UNIVERSITAS INDONESIA

KENDALI LENGAN ROBOT LIMA DERAJAT KEBEBASAN MITSUBISHI MOVEMASTER RV-M1

SKRIPSI

ADRIAN BASKORO
0706198972

FAKULTAS TEKNIK
PROGRAM SARJANA EKSTENSI
DEPOK
JUNI 2009

UNIVERSITAS INDONESIA

KENDALI LENGAN ROBOT LIMA DERAJAT KEBEBASAN MITSUBISHI MOVEMASTER RV-M1

SKRIPSI

Diajukan sebagai salah satu syarat untuk memperoleh gelar sarjana teknik

ADRIAN BASKORO
0706198972

FAKULTAS TEKNIK
PROGRAM SARJANA EKSTENSI
DEPOK
JUNI 2009

PERNYATAAN ORISINALITAS

Skripsi ini adalah hasil karya saya sendiri,
dan semua sumber baik yang dikutip maupun dirujuk
telah saya nyatakan dengan benar.

Nama : Adrian Baskoro
NPM : 0706198972

Tanda Tangan :
Tanggal : 17 Juni 2009

PENGESAHAN

Skripsi ini diajukan oleh :
Nama : Adrian Baskoro
NPM : 0706198972
Program Studi : Teknik Elektro
Judul Skripsi : Kendali Lengan Robot Lima Derajat Kebebasan Mitsubishi Movemaster RV-M1

Telah berhasil dipertahankan di hadapan Dewan Penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Sarjana Teknik pada Program Studi Teknik Elektro, Fakultas Teknik, Universitas Indonesia

DEWAN PENGUJI

Pembimbing : Dr. Abdul Muis, ST, M.Eng ( )

Penguji : Ir. Wahidin Wahab, MSc, Ph.D ( )

Penguji : Dr. Ir. Ridwan Gunawan, MT. ( )

Ditetapkan di : Ruang MULTIMEDIA B LT.2 DTE DEPOK.

Hari / Tanggal : Rabu / 01 Juli 2009

UCAPAN TERIMA KASIH

Puji syukur saya panjatkan kepada **Tuhan Yang Maha Esa**, karena atas berkat dan rahmat-Nya, saya dapat menyelesaikan skripsi ini. Penulisan skripsi ini dilakukan untuk memenuhi salah satu syarat untuk mencapai gelar Sarjana Teknik Jurusan Elektro pada Fakultas Teknik Universitas Indonesia. Saya menyadari bahwa, tanpa bantuan dan bimbingan dari berbagai pihak, mulai dari masa perkuliahan sampai pada penyusunan skripsi ini, sangatlah sulit bagi saya untuk menyelesaikan skripsi ini. Oleh karena itu, saya mengucapkan terima kasih kepada:

(1) Dr. Abdul Muis, ST, M.Eng, selaku dosen pembimbing yang telah menyediakan waktu, tenaga dan pikiran untuk mengarahkan saya dalam penyusunan skripsi ini;
(2) Orang tua dan keluarga yang telah memberikan bantuan dukungan moral dan material; dan
(3) sahabat yang telah banyak membantu saya dalam menyelesaikan skripsi ini.

Akhir kata, saya berharap Tuhan Yang Maha Esa berkenan membalas segala kebaikan semua pihak yang telah membantu. Semoga skripsi ini membawa manfaat bagi pengembangan ilmu pengetahuan.

Depok, 17 Juni 2009

Penulis

PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS

Sebagai sivitas akademik Universitas Indonesia, saya yang bertanda tangan di bawah ini:

Nama : Adrian Baskoro
NPM : 0706198972
Program Studi : Teknik Elektro
Departemen : Teknik Elektro
Fakultas : Teknik
Jenis karya : Skripsi

demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Indonesia **Hak Bebas Royalti Noneksklusif (*Non-exclusive Royalty-Free Right*)** atas karya ilmiah saya yang berjudul :

Kendali Lengan Robot Lima Derajat Kebebasan Mitsubishi Movemaster RV-M1

beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Noneksklusif ini Universitas Indonesia berhak menyimpan, mengalihmedia/formatkan, mengelola dalam bentuk pangkalan data (*database*), merawat, dan mempublikasikan tugas akhir saya tanpa meminta izin dari saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Depok
Pada tanggal : 17 Juni 2009
Yang menyatakan

(Adrian Baskoro)

ABSTRAK

Nama : Adrian Baskoro
Program Studi : Teknik Elektro
Judul : Kendali Lengan Robot Lima Derajat Kebebasan Mitsubishi Movemaster RV-M1

Mitsubishi Movemaster RV-M1 merupakan salah satu mikro-robot industri dengan lima derajat kebebasan, ditambah sebuah manipulator akhir berupa *gripper (optional)*. Robot lengan produksi Mitsubishi ini digerakkan oleh motor *dc servo* pada setiap *joint*-nya. Posisi pergerakan setiap *joint* dapat diketahui dengan menggunakan pembacaan *rotary encoder* tipe *incremental*.

Skripsi ini mengimplementasikan perancangan kendali pergerakan lengan robot *Mitsubishi Movemaster RV-M1* yang dimiliki oleh Laboratorium Kendali Departemen Teknik Elektro Universitas Indonesia. Sistem kendali pergerakan lengan robot dirancang dan dibuat menyerupai fungsi unit penggerak robot *(drive unit)* dengan memanfaatkan komponen-komponen konstruksi robot yang tersedia.

Kendali pergerakan lengan robot terdiri dari beberapa rangkaian *microcontroller* berbasis *AT89S52* sesuai jumlah *joint* robot. Sebuah *microcontroller* jenis yang sama ditambahkan sebagai pusat pengendali. *Microcontroller* menggunakan *i²c bus* sebagai media komunikasi. Keseluruhan *joint* robot dapat digerakkan secara bersamaan.

Metode pemrograman *leadthrough* diaplikasikan pada kendali robot ini dimana manipulator digerakkan atau dipindahkan terlebih dahulu secara manual melalui lintasan pergerakan tertentu. Metode ini dikenal sebagai metode *'teaching by showing'*.

Kata kunci : *Mitsubishi Movemaster RV-M1*, Lengan Robot, Derajat Kebebasan.

ABSTRACT

Name : Adrian Baskoro
Study Program : Electrical Engineering
Title : Five Axis Arm Robot Control of Mitsubishi Movemaster RV-M1

A Mitsubishi Movemaster RV-M1 is one of the industrial micro-robots produced by Mitsubishi Corp. It has five degrees of freedom, not include hand gripper as its end-effector (optional). The movemaster RV-M1 is driven by dc servo motors with a toothed-belt transmission system. The current position of each joint can be determined using incremental rotary encoder coupled to each motors.

This Final Project applies a control system and design of the movemaster RV-M1 arm robot. The robot is a property of Electrical Department's Control Lab., University of Indonesia. The arm robot control system is designed to have similar function with its original drive unit utilising available components of the robot.

The arm robot control consist of some microcontrollers system based on AT89S52 (the same amount with joint number). Another one microcontroller system (same type) is added as its main control system. The microcontrollers using i²c bus as a communication media. All five joint can be moved simultaneously.

A leadthrough programming method is applied. The manipulator is manually moved beforehand to get a particular track path. This method is commonly known as 'teaching by showing' method.

Keywords: Mitsubishi Movemaster RV-M1, Arm Robot, Degree of Freedom.

DAFTAR ISI

Halaman

DAFTAR GAMBAR

Halaman

DAFTAR TABEL

Halaman

BAB I
PENDAHULUAN

1.1 LATAR BELAKANG MASALAH

Mitsubishi Movemaster RV-M1 micro-robot merupakan robot lengan dengan lima derajat kebebasan, ditambah dengan manipulator akhir *(end effector)* berupa sebuah *gripper*. Robot lengan ini digerakkan menggunakan beberapa motor dc servo. Untuk memperoleh torsi, konstruksi mekaniknya menggunakan beberapa roda gigi pembagi *(reduction gear)* dan sabuk gilir *(synchronous belt)* sebagai elemen transmisinya. Keseluruhan *joint* robot dapat digerakkan secara simultan dan kendali posisinya memanfaatkan pembacaan *rotary encoder* tipe *incremental* yang terhubung pada poros motor penggerak setiap *joint*.

Salah satu *industrial micro-robot* model *RV-M1* produksi *Mitsubishi* ini dimiliki oleh Departemen Teknik Elektro Universitas Indonesia dan menjadi salah satu fasilitas laboratorium kendali. Robot ini memiliki *teaching box* dan *drive unit*, namun kondisi *drive unit*-nya tidak berfungsi lagi. Oleh karena itu sebuah rangkaian sistem kendali pengganti dirancang dan dibuat menyerupai fungsi *drive unit* dengan memanfaatkan komponen-komponen konstruksi robot yang masih tersedia.

Perancangan sistem kendali lengan robot yang dibuat terdiri dari beberapa rangkaian *microcontroller* berbasis *AT89S52* produksi *Atmel* sesuai jumlah *joint* robot dan motor penggeraknya, ditambah dengan sebuah *microcontroller* jenis yang sama sebagai pusat pengendali. *Microcontroller* menggunakan *i²c bus (inter integrated circuit)* sebagai media komunikasi, dimana sebuah *serial EEPROM AT24C256* digunakan untuk menyimpan beberapa posisi per *step* pergerakan lengan robot. Data lintasan pergerakan dikirim secara serial ke komputer untuk analisis pergerakan *joint* robot. Beberapa rangkaian elektronik pendukung juga dibuat untuk kebutuhan antar muka perangkat keras robot dengan *microcontroller*.

.

1.2 TUJUAN PENULISAN

Tujuan penulisan skripsi ini adalah untuk merencanakan dan membuat sistem kendali robot lengan *Mitsubishi Movemaster RV-M1* menggunakan

beberapa buah *microcontroller* berbasis *AT89S52* untuk setiap *joint*-nya dan terintegrasi dengan sebuah *microcontroller* utama sebagai pusat kendali.

1.3 RUANG LINGKUP DAN PEMBATASAN MASALAH

1.3.1 Ruang Lingkup

Ruang lingkup penulisan skripsi ini adalah sebagai berikut:

1. Penggunaan beberapa sistem *microcontroller* berbasis *AT89S52* sebagai unit kendali pergerakan lengan robot.
2. Antar muka *microcontroller* dengan *rotary encoder* tipe *incremental* pada poros motor penggerak setiap *joint* yang digunakan sebagai sensor kendali pergerakan lengan robot.
3. Komunikasi antar *microcontroller* serta pengiriman data pergerakan lengan robot secara *serial* ke sebuah *personal computer (pc).*
4. Pengujian performa sistem kendali pergerakan lengan robot.

1.3.2 Pembatasan Masalah

Pembatasan masalah pada penulisan skripsi ini adalah:

1. Perancangan dan pembuatan sistem kendali lengan robot lima derajat kebebasan menggunakan metode *leadthrough (teaching by showing)* berdasarkan beberapa *step* posisi hasil pergerakan setiap *joint*-nya. Data posisi per *step* pergerakan diperoleh melalui pergerakan manual dan disimpan dalam memori pengendali.
2. Pengujian sistem dalam pembacaan beberapa data *step* posisi, penyimpanan data, lintasan pergerakan lengan robot, serta pengiriman data secara *serial* ke sebuah komputer.
3. Pengujian performa dan kemampuan *repeatability* sistem kendali lengan robot untuk pergerakan yang berulang.
4. Adapun kondisi *rotary encoder* sebagai sensor posisi pada *joint 1 (waist)* dalam kondisi tidak berfungsi, sehingga untuk *joint 1* kendalinya bersifat *open loop*. Demikian pula dengan *end-effector* berupa *gripper* yang bersifat tambahan *(optional)*.

1.4 METODE PERANCANGAN

Perancangan dimulai dengan mempelajari susunan sistem kendali robot yang dibuat oleh pabrikannya. Observasi konstruksi mekanik robot dan inventarisasi komponen perangkat keras penunjang sistem kendali dilakukan untuk menyesuaikan perancangan dengan kondisi robot yang ada.

Perancangan dilanjutkan dengan pembuatan beberapa rangkaian sistem *microcontroller* sebagai kendali lengan robot untuk setiap pasangan *joint* dan motor penggerak dengan beberapa rangkaian elektronik pendukung lainnya. Kemudian ditambahkan sebuah *microcontroller* yang digunakan sebagai pusat kendali untuk memonitor pergerakan setiap *joint* dan mengatur proses pengiriman data posisi secara *serial* ke sebuah komputer. Menggunakan beberapa *microcontroller* tadi, setiap *joint* robot dapat digerakkan bergantian maupun bersamaan secara simultan. Data posisi dan pergerakan yang dihasilkan dapat dianalisis untuk evaluasi performa pergerakan robot.

1.5 SISTEMATIKA PENULISAN

Sistematika penulisan pada skripsi ini yaitu pendahuluan pada bab I yang berisi latar belakang masalah, tujuan penelitian, ruang lingkup dan pembatasan masalah, metode penelitian, dan sistematika penulisan. Teori dasar pada bab II berisi teori-teori dasar mengenai peralatan dan sistem yang digunakan dalam perancangan sistem kendali lengan robot *Mitsubishi Movemaster RV-M1*. Selanjutnya perancangan sistem pada bab III berisi prinsip kerja dan perancangan sistem yang dibuat. Analisis dan pengujian sistem pada bab IV berisi pengujian sistem kendali pergerakan robot, baik secara manual maupun otomatis. Akhirnya, kesimpulan pada bab V berisi kesimpulan akhir dari hasil penelitian terhadap rancangan yang dibuat.

BAB II
TEORI DASAR

2.1 ROBOT DAN ROBOTIKA

Robot berasal dari bahasa *Czech* yaitu *robota* yang berarti bekerja. *Webster*, di dalam kamusnya, menyatakan bahwa robot adalah suatu peralatan otomatis yang dapat mencontoh gerakan manusia. Robot adalah manipulator multifungsi yang dapat diprogram ulang dan didesain untuk memindahkan material, peralatan atau piranti tertentu lainnya melalui variasi program pergerakan untuk menyelesaikan berbagai macam pekerjaan. Dengan kata lain, robot adalah manipulator yang bermanfaat secara umum untuk melakukan beraneka ragam pekerjaan dan dapat diprogram ulang serta dilengkapi sensor-sensor eksternal penunjangnya.

2.1.1 Klasifikasi Robot

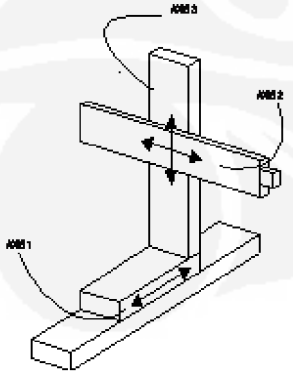
Robot dapat diklasifikasikan berdasarkan:

1. Konfigurasi Fisik.
2. Sistem Pengendalian.
3. Jenis Pengendali.

2.1.1.1 Klasifikasi Robot Berdasarkan Konfigurasi Fisik

Berdasarkan konfigurasi fisiknya, jenis robot dapat dibagi menjadi:

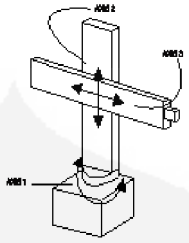
a. Tipe Koordinat Kartesian



Gambar 2.1 Jenis robot tipe koordinat kartesian

(Sumber: Richard D.Klafter, *Robotic Engineering An Integrated Approach*, Prentice-Hall)

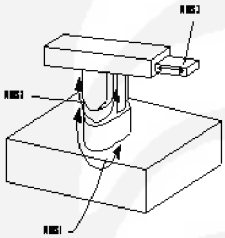
b. Tipe Koordinat Silinder



Gambar 2.2 Jenis robot tipe koordinat silinder

(Sumber: Richard D.Klafter, *Robotic Engineering An Integrated Approach*, Prentice-Hall)

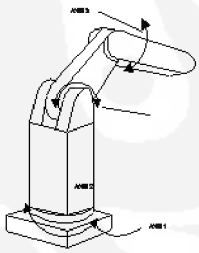
c. Tipe Koordinat Polar



Gambar 2.3 Jenis robot tipe koordinat polar

(Sumber: Richard D.Klafter, *Robotic Engineering An Integrated Approach*, Prentice-Hall)

d. Tipe Koordinat *Revolute*



Gambar 2.4 Jenis robot tipe koordinat *revolute*

(Sumber: Richard D.Klafter, *Robotic Engineering An Integrated Approach*, Prentice-Hall)

e. Tipe *SCARA (Selective Compliance Arm for Robotic Assembly)*

Gambar 2.5 Jenis robot tipe *SCARA*

(Sumber: Richard D.Klafter, *Robotic Engineering An Integrated Approach*, Prentice-Hall)

2.1.1.2 Klasifikasi Robot Berdasarkan Sistem Pengendalian

Berdasarkan sistem pengendaliannya, robot diklasifikasikan menjadi:

a. Robot yang dikendalikan secara elektrik.
b. Robot yang dikendalikan oleh sistem pneumatik.
c. Robot yang dikendalikan oleh sistem hidraulik.

2.1.1.3 Klasifikasi Robot Berdasarkan Jenis Pengendali

Berikut adalah klasifikasi robot berdasarkan tipe pengendaliannya:

a. Robot dengan Gerak Titik ke Titik

   Robot dengan tipe ini tidak terlalu memperhatikan lintasan gerak. Hal yang paling utama adalah bagaimana mencapai titik-titik lokasi yang telah ditentukan dengan waktu yang cepat dan sesuai dengan proses yang diinginkan. Salah satu contoh robot jenis ini antara lain adalah robot pengelasan titik, robot untuk fungsi *assembly*, robot pengambil dan penempat barang *(pick and place robot)*.

b. Robot dengan Jejak Kontinu

   Pada robot dengan jejak kontinu ini titik awal dan akhir pergerakan, bentuk lintasan dan kecepatan gerak sangat dipentingkan. Contoh dari robot jenis ini antara lain robot pengecatan dan robot pengelasan busur.

2.1.2 Metode Pemrograman Robot Industri

Metode pemrograman robot industri yang paling banyak digunakan adalah metode *leadthrough* dan pemrograman secara *off-line*. Perbedaan antara kedua metode ini dapat dijelaskan sebagai berikut:

- Metode *Leadthrough*

  Metode ini membutuhkan *programmer* untuk memindahkan manipulator melalui lintasan pergerakan yang diinginkan dan lintasan tersebut direkam dalam memori oleh pengendali. Metode ini sering disebut sebagai *'teach by showing'*.

- Metode Pemrograman Secara *Off-Line*

  Pemrograman secara *off-line* dilakukan dengan menuliskan perintah gerakan robot yang ditampilkan pada monitor terlebih dahulu seperti

halnya menuliskan sebuah program komputer sebelum gerakan tersebut dieksekusi.

2.1.3 Karakteristik Robot

Karakteristik dan performa sebuah robot dapat dilihat dari berbagai aspek, antara lain:

- Perulangan *(Repeatability)*

  Merupakan sebuah ukuran kemampuan sebuah robot untuk kembali pada posisi-posisi yang sebelumnya telah direkam menggunakan metode pemrograman *leadthrough*. Dengan kata lain, merupakan suatu parameter kemampuan robot untuk melakukan pergerakan berulang.

- Akurasi *(Accuracy)*

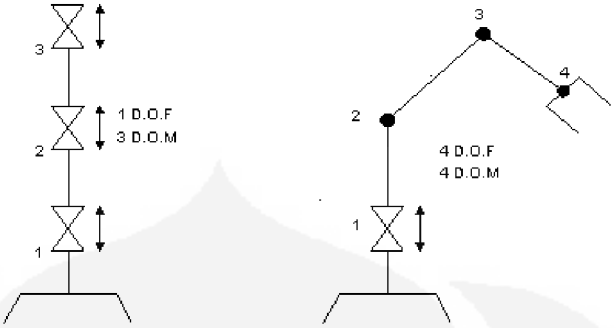
  Merupakan ukuran kemampuan sebuah robot untuk sampai pada posisi yang telah dispesifikasikan (biasanya dengan menggunakan permograman secara *off-line*).

- Derajat Kebebasan *(Degree of Freedom)*

  Derajat kebebasan sebuah robot sama dengan jumlah *joint* yang berdiri sendiri *(independent)* dalam struktur robot yang memiliki referensi koordinat ruang. Dengan kata lain, derajat kebebasan sebuah robot merupakan jumlah pergerakan *independent* dari lengan robot, dilihat dari sudut pandang manipulator akhir *(end-effector)* atau setiap titik sumbu gerakan mekanik pada robot (tidak termasuk *end-effector*).

- Derajat Mobilitas *(Degree of Mobility)*

  Derajat mobilitas dihitung dari total aktuator tanpa melihat referensi koordinat ruang. Derajat mobilitas merupakan kemampuan untuk melakukan sebuah pergerakan (dari setiap pasangan *joint* dan *link-joint*). Sehingga sebuah lengan robot dengan tiga buah *link-joint* yang dihubungkan dengan teleskopik *joint* dapat memiliki tiga buah derajat mobilitas namun hanya memiliki sebuah derajat kebebasan.



Gambar 2.6 *Degree of Freedom (DOF) vs Degree of Mobility (DOM)*

2.2 SENSOR

Sensor berasal dari kata *"to sense"* yang artinya merasakan. Dengan kata lain, secara harafiah sensor dapat berfungsi seolah-olah sebagai suatu indera yang dapat digunakan oleh sistem.

Dalam robotika, elemen sensor digunakan untuk menginformasikan pengendali robot mengenai status/kondisi suatu manipulator. Pada beberapa aplikasi robot, sensor banyak digunakan sebagai elemen pendeteksi suatu posisi, kecepatan, percepatan ataupun informasi lain tentang kondisi manipulator/sistem yang dapat digunakan sebagai umpan balik *(feed-back)* untuk unit kendali sistem tersebut untuk menghasilkan kendali yang diinginkan.

Sensor yang umum digunakan pada robot modern dapat dibedakan menjadi dua kelompok besar, yaitu :

- *Non-Visual Sensor*
- *Visual Sensor*

Kelompok yang pertama meliputi saklar pembatas *(limit/proximity switch)*, sensor posisi (*optical encoder* dan *potensiometer*), sensor kecepatan *(tachometer)*, atau sensor gaya (untuk proteksi pembebanan). Sedangkan kelompok sensor yang kedua meliputi sensor-sensor yang berhubungan dengan pencitraan *(imaging sensor)* seperti kamera dan lainnya.

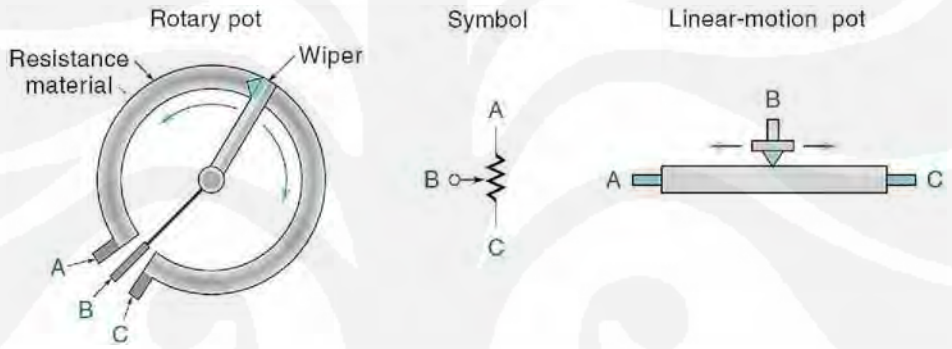
2.2.1 Saklar Pembatas *(Limit/Proximity Switch)*

Saklar pembatas dapat digunakan sebagai sensor pembatas posisi *(proximity sensor)* untuk menentukan posisi suatu manipulator. Elemen sensor ini

dapat dikatakan sangat sederhana karena fungsinya tidak lebih seperti saklar biasa *(on/off)*, namun biasanya melekat pada konstruksi sistem yang digunakan.

2.2.2 *Potensiometer*

Elemen sensor lain yang dapat digunakan untuk mengukur/menyatakan suatu posisi adalah *potensiometer*. Dalam aplikasi robot, sensor seperti ini dapat digunakan untuk memonitor posisi sudut dari *joint* yang berputar *(angular)* ataupun suatu posisi linier. Berikut adalah ilustrasi dari konstruksi umum dari beberapa jenis *potensiometer* yang banyak digunakan untuk aplikasi sensor dan robotika.



Gambar 2.7 Konstruksi umum potensiometer (rotasi dan linier)
(Sumber: www.creativecommons.org)

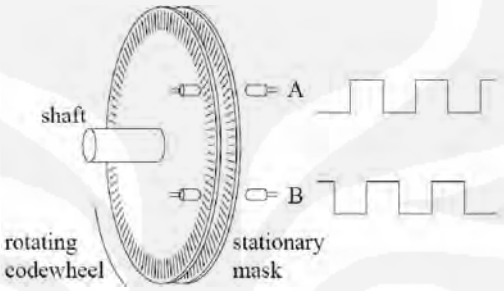
Konstruksi sebuah potensiometer terdiri dari elemen belitan resistif dalam konfigurasi suatu koil. Dengan memberikan tegangan dc melewati hambatan (reistansi material), maka tegangan keluaran akan proporsional dengan kelinieran atau jarak rotasi dari kontaktor geser *(sliding contactor* atau *wiper)* dari titik referensi tertentu.

2.2.3 *Encoder*

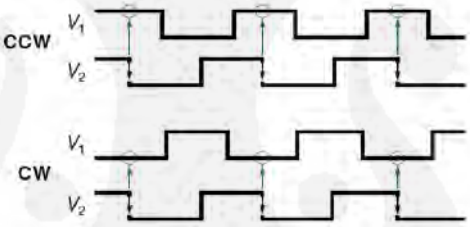
Salah satu komponen yang paling sering digunakan sebagai sensor posisi adalah *encoder*. Berdasarkan kemampuan atau resolusi yang dibutuhkan untuk aplikasi robot, maka perangkat sensor ini dibagi menjadi dua kelas yang berbeda, yaitu:

- *Optical Incremental Encoder*
- *Rotary Absolute Encoder*

Encoder tipe *incremental* adalah tipe *encoder* yang digunakan untuk mengubah pergerakan putar menjadi sinyal digital dengan teknik perbedaan pulsa keluaran. *Encoder* ini biasanya dipasang menjadi satu *(coupled)* dengan bagian rotor motor dc. *Encoder* jenis ini umumnya memiliki sinyal keluaran dengan perbedaan fasa ±90º dimana perbedaan fasa pulsa ±90º ini digunakan untuk membedakan arah pergerakan putarnya.

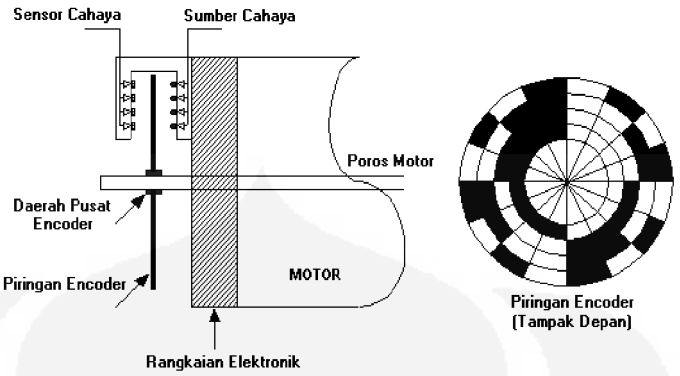


Gambar 2.8 Konstruksi umum sebuah *incremental encoder*
(Sumber: Kevin M. Lynch, *Linear and Rotational Sensors*, Northwestern University)



Gambar 2.9 *Decoding* pulsa keluaran *encoder* penentuan arah putaran
(Sumber: www.creativecommons.org)

Tipe lain dari sensor *encoder* adalah *absolute encoder* yang mampu memberi informasi posisi (berdasarkan putaran) yang lebih akurat dan tepat setiap waktu. *Encoder* jenis ini menghasilkan kode yang khusus *(grey code → binary code)* dan terpisah untuk tiap posisi poros putar. Hal lain yang menjadi keuntungan *encoder* jenis ini adalah kemampuannya untuk "mengingat" posisi terakhir sistem, meskipun daya pada sistem hilang (misalnya kehilangan tenaga/daya sesaat). Informasi tersebut akan disampaikan sistem utama sesaat setelah daya kembali diberikan.



Gambar 2.10 *Absolute encoder* yang dihubungkan pada poros motor (Sumber: www.bipom.com)

2.3 MOTOR DC

2.3.1 Prinsip Kerja Motor DC

Motor dc adalah alat yang berfungsi untuk mengubah energi listrik menjadi energi mekanik berupa gerak rotasi. Bagian motor yang berputar disebut rotor, dan bagian yang diam disebut stator. Medan magnet dihasilkan oleh lilitan pada *frame* yang disebut armatur. Armatur merupakan bagian dari motor dimana input tegangan listrik disuplai. Tipe motor dc yang populer adalah jenis *brushed* (bersikat) dan jenis *brushless* (tanpa sikat). Masing-masing kerja motor menggunakan komutasi untuk menghasilkan arus osilasi dari sumber arus searah. *Brushed dc motor* menghasilkan arus osilasi dalam sebuah lilitan rotor dengan sebuah *commutator split ring*, dan sebuah stator berupa magnet dari induksi kumparan listrik maupun magnet permanen. Rotornya terdiri dari sebuah koil yang melilit rotor yang bertenaga arus listrik searah. Sedangkan pada *brushless dc motor*, *"rotating switch"* mekanis atau susunan *commutator/brushgear* diganti dengan sebuah *switch* elektronik yang disinkronkan dengan posisi rotor. Motor dc jenis ini biasa digunakan untuk aplikasi-aplikasi yang membutuhkan kendali kecepatan yang presisi.

Pada prinsipnya, jika sebuah penghantar dilalui arus listrik, akan dihasilkan medan magnet di sekelilingnya. Bila penghantar ini ditempatkan dalam induksi magnetik, diperoleh gaya *F*. Besarnya gaya yang ditimbulkan sebanding dengan arus listrik dan panjang penghantar *l* yang memotong induksi magnetik *B*. Fenomena ini dapat dinyatakan dengan persamaan sebagai berikut:

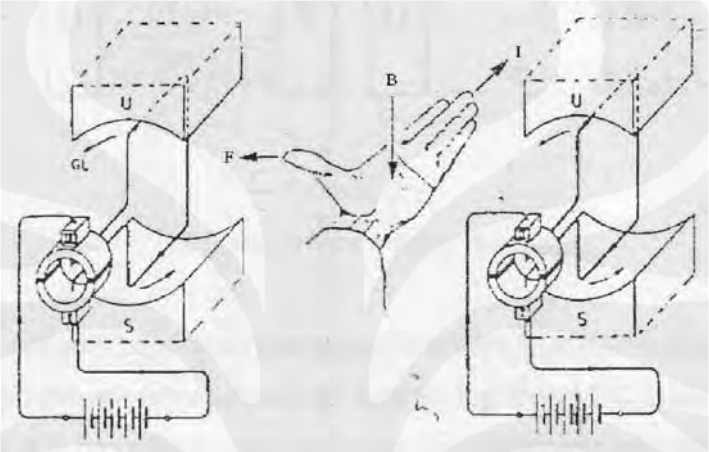
$$\overrightarrow{F} = \overrightarrow{I} \times \overrightarrow{B} \bullet l \quad \textit{(Newton)} \qquad (2.1)$$

dimana: F = gaya yang bekerja pada kawat penghantar *(Newton)*

B = kerapatan fluks magnet *(webber/A)*

l = panjang kawat penghantar *(m)*

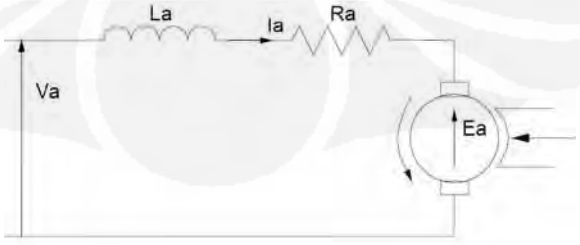
i = arus listrik yang mengalir *(Ampere)*



Gambar 2.11 Hubungan antara gaya (F), kepadatan fluks (B) dan arus (I).
(Sumber: Pancono, Suharyadi, Dipl.Ing,HTL, *Mesin Listrik 1*, Diktat, Polman, Bandung)

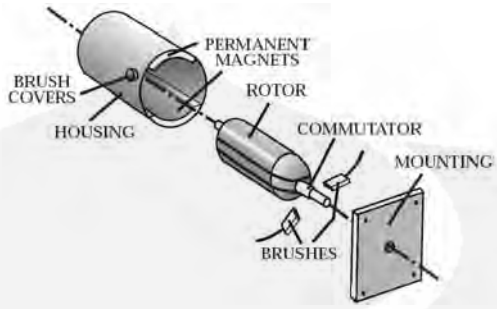
2.3.2 Motor DC Servo

Motor dc servo sebagaimana motor dc lainnya adalah suatu alat untuk mengubah energi listrik menjadi energi mekanik. Magnet permanen motor dc servo mengubah energi listrik ke dalam energi mekanik melalui interaksi dari dua medan magnet. Salah satu medan dihasilkan oleh magnet permanen dan yang lainnya dihasilkan oleh arus yang mengalir dalam kumparan motor. Resultan dari dua medan magnet tersebut menghasilkan torsi yang membangkitkan putaran motor. Saat motor berputar, arus pada kumparan motor menghasilkan torsi yang nilainya konstan.



Gambar 2.12 Rangkaian ekivalen dc servo
(Sumber: www.elektro-terapan.blogspot.com)

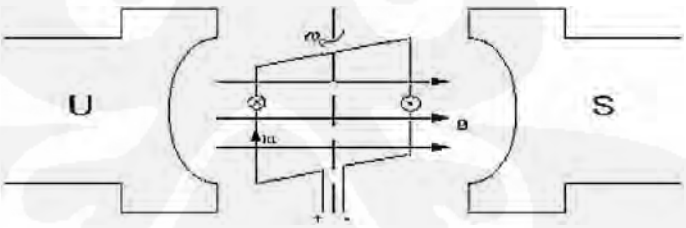
Pada motor dc servo terdapat tiga komponen utama, yaitu armatur, magnet permanen dan komutator.



Gambar 2.13 Konstruksi umum motor dc

(Sumber: *Baldor motor and drives, handbook, Baldor Electric Company*)

Prinsip kerja motor didasarkan pada peletakan suatu konduktor dalam suatu medan magnet. Jika suatu konduktor dililitkan dengan kawat berarus maka akan dibangkitkan medan magnet berputar. Kontribusi dari setiap putaran akan merubah intensitas medan magnet yang ada dalam bidang yang tertutup kumparan. Dengan cara inilah medan magnet yang kuat terbentuk. Tenaga yang digunakan untuk mendorong fluks magnet tersebut disebut *Magnetomotive Force (MMF).* Fluks magnet digunakan untuk mengetahui seberapa banyak fluks pada daerah disekitar koil atau magnet permanen.



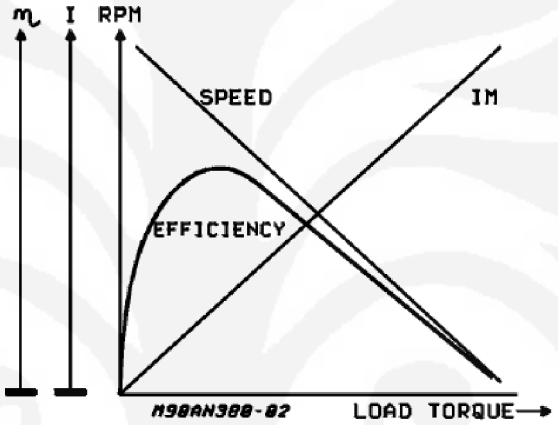
Gambar 2.14 Ilustrasi kerja rangkaian motor servo dc

(Sumber: www.elektro-terapan.blogspot.com)

Medan magnet pada motor dc servo dibangkitkan oleh magnet permanen. Fluks medan magnet pada stator tidak dipengaruhi oleh arus armatur. Oleh karena itu, kurva perbandingan antara kecepatan dengan torsi adalah linier (dibahas pada sub bab selanjutnya mengenai torsi motor dc). Hal inilah yang membedakan motor dc servo dengan motor dc lainnya, sehingga sering digunakan untuk aplikasi yang membutuhkan kendali posisi yang presisi.

2.3.3 Torsi

Torsi motor merupakan perputaran dari suatu gaya terhadap suatu poros. Karakteristik torsi motor dc adalah bahwa saat kecepatan armatur tanpa beban (N_A) maka *output* torsi adalah nol. Jadi, selama motor tidak diberi beban maka tidak diperlukan torsi yang besar. Pada kondisi ini arus armatur harus mengalir untuk mengatasi rugi-rugi mekanik dan elektrik dari motor pada saat motor tanpa beban. Rugi-rugi mekanik dapat diakibatkan oleh gesekan. Sedangkan rugi-rugi elektrik antara lain diakibatkan oleh rugi-rugi daya, histeresis dan juga arus *eddy* pada rotor.



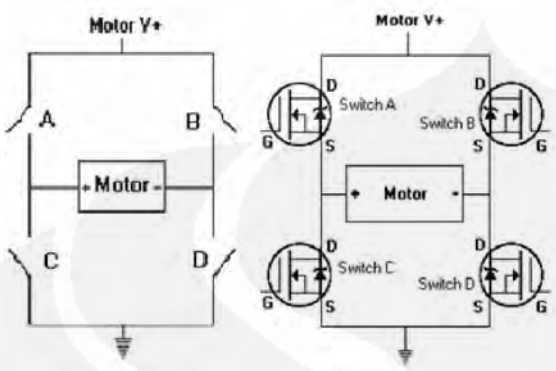
Gambar 2.15 Kurva karakteristik arus, kecepatan, efisiensi dan torsi motor dc
(Sumber: *AN380 Application Note, ST Microelectronics*)

Pada saat motor terbebani, daya keluaran dan arus armatur akan meningkat. Efisiensi akan naik dengan dengan cepatnya sampai nilai maksimum dan kemudian akan turun kembali. Daya keluaran maksimum akan terjadi pada saat torsi maksimum.

2.3.4 *Driver* Motor DC

Motor dc yang mempunyai arus besar tidak bisa langsung dihubungkan dengan *microcontroller* sebagai pengendali dengan arus keluarannya yang kecil sehingga membutuhkan sebuah antarmuka berupa *driver* motor. Prinsip dasar kerja sebuah *driver* motor dc adalah proses *switching* kedua terminal motor. Fungsi *switching* tersebut dapat digantikan oleh piranti *switching* seperti transistor/*MOSFET*. Berikut adalah ilustrasi kerja suatu rangkaian *driver* motor dc

(logika ”1” adalah kondisi *high/on*, sedangkan logika ”0” menyatakan kondisi *low/off*):



A	B	C	D	State	A	B	C	D	State
0	0	0	0	Off	1	0	0	0	Off
0	0	0	1	Off	1	0	0	1	Forward
0	0	1	0	Off	1	0	1	0	SHORT!!
0	0	1	1	Brake	1	0	1	1	SHORT!!
0	1	0	0	Off	1	1	0	0	Brake
0	1	0	1	SHORT!!	1	1	0	1	SHORT!!
0	1	1	0	Reverse	1	1	1	0	SHORT!!
0	1	1	1	SHORT!!	1	1	1	1	SHORT!!

Gambar 2.16 Prinsip kerja driver motor dc

2.3.5 Teknik *PWM (Pulse Width Modulation)*

Teknik *PWM (Pulse Width Modulation)* merupakan salah satu teknik manipulasi kendali motor dc atau perangkat elektronik lainnya yang berarus besar. Metode ini menggunakan teknik *cut-off* dan saturasi yaitu dengan mengatur lebar pulsa tegangan.

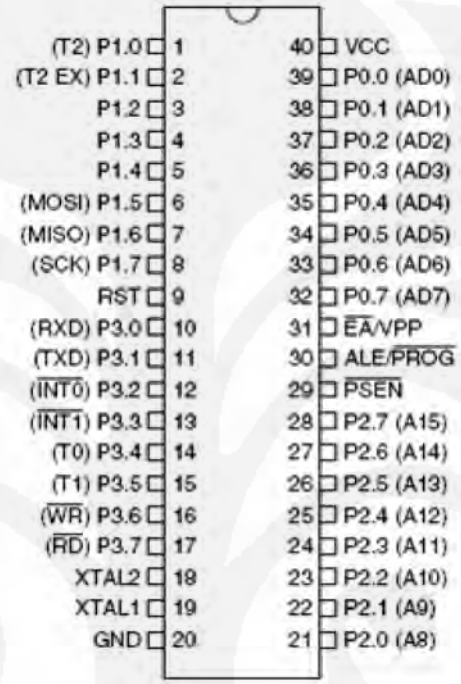
Gambar 2.17 Bentuk pulsa *PWM*

Gambar 2.16 mengilustrasikan bentuk pulsa *PWM*. Lebar pulsa *PWM* dinyatakan dengan *duty cycle (%)*. Dengan mengatur lebar pulsa *ton* dan *toff* dalam satu perioda pulsa *(T)* maka kecepatan motor yang dihasilkan dapat diatur bervariasi sesuai kebutuhan. Semakin lebar pulsa *ton* mendekati periode *T* maka kecepatan motor akan semakin mendekati kecepatan maksimumnya.

$$DutyCycle = \frac{Ton}{T} \times 100\% \qquad (2.2)$$

2.4 *MICROCONTROLLER* AT89S52

Microcontroller AT89S52 adalah mikroprosesor yang dilengkapi memori, dan perangkat masukan serta keluaran di dalamnya. *Microcontroller* ini diproduksi oleh *ATMEL* dan termasuk keluarga *MCS51*.



Gambar 2.18 Konfigurasi pin *AT89S52*
(Sumber: *Datasheet AT89S52,* Atmel)

Spesifikasi dari *microcontroller AT89S52* antara lain:

- Merupakan *CPU* 8-bit yang termasuk dalam keluarga *MCS51*.
- 8-Kbyte *In-System Programmable (ISP) Flash Memory.*
- 256 x 8-bit *Internal RAM.*
- 32 jalur *I/O* yang dapat diprogram.
- Tiga buah *timer/counter* 16 *bit.*
- Delapan buah sumber interupsi.
- Sebuah port serial dengan *serial full duplex UART control.*
- Kecepatan pelaksanaan instruksi per siklus adalah 1 mikrodetik pada frekuensi *clock* 11.5902 MHz.
- Waktu pemrograman yang relatif singkat.

- Fleksibel *ISP Programming (byte* dan *page mode).*
- Kemampuan tulis/hapus program sampai dengan 10000 kali.

2.4.1 *Timer/Counter AT89S52*

AT89S52 menyediakan fasilitas *timer/counter* 16-bit sebanyak tiga buah, yaitu *timer0, timer1* dan *timer2*. *Timer* bekerja dengan cara menghitung pulsa *clock internal microcontroller* yang dihasilkan dari rangkaian osilator. Jumlah pulsa *clock* tersebut dibandingkan dengan sebuah nilai yang terdapat dalam *register timer (TH dan TL).* Jika jumlah pulsa *clock* sama sengan nilai *timer*, maka sebuah *interrupt* akan terjadi (ditandai oleh *flag TF*). *Interrupt* ini dapat dipantau oleh program sebagai tanda bahwa *timer* telah mengalami *overflow.* Sub-rutin *timer interrupt* dieksekusi pada kondisi ini.

Tabel 2.1 *Timer2 register* dan mode operasinya

T2CON Address = 0C8H — Reset Value = 0000 0000B

Bit Addressable

Bit	TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/$\overline{T2}$	CP/$\overline{RL2}$
	7	6	5	4	3	2	1	0

RCLK +TCLK	CP/$\overline{RL2}$	TR2	MODE
0	0	1	16-bit Auto-reload
0	1	1	16-bit Capture
1	X	1	Baud Rate Generator
X	X	0	(Off)

T2MOD Address = 0C9H — Reset Value = XXXX XX00B

Not Bit Addressable

Bit	–	–	–	–	–	–	T2OE	DCEN
	7	6	5	4	3	2	1	0

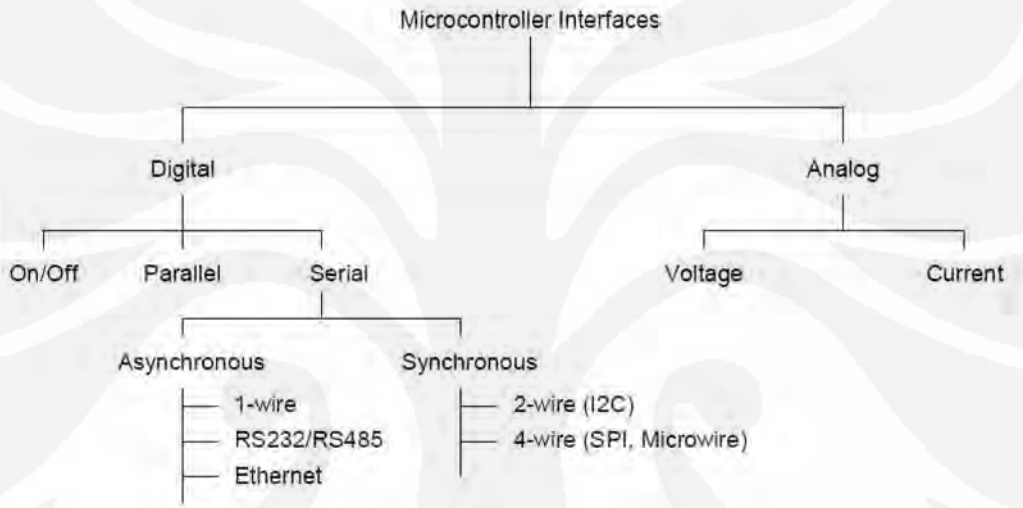
(Sumber: *Datasheet AT89S52,* Atmel)

Timer2 pada *microcontroller AT89S52* merupakan fasilitas timer tambahan yang tidak dimiliki oleh *microcontroller MCS51* versi lain di bawahnya. *Timer2* merupakan 16-bit *timer/counter* yang dapat dioperasikan sebagai *timer* ataupun *counter* dan mampu melakukan mode operasi khusus yang tidak dapat dilakukan oleh *timer0* dan *timer1*. Tiga mode operasi untuk *timer2* adalah mode *capture, auto-reload (up or down counting)* dan *baud rate*

generator. Khusus untuk penggunaan sebagai *counter*, cacah naik atau turun dapat diperoleh dengan mengatur nilai bit *DCEN (Down Counter Enable)* pada register *t2mod*.

2.4.2 Teknik Antarmuka *Microcontroller*

Berbagai teknik antarmuka *microcontroller* telah dikembangkan untuk mengatasi tuntutan kompleksitas dan fitur sistem yang ada. Berikut adalah gambaran beberapa di antaranya:



Gambar 2.19 Beberapa teknik antarmuka *microcontroller*

Beragam pilihan teknik antarmuka di atas digunakan untuk kebutuhan komunikasi *microcontroller* dengan piranti lain seperti sensor, memori, tampilan *LCD*, bahkan untuk kebutuhan komunikasi dengan *microcontroller* lainnya.

2.4.3.1 Komunikasi Data Serial

Komunikasi data secara serial dibedakan menjadi dua, yaitu: komunikasi data serial secara sinkron dan asinkron. Pada komunikasi data serial secara sinkron, *clock* dikirimkan bersama-sama dengan data serial, sedangkan *clock* pada komunikasi data asinkron tidak dikirimkan bersama data serial, melainkan dibangkitkan terpisah baik pada sisi pengirim *(transmitter)* maupun pada sisi

penerima *(receiver)*. Port serial pada *IBM PC* adalah contoh komunikasi serial secara asinkron.

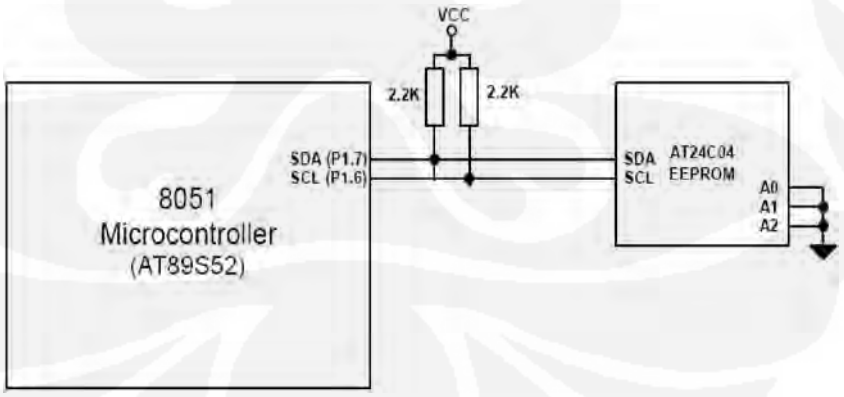
2.4.3.2 *Inter Integrated Circuit (I²C Bus)*

Teknik antarmuka menggunakan *Inter Integrated Circuit (I²C Bus)* merupakan salah satu teknik antarmuka *master-slave microcontroller*. Sinyal komunikasi pada teknik ini menggunakan dua buah jalur komunikasi, yaitu:

- *Serial Data (SDA)*
- *Serial Clock (CLK)*

Untuk memulai komunikasi, *bus master* menunjuk alamat dari *device* yang akan dihubungi *(slave)*. Setiap *slave device* selalu memonitor jalur komunikasi *(bus)* apabila *master device* mengirimkan alamat *slave*. *Master device* hanya akan berkomunikasi dengan *slave device* dengan alamat yang benar.

Beberapa keuntungan menggunakan teknik ini antara lain adalah kemudahan akses beberapa *slave device* yang hanya menggunakan dua buah jalur/kabel serta implementasinya yang mudah dalam konfigurasi *master-slave*. Namun, beberapa kerugian dari teknik ini antara lain jangkauan jarak yang pendek, kecepatan yang relatif lambat (100/400KHz) serta terbatasnya pengalamatan *device*.



Gambar 2.20 Contoh teknik antarmuka menggunakan *i²c bus*

BAB III
PERANCANGAN SISTEM

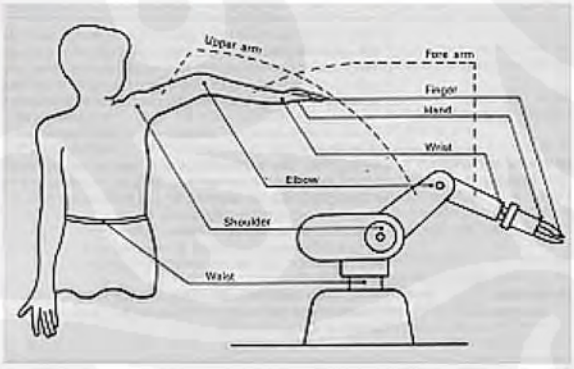
3.1 KONFIGURASI UMUM

3.1.1 Konstruksi Robot

Mitsubishi Movemaster RV-M1 Micro-Robot memiliki lima *joint* pergerakan, ditambah dengan sebuah *end-effector* berupa *gripper*. Masing-masing *joint* digerakkan oleh motor dc servo yang pada porosnya terhubung pada sebuah *encoder* tipe *incremental* sebagai sensor posisi pergerakan motor. Resolusi *encoder* yang digunakan pada robot ini adalah 200 pulsa per putaran (untuk *joint* 1, 2 dan 3) serta 96 pulsa per putaran (untuk *joint* 4 dan 5).

Berikut adalah bagian-bagian konstruksi robot beserta ilustrasinya:

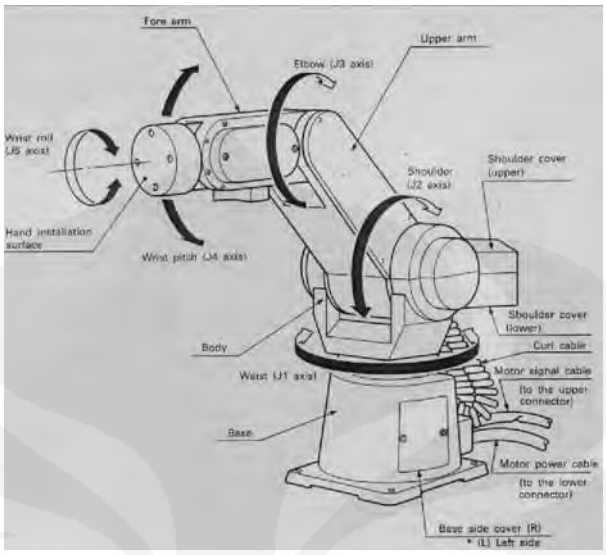
Joint 1 (J1) : bagian pinggang *(waist)*
Joint 2 (J2) : bagian bahu *(shoulder)*
Joint 3 (J3) : bagian siku lengan *(elbow)*
Joint 4 (J4) : pergerakan naik/turun pergelangan tangan *(wrist pitch)*
Joint 5 (J5) : pergerakan putar pergelangan tangan *(wrist roll)*
Gripper : bagian tangan robot *(end-effector)*



Gambar 3.1 *Mitsubishi Movemaster RV-M1 Micro-Robot*
(Sumber: diolah kembali dari buku manual *Mitsubishi Movemaster RV-M1*)

3.1.2 Spesifikasi Standar Pergerakan Robot

Spesifikasi standar jangkauan pergerakan setiap *joint* dari lengan robot dapat dilihat pada gambar dan tabel berikut:



Gambar 3.2 Pergerakan tiap *joint* lengan robot

(Sumber: Buku manual *Mitsubishi Movemaster RV-M1*)

Tabel 3.1 Spesifikasi standar pergerakan robot

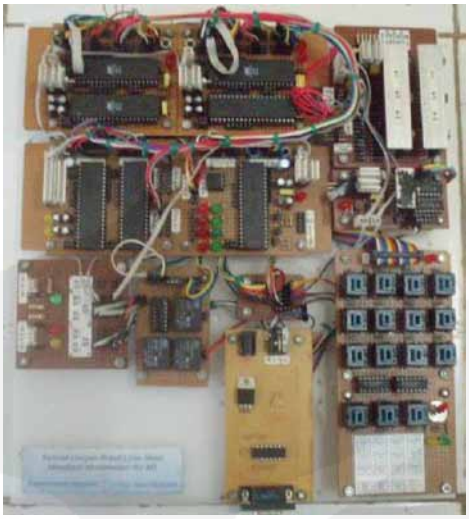
Item		Spesifikasi	Keterangan
Konstruksi Mekanik		Lima derajat kebebasan	
Jangkauan Pergerakan	*Waist Rotation*	300° (max. 120°/sec)	*Joint* J1
	Shoulder Rotation	130° (max. 72°/sec)	*Joint* J2
	Elbow Rotation	110° (max. 109°/sec)	*Joint* J3
	Wrist Pitch	± 90° (max. 100°/sec)	*Joint* J4
	Wrist Roll	± 180° (max. 163°/sec)	*Joint* J5
Panjang Lengan	Lengan Atas	250 mm	
	Lengan Bawah	160 mm	
Kecepatan maksimum *(path)*		1000 mm/sec	
Sistem Penggerak		*DC Servo*	
Berat Keseluruhan		± 19 kg	
Kapasitas Motor		*Joint* J1-J3 : 30 W	
		Joint J4, J5 : 11 W	

(Sumber: diolah kembali dari buku manual *Mitsubishi Movemaster RV-M1*)

3.1.3 Konfigurasi Kendali Robot

3.1.3.1 Perangkat Keras

Perancangan dan implementasi perangkat keras sistem kendali pergerakan robot yang dibuat dapat dilihat melalui ilustrasi pada gambar 3.3.

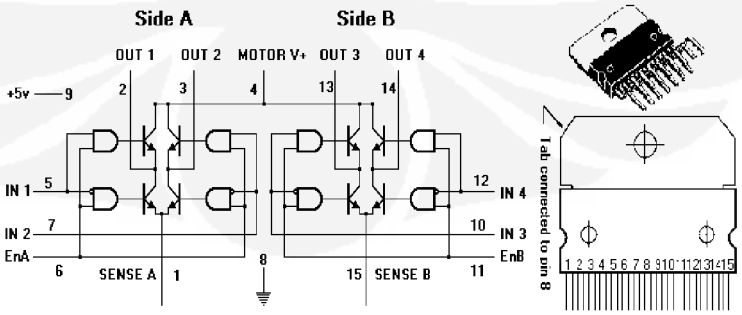


Gambar 3.3 Perangkat keras sistem kendali pergerakan robot

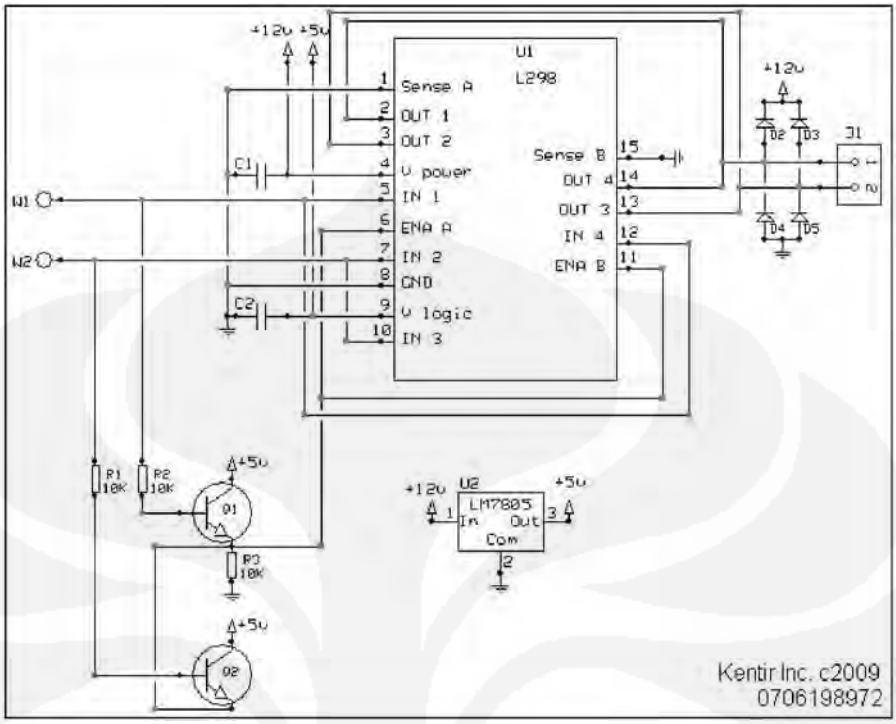
Perancangan sistem kendali lengan robot yang dibuat terdiri dari beberapa rangkaian *microcontroller* berbasis *AT89S52* produksi *Atmel* sesuai jumlah *joint* robot dan motor penggeraknya *(slaves)*. Sebuah *microcontroller* jenis yang sama ditambahkan sebagai kendali pusat *(master)*. Komunikasi antar *microcontroller* menggunakan *i²c bus (inter integrated circuit)*, dimana sebuah *serial EEPROM AT24C256* digunakan untuk menyimpan beberapa data *step* posisi pergerakan setiap *joint*. Data posisi dan lintasan pergerakan setiap *joint* dikirim secara *serial* ke sebuah komputer untuk ditampilkan ke layar monitor melalui *hyperterminal*.

3.1.3.2 *Driver* Motor

Driver motor penggerak lengan robot menggunakan *dual full h-bridge* yang dikemas dalam sebuah *IC L298D*. Dua rangkaian *h-bridge* disusun paralel untuk meningkatkan kerja motor dengan arus yang lebih besar:



Gambar 3.4 Diagram blok *IC L298D dual full h-bridge*
(Sumber : *Datasheet L298D,* ST Microelectronics)



Gambar 3.5 Skematik rangkaian *driver* motor menggunakan *IC L298D*

3.1.3.3 Perangkat Lunak

Secara garis besar, perancangan program sistem kendali pergerakan lengan robot dapat dibagi menjadi dua bagian yaitu:

- Pergerakan robot secara manual

  Pergerakan robot secara manual dilakukan dengan menggunakan *keypad control*. Beberapa *step* posisi pergerakan robot kemudian disimpan untuk selanjutnya digunakan sebagai *set point* langkah pergerakan robot secara otomatis.

- Pergerakan robot secara otomatis

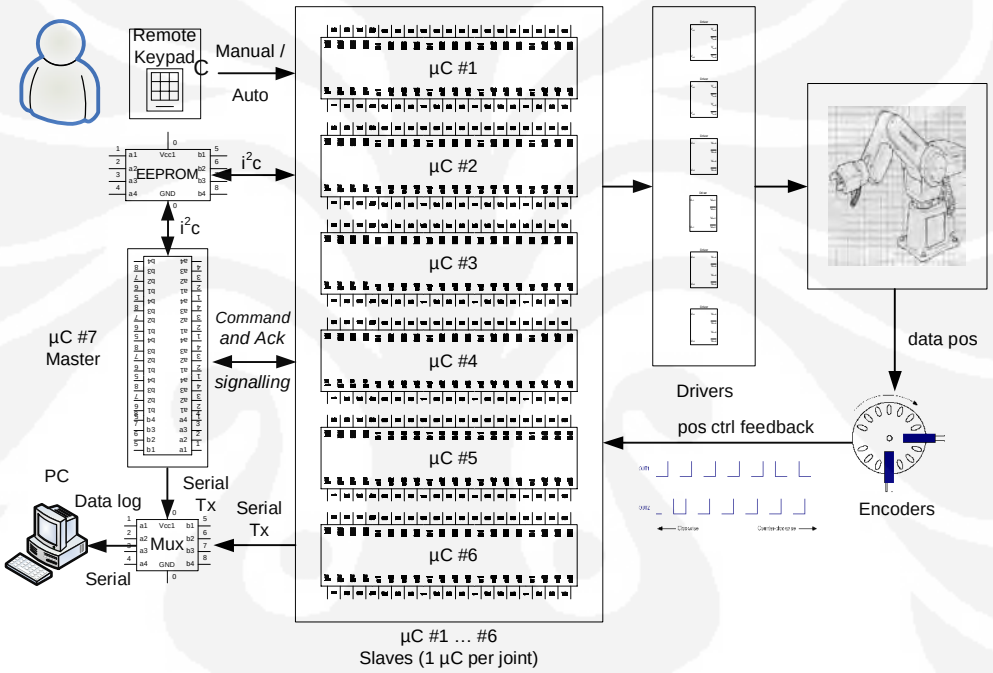
  Pergerakan robot secara otomatis dilakukan berdasarkan beberapa data *step* posisi pergerakan setiap aksis yang diperoleh dari hasil penyimpanan data posisi pergerakan manual sebelumnya melalui proses pembelajaran *(metode leadthrough – teaching by showing)*

3.2 PRINSIP KERJA SISTEM

Pergerakan robot secara manual dilakukan menggunakan tombol *keypad control*. Beberapa fungsi umum operasi robot yang terdapat pada *keypad* antara

lain *toggle switch* pilihan mode operasi (manual dan otomatis), tombol kendali pergerakan robot untuk setiap *joint* (dua buah untuk setiap *joint*-nya), tombol simpan data posisi per *step*, tombol *load* data posisi dan tombol *reset* data posisi.

Setiap *joint* dikendalikan menggunakan sebuah *microcontroller* terpisah dengan tujuan agar setiap aksis dapat digerakkan secara simultan/bersamaan. Masing-masing aksis digerakkan oleh sebuah motor dc servo yang pada porosnya terpasang *rotary encoder* tipe *incremental* sebagai sensor posisi. Gambar berikut adalah blok diagram sistem kendali robot yang dibuat:

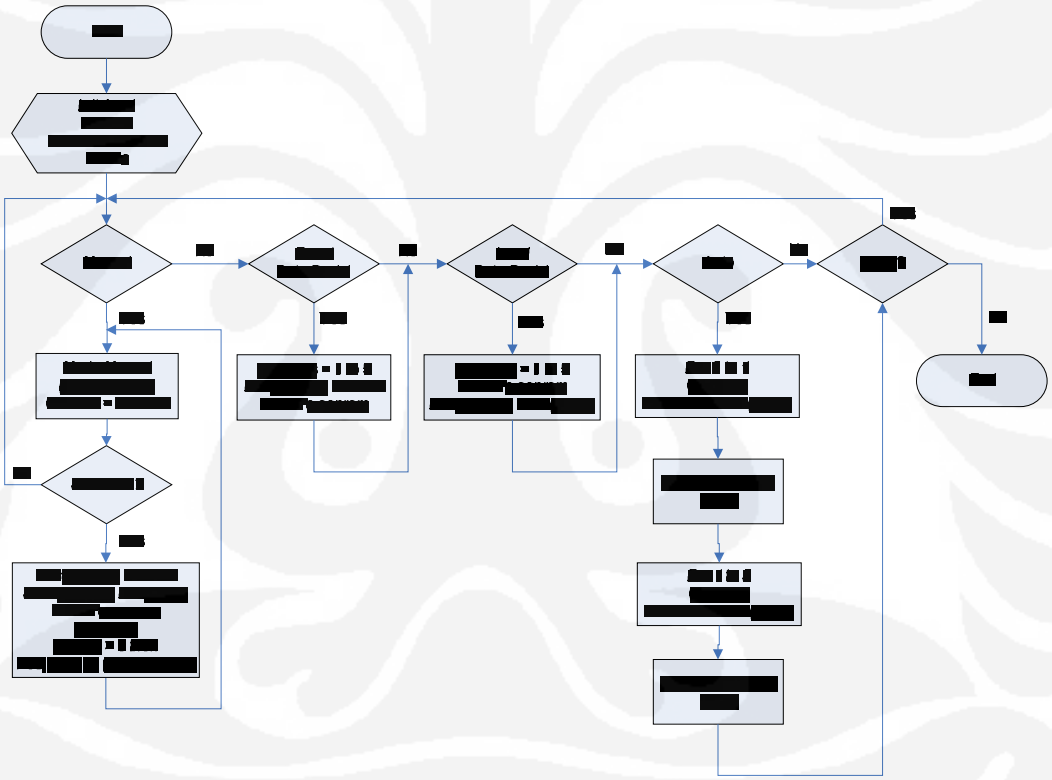


Gambar 3.6 Blok diagram sistem kendali pergerakan robot

Beberapa *step* posisi pergerakan robot berupa nilai pembacaan *encoder* dapat disimpan dalam memori pengendali untuk digunakan sebagai nilai *set point* pergerakan robot secara otomatis (tombol simpan data posisi per *step*). Karena keterbatasan memori internal pengendali, maka data *step* posisi maksimal yang dapat disimpan adalah lima buah (lima *step* posisi). Apabila data posisi yang dimasukkan lebih dari lima *step*, maka posisi terakhir tersebut akan menjadi *step* posisi yang ke lima. Data *step* posisi pergerakan setiap *joint* juga disimpan dalam sebuah *i²c eeprom AT24C256* menggunakan jalur *i²c bus*. Proses *handshaking*

(acknowledge signalling) antar *microcontroller* menjadi pengatur akses jalur serial data ini.

Proses simpan/tulis data *step* pergerakan tiap *joint* diatur oleh sinyal *request* dari *master microcontroller* ke *slaves microcontroller*. Sedangkan proses baca data pergerakan tiap *joint* dari *EEPROM* diatur berdasarkan sinyal *acknowledgement* dari *slaves microcontrollers* yang diterima oleh *microcontroller* kendali utama. Sinyal *acknowledgment* digunakan sebagai tanda bahwa data telah selesai ditulis oleh *slave microcontroller* dan siap untuk dibaca oleh *microcontroller* kendali utama. Berikut adalah diagram alir kerja sistem kendali secara umum untuk pergerakan setiap *joint* robot:



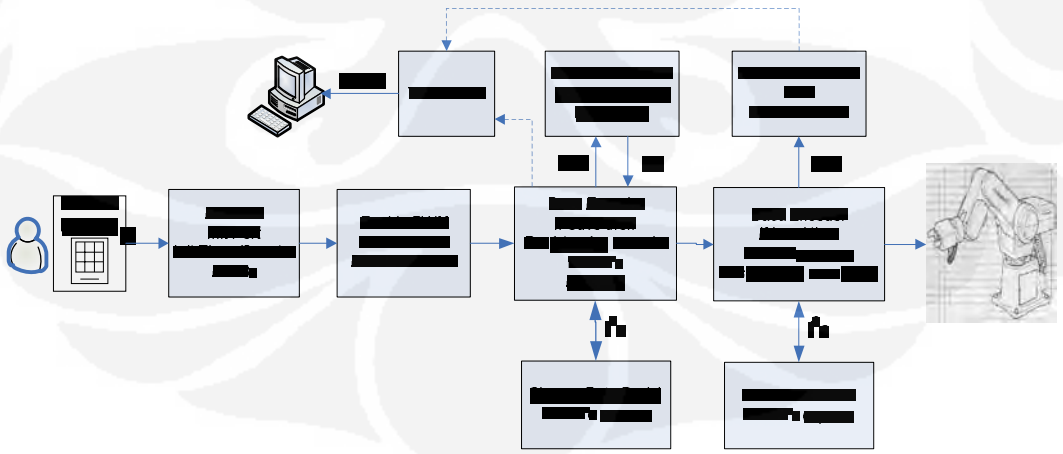
Gambar 3.7 Diagram alir kerja sistem secara umum

Pergerakan robot secara otomatis dilakukan berdasarkan beberapa data *step* posisi pergerakan setiap *joint* yang sudah disimpan terlebih dahulu saat pergerakan manual robot. Data posisi tersebut berupa nilai pembacaan *encoder* setiap *step* pergerakan manual robot, yaitu saat tombol *keypad* untuk perintah simpan data posisi per *step* ditekan. Kemudian, masing-masing *joint* robot akan

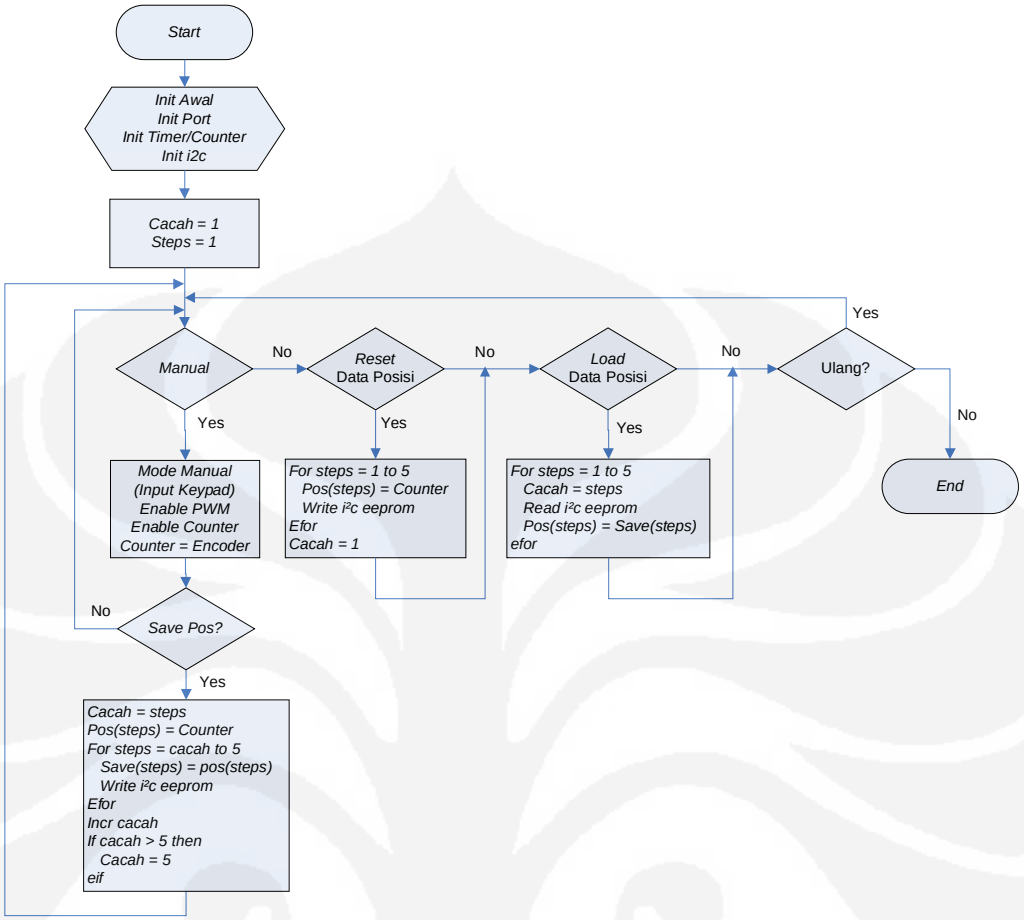
digerakkan secara otomatis oleh motor menuju *step* posisi tersebut dengan membandingkan pembacaan *rotary encoder* dengan data posisi tadi sebagai nilai *set point* yang harus dicapai. Lebih lanjut, data pergerakan robot berupa nilai pembacaan *encoder* pada setiap *joint* dikirim secara serial ke komputer dan ditampilkan melalui *hyperterminal* untuk digunakan sebagai monitor proses pergerakan lintasan lengan robot.

3.3 PERANCANGAN PROGRAM PERGERAKAN MANUAL

Bahasa pemrograman yang digunakan adalah *BASIC* untuk *MCS-51®* dengan *Basic Compiler BASCOM-8051©*. Perancangan program pergerakan manual dibuat untuk beberapa *microcontroller* kendali pergerakan tiap *joint* (satu *microcontroller* per *joint*). Namun, algoritmanya relatif sama. Apabila terdapat beberapa perbedaan kecil dikarenakan untuk menyesuaikan kondisi dan karakteristik pergerakan masing-masing *joint*. Nilai pembacaan *encoder* sebagai sensor posisi pergerakan setiap *joint* dikirim secara serial ke sebuah komputer menggunakan jalur *serial Tx microcontroller* masing-masing *joint*. Jalur *serial Tx* ini digunakan secara bergantian oleh beberapa *microcontroller* (*master* dan *slave*) menggunakan *multiplexer 8x1 (74HC151)* yang diatur oleh *microcontroller* kendali pusat *(master)*. Berikut adalah blok fungsional dan diagram alir program perrgerakan robot secara manual:



Gambar 3.8 Blok fungsional program pergerakan manual setiap *joint*



Gambar 3.9 Diagram alir program pergerakan manual robot

Pergerakan setiap aksis dibuat terpisah karena menggunakan sebuah *microcontroller* untuk setiap aksisnya. Teknik *PWM (Pulse Width Modulation)* digunakan untuk mengatur kecepatan pergerakan setiap aksis. *Timer0* sebagai *timer* 16-bit (mode 1) digunakan pada program untuk mengatur lebar pulsa *PWM*. Lebar pulsa *PWM* ini bervariasi pada setiap aksisnya tergantung karakteristik dan beban pergerakan motor masing-masing *joint*.

Tabel 3.2 *PWM duty cycle* pergerakan tiap *joint*

Duty Cycle	*Clockwise*	*Counter Clockwise*	Keterangan
Joint #1	75%	75%	
Joint #2	50% (turun)	90% (naik)	beban naik > beban turun
Joint #3	50% (turun)	90% (naik)	beban naik > beban turun
Joint #4	80%	80%	
Joint #5	75%	75%	
Gripper	75%	75%	

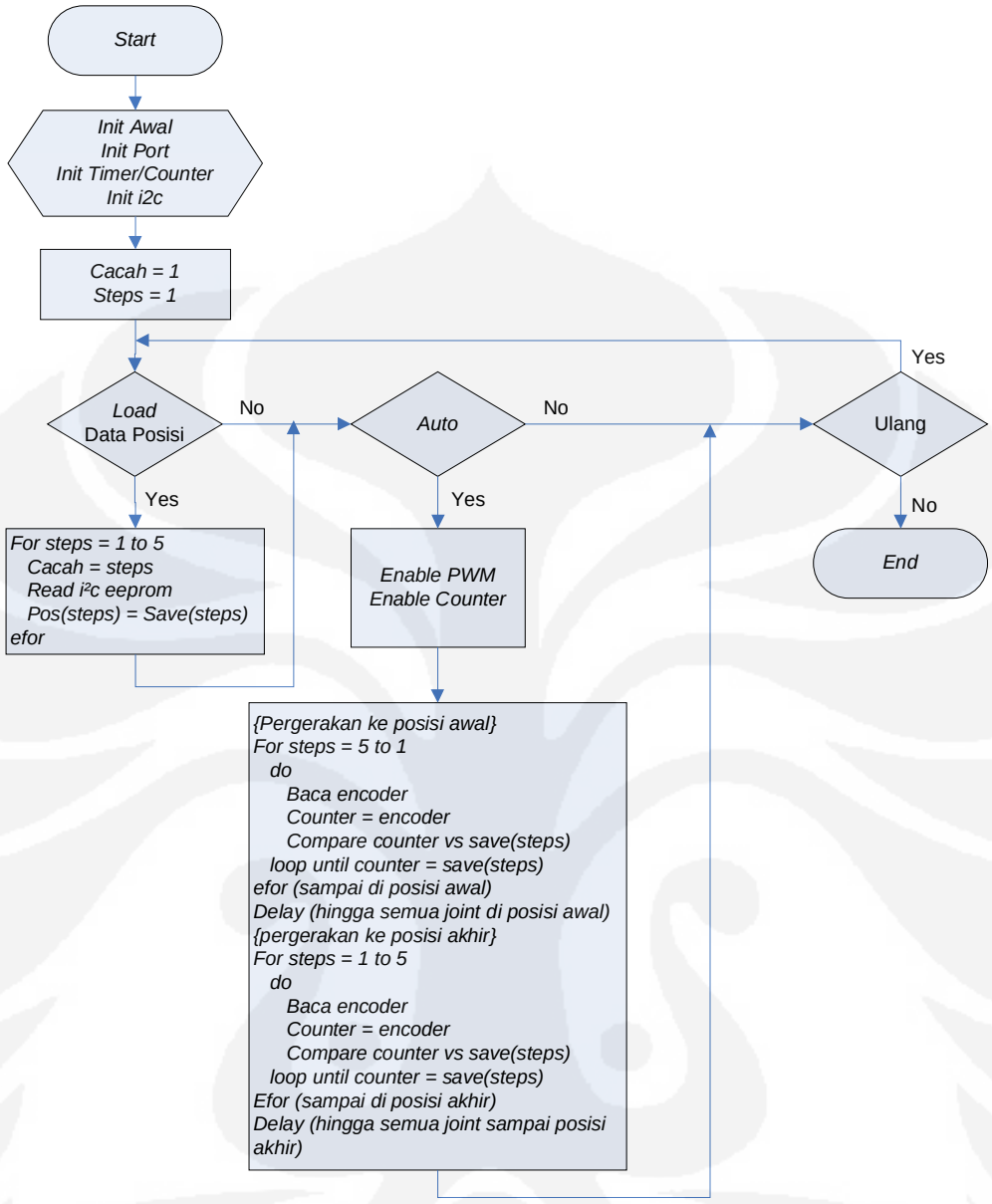
Timer0 register Th0 dan *Tl0* diberi nilai tertentu sesuai lebar pulsa *Ton PWM* yang dikehendaki. Saat terjadi *over-flow, timer0 interrupt* terjadi dan *timer0 register Th0* dan *Tl0* kembali diberi nilai tertentu sesuai lebar pulsa *Toff PWM*. Proses tersebut berlangsung berulang-ulang sehingga menghasilkan pulsa *on-off* dengan *duty cycle* tertentu yang dikehendaki untuk mengatur kecepatan motor.

Data nilai pembacaan *encoder* disimpan pada *array* variabel data posisi dan juga disimpan pada memori serial *AT24C256* setiap kali tombol simpan data posisi per *step* ditekan. Selanjutnya, data ini akan menjadi nilai *set point* pada mode operasi pergerakan robot secara otomatis. Tombol *reset* data posisi akan mengisi semua nilai *array* variabel data posisi berupa pembacaan *encoder* pada posisi terakhir. Sedangkan, tombol *load* data posisi akan mengisi *array* variabel data posisi dengan nilai pembacaan *encoder* yang sudah disimpan sebelumnya.

3.4 PERANCANGAN PROGRAM PERGERAKAN OTOMATIS

Pergerakan robot secara otomatis dilakukan berdasarkan beberapa data posisi berupa nilai pembacaan *encoder* setiap aksis hasil penyimpanan data posisi pergerakan manual sebelumnya. Data posisi nilai pembacaan *encoder* ini dibaca kembali, kemudian dibandingkan dengan pembacaan *encoder* saat pergerakan lengan robot secara otomatis. Berikut adalah blok fungsional dan diagram alirnya:

Gambar 3.10 Blok fungsional program pergerakan otomatis setiap joint



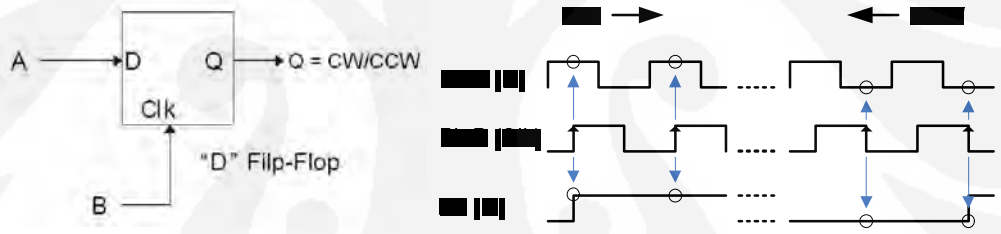
Gambar 3.11 Diagram alir program pergerakan otomatis

Pergerakan otomatis setiap *joint* dimulai dengan pencarian posisi awalnya dengan membandingkan pembacaan *encoder* terhadap *set point* berupa nilai pembacaan *encoder* yang sudah disimpan sebelemnya dalam memori pengendali (mode manual). Setelah posisi awal tercapai, setiap *joint* akan menunggu sampai keseluruhan *joint* sampai pada posisi awalnya masing-masing. Keseluruhan *joint* kemudian kembali bergerak menuju posisi akhirnya berdasarkan nilai *set point* masing-masing nilai pembacaan *encoder* per *step* posisi pergerakan yang sudah disimpan sebelumnya. Pergerakan tersebut akan berlangsung berulang-ulang

hingga pilihan mode otomatis dihentikan (*toggle switch* mode *auto off*). Selama pergerakan dalam mode otomatis berjalan, data pergerakan setiap *joint* berupa nilai pembacaan *encoder* dikirim secara serial untuk ditampilkan melalui *hyperterminal*.

3.5 DATA POSISI PERGERAKAN TIAP *JOINT*

Data posisi pergerakan tiap *joint* diperoleh melalui pembacaan *rotary encoder* yang terhubung pada motor penggerak setiap *joint*. Pulsa keluaran *phase A* menyatakan posisi, sedangkan pulsa keluaran *phase B* menyatakan arah putaran. Sebuah *d-flip-flop* yang dikemas dalam *IC 74LS74* digunakan sebagai detektor arah pergerakan motor. *Phase A encoder* digunakan sebagai masukan *d-flip-flop (D)* dan *phase B encoder* digunakan sebagai *clock (Clk)*. Kondisi keluaran *(Q)* akan sama dengan kondisi masukan *d-flip-flop (D)* setiap *clock* positif terjadi (kondisi *low to high*).

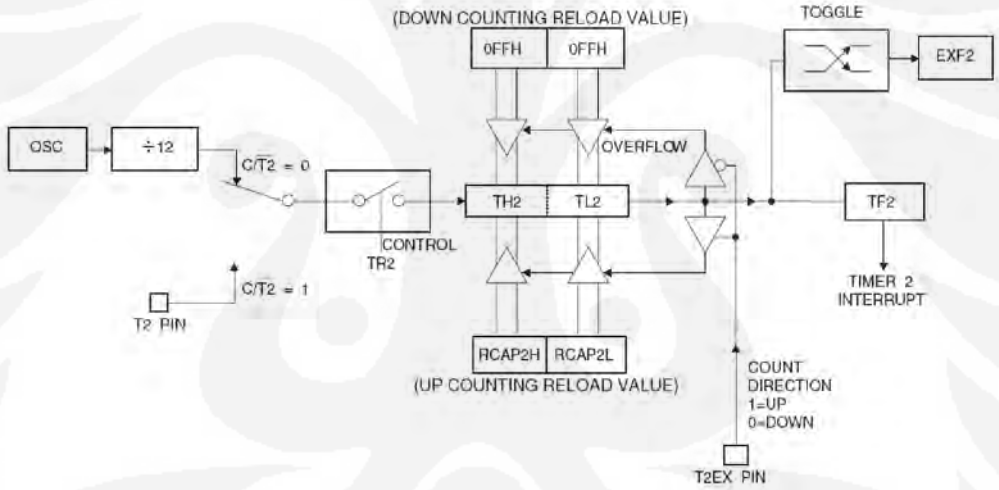


Gambar 3.12 Detektor arah putaran *rotary encoder*

Timer2 digunakan sebagai 16-bit *autoreload counter (mode 0)* untuk menghitung pulsa keluaran *encoder* yang menyatakan posisi pergerakan setiap *joint* robot. *Timer2 register T2CON* diisi nilai tertentu untuk mengaktifkan *timer2 external enable input T2EX*. Keluaran *phase A encoder* dihubungkan dengan pin P1.0 (T2) *AT89S52* sebagai *external count input timer2*, sedangkan keluaran *d-flip-flop* sebagai detektor arah putaran *(Q)* dihubungkan dengan pin P1.1 (T2EX) sebagai *timer/counter2 direction control*. Jadi, pulsa keluaran *phase A* akan dihitung naik atau turun tergantung kondisi keluaran *d-flip-flop (Q)* sebagai detektor arah pergerakan motor. Saat pulsa *phase A encoder* dalam kondisi transisi *high to low* dan keluaran *d-flip-flop (Q)* dalam kondisi *high*, maka *counter*

akan menghitung naik. Artinya, motor berputar searah jarum jam. Sebaliknya, apabila dalam kondisi yang sama, keluaran *d-flip-flop (Q)* dalam kondisi *low*, maka *counter* akan menghitung mundur atau motor berputar berlawanan arah jarum jam. *Overflow* tidak digunakan (tidak pernah terjadi) karena *encoder* yang digunakan hanya memiliki maksimum resolusi 200 pulsa per putaran, sedangkan *timer/counter2* adalah *timer/counter* 16-bit.

Data pergerakan berupa nilai pembacaan *encoder* disimpan pada *array* variabel data posisi per step masing-masing *microcontroller* dan secara bergiliran disimpan pada serial memori menggunakan jalur *i²c bus* setiap kali tombol *keypad* simpan data posisi ditekan. Nilai pembacaan *encoder* setiap *joint* juga dikirim secara bergiliran secara serial untuk ditampilkan pada layar monitor melalui *hyperterminal*.

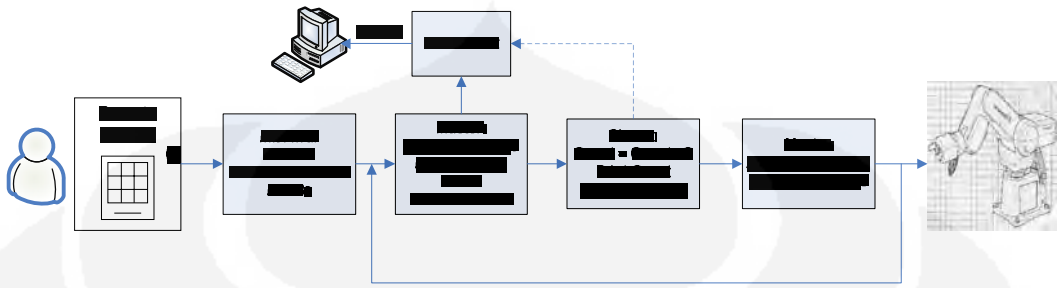


Gambar 3.13 *Timer2 auto-reload mode*
(Sumber: *Datasheet AT89S52*, Atmel)

3.6 PENGIRIMAN DATA PERGERAKAN ROBOT

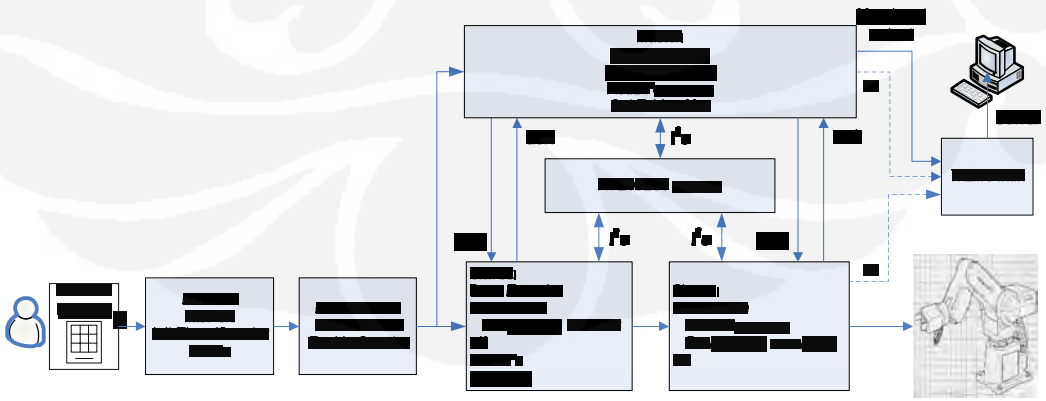
Data pergerakan berupa nilai pembacaan *encoder* sebagai sensor posisi pergerakan setiap *joint* dikirim secara serial ke sebuah komputer menggunakan jalur *serial Tx microcontroller* masing-masing *joint*. Jalur *serial Tx* ini digunakan secara bergantian oleh beberapa *microcontroller* (*master* dan *slave*). *Microcontroller* kendali pusat bertanggung jawab mengatur penggunaan jalur

serial Tx ini menggunakan *multiplexer 8x1* dengan mengatur nilai *chip select input IC Mux 74HC151*.

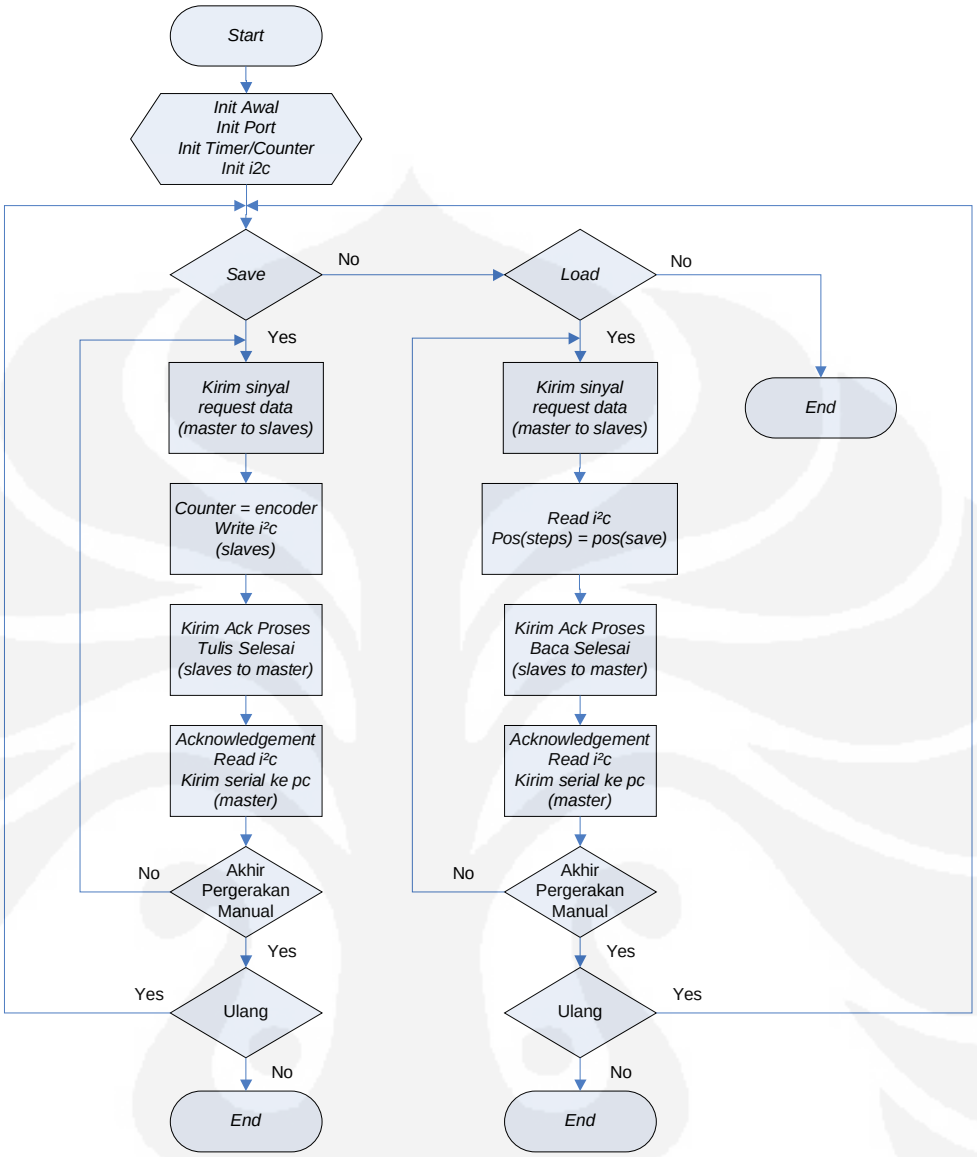


Gambar 3.14 Blok fungsional pengaturan jalur *serial Tx*

Data pembacaan *encoder* ini juga disimpan secara berkala pada serial memori setiap kali tombol *keypad* simpan data posisi ditekan. Akses jalur *i²c bus* untuk proses baca/tulis serial memori ini diatur melalui proses *handshaking* antar *micorcontroller*. Sinyal *request* data posisi pergerakan dikirim oleh *microcontroller* pusat. *Slave microcontrollers* kemudian melakukan proses tulis data posisi melalui *i^2c EEPROM* dan mengirimkan sinyal balik berupa *acknowledgement* saat proses tulis selesai. Sinyal tersebut diterima *microcontroller* pusat untuk melakukan proses baca data posisi. Kemudian, *microcontroller* pusat akan mengirimkan hasil penyimpanan data berupa pembacaan *encoder* tersebut secara serial untuk ditampilkan melalui *hyperterminal*, sekaligus menjadi verifikasi data yang tersimpan dalam memori.



Gambar 3.15 Blok fungsional *handshaking* proses baca/tulis data

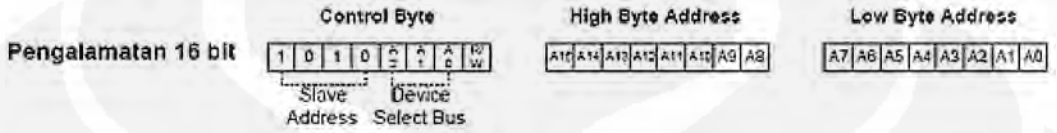


Gambar 3.16 Diagram alir proses baca/tulis data pergerakan robot

3.6.1 *I²C Serial EEPROM AT24C256*

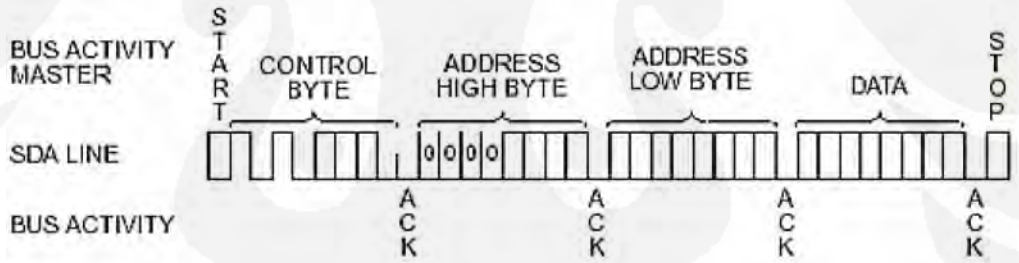
Untuk keperluan penyimpanan data digunakan *serial EEPROM* tipe *24xx* yang merupakan memori serial *i²c (inter integrated circuit). Serial EEPROM* yang digunakan adalah *AT24C256* produksi *Atmel* yang memiliki kapasitas *32Kbyte (32,768 x 8)*. Memori serial ini digunakan bersama oleh *microcontroller* untuk keperluan tulis dan baca data pergerakan setiap *joint* robot (*slave* → tulis dan *master* → baca).

Dua buah jalur *(bus)* digunakan untuk komunikasi antar *microcontroller* melalui memori serial ini yaitu *SDA (serial data)* dan *SCL (serial clock)*. *SDA* merupakan jalur data pada komunikasi *i²c* sedangkan *SCL* merupakan jalur *clock* . Sinyal *clock* ini selalu muncul setiap transfer *bit* data. Adapun karena kapasitasnya lebih dari 4 *Kbyte*, memori serial ini menggunakan metode pengalamatan 16 *bit*.

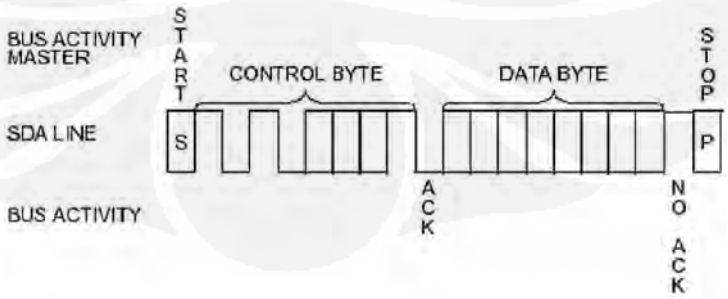


Gambar 3.17 Blok diagram pengalamatan *16-bit*
(Sumber: diolah kembali dari *datasheet AT24C256,* Atmel)

Penulisan data dilakukan per *byte* dengan mengirimkan *control byte*, alamat tujuan dan data yang akan ditulis. Sedangkan pembacaan data dilakukan dengan membaca alamat sekarang yang ingin diakses *(current read address).*



Gambar 3.18 Siklus pengiriman data *(per byte)*
(Sumber: *Datasheet AT24C256,* Atmel)



Gambar 3.19 Siklus pembacaan data *(current read address)*
(Sumber: *datasheet AT24C256, Atmel*)

3.6.2 Akses dan Pembagian Alamat Memori

Memori serial *AT24C256* memiliki kapasitas *32Kbyte (32,768 x 8)* yang dapat diakses oleh *microcontroller* masing-masing *joint* untuk keperluan baca dan tulis data. Namun, waktu akses dan pengalamatan memori ini harus diatur untuk menghindari kehilangan data saat proses baca/tulis dan menumpuknya data pada lokasi yang sama. Berikut adalah pembagian akses lokasi serial memori untuk tiap pergerakan *joint* robot:

Tabel 3.3 Pembagian lokasi memori serial *AT24C256*

24C256 — 32,768 x 8 (256K)

Blok Alamat	*Word* Alamat		Lokasi Memori		Keterangan
	Awal	Akhir	Awal	Akhir	
Joint #1	0000	0FFF	0	4095	sementara tidak digunakan karena *encoder* tidak berfungsi
Joint #2	1001	1FFF	4096	8191	-
Joint #3	2001	2FFF	8192	12287	-
Joint #4	3001	3FFF	12288	16383	-
Joint #5	4001	4FFF	16384	20479	-
Gripper	5000	5FFF	20480	24575	sementara tidak digunakan *(feedback sensor n/a)*
n/a	6000	6FFF	24576	28671	*Spare*
n/a	7000	7FFF	28672	32767	*Spare*

1,023 lokasi memori per *joint*

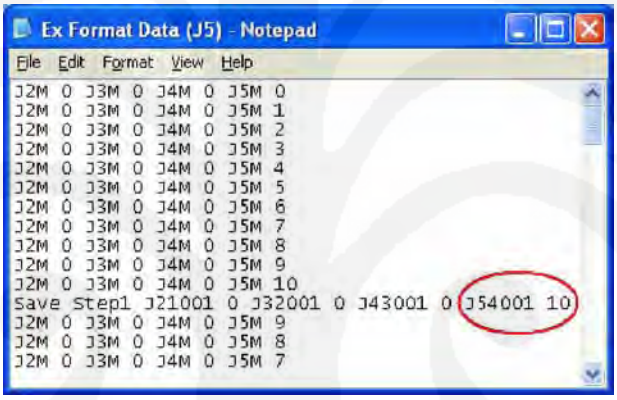
Dari tabel di atas, setiap *joint* memperoleh 1023 lokasi alamat memori untuk menyimpan data pergerakannya. Masing-masing lokasi alamat tersebut dapat diisi data berukuran 8 *bit*. Karena tipe data *integer* digunakan untuk menyatakan nilai pembacaan *encoder*, maka digunakan dua buah alamat untuk menyimpan setiap nilai tersebut. Pada implementasi program, nilai pembacaan *encoder (integer)* dibagi menjadi dua bagian yaitu *upper byte* dan *lower byte*. Kemudian, keduanya disimpan pada dua alamat berbeda namun saling berurutan dalam *i²c eeprom*. Saat dibaca, isi kedua alamat memori kembali dijadikan bentuk *integer* seperti semula oleh program.

BAB IV
ANALISIS DAN PENGUJIAN PROGRAM

Pengujian dan analisis sistem dibagi menjadi dua bagian, yaitu:

- Sistem Pergerakan Manual
- Sistem Pergerakan Otomatis (metode *leadthrough – teaching mode*)

Data posisi pergerakan berupa nilai pembacaan *rotary encoder* pada setiap *joint* dapat disimpan dalam memori pengendali dan memori serial. Adapun format data pergerakan setiap *joint* robot yang digunakan dalam pengujian program adalah sebagai berikut:



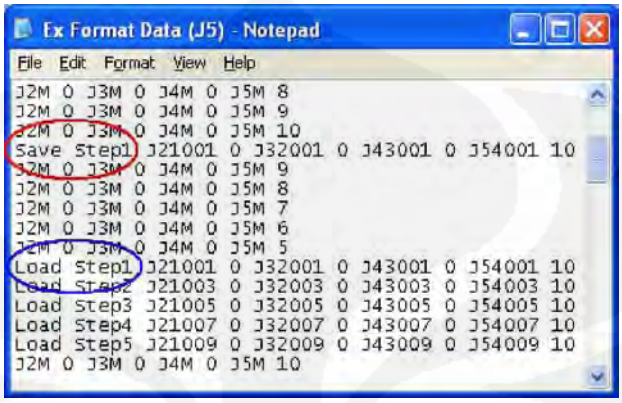
Gambar 4.1 Format data posisi pergerakan

Format umum data pergerakan tiap *joint* robot terdiri dari enam digit *header* yang menandakan nomor *joint* robot dan alamat penyimpanan data pada memori serial. Kemudian dilanjutkan dengan 1-3 digit data nilai pembacaan *encoder*. Gambar 4.1 adalah contoh format data pergerakan manual *joint* 5 (dilingkari), sedangkan ilustrasi penjelasannya dapat dilihat pada gambar 4.2 di bawah ini:

Gambar 4.2 Penjelasan format data pergerakan

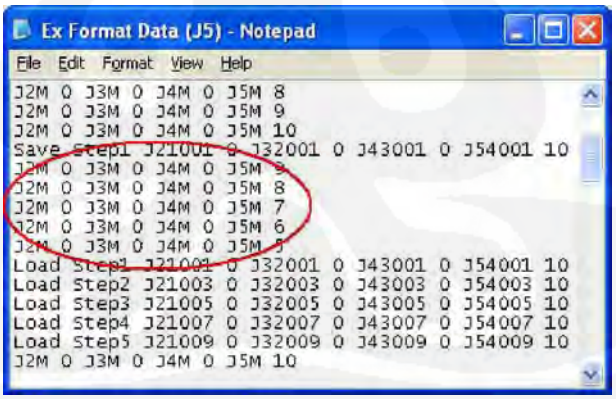
Format lain untuk proses simpan data *step* posisi pergerakan dan *load* data *step* pergerakan hampir sama dengan format sebelumnya. Perbedaannya hanya terletak

pada *header* awal setiap baris data, yaitu *"Save Step"X* untuk simpan data posisi pergerakan dan *"Load Step"X* untuk *load* data posisi pergerakan (*"X"* mengidentifikasikan nomor *step*). Berikut adalah ilustrasinya:



Gambar 4.3 Format *save data* (merah) dan *load* data pergerakan (biru)

Data nilai pembacaan *encoder* secara *online* (lintasan pergerakan lengan robot) dikirim secara serial untuk ditampilkan melalui hyperterminal dengan format yang sama namun tanpa empat digit alamat memori di depannya. Di belakang nomor joint ditambahkan karakter "M" untuk jenis pergerakan manual atau "A" untuk pergerakan otomatis (gambar 4.4).

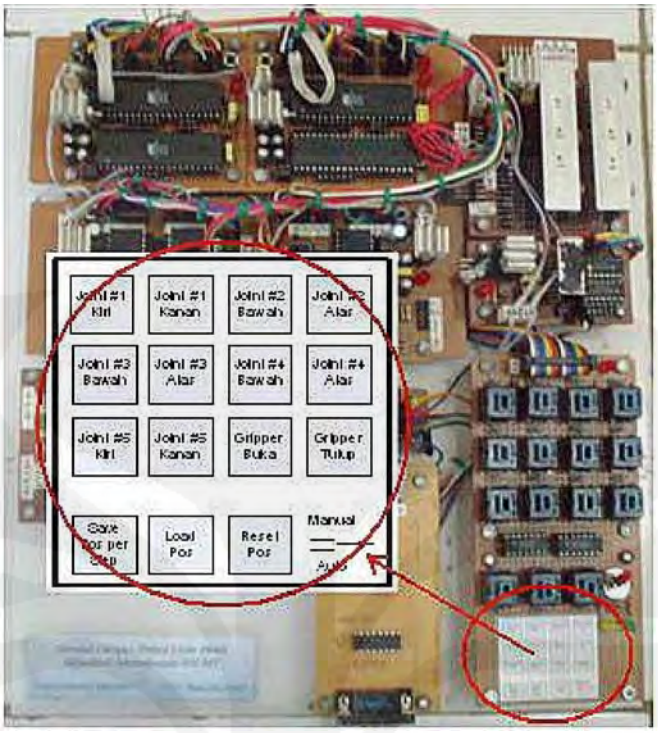


Gambar 4.4 Format data nilai pembacaan *encoder (online)*

4.1 PENGUJIAN DAN ANALISIS PERGERAKAN MANUAL

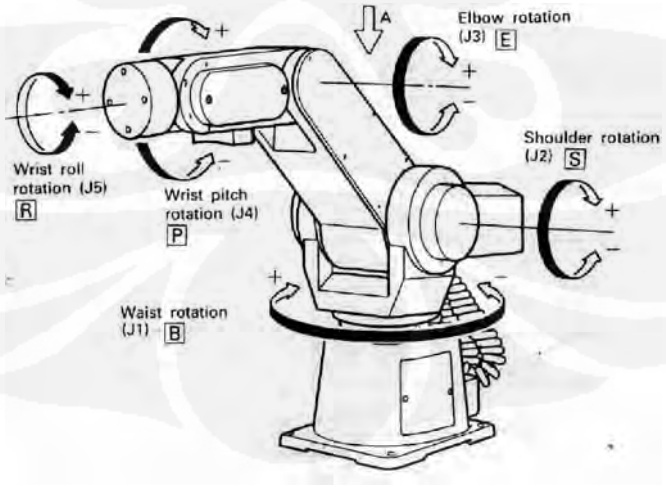
4.1.1 Sistem Pergerakan Manual

Pergerakan manual robot dioperasikan menggunakan tombol-tombol *keypad control*. Setiap *joint* dikendalikan oleh sebuah *microcontroller* terpisah dengan tujuan agar dapat digerakkan secara simultan/bersamaan.



Gambar 4.5 *Keypad* kendali pergerakan robot

Beberapa *step* posisi pergerakan robot disimpan dalam sebuah *array* variabel data posisi sebagai nilai *set point* yang selanjutnya digunakan pada mode pergerakan robot secara otomatis. Data tersebut berupa nilai pembacaan *encoder* pada masing-masing *joint* saat tombol simpan data posisi ditekan. Berikut adalah ilustrasi pergerakan setiap aksis robot lengan *Mitsubishi Movemaster RV-M1*.

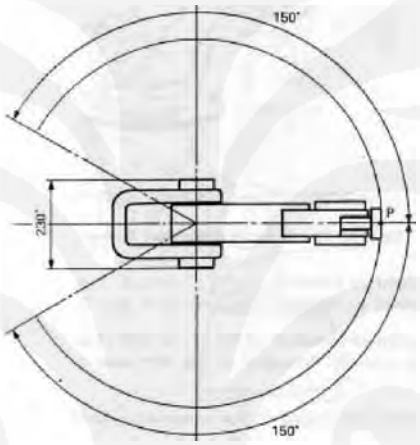


Gambar 4.6 Ilustrasi pergerakan setiap *joint* lengan robot
(Sumber: Buku manual *Mitsubishi Movemaster RV-M1*)

4.1.2 Pergerakan *Waist (Joint 1)*

Jangkauan pergerakan bagian pinggang robot *(joint 1)* adalah sejauh 300°. Jangkauan pergerakan *joint 1* dibatasi oleh *proximity sensor* berupa *limit switch* pada salah satu ujung akhir pergerakannya.

Kendali posisi pergerakan motor pada *joint* ini bersifat *open loop* karena sensor posisi pergerakan berupa *rotary encoder* yang terhubung pada poros motor penggerak *joint* ini tidak berfungsi. Kendali posisi hanya diperoleh dari *proximity sensor* berupa *limit switch* sebagai batas jangkauan pergerakan



Gambar 4.7 Jangkauan pergerakan *joint 1* (tampak atas)
(Sumber: Buku manual *Mitsubishi Movemaster RV-M1*)

.Berikut adalah tabel kendali pergerakan manual *joint 1* berdasarkan kondisi tegangan logika *TTL* yang diberikan pada *keypad push-button (Pb)* dan kondisi sensor *limit switch* (logika ”1” adalah kondisi *high* 5 V, sedangkan logika ”0” adalah kondisi *low* 0 V).

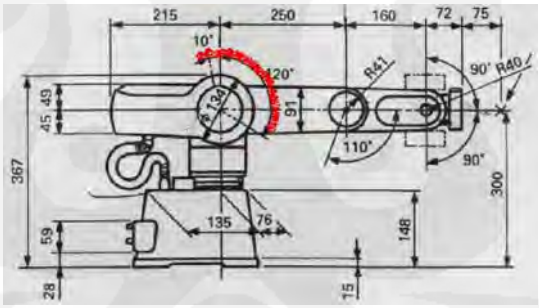
Tabel 4.1 Kendali pergerakan manual *joint 1*

Pb J1 Kiri	*Pb J1* Kanan	*Limit Switch*	Keterangan
0	0	0	*J1 off*
1	0	0	*J1* putar kiri
0	1	0	*J1* putar kanan
1	1	0	*J1 stop*
0	0	1	*J1 off*
1	0	1	*J1 stop* putar kiri
0	1	1	*J1* putar kanan
1	1	1	*J1 stop*

Apabila sensor *limit switch* dalam kondisi tertekan (aktif → logika 1), maka *joint 1* tidak dapat bergerak lebih jauh lagi ke arah kiri meskipun tombol *keypad "Pb J1 Kiri"* ditekan. *Joint 1* dapat digerakkan ke arah kiri kembali apabila kondisi *limit switch* tidak tertekan. Sebagai pengaman kendali pergerakan, pada saat kondisi dua tombol *"Pb J1 Kiri"* dan *"Pb J1 Kanan"* ditekan secara bersamaan, maka *joint 1* akan berhenti bergerak *(J1 stop)*.

4.1.3 Pergerakan *Shoulder (Joint 2)*

Jangkauan pergerakan bagian bahu robot *(joint 2)* adalah sejauh 130°. Jangkauan pergerakan *joint* ini juga dibatasi oleh sebuah *proximity sensor* berupa *limit switch* pada salah satu ujung akhir pergerakannya. Resolusi *encoder* yang digunakan pada *joint* ini adalah 200 pulsa per putaran *(200 ppr – pulse per rotation)* atau 1.8° per pulsanya. Pergerakan manual pada *joint 2* dapat dianalisis menggunakan data lintasan pergerakan manual berupa nilai pembacaan *encoder* yang secara berkala dikirim secara serial melalui *hyperterminal*. Berikut adalah gambar dan ilustrasi jangkauan pergerakan bahu lengan robot *(joint 2):*



Gambar 4.8 Jangkauan pergerakan *joint 2*

(Sumber: diolah kembali dari buku manual *Mitsubishi Movemaster RV-M1*)

Tabel 4.2 Kendali pergerakan manual *joint 2*

Pb J2 Bawah	*Pb J2* Atas	*Limit Switch*	Keterangan
0	0	0	J2 *off*
1	0	0	J2 putar bawah
0	1	0	J2 putar atas
1	1	0	J2 stop
0	0	1	J2 *off*
1	0	1	J2 putar bawah
0	1	1	J2 stop putar atas
1	1	1	J2 stop

Pengujian dilakukan untuk membandingkan nilai pembacaan *encoder* pada pergerakan *joint* menuju jangkauan maksimumnya terhadap hasil perhitungan. Pergerakan *joint 2* dimulai dari posisi vertikal +10° (kondisi *limit switch* tertekan) menuju posisi horisontal pada jangkauan maksimumnya yaitu -120° (total pergerakan 130°). Perhitungan nilai pembacaan *encoder* berdasarkan gerak rotasi yang terjadi dapat diperoleh dengan persamaan berikut:

$$Rotasi\ (°) = \frac{Pulsa\ encoder\ yang\ terjadi}{Jumlah\ pulsa\ per\ rotasi} \times 360° \quad (4.1)$$

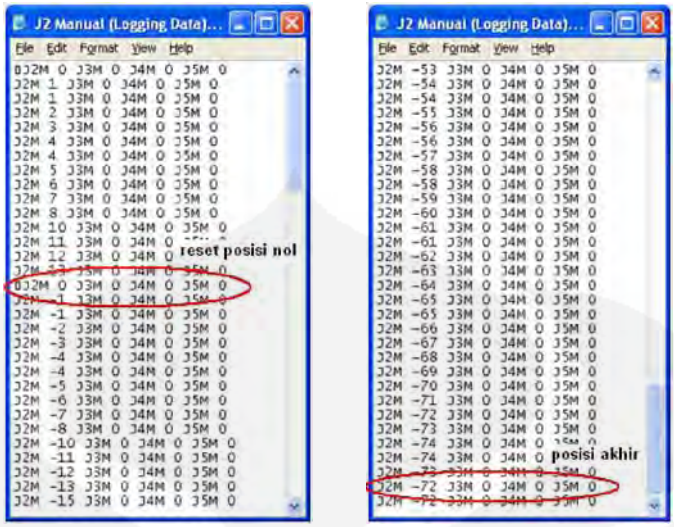
Menggunakan persamaan 4.1 di atas, maka dengan resolusi encoder yang digunakan sebesar 200 *ppr*, maka pulsa *encoder* yang terjadi untuk gerak rotasi sejauh 130° adalah 72 pulsa.

$$Rotasi\ (°) = \frac{Pulsa\ encoder\ yang\ terjadi}{Jumlah\ pulsa\ per\ rotasi} \times 360°$$

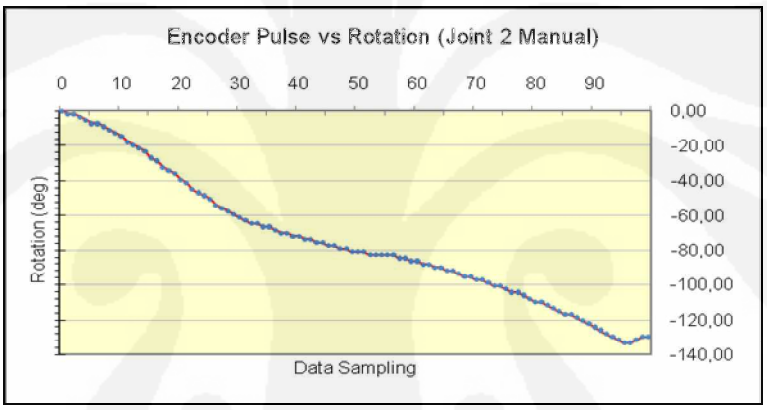
$$Pulsa\ encoder = \frac{130° \times 200\ ppr}{360} = 72{,}22\ pulsa\ (\text{kondisi ideal})$$

Pada saat pengujian, *joint 2* digerakkan terlebih dahulu menuju titik awalnya (posisi vertikal) hingga *proximity sensor* berupa *limit switch* dalam kondisi tertekan. Kemudian, *microcontroller* di-*reset* sehingga pada posisi tersebut nilai pembacaan *encoder* adalah nol. Selanjutnya, lengan robot *joint 2* digerakkan menuju jangkauan maksimumnya sejauh -130° dan pergerakannya dimonitor melalui *hyperterminal* yang menampilkan nilai pembacaan *encoder* secara *online*. Pergerakan manual *joint 2* kemudian dihentikan saat nilai pembacaan *encoder* melalui *hyperterminal* menunjukkan nilai 72. Berikut adalah ilustrasi pergerakan manual *joint 2* dan hasil pembacaan *encoder* melalui *hyperterminal*:

Gambar 4.9 Pergerakan manual *joint 2* hasil pengujian



Gambar 4.10 Data pergerakan manual *joint 2*



Gambar 4.11 Kurva plot pergerakan manual *joint 2*

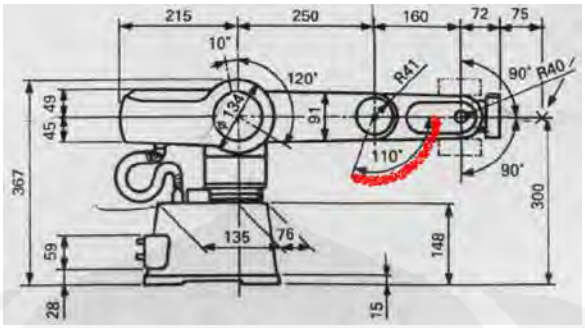
Secara perhitungan, maka dengan 72 pulsa *encoder* yang terjadi diperoleh pergerakan sejauh 129,6°. Kondisi posisi tepat 130° sulit diperoleh karena ketelitian *encoder* yang digunakan adalah 200 *ppr* (1.8° per pulsa).

$$Rotasi\ (^\circ) = \frac{Pulsa\ encoder\ yang\ terjadi}{Jumlah\ pulsa\ per\ rotasi} \times 360^\circ$$

$$Rotasi\ (^\circ) = \frac{72}{200} \times 360^\circ = 129{,}6^\circ$$

4.1.4 Pergerakan *Elbow (Joint 3)*

Jangkauan pergerakan bagian siku lengan robot *(joint 3)* adalah sebesar 110°. Jangkauan pergerakan *joint* ini juga dibatasi oleh sebuah *proximity sensor* berupa *limit switch* pada salah satu ujung akhir pergerakannya. Resolusi *encoder* yang digunakan adalah 200 *ppr* (1.8° per pulsa).



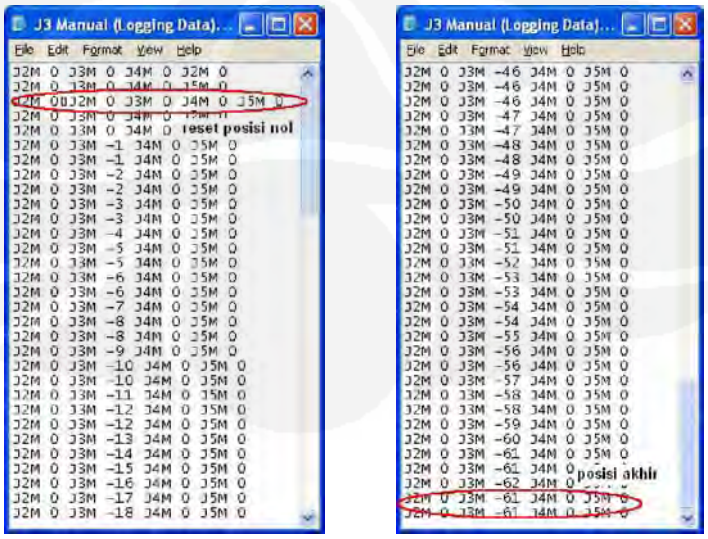
Gambar 4.12 Jangkauan pergerakan *joint 3*

(Sumber: diolah kembali dari buku manual *Mitsubishi Movemaster RV-M1*)

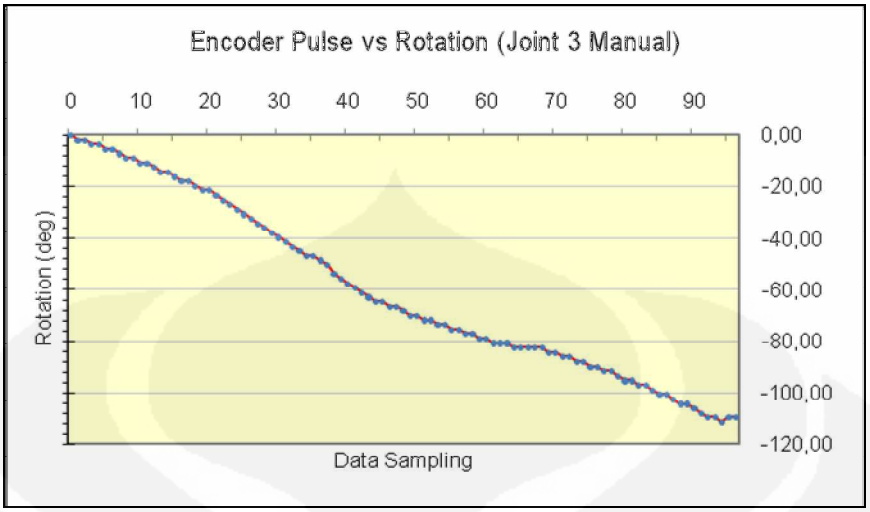
Tabel 4.3 Kendali pergerakan manual *joint 3*

Pb J3 Bawah	*Pb J3* Atas	*Limit Switch*	Keterangan
0	0	0	J3 *off*
1	0	0	J3 putar bawah
0	1	0	J3 putar atas
1	1	0	J3 stop
0	0	1	J3 *off*
1	0	1	J3 putar bawah
0	1	1	J3 stop putar atas
1	1	1	J3 stop

Prosedur pengujian pergerakan manual *joint* ini adalah sama dengan *joint 2* sebelumnya. Rentang pergerakan *joint* 3 sejauh 110° dimulai dari posisi vertikal 0° menuju posisi horisontal -110°. Berikut adalah hasil nilai pembacaan *encoder* berdasarkan pergerakan manual *joint 3:*



Gambar 4.13 Data pergerakan manual *joint 3*



Gambar 4.14 Kurva plot pergerakan manual *joint 3*

Menggunakan persamaan 4.1, dengan resolusi *encoder* 200 *ppr*, maka pulsa *encoder* yang terjadi untuk gerak rotasi sejauh 110° adalah sebagai berikut:

$$Rotasi\ (°) = \frac{Pulsa\ encoder\ yang\ terjadi}{Jumlah\ pulsa\ per\ rotasi} \times 360°$$

$$Pulsa\ encoder = \frac{110° \times 200\ ppr}{360} = 61{,}11\ pulsa\ (\text{kondisi ideal})$$

Dengan mengambil pembulatan nilai 61 pulsa *encoder* pada saat pengujian, maka akan diperoleh pergerakan sejauh 109,8°.

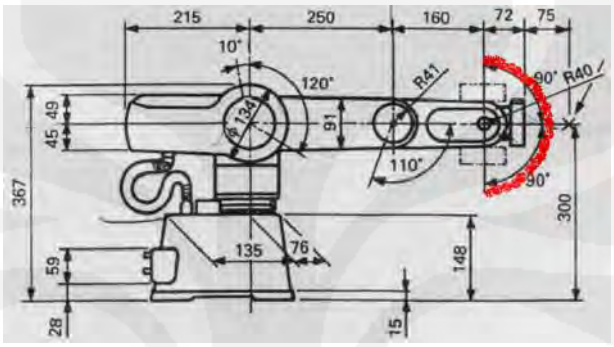
$$Rotasi\ (°) = \frac{61}{200} \times 360° = 109{,}8°$$

Dengan resolusi yang sama dengan *joint* sebelumnya (200 *ppr*), maka kondisi posisi tepat 110° akan sulit diperoleh karena ketelitian *encoder* yang digunakan adalah 1.8° per pulsanya.

Gambar 4.15 Pergerakan manual *joint 3* hasil pengujian

4.1.5 Pergerakan *Wrist Pitch (Joint 4)*

Pergerakan naik/turun pergelangan tangan robot *(joint 4)* memiliki jangkauan sebesar ±90°. Jangkauan pergerakan *joint* ini juga dibatasi oleh sebuah *proximity sensor* berupa *limit switch* pada salah satu ujung akhir pergerakannya. Sensor posisi pergerakan yang digunakan pada *joint* ini berupa *rotary encoder* dengan resolusi 96 pulsa per putaran (3.75° per pulsa).



Gambar 4.16 Jangkauan pergerakan *joint 4*

(Sumber: diolah kembali dari buku manual *Mitsubishi Movemaster RV-M1*)

Tabel 4.4 Kendali pergerakan manual *joint 4*

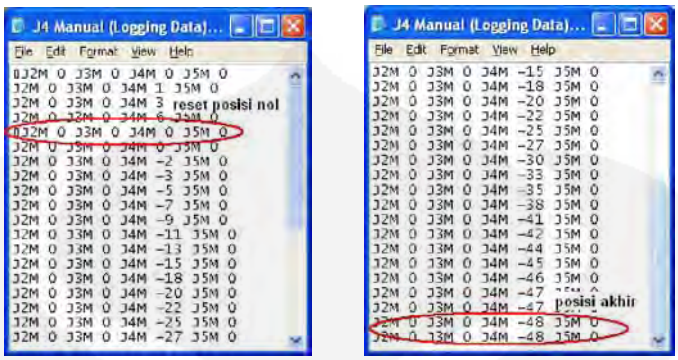
Pb J4 Bawah	*Pb J4* Atas	*Limit Switch*	Keterangan
0	0	0	J4 *off*
1	0	0	J4 putar bawah
0	1	0	J4 putar atas
1	1	0	J4 stop
0	0	1	J4 *off*
1	0	1	J4 stop putar bawah
0	1	1	J4 putar atas
1	1	1	J4 stop

Pengujian pergerakan manual *joint* 4 dilakukan untuk rentang jangkauan maksimumnya sejauh ±90° (180°). Pergerakan manual *joint* 4 dimulai dari posisi +90° menuju posisi -90°. Hasil perhitungan (menggunaan persamaan 4.1) menunjukkan bahwa dengan resolusi *encoder* sebesar 96 *ppr*, maka jumlah pulsa *encoder* yang terjadi untuk rentang pergerakan rotasi sejauh 180° adalah 48 pulsa.

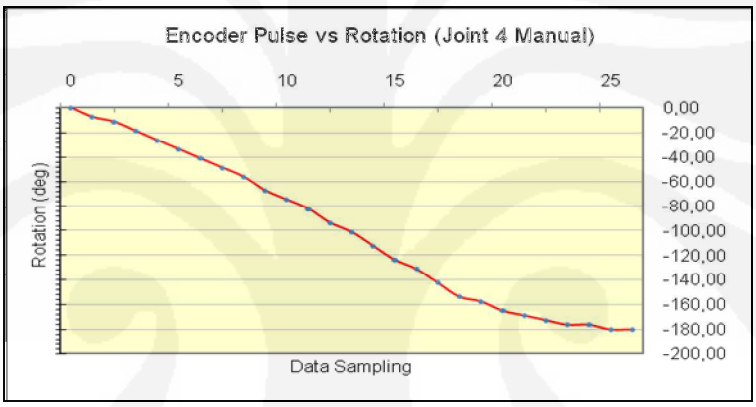
$$Rotasi\ (°) = \frac{Pulsa\ encoder\ yang\ terjadi}{Jumlah\ pulsa\ per\ rotasi} \times 360°$$

$$Pulsa\ encoder = \frac{180° \times 96\ ppr}{360} = 48\ pulsa$$

Gambar 4.17 merupakan data nilai pembacaan *encoder* pada pengujian pergerakan manual *joint 4* dengan rentang pergerakan sejauh 180°:



Gambar 4.17 Data pergerakan manual *joint 4*



Gambar 4.18 Kurva plot pergerakan manual *joint 4*

Gambar 4.19 Pergerakan manual *joint 4* hasil pengujian

4.1.6 Pergerakan *Wrist Roll (Joint 5)*

Pergerakan putar pergelangan tangan robot *(joint 5)* memiliki jangkauan sebesar ±180°. Jangkauan pergerakan *joint* ini juga dibatasi oleh *proximity sensor*

berupa *limit switch*. Sama halnya dengan *joint 4*, sensor posisi pergerakan yang digunakan pada *joint* ini berupa *rotary encoder* dengan resolusi 96 pulsa per putaran (3.75° per pulsa).

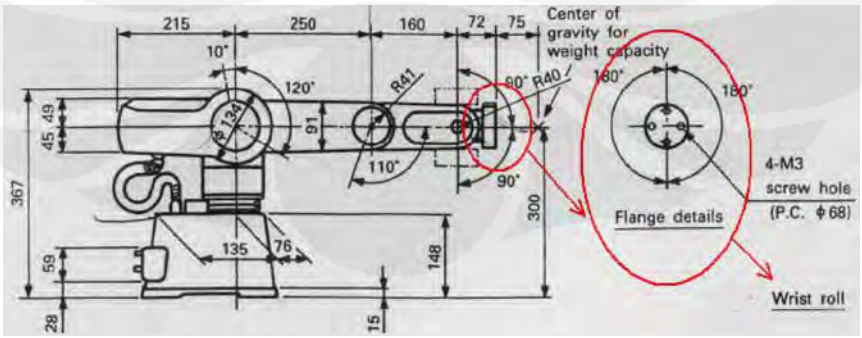
Tabel 4.5 Kendali pergerakan manual *joint 5*

Pb J5 Kiri	*Pb J5* Kanan	*Limit Switch*	Keterangan
0	0	0	J5 *off*
1	0	0	J5 putar kiri
0	1	0	J5 putar kanan
1	1	0	J5 stop
0	0	1	J5 *off*
1	0	1	J5 stop putar kiri
0	1	1	J5 stop putar kanan
1	1	1	J5 stop

Jangkauan pergerakan pada *joint 5* ini adalah ±180°. Artinya, *joint* ini dapat bergerak berputar 180° searah atau berlawanan arah jarum jam. Pengujian pergerakan manual dilakukan dengan memutar joint *ini* mencapai jangkauan maksimumnya sejauh ±180°. Pergerakan manual *joint* 5 dimulai dari posisi 0° menuju posisi -180°. Menurut perhitungan menggunakan persamaan 4.1, jumlah pulsa *encoder* yang terjadi untuk mencapai titik tersebut adalah 48 pulsa (resolusi *encoder* 96 *ppr*). Kemudian, *joint* diputar dari posisi 0° menuju arah +180° dengan jumlah pulsa *encoder* yang sama.

$$Rotasi\ (°) = \frac{Pulsa\ encoder\ yang\ terjadi}{Jumlah\ pulsa\ per\ rotasi} \times 360°$$

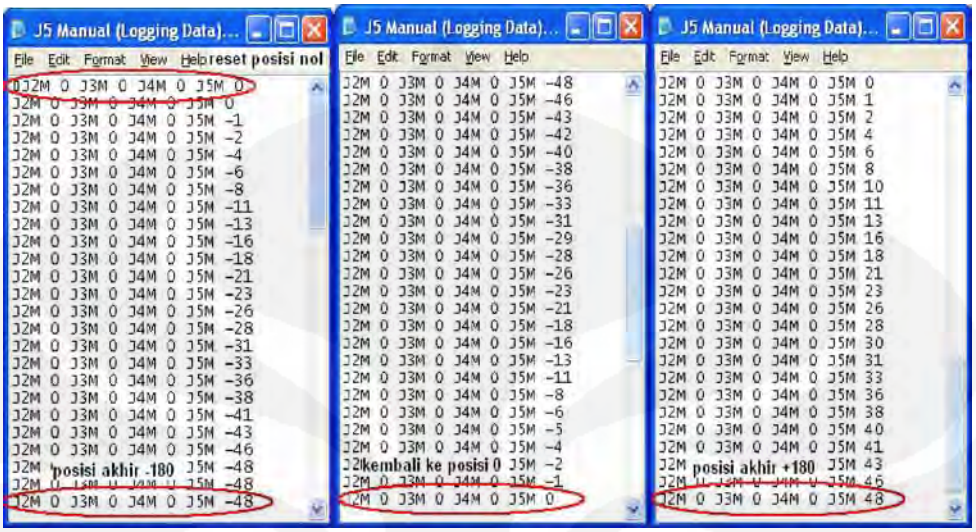
$$Pulsa\ encoder = \frac{180° \times 96\ ppr}{360} = 48\ pulsa$$



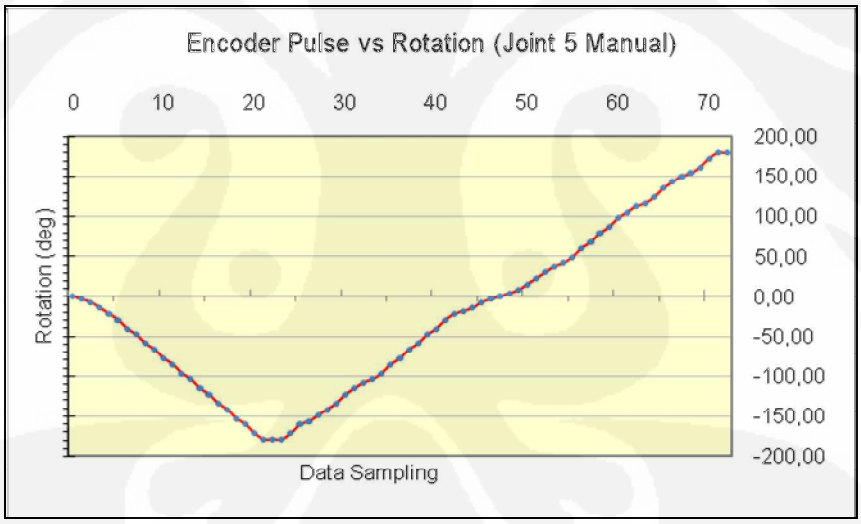
Gambar 4.20 Jangkauan pergerakan *joint 5*

. (Sumber: diolah kembali dari buku manual *Mitsubishi Movemaster RV-M1*)

Berikut adalah data hasil pembacaan *encoder* berdasarkan pengujian yang dilakukan:



Gambar 4.21 Data pergerakan manual *joint 5*



Gambar 4.22 Kurva plot pergerakan manual *joint 5*

Gambar 4.23 Pergerakan manual *joint 5* hasil pengujian

4.2 PENGUJIAN DAN ANALISIS SISTEM PERGERAKAN OTOMATIS

4.2.1 Sistem Pergerakan Otomatis

Pergerakan otomatis robot dilakukan melalui proses pembelajaran dimana manipulator digerakkan atau dipindahkan terlebih dahulu secara manual melalui lintasan pergerakan tertentu *(metode leadthrough – teaching by showing).* Data lintasan pergerakan berupa pembacaan *encoder* beberapa *step* pergerakan manual yang disimpan dalam memori pengendali. Selanjutnya, pergerakan otomatis robot akan mengikuti lintasan pergerakan yang telah disimpan tersebut. Pada implementasinya, program pergerakan lengan robot akan terus-menerus membandingkan hasil pembacaan *encoder* dengan nilai variabel dalam *array* data posisi yang telah disimpan sebelumnya sebagai nilai *set point* yang harus dicapai.

4.2.2 Pergerakan Otomatis Satu *Joint*

Pengujian pergerakan otomatis satu *joint* dilakukan pada *joint 4 (wrist pitch)* untuk pergerakan naik/turun pergelangan tangan robot. Pengujian dilakukan dengan memasukkan tiga *step* pergerakan (maksimum lima *step*). *Step* pertama adalah posisi awal 0° (vertikal), kemudian dilanjutkan pergerakan *step* ke dua menuju posisi -60° dan *step* terakhir pergerakan menuju -30°. Setiap *step* posisi pergerakan adalah berdasarkan nilai pembacaan *encoder* pada *joint* ini. Berikut adalah tabel yang menunjukkan nilai pembacaan *encoder* yang harus dicapai untuk melakukan pengujian ini:

$$Rotasi\ (^\circ) = \frac{Pulsa\ encoder\ yang\ terjadi}{Jumlah\ pulsa\ per\ rotasi} \times 360^\circ$$

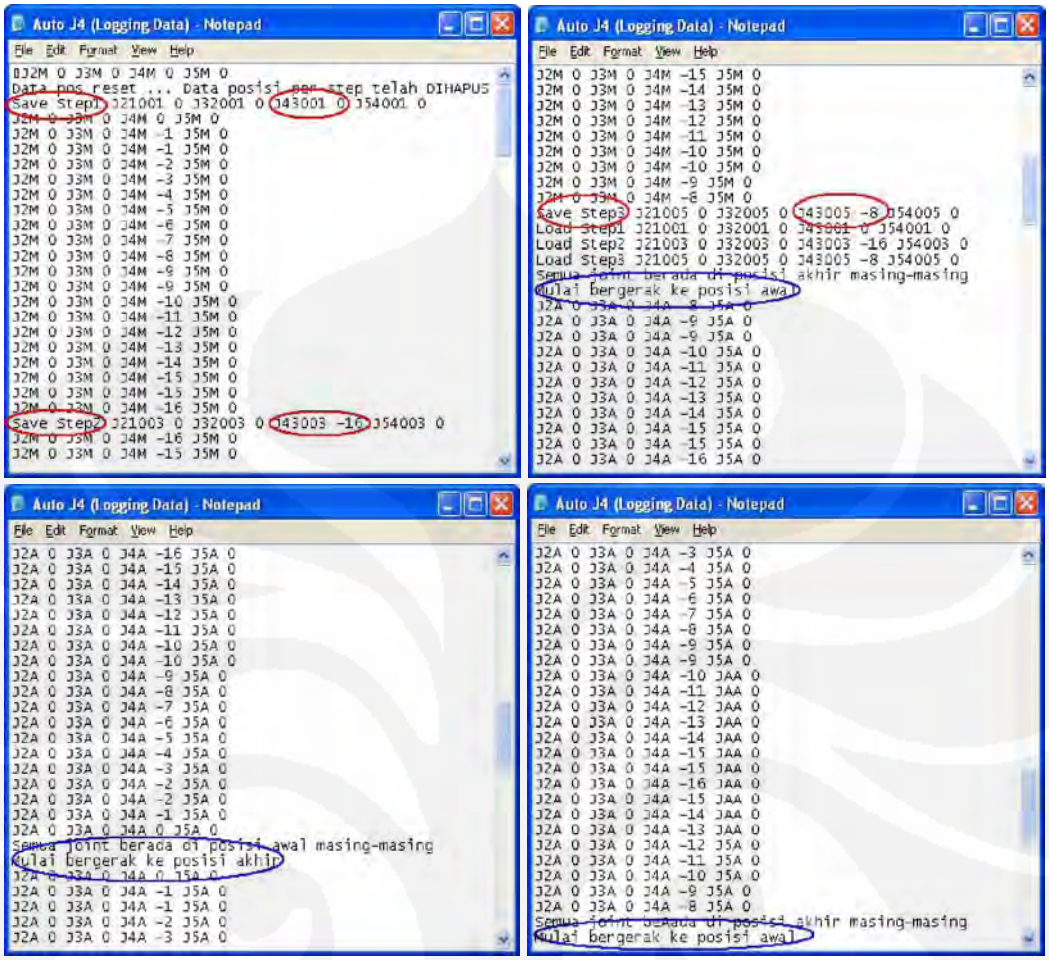
$$Pulsa\ encoder = \frac{60^\circ \times 96\ ppr}{360} = 16\ pulsa\ (0^\circ \rightarrow -60^\circ)$$

$$Pulsa\ encoder = \frac{30^\circ \times 96\ ppr}{360} = 8\ pulsa\ (-60^\circ \rightarrow -30^\circ)$$

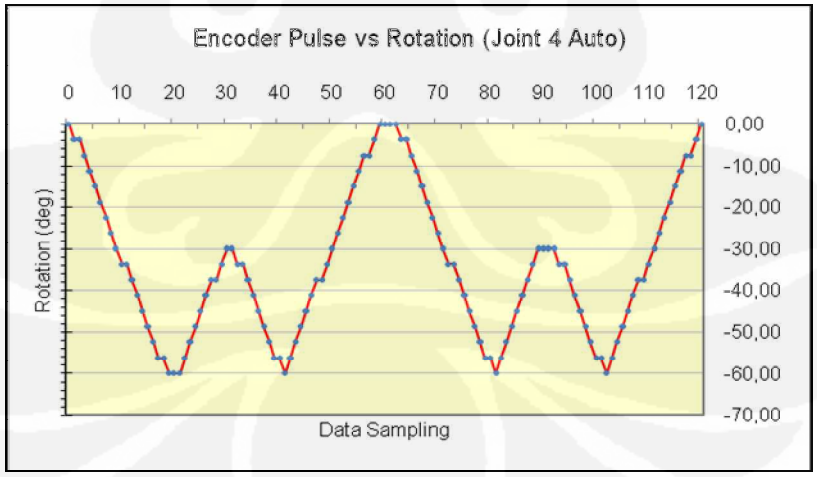
Tabel 4.6 Pengujian pergerakan otomatis satu *joint (joint 4)*

Langkah pergerakan	Posisi sudut	*Counter* pulsa *encoder*	Keterangan
Step 1	0° (posisi awal)	0	Resolusi 96 *ppr*
Step 2	0° menuju -60°	-16	
Step 3	-60° menuju -30°	-8	

Data hasil pengujian pergerakan otomatis satu *joint* ini dapat diamati pada gambar di bawah ini:



Gambar 4.24 Data pergerakan otomatis satu *joint (joint 4)*



Gambar 4.25 Kurva plot pergerakan otomatis satu *joint (joint 4)*

4.2.3 Pergerakan Otomatis Simultan Dua *Joint*

Pengujian pergerakan otomatis dua *joint* dilakukan *joint 4 (wrist pitch)* dan *joint 5 (wrist roll)* untuk pergerakan naik/turun dan pergerakan putar pergelangan

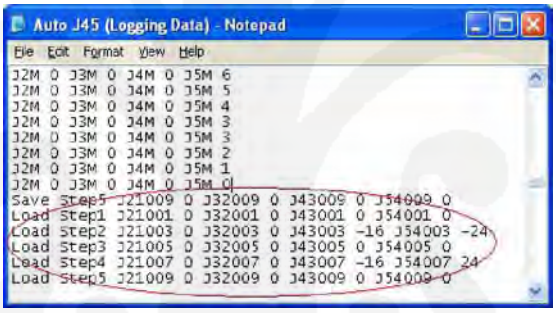
tangan robot. Pengujian dilakukan dengan memasukkan lima *step* pergerakan (maksimum) yang disimpan saat pergerakan manual. Berikut adalah tabel yang menunjukkan nilai pembacaan *encoder* yang harus dicapai pada pengujian ini:

$$Pulsa\ encoder = \frac{60° \times 96\ ppr}{360} = 16\ pulsa\ (0° \rightarrow -60°)$$

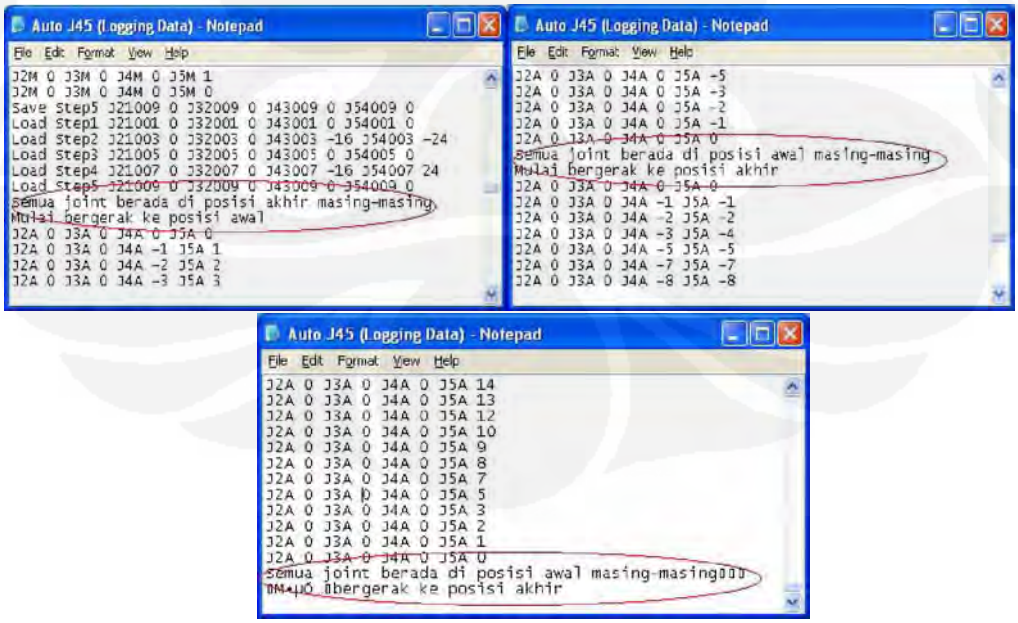
$$Pulsa\ encoder = \frac{90° \times 96\ ppr}{360} = 24\ pulsa\ (0° \rightarrow -90°)$$

Tabel 4.7 Pengujian pergerakan otomatis simultan dua *joint*

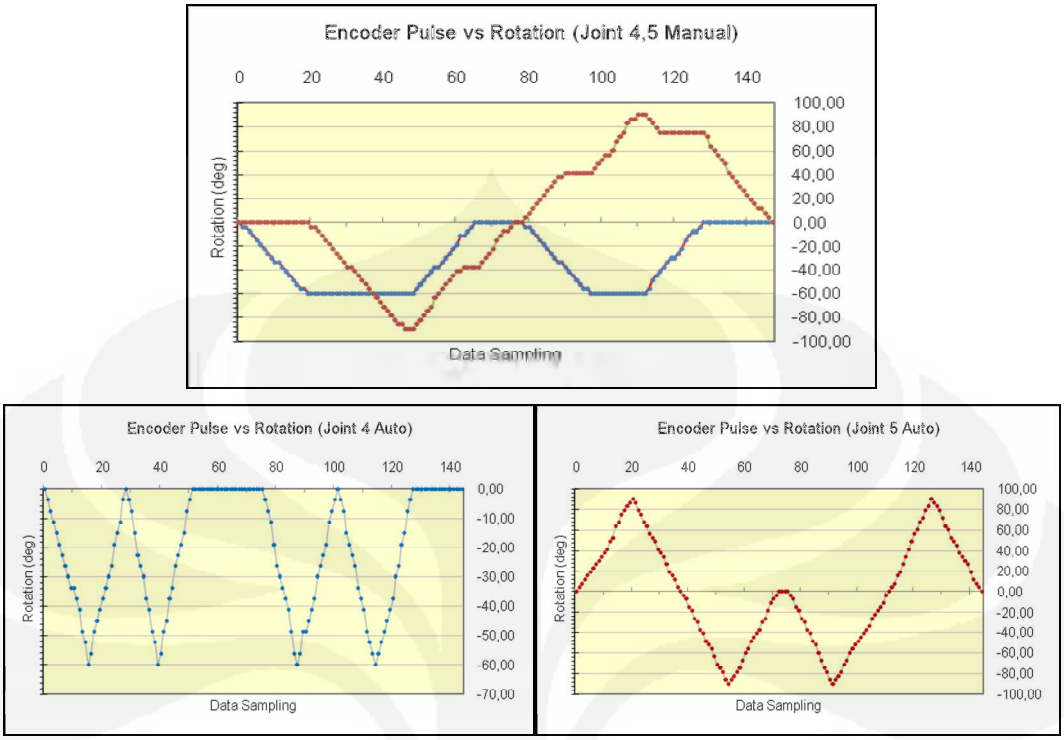
Langkah pergerakan	Posisi *J4*	Posisi *J5*	*Counter* pulsa *encoder*	Ket.
Step 1	0° (posisi awal)	0° (posisi awal)	0 (J4) / 0(J5)	Res. 96 *ppr*
Step 2	0° menuju -60°	0° menuju -90°	-16 (J4) / -24 (J5)	
Step 3	-60° menuju 0°	-90° menuju 0°	0 (J5) / 0 (J5)	
Step 4	0° menuju -60°	0° menuju +90°	-16 (J4) / +24 (J5)	
Step 5	-60° menuju 0°	+90° menuju 0°	0 (J4) / 0 (J5)	



Gambar 4.26 Penyimpanan data pergerakan per *step (joint 4* dan *joint 5)*



Gambar 4.27 Data pergerakan otomatis dua *joint*



Gambar 4.28 Kurva plot pergerakan otomatis dua *joint (joint 4* dan *5)*

4.2.4 Pengujian Pergerakan Berulang

Pengujian program untuk pergerakan berulang dilakukan untuk mengetahui unjuk kerja dan kemampuan *repeatability* sistem yang dibuat. Pengujian dilakukan dengan memasukkan dua *step* pergerakan yang disimpan saat mode operasi pergerakan manual. Nilai pembacaan *encoder* kedua *step* tersebut dijadikan *set point* yang harus dicapai saat pergerakan lengan robot secara otomatis. Pergerakan robot secara otomatis dilakukan berulang-ulang. Lintasan pergerakan berupa nilai pembacaan *encoder* ditampilkan melalui *hyperterminal, kemudian* diamati dan dibandingkan dengan pengamatan secara visual. Pengamatan secara visual diperoleh dengan memperhatikan tanda berupa garis pada konstruksi *joint* yang menandakan posisi awal dan akhir. Terjadinya perubahan diamati saat pergerakan secara otomatis dilakukan berulang-ulang untuk beberapa kali siklus pergerakan. Pengujian dilakukan dengan mengambil data pada *joint 4* dengan pergerakan sejauh 45° (resolusi *encoder* 96 *ppr*).

$$Pulsa\ encoder = \frac{45° \times 96\ ppr}{360} = 12\ pulsa\ (0° \rightarrow 45°)$$

Tabel 4.8 Pengujian pergerakan berulang *joint 4 (96 ppr)*

Langkah pergerakan	Posisi sudut	*Counter* pulsa *encoder*	Keterangan
Step 1	0° (posisi awal)	0	Resolusi 96 *ppr*
Step 2	0° menuju +45°	12	

Berdasarkan data pengujian yang diperoleh pada pergerakan berulang, setelah beberapa kali siklus pergerakan terjadi perubahan/pergeseran kecil pada pengamatan visual pergerakan *joint.*

Gambar 4.29 Pergeseran posisi setelah beberapa siklus pergerakan

Pergeseran itu tampak pada pengamatan tanda garis posisi awal dan akhir *(marking position)* pada konstruksi *joint* yang diuji. Hal tersebut dapat disebabkan oleh beberapa hal, antara lain:

- Resolusi *encoder* yang digunakan. Ketelitian *encoder* yang digunakan pada *joint 4* adalah *96 ppr* (3.75° per pulsa), sehingga perubahan terkecil yang mampu dibaca oleh *encoder* adalah sebesar 3.75°.
- Metode *counter* pembacaan pulsa *encoder* yang digunakan hanya 1x setiap sisi turun saja (transisi kondisi *high to low*). Hal ini karena karakteristik *timer2* yang digunakan pada program sebagai *up/down counter* pulsa *encoder* hanya menghitung naik/turun saat transisi kondisi *high to low* saja. Metode *quadrature* untuk pembacaan *encoder* dapat diaplikasikan pada program, namun banyaknya frekuensi *interrupt* yang dibutuhkan akan mengganggu jalannya program lain (termasuk program utama pergerakan lengan robot). Sehingga, metode ini tidak digunakan pada program yang dibuat.

- Pergerakan mekanik akibat momentum putaran motor penggerak *joint* sesaat sebelum motor benar-benar berhenti berputar. Meskipun relatif sangat kecil, namun dengan frekuensi pergerakan berulang yang semakin tinggi maka pergeseran yang terjadi akan menjadi semakin lebar.
- Usia pemakaian robot ini relatif sudah cukup lama. Perawatan dan inspeksi rutin pada komponen robot tidak dilakukan sebagaimana dianjurkan dalam buku manualnya. Sehingga ketelitian konstruksi mekanik, seperti *alignment* yang kurang baik, *backlash* dan *slip* yang relatif besar akan mempengaruhi kepekaan sensor yang digunakan dalam aplikasi robot.

BAB V
KESIMPULAN

Setelah sistem dianalisis dan diuji, maka dapat disimpulkan beberapa poin sebagai berikut:

1. Skripsi ini telah berhasil mengimplementasikan perancangan dan pembuatan sistem kendali robot lengan *Mitsubishi Movemaster RV-M1* menggunakan beberapa buah *microcontroller* berbasis *AT89S52* untuk setiap *joint*-nya dan terintegrasi dengan sebuah *microcontroller* jenis yang sama sebagai kendali pusatnya.
2. Teknik *handshaking* yang baik dengan *acknowledge signalling* yang tidak terputus dan didukung oleh algoritma program yang dibuat sesederhana mungkin sangat diperlukan untuk koordinasi/komunikasi sistem kendali *multi-controller* dengan konfigurasi *master-slaves.*
3. Resolusi *encoder* yang digunakan sebagai sensor posisi pergerakan lengan robot akan sangat mempengaruhi performa sistem yang dibuat. Semakin tinggi ketelitian sensor yang digunakan, maka semakin baik pula performa sistem kendali yang dihasilkan.
4. Ketelitian konstruksi mekanik, seperti *alignment* elemen transmisi yang digunakan, minimalisasi terjadinya *backlash* dan *slip* akan berpengaruh terhadap kepekaan sensor yang digunakan dalam aplikasi robot.

DAFTAR REFERENSI

Klafter ,D. Richard, Chmielewski, A.Thomas, Negin, Michael (1989), *Robotic Enginnering An Integrated Approach*, Prentice Hall International Editions, New Jersey-USA.

Schiling, J. Robert (1990), *Fundamentals of Robotics*, Prentice Hall International Editions, New Jersey-USA.

McComb, Gordon (1987), *Robot Builder's Bonanza 99 Inexpensive Robotics Projects*, McGraw Hill.

Mitsubishi Corp., *Industrial Micro-Robot System Model RV-M1, Instruction Manual.*

Dwiartomo, Bolo, Ir., M.Eng (2000), *Robotik*, Politeknik Manufaktur Bandung, Bandung.

Pancono, Suharyadi, Dipl.Ing, HTL, *Mesin Listrik 1*, Politeknik Manufaktur Bandung, Bandung.

Baldor motor and drives, handbook, Baldor Electric Company.

ST Microelectronics, AN380 Application Note, *How to Drive DC Motors with Smart Power Ics.*

ST Microelectronics, Data Sheet L298D.

Atmel Corp (2005), *Data Sheet AT89S52.*

Atmel Corp (2004), *Data Sheet AT24C256.*

www.elektro-terapan.blogspot.com

www.creativecommons.org

www.bipom.com

DAFTAR ISTILAH

Alignment : Pemeliharaan kondisi elemen mesin dan elemen transmisi pemindah putaran atau daya.

Backlash : Rugi-rugi mekanik akibat keausan elemen transmisi yang digunakan dalam suatu sistem (misal. hubungan *belt* dan *gear*).

DOF : *Degree of Freedom.* Merupakan jumlah pergerakan *independent* dari lengan robot, dilihat dari sudut pandang manipulator akhir *(end-effector)* atau setiap titik sumbu gerakan mekanik pada robot (tidak termasuk *end-effector*).

DOM : *Degree of Mobility.* Merupakan kemampuan untuk melakukan sebuah pergerakan (dari setiap pasangan *joint* dan *link-joint*).

I²C : *Inter Integrated Circuit.* Salah satu teknik antarmuka *master-slave microcontroller* dimana sinyal komunikasi pada teknik ini menggunakan dua buah jalur komunikasi yaitu *Serial Data (SDA)* dan *Serial Clock (CLK).*

Incrementally Encoders : Tipe *encoder* yang digunakan untuk mengubah gerakan linear atau putaran menjadi sinyal digital dengan teknik perbedaan fasa pulsa keluaran ±90º. Sinyal keluaran dengan perbedaan fasa ini digunakan untuk membedakan arah perputaran *encoder*

Joint dan *link* : *Joint* memungkinkan terjadinya suatu gerakan pada dua bagian tubuh atau konstruksi robot, sedangkan *link* menghubungkan tiap-tiap *joint* tersebut.

Leadthrough : Metode pemrograman robot dimana manipulator dipindahkan terlebih dahulu melalui lintasan pergerakan yang diinginkan dan direkam dalam memori oleh pengendali. Lintasan tersebut nantinya akan menjadi nilai setpoint pergerakan robot (*teaching by showing).*

Multiplexer : Rangkaian *digital* pemilih data dimana dari beberapa masukan hanya akan dipilih satu buah sebagai keluarannya. Sinyal-sinyal masukan yang akan dikeluarkan akan dipilih oleh suatu selektor.

Ppr : *Pulse per Rotation.* Merupakan jumlah pulsa keluaran per putaran yang menyatakan resolusi atau ketelitian *encoder*.

Quadrature : Teknik *decoding* pulsa keluaran *encoder* dimana proses *decoding* pulsa keluaran *encoder* dengan perbedaan fasa yang tetap (90°) dilakukan empat kali untuk setiap periode pulsa, yaitu pada sisi naik dan turun dari kedua fasa keluaran *encoder*.

SCL : Jalur *Serial Clock* yang digunakan pada komunikasi *i²c*.

SDA : Jalur *Serial Data* yang digunakan pada komunikasi *i²c*.

Serial Tx : Jalur serial untuk *transmit data (Tx)* ke *peripheral* lain.

Slip : Rugi-rugi akibat ketelitian konstruksi mekanik sesaat sebelum elemen pemindah putaran atau daya bergerak.

Timer/Counter : Fasilitas yang dimiliki oleh *microcontroller* untuk fungsi pewaktu *(timer)* maupun penghitung cacah naik/turun *(counter).*

Torque : Torsi motor merupakan perputaran dari suatu gaya terhadap suatu poros putarnya.

DAFTAR LAMPIRAN

Skematik dan *Port Config (Master-Slaves)*.

Joint #1 *(µc 1)*

Alias	Port #0
Pb_Cw	P0.0
Pb_Ccw	P0.1
Pb_Save	P0.2
Pb_Load	P0.3
Pb_Reset	P0.4
Pb_Auto	P0.5
n/a	P0.6
n/a	P0.7

Alias	Port #1
n/a	P1.0
n/a	P1.1
n/a	P1.2
n/a	P1.3
n/a	P1.4
n/a	P1.5
n/a	P1.6
n/a	P1.7

Alias	Port #2
Cw	P2.0
Ccw	P2.1
Ena_bridge	P2.2
n/a	P2.3
n/a	P2.4
n/a	P2.5
n/a	P2.6
n/a	P2.7

Alias	Port #3
n/a	P3.0
n/a	P3.1
n/a	P3.2
n/a	P3.3
limit_switch	P3.4
n/a	P3.5
n/a	P3.6
n/a	P3.7

Joint #2 *(µc 2)*

Alias	Port #0
Pb_Cw	P0.0
Pb_Ccw	P0.1
Pb_Save	P0.2
Pb_Load	P0.3
Pb_Reset	P0.4
Pb_Auto	P0.5
n/a	P0.6
n/a	P0.7

Alias	Port #1
Ph A (T2)	P1.0
Ph B (T2EX)	P1.1
i2c SCL	P1.2
i2c SDA	P1.3
limit_switch	P1.4
n/a	P1.5
n/a	P1.6
n/a	P1.7

Alias	Port #2
Cw	P2.0
Ccw	P2.1
Ena_bridge	P2.2
Brake	P2.3
n/a	P2.4
n/a	P2.5
n/a	P2.6
n/a	P2.7

Alias	Port #3
out_led	P3.0
J2 Tx	P3.1
Int0	P3.2
Ackw	P3.3
flag_in	P3.4
flag_out	P3.5
Stop2	P3.6
Cont2	P3.7

Joint #3 *(µc 3)*

Alias	Port #0
Pb_Cw	P0.0
Pb_Ccw	P0.1
Pb_Save	P0.2
Pb_Load	P0.3
Pb_Reset	P0.4
Pb_Auto	P0.5
n/a	P0.6
n/a	P0.7

Alias	Port #1
Ph A (T2)	P1.0
Ph B (T2EX)	P1.1
i2c SCL	P1.2
i2c SDA	P1.3
limit_switch	P1.4
n/a	P1.5
n/a	P1.6
n/a	P1.7

Alias	Port #2
Cw	P2.0
Ccw	P2.1
Ena_bridge	P2.2
Brake	P2.3
n/a	P2.4
n/a	P2.5
n/a	P2.6
n/a	P2.7

Alias	Port #3
out_led	P3.0
J3 Tx	P3.1
Int0	P3.2
Ackw	P3.3
flag_in	P3.4
flag_out	P3.5
stop3	P3.6
cont3	P3.7

Joint #4 *(µc 4)*

Alias	Port #0
Pb_Cw	P0.0
Pb_Ccw	P0.1
Pb_Save	P0.2
Pb_Load	P0.3
Pb_Reset	P0.4
Pb_Auto	P0.5
n/a	P0.6
n/a	P0.7

Alias	Port #1
Ph A (T2)	P1.0
Ph B (T2EX)	P1.1
i2c SCL	P1.2
i2c SDA	P1.3
limit_switch	P1.4
n/a	P1.5
n/a	P1.6
n/a	P1.7

Alias	Port #2
Cw	P2.0
Ccw	P2.1
Ena_bridge	P2.2
n/a	P2.3
n/a	P2.4
n/a	P2.5
n/a	P2.6
n/a	P2.7

Alias	Port #3
out_led	P3.0
J4 Tx	P3.1
Int0	P3.2
Ackw	P3.3
flag_in	P3.4
flag_out	P3.5
stop4	P3.6
cont4	P3.7

Joint #5 *(µc 5)*

Alias	Port #0
Pb_Cw	P0.0
Pb_Ccw	P0.1
Pb_Save	P0.2
Pb_Load	P0.3
Pb_Reset	P0.4
Pb_Auto	P0.5
n/a	P0.6
n/a	P0.7

Alias	Port #1
Ph A (T2)	P1.0
Ph B (T2EX)	P1.1
i2c SCL	P1.2
i2c SDA	P1.3
limit_switch	P1.4
n/a	P1.5
n/a	P1.6
n/a	P1.7

Alias	Port #2
Cw	P2.0
Ccw	P2.1
Ena_bridge	P2.2
n/a	P2.3
n/a	P2.4
n/a	P2.5
n/a	P2.6
n/a	P2.7

Alias	Port #3
out_led	P3.0
J5 Tx	P3.1
Int0	P3.2
Ackw	P3.3
flag_in	P3.4
flag_out	P3.5
stop5	P3.6
cont5	P3.7

Gripper *(µc 6)*

Alias	Port #0
Pb_Cw	P0.0
Pb_Ccw	P0.1
n/a	P0.2
n/a	P0.3
n/a	P0.4
n/a	P0.5
n/a	P0.6
n/a	P0.7

Alias	Port #1
n/a	P1.0
n/a	P1.1
n/a	P1.2
n/a	P1.3
n/a	P1.4
n/a	P1.5
n/a	P1.6
n/a	P1.7

Alias	Port #2
Cw	P2.0
Ccw	P2.1
Ena_bridge	P2.2
n/a	P2.3
n/a	P2.4
n/a	P2.5
n/a	P2.6
n/a	P2.7

Alias	Port #3
n/a	P3.0
n/a	P3.1
n/a	P3.2
n/a	P3.3
n/a	P3.4
n/a	P3.5
n/a	P3.6
n/a	P3.7

Master *(µc 7)*

Alias	Port #0
Pb_manual	P0.0
Pb_auto	P0.1
Pb_save	P0.2
Pb_load	P0.3
Pb_reset	P0.4
n/a	P0.5
n/a	P0.6
n/a	P0.7

Alias	Port #1
Cont25	P1.0
Cont34	P1.1
i2c SCL	P1.2
i2c SDA	P1.3
Flag_in	P1.4
n/a	P1.5
n/a	P1.6
n/a	P1.7

Alias	Port #2
Ackw2	P2.0
Ackw3	P2.1
Ackw4	P2.2
Ackw5	P2.3
Stop2	P2.4
Stop3	P2.5
Stop4	P2.6
Stop5	P2.7

Alias	Port #3
Intr2	P3.0
Master_Tx	P3.1
Intr3	P3.2
Intr4	P3.3
Intr5	P3.4
mux_lsb	P3.5
mux_mid	P3.6
mux_msb	P3.7

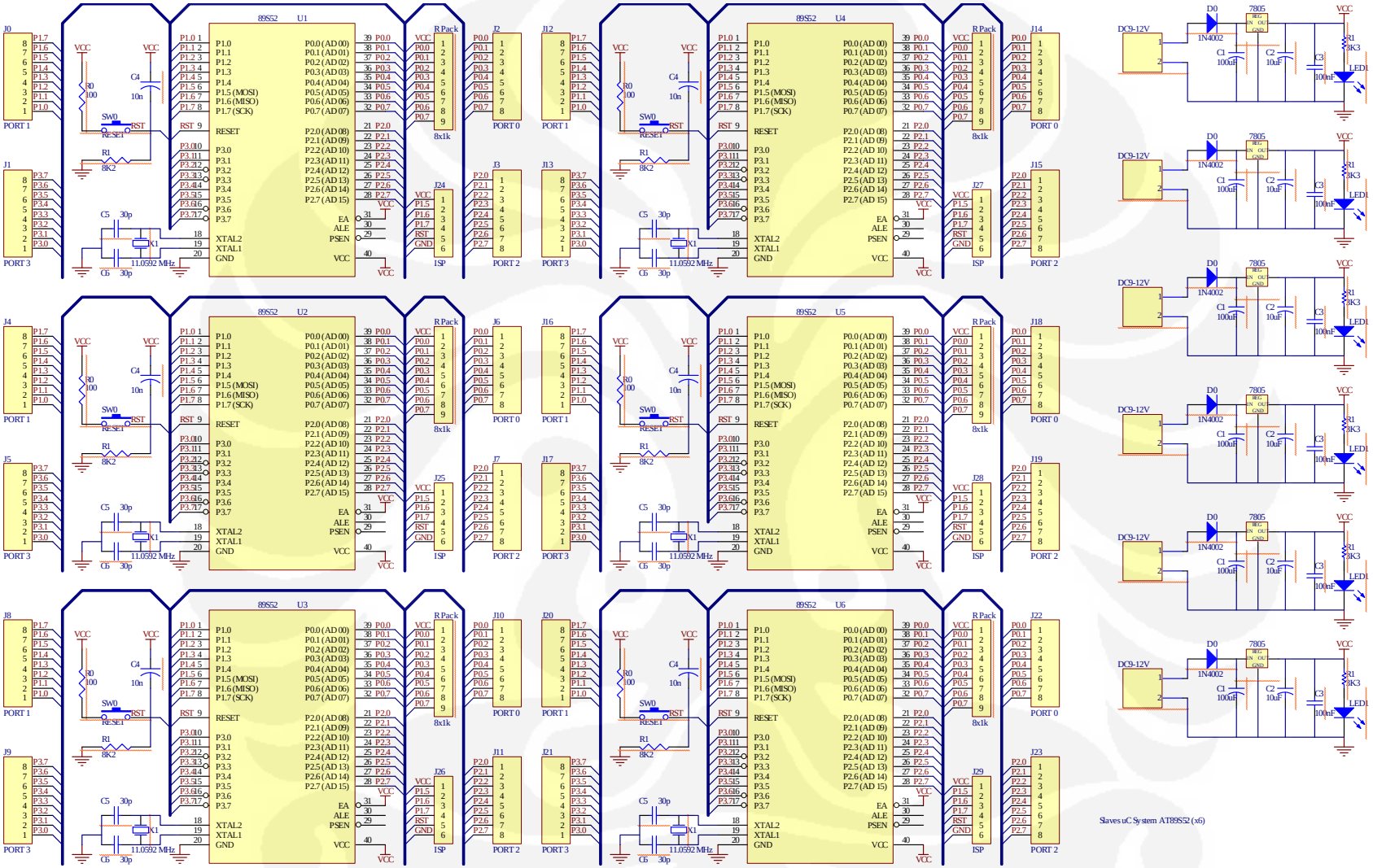
In
Out
In/Out

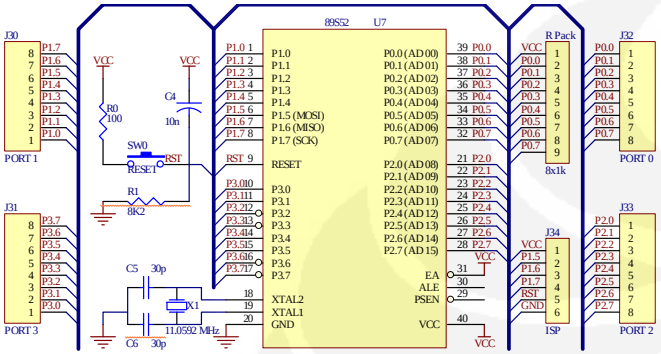
Alokasi Memori Serial *(Save Pos/Step)*.

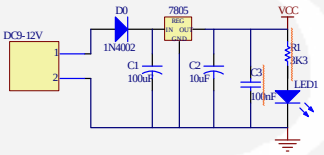
24C256

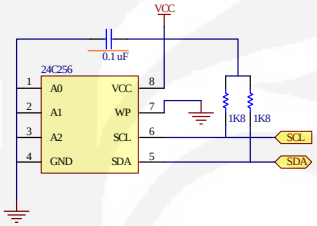
Address Block	*Address Word*		*Address*		Keterangan
	start	*end*	*start*	*end*	
Joint #1	0000	0FFF	0	4095	*not used*
Joint #2	1001	1FFF	4097	8191	*Save Pos J2*
Joint #3	2001	2FFF	8193	12287	*Save Pos J3*
Joint #4	3001	3FFF	12289	16383	*Save Pos J4*
Joint #5	4001	4FFF	16385	20479	*Save Pos J5*
Gripper	5000	5FFF	20480	24575	*not used*
n/a	6000	6FFF	24576	28671	spare
n/a	7000	7FFF	28672	32767	spare

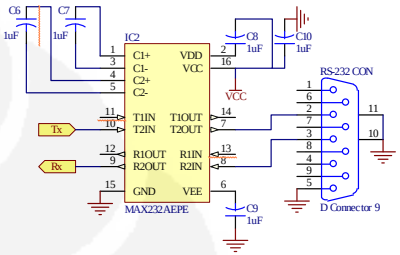
511slots per joint

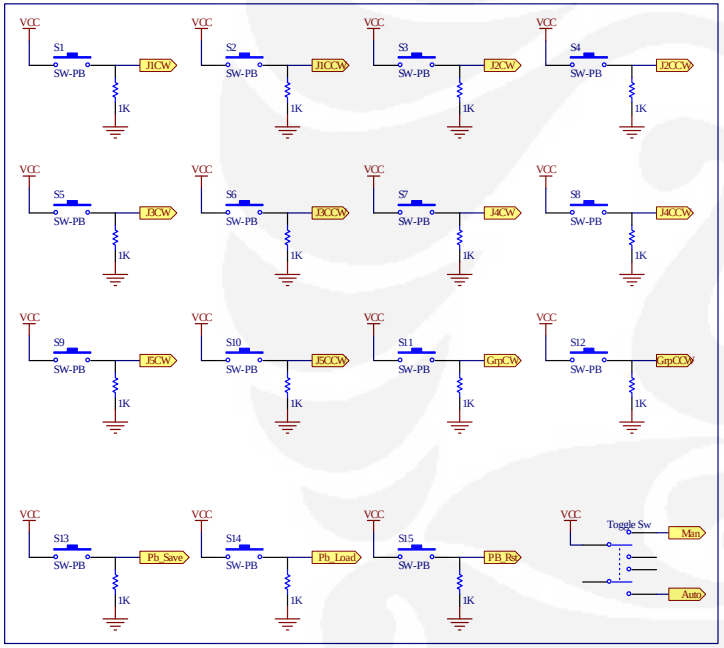

89S52 U1
89S52 U2
89S52 U3
89S52 U4
89S52 U5
89S52 U6
RESET
XTAL2
XTAL1
GND
VCC
EA
ALE
PSEN
11.0592 MHz
DC9-12V
1N4002
7805
LED1
Slaves uC System AT89S52 (x6)











Master uC System AT89S52 (with RS-232 Converter, Keypad, i2c eeprom AT24C256)