

**PERANCANGAN SISTEM MONITORING *LOADING ARM* DI PT BADAK
BONTANG BERBASIS MIKROKONTROLER ATMega8**

**SKRIPSI
KONSENTRASI TEKNIK ELEKTRONIKA**

Diajukan Untuk Memenuhi Persyaratan

Memperoleh Gelar Sarjana Teknik



Disusun oleh:

EDWIN GUTAMA

NIM. 0810633007-63

**KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
UNIVERSITAS BRAWIJAYA
FAKULTAS TEKNIK
MALANG
2013**

LEMBAR PERSETUJUAN

PERANCANGAN SISTEM MONITORING *LOADING ARM* DI PT BADAK
BONTANG BERBASIS MIKROKONTROLER ATMega8

SKRIPSI
KONSENTRASI TEKNIK ELEKTRONIKA

Diajukan untuk memenuhi persyaratan
memperoleh gelar Sarjana Teknik



Disusun oleh:

EDWIN GUTAMA

NIM. 0810633007-63

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Ir. Nurussa'adah, MT

NIP. 19680706 199203 2 001

Dosen Pembimbing II

Mochammad Rif'an, ST., MT

NIP. 19710301 200012 1 001

LEMBAR PENGESAHAN

PERANCANGAN SISTEM MONITORING *LOADING ARM* DI PT BADAK
BONTANG BERBASIS MIKROKONTROLER ATMega8

SKRIPSI

JURUSAN TEKNIK ELEKTRO

Diajukan Untuk Memenuhi Sebagian Persyaratan
Memperoleh Gelar Sarjana Teknik

Disusun oleh:

EDWIN GUTAMA

NIM. 0810633007-63

Skripsi ini telah diuji dan dinyatakan lulus pada
tanggal 25 Juli 2013

DOSEN PENGUJI

Ir Muhammad Julius St., MS.
NIP. 19540720 198203 1 002

Ir. Ponco Siwindarto, M.Eng.Sc.
NIP. 19590304 198903 1 001

Akhmad Zainuri, ST., MT.
NIK. 840120 06 1 1 0052

Mengetahui,

Ketua Jurusan Teknik Elektro

Dr. Ir. Sholeh Hadi Pramono., MS.
NIP. 19580728 198701 1 001



PENGANTAR

Dengan memanjangkan puji syukur kehadirat Allah SWT atas karunia dan rahmat-Nya, penulis dapat menyelesaikan pembuatan laporan skripsi yang berjudul **“Perancangan Sistem Monitoring Loading Arm di PT Badak Bontang Berbasis Mikrokontroler ATMega8”**

Laporan ini dibuat untuk memenuhi matakuliah SKRIPSI yang merupakan persyaratan akademik bagi setiap mahasiswa S1 Jurusan Teknik Elektro Fakultas Teknik Universitas Brawijaya.

Proses pelaksanaan dan pembuatan laporan Skripsi ini tidak bisa lepas dari dukungan, bantuan, serta sumbangan saran dari berbagai pihak. Untuk itu pada kesempatan ini penulis ingin mengucapkan terima kasih yang sebesar-besarnya kepada:

1. Bapak Dr. Ir. Sholeh Hadi, MS, selaku Ketua Jurusan Elektro Fakultas Teknik Universitas Brawijaya Malang.
2. Bapak M. Aziz Muslim, ST., MT, selaku Sekretaris Jurusan Elektro Fakultas Teknik Universitas Brawijaya Malang.
3. Ibu Ir. Nurrusadah, MT, selaku dosen pembimbing 1 dan Bapak Moch. Rif'an, MT, selaku dosen pembimbing 2 Jurusan Elektro Fakultas Teknik Universitas Brawijaya Malang.
4. Kedua orang tua serta keluarga tercinta yang telah memberikan do'a dan dukungan, baik moril maupun materil.
5. Karyawan PT Badak Departement Instrument yang telah membantu penulis selama mengambil data di PT Badak Bontang.
6. Annisa Triandini yang telah memberikan motivasi dan semangat, serta menemani selama penulis membuat laporan ini.
7. Semua teman-teman Teknik Elektro Universitas Brawijaya, khususnya teman-teman CONCORDES 2008 dan teman-teman paket B yang telah membantu penulis dalam urusan kampus selama penulis tidak berada di tempat.
8. Keluarga besar Teknik Elektro Universitas Brawijaya khususnya karyawan sekretariat Jurusan Teknik Elektro yang telah banyak memberikan bantuan dalam urusan administrasi.

Penulis menyadari bahwa laporan ini masih perlu banyak penyempurnaan. Oleh karena itu penulis mengharapkan kritik dan saran yang bersifat membangun untuk kesempurnaan laporan ini. Akhir kata penulis berharap laporan ini dapat bermanfaat bagi penulis maupun pihak yang memerlukan serta mahasiswa Universitas Brawijaya Malang pada umumnya.

Malang, Juli 2013

Penulis



ABSTRAK

Edwin Gutama, Jurusan Teknik Elektro Fakultas Teknik Universitas Brawijaya, Juni 2013, *Perancangan Sistem Monitoring Loading Arm di PT Badak Bontang Berbasis Mikrokontroler ATMega8*,
Dosen Pembimbing: Ir. Nuruss'adah, MT. dan Mohammad Rif'an, ST., MT.

PT Badak merupakan penghasil gas alam cair yang ditunjang peralatan elektronik modern. Gas alam diolah melalui tahap tertentu hingga menjadi cair dan disimpan dalam tanki. Hasilnya dikirim ke negara-negara lain dengan kapal tanker. Terdapat *Loading Arm* yang berfungsi sebagai lengan penghubung antara dermaga dengan kapal untuk mengalirkan gas alam cair. Pada awalnya *Loading Arm* tidak memiliki sistem monitoring untuk memantau pergerakannya dan hanya mengandalkan *limit switch* sebagai sensor pengaman. Pada akhirnya digunakanlah *Position Monitoring System* (PMS) buatan FMC Europe. Namun, karena rusak dan harganya yang mahal maka PT Badak tidak memakai sistem tersebut dan kembali menggunakan *limit switch* sebagai pengaman. Oleh karena itu, penulis mencoba merancang dan membuat sistem monitoring yang dapat digunakan untuk memantau pergerakan *Loading Arm*.

Pada perancangan dan pembuatan sistem ini, sensor yang digunakan untuk membaca sudut pergerakan *Loading Arm* menggunakan modul sensor potensiometer dan dikonversi menjadi sudut yang kemudian dikirim melalui komunikasi serial untuk ditampilkan di komputer. Terdapat beberapa parameter yang ditampilkan seperti DEPTH, WIDTH, DISTANCE dan HEIGHT serta alarm peringatan dan gambar pergerakan *Loading Arm*.

Kata kunci: *Loading Arm*, PMS, Potensiometer, PT BADAK.



DAFTAR ISI

LEMBAR PERSETUJUAN	ii
LEMBAR PENGESAHAN	iii
PENGANTAR	iv
ABSTRAK	vi
DAFTAR ISI.....	vii
DAFTAR GAMBAR.....	x
DAFTAR TABEL	xi
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah	2
1.4 Tujuan	2
1.5 Sistematika Penulisan	2
BAB II TINJAUAN PUSTAKA.....	4
2.1 PT Badak NGL (<i>Natural Gas Liquification</i>)	4
2.2 <i>Loading Arm</i>	5
2.3 Modul Sensor Potensiometer	7
2.4 <i>Analog to Digital Converter</i> (ADC)	7
2.5 Mikrokontroler	10
2.5.1 ATMEL AVR ATmega8.....	10
2.5.2 ADC Mikrokontroler ATmega8	12
2.5.3 Komunikasi Serial ATmega8	12
2.6 Komunikasi Serial.....	13
2.6.1 RS-232	13
2.6.2 RS-485	13
2.6.3 <i>Converter RS-232 to RS-485</i>	14
2.7 Pemrograman Visual	15



BAB III METODOLOGI	17
3.1 Spesifikasi Alat	17
3.2 Perancangan Alat	17
3.2.1 Diagram Blok.....	17
3.2.2 Deskripsi Cara Kerja Alat.....	18
3.2.3 Perancangan Perangkat Keras.....	18
3.2.4 Perancangan Perangkat Lunak.....	21
3.3 <i>Position Monitoring System (PMS)</i>	23
3.4 Pengujian Alat.....	25
3.4.1 Pengujian Perangkat Keras	25
3.4.2 Pengujian Perangkat Lunak	26
3.4.3 Pengujian Keseluruhan	26
BAB IV PERANCANGAN DAN PEMBUATAN ALAT	27
4.1 Perancangan Perangkat Keras	27
4.1.1 Perancangan Mekanik.....	27
4.1.2 Perancangan Elektrik	29
4.2 Perancangan Perangkat Lunak	34
4.2.1 Perancangan Algoritma Program Mikrokontroler	34
4.2.2 Perancangan Algoritma Program PMS Visual Studio	35
BAB V PENGUJIAN DAN ANALISIS	39
5.1 Pengujian Perangkat Keras	39
5.1.1 Pengujian Modul Sensor Potensiometer	39
5.1.2 Pengujian Catu Daya	41
5.1.3 Pengujian Mikrokontroler.....	43
5.1.4 Pengujian Rangkaian MAX-232.....	44
5.1.5 Pengujian Komunikasi Serial.....	46
5.2 Pengujian Perangkat Lunak.....	47
5.2.1 Tujuan	47

5.2.2 Alat yang Digunakan	47
5.2.3 Prosedur Pengujian	48
5.2.4 Hasil Pengujian dan Analisis	48
5.3 Pengujian Keseluruhan Sistem.....	49
5.3.1 Tujuan	49
5.3.2 Alat yang Digunakan	49
5.3.3 Prosedur Pengujian	49
5.3.4 Hasil Pengujian dan Analisis	50
BAB VI KESIMPULAN DAN SARAN.....	53
6.1 Kesimpulan	53
6.2 Saran	53
DAFTAR PUSTAKA.....	54
LAMPIRAN.....	55

DAFTAR GAMBAR

Gambar 2.1 Skema Proses Pembuatan LNG di PT Badak	5
Gambar 2.2 <i>Loading Arm</i> PT Badak	6
Gambar 2.3 Modul Sensor Potensiometer	7
Gambar 2.4 Skematik Rangkaian ADC	8
Gambar 2.5 <i>Sampling ADC</i>	8
Gambar 2.6 Gelombang Sinus Terdigitasi dengan resolusi 3 bit	9
Gambar 2.7 Contoh Beberapa Bentuk Mikrokontroller ATMEL	10
Gambar 2.8 Konfigurasi Pin ATMega8	11
Gambar 2.9 Contoh Konverter RS-232 to RS-485	14
Gambar 2.10 Tampilan awal aplikasi <i>Microsoft Visual Studio 2012 Express</i>	15
Gambar 3.1 Diagram Blok Sistem Monitoring <i>Loading Arm</i>	17
Gambar 3.2 Diagram Blok Catu Daya	19
Gambar 3.3 Diagram Blok Rangkaian Mikrokontroler	20
Gambar 3.4 Bentuk Kerangka Miniatur <i>Loading Arm</i> dan Box Elektrik	21
Gambar 3.5 Diagram Alur Program di CV AVR	22
Gambar 3.6 Tampilan Program di Komputer	22
Gambar 3.7 Parameter <i>Loading Arm</i>	23
Gambar 4.1 Perancangan Miniatur <i>Loading Arm</i>	28
Gambar 4.2 Rangkaian Catu Daya	29
Gambar 4.3 Rangkaian Mikrokontroler ATMega8	31
Gambar 4.4 Rangkaian Reset	31
Gambar 4.5 Rangkaian Penyesuai Level Tegangan (MAX232)	33
Gambar 4.6 Diagram Alir Program Utama	35
Gambar 4.7 Diagram alir batas alarm <i>Loading Arm</i>	37
Gambar 4.8 Algoritma Program PMS	38
Gambar 5.1 Rangkaian pengujian modul sensor potensiometer	39
Gambar 5.2 Pengukuran Tegangan Modul Sensor Potensiometer dengan $\theta=90^\circ$	40
Gambar 5.3 Rangkaian Pengujian Catu Daya Tanpa Beban	41
Gambar 5.4 Rangkaian Pengujian Catu Daya Berbeban	42
Gambar 5.5 Pengukuran Catu Daya Berbeban	42
Gambar 5.6 Pengukuran Catu Daya Tanpa Beban	42
Gambar 5.7 Rangkaian Pengujian Tegangan Tx/Rx Mikrokontroler	43
Gambar 5.8 Pengujian Pengukuran Tegangan Tx/Rx Mikrokontroler	44
Gambar 5.9 Diagram Blok Pengujian Rangkaian MAX-232	45
Gambar 5.10 Pengujian Rangkaian MAX-232	45
Gambar 5.11 Diagram Blok Pengujian Komunikasi Serial	46
Gambar 5.12 Pengujian Komunikasi Serial	47
Gambar 5.13 Pengujian Gerak Program <i>Loading Arm</i> Dengan $a=0$, $b=180$ dan $c=45$	48
Gambar 5.14 Pengujian Gerak Program <i>Loading Arm</i> Dengan $a=0$, $b=90$ dan $c=90$	48
Gambar 5.15 Diagram Blok Pengujian Keseluruhan Sistem	50
Gambar 5.16 Pengujian Keseluruhan Sistem Monitoring <i>Loading Arm</i>	50

DAFTAR TABEL

Tabel 3.1 Pengaturan Fuse Bit	20
Tabel 4.1 Pin yang digunakan pada mikrokontroler ATMega8.....	30
Tabel 4.2 Batas-batas Sudut Lengan <i>Loading Arm</i>	35
Tabel 5.1 Pengujian Tegangan Modul Sensor Potensiometer.....	40
Tabel 5.2 Pengujian keseluruhan sistem monitoring <i>Loading Arm</i>	51



BAB I

PENDAHULUAN

1.1 Latar Belakang

PT BADAK adalah salah satu perusahaan penghasil gas alam cair/*Liquid Natural Gas* (LNG) yang dalam proses produksinya ditunjang dengan peralatan-peralatan berteknologi modern. Gas alam cair yang sudah jadi nantinya akan dikirim ke negara-negara lain untuk dikonsumsi. Pada tahun 1990-an FMC Europe S.A mendirikan Loading Dock di PT Badak. Loading Dock merupakan sarana yang dibangun PT Badak untuk melayani pengapalan LNG kepada pihak konsumen. Di Loading Dock terdapat beberapa *Loading Arm* yang sangat penting. *Loading Arm* merupakan salah satu alat yang dapat membantu dalam pengiriman hasil gas alam cair tersebut.

Loading Arm adalah lengan penghubung antara dermaga dengan kapal tanker LNG. Gas alam yang sudah dicairkan tersebut dialirkan dari tangki-tangki LNG dan mengalir melalui pipa-pipa yang akhirnya akan dihubungkan oleh *Loading Arm* untuk selanjutnya ditampung di kapal tanker LNG. Terdapat 3 tempat *Loading Arm* ini dibangun, yaitu di Dock-1, Dock-2 dan Dock-3. *Loading Arm* pada Dock-1 digunakan untuk menyalurkan LNG saja dan memiliki 5 *Loading Arm*, sedangkan untuk Dock-2 dan Dock-3 digunakan untuk menyalurkan LNG dan LPG dan memiliki 5 *Loading Arm* untuk LNG dan 4 *Loading Arm* untuk LPG.

Dalam setiap prosesnya, alat-alat yang ada di PT Badak memiliki sistem yang dapat mengawasi suatu pekerjaan agar kesalahan-kesalahan yang ada bisa dilihat dan diatasi. Salah satunya yaitu sistem monitoring yang ada pada *Loading Arm* ini.

Masalah yang terjadi pada *Loading Arm* dikarenakan tidak berfungsinya sistem monitoring Dock-1 yang terletak di bagian operator sehingga para operator tidak dapat mengawasi pergerakan dari *Loading Arm* tersebut. Maka pada penelitian kali ini penulis mencoba menyelesaikan masalah tersebut dengan merancang suatu sistem monitoring yang dapat mengawasi pergerakan *Loading Arm* tersebut menggunakan prinsip dasar elektronika.

1.2 Rumusan Masalah

Berdasarkan masalah yang telah dijelaskan pada latar belakang, dapat dibuat rumusan sebagai berikut:

- 1) Bagaimana merancang sistem monitoring *Loading Arm* menggunakan mikrokontroler ATMega8.
- 2) Bagaimana membuat batas pergerakan *Loading Arm* sesuai dengan *range* yang telah ditentukan.
- 3) Bagaimana merancang sistem antarmuka dari mikrokontroler ke komputer/*Console*.
- 4) Bagaimana menampilkan program keluaran dalam *software* Visual Studio.

1.3 Batasan Masalah

Adapun batasan masalah penelitian ini adalah:

- 1) Miniatur *Loading Arm* tidak dibuat sek kompleks aslinya, tetapi secara prinsip kerjanya sama, hanya saja sistem monitoringnya berbeda dengan aslinya.
- 2) Sensor yang digunakan untuk membaca tegangan menggunakan *variable resistor*.
- 3) Hanya membahas sistem monitoring *Loading Arm* saja.

1.4 Tujuan

Tujuan penelitian ini adalah untuk merancang dan membuat sistem monitoring yang ada pada *Loading Arm* PT Badak.

1.5 Sistematika Penulisan

Sistematika penulisan yang digunakan dalam penyusunan laporan penelitian ini adalah sebagai berikut.

BAB I PENDAHULUAN

Memuat latar belakang, rumusan masalah, ruang lingkup, tujuan, dan sistematika penulisan.

BAB II TINJAUAN PUSTAKA



Tinjauan pustaka membahas tentang perancangan sistem monitoring *Loading Arm* dan teori-teori lain yang terkait.

BAB III METODOLOGI

Memberikan penjelasan tentang metode yang digunakan dalam perancangan sistem monitoring *Loading Arm* ini yang terdiri atas metode pengambilan data, pengujian, dan penarikan kesimpulan.

BAB IV PERANCANGAN

Melakukan pembuatan sistem monitoring *Loading Arm* berdasarkan data primer maupun data sekunder yang diperlukan.

BAB V PENGUJIAN DAN ANALISIS

Membahas tentang peroses pengujian dan analisis data yang diperoleh dari sistem yang telah dibuat.

BAB VI PENUTUP

Berisi kesimpulan perancangan dan saran yang diperoleh dari hasil pengujian untuk melakukan pengembangan aplikasi selanjutnya.



BAB II

TINJAUAN PUSTAKA

Pada umumnya sebuah proses memiliki sistem monitoring untuk memonitor pekerjaan tersebut. Salah satunya sistem monitoring yang ada pada *Loading Arm*. Sistem monitoring pada *Loading Arm* berfungsi untuk memonitor pergerakan dari *Loading Arm*. Ada beberapa perangkat penting dalam sistem monitoring *Loading Arm* tersebut, yaitu sensor yang berada di *Loading Arm* berfungsi untuk membaca suatu besaran, lalu mikrokontroler sebagai otak dari sistem monitoring, dan komputer sebagai keluaran atau *display* dari pergerakan *Loading Arm* ini.

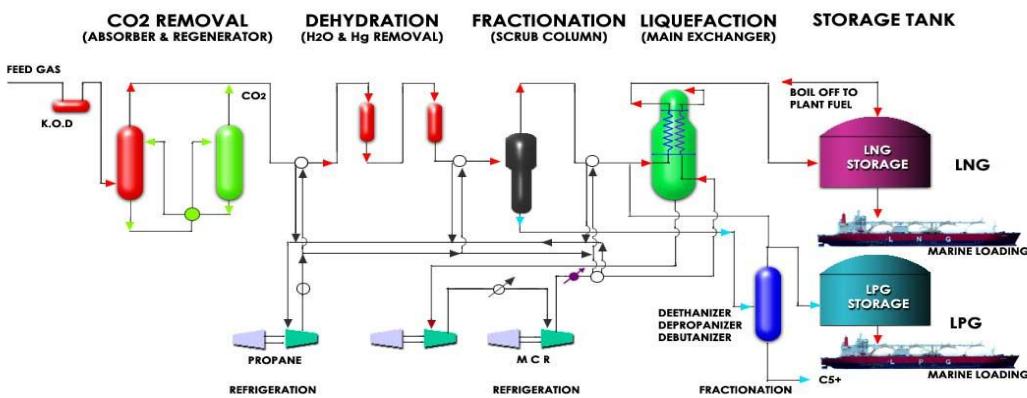
2.1 PT Badak NGL (*Natural Gas Liquification*)

PT Badak merupakan salah satu perusahaan Pertamina yang dapat mencairkan gas alam untuk dieksport ke negara-negara lain seperti Jepang, Taiwan, Korea bahkan ke negara di Timur Tengah. Pencairan gas alam menjadi LNG/LPG bertujuan untuk memudahkan dalam penyimpanan dan transportasi. Gas alam yang diolah di kilang LNG/LPG PT BADAQ NGL, Bontang, berasal dari beberapa sumur gas, yaitu Muara Badak, Nilam, Attaka, Tunu, Tambora, Samberah, Bekapai dan Handil. Secara singkat proses pencairan gas alam ini dilakukan pada suatu rangkaian peralatan yang disebut *Process Train*.

Pada saat ini, di kilang LNG/LPG Bontang terdapat delapan unit *train*, *train A, B, C, D, E, F, G, dan H* yang masing-masing berkapasitas produksi $\pm 710 \text{ m}^3/\text{hr}$. Dan untuk mendukung kelancaran produksi, LNG *Plant* Bontang dilengkapi dengan *Utilities Plant* serta *Storage and Loading*.

Process Train adalah unit pengolahan gas alam hingga menjadi LNG serta produk-produk lainnya (pencairan fraksi berat dari gas alam). Dalam pengolahan gas alam di *process train* dilakukan proses pemurnian, pemisahan H_2O dan Hg , serta pendinginan dan penurunan tekanan secara bertahap hingga hasil akhir berupa LNG. Skema dari pembuatan LNG ditunjukkan dalam Gambar 2.1.





Gambar 2.1 Skema Proses Pembuatan LNG di PT Badak

Sumber: PT Badak

Utilities merupakan sarana penunjang untuk memenuhi kebutuhan yang diperlukan untuk proses pengolahan gas alam serta penyediaan air dan listrik untuk *community*. *Utilities* terbagi atas *On-Plot Utilities* dan *Off-Plot Utilities*.

Tugas utama yang ditangani oleh *Storage & Loading Section* adalah sebagai berikut:

- Menampung dan menyimpan hasil produksi LNG dan LPG hingga pengapalannya.
- Menanggulangi sistem *blowdown*, *flare* dan *burn pit*.
- Memproduksi nitrogen gas dan gas cair untuk keperluan LNG tanker.

PT Badak memiliki 3 *Loading Section*, yaitu *Dock-1* yang memiliki 5 *Arm* untuk LNG, *Dock-2* dan *Dock-3* yang memiliki 5 *Arm* untuk LNG dan 4 *Arm* untuk LPG.

2.2 Loading Arm

Loading Arm adalah lengan silinder besar yang berfungsi sebagai penghubung antara dermaga dan kapal tanker LNG untuk menyalurkan LNG dan LPG. *Loading Arm* ini dapat bergerak bebas yang dioperasikan melalui *Jetty Head Control Panel* maupun menggunakan *radio cordless* dengan PLC sebagai *interface*.

Gambar *Loading Arm* PT Badak ditunjukkan dalam Gambar 2.2.



Gambar 2.2 *Loading Arm* PT Badak

Sumber: PT Badak

Terdapat 2 jenis *Loading Arm* yang ada di PT Badak, yaitu jenis *Double Counterweight Marine Arm* (DCMA) untuk LNG dan jenis *Rotating Counterweight Marine Arm* (RCMA).

Pada dasarnya jenis arm ini tidak berbeda, yaitu memiliki *base riser/ slewing*, *inboard arm*, *outboard arm*, dan *elbow*.

- *Base Riser*: Merupakan bagian yang digunakan untuk menopang tempat *arm* diletakkan dan berfungsi menyalurkan LNG antara pipa penyalur dari *storage tank* ke *arm*. Terdapat *swivel joint* yang mendukung pada bagian *inboard arm*. Dan terdapat bagian *slewing right* serta *slewing left*.
- *Inboard Arm*: Merupakan lengan penghubung antara *base riser* dengan *outboard arm* untuk menyalurkan LNG. Terdapat *swivel joint* yang bergerak ke atas dan ke bawah untuk mendukung bagian *outboard arm*.
- *Outboard Arm*: Merupakan lengan penghubung antara *inboard arm* dengan *elbow*. Pada ujung *outboard arm* terdapat *joint* yang dapat bergerak bebas, sehingga *elbow* dapat ditempatkan pada kapal dengan tepat.

- *Elbow*: Merupakan bagian akhir dari *Loading Arm*. Terdapat *Double Ball Valve* yaitu *upper valve* dan *lower valve*. Bagian ini berfungsi sebagai pengaman jika terjadi pergerakan kapal yang ekstrim. Jadi *upper valve* dan *lower valve* akan terpisah. Bagian *upper valve* akan tertinggal di *arm* sedangkan *lower valve* akan tertinggal di kapal.

2.3 Modul Sensor Potensiometer

Sensor adalah suatu alat yang dapat mengkonversi suatu besaran fisik menjadi besaran elektrik berupa sinyal analog. Jenis Sensor yang dipakai dalam perancangan ini adalah modul sensor potensiometer yang ditunjukkan dalam Gambar 2.3.

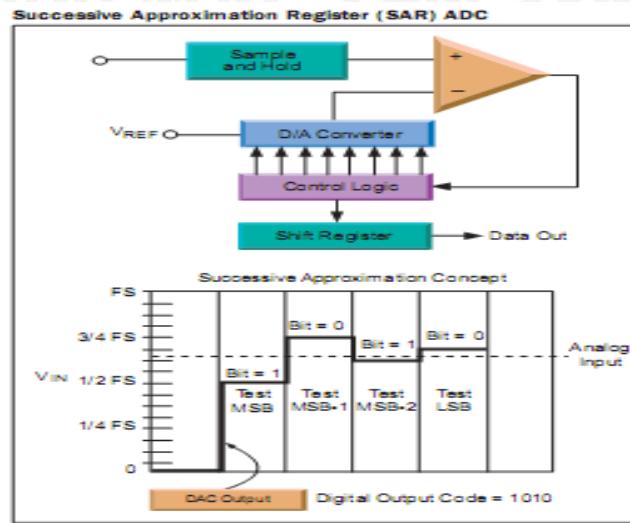


Gambar 2.3 Modul Sensor Potensiometer

Potensiometer tersebut berfungsi sebagai sensor yang mengeluarkan tegangan $0 \sim 5$ volt dan nantinya akan dikonversi oleh ADC. Potensiometer ini memiliki nilai resistansi 0Ω yang mewakili 0 volt sampai dengan $2,2 \text{ k}\Omega$ yang mewakili 5 volt dan memiliki *error* $\pm 10\%$. Modul sensor ini biasa digunakan sebagai pengukur sudut dalam suatu putaran.

2.4 Analog to Digital Converter (ADC)

Rangkaian ini berfungsi sebagai pengubah besaran sinyal analog menjadi sinyal digital. Jadi sinyal masuk yang berupa sinyal analog seperti *transmitter* tersebut akan diolah menjadi sinyal digital (1 atau 0). Proses pengubahan ini dikenal juga dengan nama sistem akusisi data. ADC umumnya digunakan sebagai perantara antara sensor dengan kontroler. ADC banyak digunakan dalam proses industri, komunikasi data dan rangkaian pengukuran. Skema dari rangkaian ADC ditunjukkan dalam Gambar 2.4.

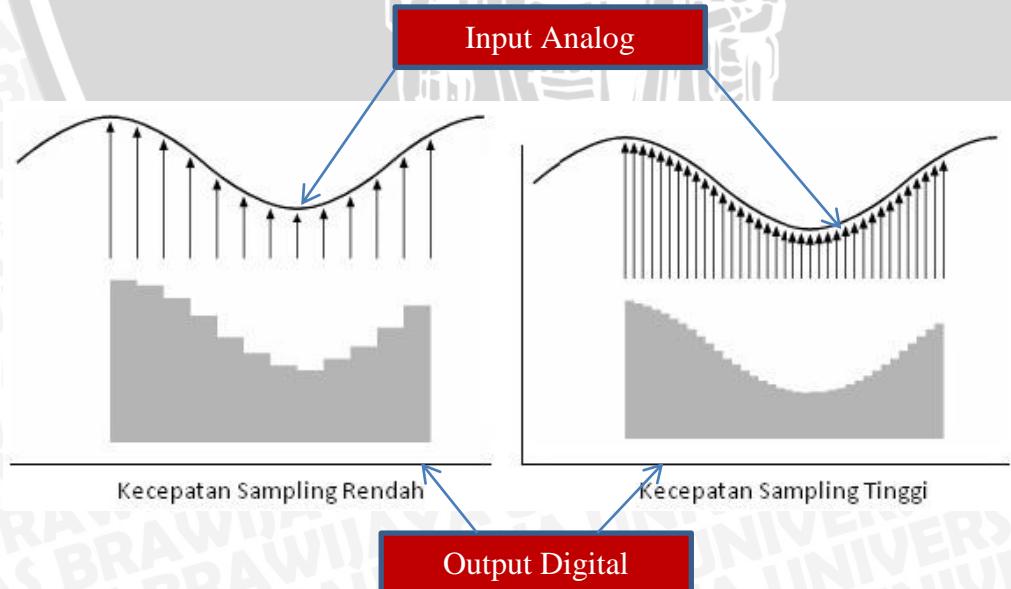


Gambar 2.4 Skematik Rangkaian ADC

Sumber: liu.diva-portal.org

Terdapat empat macam ADC yang memenuhi standar industri, yaitu *integrating*, *tracking converter*, *successive approximation* dan *flash/parallel*. Keempat jenis ADC tersebut mewakili beberapa macam pertimbangan diantaranya resolusi, kecepatan *sampling* dan biaya.

1. Kecepatan *sampling* suatu ADC menyatakan seberapa sering sinyal analog dikonversikan ke bentuk sinyal digital pada selang waktu tertentu. Kecepatan *sampling* biasanya dinyatakan dalam *sample per second* (SPS). *Sampling* dari ADC ditunjukkan dalam Gambar 2.5



Gambar 2.5 Sampling ADC

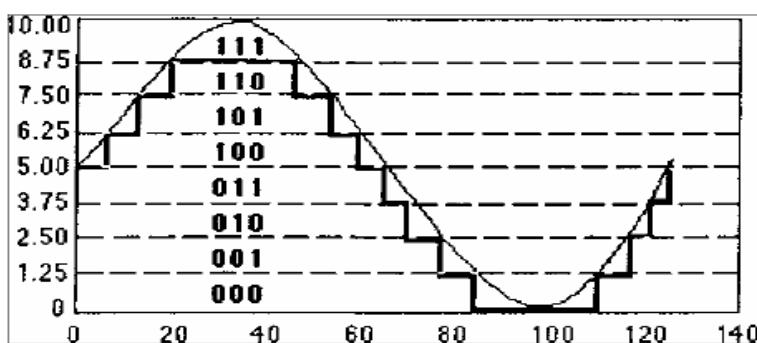
Sumber: <http://www.elektronika-dasar.web.id>

2. Resolusi ADC menentukan ketelitian nilai hasil konversi ADC.
- Contoh 1 : ADC 8 bit akan memiliki *output* 8 bit data digital, ini berarti sinyal *input* dapat dinyatakan dalam 255 ($2^n - 1$)
 - Contoh 2 : ADC 12 bit nilai diskrit. ADC 12 bit memiliki 12 bit *output* data digital, ini berarti sinyal *input* dapat dinyatakan dalam 4096 nilai diskrit.

Prinsip kerja ADC adalah mengkonversi sinyal analog ke dalam bentuk besaran yang merupakan rasio perbandingan sinyal *input* dan tegangan referensi, dan *output* dari rangkaian ini berupa sinyal digital (1 atau 0). Sinyal analog berarti banyaknya masukan yang berasal dari *transmitter* karena adanya perubahan kondisi dari sensor. Masukan tersebut akan diolah terlebih dahulu menjadi sinyal digital sebelum sinyal tersebut diolah di dalam kontroler. Keluaran digital bisa berderet (bit demi bit) atau berjajar dengan semua bit yang disandikan disajikan serentak.

Contoh bila tegangan referensi 5 volt, tegangan *input* 3 volt, (rasio *input* terhadap referensi adalah 60%). Jadi, jika menggunakan ADC 8 bit dengan skala maksimum 255, akan didapatkan sinyal digital sebesar $60\% \times 255 = 153$ (bentuk desimal) atau 10011001 (bentuk biner).

$$\begin{aligned}\text{Sinyal} &= (\text{sample/nilai maks}) * \text{reference_voltage} \\ &= (153/255) * 5 \\ &= 3 \text{ volt}\end{aligned}$$



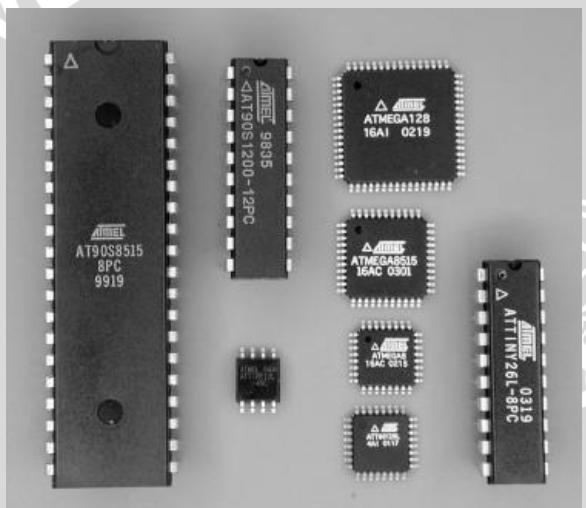
Gambar 2.6 Gelombang Sinus Terdigitasi dengan resolusi 3 bit

Sumber: www.ese.upenn.edu

Gambar 2.6 merupakan hasil dari gelombang sinus yang terdigitalisasi dengan resolusi 3 bit. Semakin tinggi nilai resolusi, maka perubahannya semakin halus.

2.5 Mikrokontroler

Mikrokontroler adalah piranti elektronik berupa IC (*Integrated Circuit*) yang memiliki kemampuan manipulasi data (informasi) berdasarkan suatu urutan instruksi (program) yang dibuat oleh *programmer*. Mikrokontroler merupakan contoh suatu sistem komputer sederhana yang masuk dalam kategori *embedded* komputer. Dalam sebuah struktur mikrokontroler akan kita temukan juga komponen-komponen seperti: *processor*, *memory*, *clock*, dan lain-lain. Contoh mikrokontroler ditunjukkan dalam Gambar 2.7.



Gambar 2.7 Contoh Beberapa Bentuk Mikrokontroller ATMEL

Sumber: www.atmelmikrokontroler.wordpress.com

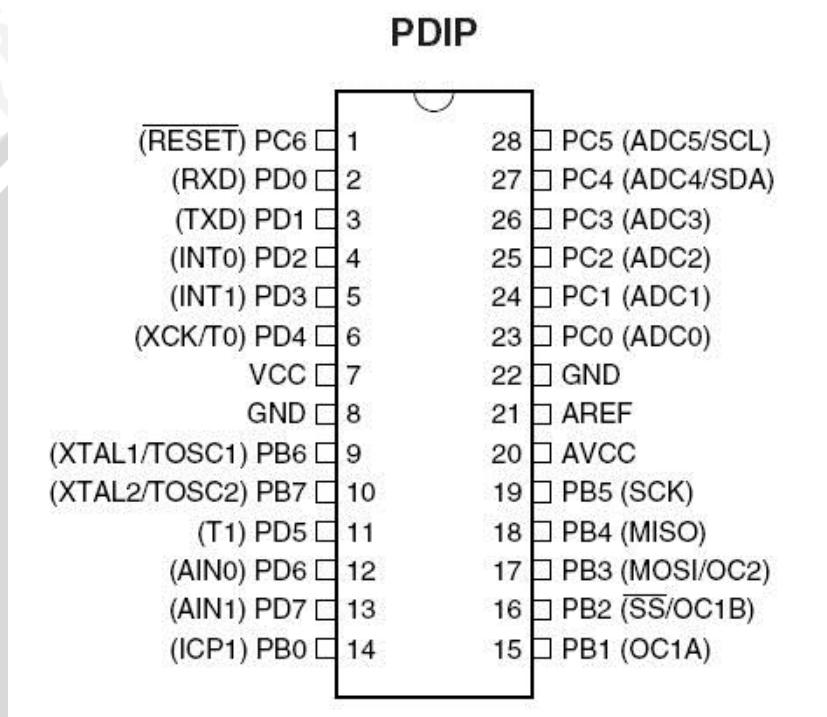
2.5.1 ATMEL AVR ATMega8

ATMega8 merupakan jenis IC mikrokontroler ATMEL AVR RISC yang memiliki beberapa karakteristik sebagai berikut:

- ATMEL AVR RISC memiliki fasilitas dan kefungsian yang lengkap dengan harga yang relatif murah.
- Kecepatan maksimum eksekusi instruksi mikrokontroler mencapai 16 MIPS (*Million Instruction per Second*), yang berarti hanya dibutuhkan 1 *clock* untuk 1 eksekusi instruksi.

- Konsumsi daya yang rendah jika dibandingkan dengan kecepatan eksekusi instruksi.
- Ketersediaan kompiler C (CVAVR) yang memudahkan *user* dalam menggunakan bahasa C.

Chip AVR ATMega8 memiliki 28 pin kaki dengan fungsi pin yang berbeda-beda, berikut skema kaki ATMega8:



Gambar 2.8 Konfigurasi Pin ATMega8

Sumber: *Datasheet*

Dari skema Gambar 2.8 ditunjukkan bahwa ATMega8 memiliki 3 buah *port input/output* 8 bit, yaitu PORT B, PORT C, dan PORT D. Selain sebagai *input/output* masing-masing *port* juga memiliki fungsi yang lain. Misalnya PORT C dapat difungsikan sebagai ADC (*Analog to Digital Converter*). Fungsi-fungsi yang lain dapat dilihat pada *datasheet* ATMega8.

Mikrokontroler ATMega8 memiliki karakteristik utama sebagai berikut:

- 1) Mikrokontroler 8 bit dengan performasi tinggi dan daya rendah.
- 2) Sistem *self-programmable flash* 8 byte.

- 3) EEPROM sebesar 512 byte dan 1 KByte SRAM internal.
- 4) Memiliki 23 saluran I/O dan 32 *general purpose register*.
- 5) Dua *timer/counter* 8 bit dengan *prescaler* terpisah, satu mode pembanding (*compare mode*).
- 6) Satu *timer/counter* 16 bit dengan *prescaler* terpisah, mode pembanding dan perekam (*capture*).
- 7) *Internal* dan *eksternal interrupt*.
- 8) Enam pin ADC dengan resolusi konversi sebesar 10 bit.
- 9) USART (*Universal Syncronous and Asynchronous Receiver Transmitter*).

2.5.2 ADC Mikrokontroler ATMega8

Mikrokontroler ATMega8 memiliki fasilitas 6 *port* ADC yang terintegrasi dengan pilihan 8 dan 10 bit. ADC tersebut terhubung dengan analog *multiplexer* 8 *channel* untuk memproses keenam sinyal analog tersebut. Enam *channel* tersebut terhubung pada PIN.C0 sampai dengan PIN.C5.

Beberapa fitur ADC pada ATMega8, yaitu:

- 1) Resolusi 8 bit dan 10 bit.
- 2) Batas tegangan *input* ADC 0 ~ 5 volt.
- 3) Memiliki tegangan referensi internal 2,56 volt dan 1,1 volt.
- 4) Memiliki mode *free running* atau *single conversion*.
- 5) *Interupt* ketika konversi selesai.

2.5.3 Komunikasi Serial ATMega8

Mikrokontroler AVR Atmega 8 memiliki *Port* USART pada pin 2 dan pin 3 untuk melakukan komunikasi data antara mikrokontroler dengan mikrokontroler ataupun mikrokontroler dengan komputer. USART dapat difungsikan sebagai transmisi data sinkron, dan asinkron. Sinkron berarti *clock* yang digunakan antara *transmiter* dan *receiver* satu sumber *clock*. Sedangkan asinkron berarti *transmitter* dan *receiver* mempunyai sumber *clock* sendiri-sendiri. USART terdiri atas tiga blok yaitu *clock generator*, *transmitter* dan *receiver*.

2.6 Komunikasi Serial

2.6.1 RS-232

RS-232 merupakan rangkaian elektronik yang berfungsi sebagai antarmuka (*interface*) dalam proses pengiriman data antar komputer dalam bentuk serial. RS-232 merupakan kependekan dari *Recommended Standard Number 232*. Antarmuka ini dibuat untuk *interface* antara peralatan terminal data dan peralatan komunikasi data, dengan menggunakan data biner serial sebagai data yang ditransmisikan. Serial *interface* RS-232 memberi ketentuan logika tegangan sebagai berikut :

- Logika 1 disebut “*mark*” merupakan tegangan antara -3 V hingga -15 V.
- Logika 0 disebut “*space*” merupakan tegangan antara +3 V hingga +15 V.

Daerah tegangan antara -3 V hingga +3 V adalah *invalid level*, yaitu daerah tegangan yang tidak memiliki logika sehingga daerah itu harus dihindari. Suatu saluran data RS-232 yang memberi keadaan tegangan ini berarti ada kesalahan. Demikian pula pada saluran pada daerah lebih negatif dari -15 V daerah lebih positif dari +15 V.

2.6.2 RS-485

RS-485 adalah komunikasi data serial yang metode pengoperasianya adalah *differential*, artinya level tegangan pengoperasianya adalah diferensial dari *Line A* dan *Line B*. *Line* untuk komunikasi serial RS-485 adalah 4 *wire*, 2 *wire* untuk *transmitter* dan 2 *wire* lagi untuk *receiver*, 2 *wire* yang satu disebut dengan *Line A*, *B Transmitter* dan 2 *wire* lainnya *Line A, B Receiver*, nah sinyal yang disebut sinyal diferensial itu yaitu adalah diferensial antara *Line A* dan *Line B*. Komunikasi ini dapat melakukan pemindahan data sampai dengan jarak 1,2 km. Terdapat 3 Pin dalam rangkaian komunikasi RS-485 yaitu *Data+*, *Data-* dan *Ground*. RS-485 dan RS-232 memiliki *interface* yang sama, hanya pengoperasianya saja yang berbeda.

Komunikasi serial RS-485 menggunakan sepasang kabel untuk mengirimkan satu sinyal tegangan antara kedua kabel saluran selalu berlawanan. Logika ditentukan dari beda tegangan antara kedua kabel tersebut. SN75176 merupakan IC *multipoint RS-485 transceiver*. SN75176 dapat mendukung 32 unit parallel dalam satu jalur. Sensitivitas tegangan input receiver 0,2 V dan jarak maksimum 4000 feet.

Pada mode pengiriman (*transmitting*), kaki *enable* kirim DE diberi logika 1. Keluaran A dan B ditentukan oleh masukan *driver* D, dimana keluaran A akan sesuai dengan logika *driver* D, sedangkan B berkebalikan. Jika *input* D berlogika 1, maka *output* A akan bertegangan 5 Volt dan *output* B 0 Volt. Sebaliknya jika *input* D berlogika 0 maka *output* A bertegangan 0 Volt dan *output* B 5 Volt. Pada mode penerimaan (*receiving*), kaki *enable* terima RE diberi logika 0. *Output receiver* R ditentukan oleh tegangan diferensial antara *input* A dan B (VA - VB). Jika tegangan diferensial VA - VB lebih besar dari +0,2 Volt, maka *receiver* R akan berlogika 1, sedangkan jika VA - VB lebih kecil dari -0,2 Volt maka *receiver* R akan berlogika 0. Untuk tegangan VA - VB antara -0,2 Volt sampai +0,2 Volt, maka level logika keluaran tidak terdefinisi.

2.6.3 Converter RS-232 to RS-485

Converter RS-232 to RS-485 adalah alat konverter komunikasi serial dari RS-232 menjadi RS-485 atau sebaliknya. Perangkat ini biasanya digunakan untuk menghubungkan perangkat yang umumnya menggunakan komunikasi serial RS-232 dan jarak antar perangkat yang cukup jauh. Perangkat ini mempunyai *socket female* DB-9 pada bagian RS-232 dan *socket male* DB-9 pada bagian RS-485. Contoh konverter RS-232 to RS-485 ditunjukkan dalam Gambar 2.9.



Gambar 2.9 Contoh Konverter RS-232 to RS-485

Sumber: www.annesa.com

2.7 Pemrograman Visual

Pemrograman *visual* adalah metode pembuatan program dimana *programmer* membuat koneksi antara objek-objek dengan cara menggambar, menunjuk dan mengklik pada diagram dan *icon*, dan berinteraksi dengan diagram alur. Jadi, *programmer* bisa menciptakan program dengan cara mengklik pada *icon* yang mewakili rutin-rutin pemrograman secara umum. Contoh pemrograman *visual* antara lain yaitu *Microsoft Visual Studio 2012 Express*. Tampilan aplikasi *software* *Microsoft Visual Studio 2012 Express* ditunjukkan dalam Gambar 2.10.



Gambar 2.10 Tampilan awal aplikasi *Microsoft Visual Studio 2012 Express*

Microsoft Visual Studio Express 2012 merupakan salah satu bahasa pemograman (*development language*) yang digunakan untuk merancang suatu aplikasi program berbasis grafis. Selain itu *Microsoft Visual Studio Express 2012* dapat juga membuat program berbasis jaringan dan internet.

Keunggulan dari *Microsoft Visual Studio* antara lain:

- 1) IDE (*Integrated Development Environment*) atau lingkungan pengembangan aplikasi sendiri, di dalamnya terdapat menu-menu yang memudahkan kita untuk membuat suatu proyek program.
- 2) Proses kompilasi cepat, pada saat aplikasi yang kita buat dijalankan pada *Microsoft Visual Studio*, maka secara otomatis akan dibaca sebagai sebuah program, tanpa dijalankan terpisah.

- 3) Mudah digunakan, *source code Microsoft Visual Studio* yang merupakan turunan dari *basic*, sehingga tidak diperlukan suatu penyesuaian lagi.
- 4) Bersifat *multi purphase*, artinya bahasa pemrograman *Microsoft Visual Studio* dapat digunakan untuk mengembangkan berbagai keperluan pengembangan aplikasi.



UNIVERSITAS BRAWIJAYA



BAB III

METODOLOGI

Untuk menyelesaikan perumusan masalah dan merealisasikan tujuan penelitian yang ditunjukkan dalam bab pendahuluan, maka diperlukan metode untuk menyelesaikan masalah tersebut. Langkah-langkah yang diperlukan untuk menyelesaikan alat yang dirancang adalah penentuan spesifikasi alat, perancangan dan pembuatan alat, pengujian alat, dan pengambilan kesimpulan.

3.1 Spesifikasi Alat

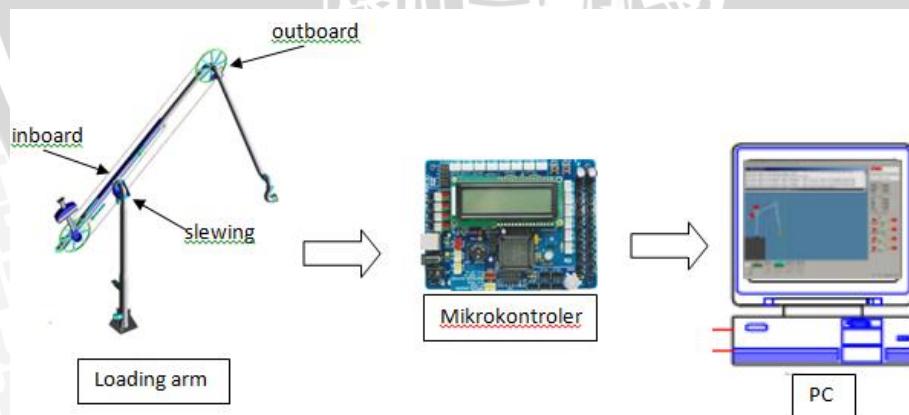
Alat yang diteliti memenuhi spesifikasi sebagai berikut:

- 1) Modul Sensor potensiometer $2,2\text{ k}\Omega \pm 10\%$, linearitas $0,1\%$.
- 2) ADC 10bit dengan frekuensi 1MHz.
- 3) RS-232 dengan Baudrate 9600 sebagai antarmukanya.
- 4) ATMega8 untuk mengkonversi nilai ADC menjadi besaran sudut.

3.2 Perancangan Alat

3.2.1 Diagram Blok

Pembuatan diagram blok merupakan dasar dari perancangan sistem agar perancangan dan perealisasian alat berjalan secara sistematis. Gambar 3.1 menunjukkan diagram blok alat.



Gambar 3.1 Diagram Blok Sistem Monitoring *Loading Arm*

Penjelasan mengenai diagram blok rancangan elektronik dalam Gambar 3.1 adalah sebagai berikut:

- 1) Sensor potensiometer yang terdapat pada bagian *slewing*, *inboard*, dan *outboard Loading Arm* merupakan sensor yang dapat membaca gerakan dari bagian-bagian tersebut sehingga dapat diukur berapa besar sudutnya.
- 2) ADC pada mikrokontroler merupakan rangkaian untuk merubah besaran sinyal analog dari sensor antara 0 ~ 5 volt menjadi bilangan biner 1 dan 0.
- 3) Mikrokontroler ATMega8 digunakan sebagai proses pengkonversian data dari bentuk digital menjadi bentuk sudut.
- 4) PC/komputer digunakan sebagai tampilan akhir yang dihubungkan dengan converter RS-232 to RS-485 dengan mikrokontroler.

3.2.2 Deskripsi Cara Kerja Alat

Sensor pada *Loading Arm* akan mengeluarkan sinyal analog antara 0 ~ 5 volt. Sinyal tersebut masuk melalui port ADC yang terdapat pada mikrokontroler. Sinyal ADC dikonversi dari bentuk analog menjadi sinyal digital dan akan diubah dalam bentuk sudut. Nilai dari sudut inilah yang nantinya akan deprogram menjadi batasan-batasan untuk memonitor *Loading Arm*. Terdapat 3 alarm peringatan dalam batasan tersebut. Hasil dari perubahan gerak itu ditampilkan dalam pemograman *visual* pada komputer agar perubahan gerak tersebut lebih dapat terlihat.

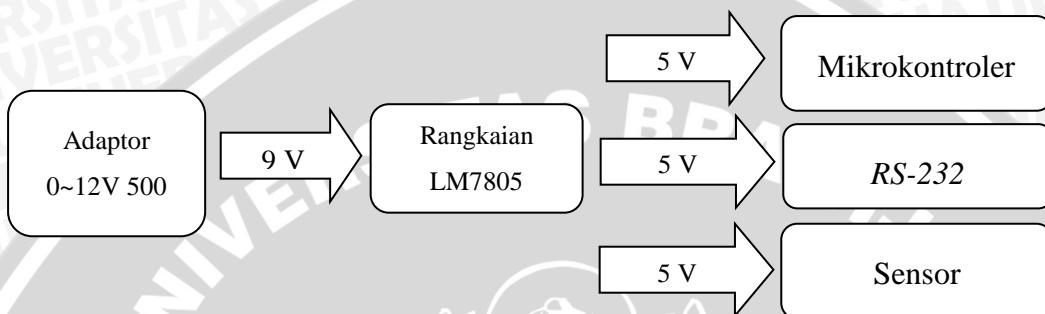
3.2.3 Perancangan Perangkat Keras

3.2.3.1 Rangkaian Catu Daya

Dalam perancangan catu daya dirancang berdasarkan hasil perhitungan daya beban. Beban-beban tersebut adalah sensor potensiometer, mikrokontroler, dan rangkaian MAX-232.

- Arus untuk mencatu mikrokontroler sebesar 15 mA. Daya yang dibutuhkan mikrokontroler sebesar $5 \text{ V} \times 15 \text{ mA} = 75 \text{ mW}$.
- Arus yang dibutuhkan oleh 1 buah sensor potensiometer sebesar 2,3 mA. Daya yang dibutuhkan oleh 3 buah sensor potensiometer sebesar $5 \text{ V} \times 2,3 \text{ mA} \times 3 \text{ buah} = 34,5 \text{ mW}$.
- Arus untuk mencatu rangkaian MAX-232 sebesar 22 mA. Daya yang dibutuhkan rangkaian MAX-232 sebesar $5 \text{ V} \times 22 \text{ mA} = 120 \text{ mW}$.

- Arus untuk mencatu modul converter RS-232 to RS-485 sebesar 40mA. Daya yang dibutuhkan oleh 2 buah modul converter RS-232 to RS-485 sebesar $5\text{ V} \times 40\text{ mA} \times 2\text{ buah} = 400\text{ mW}$
- Daya untuk mencatu rangkaian led sebesar $5\text{ V} \times (20\text{ mA} + 15\text{ mA} + 15\text{ mA}) = 250\text{ mW}$



Gambar 3.2 Diagram Blok Catu Daya

Rangkaian catu daya yang digunakan adalah satu buah rangkaian catu daya menggunakan LM7805. Diagram blok rangkaian catu daya ditunjukkan dalam Gambar 3.2.

Berdasarkan perhitungan daya beban adaptor dengan spesifikasi 0~12 V, 500 mA cukup untuk mencatu seluruh beban. Dengan total arus $133.6\text{ mA} \approx 150\text{ mA}$, sehingga arus beban lebih kecil dibandingkan arus adaptor.

Dioda D1 berfungsi sebagai pengaman rangkaian jika terjadi pemasangan polaritas catu terbalik. Tegangan A ditunjukkan dalam Persamaan (3.1)

$$V_A = 9 - V_{D1}$$

$$V_{D1} = 0.7 \text{ volt}$$

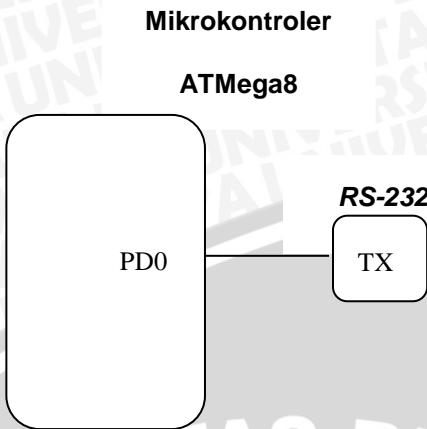
$$V_A = 9 - 0.7$$

$$V_A = 8.3 \text{ volt} \quad (3.1)$$

3.2.3.2 Perancangan Mikrokontroler

Perencanaan pin-pin mikrokontroler ATMega8 dijelaskan dalam diagram blok rangkaian mikrokontroler yang ditunjukkan dalam Gambar 3.3.





Gambar 3.3 Diagram Blok Rangkaian Mikrokontroler

Reset yang digunakan adalah *power on reset* yang bekerja ketika tegangan catu di bawah *power on reset threshold* (V_{pot}) yang memiliki tegangan tipikal sebesar 2,3 V.

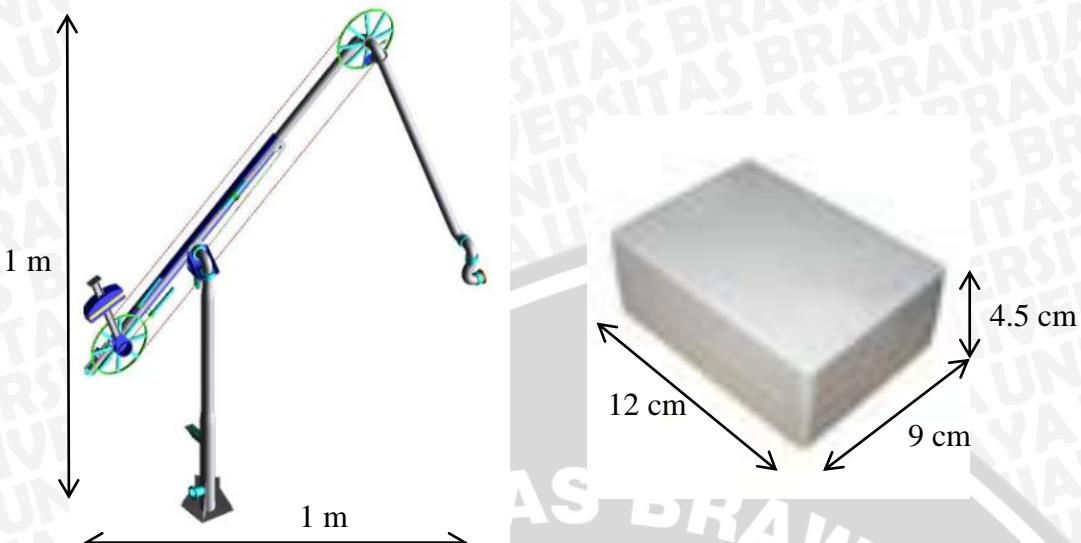
Clock yang digunakan adalah *clock* eksternal mikrokontroler ATMega 8 sebesar 16 MHz sehingga mengatur *fuse bit* seperti ditunjukkan dalam Tabel 3.1

Tabel 3.1 Pengaturan Fuse Bit

No	Fuse bit	Nilai
1	CKSEL 3..0	1111
2	CKOPT	0
3	SUT 1..0	10

3.2.3.3 Perancangan Kerangka Alat

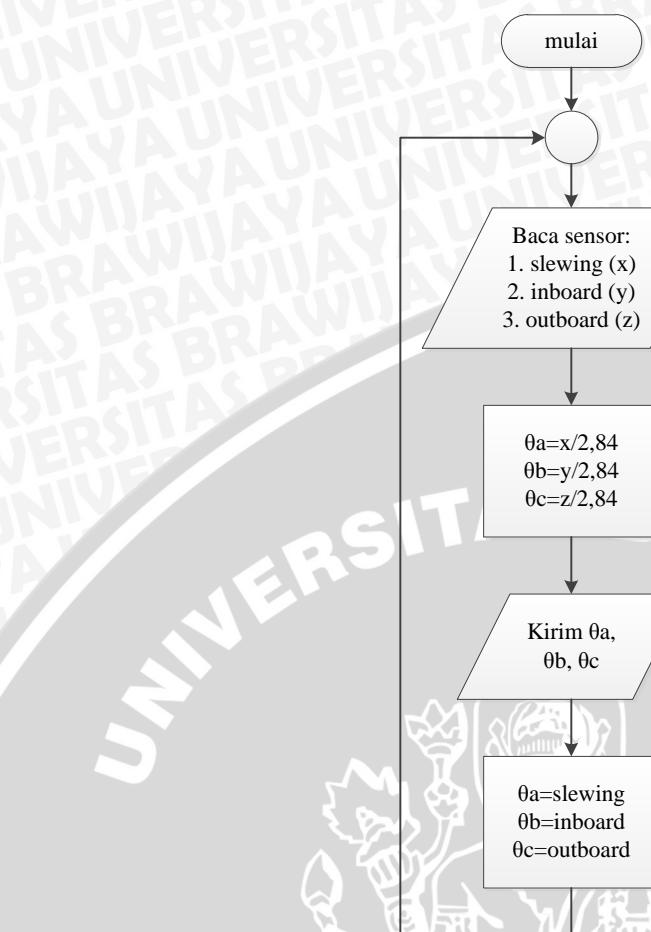
Bentuk kerangka miniature *Loading Arm* dirancang memiliki tinggi ± 1 m dan panjang ± 1 m dan untuk box elektriknya berukuran 12 cm x 9 cm x 4,5 cm. Bentuk kerangka *Loading Arm* dan box eletrik ditunjukkan dalam Gambar 3.4.



Gambar 3.4 Bentuk Kerangka Miniatur *Loading Arm* dan Box Elektrik

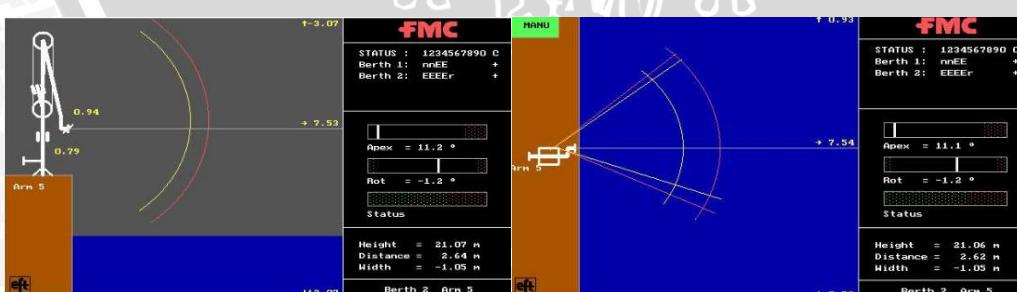
3.2.4 Perancangan Perangkat Lunak

Untuk menghubungkan perangkat keras dengan hasil keluaran yang diinginkan, maka dibuatlah perangkat lunak. Perancangan perangkat lunak ini bertujuan untuk memonitor dan mengatur kerja dari perangkat keras alat ini. Rancangan dan parameter-parameter yang didapat dari data-data diterapkan ke dalam mikrokontroler ATMega8 dengan bahasa C dan *compiler* CodeVision AVR. Diagram alur perencanaan perangkat lunak pada CV AVR ditunjukkan dalam Gambar 3.5.



Gambar 3.5 Diagram Alur Program di CV AVR

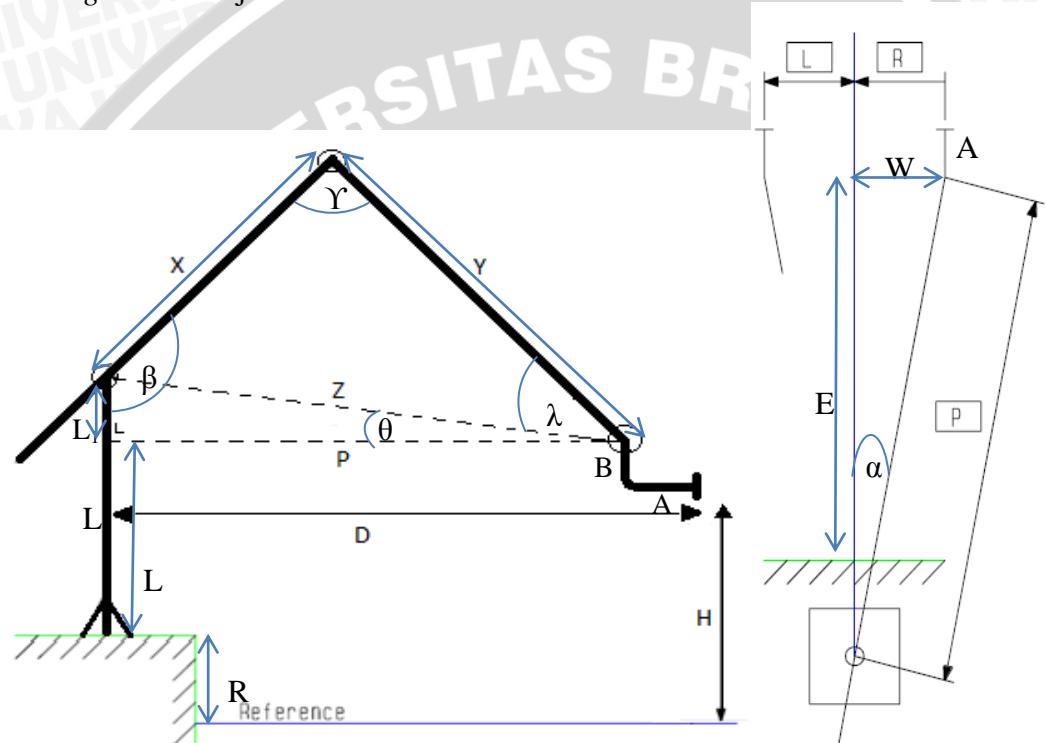
Selanjutnya keluaran tersebut dikirim melalui komunikasi serial untuk ditampilkan di dalam komputer/*console*. Hasil tersebut dihubungkan dengan *software* Visual Studio dan akan keluar berupa gambar yang dapat bergerak. Contoh tampilan program ditunjukkan dalam Gambar 3.6.



Gambar 3.6 Tampilan Program di Komputer

3.3 Position Monitoring System (PMS)

Dalam pembuatan sistem monitoring *Loading Arm* ini diperlukan suatu metode untuk mendapatkan nilai-nilai parameter yang akan dicari. Terdapat beberapa parameter yang harus dicari yaitu jarak antara siku *elbow* dengan *base riser* (DEPTH), jarak antar ujung *elbow* dengan titik tengah *slewing* (WIDTH), jarak antara ujung *elbow* dengan *base riser* secara tegak lurus (DISTANCE), jarak antara ujung *elbow* dengan referensi tinggi permukaan air laut (HEIGHT). Parameter *Loading Arm* ditunjukkan dalam Gambar 3.7.



Gambar 3.7 Parameter *Loading Arm*

Diketahui nilai X, Y, L, B, A, R, β , γ . Berdasarkan Gambar 3.7 dan parameter yang diketahui, untuk menentukan DEPTH (P) didapat rumus untuk mendapatkan nilai Z yang ditunjukkan dalam Persamaan (3.2).

$$\begin{aligned} Z^2 &= X^2 + Y^2 - 2XY\cos \gamma \\ Z &= \sqrt{X^2 + Y^2 - 2XY\cos \gamma} \quad (3.2) \end{aligned}$$

Setelah mendapatkan nilai Z, nilai sudut didapatkan dalam Persamaan (3.3).

$$\begin{aligned} X^2 &= Y^2 + Z^2 - 2YZ \cos \lambda \\ 2YZ \cos \lambda &= Y^2 + Z^2 - X^2 \\ \cos \lambda &= \frac{Y^2 + Z^2 - X^2}{2YZ} \\ \lambda &= \cos^{-1} \frac{Y^2 + Z^2 - X^2}{2YZ} \end{aligned} \quad (3.3)$$

Untuk mencari θ menggunakan rumus yang ditunjukkan dalam Persamaan (3.4).

$$\begin{aligned} 360^\circ &= \beta + \gamma + \lambda + \theta + 90^\circ \\ \theta &= 270^\circ - \beta - \gamma - \lambda \end{aligned} \quad (3.4)$$

Untuk mendapatkan nilai DEPTH (P) menggunakan rumus yang ditunjukkan dalam Persamaan (3.5).

$$\begin{aligned} e &= 51 * \cos(\beta) \\ f &= 61 * \sin(\beta) \\ P &= e + f \end{aligned} \quad (3.5)$$

Untuk mendapatkan nilai WIDTH (W) menggunakan rumus yang ditunjukkan dalam Persamaan (3.6).

$$\begin{aligned} G &= \sin(\beta) * p \\ W &= g * \tan(\beta) + (8) / \cos(\beta) \end{aligned} \quad (3.6)$$

Untuk mendapatkan nilai DISTANCE (D) menggunakan rumus yang ditunjukkan dalam Persamaan (3.7).

$$D = g + 5 \quad (3.7)$$

Dan untuk menentukan nilai HEIGHT (H) menggunakan rumus yang ditunjukkan dalam Persamaan (3.8).

$$\begin{aligned} \sin \theta &= \frac{L_1}{Z} \\ L_1 &= \sin \theta * Z \\ L_2 &= L - L_1 - B \\ H &= L_2 + R \end{aligned} \quad (3.8)$$

3.4 Pengujian Alat

Untuk memastikan agar sistem ini berjalan sesuai dengan yang diharapkan, maka diperlukan pengujian perangkat keras yang dilakukan secara per blok maupun pengujian keseluruhan sistem.

3.4.1 Pengujian Perangkat Keras

Pengujian ini dilakukan dengan tujuan untuk menyesuaikan nilai-nilai parameter yang ada pada *datasheet* perangkat dengan nilai-nilai yang terukur pada keadaan sebenarnya seperti nilai tegangan dan arus, sehingga komponen dapat bekerja dengan baik. Pengujian perangkat keras ini meliputi :

- Pengujian Modul Sensor Potensiometer

Pengujian ini dilakukan untuk melihat tegangan yang keluar dari modul sensor dengan memberikan sumber tegangan pada inputnya.

- Pengujian Catu Daya

Pengujian ini dilakukan untuk melihat tegangan yang dihasilkan oleh rangkaian catu daya dengan beban dan tanpa beban.

- Pengujian Mikrokontroler

Pengujian ini dilakukan untuk melihat apakah mikrokontroler dapat bekerja dengan baik dengan mengeluarkan tegangan pada pin Tx.

- Pengujian Rangkaian MAX-232

Pengujian ini dilakukan untuk melihat apakah MAX-232 dapat bekerja dengan baik dengan melihat tegangan yg keluar pada pin MAX-232.

- Pengujian Komunikasi Serial

Pengujian ini dilakukan untuk melihat apakah program yang ada di dalam mikrokontroler dapat di terima dengan baik oleh komputer/laptop.



3.4.2 Pengujian Perangkat Lunak

Pengujian ini dilakukan dengan tujuan untuk mengkalibrasi parameter-parameter yang ada pada perangkat keras seperti sensor agar sesuai dengan nilai-nilai parameter yang diinginkan. Pengolahan data dilakukan dengan *software* yang ada pada komputer. Hasil yang sudah dikalibrasi tersebut ditampilkan dalam bentuk *visual* agar lebih terlihat menarik.

3.4.3 Pengujian Keseluruhan

Pengujian sistem secara keseluruhan dilakukan untuk menganalisis kerja dari alat yang dibuat dari segi perangkat keras dan perangkat lunak yang diintegrasikan bersama.





BAB IV

PERANCANGAN DAN PEMBUATAN ALAT

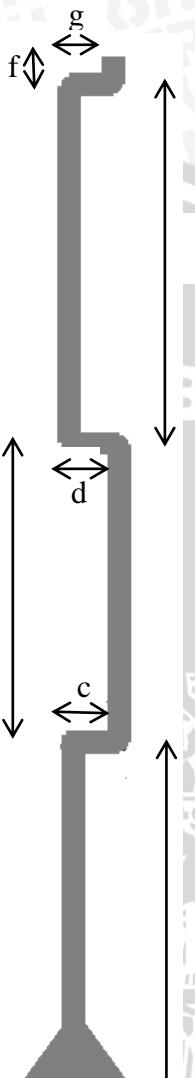
Perancangan alat ini dilakukan secara bertahap dalam bentuk blok, sehingga memudahkan dalam menganalisis setiap bloknya maupun secara keseluruhan. Perancangan ini terdiri atas:

- Perancangan perangkat keras (perancangan rangkaian catu daya, perancangan mikrokontroler ATMega8, Perancangan sistem antarmuka komunikasi serial dengan PC, dan bentuk keseluruhan sistem monitoring *Loading Arm*).
- Perancangan perangkat lunak (perancangan algoritma perangkat lunak pada mikrokontroler sebagai pengkonversi nilai ADC dan pada Visual Studio sebagai keluaran yang diinginkan).

4.1 Perancangan Perangkat Keras

4.1.1 Perancangan Mekanik

Dalam perancangan mekanik dibutuhkan 4 *joint* untuk membuat miniatur *Loading Arm* yang dihubungkan dengan lengan-lengan pipa. Terdapat 1 *joint* seperti engsel pintu yang bergerak ke kanan dan ke kiri, dan 2 *joint* yang bergerak ke atas dan ke bawah, serta 1 *joint* tambahan yang bergerak bebas. Untuk 3 *joint* pertama bergerak aktif, sedangkan 1 *joint* yang bebas bergerak pasif. Pada *joint* pertama terhubung dengan pipa dan menjadi *base* sebagai titik tempat *Loading Arm* berdiri. Perancangan miniatur *Loading Arm* ditunjukkan dalam Gambar 4.1.



Gambar 4.1 Perancangan Miniatur *Loading Arm*

Keterangan gambar:

- | | |
|-------------------------|-------------------------|
| a) Slewing = 62 cm | e) Outboard = 61 cm |
| b) Inboard = 51 cm | f) Elbow = 5 cm |
| c) Engsel Inb. = 9 cm | g) Engsel Elb. = 7,5 cm |
| d) Engsel Out. = 8,5 cm | |

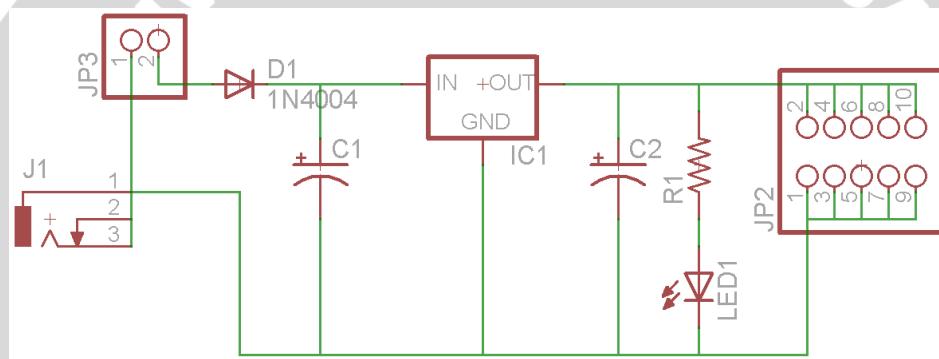
4.1.2 Perancangan Elektrik

Perencanaan bagian perangkat keras ini dapat diuraikan sebagai berikut.

- 1) Perancangan rangkaian Catu Daya.
- 2) Perancangan rangkaian minimum sistem ATMega8.
- 3) Perancangan rangkaian Komunikasi.

4.1.2.1 Perancangan Rangkaian Catu Daya

Pada perancangan catu daya tegangan yang dibutuhkan untuk mengaktifkan sistem adalah 5 volt DC. Rangkaian catu daya secara keseluruhan ditunjukkan dalam Gambar 4.2.



Gambar 4.2 Rangkaian Catu Daya

Catu sumber yang digunakan adalah adaptor DC dengan tegangan 9 volt. Terdapat regulator LM7805 agar menghasilkan tegangan 5 volt. JP3 digunakan sebagai saklar untuk menyalakan atau mematikan sistem.

Besar nilai kedua kapasitor yang digunakan didapat berdasarkan *datasheet* regulator LM7805 yaitu sebesar 0.33 µF untuk C1 dan 0.1 µF untuk C2.

Kemudian pada Ground regulator diberi diode agar jika polaritas dari catu sumber terbalik maka tidak terjadi kerusakan sistem. Perhitungan tegangan sumber yang masuk ke dalam regulator ditunjukkan dalam Persamaan (4.1)

$$V_{in} = V_s - V_{diode}$$

$$V_{in} = 9 - 0,7$$

$$V_{in} = 8,3 \text{ V}$$

(4.1)

LED digunakan sebagai indikator yang menyala jika sudah diberi catu sumber. Perhitungan nilai resistor yang digunakan ditunjukkan dalam Persamaan (4.2).

$$V_o = 5 \text{ volt} \quad V_{f(LED)} = 2,2 \text{ volt} \quad I_{f \max LED} = 20 \text{ mA}$$

$$R_1 = \frac{V_o - V_{f(LED)}}{I_{f \max LED}}$$

$$R_1 = \frac{5-2,2}{20.10^{-3}} = 140 \Omega \quad (4.2)$$

Jadi, R_1 yang digunakan adalah 150Ω .

4.1.2.2 Perancangan Rangkaian Mikrokontroler ATMega8

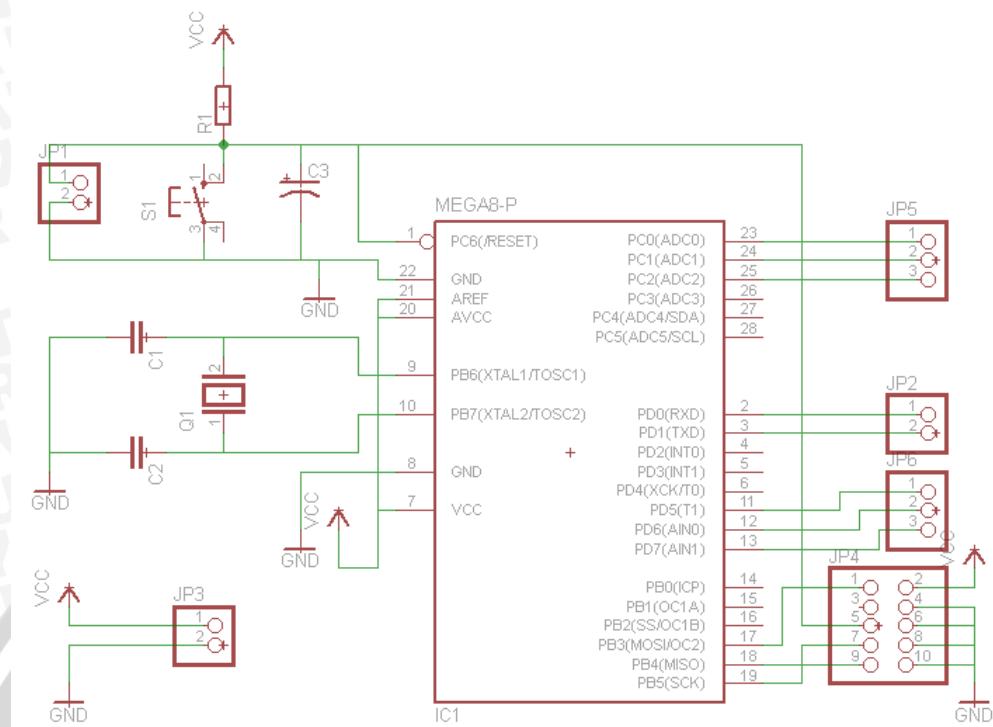
IC mikrokontroler ATMega8 memiliki 3 buah port *input/output* yaitu port B, C, dan D. Masing-masing port memiliki 8 bit *input/output*. Selain berfungsi sebagai masukan dan keluaran pin mikrokontroler juga dapat digunakan dalam fungsi lainnya. Dalam perancangan, pin yang digunakan diperlihatkan dalam Tabel 4.1

Tabel 4.1 Pin yang digunakan pada mikrokontroler ATMega8

No.	Fungsional	Pin Mikrokontroler	No. Pin
1	Pin Reset	C6	1
2	Pin Kristal	B6,B7	9,10
3	Potensiometer Slewing	C0	23
4	Potensiometer Inboard	C1	22
5	Potensiometer Outboard	C2	21
6	Pin Tx/Rx	D1,D0	3,2
7	Pin SPI	B3,B4,B5	17,18,19

Untuk membuat rangkaian mikrokontroler dibentuk sebuah sistem minimum yang dihubungkan dengan beberapa rangkaian pendukung seperti rangkaian reset dan rangkaian kristal. Rangkaian minimum mikrokontroler ATMega8 ditunjukkan dalam Gambar 4.3.

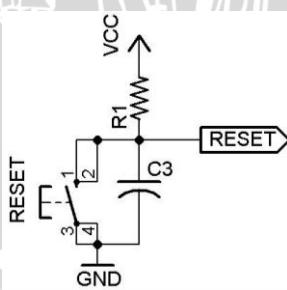




Gambar 4.3 Rangkaian Mikrokontroler ATMega8

Agar mikrokontroler dapat bekerja secara maksimal maka dipilih Kristal sebesar 16 MHz dan minimum sistem untuk rangkaian clock terdapat dalam datasheet ATMega8. Terdapat 2 buah kapasitor pendukung pada kaki-kaki Kristal yang dihubungkan dengan ground.

Terdapat rangkaian reset berfungsi untuk mengembalikan keadaan awal program saat diberi catu ground. Gambar rangkaian reset pada mikrokontroler ditunjukkan dalam Gambar 4.4.



Gambar 4.4 Rangkaian Reset

Rangkaian reset terdiri atas resistor R₁ dan kapasitor C₃ yang dihubungkan ke PIN RESET yang terdapat pada mikrokontroler. Proses reset terjadi ketika diberi

logika rendah selama 1,5 μ s sesuai dengan datasheet ATMega8. Hubungan R_1 dan C_3 ditunjukkan dalam Persamaan 4.3.

$$t = -R_1 C_3 \ln\left(1 - \frac{V_{RST}}{V_{POT}}\right) \quad (4.3)$$

dengan:

t = waktu yang dibutuhkan untuk mereset mikrokontroler, minimal sebesar 1,5 μ s (berdasarkan datasheet ATMega8).

V_{RST} = tegangan logika rendah pada pin reset sebesar $0,1 \times V_{CC}$ (berdasarkan datasheet ATMega8).

V_{POT} = Voltage Power On Threshold sebesar 2.3 volt (berdasarkan datasheet ATMega8).

R_1 = bernilai antara $30 \text{ k}\Omega$ – $80 \text{ k}\Omega$ (berdasarkan datasheet ATMega8).

Dengan menggunakan R_1 sebesar $47 \text{ k}\Omega$ dan C_3 sebesar 1 nF maka:

$$t = -R_1 C_3 \ln\left(1 - \frac{V_{RST}}{V_{POT}}\right)$$

$$t = -47 \cdot 10^3 \cdot 1 \cdot 10^{-9} \ln\left(1 - \frac{0,5}{2,3}\right)$$

$$t = 11.52 \mu\text{s}$$

Nilai t ini sudah memenuhi waktu yang dibutuhkan untuk mereset mikrokontroler.

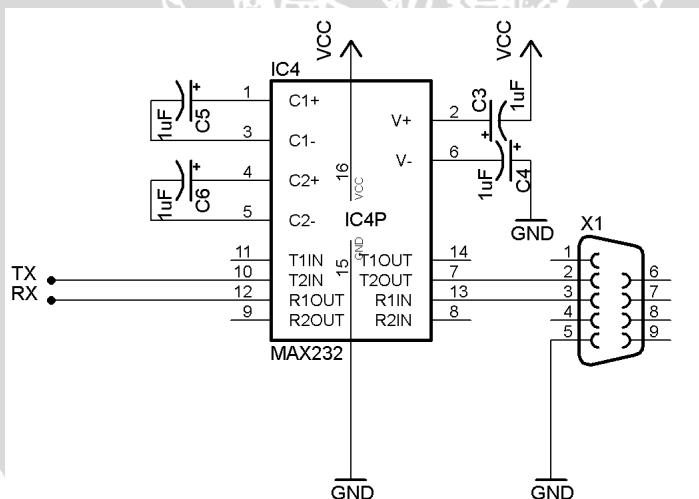
4.1.2.3 Perancangan Rangkaian Komunikasi

Data yang keluar dari rangkaian mikrokontroler berupa level tegangan CMOS. Sedangkan data yang masuk dari PC menggunakan standar komunikasi serial RS-232, level tegangan yang digunakan adalah level tegangan yang berkisar antara -3 volt dan -15 volt untuk kondisi logika "1" atau yang disebut dengan keadaan *mark* dan antara +3 volt dan +15 volt untuk kondisi logika "0" atau disebut dengan keadaan *space* atau dengan kata lain standar RS-232 menggunakan logika negatif/terbalik. Oleh karena itu dibutuhkan sebuah rangkaian untuk mengkonversi perbedaan level tegangan untuk masing-masing logika pada kedua bagian ini, dalam hal ini digunakan komponen MAX232.

IC MAX232 mempunyai 2 *receivers* yang berfungsi sebagai penyesuaian level tegangan level RS-232 ke level *Transistor Transistor Logic* (TTL) ataupun CMOS, dan juga mempunyai 2 *drivers* yang berfungsi mengubah level tegangan dari level TTL/CMOS ke level RS-232. Pasangan *driver/receiver* ini berfungsi sebagai *TX* dan *RX*, sedangkan pasangan yang lainnya berfungsi sebagai *CTS* dan *RTS*.

Nilai kapasitor (C) yang digunakan telah ditentukan sebesar 1uF, nilai ini merupakan konfigurasi datasheet MAX232. Gambar 4.5 menunjukkan rangkaian MAX232. Empat kapasitor dalam rangkaian ini digunakan pada:

- C₁ terhubung pada pin 1 (+) dan pin 3(-), sebagai pengganda tegangan (+5 volt menjadi +10 volt). keluaran hasil proses pengganda tegangan ada di pin 2.
- C₂ terhubung pada pin 4(+) dengan pin 5(-), sebagai pembalik tegangan (+10 volt menjadi -10 volt), keluaran ada di pin 6.
- C₃ terhubung pada pin 6 (-) dan dengan *Ground* (+), untuk penstabil nilai arus yang mengalir.
- C₄ terhubung pada pin 2(+) dengan pin 16(-), untuk penstabil nilai arus yang mengalir.



Gambar 4.5 Rangkaian Penyesuaian Level Tegangan (MAX232)

Jarak antara modul dengan komputer berada sekitar 10 meter, sedangkan RS-232 hanya dapat mengirim dan menerima data sampai dengan jarak \pm 5 meter. Maka diperlukan RS-485 sebagai komunikasi tambahan karena RS-485 dapat mentransmisikan data sampai dengan jarak \pm 1 km.

Karena keluaran dari sistem ini menggunakan komunikasi RS-232 dan komunikasi komputer juga menggunakan RS-232, maka dibutuhkan 2 konverter RS-232 *to* RS-485. Spesifikasi dari *converter RS-232 to RS-485* sebagai berikut:

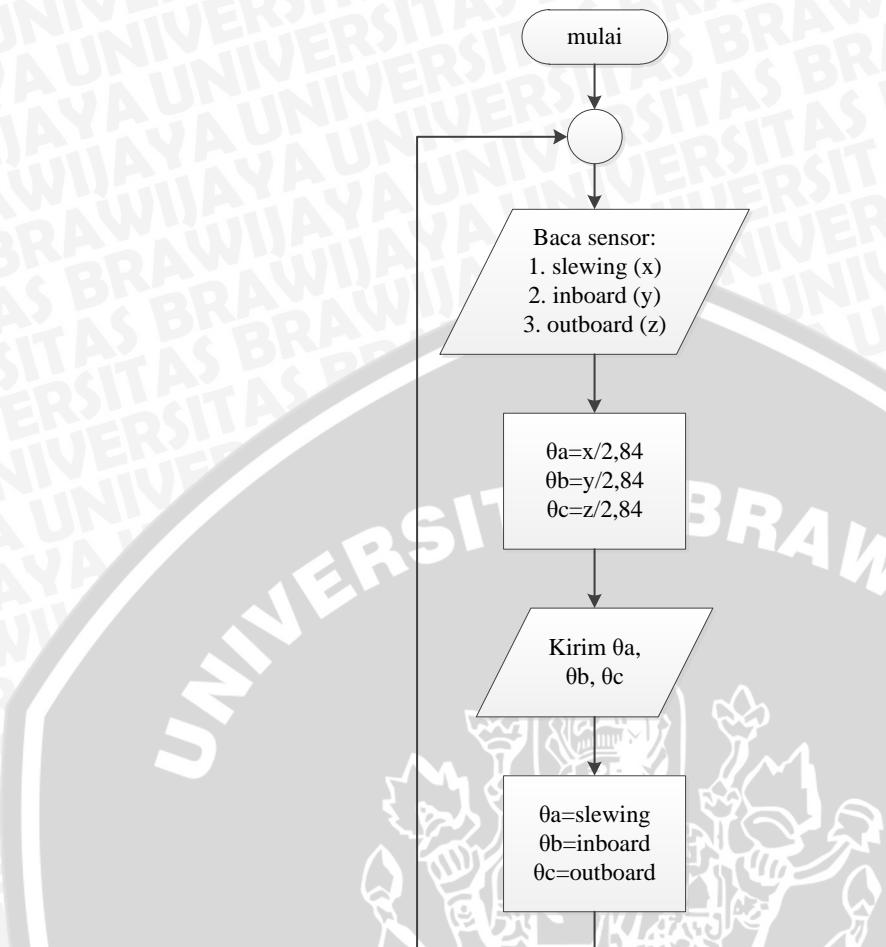
1. *Standard: Accord EIA RS-232, RS-485 standard*
2. *RS-232 signal: TX, RX, GND*
3. *RS-485 signal: D+, D-, GND*
4. *Baud rate: 300-115200bps, auto sense serial rate*
5. *Loading: Support 32 Nodes (can custom 128 nodes) RS-485 side 1200M(9600bps), RS-232 side 5M*
6. *Connector: RS-232 side DB-9 female, RS-485 side 3 terminal block*
7. *Working temperature: -20°C to 60°C*
8. *Storage temperature: -40°C to 85°C*
9. *Power input: No power supply needed, powered from TXD, RTS, DTR signal in RS-232 port.*

4.2 Perancangan Perangkat Lunak

Terdapat 2 jenis perancangan perangkat lunak yaitu, perancangan program utama yang terdapat di dalam mikrokontroler dan di dalam Visual Studio sebagai penampilnya.

4.2.1 Perancangan Algoritma Program Mikrokontroler

Perancangan ini dilakukan agar sistem dapat bekerja sesuai dengan algoritma yang diberikan. Perancangan ini menggunakan bahasa C dan melalui *compiler* program CodeVisionAVR. Keluaran yang dihasilkan program ini bernilai sudut dari 3 masukan dari sensor-sensor yang terdapat di *Loading Arm* yaitu *slewing, inboard* dan *outboard*. Diagram alir program utama ditunjukkan dalam Gambar 4.6. Data-data sudut tersebut selanjutnya dapat dilihat di komputer dan dikirim melalui komunikasi serial RS-232.



Gambar 4.6 Diagram Alir Program Utama

4.2.2 Perancangan Algoritma Program PMS Visual Studio

Dalam perancangan program PMS perlu diperhatikan batas-batas sudut yang dihasilkan oleh lengan *slewing*, *inboard* dan *outboard*. Seperti yang ditunjukkan dalam Lampiran. Berdasarkan sudut minimum dan maksimum lengan-lengan *Loading Arm* ditunjukkan dalam Tabel 4.2.

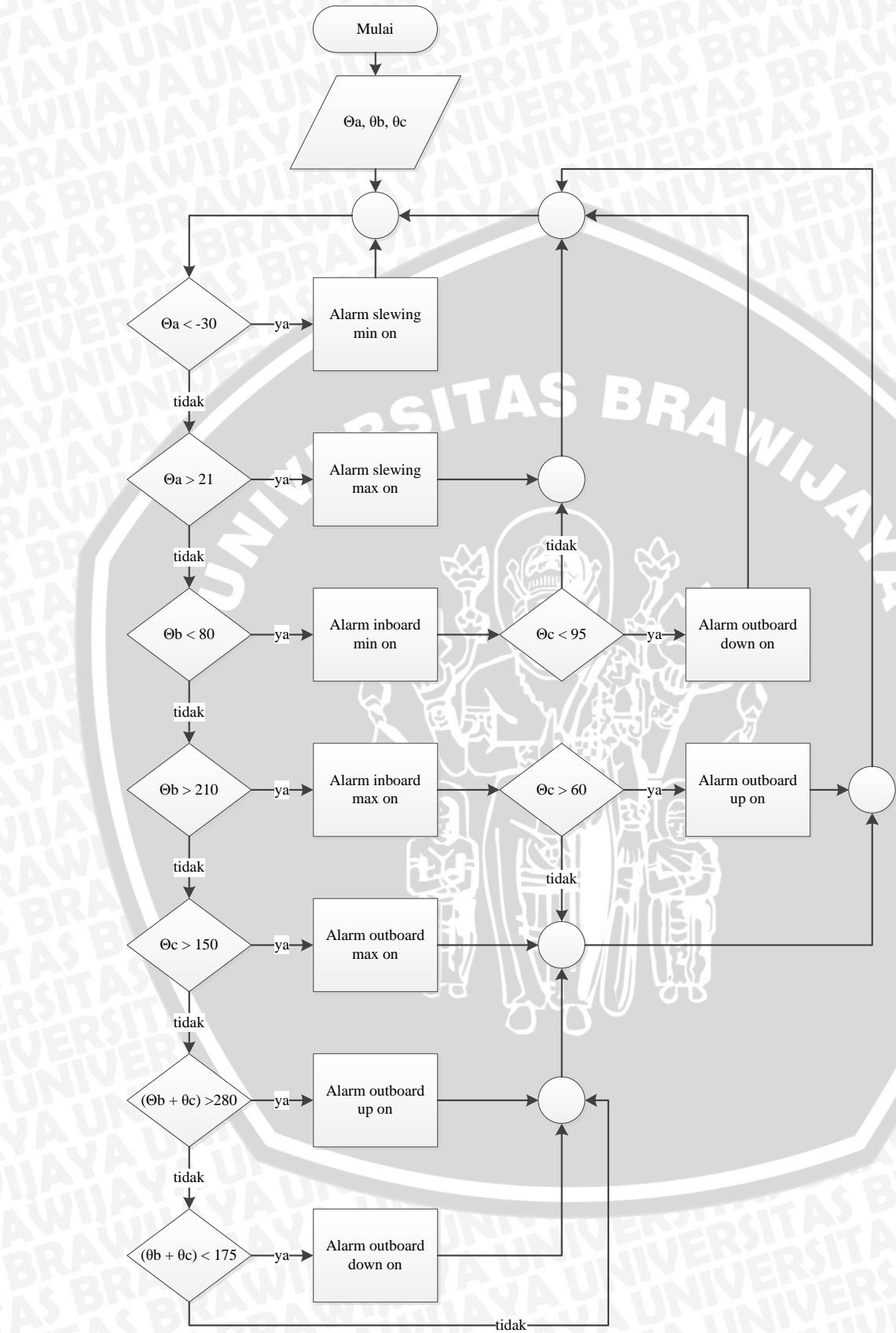
Tabel 4.2 Batas-batas Sudut Lengan *Loading Arm*

Lengan	Batas Sudut	
	Minimum	Maksimum
Slewing	-30°	21°
Inboard	80°	210°
Outboard	*	*

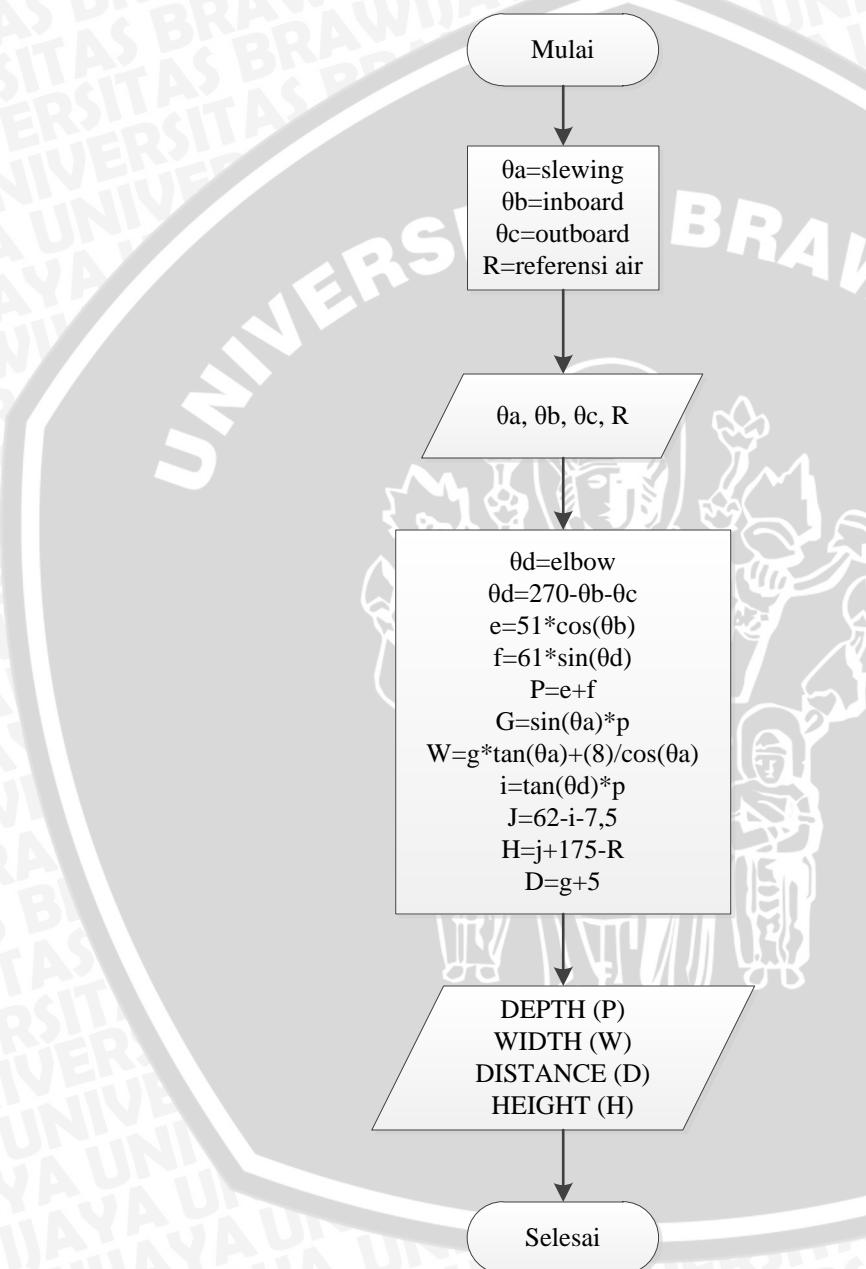
* nilai minimum *outboard* ditunjukkan dalam Lampiran pada bagian PMS ANGELS

Batas tersebut digunakan sebagai batas aman pergerakan *Loading Arm* dan sebagai *feedback* alarm dalam program visual. Jika sudut yang dihasilkan lengan keluar dari batas antara minimum dan maksimum , maka program akan mengeluarkan peringatan dan operator akan mengontrol pergerakan *Loading Arm* agar *Loading Arm* kembali dalam daerah aman. Diagram alir batas alarm *Loading Arm* ditunjukkan dalam Gambar 4.7.



Gambar 4.7 Diagram alir batas alarm *Loading Arm*

Sudut-sudut yang dihasilkan lengan kemudian mejadi masukan yang diketahui untuk mendapatkan parameter-parameter yang akan dicari. Terdapat beberapa parameter yang dicari, yaitu DEPTH, WITDH, DISTANCE, dan HEIGH. Parameter tersebut ditampilkan dalam komputer dengan menggunakan program Visual Studio. Algoritma parameter *Loading Arm* ditunjukkan dalam Gambar 4.8.



Gambar 4.8 Algoritma Program PMS

BAB V

PENGUJIAN DAN ANALISIS

5.1 Pengujian Perangkat Keras

5.1.1 Pengujian Modul Sensor Potensiometer

5.1.1.1 Tujuan

Pengujian modul sensor potensiometer bertujuan untuk melihat tegangan yang keluar dari sensor.

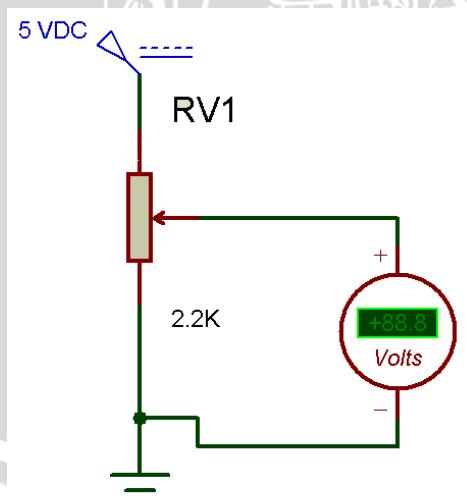
5.1.1.2 Alat yang Digunakan

Alat yang digunakan untuk pengujian ini adalah sebagai berikut:

1. Sumber tegangan 5VDC
2. Multimeter Digital
3. Modul sensor potensiometer

5.1.1.3 Prosedur Pengujian

Pengujian ini dilakukan dengan cara mengukur nilai resistansi pada potensiometer terlebih dahulu kemudian hubungkan rangkaian seperti ditunjukkan dalam Gambar 5.1 lalu ukur tegangan keluaran potensiometer di kaki ke 2.



Gambar 5.1 Rangkaian pengujian modul sensor potensiometer

5.1.1.4 Hasil Pengujian dan Analisis

Hasil pengujian sensor potensiometer ditunjukkan dalam Gambar 5.2 dan diambil ketika sudut *Loading Arm* pada bagian *Outboard* sebesar 90° terhadap bagian *Inboard*.



Gambar 5.2 Pengukuran Tegangan Modul Sensor Potensiometer dengan $\theta=90^\circ$

Tabel 5.1 Pengujian Tegangan Modul Sensor Potensiometer

No.	θ	Tegangan terukur (V)	Tegangan teori (V)	Selisih	
				V	%
1	0°	0	0	0	0
2	45°	0,596	0,625	0,029	4,64
3	90°	1,210	1,250	0,040	3,20
4	135°	1,837	1,875	0,038	2,02
5	180°	2,489	2,500	0,011	0,44
6	225°	3,084	3,125	0,041	1,31
7	270°	3,713	3,750	0,037	0,98
8	315°	4,345	4,375	0,030	0,68
Rata-rata				1,65	

Tegangan teori didapat dari hubungan antara sudut $0^\circ \sim 360^\circ$ dengan tegangan $0 \sim 5$ volt. Hubungan persamaan tegangan dengan sudut ditunjukkan dalam Persamaan (5.1).

$$\frac{V(\text{teori}) - V(1)}{V(2) - V(1)} = \frac{\theta - \theta(1)}{\theta(2) - \theta(1)}$$

$$\frac{V(\text{teori}) - 0}{5 - 0} = \frac{\theta - 0}{360 - 0}$$

$$V(\text{teori}) = 0,014 * \theta \quad (5.1)$$

Berdasarkan hasil pengujian dalam Gambar 5.2 dan Tabel 5.1 menunjukkan ketika modul potensiometer diberi catu 5 V dan sensor diputar, maka rata-rata kesalahan yang diperoleh sebesar 1,65%. Berdasarkan datasheet modul sensor potensiometer, nilai kesalahan sensor masih di bawah 10% dan modul sensor dapat dikatakan dalam kondisi baik.

5.1.2 Pengujian Catu Daya

5.1.2.1 Tujuan

Pengujian Catu daya bertujuan untuk melihat keluaran rangkaian catu daya 5V apakah sesuai dengan perancangan saat berbeban atau tanpa beban.

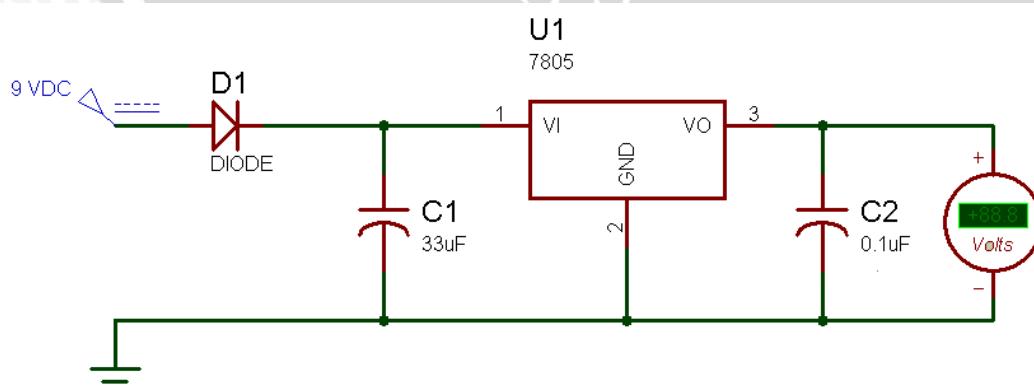
5.1.2.2 Alat yang Digunakan

Alat yang digunakan untuk pengujian ini adalah sebagai berikut:

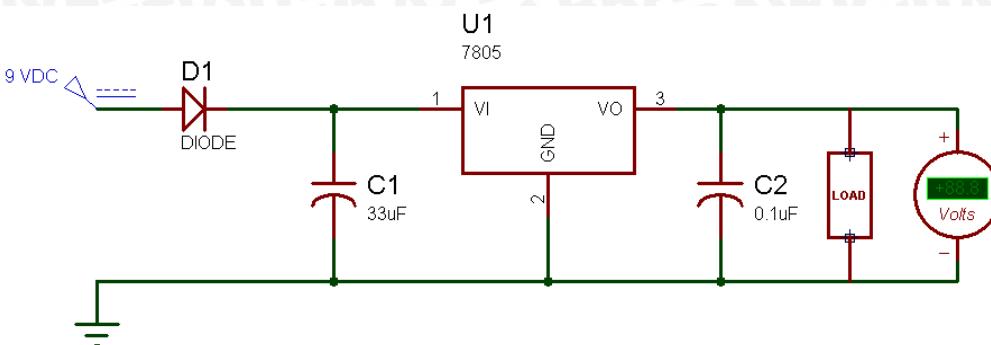
1. Sumber tegangan adaptor 9VDC 500mA
2. Multimeter Digital
3. Rangkaian Catu Daya yang diuji
4. Rangkaian Beban (sensor, mikrokontroler, dan rangkaian serial)

5.1.2.3 Prosedur Pengujian

Pengujian ini dilakukan dengan menghubungkan rangkaian tanpa beban seperti ditunjukkan dalam Gambar 5.3 dan rangkaian berbeban seperti ditunjukkan dalam Gambar 5.4 kemudian mengukur tegangan keluarann catu daya 5V saat berbeban dan saat tanpa beban.



Gambar 5.3 Rangkaian Pengujian Catu Daya Tanpa Beban



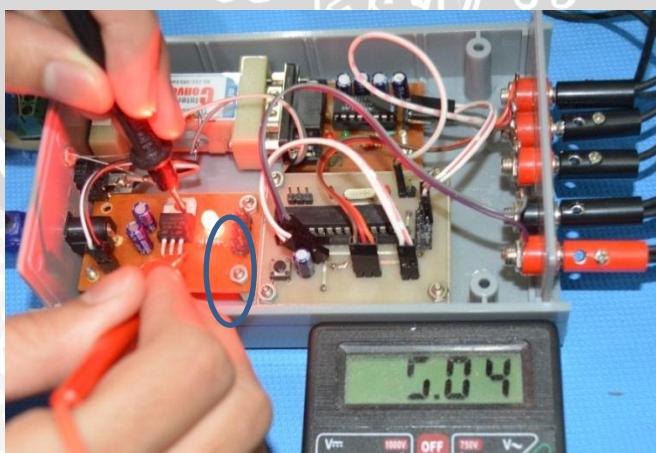
Gambar 5.4 Rangkaian Pengujian Catu Daya Berbeban

5.1.2.4 Hasil Pengujian dan Analisis

Hasil pengujian catu daya untuk yang berbeban diambil ketika semua rangkaian seperti rangkaian MAX-232, rangkaian Mikrokontroler, dan sensor potensiometer dicatu oleh catu daya, sedangkan untuk yang tidak berbeban rangkaian beban tidak dicatu oleh catu daya.



Gambar 5.5 Pengukuran Catu Daya Berbeban



Gambar 5.6 Pengukuran Catu Daya Tanpa Beban

Berdasarkan hasil pengujian dalam Gambar 5.5 dan Gambar 5.6 rangkaian catu daya memiliki tegangan keluaran sebesar 5,03 V untuk yang berbeban dan 5,04 V untuk yang tidak berbeban. Hal ini menunjukkan bahwa rangkaian catu daya berfungsi dengan baik.

5.1.3 Pengujian Mikrokontroler

5.1.3.1 Tujuan

Pengujian mikrokontroler bertujuan untuk melihat keluaran pada mikrokontroler ke rangkaian serial apakah sesuai dengan datasheet.

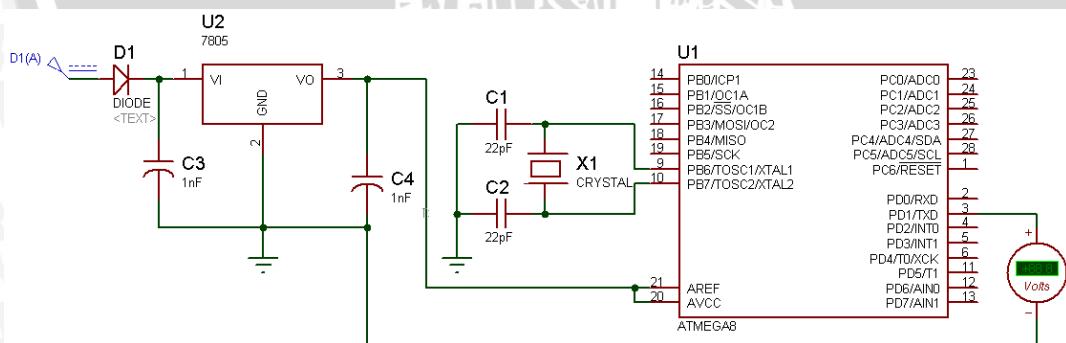
5.1.3.2 Alat yang Digunakan

Alat yang digunakan untuk pengujian ini adalah sebagai berikut:

1. Modul mikrokontroler
2. Multimeter digital
3. Catu daya 5V DC

5.1.3.3 Prosedur Pengujian

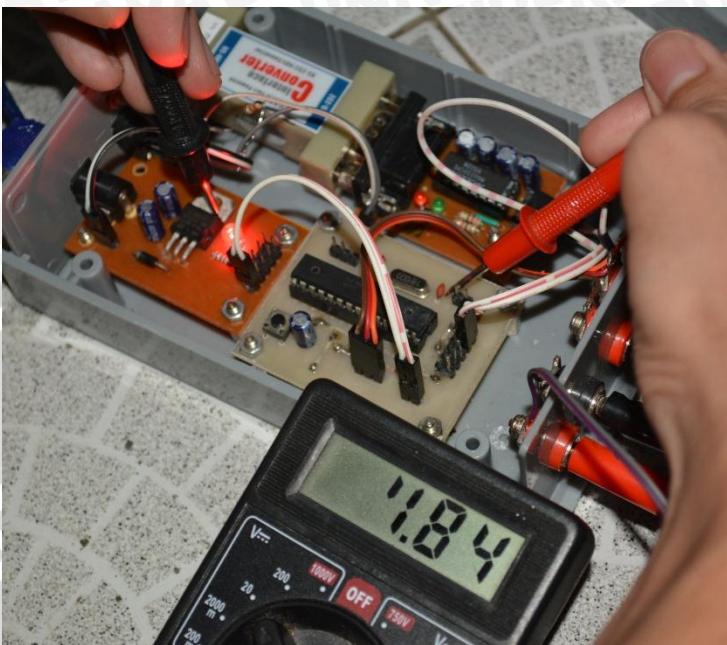
Pengujian ini dilakukan dengan cara mengukur keluaran yang terdapat pada PIN Tx/Rx mikrokontroler seperti yang ditunjukkan dalam Gambar 5.7.



Gambar 5.7 Rangkaian Pengujian Tegangan Tx/Rx Mikrokontroler

5.1.3.4 Hasil Pengujian dan Analisis

Hasil pengujian mikrokontroler ditunjukkan dalam Gambar 5.8 dan diambil ketika port Tx pada mikrokontroler berada dalam kondisi logika 1.



Gambar 5.8 Pengujian Pengukuran Tegangan Tx/Rx Mikrokontroler

Dalam Gambar 5.8 menunjukkan bahwa tegangan yang terukur pada PIN Tx mikrokontroler sebesar 4,84 V. Berdasarkan datasheet MAX232, minimum *input threshold high* yang dapat diterima sebesar 2 V, jadi mikrokontroler dapat bekerja dengan baik.

5.1.4 Pengujian Rangkaian MAX-232

5.1.4.1 Tujuan

Pengujian Rangkaian MAX-232 bertujuan untuk melihat keluaran rangkaian MAX-232 apakah sudah sesuai dengan datasheet.

5.1.4.2 Alat yang Digunakan

Alat yang digunakan untuk pengujian ini adalah sebagai berikut:

1. Modul rangkaian RS-232
2. Modul mikrokontroler
3. Multimeter digital
4. Catu daya 5 V DC

5.1.4.3 Prosedur Pengujian

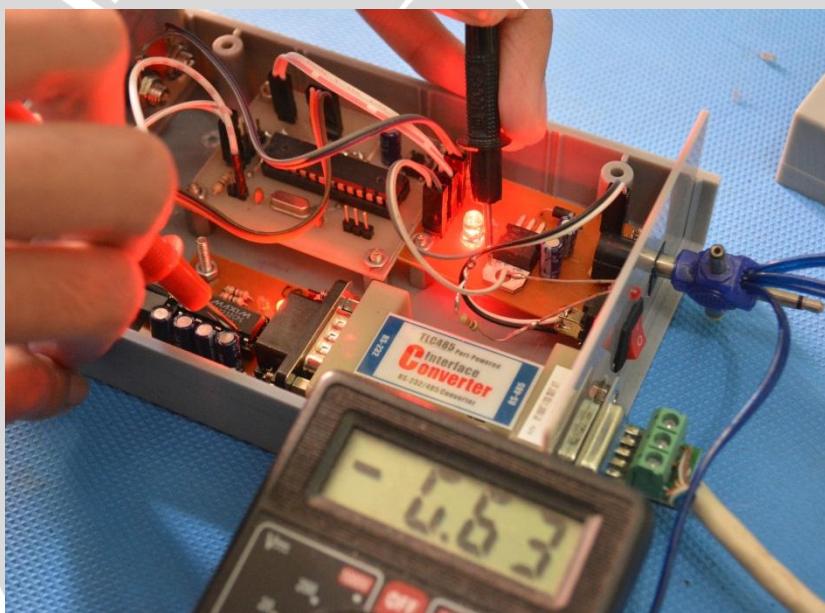
Pengujian dilakukan dengan cara menghubungkan pin Tx mikrokontroler dengan rangkaian MAX-232, kemudian ukur tegangan keluarannya. Diagram blok pengujian rangkaian MAX-232 ditunjukkan dalam Gambar 5.9.



Gambar 5.9 Diagram Blok Pengujian Rangkaian MAX-232

5.1.4.4 Hasil Pengujian dan Analisis

Hasil pengujian MAX-232 ditunjukkan dalam Gambar 5.10 dan diambil pada saat port *Transfer Out* pada IC MAX-232 dalam kondisi logika 1.



Gambar 5.10 Pengujian Rangkaian MAX-232

Dalam Gambar 5.10 menunjukkan bahwa keluaran rangkaian MAX-232 sebesar $-6,63$ V, dan berdasarkan *datasheet* MAX-232 tegangan antara -3 V sampai dengan -15 V merupakan tegangan standar RS-232 untuk mendapatkan logika 1.

Jadi rangkaian MAX-232 ini dapat bekerja dengan baik.

5.1.5 Pengujian Komunikasi Serial

5.1.5.1 Tujuan

Tujuan Pengujian ini adalah untuk menganalisis apakah program mikrokontroler dapat diterima dengan baik oleh komputer.

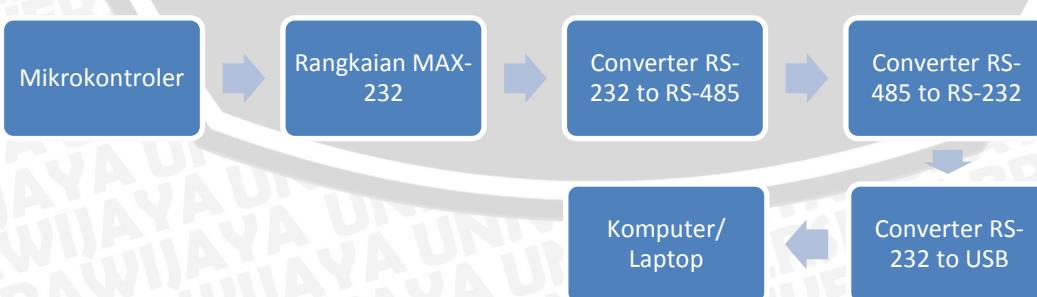
5.1.5.2 Alat yang Digunakan

Alat yang digunakan untuk pengujian ini adalah sebagai berikut:

1. Rangkaian mikrokontroler
2. Rangkaian MAX-232
3. Dua buah converter RS-232 to RS-485
4. Kabel penghubung ± 10 meter
5. Modul RS-232 to USB
6. Komputer/ Laptop
7. Hyperterminal

5.1.5.3 Prosedur Percobaan

Pengujian dilakukan dengan cara menghubungkan seluruh rangkaian dengan 2 buah converter RS-232 to RS-485 yang diantaranya dihubungkan oleh kabel penghubung. Diagram blok ditunjukkan dalam Gambar 5.11. Hyperterminal digunakan untuk melihat data yang dikirim oleh mikrokontroler. Kecepatan koneksi sebesar 9600 bps dengan protokol komunikasi 8 bit data, 1 stop bit, dan tanpa parity.



Gambar 5.11 Diagram Blok Pengujian Komunikasi Serial

5.1.5.4 Hasil Pengujian dan Analasis

Hasil pengujian komunikasi serial ditunjukkan dalam Gambar 5.12 dan diambil pada saat hyperterminal pada computer/laptop dalam kondisi *connect*.

```

Terminal
Disconnect Hex Code: Send Rx File Tx File Hex Clear
Pengujian Komunikasi Serial Monitoring Loading Arm
Serial Monitoring Loading Arm Pengujian Komunikasi
m Pengujian Komunikasi Serial Monitoring Loading Arm Pengujian Komunikasi Serial
Monitoring Loading Arm Pengujian Komunikasi Serial Monitoring Loading Arm Pengujian
jian Komunikasi Serial Monitoring Loading Arm Pengujian Komunikasi Serial Monito
ring Loading Arm Pengujian Komunikasi Serial Monitoring Loading Arm Pengujian Ko
municasi Serial Monitoring Loading Arm Pengujian Komunikasi Serial Monitoring Lo
ading Arm Pengujian Komunikasi Serial Monitoring Loading Arm Pengujian Komu
i Serial Monitoring Loading Arm Pengujian Komunikasi Serial Monitoring Loading A
rm Pengujian Komunikasi Serial Monitoring Loading Arm Pengujian Komu
COM2:9600,8N1 No handsh. ASCII TTY Echo on

```

Gambar 5.12 Pengujian Komunikasi Serial

Pada tampilan hyperterminal dalam Gambar 5.12, karakter yang terbaca sama dengan karakter yang dibangkitkan oleh mikrokontroler, yaitu "Pengujian Komunikasi Serial Monitoring *Loading Arm*". Data hasil pembacaan pada hyperterminal juga tidak terdapat cacat karakter. Hal ini membuktikan *hardware* telah mampu mendukung komunikasi serial dengan kecepatan 9600 bps dengan 8 bit data, 1 bit *stop*, dan tanpa *parity* dan dapat bekerja dengan baik.

5.2 Pengujian Perangkat Lunak

5.2.1 Tujuan

Pengujian ini bertujuan untuk mencocokkan nilai input dari sistem dengan gerakan program visual.

5.2.2 Alat yang Digunakan

Alat yang digunakan untuk pengujian ini adalah sebagai berikut:

1. Komputer/ Laptop
2. Program Visual Studio

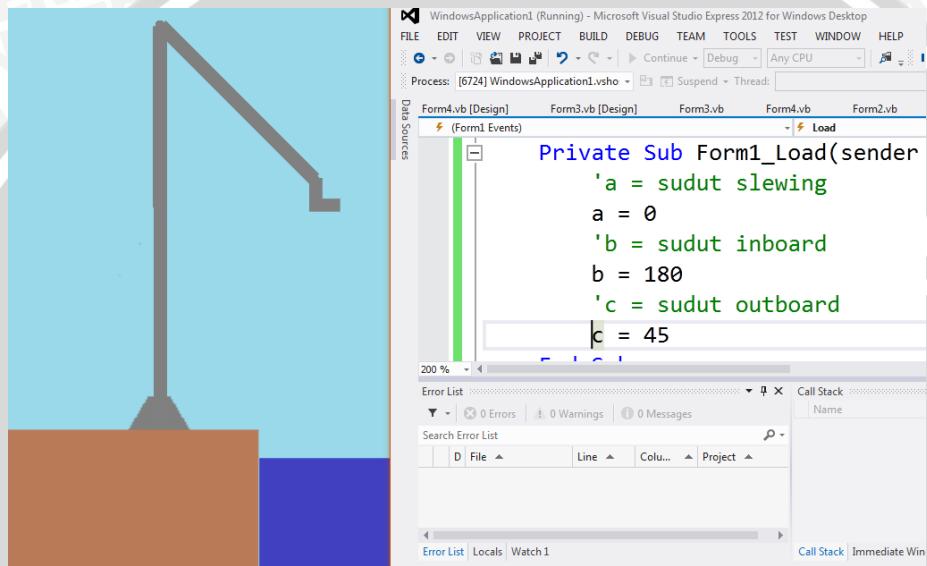


5.2.3 Prosedur Pengujian

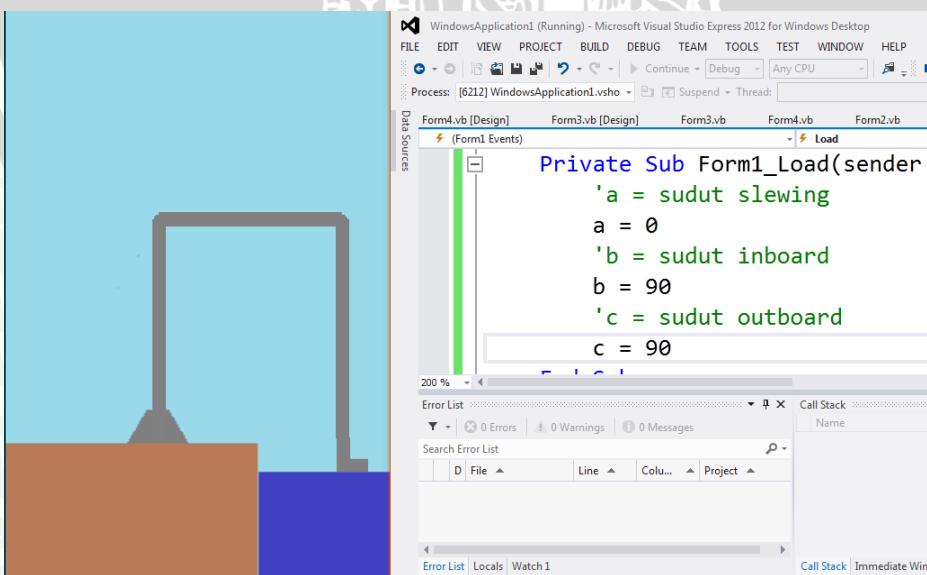
Pengujian dilakukan dengan cara memberikan nilai sudut secara manual dengan menggunakan keyboard. Lalu cek gambar visual dan hasil keluaran nilai visual apakah sama.

5.2.4 Hasil Pengujian dan Analisis

Hasil pengujian perangkat lunak ditunjukkan dalam Gambar 5.13 dan Gambar 5.14.



Gambar 5.13 Pengujian Gerak Program *Loading Arm* Dengan $a=0$, $b=180$ dan $c=45$



Gambar 5.14 Pengujian Gerak Program *Loading Arm* Dengan $a=0$, $b=90$ dan $c=90$

Berdasarkan hasil pengujian gerak program *Loading Arm* dengan *software* Visual Studio dalam Gambar 5.13 dan Gambar 5.14 menunjukkan bahwa program dapat berjalan sesuai dengan masukan yang didapat dengan baik.

5.3 Pengujian Keseluruhan Sistem

5.3.1 Tujuan

Pengujian keseluruhan bertujuan untuk menganalisis kemampuan sistem *Loading Arm* dalam mengukur sudut orientasi.

5.3.2 Alat yang Digunakan

Alat yang digunakan untuk pengujian ini adalah sebagai berikut:

1. Adaptor 9 VDC
2. Miniatur *Loading Arm*
3. Rangkaian Catu Daya 5 VDC
4. Rangkaian Mikrokontroler ATMega8
5. Rangkaian Komunikasi
6. Dua buah *Converter RS-232 to RS-485*
7. Modul RS-232 to USB
8. Komputer/ Laptop
9. Busur Derajat dan Penggaris

5.3.3 Prosedur Pengujian

Pengujian ini dilakukan dengan cara menghubungkan rangkaian sesuai diagram blok dalam Gambar 5.15. Busur derajat digunakan sebagai acuan untuk membandingkan besarnya sudut orientasi *Loading Arm*. Pada kondisi awal, sudut *Loading Arm* diposisikan ke sudut 0° untuk slewing, 180° untuk inboard dan 45° untuk outboard. Kemudian sambungkan koneksi antara rangkaian sistem *Loading Arm* dan komputer/ Laptop dengan menggunakan modul RS-232 to USB. Ubah sudut *Loading Arm* hingga sudut tertentu. Lalu catat dan amati data yang ditampilkan program Visual Studio untuk di Analisis. Pengujian keseluruhan ditunjukkan dalam Gambar 5.16.



Gambar 5.15 Diagram Blok Pengujian Keseluruhan Sistem



Gambar 5.16 Pengujian Keseluruhan Sistem Monitoring *Loading Arm*

5.3.4 Hasil Pengujian dan Analisis

Pada saat miniatur *Loading Arm* berada di posisi awal yaitu dengan sudut slewing sebesar 0° , inboard sebesar 180° , dan outboard sebesar 45° , program alarm tidak ada yang menyala.

Ketika posisi sudut miniatur *Loading Arm* diubah ke sudut-sudut kritis, maka *feedback* program alarm akan menyala sesuai dengan *range* yang telah ditentukan. Parameter-parameter *Loading Arm* dan program alarm ditunjukkan dalam Tabel 5.2.

Tabel 5.2 Pengujian Keseluruhan Sistem Monitoring *Loading Arm*

No.	sudut arm (°)			sudut termonitor (°)			parameter teori (cm)				parameter termonitor (cm)				alarm						
	slewing	inboard	outboard	slewing	Inboard	outboard	P	W	D	H	P	W	D	H	s. max	s. min	i. max	i. min	o. up	o. down	o. max
1	0	180	45	1	179	46	43.1	8.0	48.1	179.3	44.0	8.8	49.0	179.6							
2	-40	215	30	-40	215	32	26.0	-6.3	24.9	202.4	26.9	-6.8	25.6	204.9	V	V					
3	-30	180	90	-30	179	89	61.0	-21.3	57.8	236.0	61.9	-21.7	58.6	233.2							
4	-20	135	135	-19	136	136	97.1	-24.7	96.2	236.0	96.4	-22.9	96.1	238.7							
5	-10	120	150	-8	122	151	105.2	-10.1	108.6	236.0	104.2	-6.4	108.2	240.1							V
6	0	115	155	-1	115	154	107.2	8.0	112.2	236.0	107.2	6.1	112.2	234.6							V
7	10	100	45	10	100	46	15.2	10.8	20.0	169.1	16.1	10.9	20.9	168.7							V
8	20	90	180	21	89	179	112.0	46.8	110.2	236.0	112.0	48.7	109.5	233.2							V
9	30	80	90	32	79	89	39.6	29.1	39.3	156.7	37.4	29.2	36.7	156.8	V		V				
10	40	200	95	39	201	96	37.8	34.8	34.0	269.6	36.1	33.0	33.0	271.9	V		V				
No.	parameter teori (cm)				parameter termonitor (cm)				error P		error W		error D		error H						
	P	W	D	H	P	W	D	H	(cm)	%	(cm)	%	(cm)	%	(cm)	%					
1	43.1	8.0	48.1	179.3	44.0	8.8	49.0	179.6	0.9	0.79	0.8	0.14	0.9	0.77	0.3	0.10					
2	26.0	-6.3	24.9	202.4	26.9	-6.8	25.6	204.9	0.9	0.77	0.5	0.09	0.7	0.63	2.5	0.80					
3	61.0	-21.3	57.8	236.0	61.9	-21.7	58.6	233.2	0.9	0.76	0.4	0.07	0.8	0.71	2.8	0.89					
4	97.1	-24.7	96.2	236.0	96.4	-22.9	96.1	238.7	0.7	0.60	1.8	0.32	0.1	0.09	2.7	0.86					
5	105.2	-10.1	108.6	236.0	104.2	-6.4	108.2	240.1	1.0	0.89	3.7	0.65	0.4	0.36	4.1	1.30					
6	107.2	8.0	112.2	236.0	107.2	6.1	112.2	234.6	0.0	0.01	1.9	0.33	0.0	0.01	1.4	0.45					
7	15.2	10.8	20.0	169.1	16.1	10.9	20.9	168.7	0.9	0.78	0.1	0.02	0.9	0.80	0.4	0.13					
8	112.0	46.8	110.2	236.0	112.0	48.7	109.5	233.2	0.0	0.04	1.9	0.33	0.7	0.63	2.8	0.89					
9	39.6	29.1	39.3	156.7	37.4	29.2	36.7	156.8	2.3	2.01	0.1	0.02	2.6	2.32	0.1	0.03					
10	37.8	34.8	34.0	269.6	36.1	33.0	33.0	271.9	1.8	1.58	1.8	0.32	1.0	0.89	2.3	0.72					
									rata2		0.82		0.23		0.72		0.62				

Terdapat *error* antara parameter-parameter yang ditunjukkan program dengan parameter-parameter yang dihitung dengan teori pada bagian miniatur *Loading Arm*. Pada parameter DEPTH terdapat *error* rata-rata sebesar 0,82%, parameter WIDTH 0,23%, parameter DISTANCE 0,72%, dan parameter HEIGHT 0,62%. Dalam Tabel 5.2 ditunjukkan bahwa pada saat miniatur *Loading Arm* berada pada sudut kritis, maka *feedback alarm* program *Loading Arm* di komputer menyala. Hal ini menunjukkan bahwa alat monitoring *Loading Arm* dapat bekerja dengan baik.



BAB VI

KESIMPULAN DAN SARAN

6.1 Kesimpulan

Berdasarkan hasil pengujian dan rumusan masalah yang telah dibuat, didapatkan kesimpulan sebagai berikut :

- 1) Sistem monitoring *Loading Arm* dapat berfungsi dengan baik untuk memonitoring *Loading Arm* dengan *error* rata-rata sensor sebesar 1,65%, karena *error* ini masih dibawah *error* maksimum yang bernilai $\pm 10\%$ dan kurang presisinya miniatur *Loading Arm* yang terpasang sensor sehingga terdapat perbedaan ukuran antara sudut dan parameter dalam program dengan pengukuran dan teori sebenarnya.
- 2) Program alarm sistem monitoring *Loading Arm* dapat bekerja dengan baik sesuai pembacaan sensor saat miniatur *Loading Arm* berada dalam sudut kritis.
- 3) Data modul sistem monitoring dengan komputer/*console* terkirim sesuai yang diharapkan dengan jarak 10 meter, karena RS-485 mampu mengirimkan data maksimum sejauh 1220 meter.
- 4) Program Visual Studio mampu bekerja dengan baik sesuai dengan masukan dari sensor potensiometer yang berupa tegangan yang kemudian dikonversi menjadi sudut oleh mikrokontroler.

6.2 Saran

Agar sistem monitoring *Loading Arm* ini dapat bekerja lebih baik, maka perlu diperhatikan bentuk mekanika *Loading Arm* yang digunakan. Semakin presisi bentuk *Loading Arm*, maka semakin presisi data yang ditampilkan.



DAFTAR PUSTAKA

- ATMEL. "ATMega8". <http://www.atmel.com/images/doc2486.pdf> (diakses pada 28 Agustus 2012, 21.47 WIB)
- Bejo, Agus. 2008. "*C dan AVR Rahasia Kemudahan Bahasa C dalam mikrokontroler ATMega 8535*". Yogyakarta: Graha Ilmu.
- FMC Energy System. 2000. "*MAGNUM – Loading Arm*". Europa: FMC Europa SA.
- FMC Energy & Transportation Equipment Group. 1998. "*Bontang LNG Expansion Project 3rd Book*". Europe: FMC Group.
- FMC Energy & Transportation Equipment Group. 1998. "*LPG Storage Loading Arm Technical Data Book*". Europe: FMC Group.
- Hendrayudi. 2009. "*VB 2008 untuk Berbagai Keperluan Pemrograman*". Jakarta: Elex Media Komputindo.
- Kontrol, Elektro. "*AVR Tutorial*". <http://elektro-kontrol.blogspot.com> (diakses pada 28 Agustus 2012, 21.35 WIB)
- Kurniawan, Dayat. 2009. "*ATMega8 dan Aplikasinya*". Jakarta: Elex Media Komputindo.
- PT Badak NGL. "*Sejarah Kami*". http://www.badaklng.co.id/in/sejarah_ptbadak.html (diakses pada 23 Agustus 2012, 20.34 WIB)
- Rusmawan, Uus. 2007. "*Konsep dan Implementasi Visual Basic*". Jakarta: Elex Media Komputindo
- Winoto, Ardi. 2008. "*Mikrokontroler AVR ATMega8/32/16/8535 dan Pemrogramannya Dengan Bahasa C Pada WinAVR + CD*". Bandung: Informatika.



UNIVERSITAS BRAWIJAYA



LAMPIRAN

Dokumentasi Alat



Gambar 1 Tampak dari Atas



Gambar 2 Tampak dari Depan



Gambar 3 Box Elektrik Tertutup



Gambar 4 Miniatur *Loading Arm* Tampak dari Samping

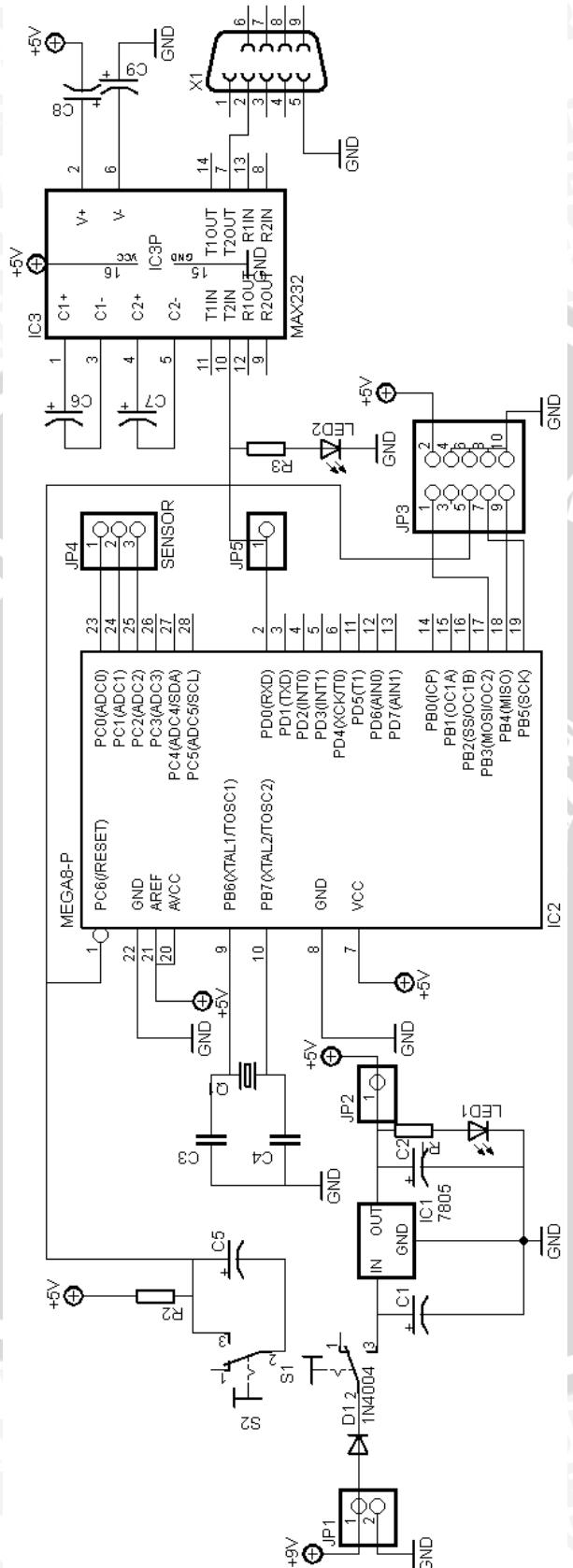


Gambar 5 Miniatur *Loading Arm* Tampak dari Depan



LAMPIRAN II

Skema Rangkaian



Gambar 6 Skema Rangkaian Sistem Monitoring Loading Arm



UNIVERSITAS BRAWIJAYA



LAMPIRAN III

Listing Program Utama

```
*****  
*****
```

This program was produced by the
CodeWizardAVR V2.05.0 Professional
Automatic Program Generator
© Copyright 1998-2010 Pavel Haiduc,
HP InfoTech s.r.l.
<http://www.hpinfotech.com>

Project :

Version :

Date : 1/13/2013

Author : NeVaDa

Company :

Comments:

Chip type : ATmega8
Program type : Application
AVR Core Clock frequency: 16.000000
MHz
Memory model : Small
External RAM size : 0
Data Stack size : 256


```
#include <mega8.h>  
  
#include <delay.h>  
  
// Standard Input/Output functions  
#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 & 0xff);  
// Delay needed for the stabilization of the  
ADC input voltage  
delay_us(10);  
// Start the AD conversion  
ADCSRA|=0x40;  
// Wait for the AD conversion to complete  
while ((ADCSRA & 0x10)==0);  
ADCSRA|=0x10;  
return ADCW;  
}
```

```
// Declare your global variables here  
int a,b,c,x,y,z;  
char  
d,e,f,buff[33],byte[9]={0,0,0,0,0,0,0,0,0};  
  
void main(void)  
{  
// Declare your local variables here  
  
// Input/Output Ports initialization  
// Port B initialization  
// Func7=Out Func6=Out Func5=Out  
Func4=Out Func3=Out Func2=Out  
Func1=Out Func0=Out  
// State7=0 State6=0 State5=0 State4=0  
State3=0 State2=0 State1=0 State0=0  
PORTB=0x00;  
DDRB=0xFF;  
  
// Port C initialization  
// Func6=Out Func5=Out Func4=Out  
Func3=Out Func2=Out Func1=Out  
Func0=Out  
// State6=0 State5=0 State4=0 State3=0  
State2=0 State1=0 State0=0  
PORTC=0x00;  
DDRC=0x00;  
  
// Port D initialization  
// Func7=Out Func6=Out Func5=Out  
Func4=Out Func3=Out Func2=Out  
Func1=Out Func0=Out  
// State7=0 State6=0 State5=0 State4=0  
State3=0 State2=0 State1=0 State0=0  
PORTD=0x00;  
DDRD=0xFF;  
  
// Timer/Counter 0 initialization  
// Clock source: System Clock  
// Clock value: Timer 0 Stopped  
TCCR0=0x00;  
TCNT0=0x00;  
  
// Timer/Counter 1 initialization  
// Clock source: System Clock  
// Clock value: Timer1 Stopped  
// Mode: Normal top=0xFFFF  
// OC1A output: Discon.  
// OC1B output: Discon.  
// Noise Canceler: Off  
// Input Capture on Falling Edge  
// Timer1 Overflow Interrupt: Off  
// Input Capture Interrupt: Off
```

```

// Compare A Match Interrupt: Off
// Compare B Match Interrupt: Off
TCCR1A=0x00;
TCCR1B=0x00;
TCNT1H=0x00;
TCNT1L=0x00;
ICR1H=0x00;
ICR1L=0x00;
OCR1AH=0x00;
OCR1AL=0x00;
OCR1BH=0x00;
OCR1BL=0x00;

// Timer/Counter 2 initialization
// Clock source: System Clock
// Clock value: Timer2 Stopped
// Mode: Normal top=0xFF
// OC2 output: Disconnected
ASSR=0x00;
TCCR2=0x00;
TCNT2=0x00;
OCR2=0x00;

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

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

// USART initialization
// Communication Parameters: 8 Data, 1
Stop, No Parity
// USART Receiver: On
// USART Transmitter: On
// USART Mode: Asynchronous
// USART Baud Rate: 9600
UCSRA=0x00;
UCSRB=0x18;
UCSRC=0x86;
UBRRH=0x00;
UBRRL=0x67;

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

// ADC initialization
// ADC Clock frequency: 1000.000 kHz
// ADC Voltage Reference: AVCC pin
ADMUX=ADC_VREF_TYPE & 0xff;
ADCSRA=0x84;

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

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

while (1)
{
    // Place your code here
    x=read_adc(0);
    y=read_adc(1);
    z=read_adc(2);
    a=(x/2.846667);
    b=(y/2.846667);
    c=(z/2.846667);

    byte[0] = (a/100)+48;
    byte[1] = ((a% 100)/10)+48;
    byte[2] = (a% 100)% 10+48;
    byte[3] = (b/100)+48;
    byte[4] = ((b% 100)/10)+48;
    byte[5] = (b% 100)% 10+48;
    byte[6] = (c/100)+48;
    byte[7] = ((c% 100)/10)+48;
    byte[8] = (c% 100)% 10+48;

    //lcd_clear();

    printf("1%c%c%c*1%c%c%c*1%c%c%c"
        , byte[0], byte[1], byte[2], byte[3],
        byte[4], byte[5], byte[6], byte[7],
        byte[8]);
    delay_ms(50);
    //lcd_puts(buff);
}
}

```



LAMPIRAN IV

Listing Program Visual Studio Form 1

```

Public Class Form1
    Dim data As String
    Public motor() As String
    Dim z As Double
    Dim i, j, r, s, t, u As Double
    Dim p, w, d, h As String
    Dim x, y As Double
    Public a, b, c As Integer

    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
        btnset.Click
            Form2.Show()
        End Sub

    Private Sub Button2_Click(sender As Object, e As EventArgs) Handles
        btnclose.Click
            Me.Close()
        End Sub

    Private Sub SerialPort1_DataReceived(sender As Object, e As
        System.IO.Ports.SerialDataReceivedEventArgs) Handles SerialPort1.DataReceived
        Me.Invoke(New myDelegate(AddressOf updateTextBox))
    End Sub
    Public Delegate Sub myDelegate()
    Public Sub updateTextBox()
        Dim panjang As Integer

        data = SerialPort1.ReadExisting
        panjang = data.Length
        If panjang = 14 Then
            tbslew.Clear()
            tbinb.Clear()
            tbout.Clear()
            motor = Split(data, "*")
            a = Convert.ToInt16(motor(0))
            b = Convert.ToInt16(motor(1))
            c = Convert.ToInt16(motor(2))
            a = a - 1180
            b = b - 1000
            c = c - 1000

            tbslew.AppendText(a)
            tbinb.AppendText(b)
            tbout.AppendText(c)

            TrackBar2.Value = a
            TrackBar1.Value = b
            TrackBar3.Value = c

            If a < -30 Then
                OvalShape3.FillColor = Color.Maroon
            ElseIf a > 21 Then
                OvalShape2.FillColor = Color.Maroon
            Else : OvalShape3.FillColor = Color.Transparent
                OvalShape2.FillColor = Color.Transparent
            End If

            If b < 80 Then
                OvalShape1.FillColor = Color.Maroon
            End If
        End If
    End Sub
End Class

```

```

        If c < 95 Then
            OvalShape6.FillColor = Color.Maroon
        Else : OvalShape6.FillColor = Color.Transparent
        End If
    ElseIf b > 210 Then
        OvalShape4.FillColor = Color.Maroon
        If c > 60 Then
            OvalShape5.FillColor = Color.Maroon
        Else : OvalShape5.FillColor = Color.Transparent
        End If
    Else : OvalShape4.FillColor = Color.Transparent
        OvalShape1.FillColor = Color.Transparent
        OvalShape5.FillColor = Color.Transparent
        OvalShape6.FillColor = Color.Transparent
    End If

    If c > 150 Then
        OvalShape7.FillColor = Color.Maroon
    Else : OvalShape7.FillColor = Color.Transparent
    End If

    If (b + c) > 280 Then
        OvalShape5.FillColor = Color.Maroon
    Else : OvalShape5.FillColor = Color.Transparent
    End If

    If (b + c) < 175 Then
        OvalShape6.FillColor = Color.Maroon
    Else : OvalShape6.FillColor = Color.Transparent
    End If

End If

End Sub

Private Sub samping_Click_2(sender As Object, e As EventArgs) Handles samping.Click
    Form3.Show()
    Form3.Location = New Point(50, 50)
    samping.Enabled = False
    Form4.Hide()
    atas.Enabled = True
End Sub

Private Sub atas_Click_1(sender As Object, e As EventArgs) Handles atas.Click
    Form4.Show()
    Form4.Location = New Point(50, 50)
    atas.Enabled = False
    Form3.Hide()
    samping.Enabled = True
End Sub

End Class

```





UNIVERSITAS BRAWIJAYA

LAMPIRAN V

Listing Program Visual Studio Form 2



Public Class Form2

```

    Private Sub Form2_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        btnok.Enabled = True
        btncancel.Enabled = True
        cbcom.Enabled = True
        cbbaud.Enabled = True
        cbcom.Items.Clear()
        For i As Integer = 0 To My.Computer.Ports.SerialPortNames.Count - 1
            cbcom.Items.Add(My.Computer.Ports.SerialPortNames(i))
        Next

        cbbaud.Items.Add(1200)
        cbbaud.Items.Add(2400)
        cbbaud.Items.Add(4800)
        cbbaud.Items.Add(9600)
        cbbaud.Items.Add(14400)
        cbbaud.Items.Add(19200)
        cbbaud.Items.Add(38400)
        cbbaud.Items.Add(57600)
        cbbaud.Items.Add(115200)
    End Sub

    Private Sub btnok_Click(sender As Object, e As EventArgs) Handles btnok.Click
        If cbcom.Text = "" Then
            MsgBox("Silahkan Pilih COM Port", MsgBoxStyle.OkOnly, " ")
        ElseIf cbbaud.Text = "" Then
            MsgBox("Silahkan Pilih Baudrate", MsgBoxStyle.OkOnly, " ")
        Else
            btnok.Enabled = True
            btncancel.Enabled = True
            cbcom.Enabled = False
            cbbaud.Enabled = False
            If Form1.SerialPort1.IsOpen Then
                Form1.SerialPort1.Close()
            End If
            Form1.SerialPort1.PortName = cbcom.SelectedItem
            Form1.SerialPort1.BaudRate = cbbaud.SelectedItem
            Form1.SerialPort1.Parity = IO.Ports.Parity.None
            Form1.SerialPort1.DataBits = 8
            Form1.SerialPort1.Open()
            Me.Hide()
        End If
    End Sub

    Private Sub btncancel_Click(sender As Object, e As EventArgs) Handles btncancel.Click
        btnok.Enabled = True
        btncancel.Enabled = True
        cbcom.Enabled = True
        cbbaud.Enabled = True
        Form1.SerialPort1.Close()
        Me.Hide()
    End Sub
End Class

```





LAMPIRAN VI

Listing Program Visual Studio Form 3

```

Public Class Form3
    Dim cbimage As Bitmap
    Dim slewing As Bitmap
    Dim inboard As Bitmap
    Dim outbound As Bitmap
    Dim elbow As Bitmap
    Dim engsel As Bitmap
    Dim laut As Bitmap
    Public inb, out As Double
    Public a, b, c, d, g, f, h, i, j, k, l, m, n, o, p, q As Double
    Public r As Integer
    Dim x As Double
    Dim y As Double
    Dim z As Double
    Dim s() As String
    Dim t() As String
    Private Sub Form3_Load(sender As Object, e As EventArgs) Handles Me.Load
        cbimage = New Bitmap("D:\-kuliah-
\edwin\VB\WindowsApplication1\WindowsApplication1\Resources\cbimage.bmp",
True)
        cbimage.MakeTransparent(Color.Blue)
        inboard = New Bitmap("D:\-kuliah-
\edwin\VB\WindowsApplication1\WindowsApplication1\Resources\inboard.bmp",
True)
        inboard.MakeTransparent(Color.White)
        elbow = New Bitmap("D:\-kuliah-
\edwin\VB\WindowsApplication1\WindowsApplication1\Resources\elbow.bmp", True)
        elbow.MakeTransparent(Color.White)
        engsel = New Bitmap("D:\-kuliah-
\edwin\VB\WindowsApplication1\WindowsApplication1\Resources\engsel.bmp", True)
        engsel.MakeTransparent(Color.White)
        laut = New Bitmap("D:\-kuliah-
\edwin\VB\WindowsApplication1\WindowsApplication1\Resources\laut.bmp", True)
        laut.MakeTransparent(Color.White)

    End Sub
    Private Sub Form3_Paint(sender As Object, e As PaintEventArgs) Handles
Me.Paint
        inb = Form1.b
        out = Form1.c

        'inb = 90
        'out = 180
        r = 145

        c = (inb - 90) * Math.PI / 180
        f = (90 - z) * Math.PI / 180

        b = 204 * Math.Sin(c)
        a = 204 * Math.Cos(c)
        z = 90 - (out - (90 - (inb - 90)))
        d = 244 * Math.Sin(f)
        g = 244 * Math.Cos(f)

        'DEPTH
        p = a + d
        Form1.depths.Clear()
        Form1.depths.AppendText(CInt(p / 4))
    End Sub

```



```

'WIDTH
h = Math.Sin(((90 - Form4.slew) * Math.PI) / 180) * p
i = h * Math.Tan(((Form4.slew) * Math.PI) / 180) + (32) /
Math.Cos(((Form4.slew) * Math.PI) / 180)
Form1.widths.Clear()
Form1.widths.AppendText(CInt(i / 4))

'DISTANCE
j = h + 20
Form1.distances.Clear()
Form1.distances.AppendText(CInt(j / 4))

'HEIGHT
k = Math.Sqrt((204 * 204) + (244 * 244) - (2 * 204 * 244 *
(Math.Cos(out * Math.PI) / 180)))
l = Math.Acos((244 * 244) + (k * k) - (204 * 204)) / (2 * 244 * k)
m = 270 - (inb + out + 1)
n = Math.Sin((m * Math.PI) / 180) * k
o = 248 - n - 30
q = o + 175 - r
Form1.heights.Clear()
Form1.heights.AppendText(CInt(q / 4) + 173)

With cbimage
    e.Graphics.TranslateTransform(390, 330)
    e.Graphics.RotateTransform(0)
    e.Graphics.DrawImage(cbimage, -390, -330, 780, 660)
    e.Graphics.ResetTransform()
End With
With slewing
    e.Graphics.TranslateTransform(170, 236)
    e.Graphics.RotateTransform(90)
    e.Graphics.DrawImage(inboard, -0, -8, 248, 16)
    e.Graphics.ResetTransform()
End With
With inboard
    x = 360 - inb
    e.Graphics.TranslateTransform(170, 236)
    e.Graphics.RotateTransform(x + 90)
    e.Graphics.DrawImage(inboard, -0, -8, 204, 16)
    e.Graphics.ResetTransform()
End With
With outboard
    e.Graphics.TranslateTransform(170 + a, 236 - b)
    e.Graphics.RotateTransform(z)
    e.Graphics.DrawImage(inboard, -0, -8, 244, 16)
    e.Graphics.ResetTransform()
End With
With elbow
    e.Graphics.TranslateTransform(164 + a + d, 236 - b + g)
    e.Graphics.RotateTransform(0)
    e.Graphics.DrawImage(elbow, -0, -2, 35, 38)
    e.Graphics.ResetTransform()
End With
With engsel
    e.Graphics.TranslateTransform(170, 236)
    e.Graphics.RotateTransform(0)
    e.Graphics.DrawImage(engsel, -8, -8, 16, 16)
    e.Graphics.ResetTransform()
End With

```

```

With engsel
    e.Graphics.TranslateTransform(170 + a, 236 - b)
    e.Graphics.RotateTransform(0)
    e.Graphics.DrawImage(engsel, -8, -8, 16, 16)
    e.Graphics.ResetTransform()
End With
With engsel
    e.Graphics.TranslateTransform(170 + a + d, 236 - b + g)
    e.Graphics.RotateTransform(0)
    e.Graphics.DrawImage(engsel, -8, -8, 16, 16)
    e.Graphics.ResetTransform()
End With
With laut
    e.Graphics.TranslateTransform(780, 660)
    e.Graphics.RotateTransform(180)
    e.Graphics.DrawImage(laut, -0, -0, 500, r)
    e.Graphics.ResetTransform()
End With
End Sub

Private Sub Timer1_Tick(sender As Object, e As EventArgs) Handles
Timer1.Tick
    Me.Refresh()
End Sub
End Class

```





LAMPIRAN VII

Listing Program Visual Studio Form 4

```

Public Class Form4
    Dim atas As Bitmap
    Dim inboard As Bitmap
    Dim outbound As Bitmap
    Dim elbow As Bitmap
    Dim engsel As Bitmap
    Public slew As Double
    Dim h As Double
    Dim i As Double
    Dim j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z As Double

    Private Sub Form4_Load(sender As Object, e As EventArgs) Handles Me.Load
        atas = New Bitmap("D:\kuliah-
\edwin\VB\WindowsApplication1\WindowsApplication1\Resources\atas.bmp", True)
        atas.MakeTransparent(Color.Blue)
        inboard = New Bitmap("D:\kuliah-
\edwin\VB\WindowsApplication1\WindowsApplication1\Resources\inboard.bmp",
True)
        inboard.MakeTransparent(Color.White)
        engsel = New Bitmap("D:\kuliah-
\edwin\VB\WindowsApplication1\WindowsApplication1\Resources\engsel.bmp", True)
        engsel.MakeTransparent(Color.White)
    End Sub

    Private Sub Form4_Paint(sender As Object, e As PaintEventArgs) Handles
Me.Paint
    slew = Form1.a

    'slew = 0
    h = Form3.a
    i = Form3.d
    l = Math.Atan(204 / 36)
    m = 180 - (slew + 1) - (90 - 1)
    n = Math.Sin((m * Math.PI) / 180) * 36
    o = Math.Cos((m * Math.PI) / 180) * 36
    p = 90 - slew
    q = Math.Cos((p * Math.PI) / 180) * h
    r = Math.Sin((p * Math.PI) / 180) * h
    s = Math.Atan(244 / 34)
    t = 180 - (slew + s) - (90 - s)
    u = Math.Sin((t * Math.PI) / 180) * 34
    v = Math.Cos((t * Math.PI) / 180) * 34
    w = Math.Cos((p * Math.PI) / 180) * i
    x = Math.Sin((p * Math.PI) / 180) * i
    y = Math.Sin((p * Math.PI) / 180) * 30
    z = Math.Cos((p * Math.PI) / 180) * 30

    With atas
        e.Graphics.TranslateTransform(390, 330)
        e.Graphics.RotateTransform(0)
        e.Graphics.DrawImage(atas, -390, -330, 780, 660)
        e.Graphics.ResetTransform()
    End With
    With inboard
        e.Graphics.TranslateTransform(385, 630)
        e.Graphics.RotateTransform(slew)
        e.Graphics.DrawImage(inboard, -0, -8, 36, 16)
        e.Graphics.ResetTransform()
    End With
End Class

```

```

End With
With inboard
    e.Graphics.TranslateTransform(385 + n, 630 + o)
    e.Graphics.RotateTransform(slew + 270)
    e.Graphics.DrawImage(inboard, -0, -8, CInt(h), 16)
    e.Graphics.ResetTransform()
End With
With inboard
    e.Graphics.TranslateTransform(385 + n + q, 630 + o - r)
    e.Graphics.RotateTransform(slew + 180)
    e.Graphics.DrawImage(inboard, -0, -8, 34, 16)
    e.Graphics.ResetTransform()
End With
With outboard
    e.Graphics.TranslateTransform(385 + n + q - u, 630 + o - r - v)
    e.Graphics.RotateTransform(slew + 270)
    e.Graphics.DrawImage(inboard, -0, -8, CInt(i), 16)
    e.Graphics.ResetTransform()
End With
With outboard
    e.Graphics.TranslateTransform(385 + n + q - u + w, 630 + o - r - v
- x)
        e.Graphics.RotateTransform(slew)
        e.Graphics.DrawImage(inboard, -0, -8, 30, 16)
        e.Graphics.ResetTransform()
End With
With elbow
    e.Graphics.TranslateTransform(385 + n + q - u + w + y, 630 + o - r
- v - x + z)
        e.Graphics.RotateTransform(270)
        e.Graphics.DrawImage(inboard, -0, -8, 20, 16)
        e.Graphics.ResetTransform()
End With
With engsel
    e.Graphics.TranslateTransform(385, 630)
    e.Graphics.RotateTransform(0)
    e.Graphics.DrawImage(engsel, -8, -8, 16, 16)
    e.Graphics.ResetTransform()
End With
With engsel
    e.Graphics.TranslateTransform(385 + n, 630 + o)
    e.Graphics.RotateTransform(0)
    e.Graphics.DrawImage(engsel, -8, -8, 16, 16)
    e.Graphics.ResetTransform()
End With
With engsel
    e.Graphics.TranslateTransform(385 + n + q, 630 + o - r)
    e.Graphics.RotateTransform(0)
    e.Graphics.DrawImage(engsel, -8, -8, 16, 16)
    e.Graphics.ResetTransform()
End With
With engsel
    e.Graphics.TranslateTransform(385 + n + q - u, 630 + o - r - v)
    e.Graphics.RotateTransform(0)
    e.Graphics.DrawImage(engsel, -8, -8, 16, 16)
    e.Graphics.ResetTransform()
End With
With engsel
    e.Graphics.TranslateTransform(385 + n + q - u + w, 630 + o - r - v
- x)

```



```
        e.Graphics.RotateTransform(0)
        e.Graphics.DrawImage(engsel, -8, -8, 16, 16)
        e.Graphics.ResetTransform()
    End With
    With engsel
        e.Graphics.TranslateTransform(385 + n + q - u + w + y, 630 + o - r
- v - x + z)
        e.Graphics.RotateTransform(0)
        e.Graphics.DrawImage(engsel, -8, -8, 16, 16)
        e.Graphics.ResetTransform()
    End With
End Sub

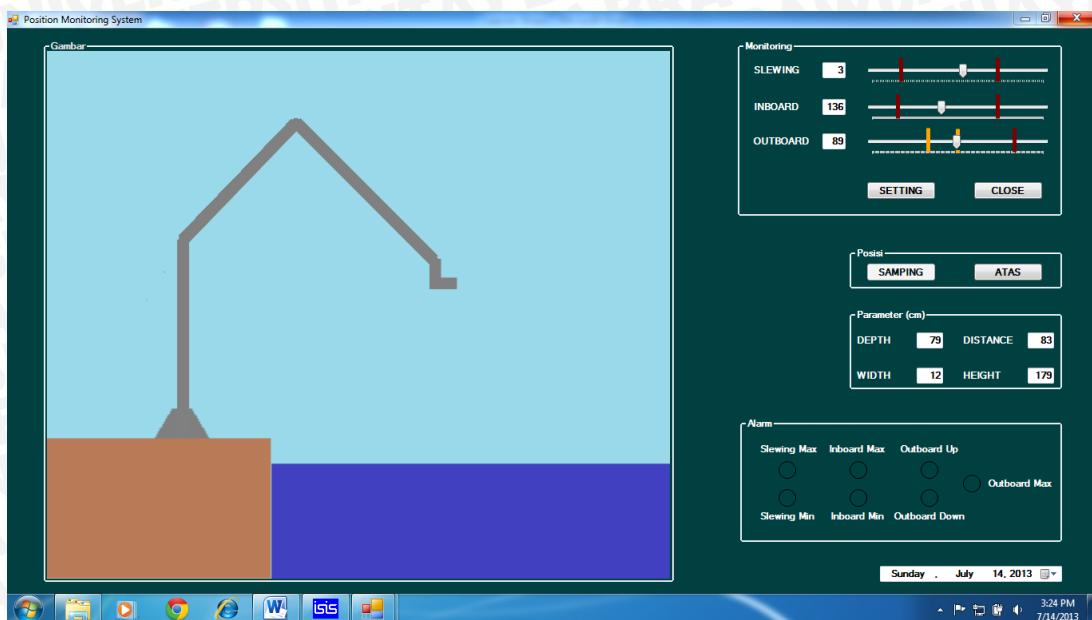
Private Sub Timer1_Tick(sender As Object, e As EventArgs) Handles
Timer1.Tick
    Me.Refresh()
End Sub
End Class
```



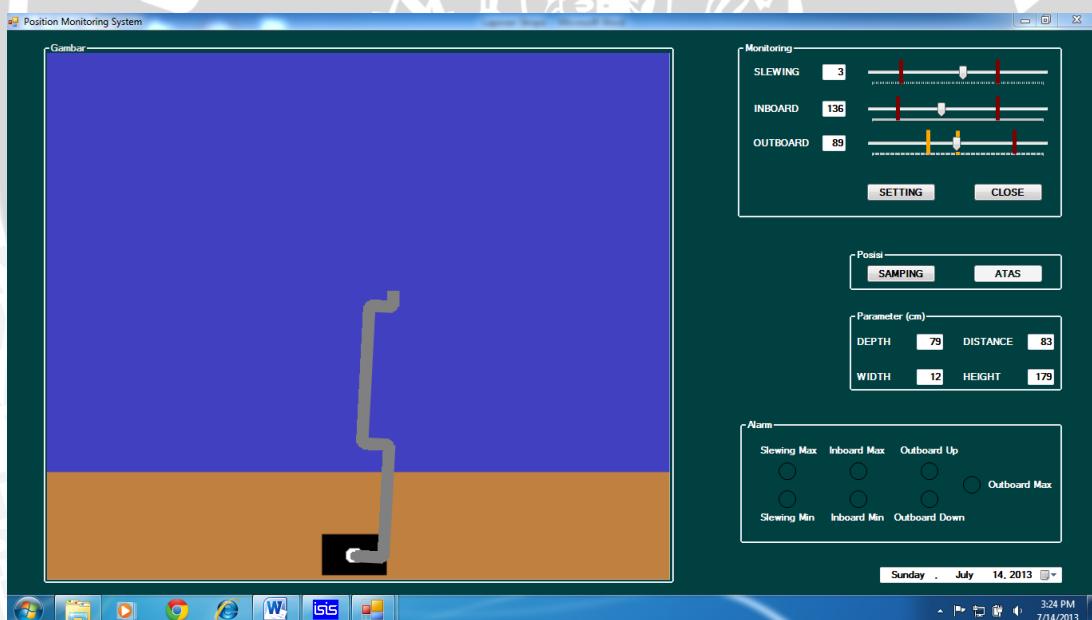


LAMPIRAN VIII

Tampilan Program Visual Studio



Gambar 7 Hasil Program PMS *Loading Arm* Tampak dari Samping

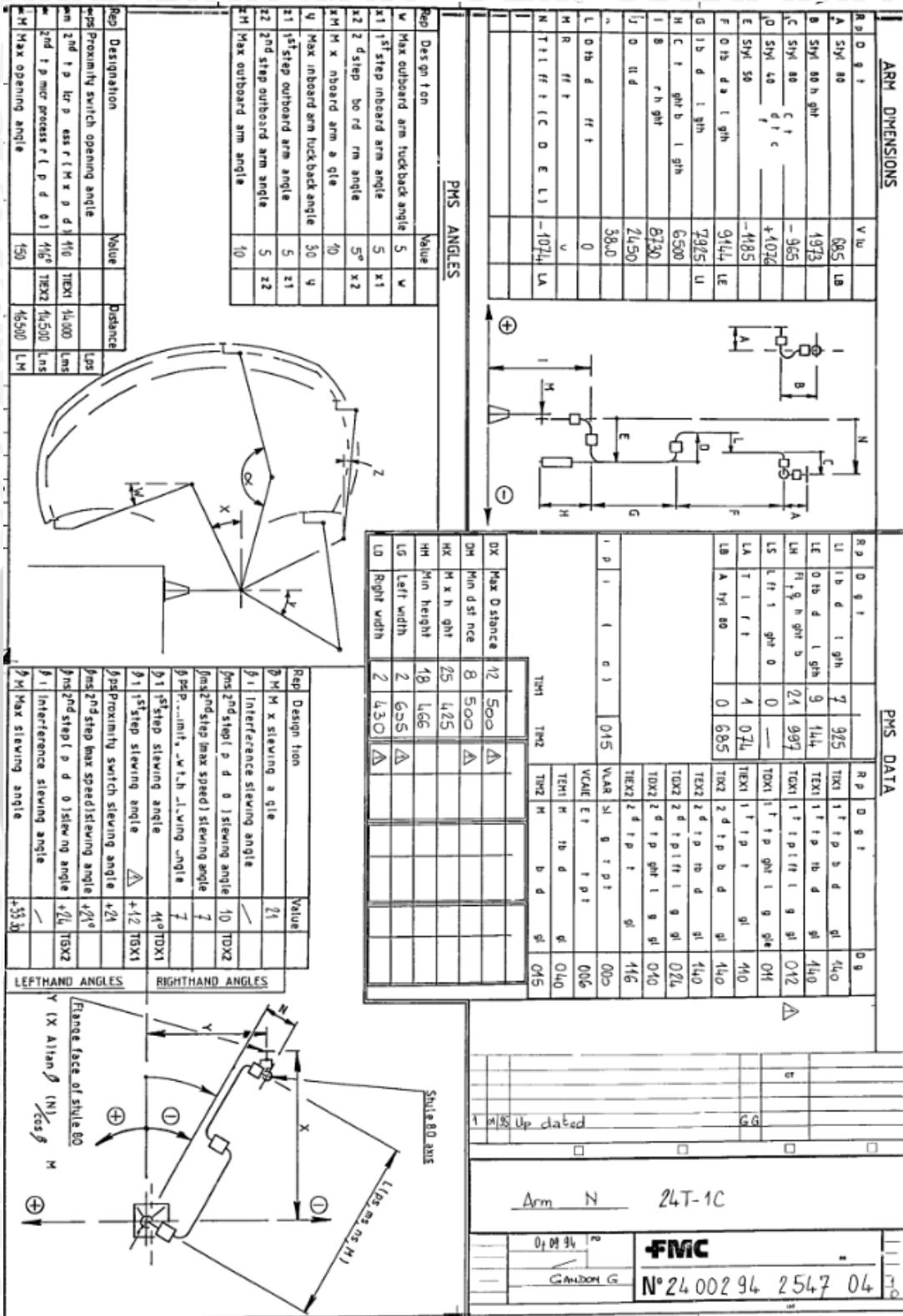


Gambar 8 Hasil Program PMS *Loading Arm* Tampak dari Atas



LAMPIRAN IX

Data PMS





LAMPIRAN X

Datasheet

Features

- High-performance, Low-power Atmel® AVR® 8-bit Microcontroller
- Advanced RISC Architecture
 - 130 Powerful Instructions – Most Single-clock Cycle Execution
 - 32 × 8 General Purpose Working Registers
 - Fully Static Operation
 - Up to 16MIPS Throughput at 16MHz
 - On-chip 2-cycle Multiplier
- High Endurance Non-volatile Memory segments
 - 8Kbytes of In-System Self-programmable Flash program memory
 - 512Bytes EEPROM
 - 1Kbyte Internal SRAM
 - Write/Erase Cycles: 10,000 Flash/100,000 EEPROM
 - Data retention: 20 years at 85°C/100 years at 25°C⁽¹⁾
 - Optional Boot Code Section with Independent Lock Bits
 - In-System Programming by On-chip Boot Program
 - True Read-While-Write Operation
 - Programming Lock for Software Security
- Peripheral Features
 - Two 8-bit Timer/Counters with Separate Prescaler, one Compare Mode
 - One 16-bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture Mode
 - Real Time Counter with Separate Oscillator
 - Three PWM Channels
 - 8-channel ADC in TQFP and QFN/MLF package
 - Eight Channels 10-bit Accuracy
 - 6-channel ADC in PDIP package
 - Six Channels 10-bit Accuracy
 - Byte-oriented Two-wire Serial Interface
 - Programmable Serial USART
 - Master/Slave SPI Serial Interface
 - Programmable Watchdog Timer with Separate On-chip Oscillator
 - On-chip Analog Comparator
- Special Microcontroller Features
 - Power-on Reset and Programmable Brown-out Detection
 - Internal Calibrated RC Oscillator
 - External and Internal Interrupt Sources
 - Five Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, and Standby
- I/O and Packages
 - 23 Programmable I/O Lines
 - 28-lead PDIP, 32-lead TQFP, and 32-pad QFN/MLF
- Operating Voltages
 - 2.7V - 5.5V (ATmega8L)
 - 4.5V - 5.5V (ATmega8)
- Speed Grades
 - 0 - 8MHz (ATmega8L)
 - 0 - 16MHz (ATmega8)
- Power Consumption at 4Mhz, 3V, 25°C
 - Active: 3.6mA
 - Idle Mode: 1.0mA
 - Power-down Mode: 0.5µA



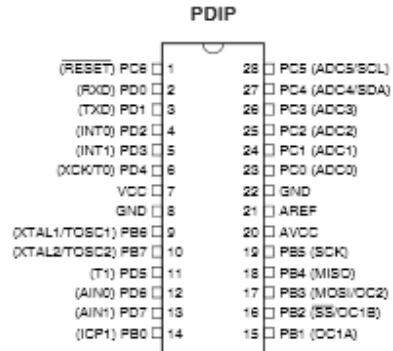
8-bit AVR® with 8KBytes In-System Programmable Flash

**ATmega8
ATmega8L**

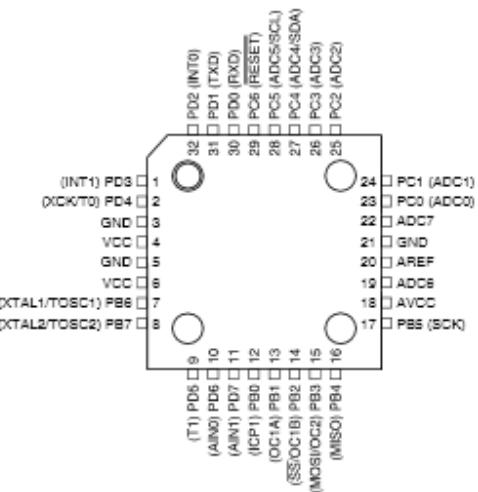




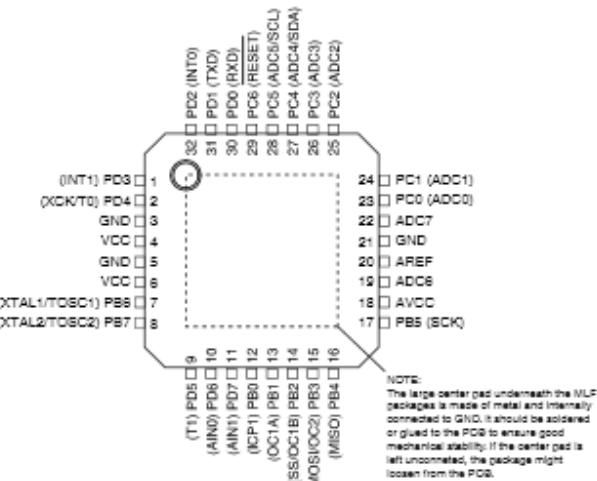
Pin Configurations



TQFP Top View



MLF Top View



Pin Descriptions

VCC	Digital supply voltage.
GND	Ground.
Port B (PB7..PB0)	Port B is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port B output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port B pins that are externally pulled low will source current if the pull-up resistors are activated. The Port B pins are tri-stated when a reset condition becomes active, even if the clock is not running.
XTAL1/XTAL2/TOSC1/ TOSC2	Depending on the clock selection fuse settings, PB6 can be used as input to the inverting Oscillator amplifier and input to the internal clock operating circuit. Depending on the clock selection fuse settings, PB7 can be used as output from the inverting Oscillator amplifier. If the Internal Calibrated RC Oscillator is used as chip clock source, PB7..6 is used as TOSC2..1 input for the Asynchronous Timer/Counter2 if the AS2 bit in ASSR is set. The various special features of Port B are elaborated in " Alternate Functions of Port B " on page 58 and " System Clock and Clock Options " on page 25.
Port C (PC5..PC0)	Port C is an 7-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port C output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port C pins that are externally pulled low will source current if the pull-up resistors are activated. The Port C pins are tri-stated when a reset condition becomes active, even if the clock is not running.
PC6/RESET	If the RSTDISBL Fuse is programmed, PC6 is used as an I/O pin. Note that the electrical characteristics of PC6 differ from those of the other pins of Port C. If the RSTDISBL Fuse is unprogrammed, PC6 is used as a Reset input. A low level on this pin for longer than the minimum pulse length will generate a Reset, even if the clock is not running. The minimum pulse length is given in Table 15 on page 38 . Shorter pulses are not guaranteed to generate a Reset. The various special features of Port C are elaborated on page 61 .
Port D (PD7..PD0)	Port D is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port D output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port D pins that are externally pulled low will source current if the pull-up resistors are activated. The Port D pins are tri-stated when a reset condition becomes active, even if the clock is not running. Port D also serves the functions of various special features of the ATmega8 as listed on page 63 .
RESET	Reset input. A low level on this pin for longer than the minimum pulse length will generate a reset, even if the clock is not running. The minimum pulse length is given in Table 15 on page 38 . Shorter pulses are not guaranteed to generate a reset.



AV_{CC}	AV _{CC} is the supply voltage pin for the A/D Converter, Port C (3..0), and ADC (7..6). It should be externally connected to V _{CC} , even if the ADC is not used. If the ADC is used, it should be connected to V _{CC} through a low-pass filter. Note that Port C (5..4) use digital supply voltage, V _{CC} .
AREF	AREF is the analog reference pin for the A/D Converter.
ADC7..6 (TQFP and QFN/MLF Package Only)	In the TQFP and QFN/MLF package, ADC7..6 serve as analog inputs to the A/D converter. These pins are powered from the analog supply and serve as 10-bit ADC channels.



Figure 14. Reset Logic

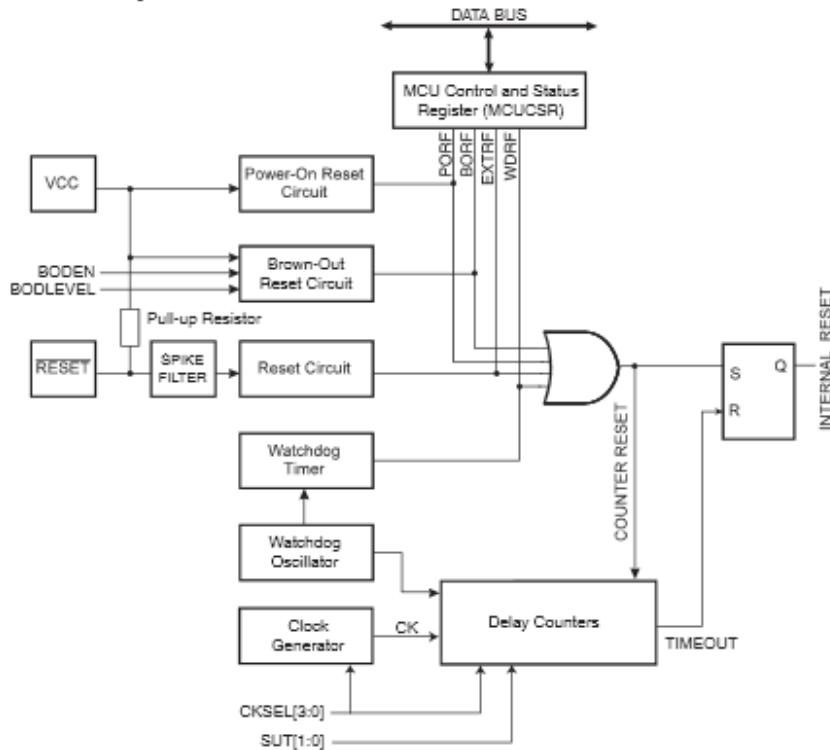


Table 15. Reset Characteristics

Symbol	Parameter	Condition	Min	Typ	Max	Units
V_{POT}	Power-on Reset Threshold Voltage (rising) ⁽¹⁾			1.4	2.3	V
	Power-on Reset Threshold Voltage (falling)			1.3	2.3	
V_{RST}	RESET Pin Threshold Voltage		0.2		0.9	V_{CC}
t_{RST}	Minimum pulse width on RESET Pin				1.5	μs
V_{BOT}	Brown-out Reset Threshold Voltage ⁽²⁾	BODLEVEL = 1	2.4	2.6	2.9	V
		BODLEVEL = 0	3.7	4.0	4.5	
t_{BOD}	Minimum low voltage period for Brown-out Detection	BODLEVEL = 1	2			μs
		BODLEVEL = 0	2			
V_{HYST}	Brown-out Detector hysteresis			130		mV

Notes:

- The Power-on Reset will not work unless the supply voltage has been below V_{POT} (falling).
- V_{POT} may be below nominal minimum operating voltage for some devices. For devices where this is the case, the device is tested down to $V_{CC} = V_{POT}$ during the production test. This guarantees that a Brown-out Reset will occur before V_{CC} drops to a voltage where correct operation of the microcontroller is no longer guaranteed. The test is performed using BODLEVEL = 1 for ATmega8L and BODLEVEL = 0 for ATmega8. BODLEVEL = 1 is not applicable for ATmega8.

USART

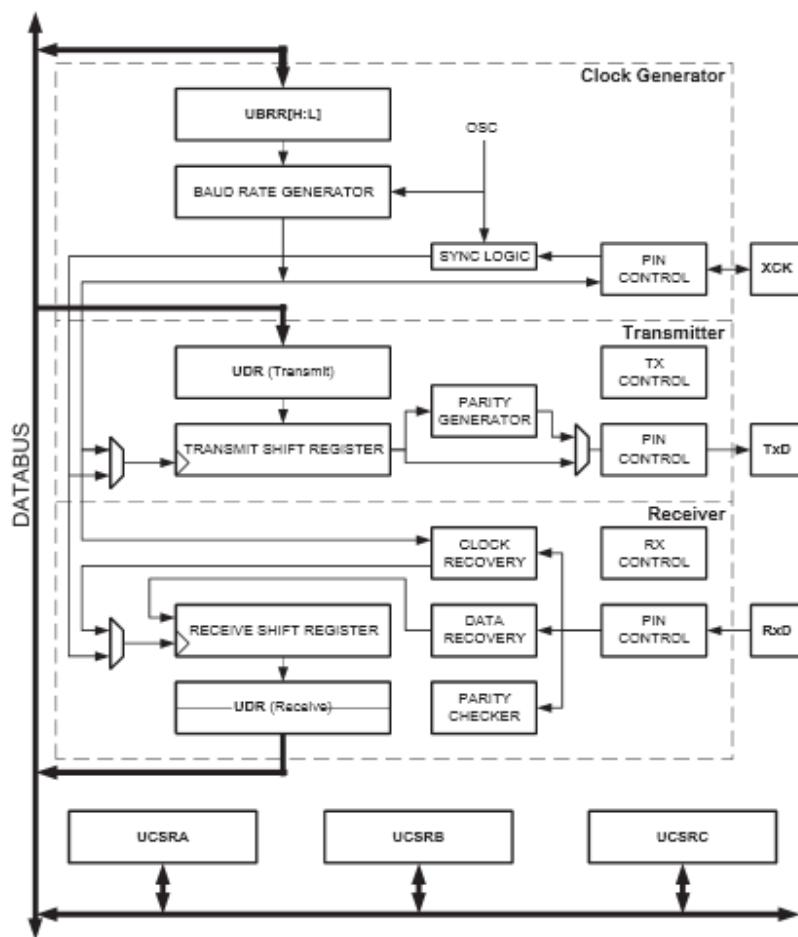
The Universal Synchronous and Asynchronous serial Receiver and Transmitter (USART) is a highly-flexible serial communication device. The main features are:

- Full Duplex Operation (Independent Serial Receive and Transmit Registers)
- Asynchronous or Synchronous Operation
- Master or Slave Clocked Synchronous Operation
- High Resolution Baud Rate Generator
- Supports Serial Frames with 5, 6, 7, 8, or 9 Databits and 1 or 2 Stop Bits
- Odd or Even Parity Generation and Parity Check Supported by Hardware
- Data OverRun Detection
- Framing Error Detection
- Noise Filtering Includes False Start Bit Detection and Digital Low Pass Filter
- Three Separate Interrupts on TX Complete, TX Data Register Empty and RX Complete
- Multi-processor Communication Mode
- Double Speed Asynchronous Communication Mode

Overview

A simplified block diagram of the USART Transmitter is shown in [Figure 61](#). CPU accessible I/O Registers and I/O pins are shown in bold.

Figure 61. USART Block Diagram⁽¹⁾



Note: 1. Refer to "Pin Configurations" on page 2, Table 30 on page 64, and Table 29 on page 64 for USART pin placement

Analog-to-Digital Converter

Features

- 10-bit Resolution
- 0.5 LSB Integral Non-linearity
- ± 2 LSB Absolute Accuracy
- 13 μ s - 260 μ s Conversion Time
- Up to 15 kSPS at Maximum Resolution
- 6 Multiplexed Single Ended Input Channels
- 2 Additional Multiplexed Single Ended Input Channels (TQFP and QFN/MLF Package only)
- Optional Left Adjustment for ADC Result Readout
- 0 - V_{cc} ADC Input Voltage Range
- Selectable 2.56V ADC Reference Voltage
- Free Running or Single Conversion Mode
- Interrupt on ADC Conversion Complete
- Sleep Mode Noise Canceler

The ATmega8 features a 10-bit successive approximation ADC. The ADC is connected to an 8-channel Analog Multiplexer which allows eight single-ended voltage inputs constructed from the pins of Port C. The single-ended voltage inputs refer to 0V (GND).

The ADC contains a Sample and Hold circuit which ensures that the input voltage to the ADC is held at a constant level during conversion. A block diagram of the ADC is shown in [Figure 90 on page 190](#).

The ADC has a separate analog supply voltage pin, AV_{cc}. AV_{cc} must not differ more than ± 0.3 V from V_{cc}. See the paragraph "[ADC Noise Canceler](#)" on page 195 on how to connect this pin.

Internal reference voltages of nominally 2.56V or AV_{cc} are provided On-chip. The voltage reference may be externally decoupled at the AREF pin by a capacitor for better noise performance.



+5V-Powered, Multichannel RS-232 Drivers/Receivers

MAX220-MAX249

ABSOLUTE MAXIMUM RATINGS—MAX220/222/232A/233A/242/243

Supply Voltage (V_{CC})	-0.3V to +6V	20-Pin Plastic DIP (derate 8.00mW/ $^{\circ}C$ above +70 $^{\circ}C$)	..440mW
Input Voltages		16-Pin Narrow SO (derate 8.70mW/ $^{\circ}C$ above +70 $^{\circ}C$)	..696mW
T_{IN}	-0.3V to (V_{CC} - 0.3V)	16-Pin Wide SO (derate 9.52mW/ $^{\circ}C$ above +70 $^{\circ}C$)	..762mW
R_{IN} (Except MAX220)	±30V	18-Pin Wide SO (derate 9.52mW/ $^{\circ}C$ above +70 $^{\circ}C$)	..762mW
R_{IN} (MAX220)	±25V	20-Pin Wide SO (derate 10.00mW/ $^{\circ}C$ above +70 $^{\circ}C$)	..800mW
T_{OUT} (Except MAX220) (Note 1)	±15V	20-Pin SSOP (derate 8.00mW/ $^{\circ}C$ above +70 $^{\circ}C$)	..640mW
T_{OUT} (MAX220)	±13.2V	16-Pin CERDIP (derate 10.00mW/ $^{\circ}C$ above +70 $^{\circ}C$)	..800mW
Output Voltages		18-Pin CERDIP (derate 10.53mW/ $^{\circ}C$ above +70 $^{\circ}C$)	..842mW
T_{OUT}	±15V	Operating Temperature Ranges	
R_{OUT}	-0.3V to (V_{CC} + 0.3V)	MAX22_ _AC_ _ MAX2_ _C_ _	0 $^{\circ}C$ to +70 $^{\circ}C$
Driver/Receiver Output Short Circuited to GND	Continuous	MAX22_ _AE_ _ MAX2_ _E_ _	-40 $^{\circ}C$ to +85 $^{\circ}C$
Continuous Power Dissipation ($T_A = +70^{\circ}C$)		MAX22_ _AM_ _ MAX2_ _M_ _	-55 $^{\circ}C$ to +125 $^{\circ}C$
16-Pin Plastic DIP (derate 10.53mW/ $^{\circ}C$ above +70 $^{\circ}C$)	..842mW	Storage Temperature Range	-65 $^{\circ}C$ to +160 $^{\circ}C$
18-Pin Plastic DIP (derate 11.11mW/ $^{\circ}C$ above +70 $^{\circ}C$)	..889mW	Lead Temperature (soldering, 10s)	..+300 $^{\circ}C$

Note 1: Input voltage measured with T_{OUT} in high-impedance state, $SHDN$ or $V_{CC} = 0V$.

Note 2: For the MAX220, V_+ and V_- can have a maximum magnitude of 7V, but their absolute difference cannot exceed 13V.

Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of the specifications is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

ELECTRICAL CHARACTERISTICS—MAX220/222/232A/233A/242/243

($V_{CC} = +5V \pm 10\%$, $C_1-C_4 = 0.1\mu F$, MAX220, $C_1 = 0.047\mu F$, $C_2-C_4 = 0.33\mu F$, $TA = T_{MIN}$ to T_{MAX} , unless otherwise noted.)

PARAMETER	CONDITIONS	MIN	TYP	MAX	UNITS
RS-232 TRANSMITTERS					
Output Voltage Swing	All transmitter outputs loaded with $3k\Omega$ to GND	±5	±8		V
Input Logic Threshold Low			1.4	0.8	V
Input Logic Threshold High	All devices except MAX220	2	1.4		V
	MAX220: $V_{CC} = 5.0V$	2.4			
Logic Pull-Up/Input Current	All except MAX220, normal operation	5	40		μA
	$SHDN = 0V$, MAX222/242, shutdown, MAX220	±0.01	±1		
Output Leakage Current	$V_{CC} = 5.6V$, $SHDN = 0V$, $V_{OUT} = \pm 15V$, MAX222/242	±0.01	±10		μA
	$V_{CC} = SHDN = 0V$, $V_{OUT} = \pm 15V$	±0.01	±10		
Data Rate			200	116	kbps
Transmitter Output Resistance	$V_{CC} = V_+ = V_- = 0V$, $V_{OUT} = \pm 2V$	300	10M		Ω
Output Short-Circuit Current	$V_{OUT} = 0V$	±7	±22		mA
RS-232 RECEIVERS					
RS-232 Input Voltage Operating Range				±30	V
RS-232 Input Threshold Low	$V_{CC} = 5V$	All except MAX243 R_{2IN}	0.8	1.3	V
		MAX243 R_{2IN} (Note 2)	-3		
RS-232 Input Threshold High	$V_{CC} = 5V$	All except MAX243 R_{2IN}	1.8	2.4	V
		MAX243 R_{2IN} (Note 2)	-0.5	-0.1	
RS-232 Input Hysteresis	All except MAX243, $V_{CC} = 5V$, no hysteresis in shdn.	0.2	0.5	1	V
		MAX243	1		
RS-232 Input Resistance			3	5	$k\Omega$
TTL/CMOS Output Voltage Low	$I_{OUT} = 3.2mA$			0.2	0.4
TTL/CMOS Output Voltage High	$I_{OUT} = -1.0mA$		3.5	$V_{CC} - 0.2$	V
TTL/CMOS Output Short-Circuit Current	Sourcing $V_{OUT} = GND$	-2	-10		
	Shrinking $V_{OUT} = V_{CC}$	10	30		mA

TLC485

Port-powered RS232 to RS485 converter

Features

- ◊ Plug and play, small volume, easy to installation
- ◊ Convert RS-232 data transmit (TXD) and data receive (RXD) signal into balance half duplex RS-485 signals.
- ◊ Powered from RS-232 serial lines — no power supply needed!
- ◊ Point to multiple point
- ◊ Automatic Send Data Control - no software drivers needed!
- ◊ Auto sense serial signal rate, zero delay automatic transmit

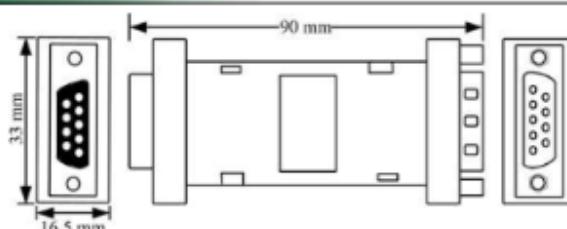


Introduction

TLC485 change TD and RD RS-232 lines to RS-485 signals. All are port powered and feature Automatic Send Data Control. SDC automatically enables the RS-485 driver when data is present on the RS-232 side. Control of the driver is automatic at rates up to

115.2 kbps. It is used widely in the field of electricity, industrial automatic control, IC card billing system. It is a high quality, economical data interface conversion product.

Installation



Specifications

Standard: Accord EIA RS-232, RS-485 standard	terminal block
RS-232 signal: TX, RX, GND	Working temperature: -20°C to 60°C
RS-485 signal: D+, D-, GND	Storage temperature: -40°C to 85°C
Working format: Asynchronism, point to point or multi point, 2 wire half-duplex	Relative humidity: 5% to 95%
Direction: Automatic send data control, auto distinguish and sense direction about data transfer	Power input: No power supply needed, powered from TXD, RTS, DTR signal in RS-232 port
Band rate: 300-115200bps, auto sense serial rate	Consumption:
Loading: Support 32 Nodes (can custom 128 nodes)	Static consumption less than 10mA
RS-485 side 1200M(9600bps), RS-232 side 5M	Dynamic consumption less than 40mA
Protection: 600W surge protection, 15KV ESD protection	L*W*H: 90mm*33mm*16.5mm
Connector: RS-232 side DB-9 female, RS-485 side 3	Case: Plastic
	Color: White
	Weight: 40g