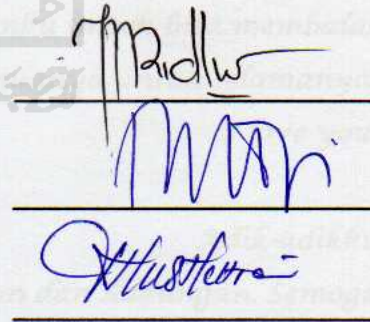
Ketua

(Purtojo, S.T., M.Sc.)

Anggota I

(Tri Setya Putra, S.T.)

Anggota II



Mengetahui,

Ketua Jurusan Teknik Mesin

Fakultas Teknologi Industri

Universitas Islam Indonesia



(Agung Nugroho Adi, S.T., M.T.)

HALAMAN PERSEMBAHAN

Kupersembahkan Kepada :
Allah SWT yang telah mencurahkan rahmat, hidayah serta karunia-Nya.
Nabi Muhamad SAW sebagai teladan dalam menapaki kehidupan.

Bapak & Ibu
Yang selalu mendo'akan dari hembusan nafas pertamaku sampai saat ini,
Yang tak pernah letih kau menuang kasih sayang
Tak pernah letih kau memberi, mendidik,
menuntun setiap langkahku
jangan berhenti kau memberi
jangan berhenti do'amu mengalir untukku.
Pengorbananmu yang begitu besar takkan pernah terbalaskan olehku.
Semoga ALLAH SWT yang membalas kalian dengan limpahan rahmat
dan ridho-Nya, karena ketidak sanggupanku untuk bisa membalas
semuanya untuk selamanya.
I love you.

Adik-adikku
Makasih buat kasih sayang, bimbingan, bantuan dan dukungan. Semoga
ALLAH SWT selalu memberikan rahmat dan ridho-Nya.
Amin.

Teman-temanku dan saudaraku terimakasih untuk semua detik, menit,
jam, hari, bulan dan tahun penuh arti yang berisikan semangat, do'a,
pengorbanan atas bantuannya.
Pakde Eko serta istri, Pak Sugeng, Bu Erni, Pak Wisnu, dan semua
keluarga

*Terima kasih untuk semua bantuan, do'a dan perhatiannya.
Semoga ALLAH SWT membalas kalian dengan limpahkan rahmat dan
karunia-Nya.*

Amin.

*Pak Suroto, terimakasih atas ilmu yang diberikan, bantuan serta tempat
Semoga ALLAH SWT selalu memberikan rahmat dan ridho-Nya, sukses
dan lancar selalu pekerjaannya.*

Amin.

*Teman TA seperjuanganku, Nur Akhfridho, Angga Prima, Figor
Rahman, Amunk Zuhri, Raju dan Nur Indragiri, temen-temen satu kost
Bari, Feri, Yosa, Putra, Asep dan Bobi. Terima kasih selama ini telah
banyak membantu, berbagi, dan menolongku dalam berbagai hal.
Semoga kalian cepet selesai kuliahnya, sukses dan bisa membanggakan
orang tua dan keluarga.*

Amin.

*Temen-temen Mesin UII angkatan 2002 yang lagi skripsi dan semua
temen-temen di Teknik Mesin UII. Buktikan kalo kita bisa dan mampu!
serta buktikan bahwa kita adalah yang terbaik! Semoga semuanya cepet*

lulus dan sukses!

Amin.

HALAMAN MOTTO

“... Allah akan meninggikan orang yang beriman di antara kamu dan orang-orang yang diberi ilmu pengetahuan beberapa derajat ...”

(Q.S Al Mujaadilah ayat 11)

“Sesungguhnya sesudah kesulitan itu ada kemudahan, maka apabila kamu telah selesai (dari suatu urusan) maka kerjakanlah dengan sungguh-sungguh urusan yang lain. Dan hendaklah hanya kepada tuhanmulah kamu berharap”

(Qs. Al-Insraf 6-8)

“Apabila kamu tidak dapat memberikan kebaikan kepada orang lain dengan kekayaanmu, berilah mereka kebaikan dengan wajahmu yang berseri-seri disertai ahlak yang baik”

(Nabi Muhammad SAW)

“Jangan lihat masa lampau dengan penyesalan, jangan pula lihat masa depan dengan ketakutan, tapi lihatlah sekitarmu dengan penuh kesadaran”

(James Thurber)

KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Assalamu'alaikum. Wr. Wb.,

Puji syukur kehadirat Allah SWT yang telah melimpahkan segala nikmat, rahmat dan hidayah-Nya sehingga laporan tugas akhir dengan judul “ **Sistem Kendali Penggerak Pada Jari tangan Robot** ” ini dapat terselesaikan dengan baik. Shalawat serta salam semoga selalu tercurah kepada Nabi besar kita Nabi Muhammad SAW beserta keluarga serta sahabatnya.

Tugas akhir ini adalah sebagai salah satu syarat yang harus ditempuh untuk mendapatkan gelar sarjana Jurusan Teknik Mesin, Fakultas Teknologi Industri, Universitas Islam Indonesia.

Selama penulisan dan penyusunan laporan tugas akhir ini penulis banyak mendapatkan bantuan dari berbagai pihak, untuk itu penulis ingin menyampaikan rasa terima kasih yang sebesar-besarnya kepada :

1. Ayahanda Suyono & Ibunda Ngatimah, Saudaraku Eni dan suami, Saudaraku Tri Wulan Dari. Serta keluarga untuk semua do'a dan dukungannya.
2. Bapak Ir. Gumbolo HS., M.Sc selaku Dekan Fakultas Teknologi Industri, Universitas Islam Indonesia.
3. Bapak Agung Nugroho Adi , ST., MT selaku Ketua Jurusan Teknik Mesin Universitas Islam Indonesia dan Dosen Pembimbing tugas akhir yang telah sangat banyak membantu dan membimbing dengan penuh kesabaran selama proses pengerjaan dan penyusunan tugas akhir ini .
4. Bapak Muhammad Ridlwan, ST.,MT selaku Dosen Pembimbing tugas akhir yang telah sangat banyak membantu dan membimbing dengan

penyusunan tugas akhir ini.

5. Segenap Dosen Jurusan Teknik Mesin, Universitas Islam Indonesia.
6. Mba Indah selaku *Front Office* Jurusan Teknik Mesin Universitas Islam Indonesia yang telah banyak membantu untuk semua urusan birokrasi dan administrasi selama pengerjaan dan penyusunan laporan tugas akhir ini.
7. Teman-teman bimbingan tugas akhir Bapak Muhammad Ridlwan, ST.,MT.
8. Teman-teman angkatan 2002 Jurusan Teknik Mesin dan semua mahasiswa Jurusan Teknik Mesin untuk dukungan dan bantuannya, “Tetap semangat”.
9. Serta ucapan terima kasih yang sebesar-besarnya kepada semua pihak yang tidak bisa penulis sebutkan namanya satu persatu di sini. Semoga Allah membalas kebaikan kalian semua dengan berlipat ganda. Amin.

Penulis sangat menyadari bahwa dalam penulisan laporan tugas akhir ini terdapat banyak kesalahan. Untuk itu penulis sangat mengharapkan adanya kritik serta saran yang membangun dari semua kalangan pembaca, sehingga penulis dapat memperbaikinya pada kesempatan yang akan datang. Akhir kata semoga tugas akhir ini dapat bermanfaat bagi kita semua. Amin.

Wassalamu’alaikum Wr.Wb.,

Jogjakarta, Mei 2011

(Dwi Setiyanto)

SISTEM KENDALI JARI TANGAN ROBOT

ABSTRAKSI

Pembuatan Tugas akhir ini dilatar belakangi sebagai perbaikan pembuatan tangan robot sebelumnya. Dimana pada proses pembuatan tangan robot sebelumnya motor servo yang digunakan adalah satu dan sekarang digunakan lima buah motor servo. Alat ini dibuat dengan tujuan untuk membuat rancangan sistem kendali gerak untuk menggerakkan jari tangan robot.

Metode yang digunakan pada pembuatan rangkaian sistem kendali adalah membuat desain rangkaian menggunakan software PCB layout dengan langkah yaitu: pemindahan desain pada PCB, pelarut yang bukan desain pada PCB, proses perakitan, pengujian rangkaian dan pemrograman.

Model tangan robot yang dibuat dari jari tangan sampai sikut tangan. Tangan robot dapat digerakan dengan menggunakan dua metode yaitu dengan potensiometer dan keyboard. Software Code Vision AVR digunakan dalam pemrograman. Untuk pemrogramannya dengan menuliskan kode program kemudian diprogram pada mikrokontroler. Mikrokontroler yang digunakan ATmega 8535 adalah salah satu jenis mikrokontroler keluaran AVR dan dalam pemrogramannya dengan menghubungkan pin VCC+5V, pin GND 0V, pin kontrol dengan port komputer. Jari tangan robot dapat bergerak menggenggam dan membuka serta dapat juga bergerak menggenggam satu persatu.

Kata kunci : Tangan Robot, Rangkaian Mikrokontroler, Software CVAVR

ROBOT FINGER CONTROL SYSTEM

ABSTRACT

This final task is conducted to improve previous robot hand making. Previous robot hand making use one servo motor and this improvement used five servo motor. This device was made to produce movement control system design to move robot finger.

Method used in making the control system circuit is to make circuit design using PCB layout program with following steps: transferring design on PCB, leaching non-design on PCB, assembly process, circuit testing, and programming.

Robot hand model was made from finger to elbow. Robot hand can be moved using two methods: potentiometer and keyboard. Code Vision AVR software was used in programming. The programming was done by writing program code and then the code was programmed into microcontroller. It used microcontroller ATmega 8535, a production of AVR and the programming was done by connecting pin VCC +5V, pin GND 0V, and control pin to computer port. The robot can move its fingers gripping and opening and can also grip its fingers one by one.

Key words : **Robot Hand, Microcontroller Circuit, CV AVR Software**

DAFTAR ISI

HALAMAN JUDUL.....	i
LEMBAR PENGESAHAN PEMBIMBING	i
LEMBAR PENGESAHAN PENGUJI.....	Error! Bookmark not defined.
HALAMAN PERSEMBAHAN	ii
HALAMAN MOTTO	vi
KATA PENGANTAR.....	vii
ABSTRAKSI.....	ix
DAFTAR ISI	xi
DAFTAR GAMBAR.....	xiii
DAFTAR TABEL	xv
Bab 1 PENDAHULUAN	xv
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	1
1.3 Batasan Masalah	2
1.4 Tujuan Penelitian	2
1.5 Manfaat Penelitian	2
1.6 Sistematika Penulisan	2
Bab 2 LANDASAN TEORI.....	4
2.1 Robot.....	4
2.2. Gripper	7
2.3 Motor Servo	10
2.4 Sistem Kontrol	13
Bab 3 METODOLOGI PENELITIAN.....	19
3.1 Diagram Alir.....	19
3.2 Kajian Referensi	20
3.3 Alat dan Bahan.....	20

3.4	Bagian-bagian Tangan Robot	20
3.5	Perakitan Sistem Kendali Penggerak Jari Tangan Robot	22
3.6	Pembuatan Rangkaian Mikrokontroler	26
3.7	Komunikasi Serial.....	31
3.8	Code Vision AVR.....	36
Bab 4 ANALISA DAN PEMBAHASAN		37
4.1	Peralatan.....	37
4.2	Pemrograman	37
4.3	Pengukuran Derajat Motor servo	43
4.4	Pengukuran Derajat Putaran Jari Tangan Robot.....	43
4.5	Tangan Robot Terdahulu dan Tangan Robot Pengembangan	48
4.6	Rangkaian Setup Penelitian	49
4.7	Mode Input.....	50
4.7	Biaya Pembuatan	53
Bab 5 PENUTUP.....		55
5.1	Kesimpulan	55
5.2	Saran	55
DAFTAR PUSTAKA.....		56
LAMPIRAN		57

DAFTAR GAMBAR

Gambar 2-1 Manipulator Robot Industri (Endra Pitowarno 2006)	5
Gambar 2-2 Proses Pengelasan Pada Industri Mobil.....	6
Gambar 2-3 <i>End Effector</i> Tangan <i>Gripper</i>	7
Gambar 2-4 <i>Gripper</i> Mekanik (Wirawan Sumbodo ,2008)	8
Gambar 2-5 <i>Gripper</i> Magnet (Wirawan Sumbodo , 2008).....	9
Gambar 2-6 Contoh Sistem <i>Vakum</i> (Wirawan Sumbodo , 2008)	9
Gambar 2-7 <i>Gripper</i> <i>Vacuum</i> (Wirawan Sumbodo , 2008)	10
Gambar 2-8 Motor Servo (Wirawan Sumbodo , 2008).....	11
Gambar 2-9 Hubungan Pulsa Dan Posisi Motor Servo	12
Gambar 2-10 Sistem Robotik	13
Gambar 2-11 Diagram kontrol loop terbuka (Endra Pitowarno, 2006).....	14
Gambar 2-12 Diagram kontrol loop tertutup (Endra Pitowarno, 2006)	14
Gambar 2-13 Konfigurasi pin ATmega 8535 (Data chip ATmega 8535).....	18
Gambar 3-1 Diagram alir.....	19
Gambar 3-2 Tangan robot	21
Gambar 3-3 Tali senar sebagai otot penggerak	21
Gambar 3-4 Motor servo	22
Gambar 3-5 Skema sistem kendali penggerak tangan robot	23
Gambar 3-6 Pemasangan otot penggerak pada jari tangan robot.....	24
Gambar 3-7 Posisi pemasangan motor servo pada lengan robot.....	25
Gambar 3-8 Desain rangkaian	26
Gambar 3-9 Desain menempel pada rangkaian pada PCB.....	27
Gambar 3-10 Pelarutan desain rangkaian pada PCB.....	27
Gambar 3-11 Proses pengeboran lubang kaki komponen	28
Gambar 3-12 Komponen yang telah dirangkai.....	28
Gambar 3-13 Rangkaian regulator	29
Gambar 3-14 Rangkaian arus motor servo dan pin motor servo.....	30
Gambar 3-15 Rangkaian potensiometer	31

Gambar 3-16 Komunikasi serial.....	31
Gambar 3-17 Transmisi simplex dan duplex.....	32
Gambar 3-18 Komunikasi serial sinkron.....	33
Gambar 3-19 Komunikasi serial asinkron.....	33
Gambar 3-20 IC Max 232.....	34
Gambar 3-21 Pin pada konektor DB9	35
Gambar 3-22 Program Code Vision AVR.....	36
Gambar 4-1 Nilai motor servo yang ditampilkan pada <i>Hyperterminal</i>	40
Gambar 4-2 <i>Software extrime burner</i>	42
Gambar 4-3 Nilai ADC dan pulsa	43
Gambar 4-4 Nilai pulsa (motor servo) dan nilai sudut	44
Gambar 4-5 Posisi motor servo menggunakan busur derajat sudut 0°	45
Gambar 4-6 Posisi jari tangan pada sudut 0°	45
Gambar 4-7 Nilai sudut 0° pada LCD	46
Gambar 4-8 Posisi jari tangan pada sudut 30°	46
Gambar 4-9 Posisi jari tangan pada sudut 60°	47
Gambar 4-10 Posisi jari tangan pada sudut 90°	47
Gambar 4-11 Posisi jari tangan pada sudut 180°	48
Gambar 4-12 Tangan robot pengembangan	48
Gambar 4-13 Rangkaian setup penelitian.....	49
Gambar 4-13 Pin1 ditekan (b) potensiometer diputar.....	50
Gambar 4-14 Pin1 dan pin 2 ditekan (b) potensiometer.....	50
Gambar 4-7 (a) kabel RS-232 pada port CPU (b) kabel RS-232 yang terhubung pada mikrokontroler (c) keyboard komputer.....	51

DAFTAR TABEL

Tabel 4-1 Perbedaan logika RS-232 dengan logika TTL.....	34
Tabel 4-2 Fungsi pin pada konektor DB9	35
Tabel 4-3 Biaya pembuatan rangkaian mikrokontroler.....	53
Tabel 4-4 Biaya pembuatan rangkaian serial RS-232	54
Tabel 4-5 Biaya pembelian motor servo	54



BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Tangan robot adalah tangan mekanik, dimana tangan robot memiliki kekuatan yang sama dengan tangan manusia, tetapi tidak memiliki kepekaan sentuhan, kesempurnaan daya gerak, dan kemampuan melakukan beragam pekerjaan. Tangan robot yang banyak digunakan umumnya berbentuk lengan robot, dimana bentuk tangan seperti penjepit atau tangan *gripper* yang digunakan sebagai penggenggam. Bentuk tangan *gripper* sangat mekanik, dikarenakan fungsinya untuk menjepit.

Banyak ditemui bentuk tangan robot yang menyerupai tangan manusia, tetapi bentuk dari jari-jarinya masih terlihat kaku dan berbentuk rangka. Rangka antar sendi pada jari tangan robot yang bergerak menutup dan membuka terbuat dari rantai mesin sepeda motor dimana ukurannya menyesuaikan bentuk tangan. Untuk menggerakkan setiap jari tangan robot menggunakan servo motor yang memiliki *torsi* yang cukup serta membutuhkan kontroler untuk menggerakannya. Robot harus diprogram untuk melakukan semua *step* gerakan atau kerja sebelum digunakan atau dengan kata lain tahap awal merangkai dan membangun pola pikir robot.

Pemrograman yang digunakan untuk menggerakkan robot adalah *Software Code Vision AVR*, dimana *software* ini menggunakan bahasa C dalam pemrogramannya. Pemrograman bahasa C untuk AVR luas digunakan untuk pemrograman berbagai jenis perangkat, termasuk mikrokontroler.

Berdasarkan permasalahan diatas, maka pada penelitian ini, penulis akan membuat sistem kendali penggerak untuk menggerakkan jari tangan robot.

1.2 Rumusan Masalah

Berdasarkan latar belakang masalah diatas, dapat dirumuskan permasalahan yaitu bagaimana sistem mekanisme jari tangan robot agar dapat

bekerja dan bagaimana cara membuat program untuk mengontrol jari tangan robot agar dapat bergerak dengan baik.

1.3 Batasan Masalah

Agar tidak menyimpang dan lebih terarah pada objek pembahasan maka diperlukan batasan-batasan penelitian sebagai berikut :

1. Penelitian difokuskan pada sistem pengendali gerak pada jari tangan robot yang bergerak menutup dan membuka.
2. Jari tangan robot dapat bergerak menutup dan membuka secara tidak bersamaan.
3. Kontrol yang digunakan adalah AT MEGA 8535
4. Sistem kontrol loop terbuka (*open loop*).

1.4 Tujuan Penelitian

Tujuan penelitian ini adalah untuk mengetahui sistem mekanisme jari tangan robot agar dapat bekerja dan membuat program untuk mengendalikan jari tangan robot agar dapat bergerak dengan baik.

1.5 Manfaat Penelitian

1. Sebagai referensi bagi penelitian lain mengenai robot pada masa yang akan datang.
2. Sebagai peraga mode input kendali robot dengan menggunakan potensiometer dan keyboard.

1.6 Sistematika Penulisan

Laporan tugas akhir merupakan dokumentasi pelaksanaan dan hasil penelitian. Sistematika penulisan tugas akhir ini dibagi menjadi lima bab yang terdiri dari, bab 1 merupakan bab pendahuluan berisi tentang latar belakang masalah, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian dan sistematika penulisan. BAB II merupakan tinjauan pustaka, dasar-dasar teori

yang berhubungan dengan penelitian yang akan dilakukan. BAB III berisi metodologi penelitian, peralatan dan bahan yang digunakan, serta alur penelitian. BAB IV diuraikan pengolahan data dan perhitungan yang berisi data-data yang dibutuhkan dalam penelitian ini, proses ketika penelitian berlangsung dan analisa hasil penelitian, proses-proses yang terjadi, perubahan-perubahan yang terjadi, perbandingan dengan penelitian lain. BAB V merupakan bab penutup yang berisi kesimpulan serta saran-saran untuk kemajuan penelitian.



BAB II

LANDASAN TEORI

2.1 Robot

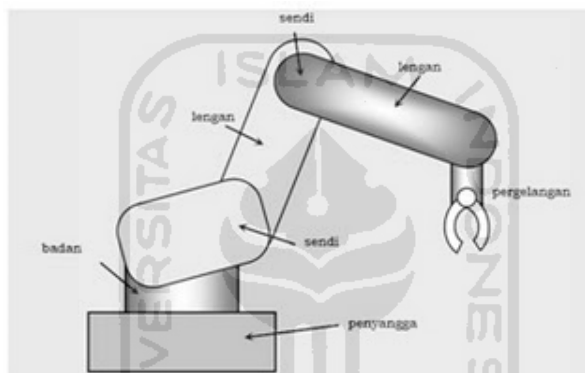
Robot berasal dari bahasa ceko-slowakia, *Robota*, yang berarti pekerja. Istilah ini muncul pada tahun 1921 oleh pengarang sandiwara bernama Karel Capek. Karyanya saat itu berjudul *Rossum's Universal Robot* yang artinya Robot Dunia milik Rossum, mengisahkan mesin yang menyerupai manusia yang dapat bekerja tanpa lelah. Gaung istilah robot kemudian memperoleh sambutan dengan diperkenalkannya robot Jerman dalam film *Metropolis* tahun 1926 yang sempat dipamerkan dalam *New York World's Fair 1939*. Film yang mengisahkan tentang robot berjalan mirip manusia beserta hewan peliharaannya. Kembali atas jasa insan film, istilah robot ini makin populer dengan lahirnya robot C3PO dalam film *Star Wars* pertama pada tahun 1977 (Pitowarno, 2006).

2.1.1 Perkembangan Robot

Menurut Fu, et al. (1987) penelitian dan pengembangan pertama yang berbuah produk robotik dapat dilacak mulai dari tahun 1940-an ketika *Argonne Nasional Laboratorius* di Oak Ridge, Amerika, memperkenalkan sebuah mekanisme robotik yang dinamai *master-slave manipulator* yang digunakan untuk menangani material radioaktif. Produk robot komersial pertama dikenalkan oleh *Unimation Incorporated*, Amerika, pada tahun 1950-an (Pitowarno, 2006).

Setelah dunia mulai menampak kejaman industri pada pertengahan tahun 1960-an kebutuhan akan otomasi makin menjadi-jadi, ketika itulah robotik diterima sebagai disiplin ilmu baru yang mendampingi ilmu-ilmu dasar dan teknik yang telah mapan sebelumnya. Hal inilah yang menyebabkan munculnya peneliti-peneliti dalam bidang robotik. Jepang menjadi negara termaju dalam melakukan penelitian dikarenakan Jepang melakukan penelitian teknologi infrastruktur seperti komponen dan piranti mikro yang akhirnya menjadi inti dari pengembangan robot modern (Pitowarno, 2006).

Aplikasi robot dalam dunia industri adalah robot pekerja industri atau berupa tangan dan lengan yang dikenal dengan istilah industrial robot dan robot manipulator. Manipulator adalah bagian mekanik yang dapat difungsikan untuk memindah, mengangkat dan memanipulasi benda kerja. Dalam dunia mekanikal, manipulator memiliki dua bagian yaitu tangan atau lengan (*arm*) dan pergelangan (*wrist*). Pada pergelangan dipasang berbagai tool sesuai kebutuhan dan kegunaannya (Pitowarno, 2006). Gambar 2.1 menunjukkan skema manipulator robot industri.



Gambar 2.1 Manipulator robot industri.

Sumber : Pitowarno (2006).

Secara umum kegunaan robot adalah menggantikan kerja manusia yang membutuhkan ketelitian tinggi, mempunyai resiko yang sangat besar dan mengancam keselamatan manusia. Robot industri adalah komponen utama dalam teknologi otomasi yang dapat berfungsi sebagai layaknya buruh atau pekerja manusia dalam pabrik namun memiliki kemampuan bekerja yang terus menerus tanpa lelah. Robot industri dapat diciptakan untuk menggantikan posisi-posisi pekerja dalam bagian produksi (Pitowarno, 2006).

Sebagai contoh dalam bidang produksi robot digunakan untuk tugas yang berat, berbahaya, pekerjaan yang berulang dan kotor seperti sebagai pembersihan limbah beracun, proses pengelasan (*welding*). Kedua bidang tersebut dalam

industri perakitan kendaraan memiliki resiko kecelakaan yang tinggi. Dengan adanya robot maka akan mengurangi resiko kecelakaan pada pekerja, bahkan bisa menghilangkan resiko kecelakaan. Gambar 2.2 menunjukkan proses pengelasan pada industri mobil.



Gambar 2.2 Proses pengelasan pada industri mobil.

Sumber : www.RequestArtikel.com_penggunaan komputer dipabrik mobil (2010).

2.1.2 Dasar Operasional Robot

Robot memiliki dasar operasional, yang terdiri dari beberapa komponen sebagai berikut (digilib.petra.ac.id/berkaki_enam-chapter2.pdf):

1. Mekanik robot merupakan anggota badan robot, yang terbuat dari bahan logam maupun dari bahan bukan logam, seperti: kayu, akrilik dan lain-lainnya.
2. Sumber energi atau catur daya dapat berupa listrik, hidrolik, dan lainnya.
3. Peralatan transmisi, dapat berupa kabel dan roda gigi. Alat-alat ini merupakan penghubung antar mekanik robot dengan sumber energi.
4. Otak dari robot merupakan suatu alat yang berfungsi untuk mengolahnya untuk melakukan suatu pekerjaan sesuai dengan program yang telah ditentukan, biasanya berupa suatu chip mikrokontroler.

2.2 *Gripper*

Gripper adalah sebuah *end effector* yang berfungsi untuk menggenggam dan menahan obyek. *End effector* berfungsi sebagai bagian terakhir yang menghubungkan antara manipulator dengan obyek yang akan dijadikan kerja dari robot (Sumbodo, 2008). Gambar 2.3 menunjukkan *end effector* tangan *gripper*.



Gambar 2.3 *End effector* tangan *gripper*.

Sumber : www.robotmatrix.org/robotics-hand.htm (2009).

Sebuah *end effector* robot adalah tangan dari lengan robot. Dengan melampirkan alat pada pergelangan tangan robot, lengan robot dapat melakukan tugas-tugas yang diberikan. Dalam banyak hal, *end effector* robot membutuhkan masukan listrik tambahan untuk beroperasi, tergantung pada jenis fungsi *end effector*. *End effector* yang banyak digunakan adalah pneumatik, karena lebih mudah untuk memberikan udara keujung lengan robot. Untuk aplikasi tertentu beberapa derajat umpan balik sensor yang diperlukan. Sebagai contoh penyisipan atau pencengkraman kekuatan pengukuran, sensor jarak untuk mendeteksi keberadaan benda-benda antar rahang *gripper*, tabrakan unit deteksi yang melekat antara pergelangan tangan robot dan *end effector* sehingga jika kekuatan yang

berlebihan diterapkan untuk alat robot lengan akan berhenti (www.robotmatrik.org/robotics-hand.htm, 2009).

2.2.1 Beberapa Jenis *Gripper*

a. *Gripper* Mekanik

Mechanical Gripper didesain untuk menggenggam dan menahan obyek dengan memberikan kontak pada obyek, pada umumnya menggunakan *finger* atau jari mekanik yang disebut dengan *jaws*. Jari dapat dilepas dan dipasang sehingga sangat fleksibel dalam pemakaiannya. Sumber tenaga bisa berupa *pneumatik*, *hidrolik* dan *elektrik* (Sumbodo, 2008). Contoh *gripper* mekanik dapat ditunjukkan pada Gambar 2.4 berikut ini.



Gambar 2.4 *Gripper* mekanik.

Sumber : Sumbodo (2008).

b. *Gripper* Magnetic

Gripper Magnetic bekerja karena efek bidang magnet, sehingga menimbulkan hisapan atau tarikan pada komponen yang akan ditangani. *Gripper Magnetic* dapat dibagi menjadi dua kategori, yaitu menggunakan elektromagnet dan menggunakan magnet permanen. Elektromagnet menggunakan sumber arus dan lebih mudah untuk dikontrol dibanding dengan menggunakan magnet permanen. Pada *gripper* magnet menggunakan elektromagnet saat menghisap dan melepas komponen, sehingga menimbulkan hisapan atau tarikan pada komponen

yang akan ditangani dengan menggunakan metode *on and off* arus yang mengalir pada *electromagnet* (Sumbodo, 2008). Contoh aplikasi *gripper* magnet dapat ditunjukkan pada Gambar 2.5 berikut ini.

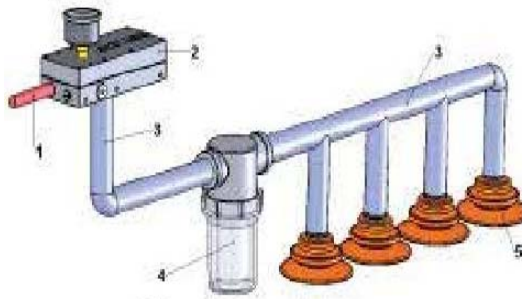


Gambar 2.5 *Gripper* Magnet.

Sumber : Sumbodo (2008).

c. *Gripper* Ruang Hampa (Mangkok Vakum)

Mangkok vakum disebut juga mangkok hisap digunakan sebagai *gripper* yang berfungsi untuk mengangkat dan menahan obyek. Obyek yang ditangani oleh jenis *gripper* ini adalah obyek rata, bersih, dan halus (Sumbodo, 2008). Contoh sistem vakum dapat ditunjukkan pada Gambar 2.6 berikut ini.



Gambar 2.6 Contoh sistem vakum.

Sumber : Sumbodo (2008).

Keterangan dari Gambar 2.6 sistem vakum yaitu :

1. Udara terkompresi
2. Pembangkit kevakuman
3. Aliran
4. Penyaring
5. Mangkok

Gripper ini memiliki dua komponen, yaitu : mangkok dan sistem ruang hampa. Mangkok berbatasan dengan obyek dan berfungsi untuk menggengam dan menahan obyek. Mangkok terbuat dari karet, dan membuat tekanan menghisap sehingga obyek akan menempel pada mangkok. Sistem vakum menghasilkan hisapan pada mangkok, dibagi menjadi dua yaitu pompa vakum dan sistem venturi. Pompa vakum menggunakan piston (dengan menggunakan motor listrik) untuk membuat hampa udara. Pompa memberikan kehampaan yang tinggi dibandingkan dengan sistem venturi. Sistem venturi menggunakan sebuah nosel yang dilewati oleh udara, sehingga menimbulkan kevakuman (Sumbodo, 2008). Sebagai contoh *gripper* vakum dapat ditunjukkan pada Gambar 2.7 berikut ini.



Gambar 2.7 *Gripper* Vakum

Sumber : Sumbodo (2008).

2.3 Motor Servo

Motor servo adalah motor yang mampu bekerja dua arah (CW dan CCW) dimana arah dan sudut pergerakan rotornya dikendalikan hanya dengan memberikan pengaturan *duty cycle* sinyal PWM pada bagian pin kontrolnya.

Motor servo merupakan sebuah motor DC yang memiliki rangkaian kontrol elektronik dan *internal* gear untuk mengendalikan pergerakan dan sudut angularnya (Sigit, 2007). Gambar motor servo yang digunakan dapat dilihat pada Gambar 2.8 berikut ini.



Gambar 2.8 Motor servo.

Motor servo adalah motor yang berputar lambat, dimana biasa ditunjukkan oleh *rate* putarannya yang lambat, namun demikian memiliki torsi yang kuat karena *internal* gearnya (Sigit, 2007). Sebuah motor servo terdiri dari komponen-komponen sebagai berikut:

1. Tiga jalur kabel : *power*, *ground*, dan *control*.
2. Sinyal kontrol mengendalikan posisi.
3. Operasional dari servo motor dikendalikan oleh sebuah pulsa selebar ± 20 ms, dimana lebar pulsa antara 0,5 ms dan 2 ms menyatakan akhir dari *range* sudut maksimum.
4. Kontruksi didalamnya meliputi *internal* gear, potensiometer, dan *feedback control*.

Motor servo terbagi dalam dua jenis, yaitu :

1. Motor Servo Standar 180°

Motor servo standar hanya mampu bergerak dua arah (CW dan CCW) dengan defleksi masing-masing sudut mencapai 90° sehingga total defleksi sudut dari kanan-tengah-kiri adalah 180°.

2. Motor Servo Continuous

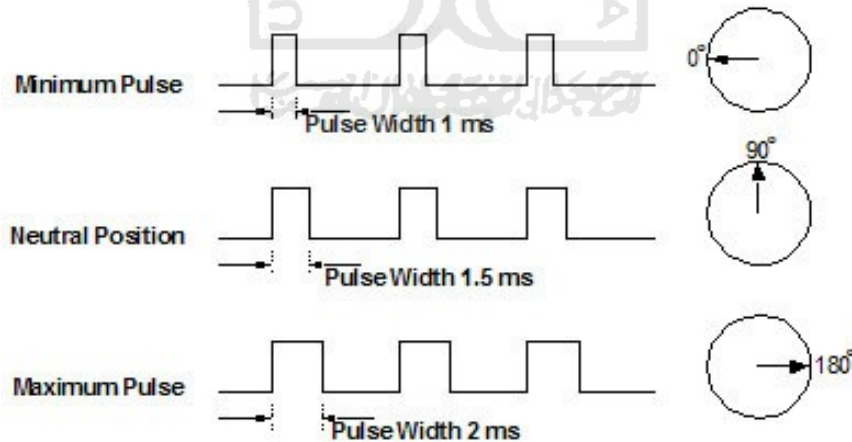
Motor *Servo Continuous* mampu bergerak dua arah (CW dan CCW) tanpa batasan defleksi sudut putar (dapat berputar searah kontinyu).

2.3.1 Pulse Width Modulation

Pengaturan posisi poros motor servo menggunakan sistem *pulse width modulation* (PWM). PWM adalah suatu teknik manipulasi dalam mengendalikan motor yang menggunakan prinsip *high-low* (Sigit, 2007). Sebagai contoh pemberian pulsa pada motor servo yaitu :

1. Bila diberikan pulsa dengan besar 1,5 ms mencapai gerakan 90 derajat (posisi tengah).
2. Bila diberikan pulsa kurang dari 1,5 ms maka posisi mendekati 0 derajat (posisi kiri).
3. Bila diberikan pulsa lebih dari 1,5 ms maka posisi mendekati 180 derajat (posisi kanan).

Hubungan pulsa dan posisi motor servo dapat ditunjukkan pada Gambar 2.9 berikut ini.

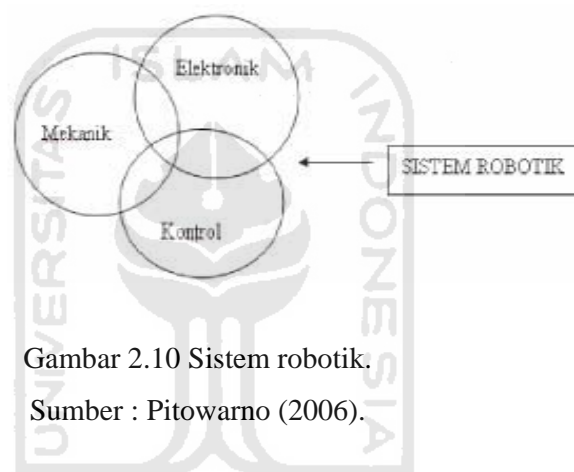


Gambar 2.9 Hubungan pulsa dan posisi motor servo.

Sumber : ([www.andifajar.com-pemrograman servo](http://www.andifajar.com-pemrograman-servo)).

2.4 Sistem Kontrol

Kontrol adalah bagian yang amat penting dalam robotik. Sistem robotik tanpa kontrol hanya akan menjadi benda mekatronik yang mati. Kontrol merupakan rangkaian elektronik yang setidaknya-tidaknya terdiri dari rangkaian prosesor ataupun mikrokontroler, pengondisi sinyal untuk sensor analog dan digital, dan driver untuk aktuator. Bila diperlukan dapat dilengkapi dengan sistem monitor seperti seven segment, LCD (*liquid crystal display*), atau CRT (*cathode ray-tube*) (Pitowarno, 2006). Sistem robotik dapat dilihat pada Gambar 2.10 berikut ini.



Gambar 2.10 Sistem robotik.

Sumber : Pitowarno (2006).

Dalam Gambar sistem kontrol robotik, kontrol adalah bagian yang tak terpisahkan dalam sistem robotik. Dalam hal ini sistem kontrol bertugas mengkolaborasikan sistem elektronik dengan mekanik dengan baik agar mencapai fungsi seperti yang dikehendaki (Pitowarno, 2006).

2.4.1 Dasar Sistem Kontrol Robotik

Sistem kontrol robotik pada dasarnya terbagi dalam dua kelompok yaitu, sistem kontrol loop terbuka dan sistem kontrol loop tertutup.

1. Sistem Kontrol Loop Terbuka

Sistem kendali loop terbuka (*open loop*) umumnya mempergunakan pengatur (kontroler) serta aktuator kendali (kontrol aktuator) yang berguna untuk memperoleh respon sistem yang baik. Sistem kendali ini keluarannya tidak

diperhitungkan ulang oleh kontroler. Suatu keadaan apakah robot benar-benar telah mencapai target seperti yang dikehendaki masukan atau referensi, tidak dapat mempengaruhi kinerja kontroler. Oleh karena itu pengendalian sistem loop terbuka hanya bisa digunakan jika hubungan antara masukan dan keluaran diketahui dan tidak ada gangguan *eksternal* dan *internal* (Pitowarno, 2006).

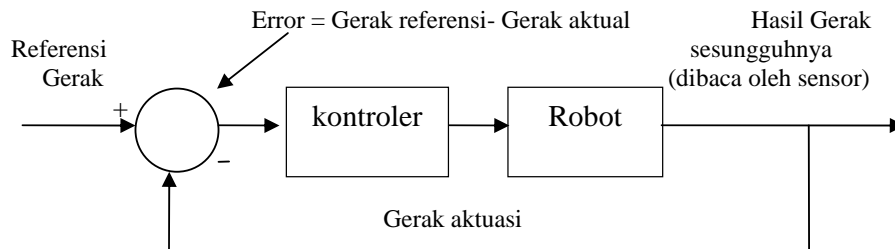
Pada sistem ini komponennya hanya ada masukan kontroler, aktuator dan keluaran, misalnya memiliki urutan sebagai berikut: menuju keposisi obyek, mengambil obyek, mengangkat obyek, memindahkan obyek keposisi tertentu, dan meletakkan obyek. Dapat saja langkah menuju posisi obyek dan memindahkan obyek menuju posisi akhir adalah gerak berdasarkan loop tertutup (Pitowarno, 2006). Diagram kontrol loop terbuka dapat ditunjukkan pada Gambar 2.11.



Gambar 2.11 Diagram kontrol loop terbuka.

Sumber : Pitowarno (2006).

2. Sistem Kontrol Loop Tertutup



Gambar 2.12 Diagram kontrol loop tertutup.

Sumber : Piowarno (2006).

Pada Gambar 2.12, jika hasil gerak aktual telah sama dengan referensi maka input kontroler akan nol. Artinya kontroler tidak memberikan sinyal aktuasi

kepada robot karena target akhir perintah gerak telah diperoleh. Makin kecil *error* terhitung maka makin kecil pula sinyal pengemudian kontroler terhadap robot, sampai akhirnya mencapai kondisi tenang. Referensi gerak dan gerak aktual dapat berupa posisi, kecepatan, akselerasi, atau gabungan diantaranya. Kontrol bersifat konvergen jika dalam rentang waktu pengontrolan nilai error menuju nol, dan keadaan dikatakan stabil jika setelah konvergen kontroler mampu menjaga agar error selalu nol (Pitowarno, 2006).

Jika referensinya adalah kecepatan maka disebut stabil jika pada keadaan tenang kecepatan akhir sama dengan referensi dan kontroler mampu menjaga 'kesamaan' pada masa-masa berikutnya. Dalam hal kecepatan, keadaan tenang yang dimaksud adalah bukan berarti keluaran kontroler bernilai nol (tegangan nol volt) seperti keadaan sesungguhnya pada kontrol posisi, namun kontroler tidak lagi memberikan penguatan atau pelemahan pada actuator (Pitowarno, 2006).

2.4.2 Sistem Kontrol Berbasis Mikrokontroler

Mikrokontroler pada dasarnya adalah komputer dalam satu chip, yang didalamnya terdiri dari, mikroprosesor, memori, jalur Input/Output (I/O) dan unit pendukungnya, misalnya ADC (*Analog to Digital Converter*).

CPU pada mikrokontroler berupa mikroprosesor yang berfungsi sebagai otak dari mikrokontroler. Dalam suatu mikrokontroler terdapat tiga buah memori, yaitu RAM, ROM, dan EEPROM. RAM dan ROM hampir selalu ada pada setiap mikrokontroler, sedangkan EEPROM hanya terdapat pada beberapa jenis mikrokontroler tertentu (modul praktikum mekatronika, 2009).

1. RAM (*Random Acces Memory*)

RAM digunakan sebagai penyimpan data sementara yang berupa register-register. Register adalah tempat penyimpanan data, misalnya variable dalam program, keadaan input/output, serta pengaturan timer/counter dan komunikasi serial. Data yang ada pada RAM akan hilang bila catu daya dari RAM dimatikan sehingga RAM hanya dapat digunakan untuk menyimpan data sementara.

2. ROM (*Read Only Memory*)

ROM digunakan sebagai tempat penyimpanan program, data yang disimpan di ROM tidak akan hilang meskipun tegangan supply dimatikan. hanya dapat dihapus dan ditulis secara sekaligus. Data yang disimpan di ROM tidak akan hilang meskipun tegangan supply dimatikan.

2. EEPROM

EEPROM biasanya digunakan untuk menyimpan data, EEPROM tetap dapat menyimpan data saat tidak dicatu daya dan juga dapat diubah saat program berjalan.

3. Bus Alamat

Bus alamat berfungsi sebagai sejumlah lintasan pengalamatan antara alat dengan sebuah komputer, ditentukan terlebih dahulu untuk menghindari terjadinya kesalahan pengiriman sebuah intruksi dan terjadinya bentrok antara dua buah alat yang bekerja secara bersamaan.

4. Bus Data

Bus Data merupakan sebuah lintasan saluran keluar masuknya data dalam suatu mikrokontroler. Pada umumnya saluran data yang masuk sama dengan saluran data keluar.

5. Bus Kontrol

Bus kontrol ataupun bus pengendali ini berfungsi untuk menyerempakkan operasi mikrokontroler dengan operasi rangkaian luar.

2.5 ATMega 8535

ATMega 8535 adalah salah satu jenis mikrokontroler keluarga AVR buatan Atmel, menggunakan arsitektur RISC (*Reduced Instruction Set Computing*) dengan lebar pulsa 8 bit, dimana semua intruksi dikemas dalam kode 16-bit dan sebagian besar intruksi dikemas dalam satu siklus clock. (Wardhana, 2006).

2.5.1 Arsitektur ATmega 8535

Arsitektur ATmega 8535 memiliki bagian yaitu :

1. Saluran I/O sebanyak 32 buah, yaitu Port A, Port B, Port C, dan Port D.
2. ADC 10 bit sebanyak 8 saluran.
3. Tiga buah Timer/Counter dengan kemampuan pembandingan.
4. CPU yang terdiri atas 32 buah register.
5. Watchdog Timer dengan osilator internal.
6. SRAM sebesar 512 byte.
7. Memori Flash sebesar 8 kb dengan kemampuan Read While Write.
8. Unit Interupsi internal dan eksternal.
9. Port antar muka SPI.
10. EEPROM sebesar 512 byte yang dapat diprogram saat operasi.
11. Antarmuka komparator analog.
12. Port USART untuk komunikasi serial.

2.5.2 Fitur ATmega 8535

Kapabilitas detail dari ATmega 8535 adalah:

1. Sistem mikroprosesor 8 bit berbasis *RISC* dengan kecepatan maksimal 16 MHz.
2. Kapabilitas memori flash 8 KB, SRAM 512 byte, dan EEPROM (*Electrically Erasable Programmable Read Only memory*) sebesar 512 byte.
3. ADC internal dengan fidelitas 10 bit sebanyak 8 channel.
4. Portal komunikasi serial (USART) dengan kecepatan maksimal 2,5 Mbps.
5. Enam pilihan mode *sleep* menghemat penggunaan daya listrik.

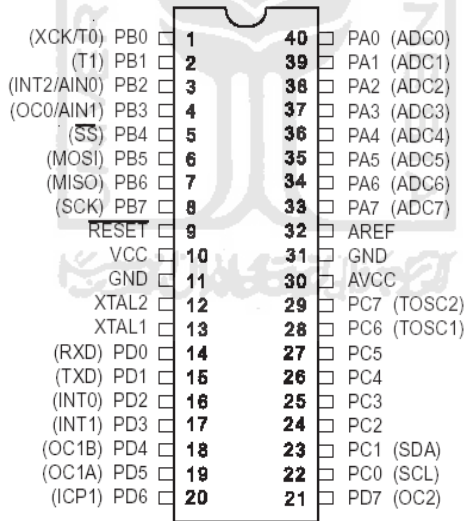
2.5.3 Konfigurasi Pin ATmega 8535

Konfigurasi pin ATmega 8535 terdiri dari :

1. VCC merupakan pin yang berfungsi sebagai pin masuknya catu daya.

2. GND merupakan pin ground.
3. Port A (PA0..PA7) merupakan pin I/O dua arah dan pin masuk *ADC*.
4. Port B (PBO..PB7) merupakan pin I/O dua arah dan pin fungsi khusus, yaitu Timer/Counter, komparator analog, dan SPI.
5. Port C (PCO..PC7) merupakan pin I/O dua arah dan pin fungsi khusus, yaitu TWI, komparator analog, dan Timer Oscilator.
6. *Port D* (PDO..PD7) merupakan pin I/O dua arah dan pin fungsi khusus, yaitu komparator analog, interuksi eksternal, dan komunikasi serial.
7. RESET merupakan pin yang digunakan untuk me-reset mikrokontroler.
8. XTAL1 dan XTAL2 merupakan pin masukan *clock* eksternal.
9. AVCC merupakan pin masukan tegangan untuk ADC.
10. AREF merupakan pin masukan tegangan referensi ADC.

Gambar 2.13 berikut ini merupakan konfigurasi pin ATMega 8535.

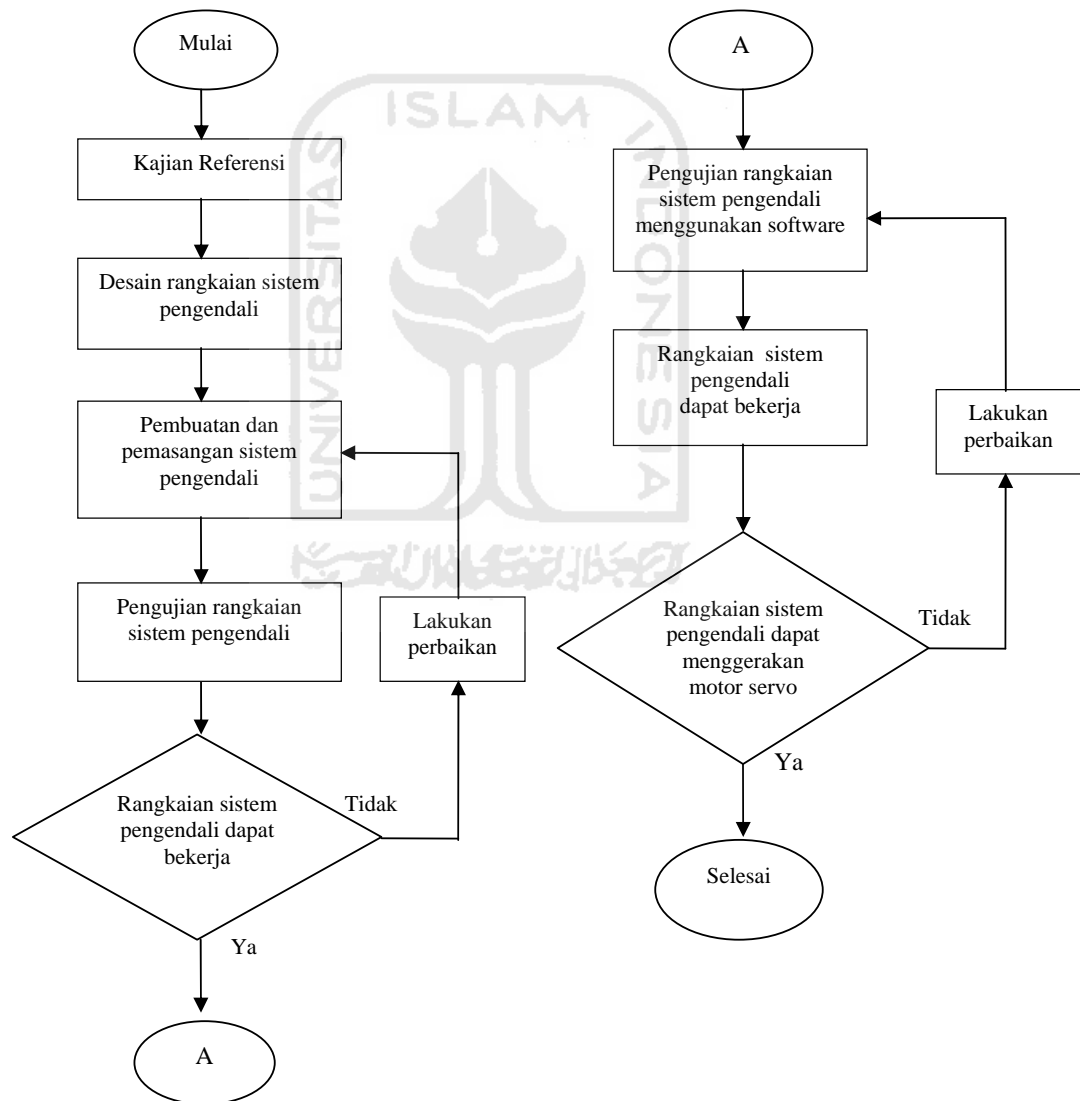


Gambar 2.13 Konfigurasi pin ATMega 8535 (Data Chip ATMega 8535).

BAB III METODOLOGI PENELITIAN

3.1 Alur Penelitian

Dalam perancangan rangkaian sistem kendali penggerak pada jari tangan robot melalui beberapa tahapan yang dilalui, ditunjukkan pada gambar 3.1.



Gambar 3.1 Diagram alir.

3.2 Kajian Referensi

Sebagai bahan acuan untuk membuat sistem kendali penggerak pada jari tangan robot, referensi diambil dari beberapa sumber yang berhubungan dengan tangan robot, seperti buku, *ebook*, *website*, dan lain-lain.

3.3 Alat dan Bahan

3.3.1 Alat

Peralatan yang digunakan dalam penelitian ini adalah :

1. Gunting atau cutter.
2. Amplas.
3. Bor PCB.
4. Obeng.
5. Tang.
6. Solder.

3.3.2 Bahan

Bahan bahan yang digunakan adalah :

1. Model tangan robot (resin).
2. Rangkaian pengendali motor servo.
3. Tali senar.
4. Motor servo.

3.4 Bagian Bagian Tangan Robot

3.4.1 Model Tangan Robot

Model tangan robot ini telah dibuat sebelumnya (Adityo, 2011). Tangan robot ini dibuat menyerupai tangan manusia, dimana bentuk dari jari-jari tangannya menyerupai jari tangan manusia dan menggunakan lima buah motor servo sebagai penggerak, serta tali senar sebagai mekanis penggeraknya. Tangan robot dibentuk dengan menggunakan salah satu metode *layer deposition manufacturing*. Metode ini yaitu dengan membagi desain suatu *part* menjadi

beberapa layer, yang kemudian akan disusun menjadi satu bagian. Bagian-bagian tangan robot yang telah disatukan dapat ditunjukkan pada Gambar 3.2 berikut ini.



Gambar 3.2 Tangan robot.

3.4.2 Otot Penggerak pada Jari Tangan Robot

Otot penggerak menggunakan Tali senar, pada jari tangan robot diberikan jalur agar senar yang melewati rangka tidak saling mengganggu dan bersentuhan. Otot penggerak yang digunakan dapat ditunjukkan pada Gambar 3.3 berikut ini.



Gambar 3.3 Tali senar sebagai otot penggerak.

3.4.3 Motor Penggerak

Motor servo digunakan sebagai penggerak, untuk menggerakkan jari-jari tangan robot. Fungsi dari motor servo adalah untuk menggerakkan jari-jari tangan robot agar dapat bergerak menggenggam dan membuka kembali keposisi semula. Motor servo membutuhkan daya listrik 6 volt untuk menggerakannya serta dapat menghasilkan torsi hingga 2,2 kg.cm. Motor servo yang digunakan dapat dilihat pada Gambar 3.4 berikut ini.



Gambar 3.4 Motor servo.

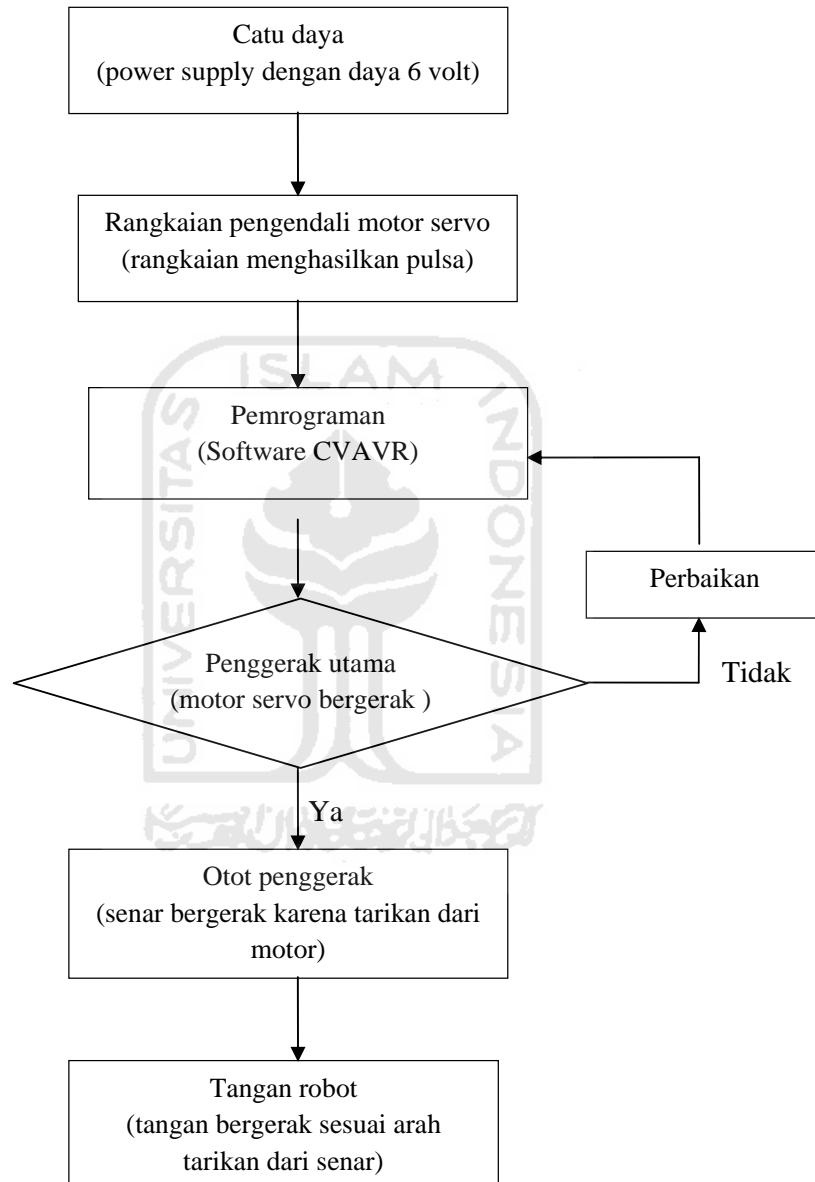
Motor servo yang digunakan memiliki spesifikasi sebagai berikut:

- Dimensi 22,8 mm (panjang), 12,2 mm (lebar), 28,5 mm (tinggi).
- Berat 13,4 gram.
- Torsi 1,8 kg.cm - 2,2 kg.cm.
- Tegangan 4,8V - 6V.

3.5 Perakitan Sistem Kendali Penggerak Tangan Robot

Gerak yang ingin dihasilkan sama dengan gerak tangan manusia yaitu menggenggam dan membuka kembali. Untuk menggerakkan jari tangan robot

melalui beberapa tahapan seperti skema sistem kendali penggerak pada Gambar 3.5 berikut ini.



Gambar 3.5 Skema sistem kendali penggerak tangan robot.

3.5.1 Pemasangan Otot Penggerak

Otot sebagai penggerak jari tangan robot menggunakan senar yang akan ditarik oleh motor servo, senar dimasukan didalam jalur pada lubang yang telah dibuat pada bagian-bagian tangan robot. Untuk jalur lubang senar pada rangka menggunakan *Cotton swab* (batang plastik yang terdapat pada permen), jalur senar berfungsi agar senar yang melewati rangka tidak saling mengganggu dan bersentuhan. Pemasangan otot penggerak yaitu pada telapak tangan, sendi-sendi pada jari tangan dan disimpul mati pada ujung jari tangan robot. Pemasangan otot penggerak dapat dilihat pada Gambar 3.6 berikut ini.



Gambar 3.6 Pemasangan otot penggerak pada jari tangan robot.

3.5.2 Pemasangan Motor Servo

Motor servo digunakan sebagai penggerak utama yang diletakan didalam lengan sehingga tangan robot terlihat lebih rapih. Untuk posisi motor servo disusun secara tidak searah yaitu berlawanan, ini dilakukan untuk menghindari tabrakan antara *horn* dengan senar pada saat motor servo berputar. Apabila saat motor servo bergerak dan senar bersentuhan dengan *horn* maka pergerakan motor servo akan menjadi tidak maksimal, karena senar tertahan terlebih dahulu

kemudian berjalan normal, sehingga tarikan motor servo menjadi tidak sempurna (Gambar 3.7a), dan (Gambar 3.7b) merupakan posisi motor servo pada lengan robot.



Gambar 3.7(a) Pemasangan motor servo pada lengan robot.



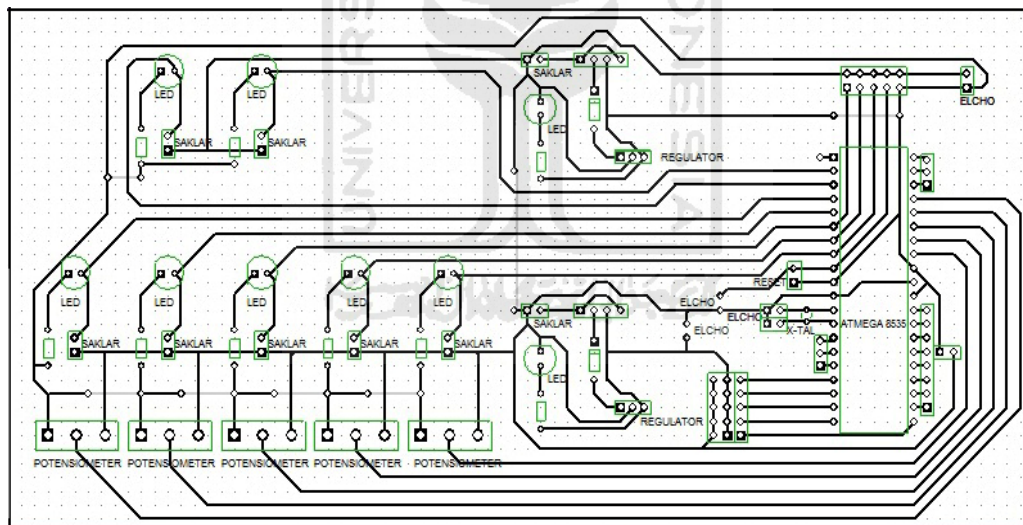
Gambar 3.7(b) Posisi motor servo pada lengan robot.

3.6 Pembuatan Rangkaian Mikrokontroler

Pembuatan rangkaian mikrokontroler menggunakan alat dan bahan seperti gunting atau cutter, amplas, bor PCB, tang, solder, amplas, dan bahan yang digunakan yaitu PCB, motor servo, tali senar, rangkaian pengendali motor servo dan lain-lainnya. Dalam proses pembuatan rangkaian mikrokontroler melalui beberapa tahapan dari proses pembuatan desain rangkaian, pelarutan desain hingga proses penyolderan dan pemasangan.

3.6.1 Desain Rangkaian

Desain rangkaian menggunakan *software PCB Layout*. *Software PCB Layout* digunakan untuk mempermudah dalam perangkaian desain. Desain rangkaian dapat dilihat pada Gambar 3.8 berikut ini.

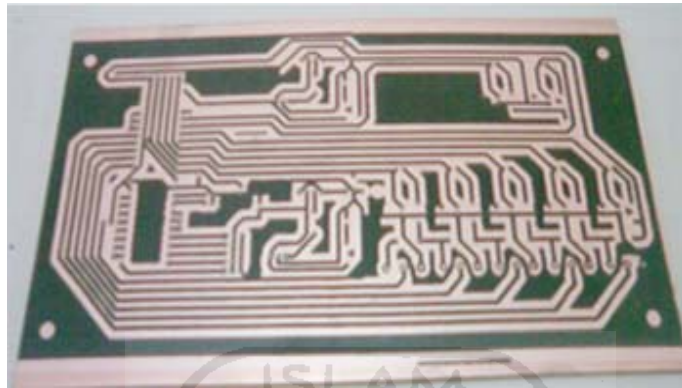


Gambar 3.8 Desain rangkaian.

3.6.2 Pemindahan Desain Rangkaian

Pemindahan desain rangkaian pada PCB, yaitu mencetak jalur rangkaian dengan menggunakan fotokopi berbahan plastik. Agar jalur rangkaian membentuk

pada PCB digunakan setrika sebagai proses pemindahan jalur rangkaian. Desain rangkaian yang telah berpindah pada PCB dapat dilihat pada Gambar 3.9.



Gambar 3.9 Desain rangkaian yang telah berpindah pada PCB.

3.6.3 Pelarutan Rangkaian

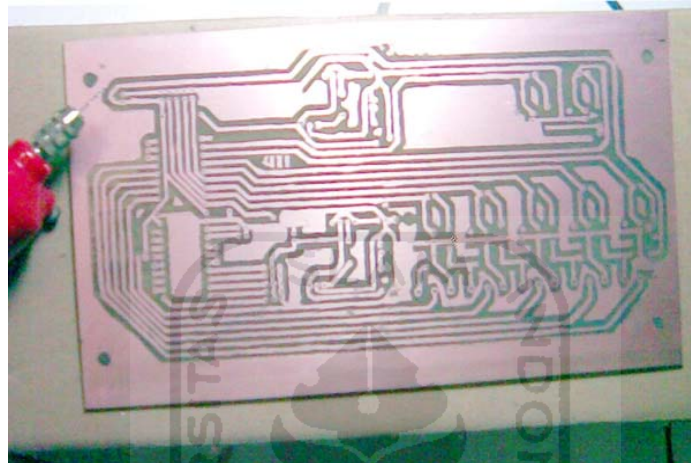
Proses pelarutan desain rangkaian menggunakan bahan pelarut PCB. Fungsi pelarut untuk melarutkan PCB dan yang tertinggal pada PCB hanya desain rangkaian. Proses pelarutan dapat ditunjukkan pada Gambar 3.10 berikut ini.



Gambar 3.10 Pelarutan desain rangkaian pada PCB.

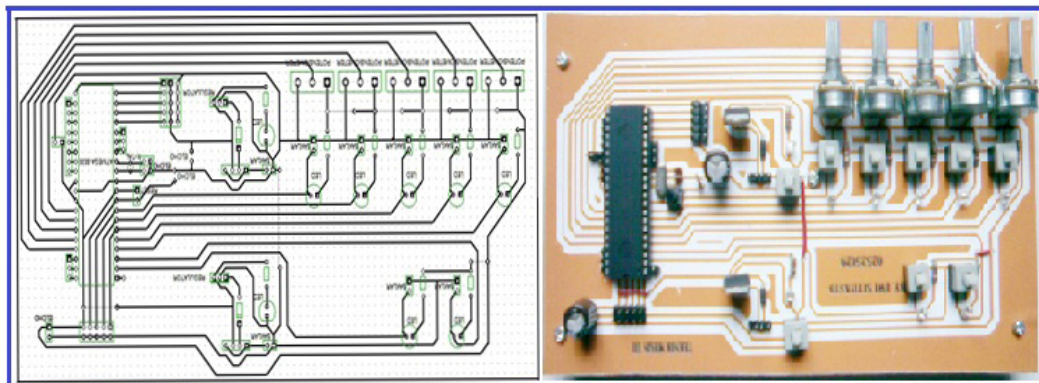
3.6.4 Pelubangan Desain Rangkaian

Proses pengeboran lubang kaki-kaki komponen pada PCB menggunakan bor PCB. Kemudian pemasangan komponen pada jalur rangkaian dan proses penyolderan kaki komponen. Pelubangan seperti pada Gambar 3.11 berikut ini.



Gambar 3.11 Proses pengeboran lubang kaki komponen.

Selanjutnya pemasangan komponen, komponen dipasang dan disolder. Komponen yang telah dipasang dapat ditunjukkan pada Gambar 3.12.



Gambar 3.12 komponen yang telah dirangkai.

3.6.5 Pengujian Rangkaian

Rangkaian mikrokontroler yang telah dirangkai harus melalui pengujian. Pengujian dilakukan untuk mengetahui rangkaian sudah dapat bekerja atau tidak dan untuk mengetahui ada atau tidaknya kesalahan dalam proses pemasangan. Proses pengujian menggunakan *power supply* sebagai catu daya, kabel-kabel penghubung sebagai penghubung arus listrik pada rangkaian. Arus 12 volt dari catu daya akan dikonversi oleh regulator menjadi 5 volt dan menuju mikrokontroler (ATMega 8535). Arus positif 5 volt dari catu daya masuk pada *port* catu daya arus positif rangkaian dan arus negatif dari catu daya masuk pada *port* catu daya arus negatif rangkaian. Dapat dilihat pada Gambar 3.13 berikut ini.



Gambar 3.13 Rangkaian regulator.

3.6.6 Posisi Pin Motor Servo Pada Rangkaian

Rangkaian arus motor servo menggunakan *port* D pada kaki ATMega 8535. Pada perakitanya sesuai dengan urutan motor servo, karena menggunakan 5 motor servo. Untuk menggerakkan motor servo membutuhkan arus sebesar 5 volt. Pada motor servo terdiri dari tiga kabel yaitu arus positif masuk *port* tengah (lubang bagian tengah pin motor servo) kabel warna merah, arus negative masuk

port kiri (lubang bagian kiri pin motor servo) kabel warna coklat dan kontrol masuk *port* kanan (lubang bagian kanan pin motor servo) kabel warna kuning. Seperti pada Gambar 3.14 berikut ini.

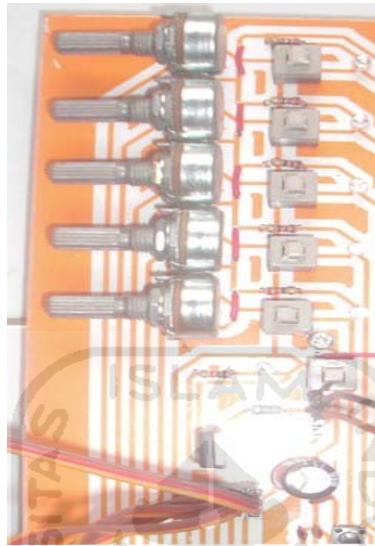


Gambar 3.14 Rangkaian arus motor servo dan pin motor servo.

3.6.7 Potensiometer

Rangkaian potensiometer terletak pada port A, terdapat tiga kaki yaitu kaki kiri dialiri arus positif, kaki bagian tengah sebagai kontrol dan kaki bagian kanan dialiri arus negatif yang dihubungkan pada mikrokontroler. Untuk melihat nilai potensiometer yang digunakan dan untuk mempermudah dalam proses pengukuran derajat putaran motor servo maka dilengkapi dengan LCD. Cara

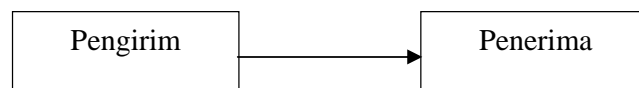
menjalankan potensiometer yaitu dengan memutar potensiometer kearah kiri dan kanan. Potensiometer yang digunakan seperti pada Gambar 3.15 berikut ini.



Gambar 3.15 Rangkaian Potensiometer.

3.7 Komunikasi Serial

Untuk melaksanakan komunikasi serial, *byte* data harus dikonversikan menjadi bit serial menggunakan paralel in serial *out shift register*. Setelah itu data ditransmisikan menjadi jalur data tunggal, untuk kemudian pada penerima data diubah menjadi *byte* data kembali menggunakan serial in paralel *out shift register*. Komunikasi serial dapat ditunjukkan pada Gambar 3.16 berikut ini.



Gambar 3.16 Komunikasi serial.

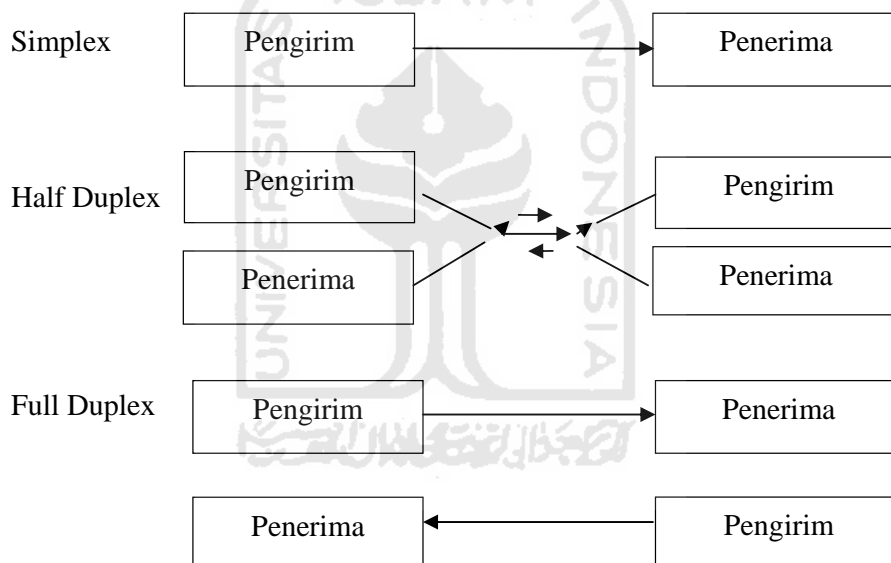
Sumber : Adi (2009).

Pada jarak dekat komunikasi serial dapat dilakukan sederhana, yaitu komunikasi antara keyboard dengan motherboard.

Pengiriman dan penerimaan data, transmisi data dapat dibagi yaitu :

1. Transmisi simplex, komunikasi dilakukan satu arah, satu peranti berfungsi sebagai pengirim dan peranti lainnya sebagai penerima.
2. Transmisi duplex, komunikasi dilakukan dua arah, pada pada masing-masing peranti terdapat pengirim dan penerima. Transmisi duplex, dibagi menjadi dua, yaitu :
3. Full duplex, data dapat dikirim dan diterima secara bersamaan.
4. Half duplex, data hanya dapat dikirim atau diterima saja pada waktu bersamaan.

Gambar 3.17 dibawah ini menunjukkan sistem transmisi simplex dan duplex



Gambar 3.17 Transmisi Simplex dan duplex.

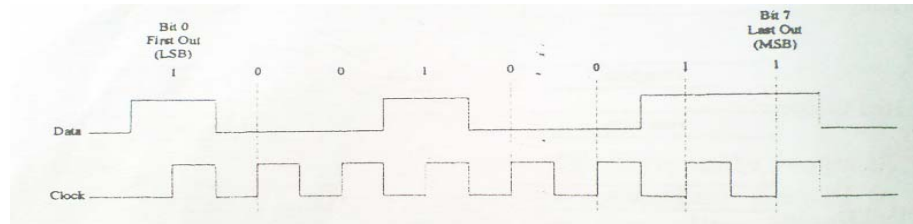
Sumber : Adi (2009).

Terdapat dua metode yang digunakan pada komunikasi serial, yaitu sinkron dan asinkron.

a. Serial Sinkron

Pada komunikasi serial sinkron, selain jalur data terdapat satu jalur tambahan yang dipergunakan sebagai clock yang berfungsi untuk mengkoordinasi

data yang dikirim atau diterima. Gambar 3.18 merupakan contoh jalur komunikasi sinkron.

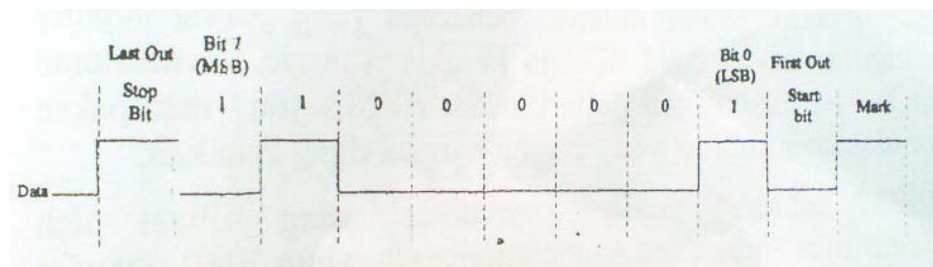


Gambar 3.18 Komunikasi serial sinkron.

Sumber : Adi (2009).

b. Serial Asinkron

Komunikasi serial asinkron mempunyai satu buah jalur data untuk input/output. Data diletakkan diantara start bit dan stop bit. Saat tidak ada transfer data sinyal berada pada kondisi high, atau disebut mark. Transmisi dimulai dengan start bit, selanjutnya diikuti dengan data yang dikirim berturut-turut mulai LSB hingga MSB dan diakhiri dengan stop bit. Pada komunikasi asinkron piranti dapat diprogram untuk data dengan lebar data 7 sampai 8 bit, sedangkan stop bit dapat berjumlah 1 atau 2. Pengiriman data yang umum dilakukan berupa data ASCII 8 bit dan diakhiri dengan 1 buah stop bit. Gambar 3.19 merupakan contoh jalur komunikasi asinkron.



Gambar 3.19 Komunikasi serial asinkron.

Sumber : Adi (2009).

3.7.1 RS232

Supaya berbagai peranti dapat berkomunikasi dengan lancar, maka dibuat standar-standar komunikasi. Saat ini terdapat banyak standar komunikasi, beberapa yang paling populer adalah TTL dan RS-232.

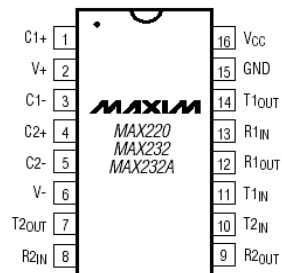
RS-232 adalah standar komunikasi yang dibuat oleh *Electronics Industries Association* pada tahun 1960. Standar ini telah dipergunakan secara luas pada mini computer, modem, dan berbagai macam peralatan sejak masa sebelum PC berkembang. Karena RS-232 dikembangkan sebelum era TTL, level tegangan *input* dan *output* nya berbeda dan tidak kompatibel dengan level tegangan TTL. Perbedaan logika RS-232 dengan logika TTL dapat ditunjukkan pada Tabel 4.1.

Tabel 4.1 Perbedaan logika RS-232 dengan logika TTL.

Logika	TTL	RS232
Logika 0	0 V	+3 hingga +25V
Logika 1	5 V	-3 hingga -25V

Sumber : Adi (2009).

Untuk menghubungkan RS232 dengan mikrokontroler (yang mempunyai level tegangan TTL 0 V dan 5 V) maka diperlukan converter tegangan. Converter tegangan TTL dan RS-232 yaitu IC MAX232. Gambar 3.20 merupakan IC Max 232.



Gambar 3.20 IC Max 232.

Sumber : Adi (2009).

Sebagai penghubung pada komunikasi serial dipergunakan kabel serial menggunakan soket DB9 yang mempunyai 9 pin. Fungsi dari masing-masing pin dapat ditunjukkan pada Tabel 4.2 berikut ini.

Tabel 4.2 Fungsi pin pada konektor DB9

Pin	Description
1	Data carrier detect (DCD)
2	Received data (RxD)
3	Transmitted data (TxD)
4	Data terminal ready (DTR)
5	Signal ground (GND)
6	Data set ready (DSR)
7	Request to send (RTS)
8	Clear to send (CTS)
9	Ring indicator (RI)

Sumber : Mazidi & Mazidi (2000).

Gambar 3.21 berikut ini merupakan posisi pin pada konektor DB9.



Gambar 3.21 Pin pada konektor DB9.

Untuk koneksi paling sederhana cukup digunakan 3 pin, yaitu TxD, RxD, dan GND. Pada MAX232 terdapat 2 pin yang digunakan untuk transfer data, yaitu TxD untuk mengirimkan data dan RxD untuk menerima data. Pada

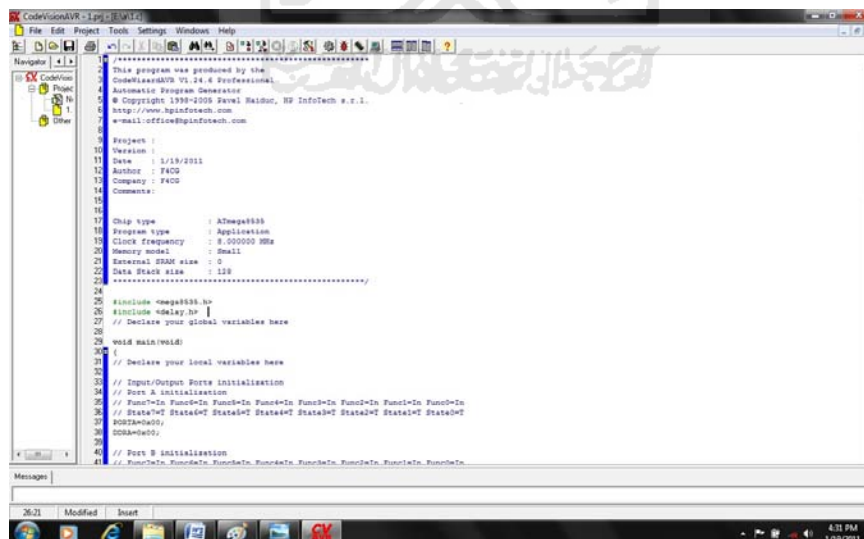
mikrokontroler pin yang dipergunakan adalah pin P3.1 (sebagai Tx) dan pin 3.0 (sebagai Rx). Untuk dapat bekerja rangkaian memerlukan 4 buah kapasitor dengan nilai antara 1 hingga 22 μ F.

3.8 Code Vision AVR

Untuk program suatu mikrokontroler terdapat banyak bahasa pemrograman yang digunakan. Bahasa pemrograman yang digunakan dalam pemrograman ini adalah Code Vision AVR (CVAVR). Code Vision AVR (CVAVR) adalah kompiler C. ATmega 8535 adalah salah satu jenis mikrokontroler keluarga AVR dan dalam pemrogramannya dengan menghubungkan pin VCC + 5V, pin GND 0V, pin kontrol dengan port mikrokontroler.

Langkah untuk membuat program yaitu: membuka program Code Vision AVR, mengatur konfigurasi chip ATmega, nilai clock, port yang digunakan kemudian *Generate, save dan exit*.

Contoh pemrograman Code Vision AVR dapat dinyatakan pada Gambar 3.22 berikut ini.



```
10 *****
11 This program was produced by the
12 CodeVisionAVR V1.24.6 Professional
13 Automatic Program Generator
14 Copyright 1998-2006 Ravel Maidor, HP InfoTech s.r.l.
15 http://www.hpinform.com
16 email:office@hpinform.com
17
18 #include <avr/io.h>
19 #include <delay.h>
20 // Declare your global variables here
21
22 void main(void)
23 {
24 // Declare your local variables here
25
26 // Input/Output Ports Initialization
27 // Port B initialization
28 // Func0=In Func1=In Func2=In Func3=In Func4=In Func5=In Func6=In
29 // State0=I State1=I State2=I State3=I State4=I State5=I State6=I State7=I
30 // DDRB=0x00;
31 // Port B initialization
32 // Func0=In Func1=In Func2=In Func3=In Func4=In Func5=In Func6=In Func7=In
```

Gambar 3.22 Program Code Vision AVR.

BAB IV

ANALISA DAN PEMBAHASAN

Penelitian ini bertujuan untuk membuat rangkaian sistem pengendali dan mengaplikasikannya pada jari tangan robot terhadap derajat putaran motor servo. Pembuatan rangkaian yang dilakukan untuk mengetahui posisi jari tangan robot pada saat motor servo melakukan putaran, dan putaran motor servo pada busur derajat.

4.1 Peralatan

Peralatan yang digunakan dalam pengujian rangkaian pengendali yaitu :

1. *Power supply* (catu daya).
2. Kabel-kabel penghubung.
3. Rangkaian downloader (catu daya).
4. Serial RS232

4.2 Pemrograman

4.2.1 Program Code Vision AVR

Untuk menjalankan program, masukkan kode program yang akan deprogram. Peneliti menggunakan ATmega 8535 maka data yang ada adalah data ATmega 8535. Kemudian masukan kode program yang akan digunakan dan coba program (*make the project*) , open pada extreme burner AVR, ambil program kemudian download (*write all*).

Seperti contoh penggalan program berikut ini:

```
*****  
#include <mega8535.h>  
#include <delay.h>  
char data_LCD [16];  
int z,a,b,c,d,e;  
char i;  
// Alphanumeric LCD Module functions
```

```

#asm
.equ __lcd_port=0x15 ;PORTC
#endasm
#include <lcd.h>
#include <stdio.h>

```

Posisi pin motor servo pada port D dan nilai pwm 2400 pada posisi membuka, seperti pada penggalan kode program berikut ini.

```

#define servo1_pin PORTD.7
#define servo2_pin PORTD.6
#define servo3_pin PORTD.5
#define servo4_pin PORTD.4
#define servo5_pin PORTD.3
#define servo6_pin PORTD.2
unsigned int pwm1=2400;
unsigned char servo_index=0;

```

Program akan mengeksekusi kode program yang ditulis, yaitu program akan membaca *if (servo_index==0)*, jika kode program tersebut tidak terbaca maka program akan mengeksekusi kode program berikutnya *else if (servo_index==1)* dan selanjutnya sampai kode program terakhir yang dituliskan. Karena cara pembacaannya sama maka ditulis satu saja. Seperti pada penggalan kode program berikut ini.

```

//Timer 1 output compare A interrupt service routine
interrupt [TIM1_COMPA] void timer1_compa_isr(void)
{
    if(servo_index==0)
    {
        servo1_pin=1;
        OCR1A=pwm1;
        servo_index=1;
    }
}

```

Untuk menampilkan nilai dan tulisan pada LCD, pada nilai (0,0) yaitu tampilan pada kolom pertama baris pertama. Sedangkan pada nilai (0,1) yaitu tampilan pada kolom pertama tetapi ditampilkan pada baris kedua, seperti pada penggalan kode program berikut ini.

```
sprintf(data_LCD, "%i-%i-%i",a,b,c);  
lcd_gotoxy(0,0);  
lcd_puts(data_LCD);  
sprintf(data_LCD, "%i-%i",d,e);  
lcd_gotoxy(0,1);  
lcd_puts(data_LCD);
```

Pada nilai $(7.058 * \text{read_adc}(7)) + 600$ merupakan hasil perhitungan menggunakan Microsoft Office Excel, untuk mengetahui nilai yang ada pada potensiometer yaitu dari putaran awal potensiometer 600 sampai titik putaran maksimum potensiometer 2400. Seperti pada penggalan kode program berikut ini.

```
pwm1=(7.058*read_adc(7))+600;
```

Pada nilai $(0.1 * \text{pwm1}) - 60$ merupakan hasil perhitungan dengan menggunakan Microsoft Office Excel, untuk menentukan nilai derajat pada motor servo yang akan terlihat pada LCD dimana nilai yang digunakan yaitu 0 sampai 180. Untuk menjalankannya dengan memutar potensiometer. Seperti pada penggalan kode program berikut ini.

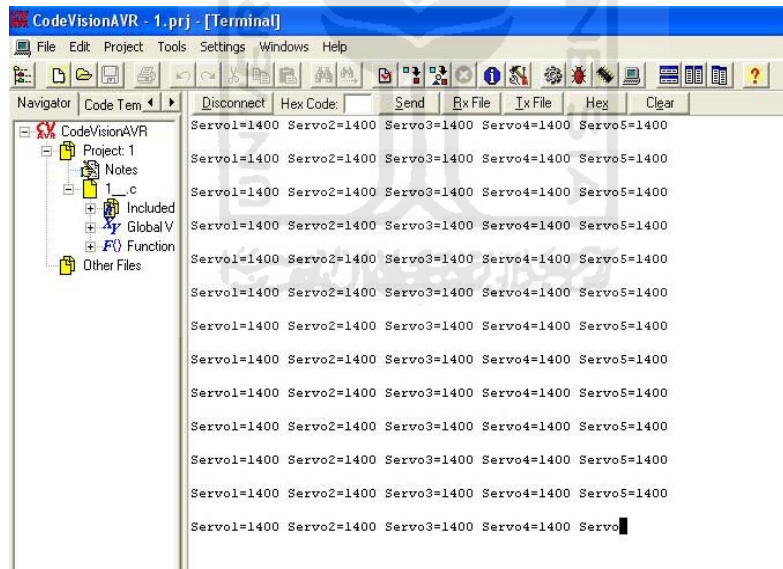
```
a=(0.1*pwm1)-60;
```

Rangkaian saklar untuk mengatur arus yang masuk pada rangkaian dan potensiometer data dapat dilihat pada LCD. Saklar ini terdiri dari tujuh pin yang terhubung dengan potensiometer, pin 1 jika ditekan maka sebagai penggerak manual potensiometer dan jika pin 1 dan pin 2 ditekan maka sebagai penggerak

otomatis potensiometer. Pin 1 dan 2 tidak ditampilkan pada LCD, hanya pin 3 sampai 7 yang ditampilkan. Seperti pada kode program berikut ini.

```
sprintf(data_LCD, "%i-%i-%i-",pwm1,pwm2,pwm3);  
  lcd_gotoxy(0,0);  
  lcd_puts(data_LCD);  
  
sprintf(data_LCD,"%i-%i-  
%%i%%i%%i%%i",pwm4,pwm5,PINB.3,PINB.4,PINB.5,PINB.6,PINB.7,);  
  lcd_gotoxy(0,1);  
  lcd_puts(data_LCD);
```

Pada Gambar 4.1 berikut ini merupakan nilai servo yang ditampilkan pada HyperTerminal.



Gambar 4.1 Nilai motor servo yang ditampilkan pada Hyperterminal.

Penjelasan dari kode program *if (PINB.1==0&&PINB.2==1)* adalah PINB.1 = 0 adalah hidup dan PINB.2 = 1 adalah mati, maka pin dua tidak

berfungsi atau tidak aktif dan yang berfungsi yang digunakan adalah pin satu) seperti pada kode program berikut ini:

```

if(PINB.1==0&&PINB.2==1)
{

if(PINB.3==0&&PINB.4==0&&PINB.5==0&&PINB.6==0&&PINB.7==0)//00000
{
    pwm2=pwm1;
    pwm3=pwm1;
    pwm4=pwm1;
    pwm5=pwm1;
    pwm1=(7.058*read_adc(3))+600;
    lcd_gotoxy(0,1);
    lcd_putsf("A");
};

```

Penjelasan dari kode program *If (PINB.1==1&&PINB.2==0)* adalah PINB.1 = 1 adalah mati dan PINB.2 = 0 adalah hidup, maka pin satu tidak berfungsi atau tidak aktif dan yang berfungsi digunakan adalah pin dua. Untuk nilai 600 motor servo akan bergerak kekanan posisi menggenggam dan pada nilai 2400 motor servo akan bergerak kekiri pada posisi membuka. Seperti pada kode berikut ini:

```

If(PINB.1==1&&PINB.2==0)
{
    pwm1=600; pwm2=2400; pwm3=2400; pwm4=2400; pwm5=2400; delay_ms(1000);
};

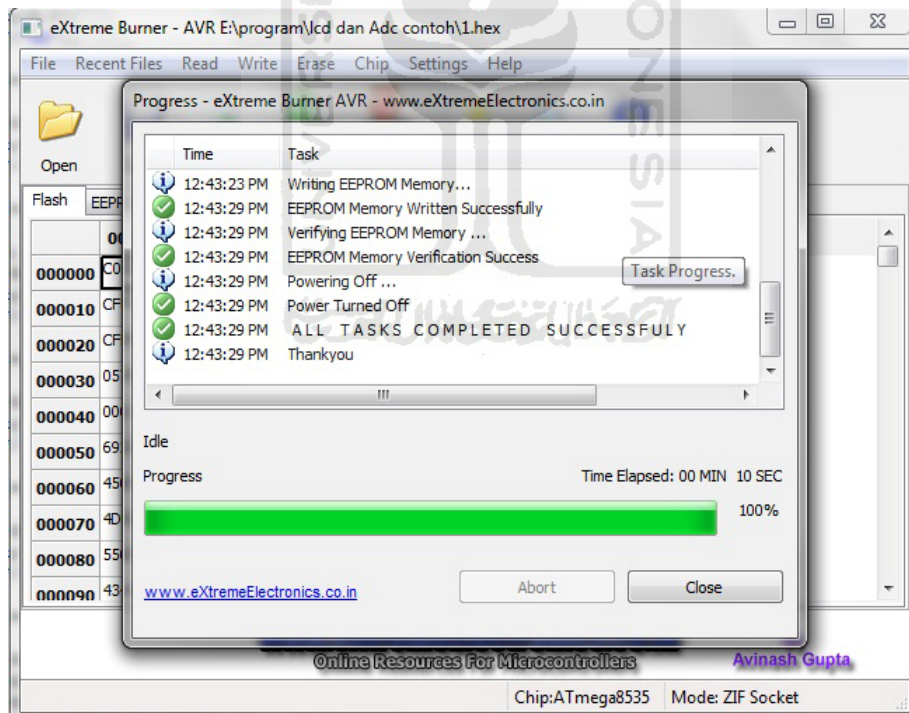
```

Pada kode program *If (PINB.1==0&&PINB.2==0)* adalah sebagai kontrol melalui serial port dan digerakan menggunakan keyboard, dimana pada PINB.1 dan PINB.2 adalah 0 maka kode program tidak dieksekusi atau tidak dikerjakan. Pada kode program *if (getchar()=='a'){pwm1=pwm1+100;};* nilai pwm1 ditambah 100 dan pada *if (getchar()=='b'){pwm1=pwm1-100;};* nilai pwm1

dikurangi 100, maka motor servo akan bergerak sesuai dengan perintah yang diberikan. Seperti pada kode program berikut ini.

```
if(PINB.1==0&&PINB.2==0)
{
if(getchar()=='a'){pwm1=pwm1+100;};
if(getchar()=='b'){pwm1=pwm1-100;};
};
```

Selanjutnya untuk memasukan kode program kedalam mikrokontroler menggunakan software extreme burner, masukan kode program dengan klik open kemudian reload dan write all (download), maka data program telah tersimpan pada mikrokontroler. Dapat ditunjukkan pada gambar 4.2 berikut ini.



Gambar 4.2 Software extreme burner.

4.3 Pengukuran Derajat Motor Servo

Pengukuran putaran derajat motor servo menggunakan alat seperti :

1. Pensil.
2. Busur derajat.
3. Penggaris.
4. Jarum.

4.4 Pengukuran Derajat Putaran Jari Tangan Robot

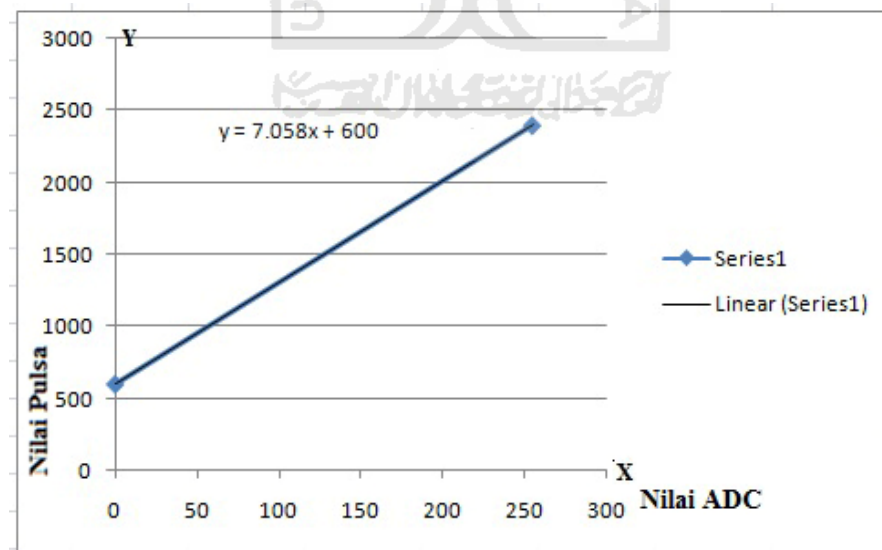
Mengkonversi nilai ADC yang dihasilkan oleh potensiometer yaitu 0-255 menjadi nilai pulsa (frekuensi penggerak servo) 600-2400, didapat hasil yaitu $y = 7.058 \times \text{ADC} + 600$, seperti data berikut ini.

Jika nilai ADC = 0

Servo = $7.058 \times 0 + 600 = 600$ (600 adalah nilai pulsa).

Jika nilai ADC = 255

Servo = $7.058 \times 255 + 600 = 2399,79$ (2399,79 adalah nilai pulsa). Nilai yang diperoleh dapat dilihat pada Gambar 4.3 berikut ini.

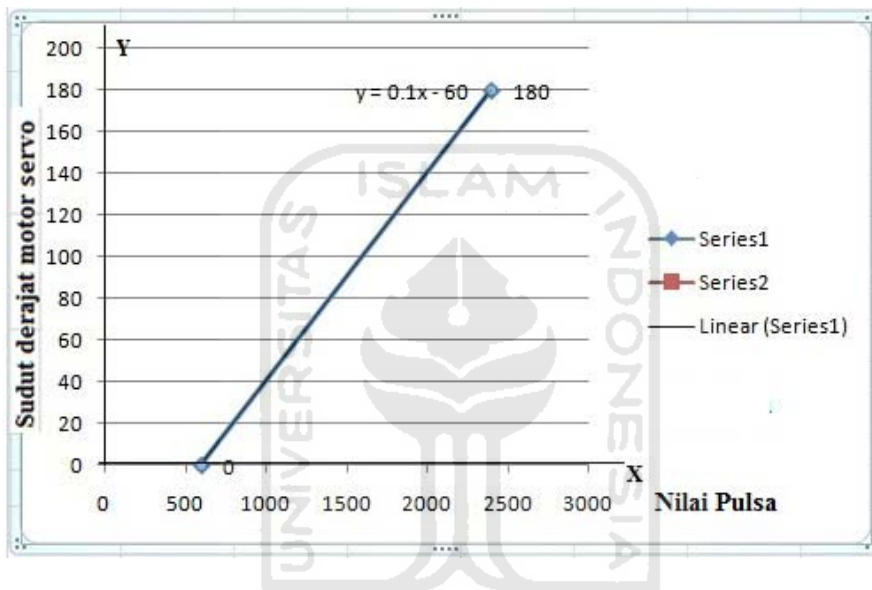


Gambar 4.3 Nilai ADC dan pulsa (motor servo).

Selanjutnya mencari nilai sudut derajat motor servo dan nilai pulsa penggerak motor servo, didapat hasil $y = 0.1 \times \text{servo}(\text{pulsa}) - 60$, seperti data berikut ini.

Sudut (y) = $0.1 \times 600 - 60 = 0$, maka didapat nilai sudut 0° .

Sudut (y) = $0.1 \times 2399,79 - 60 = 179,979^\circ$, maka didapat nilai sudut $179,979^\circ$. Nilai yang diperoleh dapat dilihat pada Gambar 4.4 berikut ini.



Gambar 4.4 Nilai pulsa (motor servo) dan nilai sudut.

Keterangan Gambar yaitu :

- 1) $y = 0.1 \times -60$ adalah nilai hasil perhitungan pulsa dan sudut derajat motor servo.
- 2) 0-180 adalah derajat motor servo.
- 3) 600-2400 adalah nilai pulsa motor servo.

4.4.1 Pengukuran pada Sudut 0° Motor Servo

Pada sudut 0° motor servo bergerak kearah kanan atau menggenggam, dikarenakan sudut awalnya adalah 0° pada posisi menggenggam. Pada LCD akan terlihat nilai 0° , kemudian motor servo akan bergerak sesuai sudut putaran yaitu

0°. Selanjutnya ukur *horn* dengan menggunakan busur derajat sebagai titik acuan adalah titik awal busur derajat yaitu 0°. Seperti pada Gambar 4.5 berikut ini.



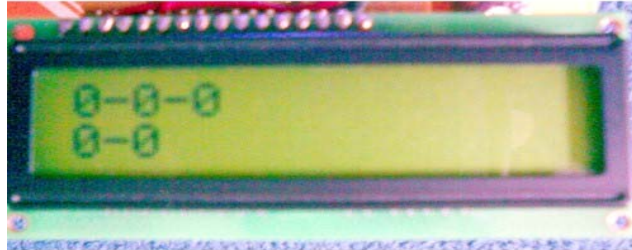
Gambar 4.5 Pengukuran posisi motor servo menggunakan busur derajat sudut 0°.

Posisi jari tangan robot pada sudut 0° adalah menggenggam. Dapat dilihat pada Gambar 4.6 berikut ini.



Gambar 4.6 Jari tangan robot pada posisi menggenggam pada sudut 0°.

Pada LCD akan terlihat nilai yang diinginkan, seperti pada Gambar 4.7 merupakan nilai sudut 0° .



Gambar 4.7 Nilai sudut 0° pada lcd.

Yang ditampilkan pada penggambaran sudut 30° , 60° , 90° , 180° hanya posisi jari tangan robot saja, karena penggambaran dengan menggunakan busur derajat dan nilai pada lcd sama seperti pada sudut 0° , mengikuti sudut derajat dan putaran potensiometer.

4.4.2 Pengukuran pada Sudut 30° Motor Servo

Penggambaran posisi susunan jari tangan robot pada sudut 30° motor servo dapat dilihat seperti pada Gambar 4.8 berikut ini.



Gambar 4.8 Posisi jari tangan pada sudut 30° .

4.4.3 Pengukuran pada Sudut 60° Motor Servo

Penggambaran posisi susunan jari tangan robot pada sudut 60° motor servo dapat dilihat seperti pada Gambar 4.9 berikut ini.



Pada Gambar 4.9 Posisi jari tangan pada sudut 60°.

4.4.4 Pengukuran pada Sudut 90° Motor Servo

Penggambaran posisi susunan jari tangan robot pada sudut 90° motor servo dapat dilihat seperti pada Gambar 4.10 berikut ini.



Pada Gambar 4.10 Posisi jari tangan pada sudut 90°.

4.4.5 Pengukuran pada Sudut 180° Motor Servo

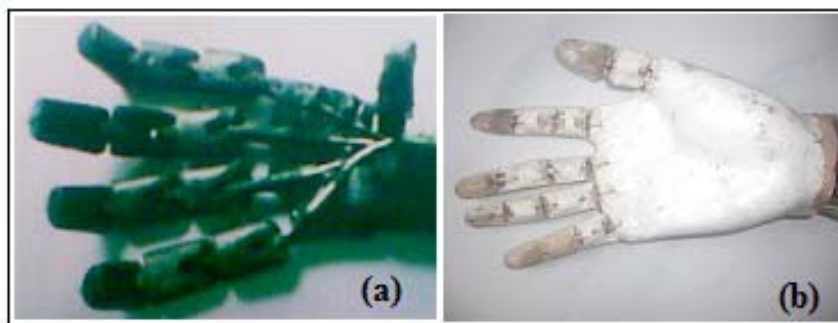
Penggambaran posisi susunan jari tangan robot pada sudut 180° motor servo dapat dilihat seperti pada Gambar 4.11 berikut ini.



Pada gambar 4.11 Posisi tangan pada sudut 180°.

4.5 Tangan Robot Terdahulu dan Tangan Robot Pengembangan

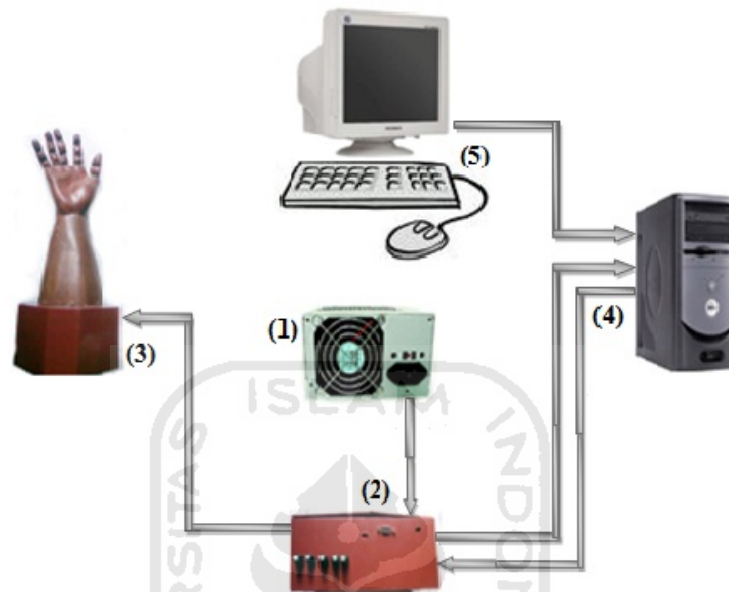
Tangan robot telah berhasil dibuat, tetapi fokus dari perancangan adalah untuk membuat sistem kendali untuk menggerakkan jari tangan robot. Tangan robot terdahulu (Gambar 4.12a) dan yang telah berhasil dibuat (Gambar 4.12b).



Gambar 4.12 (a) Tangan robot terdahulu (b) Tangan robot pengembangan.

4.6 Rangkaian Setup Penelitian

Rangkaian setup penelitian dapat dilihat pada Gambar 4.13 berikut ini.



Gambar 4.13 rangkaian setup penelitian

Keterangan dari Gambar 4.13 adalah :

1. Power *Supply* (catu daya).
2. Mikrokontroler.
3. Tangan Robot.
4. CPU.
5. Monitor dan Keyboard.

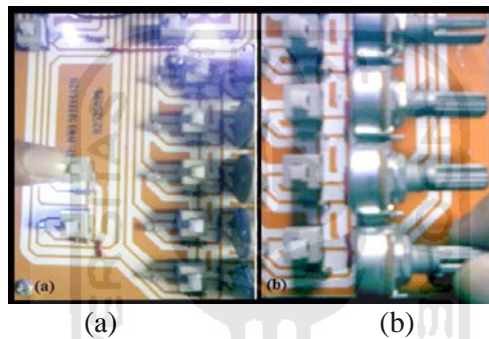
Langkah untuk menggerakkan tangan robot adalah :

1. Power supply (Catu daya) dihubungkan pada mikrokontroler.
2. Motor servo (Tangan Robot) dihubungkan pada mikrokontroler.
3. Serial RS232 dihubungkan pada mikrokontroler dan port CPU.
4. Mikrokontroler dihubungkan pada CPU.
5. Program (Code Vision AVR) dimasukkan pada mikrokontroler.
6. Tangan robot dapat digerakan menggunakan keyboard dan potensiometer.

4.7 Mode Input

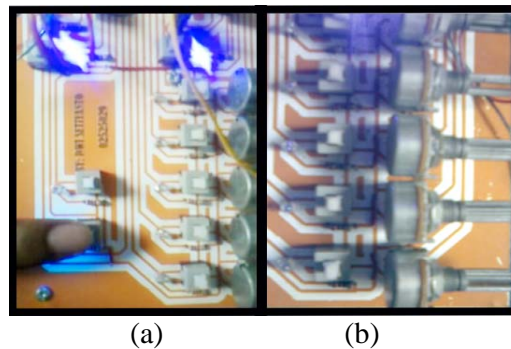
Untuk menggerakkan jari tangan robot dapat dilakukan dengan memutar potensiometer dan dapat juga digerakan dengan menggunakan keyboard pada komputer.

1. Menggunakan Potensiometer.
 - a. Dengan menekan tombol saklar pada Pin 1 kemudian potensiometer diputar maka jari tangan robot dapat bergerak. Seperti Gambar 4.14 berikut ini.



Gambar 4.14 (a) Pin1 ditekan (b) Potensiometer diputar.

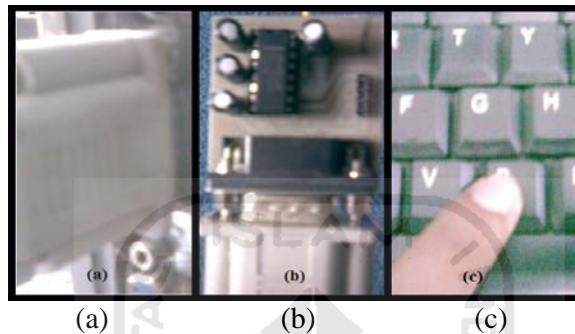
- b. Dengan menekan tombol saklar pada Pin 2 ditekan maka jari tangan robot dapat bergerak tanpa harus memutar potensiometer. Seperti pada gambar 4.15 berikut ini.



Gambar 4.15 (a) Pin1 dan pin 2 ditekan (b) Potensiometer.

2 Menggunakan Keyboard.

- a. Rangkaian mikrokontroler dihubungkan dengan rangkaian serial RS-232 kemudian dihubungkan pada *port* komputer. Jika keyboard ditekan maka jari tangan robot akan bergerak. Seperti pada Gambar 4.15 berikut ini.



Gambar 4.15 (a) Kabel RS-232 pada port CPU (b) Kabel RS-232 yang terhubung pada mikrokontroler (c) Keyboard komputer.

Hasil perancangan tangan robot pada penelitian ini memiliki kelebihan dan kekurangan dibandingkan perancangan tangan robot terdahulu. Pada penelitian ini terdapat kesulitan pada saat proses pembuatan sistem kendali untuk menggerakkan jari tangan robot. Kelebihan dan kekurangan penelitian ini antara lain:

Kelebihan :

- a. Jari tangan robot dapat bergerak menggenggam satu persatu dan juga dapat bergerak secara acak tidak beraturan.
- b. Bentuk jari tangan robot sudah lebih baik dari pembuatan sebelumnya.
- c. Tangan robot dapat digerakan menggunakan keyboard dengan menggunakan serial kabel RS-232 yang dihubungkan pada mikrokontroler dan komputer, pada pembuatan tangan robot sebelumnya belum dapat digerakan menggunakan keyboard.

Kekurangan :

- a. Jari tangan robot masih sedikit terlihat mekanik pada posisi sendi-sendinya karena menggunakan pegas sebagai pembalik pada saat jari

tangan membuka. Akan lebih baik apabila diberi sarung tangan sehingga terlihat lebih menyerupai jari tangan.

- b. Tangan robot tidak dapat memegang benda, karena jari tangan robot dirancang untuk menggenggam dan membuka.
- c. Pada saat jari tangan robot bergerak bersamaan sering terjadi eror, karena kekurangan sumber arus.



4.8 Biaya Pembuatan

Dalam pembuatan rangkaian sistem pengendali membutuhkan biaya, antara lain: biaya pembuatan rangkaian mikrokontroler, biaya pembuatan rangkaian Serial RS232, biaya pembelian motor servo (mikro servo TGY-90 S) sebagaimana table di bawah ini.

Table 4.3 Biaya pembuatan rangkaian mikrokontrler.

No	Bahan	Jumlah	Harga satuan (Rp)	Total (Rp)
1	PCB polos	3	3.000	9.000
2	Glue Stick kecil 30 cm	2	900	1.800
3	Pelarut	2	2.250	4.500
4	Tenol	1	8.500	8.500
5	Cover Komputer DB 25 Pin	1	2.750	2.750
6	Jeck Komputer DB 25P LK	1	2.750	2.750
7	Pin Deret 1X40 lurus	1	1.500	1.500
8	Pin Deret 2X40 lurus	1	2.500	2.500
9	Soket IDC 10 Pin	1	1.950	1.950
10	Black Housing 5 Pin	2	1.750	1.750
11	Soket IC 40 Pin	1	1.000	1.000
12	Elco 330/25	2	400	800
13	IC ATMega 8535	1	39.500	39.500
14	SK Tv 6 Pin (2X3) 25X	7	1.500	10.500
15	Black Housing 2Pin	6	500	3.000
16	Mikro Swith kecil 4 Pin	1	350	350
17	IC AN7805 kw	2	2.250	4.500
18	Kristal 8.000 Mhz	1	2.000	2.000
19	LED biru 3 mm	7	500	3.500
20	Mata bor 0,8 mm	1	800	800
21	Mata bor 1 mm	1	800	800
22	R1/4W	7	25	175
23	Dioda	2	200	400
24	Kabel 1X0,6 merah	1	600	600
Jumlah				106.675

Table 4.4 Biaya pembuatan rangkaian serial RS232.

No	Bahan	Jumlah	Harga satuan (Rp)	Total (Rp)
1	Jeck + kabel Komputer VGA 9P	1	19.500	19.500
2	Pin deret 1X40	1	1.750	1.750
3	Soket Komputer DB9P	1	6.500	6.500
4	Black Konektor 8 Pin	1	2.250	2.250
5	Kabel	2	300	600
6	Soket IC 16 Pin	1	400	400
7	IC MAX 232	1	8.500	8.500
8	Elco 10/16	4	150	600
Jumlah				40.100

Table 4.5 Biaya pembelian motor servo.

No	Bahan	Jumlah	Harga satuan (Rp)	Total (Rp)
1	Motor Servo	5	85.000	425.000
1	Motor Servo	1	85.000	85.000
Jumlah				510.000

Motor servo yang digunakan dalam penelitian ini sebanyak 5 (lima) buah. Terdapat kerusakan pada salah satu motor servo sehingga perlu mengganti dengan yang baru.

BAB V

PENUTUP

5.1 Kesimpulan

Dari penelitian yang telah dilakukan, dapat diambil kesimpulan sebagai berikut:

Sistem kendali jari tangan robot telah berhasil dibuat. Lima buah motor servo digunakan untuk menggerakkan jari-jari tangan robot. Jari tangan robot dapat bergerak menggenggam satu persatu serta dapat bergerak menggenggam dan membuka secara tidak bersamaan dengan memasukkan kode program ke dalam mikrokontroler. Rangkaian sistem pengendali untuk menggerakkan motor servo menggunakan potensiometer dan keyboard dengan menggunakan serial kabel pada *port* komputer.

Program untuk mengontrol jari tangan robot agar dapat bergerak dengan baik yaitu menggunakan software Code Vision AVR Bahasa C. Kode-kode program dimasukkan ke dalam mikrokontroler kemudian *download*, maka program telah tersimpan pada mikrokontroler.

5.2 Saran

Berdasarkan penelitian yang telah dilakukan, maka penulis dapat member saran bagi pengembangan penelitian lain mengenai robot, khususnya tangan robot. Pengembangan yang dapat dilakukan antara lain:

- a. Menambahkan sensor untuk menggerakkan jari tangan robot;
- b. Pembuatan tangan robot yang dapat memegang benda. Pada penelitian ini jari tangan robot difokuskan hanya untuk bergerak menggenggam dan membuka serta bergerak secara tidak bersamaan.

DAFTAR PUSTAKA

- Agung, N.A. (2009). *Buku Ajar Kuliah Mekanika*, Prodi Teknik Mesin Fakultas Teknologi Industri Universitas Islam Indonesia Yogyakarta.
- Andre, D.F. (2008). *Layer Deposition Manufacturing*. Skripsi. Tidak diterbitkan. Teknik Mesin Universitas Islam Indonesia Yogyakarta.
- Endra, P. (2006). *Robotika Desain, dan Kecerdasan Buatan*. Andi : Yogyakarta
- Erlangga, R. (2008). *Pembuatan Mekanisme Menggenggam Pada Tangan Robot Humanoid*. Skripsi. Tidak diterbitkan. Teknik Mesin Universitas Islam Indonesia Yogyakarta.
- Lingga, W. (2006). *Belajar Sendiri Mikrokontroler AVR Seri ATmega8535, Simulasi, Hardware, dan Aplikasi*. Andi : Yogyakarta.
- Petunjuk Praktikum Mekanika, (2009). *Tim Laboratorium Mekanika, Jurusan Teknik Mesin*, Universitas Islam Indonesia.
- Riyanto, S. (2007). *Robotika, Sensor, dan Aktuator*. Graha Ilmu : Yogyakarta.
- Wirawan, S. (2008) *Teknik Produksi Mesin Industri Jilid III : untuk SMK*. Jakarta : Direktorat Pembinaan Sekolah Menengah Kejuruan, Departemen Pendidikan Nasional.



LAMPIRAN



This program was produced by the
CodeWizardAVR V1.24.6 Professional
Automatic Program Generator
© Copyright 1998-2005 Pavel Haiduc, HP InfoTech s.r.l.
<http://www.hpinfotech.com>
e-mail:office@hpinfotech.com

Project :
Version :
Date : 2/8/2011
Author : F4CG
Company : F4CG
Comments:

Chip type : ATmega8535
Program type : Application
Clock frequency : 8.000000 MHz
Memory model : Small
External SRAM size : 0
Data Stack size : 128

*****/

```
#include <mega8535.h>
#include <delay.h>
char data_LCD [16];
int z,a,b,c,d,e;
char i;
// Alphanumeric LCD Module functions
#asm
.equ __lcd_port=0x15 ;PORTC
#endasm
#include <lcd.h>
#include <stdio.h>
//
#define servo1_pin PORTD.7
#define servo2_pin PORTD.6
#define servo3_pin PORTD.5
#define servo4_pin PORTD.4
#define servo5_pin PORTD.3
#define servo6_pin PORTD.2
unsigned int pwm1=1400;
unsigned int pwm2=1400;
unsigned int pwm3=1400;
unsigned int pwm4=1400;
unsigned int pwm5=1400;
```

```

unsigned int pwm6=1400;
unsigned char servo_index=0;
// Timer 1 output compare A interrupt service routine
interrupt [TIM1_COMPA] void timer1_compa_isr(void)
{
  if(servo_index==0)
  {
    servo1_pin=1;
    OCR1A=pwm1;
    servo_index=1;
  }
  else if(servo_index==1)
  {
    servo1_pin=0;
    servo2_pin=1;
    OCR1A=pwm2;
    servo_index=2;
  }
  else if(servo_index==2)
  {
    servo2_pin=0;
    servo3_pin=1;
    OCR1A=pwm3;
    servo_index=3;
  }
  else if(servo_index==3)
  {
    servo3_pin=0;
    servo4_pin=1;
    OCR1A=pwm4;
    servo_index=4;
  }
  else if(servo_index==4)
  {
    servo4_pin=0;
    servo5_pin=1;
    OCR1A=pwm5;
    servo_index=5;
  }
  else if(servo_index==5)
  {
    servo5_pin=0;
    servo6_pin=1;
    OCR1A=pwm6;
    servo_index=6;
  }
}

```



```

else if(servo_index==6)
{
servo6_pin=0;
OCR1A=2000;
servo_index=0;
}
}
/////
#define ADC_VREF_TYPE 0x60

// Read the 8 most significant bits
// of the AD conversion result
unsigned char read_adc(unsigned char adc_input)
{
ADMUX=adc_input|ADC_VREF_TYPE;
// Start the AD conversion
ADCSRA|=0x40;
// Wait for the AD conversion to complete
while ((ADCSRA & 0x10)==0);
ADCSRA|=0x10;
return ADCH;
}

// Declare your global variables here

void main(void)
{
// Declare your local variables here

// Input/Output Ports initialization
// Port A initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In
Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
PORTA=0x00;
DDRA=0x00;

// Port B initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In
Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
PORTB=0xFF;
DDRB=0x00;

// Port C initialization

```

```

// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In
Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
PORTC=0x00;
DDRC=0x00;

// Port D initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In
Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
PORTD=0x00;
DDRD=0xFF;

// Timer/Counter 0 initialization
// Clock source: System Clock
// Clock value: Timer 0 Stopped
// Mode: Normal top=FFh
// OC0 output: Disconnected
TCCR0=0x00;
TCNT0=0x00;
OCR0=0x00;
// Timer/Counter 1 initialization
// Clock source: System Clock
// Clock value: 1000.000 kHz
// Mode: CTC top=OCR1A
// OC1A output: Discon.
// OC1B output: Discon.
// Noise Canceler: Off
// Input Capture on Falling Edge
// Timer 1 Overflow Interrupt: Off
// Input Capture Interrupt: Off
// Compare A Match Interrupt: On
// Compare B Match Interrupt: Off
TCCR1A=0x00;
TCCR1B=0x0A;
OCR1A=5000;

// Timer/Counter 2 initialization
// Clock source: System Clock
// Clock value: Timer 2 Stopped
// Mode: Normal top=FFh
// OC2 output: Disconnected
ASSR=0x00;
TCCR2=0x00;
TCNT2=0x00;
OCR2=0x00;

```

```

// External Interrupt(s) initialization
// INT0: Off
// INT1: Off
// INT2: Off
MCUCR=0x00;
MCUCSR=0x00;

// Timer(s)/Counter(s) Interrupt(s) initialization
TIMSK=0x10;

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

// ADC initialization
// ADC Clock frequency: 125.000 kHz
// ADC Voltage Reference: AVCC pin
// ADC High Speed Mode: Off
// ADC Auto Trigger Source: None
// Only the 8 most significant bits of
// the AD conversion result are used
ADMUX=ADC_VREF_TYPE;
ADCSRA=0x86;
SFIOR&=0xEF;

/// 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=0x33;
// Global enable interrupts
#asm("sei")

// LCD module initialization
lcd_init(16);

while (1)
{
    // Place your code here

```

```

// sprintf(data_LCD, "%i-%i-%i",a,b,c);
// lcd_gotoxy(0,0);
// lcd_puts(data_LCD);
// sprintf(data_LCD, "%i-%i",d,e);
// lcd_gotoxy(0,1);
// lcd_puts(data_LCD);
// pwm1=(7.058*read_adc(7))+600;
// pwm2=(7.058*read_adc(6))+600;
// pwm3=(7.058*read_adc(5))+600;
// pwm4=(7.058*read_adc(4))+600;
// pwm5=(7.058*read_adc(3))+600;
// a=(0.1*pwm1)-60;
// b=(0.1*pwm2)-60;
// c=(0.1*pwm3)-60;
// d=(0.1*pwm4)-60;
// e=(0.1*pwm5)-60;

//SAKLAR
sprintf(data_LCD, "%i-%i-%i-",pwm1,pwm2,pwm3);
lcd_gotoxy(0,0);
lcd_puts(data_LCD);

sprintf(data_LCD,"%i-%i-
%i%i%i%i%i",pwm4,pwm5,PINB.3,PINB.4,PINB.5,PINB.6,PINB.7.);
lcd_gotoxy(0,1);
lcd_puts(data_LCD);

printf("Servo1=%i° Servo2=%i° Servo3=%i° Servo4=%i° Servo5=%i°
\r\n",pwm1,pwm2,pwm3,pwm4,pwm5); //Data yang ditampilkan pada
HyperTerminal

if(PINB.1==0&&PINB.2==1)
{

if(PINB.3==0&&PINB.4==0&&PINB.5==0&&PINB.6==0&&PINB.7==0)//00
00 (0)
{
pwm2=pwm1;
pwm3=pwm1;
pwm4=pwm1;
pwm5=pwm1;
pwm1=(7.058*read_adc(3))+600;
lcd_gotoxy(0,1);
lcd_putsf("A");
};

```

```

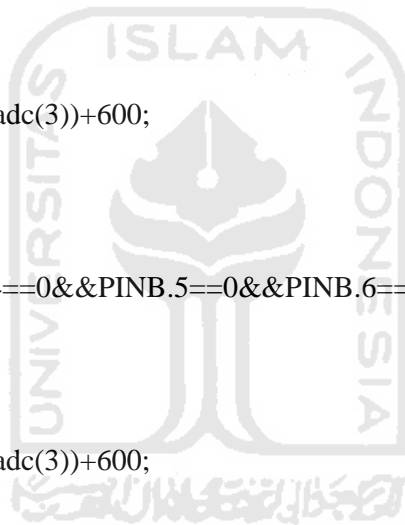
if(PINB.3==0&&PINB.4==0&&PINB.5==0&&PINB.6==0&&PINB.7==1)//000
01 (1)
{
    pwm2=pwm1;
    pwm3=pwm1;
    pwm4=pwm1;
    pwm1=(7.058*read_adc(3))+600;
    lcd_gotoxy(0,1);
    lcd_putsf("B");
};

if(PINB.3==0&&PINB.4==0&&PINB.5==0&&PINB.6==1&&PINB.7==0)//000
10 (2)
{
    pwm2=pwm1;
    pwm3=pwm1;
    pwm5=pwm1;
    pwm1=(7.058*read_adc(3))+600;
    lcd_gotoxy(0,1);
    lcd_putsf("C");
};

if(PINB.3==0&&PINB.4==0&&PINB.5==0&&PINB.6==1&&PINB.7==1)//000
11 (3)
{
    pwm2=pwm1;
    pwm3=pwm1;
    pwm1=(7.058*read_adc(3))+600;
    lcd_gotoxy(0,1);
    lcd_putsf("D");
};

if(PINB.3==0&&PINB.4==0&&PINB.5==1&&PINB.6==0&&PINB.7==0)//001
00 (4)
{
    pwm2=pwm1;
    pwm4=pwm1;
    pwm5=pwm1;
    pwm1=(7.058*read_adc(3))+600;
    lcd_gotoxy(0,1);
    lcd_putsf("E");
};

```



```

if(PINB.3==0&&PINB.4==0&&PINB.5==1&&PINB.6==0&&PINB.7==1)//001
01 (5)
{
    pwm2=pwm1;
    pwm4=pwm1;
    pwm1=(7.058*read_adc(3))+600;
    lcd_gotoxy(0,1);
    lcd_putsf("F");
};

```

```

if(PINB.3==0&&PINB.4==0&&PINB.5==1&&PINB.6==1&&PINB.7==0)//001
10 (6)
{
    pwm2=pwm1;
    pwm5=pwm1;
    pwm1=(7.058*read_adc(3))+600;
    lcd_gotoxy(0,1);
    lcd_putsf("G");
};

```

```

if(PINB.3==0&&PINB.4==0&&PINB.5==1&&PINB.6==1&&PINB.7==1)//001
11 (7)
{
    pwm2=pwm1;
    pwm1=(7.058*read_adc(3))+600;
    lcd_gotoxy(0,1);
    lcd_putsf("H");
};

```

```

if(PINB.3==0&&PINB.4==1&&PINB.5==0&&PINB.6==0&&PINB.7==0)//010
00 (8)
{
    pwm3=pwm1;
    pwm4=pwm1;
    pwm5=pwm1;
    pwm1=(7.058*read_adc(3))+600;
    lcd_gotoxy(0,1);
    lcd_putsf("I");
};

```

```

if(PINB.3==0&&PINB.4==1&&PINB.5==0&&PINB.6==0&&PINB.7==1)//010
01 (9)
{
    pwm3=pwm1;
    pwm4=pwm1;
    pwm1=(7.058*read_adc(3))+600;
};

```



```

    lcd_gotoxy(0,1);
    lcd_putsf("J");
};

if(PINB.3==0&&PINB.4==1&&PINB.5==0&&PINB.6==1&&PINB.7==0)//010
10 (10)
{
    pwm3=pwm1;
    pwm5=pwm1;
    pwm1=(7.058*read_adc(3))+600;
    lcd_gotoxy(0,1);
    lcd_putsf("K");
};

if(PINB.3==0&&PINB.4==1&&PINB.5==0&&PINB.6==1&&PINB.7==1)//010
11 (11)
{
    pwm3=pwm1;
    pwm1=(7.058*read_adc(3))+600;
    lcd_gotoxy(0,1);
    lcd_putsf("L");
};

if(PINB.3==0&&PINB.4==1&&PINB.5==1&&PINB.6==0&&PINB.7==0)//011
00 (12)
{
    pwm4=pwm1;
    pwm5=pwm1;
    pwm1=(7.058*read_adc(3))+600;
    lcd_gotoxy(0,1);
    lcd_putsf("M");
};

if(PINB.3==0&&PINB.4==1&&PINB.5==1&&PINB.6==0&&PINB.7==1)//011
01 (13)
{
    pwm4=pwm1;
    pwm1=(7.058*read_adc(3))+600;
    lcd_gotoxy(0,1);
    lcd_putsf("N");
};

if(PINB.3==0&&PINB.4==1&&PINB.5==1&&PINB.6==1&&PINB.7==0)//011
10 (14)
{
    pwm5=pwm1;

```

```

    pwm1=(7.058*read_adc(3))+600;
    lcd_gotoxy(0,1);
    lcd_putsf("O");
};

if(PINB.3==0&&PINB.4==1&&PINB.5==1&&PINB.6==1&&PINB.7==1)//011
11 (15)
{
    pwm1=(7.058*read_adc(3))+600;
    lcd_gotoxy(0,1);
    lcd_putsf("P");
};

if(PINB.3==1&&PINB.4==0&&PINB.5==0&&PINB.6==0&&PINB.7==0)//100
00 (16)
{
    pwm3=pwm2;
    pwm4=pwm2;
    pwm5=pwm2;
    pwm2=(7.058*read_adc(4))+600;
    lcd_gotoxy(0,1);
    lcd_putsf("Q");
};

if(PINB.3==1&&PINB.4==0&&PINB.5==0&&PINB.6==0&&PINB.7==1)//100
01 (17)
{
    pwm3=pwm2;
    pwm4=pwm2;
    pwm2=(7.058*read_adc(4))+600;
    lcd_gotoxy(0,1);
    lcd_putsf("R");
};

if(PINB.3==1&&PINB.4==0&&PINB.5==0&&PINB.6==1&&PINB.7==0)//100
10 (18)
{
    pwm3=pwm2;
    pwm5=pwm2;
    pwm2=(7.058*read_adc(4))+600;
    lcd_gotoxy(0,1);
    lcd_putsf("S");
};

```

```

if(PINB.3==1&&PINB.4==0&&PINB.5==0&&PINB.6==1&&PINB.7==1)//100
11 (19)
{
    pwm3=pwm2;
    pwm2=(7.058*read_adc(4))+600;
    lcd_gotoxy(0,1);
    lcd_putsf("T");
};

```

```

if(PINB.3==1&&PINB.4==0&&PINB.5==1&&PINB.6==0&&PINB.7==0)//101
00 (20)
{
    pwm4=pwm2;
    pwm5=pwm2;
    pwm2=(7.058*read_adc(4))+600;
    lcd_gotoxy(0,1);
    lcd_putsf("U");
};

```

```

if(PINB.3==1&&PINB.4==0&&PINB.5==1&&PINB.6==0&&PINB.7==1)//101
01 (21)
{
    pwm4=pwm2;
    pwm2=(7.058*read_adc(4))+600;
    lcd_gotoxy(0,1);
    lcd_putsf("V");
};

```

```

if(PINB.3==1&&PINB.4==0&&PINB.5==1&&PINB.6==1&&PINB.7==0)//101
10 (22)
{
    pwm5=pwm2;
    pwm2=(7.058*read_adc(4))+600;
    lcd_gotoxy(0,1);
    lcd_putsf("W");
};

```

```

if(PINB.3==1&&PINB.4==0&&PINB.5==1&&PINB.6==1&&PINB.7==1)//101
11 (23)
{
    pwm2=(7.058*read_adc(4))+600;
    lcd_gotoxy(0,1);
    lcd_putsf("X");
};

```

```

if(PINB.3==1&&PINB.4==1&&PINB.5==0&&PINB.6==0&&PINB.7==0)//110
00 (24)
{
    pwm4=pwm3;
    pwm5=pwm3;
    pwm3=(7.058*read_adc(5))+600;
    lcd_gotoxy(0,1);
    lcd_putsf("Y");
};

```

```

if(PINB.3==1&&PINB.4==1&&PINB.5==0&&PINB.6==0&&PINB.7==1)//110
01 (25)
{
    pwm4=pwm3;
    pwm3=(7.058*read_adc(5))+600;
    lcd_gotoxy(0,1);
    lcd_putsf("Z");
};

```

```

if(PINB.3==1&&PINB.4==1&&PINB.5==0&&PINB.6==1&&PINB.7==0)//110
10 (26)
{
    pwm5=pwm3;
    pwm3=(7.058*read_adc(5))+600;
    lcd_gotoxy(0,1);
    lcd_putsf("AB");
};

```

```

if(PINB.3==1&&PINB.4==1&&PINB.5==0&&PINB.6==1&&PINB.7==1)//110
11 (27)
{
    pwm3=(7.058*read_adc(5))+600;
    lcd_gotoxy(0,1);
    lcd_putsf("AC");
};

```

```

if(PINB.3==1&&PINB.4==1&&PINB.5==1&&PINB.6==0&&PINB.7==0)//111
00 (28)
{
    pwm5=pwm4;
    pwm4=(7.058*read_adc(6))+600;
    lcd_gotoxy(0,1);
    lcd_putsf("AD");
};

```

```

if(PINB.3==1&&PINB.4==1&&PINB.5==1&&PINB.6==0&&PINB.7==1)//111
01 (29)
{
    pwm4=(7.058*read_adc(6))+600;
    lcd_gotoxy(0,1);
    lcd_putsf("AE");
};

```

```

if(PINB.3==1&&PINB.4==1&&PINB.5==1&&PINB.6==1&&PINB.7==0)//111
10 (30)
{
    pwm5=(7.058*read_adc(7))+600;
    lcd_gotoxy(0,1);
    lcd_putsf("AF");
};

```

```

if(PINB.3==1&&PINB.4==1&&PINB.5==1&&PINB.6==1&&PINB.7==1)//111
11 (31)
{
    pwm1=(7.058*read_adc(3))+600;
    pwm2=(7.058*read_adc(4))+600;
    pwm3=(7.058*read_adc(5))+600;
    pwm4=(7.058*read_adc(6))+600;
    pwm5=(7.058*read_adc(7))+600;
    lcd_gotoxy(0,1);
    lcd_putsf("AG");
};
};

```

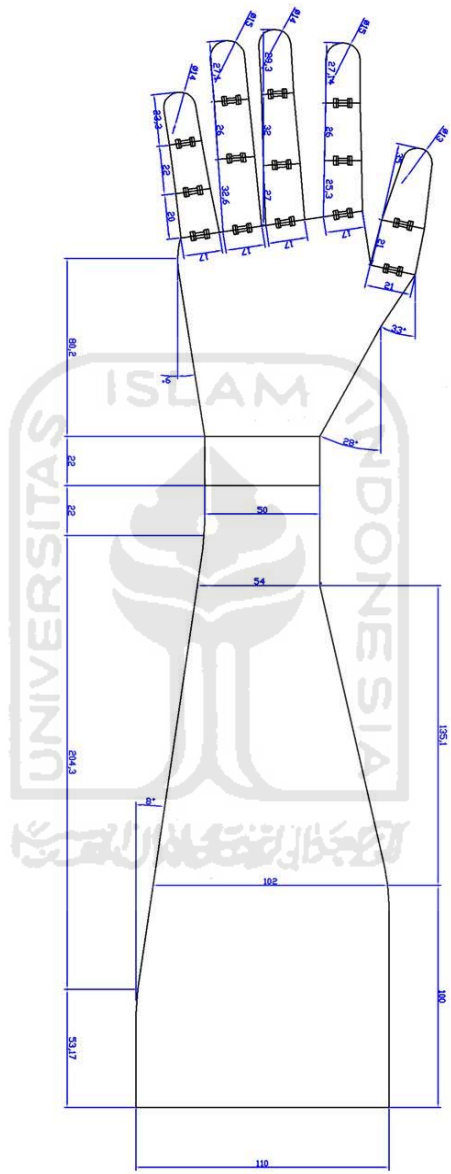
```

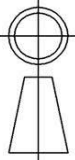
if(PINB.1==1&&PINB.2==0)
{
    pwm1=600;          pwm2=2400;          pwm3=2400;          pwm4=2400;
    pwm5=2400;delay_ms(1000);
    pwm1=600;          pwm2=600;          pwm3=2400;          pwm4=2400;
    pwm5=2400;delay_ms(1000);
    pwm1=600;          pwm2=600;          pwm3=600;          pwm4=2400;
    pwm5=2400;delay_ms(1000);
    pwm1=600; pwm2=600; pwm3=600; pwm4=600; pwm5=2400;delay_ms(1000);
    pwm1=600; pwm2=600; pwm3=600; pwm4=600; pwm5=600;delay_ms(1000);
    pwm1=2400;          pwm2=2400;          pwm3=2400;          pwm4=2400;
    pwm5=2400;delay_ms(1000);
};
if(PINB.1==0&&PINB.2==0)
{
    if (getchar()=='a'){ pwm1=pwm1+100;};
    if (getchar()=='b'){ pwm1=pwm1-100;};
};

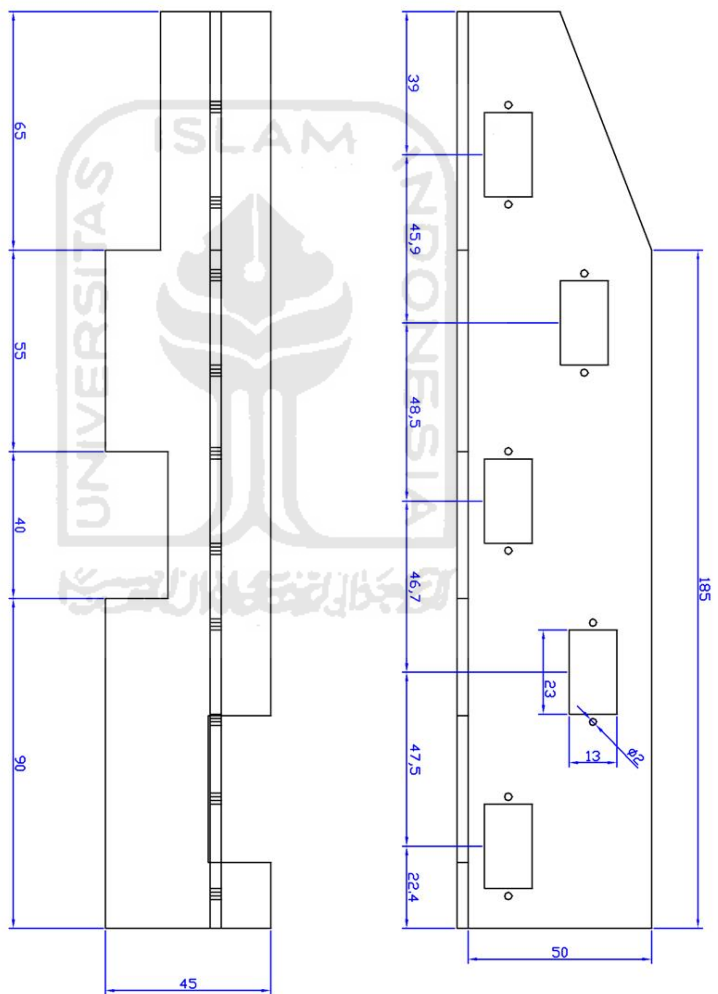
```

```
if (getchar()=='c'){pwm2=pwm2+100;};  
if (getchar()=='d'){pwm2=pwm2-100;};  
if (getchar()=='e'){pwm3=pwm3+100;};  
if (getchar()=='f'){pwm3=pwm3-100;};  
if (getchar()=='g'){pwm4=pwm4+100;};  
if (getchar()=='h'){pwm4=pwm4-100;};  
if (getchar()=='i'){pwm5=pwm5+100;};  
if (getchar()=='j'){pwm5=pwm5-100;};  
};  
};  
}
```





	SKALA :	mm	CATATAN :
	SATUAN :		
	NAMA :		
	DILIHAT :		
Mekanik Tangan			A ₄



	SKALA :	mm	CATATAN :
	SATUAN :		
	TANGGAL :		
U11	NAMA :		A4
	DILIHAT :		
Motor Housing			

SELESAI

