

ROBOT SEGWAY LINE TRACER

TUGAS AKHIR

**Diajukan Sebagai Salah Satu Syarat
Untuk Memperoleh Gelar Sarjana Jurusan Teknik Informatika**



Oleh :

Nama : Prasyda Nur Windraya

No. Mahasiswa : 08 523 051

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

2012

LEMBAR PENGESAHAN PEMBIMBING
ROBOT SEGWAY LINE TRACER
TUGAS AKHIR



Affan Mahtarami, S.Kom., M.T.

LEMBAR PENGESAHAN PENGUJI

ROBOT SEGWAY LINE TRACER

Oleh :

Nama : Prasidya Nur Windraya

No. Mahasiswa : 08 523 051

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

Yogyakarta, April 2012

Tim Penguji

Affan Mahtarami, S.Kom., M.T.

Zainudin Zuhri, S.T., M.I.T.

Izzati Muhimmah, S.T., M.Sc., Ph.D.



Mengetahui,

Ketua Program Studi Teknik Informatika

Universitas Islam Indonesia

Yudi Prayudi, S.Si., M.Kom

LEMBAR PERNYATAAN KEASLIAN
HASIL TUGAS AKHIR

Saya yang bertanda tangan dibawah ini,

Nama : Prasyda Nur Windraya

No. Mahasiswa : 08 523 051

Menyatakan bahwa seluruh komponen dan isi dalam laporan Tugas Akhir ini adalah hasil karya sendiri, dan semua sumber baik yang dikutip maupun yang dirujuk telah saya nyatakan dengan benar. Bila di kemudian hari ternyata pernyataan saya terbukti tidak benar, maka saya bersedia menerima sanksi yang telah ditetapkan oleh Fakultas Teknologi Industri Universitas Islam Indonesia.

Demikian pernyataan ini saya buat, semoga dapat dipergunakan sebagaimana mestinya.

Yogyakarta, Maret 2012

Penulis



Prasyda Nur Windraya

HALAMAN PERSEMBAHAN

Tugas akhir ini ku persembahkan untuk :

Ayah dan Ibunda tercinta

(Prajoga dan Hindun Rahayu Winarni)

Terima kasih atas semua kasih sayangnya.

Guru dan Dosenku

*Terima kasih atas semua bimbingan serta arahan sehingga aku
bisa seperti sekarang ini*

Semua sahabatku tercinta

Terimakasih atas dorongan dan motivasinya !!!

MOTTO

“Sesungguhnya bersama kesulitan ada kemudahan. Maka apabila engkau telah selesai (dari sesuatu urusan), tetaplah bekerja keras (untuk urusan yang lain)”

(QS. Al Insyirah : 6-7)



KATA PENGANTAR

Segala puji dan syukur kehadirat Allah SWT atas segala limpahan rahmat dan karunia-Nya serta ridho-Nya sehingga tugas akhir dengan judul “**Robot Segway Line Tracer**” dapat diselesaikan.

Penulis menyadari bahwa penulisan tugas akhir ini terwujud berkat uluran tangan dari berbagai pihak. Pada kesempatan ini, penulis ingin mengucapkan terima kasih kepada :

1. Bapak Yudi Prayudi, S.Si, M.Kom. selaku Ketua Jurusan Teknik Informatika Universitas Islam Indonesia.
2. Bapak Affan Mahtarami, S.Kom., M.T. selaku dosen pembimbing yang telah memberikan pengarahan, bimbingan, serta masukan selama pelaksanaan tugas akhir dan penulisan laporan.
3. Mas Galang Prihadi Mahardhika, S.Kom yang telah memberikan masukan dan arahan dalam pembuatan robot ini.
4. Ayahanda Prajoga dan ibunda Hindun Rahayu Winarni yang telah banyak memberikan support dan doa serta kasih sayang yang begitu tulus dan ikhlas.
5. Teman-teman seperjuanganku (Tulus, Fachreza, Reza, Aziz) yang telah memberikan dukungan dan supportnya. “ Arigatou Gozaimasu Aibo !!! ”.
6. Seluruh staff dosen dan pengajar di Jurusan Teknik Informatika yang telah banyak menyampaikan pengetahuan dan wawasan teknologi informasi kepada penulis.
7. Teman-teman yang tergabung dalam SIDEV (Student Inside Development Program) yang telah memberikan dukungan dan supportnya. Terima kasih kawan. Semoga kalian sukses.
8. Teman-teman KKN yang sudah seperti keluarga bagiku. Terima kasih atas supportnya.

9. Semua pihak yang secara langsung maupun tidak langsung terlibat dalam penyelesaian penelitian Tugas Akhir ini.

Sebagai sebuah karya manusia, tulisan ini tentulah tidak sempurna. Untuk itu penulis akan senantiasa berlapang dada menerima saran dan kritik yang sifatnya membangun demi penyempurnaan tugas akhir ini.

Akhir kata, semoga tugas akhir ini dapat memberikan manfaat bagi pembaca. Amiin.



Yogyakarta, Maret 2012

Penulis

ABSTRAK

Dewasa ini keberadaan robot sebagai mesin yang dapat membantu pekerjaan manusia telah banyak dijumpai. Hal tersebut nampak dari banyaknya tugas-tugas manusia yang dilakukan oleh robot. Jenis robot ada bermacam-macam, mulai dari robot industri hingga robot rumah tangga. Struktur bentuk robot juga berbeda-beda, ada robot berkaki dan ada pula robot beroda. Salah satu robot beroda adalah Robot Segway. Robot ini terinspirasi oleh pembuatan kendaraan Segway PT karya Dean Kamen. Prinsip yang digunakan dalam pembuatan robot ini adalah pendulum terbalik. Robot juga diberi tambahan kecerdasan berupa line follower atau line tracer skill.

Dalam laporan Tugas Akhir ini akan dipaparkan tentang perancangan dan pembuatan Robot Segway Line Tracer menggunakan NXT Lego Mindstorms. Perancangan desain body robot menggunakan software Lego Digital Designer (LDD). Program dibuat menggunakan bahasa pemrograman Java dengan firmware LeJOS NXJ. Pengujian konsep line tracer menggunakan arena line follower yang berupa garis hitam diatas permukaan berwarna putih.

Kata kunci : Robot, Segway, Line Tracer

TAKARIR

<i>Autonomous Robot</i>	<i>Robot yang dapat bekerja secara otomatis sesuai instruksi program yang diberikan dan tanpa kontrol dari manusia</i>
<i>Degree of Freedom</i>	<i>Derajat independensi yang diperlukan untuk menyatakan posisi suatu sistem pada setiap saat</i>
<i>Degree per Second</i>	<i>Satuan SI untuk kecepatan sudut</i>
<i>Drag and Drop</i>	<i>Merupakan teknik untuk memindahkan item-item tertentu seperti ikon dan file pada layar monitor</i>
<i>Derivative Error</i>	<i>Nilai error yang diperoleh dari selisih nilai error dan previous error dalam PID Controller</i>
<i>Error</i>	<i>Suatu kesalahan proses yang terjadi pada program</i>
<i>Hardware</i>	<i>Salah satu komponen pada komputer yang sifat alatnya bisa dilihat dan diraba secara langsung</i>
<i>Inverted Pendulum</i>	<i>Sistem non linear dan tidak stabil yang sering digunakan untuk menguji performansi dan efisiensi metode kontrol</i>
<i>Integral Error</i>	<i>Nilai yang diperoleh dari penjumlahan nilai error sebelumnya pada PID Controller</i>
<i>Line Follower</i>	<i>Konsep kecerdasan robot dalam bentuk mengikuti garis</i>
<i>Motor Angle</i>	<i>Nilai sudut yang dihasilkan oleh servo motor</i>

<i>Motor Velocity</i>	<i>Nilai kecepatan sudut yang dihasilkan oleh servo motor</i>
<i>Motor Power</i>	<i>Nilai power yang dikeluarkan oleh servo motor berdasarkan perhitungan</i>
<i>Pseudocode</i>	<i>Deskripsi dari algoritma pemrograman komputer yang menggunakan struktur sederhana dengan tujuan untuk memudahkan manusia dalam memahami prinsip dari suatu algoritma</i>
<i>Port</i>	<i>Mekanisme yang mengizinkan sebuah komputer untuk mendukung beberapa sesi koneksi dengan komputer lain atau dengan alat sejenis</i>
<i>Programming Environment</i>	<i>Sarana pendukung untuk membuat program seperti Netbeans dan Eclipse</i>
<i>Pulse Mode</i>	<i>Satuan nilai power yang dihasilkan oleh servo motor</i>
<i>Software</i>	<i>Sekumpulan data elektronik yang disimpan dan diatur oleh komputer</i>
<i>Semi Autonomous</i>	<i>Robot yang dapat dikontrol manusia dan juga diberi program tertentu sesuai fungsi robot tersebut</i>
<i>Tilt Angle</i>	<i>Nilai sudut yang dihasilkan oleh sensor gyro</i>
<i>Tilt Velocity</i>	<i>Nilai kecepatan sudut yang dihasilkan oleh sensor gyro</i>

DAFTAR ISI

LEMBAR PENGESAHAN PEMBIMBING	i
LEMBAR PENGESAHAN PENGUJI	ii
LEMBAR PERNYATAAN KEASLIAN.....	iii
HALAMAN PERSEMBAHAN	iv
MOTTO.....	v
KATA PENGANTAR	vi
ABSTRAK	viii
TAKARIR	ix
DAFTAR ISI	xi
DAFTAR TABEL	xiv
DAFTAR GAMBAR	xv
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah	2
1.4 Tujuan Penelitian	2
1.5 Sistematika Penulisan.....	3
BAB II LANDASAN TEORI.....	4
2.1 Sejarah Robot.....	4

2.2 Jenis-jenis Robot Beroda	4
2.3 Derajat Kebebasan.....	7
2.4 Segway.....	9
2.5 Inverted Pendulum.....	9
2.6 PID Controller.....	10
2.7 Algoritma Line Follower.....	12
2.8 Lego Mindstorm.....	14
2.9 Perkembangan Lego Segway Robot.....	18
2.9.1 LegWay(Segway Robot)-RCX dan Electro Optical Proximity Detectors.....	18
2.9.2 NXTway(Segway Robot)-NXT dan Light Sensor.....	18
2.9.3 NXTway-G-NXT dan Gyroscope.....	19
2.10 Bahasa Pemrograman.....	20
BAB III METODOLOGI.....	22
3.1 Pendahuluan.....	22
3.2 Langkah-langkah Pembuatan	22
3.3 Struktur Program	33
3.4 Desain Arena Line Follower.....	35
BAB IV HASIL DAN PEMBAHASAN.....	38
4.1 Deskripsi Kelas	38
4.1.1 RobotMain.....	38

4.1.2 BalanceKontrol	40
4.1.3 MotorKontrol.....	42
4.1.4 GiroSensor.....	43
4.1.5 ParameterKontrol.....	45
4.1.6 MotorDirection	45
4.1.7 LineFollower	46
4.1.8 BT.....	48
4.1.9 BT_reader.....	50
4.2 Hasil Pengujian Robot	51
BAB V KESIMPULAN DAN SARAN.....	55
5.1 Kesimpulan.....	55
5.2 Saran.....	55
DAFTAR PUSTAKA	56
LAMPIRAN	58

DAFTAR TABEL

Tabel 2.1 Efek dari setiap kontroler (K_p , K_i , K_d) dalam loop tertutup	11
Tabel 4.1 Nilai Bobot Konstanta PID Controller	40
Tabel 4.2 Hasil Uji Robot Pada Arena I	51
Tabel 4.3 Hasil Uji Robot Pada Arena II	52
Tabel 4.4 Hasil Uji Robot Pada Arena III	53



DAFTAR GAMBAR

Gambar 2.1 Bentuk Robot Roda 2	5
Gambar 2.2 Roomba Vacuum Cleaner.....	5
Gambar 2.3 Mekanisme 3-Wheeled Robot	6
Gambar 2.4 Omni Wheels	6
Gambar 2.5 4-Wheeled Robot.....	7
Gambar 2.6 Degree Of Freedom.....	8
Gambar 2.7 Robot Arm.....	8
Gambar 2.8 Dean Kamen Segway PT.....	9
Gambar 2.9 Inverted Pendulum.....	10
Gambar 2.10 Blok Diagram PID Controller	11
Gambar 2.11 Pseudo Code PID Controller	12
Gambar 2.12 Tombol Fungsi RCX	15
Gambar 2.13 Port RCX.....	15
Gambar 2.14 Tombol Fungsi NXT	16
Gambar 2.15 Port NXT.....	17
Gambar 2.16 HiTechnic Gyro Sensor.....	17
Gambar 2.17 Robot LegWay	18
Gambar 2.18 NXTway Robot.....	19
Gambar 2.19 NXTway-G.....	19

Gambar 3.1 NXT Brick.....	24
Gambar 3.2 Servo Motor.....	24
Gambar 3.3 Sensor Gyro.....	25
Gambar 3.4 Sensor Cahaya	25
Gambar 3.5 Kabel Konektor NXT.....	25
Gambar 3.6(a) Robot Tampak Depan.....	26
Gambar 3.6(b) Robot Tampak Samping	26
Gambar 3.6(c) Robot Tampak Belakang	26
Gambar 3.6(d) Robot Tampak Atas	26
Gambar 3.7 Lego NXT Mindstorm Edu.....	27
Gambar 3.8 Logo leJOS.....	28
Gambar 3.9 Logo Eclipse	28
Gambar 3.10 Sistem Kontrol	29
Gambar 3.11 Sistem Kontrol dengan PID Controller.....	30
Gambar 3.12 Diagram Nyquist	31
Gambar 3.13 Flowchart Alur Program.....	32
Gambar 3.14 Diagram Kelas	34
Gambar 3.15 Robot Dengan Dua Sensor Cahaya	35
Gambar 3.16 Desain Track 1	36
Gambar 3.17 Desain Track 2	36
Gambar 3.18 Desain Track 3	37

Gambar 4.1 Source Code-1 RobotMain	38
Gambar 4.2 Source Code-2 RobotMain.....	39
Gambar 4.3 Source Code BalanceKontrol.....	41
Gambar 4.4 Source Code MotorKontrol.....	42
Gambar 4.5 Source Code -1 GiroSensor.....	43
Gambar 4.6 Source Code-2 GiroSensor.....	44
Gambar 4.7 Source CodeParameterKontrol	45
Gambar 4.8 Source Code -1 MotorDirection	46
Gambar 4.9 Source Code-2 MotorDirection.....	46
Gambar 4.10 Source Code-1 LineFollower.....	47
Gambar 4.11 Source Code-2 LineFollower	47
Gambar 4.12 Source Code-3 LineFollower.....	48
Gambar 4.13 Source Code -1 BT	49
Gambar 4.14 Source Code-2 BT.....	49
Gambar 4.15 Source Code-3 BT.....	49
Gambar 4.16 Source Code BT_reader.....	50
Gambar 4.17 Uji-1	54
Gambar 4.18 Uji-2	54

BAB I

PENDAHULUAN

1.1 Latar Belakang

Dewasa ini keberadaan robot sebagai mesin yang dapat membantu pekerjaan manusia telah banyak dijumpai. Hal tersebut nampak dari banyaknya tugas-tugas manusia yang dilakukan oleh robot. Robot banyak dijadikan sebagai alat bantu manusia dalam mengerjakan pekerjaannya, sebagai contoh robot arms yang terdapat pada pabrik-pabrik. Robot tersebut berguna untuk membantu manusia dalam proses industri. Selain itu terdapat pula robot-robot yang berguna membantu pekerjaan rumah tangga seperti robot pembersih lantai.

Bentuk robot ada bermacam-macam, mulai dari robot berkaki hingga robot beroda. Salah satu bentuk robot beroda adalah Robot Segway. Robot Segway ini memiliki prinsip kerja yang hampir mirip dengan sistem pendulum terbalik (Geoffrey D, Benneth, 2007). Segway pada awalnya adalah sebuah kendaraan listrik beroda dua yang diciptakan oleh Dean Kamen yang dapat menyeimbangkan diri. Konsep inilah yang mendasari pembuatan robot segway dalam tugas akhir ini.

Robot dapat dikatakan cerdas bila dapat menyelesaikan atau memecahkan suatu masalah. Salah satu permasalahan yang dapat dijadikan tantangan adalah *line follower*.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dijelaskan di atas, maka dapat diambil suatu rumusan masalah sebagai berikut :

- a. Bagaimana membuat sebuah robot beroda yang tetap dapat menjaga keseimbangan walaupun hanya menggunakan dua roda saja (Robot Segway).
- b. Bagaimana mengimplementasikan alat-alat berupa paket Lego NXT MINDSTORM yang terdiri dari NXT Brick, servo motor, dan sensor-sensor untuk membentuk robot segway tersebut.
- c. Bagaimana robot tersebut dapat menyelesaikan tantangan berupa arena line follower.

1.3 Batasan Masalah

Batasan masalah yang terdapat pada pembuatan robot segway ini adalah :

- a. Body robot dibuat menggunakan Lego NXT Mindstorm.
- b. Bahasa pemrograman yang digunakan adalah java.
- c. Sensor yang digunakan sebagai pusat kendali keseimbangan adalah sensor gyro, sedangkan sensor yang digunakan untuk berjalan mengikuti garis adalah sensor cahaya.
- d. Arena berupa garis hitam pada landasan berwarna putih.

1.4 Tujuan Penelitian

Adapun tujuan dalam pengerjaan tugas akhir yang berjudul “ROBOT SEGWAY LINE TRACER” ini adalah untuk memperoleh suatu rancangan robot keseimbangan dengan dua roda (segway) dan mengeksplorasi kemampuan dari Lego NXT Mindstorm sebagai robot kit yang dapat diprogram.

1.5 Sistematika Penulisan

Penulisan laporan Tugas Akhir ini terdiri dari beberapa bagian, yaitu :

BAB I PENDAHULUAN

Bab ini merupakan pengantar permasalahan yang akan dibahas. Berisi tentang gambaran umum mengenai objek penelitian dimana dalam tugas akhir ini adalah pembuatan Robot Segway Line Tracer. Terdiri dari latar belakang masalah, batasan masalah, tujuan penelitian, dan sistematika penulisan.

BAB II LANDASAN TEORI

Bab ini membahas tentang teori-teori pendukung dalam pembuatan Robot Segway Line Tracer. Meliputi teori-teori yang berhubungan dengan definisi robot, peralatan, komponen *hardware* dan *software* yang akan digunakan, bahasa pemrograman yang digunakan, serta ulasan penelitian yang sejenis yang pernah dilakukan.

BAB III METODOLOGI

Bab ini membahas tentang langkah-langkah yang akan dilakukan dalam pembuatan Robot Segway Line Tracer.

BAB IV HASIL DAN PEMBAHASAN

Bab ini berisi uraian tentang inti program dan potongan source code program serta hasil pengujian robot pada lintasan yang telah dibuat.

BAB V KESIMPULAN DAN SARAN

Bab ini berisi kesimpulan dan saran. Kesimpulan yang merupakan rangkuman dari hasil penelitian dan saran yang berisi masukan yang perlu diperhatikan berdasarkan keterbatasan yang ditemukan.

DAFTAR PUSTAKA

LAMPIRAN

BAB II

LANDASAN TEORI

2.1 Sejarah Robot

Istilah robot berasal dari bahasa Cheko yaitu “Robota” yang berarti pekerja, kuli atau budak yang tidak mengenal lelah atau bosan. Istilah ini pertama kali muncul pada tahun 1920 dalam drama pentas Rossum’s Universal Robots karya Karel Capek. Robot dapat melakukan tugas fisik, baik menggunakan pengawasan dan kontrol manusia, maupun menggunakan program yang telah didefinisikan terlebih dahulu. Menurut tingkat interaksi antara manusia dengan robot, robot dapat dibagi menjadi 3 jenis yaitu *autonomous robot*, *semi-autonomous*, dan remot kontrol robot. (Pitowarno, E. 2007).

Pada awalnya aplikasi robot hampir tidak dapat dipisahkan dengan dunia industri sehingga muncul istilah robot industri. Robot industri adalah robot yang diciptakan untuk menggantikan peran manusia dalam proses industri (buruh pabrik). Robot ini banyak digunakan untuk melakukan tugas yang berbahaya, nonstop, memerlukan kecepatan, dan tingkat konsentrasi yang tinggi. Namun belakangan ini, robot mulai memasuki pasaran konsumen di bidang hiburan, dan alat pembantu rumah tangga, seperti robot penyedot debu atau robot pembersih lantai.

2.2 Jenis-Jenis Robot Beroda

Robot yang berkembang saat ini pada umumnya bertipe robot beroda. Robot beroda adalah robot yang mengandalkan roda sebagai sarana berjalan. Desain ini lebih mudah untuk dibuat dibandingkan dengan

menggunakan kaki seperti pada robot humanoid dan animaloid. Jenis-jenis robot beroda diantaranya adalah :

1. *2-wheeled robots* (roda 2)
2. *3-wheeled robots* (roda 3)
3. *Omni wheels*
4. *4-wheeled robots* (roda 4)

2-wheeled robot adalah robot yang berporos pada dua roda sehingga menyulitkannya untuk berdiri seimbang. Karena itu robot ini harus aktif bergerak untuk tetap menjaga keseimbangannya. Tipe robot ini sekarang banyak dijumpai pada robot-robot rumah tangga yaitu robot vacuum cleaner atau penyedot debu. Bentuk tipe robot roda dua ditunjukkan oleh gambar 2.1 dan gambar 2.2 dibawah ini (Sumber : <http://nextsys.web.id/edukasi/motor-dan-gearbox>).

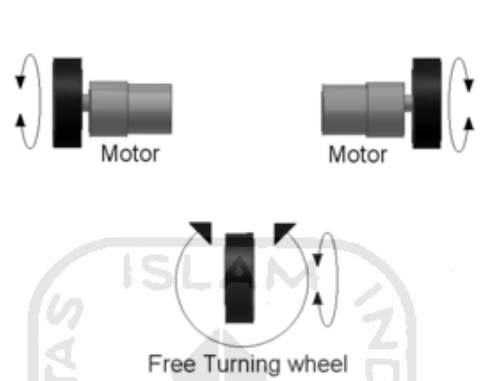


Gambar 2.1 Bentuk Robot Roda 2



Gambar 2.2 Roomba Vacuum Cleaner

3-wheeled robot adalah robot beroda dua yang menambahkan satu roda lagi di belakang sebagai roda bebas. Dengan adanya roda bebas tersebut, maka robot dapat berputar dengan mudah. Mekanisme *3-wheeled robot* ditunjukkan oleh gambar 2.3 berikut ini.



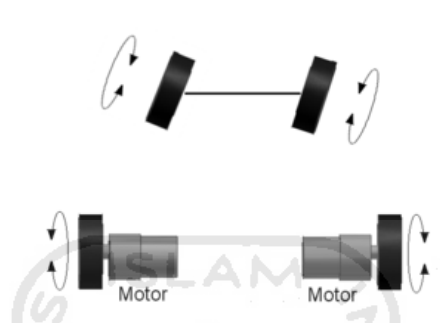
Gambar 2.3 Mekanisme 3-Wheeled Robot

Omni wheels adalah tipe robot beroda yang memiliki banyak roda-roda kecil dalam satu lingkaran roda. Hal ini membuatnya mudah untuk berputar dan bergerak ke segala arah. Omni wheels ditunjukkan oleh gambar 2.4 dibawah ini.



Gambar 2.4 Omni Wheels

4-wheeled robot adalah robot yang memiliki 4 buah roda. Sama seperti mobil, robot ini mengandalkan dua roda belakang sebagai motor penggerak dan dua roda depan sebagai kontrol arah. *4-wheeled robot* ditunjukkan oleh gambar 2.5 dibawah ini.



Gambar 2.5 4-Wheeled Robot

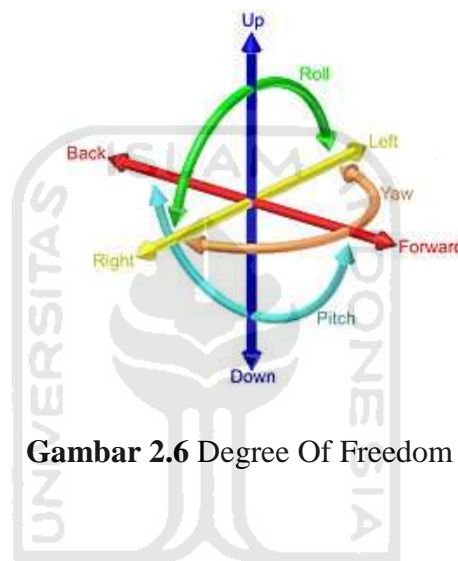
2.3 Derajat Kebebasan

Derajat kebebasan (*degree of freedom*) adalah derajat independensi yang diperlukan untuk menyatakan posisi suatu sistem pada setiap saat. Pada masalah dinamika, setiap titik atau massa pada umumnya hanya diperhitungkan berpindah tempat dalam satu arah saja yaitu arah horizontal (Kuningsih, T.W. 2011).

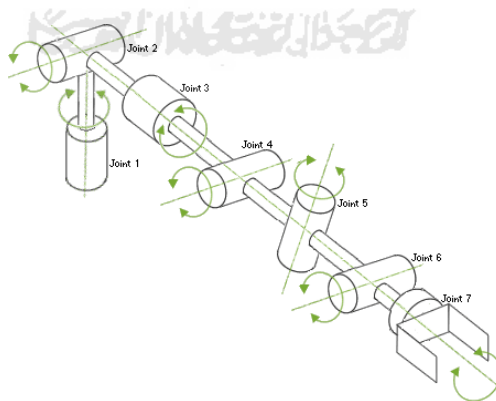
Dalam dunia robotika, konsep *degree of freedom* sangatlah penting karena dalam membuat sebuah robot perlu mengetahui jumlah derajat kebebasan yang akan dipakai. Misalnya pada robot arms atau robot tangan, robot ini merupakan tiruan dari tangan manusia. Seperti halnya tangan manusia yang memiliki sendi, maka robot memiliki derajat kebebasan. Dengan kata lain derajat kebebasan adalah sendi.

Dalam teori analisis struktur dijelaskan bahwa *degree of freedom* ada enam yaitu tiga buah translasi (arah x, y, z) yang menghasilkan arah x, y, z dan tiga buah rotasi (arah x, y, z) yang menghasilkan moment arah x, y, dan z.

Robot Segway hanya menggunakan satu *degree of freedom* yang menyebabkan robot ini tidak seimbang. Untuk menyeimbangkannya dibutuhkan suatu proses perhitungan yang tepat. Struktur *degree of freedom* ditunjukkan oleh gambar 2.6 dan 2.7.



Gambar 2.6 Degree Of Freedom



Gambar 2.7 Robot Arms

2.4 Segway

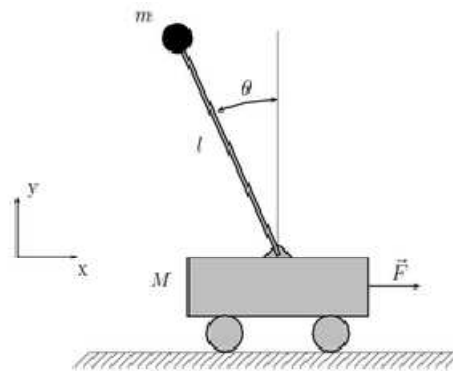
Segway diperkenalkan pertama kali oleh Dean Kamen pada 27 Juli 1999. Dean Kamen menyebutnya dengan istilah Segway Personal Transporter. Segway dibuat dengan menggunakan konsep *inverted pendulum* atau pendulum terbalik. Segway PT dapat dikendalikan dengan cara mencodongkan tubuh si pemakai ke arah depan atau belakang agar dapat bergerak maju atau mundur. Alat ini dapat berjalan dengan kecepatan 20 kph. Segway PT ini menggunakan 5 mikro sensor giroskopik dan dua buah akselerometer yang berguna untuk menyeimbangkan diri. Segway PT ditunjukkan oleh gambar 2.8.



Gambar 2.8 Dean Kamen Segway PT

2.5 Inverted Pendulum

Robot segway adalah robot yang memiliki prinsip kerja yang hampir mirip dengan sistem pendulum terbalik (*inverted pendulum*). *Inverted pendulum* adalah bandul yang memiliki massa yang berputar di atas titik. Hal ini sering dilakukan dengan berporos titik menjulang pada kereta yang dapat bergerak horizontal dan dapat disebut sebagai tiang dan kereta. Bandul biasa stabil bila digantung ke bawah. Sebuah bandul terbalik memiliki sifat tidak stabil sehingga harus aktif bergerak agar posisi selalu seimbang (Fachri, et.al., 2010). Inverted Pendulum ditunjukkan oleh gambar 2.9.



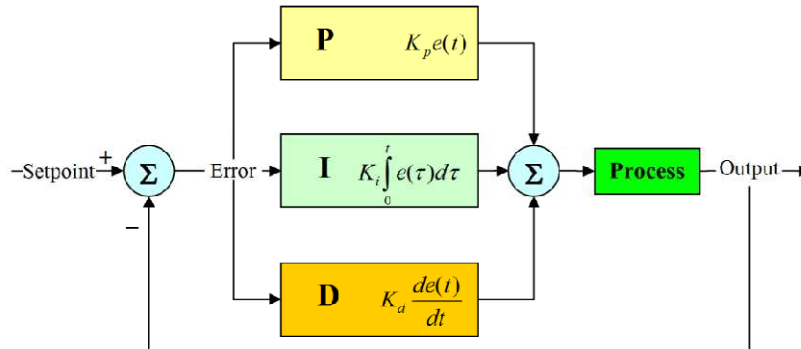
Gambar 2.9 Inverted Pendulum

2.6 PID Controller

PID Controller adalah suatu sistem yang akan mengatur nilai power yang akan dikirimkan kepada motor. Jika tidak ada PID Controller maka nilai power yang dikeluarkan akan langsung dikirimkan ke motor sehingga menyebabkan robot berjalan laju dan tidak seimbang. Dengan adanya PID Controller ini, maka nilai power tersebut dihitung dahulu untuk menghasilkan nilai power yang sesuai agar robot dapat menyeimbangkan diri.

Karakteristik PID Controller sangat dipengaruhi oleh kontribusi besar dari ketiga parameter yaitu P, I, dan D. Penyetelan konstanta K_p , K_i , dan K_d akan mengakibatkan penonjolan sifat dari masing-masing elemen. Satu atau dua dari ketiga konstanta tersebut dapat diset lebih menonjol dibanding yang lain. Konstanta yang menonjol itulah yang akan memberikan kontribusi pengaruh pada respon sistem secara keseluruhan.

Setiap kekurangan dan kelebihan masing-masing kontroler dapat saling menutupi dengan menggabungkan ketiganya secara parallel menjadi kontroler PID. Gambar 2.10 menunjukkan blok diagram PID Controller (Fachri, et.al., 2010).



Gambar 2.10 Blok Diagram PID Controller

Proporsional kontroler (K_p) akan memberikan efek mengurangi waktu naik, tetapi tidak menghapus kesalahan keadaan tunak. Integral kontroler (K_i) akan memberikan efek menghapus kesalahan keadaan tunak, tetapi berakibat memburuknya respon transien. Diferensial kontroler (K_d) akan memberikan efek meningkatnya stabilitas sistem, mengurangi over-shoot, dan menaikkan respon transfer. Efek dari setiap kontroler (K_p , K_i , K_d) dalam sistem loop tertutup diperlihatkan pada tabel 2.1.

Tabel 2.1 Efek dari setiap kontroler (K_p , K_i , K_d) dalam loop tertutup

Respon Loop tertutup	Waktu naik	Over-shoot	Waktu turun	Kesalahan keadaan tunak
K_p	Menurun	Meningkat	Perubahan kecil	Menurun
K_i	Menurun	Meningkat	Meningkat	Hilang
K_d	Perubahan Kecil	Menurun	Menurun	Perubahan kecil

Penjelasan mengenai PID Controller dapat dijabarkan dalam bentuk *pseudo code*. *Pseudo code* dapat memberikan gambaran yang jelas tentang alur program (Rieper, et. al., 2009).

Pseudo code algoritma PID Controller ditunjukkan oleh gambar 2.11 dibawah ini.

```

previous_error = setpoint - process_feedback
integral = 0
start:
  wait(dt)
  error = setpoint - process_feedback
  integral = integral + (error*dt)
  derivative = (error - previous_error)/dt
  output = (Kp*error) + (Ki*integral) + (Kd*derivative)
  previous_error = error
  goto start

```

Gambar 2.11 Pseudo Code PID Controller

2.7 Algoritma Line Follower

Line Follower Robot adalah robot yang dapat berjalan mengikuti sebuah lintasan dalam bentuk garis, ada yang menyebutnya dengan Line Tracker, Line Tracer dan sebagainya. Garis yang digunakan adalah garis berwarna hitam dan berada diatas permukaan berwarna putih atau sebaliknya. Untuk dapat berjalan mengikuti garis diperlukan komponen berupa sensor cahaya. Dengan sensor ini, robot dapat mendeteksi perubahan intensitas cahaya berupa warna gelap maupun terang.

Permasalahan line follower adalah permasalahan klasik yang paling banyak diterapkan pada robot. Hal tersebut nampak dari banyaknya perlombaan-perlombaan robot cerdas yang diadakan yang bertemakan line follower. Untuk menyelesaikan permasalahan line follower atau line tracer dapat menggunakan dua buah sensor cahaya.

Namun sebelum robot dapat mendeteksi warna hitam maupun putih, perlu dilakukan kaliberasi cahaya terlebih dahulu. Kaliberasi dilakukan untuk menentukan batas antara warna hitam dan putih. Rumus yang digunakan untuk mengkaliberasi cahaya adalah :

Batas Hitam_Putih = (Nilai pada bid. Putih + Nilai pada bid.Hitam) / 2

Setelah kalibrasi dilakukan, lalu dideskripsikan nilai hitam dan putih.
Sebagai contoh :

Misalkan,

Nilai Batas Hitam_Putih = 45

Maka,

Hitam = Sensor.readValue < Batas Hitam_Putih

Putih = Sensor.readValue > Batas Hitam_Putih

Selanjutnya digunakan algoritma line following agar robot dapat mengikuti garis.Pseudo code algoritma line following dengan dua sensor cahaya adalah sebagai berikut.

```

If (Sensor_right on black line and Sensor_left on white)
{
    Forward (left_motor);
    Off (right_motor);
    Wait (time);
}
Else if (Sensor_right on white and Sensor_left on black line)
{
    Forward (right_motor);
    Off (left_motor);
    Wait (time);
}
Else if(Sensor_right on white and Sensor_left on white)
{
    Forward (right_motor + left_motor);
}

```

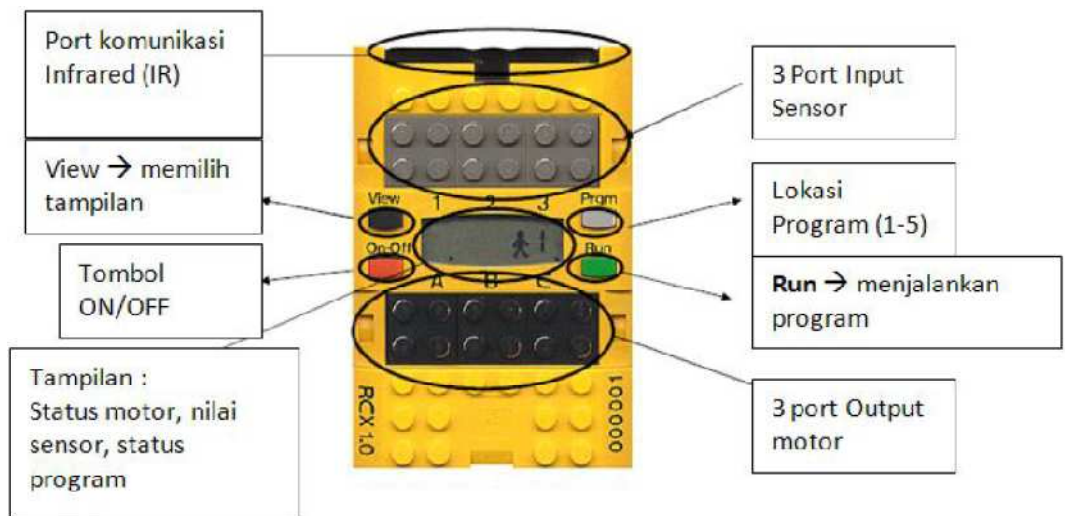
2.8 Lego Mindstorm

Lego Mindstorm adalah robot kit keluaran perusahaan Lego yang dapat dirakit sendiri. Seperti permainan bongkar pasang Lego lainnya, Lego Mindstorm juga terdiri dari bagian-bagian kecil yang dapat dirangkai atau dirakit sesuai daya kreatifitas masing-masing individu. Lego Mindstorm juga menambahkan NXT Brick yaitu sejenis komputer yang dapat diprogram. Dengan kelebihan tersebut, maka robot kit dari Lego menjadi alternatif permainan anak yang mendidik. Tidak hanya anak-anak saja yang menyukai permainan ini, bahkan orang dewasa pun sangat menyukainya.

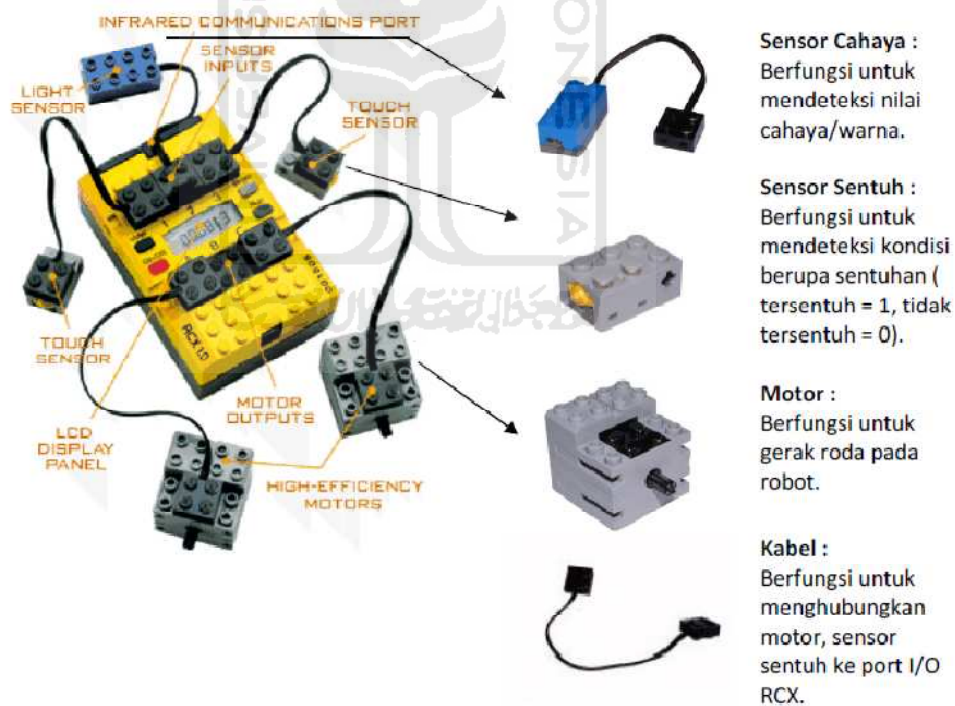
Dewasa ini perlombaan-perlombaan robot baik itu tingkat junior maupun senior telah banyak yang menggunakan paket robot kit dari Lego ini. Dari program sederhana seperti *line follower* hingga yang rumit seperti robot rubik solver telah dibuat dengan menggunakan robot lego. Hal tersebut menunjukkan bahwa Lego Mindstorm dapat menjadi sarana pendidikan yang menjanjikan.

Pada mulanya Lego meluncurkan seri robot kit yang diberi nama RCX (Robotic Command eXplorer). RCX ini berisi 8-bit Renesas H8/300 mikrokontroler sebagai CPU internal, 32K RAM yang menyimpan firmware dan program pengguna. RCX Brick diprogram dengan menggunakan bahasa pemrograman seperti C, C++, NXC, dan Java. Program yang sudah dibuat dengan menggunakan bahasa pemrograman tersebut lalu diupload ke RCX Brick. Jadi dapat dikatakan bahwa RCX Brick adalah otak pada robot lego tersebut (Very, B. 2010).

RCX mempunyai beberapa tombol fungsi dan beberapa *port input* dan *port output* (lihat pada gambar 2.12 dan gambar 2.13). Berikut penjelasan masing-masing tombol dan port pada RCX.

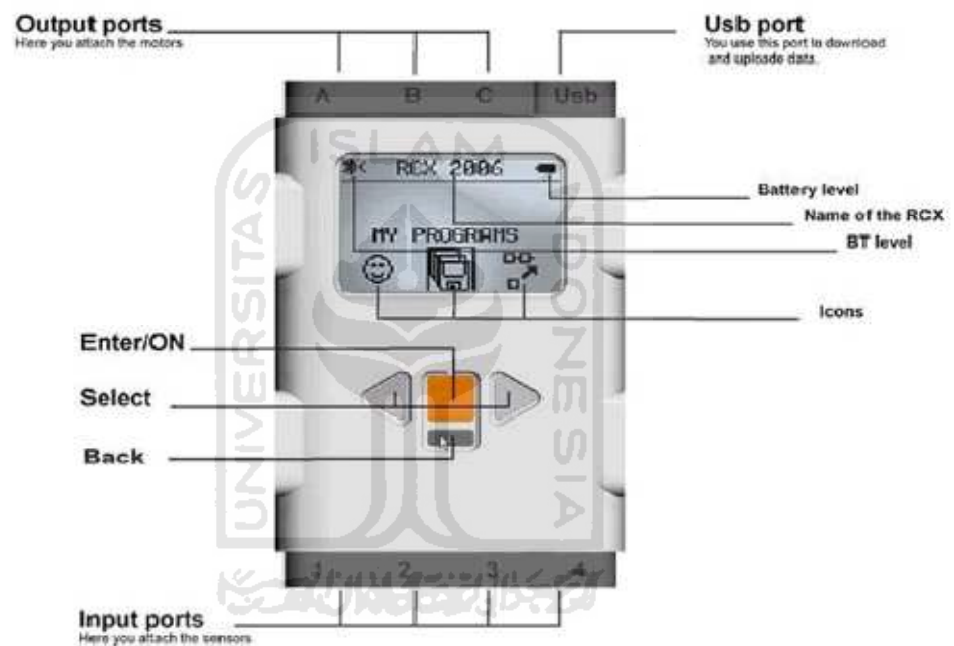


Gambar 2.12 Tombol Fungsi RCX

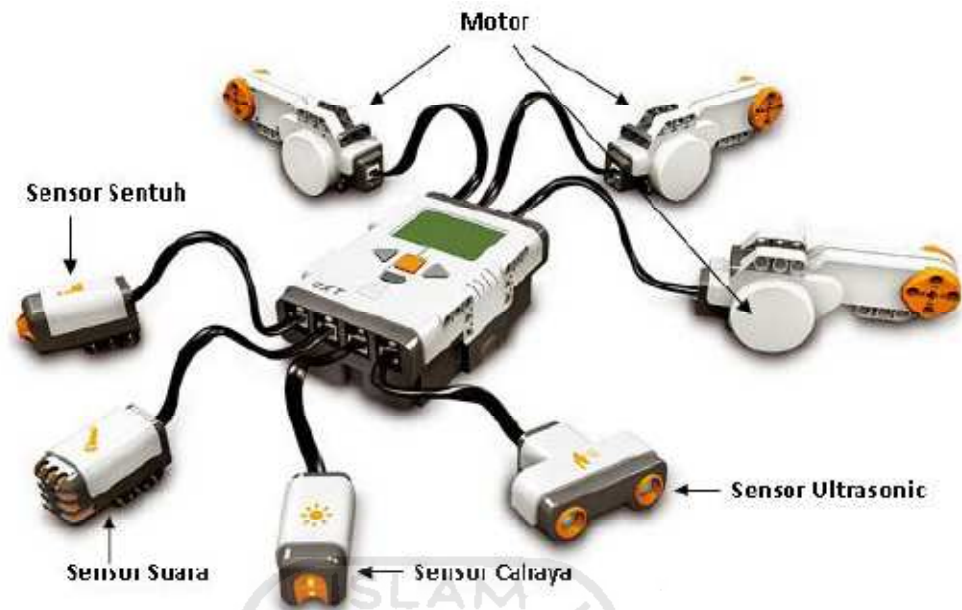


Gambar 2.13 Port RCX

Pada perkembangan selanjutnya muncullah NXT. NXT sama dengan RCX yang merupakan brick Lego yang dapat diprogram, yang merubah sebuah model menjadi robot dan mengontrol semua gerakan robot. Tidak berbeda halnya dengan RCX, NXT mempunyai beberapa tombol fungsi dan beberapa *port input* dan *port output* (lihat gambar 2.14 dan 2.15). Berikut penjelasan masing-masing fungsi tombol dan port pada NXT (Very, B. 2010).

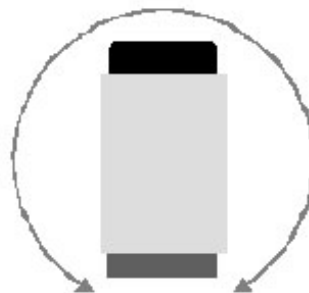


Gambar 2.14 Tombol Fungsi NXT



Gambar 2.15 Port NXT

Sensor Gyro adalah sensor yang berguna untuk mendeteksi dan mengukur derajat kemiringan atau rotasi pada robot. NXT Gyro Sensor memiliki satuan ukur yaitu *degree per second*. Sensor Gyro ini dapat mengukur $\pm 360^\circ$ per detik dan dapat membantu robot menyeimbangkan diri. Sensor Gyro ditunjukkan oleh gambar 2.16 dibawah ini.



Gambar 2.16 HiTechnic Gyro Sensor

2.9 Perkembangan Lego Segway Robot

Berikut ini akan dijelaskan mengenai perkembangan robot segway menggunakan Lego Mindstrom Kit.

2.9.1. LegWay (Segway Robot)- RCX dan Electro Optical Proximity Detectors

Steve Hassenplug berhasil membuat self balancing robot menggunakan Lego Mindstrom RCX brick dan dua Electro Optical Proximity Detectors (EOPD) sensor. Robot yang diberi nama LegWay ini memiliki kemampuan untuk menjaga keseimbangan dan mengikuti garis hitam.

LegWay menggunakan sensor EOPD untuk menjaga kestabilan dan kesimbangannya. Robot ini diprogram untuk bergerak maju jika jarak sensor dengan tanah berkurang dan sebaliknya akan bergerak mundur jika jarak sensor dengan tanah bertambah. Sensor EOPD bekerja dengan cara memancarkan cahaya dan akan terdeteksi jarak antara sensor dengan benda didepannya berdasarkan intensitas cahaya yang dikembalikan ke sensor. LegWay ditunjukkan oleh gambar 2.17 dibawah ini.



Gambar 2.17 Robot LegWay

2.9.2. NXTway (Segway Robot)- NXT dan Light Sensor

Percobaan robot segway selanjutnya adalah NXT way. Robot ini terinspirasi juga oleh Robot Legway buatan Steve Hassenplug. NXTway dibuat oleh Philippe Hurbain menggunakan NXT Brick dan satu buah sensor

cahaya.NXTway mampu seimbang untuk beberapa waktu saja.NXTway ditunjukkan oleh gambar 2.18 dibawah ini.

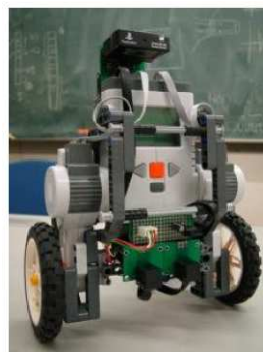


Gambar 2.18 NXTway Robot

2.9.3. NXTway-G – NXT dan Gyroscope

Percobaan robot berikutnya adalah NXTway-G karya Ryo Watanabe. Berbeda dari dua segway sebelumnya, Ryo Watanabe tidak menggunakan sensor cahaya maupun sensor EOPD melainkan menggunakan sensor giroskopik. Sensor ini dapat mendeteksi perubahan sudut atau angle.

Dengan perhitungan yang rumit, Ryo Watanabe mampu menciptakan robot yang stabil dan dapat dikendalikan dengan remot kontrol. Beliau menggunakan software RobotC sebagai *programming environment* dan menggunakan bahasa pemrograman yaitu bahasa C. NXTway-G ditunjukkan oleh gambar 2.19.



Gambar 2.19 NXTway-G

2.10 Bahasa Pemrograman

Lego Mindstorm memberikan kebebasan bagi pengembangnya untuk memilih bahasa pemrograman yang akan digunakan. Ada berbagai macam bahasa pemrograman yaitu diantaranya bahasa C, C++, dan Java. Masing-masing bahasa pemrograman tersebut dapat diterapkan kedalam robot Lego. Perbedaannya adalah terdapat pada firmware yang digunakan.

Bahasa pemrograman Java untuk Lego adalah bahasa program yang dikembangkan oleh pihak ketiga yang disebut The LeJOS Team. Mereka membuat bahasa pemrograman yang berbasis pemrograman berorientasi objek. Untuk dapat menggunakan bahasa ini, dibutuhkan firmware LeJOS NXJ yang diupload ke dalam NXT Brick. Software atau Editor yang digunakan yaitu Eclipse.

Lejos NXJ firmware adalah Java Virtual Machine (JVM) yang diperkenalkan pada Lego NXT Brick pada tahun 2006. Di dalam paket LeJOS NXJ sudah berisi library-library Java.

Beberapa keuntungan atau kelebihan LeJOS NXJ diantaranya adalah :

1. Merupakan object-oriented programming yang memberikan kemudahan dalam membuat class-class yang terpisah sehingga program lebih mudah dipahami.
2. Bersifat open source. Seperti halnya Linux, LeJOS juga bisa dikembangkan oleh banyak orang.
3. Dapat diimplementasikan kedalam Integrated Development Environment seperti Eclipse dan Netbeans. LeJOS dibuat menjadi plugin yang dapat diinstall pada Eclipse dan Netbeans.
4. Full support Bluetooth.
5. Support third party sensor seperti HiTechnic gyro sensor.
6. Terdapat behavior class yang membuat program menjadi mudah dipahami karena dapat dibagi menjadi kelas-kelas yang terpisah kemudian dapat dipanggil dengan arbitrator.

7. Support multithreading. Mutithreading berguna untuk menjalankan task-task secara paralel.

Mengingat dasar atau basic kemampuan programming yang dimiliki oleh penulis adalah java, maka diputuskan untuk menggunakan LeJOS NXJ sebagai bahasa program dalam membuat robot segway ini.



BAB III

METODOLOGI

3.1 Pendahuluan

Robot adalah alat mekanik yang terdiri dari berbagai komponen yang saling terhubung dan bekerja sama. Komponen pada robot meliputi *hardware*, *software*, dan program.

Hardware meliputi semua bagian fisik pada sebuah robot yang dapat berupa sensor, servo motor, microcontroller, dan body robot. *Software* adalah perangkat lunak pada robot yang berisi perintah-perintah program. Sedangkan program adalah inti dari sebuah robot.

Pada bab ini akan dijelaskan mengenai langkah-langkah apa saja yang perlu dilakukan dalam membuat Robot Segway Line Tracer. Dimulai dari merancang *hardware*, mempersiapkan *software*, sistem kontrol, dan merancang alur program.

3.2 Langkah-langkah Pembuatan

Langkah-langkah yang dilakukan dalam pembuatan robot segway line tracer adalah sebagai berikut :

1. Merancang Hardware

Komponen *hardware* adalah komponen dasar yang ada pada sebuah robot. Untuk itu hal pertama yang harus dipersiapkan dalam membuat robot adalah merancang *hardware*. Robot biasanya terdiri dari sensor, servo motor, dan microcontroller. Robot Segway Line Tracer pada proyek tugas akhir ini menggunakan paket robot kit

buatan perusahaan Lego yang dikenal dengan nama Lego NXT Mindstorm.

Lego NXT Mindstorm adalah robot kit yang dapat dirakit sendiri sesuai daya kreatifitas si pembuat. Lego NXT Mindstorm berisi sensor yang berfungsi sebagai alat pendeteksi lingkungan sekitar. Ada berbagai macam jenis sensor pada Lego NXT Mindstorm, diantaranya sensor ultrasonic atau sonar, sensor cahaya, sensor sentuh, sensor suara dan sensor gyro. Sensor-sensor tersebut memiliki fungsi yang berbeda-beda. Sebagai contoh apabila ingin mendeteksi jarak antara robot dengan dinding pembatas, maka sensor yang digunakan adalah sensor ultrasonic.

Robot Segway Line Tracer ini adalah robot yang unik karena hanya menggunakan dua roda bersisian sebagai mekanisme gerakannya. Dilihat dari desain body pada robot, sudah jelas bahwa robot ini tidak seimbang. Maka dari itu dibutuhkan sebuah sensor keseimbangan yang disebut sebagai sensor gyro.

Berikut ini adalah komponen hardware yang dibutuhkan dalam membuat Robot Segway Line Tracer.

- a. 1 buah NXT Brick
- b. 2 buah servo motor
- c. 1 buah sensor gyro
- d. 2 buah sensor cahaya
- e. 5 buah kabel konektor
- f. 1 paket Lego

NXT Brick berfungsi sebagai otak pada robot yang dapat dimasukkan perintah-perintah program. NXT Brick memiliki spesifikasi yaitu :

- a. 32-bit ARM7 microprocessor

- b. 1 USB 2.0 port
- c. 4 port sensor
- d. 3 port motor
- e. Support for Bluetooth wireless communication
- f. Powered by 6AA(1.5v) batteries atau lithium Rechargeable Battery



Gambar 3.1 NXT Brick

Servo motor berfungsi sebagai motor penggerak. Terdiri dari gear-gear roda yang saling terhubung. Gambar 3.2 menunjukkan bentuk servo motor.



Gambar 3.2 Servo Motor

Sensor Gyro berfungsi sebagai sensor keseimbangan. Sensor ini akan mendeteksi perubahan kemiringan atau rotasi sebesar +/- 360° per detik. Gambar 3.3 menunjukkan bentuk Sensor Gyro.



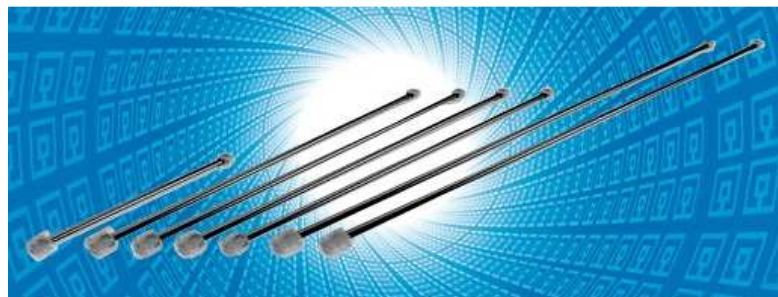
Gambar 3.3 Sensor Gyro

Sensor cahaya berfungsi sebagai alat pembeda gelap dan terang. Sensor ini mendeteksi perbedaan intensitas cahaya yang dipantulkan oleh benda disekitarnya. Sensor ini sangat cocok digunakan pada robot line follower atau line tracer. Gambar 3.4 menunjukkan bentuk sensor cahaya.



Gambar 3.4 Sensor Cahaya

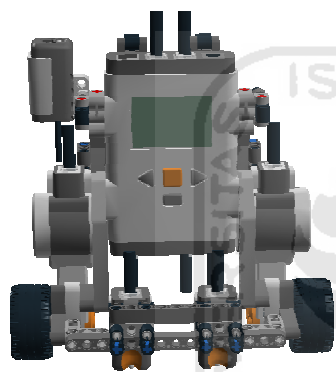
Kabel konektor berfungsi menghubungkan sensor-sensor dan servo motor ke NXT Brick. Gambar 3.5 menunjukkan bentuk kabel konektor NXT.



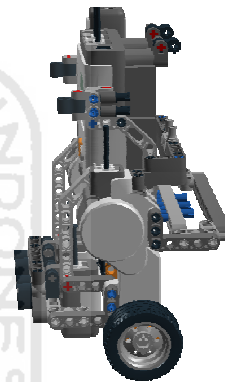
Gambar 3.5 Kabel Konektor NXT

Lego Digital Designer (LDD) adalah sebuah software yang dibuat oleh perusahaan Lego yang berguna untuk mendesain bentuk robot. Software keluaran Lego tersebut memang dibuat khusus untuk memberikan gambaran visual bentuk robot sebelum merakit robot tersebut. Software Lego Digital Designer terdapat dalam CD Laporan Tugas Akhir ini.

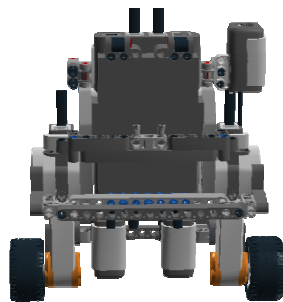
Gambar 3.6 menunjukkan bentuk body robot yang telah selesai dibuat menggunakan Lego Digital Designer (LDD).



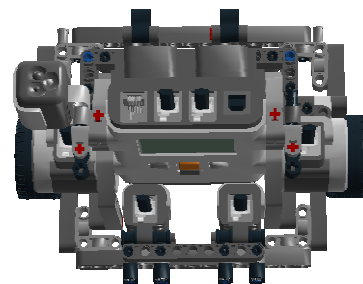
Gambar 3.6(a) Robot Tampak
Depan



Gambar 3.6(b) Robot Tampak
Samping



Gambar 3.6(c) Robot Tampak
Belakang

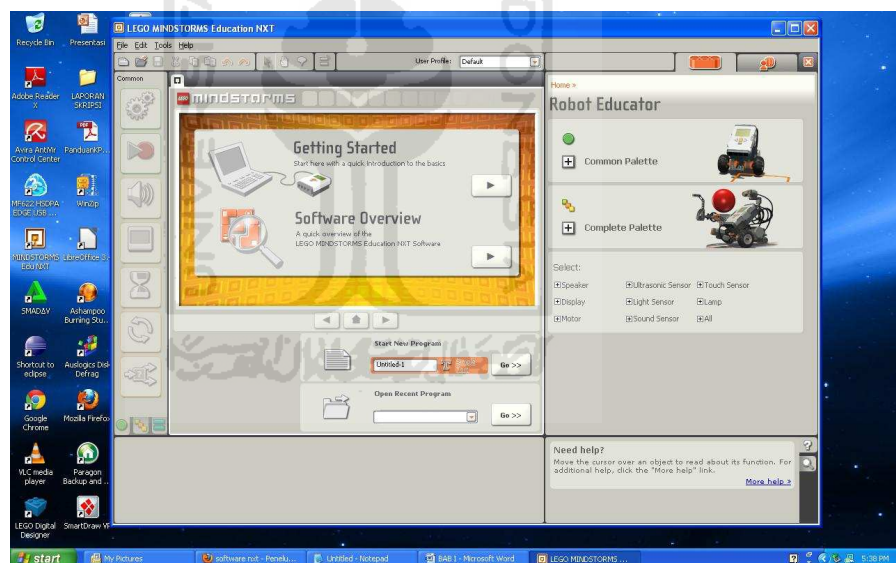


Gambar 3.6(d) Robot Tampak Atas

2. Mempersiapkan Software

Tahap kedua adalah mempersiapkan software pendukung. Banyak sekali software-software untuk membuat program pada NXT Mindstrom. Mulai dari software berbasis GUI yang dikenal dengan Lego NXT Mindstorm Edu sampai software yang berbasis bahasa pemrograman java.

Lego Mindstorm Edu diperuntukan bagi kalangan pemula atau yang tidak mengerti bahasa pemrograman. Kelebihan dari software ini adalah tampilan yang menarik dan mudah dimengerti. Pengguna cukup melakukan *drag and drop* gambar yang merupakan perlambangan program. Berikut ini adalah tampilan GUI Lego Mindstorm Edu.



Gambar 3.7 Lego NXT Mindstorm Edu

Namun pada tugas akhir ini bahasa pemrograman yang digunakan adalah java. Program dibuat menggunakan language editor yaitu Eclipse dan LeJOS firmware.



Gambar 3.8 Logo LeJOS

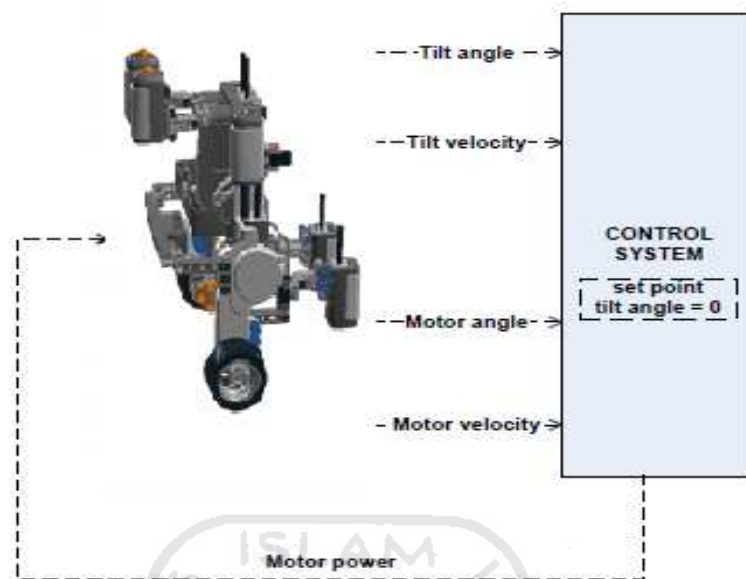


Gambar 3.9 Logo Eclipse

3. Merancang Sistem Kontrol

Tahap ketiga adalah merancang sistem kontrol. Sistem kontrol adalah keseluruhan sistem pada robot yang mengatur gerak robot. Jika dilihat dari struktur fisiknya, robot segway ini akan sulit untuk berdiri tegak seimbang. Untuk itu diperlukan sebuah proses perhitungan atau kalkulasi yang tepat agar robot dapat berdiri tegak dan seimbang.

Sistem kontrol akan mendeteksi kemiringan robot dan akan membuat robot seimbang dengan cara memberikan jumlah power yang sesuai kepada servo motor. Proses pada sistem kontrol ditunjukkan oleh gambar 3.10.



Gambar 3.10 Sistem Kontrol

Pada gambar tersebut dijelaskan bahwa pada saat robot diaktifkan dan robot tersebut mulai bergerak, maka robot akan terjatuh kedepan atau kebelakang yang disebabkan adanya berat body robot yang ditarik oleh gaya gravitasi.

Dari pergerakan maju atau mundur tersebut akan didapat beberapa nilai yaitu *tilt angle*, *tilt velocity*, *motor angle*, dan *motor velocity*. Masing-masing nilai yang didapat tersebut akan diolah oleh sistem untuk mendapatkan nilai *motor power* yang sesuai agar robot dapat bergerak menyeimbangkan diri.

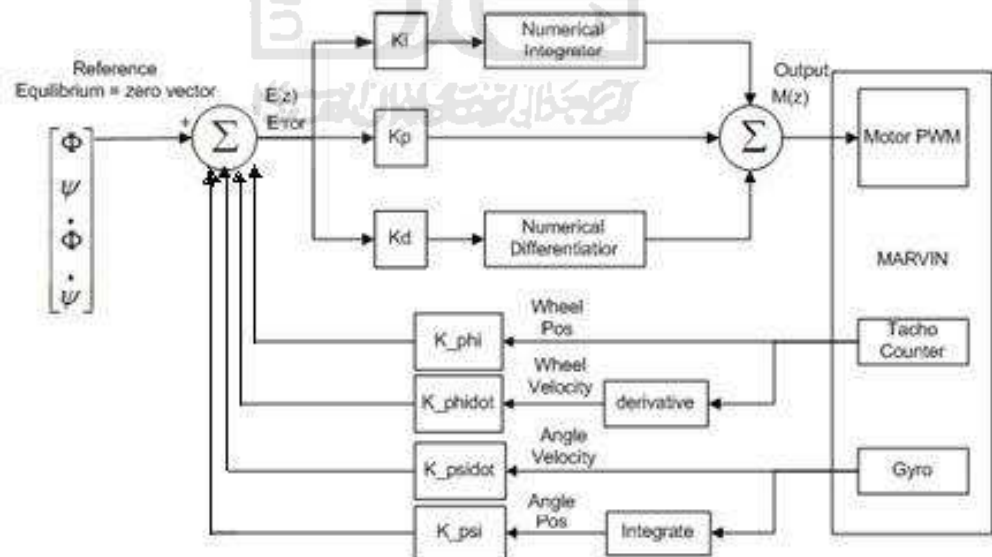
Sistem kontrol yang terdapat pada robot segway ini adalah menggunakan konsep PID Controller. Konsep ini digunakan karena memiliki kemampuan untuk mengolah *motor power* sehingga robot dapat stabil.

Nilai *output* atau keluaran dari sistem kontrol adalah berupa nilai power dalam *pulse moded*an mengandung nilai error. Jika nilai

output tersebut langsung dikirimkan ke servo motor tanpa diolah terlebih dahulu, maka akan menyebabkan robot kehilangan keseimbangan. Untuk itu dibutuhkan PID Controller didalam sistem kontrol.

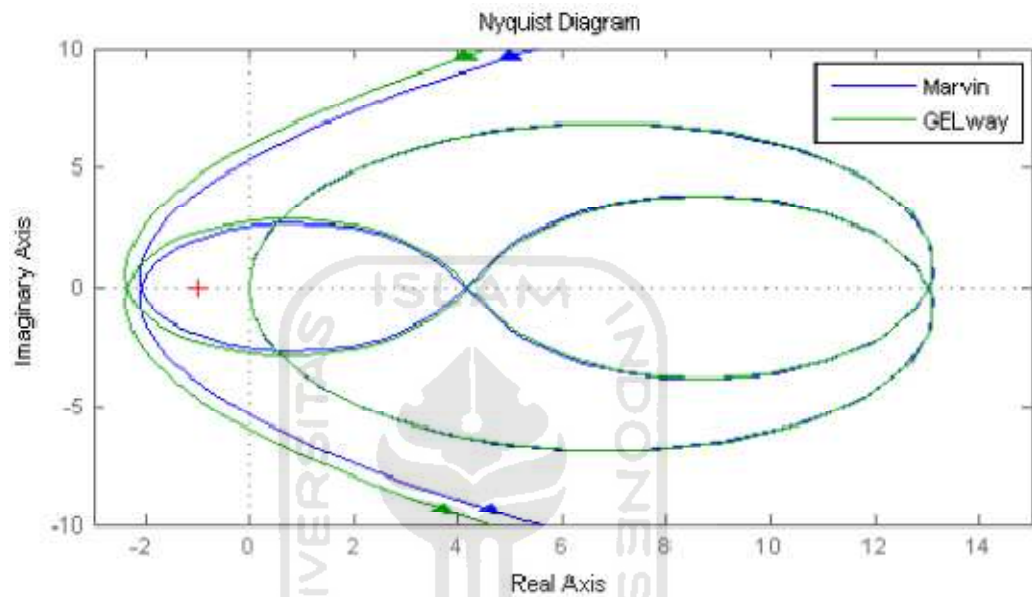
PID Controller terdiri dari tiga komponen yaitu proporsional, integral, dan diferensial. Masing-masing komponen tersebut memberikan dampak atau efek yang berbeda bagi sistem. Proporsional kontroler (K_p) akan memberikan efek mengurangi waktu naik, tetapi tidak menghapus kesalahan keadaan tunak. Integral kontroler (K_i) akan memberikan efek menghapus kesalahan keadaan tunak, tetapi berakibat memburuknya respon transien. Diferensial kontroler (K_d) akan memberikan efek meningkatnya stabilitas sistem, mengurangi over-shoot, dan menaikkan respon transfer.

Sistem kontrol dengan PID Controller dalam loop tertutup ditunjukkan oleh gambar 3.11 dibawah ini.



Gambar 3.11 Sistem Kontrol dengan PID Controller

Menentukan nilai K_p , K_i , dan K_d adalah tugas yang sulit dan membutuhkan pengetahuan tentang teorema Nyquist. Teorema Nyquist dicetuskan oleh Harry Theodor Nyquist. Teorema Nyquist merupakan semi-graphical method yang dapat digunakan untuk mengecek kestabilan sistem. Diagram Nyquist ditunjukkan oleh gambar 3.12.

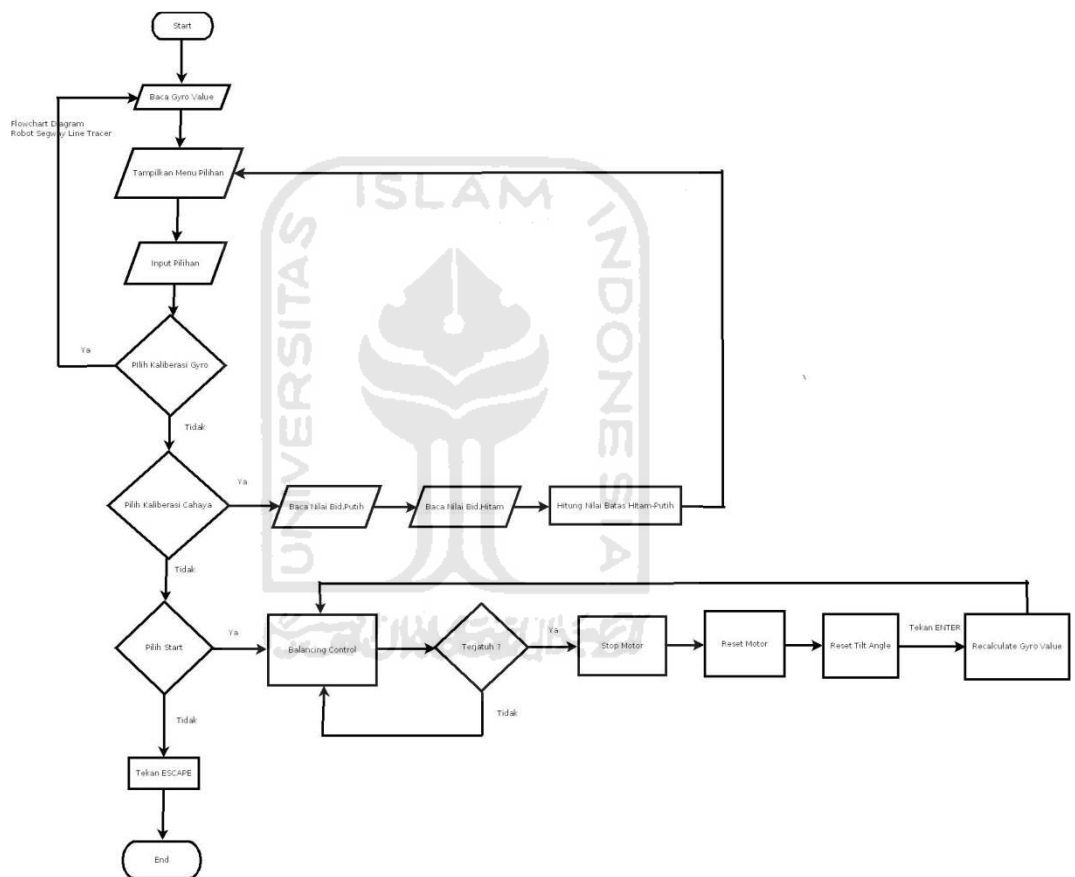


Gambar 3.12 Diagram Nyquist

Nilai konstanta pada pembuatan robot segway ini didapat dari percobaan serupa yaitu RobotMarvin karya Johnny Rieper, Bent Brisballe Nyeng, dan Kasper Sohn. Kemudian nilai konstanta tersebut disesuaikan dengan kondisi robot dengan menggunakan metode *trial and error*. Berdasarkan proses *trial and error* didapat hasil nilai konstanta $K_p = 1.2$, konstanta $K_i = 0.25$, konstanta $K_d = 0.1$.

4. Merancang Alur Program

Alur program pada pembuatan Robot Segway Line Tracer ini menggunakan diagram flowchart. Flowchart terdiri dari bagan-bagan arus yang menggambarkan algoritma program. Alur program ditunjukkan oleh gambar 3.13 dibawah ini.



Gambar 3.13 Flowchart Alur Program

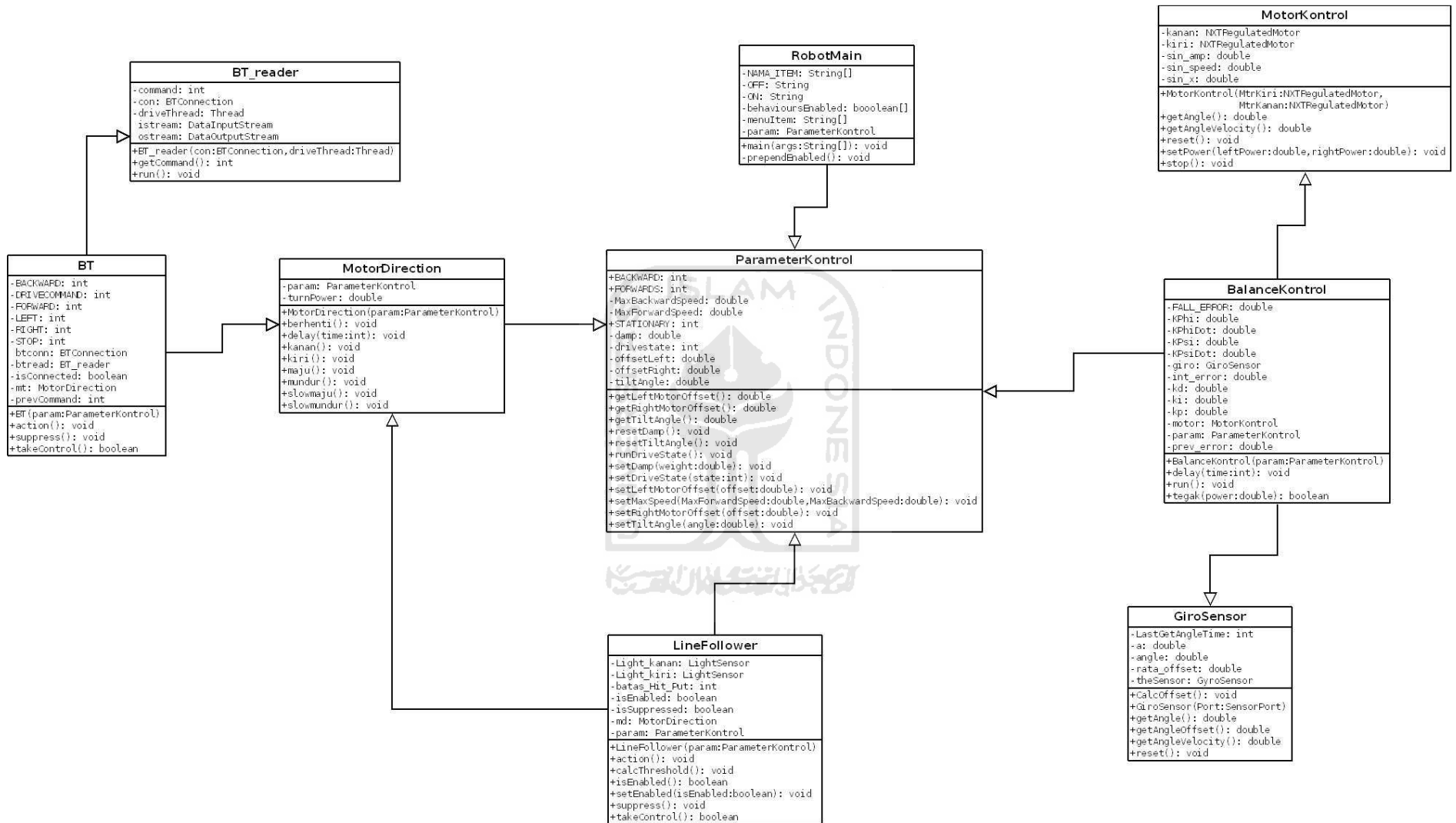
Penjelasan alur program diatas adalah sebagai berikut :

1. Robot dinyalakan dan program mulai berjalan.
2. Robot akan membaca nilai gyro yang dihasilkan oleh sensor gyro, lalu tekan ENTER.
3. Muncul tampilan menu utama yang terdiri dari :

- a. Start
- b. Kaliberasi Gyro
- c. Kaliberasi Cahaya
4. Jika memilih Kaliberasi Gyro, maka robot akan melakukan kaliberasi ulang nilai gyro.
5. Jika memilih Kaliberasi Cahaya, maka sensor cahaya akan menyala dan meminta masukan berupa nilai intensitas cahaya pada bidang putih dan hitam. Selanjutnya dilakukan proses perhitungan nilai batas hitam_putih.
6. Jika memilih Start, maka proses keseimbangan dijalankan.
7. Saat robot terjatuh, robot akan mereset motor power dan nilai gyro.
8. Tekan ENTER untuk melanjutkan proses keseimbangan.
9. Tekan ESCAPE untuk menyelesaikan proses.
10. Robot berhenti.

3.3 Struktur Program

Diagram Kelas (*class diagram*) menunjukkan keberadaan kelas-kelas dan hubungan antar kelas. Diagram kelas dapat merepresentasikan seluruh atau sebagian struktur kelas di sistem (Hariyanto, B. 2010. *Esensi-esensi Bahasa Pemrograman Java*. Bandung: Informatika Bandung). Gambar 3.14 berikut ini menjelaskan struktur program yang disajikan dalam bentuk diagram kelas.

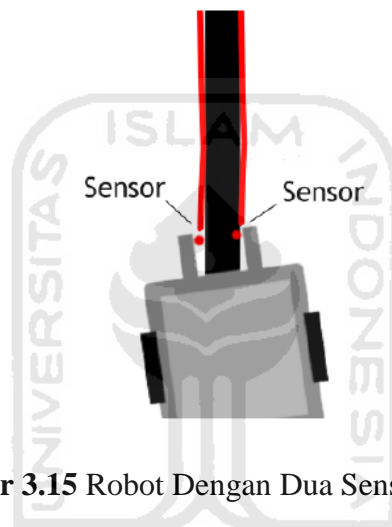


Gambar 3.14 Diagram Kelas

3.4 Desain Arena Line Follower

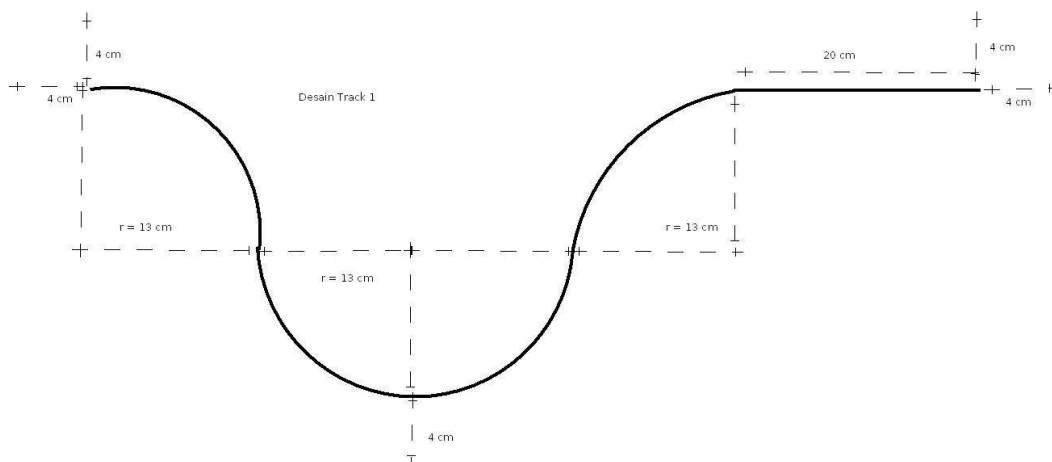
Pada subbab berikut ini akan dijelaskan tentang gambaran bentuk arena line follower yang akan digunakan untuk pengujian robot. Arena tersebut dibuat menggunakan selotip hitam sebagai garis alur yang harus dilalui oleh robot dan kertas putih besar sebagai bidang alasnya.

Robot segway line tracer ini menggunakan dua sensor cahaya sebagai sensor pendeteksi garis. Gambar robot ditunjukkan oleh gambar 3.15.

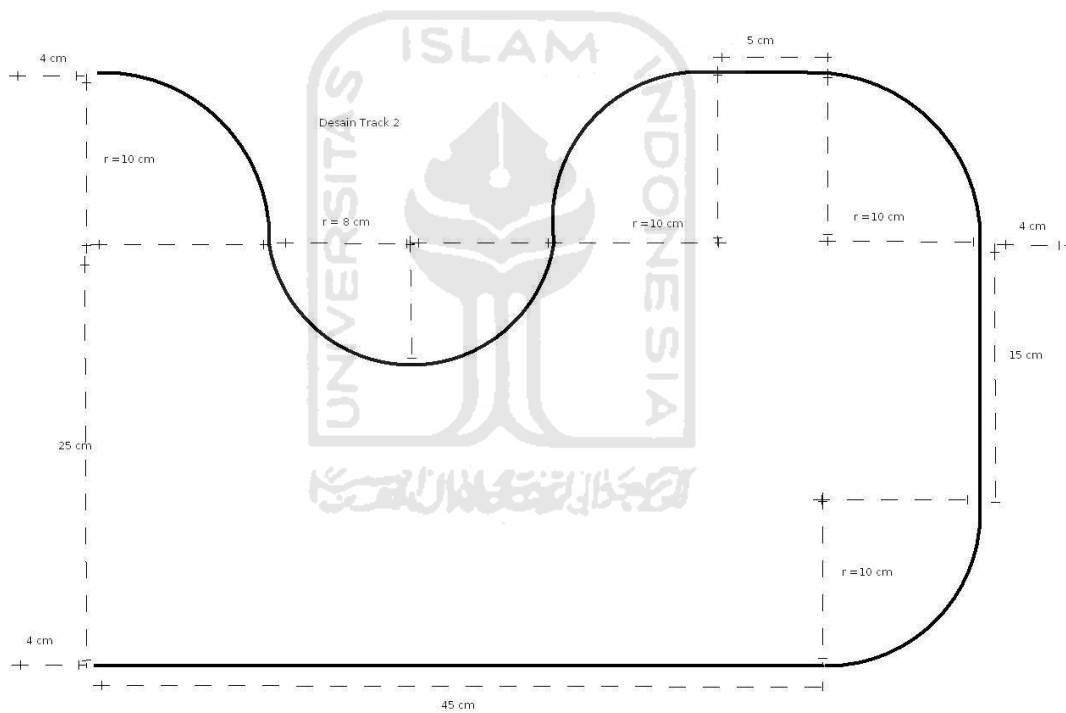


Gambar 3.15 Robot Dengan Dua Sensor Cahaya

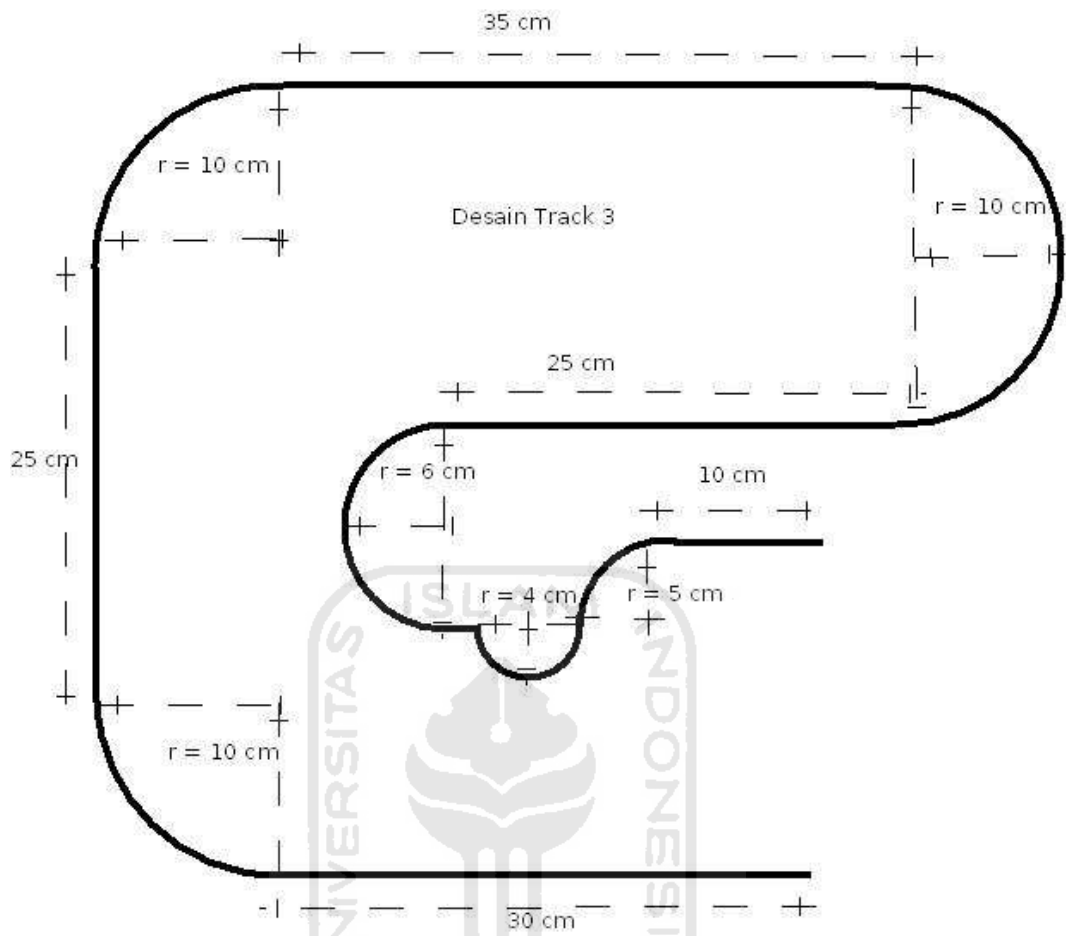
Untuk pengujian robot akan digunakan jenis arena *line follower* sederhana. Robot nantinya akan ditempatkan diatas arena tersebut dan mencoba berjalan mengikuti garis. Arena 1 dibuat dengan panjang 101.68 cm dan merupakan arena dengan kategori mudah karena memiliki panjang lintasan yang paling pendek dibandingkan dengan arena lainnya, sudut belokan yang besar serta memiliki jumlah belokan sebanyak empat buah. Arena 2 memiliki panjang 152.92 cm dan merupakan arena dengan kategori sedang karena memiliki jumlah belokan yang lebih banyak dibandingkan dengan arena 1. Arena 3 dibuat dengan panjang 227.05 cm dan merupakan arena dengan kategori sulit karena arena ini merupakan arena terpanjang dan memiliki besar sudut dan jumlah belokan yang paling banyak. Gambar 3.16, 3.17 dan 3.18 menunjukkan desain arena robot yang akan digunakan.



Gambar 3.16 Desain Track 1



Gambar 3.17 Desain Track 2



Gambar 3.18 Desain Track 3

BAB IV

HASIL DAN PEMBAHASAN

4.1 Deskripsi Kelas

Pada bab sebelumnya telah dijabarkan diagram kelas dan hubungannya. Pada bab ini, akan dijelaskan lebih detail mengenai kelas-kelas yang ada pada diagram tersebut.

4.1.1. RobotMain

Kelas RobotMain.java merupakan kelas main yang berfungsi menginisialisasi BalanceKontrol (keseimbangan) dan behavior robot (perilaku robot). Terdapat pula button listener yang berguna untuk keluar dari program sewaktu-waktu. Potongan *source code* program ditunjukkan oleh gambar 4.1 dan gambar 4.2 dibawah ini.

```
public static void main(String[] args) {  
  
    Button.ESCAPE.addActionListener(new ButtonListener() { /*Button  
listener*/  
  
        @Override  
        public void buttonReleased(Button b) {  
            // TODO Auto-generated method stub  
        }  
  
        @Override  
        public void buttonPressed(Button b) {  
            // TODO Auto-generated method stub  
            Sound.buzz();  
            System.exit(0);  
        }  
    });  
}
```

Gambar 4.1 Source Code-1 RobotMain

Pada gambar 4.1 terdapat method button listener yang berguna untuk keluar dari program sewaktu-waktu. Yang dijadikan sebagai button listener adalah tombol “ESCAPE”. Jadi saat tombol “ESCAPE” ditekan maka robot akan berhenti bergerak disertai bunyi peringatan “buzz”.

```

/*inisialisasi BalanceKontrol & Parameter*/
param = new ParameterKontrol();
BalanceKontrol bk = new BalanceKontrol(param);

/*Behavior*/
Behavior b1 = new LineFollower(param);
Behavior b2 = new BT(param);
Behavior[] bArray = {b1, b2};
Arbitrator arby = new Arbitrator(bArray);

//Menu pada layar
prependEnabled(); //tampilan on/off
TextMenu menu = new TextMenu(menuItem, 1, "MENU");

```

Gambar 4.2 Source Code-2 RobotMain

Pada gambar 4.2 terdapat fungsi behavior dan arbitrator. Behavior merupakan sebuah fungsi yang membuat program menjadi mudah dipahami karena dapat dibagi menjadi beberapa kelas yang terpisah kemudian baru dipanggil di arbitrator. Sedangkan arbitrator merupakan sebuah kontrol yang mengatur behavior mana yang harus aktif pada behavior control system.

Pada program terdapat dua behavior yaitu “LineFollower”, dan “BT”. Behavior “LineFollower” berisi baris perintah linefollower yang membuat robot dapat mengikuti garis. Sedangkan behaviour “BT” berisi perintah koneksi bluetooth yang membuat robot dapat dikendalikan dengan remot kontrol. Arbitrator bertugas mengontrol behavior mana yang harus aktif diantara beberapa behavior yang ada tersebut.

4.1.2. BalanceKontrol

Kelas BalanceKontrol.java merupakan kelas yang berisi fungsi-fungsi PID yang berguna untuk mengatur power yang akan dikirimkan ke servo motor. Modul BalanceKontrol ini akan membaca nilai *angle* dan *angle velocity* yang dihasilkan oleh sensor gyro dan servo motor dan menggunakan nilai tersebut sebagai nilai bobot. Nilai bobot yang dihasilkan akan digunakan untuk mendapatkan nilai *error*, *integral error* dan *derivative error*. Setelah itu akan dihitung nilai power dengan memakai rumus PID Controller. Bobot konstanta PID Controller ditunjukkan oleh Tabel 4.1.

Tabel 4.1 Nilai Bobot Konstanta PID Controller

Nama Konstanta	Nilai Bobot
Kp	1.2
Ki	0.25
Kd	0.1
KPhi	0.806876
KPhiDot	0.039711
KPsi	44.257035
KPsiDot	0.620882

Potongan source code program :

```

public class BalanceKontrol extends Thread {
-
-
-
public void run(){

    //proses setelah menekan enter
    while (true){
        delay(6);
        param.runDriveState();//utk mendapatkan nilai tiltangle,jika (+
maju) (-mundur) (0 diam)

        double Psi = giro.getAngle(); //nilai angle
dari giro dlm satuan degree
        double PsiDot = giro.getAngleVelocity(); //nilai angle velocity
dari giro dlm satuan degree/second

        double Phi = MotorKontrol.getAngle() -
ParameterKontrol.getTiltAngle(); //degree
        double PhiDot = motor.getAngleVelocity(); //nilai velocity(kec)
motor--degree/second

        double error = Psi * KPsi + PsiDot * KPsiDot + Phi * KPhi +
PhiDot * KPhiDot ;
        int_error += error; //integral error
        double deriv_error = error - prev_error; //derivative error
        prev_error = error;

        double pw = error * kp + deriv_error * kd + int_error * ki;
//power
        if (!tegak (pw)){ //ketika robot terjatuh maka akan mereset
semua nilai
            continue;
        }
        //LCD.drawString("pw "+pw, 3, 0);
        motor.setPower(pw + param.getLeftMotorOffset(), pw +
param.getRightMotorOffset());
    }
}
-
-
-
}

```

Gambar 4.3Source Code BalanceKontrol

Pada potongan *source code* diatas terdapat proses pengaturan keseimbangan dimana proses dilakukan dalam loop tertutup.Terdapat beberapa variable yaitu Psi, PsiDot, Phi, PhiDot, error, int_error, derive_error, prev_error dan pw. Variabel Psi, PsiDot, Phi dan PhiDot merupakan hasil perhitungan sudut dan kecepatan sudut yang dihasilkan oleh sensor gyro dan servo motor. Sedangkan error, derive_error dan int_error adalah nilai yang

akan dikalikan dengan konstanta PID sehingga menghasilkan bobot power motor yang sesuai agar robot dapat berdiri seimbang.

4.1.3. MotorKontrol

Kelas MotorKontrol.java merupakan kelas yang berisi method setPower. Method ini akan mengirimkan nilai power untuk diolah di BalanceKontrol. Di dalam kelas ini juga terdapat method getAngle dan getAngleVelocity yang memiliki fungsi membaca nilai angle dan angle velocity yang dihasilkan oleh servo motor.

Potongan source code program :

```

public class MotorKontrol {
-
-
-
public void setPower(double leftPower, double rightPower){
    sin_x += sin_speed ;

    int pwl = (int) (leftPower + Math.sin(sin_x)*sin_amp);
    int pwr = (int) (rightPower - Math.sin(sin_x)*sin_amp);

    kiri.setSpeed(pwl);
    if(pwl<0){
        kiri.backward();
    }else if(pwl>0){
        kiri.forward();
    }else{
        kiri.stop();//stop
    }

    kanan.setSpeed(pwr);
    if(pwr<0){
        kanan.backward();
    }else if(pwr>0){
        kanan.forward();
    }else {
        kanan.stop();//stop
    }
}

//nilai rata2 degree motor(sudut putaran motor)
public static double getAngle(){
    return (kiri.getTachoCount()+kanan.getTachoCount())/2.0;
}

//kec rata2 motor dlm degree/second
public double getAngleVelocity(){
    return (kiri.getRotationSpeed()+kanan.getRotationSpeed())/2.0;
}
}

```

Gambar 4.4 Source Code MotorKontrol

Pada potongan *source code* diatas terdapat variable *pwl* dan *pwr*. Variable tersebut merupakan nilai yang akan mengatur kecepatan atau *setSpeed* pada motor kiri dan motor kanan. Jika bernilai negatif maka motor akan bergerak mudur, jika bernilai positif maka motor akan bergerak maju, dan jika bernilai nol (0) maka motor akan berhenti. Sedangkan method *getAngle* dan *getAngleVelocity* akan menghitung nilai sudut dan kecepatan sudut yang dihasilkan akibat dari pergerakan motor.

4.1.4. GiroSensor

Kelas *GiroSensor.java* merupakan kelas yang berisi method untuk menghitung nilai gyro offset. Selain itu terdapat pula method *getAngle* dan *getAngleVelocity* yang memiliki fungsi membaca nilai angle dan angle velocity yang dihasilkan oleh gyro. Potongan source code program :

```
public class GiroSensor {
-
-
-
    /*
    * Method utk mengkalkulasi gyro offset
    */
    public static void CalcOffset(){
        rata_offset=0.0;
        int TotOffset = 0;
        LCD.clear();
        LCD.drawString("Hitung Gyro:", 0, 2);
        //membaca nilai gyro offset selama 100x
        for(int i=0;i<100;i++){
            TotOffset += theSensor.readValue();
            try {
                Thread.sleep(4);
            } catch (InterruptedException e) {
                // TODO Auto-generated catch block
            }
        }
        //menghitung rata2 nilai gyro offset yg dihasilkan
        rata_offset = Math.ceil((double) TotOffset / (double) 100) + 1.05;
        while(!Button.ENTER.isPressed()){
            LCD.drawString("Kalibrasi OK", 0, 4);
            LCD.drawString("offset : "+rata_offset, 0, 5);
            LCD.drawString("<Tekan ENTER>", 1, 6);
        }
        LCD.clear();
        try {
            Thread.sleep(500);
        } catch (InterruptedException e) {
            // TODO Auto-generated catch block
        }
    }
}
```

Gambar 4.5 Source Code-1 GiroSensor

```

public double getAngleOffset(){
    double offset = rata_offset * a + (1.0 - a) *
(double) theSensor.readValue();
    rata_offset = offset;
    return offset;
}
/**
 * get angle velocity (kec sudut) gyro dalam degree per second
 */
public double getAngleVelocity(){ //nilai angle velocity dalam
degree/second
    return theSensor.readValue() - getAngleOffset();
}
/**
 *rumus hitung sudut(angle) dan mengembalikan nilai angle dalam degree
 */
public double getAngle(){
    int now = (int) System.currentTimeMillis();
    int delta_t = now - LastGetAngleTime;

    if(delta_t != now){
        angle += getAngleVelocity() * delta_t /1000;
    }
    LastGetAngleTime = now;
    return angle;
}
}
}
}

```

Gambar 4.6Source Code-2 GiroSensor

Pada potongan source code-1 diatas terdapat method CalcOffset untuk mengkalkulasi nilai gyro offset dimana nilai tersebut merupakan representasi dari posisi tegak robot. Dari nilai offset hasil kalkulasi tersebut maka akan dikirimkan ke method getAngleOffset (lihat gambar 4.6) untuk mendapatkan nilai offset sudut. Setelah itu akan didapat nilai kecepatan sudut yang dihitung pada method getAngleVelocity.Rumus hitung sudut terdapat pada method getAngle.Pada method getAngle terdapat variable now dan delta_t dengan tipe data int, variable now adalah variable yang menghitung waktu dalam millisecond, sedangkan delta_t adalah selisih waktu.Fungsi dari penghitungan waktu tersebut adalah untuk mendapatkan nilai angle dalam degree.

4.1.5. ParameterKontrol

Kelas ParameterKontrol.java merupakan kelas yang berisi method-method yang menjadi parameter. Diantaranya adalah setTiltAngle, setLeftMotorOffset dan setRightMotorOffset. Parameter-parameter tersebut akan dipanggil oleh kelas-kelas yang lain. Jadi dapat dikatakan bahwa kelas ParameterKontrol merupakan persimpangan kelas-kelas dari keseluruhan program. Method setTiltAngle berguna untuk setting nilai angle, method setLeftMotorOffset dan setRightMotorOffset berguna untuk setting nilai offset pada motor kiri dan pada motor kanan. Potongan source code program:

```
public class ParameterKontrol {
-
-
-
    public synchronized void setTiltAngle(double angle) {
        ParameterKontrol.tiltAngle += angle;
    }
    public synchronized void setLeftMotorOffset(double offset){
        this.offsetLeft = offset;
    }
    public synchronized void setRightMotorOffset(double offset){
        this.offsetRight = offset;
    }
-
-
-
}
```

Gambar 4.7 Source Code ParameterKontrol

4.1.6. MotorDirection

Kelas MotorDirection.java merupakan kelas yang berisi method yang mendeskripsikan arah pergerakan robot, seperti berhenti, maju, mundur, belok kanan dan kiri. Method-method tersebut dapat dipanggil dengan melakukan inisialisasi terlebih dahulu. Potongan source code program :


```

public class MotorDirection {
-
-
-
    public void berhenti() {
        param.setLeftMotorOffset(0);
        param.setRightMotorOffset(0);
        param.resetDamp();
        param.setDriveState(ParameterKontrol.STATIONARY);
    }
    public void kiri() {
        param.setLeftMotorOffset(-turnPower);
        param.setRightMotorOffset(turnPower);
    }
}

```

Gambar 4.8Source Code-1 MotorDirection

```

    public void kanan() {
        param.setLeftMotorOffset(turnPower);
        param.setRightMotorOffset(-turnPower);
    }
    public void maju() {
        param.setLeftMotorOffset(0);
        param.setRightMotorOffset(0);
        param.setMaxSpeed(1.5,4);
        param.setDriveState(ParameterKontrol.FORWARDS);
    }
    public void mundur() {
        param.setLeftMotorOffset(0);
        param.setRightMotorOffset(0);
        param.setMaxSpeed(2.5,3);
        param.setDriveState(ParameterKontrol.BACKWARDS);
    }
}
-
-
-
}

```

Gambar 4.9Source Code-2 MotorDirection

4.1.7. LineFollower

Kelas LineFollower.java merupakan kelas yang berisi method follow line sehingga robot dapat berjalan mengikuti garis hitam pada permukaan berwarna putih. Supaya robot dapat membedakan antara warna hitam dan warna putih maka perlu dilakukan kalibrasi cahaya terlebih dahulu. Proses kalibrasi terdapat pada method calcThreshold dimana dalam method tersebut terdapat fungsi kalkulasi nilai intensitas cahaya pada bidang putih dan bidang hitam. Setelah itu proses linefollower terdapat pada method

action. Didalam method action tersebut berisi algoritma linefollower problem.

Potongan source code program :

```
public class LineFollower implements Behavior {
-
-
-
    public void action() {
        isSuppressed = false;
        Light_kiri.setFloodlight(true);
        Light_kanan.setFloodlight(true);

        boolean isLeftblack, isLeftWhite;
        boolean isRightblack, isRightWhite;
        while (!isSuppressed) {
            isLeftblack = Light_kiri.readValue() < batas_Hit_Put;
            isRightblack = Light_kanan.readValue() < batas_Hit_Put;

            isLeftWhite = Light_kiri.readValue() > batas_Hit_Put;
            isRightWhite = Light_kanan.readValue() > batas_Hit_Put;

            if (isLeftWhite && isRightWhite) { // putih && putih
                md.slowmaju();
            }
        }
    }
}
```

Gambar 4.10 Source Code-1 LineFollower

```
        else if (isLeftblack) { // Left = black
            param.setLeftMotorOffset(-180);
            param.setRightMotorOffset(180);
        }
        else if (isRightblack) { // Right = black

            param.setLeftMotorOffset(180);
            param.setRightMotorOffset(-180);

        } else if (isLeftblack && isRightblack) {
            md.slowmaju();
        }
        try {
            Thread.sleep(10);
        } catch (InterruptedException e) {}
    }
    Light_kiri.setFloodlight(false);
    Light_kanan.setFloodlight(false);
}
```

Gambar 4.11 Source Code-2 LineFollower

```

public static void calcThreshold() throws Exception {

    int whiteSum = 0;
    int blackSum = 0;
    LCD.clear();
    Light_kiri.setFloodlight(true);
    Light_kanan.setFloodlight(true);

    LCD.drawString("Letakkan sensor", 0, 1);
    LCD.drawString("pada bidang", 0, 2);
    LCD.drawString("putih dan", 0, 3);
    LCD.drawString("tekan Tombol Kanan", 0, 4);
    Thread.sleep(400);
    Button.RIGHT.waitForPressAndRelease();

    for(int i = 0; i < 10; i++) {
        whiteSum += Light_kiri.readValue() +
Light_kanan.readValue();
        Thread.sleep(10);
    }

    Sound.beep();
    LCD.drawString("Letakkan sensor", 0, 1);
    LCD.drawString("pada garis", 0, 2);
    LCD.drawString("hitam dan", 0, 3);
    LCD.drawString("tekan tombol Kanan", 0, 4);
    Button.RIGHT.waitForPressAndRelease();

    for(int i = 0; i < 10; i++) {
        blackSum += Light_kiri.readValue() +
Light_kanan.readValue();
        Thread.sleep(10);
    }

    Sound.twoBeeps();
    batas_Hit_Put = (int) Math.round((whiteSum + blackSum)/40.0);
    Light_kiri.setFloodlight(false);
    Light_kanan.setFloodlight(false);
}

```

Gambar 4.12Source Code-3 LineFollower

Penjelasan source code diatas adalah sebagai berikut :

Pada method calcThreshold (lihat gambar 4.12) terdapat variable whitesum dan blacksum.Kedua variable tersebut merupakan penjumlahan nilai value yang dibaca pada bidang putih dan pada bidang hitam.Kedua nilai tersebut berguna untuk menghitung batas nilai hitam dan putih.

4.1.8. BT

Kelas BT.java merupakan kelas yang berguna sebagai pusat penerima perintah kendali robot melalui Bluetooth. Bunyi beep menandakan bahwa terdapat koneksi yang masuk. Pada method action terdapat baris

perintah pergerakan robot seperti belok kiri, belok kanan, maju kedepan dan mundur kebelakang. Potongan source code program :

```
public void run(){
    while(true) {
        //LCD.drawString("Waiting..", 0, 6);
        btconn = Bluetooth.waitForConnection();
        if(btconn == null){
            continue;
        }
        //LCD.drawString("Connected ", 0, 6);
        btread = new BT_reader(btconn, this);
        btread.start();
        isConnected = true;
        Sound.beepSequenceUP();
        try{
            Thread.sleep(Integer.MAX_VALUE);
        }catch (InterruptedException e) {
            isConnected = false;
            Sound.beepSequence();
        }
    }
}
```

Gambar 4.13 Source Code-1 BT

```
public void action() {
    try{
        int command = btread.getCommand();
        if(command != prevCommand) {
            switch (command & DRIVECOMMAND) {
                case LEFT:
                    mt.kiri();
                    break;
                case RIGHT:
                    mt.kanan();
                    break;
```

Gambar 4.14 Source Code-2 BT

```
                case FORWARD:
                    mt.maju();
                    break;
                case BACKWARD:
                    mt.mundur();
                    break;
                case STOP:
                    mt.berhenti();
                    break;
            }
            prevCommand = command;
        }catch (Exception e) {}
    }
}
```

Gambar 4.15 Source Code-3 BT

4.1.9. BT_reader

Kelas BT_reader.java merupakan kelas yang mendeteksi ada tidaknya koneksi Bluetooth yang sedang belangsung. Potongan source code program :

```
public void run(){  
  
    command = 0;  
  
    try{  
        while(true) {  
            command = istream.readInt();  
            LCD.drawInt(command, 0, 0);  
            try{  
                Thread.sleep(100);  
            }catch (InterruptedException e) {}  
        }  
  
    }catch (Exception e) { //jika koneksi lost  
        driveThread.interrupt();  
    }  
    try{  
        if(istream != null){  
            istream.close();  
        }else if(con != null){  
            con.close();  
        }  
    }catch (IOException e1) {}  
    }  
    }  
    public int getCommand(){  
        return command;  
    }  
}
```

Gambar 4.16 Source Code BT_reader

4.2 Hasil Pengujian Robot

Hasil pengujian robot pada lintasan dijelaskan dalam tabel 4.2, tabel 4.3 dan tabel 4.4.

Tabel 4.2 Hasil Uji Robot Pada Arena I

Arena I			
Panjang lintasan = 101.68 cm			
Pengujian ke -	Waktu Penyelesaian (detik)	Freq Jatuh	Kecepatan (cm/detik)
1	28.38	0	3.58
2	31.11	1	3.26
3	26.41	1	3.58
4	30.60	2	3.32
5	30.81	3	3.30
6	29.55	1	3.44
7	29.30	0	3.47
8	24.95	2	4.07
9	26.64	1	3.81
10	21.30	0	4.77
Rata-rata :	27.91 detik	1 kali	3.32 cm/detik
Standar Deviasi	3.10		0.46

Tabel 4.3 Hasil Uji Robot Pada Arena II

Arena II			
Panjang lintasan = 152.92 cm			
Pengujian ke -	Waktu Penyelesaian (detik)	Freq Jatuh	Kecepatan (cm/detik)
1	52.73	2	2.90
2	45.18	2	3.38
3	44.35	3	3.44
4	48.47	3	3.15
5	42.48	2	3.59
6	43.57	3	3.51
7	45.66	3	3.35
8	40.82	2	3.75
9	40.66	0	3.76
10	43.60	3	3.51
Rata-rata :	44.75 detik	2 kali	3.43 cm/detik
Standar Deviasi	3.63		0.26

Tabel 4.4 Hasil Uji Robot Pada Arena III

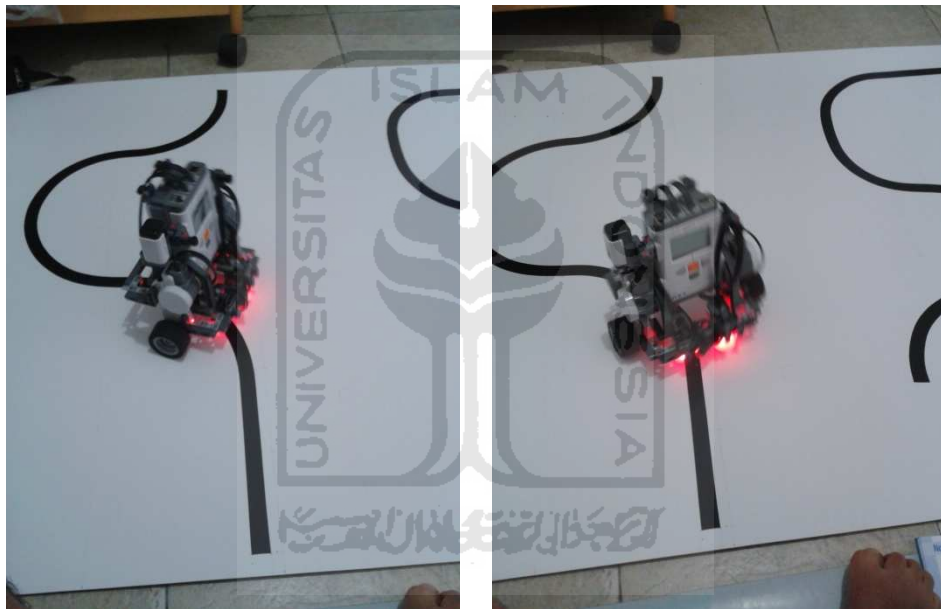
Arena III			
Panjang lintasan = 227.05 cm			
Pengujian ke -	Waktu Penyelesaian (detik)	Freq Jatuh	Kecepatan (cm/detik)
1	45.21	2	5.02
2	48.25	3	4.70
3	63.06	3	3.60
4	61.76	3	3.67
5	60.22	3	3.77
6	60.64	3	3.74
7	60.46	3	3.75
8	53.71	3	4.22
9	50.80	3	4.46
10	57.69	3	3.93
Rata-rata :	56.18 detik	3 kali	4.08 cm/detik
Standar Deviasi	6.26		0.49

Berdasarkan hasil uji diatas maka dapat disimpulkan bahwa :

- a. Pada arena 1 (mudah) robot dapat menyelesaikan tantangan dengan rata-rata waktu 27.91 detik, dengan frequensi terjatuh sebanyak satu kali dalam setiap kali pengujian.

- b. Pada arena 2 (sedang) robot dapat menyelesaikan tantangan dengan rata-rata waktu 44.75 detik, dengan frekuensi terjatuh sebanyak dua kali dalam setiap kali pengujian.
- c. Pada arena 3 (sulit) robot dapat menyelesaikan tantangan dengan rata-rata waktu 56.18 detik, dengan frekuensi terjatuh sebanyak 3 kali dalam setiap kali pengujian.

Gambar 4.17 dan 4.18 menggambarkan proses pengujian robot pada lintasan line tracer.



Gambar 4.17 Uji-1

Gambar 4.18 Uji-2

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Robot Segway Line Tracer yang telah dibuat ini memiliki kemampuan untuk menyeimbangkan diri walau hanya menggunakan dua roda yang bersisian. Metode pengaturan keseimbangan pada robot segway ini adalah PID Controller. Robot juga diberi tambahan kemampuan berupa line follower skill yang membuat robot dapat berjalan mengikuti garis hitam pada permukaan berwarna putih.

Kekurangan yang didapat dari hasil uji coba adalah robot masih belum bisa berdiri seimbang untuk waktu yang lama, robot masih sering terjatuh jika mendapat gangguan berupa kondisi arena yang tidak rata atau tidak datar sempurna, serta jika ada sentuhan dorongan terhadap robot.

5.2 Saran

Berdasarkan kekurangan-kekurangan yang ditemui pada robot maka saran yang diberikan yaitu :

1. Perbaikan desain robot masih perlu dilakukan demi mendapat hasil yang lebih baik lagi.
2. Tidak menutup kemungkinan algoritma PID Controller yang digunakan saat ini bisa diganti dengan algoritma lain yang lebih baik. Hal tersebut bisa dilakukan dengan penelitian lebih lanjut.

DAFTAR PUSTAKA

- Hariyanto, Bambang. 2010. *Esensi-esensi Bahasa Pemrograman Java*. Bandung: Informatika Bandung.
- Antonio, Juan. 2008. *Develop leJOS Programs Step by Step*. <http://www.juanantonio.info/lejos-ebook/>. Diakses pada tanggal 04 January 2012.
- Butka, Jason. 2008. *Installing Lejos on the NXT Mindstorm Using the Eclipse IDE and Introduction to Lejos / Java Programming*. <http://ebookbrowse.com/ltu-lejos-butka-pdf-d88150121>. Diakses pada tanggal 06 January 2012.
- Van Putten, B.J.S. 2006. *Design of a maze solving robot using Lego MINDSTORMS*. Skripsi. Eindhoven: Department of mechanical engineering Dynamics and Control group.
- Dung, Vu Ba Tien. 2010. *DEVELOPING A SELF-BALANCING ROBOT WITH LEGO NXT MINDSTORMS*. Skripsi. Turku: Information Technology.
- Kuningsih, T.W. 2011. *Derajat Kebebasan Degree Of Freedom*. (Online), <http://triwahyukuningsih.wordpress.com/2011/05/16/derajat-kebebasan-degree-of-freedom-dof/>, diakses 6 Maret 2012.
- Pitowarno, E. 2007. *Introduction to Robotics*. Makalah disajikan dalam Seminar New Concept Robotics : Robot Vision, Universitas Gunadarma, Jakarta, 22 Februari.
- Very, B. 2010. *Pemrograman Robot LEGO (RCX & NXT)*. Semarang : Robokidz.

Fachri, Z., Akhmad, H, & Ardik, W. 2010. *Perencanaan Balancing Robot Dengan Dua Roda*. Skripsi. Surabaya : Politeknik Elektronika Negeri.

Geoffrey D. Bennett. 2007. *Building an Open-Source Segway*.



Lampiran 1

LANGKAH-LANGKAH INSTALASI LEJOS NXJ DAN ECLIPSE HELIOS

Langkah 1

Persiapkan software pendukung yang terdiri dari (software sudah disertakan dalam CD laporan) :

1. NXT MINDSTORMS EDU.
2. Java JDK (jdk-6u27-windows-i586.exe).
3. leJOS_NXJ_0.9.0-Setup.exe
4. eclipse-java-helios-SR2-win32.zip

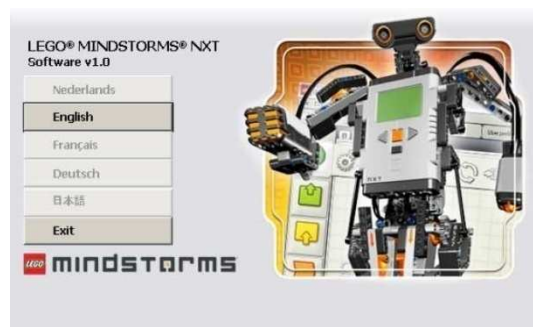
Langkah 2

Lakukan proses instalasi NXT MINDSTORMS EDU dengan cara :

1. Klik Icon **Autorun.exe**

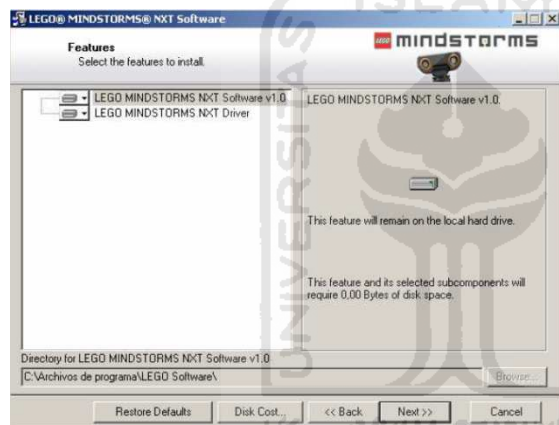


2. Selanjutnya akan muncul tampilan sebagai berikut



Klik menu “English”

- Selanjutnya pilih komponen yang mau diinstall lalu Next dan ikuti langkah instalasi hingga selesai



Langkah 3

Install Java JDK (jdk-6u27-windows-i586.exe)



jdk-6u27-windows-i586

Langkah 4

Install lejos nxj dengan cara double klik icon setup

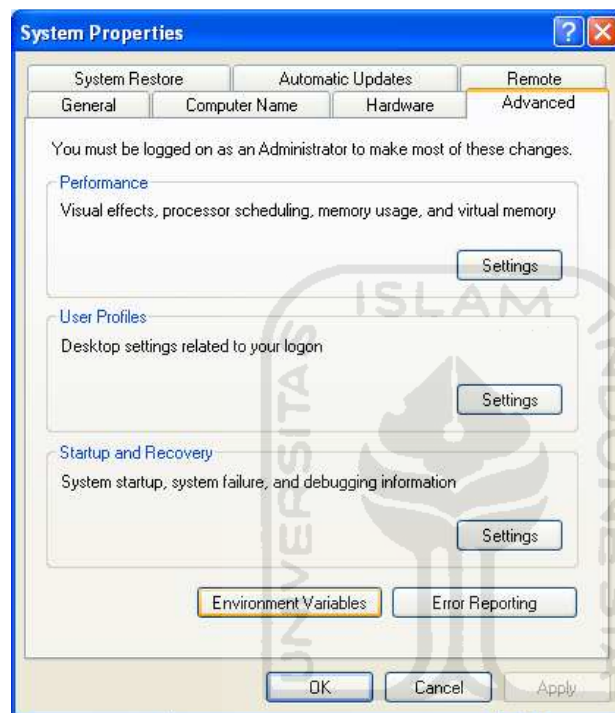


leJOS_NXJ_0.9.0-Setup
leJOS NXJ 0.9.0 Setup
lejos.org

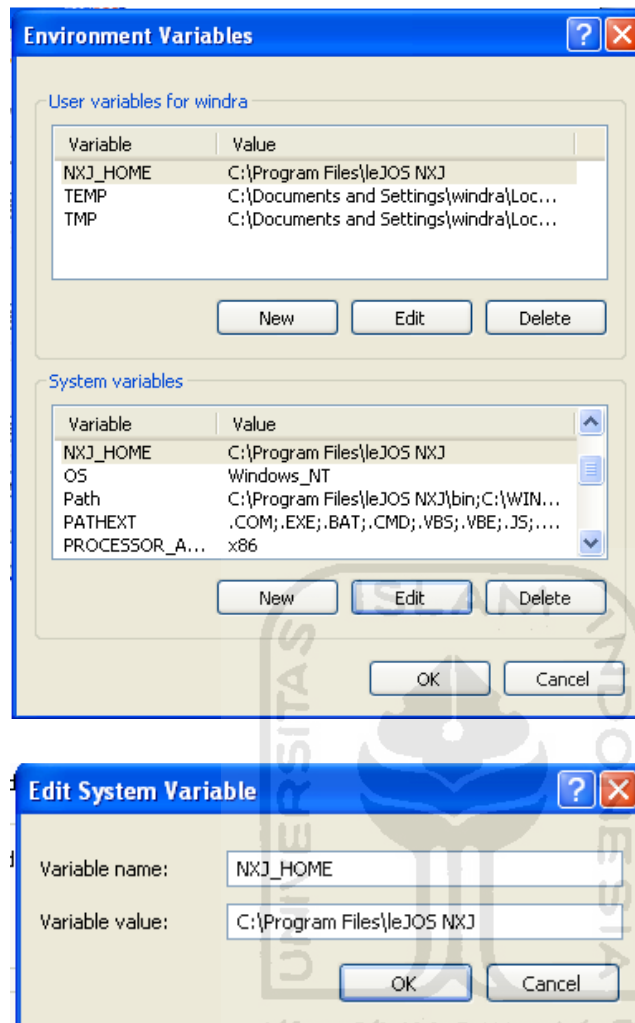
Langkah 5

Setting environment variable dengan cara :

1. Klik **Start>Control Panel>System** atau Klik kanan pada icon **My Computer>Properties>Advanced>Environment Variables**



2. Set Up System Variable dengan cara :
 - a. Klik **New**
 - b. Isi variable name : **NXJ_HOME**
 - c. Isi variable value : **C:\Program Files\leJOS NXJ**
 - d. Klik **Ok**



3. Update the environment variable PATH. Berisi :

- a. lokasi bin folder dari JDK installation
- b. lokasi bin folder dari NXJ installation

contoh : tambahkan script seperti di bawah ini pada value dari variable Path

```
%Program% Files\Java\jdk1.6.0_27\bin\;C:\ProgramFiles\leJOS NXJ\bin;
```

Langkah 6

Reboot/Restart Komputer Anda

Langkah 7

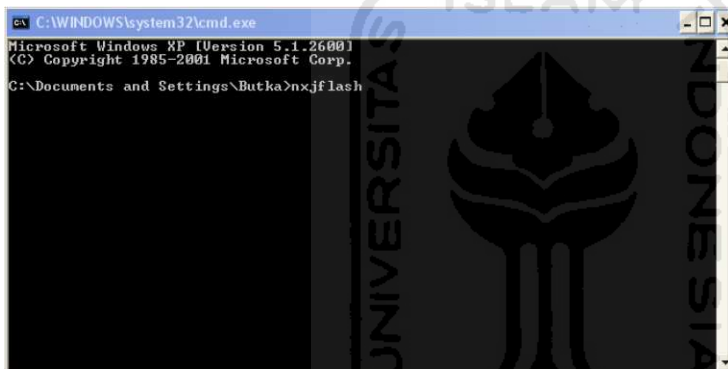
Install NXJ into your NXT Brick dengan cara :

Buka **commandprompt**> ketikkan perintah "**nxjflash**"

Jika belum me-reset NXT Brick dan NXT tidak terkoneksi pada usb maka akan muncul pesan :

"No devices in firmware update mode were found. Searching for other NXT devices.

No NXT found. Please check that the device is turned on and connected."



Langkah 8

Tekan **Reset Button** pada NXT Brick selama 5 detik sampai terdengar bunyi. Ulangi langkah 7.



These images show where the reset button is on the NXT brick. If you look carefully you will see a small silver button which is fairly simple to depress using a paperclip or toothpick. This button allows you to place the NXT brick in reset mode

Langkah 9

Selanjutnya melakukan setting Eclipse plugin dengan cara :

1. Pastikan komputer Anda terkoneksi dengan internet
2. Ekstrack file **eclipse-java-helios-SR2-win32.zip**
3. Adding the NXJ Plug in :
 - a. Jalankan **Eclipse.exe**
 - b. Go to **Help>Install new software>** Klik **Add**
 - c. Pada Kotak dialog **Add Repository>** isi location :
<http://lejos.sourceforge.net/tools/eclipse/plugin/nxj/> >**Ok**
 - d. Klik **LeJOS NXJ**> ikuti langkah instalasi >Selesai

