



UNIVERSITAS DIPONEGORO

**PERANCANGAN DAN PEMBUATAN PROTOTIPE ROBOT
INSPEKSI REL KERETA API**

TUGAS SARJANA

CHOIRUN NIAM

L2E604200

**FAKULTAS TEKNIK
JURUSAN TEKNIK MESIN**

SEMARANG

MARET 2011



**DEPARTEMEN PENDIDIKAN NASIONAL
UNIVERSITAS DIPONEGORO SEMARANG
FAKULTAS TEKNIK JURUSAN TEKNIK MESIN**

TUGAS SARJANA

Diberikan kepada:

- Nama : Choirun Niam
NIM : L2E 604 200
Dosen Pembimbing : 1. Joga Dharma Setiawan, PhD.
2. Ir. Yurianto, MT.
Jangka Waktu : 6 (enam) bulan
Judul : PERANCANGAN DAN PEMBUATAN PROTOTIPE
ROBOT INSPEKSI REL KERETA API.
Isi Tugas : 1. Belajar merancang dan membuat prototipe robot inspeksi
rel kereta api.
2. Belajar membuat sistem kontrol menggunakan *optocoupler*
sebagai sensor kecepatan, dan Microcontroller *ATMega*
8535.
3. Belajar membuat sistem penggerak motor dengan
kendali jarak jauh via internet serta memberikan informasi
data berupa gambar via internet.
4. Penguasaan untuk *Software AutoCad 2006*, *Software*
Visual Studio 2008 dan *Software Team viewer*.

Dosen Pembimbing I

Dosen Pembimbing II

Joga Dharma Setiawan, Ph.D
NIP. 196811102005011001

Ir. Yurianto, MT
NIP. 195907271986031008

HALAMAN PERNYATAAN ORISINALITAS

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

NAMA : CHOIRUN NIAM

NIM : L2E 604 200

Tanda Tangan :

Tanggal : 25 Maret 2011

HALAMAN PENGESAHAN

Skripsi ini diajukan oleh :
NAMA : Choirun Niam
NIM : L2E 604 200
Jurusan/Program Studi : Teknik Mesin
Judul Skripsi : Perancangan Dan Pembuatan Prototipe Robot
Inspeksi Rel Kereta Api

Telah berhasil dipertahankan dihadapan Tim Penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Sarjana Teknik pada Jurusan Teknik Mesin, Fakultas Teknik, Universitas Diponegoro.

TIM PENGUJI

Pembimbing : Joga Dharma Setiawan, Ph.D ()
Penguji : Ir. Yurianto, MT ()
Penguji : Ir. Djoeli Satrijo, MT ()
Penguji : Ir. Sudargana, MT ()

Semarang, 25 Maret 2011

Ketua
Jurusan Teknik Mesin,

Dr.Ir.Dipl Ing Berkah Fajar TK.
NIP. 195907221987031003

HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS

Sebagai sivitas akademika Universitas Diponegoro, saya yang bertanda tangan di bawah ini:

Nama : Choirun Niam
NIM : L2E 604 200
Jurusan/Program Studi : Teknik Mesin
Fakultas : Teknik
Jenis Karya : Tugas Akhir

demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Diponegoro **Hak Bebas Royalti Noneksklusif** (*None-exclusive Royalty Free Right*) atas karya ilmiah saya dan Joga Dharma Setiawan, Ph.D sebagai pembimbing serta Ir. Yurianto, MT sebagai Co.pembimbing saya yang berjudul :

“Perancangan Dan Pembuatan Prototipe Robot Inspeksi Rel Kereta Api”

beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti/Noneksklusif ini Universitas Diponegoro berhak menyimpan, mengalihmedia/formatkan, mengelola dalam bentuk pangkalan data (*database*), merawat dan mempublikasikan tugas akhir saya dan Joga Dharma Setiawan, Ph.D sebagai pembimbing serta Ir.Yurianto, MT sebagai Co.pembimbing saya selama tetap mencantumkan nama kami sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Semarang
Pada Tanggal : 25 Maret 2011

Yang menyatakan

(Choirun Niam)

HALAMAN MOTTO

”Berjalan apa adanya, belajar dari kesalahan, mengalir menyatu dengan alam berfikir optimis serta terbuka, berusaha dan selalu berdoa kepada Allah SWT
(*LIFE IS AN ADVENTURE*)”

LEMBAR PERSEMBAHAN

Dalam pelaksanaan dan pembuatan tugas akhir ini kami banyak menerima dan bantuan dari berbagai pihak. Kami bersyukur sebesar-besarnya kepada Allah SWT atas semua karunia yang telah diberikan-Nya kepada kami sehingga kami dapat menyelesaikannya dengan baik. Juga tak lupa kami haturkan sholawat dan salam kepada Nabi Besar Muhammad SAW. Dan tanpa menghilangkan rasa hormat kami mengucapkan terima kasih kepada pihak yang telah membantu kami terutama kepada :

1. Fadhilatun Ni'mah, Nurul Wilda Alfiani serta Muhammad Zainal Fikri terima kasih doa dan supportnya.
2. Budi Heriyanto, terima kasih atas bimbingannya.
3. Zuniawan Prasetya, terima kasih atas bimbingannya.
4. Hendri Setiawan, terima kasih atas bimbingannya.
5. Herman martono, terima kasih atas supportnya.
6. Yusdi Ghozali, terima kasih atas supportnya.
7. Tendi, terima kasih atas bantuannya.
8. Paryanto, terima kasih atas supportnya.

KATA PENGANTAR

Puji dan syukur penulis ucapkan ke hadirat Allah SWT, atas segala rahmat dan hidayahnya-Nya yang telah diberikan sehingga penulis dapat menyelesaikan laporan Tugas Akhir ini yang berjudul “PERANCANGAN DAN PEMBUATAN PROTOTIPE ROBOT INSPEKSI REL KERETA API” Pada kesempatan ini perkenankan penulis mengucapkan terima kasih yang sebesar-besarnya kepada :

1. Joga Dharma Setiawan, BSc, MSc, PhD, selaku dosen pembimbing pertama Tugas Akhir ini.
2. Ir. Yuriyanto, MT, selaku dosen pembimbing kedua Tugas Akhir ini.
3. H. Sulchi Ghozali ayahanda tercinta, terima kasih atas do’a dan dukungannya
4. Hj. Sa’diyah ibuhanda tercinta, terima kasih atas do’a dan dukungannya.
5. Heri Susianto, selaku Partner Tugas akhir.

Penulis menyadari bahwa laporan ini masih banyak terdapat kekurangan, untuk itu penulis mengharap saran dan kritik yang membangun dari semua pihak.

Akhir kata semoga laporan ini dapat bermanfaat bagi pembaca dan semakin menambah pengetahuan kita.

Semarang, Maret 2011

Penulis

ABSTRACT

Daily care profession railways in Indonesia is still done manually. This final report will present the design and manufacture of inspection robot prototype railroads. Monitoring of a railway line using a robot-based camera. DC motor controlling is done through visual studio software on the computer and to drive the robot via the internet is required software team viewer. The sensor optocoupler works as a reader turns the DC motor. Readings are transmitted via RS232 serial communication to the microcontroller which will determine the working motor driver so that the robot can be driven. Images received from the webcam is processed through a laptop using software visual studio and received by the webcam images are sent using an Internet connection to a computer server. In this monitoring system consists of the Client and Server.

The software will be used consist of: Software Visual Studio 2008, C # Programming Language, Software Team viewer, Software Code Vision AVR V2.03.9 Evaluation, Software Autocad 2006. Hardware that will be used consisting of: a DC motor driver module, DC motor, microcontroller AT8535, optocoupler sensor, potensio meter, serial RS232,netbook.

Testing have been conducted in several aspects including the testing movement forward, backward, automatic with maximum speed. The dimensions of this robot is the length of 247cm, width of 78cm, 23cm tall weighing 15 kg. The test show that the robot has been designed to have good performance.

Keywords: Robot , RailRoad Inspection, Maintenance RailRoad..

ABSTRAK

Profesi perawatan harian rel kereta api di Indonesia masih dilakukan secara manual. Dalam laporan tugas akhir ini akan dipaparkan tentang perancangan dan pembuatan prototipe robot inspeksi rel kereta api. *Monitoring* suatu rel kereta api menggunakan robot yang berbasis *camera*. Pengontrolan motor DC dilakukan melalui *Software visual studio* pada komputer dan untuk menggerakkan robot via internet diperlukan *Software team viewer*. Sensor *optocoupler* bekerja sebagai pembaca putaran motor DC. Hasil pembacaan tersebut dikirim melalui komunikasi *serial RS232* ke mikrokontroler yang nantinya akan menentukan kerja driver motor sehingga robot dapat digerakkan. Gambar yang diterima dari *webcam* diolah melalui laptop dengan menggunakan *software visual studio* dan gambar yang diterima oleh *webcam* dikirim menggunakan koneksi internet ke komputer *server*. Pada sistem *monitoring* ini terdiri dari *Client* dan *Server*.

Perangkat lunak yang akan digunakan terdiri dari : *Software Visual Studio 2008*, Bahasa Pemrograman *C#*, *Software Team viewer*, *Software Code Vision AVR Evaluation V2.03.9*, *Software Autocad 2006*. Perangkat keras yang akan digunakan terdiri dari : modul *driver* motor DC, motor Dc, mikrokontroler AT8535, sensor *optocoupler*, *potensiometer*, *serial RS232*, *netbook*.

Pengujian telah dilakukan dalam beberapa aspek termasuk pengujian gerakan maju, mundur, otomatis dengan kecepatan maksimum. Dimensi robot ini adalah panjang 247cm, lebar 78cm, tinggi 23cm dengan berat 15 kg. Hasil pengujian menunjukkan bahwa robot yang telah dirancang memiliki *performance* yang baik.

Kata kunci: Robot, Robot Inspeksi Rel Kereta Api, Perawatan Rel.

.DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN TUGAS SARJANA	ii
HALAMAN PERNYATAAN ORISINALITAS	iii
HALAMAN PENGESAHAN	iv
HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI	v
HALAMAN MOTTO	vi
LEMBAR PERSEMBAHAN	vii
KATA PENGANTAR	viii
ABSTRACT	ix
ABSTRAK	x
DAFTAR ISI	xi
DAFTAR TABEL	xv
DAFTAR GAMBAR	xvi
NOMENKLATUR	xix

BAB I PENDAHULUAN

1.1 Latar Belakang	1
1.2 Tujuan Penelitian.....	2
1.3 Batasan Masalah.....	2
1.4 Metode yang digunakan	3
1.5 Sistematika Penulisan.....	4

BAB II DASAR TEORI

II.1 Perancangan Dan Gambar Teknik.....	5
II.1.1 Fase-Fase Dalam Proses Perancangan.....	5
II.1.2 Definisi Proyek, Perencanaan Proyek Dan Penyusunan Spesifikasi Teknis Proyek	6
II.1.3 Fase Perancangan Konsep Produk.....	6

II.1.4	Fase Perancangan Produk	7
II.1.4.1	Matriks Keputusan Dasar.....	7
II.1.4.2	Pemilihan Konsep Produk	7
II.1.5	Gambar Dan Spesifikasi Pembuatan Produk.....	8
II.2	Definisi Robot	9
II.2.1	Galeri Robot	10
II.2.1.1	<i>Mobile Robot</i>	10
II.2.1.2	Robot <i>Humanoid</i>	11
II.2.1.3	Robot Jaringan	11
II.2.1.4	<i>Under Water Robot</i> (Robot dalam air)	12
II.2.1.5	<i>Flying Robot</i> (Robot terbang)	12
II.2.1.6	Robot Manipulator (tangan)	13
II.2.2	Kendali Jarak Jauh.....	13
II.2.3	<i>Wireless Lan</i>	13
II.3	Perangkat Keras Dan Komponen Utama Robot.....	14
II.3.1	Rangka Robot, Roda Dan Sistem Penggerak Robot.....	14
II.3.2	Motor DC.....	14
II.3.3	<i>Accumulator</i> (Aki)	15
II.3.4	Kamera.....	16
II.3.5	Mikrokontroler.....	17
II.3.6	Motor <i>Driver</i> (<i>H-Bridge</i>)	18
II.3.7	Sensor <i>Optocoupler</i>	19
II.3.8	<i>Potensiometer</i>	20
II.3.9	Komunikasi <i>Serial RS232</i>	22
II.3.10	<i>Notebook</i>	26
II.3.11	Modem.....	27
II.4	Bagian Perangkat Lunak	27
II.4.1	<i>Software Team Viewer</i>	27
II.4.2	<i>Software Interface Mikrokontroler Codevision AVR</i>	28
II.4.3	<i>Software Visual Studio 2008</i>	31
II.5	Teori Dasar Perhitungan Mekanik Robot	32

II.5.1 Penentuan Motor Penggerak	32
II.5.2 Perhitungan Baterai	33
II.5.2.1 Perhitungan Energi Baterai.....	33
II.5.2.2 Waktu Pemakaian Baterai (Tpemakaian)	34
II.5.2.3 Jarak Tempuh Yang Dihasilkan (S).....	34

BAB III PERANCANGAN DAN PEMBUATAN PROTOTIPE ROBOT

III.1 Konsep Robot	37
III.1.1 Perancangan <i>Assembling Part</i> Robot Menggunakan <i>Software Autocad 2006</i>	38
III.1.2 Dimensi Umum Dan Bentuk Robot	39
III.1.3 Konfigurasi Roda Pada Robot	39
III.2 Perangkat Keras Robot	40
III.2.1 Kerangka, Roda, Baterai Dan Motor Penggerak	40
III.2.2 Mikrokontroler	41
III.2.3 Kamera	42
III.3 Perakitan Robot	43
III.4 Perangkat Lunak	43
III.4.1 Pemograman Dengan <i>Software Visual Studio 2008</i>	43
III.4.2 Pemograman Mikrokontroler	63
III.4.3 <i>Remote</i> Komputer Dengan <i>Software Team Viewer</i>	69
III.5 Prinsip Kerja.....	71

BAB IV HASIL DAN PEMBAHASAN

IV.1 Hasil Perancangan Robot	74
IV.2 Model Atau Prototipe Robot	74
IV.3 Pengoperasian Sistem	75
III.3.1 Monitoring Dengan <i>Form Visual C#</i>	75
III.3.2 Hasil <i>Capturing</i> Foto Inspeksi Rel	76
IV.4 Perhitungan Mekanik Dan Perhitungan Baterai Robot.....	76
IV.4.1 Perhitungan Mekanik Robot.....	76

IV.4.2 Perhitungan Baterai Robot.....	78
IV.5 Pengujian Baterai	79
BAB V PENUTUP	
V.1 Kesimpulan.....	80
V.2 Saran.....	81

DAFTAR PUSTAKA

LAMPIRAN

DAFTAR TABEL

Tabel 2.1 Karakteristik <i>Worm Gear</i> DC Motor	15
Tabel 2.2 Keterangan <i>Pin DB9</i>	23
Tabel 3.1 Dimensi Umum Robot.....	39
Tabel 3.2 Properti Komponen Pembuatan Kamera Dengan <i>Visual Studio 2008</i>	45
Tabel 3.3 Properti Komponen Komunikasi <i>Serial Port Visual Studio 2008</i>	54
Tabel 3.4 Konfigurasi Pengaktifan <i>Port</i> Pada Mikrokontroler.....	66
Tabel 4.1 Pengujian Konsumsi Arus Listrik Baterai Dengan Beban Rangka 15 kg....	79

DAFTAR GAMBAR

Gambar 2.1 :	Robot <i>mobile Sojourner Rover</i>	10
Gambar 2.2:	Contoh Robot <i>Humanoid</i>	11
Gambar 2.3:	<i>Worm Gear DC Motor</i>	15
Gambar 2.4:	Contoh baterai kering.....	15
Gambar 2.5:	<i>Webcam Logitech QuickCam®Cool</i>	16
Gambar 2.6:	Microcontroler <i>ATMEGA8535</i>	17
Gambar 2.7:	Skema Kontrol Pengendali Sistem Penggerak.....	18
Gambar 2.8:	Sirkuit <i>H-Bridge</i>	18
Gambar 2.9:	Skema Prinsip Kontrol <i>H-Bridge</i>	19
Gambar 2.10:	Sensor <i>Optocoupler</i>	20
Gambar 2.11:	Bentuk dan Rangkaian Dasar Sensor <i>Optocoupler</i>	20
Gambar 2.12:	Jenis <i>Potensiometer</i>	21
Gambar 2.13:	Grafik resistansi <i>potensiometer</i>	22
Gambar 2.14:	<i>DB9 Male</i>	22
Gambar 2.15:	<i>DB9 Female</i>	22
Gambar 2.16:	<i>Male Connector</i>	24
Gambar 2.17:	<i>Pins commonly used for RS-232 (serial)</i>	24
Gambar 2.18:	<i>Notebook</i>	26
Gambar 2.19:	Tampilan <i>Software Team Viewer</i>	28
Gambar 2.20:	<i>IDE</i> perangkat lunak <i>CodeVisionAVR</i>	29
Gambar 2.21:	<i>Code Generator</i> yang dapat digunakan untuk menginisialisasi <i>register-register</i> pada mikrokontroler <i>AVR</i>	30
Gambar 2.22:	Kode-kode program yang dibangkitkan otomatis oleh <i>code generator</i> .	30
Gambar 2.23:	Tampilan <i>Visual C# 2008</i>	31
Gambar 2.24:	<i>Design form Visual C# 2008</i>	31
Gambar 2.25:	Diagram benda bebas <i>simple</i> gerak robot.....	32
Gambar 3.1:	Diagram alir perancangan dan pembuatan prototipe robot.....	37
Gambar 3.2:	Jendela <i>softwareAutoCad 2006</i>	38
Gambar 3.3:	Konsep Desain Robot Inspeksi Rel.....	38

Gambar 3.4:	Bentuk Roda robot inspeksi rel kereta api.....	40
Gambar 3.5:	Konfigurasi roda pada robot inspeksi rel.....	40
Gambar 3.6:	Mikrokontroler AT8535.....	41
Gambar 3.7:	<i>Driver</i> Motor	41
Gambar 3.8:	Kendali Motor robot inspeksi rel kereta api dengan AT8535.....	42
Gambar 3.9:	Tampilan <i>form</i> kamera dengan <i>Visual C#</i>	42
Gambar 3.10:	Tampilan <i>New Project</i>	44
Gambar 3.11:	Posisi komponen yang ditambahkan pada <i>form</i>	44
Gambar 3.12:	<i>Outputpin Connection editor</i>	45
Gambar 3.13:	<i>Audio Outputpin dsVideoLogger1</i>	46
Gambar 3.14:	<i>OutputPin SnapShot</i>	46
Gambar 3.15:	Menambahkan <i>matlab</i> ke <i>visual C#</i>	49
Gambar 3.16:	<i>Tab</i> baru dengan nama <i>gauge</i>	49
Gambar 3.17:	Penambahan <i>Choose Items gauge</i>	50
Gambar 3.18:	<i>Form choose toolbox items</i>	50
Gambar 3.19:	<i>Browse</i> dari <i>choose toolbox items</i>	51
Gambar 3.20:	Centang <i>GMS Aircraft Instrument</i>	51
Gambar 3.21:	Tampilan <i>design form aircraft</i>	52
Gambar 3.22:	Edit <i>properties aircraft</i>	52
Gambar 3.23:	Tampilan <i>artifical horizon matlab 2008B</i>	53
Gambar 3.24:	<i>form application design serial conection</i>	54
Gambar 3.25:	Tampilan Desain <i>GUI Form C# Robot Inspeksi Rel</i>	63
Gambar 3.25:	<i>CodeVisionAVR</i>	64
Gambar 3.26:	<i>Create new project</i>	64
Gambar 3.27:	Pilihan untuk menggunakan <i>Code Wizard</i>	65
Gambar 3.28:	<i>Setting chip,pin Input/Output, dan LCD</i>	66
Gambar 3.29:	<i>Generate, save, dan exit</i>	67
Gambar 3.30:	Proses <i>download</i> program kedalam <i>AVR</i>	68
Gambar 3.31:	<i>Flowchart</i> pemrograman microcontroller.....	68
Gambar 3.32:	Tampilan <i>Software Team viewer versi 5</i>	69
Gambar 3.33:	<i>Create session</i> isi <i>ID Client</i>	70

Gambar 3.34: <i>Authentication password team viewer</i>	70
Gambar 3.35: Tampilan <i>team viewer</i> pada komputer <i>server</i>	71
Gambar 3.36: Konsep robot inspeksi rel kereta api.....	72
Gambar 3.37: Blok Diagram prinsip kerja sistem robot inspeksi rel	72
Gambar 4.1 : Model atau Prototipe Robot.....	74
Gambar 4.2 : Tampilan <i>Visual Monitoring</i> Kamera.....	75
Gambar 4.2 : Hasil <i>capture</i> permukaan rel.....	76
Gambar 4.3 : Hasil <i>capture</i> baut rel.....	76

NOMENKLATUR

f_s	Gaya gesek
g	Gravitasi (m/s^2)
I	Arus (Ampere)
m	Massa (kg)
t	waktu baterai habis (second)
V	Kecepatan max (m/s)
V	Voltase (Volt)
ω	Kecepatan sudut (Rps)
μ	Koefisien gesekan

DAFTAR PUSTAKA

1. Ahmadcyodyavalla2011.Com.”Job5: Bab Teori Dasar Pengkabelan RS232 Pada Konektor DB9”, Akses tanggal 11/02/20011.
2. Bambang Hermanto.2008. Perancangan Sistem Penggerak Robot Modular Nirkabel Berbasis Web. Institut Sains Dan Teknologi Nasional, Jakarta.
3. Dayat Kurniawan.2010. Aplikasi Elektronika Dengan Visual C# 2008 Express Edition. PT. Elex Media Komputindo, Jakarta.
4. Gesit Ari Nugroho.2006. Proyek Akhir Sistem Power Window Pada Suzuki Baleno. Jurusan Teknik Mesin : Universitas Negeri Semarang.
5. H.Darmawan Harsokoesoema.2004. Pengantar Perancangan Teknik Edisi Kedua. Bandung : Penerbit ITB.
6. Http: // InfoZone.com.: Berbagai macam bentuk robot Dan kontrol yang digunakan pada robot, Akses tanggal 24/05/2009.
7. http; // Logitech.com. *Camera Logitech QuickCam Cool*, Akses tanggal 10/06/2010.
8. Http: // Padepokan-IT.com. *Software Team Viewer*, Akses tanggal 20/03/2011.
9. Http: // Wikipedia.com. Mikrokontroler AVR, Akses tanggal 19/03/2011.
10. Http: // Wikipedia.com. *Rolling Resistance*, Akses tanggal 18/04/ 2011 at 15:37.
11. Http: //www.mikron123.com: Aplikasi mikrokontroler, Akses tanggal 26/04/20011.
12. Joseph Edward Shigley, Larry D.Mitchell. 1983. Perencanaan Teknik Mesin Edisi Keempat. Jakarta : Erlangga.
13. Kristof Goris. 2004-2005. Autonomous Mobile Robot Mechanical Design. Vrije Universiteit Brussel.
14. Thomas D.Gillespie : Fundamentals Of Vehicle Dynamics Society Of Automotive Engineers Inc 400 Common Wealth Drive Warrendale. PA 15096-0001.
15. Thiang, Handry Khoswanto, Yosafat Wahyudi D.S. Robot Mobil Omnidirection Beroda Empat Dengan Menggunakan Sistem Synchro Drive. Jurusan Teknik Elektro : Universitas Kristen Petra.

BAB I

PENDAHULUAN

1.1 LATAR BELAKANG

Profesi perawatan harian rel kereta api di Indonesia masih dilakukan secara manual. Petugas bagian perawatan rel biasanya hanya terdiri atas satu atau dua orang, sehingga akan sangat merepotkan dan melelahkan bila petugas tersebut harus berkeliling untuk memeriksa jalan rel yang sangat panjang tersebut. Efisiensi waktu dan tenaga yang dibutuhkan dalam perawatan rel tidaklah hemat oleh karena itu diperlukan suatu alat yang dapat menggantikan tugas berkeliling tersebut.

Hal tersebut yang mendasari pembuatan alat yang berbentuk prototipe robot dengan memanfaatkan teknologi yang ada. Pesatnya perkembangan dunia internet akhir-akhir ini telah memicu perkembangannya teknologi baru yang memanfaatkan teknologi internet tersebut sebagai media untuk mewujudkan impian manusia akan sebuah aplikasi pengoperasian peralatan jarak jauh. Robot inspeksi rel ini mampu memonitor keadaan rel, di antaranya: pengecekan ketinggian rel kanan dan kiri, memonitor baut, dan sambungan plat.

Perancangan desain robot dibuat dengan menggunakan *Software CAD* yang mempunyai kemampuan visualisasi, yaitu memodelkan bentuk 3 dimensi, program grafis berbasis *CAD* seperti: *Autocad* sangat membantu dalam merancang part – part serta dimensi robot sebelum robot direalisasikan.

Robot ini dikendalikan melalui sebuah perangkat netbook yang dilengkapi dengan program pengontrol. Untuk desain program pengontrol robot dibuat menggunakan *software microsoft visual C# 2008*. Dimana pada program ini dilengkapi dengan indikator ketinggian rel, tampilan kamera, dan mode arah gerak (maju/mundur).

1.2 TUJUAN PENELITIAN

Adapun tujuan dalam pengerjaan tugas akhir yang berjudul "**PERANCANGAN DAN PEMBUATAN PROTOTIPE ROBOT INSPEKSI REL KERETA API**" ini antara lain:

1. Merencanakan dan kemudian membuat suatu prototipe robot yang dapat berjalan diatas lintasan rel.
2. Membuat sistem penggerak motor dengan kendali jarak jauh via internet serta memberikan informasi data berupa gambar.
3. Menguji lama pemakaian baterai yang digunakan.

1.3 BATASAN MASALAH

1. Melakukan pengontrolan netbook via internet dengan menggunakan *software team viewer*.
2. Prototipe robot inspeksi ini menggunakan sensor *optocoupler* dan potensiometer digunakan untuk pengukuran kerataan rel.
3. *Software* analisis yang digunakan adalah: *Software CodeVision AVR Evaluation 2.03*, *Software Visual Studio 2005*, *Software team viewer*.
4. Baterai yang digunakan: aki 12 V 5 Ah dan 9 V 860 mA.

1.4 METODE YANG DIGUNAKAN

Perlu penulis tekankan bahwa tugas akhir yang berupa perancangan dan pembuatan prototipe robot inspeksi rel kereta api dalam pengambilan data digunakan beberapa metode, yaitu:

1. Studi lapangan

Melalui pengamatan dan pengalaman yang didapat untuk mengetahui proses pengerjaan suatu bahan juga peralatan yang akan digunakan dalam membuat model atau prototipe robot inspeksi rel kereta api.

Melalui pengamatan lapangan untuk memperoleh informasi tentang jenis bahan yang dipakai sehingga didapat bahan yang ekonomis dan terjangkau, namun memenuhi kriteria yang diperlukan dalam membangun sistem ini.

2. Studi literatur

Mengumpulkan data dengan mencatat atau membaca dari buku-buku yang berguna dengan pokok permasalahan ataupun referensi lain. Sebagian besar metode diambil dari situs – situs internet, dan buku cetak. Penulis melakukan pengumpulan data dengan cara *browsing*, membaca beberapa bagian dari buku-buku referensi internet dan materi-materi perkuliahan yang berhubungan dengan tugas akhir ini. Sehingga diperoleh gambaran dari prinsip kerja robot inspeksi rel kereta api yang dapat penulis gunakan sebagai landasan pemahaman dalam perancangan dan pengerjaan model atau prototipe robot ini.

3. Diskusi ilmiah

Mengumpulkan data dengan melakukan serangkaian diskusi dengan pihak lain yang lebih menguasai, sehingga didapat pemecahan masalah yang dihadapi.

4. Desain Prototipe Robot Inspeksi Rel kereta api

Melalui pembuatan dengan gambar bentuk model atau prototipe robot yang akan dibuat dan penyusunan aplikasi program.

5. Perancangan Dan Pembuatan Prototipe Robot

Merancang model atau prototipe robot dengan berbagai bahan dasar dari beberapa komponen pendukung, instalasi *software* dan memasang komponen lain menjadi satu kesatuan sistem yang kompak.

6. Uji coba Dan Analisa

Pengujian model yang telah selesai dibuat dan menganalisisnya untuk mengetahui fungsi-fungsi prototipe robot inspeksi rel sehingga diperoleh hasil dan kesimpulan.

I.5 SISTEMATIKA PENULISAN

Penulisan laporan Tugas Akhir ini terdiri dari beberapa bagian:

BAB I PENDAHULUAN

Berisikan latar belakang masalah, tujuan penulisan, pembatasan masalah, metode penulisan dan sistematika penulisan.

BAB II DASAR TEORI

Berisi tentang definisi robot, peralatan dan komponen-komponen *hardware* dan *software* yang akan digunakan, serta teori dasar perhitungan mekanik robot.

BAB III PERANCANGAN DAN PEMBUATAN PROTOTIPE ROBOT

Berisi uraian tentang konsep robot, penentuan dimensi / ukuran robot, konfigurasi roda robot, pembuatan prototipe dan perakitan *hardware*, pemrograman dengan *visual C#*, pemrograman mikrokontroler, koneksi dengan *software team viewer*.

BAB IV PENGUJIAN DAN HASIL

Berisi tentang pengujian baterai dan pengujian jalan di rel.

BAB V PENUTUP

Berisi kesimpulan dan saran.

DAFTAR PUSTAKA

LAMPIRAN

BAB II

DASAR TEORI

II.1 PERANCANGAN DAN GAMBAR TEKNIK

Sebelum sebuah produk dibuat, maka produk tersebut haruslah dirancang terlebih dahulu. Dalam bentuknya yang paling sederhana, hasil rancangan tersebut berupa sebuah sketsa atau gambar sederhana dari sebuah produk atau benda teknik yang akan dibuat. Apabila pembuat produk adalah perancangya sendiri, maka sketsa atau gambar cukup sederhana asalkan gambar sketsa sudah dapat dimengerti. Pada zaman modern ini, sebagian produk merupakan benda teknik yang rumit yang mempunyai banyak elemen penyusun benda teknik tersebut, dan pada umumnya sudah tidak dapat dibuat lagi oleh satu orang saja. Gambar yang dibuat tidak sederhana tetapi sudah rumit, sehingga harus dibuat dengan aturan dan cara menggambar yang jelas supaya dapat dibaca dan dimengerti oleh semua orang yang terlibat dalam kegiatan pembuatan produk.[5].

Gambar merupakan alat penghubung atau alat komunikasi antara perancang dan pembuat produk, dan antara semua orang yang terlibat dalam kegiatan perancangan dan pembuatan. Gambar teknik merupakan bahasa *universal* yang dipakai dalam kegiatan dan komunikasi antara orang-orang teknik. Dalam bentuk yang modern gambar rancangan produk berupa informasi digital yang disimpan dalam memori komputer. Gambar digital tersebut dapat pula dibaca oleh *software* dan hasil bacaannya diteruskan ke alat pembuat produk, yaitu alat pembuat produk yang dikendalikan oleh komputer dan yang akan membuat produk tersebut. [5].

II.1.1 Fase-Fase Dalam Proses Perancangan

Perancangan merupakan serangkaian yang berurutan, karena perancangan kemudian disebut sebagai proses perancangan yang mencakup seluruh kegiatan yang terdapat dalam proses perancangan. Kegiatan-kegiatan dalam proses perancangan tersebut dinamakan fase. Fase-fase dalam proses perancangan berbeda satu dengan yang lainnya. Setiap fase dari proses perancangan tersebut masih terdiri dari beberapa kegiatan yang dinamakan langkah-langkah dalam fase.[5].

II.1.2 Definisi Proyek, Perencanaan Proyek Dan Penyusunan Spesifikasi

Teknis Proyek

Ide produk yang akan dirancang dan dibuat dipilih dari daftar ide-ide produk yang diusulkan baik oleh bagian pemasaran maupun dari bagian-bagian lain dalam perusahaan pada langkah / kegiatan definisi proyek. Definisi proyek dan kegiatan-kegiatan lain dalam fase ini menghasilkan antara lain:

1. Pernyataan tentang masalah / produk yang akan dirancang.
2. Beberapa kendala atau *constraints* yang membatasi masalah tersebut.
3. Spesifikasi teknis.
4. Kriteria penerimaan (*acceptability criteria*).
5. Rancangan proyek.

Spesifikasi teknis adalah dinamis sifatnya, yaitu dapat mengalami perubahan selama proses perancangan dan pembuatan produk berlangsung. Spesifikasi teknis produk mengandung hal-hal berikut :

- a. Kinerja atau *performance* yang harus dicapai produk.
- b. Kondisi lingkungan, seperti temperatur, tekanan yang akan dialami oleh produk.
- c. Kondisi pengoperasian dari produk.
- d. Jumlah produk yang akan dibuat.
- e. Dimensi produk.
- f. Berat produk.
- g. Ergonomi.
- h. Keamanan dan *Safety*.
- i. Harga produk.

Jika dalam waktu penyelesaian perancangan dan pembuatan produk tercantum dalam spesifikasi, maka perlu dibuat jadwal penyelesaian setiap fase dan langkah dalam proses perancangan dan pembuatan produk.[5].

II.1.3 Fase Perancangan Konsep Produk

Tujuan dari fase perancangan konsep produk adalah menghasilkan alternatif produk sebanyak mungkin. Konsep produk yang dihasilkan fase ini masih berupa

skema atau dalam bentuk sketsa atau *skeleton*. Pada prinsipnya, semua alternatif konsep produk tersebut memenuhi spesifikasi teknik produk. Pada akhir fase perancangan konsep produk, dilakukan evaluasi pada hasil perancangan konsep produk untuk memilih salah satu atau beberapa konsep produk terbaik untuk dikembangkan pada fase ketiga yaitu fase perancangan produk atau fase pemberian bentuk pada konsep produk.[5].

II.1.4 Fase Perancangan Produk

Fase perancangan produk terdiri dari beberapa langkah, tetapi pada intinya pada fase ini solusi alternatif dalam bentuk skema atau sketsa dikembangkan lebih lanjut menjadi produk atau benda teknis yang bentuk, material dan dimensi elemen-elemennya ditentukan. Jika terdapat lebih dari satu solusi alternatif maka harus ditentukan satu solusi terakhir yang terbaik. Fase perancangan produk ini diakhiri dengan perancangan detail elemen-elemen produk yang kemudian dituangkan dalam gambar-gambar detail untuk proses pembuatan.[5].

II.1.4.1 Matriks Keputusan Dasar

Matriks keputusan dasar adalah sebuah matriks yang berfungsi sebagai petunjuk dalam pengambilan keputusan dari hasil rancangan yang akan di buat. Dalam matriks ini memuat aspek-aspek yang menjadi prioritas dalam pembuatan rancangan suatu produk. Matriks ini didapat dari penilaian para pelanggan tentang produk yang telah ada, dengan memberikan *quistionair* kepada pelanggan aspek mana yang menurut mereka sangat penting, penting, kurang penting, tidak penting dan sangat penting. Kemudian diambil sebagai dasar pertimbangan dalam suatu rancangan produk. Untuk membuat matriks keputusan dasar diperlukan pemilihan konsep produk.[5].

II.1.4.2 Pemilihan Konsep Produk

Kriteria perbandingan Kendaraan Inspeksi Rel Kereta Api dapat disusun sebagai berikut:

1. Keakuratan data: Diinginkan perancangan kendaraan inspeksi rel kereta api yang memberikan keakuratan data.
2. Daya tahan alat: Daya tahan dari kendaraan inspeksi rel kereta api ini diharapkan mampu beroperasi berulang kali dan bertahan dalam waktu yang lama.
3. Pengoperasian: Diinginkan dalam perancangan ini dihasilkan produk yang pengoperasiannya mudah.
4. Harga: Diharapkan dalam pembuatan produk ini tidak memakan biaya yang banyak, sehingga harganya sangat murah.
5. Estetika/desain: Estetika desain dari produk ini diharapkan mampu menyesuaikan dengan keinginan konsumen.
6. Fungsi: Diharapkan fungsi dari kendaraan ini dapat bekerja dengan baik sesuai dengan kebutuhan.
7. Perakitan: Diharapkan dari perancangan produk ini dihasilkan produk yang mudah dirakit dalam penggunaannya.
8. Ringan: Diharapkan dari perancangan produk ini dihasilkan produk dengan berat yang ringan, agar dalam penggunaannya sendiri mudah dibawa.
9. Perawatan: Dalam pembuatan kendaraan inspeksi rel kereta api ini diharapkan mudah dalam hal perawatannya.
10. Kualitas: Diharapkan kualitas dari produk yang telah dirancang mampu bersaing dengan produk yang ada dipasaran.

II.1.5 Gambar Dan Spesifikasi Pembuatan Produk

Gambar dan spesifikasi pembuatan produk terdiri dari:

- a. Gambar semua elemen produk lengkap dengan bentuk geometrinya, dimensi, kekasaran permukaan dan material.
- b. Gambar susunan komponen (*Assembly*).
- c. Gambar susunan produk.
- d. Spesifikasi yang memuat keterangan-keterangan yang tidak dapat dimuat dalam gambar.
- e. *Bill of material* dari semua komponen produk.

II.2 DEFINISI ROBOT

Robot adalah sebuah alat mekanik yang dapat melakukan tugas fisik, baik menggunakan pengawasan dan kontrol manusia, ataupun menggunakan program yang telah didefinisikan terlebih dulu (kecerdasan buatan). Robot biasanya digunakan untuk tugas yang berat, berbahaya, pekerjaan yang berulang dan kotor. Biasanya kebanyakan robot industri digunakan dalam bidang produksi. Penggunaan robot lainnya termasuk untuk pembersihan limbah beracun, penjelajahan bawah air dan luar angkasa, pertambangan, pekerjaan "cari dan tolong" (*search and rescue*), dan untuk pencarian tambang. Belakangan ini robot mulai memasuki pasaran konsumen dibidang hiburan, dan alat pembantu rumah tangga, seperti penyedot debu, dan pemotong rumput. [2].

Mobile Robot adalah konstruksi robot yang ciri khasnya adalah mempunyai aktuator berupa roda untuk menggerakkan keseluruhan badan robot tersebut, sehingga robot tersebut dapat melakukan perpindahan posisi dari satu titik ke titik yang lain. Robot mobil ini sangat disukai bagi orang yang mulai mempelajari robot. Hal ini karena membuat robot mobil tidak memerlukan kerja fisik yang berat. Untuk dapat membuat sebuah robot *mobile* minimal diperlukan pengetahuan tentang mikrokontroler dan sensor – sensor elektronik. *Base* robot mobil dapat dengan mudah dibuat dengan menggunakan *plywood* atau triplek, akrilik sampai menggunakan logam (aluminium) maupun plat besi. [2].

Kendaraan inspeksi ini dimodelkan sebagai sebuah robot yang dapat berjalan otomatis di atas rel kereta api, yang digerakkan oleh motor DC bertenaga *battery accu*. Kendaraan tersebut dilengkapi dengan kamera *webcam* yang berfungsi untuk mengambil rekaman dan gambar rel dari beberapa sudut. Kemudian data rekaman diolah oleh *processor* (berupa laptop) dan diteruskan ke *transmitter* yang berfungsi sebagai penguat sinyal dengan menggunakan bantuan *software team viewer* sebagai media komunikasi. Kemudian data dari *transmitter* dikirimkan ke *receiver* (berupa laptop) secara *online* via internet. Operator dapat memantau perjalanan robot inspeksi melalui monitor PC *receiver*. Sistem penggerak robot dapat diakses via *online*. [2].

II.2.1 Galeri Robot

Sebagai ilustrasi robot yang ada maka perlu di ketahui bahwa sangat beragam jenis robot yang dikembangkan dan di buat baik perorangan, kelompok penelitian, perusahaan ataupun badan – badan pemerintah yang tentunya dibuat dengan tujuan tertentu. Dengan adanya robot – robot yang telah di buat, maka kita bisa belajar dengan apa yang sudah ada, atau mengembangkannya, atau bahkan jika kita sedang mengembangkan suatu proyek robot maka bisa membandingkan nilai kebaruan robot yang sedang di buat dengan robot – robot yang sudah ada. [2].

II.2.1.1 Mobile Robot

Robot Mobil atau *Mobile Robot* adalah konstruksi robot yang ciri khasnya adalah mempunyai aktuator berupa roda untuk menggerakkan keseluruhan badan robot tersebut, sehingga robot tersebut dapat melakukan perpindahan posisi dari satu titik ke titik yang lain.

Robot mobil ini sangat disukai bagi orang yang mulai mempelajari robot. Hal ini karena membuat robot mobil tidak memerlukan kerja fisik yang berat. Untuk dapat membuat sebuah robot mobile minimal diperlukan pengetahuan tentang mikrokontroler dan sensor-sensor elektronik.

Base robot mobil dapat dengan mudah dibuat dengan menggunakan *plywood* /triplek, akrilik sampai menggunakan logam (aluminium). Robot mobil dapat dibuat sebagai pengikut garis (*Line Follower*) atau pengikut dinding (*Wall Follower*) ataupun pengikut cahaya.[2].



Gambar 2.1 :Robot *mobile Sojourner Rover* [2]

II.2.1.2 Robot *Humanoid*

Robot *Humanoid* adalah robot yang memiliki bentuk seperti manusia (*human*). Oleh karena itu ia memiliki kemampuan seperti layaknya manusia, yaitu dapat berjalan dengan kedua kaki (*bi-pedal*), memiliki tangan dengan jari-jari tangan. Robot *humanoid* yang cerdas biasanya dilengkapi dengan kamera yang pembuatnya dapat mengenali orang. [2].

Honda *ASIMO* adalah robot yang di buat oleh Honda Motor *Company*. Perusahaan ini telah melakukan penelitian rahasia tentang *humanoid* robot beberapa tahun lalu dan hasilnya adalah *Asimo*. *Asimo* kini tersedia versi yang berukuran lebih kecil. Penelitian robot ini telah dimulai tahun 1986. Kelebihan khusus *Asimo* adalah lebih kecil dan ringan, teknologi gerak yang lebih baik, operasional yang sederhana, berdisain seperti orang (*human-like robot*).

Perusahaan *Sony* merupakan perusahaan elektronik raksasa yang telah lama berkecimpung dalam bidang perobotan. Robot *humanoid* buatan *Sony* diberi nama *QRIO*. Robot ini harganya masih sangat mahal, bahkan menyamai sebuah mobil mewah. [2].



Gambar 2.2: Contoh Robot *Humanoid* [2]

II.2.1.3 Robot Jaringan

Robot jaringan adalah pendekatan baru untuk melakukan kontrol robot menggunakan jaringan internet dengan protokol TCP/IP. Perkembangan robot jaringan dipicu oleh kemajuan jaringan dan internet yang pesat. Dengan koneksi jaringan, proses kontrol dan monitoring, termasuk akuisisi data bila ada,

seluruhnya dilakukan melalui jaringan. Keuntungan lain, koneksi ini bisa dilakukan secara nirkabel.

Di Indonesia, pengembang robot jaringan belum banyak, meski pengembang dan komunitas robot secara umum sudah banyak. Hal ini disebabkan tuntutan teknis yang jauh lebih kompleks. Salah satu robot jaringan yang sudah berhasil dikembangkan adalah LIPI *Wireless Robot* (LWR) yang dikembangkan oleh Grup Fisika Teoritik dan Komputasi GFTK LIPI. Seperti ditunjukkan di *LWR*, seluruh proses kontrol dan *monitoring* bisa dilakukan melalui perambah internet. Lebih jauh, seluruh sistem dan protokol yang dikembangkan untuk *LWR* ini telah dibuka sebagai *open-source* dengan lisensi GNU *Public License (GPL)* di *Source Forge* dengan nama *open NR*. [6].

II.2.1.4 Under Water Robot (Robot dalam air)

Robot ini digunakan di bawah laut untuk memonitor kondisi bawah laut dan juga untuk mengambil sesuatu di bawah laut.

Ada beberapa unjuk kerja robot yang perlu diketahui, antara lain:

- Resolusi adalah perubahan gerak terkecil yang dapat diperintahkan oleh sistem kontrol pada lingkup kerja manipulator.
- Akurasi adalah besarnya penyimpangan / *deviasi* terhadap masukan yang diketahui.
- *Repeatability* adalah kemampuan robot untuk mengembalikan *end effector* (pemegang / *griper*) pada posisinya semula.

Fleksibilitas merupakan kelebihan yang dimiliki oleh robot secara umum jika dibandingkan dengan mesin konvensional. Hal ini pun tergantung kepada pemrogram dalam merencanakan pola gerakannya. [6].

II.2.1.5 Flying Robot (Robot Terbang)

Robot yang mampu terbang, robot ini menyerupai pesawat model yang diprogram khusus untuk memonitor keadaan di tanah dari atas, dan juga untuk meneruskan komunikasi. [6].

II.2.1.6 Robot Manipulator (tangan)

Robot ini hanya memiliki satu tangan seperti tangan manusia yang fungsinya untuk memegang atau memindahkan barang, contoh robot ini adalah robot las di Industri mobil, robot merakit elektronik dll.[6].

II.2.2 Kendali Jarak Jauh

Salah satu aspek dalam merancang suatu robot adalah bagaimana agar robot tersebut dapat di operasikan dari jarak tertentu. Ini biasanya disebut *Teleoperated* (terkendali-jauh). Robot terkendali jauh ini telah dibuat oleh beberapa ahli robot luar negeri. Dalam suatu lingkungan tertentu dimana manusia sangat berbahaya bila berada di tempat tertentu tersebut (misalnya fasilitas nuklir dan lainnya), maka hal ini dapat diselesaikan dengan menggunakan robot terkendali jauh. Bahkan sistem yang lebih canggih, memungkinkan mengendalikan robot dengan menggunakan internet. [2].

II.2.3 WIRELESS LAN

Istilah jaringan komputer tanpa kabel merupakan terjemahan dari istilah *wireless networking* yang berarti komunikasi data dalam sebuah jaringan komputer yang tidak memanfaatkan kabel sebagai media transmisi. Jaringan ini memanfaatkan gelombang mikro atau gelombang elektromagnetik sebagai media transmisinya. [2]. Kelebihan *wireless networking* antara lain:

1. Kebutuhan akan informasi di era pasar bebas membuat para pecandu informasi ingin *online* terus-menerus. Untuk keperluan ini komunikasi *wireless* sangat tepat karena mempunyai banyak aplikasi penting disamping juga menyediakan konektivitas bagi pengguna.
2. Untuk medan yang berat dan sulit seperti di pegunungan, hutan, dan rawa teknologi *wireless* lebih tepat digunakan daripada kabel.
3. Teknologi *wireless* dapat menjawab kebutuhan pemakai jaringan yang memiliki mobilitas tinggi. Dengan *wireless* jangkauan jaringan menjadi jauh lebih luas.

Kekurangan *wireless networking* antara lain:

1. Jangkauannya masih tetap terbatas.
2. Kemampuan transfer data lebih kecil dibandingkan jaringan kabel.
3. Keamanan data belum terjamin karena masih memungkinkan terjadinya penyadapan.
4. Proses instalasi sulit
5. Masih sering mengalami gangguan sehingga kemungkinan terjadinya kehilangan data masih cukup besar.

II.3 PERANGKAT KERAS DAN KOMPONEN UTAMA ROBOT

II.3.1. Rangka Robot, Roda dan Sistem Penggerak Robot

Dalam pembuatan robot inspeksi rel ini, dibuat sesederhana mungkin karena masih dalam tahap pengembangan. Rangka yang di gunakan terbuat dari bahan plat besi dengan ketebalan 1–2 mm untuk menopang beban yang ada. Sebagai penggerak, robot dilengkapi dengan dua pasang roda, dua roda di depan dan dua roda di bagian belakang. Pada bagian pasangan roda bagian belakang di lengkapidengan penggerak motor DC 12 V. Motor penggerak dikombinasikan dengan gir–gir untuk mereduksi putaran dan untuk mendapatkan torsi putaran yang lebih kuat agar bisa menggerakkan rangka beserta beban.

II.3.2. Motor DC

Worm gear dc motor merupakan salah satu dari jenis DC motor yang mempunyai torsi tinggi namun dengan rpm yang rendah. *Worm gear* dc motor merupakan DC motor yang sudah dilengkapi dengan *gear reducer* didalamnya. *Worm gear* dc motor dalam pemanfaatanya menggunakan tegangan sebesar 24 volt dengan arus 30 *ampere*. *Worm gear* dc motor juga memiliki bentuk yang ramping. Roda gigi yang ada didalam *Worm gear* dc motor merupakan roda gigi *worm* sehingga dengan adanya roda gigi *worm* ini tidak akan terjadi *blackless* akibat kontak roda gigi dan dapat mengunci pergerakan seperti *screw*. Pemanfaatan dari *Worm gear* dc motor ini adalah untuk otomatisasi membuka dan menutupnya kaca mobil, dan sistem *autolock* pada pintu mobil. [4].



Gambar 2.3: *Worm Gear DC Motor* [4]

Tabel 2.1 Karakteristik *Worm Gear DC Motor*

Nama	<i>Power Window Relay</i>
Seri	4 Ra 003 510-08 <i>Made in Jerman</i>
Current	200 mA (<i>coil load</i>)
Voltage	12 volt 10 A
Load	4 N. m
Speed	80 rpm
Operating Voltage Range	DC 10 – 12 volt
Operating Temperature Range	300 C – (+) 800 C

Dari segi *mechanical worm gear* dc motor ini mempunyai torsi yang besar karena digunakan untuk sistem *open/close* pada kaca jendela mobil.

II.3.3. *Accumulator (Aki)*

Accumulator (aki) digunakan sebagai sumber tegangan penggerak. Pada perangkat dinamis ini menggunakan aki merk yuasa 12V-5Ah. Contoh baterai aki merk yuasa seperti terlihat pada gambar 2.5.



Gambar 2.4: Contoh baterai aki merk yuasa [2]

II.3.4. Kamera

Kamera digunakan untuk memantau perangkat dari jauh. Dengan aplikasi *browser*, sehingga seorang dapat berkomunikasi lewat perangkat modem yang terkoneksi internet. Kamera akan ditampilkan pada *form software visual studio 2005*, Gambar webcam *Logitech QuickCam® Cool* seperti terlihat pada gambar 2.9.



Gambar 2.5: Webcam *Logitech QuickCam® Cool* [7]

II.3.5. Mikrokontroler

Umumnya pada suatu robot sudah dipastikan terdapat komponen mikrokontroler yang berisi algoritma program untuk memberikan logika pendeteksian berupa tingkah laku dari robot tersebut baik berupa pergerakan mekanis maupun perubahan visual. Untuk melakukan pergerakan mekanis tersebut dibutuhkan suatu pemacu gerak seperti motor dan beberapa kombinasi roda gir.

Mikrokontroler digunakan sebagai penyedia antara *server* dengan berbagai komponen baik berupa kontrol penggerak roda ataupun pembacaan data dari berbagai sensor. Pada perancangan model atau prototipe robot inspeksi rel ini digunakan mikrokontroler seri *AT8535* yang berfungsi sebagai pengendali gerak.

Mikrokontroler, sebagai suatu terobosan teknologi mikro prosesor dan mikro komputer, hadir memenuhi kebutuhan pasar (*market need*) dan teknologi baru.

Tidak seperti sistem komputer, yang mampu menangani berbagai macam program aplikasi (misalnya pengolah kata, pengolah angka, dan lain sebagainya), mikrokontroler hanya bisa digunakan untuk suatu aplikasi tertentu saja (hanya satu program saja yang bisa tersimpan). Perbedaan lainnya terletak pada perbandingan *RAM* dan *ROM*. Pada sistem komputer perbandingan *RAM* dan *ROM*-nya besar, artinya program-program pengguna disimpan dalam ruang *RAM* yang relatif besar, sedangkan rutin-rutin antarmuka perangkat keras disimpan dalam ruang *ROM* yang kecil. Sedangkan pada mikrokontroler, perbandingan *ROM* dan *RAM*-nya yang besar, artinya program kontrol disimpan dalam *ROM* (bisa *Masked ROM* atau *Flash PEROM*) yang ukurannya relatif lebih besar, sedangkan *RAM* digunakan sebagai tempat penyimpanan sementara, termasuk *register-register* yang digunakan pada mikrokontroler yang bersangkutan.

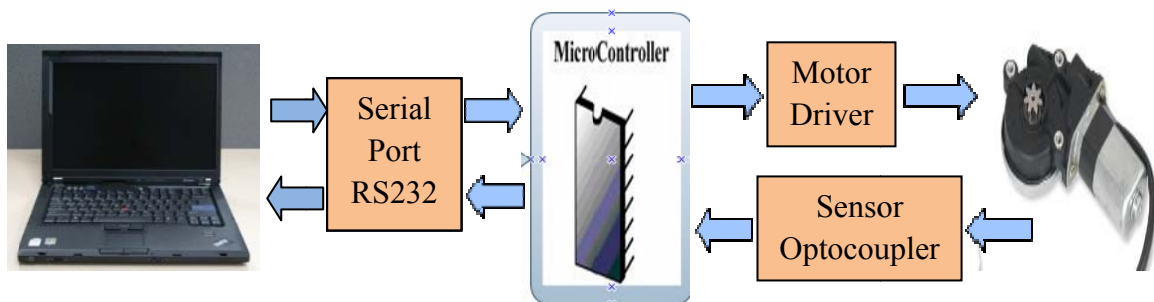
AT8535 adalah mikrokontroler keluaran *Atmel* dengan 4K byte *Flash PEROM* (*Programmable and Erasable Read Only Memory*), *AT8535* merupakan memori dengan teknologi *nonvolatile memory*, isi *memory* tersebut dapat diisi ulang ataupun dihapus berkali – kali. Memori ini biasa digunakan untuk menyimpan intruksi (perintah) berstandar *MCS-51 code* sehingga memungkinkan mikrokontroler ini untuk bekerja dalam *mode single chip operation* (*mode* operasi keping tunggal) yang tidak memerlukan *external memory* (memori luar) untuk menyimpan *source code* tersebut. [2].



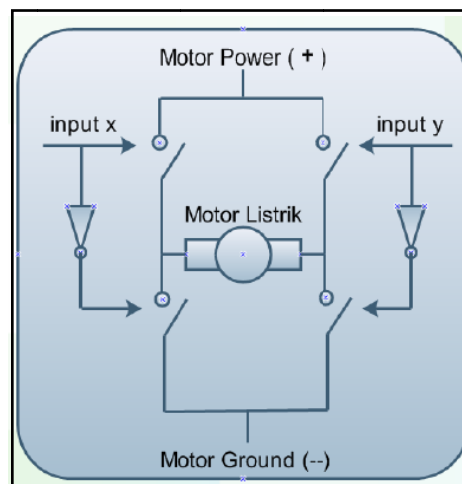
Gambar 2.6: *Microcontroller ATMEGA8535* [2]

II.3.6. Motor Driver (H-Bridge)

Salah satu metoda yang dapat digunakan dalam pengontrolan motor DC yaitu dengan menggunakan metoda jembatan-H atau *H-bridge*. Pada dasarnya rangkaian ini terdiri dari empat element switch yang dapat dikontrol baik pada posisi *on* atau *off* seperti terlihat pada gambar 2.11. [2].



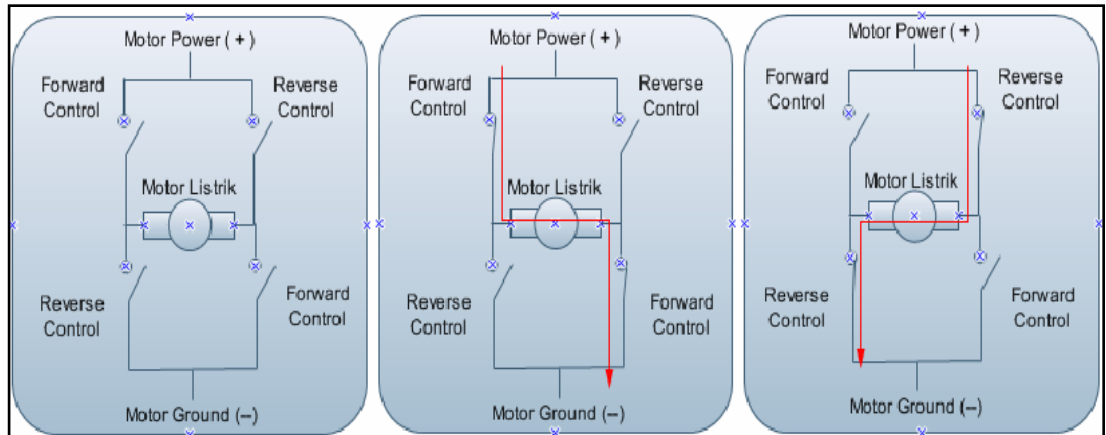
Gambar 2.7: Skema Kontrol Pengendali Sistem Penggerak [2]



Gambar 2.8: Sirkuit *H-Bridge* [2]

Ketika *switch* dalam kondisi *on* artinya motor akan terhubung baik itu terhadap *power* atau terhadap *ground* tergantung posisi *switch* tadi. Untuk mengendalikan supaya motor bergerak kearah maju atau mundur maka posisi *switch* yang berseberangan harus dalam kondisi *on* sehingga motor akan terhubung baik terhadap *power* dan terhadap *ground*, yang pada akhirnya motor akan berputar. Untuk menggerakkan motor dalam arah yang berlawanan dapat dikontrol dengan mengubah

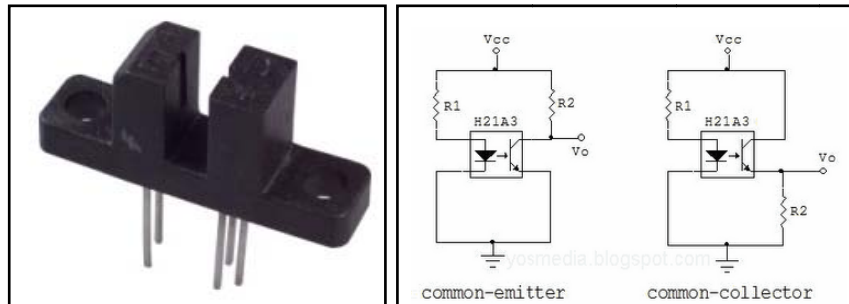
kondisi *switch* secara berlawanan dengan kondisi tadi seperti dapat terlihat pada gambar 2.11. [2].



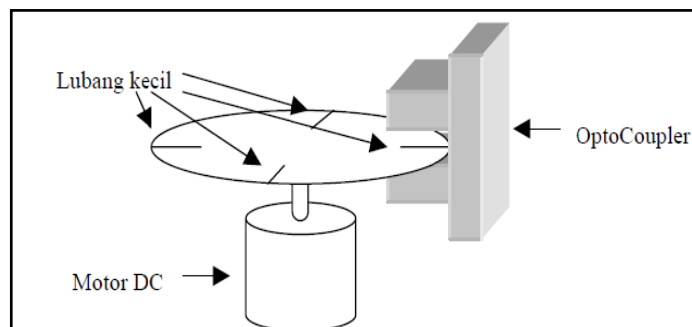
Gambar 2.9: Skema Prinsip Kontrol *H-Bridge* [2]

II.3.7. Sensor *Optocoupler*

Optocoupler adalah perangkat keras yang akan digunakan sebagai sensor kecepatan. *Optocoupler* merupakan gabungan dari *LED* infra merah dengan fototransistor yang terbungkus menjadi satu *chips*. Cahaya infra merah termasuk dalam gelombang elektromagnetik yang tidak tampak oleh mata telanjang. Sinar ini tidak tampak oleh mata karena mempunyai panjang gelombang berkas cahaya yang terlalu panjang bagi tanggapan mata manusia. Sinar infra merah mempunyai daerah frekuensi 1×10^{12} Hz sampai dengan 1×10^{14} GHz atau daerah frekuensi dengan panjang gelombang $1\mu\text{m} - 1\text{mm}$. [2].



Gambar 2.10: Sensor *Optocoupler* [2]



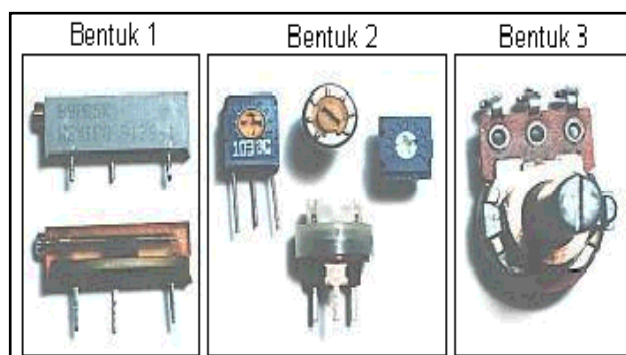
Gambar 2.11: Bentuk dan Rangkaian Dasar Sensor *Optocoupler* [2]

II.3.8. Potensiometer

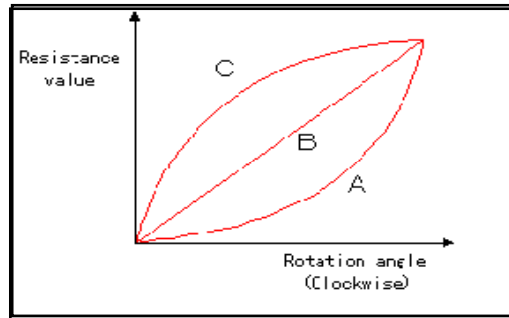
Potensiometer termasuk kedalam kategori resistor karena mempunyai nilai resistansi, namun nilai resistansinya dapat diubah tergantung nilai maksimal dan nilai minimal resistansi dari *potensiometer* tersebut. Untuk kelas resistor yang kedua ini terdapat 2 tipe. Untuk tipe pertama dinamakan *variable* resistor dan nilainya dapat diubah sesuai keinginan dengan mudah dan sering digunakan untuk pengaturan *volume*, *bass*, *balance*, dll. Sedangkan yang kedua adalah *semi-fixed* resistor. Nilai dari resistor ini biasanya hanya diubah pada kondisi tertentu saja. Contoh penggunaan dari *semi-fixed* resistor adalah tegangan referensi yang digunakan untuk ADC, *fine tune circuit*, dll. Ada beberapa model pengaturan nilai *variable* resistor, yang sering digunakan adalah dengan cara memutar. Pengubahan nilai dengan cara memutar biasanya terbatas sampai 300 derajat putaran. Ada beberapa model *variable* resistor yang harus diputar berkali-kali untuk Contoh

penggunaan dari *semi-fixed* resistor adalah tegangan referensi yang digunakan untuk ADC, *fine tune circuit*, dll. Ada beberapa model pengaturan nilai *variable* resistor, yang sering digunakan adalah dengan cara memutar. Perubahan nilai dengan cara memutar biasanya terbatas sampai 300 derajat putaran. Ada beberapa model *variable* resistor yang harus diputar berkali-kali untuk mendapatkan semua nilai resistor. Model ini dinamakan “*Potentiometers*” atau “*Trimmer Potentiometers*”. Pada gambar 4 disamping untuk bentuk 3 biasanya digunakan untuk *volume* kontrol. Bentuk yang ke 2 merupakan *semi fixed* resistor dan biasanya di pasang pada PCB (*Printed Circuit Board*). Sedangkan bentuk 1 disebut juga *trimmer potentiometers*.

Ada 3 tipe didalam perubahan nilai dari resistor variabel, perubahan tersebut dapat dilihat pada gambar 2.16. Pada saat tipe A diputar searah jarum jam, awalnya perubahan nilai resistansi lambat tetapi ketika putarannya mencapai setengah atau lebih nilai perubahannya menjadi sangat cepat. Tipe ini sangat cocok dengan karakteristik telinga manusia. Karena telinga sangat peka ketika membedakan suara dengan *volume* yang lemah, tetapi tidak terlalu sensitif untuk membedakan perubahan suara yang keras. Biasanya tipe A ini juga disebut sebagai “*Audio Taper*” *potensiometer*. Untuk tipe B perubahan resistansinya adalah linier dan cocok digunakan untuk Aplikasi *Balance Control, resistance value adjustment in circuit*, dll. Sedangkan untuk tipe C perubahan resistansinya kebalikan dari tipe A. Biasanya tipe ini digunakan untuk fungsi-fungsi yang khusus. Kebanyakan untuk resistor variabel digunakan tipe A dan tipe B.



Gambar 2.12: Jenis *potensiometer* [2]

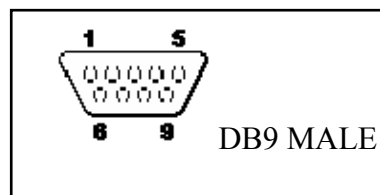


Gambar 2.13: Grafik resistansi *potensiometer* [2]

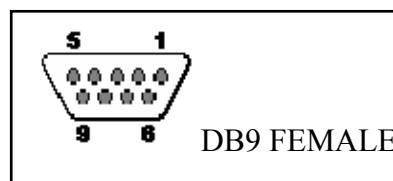
II.3.9. Komunikasi Serial RS232

Dalam tugas akhir ini akan membuat robot otomatis dengan menggunakan komunikasi serial. Pada bagian pemroses gambar dalam hal ini warna terdiri dari kamera, *USB (Universal Serial Bus)*, komputer, rangkaian pengubah level tegangan, sedangkan bagian pengaturan gerak robot terdiri dari rangkaian mikrokontroler dan rangkaian penggerak sebagai penggerak motor. Sedangkan untuk perancangan komunikasi data menggunakan *RS232*. [1].

Serial port bersifat asinkron dimana dapat mengirimkan data sebanyak 1 *bit* dalam tiap satu waktu. *Port* yang digunakan biasanya menggunakan konektor *DB9*. *DB9* mempunyai 9 *pin* yaitu:



Gambar 2.14: *DB9 Male* [1]



Gambar 2.15: *DB9 Female* [1]

keterangan:

- pin 1 = *Data Carrier Detect (DCD)*
- pin 2 = *Received Data (RxD)*
- pin 3 = *Transmitted Data (TxD)*
- pin 4 = *Data Terminal Ready (DTR)*
- pin 5 = *Signal Ground (common)*
- pin 6 = *Data Set Ready (DSR)*
- pin 7 = *Request To Send (RTS)*
- pin 8 = *Clear To Send (CTS)*
- pin 9 = *Ring Indicator (RI)*

Tabel 2.2 Keterangan *Pin DB9* [1]

Function	Signal	Pin	DTE	DCE
Data	TxD	3	O	I
	RxD	2	I	O
Handshake	RTS	7	O	I
	CTS	8	I	O
	DSR	6	I	O
	DCD	1	I	O
	DTR	4	O	I
Common	Com	5	-	-
Other	RI	9	I	O

Sinyal fungsi yang dijelaskan secara rinci dalam sinyal/*pin* primer, kolom yang ditandai *Dir* menunjukkan arah sinyal sehubungan dengan *DTE*. [1]

Pin No.	Name	Dir	Notes/Description
1	DCD	IN	Data Carrier Detect. Raised by DCE when modem synchronized.
2	RD	IN	Receive Data (a.k.a RxD, Rx). Arriving data from DCE.
3	TD	OUT	Transmit Data (a.k.a TxD, Tx). Sending data from DTE.
4	DTR	OUT	Data Terminal Ready. Raised by DTE when powered on. In auto-answer mode raised only when RI arrives from DCE.
5	SGND	-	Ground
6	DSR	IN	Data Set Ready. Raised by DCE to indicate ready.
7	RTS	OUT	Request To Send. Raised by DTE when it wishes to send. Expects CTS from DCE.
8	CTS	IN	Clear To Send. Raised by DCE in response to RTS from DTE.
9	RI	IN	Ring Indicator. Set when incoming ring detected - used for auto-answer application. DTE raised DTR to answer.

Gambar 2.16: Male Connector [1]

Berikut adalah pengaturan atau urutan kabel dari konektor DB9 jantan dan betina di lihat dari sisi pin nya.

Pins commonly used for RS-232/EIA-232 (serial):

DB-25	D-sub-9	Signal Direction	Signal Name
1		x	Protective Ground
2	3	DTE-to-DCE	Transmitted Data
3	2	DCE-to-DTE	Received Data
4	7	DTE-to-DCE	Request To Send
5	8	DCE-to-DTE	Clear To Send
6	6	DCE-to-DTE	Data Set Ready
7	5	x	Signal Ground
8	1	DCE-to-DTE	Received Line Signal Detector (Carrier Detect)
20	4	DTE-to-DCE	Data Terminal Ready
22	9	DCE-to-DTE	Ring Indicator

Gambar 2.17: Pins commonly used for RS-232 (serial) [1]

Komunikasi serial ialah pengiriman data secara *serial* (data dikirim satu persatu secara berurutan), sehingga komunikasi serial jauh lebih lambat daripada komunikasi paralel. *Serial port* lebih sulit ditangani karena peralatan yang dihubungkan ke *serial port* harus berkomunikasi dengan menggunakan transmisi *serial*, sedang data di komputer diolah secara *paralell*. [1].

Oleh karena itu data dari dan ke *serial port* harus dikonversikan ke dan dari bentuk *paralell* untuk bisa digunakan. Menggunakan *hardware*, hal ini bisa dilakukan oleh *Universal Asynchronous Receiver Transmimeter (UART)*, kelemahannya kita butuh *software* yang menangani *register UART* yang cukup rumit dibanding pada *parallel port*. Kelebihan dari komunikasi *serial* ialah panjang kabel jauh dibanding paralel, karena *serial port* mengirimkan logika “1” dengan kisaran tegangan -3 V hingga -25 V dan logika 0 sebagai $+3\text{ V}$ hingga $+25\text{ V}$ sehingga kehilangan daya karena panjangnya kabel bukan masalah utama. [1].

Bandingkan dengan *port paralel* yang menggunakan *level TTL* berkisar dari 0 V untuk logika 0 dan $+5\text{ V}$ untuk logika 1. Berikut contoh bentuk sinyal komunikasi serial . Umumnya sinyal serial diawali dengan *start bit*, data *bit* dan sebagai pengecekan data menggunakan *parity bit* serta ditutup dengan 2 *stop bit*. *Level* tengangan -3 V hingga $+3\text{ V}$ dianggap sebagai *undetermined region*. [1].

Komunikasi melalui *serial port* adalah asinkron, yakni sinyal detak tidak dikirim bersama dengan data. Setiap *word* disinkronkan dengan *start bit*, dan sebuah *clock* internal di kedua sisi menjaga bagian data saat pewaktuan (*timing*). [1].

Peralatan Komunikasi Serial

Hardware pada komunikasi *serial port* dibagi menjadi 2 (dua) kelompok yaitu Data *Communication Equipment (DCE)* dan Data *Terminal Equipment (DTE)*. Contoh dari *DCE* ialah *modem, plotter, scanner* dan lain lain sedangkan contoh dari *DTE* ialah terminal di komputer. Spesifikasi elektronik dari *serial port* merujuk pada *Electronic Industry Association (EIA)* :

“*Space*” (logika 0) ialah tegangan antara $+3$ hingga $+25\text{ V}$. “*Mark*” (logika 1) ialah tegangan antara -3 hingga -25 V . Daerah antara $+3\text{V}$ hingga -3V tidak

didefinisikan / tidak terpakai. Tegangan *open circuit* tidak boleh melebihi 25 V. Arus hubungan singkat tidak boleh melebihi 500mA.

Konektor *port serial* terdiri dari 2 jenis, yaitu konektor 25 *pin* (*DB25* dan 9 *pin* (*DB9*) yang berpasangan (jantan dan betina). Bentuk dari konektor *DB-25* sama persis dengan *port* paralel. [1].

Alamat Port dan IRQ

Alamat standar *serial port* adalah sebagaimana tampak dalam Tabel 12.2. Hal ini masih dengan catatan bahwa mungkin ada komputer yang memiliki alamat *port dan IRQ* yang berbeda dengan alarnat di bawah ini, misalnya pada komputer *PS/2* yang menggunakan *Micro Channel Bus*.

Konverter Logika RS-232

Jika peralatan yang kita gunakan menggunakan logika *TTL* maka sinyal serial *port* harus kita konversikan dahulu ke pulsa *TTL* sebelum kita gunakan, dan sebaliknya sinyal dari peralatan kita harus dikonversikan ke logika *RS-232* sebelum diinputkan ke *serial port*. Konverter yang paling mudah digunakan adalah *MAX-232*. Di dalam *IC* ini terdapat *Charge Pump* yang akan membangkitkan +10 Volt dan -10 Volt dari sumber +5 Volt tunggal. Dalam *IC DIP* (*Dual In-line Package*) 16 *pin* (8 *pin* x 2 baris) ini terdapat 2 buah *transmitter* dan 2 *receiver*. [1].

II.3.10. Notebook

Pada kendaraan inspeksi rel kereta api ini terdapat sebuah *netbook* yang berguna sebagai *receiver* gambar yang di tangkap dari *webcam* yang selanjutnya akan diteruskan ke komputer operator untuk memberikan informasi yang diperlukan menggunakan *wireless* secara *online*.



Gambar 2.18: *Notebook*

II.3.11. Modem

Modem merupakan suatu *hardware* tambahan yang akan terhubung dengan laptop sebagai perantara untuk dapat koneksi *via internet* secara online. *Modem* yang akan dipakai dalam pembuatan robot inspeksi rel kereta api adalah *modem* bentuk *USB* karena *modem* jenis tersebut sangat efektif dan efisien untuk kecepatan aksesnya juga baik.

II.4 BAGIAN PERANGKAT LUNAK

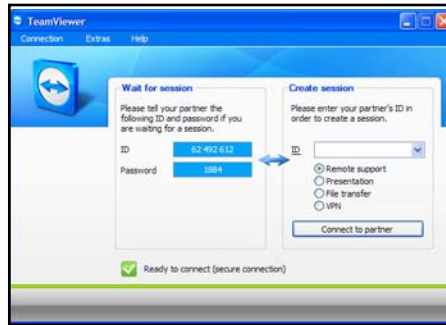
II.4.1. Software Team Viewer

Team Viewer adalah suatu program yang cukup sederhana dan sangat mudah digunakan untuk beberapa keperluan terutama melakukan akses *PC* secara *remote* melalui internet. *Team Viewer* saat ini sudah mencapai versi 5, merupakan aplikasi yang sangat cocok digunakan untuk mengakses *PC* lain melalui internet.[6],

Fitur utama *TeamViewer* adalah:

1. *Remote Support*
2. *Presentation*
3. *File Transfer*
4. *VPN*

Yang akan digunakan untuk tugas akhir ini adalah *Remote Support* (akses *PC* Klien melalui jalur internet), fitur / manfaat *teamviewer* yang satu ini sangat membantu jika kita perlu mengakses *PC* yang jauh dari posisi kita sekarang, karena cukup terkoneksi internet dan mengetahui *ID* serta *Password PC* tersebut dalam hitungan detik kita sudah bisa akses *PC* itu secara *remote* seperti kita ada didepan *PC* tersebut secara langsung. [8].



Gambar 2.19: Tampilan *Software Team Viewer* [8]

II.4.2. *Software Interface Mikrokontroler CodeVisionAVR Evaluation V2.03.9*

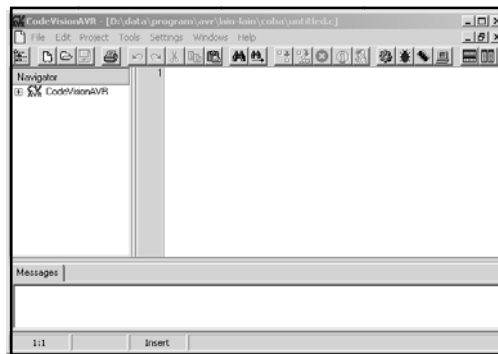
Pada dasarnya *CodeVisionAVR* merupakan perangkat lunak pemrograman mikrokontroler keluarga *AVR* berbasis bahasa *C*. Ada tiga komponen penting yang telah diintegrasikan dalam perangkat lunak ini: *Compiler C*, *IDE* dan *Program generator*. [9].

Berdasarkan spesifikasi yang dikeluarkan oleh perusahaan pengembangnya, *Compiler C* yang digunakan hampir mengimplementasikan semua komponen standar yang ada pada bahasa *C* standar *ANSI* (seperti struktur program, jenis tipe data, jenis operator, dan *library* fungsi standar-berikut penamaannya). Meskipun demikian, dibandingkan bahasa *C* untuk aplikasi komputer, *compiler C* untuk mikrokontroler ini memiliki sedikit perbedaan yang disesuaikan dengan arsitektur *AVR* tempat program *C* tersebut ditanamkan (*embedded*). [9].

Khusus untuk *library* fungsi, disamping *library* standar (seperti fungsi-fungsi matematik, manipulasi *String*, pengaksesan memori dan sebagainya), *CodeVision AVR* juga menyediakan fungsi-fungsi tambahan yang sangat bermanfaat dalam pemrograman antarmuka *AVR* dengan perangkat. Beberapa fungsi *library* yang penting diantaranya adalah fungsi-fungsi untuk pengaksesan *LCD*, komunikasi *I2C*, *IC RTC (Real time Clock)*, sensor suhu *LM75*, *SPI (Serial Peripheral Interface)* dan lain sebagainya. [9].

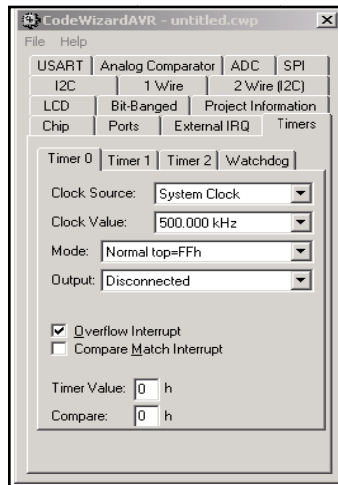
Untuk memudahkan pengembangan program aplikasi, *CodeVision AVR* juga dilengkapi *IDE* yang sangat *user friendly* (lihat gambar 2.24). Selain menu-menu pilihan yang umum dijumpai pada setiap perangkat lunak berbasis *Windows*,

CodeVision AVR ini telah mengintegrasikan perangkat lunak *downloader (in system programmer)* yang dapat digunakan untuk mentransfer kode mesin hasil kompilasi ke dalam sistem memori mikrokontroler AVR yang sedang deprogram. [9].



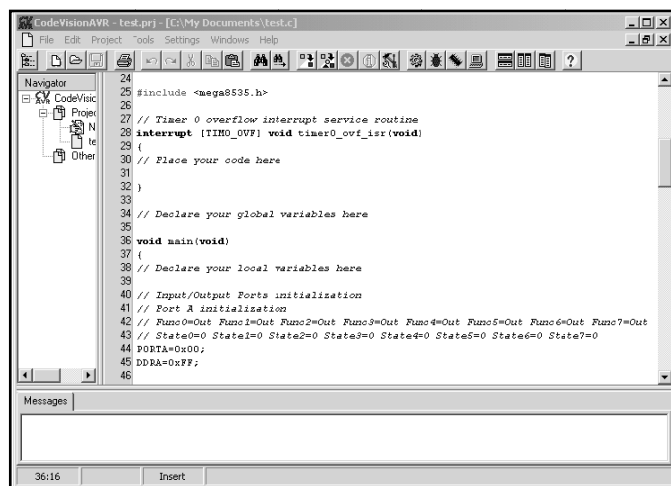
Gambar 2.20: IDE perangkat lunak *CodeVisionAVR* [9]

Selain itu, *CodeVision AVR* juga menyediakan sebuah *tool* yang dinamakan dengan *Code Generator* atau *Code Wizard AVR* (lihat gambar 2.23). Secara praktis, *tool* ini sangat bermanfaat membentuk sebuah kerangka program (*template*), dan juga memberi kemudahan bagi *programmer* dalam penginisialisasian *register-register* yang terdapat pada mikrokontroler AVR yang sedang diprogram. Dinamakan *Code Generator*, karena perangkat lunak *CodeVision* ini akan membangkitkan kode-kode program secara otomatis setelah fase inisialisasi pada jendela *CodeWizard AVR* selesai dilakukan. Gambar 2.24 berikut memperlihatkan beberapa penggal baris kode program yang dibangkitkan secara otomatis oleh *CodeWizard AVR*. Secara teknis, penggunaan *tool* ini pada dasarnya hampir sama dengan aplikasi *wizard* pada bahasa-bahasa pemrograman *Visual* untuk komputer (seperti *Visual C*, *Borland Delphi*, dan sebagainya). [9].



Gambar 2.21: *Code Generator* yang dapat digunakan untuk menginisialisasi *register-register* pada mikrokontroler AVR. [9]

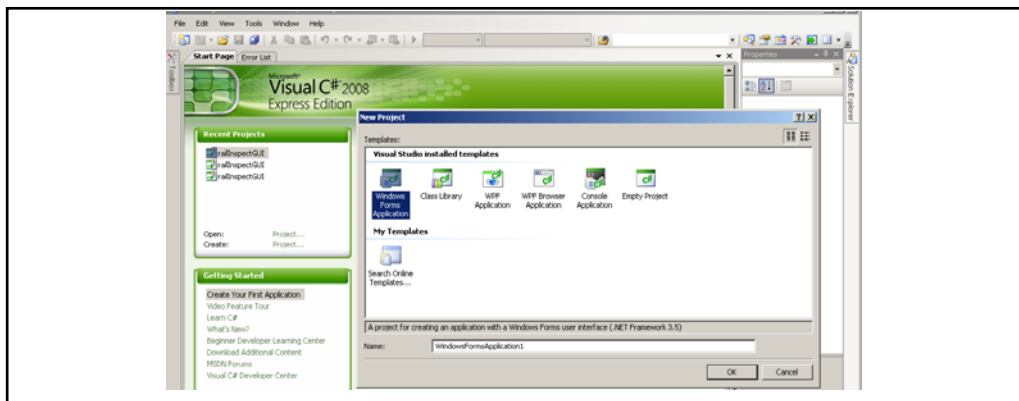
Disamping versi yang komersil, Perusahaan Pavel Haiduc juga mengeluarkan *CodeVision AVR* versi Demo yang dapat didownload dari internet secara gratis (lihat alamat *URL*: <http://www.hpinfo.tech.ro>) Dalam versi ini, memori *flash* yang dapat diprogram dibatasi maksimal 2K, selain itu tidak semua fungsi *library* yang tersedia dapat dipanggil secara bebas. [9].



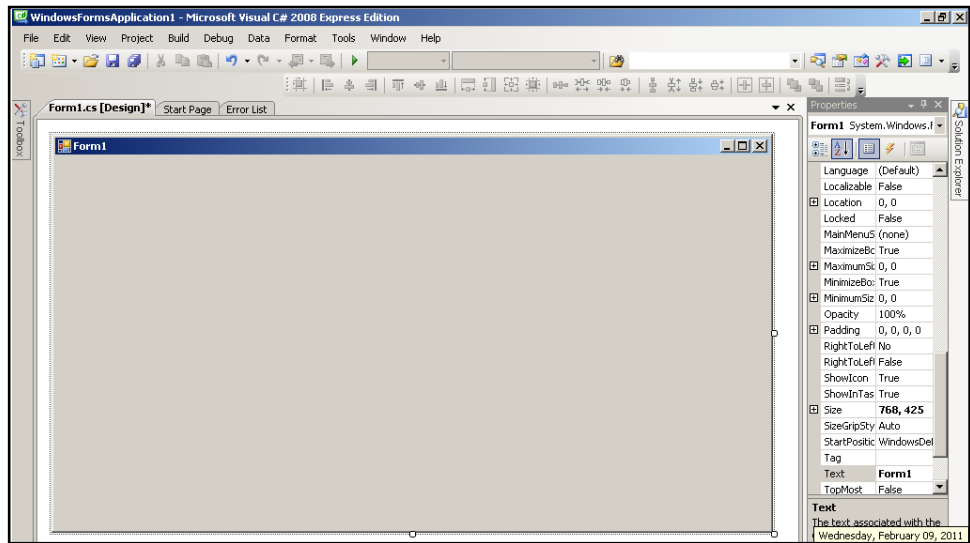
Gambar 2.22: Kode-kode program yang dibangkitkan otomatis oleh *code generator* [9]

II.4.3. Software Visual Studio 2008

Software visual studio merupakan keluaran *Microsoft*, dengan menggunakan *visual C#* dimana bahasa pemrogramannya menggunakan bahasa *C#*. Pada tugas akhir ini *software visual C#* digunakan untuk melaporkan dan menerima data dari mikrokontroler, dengan menggunakan *interface GUI* sebagai media komunikasi *serial RS232* untuk transfer data ke mikrokontroler *AVR*. Sebuah form menggunakan *Visual C# 2008*. Inti dari proyek ini masih dasar dan bisa dikembangkan menjadi program aplikasi yang bermanfaat. Komunikasi antara *PC* dengan *device* luar menggunakan *serial port* yang sangat berguna untuk aplikasi di bidang elektronika, baik itu sistem *control*, pengukuran, dan sebagainya. Pada *visual C#* sudah terdapat komponen *serial port*. Berikut tampilan dari *software visual studio 2008*. [3].



Gambar 2.23: Tampilan *Visual C# 2008* [3]

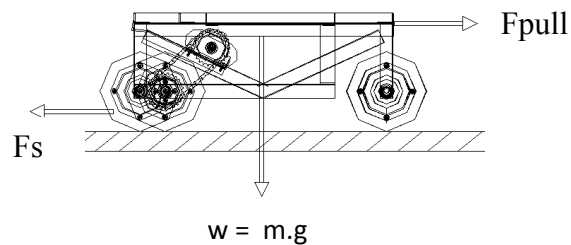


Gambar 2.24: Design form Visual C# 2008 [3]

II.5. TEORI DASAR PERHITUNGAN MEKANIK ROBOT

II.5.1 Penentuan Motor Penggerak

Perancangan robot inspeksi rel ini menggunakan satu motor sebagai penggerak. Tahapan pertama memeriksa motor *drive*. Motor ini harus mampu mendorong empat roda bersama-sama dan menghasilkan gerakan. Untuk menentukan tenaga yang diperlukan dari motor, perancang menggunakan bantuan dengan diagram benda bebas untuk gerak robot serta gaya-gaya yang terjadi, ditunjukkan pada gambar 2.27 [11].



Gambar 2.25: Diagram benda bebas *simple* gerak robot

Robot dengan *massa* m ditarik kearah *horizontal*, kekuatan yang dibutuhkan untuk melakukan ini disebut $F_{pull} = \sum F = 0$

$$F - F_s = 0$$

$$F = F_s$$

$$F = \mu \cdot m \cdot g$$

$$\text{Dimana, } \mu = C_{TT}$$

$$= \sqrt{z/d} \quad (2.1)$$

Dimana, $z = \text{sinkage depth}$

$d = \text{diameter roda}$

Maka daya yang dibutuhkan,

$$P = F_{pull} \cdot v \quad (2.2)$$

Dimana :

F_{pull} = Gaya tarik (*Newton*)

v = Kecepatan (*m/s*)

Karena menggunakan reduksi gear maka ; rumus perhitungan reduksi gear adalah:

$$N_2 = \frac{\text{jumlah gigi 1 (z1)}}{\text{jumlah gigi 2 (z2)}} \times N_1 \quad (2.3)$$

Jadi untuk rumus perhitungan torsi motor adalah:

$$T = \frac{P}{\omega} \quad (2.4)$$

Dimana :

P = Daya tarik

ω = Kecepatan Sudut

II.5.2 PERHITUNGAN BATERAI

II.5.2.1 Perhitungan Energi baterai

Energi yang dihitung adalah Energi yang dihasilkan oleh baterai (E), di mana Energi tersebut dapat dicari dengan rumus.[11] :

$$W = V I . t \quad (2.5)$$

Di mana :

V = Voltase (*Volt*)

I = Arus (*Ampere*)

t = waktu baterai habis (*second*)

II.5.2.2 Waktu Pemakaian Baterai ($T_{\text{pemakaian}}$)

Perhitungan waktu pemakaian baterai ini bertujuan untuk mencari berapa lama pemakaian baterai aki dalam kondisi baterai penuh. Waktu pemakaian baterai dapat dicari dengan rumus. [10] :

$$T_{\text{pemakaian}} = \frac{\text{Energi baterai}}{\text{daya motor}} \quad (2.6)$$

II.5.2.3 Jarak Tempuh Yang Dihasilkan (S)

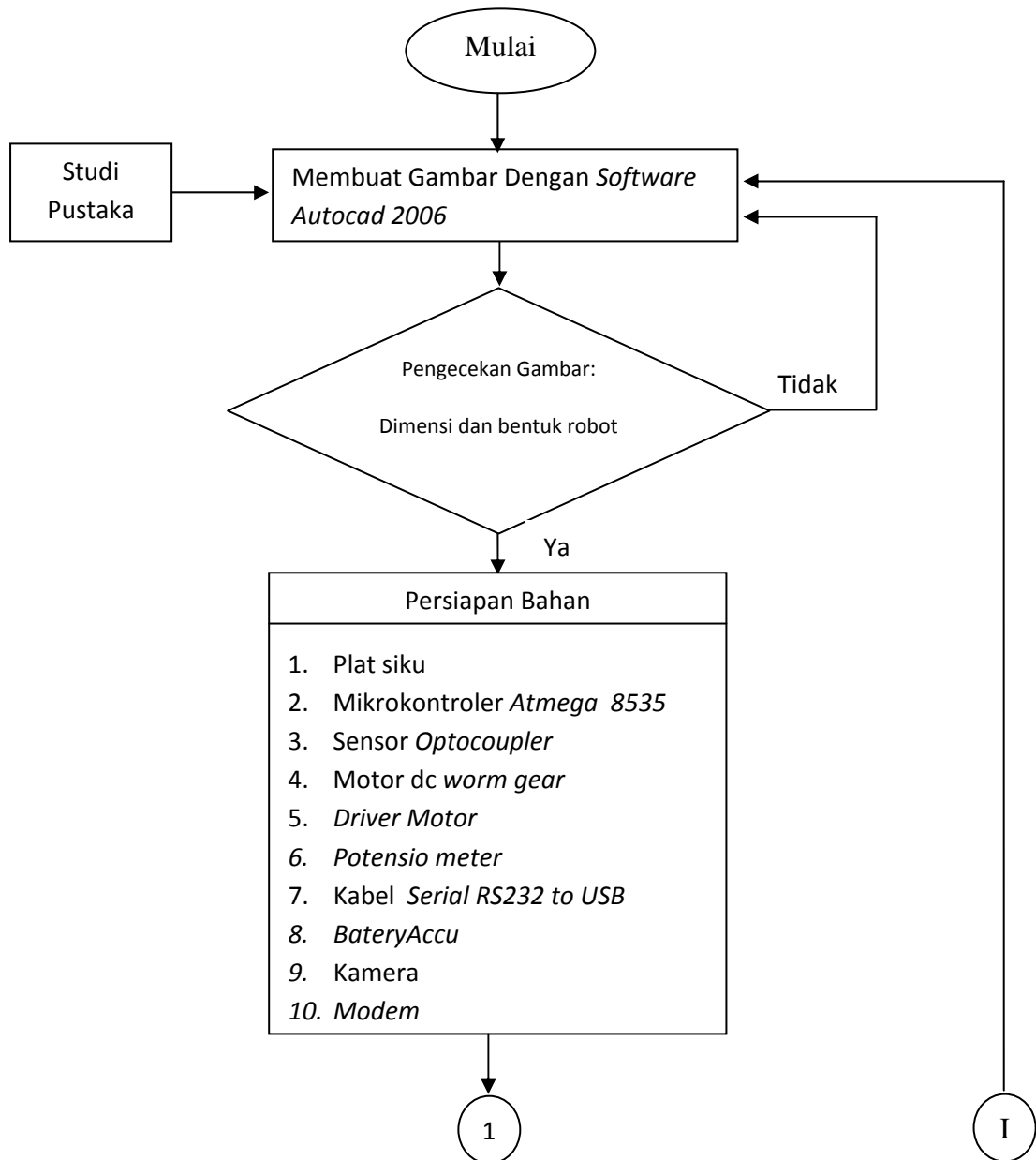
Di sini kita dapat mengetahui berapa jauh jarak tempuh dari robot dalam kondisi baterai terisi penuh. Jarak tersebut dapat dicari dengan rumus :

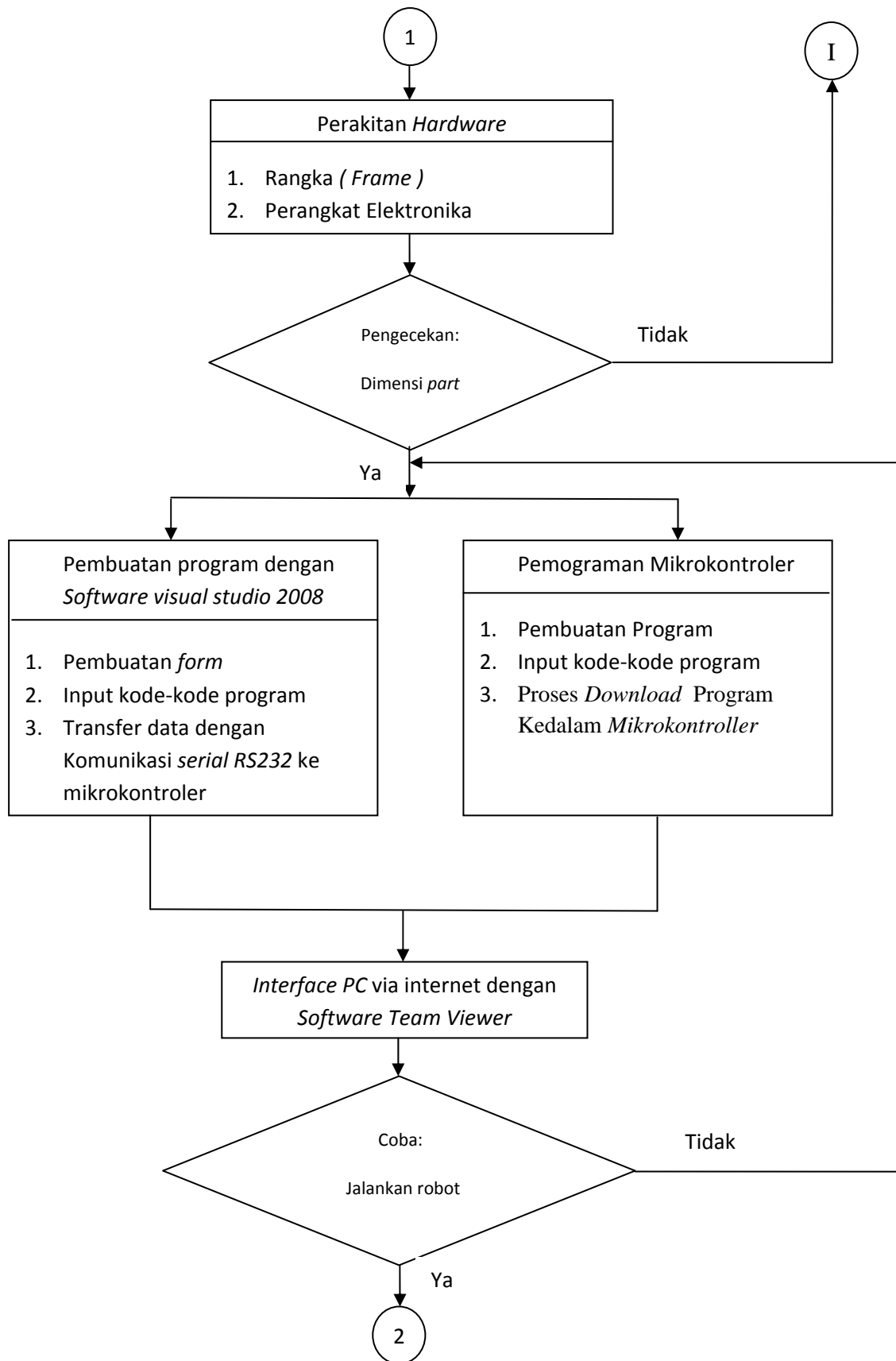
$$S = v \times T_{\text{pemakaian}} \quad (2.7)$$

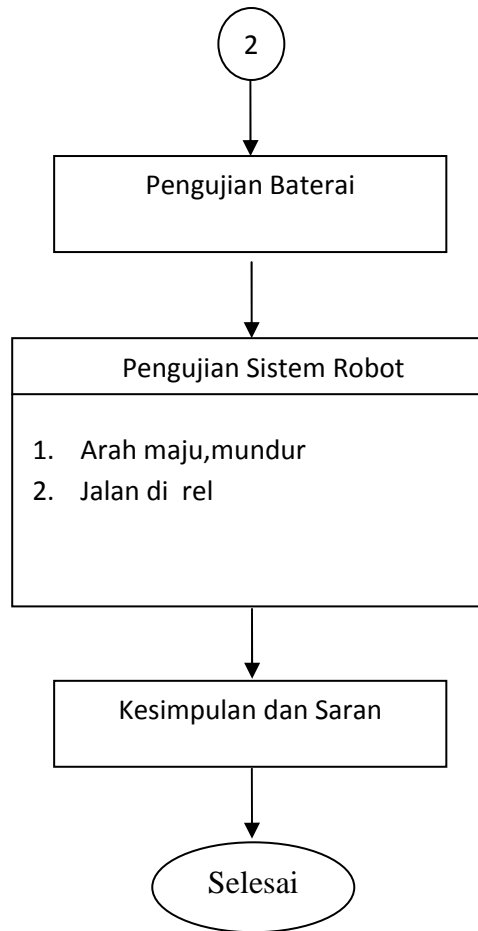
BAB III

PERANCANGAN DAN PEMBUATAN PROTOTIPE ROBOT

Realisasi dalam membangun suatu perangkat dinamis dalam hal ini adalah sebuah robot di perlukan suatu tahapan-tahapan dalam proses pembuatannya hingga menghasilkan model jadi. Model atau prototipe robot yang dibuat pada dasarnya terdiri dari dua aspek utama, yaitu aspek perangkat keras (*Hardware*) dan aspek perangkat lunak (*Software*).







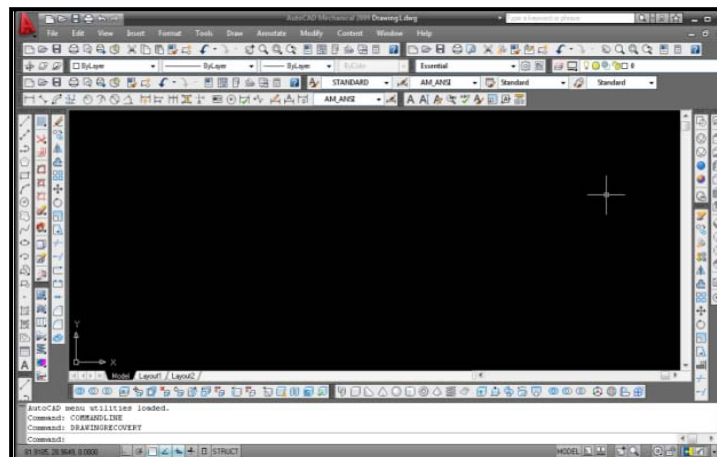
Gambar 3.1: Diagram alir perancangan dan pembuatan prototipe robot

III.1. KONSEP ROBOT

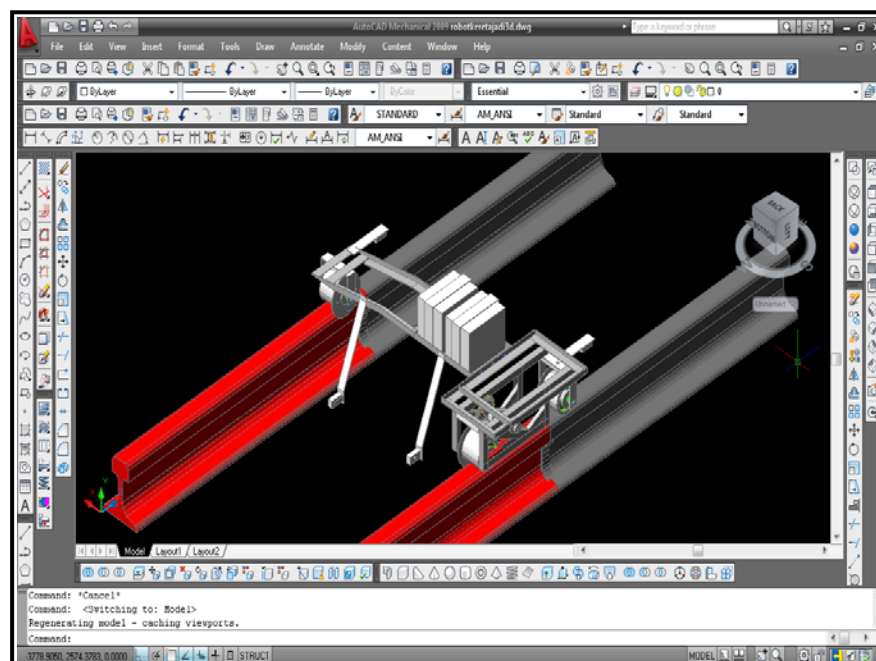
Sebelum merancang dan mengembangkan suatu robot baru, perancang harus mempunyai konsep robot. Pada tugas akhir ini membahas tentang pembuatan robot inspeksi rel kereta api, robot ini akan digunakan untuk *memonitoring* kondisi rel, sambungan plat, baut rel, dan ketinggian rel kanan dan kiri. Kegiatan merancang robot inspeksi rel ini diawali dengan konsep bentuk robot yang pada akhirnya diharapkan dapat berfungsi sebagaimana mestinya. Konsep robot ini dibuat dengan menggunakan alat bantu gambar *Software Auto Cad 2006*.

III.1.1. Perancangan *Assembling Part Inpection Robot Menggunakan Software Autocad 2006*

Software Autocad 2006 merupakan *software design CAD 3D*, *Autocad 2006* digunakan untuk merancang desain *part* yang akan digunakan robot sebelum robot dibuat.



Gambar 3.2: Jendela *software AutoCad 2006*



Gambar 3.3: Konsep Desain Robot Inspeksi Rel

Desain ini dibuat dengan tampilan 3 dimensi sehingga dapat mengetahui bentuk robot sebelum robot direalisasikan.

III.1.2. Dimensi Umum Dan Bentuk Robot

Ketika konsep robot yang diinginkan telah sesuai, tahapan selanjutnya menentukan dimensi/ukuran robot yang akan dibuat. Adapun dimensi/ukuran robot inspeksi rel adalah:

Tabel 3.1: Dimensi umum robot

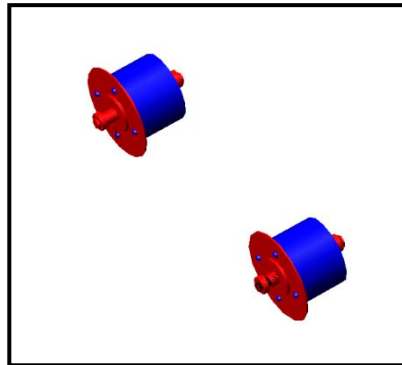
Panjang robot	247 cm
Lebar robot	78 cm
Tinggi robot	23 cm
Berat konstruksi rangka robot	6 kg
Berat total	15 kg

Bentuk robot inspeksi rel sangat penting dapat berdampak pada kinerja robot. Misalnya: bentuk robot persegi memiliki resiko lebih besar untuk terjebak oleh sebuah rintangan atau gagal untuk menemukan jalan melalui sebuah ruang sempit. Sedangkan bentuk robot silinder bisa menggunakan algoritma sederhana untuk menemukan jalan melalui lorong sempit karena mampu memutar didepan kendala tanpa terjebak. Robot persegi harus kembali terlebih dahulu bahkan memutar. Karena kita ingin menggunakan robot untuk tugas yang berbeda dan robot akan berjalan tanpa adanya rintangan, maka sebuah robot berbentuk persegi merupakan solusi terbaik untuk bentuk robot inspeksi rel.

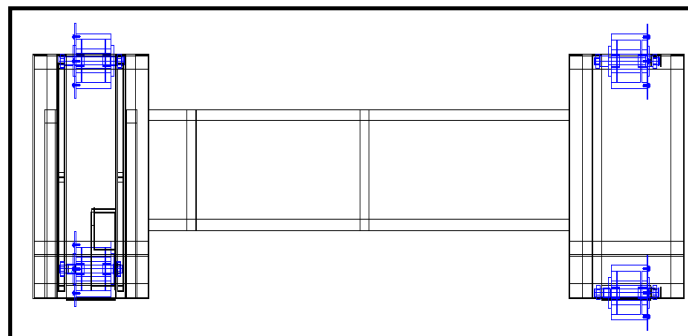
III.1.3. Konfigurasi Roda Pada Robot

Tahapan berikutnya dalam proses desain adalah pemilihan konfigurasi roda dan jenis roda yang digunakan dalam konfigurasi roda. perancang memutuskan untuk mengambil *drive* empat roda dengan mengarahkan roda standar. Roda standar dipilih karena kesederhanaan bentuk dan proses pembuatan tidak terlalu rumit. *Drive* empat roda memiliki beberapa manfaat sehubungan dengan konfigurasi roda lainnya dalam

aplikasi ini. Manfaat utama hanya satu motor yang diperlukan, satu motor roda berputar semua bersama-sama untuk menghasilkan gerakan. Suspensi Tidak perlu, konfigurasi roda empat masing-masing roda kontak langsung dengan tanah, Jaminan mekanik gerak garis lurus.



Gambar 3.4: Bentuk Roda robot inspeksi rel kereta api



Gambar 3.5: Konfigurasi roda pada robot inspeksi rel

III.2. PERANGKAT KERAS ROBOT

III.2.1. Kerangka, Roda, Baterai dan Motor Penggerak

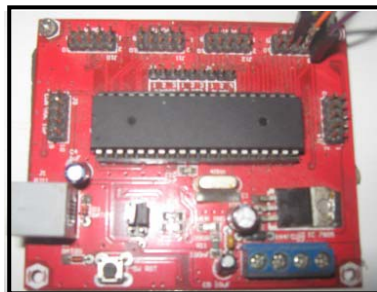
Kerangka dibuat dari bahan plat besi, dengan ukuran yang telah ditentukan dan disesuaikan dengan kebutuhan untuk menampung komponen yang lain seperti *notebook*, *modem*, mikrokontroler, baterai, kamera serta komponen pendukung lain. Mengenai pembuatan kerangka, tidak dipersyaratkan secara khusus dan detail. Kesederhanan akan tetap dipertahankan tanpa mengurangi fungsi, mengutamakan fleksibilitas pembuatan perangkat yang masih berupa model atau prototipe.

Roda merupakan bagian aktuator untuk melakukan manufer berupa gerak robot berupa maju, mundur, balik otomatis dan berhenti. Model atau prototipe robot inspeksi rel ini dipasang empat buah roda, satu pasang roda depan dan satu pasang roda belakang yang ukurannya lebih besar dibandingkan roda bagian depan. bagian tengah busur lingkaran roda dipasang *bearing*, agar roda tidak selip saat berputar, dimana untuk penggerak robot digunakan 1 buah motor *worm gear* 12 V. Sebagai sumber dayanya dipakai baterai dengan kapasitas 5 Ah.

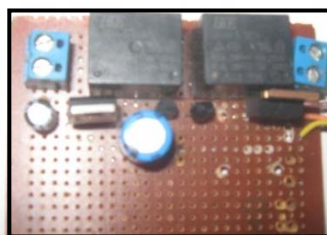
III.2.2. Mikrokontroler

Seperti yang telah disebutkan pada bab sebelumnya, untuk kontrol sistem penggerak digunakan mikrokontroler dari *Atmel* dengan tipe *AT8535*.

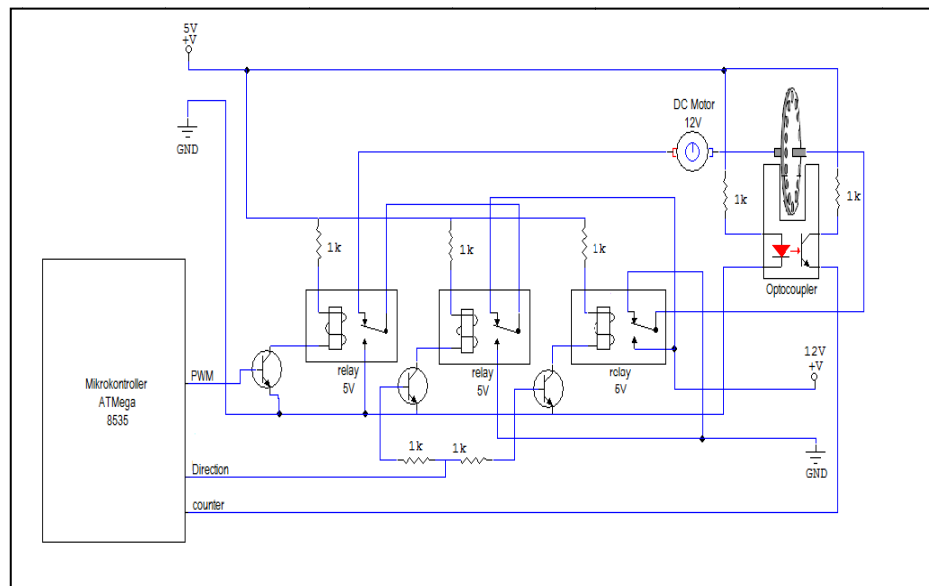
Pada sistem pengerak robot kali ini menggunakan *driver* motor dan dioda–dioda untuk melindungi sirkuit dari keluaran arus balik. *Relay* ini dapat mengontrol motor DC secara langsung untuk berputar searah atau berlawanan jarum jam, dengan *input* yang dihasilkan dari mikrokontroler *AT8535*.



Gambar 3.6: Mikrokontroler *AT8535*



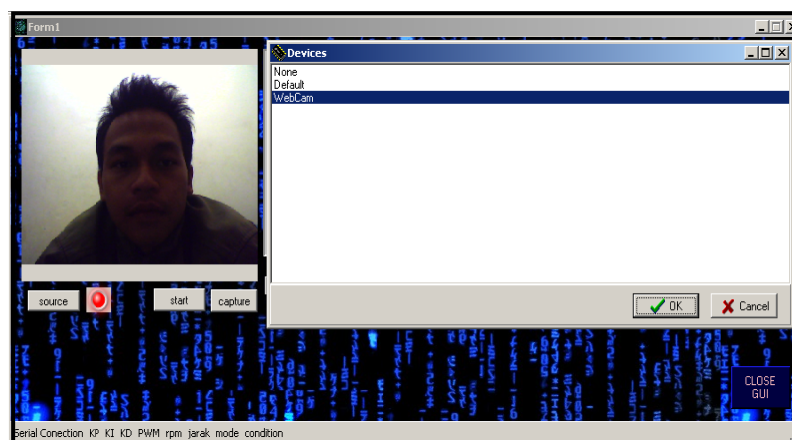
Gambar 3.7: *Driver* Motor



Gambar 3.8: Kendali Motor robot inspeksi rel kereta api dengan AT8535

III.2.3. Kamera

Pada model atau prototipe robot inspeksi rel kereta api ini kamera digunakan untuk melihat berbagai kondisi antara lain: permukaan rel dan baut pada sambungan plat. Kamera yang digunakan merk *Logitech Quickcam® Cool* berjumlah 4 buah dengan posisi *shoot* gambar yang sudah ditentukan. Akses kamera adalah *notebook* yang kemudian diintegrasikan dengan *software visual studio 2008*. Kamera diletakkan dibadan robot dengan posisi dudukan yang sudah dirancang.



Gambar 3.9: Tampilan *form* kamera dengan *Visual C#*

III.3.PERAKITAN ROBOT

Setelah semua komponen selesai dibuat dan lengkap, maka selanjutnya perangkat dinamis siap untuk dirakit. Langkah – langkah perakitan perangkat adalah sebagai berikut :

1. Semua komponen disiapkan, pertama memasang motor dc pada rangka. Karena motor yang dipasang, tidak di spesifikasikan secara khusus, hanya diutamakan menggunakan tegangan 12V dan dilengkapi *gear* untuk menguatkan torsi dengan transmisi menggunakan rantai.
2. Memasang roda belakang dan roda depan serta komponen pendukungnya.
3. Memasang baterai dibagian tengah rangka.
4. Memasang mikrokontroler, sensor *optocoupler* dan *driver* motor dc
5. Memasang laptop dibagian dudukan atas rangka.
6. Memasang kamera pada bagian depan dan samping rangka robot.
7. *Install software team viewer* kemudian setting ip *address pc server* dan klien *open form exe software visual studio*.
8. Setelah semua sudah terpasang maka robot siap untuk instalasi selanjutnya.

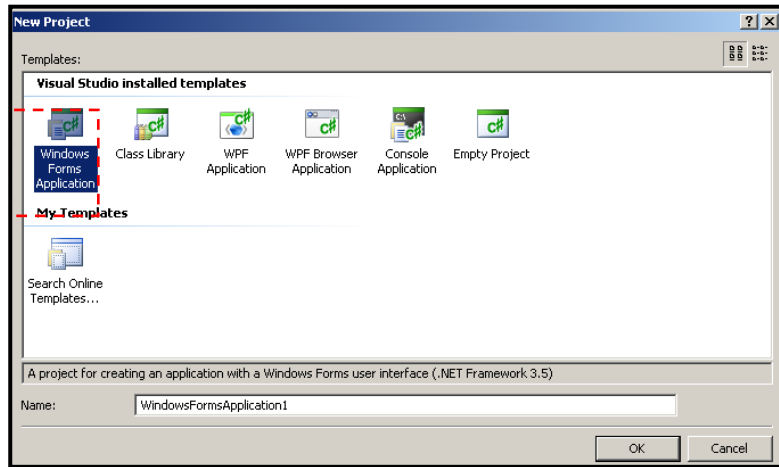
III.4. PERANGKAT LUNAK

III.4.1. Pemograman Dengan *Software Visual Studio 2008*

Pada tugas akhir ini *software visual C# 2008 express edition* digunakan sebagai indikator dan navigasi arah pada robot melalui *GUI (Graphical User Interface)* serta melakukan transfer data ke mikrokontroler . Berikut langkah-langkah pembuatan *form GUI software Visual C# 2008*:

1. Pembuatan kamera

- A. Jalankan *Visual C# 2008*, lalu klik *File → New Project*.
- B. Pada jendela *New Project*, pilih *Windows Forms Application*. Beri nama *capture_webcam*, lalu klik OK.



Gambar 3.10: Tampilan *New Project*

C. Pada *Form1*, tambahkan komponen berikut:

- 1 x *DSCapture*
- 1 x *DSImageDisplay*
- 1 x *DsVideoLogger*
- 1 x *GenericFilter*
- 1 x *Snapshot*
- 1 x *GroupBox*
- 3 x *Button*
- 1 x *PictureBox*
- 1 x *led*

D. Atur posisi komponen yang ditambahkan tadi sehingga terlihat seperti pada gambar dibawah ini.



Gambar 3.11: Posisi komponen yang ditambahkan pada *form*

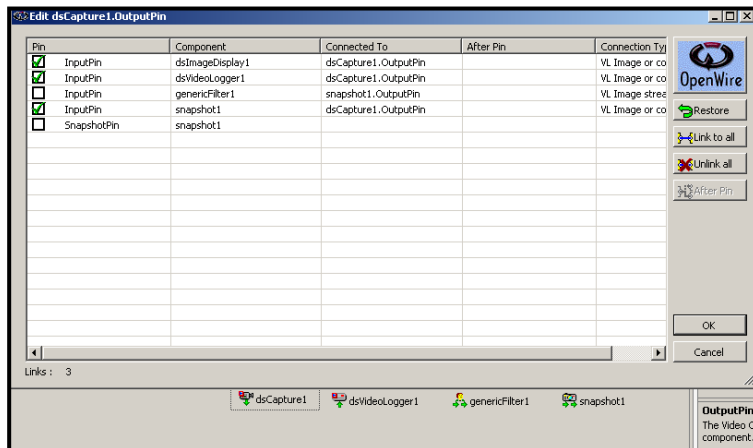
E. Atur *properti* komponen yang ditambahkan seperti pada tabel 3.2 berikut.

Tabel 3.2 *Properti* komponen pembuatan kamera dengan *visual studio 2008*

Komponen	<i>Properti</i>	Nilai <i>Properti</i>
<i>Form1</i>	<i>Text</i>	<i>Navigasi robot inspeksi</i>
<i>Button1</i>	<i>Text</i>	<i>Source</i>
<i>Button2</i>	<i>Text</i>	<i>Start</i>
	<i>Enable</i>	<i>False</i>
<i>Button3</i>	<i>Text</i>	<i>Capture Bitmap</i>
	<i>Enable</i>	<i>False</i>
<i>DsImageDisplay1</i>	<i>Anchor</i>	<i>Top, Bottom, Left, Right</i>
<i>DsCapture1</i>	<i>Enable</i>	<i>False</i>
<i>PictureBox1</i>	<i>Visible</i>	<i>False</i>

F. Pada *Dscapture1*, pilih *Outputpin*, lalu klik tombol 

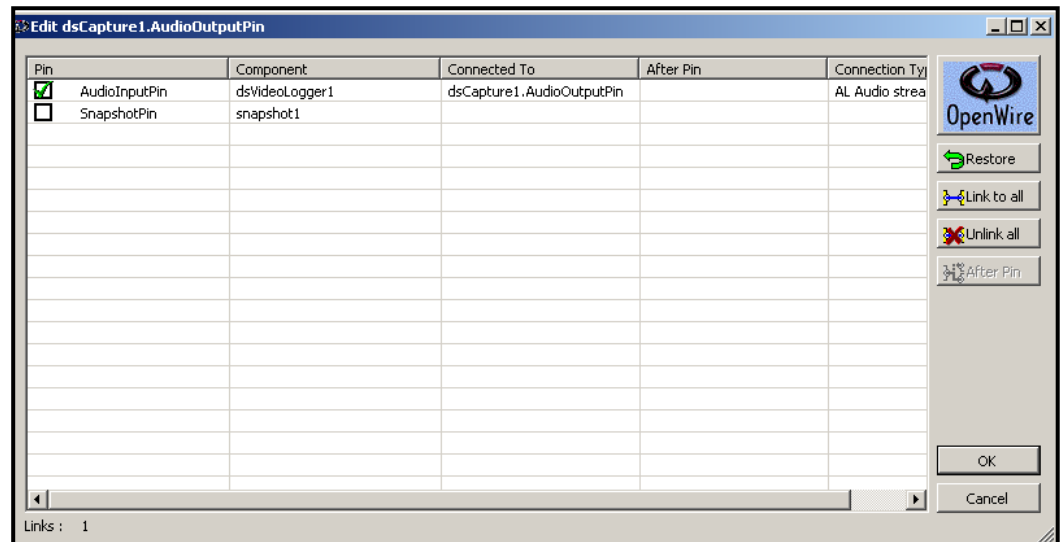
G. Pada *Connection editor* pilih *InputPin* dari *dsImageDisplay1*, *dsVideoLogger1*, dan *Snapshot1*, lalu klik OK.




Gambar 3.12: *Outputpin Connection editor*

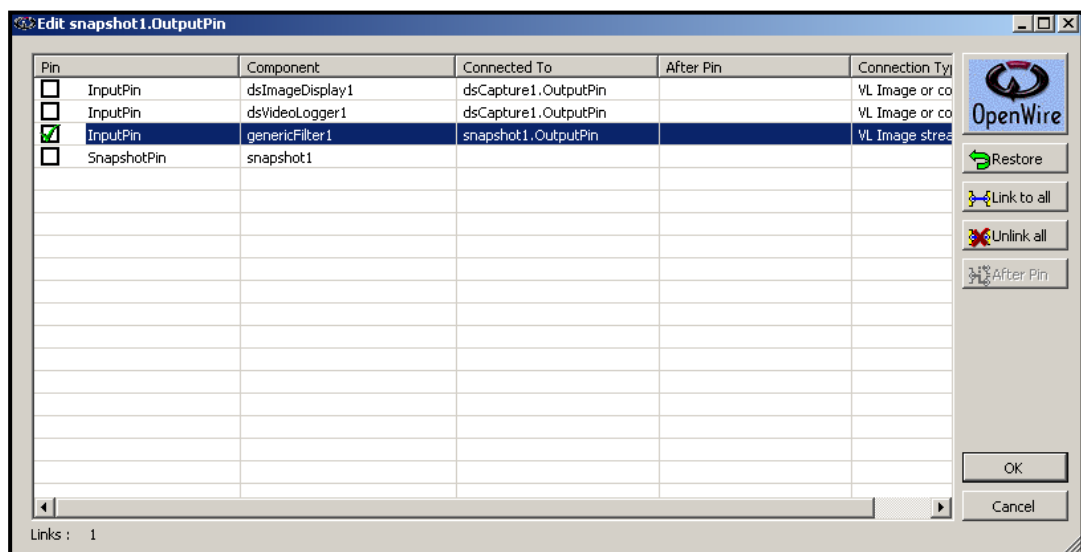
H. Pada *DsCapture1*, pilih *AudioOutputpin*, lalu klik tombol 

- I. Pada *form Edit Dscapture1.AudioOutputPin*, pilih *AudioInputPin* dari *dsVideoLogger1*, lalu klik OK.



Gambar 3.13: *Audio Outputpin dsVideoLogger1*

- J. Pada *snapShot1*, pilih *OutputPin*, lalu klik tombol 
- K. Pada *form edit snapshot1.OutputPin*, pilih *InputPin* dari *genericFilter1*, lalu klik OK.



Gambar 3.14: *OutputPin SnapShot*

L. Klik tombol *Source* pada *webcam*, tuliskan program berikut:

```
private void button1_Click(object sender, EventArgs e)
{
    dsCapture1.VideoCaptureDevice.ShowDeviceSelctDialog();
    if ((dsCapture1.VideoCaptureDevice.DeviceName == "None")
        (dsCapture1.VideoCaptureDevice.DeviceName == ""))
    {
        dsCapture1.Enabled = false;
        button2.Enabled = false;
        button3.Enabled = false;
        led2.Value = false;
    }
    else
    {
        dsCapture1.Enabled = true;
        button2.Enabled = true;
        button3.Enabled = true;
        led2.Value = true;
    }
}
```

Program diatas digunakan untuk menampilkan dialog untuk memilih *source device* yang akan digunakan, yaitu pada fungsi *dsCapture1.VideoCaptureDevice.showdeviceSelctDialog();*.

Klik tombol *Start* pada *webcam 1*. Tuliskan program di bawah ini:


```
private void button2_Click(object sender, EventArgs e)
{
    if (button2.Text == "start")
    {
        dsCapture1.Enabled = true;
        dsCapture1.Start();
        button2.Text = "stop";
        button1.Enabled = false;
        led2.Value = true;
    }
    else
    {
        dsCapture1.Stop();
        dsCapture1.Enabled = false;
        dsCapture1.VideoCaptureDevice.DeviceName = "";
        button2.Text = "start";
        button1.Enabled = true;
        button2.Enabled = false;
        button3.Enabled = false;
        led2.Value = false;
    }
}
```

Untuk memulai *capture* digunakan fungsi *dsCapture1.start();*, Untuk menghentikannya digunakan fungsi *dsCapture1.stop();*.

M. Klik tombol *Capture* pada *webcam*, tuliskan program dibawah ini:

```
private void button3_Click(object sender, EventArgs e)
{
    snapshot1.TakeSnapshot();
}
```

Fungsi `snapshot1.TakeSnapshot();` digunakan untuk mengambil *frame video* yang diteruskan ke *genericFilter* yang kemudian diolah menjadi gambar *file.bmp*.

N. Klik *genericFilter1* →  klik ganda *ProcessData* dan tuliskan program di bawah ini:

```
private void genericFilter1_ProcessData(object Sender,
Mitov.VideoLab.VideoProcessDataEventArgs Args)
{
    System.Drawing.Bitmap ABitmap1 =
    Args.InBuffer.ToBitmap();
    pictureBox1.Image = ABitmap1;
    pictureBox1.Image.Save("d:/picture1.bmp");

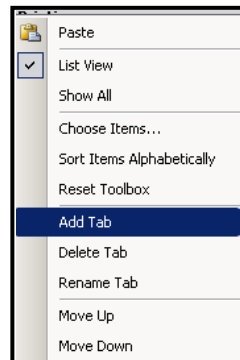
    MessageBox.Show("Picture1.bmp create");
}
```

Fungsi `System.Drawing.Bitmap ABitmap1=Args.InBuffer.ToBitmap();` digunakan untuk memproses *frame video* dari *snapshot* ke *format.bmp*. Data *stream file.bmp* disimpan pada variabel *ABitmap1*. *ABitmap1* disimpan dalam bentuk *file* melalui `PictureBox.image.save ("D:/hasil_inspeksi/picture1.bmp");`. *Folder* penyimpanan bisa diganti dengan *folder* lain.

2. Pembuatan indikator kerataan rel

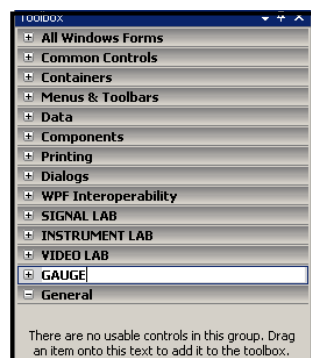
Indikator ini memanfaatkan *signal analog* dari *potensiometer* yang terpasang di badan robot kemudian disambungkan pada mikrokontroler. Berikut langkah pembuatannya:

- A. *Install software matlab 2008b*
- B. *Jalankan Software visual c#*
- C. *Tambahkan tool gauge matlab ke visual C#*
- D. Pada *toolbox*, pilih *General* → klik kanan → *Add Tab*, seperti terlihat pada gambar di bawah ini.



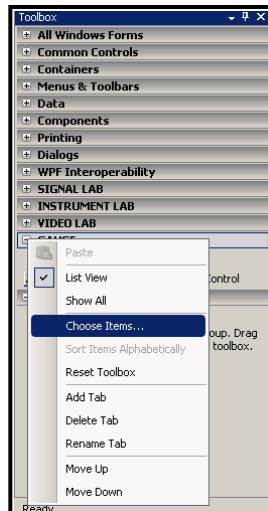
Gambar 3.15: Menambahkan *matlab* ke visual C#

E. Beri nama *tab* baru sebagai *gauge*, seperti pada gambar di bawah ini



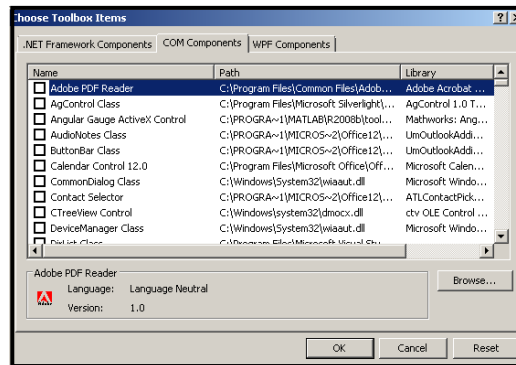
Gambar 3.16: *Tab* baru dengan nama *gauge*

F. Agar *items* pada *gauge* keluar pilih *gauge* → klik kanan → *choose items*, seperti pada gambar di bawah ini.



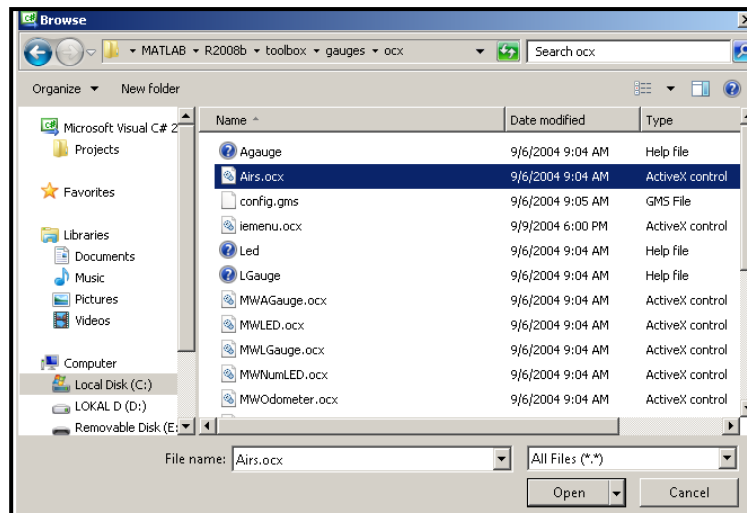
Gambar 3.17: Penambahan *Choose Items gauge*

G. Akan muncul *form choose toolbox items*, seperti yang terlihat pada gambar dibawah ini. Klik *browse*.



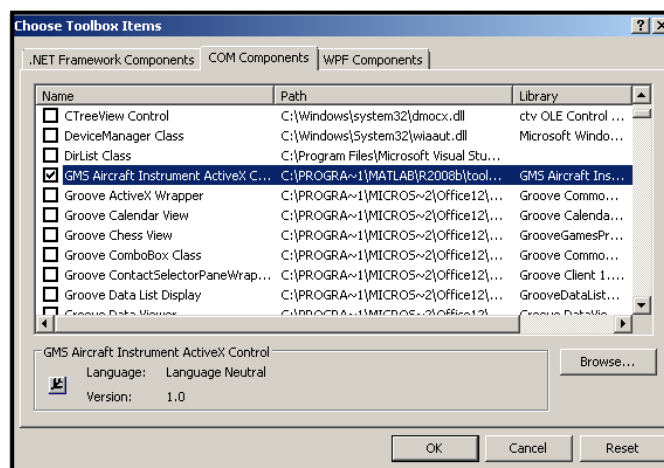
Gambar 3.18: *Form choose toolbox items*

H. Muncul kotak *dialog Open*. Arahkan ke *folder bin* tempat *gauge* di *install*. Pada proyek ini, *gauge* di *install* ke *folder C:\Program files\matlab\R2008B\toolbox\gauges\ocx*.



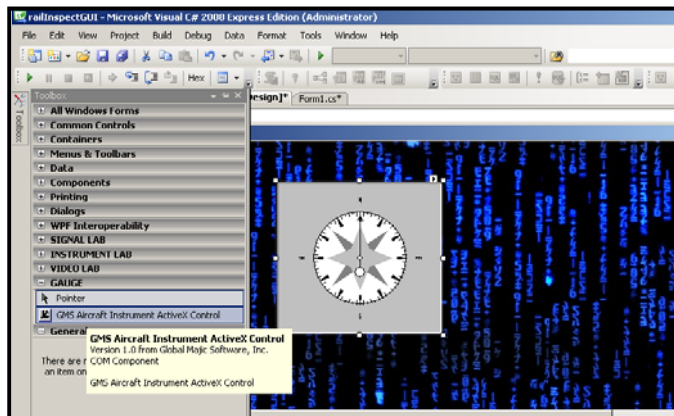
Gambar 3.19: *Browse dari choose toolbox items*

- I. Pada *form open* diatas, pilih *Airs.dll*, lalu tekan *open*. Anda akan kembali ke *form choose toolbox items*.



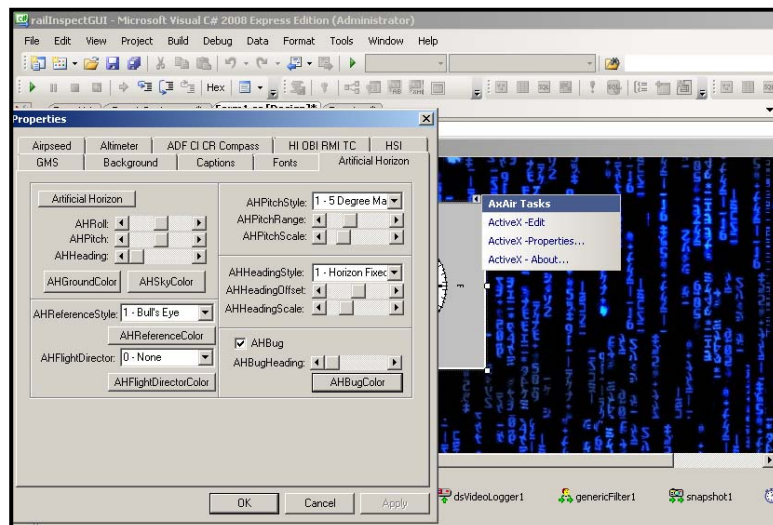
Gambar 3.20: Centang *GMS Aircraft Instrument*

- J. Kembali ke *form.Cs [design]* kemudian ke *toolbox* → pilih *gauge* → *aircraft*.



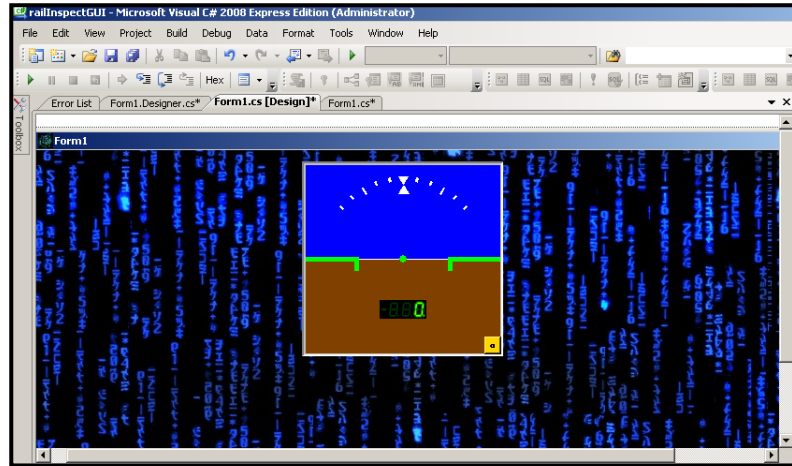
Gambar 3.21: Tampilan *design form aircraft*

K. Setelah *form* terbentuk *edit active x properties* → aktifkan *artificial horizon*



Gambar 3.22: *Edit properties aircraft*

L. Tambahkan *segment gauge* pada *artificial horizon* dan *button*



Gambar 3.23: Tampilan *artificial horizon matlab 2008B*

M. Klik tombol *button* lalu tuliskan *program* dibawah ini.

```
private void button11_Click(object sender, EventArgs e)
{
    roll10 += roll[5];
    for (int i = 4; i <= 6; i++)
    {
        roll[i] = 0;
    }
}
```

Program diatas digunakan untuk transfer data dari *potensiometer* ke mikrokontroler kemudian ditampilkan ke *form artificial horison* fungsi *string* `roll10 += roll[5];` didefinisikan di mikrokontroler sebagai inialisasi karakter ketika terjadi transfer data dan fungsi *string* `for (int i = 4; i <= 6; i++){roll[i] = 0;}` digunakan untuk *reset* data nilai sinyal input dari *potensiometer* akan dinolkan kembali.

3. Komunikasi *serial* dengan mikrokontroler AVR

Pada tugas akhir ini komunikasi *serial* digunakan untuk transfer data antara mikrokontroler dengan *visual C# studio 2008*. Pada tugas akhir ini menggunakan mikrokontroler AVR ATMEGA 8535 dan juga *usb downloader*

k510i yang sudah memiliki fitur *serial parallel RS232*. Langkah-langkah pembuatan dijelaskan dibawah ini:

A. Tambahkan komponen berikut:

1 x *GroupBox*

12 x *label*

6 x *ComboBox*

2 x *Button*

Atur letaknya sehingga seperti pada gambar dibawah ini:



Gambar 3.24: *form application design serial conection*

B. Atur semua *properties* komponen yang telah ditambahkan diatas sesuai tabel dibawah ini.

Tabel 3.3 *Properti* komponen komunikasi *serial port visual studio 2008*

Komponen	Properti	Nilai Properti
<i>GroupBox1</i>	<i>Text</i>	<i>Serial Conection</i>
<i>Button1</i>	<i>Text</i>	<i>Connect</i>
<i>Button2</i>	<i>Text</i>	<i>Play</i>
<i>Label1</i>	<i>Text</i>	<i>Port</i>
<i>Label2</i>	<i>Text</i>	<i>Baud Rate</i>
<i>Label3</i>	<i>Text</i>	<i>Data Bits</i>


<i>Label4</i>	<i>Text</i>	<i>Stop Bits</i>
<i>Label5</i>	<i>Text</i>	<i>Parity</i>
<i>Label6</i>	<i>Text</i>	<i>Handshake</i>
<i>Combobox1</i>	<i>Text</i>	
<i>Combobox2</i>	<i>Text</i>	9600
	<i>Items</i>	<i>Custom</i> 110 300 600 1200 2400 4800 9600 14400 19200 38400 56000 57600 115200 128000 256000
<i>ComboBox3</i>	<i>Text</i>	8
	<i>Items</i>	5 6 7 8
<i>ComboBox4</i>	<i>Text</i>	<i>One</i>
	<i>Items</i>	<i>None</i> <i>One</i> <i>Two</i>

		<i>OnePointfive</i>
<i>ComboBox5</i>	<i>Text</i>	<i>None</i>
	<i>Items</i>	<i>None</i> <i>Odd</i> <i>Even</i> <i>Mark</i> <i>Space</i>
<i>ComboBox6</i>	<i>Text</i>	<i>None</i>
	<i>Items</i>	<i>None</i> <i>XonXoff</i> <i>Request to send</i> <i>Request to send XonXoff</i>

- C. Tambahkan direktif *system.IO.Port* diawal program seperti pada gambar dibawah ini.

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.IO.Ports;
```

Penambahan direktif *System.IO.Port* ini bertujuan untuk memberitahukan *compiler* agar mendapatkan *class* yang digunakan dalam aplikasi ini, yaitu *port serial*.

- D. Pada *Form1*, klik tombol Event , lalu klik ganda pada event *Shown* dan tuliskan program dibawah ini.

```
private void Form1_Shown(object sender, EventArgs e)
{
    serialPort1.Close();
    button2.Text = "PLAY";
    groupBox2.Visible = false;
    groupBox3.Visible = false;
    button3.Enabled = false;
}
```

```

comboBox7.SelectedIndex = 0;

if (radioButton1.Checked == true)
{
    dir = maju;
}
else
{
    if (radioButton2.Checked == true)
    {
        dir = mundur;
    }
    else
    {
        dir = auto;
    }
}
roll0 = 512;
}

{
comboBox1.Items.Clear();
foreach (string s in SerialPort.GetPortNames())
comboBox1.Items.Add(s);
}

```

Program diatas bertujuan untuk mendapatkan *COM* yang bisa digunakan.

Sintaks yang berfungsi untuk mendapatkan *COM* ini adalah

`foreach (string s in SerialPort.GetPortNames())`.

`Serialport1.closed()`; digunakan untuk menutup *COM* dan mengakhiri sesi komunikasi dengan *COM*.

E. Klik tombol *Connect*, lalu tuliskan program dibawah ini:

```

private void button1_Click(object sender, EventArgs e)
{
    UseWaitCursor = true;

    if (button1.Text == "CONECT")
    {
        groupBox2.Visible = true;
        groupBox3.Visible = true;
        button1.Text = "DISCONNECT";
        radioButton1.Enabled = false;
        radioButton2.Enabled = false;
        radioButton3.Enabled = false;
        connecting();
    }
    else
    {
        parameterOFF();
        button3.Enabled = false;
    }
}

```

```

        button12.Visible = false;
        button1.Text = "CONNECT";
        ConStat = 0;
        if (readStat == 0) disconnecting();
    }
    UseWaitCursor = false;
}

private void conecting()
{
    button3.Enabled = false;
    if (comboBox2.Text != "")
    {
        serialPort1.PortName = comboBox1.Text;
        serialPort1.BaudRate = int.Parse(comboBox2.Text);
        serialPort1.DataBits = int.Parse(comboBox3.Text);
        serialPort1.StopBits =
        (StopBits)Enum.Parse(typeof(StopBits),
        comboBox4.Text);
        serialPort1.Parity = (Parity)Enum.Parse(typeof(Parity),
        comboBox5.Text);
        serialPort1.Handshake =
        (Handshake)Enum.Parse(typeof(Handshake), comboBox6.Text);
        serialPort1.Open();
        ConStat = 1;
    }
    else
    {
        MessageBox.Show(this, "Nothing COM selected", "Empty COM",
        MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
    if (serialPort1.IsOpen)
    {
        toolStripStatusLabel1.Text = "Serial Conected " +
        serialPort1.PortName;
        comboBox1.Enabled = false;
        comboBox2.Enabled = false;
        comboBox3.Enabled = false;
        comboBox4.Enabled = false;
        comboBox5.Enabled = false;
        comboBox6.Enabled = false;
        button3.Enabled = false;
        led1.Value = true;
    }
    else
    {
        toolStripStatusLabel1.Text = "Serial Not Conected";
    }
}

private void disconnecting()
{
    serialPort1.Close();
    if (serialPort1.IsOpen)
    {

```

```

toolStripStatusLabel1.Text = "Serial still Conected " +
serialPort1.PortName;
}
else
{
    button1.Text = "CONECT";
    button2.Text = "PLAY";
    timer1.Enabled = false;
    comboBox1.Enabled = true;
    comboBox2.Enabled = true;
    comboBox3.Enabled = true;
    comboBox4.Enabled = true;
    comboBox5.Enabled = true;
    comboBox6.Enabled = true;
    toolStripStatusLabel1.Text = "Serial Not Conected";
    ConStat = 0;
    led1.Value = false;
}
}

```

Untuk mendapatkan parameter *serial port* digunakan *sintaks* seperti dibawah ini.

```

serialPort1.PortName = comboBox1.Text;
serialPort1.BaudRate = int.Parse(comboBox2.Text);
serialPort1.DataBits = int.Parse(comboBox3.Text);
serialPort1.StopBits =
(StopBits)Enum.Parse(typeof(StopBits),
comboBox4.Text);
serialPort1.Parity = (Parity)Enum.Parse(typeof(Parity),
comboBox5.Text);
serialPort1.Handshake =
(Handshake)Enum.Parse(typeof(Handshake), comboBox6.Text);

```

Pada *sintaks* diatas, parameter *handshake* dan *namaport*, *baudrate*, *databits*, *stopbits*, *parity* diambil dari *combobox* yang terdapat pada *form*.

Untuk membuka *COM serial* digunakan fungsi

serialport1.Open();Serialport1.closed(); digunakan untuk menutup *COM* dan mengakhiri sesi komunikasi dengan *COM*.

F. Klik tombol *Play*, lalu tuliskan program dibawah ini.

```

private void button2_Click(object sender, EventArgs e)
{
    if (serialPort1.IsOpen)
    {
        if (button2.Text == "PLAY")
        {
            parameterON();
            button2.Text = "STOP";
        }
    }
}

```

```

        button3.Enabled = true;
        button12.Visible = true;
        timer1.Enabled = true;
        radioButton1.Enabled = true;
        radioButton2.Enabled = true;
        radioButton3.Enabled = true;

        datasend[0] = stop;
        serialPort1.Write(datasend, 0, 1);

        datasend[0] = 0;
        for (int i = 0; i <= 6; i++)
        {
            roll[i] = 0;
        }
        jarak = 0;
        rpm[5] = 0;
        angularGaugel.Value = 0;
        axAir1.AHRoll = 0 / 10;
        label2.Text = "";
        progressBar1.Value = 0;
    }
    else
    {
        parameterOFF();
        button2.Text = "PLAY";
        button3.Enabled = false;
        button12.Visible = false;
        timer1.Enabled = false;
        radioButton1.Enabled = false;
        radioButton2.Enabled = false;
        radioButton3.Enabled = false;
        datasend[0] = stop;
        serialPort1.Write(datasend, 0, 1);
    }
}
else
{
    MessageBox.Show(this, "Serial not conected", "COM not
selected", MessageBoxButtons.OK, MessageBoxIcon.Error);
} }

```

Untuk membuka COM serial digunakan fungsi *serialport1.Open()*;

Jika *serialport1.open()*;, maka data siap di transfer ke mikrokontroler dengan sintaks *serialPort1.Write(datasend, 0, 1)*; dan *button2*, *button3*, *button12*, *radiobutton1*, *radiobutton2*, *radiobutton3* di enable kan.

G. Klik tombol *Save*, lalu tuliskan program dibawah ini.

```

private void button6_Click(object sender, EventArgs e)
{
    if (serialPort1.IsOpen)

```

```

{
    datasend[0] = saving;
    serialPort1.Write(datasend, 0, 1);
    datasend[1] = Convert.ToByte(Kp * 10);
    datasend[2] = Convert.ToByte(Ki * 10);
    datasend[3] = Convert.ToByte(Kd * 10);
    if (Epwm > 127)
    {
        datasend[4] = 2;
    }
    datasend[5] = Convert.ToByte((Epwm % 128) * 2);
    datasend[6] = Convert.ToByte(Erpm * 2);
    datasend[7] = Convert.ToByte(setjarak * 2);
    datasend[8] = Convert.ToByte(mode * 2);
    for (int z = 1; z <= 8; z++)
    {
        serialPort1.Write(datasend, z, 1);
        int x = 0;
        while (x < 300000)
        {
            x++;
        }
        progressBar1.Max = progressBar1.Min +
(double)setjarak;
    }
    else
    {
        MessageBox.Show(this, "Serial not conected",
"missing conection", MessageBoxButtons.OK,
MessageBoxIcon.Error);
        parameterOFF();
        button3.Enabled = false;
        button1.Text = "CONECT";
        ConStat = 0;
        if (readStat == 0) disconnecting();
    }
}
}

```

Untuk mengirim data secara *serial* digunakan fungsi `serialPort1.Write (datasend, 0, 1);` yang artinya data *string* yang berada pada *data send* dikirim melalui *port parallel* ada 3 data yang akan dikirim ke mikrokontroler maka data harus dipisah-pisahkan supaya tidak tabrakan saat transfer data pemisahan data berupa angka yang sebelumnya di AVR juga sudah di inisialkan. Pemisahan data Dapat dilihat dibawah ini:

```

datasend[0] = saving;
serialPort1.Write(datasend, 0, 1);
datasend[1] = Convert.ToByte(Kp * 10);
datasend[2] = Convert.ToByte(Ki * 10);
datasend[3] = Convert.ToByte(Kd * 10);
if (Epwm > 127)

```

```

    {
        datasend[4] = 2;
    }
    datasend[5] = Convert.ToByte((Epwm % 128) * 2);
    datasend[6] = Convert.ToByte(Erpm * 2);
    datasend[7] = Convert.ToByte(setjarak * 2);
    datasend[8] = Convert.ToByte(mode * 2);

```

H. Klik *Load*, lalu tuliskan program dibawah ini:

```

private byte loading = 3;
private void button5_Click(object sender, EventArgs e)
{
    datasend[0] = loading;
    if(serialPort1.IsOpen) serialPort1.Write(datasend, 0, 1);
}

```

Program diatas berfungsi untuk meminta kembali data dari mikrokontroler untuk ditampilkan di *Form visual C# sintaks private byte loading = 3;* merupakan inisialisasi karakter.

(serialPort1.IsOpen) serialPort1.Write (datasend, 0, 1); artinya data *string* yang berada pada mikrokontroler dikirim melalui *port serial*.

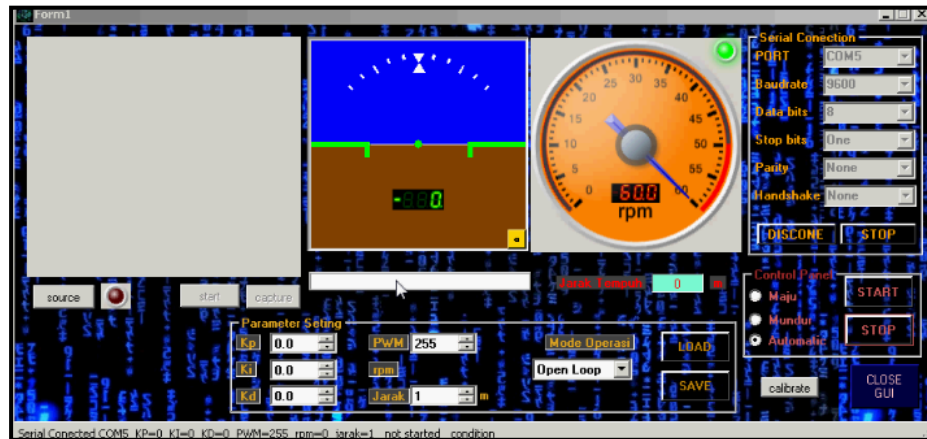
I. Klik tombol *Start*, lalu tuliskan program dibawah ini:

```

private void button3_Click(object sender, EventArgs e)
{
    if (serialPort1.IsOpen)
    {
        parameterOFF();
        datasend[0] = start;
        datasend[1] = dir;
        serialPort1.Write(datasend, 0, 1);
        int x = 0;
        while (x < 300000)
        {
            x++;
        }
        serialPort1.Write(datasend, 1, 1);
        datacek[12] = 3;
        datacek[11] = 0;
        jarak = 0;
    }
    else
    {
        MessageBox.Show(this, "Serial not conected", "missing
        conection", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}

```

Program diatas digunakan jika *COM serial* terbuka maka semua data *string* yang ada akan dikirim melalui *serial port* ke mikrokontroler.



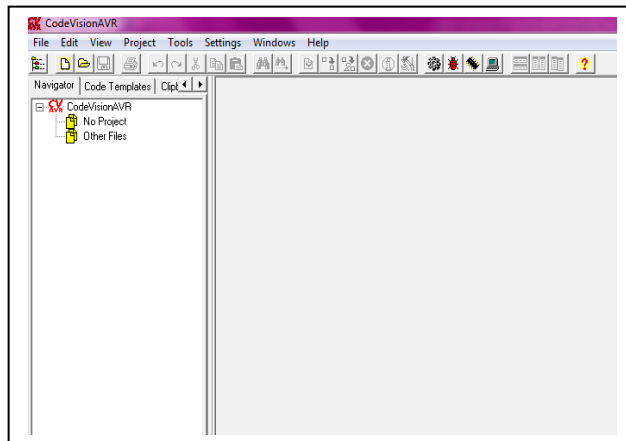
Gambar 3.25: Tampilan Desain *GUI Form C#* Robot Inspeksi Rel

III.4.2. Pemrograman *Microkontroller*

Pemrograman *microcontroller* bertujuan untuk mengendalikan proses putaran dari *Worm gear* DC motor yang merupakan *actuator* dari *inspection* robot tersebut. Sebagai *driver* kita menggunakan *relay* yang berfungsi untuk memutus atau menyambung aliran tegangan dari sumber arus. *Microcontroller* yang digunakan untuk tugas akhir ini adalah menggunakan jenis *AVR atmega8535*. Pemilihan *microcontroller Atmega8535* dalam tugas akhir ini karena *microcontroller* tersebut sudah mempunyai 4 *port* dengan 8 *pin* tiap-tiap *port*.

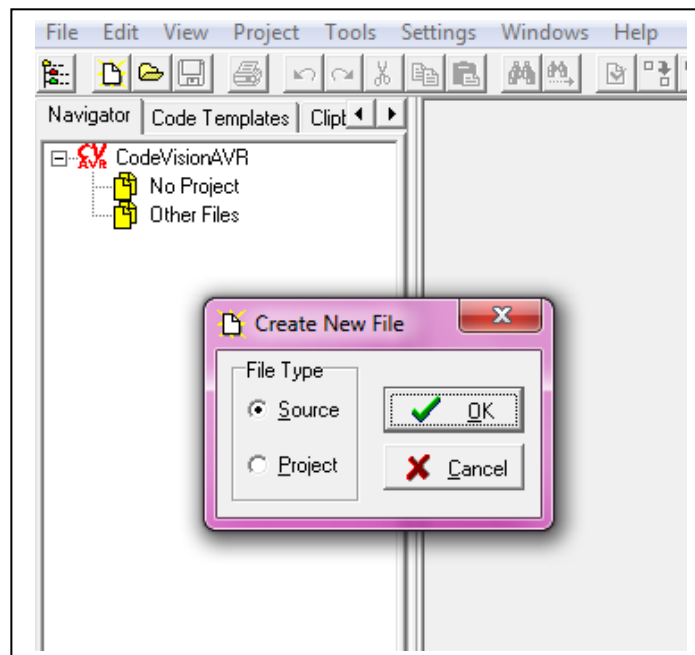
Bahasa pemrograman *microcontroller* yang digunakan untuk *inpection* robot dalam tugas akhir ini menggunakan bahasa pemrograman *C#*, sedangkan *software* yang digunakan untuk pembuatan program adalah menggunakan *software CodeVisionAVR*. Langkah-langkah dalam pemrograman adalah sebagai berikut:

1. Jalankan *software CodeVision AVR*



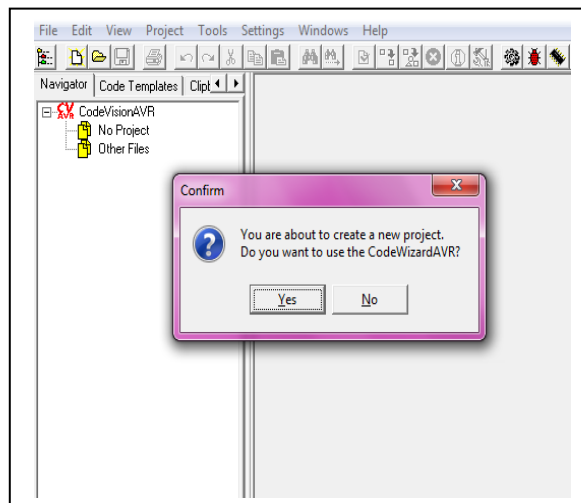
Gambar 3.25: *CodeVisionAVR.*

2. Buatlah *project* baru. Pilih *File New*. Pilih *project* lalu tekan tombol OK.



Gambar 3.26: *Create new project*

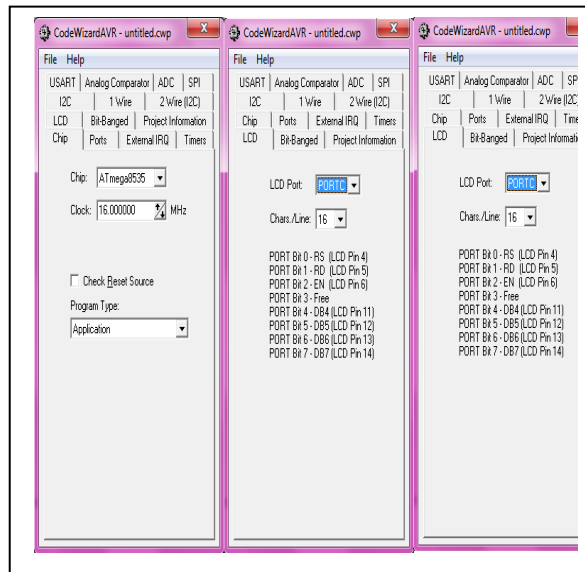
3. Kemudian muncul dialog apakah menggunakan *WizardAVR* untuk mempermudah pemrograman, pilih *YES*.



Gambar 3.27: Pilihan untuk menggunakan *Code Wizard*

4. *Setting Chip, Port Input/Output, dan LCD*

Proses awal yang dilakukan dalam pembuatan program *microcontroller* adalah *mensetting* jenis *ATMega* dan *crystal* yang digunakan. Proses selanjutnya adalah pemilihan-pemilihan *port AVR* yang nantinya akan digunakan sebagai *input* dan *output*, *setting port* untuk *Lcd*, dan *timer*. *AVR Atmega8535* memiliki 4 *port* dengan masing-masing *port* memiliki 8 *pin*. Untuk *inspection robot* dalam tugas akhir ini menggunakan jenis sensor digital yaitu *optocoupler* digunakan sebagai sensor kecepatan dan posisi.



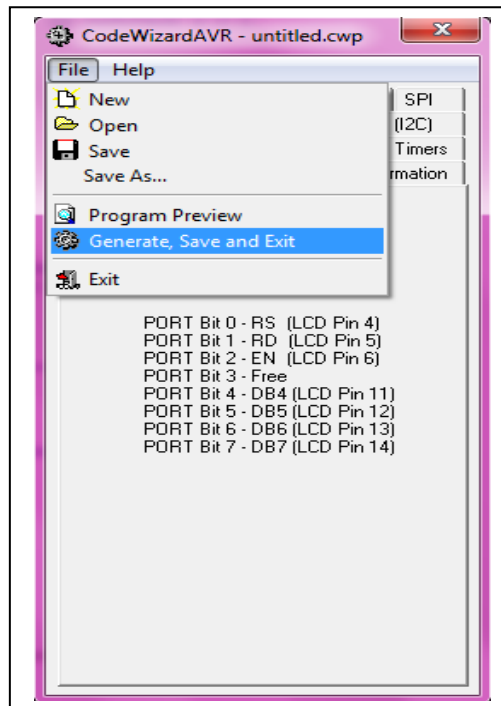
Gambar 3.28: Setting chip, pin Input/Output, dan LCD

Dalam tugas akhir ini penggunaan *pin* tiap-tiap *port input* dan *output* pada mikrokontroler digunakan sebagai mana fungsinya. Adapun konfigurasi *port* akan ditunjukkan pada tabel 3.2:

Tabel 3.4: Konfigurasi pengaktifan *port* pada mikrokontroler

PORT B		PORT C	PORT D		
PINB.0	<i>Input keypad</i> tombol 1		Dikonfigurasi sebagai <i>output LCD</i>	PIND.2	<i>Output ke serial</i> <i>RS232</i>
PINB.1	<i>Input keypad</i> tombol 2			PIND.6	<i>Output signal ke</i> <i>driver motor</i> (sebagai arah putaran motor)
PINB.2	<i>Input keypad</i> tombol 3			PIND.7	<i>Output signal</i> <i>pwm ke driver</i> motor
PINB.3	<i>Input keypad</i> tombol 4				
PINB.4	<i>Input ptocoupler</i>				

Proses selanjutnya adalah melakukan *generate* menjadi program *C#* awal yang akan diisi sesuai dengan program yang diinginkan. Hasil *generate* program ini masih berupa program *basic* yang masih umum dan belum ada perintah logika. Pembuat program harus melengkapi fungsi-fungsi pemrograman yang akan dijalankan.



Gambar 3.29: *Generate, save, dan exit*

Program yang telah digenerate akan tersimpan secara otomatis kedalam *format C#* dengan *directory* yang diinginkan. Setelah dilakukan proses, kita membuka ulang program yang hasil *generate* tersebut, kemudian melengkapi fungsi-fungsi pemrograman yang diinginkan.

5. *Download* Program ke *Chip AtMega 8535*

Setelah program yang kita buat selesai dan sesuai dengan keinginan, maka proses selanjutnya adalah melakukan *download* program kedalam *microcontroller*. Komunikasi yang digunakan dalam *download* program ini adalah menggunakan komunikasi *serial* yaitu menggunakan *RS232*. Karena komputer-komputer yang modern sudah menggunakan fitur *USB* maka kita memerlukan *hardware* bantuan

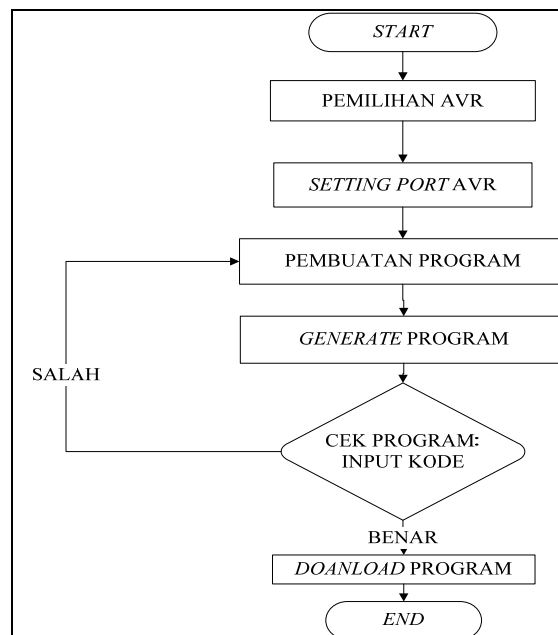
untuk mempermudah *download* data, yaitu perangkat *serial to USB Converter*, *hardware* ini berfungsi untuk mengkonversi *port* yang seharusnya menggunakan *port serial* diganti dengan *port USB*.

```

1  This program was produced by the
2  CodeVisionAVR V2.10.0 Standard
3  automatic Program Generator
4  © Copyright 1998-2010 Avra! Mikrotik, SP Infotech s.r.l.
5  http://www.avraforum.com
6
7  Project : HALI_INTERRUPT
8  Version : 2.1.0.1.10
9  Date : 7/8/2010
10 Author : kadek ngarak
11 Company : ADMTH
12
13 Comments:
14
15
16
17 Chip type : Atmega16
18 Program type : Application
19 AVR Core Clock Frequency: 16.000000 MHz
20 Memory model : Small
21 External RAM size : 0
22 Data Stack size : 512
23
24 *****
25 #include <mega16.h>
26
27 // AlphaNumeric LCD Module functions
28 #asm
29 #include <lcd.h>
30 #include <delay.h>
31 #include <stdio.h>
32
33 // subsetting macros
34 #asm
35 #define LCD_DATA_PORT PORTD
36 #define LCD_DATA_DDR DDRD
37
38 // Declare global variables
39 int i;
40 unsigned char L_state, rx_index, tx_index, button_pressed, cursor_x, cursor_y, wifil;
41 unsigned int rx_count, tx_count, rx_data, tx_data;

```

Gambar 3.30: Proses *download* program kedalam AVR



Gambar 3.31: Flowchart pemrograman microcontroller

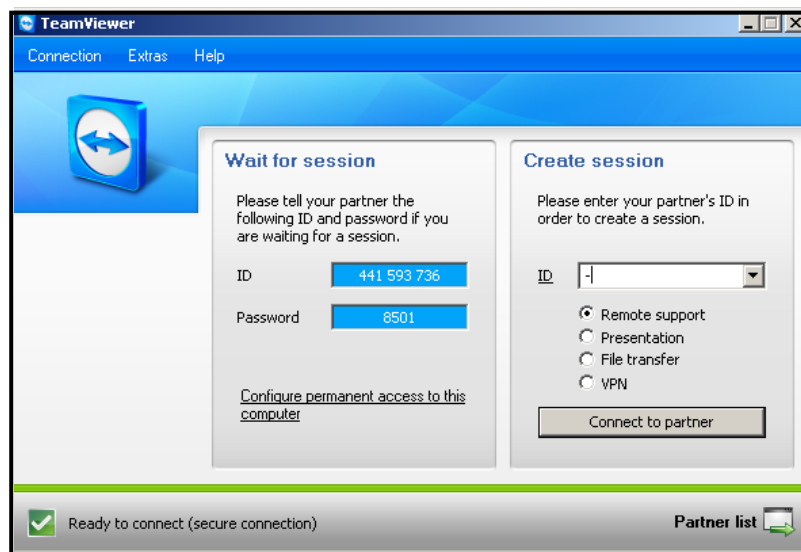
III.4.3. Remote komputer dengan Software Team viewer

Remote Desktop adalah istilah yang menggambarkan dimana sebuah komputer yang satu, bisa dikendalikan oleh komputer yang lainnya dengan menggunakan media jaringan komputer seperti Internet. Penulis tidak membahas tentang *hacking* disini, Pada tugas akhir ini *Software team viewer* digunakan untuk mengendalikan robot secara jarak jauh via online.

Adapun Langkah-langkahnya akan ditunjukkan dibawah ini:

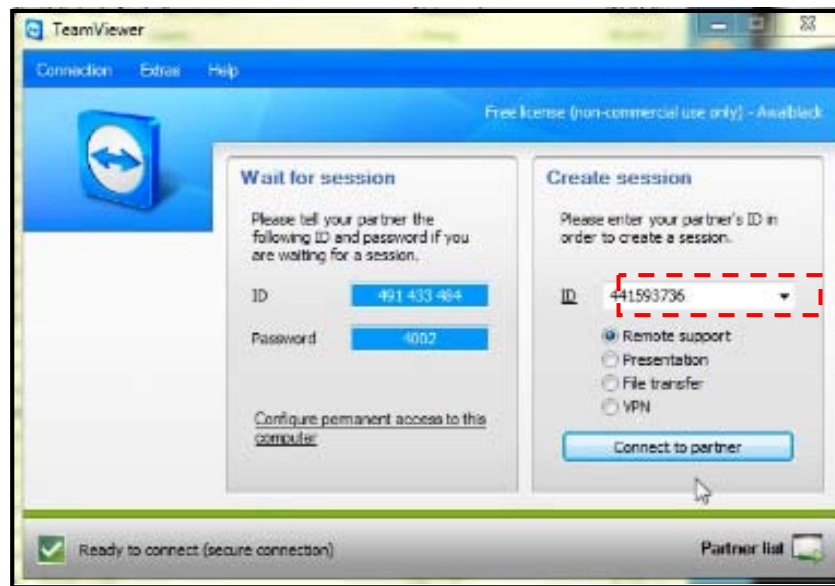
1. *Install software team viewer versi 5.*
2. *Open team viewer versi 5.*

Ketika membuka *software team viewer* otomatis akan terdeteksi *ID* dan *password*.



Gambar 3.32: Tampilan *Software Team viewer* versi 5

3. *Create session* → isi *ID* di komputer *server* → pilih opsi *remote support*, *presentation*, *file transfer*, *vpn* → kemudian *connect to partner*.

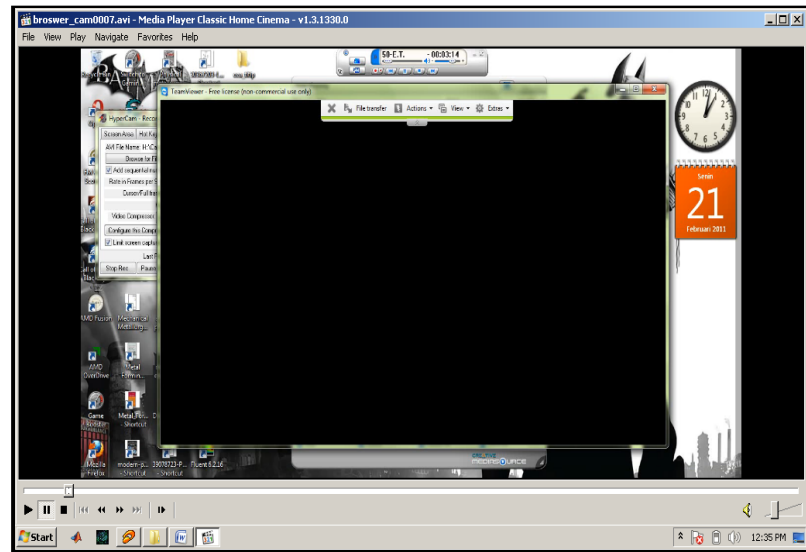


Gambar 3.33: *Create session* isi ID Client

4. Isikan *password* komputer *client* → *log on*



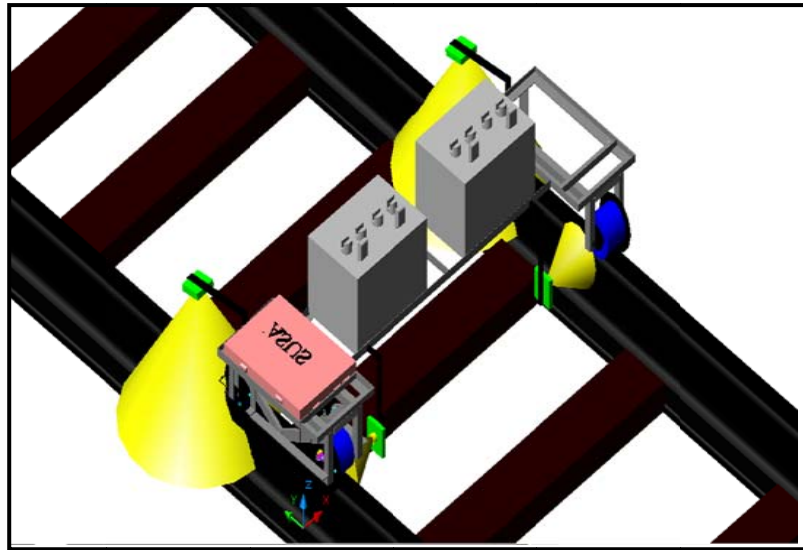
Gambar 3.34: *Authentication password team viewer*



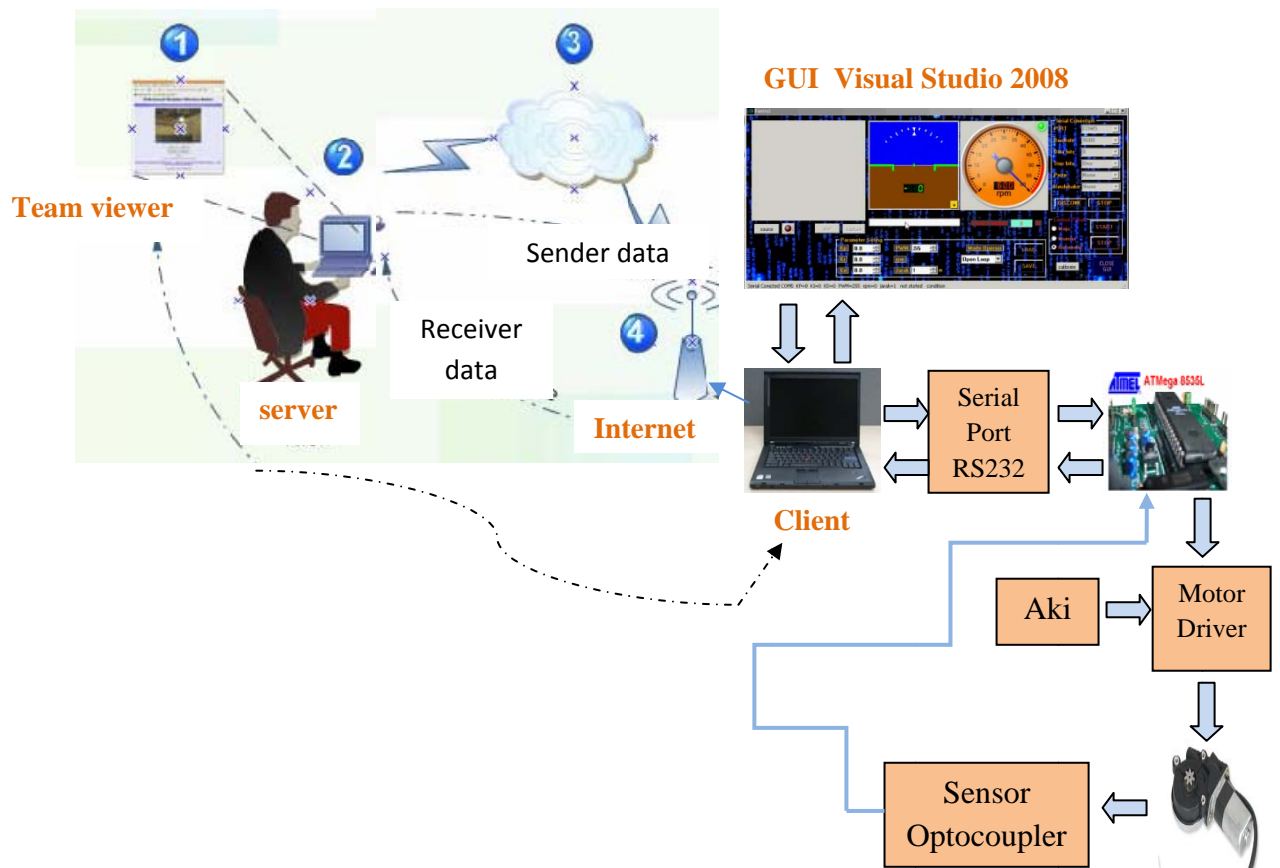
Gambar 3.35: Tampilan *team viewer* pada komputer *server*

III.5. Prinsip Kerja Robot

Robot inspeksi ini dimodelkan sebagai sebuah robot yang dapat berjalan di atas rel, yang digerakkan oleh motor *power window* bertenaga *battery accu*, sensor *optocoupler* yang berfungsi sebagai pembaca kecepatan yang kemudian diolah kedalam mikrokontroler untuk pengaturan kecepatan motor DC. Robot tersebut dilengkapi dengan kamera *webcam* yang mengambil rekaman gambar rel dari beberapa sudut. Kemudian data rekaman diolah oleh *processor* (berupa *netbook*) dan diteruskan ke *transmitter* yang berfungsi sebagai penguat sinyal dengan menggunakan bantuan *software team viewer* dan *software visual studio 2008*. Komunikasi *serial RS232* digunakan untuk transfer data mikrokontroler dengan *visual studio*. Kemudian data dari *transmitter* dikirimkan ke *receiver* (berupa laptop) secara *online* via internet. Operator dapat memantau perjalanan robot inspeksi melalui monitor *PC receiver*. Robot ini juga dilengkapi sensor untuk pendeteksi ketinggian rel kanan dan kiri yang diharapkan dapat memantau keadaan kerataan rel.



Gambar 3.36: Konsep robot inspeksi rel kereta api



Gambar 3.37: Blok Diagram prinsip kerja sistem robot inspeksi rel

Signal yang didapat dari sensor *optocoupler* akan diteruskan ke *notebook* yang dihubungkan oleh mikrokontroler dengan komunikasi *serial port RS232* seperti pada gambar. Kemudian *Signal* ini akan dikonversi menjadi data jarak ataupun kecepatan dari robot tersebut. Tampilan dari data jarak akan ditampilkan oleh *software Visual Studio 2008* dan data visualisasi didapat dari *webcam*. Data akuisisi tersebut selanjutnya di monitor melalui *PC server*. Untuk kendali jarak jauh menggunakan *software team viewer*. Dua pasang perangkat modem *gsm Huawei provider* yang digunakan adalah kartu IM2 sebagai koneksi antar *notebook Client* dengan *PC server* via internet.

BAB IV

HASIL DAN PEMBAHASAN

IV.1 Hasil Perancangan Robot

Dalam perancangan ini, telah berhasil dibuat sistem penggerak model atau Prototipe robot berbasis mikrokontroler dengan kendali jarak jauh via internet dan telah berfungsi dengan baik, dapat dikontrol dan dikendalikan melalui internet, dengan *software visual C#* dan *software team viewer* di sisi *klien*. Pada dasarnya semua aplikasi terpusat pada *client* robot, sedangkan *server* hanya perlu aplikasi *software team viewer* yang terkoneksi internet. Sangat perlu dilakukan uji mengenai sistem ini. Pengujian sistem dilakukan untuk mengetahui bagaimana sistem bekerja, fungsi kontrol melalui suatu program *visual C#*, kendali, dan fleksibilitas sistem. Sistem yang telah dihasilkan sudah bekerja sesuai dengan tujuan yang hendak dicapai penulis pada tugas akhir ini, namun disamping itu juga masih ada hal – hal yang perlu dibenahi.

IV.2 Model Atau Prototipe Robot

Pada perancangan robot inspeksi berbasis mikrokontroler ini, seperti pada gambar 4.1, struktur model merupakan bentuk yang dibuat secara sederhana, belum memperhitungkan aspek–aspek estetika perancangan, dan lebih mengutamakan berfungsinya sistem dengan baik. Oleh karena itu dengan model atau prototipe awal ini, akan memberikan kebebasan kreasi dan fitur selanjutnya dalam hal pengembangan.



Gambar 4.1: Model atau Prototipe Robot

Kerangka robot dibuat dari plat besi, dengan pengerjaan kerja bangku. Roda belakang dibuat berdiameter 8 cm, dan depan berdiameter 8 cm yang bisa bergerak memutar dengan mesin bubut. Dalam uji coba mobilitas robot, untuk gerak robot berfungsi dengan baik.

Pada prinsipnya kalibrasi putaran dilakukan dengan mengatur tegangan input ke motor DC. Kalibrasi bisa dilakukan dengan memasang sensor *counter (optocoupler)* pada roda belakang, yang digunakan untuk set dan komparasi pada sirkuit mikrokontroler khusus untuk pengatur putaran.

IV.3 Pengoperasian Sistem

Aplikasi design *form visual c#* robot inspeksi rel digunakan sebagai media interface antara mikrokontroler dengan komputer. Transfer data menggunakan komunikasi RS232.

IV.3.1 Monitoring Dengan Form Visual c#

Pemantauan rel kereta api dilakukan untuk mengetahui kondisi yang terjadi pada rel kereta api. Fungsi *monitoring* ini ditujukan untuk dapat melihat kondisi rel, kondisi baut dan sambungan plat. Desain *form visual c#* ini juga dilengkapi fitur *capture* gambar. Untuk tampilan visual dari kamera dapat dilihat seperti gambar 4.2



Gambar 4.2: Tampilan Visual *Monitoring* Kamera

IV.3.2 Hasil *Capturing* foto inspeksi rel

Capturing foto terjadi saat kondisi robot berjalan dengan menggunakan kamera 5.0 mega pixel yang terpasang dibadan robot. Hasil *capture* foto otomatis tersimpan di *harddisk netbook* yang terpasang pada robot, berikut foto hasil *capturing* :



Gambar 4.2: Hasil *capture* permukaan rel



Gambar 4.3: Hasil *capture* baut rel

IV.4 Perhitungan Mekanik dan Perhitungan Baterai Robot

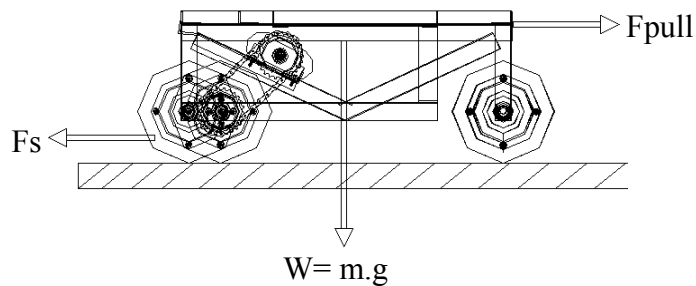
Perhitungan mekanik ini bertujuan untuk mengetahui akselerasi robot dan Perhitungan baterai bertujuan untuk mengetahui kemampuan jarak tempuh robot.

IV.4.1.1 Perhitungan Mekanik Robot

Dalam perancangan, robot data-data yang diketahui adalah:

Spesifikasi robot yang akan dirancang :

- Berat total dari robot = 15 kg
- Kecepatan motor = 60 Rpm
- Jari-jari dari roda = 4 cm = 0,04 m
- Reduksi rodagigi = 1,4



Robot dengan *massa* (m) ditarik kearah horizontal sumbu x, gaya tarik yang

dibutuhkan = $\Sigma F = 0$

$$F - fs = 0$$

$$F = fs$$

$$F = \mu \cdot m \cdot g$$

$$F = 0,078 \cdot 15 \text{ kg} \cdot 9,81 \text{ m/s}^2$$

$$= 11,48 \text{ N}$$

$$\text{Dimana } \mu = C_{TT} = \sqrt{0,05/8,2}$$

$$= 0,078$$

Untuk kecepatan (v) karena menggunakan reduksi *gear* maka kecepatan outputnya adalah;

$$N_2 = \text{Jumlah gigi 1} / \text{Jumlah gigi 2} \times N_1$$

$$= 1,4 \times 60 \text{ Rpm}$$

$$= 84 \text{ Rpm}$$

Kecepatan (v) dalam Rps adalah;

$$= 84 \text{ Rpm} \cdot 2\pi / 60$$

$$= 8,79 \text{ Rps}$$

Maka Kecepatan dalam m/s adalah;

$$= 8,79 \text{ Rps} \times 0,04 \text{ m}$$

$$= 0,352 \text{ m/s}$$

Didapatkan daya motor (P) adalah;

$$P = F.V = 11,48 \text{ N} \cdot 0,352 \text{ m/s} = 4,04 \text{ Watt}$$

Maka torsi motor adalah:

$$T = \frac{P}{\omega} = \frac{4,04}{8,79 \text{ Rps}} = 0,459 \text{ N.m}$$

IV.4.1.2 Perhitungan Baterai Robot

Spesifikasi aki yang digunakan :

- 1 Baterai aki merk *yuasa* = 5 ampere, 12 volt

A. Perhitungan Energi Baterai

$$W = V \cdot I \cdot t$$

$$\text{Maka energi aki} = W = 12 \text{ Volt} \times 5 \text{ AH}$$

$$= 60 \text{ watt Hours.}$$

B. Waktu pemakaian baterai (T pemakaian)

Perhitungan waktu pemakaian baterai ini bertujuan untuk mencari berapa lama pemakaian baterai aki dalam kondisi baterai penuh. Waktu pemakaian baterai dapat dicari dengan rumus :

$$T_{\text{pemakaian}} = \frac{\text{energi baterai}}{\text{daya motor}}$$

$$= \frac{60 \text{ watt Hours}}{4,04 \text{ watt}}$$

$$= 14,85 \text{ Hours}$$

C. Jarak tempuh yang dihasilkan

Disini kita dapat mengetahui berapa jauh jarak tempuh dari robot dalam kondisi baterai terisi penuh. Jarak tersebut dapat dicari dengan rumus :

$$\begin{aligned} S &= v \times T_{\text{pemakaian}} \\ &= 0,352 \text{ m/s} \times 53460 \text{ s} \\ &= 1882 \text{ meter} \end{aligned}$$

IV.5 Pengujian Baterai

Pengujian konsumsi arus listrik baterai dilakukan dengan tujuan untuk mengetahui lama pemakaian baterai. Pengujian konsumsi arus baterai ini diukur dengan menggunakan alat ukur multimeter dan diuji pada masing-masing bagian, yaitu rangkaian pengontrol (dua buah baterai *merk everday*), satu motor DC (satu buah baterai aki). Pengujian dilakukan pada kondisi robot diam dan bergerak. Selain itu, pengujian juga dilakukan ketika kondisi robot berbeban.

Tabel 4.1. Pengujian Konsumsi Arus Listrik Baterai Dengan Beban Rangka 15 kg

Bagian	Kondisi Diam	Kondisi Bergerak	Baterai	Voltase (Volt)	Estimasi Daya Tahan baterai (jam)	Hasil Uji (jam)
Rangkaian pengontrol	40mA	40mA	860mA	9	21,5	19,6
DC Motor	0	1295mA	5000mA	12	3,86	3,6

Dari pengujian yang ditunjukkan tabel 4.2, didapatkan hasil bahwa ketika robot berbeban 15 kg, baterai untuk rangkaian pengontrol mampu digunakan selama kurang lebih 19,6 jam terus menerus (berlaku pada kondisi diam maupun bergerak). Sedangkan baterai yang digunakan untuk DC motor mampu bertahan 3,6 jam dengan kecepatan maksimum 60 *rpm*.

BAB V

KESIMPULAN DAN SARAN

Pengembangan model atau prototipe Robot inspeksi rel kereta api merupakan perancangan mendasar sehingga perlu dikembangkan lebih lanjut. Masih banyak diperlukan unsur-unsur penunjang baik teknis maupun non teknis dalam melakukan pengembangan sistem ini. Namun bagian terpenting dari sistem ini adalah suatu sistem yang terintegrasi dalam bentuk Robot yang bisa di kontrol dan di *monitoring* melalui jaringan tanpa kabel via internet sudah dapat berfungsi.

Dengan menambahkan aplikasi *software team viewer* maka dari sisi pengguna dapat mengendalikan selama masih terhubung dengan jaringan internet dimanapun. Sedangkan untuk aplikasi – aplikasi tertentu, tentunya di sesuaikan dengan kebutuhan dan studi kasus yang sedang di hadapi.

V.1 KESIMPULAN

Setelah mengalami seluruh tahapan proses perancangan dan pembuatan prototipe robot inspeksi rel kereta api, terdapat beberapa hal yang dapat disimpulkan bahwa :

1. Spesifikasi robot yang telah dirancang adalah memiliki dimensi 247cm (P) x 78cm (L) x 23cm (t) dengan berat 15 kg. Robot menggunakan satu buah motor DC untuk menggerakkan robot.
2. Sistem robot inspeksi rel kereta api sudah bisa berfungsi dengan tujuan dan harapan yaitu robot dapat dikontrol dan dikendalikan melalui jaringan internet dengan menggunakan *software team viewer*. Dimana proses kendali dilakukan disisi *client* dengan menggunakan program hasil desain dari *software visual studio* yang dilengkapi dengan tampilan kamera.
3. Dari hasil pengujian baterai didapatkan lama adalah 19,6 jam sedangkan pada motor dc adalah 4,2 jam, untuk pengujian konsumsi arus baterai dengan beban rangka 15kg pada rangkaian pengontrol adalah 19,6 jam sedangkan pada motor dc adalah 3,6 jam.

V.2 SARAN

Dalam pembuatan model atau prototipe robot inspeksi rel kereta api ini masih banyak yang belum tergal, tentu saja kompleksitas akan sebanding dengan semakin banyaknya aspek pengontrolan. Namun dengan berhasil dibuatnya suatu model atau prototipe robot yang dapat dikontrol melalui jaringan internet ini, maka pengembangan ke yang lebih kompleks menjadi hal yang sangat mungkin untuk dilakukan. Setelah pembuatan model atau prototipe robot inspeksi rel kereta api ini, saran yang diberikan yaitu:

1. Perbaiki desain robot perlu dilakukan melihat dari tingkat kemudahan dalam Pemasangan, efisien, ringan serta mudah dibawa.
2. Sebaiknya robot dijalankan dengan kecepatan 60 *Rpm* dikarenakan jika terlalu cepat gambar yang dihasilkan oleh kamera tidaklah jelas.