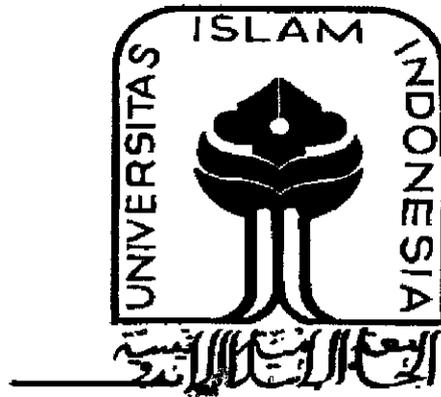


**PEMBANGUNAN PROTOTIP SAMBUNGAN 0 BERBIAYA  
RENDAH PADA LENGAN TELEROBOT INDUSTRI**

**TUGAS AKHIR**

**Diajukan Sebagai Salah Satu Syarat  
Untuk Memperoleh Gelar Sarjana Strata-1  
Teknik Industri**

Demi All  
yang seti  
pengakua  
intelektua  
Universit



Oleh

Nama : Rizqan Zaidi

No. Mahasiswa : 03 522 153

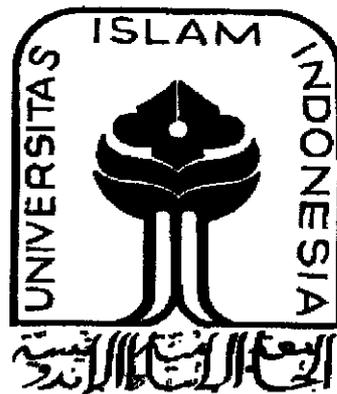
**JURUSAN TEKNIK INDUSTRI  
FAKULTAS TEKNOLOGI INDUSTRI  
UNIVERSITAS ISLAM INDONESIA  
YOGYAKARTA**



**PEMBANGUNAN PROTOTIP SAMBUNGAN 0 BERBIAYA  
RENDAH PADA LENGAN TELEROBOT INDUSTRI**

**TUGAS AKHIR**

**Diajukan Sebagai Salah Satu Syarat  
Untuk Memperoleh Gelar Sarjana Strata-1  
Teknik Industri**



Oleh

Nama : Rizqan Zaidi  
No. Mahasiswa : 03 522 153

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



## HALAMAN PENGAKUAN

Demi Allah, Saya akui karya ini adalah hasil karya saya sendiri kecuali nukilan dan ringkasan yang setiap satunya telah saya jelaskan sumbernya. Jika dikemudian hari ternyata terbukti pengakuan saya ini tidak benar dan melanggar peraturan yang sah dalam karya tulis dan hak intelektual maka saya bersedia ijazah yang telah saya terima untuk ditarik kembali oleh Universitas Islam Indonesia.

Yogyakarta, 4 Agustus 2009



Rizqan Zaidi  
03522153

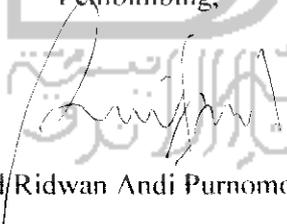
الجامعة الإسلامية  
الابتناء الاندوف

**LEMBAR PENGESAHAN PEMBIMBING****PEMBANGUNAN PROTOTIP SAMBUNGAN 0 BERBIAYA  
RENDAH PADA LENGAN TELEROBOT INDUSTRI****TUGAS AKHIR**

Oleh :  
Nama : Rizqan Zaidi  
No. Mahasiswa : 03 522 153

Yogyakarta, 13 Juli 2009

Pembimbing,

  
(Muhammad/Ridwan Andi Purnomo, ST, MSc)

## LEMBAR PENGESAHAN PENGUJI

### PEMBANGUNAN PROTOTIP SAMBUNGAN 0 BERBIAYA RENDAH PADA LENGAN TELEROBOT INDUSTRI

#### TUGAS AKHIR

Disusun Oleh :

Nama : Rizqan Zaidi

No. Mahasiswa : 03 522 153

Telah Dipertahankan di Depan Sidang Penguji Sebagai  
Salah Satu Syarat untuk Memperoleh Gelar Sarjana Strata-I  
Teknik Industri

Yogyakarta, 4 Agustus 2009

Tim Penguji

**Muhammad Ridwan Andi Purnomo, ST, MSc**

Ketua

**Taufik Immawan, ST, MM**

Anggota I

**Yuli Agusti Rochman ST, M.Eng**

Anggota II

Mengetahui,

Ka. Prodi Teknik Industri

Fakultas Teknologi Industri

Universitas Islam Indonesia



**Prof. Dr. Ir. R. Chairul Saleh, MSc.**

## MOTTO

“ Sesungguhnya bersama kesulitan ada kemudahan”

(QS. Al-Insyirah: 6)

“ Allah tidak akan membebani seseorang melainkan sesuai dengan kesanggupannya ”

(QS. Al-Baqarah: 286)

“ Sebaik-baik manusia adalah orang yang banyak manfaatnya (kebaikannya) kepada manusia lainnya ”

(H. R. Qadla'ie dan Jabir)



## KATA PENGANTAR

*Assalamu'alaikum Wr, Wb.*

Puji dan syukur kami panjatkan kehadirat Allah SWT. Yang telah melimpahkan rahmat dan hidayah-Nya. Shalawat dan salam ditujukan kepada Nabi Muhammad SAW yang telah membawa kita semua dari alam kegelapan menuju ke alam yang terang benderang ini.

Penulisan Tugas Akhir ini merupakan salah satu syarat untuk meraih gelar Sarjana (S1) di Fakultas Teknologi Industri, Jurusan Teknik Industri, Universitas Islam Indonesia. Penulis sadar didalam penyusunan tugas akhir ini masih banyak kekurangan, oleh karena itu dengan segala kerendahan hati penulis meminta maaf yang sebesar-besarnya dan menerima segala kritik dan saran yang bersifat membangun.

Didalam penyusunan tugas akhir ini penulis menemukan kesulitan, namun berkat bantuan dan dukungan dari semua pihak, penulis dapat menyelesaikan tugas akhir ini. Dengan segenap ketulusan hati pada kesempatan ini, penulis ingin menyampaikan ucapan terima kasih kepada :

1. Kedua orang tua yang tercinta serta saudara- saudara ku, terima kasih atas kesabarannya hingga saat ini.
2. Bapak Muhammad Ridwan Andi Purnomo, ST, MSc. selaku Dosen Pembimbing, yang telah banyak membantu memberi bimbingan, arahan, dan saran dalam penyusunan tugas akhir ini.

3. Ketua Prodi Teknik Industri UII, terimakasih atas segala dukungan baik secara langsung maupun secara tidak langsung kepada penulis dalam menyusun tugas akhir ini.
4. Semua pihak yang tidak dapat penulis sebut satu persatu yang telah memberikan bantuan dalam penyusunan tugas akhir ini

Dengan mengucapkan terima kasih semoga bantuan yang telah diberikan mendapat balasan dari Allah SWT, Aamien.

Akhir kata penulis berharap agar tugas akhir ini dapat bermanfaat khususnya bagi penulis dan bagi semua pihak pada umumnya.

*Wassalamu'alaikum Wr, Wb.*

Yogyakarta, 13 Juli 2009

Rizqan Zaidi



## ABSTRAK

*Tugas Akhir ini memaparkan mengenai pembangunan prototip sambungan 0 berbiaya rendah pada sistem telerobot. Tujuan dari pembangunan prototip ini adalah untuk pembelajaran pergerakan lengan telerobot industri yang dikontrol dari jarak jauh. Sistem telerobot memungkinkan sebuah pabrik beroperasi selama 24 jam. Prototip lengan telerobot di bangun dengan menggunakan motor servo dan roda-roda gigi sedangkan alat pengontrol dibangun dengan menggunakan pengontrol mikro sebagai prosesornya. Pertukaran data dalam sistem teleoperasi dilakukan melalui internet, yaitu menggunakan protocol TCP/IP standar. Hasil pengujian prototip sambungan 0 menunjukkan bahwa pergerakannya dapat dikontrol dari jarak jauh dan bergerak sendiri sesuai dengan yang diinginkan operator.*

Kata kunci: Sambungan 0, Telerobotik, Teleoperasi, Sistem berbiaya rendah



## TAKARIR

<i>Acrylic</i>	: bahan yang terbuat dari plastik
<i>Client</i>	: komponen yang berisi aplikasi dan memiliki tanggung jawab untuk melakukan interaksi dengan <i>user</i>
<i>Database</i>	: kumpulan file atau tabel yang saling berhubungan yang disimpan dalam media penyimpanan elektronik
<i>End of effector</i>	: ujung lengan
<i>Handshake</i>	: sistem jabat tangan
<i>Input</i>	: pemasukan data
<i>Intelligent Machine</i>	: mesin cerdas
<i>Joystick</i>	: alat pengontrol
<i>Keypad</i>	: tombol
<i>Low Cost</i>	: biaya rendah
<i>Loop</i>	: pengulangan
<i>Master-Slave</i>	: komponen primer-sekunder
<i>Microsurgery</i>	: bedah mikro
<i>Middleware</i>	: perangkat tengah
<i>Mobile Robot</i>	: robot berjalan
<i>Robot Arm</i>	: lengan robot
<i>Server</i>	: komputer yang menjadi pusat seluruh kegiatan untuk menangani permintaan dari <i>client</i>
<i>Tool</i>	: alat
<i>User</i>	: orang yang menggunakan sistem

## DAFTAR ISI

<b>HALAMAN JUDUL</b> .....	i
<b>HALAMAN PENGAKUAN</b> .....	ii
<b>LEMBAR PENGESAHAN PEMBIMBING</b> .....	iii
<b>LEMBAR PENGESAHAN PENGUJI</b> .....	iv
<b>HALAMAN PERSEMBAHAN</b> .....	v
<b>MOTTO</b> .....	vi
<b>KATA PENGANTAR</b> .....	vii
<b>ABSTRAK</b> .....	ix
<b>TAKARIR</b> .....	x
<b>DAFTAR ISI</b> .....	xi
<b>DAFTAR TABEL</b> .....	xiv
<b>DAFTAR GAMBAR</b> .....	xv
<b>BAB I            PENDAHULUAN</b>	
1.1    Latar Belakang Masalah .....	1
1.2    Rumusan Masalah .....	2
1.3    Batasan Masalah .....	3
1.4    Tujuan Penelitian .....	3
1.5    Manfaat Penelitian .....	3
1.6    Sistematika Penulisan .....	4
<b>BAB II            KAJIAN PUSTAKA</b>	
2.1    Robotika dalam Industri .....	5
2.2    Teleoperasi .....	7
2.3    Telerobotik.....	10
2.4    Otomasi dan Robot Industri.....	11
2.4.1 Otomasi.....	11
2.4.2 Robot Industri .....	13
2.4.3 Konfigurasi Manipulator.....	14
2.4.4 Sambungan 0 Lengan Robot.....	17

2.4.5 Perkembangan Sistem Teleoperasi dan Telerobotik di Dunia Industri.....	17
---	----

### **BAB III METODE PENELITIAN**

3.1 Identifikasi Masalah .....	20
3.2 Perumusan Masalah .....	22
3.3 Studi Pustaka .....	22
3.4 Perancangan dan Pembuatan Perangkat Keras .....	22
3.5 Perancangan dan Pembuatan Perangkat Lunak .....	23
3.6 Pengujian Sistem.....	23
3.7 Analisis Hasil.....	23
3.8 Kesimpulan dan Saran .....	24

### **BAB IV PENGUMPULAN DAN PENGOLAHAN DATA**

4.1 Pengumpulan Data .....	25
4.1.1 Motor Servo .....	26
4.1.2 Sistem Kontrol .....	26
4.1.3 Komunikasi Data Serial .....	28
4.1.4 Sistem Pemrograman Internet .....	30
4.1.5 Pemrograman pada Pengontrol Mikro .....	31
4.1.6 Pembuatan Mekanik Lengan Robot .....	32
4.2 Pengolahan Data .....	33
4.2.1 Pengujian Sistem .....	34
4.2.2 Biaya Pembuatan .....	39

### **BAB V PEMBAHASAN**

5.1 Kelebihan Prototip Sambungan 0 Lengan Telerobot .....	40
5.2 Kekurangan Prototip Sambungan 0 Lengan Telerobot .....	41

### **BAB VI PENUTUP**

6.1 Kesimpulan .....	42
6.2 Saran .....	42



DAFTAR PUSTAKA  
LAMPIRAN



**DAFTAR TABEL**

Tabel 4.1	Tabel Pengujian Sistem Kontrol .....	34
Tabel 4.2	Tabel Pengujian Sistem Kontrol dan Perangkat Tengah.....	37
Tabel 4.3	Tabel Daftar Harga Komponen.....	39



## DAFTAR GAMBAR

Gambar 2.1	Konsep Dasar Teleoperasi .....	8
Gambar 2.2	Robot Tangan dalam Sistem CIM .....	12
Gambar 2.3	Manipulator Planar .....	16
Gambar 3.1	Diagram Alir Penelitian .....	22
Gambar 4.1	Arsitektur dari Prototip yang Dibangun .....	25
Gambar 4.2	Motor Servo .....	26
Gambar 4.3	Susunan kaki Mikrokontroler ATmega16 .....	27
Gambar 4.4	Rangkaian Lengkap Sistem Kontrol .....	28
Gambar 4.5	Kode Bayangan Untuk Mengirimkan Posisi Awal Motor Servo.....	29
Gambar 4.6	Kode Bayangan Untuk Mengirimkan Perintah dari Komputer ke Sambungan 0 Lengan Robot .....	29
Gambar 4.7	Kode Bayangan Untuk Pergerakan Lengan Robot ke Komputer.....	30
Gambar 4.8	(a). Proses Jabat Tangan di Komputer Server.....	31
	(b). Sistem Jabat Tangan di Komputer Klien.....	31
Gambar 4.9	Perancangan Perangkat Lunak pada Mikrokontroler .....	32
Gambar 4.10	Rancangan Mekanik Lengan Robot .....	33
Gambar 4.11	Prototip Sambungan 0 Lengan Telerobot .....	33
Gambar 4.12	Tampilan Pengujian Pertama pada Sistem Kontrol .....	35
Gambar 4.13	Tampilan Pengujian Kedua pada Sistem Kontrol .....	35
Gambar 4.14	Tampilan Pengujian Ketiga pada Sistem Kontrol .....	36
Gambar 4.15	(a). Tampilan Pengujian 1 pada Sistem Kontrol .....	37
	(b). Tampilan Pengujian 1 pada Perangkat Tengah .....	37
Gambar 4.16	(a). Tampilan Pengujian 2 pada Sistem Kontrol.....	38
	(b). Tampilan Pengujian 2 pada Perangkat Tengah .....	38
Gambar 4.17	(a). Tampilan Pengujian 3 pada Sistem Kontrol .....	38
	(b). Tampilan Pengujian 3 pada Perangkat Tengah .....	38

## BAB I

### PENDAHULUAN

#### 1.1. Latar Belakang

Keunggulan dalam teknologi robotik tak dapat dipungkiri telah lama dijadikan ikon kebanggaan negara-negara maju di dunia. Kecanggihan teknologi yang dimiliki, gedung-gedung tinggi yang mencakar langit, tingkat kesejahteraan rakyatnya yang tinggi, kota-kotanya yang modern, belumlah terasa lengkap tanpa popularitas kepiawaian dalam dunia robotik.

Pada awalnya, aplikasi robot hampir tak dapat dipisahkan dengan industri sehingga muncul istilah industrial robot dan robot manipulator. Definisi yang populer ketika itu, robot industri adalah suatu robot tangan (*robot arm*) yang diciptakan untuk berbagai keperluan dalam meningkatkan produksi, memiliki bentuk lengan-lengan kaku yang terhubung secara seri dan memiliki sendi yang dapat bergerak berputar (rotasi) atau memanjang/memendek (translasi atau prismatic). Satu sisi lengan yang disebut sebagai pangkal ditanam pada bidang atau meja yang statis (tidak bergerak), sedangkan sisi yang lain yang disebut sebagai ujung (*end of effector*) dapat dimuati dengan *tool* tertentu sesuai dengan tugas robot.

Seiring perkembangan teknologi yang semakin mengutamakan efisiensi dan tingkat fleksibilitas yang tinggi serta munculnya masalah pengandali yang terhambat oleh jarak oleh karena itu munculah konsep teleoperasi. Teleoperasi atau yang dikenal juga dengan istilah teleotomasi merupakan konsep yang dapat

mengkombinasikan kecerdasan dan kemampuan beradaptasi dari manusia dengan kemampuan dan ketahanan sebuah robot untuk melakukan pekerjaan yang sangat sulit dikerjakan yang didalam dunia robotik disebut telerobot.

Sistem telerobot ini sangat bermanfaat dalam dunia industri. Salah satunya dapat mendukung sistem pabrik 24 jam. perusahaan dapat memantau apa yang sedang terjadi di pabrik selama 24 jam penuh serta memudahkan operator dalam mengawasi apa yang sedang terjadi di pabrik tanpa harus berada di lokasi yang dihubungkan melalui jaringan komputer.

Untuk mendemonstrasikan dan mempelajari fungsi dari suatu sistem lengan robot secara lebih mendalam pada dunia industri maka dibutuhkan suatu prototip mekanisme lengan telerobot. Alat yang dibuat ini harus dapat menunjukkan secara baik sistem kerja dari lengan telerobot industri dan berbiaya rendah serta sehingga dapat mengefisienkan biaya pembuatan.

## **1.2. Rumusan Masalah**

Dari latar belakang diatas dapat ditarik suatu permasalahan yaitu bagaimana membangun suatu sistem yang dapat menunjukkan mekanisme sambungan 0 pada suatu sistem lengan telerobot industri.

### 1.3. Batasan Masalah

Pada penelitian ini masalah dibatasi pada :

1. Penelitian terhadap perancangan alat dibatasi pada mekanisme lengan telerobot pada sambungan 0 saja.
2. Deteksi terhadap sudut dilakukan dengan menggunakan sensor berbiaya rendah, yaitu menggunakan potensiometer.
3. Sistem belum menggunakan EEPROM pada mikrokontroler sehingga belum memungkinkan dalam menyimpan data.

### 1.4 Tujuan Penelitian

Adapun tujuan penelitian adalah untuk membuat suatu prototip yang dapat menunjukkan gerakan sambungan 0 fengan telerobot berbiaya rendah.

### 1.5 Manfaat Penelitian

Manfaat yang dapat diambil dari penelitian ini adalah membantu peneliti dan mahasiswa lain dalam memahami konsep telerobotik yang akan sangat berguna dalam dunia industri.



## 1.6 Sistematika Penulisan

### **BAB II. KAJIAN PUSTAKA**

Bab ini menyajikan tentang konsep dasar yang diperlukan dalam penyelesaian masalah dan memuat hasil penelitian sebelumnya yang berhubungan dengan penelitian yang akan dilakukan.

### **BAB III METODE PENELITIAN**

Berisi tentang diagram alur penelitian serta metode-metode yang akan dilakukan dalam penelitian

### **BAB IV PENGUMPULAN DAN PENGOLAHAN DATA**

Bab ini berisi cara pengambilan data dan bagaimana pengolahan data yang kemudian ditampilkan sebagai satu kesatuan sistem. Pada bagian ini juga menjadi acuan yang digunakan pada bab selanjutnya

### **BAB V PEMBAHASAN**

Bab ini berisi analisis dan pembahasan dari bab sebelumnya serta menyesuaikan hasil dengan tujuan penelitian.

### **BAB VI KESIMPULAN DAN SARAN**

Bab ini berisi kesimpulan dan saran yang telah diambil dari tulisan yang dibuat serta implikasi yang dikemukakan oleh penulis terhadap hasil yang dibuat.

### **DAFTAR PUSTAKA**

### **LAMPIRAN**

1. Gambar
2. Tabel



## BAB II

### KAJIAN PUSTAKA

#### 2.1 Robotika dalam Industri

Keunggulan teknologi robotik telah lama dijadikan ikon kebanggaan negara-negara maju di dunia. Dalam industri manufaktur, saat ini telah banyak menggunakan teknologi ini untuk melakukan proses dalam jumlah banyak dan berulang, rumit dan memerlukan ketelitian yang tinggi. Dalam industri berat, peranan robot juga sangat diperlukan untuk melakukan penanganan material-material yang tidak mungkin dilakukan oleh manusia.

Pada kurun waktu 10 tahun terakhir ini, aplikasi robot hampir tak dapat dipisahkan dengan industri, sehingga muncul istilah robot industri. Robot industri adalah suatu robot tangan (*robot arm*) yang diciptakan untuk berbagai keperluan dalam meningkatkan produksi, memiliki bentuk lengan-lengan kaku yang terhubung secara seri dan memiliki sambungan yang dapat bergerak berputar (rotasi) atau memanjang/memendek (translasi atau prismatic). Satu sisi lengan yang disebut sebagai pangkal ditanam pada bidang atau meja yang statis (tidak bergerak), sedangkan sisi yang lain yang disebut sebagai ujung (*end of effector*) dapat dimuati dengan *tool* tertentu sesuai dengan tugas robot.

Robotik memiliki unsur yang sedikit berbeda dengan ilmu-ilmu dasar atau terapan yang lain. Ilmu dasar biasanya berkembang dari suatu asas atau hipotesis yang kemudian diteliti secara ilmiah. Ilmu terapan dikembangkan setelah ilmu-ilmu yang mendasarinya berkembang dengan baik. Sedangkan ilmu robotik lebih

sering melalui pendekatan praktis. Kemudian melalui suatu pendekatan atau perumpamaan (asumsi) dari hasil pengamatan perilaku makhluk hidup atau benda/mesin/peralatan bergerak lainnya dikembangkanlah penelitian secara teoritis. Dari teori kembali kepada praktis, dan dari sini robot berkembang menjadi lebih canggih.

Perkembangan penelitian di bidang robotik lazimnya dapat segera diketahui dengan mencermati aplikasinya di dunia industri atau produk kegiatan penelitian Skala laboratorium di grup-grup penelitian yang tersebar di berbagai institusi pendidikan dan penelitian di negara-negara maju. Dengan mudahnya mengakses internet sekarang ini, dan banyaknya sumber-sumber informasi masa kini yang tersebar terbuka di situs-situs penelitian, maka mencari tahu suatu perkembangan terbaru dalam, dunia robotik menjadi sangat mudah.

Pada dasarnya dilihat dari struktur dan fungsi fisiknya (pendekatan visual) robot terdiri dari dua bagian, yaitu *non-mobile robot* dan *mobile robot*. Kombinasi keduanya dapat menghasilkan kelompok kombinasi konvensional (*mobile & non-mobile*) dan kelompok non-konvensional. Kelompok pertama sengaja diberinama konvensional karena nama yang dipakai dalam konteks penelitian adalah nama-nama yang dianggap sudah umum, seperti *mobile manipulator*, *climbing robot* (robot pemanjat), *walking robot* (misal: bi-ped robot) dan nama-nama lain yang sudah populer. Sedangkan kelompok non-konvensional dapat berupa robot *humanoid*, *animaloid*, *extra-ordinary*, atau segala bentuk movasi penyerupaan yang bisa dilakukan. Kelompok kedua ini banyak dimanfaatkan sebagai ikon keunggulan dalam penelitian robotik, seperti robot ASIMO buatan Jepang. Sementara robot bawah air dan robot terbang lebih banyak dikembangkan sebagai peralatan

untuk membantu penelitian yang berkaitan dan untuk proyek pertahanan atau mesin perang.

Dari kelompok *non-mobile* yang sering disebut sebagai "keluarga robot" adalah robot arm atau manipulator saja. Sementara yang lebih mudah dikenali sebagai mesin cerdas (*intelligent machine*) yang tidak selalu tampak memiliki bagian tangan, kaki atau roda untuk bergerak lebih lazim disebut dengan nama khusus sesuai fungsinya. Mereka biasanya memiliki nama-nama yang tersambung. Misalnya mesin-mesin *otomatis Lathe, Milling, Drilling Machine, CNC (Computer Numerical Control) Machine, EDM (Electric Discharge Machine)*, dan berbagai peralatan otomatis yang biasa dijumpai di pabrik-pabrik modern.

*Mobile Robot* adalah tipe robot yang paling populer dalam dunia penelitian robotik. Sebutan ini biasa digunakan sebagai kata kunci utama untuk mencari rujukan atau referensi yang berkaitan dengan robotik di internet. Publikasi dengan judul yang berkaitan dengan *mobile robot* akan menjadi daya tarik tidak hanya bagi kalangan peneliti tapi juga bagi kalangan awam. Dari segi manfaat, penelitian tentang berbagai tipe *mobile robot* diharapkan dapat membantu manusia dalam melakukan otomasi dalam transportasi, platform bergerak untuk robot industri, eksplorasi tanpa awak, dan masih banyak lagi.

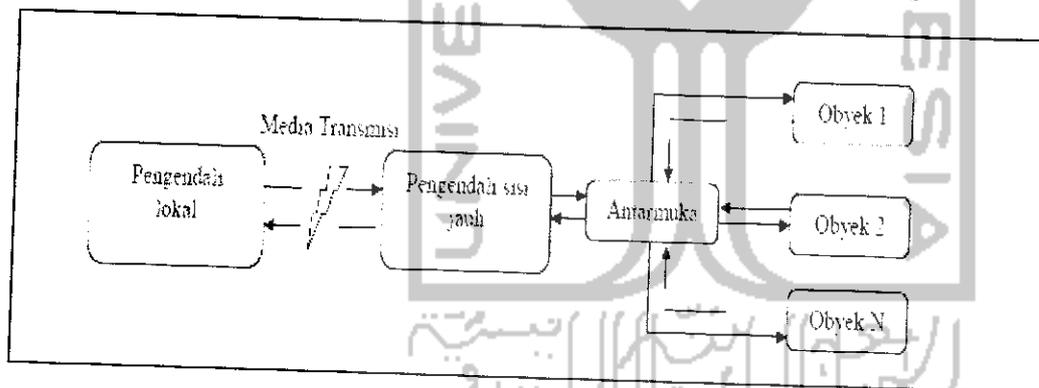
## 2.2 Teleoperasi

Aplikasi pengendalian dari jarak jauh tersebut sering disebut dengan nama teleoperasi atau sering pula disebut sebagai teleotomasi. Istilah teleoperasi mengandung dua kata kunci yang akan mempermudah untuk menebak arti dari istilah teleoperasi tersebut. Kata kunci yang pertama adalah "tele" yang berarti jauh dan kata

kunci kedua adalah “ operasi” yang berhubungan dengan melakukan sebuah aktivitas atau kerja dengan sebuah alat. Jika kedua kata ini disatukan, istilah yang diperoleh memberikan sebuah arti yang sangat khusus, dalam hal ini mengacu pada sebuah sistem pengoperasian dari jarak jauh. Teleoperasi menurut Alvares, et al. (2006) merupakan kendali langsung dan berkelanjutan atas sebuah mesin yang memungkinkan interaksi antara manusia dan mesin.

Konsep teleoperasi diinspirasi untuk diterapkan pada kondisi lingkungan yang sangat berbahaya dan tidak terstruktur sehingga situasinya lingkungannya tidak dapat diprediksikan secara tepat seperti daerah pertambangan, pembuangan limbah bom, bawah laut, dan lain sebagainya.

Didalam dunia industri, teleoperasi memungkinkan interaksi antara manusia dengan mesin yang dengan interaksi tersebut dapat menyelesaikan suatu pekerjaan berat sekalipun. Secara umum konsep teleoperasi dapat digambarkan sebagai berikut :



Gambar 2.1 Konsep Dasar Teleoperasi

Secara garis besar terdapat dua buah komponen utama yang harus dipersiapkan yaitu bagian pengendali lokal (*local site*) dan pengendali sisi jauh (*remote site*). Pengendali lokal merupakan bagian yang berhubungan dengan operator yang biasanya berupa program didalam komputer yang berhubungan dengan suatu objek yang akan

dikendalikan. Bagian selanjutnya yaitu pengendali sisi jauh. Pengendali ini berhubungan dengan pengendali lokal melalui media transmisi. Jadi pengendali dari jarak jauh bertugas untuk menerima masukan dari pengendali lokal lewat media transmisi yang terhubung dengannya dan selanjutnya mengolah data yang diterima tersebut menjadi sebuah perintah yang dikenali oleh piranti yang dikendalikan.

Dewasa ini jaringan komputer menjadi sangat kompleks di bandingkan tahun 1980-an sebagai media untuk pertukaran data dan mengatur hal tersebut menjadi sebuah tantangan (Manzoor dan Nefti, 2009). Apabila dibandingkan dengan melihat pertukaran data antara pengendali lokal dengan pengendali sisi jauh, sistem teleoperasi dapat dibedakan juga dalam (Hartanto dan Purbo, 2002):

a. Sistem Satu Arah

Pertukaran data hanya dilakukan dari salah satu pengendali lokal ke pengendali sisi jauh atau sebaliknya saja, misalnya sistem telemonitoring cuaca suatu kota.

b. Sistem Dua Arah

Dalam sistem ini terjadi pertukaran data antara penegndali lokal dengan pengendali sisi jauh secara berulang-ulang, misalnya sistem telerobotik.

Adapun komponen dasar dari sebuah sistem teleoperasi adalah sebagai berikut (Hartanto dan Purbo, 2002):

- a. *Server* yang berfungsi sebagai penghubung ke peralatan yang dikendalikan.
- b. *Client* atau antarmuka sebagai penghubung dengan operator atau pengguna.
- c. Media transmisi.

Disamping komponen-komponen tersebut beberapa peneliti juga menambahkan pula komponen tambahan dalam arsitektur sistemnya, di antaranya *database*, dan lain sebagainya.

### 2.3 Telerobotik

Dalam beberapa tahun belakangan ini sistem telerobotik merupakan sistem yang berkembang cukup pesat dan banyak peneliti yang meneliti tentang konsep dan pengembangan telerobotik (Bambang, 2007) Telerobotik merupakan teknologi yang memungkinkan pengendalian suatu sistem sekaligus mengawasi apa yang terjadi pada sistem tersebut.

Internet dan telerobotik merupakan bagian yang tidak dapat dipisahkan. Penggabungan kedua teknologi ini banyak dipakai dalam beberapa bidang seperti bidang kesehatan, industri manufaktur, militer, dunia hiburan bahkan digunakan dalam bidang antariksa. Telerobotik memungkinkan setiap orang berpartisipasi didalamnya karena cakupannya yang sangat luas.

Sistem pertama yang dibangun berupa sebuah sistem *master-slave*, dimana seorang operator menggerakkan sebuah tiruan robot yang sebenarnya berbentuk seperti sebuah *joystick* komputer pada tahun 1954 oleh Goertz di Argonne National Laboratory. *Joystick* ini, yang selanjutnya disebut sebagai *master*, akan mengirimkan sinyal-sinyal tertentu ke komputer yang selanjutnya mengolah sinyal-sinyal tersebut menjadi perintah yang dapat dikerjakan oleh robot yang terhubung dengan komputer tersebut. Jarak antara *joystick* tersebut dengan komputer cukup jauh sampai beberapa meter. Robot yang dikendalikan melalui *joystick* tersebut dinamakan *slave*. Robot akan mengikuti seluruh gerakan *joystick* yang digerakkan oleh operator tersebut. Namun sistem ini belumlah memanfaatkan teknologi komputer sebagai media transmisi, akan tetapi menggunakan infrastruktur tersambungan yang memiliki keterbatasan dalam hal jarak dan kemudahan untuk berintegrasi dengan sistem lain.

Sistem telerobotik termasuk sistem dua arah, dengan pertukaran data antara *client* dan *server* yang berlangsung dalam dua arah. Dari arah *client*, dikirimkan data

posisi yang diinginkan, sedangkan dari arah *server* di kembalikan hasil posisi robot yang telah dikerjakan bahkan dapat pula disertai dengan visualisasi gerakan robot maupun kondisi lingkungan di sekitarnya pada saat tertentu.

## 2.4 Otomasi dan Robot Industri

Otomasi didefinisikan sebagai teknologi yang berlandaskan pada aplikasi sistem mekanik, elektronik dan komputer. Otomasi sangat erat kaitannya dengan robot industri. Sebuah robot industri pasti memerlukan sebuah sistem otomasi agar dapat bergerak secara otomatis tanpa perlu bantuan seorang operator.

Robot industri adalah komponen utama dalam teknologi otomasi dalam dunia industri yang dapat berfungsi sebagai layaknya buruh/pekerja manusia dalam pabrik namun memiliki kemampuan bekerja yang terus-menerus tanpa lelah. Robot industri dapat diciptakan untuk menggantikan posisi-posisi pekerja dalam bagian produksi, seperti buruh dengan keahlian rendah hingga teknisi profesional dengan keahlian tertentu.

### 2.4.1 Otomasi

Secara umum otomasi dapat dibedakan dalam 4 kategori, yaitu:

- a. Otomasi tetap: mesin otomatis dibuat hanya untuk satu keperluan produksi saja, tidak dapat digunakan untuk produk lain. Sesuai untuk produksi massal dengan kecepatan tinggi. Investasi yang dikeluarkan pertama kali biasanya tinggi, namun biaya operasionalnya relatif rendah.
- b. Otomasi semi tetap: mesin dibuat untuk memproduksi atau menangan satu macam produk atau tugas, namun dalam beberapa parameter

- (ukuran, bentuk dan bagian produk) dapat diatur secara terbatas. Investasi awal termasuk cukup tinggi, karena mesin masih bersifat khusus. Robot yang mandiri termasuk dalam kategori ini.
- c. Otomasi fleksibel: Perangkat mesin yang dibuat dapat digunakan untuk berbagai produk, sistem otomasi lebih bersifat menyeluruh, bagianbagian produk dapat diproduksi pada waktu yang bersamaan dalam sistem otomasi ini. Yang termasuk dalam kategori ini misalnya FMS (*Flexible Automation System*) dan CIM (*Computer Integrated Manufacturing*). Robot adalah salah satu pendukung dalam kelompok otomasi ini.



Gambar 2.2 Robot Tangan dalam Sistem CIM

Pada gambar di atas adalah sebuah robot tangan yang dapat diprogram untuk mengambil barang dan mengangkatnya dari meja tetap di sebelah kiri kemudian meletakkannya ke atas ban berjalan (konveyor). Program dapat dengan mudah diubah melalui komputer konsol di sebelah kanannya. Foto ini diambil di Laboratorium CIM, Faculty of Mech. Engineering Universiti Teknologi Malaysia (UTM), Johor Bahru, Malaysia.

#### 2.4.2. Robot Industri

Robot industri yang diilustrasikan adalah robot tangan yang memiliki dua lengan (dilihat dari persambungan), dan pergelangan. Di ujung pergelangan dapat diinstal berbagai tool sesuai dengan fungsi yang diharapkan. Jika dipandang dari sudut pergerakan maka terdiri dari tiga pergerakan utama, yaitu badan robot yang dapat berputar ke kiri dan kanan, lengan yang masing-masing dapat bergerak rotasi ke arah atas dan bawah, dan gerak pergelangan sesuai dengan sifat tool. Komponen utamanya terdiri dari 4 bagian yaitu manipulator, sensor, aktuator, dan pengontrol.

Manipulator adalah bagian mekanik yang dapat difungsikan untuk memindah, mengangkat dan memanipulasi benda kerja. Sensor adalah komponen berbasis instrumentasi (pengukuran) yang berfungsi sebagai pemberi informasi tentang berbagai keadaan atau kedudukan dari bagianbagian manipulator. Output sensor dapat berupa nilai logika ataupun nilai analog. Dalam berbagai kasus dewasa ini penggunaan kamera sebagai sensor sudah menjadi lazim. Output perangkat kamera berupa citra (*image*) harus diubah terlebih dahulu ke besaran digital ataupun analog sesuai dengan kebutuhan. Kajian teknologi transformasi image ke bentuk biner (nilai acuan dalam proses perhitungan komputer) ini banyak dikaji dalam konteks terpisah, yaitu pengolahan citra (*image processing*).

Aktuator adalah komponen penggerak yang jika dilihat dari prinsip penghasil geraknya dapat dibagi menjadi 3 bagian, yaitu penggerak berbasis motor listrik (motor DC servo, stepper motor, motor AC, dsb.) penggerak pneumatik (berbasis kompresi gas: udara, nitrogen, dsb.), dan penggerak hidrolis (berbasis kompresi benda cair: minyak pelumas, dsb.).

Kontroler adalah rangkaian elektronik berbasis mikroprosesor yang berfungsi sebagai pengatur seluruh komponen dalam membentuk fungsi kerja. Tipe pengaturan yang bisa diprogramkan mulai dari prinsip pengurut (*sequencer*) yang bekerja secara *open loop* hingga prinsip umpan balik dengan melibatkan kecerdasan buatan.

### 2.4.3. Konfigurasi Manipulator

Secara klasik konfigurasi manipulator dapat dibagi dalam, 4 kelompok, yaitu polar, silindris, *cartesian* dan sambungan-lengan (*joint-arm*). Pada dasarnya badan sebuah robot dapat berputar ke kiri atau kanan. Sambungan pada badan dapat mengangkat atau menurunkan pangkal lengan secara polar. Lengan ujung dapat digerakkan maju-mundur secara translasi. Konfigurasi ini dikenal cukup tegar karena sambungan lengan dan gerakan maju-mundur memiliki Cara yang secara mekanik sangat kokoh. Kemampuan jangkauan ke atas dan bawah kurang bagus karena badan tidak mengangkat lengan secara vertikal, namun memiliki gerakan yang khas yaitu mampu memanipulasi ruang kerja yang berbentuk bola dengan algorithms gerak yang paling sederhana dibanding tipe konfigurasi yang lain.

Konfigurasi silinder mempunyai kemampuan jangkauan berbentuk ruang silinder yang lebih baik, meskipun sudut Ujung lengan terhadap garis penyangga tetap. Konfigurasi ini banyak diadopsi untuk sistem gantry atau crane karena strukturnya yang kokoh untuk tugas mengangkat beban. Pemasangan lengan Ujung yang segaris dengan badan dapat lebih menguntungkan kinematiknya menjadi lebih sederhana. Selain itu struktur secara keseluruhan bisa lebih kokoh.

Konfigurasi ini secara relatif adalah yang paling kokoh untuk tugas mengangkat beban yang berat. Struktur ini banyak dipakai secara permanen pada instalasi pabrik baik untuk mengangkat dan memindah barang produksi maupun untuk mengangkat peralatan-peralatan berat pabrik ketika melakukan kegiatan instalasi. *Crane* di galangan kapal juga banyak mengadopsi struktur ini.

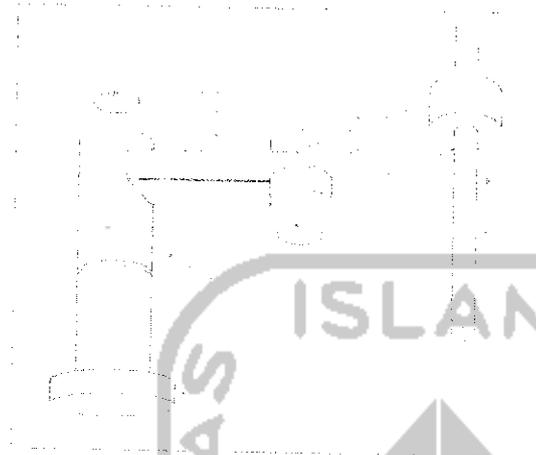
Pada aplikasi yang sesungguhnya biasanya struktur penyangga, badan dan lengan dibuat sedemikian rupa hingga tumpuan beban merata pada struktur. Misalnya, penyangga dipasang dari ujung ke ujung. Mekanik pengangkat di badan menggunakan sistem rantai dan sproket atau sistem belt. Pergerakan lengan dapat menggunakan sistem seperti rel di kiri-kanan lengan.

Konfigurasi keempat ialah struktur sambungan-lengan. Konstruksi ini adalah yang paling populer untuk tugas-tugas reguler di dalam pabrik, terutama untuk dapat melaksanakan fungsi layaknya pekerja pabrik, seperti mengangkat barang dari konveyor, mengelas, memasang komponen mur, baut pada produk, dan sebagainya. Dengan tool pergelangan yang khusus struktur lengan-sambungan ini cocok digunakan untuk menjangkau daerah kerja yang sempit dengan sudut jangkauan yang beragam.

Pada ujung lengan kedua dipasang pergelangan yang dapat berputar kiri-kanan. Robot yang secara keseluruhan berbobot sekitar 300 kg ini mampu mengangkat beban hingga 5 kg dan dapat dioperasikan pada temperatur yang relatif panas seperti ruang pengecatan bodi mobil, pengelasan, dan membantu proses *machining*.

Struktur robot ini juga populer dalam dunia penelitian. Banyak kajian kinematik dan dinamik robot mengambil contoh struktur sambungan-lengan. Kajian

dinamikanya menjadi lebih rumit karena unsur gravitasi bumi sangat berperan. Konfigurasi sambungan-lengan dapat dimodifikasi sedemikian hingga faktor gravitasi dapat diabaikan dalam analisis dinamikanya.



**Gambar 2.3** Manipulator Planar

Bentuk ini dikenal sebagai manipulator planar. Secara umum konfigurasinya dapat diilustrasikan seperti pada Gambar 2.3 di atas. Semua sambungan dirancang berputar horisontal, kecuali posisi ujung lengan (*end of effector*). Struktur ini banyak digunakan dalam kajian kinematik dan dinamik yang terutama yang berhubungan dengan analisis pembuktian dari suatu pendekatan atau teori kontrol yang dianggap baru. Dengan menghindari pengaruh gravitasi, orde dinamik dari sistem yang dikontrol relatif dapat ditekan. Salah satu pertimbangannya adalah bahwa aplikasi robotik di angkasa luar justru tidak perlu memasukkan unsur gravitasi dalam analisis. Dinamik hanya dipengaruhi oleh faktor dalaman yaitu bodi robot itu sendiri. Meski berkonstruksi planar, manipulator ini menjadi sangat kokoh karena sambungan dibuat memanjang dengan lengan yang sangat kaku. Robot ini banyak digunakan dalam pabrik elektronik seperti dalam hal pemasangan komponen.

Metode di atas sering dipakai untuk memanipulasi gerak lengan pertama dan kedua agar dapat bergerak selaras sehingga sudut sambungan kedua (posisi pergelangan) relatif tetap terhadap referensi badan robot ketika lengan pertama diayun. Sementara itu lengan kedua tetap dapat digerakkan secara independen. Untuk pergerakan berulang-ulang tanpa kontrol umpan balik (robot dioperasikan secara *loop* terbuka dan bekerja berdasarkan program sekuensial) cara ini sangat efektif terutama untuk memperoleh kesan gerakan yang lurus pada posisi pergelangan dengan cukup mengontrol satu sambungan saja.

#### **2.4.4 Sambungan 0 Lengan Robot**

Sebuah lengan robot memiliki 5 sambungan yang meliputi sambungan 0, sambungan 1 sampai dengan sambungan 4. Masing-masing sambungan memiliki fungsi masing-masing. Pergerakan dari masing-masing sambungan pun berbeda-beda, ada yang berputar sesuai sumbu vertikal maupu horizontal.

Sambungan 0 pada lengan robot merupakan bagian paling bawah dari sebuah lengan robot. Bagian ini berfungsi untuk mengatur sudut yang di inginkan dalam menggerakkan suatu lengan robot, sehingga sebuah lengan robot dapat mencapai sudut tertentu.

#### **2.4.5 Perkembangan Sistem Teleoperasi dan Telerobotik di Dunia Industri**

Fawaz, et al. (2009) menjelaskan bahwa penggunaan sistem robotik didunia industri sangat berkembang dengan pesat dalam beberapa tahun belakangan ini. Penggunaan teknologi robot ini banyak digunakan dalam berbagai bidang seperti dunia industri, kesehatan, perkebunan dan berbagai bidang lainnya. Bahkan di dalam

pengembangannya penggunaan robot dapat menggantikan peranan para pekerja terutama para pekerja yang bekerja dalam lingkungan kerja yang berbahaya.

Banyak aplikasi dari sistem robot yang kemudian dikembangkan, kebanyakan dari aplikasi tersebut berbasis simulasi dari keadaan yang sebenarnya. Robot yang menggunakan pengendali jarak jauh atau dikenal dengan teleoperasi pun berkembang dengan sangat baik beberapa tahun belakangan ini, yang kemudian dikenal dengan sistem telerobotik.

Salah satu perkembangan yang cukup menarik adalah penggunaan sistem telerobotik pada bidang kedokteran dan biologi. Di antara penggunaan sistem telerobotik pada bidang kedokteran dan biologi tersebut adalah digunakannya sistem ini untuk melakukan pembedahan jarak jauh dan untuk mengamati bakteri dari jarak jauh.

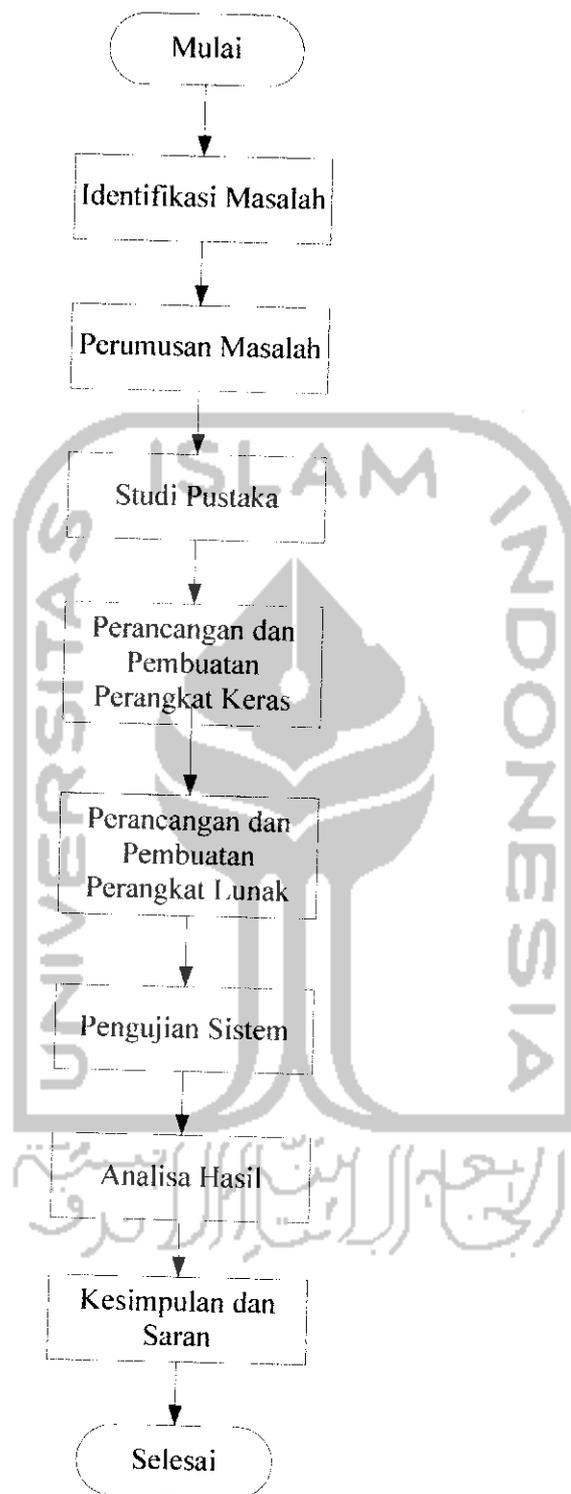
Salah satu pengembangan sistem telerobotik yang cukup berhasil dapat dilihat dari sistem *microsurgery* yang dikembangkan oleh para peneliti dari NASA-JPL dan MicroDexterity System, Inc (MDS). Pada penelitian ini, seorang dokter bedah menggerakkan sebuah simulator dari jarak jauh sambil mengamati kerja robot pembedah sistem dari jarak jauh atau teleoperasi. Sedangkan robot yang berada pada sisi jauh akan bergerak sesuai dengan perubahan posisi dari simulator tersebut.

Beberapa peristiwa penting sejarah dimulainya sebuah sistem pengendalian jarak jauh menggunakan internet dimulai pada tahun 1954, Raymond Goertz dan Thompson mengembangkan sistem master-slave yang menjadi cikal bakal dari sistem pengoperasian robot jarak jauh. Sekitar tahun 70-an muncul robot yang mampu

dikendalikan dengan menggunakan komputer diperkenalkan oleh Cincinnati-Milicron Corporation.

Pada tahun 1975 konsep jaringan komputer mulai dikembangkan untuk menghubungkan sebuah komputer dengan komputer lain. Perkembangan teknologi ini memicu munculnya sebuah jaringan komputer global yang kemudian dikenal dengan internet. Pada tahun 1994, Ken Goldberg dan rekan-rekan dari University of Southern California memperkenalkan sistem telerobotik pertama yang memungkinkan seorang pengguna untuk mengawasi atau membersihkan artifak yang tertimbun dalam sebuah kolam pasir melalui web. Proyek ini dinamakan *Mercury Project*. Sistem ini masih menggunakan teknologi HTTP standar, yaitu ISMAP yang saat itu baru berkembang.

Setahun kemudian Ken Goldberg dan rekan-rekannya mempublikasikan sebuah sistem baru yang merupakan pengembangan dari proyek mereka sebelumnya. Dalam proyek ini, seorang pengguna dapat mengamati keadaan dalam sebuah kebun kecil. Beberapa aplikasi baru dalam sistem telcrobotik bermunculan di antaranya oleh lembaga-lembaga antariksa, pertambangan, lingkungan, maupun dunia industri. Selain itu muncul pula gagasan untuk mendirikan sebuah laboratorium teknik yang segala aktifitasnya dilakukan melalui internet.



**Gambar 3.1** Diagram Alir Penelitian

### 3.2 Perumusan Masalah

Untuk mendapatkan gambaran yang jelas mengenai masalah yang akan diselesaikan, maka masalah yang telah diidentifikasi perlu untuk dirumuskan. Pada langkah ini, masalah akan dirumuskan dengan jelas, dan langkah-langkah penyelesaian masalah akan disusun berdasarkan rumusan masalah yang telah dibuat.

### 3.3 Studi Pustaka

Langkah berikutnya adalah melakukan studi pustaka. Studi pustaka ini dilakukan untuk mengetahui posisi penelitian ini dibandingkan dengan penelitian lain yang telah dilakukan oleh para peneliti sebelumnya. Ini dilakukan untuk mengetahui kontribusi akademik yang ada dalam penelitian ini. Selain itu studi pustaka juga dilakukan untuk mendapatkan gambaran mengenai langkah-langkah teknis yang akan ditempuh selama melakukan penelitian ini. Adapun studi pustaka yang dilakukan meliputi robotika dalam industri, telerobotik, teleoperasi, mikrokontroler AVR, komunikasi serial, pemrograman internet, *Analog to Digital Converter (ADC)* dan motor servo.

### 3.4 Perancangan dan Pembuatan Perangkat Keras

Pada tahap ini akan dilakukan pembangunan prototip sambungan 0 pada sistem lengan telerobot industri. Prototip yang dibangun meliputi sistem sambungan 0 pada lengan telerobot dan juga sistem kontrolnya. Sistem kontrol dibangun menggunakan mikrokontroler sebagai prosesornya dan menggunakan potensiometer sebagai sensor posisi dari lengan robotnya.

### 3.5 Perancangan dan Pembuatan Perangkat Lunak

Pada tahap ini akan dilakukan perancangan perangkat lunak yang akan mendukung sistem telerobotik. Perangkat lunak terbagi atas dua macam yaitu pembuatan perangkat lunak yang akan di *download* pada mikrokontroler ATMEGA16 yang merupakan prosesor dari sistem kontrol lengan telerobot. Perangkat lunak ini dibangun menggunakan bahasa pemrograman C. Perangkat lunak lainnya adalah antar muka pengguna untuk menjalankan sistem telerobot ini. Ada dua macam antar muka yang akan dibangun, yaitu antar muka yang akan diinstal dalam komputer *server* dan antar muka yang akan diinstal dalam komputer *client*. Komputer *client* terhubung dengan komputer *server* melalui jaringan internet.

### 3.6 Pengujian Sistem

Pada tahapan ini dilakukan pengujian sistem guna mengetahui sejauh mana sistem dapat bekerja dengan baik dan mengetahui kekurangan dari sistem tersebut. Pengujian dilakukan untuk mengetahui keberhasilan sistem komunikasi data serial dari komputer *server* ke sistem telerobot, dari komputer *client* ke komputer *server* melalui jaringan internet dan juga akurasi dari gerakan *join 0* pada lengan telerobot yang dibangun.

### 3.7 Analisis Hasil

Analisis dilakukan dengan tujuan untuk mengetahui kelebihan dan kelemahan sistem sehingga dapat digunakan sebagai dasar dalam perbaikan sistem. Analisa akan difokuskan kepada sistem pertukaran data prototip yang di bangun yang menggunakan

sistem protokol TCP/IP yang mudah digunakan. Selain itu analisa juga dilakukan penggunaan potensiometer sebagai sensor dari prototip sambungan 0 lengan telcrobot.

### 3.8 Kesimpulan dan Saran

Tahap terakhir ini adalah penarikan kesimpulan berdasarkan pengujian sistem dan analisis hasil pengujian. Kesimpulan ini sangat berguna dalam merangkum hasil akhir dari suatu penelitian. Bagian ini juga dilengkapi dengan saran-saran untuk menyempurnakan hasil penelitian.

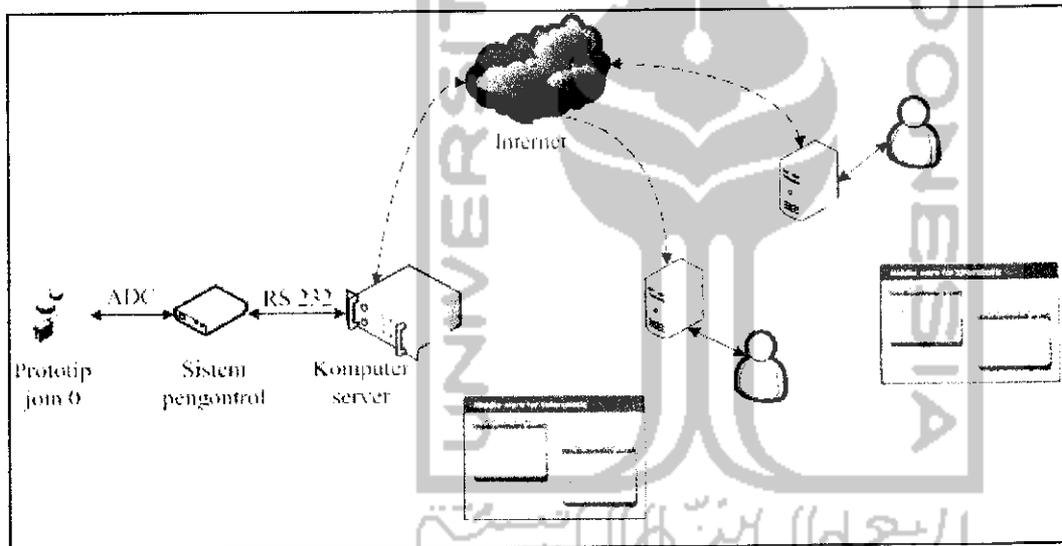


## BAB IV

### PENGUMPULAN DAN PENGOLAHAN DATA

#### 4.1 Pengumpulan Data

Pada tahap ini akan dilakukan pengumpulan data-data yang diperlukan dalam pembangunan prototip telerobotik sambungan 0 lengan telerobot industri dan sistem kontrolnya. Sebelum dilakukan pengumpulan dan pengolahan data, terlebih dahulu akan dijelaskan mengenai arsitektur dari prototip yang akan dibuat. Adapun arsitektur dari prototip yang akan dibuat adalah seperti yang ditunjukkan oleh Gambar 4.1.



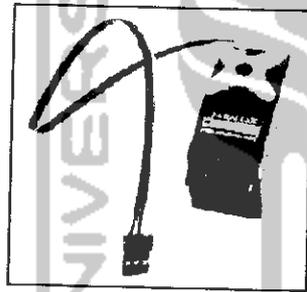
**Gambar 4.1** Arsitektur dari Prototip yang Dibangun

Data-data yang diperlukan untuk membangun prototip seperti yang ditunjukkan dalam Gambar 4.1 di atas adalah komponen motor servo beserta karakteristiknya, sistem kontrol dan juga karakteristik dari prosesor sistem kontrol yang digunakan. Dalam hal ini, prosesor sistem kontrol ialah menggunakan pengontrol mikro berjenis ATmega16. Selain itu, data lain yang diperlukan adalah sistem kerja komunikasi data serial dari

komputer ke sistem kontrol prototip sambungan 0 lengan telorobot industri. Data lainnya adalah sistem pemrograman internet untuk mewujudkan sistem teleoperasi yang akan dibangun dan juga pemrograman pengontrol mikro sebagai prosesor alat pengontrol sambungan 0. Selain itu data lain yang turut mendukung dalam pembangunan prototip sambungan 0 lengan telorobot industri adalah komponen-komponen mekanik untuk membuat prototip tersebut. Data yang dikumpulkan akan diterangkan secara lebih terperinci pada sub bab berikut.

#### 4.1.1 Motor Servo

Motor servo yang digunakan dalam penelitian ini adalah motor servo dengan putaran  $180^{\circ}$  dan torsi putar 3,40 kg-cm. Gambar motor servo yang digunakan adalah seperti yang ditunjukkan dalam Gambar 4.2.



Gambar 4.2 Motor Servo

#### 4.1.2 Sistem Kontrol

Sistem kontrol untuk prototip sambungan 0 sistem lengan telorobot industri yang dibangun adalah sistem kontrol berdasarkan pengontrol mikro ATmega16. Susunan kaki pengontrol mikro ATmega16 adalah seperti yang ditunjukkan dalam Gambar 4.3.

(XCK/T0; PB0	1	40	PA0 (ADC0)
(T1; PB1	2	39	PA1 (ADC1)
(NT2/AIN0; PB2	3	38	PA2 (ADC2)
(OC0/AIN1; PB3	4	37	PA3 (ADC3)
(SC; PB4	5	36	PA4 (ADC4)
(MOS; PB6	5	35	PA5 (ADC5)
(MISO; PB6	7	34	PA6 (ADC6)
(SCK; PB7	3	33	PA7 (ADC7)
RESET	9	32	AREF
VCC	10	31	GND
GND	11	30	AVCC
XTAL2	12	29	PC7 (TOSC2)
XTAL1	13	28	PC8 (TOSC1)
(RXD; PD0	14	27	PC5 (TDI)
(TXD; PD1	15	26	PC4 (TDO)
(INT0; PD2	16	25	PC3 (TMS)
(INT1; PD3	17	24	PC2 (TCK)
(OC1B; PD4	18	23	PC1 (SDA)
(OC1A; PD5	19	22	PC0 (SCL)
(ICP; PD6	20	21	PD7 (OC2)

Gambar 4.3 Susunan kaki Mikrokontroler ATmega16

Untuk dapat bekerja dengan baik, pengontrol mikro perlu diintegrasikan dengan rangkain eksternal. Rangkain eksternal ini berfungsi untuk membangkitkan detak agar pengontrol mikro dapat melakukan pencacahan untuk melakukan komputasi. Selain itu, rangkaian eksternal ini berfungsi sebagai antara muka pengontrol mikro dalam melakukan komunikasi serial dengan komputer server dan juga untuk menggerakkan motor servo. Rangkaian eksternal ini terdiri atas komponen-komponen elektronik seperti papan PCB, transistor, kapasitor, resistor, serta IC 7805 sebagai penurun tegangan. Untuk melindungi rangkaian dari pengaruh luar digunakan acrylic. Adapun rangkaian lengkap dari sistem kontrol yang dibangun adalah ditunjukkan dalam Gambar 4.4





Sistem jabat tangan yang digunakan untuk mengirimkan hasil pergerakan lengan 0 ke komputer dapat digambarkan seperti dalam Gambar 4.7

```

1. Data "L" mengambarkan bahwa sambungan 0 bergerak ke
kiri, apabila sambungan 0 bergerak ke kanan maka data yang
dikirimkan adalah "R".
2. Data "XXX" adalah sudut pergerakan sambungan 0
sedangkan data "#" adalah tanda akhir dari data.

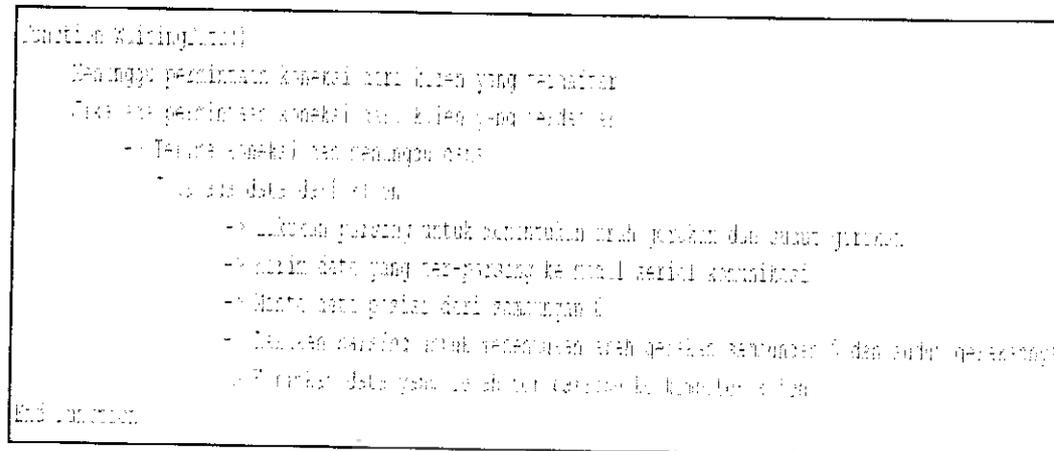
```

**Gambar 4.7** Kode Bayangan Untuk Pergerakan Lengan Telerobot ke Komputer

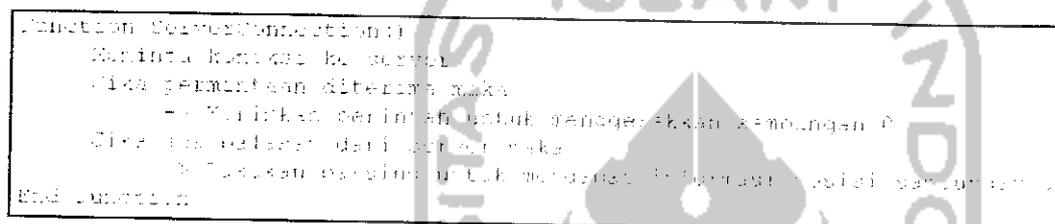
Dari Gambar 4.7 di atas, data "L" menggambarkan bahwa sambungan 0 bergerak ke kiri, apabila sambungan 0 bergerak ke kanan maka data yang dikirimkan adalah "R". Data "XXX" adalah sudut pergerakan sambungan 0 sedangkan data "#" adalah tanda akhir dari data.

#### 4.1.4 Sistem Pemrograman Internet

Sistem pertukaran data melalui internet dilakukan melalui protokol TCP/IP. Secara teknis, sistem pertukaran data ini dilakukan dengan pemrograman socket yang terpasang dalam sebuah komputer melalui sistem operasi Windows. Untuk menjamin pertukaran data dapat dilakukan secara sukses, maka diperlukan sebuah sistem jabat tangan baik di komputer *server* maupun di komputer klien. Adapun jabat tangan yang digunakan dalam pemrograman socket ini baik di komputer server maupun di komputer klien adalah seperti yang ditunjukkan dalam Gambar 4.8



(a)



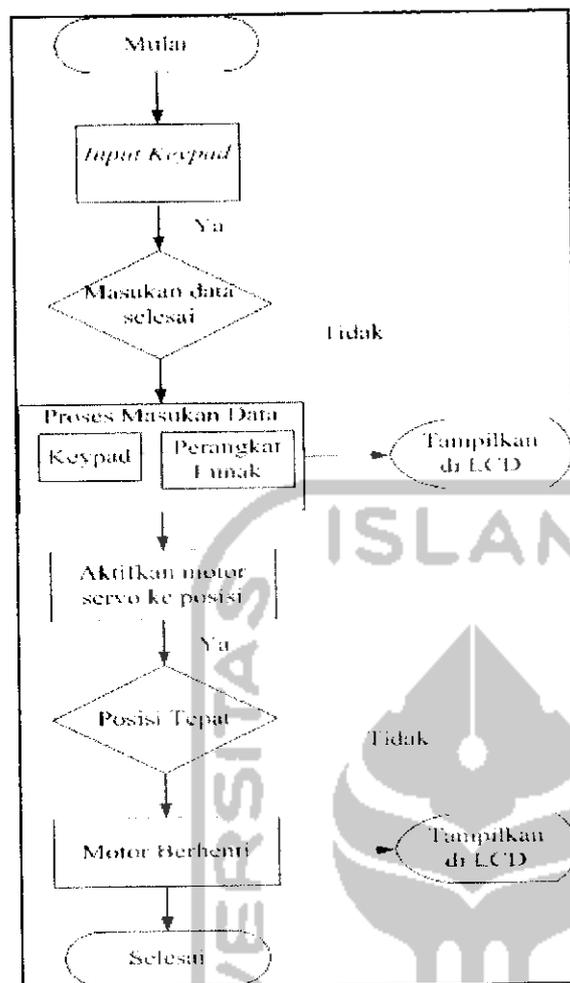
(b)

**Gambar 4.8 (a).** Proses Jabat Tangan di Komputer Server  
**(b).** Sistem Jabat Tangan di Komputer Klien

#### 4.1.5 Pemrograman Pengontrol Mikro

Sebuah IC pengontrol mikro memerlukan sebuah program untuk menjalankan suatu perintah secara berulang. Program ini digunakan untuk menggerakkan motor servo, menerima perintah dari *keypad*, mengubah isyarat analog ke digital, menjalankan LCD dan melakukan komunikasi serial dengan komputer. Carta alir program yang dimasukkan dalam IC pengontrol mikro ditunjukkan oleh Gambar 4.9

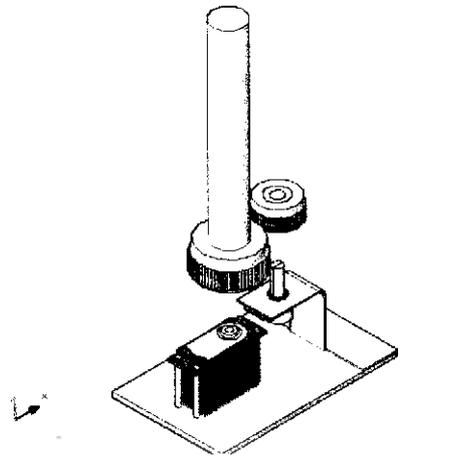




**Gambar 4.9** Perancangan Perangkat Lunak pada Mikrokontroler

#### 4.1.6 Pembuatan Mekanik Lengan Telerobot

Mekanik lengan telerobot dibangun dari suatu mekanisme poros yang dihubungkan dengan motor servo dan potensiometer sebagai sensor posisinya. Sebagai lengan telerobotnya dibangun dari suatu bahan *acrylic* karena ringan sehingga tidak menjadi faktor dinamis yang perlu dipertimbangkan dalam menjalankan motor servo. Gambar 4.9 menunjukkan mekanik lengan telerobot yang dibangun.



**Gambar 4.10** Rancangan Mekanik Lengan Telerobot

Pada rancangan diatas antara lengan telerobot, motor servo dan potensiometer memiliki korelasi yang baik sehingga dapat bergerak bersamaan. Lengan telerobot dapat digerakkan dengan menggunakan motor servo yang terhubung dengan roda gigi yang bergerak linier dengan pergerakan potensiometer melalui mekanisme roda gigi. Gambar 4.11 menunjukkan keseluruhan dari prototip sambungan 0 lengan telerobot.



**Gambar 4.11** Prototip Sambungan 0 Lengan Telerobot

#### 4.2 Pengolahan Data

Dari data-data yang ada kemudian diolah sedemikian rupa yang bertujuan merancang sebuah sistem yang dapat di implementasikan serta dapat digunakan sesuai dengan tujuan dari pembangunan sistem. Hasil pengolahan data ini berupa sebuah

prototip sambungan 0 lengan telcrobot industri, sebuah alat pengontrol, sebuah perangkat tengah dan sebuah perangkat lunak aplikasi klien untuk menjalankan prototip dari jarak jauh. Setelah prototip dibuat, untuk memvalidasi hasilnya, maka sebuah pengujian sistem juga dijalankan.

#### 4.2.1 Pengujian Sistem

Dalam perancangan suatu sistem sangat mungkin terjadi kesalahan. Untuk itu diperlukan pengujian kinerja sistem untuk memastikan suatu sistem berjalan seperti yang diharapkan. Dengan melakukan pengujian diharapkan semua kesalahan tersebut dapat ditemukan untuk diperbaiki.

##### a. Pengujian Sistem Kontrol

Pengujian sistem kontrol ini bertujuan agar mengetahui posisi sudut lengan telcrobot sesuai dengan apa yang ditampilkan pada tampilan sistem kontrol. Pengujian dilakukan beberapa kali dan memasukkan data sudut secara acak. Tabel 4.1 menunjukkan pengujian sistem kontrol.

**Tabel 4.1** Pengujian Sistem Kontrol

Masukan Sudut	Tampilan pada Sistem Kontrol
17°	17°
56°	56°
115°	115°
15°	15°
75°	75°
156°	156°
88°	88°
96°	96°
113°	113°
13°	13°
98°	98°
168°	168°
58°	58°
19°	19°
135°	135°

Sebagai contoh, sambungan 0 lengan telerobot diuji untuk bergerak ke kanan dengan sudut  $17^\circ$ . Gambar 4.12 menunjukkan hasil pengujian tersebut.



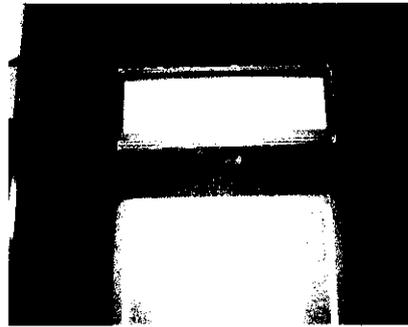
**Gambar 4.12** Tampilan Pengujian Pertama pada Sistem Kontrol

Setelah melakukan pengujian pertama, di lanjutkan dengan pengujian kedua yaitu memasukkan data sudut  $56^\circ$ . Gambar 4.13 menunjukkan tampilan pengujian kedua.



**Gambar 4.13** Tampilan Pengujian Kedua pada Sistem Kontrol

Pengujian ketiga dilakukan yaitu dengan memasukkan data sudut  $115^\circ$ . Gambar 4.14 menunjukkan tampilan pada sistem kontrol pada pengujian terakhir.



**Gambar 4.14** Tampilan Pengujian Ketiga pada Sistem Kontrol

Setelah melakukan beberapa kali pengujian di dapatkan bahwa antara masukan sudut dan tampilan yang ada pada sistem kontrol sambungan 0 lengan telerebot tidak ada perbedaan nilai antara data masukan dengan nilai yang di tampilkan.

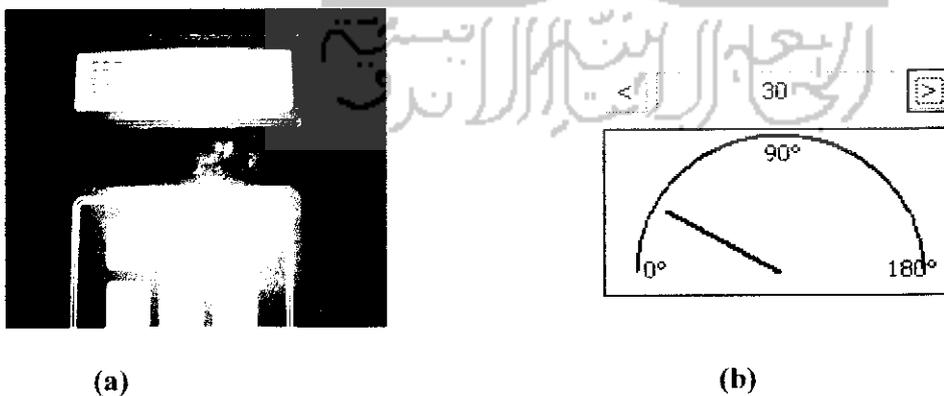
**b. Pengujian Sistem Kontrol dengan Komputer *Server***

Pengujian kali ini bertujuan untuk mengetahui apakah sistem jabat tangan antara sistem kontrol sambungan 0 lengan telerebot dan perangkat tengah berjalan dengan baik. Pengujian dilakukan beberapa kali dengan memasukkan data-data yang berbeda. Tabel 4.2 menunjukkan pengujian sistem kontrol sambungan 0 lengan telerebot dan perangkat tengah.

Tabel 4.2 Pengujian Sistem Kontrol dan Perangkat Tengah

Masukan Sudut	Tampilan pada Sistem Kontrol	Tampilan pada Komputer Server
30°	30°	30°
105°	105°	105°
180°	180°	180°
15°	15°	15°
45°	45°	45°
98°	98°	98°
135°	135°	135°
75°	75°	75°
13°	13°	13°
175°	175°	175°
39°	39°	39°
150°	150°	150°

Setelah memasukkan data pertama yaitu data sudut 30° tampilan pada sistem kontrol dan perangkat tengah dapat ditunjukkan pada Gambar 4.15.



Gambar 4.15 (a). Tampilan pada Sistem Kontrol

(b). Tampilan pada Perangkat Tengah

Kemudian pengujian dilanjutkan dengan memasukkan data masukan sudut yang kedua yaitu  $105^\circ$ . Gambar 4.16 menunjukkan tampilan pada masing-masing bagian yang di uji.



**Gambar 4.16** (a). Tampilan pada Sistem Kontrol  
(b). Tampilan pada Perangkat Tengah

Pengujian selanjutnya dilakukan dengan memasukkan data masukan sudut yang kedua yaitu  $180^\circ$ . Gambar 4.17 menunjukkan tampilan pada sistem kontrol dan perangkat tengah.



**Gambar 4.17** (a). Tampilan pada Sistem Kontrol  
(b). Tampilan pada Perangkat Tengah

Setelah melakukan beberapa kali pengujian didapatkan bahwa tampilan pada sistem kontrol sambungan 0 lengan telorobot dan perangkat tengah tidak ada perbedaan nilai.

#### 4.2.2 Biaya Pembuatan

Dalam perancangan prototip sambungan 0 ini di butuhkan biaya untuk melakukan pembuatan alat, berikut ini data biaya tiap komponen yang dibutuhkan :

Tabel 4.3 Daftar Harga Komponen

No.	Daftar Komponen	Jumlah	Harga
1.	Mikrokontroler	1	Rp. 42.500
2.	Resistor 470 Ohm	1	Rp. 30
3.	Resistor 330 Ohm	1	Rp. 30
4.	Potensiometer	1	Rp. 3.000
5.	Motor Servo	1	Rp. 300.000
6.	PCB Fiber	1	Rp. 21.000
7.	LCD	1	Rp. 67.000
8.	Transistor TIP 41	1	Rp. 3.000
9.	<i>Acrylic</i>	1	Rp. 10.000
10.	Kapsitor	3	Rp. 5.000
11.	IC 7805	1	Rp. 3.000
12.	Dioda	1	Rp. 200
<b>JUMLAH</b>			<b>Rp. 454.760</b>

Biaya ini diperlukan untuk mengetahui nilai ekonomis pembuatan dari alat yang dibuat, jika dibandingkan dengan melakukan pembelian. Dari referensi yang ada diperoleh bahwa harga termurah alat yang sejenis adalah Rp. 3.660.000, jadi sistem yang dibangun memiliki nilai ekonomis yang jauh lebih murah di bandingkan jika melakukan pembelian.



## BAB V

### PEMBAHASAN

Berdasarkan pengujian prototip sambungan 0 berbiaya rendah untuk lengan telerobot industri yang telah dilakukan pada bab sebelumnya, maka pada bab ini akan dibahas mengenai kelebihan dan kekurangan dari sistem ini. Diharapkan dengan mengetahui kelebihan dan kekurangan yang ada pada sistem ini dapat menjadi perhatian bagi peneliti selanjutnya untuk bisa mengembangkan sistem menjadi lebih handal.

#### 5.1 Kelebihan Prototip Sambungan 0 Lengan Telerobot

Prototip sambungan 0 ini memiliki kelebihan yaitu tidak memerlukan protokol yang rumit untuk melakukan pertukaran data. Dalam melakukan sistem pertukaran datanya, prototip ini menggunakan sistem protokol TCP/IP yang mudah untuk di gunakan.

Selain itu prototip ini berbiaya rendah karena komponen-komponen pendukungnya merupakan komponen yang umum di gunakan dalam keseharian serta penggunaan pengontrol mikro sebagai pengganti PLC yang harganya mahal tanpa mengurangi kemampuannya sebagai otak dari sistem.

Prototip ini menggunakan potensiometer sebagai sensor posisi sudut. Penggunaan potensiometer ini dinilai cukup akurat dalam menentukan posisi sudut dari prototip sambungan 0 lengan telerobot industri.

Prototip yang di bangun ini juga merupakan prototip berbiaya rendah karena untuk membuat prototip ini memerlukan biaya Rp. 454.760. Sedangkan harga alat sejenis yang ada dipasaran yaitu Rp. 3.660.000.

## 5.2 Kekurangan Prototip Sambungan 0 Lengan Telerobot

Prototip yang di bangun ini masih memiliki kelemahan yaitu untuk beberapa sudut tertentu terkadang motor servo tidak langsung berhenti ke posisi sudut yang di inginkan. Jadi terkadang hal ini membutuhkan sedikit waktu untuk menunggu motor servo berhenti ke posisi yang di inginkan. Hal ini disebabkan karena program dalam pengontrol mikro menggunakan teknik *looping* untuk mencapai sudut yang diinginkan. Apabila sudut aktual sambungan 0 lengan telerobot lebih besar dari sudut yang dimasukkan oleh pengguna maka sambungan 0 akan bergerak ke kiri agar sudutnya sesuai dengan yang diinginkan pengguna. Namun apabila sudut aktual lebih kecil dari sudut yang dimasukkan oleh pengguna maka sambungan 0 akan bergerak ke kanan sesuai dengan yang diinginkan pengguna.

## BAB VI

### KESIMPULAN DAN SARAN

#### 6.1 Kesimpulan

Dari hasil pembangunan prototip sambungan 0 untuk lengan telerobot industri berbiaya rendah dapat diambil kesimpulan bahwa prototip telah dapat berfungsi sesuai tujuan untuk membuat suatu sistem pengendalian terhadap posisi lengan robot berdasarkan sensor posisi. Sudut perputaran yang dapat dikendalikan adalah sejauh  $180^{\circ}$  sesuai dengan mekanisme lengan robot yang hanya dapat berputar sejauh  $180^{\circ}$ . Selain itu prototip sambungan 0 untuk lengan telerobot industri yang di bangun merupakan sistem kontrol berbiaya rendah karena jika dibandingkan dengan alat yang sejenis dipasaran harganya jauh lebih murah.

#### 6.2 Saran

Berdasarkan pembuatan sistem yang telah selesai dilakukan, maka didapatkan data-data yang dapat dijadikan bahan pengembangan produk menjadi lebih baik. Berikut ini beberapa saran yang dapat diberikan sebagai pengembangan yaitu sebagai berikut :

1. Dapat dibuat lengan robot yang lebih baik dengan mekanisme lebih dari 1 sambungan.
2. Robot dapat dikembangkan menjadi mekanisme otomatisasi dalam dunia industri yang sesungguhnya.

## DAFTAR PUSTAKA

- Alvares, A. J, Ferreira, J. C. E. (2006). WebTurning: Teleoperation of a CNC Turning Center through the Internet. *Journal of Materials Processing Technology* 179 251-259
- Bambang, R. T. (2007). Development of Architectures for Internet Telerobotics Systems. *Journal of Bionic Engineering* 4 (2007) 291-297
- Bejo, Agus. (2008). C dan AVR. Graha Ilmu, Yogyakarta
- Budiharto, W. (2005), Perancangan Sistem dan Aplikasi Mikrokontroler, Jakarta
- Cooper, William David (1985), Instrumentasi Elektronika dan Teknik Pengukuran, Erlangga, Jakarta
- Fawaz, K, Merzouki, R, Ould-Bouamama, B. (2009) . Model Based Real Time Monitoring for Collision Detection of an Industrial Robot. *Journal of Mechatronics* 19 (695-704)
- Hartanto, A.A dan Purbo, O.W (2001), Teleoperasi Menggunakan Internet, Jakarta
- Malvino, (1996), Prinsip - Prinsip Elektronika, Erlangga, Jakarta
- Manzoor, U dan Nefti, S. (2009). An agent based system for activity monitoring on network – ABSAMN. *Journal of Expert Systems with Applications* 36 (10987-10994)
- Pitowarno, Endro,(2006), Robotika Desain Kontrol dan Kecerdasan Buatan, Andi Offset, Jakarta

Wardhana, Lingga, (2006) Belajar Mikrokontroler AVR ATMEGA8535, Andi

Offset, Yogyakarta



LAMPIRAN



LISTING PROGRAM PENGONTROL  
MIKRO



```
// Alphanumeric LCD Module functions
```

```
#asm
```

```
.equ __lcd_port=0x15
```

```
#endasm
```

```
#include <lcd.h>
```

```
#define SERVO PORTD.2
```

```
#define LED_L1 PORTD.4
```

```
#define LED_R1 PORTD.6
```

```
#define LED_L2 PORTD.5
```

```
#define LED_R2 PORTD.7
```

```
char buff [33];
```

```
bit flag,idle,set;
```

```
unsigned char i,j,l,m,key,code;
```

```
unsigned char edit,pas,count,data,arah,gerak;
```

```
unsigned int seting,sudut,temp,ofset_low,ofset_high;
```

```
flash unsigned int ofset=3;
```

```
ceprom unsigned int setpoint;
```

```
#define RXB8 1
```

```
#define TXB8 0
```

```
#define UPE 2
```



```

#define OVR 3
#define FE 4
#define UDRE 5
#define RXC 7

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

// USART Receiver buffer
#define RX_BUFFER_SIZE 32
char rx_buffer[RX_BUFFER_SIZE];
unsigned char rx_wr_index,rx_rd_index,rx_counter;
bit rx_buffer_overflow;

// USART Receiver interrupt service routine
#pragma savereg-
interrupt [USART_RXC] void usart_rx_isr(void)
{
    char status,data;

    #asm
        push r26
        push r27
        push r30

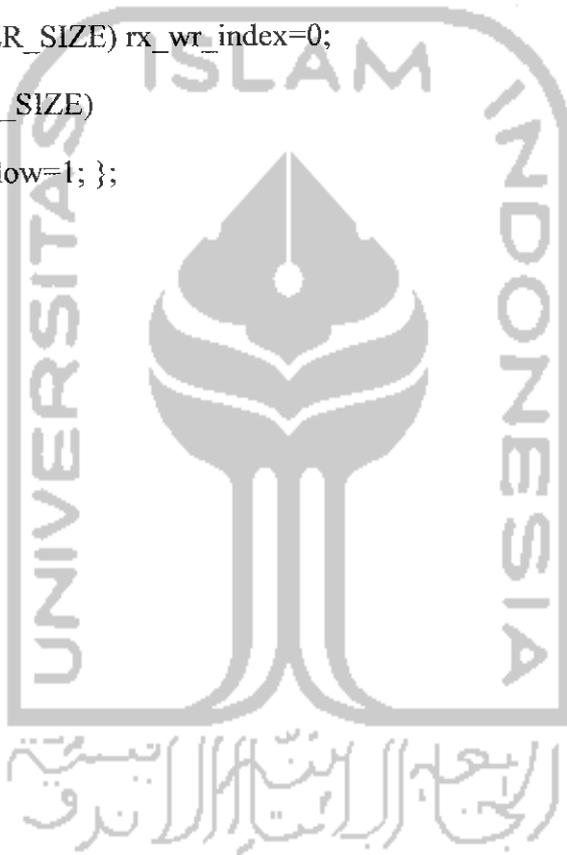
```



```

push r31
in r26,sreg
push r26
#endasm
status=UCSRA; data=UDR;
if((status & (FRAMING_ERROR | PARITY_ERROR | DATA_OVERRUN))==0)
{ rx_buffer[rx_wr_index]=data;
if(++rx_wr_index == RX_BUFFER_SIZE) rx_wr_index=0;
if(++rx_counter == RX_BUFFER_SIZE)
{ rx_counter=0; rx_buffer_overflow=1; };
};
#asm
pop r26
out sreg,r26
pop r31
pop r30
pop r27
pop r26
#endasm

```



```

}
#pragma savereg+

```

```

#ifndef _DEBUG_TERMINAL_IO_

```

```

#define _ALTERNATE_GETCHAR_

```

```

#pragma used+

```

```

char getchar(void)
{
    while (rx_counter==0);
    data=rx_buffer[rx_rd_index];
    if(++rx_rd_index == RX_BUFFER_SIZE) rx_rd_index=0;
    #asm("cli")
    --rx_counter;
    #asm("sei")
    return data;
}

#pragma used-
#endif
#include <stdio.h>

#define ADC_VREF_TYPE 0x40 // Read the AD conversion result
unsigned int read_adc(unsigned char adc_input)
{
    ADMUX=adc_input|ADC_VREF_TYPE; ADCSRA|=0x40;
    while ((ADCSRA & 0x10)==0); ADCSRA|=0x10; return ADCW;
}

// Timer 1 overflow interrupt service routine
interrupt [TIM1_OVF] void timer1_ovf_isr(void)
{
    TCNT1=0xFB1D; // 0xFB1D; 20 msec
}

```

```

if ( sct==1 && gerak!='N' )
{
  SERVO=1;
  if ( gerak=='L' ){ OCR1A=0xFB9A; } // 0xFB9A; 19 m sec
  else { OCR1A=0xFB40; } // 0xFB5B; 18 m sec
} else SERVO=0;
}

```

// Timer 1 output compare A interrupt service routine

```

interrupt [TIM1_COMPA] void timer1_compa_isr(void)

```

```

{
  SERVO=0; }

```

```

unsigned int kalkulasi ( unsigned int data )

```

```

{
  if ( l=='L' )

```

```

    { if ( setpoint > data )

```

```

      { data=(unsigned int) setpoint-data; } else data=0; }

```

```

else { data=(unsigned int) setpoint+data; if ( data >= 180 ) data=180; };

```

```

return data;
}

```

```

void new_set()

```

```

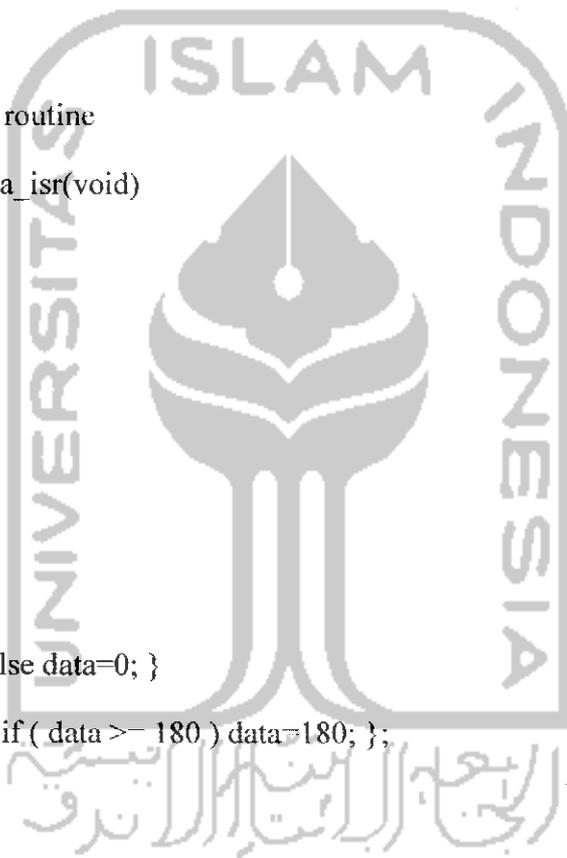
{
  arah=m; seting=temp;

```

```

  l=m; setpoint=kalkulasi( temp );
}

```



```

if ( arah=='R')
    sprintf(buff," INPUT DATA \n RIGHT >> %3d%c",temp,0xDF );
else sprintf(buff," INPUT DATA \n LEFT << %3d%c",temp,0xDF );

lcd_clear(); lcd_puts(buff);

printf( "SET:%c",0x22 ); putchar( arah );

printf( ",%03d%c#\r\n",seting,0x22 );

if ( arah=='L') { LED_L2=1; LED_R2=0; }
else { LED_L2=1; LED_R2=0; };
}

```

```
void keypad (void);
```

```
void main(void)
```

```
{
```

```
PORTA=0x00; DDRA=0x00;
```

```
// Port A initialization
```

```
PORTB=0xFF; DDRB=0x0F;
```

```
// Port B initialization
```

```
PORTC=0x00; DDRC=0x00;
```

```
// Port C initialization
```

```
PORTD=0x00; DDRD=0xFE;
```

```
// Port D initialization
```

```
TCCR0=0x00; TCNT0=0x00;
```

```
// Timer 0: Stopped
```

```
OCR0=0x00;
```

```
TCCR1A=0x00; TCCR1B=0x03;
```

```
// Timer 1 Clock: 62.500 kHz
```

```
TCNT1H=0xFB; TCNT1L=0x1D;
```

```
// 20 m sec
```

```
OCR1AH=0x00; OCR1AL=0x00;
```



```

OCR1BH=0x00; OCR1BL=0x00;

ASSR=0x00; TCCR2=0x00;           // Timer 2: Stopped
TCNT2=0x00; OCR2=0x00;

MCUCR=0x00; MCUCSR=0x00;        // External Interrupt(s): Off

//TIMSK=0x1C;
TIMSK=0x14;                       // Timer(s) initialization

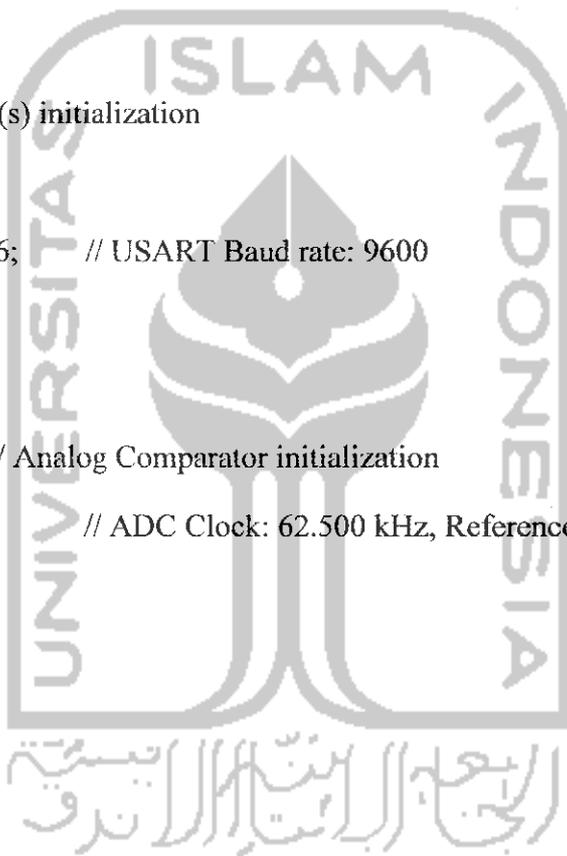
UCSRA=0x00; UCSRB=0x98; UCSRC=0x86; // USART Baud rate: 9600
UBRRH=0x00; UBRRL=0x19;

ACSR=0x80; SFIOR=0x00;          // Analog Comparator initialization
ADMUX=ADC_VREF_TYPE;             // ADC Clock: 62.500 kHz, Reference:
AVCC pin
ADCSRA=0x86; SFIOR&=0xEF;

SERVO=0;
gerak='N'; arah='L'; sudut=0;
if ( setpoint>=180 ) setpoint=180;
lcd_init(16); lcd_clear();
lcd_putsf(" ROBOTIC ARM\n CONTROLER");

LED_L1=1; delay_ms(500); LED_L1=0;

```



```
LED_L2=1; delay_ms(500); LED_L2=0;
LED_R1=1; delay_ms(500); LED_R1=0;
LED_R2=1; delay_ms(500); LED_R2=0;
```

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

```
while (1)
```

```
{
```

```
  sudut = (unsigned int)( read_adc(0)* 0.22597293904310224578044357650937 );
```

```
  if ((setpoint+offset)>=190 ) offset_high=190;
```

```
  else offset_high=setpoint+offset;
```

```
  if ( setpoint<=offset ) offset_low=0;
```

```
  else offset_low=setpoint-offset;
```

```
    if ( gerak =='R' ) sprintf(buff,"ROBOT ==> : %3d%c",sudut,0xDF );
```

```
    else if ( gerak =='L' ) sprintf(buff,"ROBOT <== : %3d%c",sudut,0xDF );
```

```
    else      sprintf(buff,"ROBOT --- : %3d%c",setpoint,0xDF );
```

```
    lcd_gotoxy(0,1); lcd_puts(buff); count=0;
```

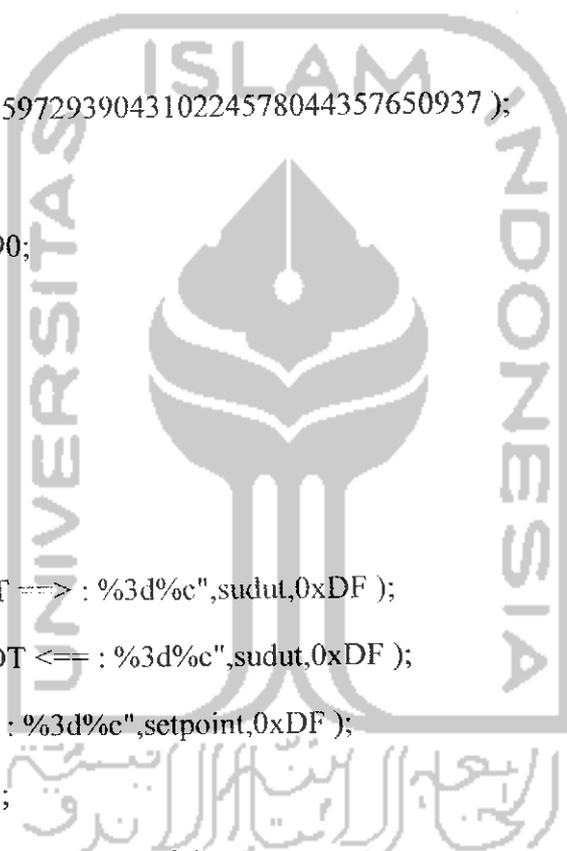
```
    printf( "RBT:%c%03d,",0x22,setpoint ); putchar( gerak );
```

```
    if ( gerak=='N' ) printf( ",%03d%c#\r\n",setpoint,0x22 );
```

```
    else printf( ",%03d%c#\r\n",sudut,0x22 );
```

```
    delay_ms( 5 );
```

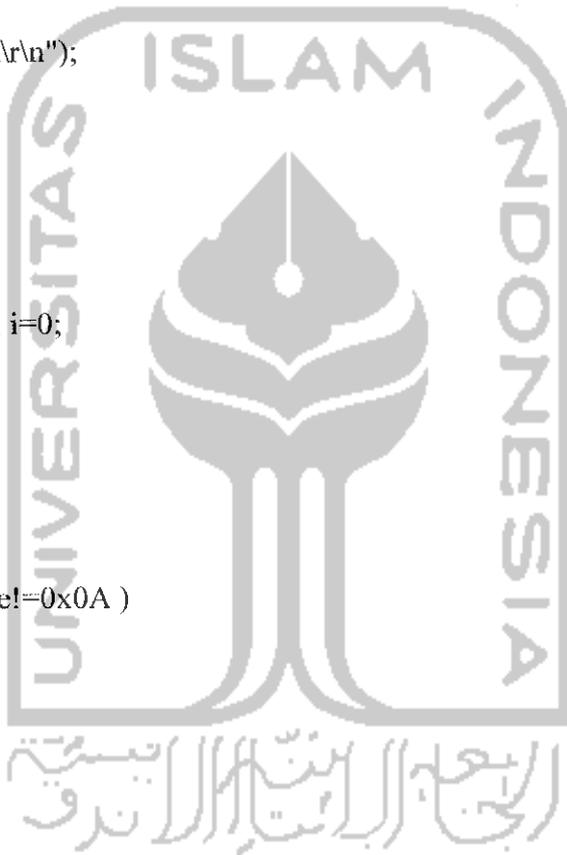
```
    if ( rx_counter!=0 )
```



```

{ if ( getchar()=='*' )
  { temp=0;
    if ( getchar()=='L' || data=='R' )
      { m=data; i=0;
        while ( getchar()!='#' )
          { temp=(unsigned int)(temp*10)+ toint( data ); };
        if ( data=='#' )
          { if ( temp>180 ) printf("ERROR\r\n");
            else new_set(); delay_ms(500);
          }
        keypad();
        if ( flag==1 && code==0x0A )
          { temp=setpoint; m=arah; pas=0; edit=0; i=0;
            while ( i==0 )
              { if ( m=='L' )
                  { keypad();
                    if ( flag==1 && pas==0 && code!=0x0A )
                      { temp=0; pas=1; };
                    if ( flag==1 && code==0x0B )
                      { if ( edit==1 )
                          { new_set(); delay_ms(1000); };
                        lcd_gotoxy(0,1);
                        if ( edit==1 ) lcd_putsf("< save data !>");
                        else lcd_putsf("< cancel ! >");
                        delay_ms(500); i=1; j=1;
                      }
                }
          }
        }

```



```

else if ( flag==1 && code==0x0A )
    { if ( m=='R' ) m='L'; else m='R'; i=0; j=1; }
else if ( flag==1 && code!=0x0A )
    { temp=(temp*10)+code; if ( temp>180 ) temp=0; i=0; j=1; edit=1; };
};
};

```

```
void keypad (void)
```

```

{ DDRB=0x0F; code=0xFF; flag=0;
PORTB.0=0; PORTB.1=1; PORTB.2=1; PORTB.3=1;
key=( ~PINB&0xF0 ); switch (key)
{ case 0x10: code = 0x01; flag = 1; break;
  case 0x20: code = 0x02; flag = 1; break;
  case 0x40: code = 0x03; flag = 1; break; };

```

```
PORTB.0=1; PORTB.1=0; PORTB.2=1; PORTB.3=1;
```

```

key=( ~PINB&0xF0 ); switch (key)
{ case 0x10: code = 0x04; flag = 1; break;
  case 0x20: code = 0x05; flag = 1; break;
  case 0x40: code = 0x06; flag = 1; break; };

```

```
PORTB.0=1; PORTB.1=1; PORTB.2=0; PORTB.3=1;
```

```

key=( ~PINB&0xF0 ); switch (key)
{ case 0x10: code = 0x07; flag = 1; break;

```



```
case 0x20: code = 0x08; flag = 1; break;
case 0x40: code = 0x09; flag = 1; break; };
```

```
PORTB.0=1; PORTB.1=1; PORTB.2=1; PORTB.3=0;
```

```
key=( ~PINB&0xF0 ); switch (key)
```

```
{ case 0x10: code = 0x0A; flag = 1; break;
```

```
case 0x20: code = 0x00; flag = 1; break;
```

```
case 0x40: code = 0x0B; flag = 1; break; };
```

```
}
```



LISTING PROGRAM VISUAL BASIC



BEGIN

MultiUse = -1 'True

Persistable = 0 'NotPersistable

DataBindingBehavior = 0 'vbNone

DataSourceBehavior = 0 'vbNone

MTSTransactionMode = 0 'NotAnMTSObject

END

Attribute VB\_Name = "INIFile"

Attribute VB\_GlobalNameSpace = False

Attribute VB\_Creatable = True

Attribute VB\_PredeclaredId = False

Attribute VB\_Exposed = False

Attribute VB\_Ext\_KEY = "SavedWithClassBuilder6", "Yes"

Attribute VB\_Ext\_KEY = "Member0", "Settings"

Attribute VB\_Ext\_KEY = "Top\_Level", "Yes"

Private Declare Function DeleteFile Lib "kernel32" Alias "DeleteFileA" (ByVal  
lpFileName As String) As Long

Private Const DftEcrBase% = &H30

Private Const DftEcrStep% = &H7

Private mLoadFlag%, mSettings As INISettings, mSetting As INISetting

Private mFileName\$, mFileSize&, mSaved%, mEncrypted%, mBadLines&

Public Property Let Setting(vName\$, Optional vDefault, vValue\$)

Dim crt As INISetting



```

sName$ = LCase(Trim(vName))

Set crt = New INISetting

For i% = 1 To mSettings.Count
    Set crt = mSettings.Item(i%)

    If LCase(crt.Name) = sName$ Then

        oldval$ = crt.Value

        newval$ = Trim(vValue)

        crt.Value = newval$

        If LCase(oldval$) <> LCase(newval$) Then mSaved = 0

        stgfound% = 1

        Exit For

    End If

Next i%

If stgfound% = 0 Then

    Set crt = mSettings.Add(vName, vValue)

    dftval$ = IIf(Not IsMissing(vDefault), vDefault, "")

    If dftval$ <> "" Then crt.Value = dftval$

    mSaved = 0

End If

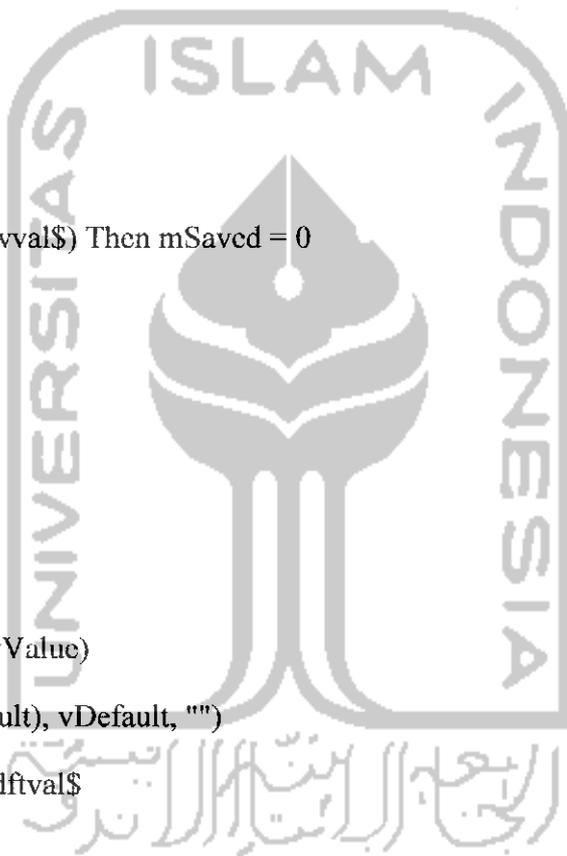
Set crt = Nothing

End Property

Public Property Get Setting$(vName$, Optional vDefault)

    Dim crt As INISetting

```



If mLoadFlag = 0 Then ReadSettings

Set crt = New INISetting

For i% = 1 To mSettings.Count

Set crt = mSettings(i%)

If LCase(crt.Name) = LCase(vName) Then

stgfound% = 1

Exit For

End If

Next i%

If stgfound% = 0 Then

dftval\$ = IIf(Not IsMissing(vDefault), vDefault, "")

Set crt = mSettings.Add(vName, dftval\$)

mSaved = 0

End If

End Property

Public Property Get FileName\$()

If mFileName = "" Then mFileName = App.EXENAME + ".INI"

FileName = mFileName

End Property

Public Property Get FileSize&()

FileSize = mFileSize

End Property

Public Property Let Saved(vData%)

mSaved = vData



End Property

Public Property Get Saved%()

Saved = mSaved

End Property

Public Property Get BadLines&()

BadLines = mBadLines

End Property

Public Function ReadSettings&()

retval& = -1

filme\$ = FileName

filnbr% = FreeFile

mSettings.Clear

gonext% = 1

If gonext% Then

If Dir(filme\$) = "" Then gonext% = 0

End If

If gonext% Then

Open filme\$ For Binary As #filnbr%

filesiz& = LOF(filnbr%)

If filesiz& > 0 Then

filbfr\$ = String(filesiz&, 0)

Get #filnbr%, , filbfr\$



```
If mEncrypted Then filbfr$ = strDcr(filbfr$)
```

```
bytptr& = 1
```

```
bfrlen& = Len(filbfr$)
```

```
dlmtxt$ = vbCrLf
```

```
Do While bytptr& <= bfrlen&
```

```
    dlmpos& = InStr(bytptr&, filbfr$, dlmtxt$)
```

```
    endpos& = IIf(dlmpos& <> 0, dlmpos&, bfrlen& + 1)
```

```
    crtlin$ = Trim(Mid(filbfr$, bytptr&, endpos& - bytptr&))
```

```
    bytptr& = IIf(dlmpos& > 0, dlmpos& + Len(dlmtxt$), bfrlen& + 1)
```

```
    If crtlin$ <> "" Then
```

```
        If InStr(crtlin$, ";") <> 1 Then
```

```
            equpos% = InStr(crtlin$, "=")
```

```
            If equpos% > 0 Then
```

```
                varnme$ = Trim(Left(crtlin$, equpos% - 1))
```

```
                valtxt$ = Right(crtlin$, Len(crtlin$) - equpos%)
```

```
                If varnme$ <> "" Then
```

```
                    Setting(varnme$) = valtxt$
```

```
                    stgcnt& = stgcnt& + 1
```

```
                    retval& = 0
```

```
                End If
```

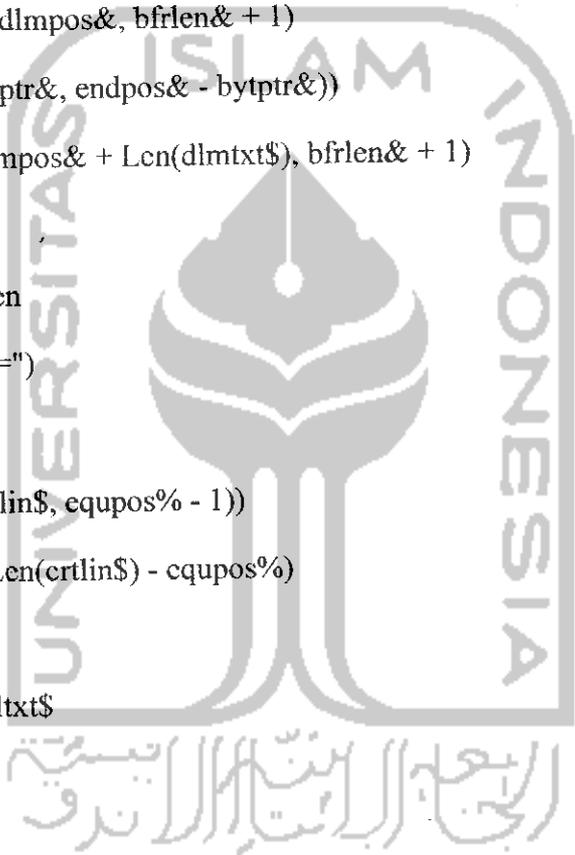
```
            Else
```

```
                inierr& = inierr& + 1
```

```
            End If
```

```
        End If
```

```
    End If
```



```
Loop
End If
Close #filnbr%
End If
```

```
mLoadFlag = 1
```

```
mSaved = 1
```

```
mBadLines = inierr&
```

```
mFileSize = filsiz&
```

```
ReadSettings = retval&
```

```
End Function
```

```
Public Function SaveSettings&()
```

```
retval& = -1
```

```
If mSettings.Count > 0 Then
```

```
For s& = 1 To mSettings.Count
```

```
wribfr$ = wribfr$ + IIf(wribfr$ <> "", Chr(13) + Chr(10), "") +  
mSettings(s&).Name
```

```
wribfr$ = wribfr$ + "=" + mSettings(s&).Value
```

```
Next s&
```

```
If mEncrypted Then wribfr$ = strEcr(wribfr$)
```

```
End If
```

```
If wribfr$ <> "" Then
```

```
filme$ = FileName
```

```
filnbr% = FreeFile
```



DeleteFile filnme\$

On Error Resume Next

Open filnme\$ For Binary As #filnbr%

If Err Then errflg% = 1

If errflg% = 0 Then

inilen& = LOF(filnbr%)

Put #filnbr%, , writbr\$

filsiz& = LOF(filnbr%)

End If

Close #filnbr%

If Err Then errflg% = 1

On Error GoTo 0

mFileSize = filsiz&

mBadLines = 0

If errflg% = 0 Then

mSaved = 1

retval& = 0

End If

End If

SaveSettings = retval&

End Function

Private Sub Class\_Initialize()



```
Set mSettings = New INISettings
```

```
mSaved = 1
```

```
'ReadSettings
```

```
End Sub
```

```
Private Sub Class_Terminate()
```

```
Set mSettings = Nothing
```

```
End Sub
```

```
' ENCRYPTION PARTS
```

```
Private Function strEcr$(srcString$, Optional vBase, Optional vStep, Optional vIndex)
```

```
esrc$ = srcString
```

```
ebse% = IIf(Not IsMissing(vBase), vBase, DftEcrBase)
```

```
estp% = IIf(Not IsMissing(vStep), vStep, DftEcrStep)
```

```
eidx% = IIf(Not IsMissing(vIndex), vIndex, 1)
```

```
temp$ = String(Len(esrc$), 0)
```

```
For idx% = 1 To eidx%
```

```
crnt& = ebse%
```

```
For i% = 1 To Len(temp$)
```

```
crnt& = (crnt& + estp% + Asc(Mid(esrc$, i%, 1))) And &H7FFFFFFF
```

```

Mid(temp$, i%, 1) = Chr(crnt& And &HFF&)
Next i%
esrc$ = temp$
Next idx%

strEcr = temp$

End Function

Private Function strDcr$(srcString$, Optional vBase, Optional vStep, Optional vIndex)
    ebse% = IIf(Not IsMissing(vBase), vBase, DftEcrBase)
    estp% = IIf(Not IsMissing(vStep), vStep, DftEcrStep)
    eidx% = IIf(Not IsMissing(vIndex), vIndex, 1)
    slen% = Len(srcString)
    If slen% > 0 Then
        encd& = Asc(Right(srcString, 1))
        dsrc$ = Chr(ebse%) + Left(srcString, slen% - 1)
        temp$ = String(slen%, 0)
        For idx% = 1 To eidx%
            For i% = slen% To 1 Step -1
                crtb& = Asc(Mid(dsrc$, i%, 1))
                crnt& = (encl& - estp% - crtb&) And &HFF&
                Mid(temp$, i%, 1) = Chr(crnt&)
                encl& = crtb&
            Next i%
            encl& = Asc(Right(temp$, 1))
            dsrc$ = Chr(dcrBase) + Left(temp$, slen% - 1)

```

```
Next idx%
End If

strDer = temp$
End Function
Public Property Let Encrypted(vData%)
    mEncrypted = vData
End Property
Public Property Get Encrypted%()
    Encrypted = mEncrypted
End Property
```



Height = 1995

Left = 240

TabIndex = 6

Top = 1920

Width = 2655

Begin VB.TextBox TxtAngle

Alignment = 2 'Center

Height = 315

Left = 480

TabIndex = 8

Top = 240

Width = 1695

End

Begin VB.CommandButton CmdNav

Caption = "<"

Height = 315

Index = 0

Left = 120

TabIndex = 7

Top = 240

Width = 315

End

Begin VB.CommandButton CmdNav

Caption = ">"

Height = 315



```
Index      = 1
Left       = 2220
TabIndex   = 9
Top        = 240
Width      = 315
```

End

Begin VB.PictureBox picAngle

```
Appearance = 0 'Flat
AutoRedraw = -1 'True
BackColor   = &H80000005&
DrawWidth   = 2
ForeColor   = &H80000008&
Height      = 1215
Left        = 120
ScaleHeight = 1185
ScaleWidth  = 2385
TabIndex    = 10
Top         = 660
Width       = 2415
```



Begin VB.Label lblAng

```
Alignment   = 1 'Right Justify
BackColor    = &H0000C000&
Caption      = "180°"
Height       = 255
Index        = 2
```

```
Left      = 1920
TabIndex  = 13
Top       = 900
Width     = 375
```

End

Begin VB.Label LblIAng

```
Alignment  = 1 'Right Justify
BackColor  = &H0000C000&
Caption    = "0°"
Height     = 255
Index      = 0
Left       = 120
TabIndex   = 11
Top        = 900
Width      = 315
```

End

Begin VB.Label LblIAng

```
Alignment  = 2 'Center
BackColor  = &H0000C000&
Caption    = "90°"
Height     = 195
Index      = 1
Left       = 1080
TabIndex   = 12
Top        = 60
```



```
Width      = 315
End
Begin VB.Shape shpAngle
  BorderStyle = 3 'Dot
  Height      = 1455
  Index       = 1
  Left        = 480
  Shape       = 2 'Oval
  Top         = 300
  Visible     = 0 'False
  Width       = 1455
```

End

```
Begin VB.Shape shpAngle
```

```
  BorderColor = &H00C0C0C0&
  Height      = 1455
  Index       = 0
  Left        = 480
  Top         = 300
  Visible     = 0 'False
  Width       = 1455
```

End

```
Begin VB.Shape ShpFocus
```

```
  BorderColor = &H00808080&
  BorderStyle = 3 'Dot
  Height      = 1830
```



```
Index      = 1
Left       = 60
Top        = 60
Visible    = 0 'False
Width      = 2250
```

```
End
```

```
End
```

```
End
```

```
Begin VB.Frame FrcComm
```

```
Height     = 1575
Left       = 240
TabIndex   = 0
Top        = 240
Width      = 2655
```

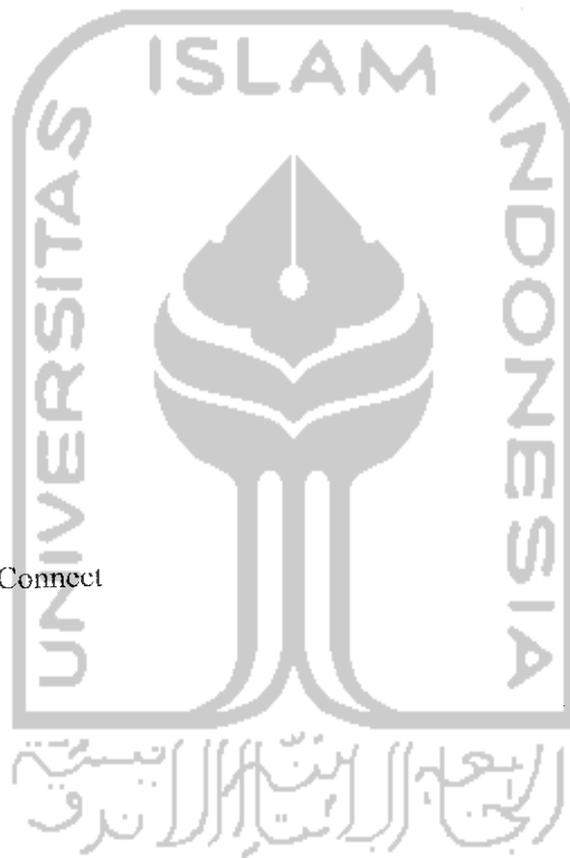
```
Begin VB.CommandButton CmdConnect
```

```
Height     = 375
Left       = 1080
TabIndex   = 5
Top        = 1080
Width      = 1395
```

```
End
```

```
Begin VB.ComboBox CmbBaud
```

```
Height     = 315
Left       = 1080
Style      = 2 'Dropdown List
```



```
TabIndex = 4
Top = 660
Width = 1395
```

End

Begin VB.ComboBox CmbPort

```
Height = 315
Left = 1080
Style = 2 'Dropdown List
TabIndex = 2
Top = 300
Width = 1395
```

End

Begin VB.Label LblBaud

```
BackColor = &H0000C000&
BackStyle = 0 'Transparent
Caption = "&Baud rate"
Height = 255
Left = 180
TabIndex = 3
Top = 720
Width = 795
```

End

Begin VB.Label LblPort

```
BackColor = &H0000C000&
BackStyle = 0 'Transparent
```



```
Caption = "&Port serial"  
Height = 255  
Left = 180  
TabIndex = 1  
Top = 360  
Width = 795
```

```
End
```

```
End
```

```
Begin VB.CommandButton CmdSetting
```

```
Caption = "&Setting"  
Height = 375  
Left = 240  
TabIndex = 19  
Top = 4620  
Visible = 0 'False  
Width = 855
```

```
End
```

```
Begin MSCommLib.MSComm Comm
```

```
Left = 2820  
Top = 2280  
_ExtentX = 1005  
_ExtentY = 1005  
_Version = 393216  
DTREnable = -1 'True
```

```
End
```



الرَّبِّعَالِ اِسْتِاْ اَلْاَنْدُوْ  
اَلْاِسْلَامِ اَلْاِنْدُوْ

Begin VB.CommandButton CmdClose

Caption = "Close"

Height = 375

Left = 6360

TabIndex = 20

Top = 4620

Width = 855

End

Begin VB.Line Line1

BorderColor = &H00808080&

X1 = 1560

X2 = 1560

Y1 = 4020

Y2 = 4320

End

Begin VB.Shape Shape2

BorderColor = &H00808080&

Height = 315

Left = 240

Top = 4020

Width = 2655

End

Begin VB.Label FldSetpoint

Alignment = 1 'Right Justify

BackColor = &H0000C000&



BackStyle = 0 'Transparent

Height = 255

Left = 1080

TabIndex = 15

Top = 4080

Width = 435

End

Begin VB.Label FldActual

Alignment = 1 'Right Justify

BackColor = &H0000C000&

BackStyle = 0 'Transparent

Height = 255

Left = 2400

TabIndex = 17

Top = 4080

Width = 435

End

Begin VB.Label LblActual

BackColor = &H0000C000&

BackStyle = 0 'Transparent

Caption = "Aktual"

Height = 255

Left = 1620

TabIndex = 16

Top = 4080



الجامعة الإسلامية  
الاندونيسية

Width = 675

End

Begin VB.Label LblSetpoint

BackColor = &H0000C000&

BackStyle = 0 'Transparent

Caption = "Setpoint"

Height = 255

Left = 300

TabIndex = 14

Top = 4080

Width = 675

End

Begin VB.Line LinSplitter

BorderColor = &H00808080&

Index = 0

X1 = 240

X2 = 7200

Y1 = 4440

Y2 = 4440

End

Begin VB.Label LblLog

BackColor = &H0000C000&

Height = 3735

Left = 3060

TabIndex = 18



Top = 480  
Width = 4095

End

Begin VB.Line LinSplitter

BorderColor = &H00FFFFFF&

Index = 1

X1 = 240

X2 = 7200

Y1 = 4455

Y2 = 4455

End

Begin VB.Shape Shape1

BorderColor = &H00808080&

FillColor = &H00C0C0C0&

FillStyle = 0 'Solid

Height = 3975

Left = 3000

Top = 360

Width = 4215

End

End

Attribute VB\_Name = "FrmMain"

Attribute VB\_GlobalNameSpace = False

Attribute VB\_Creatable = False

Attribute VB\_PredeclaredId = True



```
Attribute VB_Exposed = False
```

```
Private Const mResolution% = 15
```

```
logtxt$ = vData
```

```
dirflg% = vDirection
```

```
If lincnt% >= maxline Then
```

```
  lfdpos& = InStr(LblLog.Caption, vbLf)
```

```
  If lfdpos& > 0 Then
```

```
    LblLog.Caption = Right(LblLog.Caption, Len(LblLog.Caption) - lfdpos&)
```

```
    lincnt% = lincnt% - 1
```

```
  End If
```

```
End If
```

```
logtxt$ = Format(Now, "hh:nn:ss") + " " + If(dirflg%, "TERIMA", "KIRIM") + ": " +  
logtxt$
```

```
LblLog.Caption = LblLog.Caption + Replace(logtxt$, vbCrLf, vbLf)
```

```
lincnt% = lincnt% + 1
```

```
End Sub
```

```
Private Sub UpdateInfo()
```

```
  FldSetpoint.Caption = Format(mCrtAngle, "#0") + "°"
```

```
  FldActual.Caption = Format(mActAngle, "#0") + "°"
```

```
End Sub
```

```
Private Function SendCommand&(vCommand$, vValue%)
```

```
  retval& = -1
```

```
  ' ## debug
```



If Comm.PortOpen Then

cmdtxt\$ = "\*" + vCommand + FLStr(vValue, 3) + "#" + vbCrLf

AppendLog cmdtxt\$

On Error Resume Next

Comm.Output = cmdtxt\$

errflg& = Err

On Error GoTo 0

retval& = 0

End If

SendCommand = retval&

End Function

Private Function SetAngle(vAngle%)

mCrtAngle = vAngle

TxtAngle.Text = Format(mCrtAngle, "#0")

DrawArrow mCrtAngle

ResetButtons

End Function

Private Function AcceptCommand(vCommand\$)

' ## periodic: RBT:"XXX,L,YYY#<CR><LF>

cmdtxt\$ = vCommand

cmdtxt\$ = Replace(cmdtxt\$, vbCrLf, "")

cmdtxt\$ = Replace(cmdtxt\$, "####", "")



VERSION 5.00

Object = "{648A5603-2C6E-101B-82B6-000000000014}#1.1#0"; "MSCOMM32.OCX"

Begin VB.Form FrmMain

BorderStyle = 1 'Fixed Single

Caption = "KONTROL LENGAN ROBOT"

ClientHeight = 5220

ClientLeft = 45

ClientTop = 435

ClientWidth = 7470

BeginProperty Font

Name = "Tahoma"

Size = 8.25

Charset = 0

Weight = 400

Underline = 0 'False

Italic = 0 'False

Strikethrough = 0 'False

EndProperty

LinkTopic = "Form1"

LockControls = -1 'True

MaxButton = 0 'False

ScaleHeight = 5220

ScaleWidth = 7470

StartPosition = 1 'CenterOwner

Begin VB.Frame FrmMon



gonext% = 1

If gonext% Then

If InStr(cmdtxt\$, "RBT:") = 1 Then

cmdarg\$ = Trim(Right(cmdtxt\$, Len(cmdtxt\$) - 4))

If Right(cmdarg\$, 1) = "#" Then cmdarg\$ = Left(cmdarg\$, Len(cmdarg\$) - 1)

argptr& = 0

xxtxt\$ = Trim(ListMember(cmdarg\$, argptr&))

movtxt\$ = Trim(ListMember(cmdarg\$, argptr&))

acttxt\$ = Trim(ListMember(cmdarg\$, argptr&))

actang% = Val(acttxt\$)

mActAngle = actang%

UpdateInfo

End If

End If

AppendLog cmdtxt\$, 1

End Function

Private Function BufferCommData&(vData\$)

crtdat\$ = vData

If crtdat\$ <> "" Then

mRdBuffer = mRdBuffer + crtdat\$

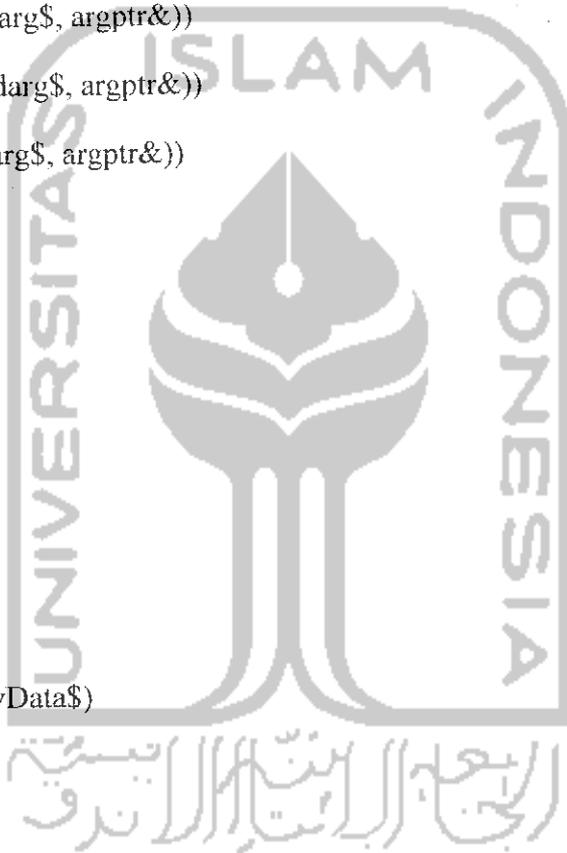
lfdpos& = InStr(mRdBuffer, vbCrLf)

Do While lfdpos& > 0

If lfdpos& > 0 Then

cmdtxt\$ = Left(mRdBuffer, lfdpos& - 1)

mRdBuffer = Right(mRdBuffer, Len(mRdBuffer) - (lfdpos& + 1))



```

    AcceptCommand cmdtxt$
End If

Ifdpos& = InStr(mRdBuffer, vbCrLf)

Loop

End If

End Function

Private Sub ResetButtons()
' ## debug
If Comm.PortOpen Then
    If mCrtAngle > 0 Then canlft% = 1
    If mCrtAngle < 180 Then canrgt% = 1
    cannav% = 1
Else
    cancom% = 1
End If
LblPort.Enabled = cancom%
LblBaud.Enabled = cancom%
CmbPort.Enabled = cancom%
CmbBaud.Enabled = cancom%
CmbPort.BackColor = If(cancom%, vbWindowBackground, vbButtonFace)
CmbBaud.BackColor = If(cancom%, vbWindowBackground, vbButtonFace)
CmdNav(0).Enabled = canlft%
CmdNav(1).Enabled = canrgt%
TxtAngle.Enabled = cannav%
TxtAngle.BackColor = If(cannav%, vbWindowBackground, vbButtonFace)

```



End Sub

Private Sub Comm\_OnComm()

If Comm.CommEvent = comEvReceive Then

    crtdat\$ = Comm.Input

    BufferCommData crtdat\$

End If

End Sub

Private Sub CmbPort\_Click()

    mCommPort = CmbPort.ListIndex + 1

End Sub

Private Sub CmbBaud\_Click()

    mBaudRate = CmbBaud.ListIndex + 1

End Sub

Private Sub CmdNav\_Click(Index As Integer)

If Index > 0 Then

    dirflg% = 1

    cmdtxt\$ = "R"

Else

    dirflg% = -1

    cmdtxt\$ = "L"

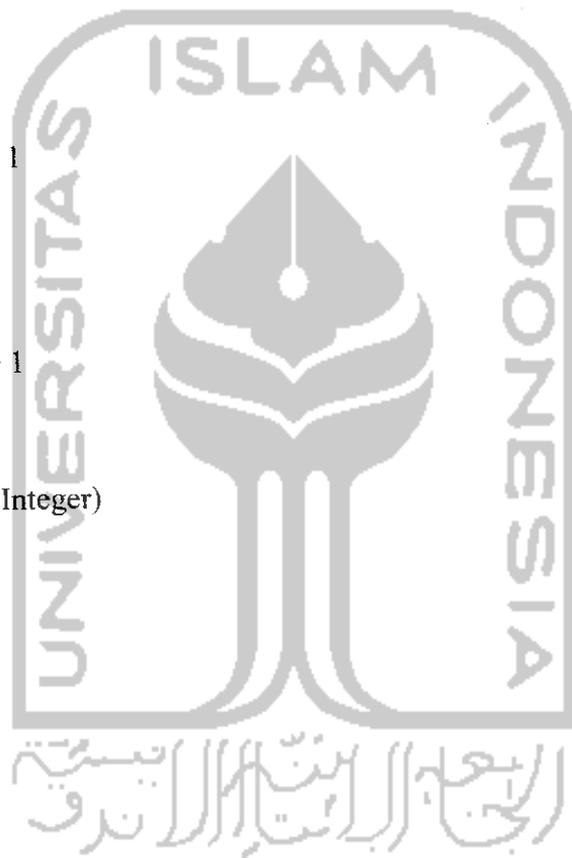
End If

angmov% = mResolution

mCrtAngle = mCrtAngle + (angmov% \* dirflg%)

If mCrtAngle > 180 Then mCrtAngle = 180

If mCrtAngle < 0 Then mCrtAngle = 0



```

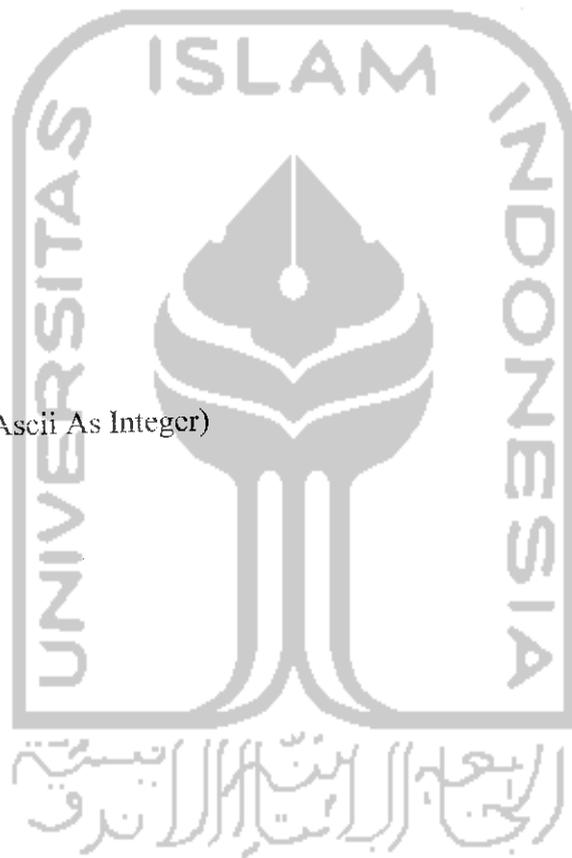
TxtAngle.Text = Format(mCrtAngle, "#0")
DrawArrow mCrtAngle
SendCommand cmdtxt$, angmov%
UpdateInfo
ResetButtons
End Sub

Private Sub TxtAngle_GotFocus()
    TBSelectText TxtAngle
End Sub

Private Sub TxtAngle_Change()
    '
End Sub

Private Sub TxtAngle_KeyPress(KeyAscii As Integer)
    If KeyAscii = vbKeyReturn Then
        KeyAscii = 0
        angtxt$ = Trim(TxtAngle.Text)
        newang% = mCrtAngle
        If IsNumeric(angtxt$) Then
            chkang% = Val(angtxt$)
            If (chkang% >= 0) And (chkang% <= 180) Then newang% = chkang%
        End If
        angmov% = newang% - mCrtAngle
        angdif% = Abs(angmov%)
        cmdtxt$ = IIf(angmov% > 0, "R", "L")
        mCrtAngle = mCrtAngle + angmov%
    End Sub

```



```

TxtAngle.Text = Format(mCrtAngle, "#0")

DrawArrow mCrtAngle

SendCommand cmdtxt$, angdif%

UpdateInfo

End If

End Sub

Private Sub picAngle_GotFocus()
'ShpFocus(1).Visible = True
End Sub

Private Sub picAngle_LostFocus()
'ShpFocus(1).Visible = False
End Sub

Private Sub CmdSetting_Click()
'FrmSetting.Show 1
End Sub

Private Sub CmdConnect_Click()
If Not Comm.PortOpen Then
Select Case mBaudRate
Case 1: bautxt$ = "4800"
Case 2: bautxt$ = "9600"
Case 3: bautxt$ = "19200"
End Select

Comm.Settings = bautxt$ + ",n,8,1"
Comm.CommPort = mCommPort

On Error Resume Next

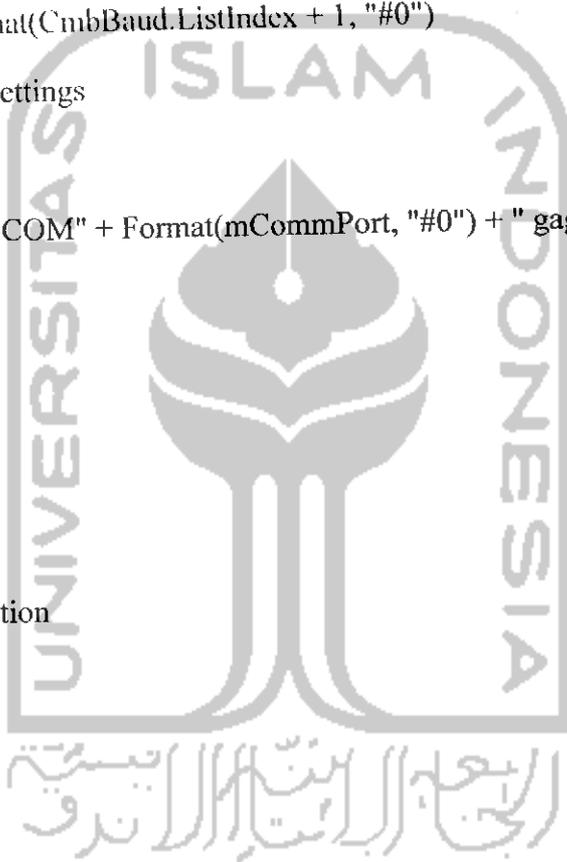
```



```

Comm.PortOpen = True
errflg& = Err
On Error GoTo 0
If errflg& = 0 Then
    LblLog.Caption = ""
    Cfg.Setting(cfgCommPort) = Format(CmbPort.ListIndex + 1, "#0")
    Cfg.Setting(cfgBaudRate) = Format(CmbBaud.ListIndex + 1, "#0")
    If Cfg.Saved = 0 Then Cfg.SaveSettings
Else
    MsgBox "Koneksi port serial via COM" + Format(mCommPort, "#0") + " gagal.",
vbInformation
End If
Else
    Comm.PortOpen = False
End If
CmdConnect.Caption = ConnectCaption
ResetButtons
End Sub
Private Sub CmdClose_Click()
    Unload Me
End Sub
Private Sub Form_DblClick()
    FunctionTest
End Sub
Private Sub Form_Load()

```



```
Set Cfg = New INIFile
```

```
For i% = 0 To 2
```

```
    LblIAng(i%).BackStyle = 0
```

```
Next i%
```

```
LblLog.BackStyle = 0
```

```
CmbPort.Clear
```

```
For i% = 1 To 8
```

```
    CmbPort.AddItem "COM" + Format(i%, "#0")
```

```
Next i%
```

```
CmbBaud.Clear
```

```
CmbBaud.AddItem "4800"
```

```
CmbBaud.AddItem "9600"
```

```
CmbBaud.AddItem "19200"
```

```
mCommPort = Val(Cfg.Setting(cfgCommPort, "1"))
```

```
mBaudRate = Val(Cfg.Setting(cfgBaudRate, "2"))
```

```
If Cfg.Saved = 0 Then Cfg.SaveSettings
```

```
CmbPort.ListIndex = mCommPort - 1
```

```
CmbBaud.ListIndex = mBaudRate - 1
```

```
TxtAngle.Text = Format(mCrtAngle, "#0")
```

