

**DESAIN DAN IMPLEMENTASI SISTEM PERGERAKAN
ROBOT BERKAKI EMPAT PADA KRCI DIVISI BERKAKI**

**SKRIPSI
JURUSAN TEKNIK ELEKTRO**

Diajukan untuk memenuhi persyaratan
memperoleh gelar Sarjana Teknik



Disusun oleh:

INDRAWAN RIZA FIRMANSYAH

NIM. 0610633034-63

KEMENTERIAN PENDIDIKAN NASIONAL

UNIVERSITAS BRAWIJAYA

FAKULTAS TEKNIK

MALANG

2011

LEMBAR PERSETUJUAN

**DESAIN DAN IMPLEMENTASI SISTEM PERGERAKAN
ROBOT BERKAKI EMPAT PADA KRCI DIVISI BERKAKI**

**SKRIPSI
JURUSAN TEKNIK ELEKTRO**

Diajukan untuk memenuhi persyaratan
memperoleh gelar Sarjana Teknik



Disusun oleh:

INDRAWAN RIZA FIRMANSYAH

NIM. 0610633034-63

Telah diperiksa dan disetujui oleh :

Dosen Pembimbing I

Dosen Pembimbing II

Panca Mudjirahardjo, ST., MT.

NIP. 19700329 200012 1 001

Ir. Nurussa'adah, MT.

NIP. 19680706 199203 2 001

LEMBAR PENGESAHAN

**DESAIN DAN IMPLEMENTASI SISTEM PERGERAKAN
ROBOT BERKAKI EMPAT PADA KRCI DIVISI
BERKAKI**

**SKRIPSI
JURUSAN TEKNIK ELEKTRO**

Diajukan untuk memenuhi persyaratan
memperoleh gelar Sarjana Teknik

Disusun oleh:

**INDRAWAN RIZA FIRMANSYAH
NIM. 0610633034-63**

Skripsi ini telah diuji dan dinyatakan lulus pada
tanggal 26 Januari 2011

DOSEN PENGUJI

Waru Djuriatno, ST., MT.
NIP. 19690725 199702 1 001

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

Ponco Siwindarto, Ir., MS.
NIP. 19590304 198903 1 001

Mengetahui
Ketua Jurusan Teknik Elektro

Rudy Yuwono ST., M.Sc.
NIP. 19710615 199802 1 003

PENGANTAR

Alhamdulillah, puji syukur penulis panjatkan ke hadirat Allah SWT, karena atas limpahan nikmat dan karunia-Nya skripsi dengan judul “Desain dan Implementasi Sistem Pergerakan Robot Berkaki Empat pada KRCI Divisi Berkaki” ini dapat diselesaikan. Skripsi ini disusun sebagai persyaratan untuk memperoleh gelar Sarjana Teknik di Jurusan Teknik Elektro Universitas Brawijaya.

Penulis menyadari bahwa penyusunan skripsi ini tidak terlepas dari bantuan berbagai pihak. Oleh karena itu, dengan ketulusan dan kerendahan hati penulis menyampaikan terima kasih dan penghargaan yang sebesar-besarnya kepada :

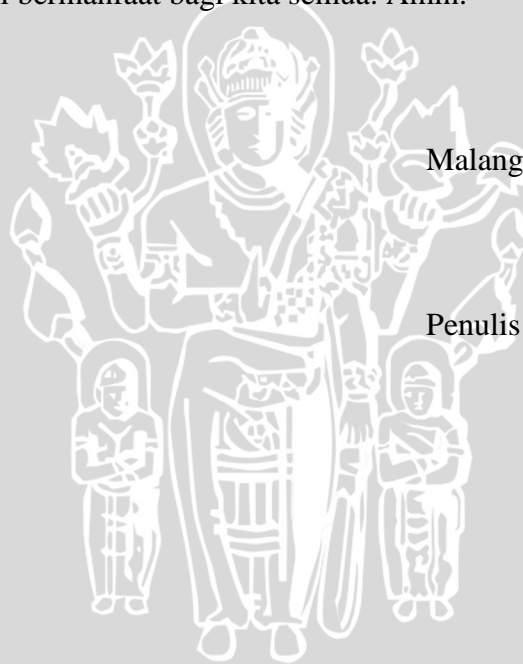
- Bapak Mashudi dan Ibu Arzakiatin, selaku orang tua penulis yang senantiasa memberikan kasih sayang, perhatian, nasehat, dukungan, dan doanya selama ini,
- Bapak Rudy Yuwono, ST., MSc. selaku Ketua Jurusan Teknik Elektro Universitas Brawijaya,
- Bapak M. Aziz Muslim, ST., MT, Ph.D. selaku Sekretaris Jurusan Teknik Elektro Universitas Brawijaya,
- Bapak Ir. M. Julius St, MS. sebagai Ketua Kelompok Dosen Keahlian Elektronika Jurusan Teknik Elektro Universitas Brawijaya,
- Bapak Panca Mudjirahardjo, ST, MT. sebagai Dosen Pembimbing I atas segala bimbingan, pengarahan, gagasan, ide, saran serta motivasi yang telah diberikan,
- Ibu Ir. Nurussa’adah, MT. sebagai Dosen Pembimbing II atas segala bimbingan, pengarahan, saran, kritik, dan masukan yang telah diberikan,
- Bapak Tibyani, ST., MT. Dan Bapak Ir. Chairuzzaini selaku dosen yang pernah menjadi pembimbing akademik.
- Staf Rekording Jurusan Teknik Elektro,
- Refita Edna Kautsar atas perhatian, motivasi, dan doanya yang selama ini diberikan kepada penulis,
- Tim Robot KRCI 2008, 2009, dan 2010 (KIMIKO, EXCALIBUR, TEUBots_Tankbercapits, N 4 HL, N 4 ML, AL-FIIL, Mas Eko, Mas Nino, Mas Rayi, Mas Riezqy, Mas Fido, Mas Zainul, Mbak Ana, Kanzi, Yudis, Aldo, Ucup, Ridho, Angga, Anas, dan Wahyu) dan seluruh Tim Robot Elektro UB 2010 atas semua perjuangan, pelajaran, pengalaman dan kenangan yang telah diberikan,

- Teman-teman seperjuangan (BESSET, Duta, Ilham, Andri, Yusniardi, Yusron, Anang, dll) yang telah memberi dan membagi berbagai hal bersama-sama,
- Keluarga besar GE-FORCE dan seluruh pengurus Himpunan Mahasiswa Elektro 2009-2010 yang telah mengajarkan banyak hal tentang organisasi, ikatan, tanggung jawab, dan kebersamaan,
- Laboratorium Elektronika dan segenap asisten yang telah membagikan berbagai ilmu dan juga telah menjadi tempat bernaung bagi penulis selama pengerjaan skripsi,
- Semua pihak yang telah memberikan bantuan serta dukungan baik secara langsung maupun tidak langsung atas penyusunan skripsi ini.

Penulis menyadari bahwa skripsi ini masih belum sempurna. Oleh karena itu, penulis mengharapkan kritik dan saran untuk penyempurnaan di masa yang akan datang. Semoga skripsi ini bermanfaat bagi kita semua. Amin.

Malang, 24 Desember 2010

Penulis



DAFTAR ISI

PENGANTAR	i
DAFTAR ISI	iii
DAFTAR GAMBAR	vi
DAFTAR TABEL	x
ABSTRAK	xi
BAB I PENDAHULUAN	
1.1. Latar Belakang	1
1.2. Rumusan Masalah	2
1.3. Ruang Lingkup	2
1.4. Tujuan	2
1.5. Sistematika Pembahasan	3
BAB II TINJAUAN PUSTAKA	
2.1 Kontes Robot Cerdas Indonesia Divisi Berkaki	4
2.2 Robot Berkaki	6
2.3 Kaki Serangga	8
2.4 Keseimbangan Robot Berkaki	10
2.4.1 Keseimbangan Statis	11
2.4.2 Keseimbangan Dinamis	12
2.5 Motor DC Servo	13
2.6 Mikrokontroler ATmega8	16
2.6.1 Arsitektur AVR	17
2.6.2 Struktur dan Operasi Port	19
2.6.3 Timer/Counter	20
BAB III METODOLOGI PENELITIAN	
3.1 Penentuan Spesifikasi Alat	22
3.2 Perancangan dan Pembuatan Alat	22
3.2.1 Perancangan dan Pembuatan Perangkat Keras	22
3.2.2 Perancangan dan Penyusunan Perangkat Lunak	23

3.3	Pengujian Alat.....	23
3.3.1	Pengujian Tiap Blok	23
3.3.2	Pengujian Keseluruhan Sistem	23
3.4	Pengambilan Kesimpulan	23

BAB IV PERANCANGAN DAN PEMBUATAN ALAT

4.1	Perancangan Mekanik Robot.....	24
4.2	Perancangan Gerakan	27
4.2.1	Daerah Kerja Sendi Kaki Robot	27
4.2.2	Posisi Dasar Robot.....	28
4.2.3	Gerakan Dasar Robot Berkaki Empat.....	30
4.2.3.1	Gerak Maju	31
4.2.3.2	Gerak Mundur.....	34
4.2.3.3	Gerak Hadap Kanan.....	36
4.2.3.4	Gerak Hadap Kiri.....	38
4.2.3.5	Gerak Berputar 180°	40
4.3	Perancangan Sistem	40
4.4	Perancangan Perangkat Keras.....	42
4.4.1	Catu Daya Sistem.....	42
4.4.2	Rangkaian Mikrokontroler Kontrol Servo.....	45
4.4.3	Rangkaian <i>Rotary Switch</i>	46
4.4.4	Motor DC Servo.....	49
4.5	Perancangan Perangkat Lunak.....	53
4.5.1	Kontrol Multi-Servo	53
4.5.2	Algoritma Mikrokontroler Kontrol Servo.....	56

BAB V PENGUJIAN DAN ANALISIS

5.1	Pengujian Rangkaian Catu Daya	60
5.2	Pengujian Sinyal Kontrol Servo	62
5.2.1	Pengujian Sinyal <i>High Servo Tunggal</i>	62
5.2.2	Pengujian Sinyal Kontrol Multi-Servo	68
5.3	Pengujian Motor DC Servo.....	70
5.3.1	Pengujian Sudut Motor DC Servo	70
5.3.2	Pengujian Arus Motor DC Servo.....	72

5.4	Pengujian Rangkaian <i>Rotary Switch</i>	73
5.5	Pengujian Keseluruhan	78
5.5.1	Pengujian Posisi Dasar.....	78
5.5.2	Pengujian Gerak Maju	82
5.5.2.1	Pengujian Gerak Maju Pelan	82
5.5.2.2	Pengujian Gerak Maju Sedang	83
5.5.2.3	Pengujian Gerak Maju Cepat.....	85
5.5.3	Pengujian Gerak Mundur.....	86
5.5.4	Pengujian Gerak Hadap Kanan.....	86
5.5.5	Pengujian Gerak Hadap Kiri.....	87
5.5.6	Pengujian Gerak Putar 180°.....	88
5.5.7	Pengujian Gerak Gabungan	89

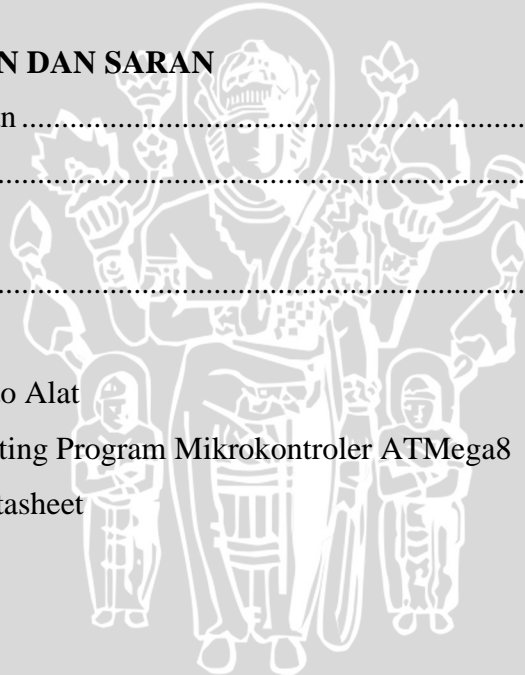
BAB VI KESIMPULAN DAN SARAN

6.1	Kesimpulan	91
6.2	Saran	91

DAFTAR PUSTAKA	92
-----------------------	-------	----

LAMPIRAN

LAMPIRAN 1	Foto Alat
LAMPIRAN 2	Listing Program Mikrokontroler ATmega8
LAMPIRAN 3	Datasheet



DAFTAR GAMBAR

Gambar 2.1.	Arena KRCI Divisi Berkaki.....	5
Gambar 2.2.	Robot Berkaki SILO4	7
Gambar 2.3.	Derajat Kebebasan pada Kaki Robot SILO4	8
Gambar 2.4.	Posisi Kaki Serangga pada Metode <i>Tripod Gait</i>	9
Gambar 2.5.	Struktur Kaki pada Kecoak	10
Gambar 2.6.	Keseimbangan pada Bidang Miring Robot TITAN VII	11
Gambar 2.7.	Keseimbangan Statis pada Robot <i>Quadrupedal</i> dan Robot <i>Hexapodal</i>	13
Gambar 2.8	Keseimbangan pada Kuda (a) Kuda saat berjalan (b) Kuda saat berlari.....	19
Gambar 2.9.	Blok Diagram Motor DC Servo	14
Gambar 2.10.	Motor Servo Parallax	14
Gambar 2.11.	Konfigurasi Pin Pengkabelan Motor Servo	15
Gambar 2.12.	Diagram waktu PWM motor servo	16
Gambar 2.13.	Arsitektur AVR	18
Gambar 2.14.	Konfigurasi pin ATmega8.....	18
Gambar 4.1.	Perspektif Desain Mekanik Robot Berkaki Empat	25
Gambar 4.2.	Robot Berkaki Empat tampak atas.....	25
Gambar 4.3.	Robot Berkaki Empat tampak depan	26
Gambar 4.4.	Bentuk kaki robot dan arah gerak tiap motor DC servo	26
Gambar 4.5.	Batas Simpangan Maksimum Servo Pinggang	28
Gambar 4.6	Batas Simpangan Maksimum Servo Lutut	28
Gambar 4.7.	Keseimbangan Robot dengan 4 Kaki.....	29
Gambar 4.8.	Keseimbangan Robot dengan 3 Kaki.....	29
Gambar 4.9.	Keseimbangan Robot dengan 2 Kaki.....	30
Gambar 4.10.	Posisi Motor DC Servo pada Robot Berkaki Empat.....	30
Gambar 4.11.	Daerah Kerja Motor DC Servo pada Robot Berkaki Empat.....	31
Gambar 4.12.	Diagram Blok Sistem Robot Berkaki Empat	41
Gambar 4.13.	Peletakkan Perangkat Keras pada Robot Berkaki Empat	42
Gambar 4.14.	Rangkaian Catu Daya Mikrokontroler ATmega8	43
Gambar 4.15.	Rangkaian Catu Daya Motor DC Servo.....	43

Gambar 4.16. Skema Rangkaian Mikrokontroler Kontrol Servo	46
Gambar 4.17. Skema Rangkaian <i>Rotary Switch</i>	47
Gambar 4.18. Hubungan Kecepatan Robot dengan Kecepatan Servo	50
Gambar 4.19. Motor DC Servo Tower Pro MG945R.....	52
Gambar 4.20. Sinyal Kontrol Motor DC Servo	53
Gambar 4.21. Pulsa Pengontrolan Multi Servo (OCRA).....	54
Gambar 4.22. Pulsa Pengontrolan Multi Servo (OCRB).....	54
Gambar 4.23. Diagram Alir Program Interupsi <i>Timer1</i>	56
Gambar 4.24. Diagram Alir Program Interupsi Timer OCRA	57
Gambar 4.25. Diagram Alir Program Interupsi Timer OCRB	58
Gambar 4.26. Diagram Alir Program Utama Mikrokontroler Kontrol Servo ...	59
Gambar 5.1. Diagram Blok Pengujian Rangkaian Catu Daya.....	60
Gambar 5.2. Tegangan Masukan dan Keluaran pada Rangkaian Catu Daya 5 V	61
Gambar 5.3. Tegangan Masukan dan Keluaran pada Rangkaian Catu Daya 6 V	62
Gambar 5.4. Diagram Blok Pengujian Sinyal <i>High</i> Servo Tunggal.....	62
Gambar 5.5. Sinyal Kontrol Servo dengan Lebar Sinyal <i>High</i> 600 μ s.....	63
Gambar 5.6. Sinyal Kontrol Servo dengan Lebar Sinyal <i>High</i> 1200 μ s.....	64
Gambar 5.7. Sinyal Kontrol Servo dengan Lebar Sinyal <i>High</i> 1300 μ s.....	64
Gambar 5.8. Sinyal Kontrol Servo dengan Lebar Sinyal <i>High</i> 1350 μ s.....	64
Gambar 5.9. Sinyal Kontrol Servo dengan Lebar Sinyal <i>High</i> 1400 μ s.....	65
Gambar 5.10. Sinyal Kontrol Servo dengan Lebar Sinyal <i>High</i> 1500 μ s.....	65
Gambar 5.11. Sinyal Kontrol Servo dengan Lebar Sinyal <i>High</i> 1600 μ s.....	65
Gambar 5.12. Sinyal Kontrol Servo dengan Lebar Sinyal <i>High</i> 1650 μ s.....	66
Gambar 5.13. Sinyal Kontrol Servo dengan Lebar Sinyal <i>High</i> 1700 μ s.....	66
Gambar 5.14. Sinyal Kontrol Servo dengan Lebar Sinyal <i>High</i> 1800 μ s.....	66
Gambar 5.15. Sinyal Kontrol Servo dengan Lebar Sinyal <i>High</i> 2400 μ s.....	67
Gambar 5.16. Diagram Blok Pengujian Sinyal Kontrol Multi-Servo.....	68
Gambar 5.17. Sinyal Kontrol Multi Servo dengan lebar pulsa <i>High</i> 1500 μ s ...	68
Gambar 5.18. Sinyal Kontrol Multi-Servo dengan lebar pulsa <i>High</i> yang berbeda-beda	69
Gambar 5.19. Diagram Blok Pengujian Motor DC Servo	70
Gambar 5.20. Diagram blok pengujian arus motor DC servo	72

Gambar 5.21. Besar Arus untuk Motor DC Servo Tunggal.....	72
Gambar 5.22. Besar Arus yang Dibutuhkan 8 Motor DC Servo	73
Gambar 5.23. Diagram Blok Pengujian Rangkaian <i>Rotary Switch</i>	73
Gambar 5.24. Nilai Tegangan Output Rangkaian <i>Rotary Switch</i> Mode A (0,45 V)	74
Gambar 5.25. Nilai Tegangan Output Rangkaian <i>Rotary Switch</i> Mode B (0,90 V)	74
Gambar 5.26. Nilai Tegangan Output Rangkaian <i>Rotary Switch</i> Mode C (1,36 V)	74
Gambar 5.27. Nilai Tegangan Output Rangkaian <i>Rotary Switch</i> Mode D (1,81 V)	75
Gambar 5.28. Nilai Tegangan Output Rangkaian <i>Rotary Switch</i> Mode E (2,27 V)	75
Gambar 5.29. Nilai Tegangan Output Rangkaian <i>Rotary Switch</i> Mode F (2,72 V)	75
Gambar 5.30. Nilai Tegangan Output Rangkaian <i>Rotary Switch</i> Mode G (3,18 V)	76
Gambar 5.31. Nilai Tegangan Output Rangkaian <i>Rotary Switch</i> Mode H (3,63 V)	76
Gambar 5.32. Nilai Tegangan Output Rangkaian <i>Rotary Switch</i> Mode I (4,09 V)	76
Gambar 5.33. Nilai Tegangan Output Rangkaian <i>Rotary Switch</i> Mode J (4,54 V)	77
Gambar 5.34. Nilai Tegangan Output Rangkaian <i>Rotary Switch</i> Mode K (5 V)	77
Gambar 5.35. Robot Berkaki Empat saat Posisi Dasar dengan 4 Kaki (a) Robot tampak Perspektif (b) Robot tampak Atas	79
Gambar 5.36. Robot Berkaki Empat saat Posisi Dasar dengan 3 Kaki.....	80
Gambar 5.37. Robot Berkaki Empat saat Posisi Dasar dengan 2 Kaki.....	81
Gambar 5.38. Gerak Maju Pelan Robot Berkaki Empat dengan Sudut Servo Pinggang 10°	82
Gambar 5.39. Gerak Maju Sedang Robot Berkaki Empat dengan Sudut Servo Pinggang 20°	84

Gambar 5.40. Gerak Maju Cepat Robot Berkaki Empat dengan Sudut Servo Pinggang 30°85

Gambar 5.41. Urutan Gerakan Gabungan Robot Berkaki Empat.....90



DAFTAR TABEL

Tabel 2.1.	Fungsi khusus Port B ATmega8	19
Tabel 2.2.	Fungsi khusus Port C ATmega8	20
Tabel 2.3	Fungsi khusus Port D ATmega8	20
Tabel 4.5.	Hasil Konversi Tegangan Analog ke Sinyal Digital	49
Tabel 4.6.	Mode Program pada <i>Rotary Switch</i>	49
Tabel 4.1.	Tahap-tahap Gerak Maju Robot Berkaki Empat	32
Tabel 4.2.	Tahap-tahap Gerak Mundur Robot Berkaki Empat.....	34
Tabel 4.3.	Tahap-tahap Gerak Hadap Kanan Robot Berkaki Empat.....	36
Tabel 4.4.	Tahap-tahap Gerak Hadap Kiri Robot Berkaki Empat.....	38
Tabel 4.7.	Lebar Sinyal <i>High</i> Sudut Servo dan Nilai TOP.....	55
Tabel 5.1.	Hasil Pengujian Sinyal <i>High</i> Servo	67
Tabel 5.2.	Hasil Pengujian Motor DC Servo.....	71
Tabel 5.3.	Hasil Pengujian Rangkaian <i>Rotary Switch</i>	78
Tabel 5.4.	Hasil Pengujian Sinyal <i>High</i> untuk Posisi Dasar dengan 4 Kaki ..	79
Tabel 5.5.	Hasil Pengujian Sinyal <i>High</i> untuk Posisi Dasar dengan 3 Kaki ..	80
Tabel 5.6.	Hasil Pengujian Sinyal <i>High</i> untuk Posisi Dasar dengan 2 Kaki ..	81
Tabel 5.7.	Hasil Pengujian Gerak Maju Pelan pada Robot Berkaki Empat ...	83
Tabel 5.8.	Hasil Pengujian Gerak Maju Sedang pada Robot Berkaki Empat	84
Tabel 5.9.	Hasil Pengujian Gerak Maju Cepat pada Robot Berkaki Empat...	85
Tabel 5.10.	Hasil Pengujian Gerak Mundur pada Robot Berkaki Empat.....	86
Tabel 5.11	Hasil Pengujian Gerak Hadap Kanan pada Robot Berkaki Empat	87
Tabel 5.12.	Hasil Pengujian Gerak Hadap Kiri pada Robot Berkaki Empat....	88
Tabel 5.13.	Hasil Pengujian Gerak Putar 180° pada Robot Berkaki Empat	89
Tabel 5.14.	Hasil Pengujian Gerak Gabungan pada Robot Berkaki Empat	90

ABSTRAK

Indrawan Riza Firmansyah, Jurusan Teknik Elektro, Fakultas Teknik Universitas Brawijaya, 2011, *Desain dan Implementasi Sistem Pergerakan Robot Berkaki Empat pada KRCI Divisi Berkaki*, Dosen Pembimbing : Panca Mudjirahardjo, ST, MT., dan Ir. Nurussa'adah, MT.

Kontes Robot Cerdas Indonesia (KRCI) merupakan ajang kompetisi robotika nasional terdiri dari divisi Beroda, divisi Berkaki, dan divisi Battle. KRCI divisi Berkaki adalah kontes dimana robot harus menggunakan kaki untuk berjalan dan mencari sumber api dalam suatu model rumah. Permasalahan utama dari divisi Berkaki adalah bagaimana menentukan metode yang baik agar robot dapat berjalan dan bernavigasi menggunakan kaki. Robot Berkaki Empat adalah salah satu alternatif yang bisa digunakan untuk mengatasi masalah tersebut. Metode gerak Robot Berkaki Empat menirukan metode gerak hewan seperti kadal atau serangga, dengan beberapa penyesuaian. Sebagai pengganti sendi pada hewan digunakan motor DC servo yang diletakkan sebagai pengganti sendi pinggang dan sendi lutut, sehingga pada satu kaki terdapat dua buah motor DC servo.

Motor DC servo digunakan sebagai penggerak pada Robot Berkaki Empat karena sudut pergerakan motor ini dapat dikendalikan dengan memberikan pengaturan *duty cycle* sinyal PWM pada bagian pin kontrolnya. Sinyal kontrol didapatkan dari mikrokontroler yang berfungsi untuk mengatur sudut putar motor DC servo. Dengan mengkombinasikan gerakan pada kedelapan motor DC servo, maka dihasilkan metode pergerakan pada Robot Berkaki Empat yang terdiri dari gerak maju, gerak mundur, gerak hadap kanan, gerak hadap kiri, dan gerak putar 180°.

Berdasarkan pengujian didapatkan beberapa hasil yaitu kecepatan rata-rata gerak maju pelan 6,98 cm/detik, kecepatan rata-rata gerak maju sedang 14,15 cm/detik, kecepatan rata-rata gerak maju cepat 17,30 cm/detik, kecepatan rata-rata gerak mundur cepat 17,24 cm/detik, waktu yang dibutuhkan untuk gerak hadap kanan 2,57 detik dengan penyimpangan sudut rata-rata sebesar +6,25°, waktu untuk hadap kiri 2,69 detik dengan penyimpangan sudut rata-rata -2,1°, dan waktu untuk berputar 180° 5,04 detik dengan penyimpangan rata-rata -2,5°.

Kata Kunci : KRCI, Robot, Kaki Empat, Motor DC Servo

BAB I

PENDAHULUAN

1.1 Latar Belakang

Dunia robotika Indonesia sudah berkembang sangat pesat. Di bidang industri dan pertanian, sudah banyak digunakan robot-robot pengganti tugas manusia yang digunakan untuk mempermudah pekerjaan. Di bidang pendidikan, teknologi robotika juga sudah mulai dikenalkan bahkan hingga tingkat Sekolah Dasar. Salah satu wadah pengembangan teknologi robotika di bidang pendidikan adalah Kontes Robot yang diadakan oleh Direktorat Jendral Pendidikan Tinggi (Dirjen DIKTI), yang terdiri dari tiga kategori, yaitu Kontes Robot Indonesia (KRI), Kontes Robot Cerdas Indonesia (KRCI), dan Kontes Robot Seni Indonesia (KRSI).

KRCI merupakan kategori kontes robot yang mempertandingkan robot untuk menyelesaikan suatu tugas tertentu. Pada kategori ini dibagi menjadi tiga divisi, yaitu divisi Beroda, divisi Berkaki, dan divisi Battle. KRCI divisi Berkaki merupakan satu-satunya divisi yang mewajibkan robot untuk menggunakan kaki dalam pergerakannya, dengan tugas robot adalah untuk memadamkan api pada salah satu ruang dalam sebuah model rumah.

Permasalahan utama KRCI divisi berkaki adalah bagaimana merancang dan menentukan metode pergerakan robot yang menggunakan kaki sebagai pergerakannya, agar robot dapat bergerak dan menyelesaikan tugasnya. Salah satu alternatif desain yang dapat digunakan adalah dengan meniru gerakan dari hewan, terutama serangga. Karena bentuk kaki serangga lebih mudah untuk diimplementasikan daripada bentuk kaki hewan lain seperti kuda atau anjing. Hal lain yang perlu diperhatikan adalah jumlah kaki yang akan digunakan, karena semakin banyak kaki yang digunakan, maka semakin banyak kaki yang harus dikendalikan, selain itu juga akan semakin menambah biaya. Robot berkaki empat merupakan salah satu alternatif terbaik, karena dari segi jumlah kaki yang digunakan akan lebih sedikit daripada robot berkaki enam (*hexapod*) atau robot berkaki delapan (*octopod*), dan dari segi kestabilan akan lebih baik daripada robot berkaki dua (*bipedal robot*).

Beberapa aspek yang perlu diperhatikan dalam merancang robot berkaki empat diantaranya adalah keseimbangan robot saat berjalan, dan juga jumlah sendi yang

digunakan pada setiap kaki robot, karena banyaknya jumlah sendi akan menentukan metode pergerakan yang akan digunakan.

1.2 Rumusan Masalah

Dari masalah yang telah disebutkan pada latar belakang maka dalam skripsi ini, rumusan masalah ditekankan pada :

- 1). Bagaimana merancang dan membuat sistem mekanik untuk Robot Berkaki Empat,
- 2). Bagaimana merancang metode pergerakan untuk Robot Berkaki Empat,
- 3). Bagaimana merancang dan membuat sistem elektronika yang menunjang penggunaan motor servo sebagai penggerak pada Robot Berkaki Empat,
- 4). Bagaimana menerapkan metode pergerakan yang telah dibuat pada Robot Berkaki Empat berbasis mikrokontroler AVR ATmega8.

1.3 Ruang Lingkup

Dalam perancangan skripsi ini, permasalahan dibatasi oleh hal-hal sebagai berikut :

- 1). Ukuran robot yang digunakan disesuaikan dengan peraturan KRCI Divisi Berkaki tahun 2010,
- 2). Arena yang digunakan memiliki permukaan datar (kemiringan 0°),
- 3). Pergerakan Robot Berkaki Empat adalah maju, mundur, hadap kanan (90° ke kanan), hadap kiri (90° ke kiri), dan putar 180° ,
- 4). Kecepatan maksimum gerak maju robot adalah 20 cm/s,
- 5). Motor servo yang digunakan adalah Tower Pro MG995R,
- 6). Mikrokontroler yang digunakan adalah ATmega8,
- 7). Pemrograman mikrokontroler ATmega8 dengan menggunakan software *CodeVisionAVR C Compiler*,
- 8). Sistem yang digunakan merupakan sistem dengan loop terbuka, dimana analisis dibatasi hanya pada metode pergerakan Robot Berkaki Empat saja.

1.4 Tujuan

Tujuan penelitian ini adalah perancangan dan pembuatan metode pergerakan Robot Berkaki Empat untuk Kontes Robot Cerdas Indonesia divisi Berkaki.

1.5 Sistematika Pembahasan

Skripsi ini terdiri dari enam bab dengan sistematika pembahasan sebagai berikut:

BAB I Pendahuluan

Membahas latar belakang, rumusan masalah, batasan masalah, tujuan dan sistematika pembahasan.

BAB II Tinjauan Pustaka

Membahas teori-teori yang mendukung dalam perencanaan dan pembuatan alat, yang meliputi: robot berkaki, kaki serangga, Kontes Robot Cerdas Indonesia Divisi Berkaki, motor servo, mikrokontroler ATmega8, keseimbangan robot berkaki, kinematika dengan persamaan trigonometri.

BAB III Metodologi Penulisan

Membahas metode penelitian dan perencanaan alat.

BAB IV Perancangan dan Pembuatan Alat

Membahas desain dan perancangan Robot Berkaki Empat, dan memodelkan sistem pergerakannya. Kemudian mengatur agar robot dapat dikendalikan melalui mikrokontroler.

BAB V Pengujian dan Analisis

Membahas hasil pengujian sistem Robot Berkaki Empat untuk tiap-tiap blok dan secara keseluruhan.

Bab VI Kesimpulan dan Saran

Membahas kesimpulan perancangan ini dan saran-saran yang diperlukan untuk melakukan pengembangan aplikasi selanjutnya.

BAB II

TINJAUAN PUSTAKA

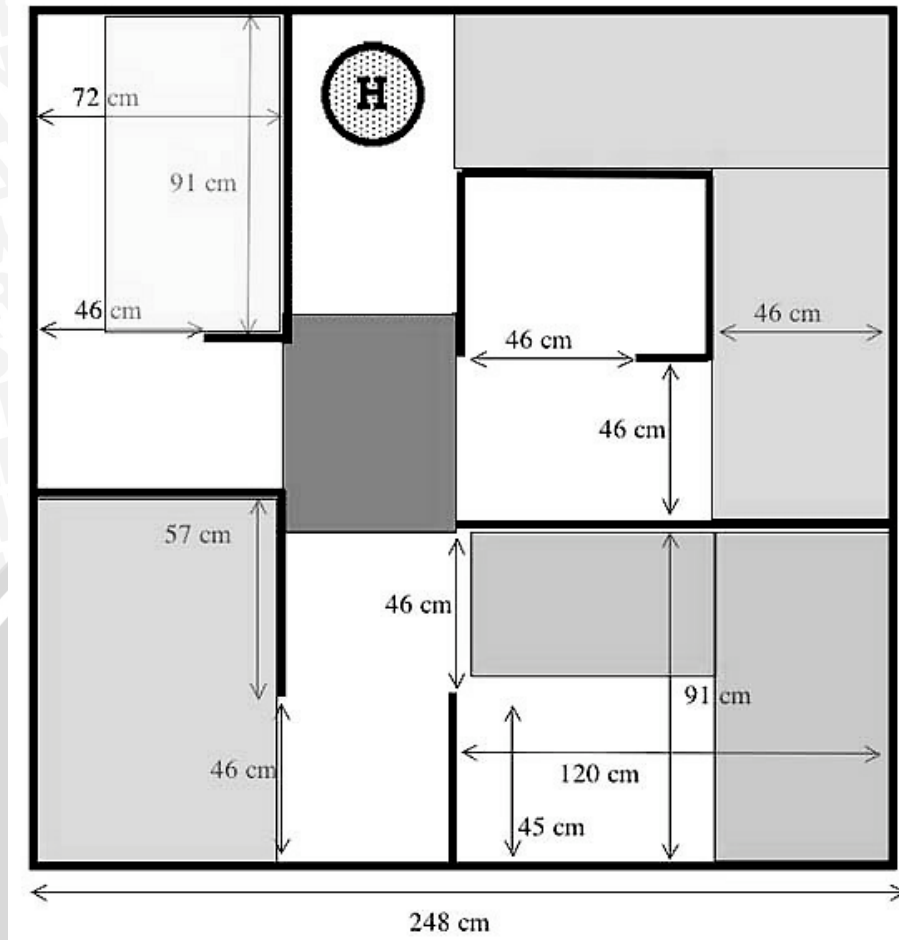
Beberapa teori pendukung yang perlu dibahas dalam pembuatan sistem *Bipedal Robot* ini meliputi :

- 1). Kontes Robot Cerdas Indonesia Divisi Berkaki
- 2). Robot Berkaki
- 3). Kaki Serangga
- 4). Keseimbangan Robot Berkaki
- 5). Motor DC Servo
- 6). Mikrokontroler ATmega8

2.1 Kontes Robot Cerdas Indonesia Divisi Berkaki

Kontes Robot Cerdas Indonesia merupakan pertandingan robot tingkat nasional yang diadakan oleh Direktorat Jendral Pendidikan Tinggi (Dirjen DIKTI). Pertandingan ini diklasifikasikan ke dalam beberapa divisi yakni divisi Beroda, divisi Berkaki, dan divisi Battle. Divisi Berkaki merupakan pertandingan robot dengan tugas mencari ruangan tempat titik api berada dan mematikan api tersebut.

Robot yang digunakan pada divisi ini harus memenuhi persyaratan dimensi yaitu memiliki panjang maksimum 46 cm, lebar maksimum 31 cm, dan tinggi maksimum 27 cm. Bagian tubuh robot juga tidak diperbolehkan melebihi ketinggian dinding. Lapangan yang digunakan berukuran 248 cm x 248 cm dan tinggi dinding 30 cm. Bentuk lapangan KRCI divisi Berkaki ditunjukkan dalam Gambar 2.1.



Gambar 2. 1. Arena KRCI Divisi Berkaki
Sumber : DIKTI, 2010

Tugas utama robot KRCI divisi Berkaki adalah mematikan lilin dan kembali ke posisi awal (HOME). Lilin diletakkan di ruangan tertentu sesuai dengan undian yang dilakukan sebelumnya. Lilin harus dimatikan dalam jarak tertentu dan hanya diperbolehkan menggunakan media angin (kipas) atau air.

Untuk tahun 2010, sistem gerak kaki robot diarahkan kemiripannya dengan gerak kaki makhluk hidup yang sudah biasa dikenal yaitu gerakan kaki manusia (*humanoid*) atau gerakan kaki binatang (*animaloid*). Bentuk robot itu sendiri secara keseluruhan tidak harus dituntut menyerupai persis manusia ataupun persis binatang yang dimaksud. Dengan demikian maka yang dimaksud kaki disini adalah alat gerak robot yang berbentuk memanjang yang menghubungkan robot dengan lantai menggunakan beberapa ruas yang saling terhubung satu dengan yang lain oleh suatu sendi mekanis. Secara jumlah, robot berkaki (*humanoid* atau *animaloid*) dapat dikelompokkan menjadi berkaki dua atau berkaki lebih dari dua. Robot berkaki satu tidak masuk dalam kriteria pembagian kelompok ini dikarenakan makhluk hidup berkaki satu

tidak lazim sebagai *humanoid* ataupun *animaloid*. Untuk dapat mempertahankan konsistensi terhadap istilah *humanoid* dan *animaloid*, maka baik robot berkaki dua atau lebih ketika melakukan gerakan maju, mundur dan berbelok, robot harus menggunakan kaki-kakinya dan tidak diperkenankan menggunakan penopang (bagian tubuh yang menempel di lantai yang tidak berfungsi sebagai kaki). Untuk dapat memenuhi kriteria kriteria di atas maka sebuah “kaki” harus memiliki minimal dua sendi untuk dapat membuat suatu gerakan memindahkan robot. Untuk itu maka robot berkaki harus memiliki setidaknya dua derajat kebebasan.

2.2 Robot Berkaki

Robot telah mengalami perkembangan yang sangat pesat. Ketertarikan dunia pada bidang robotika begitu besar, dengan ditandai banyaknya penggunaan robot di berbagai bidang, seperti bidang industri, pertanian, kedokteran, dan juga militer. Tujuan utama pemanfaatan teknologi robotika pada berbagai bidang tersebut tidak lain adalah untuk mempermudah kerja manusia. Seperti definisi dari kata robot itu sendiri yang berarti “pekerja”, keberadaan robot untuk membantu pekerjaan manusia memang tidak bisa dipungkiri. Salah satu alasan penggunaan teknologi robot adalah karena robot memiliki kemampuan khusus untuk menyelesaikan suatu pekerjaan tertentu, dengan tingkat akurasi yang tinggi dan mampu mengerjakan sesuatu dalam waktu yang relatif lebih cepat jika dibandingkan dengan manusia.

Seiring dengan perkembangan zaman, robot kini memiliki program dan fungsi yang lebih kompleks. Robot dapat dikelompokkan menjadi beberapa jenis sesuai dengan spesifikasinya, misalkan fungsi dan alat gerak. Sesuai fungsinya, ada beberapa jenis robot, diantaranya robot penjelajah, robot industri, robot pemadam api, robot medis, robot nano, robot perang, dan robot kompetisi. Jenis robot sesuai dengan alat geraknya adalah robot beroda dan robot berkaki.

Robot berkaki adalah robot yang menggunakan kaki untuk bergerak. Kelebihan utama dari robot berkaki jika dibandingkan dengan robot beroda adalah kemampuannya untuk bergerak di medan yang tidak rata. Robot berkaki dapat melalui rintangan yang tidak bisa dilewati oleh robot yang menggunakan roda. Kelemahan dari robot berkaki adalah kecepatannya yang tidak secepat robot beroda.

Robot berkaki empat atau lebih dikenal dengan istilah *quadruped robot* adalah robot yang menggunakan empat buah kaki untuk melakukan pergerakan. Robot jenis ini memiliki kestabilan yang lebih baik jika dibandingkan dengan robot berkaki

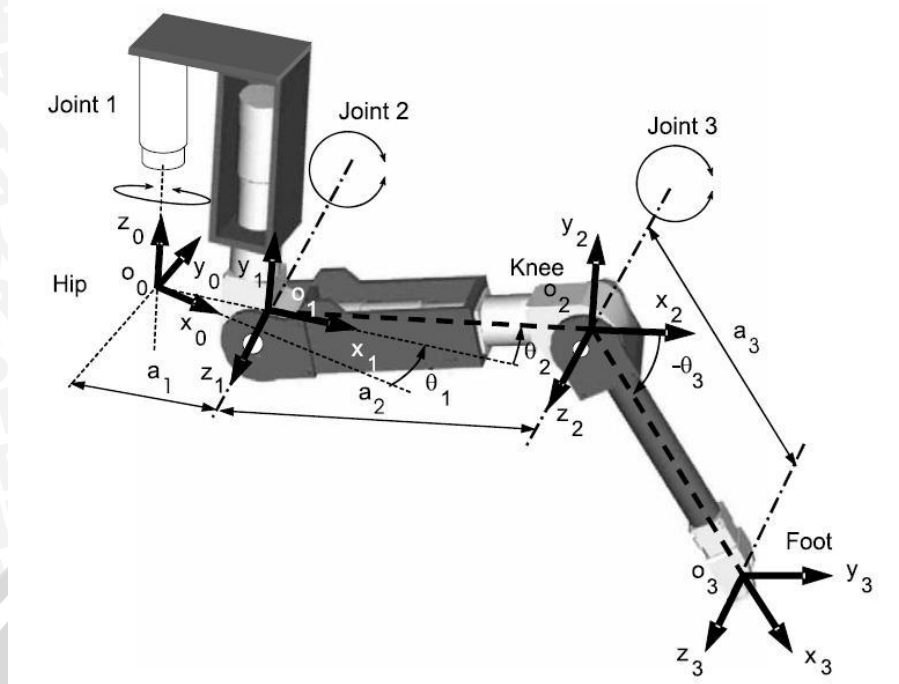
dua atau berkaki tiga. Pada saat bergerak, satu kaki mengayun ke atas sedangkan tiga kaki lain masih menapak ke tanah. Inilah yang disebut dengan kestabilan statis, dimana titik pusat beban robot ditopang oleh tiga kaki yang masih menapak ke tanah. Berbeda dengan robot berkaki dua, yang membutuhkan keseimbangan dinamis saat bergerak. Contoh dari robot berkaki empat ditunjukkan dalam Gambar 2.2.



Gambar 2. 2. Robot Berkaki SILO4

Sumber : Pablo Gonzalez de Santos, 2006 : 27

Robot berkaki empat memiliki jumlah derajat kebebasan (*degree of freedom*) sebanyak 2 sampai 3 buah pada setiap kakinya. Untuk dapat bergerak, maka perlu diatur kombinasi gerakan pada masing-masing sendi sehingga kaki robot dapat terangkat dan berpindah pada posisi yang diinginkan. Kombinasi antara keempat kaki inilah yang dapat menggerakkan robot ke depan, belakang, atau berputar. Contoh persendian dari salah satu kaki robot SILO4 ditunjukkan dalam Gambar 2.3.



Gambar 2. 3. Derajat Kebebasan pada Kaki Robot SILO4

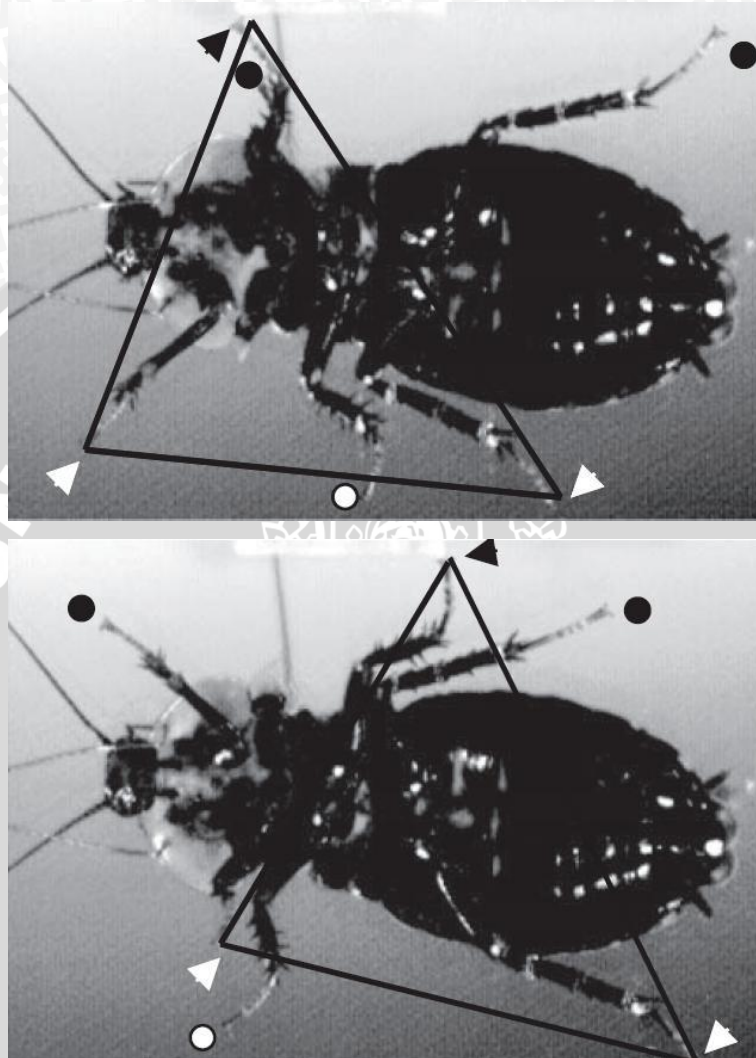
Sumber : Pablo Gonzalez de Santos, 2006 : 233

2.3 Kaki Serangga

Serangga merupakan makhluk hidup yang tergolong memiliki kemampuan bertahan pada berbagai lingkungan. Keberadaannya sejak jutaan tahun silam, dengan berbagai evolusi untuk menyesuaikan diri dengan lingkungannya, menjadikannya sebagai hewan yang paling mampu bertahan di bumi. Salah satu kelebihan serangga yang sangat disoroti oleh para pakar robotika adalah kemampuannya berjalan di berbagai medan. Hewan ini mampu berjalan melewati halangan-halangan yang bahkan melebihi ukuran tubuhnya. Salah satu contoh dari serangga yang sering kita jumpai di sekitar kita adalah kecoak. Kecoak tidak hanya mampu berjalan pada bidang horizontal saja, tetapi juga pada bidang vertikal. Kemampuan inilah yang sampai sekarang masih diteliti oleh para ahli.

Walaupun sistem pergerakan pada serangga tergolong sistem sederhana, tetapi dibalik itu sebenarnya terdapat sistem yang sangat rumit. Pergerakan kaki serangga dihasilkan oleh kombinasi dari sistem mekanik yang canggih, kontrol syaraf, dan juga sensor dari semua indera yang dimilikinya. Serangga berkaki enam melakukan pergerakan dengan metode *tripod gait*. Pada kecoak misalnya, keenam kakinya bergerak dengan kombinasi tiga buah kaki menapak ke tanah, dan tiga kaki lainnya bergerak di udara, kondisi ini terus dipertahankan, sehingga kecoak selalu dalam keadaan stabil

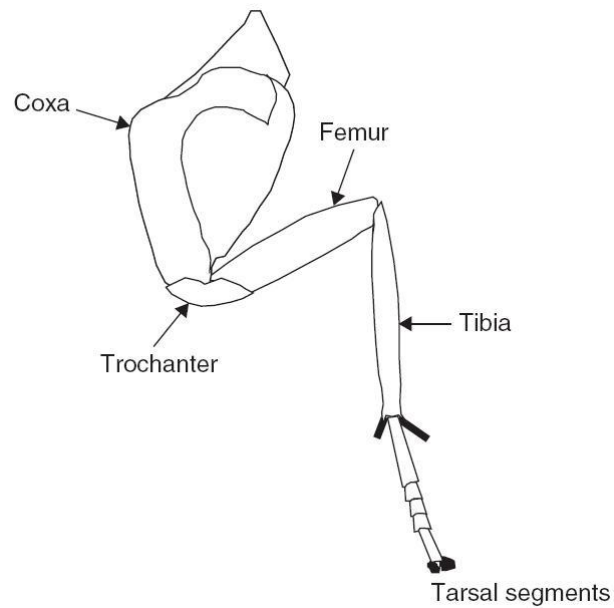
yang disebut dengan kestabilan statis. Metode *tripod gait* ini sangat stabil, karena selama bergerak, posisi pusat massa dari tubuh kecoak selalu berada di diantara tiga kaki yang menopangnya. Susunan kombinasi kaki yang menapak tanah dan yang mengayun di udara ditunjukkan dalam Gambar 2.4.



Gambar 2. 4. Posisi Kaki Serangga pada Metode *Tripod Gait*

Sumber : Vincent H. Resh, 2009 : 1045

Dari gambar di atas dapat dilihat bahwa ketiga kaki kecoak yang menapak ke tanah membentuk segitiga yang menopang tubuhnya. Kombinasi ini dijalankan bergantian dan terus dipertahankan, agar kecoak dapat berjalan. Metode inilah yang disebut *tripod gait*. Struktur kaki pada kecoak ditunjukkan dalam Gambar 2.5.



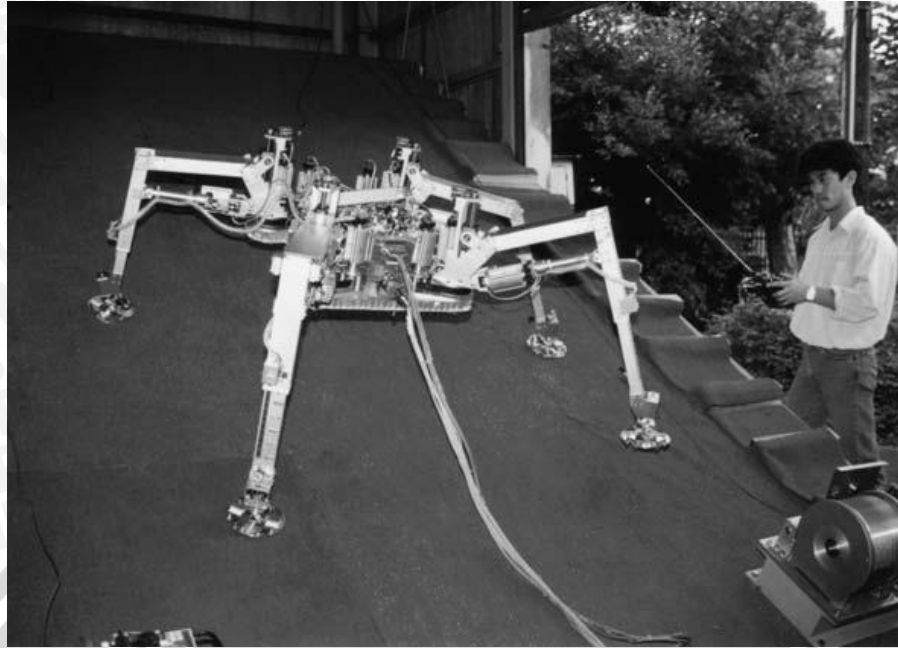
Gambar 2. 5. Struktur Kaki pada Kecoak

Sumber : Vincent H. Resh, 2009 : 1045

Kaki kecoak terbagi menjadi empat bagian utama, yaitu *coxa*, *trochanter*, *femur*, dan *tibia*. Diantara keempat bagian ini masing-masing terdapat satu sendi, jadi ada tiga sendi utama pada kecoak, yang berarti kaki kecoak mempunyai tiga derajat kebebasan (degree of freedom) yang digunakan untuk bergerak.

2.4 Keseimbangan Robot Berkaki

Salah satu faktor penting yang harus dimiliki oleh robot berkaki adalah keseimbangan. Robot dapat berjalan dengan baik jika mampu mempertahankan keseimbangan selama berjalan. Terdapat dua jenis keseimbangan pada robot, yaitu keseimbangan statis dan keseimbangan dinamis. Contoh robot dalam mempertahankan keseimbangannya ditunjukkan dalam Gambar 2.6.

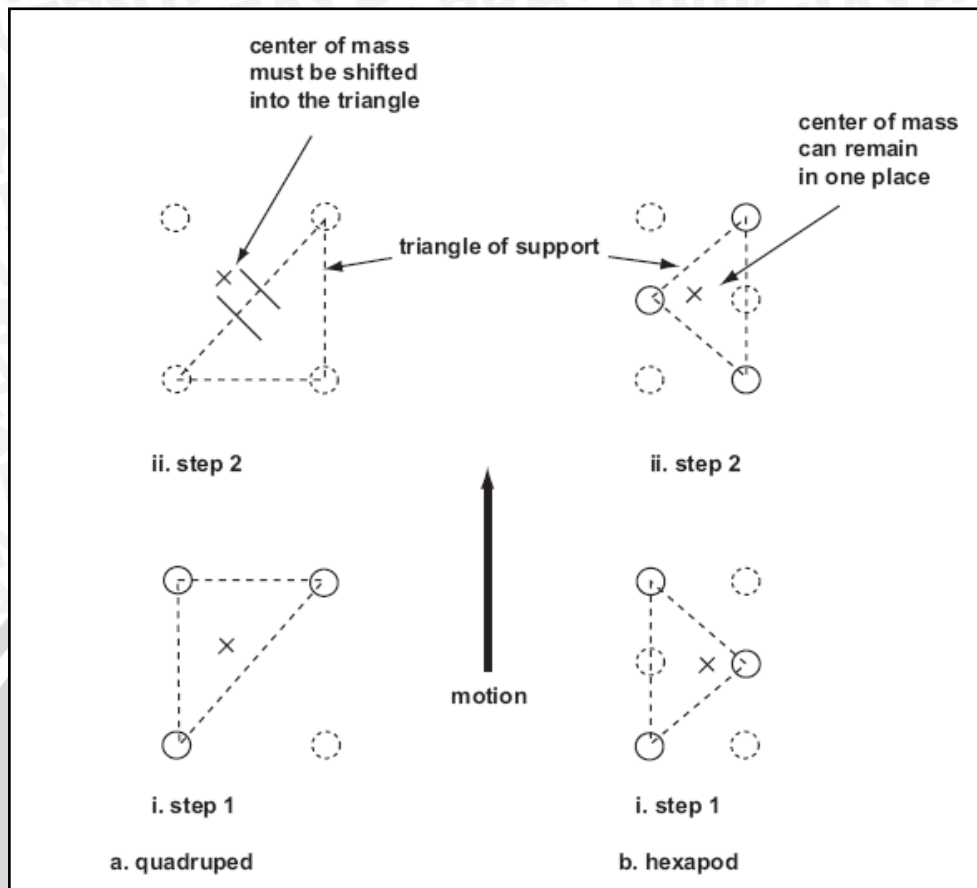


Gambar 2. 6. Keseimbangan pada Bidang Miring Robot TITAN VII

Sumber : Pablo Gonzalez de Santos, 2006 : 26

2.4.1 Keseimbangan Statis

Keseimbangan statis banyak digunakan dalam perancangan sebuah robot berkaki. Hal tersebut dikarenakan dalam mencapai keseimbangan statis tidak diperlukan adanya perpindahan titik berat robot, yang diperlukan adalah sebuah mekanisme (pada kaki) yang dapat selalu menempatkan titik berat robot pada satu titik tertentu dalam kondisi pergerakan apapun. Bisa dikatakan bahwa dalam keseimbangan statis, titik berat robot selalu terfokus dalam satu titik dalam area tertentu (sesuai struktur robot). Keseimbangan statis biasanya terdapat pada robot dengan kaki yang lebih dari empat (*quadrupedal*, *hexapodal*, dll). Pada robot jenis tersebut keseimbangan statis diciptakan dengan mempertahankan bentuk segitiga pada keenam kakinya. Titik berat robot *quadrupedal* (empat kaki) dan *hexapodal* (enam kaki) saat berjalan ditunjukkan dalam Gambar 2.7.



Gambar 2. 7. Keseimbangan Statis pada Robot *Quadrupedal* dan Robot *Hexapodal*

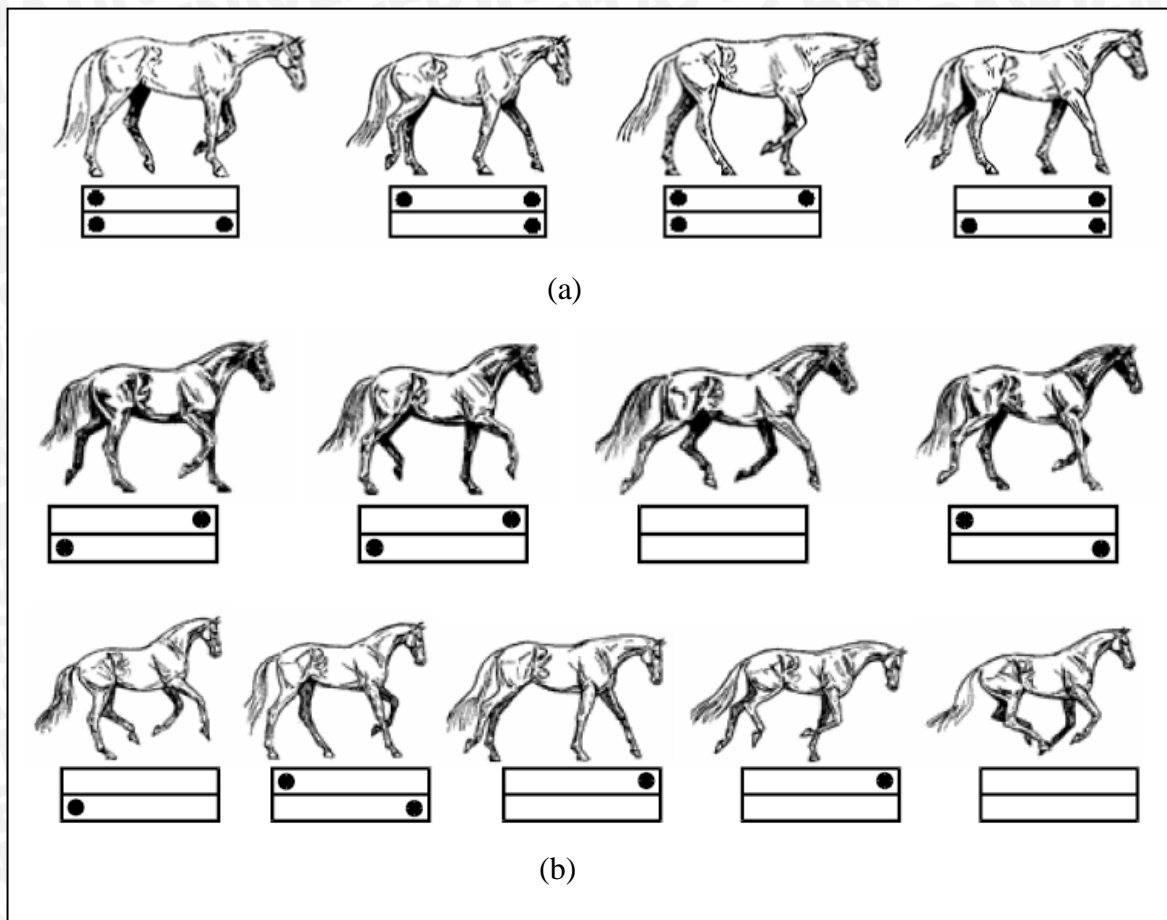
Sumber : Appin, 2007 : 294

Dalam Gambar 2.7 terlihat bahwa dalam kondisi berjalan, baik robot *quadrupedal* maupun *hexapodal* masih memiliki titik berat pada posisi di dalam segitiga yang dibentuk oleh kakinya. Syarat agar keseimbangan statis dapat terpenuhi adalah diperlukan minimal tiga buah kaki yang menapak ke tanah agar robot tetap memiliki keseimbangannya.

2.4.2 Keseimbangan Dinamis

Keseimbangan dinamis adalah sebuah kondisi keseimbangan yang terbentuk dalam kondisi (pergerakan) apapun. Diperlukan analisa kestabilan sistem pada sebuah robot sebelum hal tersebut dapat diterapkan. Keseimbangan dinamis tidak harus diciptakan dari kombinasi posisi kaki, tetapi bisa diciptakan dari pergerakan badan untuk merubah titik-beratnya. Sebagai contoh, seekor kuda saat berjalan selalu menempatkan ketiga kakinya untuk menapak ke tanah, akan tetapi saat berlari ia memiliki posisi dimana hanya dua kaki yang menapak ke tanah, bahkan ada juga posisi melayang di udara. Keseimbangan yang terbentuk saat berjalan adalah keseimbangan

statis, sedangkan saat berlari adalah keseimbangan dinamis. Posisi kaki kuda saat berjalan dan berlari ditunjukkan dalam Gambar 2.8.



Gambar 2. 8. Keseimbangan pada Kuda (a) Kuda saat berjalan (b) Kuda saat berlari

Sumber : Vedran Kordic, 2005 : 566

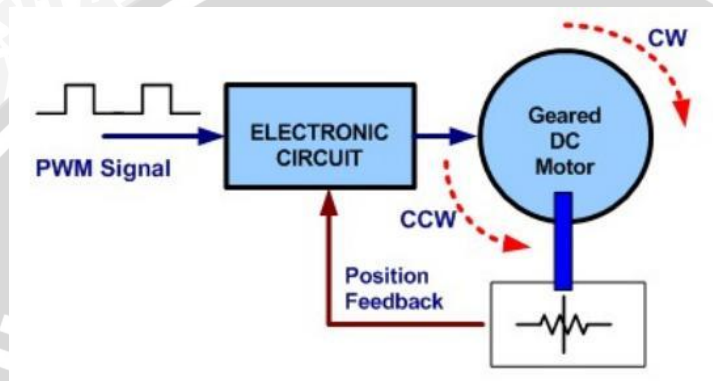
Titik-titik hitam pada Gambar 2.8 menunjukkan kaki kuda yang menapak ke tanah. Pada saat berjalan, kuda menapakkan ketiga kakinya di atas tanah, seperti ditunjukkan dalam Gambar 2.8 (a), inilah yang disebut dengan keseimbangan statis. Tetapi pada saat berlari, kaki kuda yang menapak ke tanah hanya ada dua bahkan hanya satu saja seperti ditunjukkan dalam Gambar 2.8 (b). Posisi kaki kuda pada saat berlari yang hanya menapakkan dua atau satu kakinya ke tanah inilah yang disebut dengan keseimbangan dinamis.

2.5 Motor DC Servo

Motor servo adalah motor yang mampu bekerja dua arah (searah jarum jam dan berlawanan arah jarum jam) dimana arah dan sudut pergerakan rotornya dapat dikendalikan hanya dengan memberikan pengaturan *duty cycle* sinyal PWM pada bagian pin kontrolnya. Motor servo adalah jenis motor yang digunakan sebagai

penggerak pada sistem servo (*servo-system*) seperti pada penggerak pada kontrol posisi lengan robot. Motor servo secara struktur mesin listrik ada 2 macam: DC servo motor dan AC servo motor.

Berbeda dengan motor DC dan motor stepper, motor servo adalah sebuah motor dengan sistem *closed feedback* di mana posisi dari motor akan diinformasikan kembali ke rangkaian kontrol yang ada di dalam motor servo. Motor ini terdiri atas sebuah motor, serangkaian *internal gear*, potensiometer dan rangkaian kontrol. Blok diagram motor DC servo ditunjukkan dalam Gambar 2.9.



Gambar 2. 9. Blok Diagram Motor DC Servo

Sumber : Parallax, Inc.

Potensiometer berfungsi untuk menentukan batas sudut putaran servo. Sedangkan sudut sumbu motor servo diatur berdasarkan lebar pulsa yang dikirim melalui kaki sinyal dari kabel motor. Bentuk fisik dari motor servo ditunjukkan dalam Gambar 2.10.



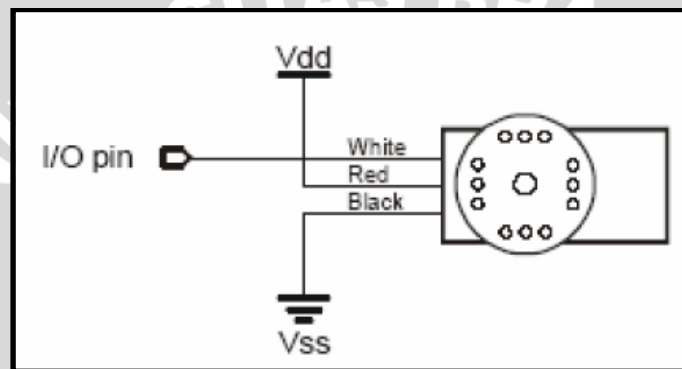
Gambar 2. 10. Motor Servo Parallax

Sumber : Parallax, Inc.

Karakteristik motor servo adalah sebagai berikut :

- 1). Operasional dari servo motor dikendalikan oleh sebuah pulsa selebar ± 20 ms, dimana lebar pulsa antara 0.5 ms dan 2 ms menyatakan akhir dari range sudut maksimum.
- 2). Memiliki 3 jalur kabel : *power*, *ground*, dan *control* seperti ditunjukkan dalam Gambar 6.
- 3). Pin *Control* untuk mengendalikan posisi.
- 4). Konstruksi didalamnya meliputi *internal gear*, *potensiometer*, dan *feedback control*.

Konfigurasi pin pengkabelan motor servo ditunjukkan dalam Gambar 2.11.



Gambar 2. 11. Konfigurasi Pin Pengkabelan Motor Servo

Sumber : Parallax, Inc.

Secara umum terdapat dua jenis motor servo, yaitu :

- 1). Motor Servo *Standar 180°*

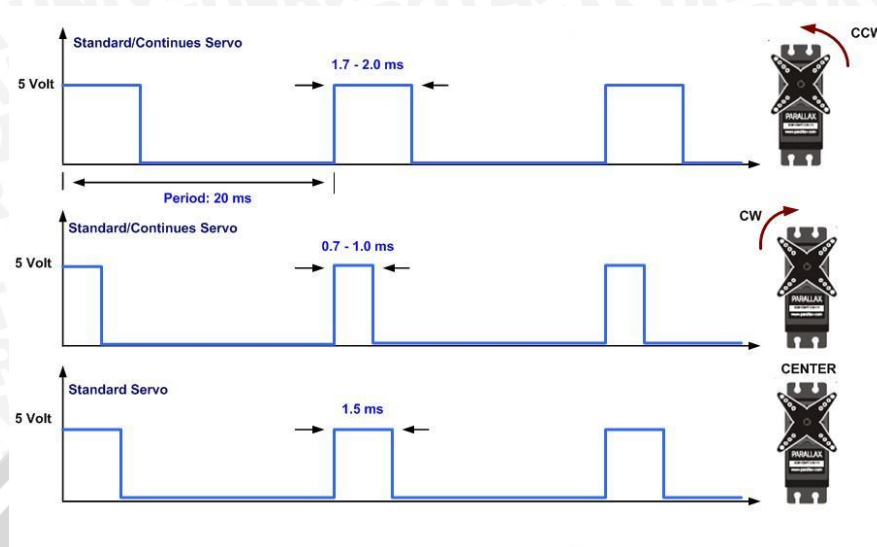
Motor servo jenis ini merupakan motor yang hanya mampu bergerak dua arah (*CW* dan *CCW*) dan mempunyai defleksi masing-masing sudut mencapai 90° sehingga total defleksi sudut dari kanan – tengah – kiri adalah 180° .

- 2). Motor Servo *Continuous*

Motor servo jenis ini mampu bergerak dua arah (*CW* dan *CCW*) dan tanpa batasan defleksi sudut putar (dapat berputar secara kontinyu).

Motor servo dapat bekerja dengan baik jika sinyal PWM (*Pulse Width Modulation*) berada pada frekuensi 50 Hz, yang berarti sinyal PWM memiliki periode 20 ms. Kontroler di dalam motor servo akan merespons sinyal PWM tersebut, lebar pulsa 0,7 ms – 1 ms akan memutar motor servo searah jarum jam, lebar pulsa 1,7 ms – 2 ms akan memutar motor servo berlawanan arah jarum jam. Sedangkan lebar pulsa

sebesar 1,5 ms akan memutar motor servo pada posisi tengah. Diagram waktu teknik PWM pada motor servo Parallax ditunjukkan dalam Gambar 2.12.



Gambar 2. 12. Diagram waktu PWM motor servo

Sumber : Parallax, Inc.

Motor servo yang digunakan adalah Tower Pro dengan tipe MG995R. Tipe tersebut memiliki spesifikasi sebagai berikut :

- Torsi: 10 kg-cm pada 4,8 V DC dan 12 kg-cm pada 6,0 V DC.
- Kecepatan: 0,23 detik/60° pada 4,8 V DC dan 0,2 detik/60° pada 6 V DC.
- Dimensi: 40,7 x 19,7 x 42,9 mm.
- Berat: 55 gr.

2.6 Mikrokontroler ATmega8

Mikrokontroler ATmega8 merupakan mikrokontroler keluaran AVR. Mikrokontroler ATmega8 merupakan mikrokontroler AVR CMOS 8 bit berdaya rendah. Mikrokontroler ATmega8 dapat mengeksekusi instruksi hingga 1 MIPS per MHz dalam satu siklus waktu. Karakteristik utama yang dimiliki oleh mikrokontroler ATmega8 adalah :

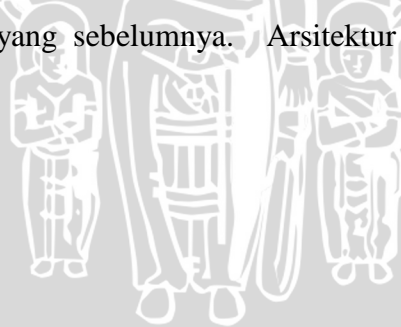
- 1). Memori program dan data yang *nonvolatile*.
- 2). Sistem *self-programable flash* 8 kByte.
- 3). EEPROM sebesar 512 Byte, dan 1kByte SRAM internal.
- 4). 23 saluran I/O dan 32 *general purpose register*.
- 5). Dua timer/counter 8 bit dengan *prescaler* terpisah, dan mode perbandingan (*compare mode*).

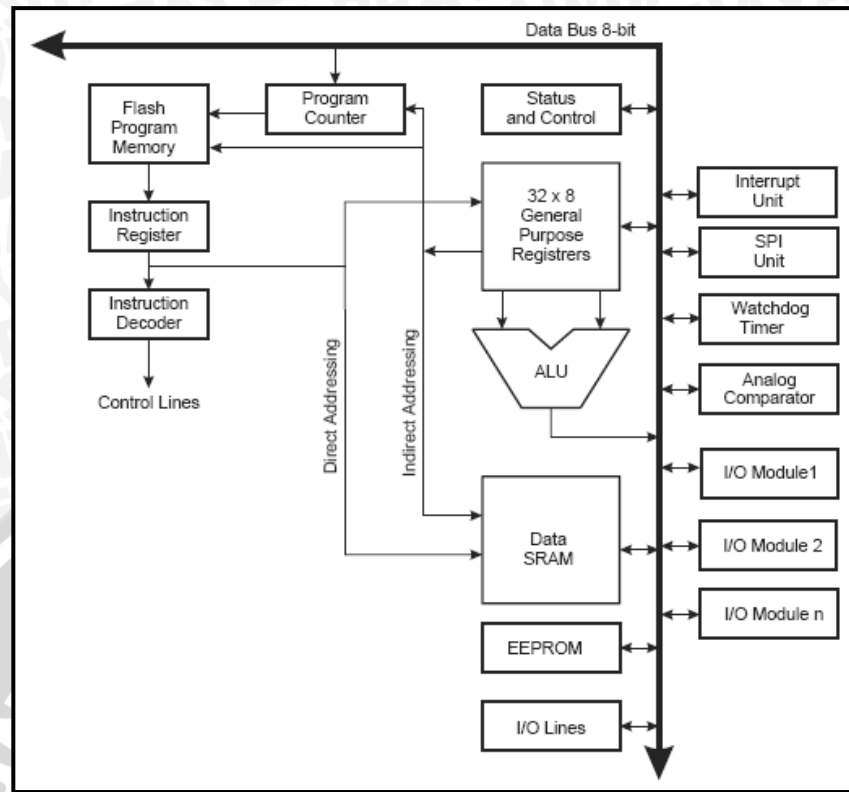
- 6). Satu buah timer/counter 16 bit dengan *prescaler* terpisah, mode pembandingan dan perekam (*capture*).
- 7). Internal dan eksternal *interrupt*.
- 8). Tiga buah pin PWM.
- 9). Serial USART.

2.6.1 Arsitektur AVR

Mikrokontroler AVR menggunakan arsitektur Harvard yang memisahkan memori dan bus untuk program dan data sehingga memaksimalkan performa. Instruksi pada memori program dieksekusi secara *pipeline*. Ketika satu instruksi masih dieksekusi, instruksi selanjutnya sudah disiapkan untuk eksekusi tanpa menunggu eksekusi pertama selesai. Konsep ini memungkinkan instruksi untuk dieksekusi setiap satu siklus waktu.

Register file terdiri atas 32 x 8 bit *general purpose working register* dengan waktu akses satu siklus waktu, yang memungkinkan operasi ALU (*Arithmetic Logic Unit*) dijalankan dalam satu siklus waktu. Dua *operand* diambil dari register, operasi ALU dijalankan, dan hasilnya disimpan kembali dalam *register file*. Operasi aritmetika dan logika dapat dijalankan oleh ALU, baik antar register atau antara register dengan konstanta. Hasil dari operasi aritmetika disimpan dalam register status (*Status Register*), menggantikan isi yang sebelumnya. Arsitektur AVR ditunjukkan dalam Gambar 2.13.





Gambar 2. 13. Arsitektur AVR

Sumber : Atmel, 2004 : 7

Program *Flash memory* dibagi dalam dua bagian, *Boot Program* dan *Application Program*. Kedua bagian memiliki *Lock bits* untuk mengunci operasi tulis (*write*) dan baca/tulis (*read/write*). Konfigurasi pin mikrokontroler ATMega8 dapat dilihat dalam Gambar 2.14.

(RESET) PC6	1	28	PC5 (ADC5/SCL)
(RXD) PD0	2	27	PC4 (ADC4/SDA)
(TXD) PD1	3	26	PC3 (ADC3)
(INT0) PD2	4	25	PC2 (ADC2)
(INT1) PD3	5	24	PC1 (ADC1)
(XCK/T0) PD4	6	23	PC0 (ADC0)
VCC	7	22	GND
GND	8	21	AREF
(XTAL1/TOSC1) PB6	9	20	AVCC
(XTAL2/TOSC2) PB7	10	19	PB5 (SCK)
(T1) PD5	11	18	PB4 (MISO)
(AIN0) PD6	12	17	PB3 (MOSI/OC2)
(AIN1) PD7	13	16	PB2 (SS/OC1B)
(ICP1) PB0	14	15	PB1 (OC1A)

Gambar 2. 14. Konfigurasi pin ATMega8

Sumber : Atmel, 2004 : 2

2.6.2 Struktur dan Operasi Port

Mikrokontroler ATmega8 ini mempunyai 4 buah port, yang memiliki 8 buah jalur I/O. Beberapa karakteristik port mikrokontroler ATmega8 dijelaskan secara singkat :

- 1). Unit I/O dapat dialamati perjalur atau per port
- 2). Setiap jalur I/O memiliki buffer, penahan (latch), kemudi input dan kemudi output.
- 3). Setiap jalur I/O terdapat register pengatur apakah dijadikan input atau dijadikan output.
- 4). Port B adalah I/O *bi-directional* 8 bit dengan resistor *pull-up* internal. Sebagai masukan, pin port B yang diberi *pull-low* secara eksternal akan mengalirkan arus bila resistor *pull-up* diaktifkan. Port B juga memiliki fungsi khusus, seperti yang terlihat dalam Tabel 2.1.

Tabel 2. 1. Fungsi khusus Port B ATmega8

Port Pin	Alternate Functions
PB7	XTAL2 (Chip Clock Oscillator pin 2) TOSC2 (Timer Oscillator pin 2)
PB6	XTAL1 (Chip Clock Oscillator pin 1 or External clock input) TOSC1 (Timer Oscillator pin 1)
PB5	SCK (SPI Bus Master clock Input)
PB4	MISO (SPI Bus Master Input/Slave Output)
PB3	MOSI (SPI Bus Master Output/Slave Input) OC2 (Timer/Counter2 Output Compare Match Output)
PB2	\overline{SS} (SPI Bus Master Slave select) OC1B (Timer/Counter1 Output Compare Match B Output)
PB1	OC1A (Timer/Counter1 Output Compare Match A Output)
PB0	ICP1 (Timer/Counter1 Input Capture Pin)

Sumber : Atmel, 2004 : 56

- 5). Port C adalah I/O *bi-directional* 8 bit dengan resistor *pull-up* internal. Sebagai masukan, pin port C yang diberi *pull-low* secara eksternal akan mengalirkan arus bila resistor *pull-up* diaktifkan. Port C juga memiliki fungsi khusus, seperti yang terlihat dalam Tabel 2.2.

Tabel 2. 2. Fungsi khusus Port C ATmega8

Port Pin	Alternate Function
PC6	RESET (Reset pin)
PC5	ADC5 (ADC Input Channel 5) SCL (Two-wire Serial Bus Clock Line)
PC4	ADC4 (ADC Input Channel 4) SDA (Two-wire Serial Bus Data Input/Output Line)
PC3	ADC3 (ADC Input Channel 3)
PC2	ADC2 (ADC Input Channel 2)
PC1	ADC1 (ADC Input Channel 1)
PC0	ADC0 (ADC Input Channel 0)

Sumber : Atmel, 2004 : 59

- 6). Port D adalah I/O *bi-directional* 8 bit dengan resistor *pull-up* internal. Sebagai masukan, pin port D yang diberi *pull-low* secara eksternal akan mengalirkan arus bila resistor *pull-up* diaktifkan. Port D juga memiliki fungsi khusus, seperti yang terlihat dalam Tabel 2.3.

Tabel 2. 3. Fungsi khusus Port D ATmega8

Port Pin	Alternate Function
PD7	AIN1 (Analog Comparator Negative Input)
PD6	AIN0 (Analog Comparator Positive Input)
PD5	T1 (Timer/Counter 1 External Counter Input)
PD4	XCK (USART External Clock Input/Output) T0 (Timer/Counter 0 External Counter Input)
PD3	INT1 (External Interrupt 1 Input)
PD2	INT0 (External Interrupt 0 Input)
PD1	TXD (USART Output Pin)
PD0	RXD (USART Input Pin)

Sumber : Atmel, 2004 : 61

2.6.3 Timer/Counter

Mikrokontroler ATmega8 memiliki 3 buah *timer/counter* yang terdiri atas 2 buah *timer/counter* 8 bit dan 1 buah *timer/counter* 16 bit. Ketiga *timer/counter* ini dapat diatur dalam mode yang berbeda. Selain itu semua *timer/counter* dapat difungsikan sebagai sumber interupsi. *Timer/counter* dapat digunakan dalam 4 mode operasi, yaitu :

- 1). Mode pertama (mode 0) adalah mode normal, timer digunakan sebagai pencacah tunggal yang dapat mencacah dari 0x00 sampai dengan 0xFF. Setelah mencapai nilai 0xFF maka register *counter* akan reset atau kembali ke 0x00.
- 2). Mode kedua (mode 1) adalah *phase correct* PWM (PCP). Mode ini digunakan untuk menghasilkan sinyal PWM dimana nilai register *counter*

yang mencacah naik dan turun secara terus menerus akan selalu dibandingkan dengan register pembanding OCRn. Hasil perbandingan register *counter* dan OCRn digunakan untuk membangkitkan sinyal PWM yang dikeluarkan pada pin OCn.

- 3). Mode ketiga (mode 2) adalah *clear timer on compare match* (CTC). Register *counter* akan mencacah naik kemudian akan direset atau kembali menjadi 0x00 pada saat nilai TCNT sama dengan OCRn.
- 4). Mode keempat (mode 3) adalah *fast PWM*. Mode ini hampir sama dengan mode *phase correct PWM*, hanya perbedaannya adalah register *counter* mencacah naik saja dan tidak mencacah turun.



BAB III

METODOLOGI PENELITIAN

Metode penyelesaian masalah diperlukan untuk menyelesaikan rumusan masalah dan merealisasikan tujuan penelitian yang terdapat di bab pendahuluan.

3.1 Penentuan Spesifikasi Alat

Sebelum melakukan perencanaan dan perealisasiian alat, maka ditentukan spesifikasi alat yang akan dibuat. Adapun spesifikasi alat yang akan direalisasikan sebagai berikut:

- 1). Robot memiliki 4 buah kaki.
- 2). Setiap kaki robot memiliki 2 derajat kebebasan ($2\ DOF$) dengan menggunakan motor servo sebagai sendinya.
- 3). Motor servo yang digunakan adalah Tower Pro dengan tipe MG995R.
- 4). Dimensi maksimum robot adalah 46 cm x 31 cm x 27 cm (sesuai Peraturan Kontes Robot Cerdas Indonesia 2010).
- 5). Mikrokontroler yang digunakan adalah ATmega8 sebagai pengontrol pergerakan motor servo.
- 6). Pemrograman mikrokontroler ATmega8 dengan software *CodeVisionAVR C Compiler*.
- 7). Arena yang digunakan berupa bidang datar (kemiringan 0°).
- 8). Pergerakan robot adalah gerak maju, gerak mundur, gerak hadap kanan (90° ke kanan), gerak hadap kiri (90° ke kiri), dan gerak putar 180° .

3.2 Perancangan dan Pembuatan Alat

Perancangan dan pembuatan alat meliputi perancangan perangkat keras dan perancangan perangkat lunak.

3.2.1 Perancangan dan Pembuatan Perangkat Keras

Secara garis besar perancangan perangkat keras dibagi dalam beberapa tahap berikut :



- 1). Penentuan spesifikasi alat
- 2). Pembuatan blok diagram keseluruhan sistem
- 3). Pembuatan mekanik robot
- 4). Penentuan desain pergerakan robot
- 5). Penentuan dan perhitungan komponen yang akan digunakan

3.2.2 Perancangan dan Penyusunan Perangkat Lunak

Penyusunan perangkat lunak digunakan untuk mengendalikan dan mengatur kerja dari alat ini. Desain dan parameter yang telah dirancang kemudian diterapkan kedalam mikrokontroler ATmega8 dengan menggunakan bahasa C dan *compiler CodeVision AVR*.

3.3 Pengujian Alat

Pengujian alat dilakukan untuk memastikan bahwa sistem ini berjalan sesuai yang direncanakan. Pengujian alat meliputi pengujian perangkat keras yang dilakukan baik per blok maupun keseluruhan sistem.

3.3.1 Pengujian Tiap Blok

Pengujian per blok dilakukan dengan tujuan untuk menyesuaikan nilai masukan dan nilai keluaran tiap-tiap blok sesuai dengan perancangan yang dilakukan sebelumnya.

3.3.2 Pengujian Keseluruhan Sistem

Pengujian sistem secara keseluruhan dilakukan dengan tujuan untuk mengetahui unjuk kerja alat setelah perangkat keras dan perangkat lunak diintegrasikan bersama.

3.4 Pengambilan Kesimpulan

Pengambilan kesimpulan dilakukan setelah didapatkan hasil dari pengujian. Jika hasil yang diperoleh telah sesuai dengan spesifikasi yang direncanakan maka alat tersebut telah memenuhi harapan dan memerlukan pengembangan untuk penyempurnaannya.

BAB IV

PERANCANGAN DAN PEMBUATAN ALAT

Perancangan Robot Berkaki Empat ini dilakukan secara bertahap sehingga akan memudahkan dalam analisis pada setiap bloknya maupun secara keseluruhan. Perancangan ini terdiri dari:

- Perancangan mekanik robot,
- Perancangan gerakan robot,
- Perancangan system,
- Perancangan perangkat keras (rangkaiian catu daya sistem, rangkaiian mikrokontroler kontrol servo, rangkaiian *rotary switch*, dan motor servo),
- Perancangan perangkat lunak (metode pengontrolan multiservo dan algoritma mikrokontroler kontrol servo).

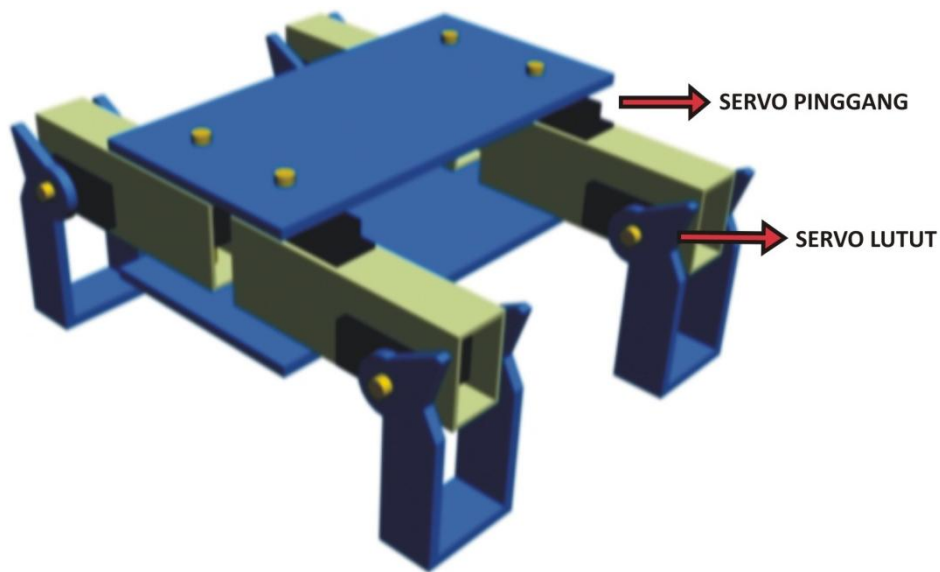
4.1 Perancangan Mekanik Robot

Berdasarkan Peraturan Kontes Robot Cerdas Indonesia (KRCI) Divisi Berkaki tahun 2010, batasan dimensi robot adalah sebagai berikut :

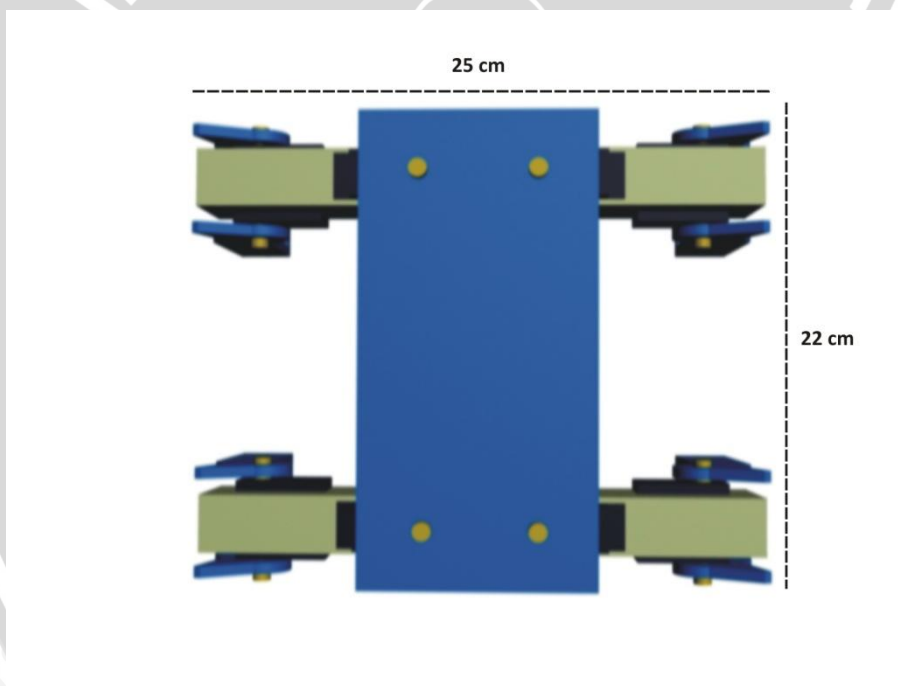
- panjang maksimum = 46 cm
- lebar maksimum = 31 cm
- tinggi maksimum = 27 cm

Dengan adanya peraturan tersebut, maka dimensi Robot Berkaki Empat harus dirancang agar tidak melebihi batas ukuran yang telah ditetapkan. Gambar perspektif dan rancangan ukuran Robot Berkaki Empat ditunjukkan dalam Gambar 4.1, Gambar 4.2, dan Gambar 4.3.

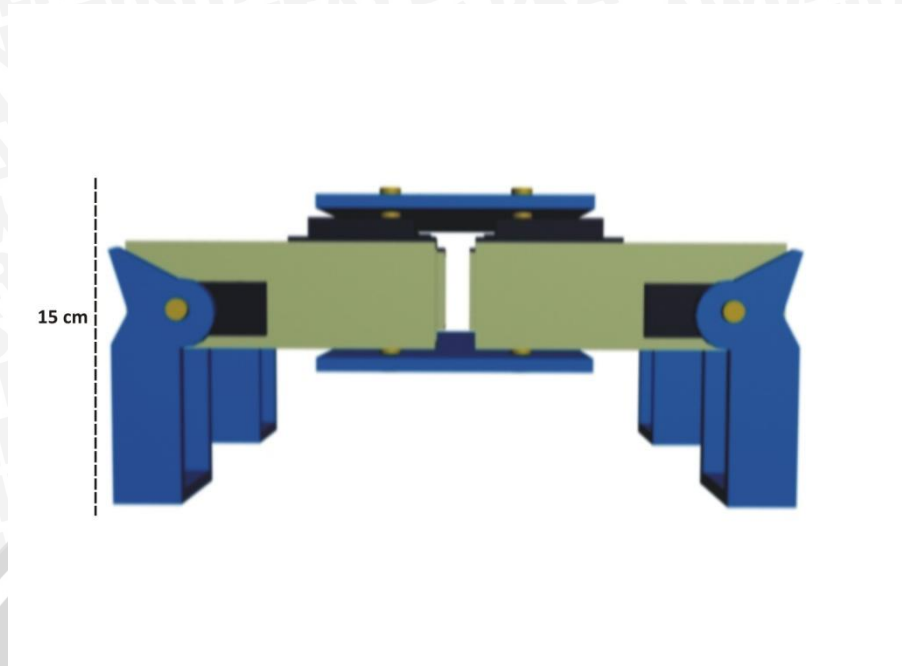




Gambar 4. 1. Perspektif Desain Mekanik Robot Berkaki Empat

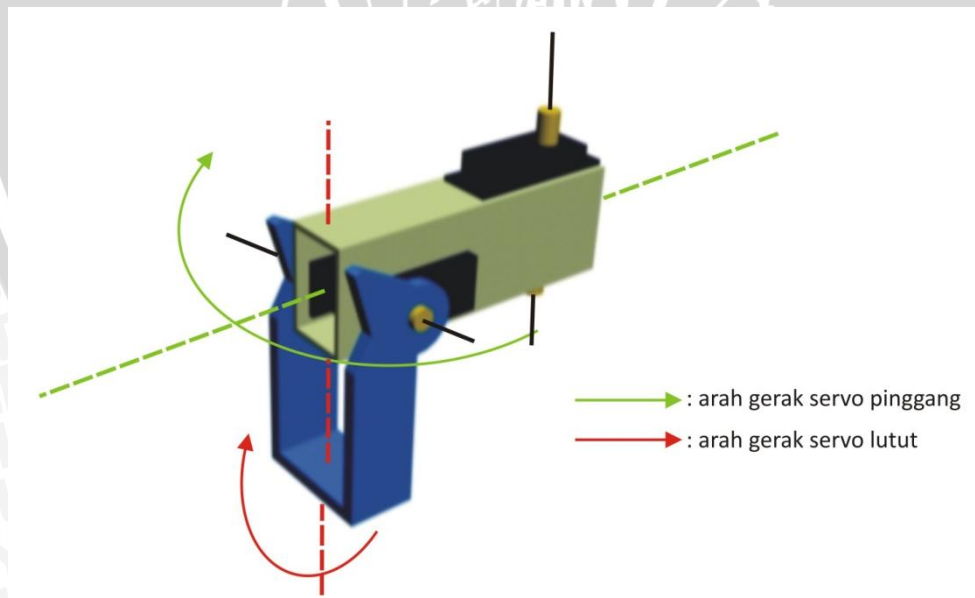


Gambar 4. 2. Robot Berkaki Empat tampak atas



Gambar 4. 3. Robot Berkaki Empat tampak depan

Dari Gambar 4.2 dan Gambar 4.3 diperoleh dimensi dari Robot Berkaki Empat yaitu panjang 22 cm, lebar 25 cm, dan tinggi 15 cm. Pada setiap kaki terdapat dua motor DC servo yang berfungsi sebagai sendi kaki robot. Posisi dan arah gerak dari masing-masing motor DC servo pada kaki robot ditunjukkan dalam Gambar 4.4.



Gambar 4. 4. Bentuk kaki robot dan arah gerak tiap motor DC servo

Servo pinggang adalah servo yang berfungsi untuk menggerakkan kaki robot ke depan dan ke belakang. Posisi servo ini berada pada bagian kaki yang menempel pada tubuh robot. Servo ini memiliki peran utama untuk menggerakkan robot

dengan mengkombinasikan gerakan antara keempat servo pinggang. Arah gerakan servo ini ditunjukkan dengan arah panah warna hijau pada Gambar 4.4.

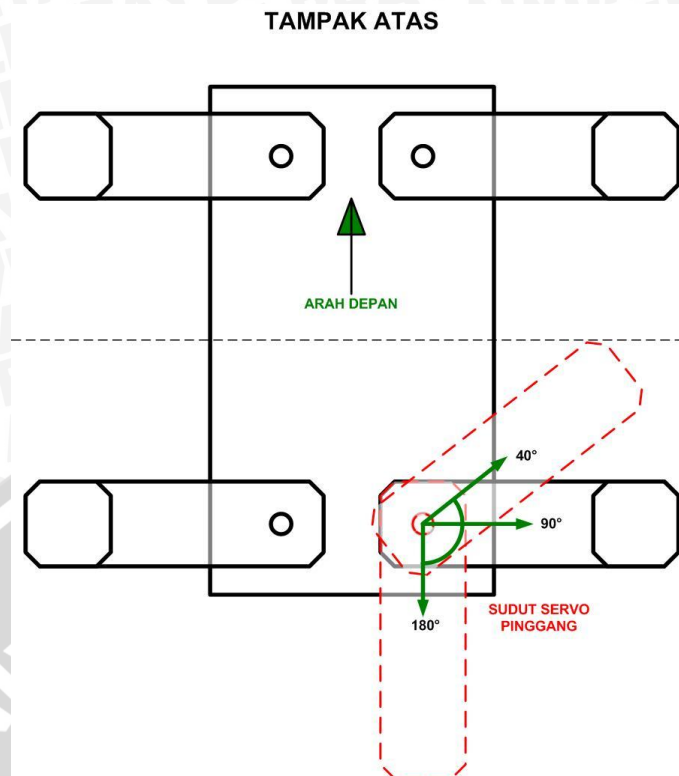
Sedangkan servo lutut adalah servo yang berfungsi untuk menggerakkan kaki bagian bawah ke atas dan ke bawah. Gerakan ini berfungsi untuk membantu robot mengurangi bidang gesekan antara kaki robot dengan alas. Pada saat melangkah, servo pinggang menggerakkan paha ke depan atau ke belakang bersamaan dengan diangkatnya servo lutut ke atas (arah luar). Arah gerakan servo ini ditunjukkan dengan arah panah warna merah dalam Gambar 4.4.

4.2 Perancangan Gerakan

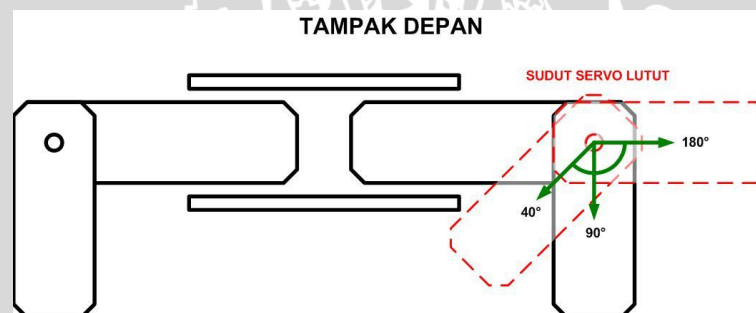
Perancangan gerakan pada Robot Berkaki Empat meliputi perancangan daerah kerja sendi robot, perancangan posisi dasar robot, dan perancangan gerakan dasar Robot Berkaki Empat.

4.2.1 Daerah Kerja Sendi Kaki Robot

Robot Berkaki Empat memiliki empat kaki dimana masing-masing kaki memiliki dua sendi (motor DC servo). Setiap servo ini memiliki rentang sudut putar tertentu sesuai letak persendian masing-masing. Motor DC Servo yang digunakan dalam perancangan ini adalah motor DC servo standar dengan kemampuan pergerakan dua arah yaitu CW (*clockwise*) dan CCW (*counter clockwise*), dengan simpangan maksimum untuk tiap arah gerakan sebesar 90° . Dengan kata lain tiap motor DC servo mampu bergerak dengan simpangan antara 0° hingga 180° . Akan tetapi desain mekanik pada Robot Berkaki Empat menyebabkan rentang sudut putar servo tidak bisa mencapai 180° . Simpangan maksimum untuk masing-masing motor DC servo ditunjukkan dalam Gambar 4.5 dan Gambar 4.6.



Gambar 4. 5. Batas Simpangan Maksimum Servo Pinggang

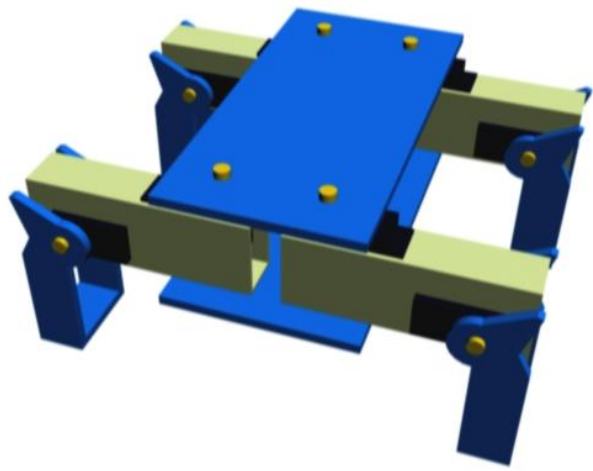


Gambar 4. 6. Batas Simpangan Maksimum Servo Lutut

Berdasarkan Gambar 4.5 dapat diketahui bahwa sudut simpangan maksimum untuk servo pinggang adalah antara 40° hingga 180° . Dan dari Gambar 4.6 dapat diketahui bahwa sudut simpangan maksimum untuk servo lutut adalah antara 40° hingga 180° .

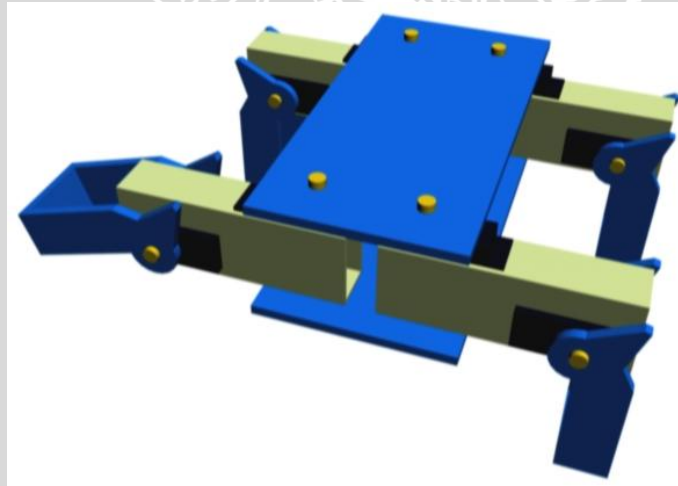
4.2.2 Posisi Dasar Robot

Robot Berkaki Empat berada pada posisi dasar pada saat diam. Pada posisi ini robot berada dalam keadaan seimbang, dimana keempat kakinya menginjak alas, dan semua motor DC servo membentuk sudut 90° . Bentuk perspektif robot pada keadaan diam ditunjukkan dalam Gambar 4.7.

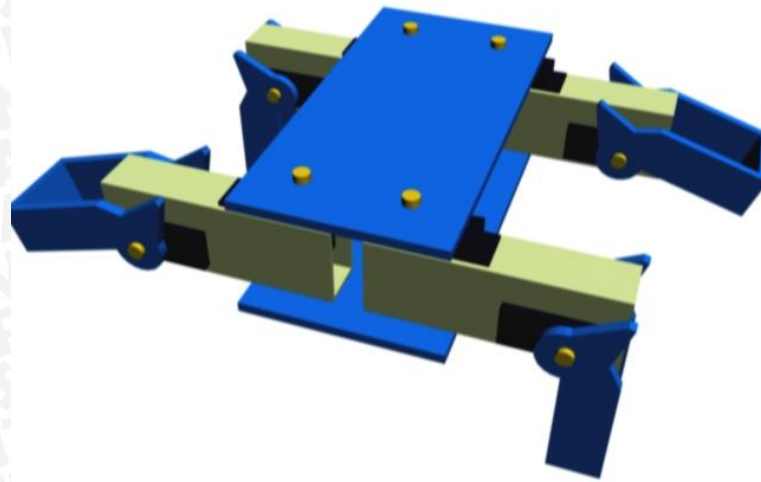


Gambar 4. 7. Keseimbangan Robot dengan 4 Kaki

Selain posisi seperti ditunjukkan dalam Gambar 4.7 di atas, terdapat beberapa posisi lain dimana robot masih berada pada kondisi seimbang dan belum jatuh, yaitu pada kondisi dimana robot ditopang oleh ketiga kakinya, dan juga pada kondisi dimana robot hanya ditopang oleh kedua kakinya. Posisi-posisi robot dalam keadaan ini ditunjukkan dalam Gambar 4.8 dan Gambar 4.9.



Gambar 4. 8. Keseimbangan Robot dengan 3 Kaki

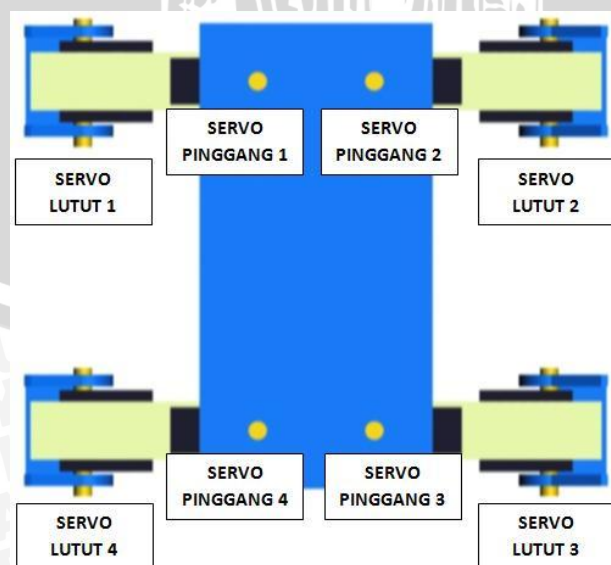


Gambar 4. 9. Keseimbangan Robot dengan 2 Kaki

Posisi kaki robot seperti dalam Gambar 4.9 di atas merupakan kunci metode pergerakan Robot Berkaki Empat. Karena pada saat melangkah robot akan berada pada kondisi hanya ditopang oleh dua kaki yang saling silang secara bergantian. Pembahasan mengenai gerak melangkah ini akan dibahas lebih lanjut dalam sub bab selanjutnya.

4.2.3 Gerakan Dasar Robot Berkaki Empat

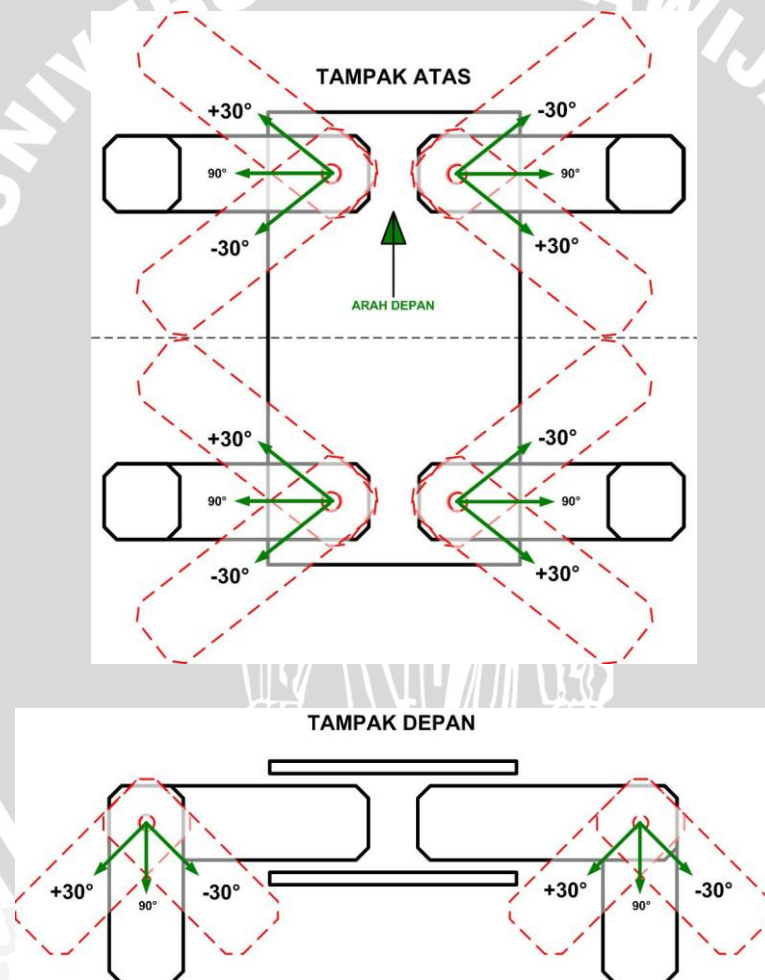
Robot Berkaki Empat memiliki beberapa gerakan dasar, yaitu gerak maju, gerak mundur, gerak hadap kanan, gerak hadap kiri, dan gerak berputar. Untuk mempermudah perancangan gerakan, maka setiap motor DC servo pada robot diberi nama sesuai letaknya. Letak dan nama masing-masing servo ditunjukkan dalam Gambar 4.10.



Gambar 4. 10. Posisi Motor DC Servo pada Robot Berkaki Empat

4.2.3.1 Gerak Maju

Gerak maju pada Robot Berkaki Empat membutuhkan kombinasi gerakan yang selaras antara ke delapan servo pada robot. Sebelum merancang metode pergerakannya, harus diketahui terlebih dahulu batas-batas daerah kerja dari tiap-tiap servo. Berdasarkan gambar 4.12 dan Gambar 4.13 dapat diketahui bahwa sudut simpangan maksimum untuk servo pinggang adalah antara 40° hingga 180° dan untuk servo lutut adalah antara 40° hingga 180° . Maka pada perancangan ini servo pinggang sudut simpangannya hanya dibatasi antara 60° hingga 120° . Begitu juga pada servo lutut sudut simpangannya dibatasi antara 60° hingga 120° . Daerah kerja motor DC servo ditunjukkan dalam Gambar 4.11.



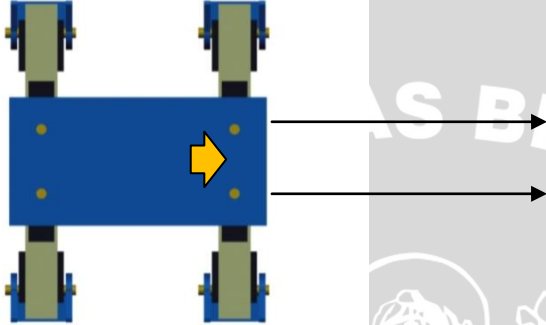
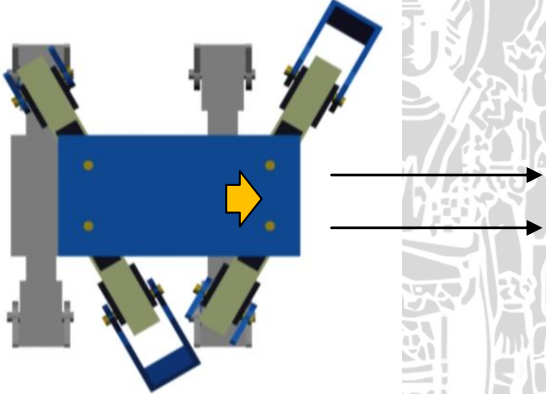
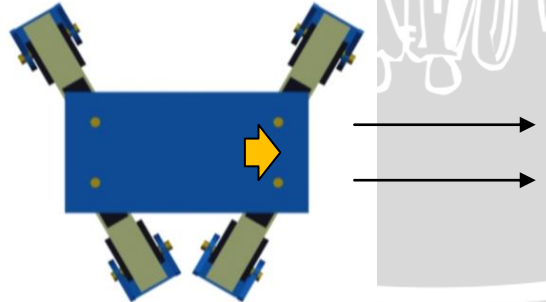
Gambar 4. 11. Daerah Kerja Motor DC Servo pada Robot Berkaki Empat

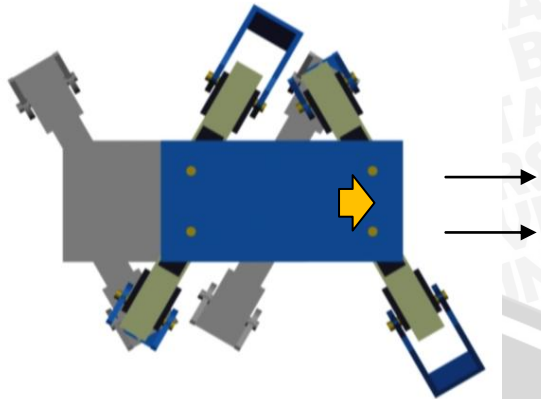
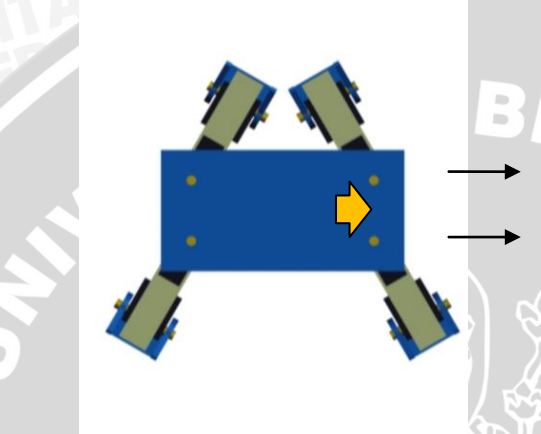
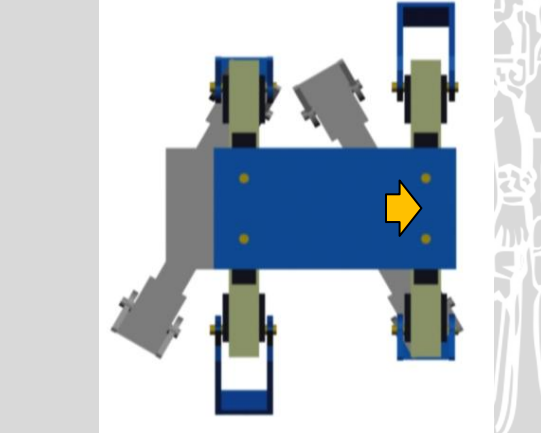
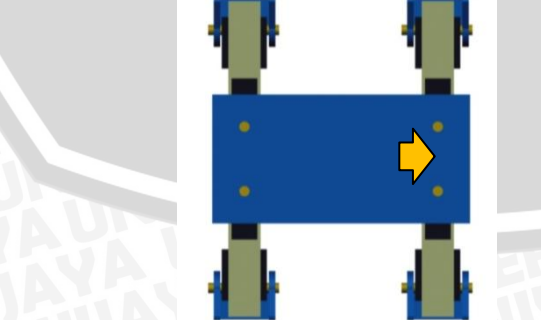
Untuk mempermudah perhitungan sudut, maka sudut-sudut motor servo dituliskan dalam bentuk matriks. Bentuk matriks sudut servo Robot Berkaki Empat adalah :

$$\text{Posisi} = \begin{bmatrix} \text{servo pinggang 1} & \text{servo lutut 1} \\ \text{servo pinggang 2} & \text{servo lutut 2} \\ \text{servo pinggang 3} & \text{servo lutut 3} \\ \text{servo pinggang 4} & \text{servo lutut 4} \end{bmatrix} = \begin{bmatrix} P1 & L1 \\ P2 & L2 \\ P3 & L3 \\ P4 & L4 \end{bmatrix}$$

Tahap-tahap gerak maju pada robot berkaki empat ditunjukkan dalam Tabel 4.1.

Tabel 4. 1. Tahap-tahap Gerak Maju Robot Berkaki Empat

Tahap	Gambar	Sudut Servo
1.		Posisi dasar = $\begin{bmatrix} 90^\circ & 90^\circ \\ 90^\circ & 90^\circ \\ 90^\circ & 90^\circ \\ 90^\circ & 90^\circ \end{bmatrix}$
2.		Posisi 1 = $\begin{bmatrix} 120^\circ & 60^\circ \\ 120^\circ & 90^\circ \\ 60^\circ & 120^\circ \\ 60^\circ & 90^\circ \end{bmatrix}$
3.		Posisi 2 = $\begin{bmatrix} 120^\circ & 90^\circ \\ 120^\circ & 90^\circ \\ 60^\circ & 90^\circ \\ 60^\circ & 90^\circ \end{bmatrix}$

4.		$\text{Posisi 3} = \begin{bmatrix} 60^\circ & 90^\circ \\ 60^\circ & 120^\circ \\ 120^\circ & 90^\circ \\ 120^\circ & 60^\circ \end{bmatrix}$
5.		$\text{Posisi 4} = \begin{bmatrix} 60^\circ & 90^\circ \\ 60^\circ & 90^\circ \\ 120^\circ & 90^\circ \\ 120^\circ & 90^\circ \end{bmatrix}$
6.		$\text{Posisi 5} = \begin{bmatrix} 90^\circ & 60^\circ \\ 90^\circ & 90^\circ \\ 90^\circ & 120^\circ \\ 90^\circ & 90^\circ \end{bmatrix}$
7.		$\text{Posisi dasar} = \begin{bmatrix} 90^\circ & 90^\circ \\ 90^\circ & 90^\circ \\ 90^\circ & 90^\circ \\ 90^\circ & 90^\circ \end{bmatrix}$

➡ : Arah depan robot

→ : Arah gerak robot

Tahap 1-7 seperti ditunjukkan dalam Tabel 4.1 di atas merupakan satu siklus gerak maju pada robot berkaki empat. Nilai-nilai sudut dalam bentuk matriks

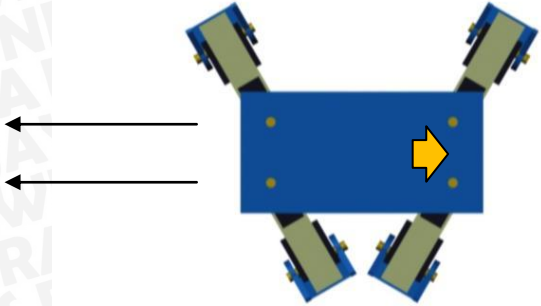
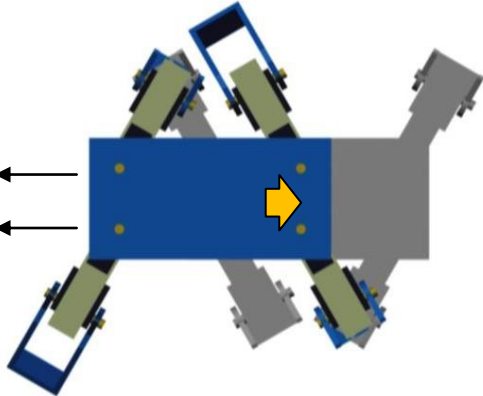
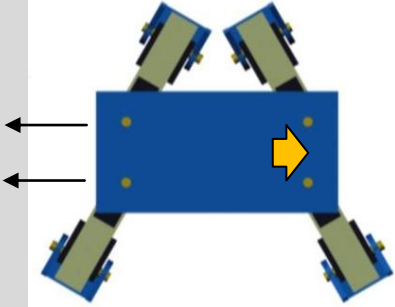
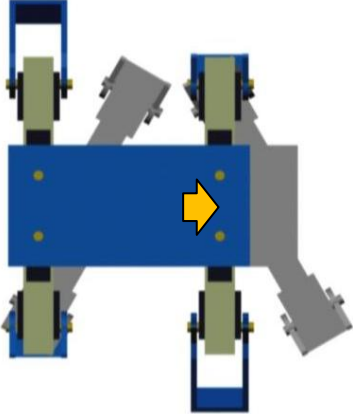
pada setiap tahap merupakan nilai yang harus dimasukkan ke dalam program agar motor DC servo dapat bergerak sesuai sudut yang sudah ditetapkan. Untuk dapat bergerak pada jarak tertentu, maka tahap-tahap di atas harus diulang beberapa kali sampai mencapai jarak yang diinginkan.

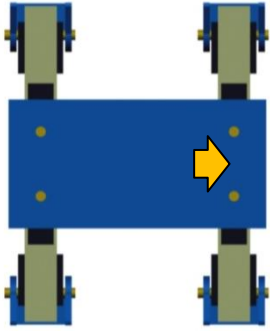
4.2.3.2 Gerak Mundur

Metode gerak mundur pada Robot Berkaki Empat sama persis dengan metode gerak majunya, hanya arahnya saja yang berlawanan. Sama seperti gerak maju, servo yang memiliki fungsi utama untuk menggerakkan tubuh robot adalah servo pinggang, sedangkan servo lutut berfungsi untuk mengangkat kaki pada saat melangkah sehingga dapat mengurangi gesekan antara telapak kaki dengan alas. Tahap-tahap gerak mundur pada robot berkaki empat ditunjukkan dalam Tabel 4.2.

Tabel 4. 2. Tahap-tahap Gerak Mundur Robot Berkaki Empat

Tahap	Gambar	Sudut Servo
1.		Posisi dasar = $\begin{bmatrix} 90^\circ & 90^\circ \\ 90^\circ & 90^\circ \\ 90^\circ & 90^\circ \\ 90^\circ & 90^\circ \end{bmatrix}$
2.		Posisi 1 = $\begin{bmatrix} 120^\circ & 90^\circ \\ 120^\circ & 120^\circ \\ 60^\circ & 90^\circ \\ 60^\circ & 60^\circ \end{bmatrix}$

<p>3.</p>		<p>Posisi 2 = $\begin{bmatrix} 120^\circ & 90^\circ \\ 120^\circ & 90^\circ \\ 60^\circ & 90^\circ \\ 60^\circ & 90^\circ \end{bmatrix}$</p>
<p>4.</p>		<p>Posisi 3 = $\begin{bmatrix} 60^\circ & 60^\circ \\ 60^\circ & 90^\circ \\ 120^\circ & 120^\circ \\ 120^\circ & 90^\circ \end{bmatrix}$</p>
<p>5.</p>		<p>Posisi 4 = $\begin{bmatrix} 60^\circ & 90^\circ \\ 60^\circ & 90^\circ \\ 120^\circ & 90^\circ \\ 120^\circ & 90^\circ \end{bmatrix}$</p>
<p>6.</p>		<p>Posisi 5 = $\begin{bmatrix} 90^\circ & 90^\circ \\ 90^\circ & 120^\circ \\ 90^\circ & 90^\circ \\ 90^\circ & 60^\circ \end{bmatrix}$</p>

7.		Posisi dasar = $\begin{bmatrix} 90^\circ & 90^\circ \\ 90^\circ & 90^\circ \\ 90^\circ & 90^\circ \\ 90^\circ & 90^\circ \end{bmatrix}$
----	---	---

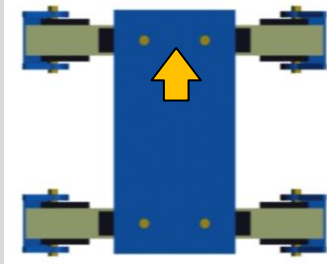
➡ : Arah depan robot

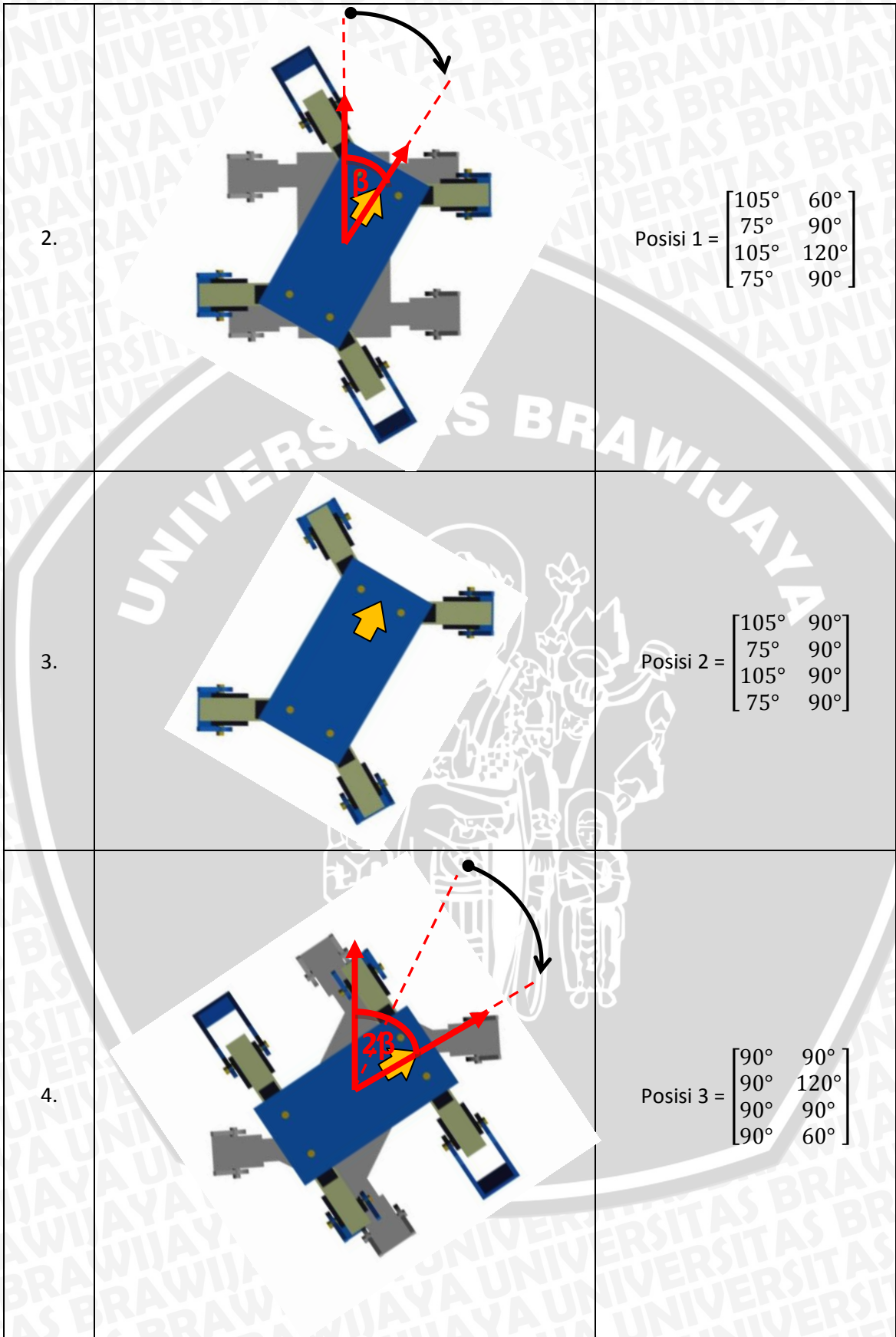
→ : Arah gerak robot

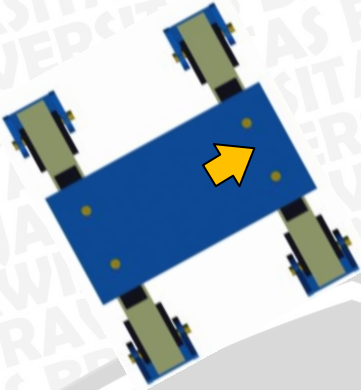
4.2.3.3 Gerak Hadap Kanan

Gerak hadap kanan pada Robot Berkaki Empat adalah gerak memutar badan robot ke kanan sebesar 90°. Tidak seperti manusia atau hewan berkaki dua lainnya yang bisa memutar tubuhnya langsung ke kanan, Robot Berkaki Empat memutar badan secara bertahap. Pada perancangan ini, Robot Berkaki Empat akan berputar sebesar 30°/siklus. Jadi untuk berputar sebesar 90°, robot harus mengulangi tahap-tahapnya sebanyak 3 kali. Siklus robot untuk berputar sebesar 30° ditunjukkan dalam Tabel 4.3.

Tabel 4. 3. Tahap-tahap Gerak Hadap Kanan Robot Berkaki Empat

Tahap	Gambar	Keterangan
1.		Posisi dasar = $\begin{bmatrix} 90^\circ & 90^\circ \\ 90^\circ & 90^\circ \\ 90^\circ & 90^\circ \\ 90^\circ & 90^\circ \end{bmatrix}$



5.		Posisi 4 = $\begin{bmatrix} 90^\circ & 90^\circ \\ 90^\circ & 90^\circ \\ 90^\circ & 90^\circ \\ 90^\circ & 90^\circ \end{bmatrix}$
----	---	---

$\beta = 15^\circ$ (sudut yang dibentuk oleh servo pinggang untuk memutar tubuh robot)

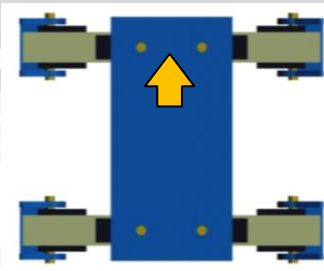
 = Arah depan robot

Dari Tabel 4.3 diatas, dapat diketahui bahwa robot berputar sebanyak dua kali, masing-masing sebesar 15° agar robot dapat memutar tubuhnya sebesar 30° dalam satu siklus. Putaran sudut 15° yang pertama terjadi pada tahap 2. Setelah itu robot menurunkan ujung kakinya pada tahap 3. Kemudian robot berputar sebesar 15° lagi pada tahap 4, dan menurunkan kakinya pada tahap 5. Sehingga total sudut yang dibentuk oleh robot adalah 30° .

4.2.3.4 Gerak Hadap Kiri

Gerak hadap kiri pada Robot Berkaki Empat identik dengan gerak hadap kanannya. Semua tahap dalam satu siklusnya sama persis dengan tahap-tahap gerak hadap kanan, hanya arah gerakanya saja yang berlawanan. Arah gerak yang berlawanan ini disebabkan oleh perbedaan kaki robot yang digunakan sebagai tumpuan dalam berpijak pada saat tubuh robot berputar. Tahap-tahap gerak hadap kiri pada Robot Berkaki Empat ditunjukkan dalam Tabel 4.4.

Tabel 4. 4. Tahap-tahap Gerak Hadap Kiri Robot Berkaki Empat

Tahap	Gambar	Keterangan
1.		Posisi dasar = $\begin{bmatrix} 90^\circ & 90^\circ \\ 90^\circ & 90^\circ \\ 90^\circ & 90^\circ \\ 90^\circ & 90^\circ \end{bmatrix}$

2.

Posisi 1 =
$$\begin{bmatrix} 105^\circ & 90^\circ \\ 75^\circ & 120^\circ \\ 105^\circ & 90^\circ \\ 75^\circ & 60^\circ \end{bmatrix}$$

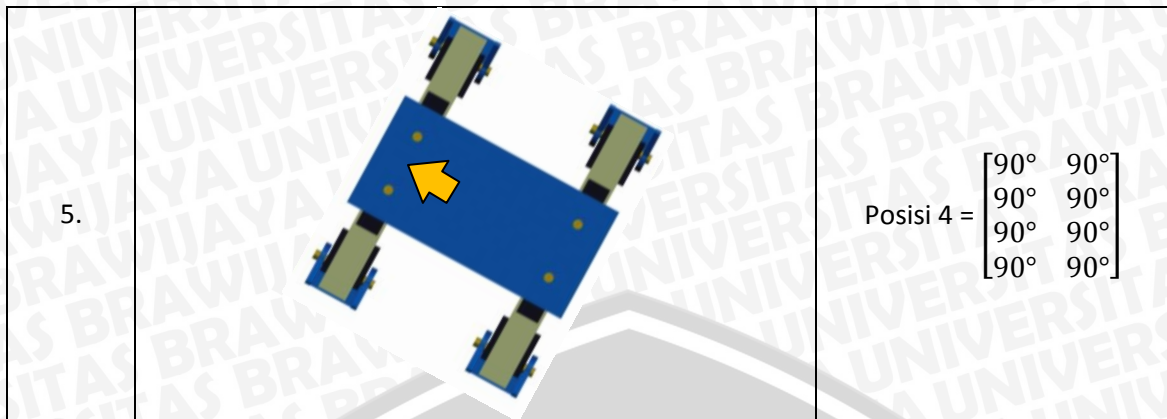
3.

Posisi 2 =
$$\begin{bmatrix} 105^\circ & 90^\circ \\ 75^\circ & 90^\circ \\ 105^\circ & 90^\circ \\ 75^\circ & 90^\circ \end{bmatrix}$$

4.

Posisi 3 =
$$\begin{bmatrix} 90^\circ & 60^\circ \\ 90^\circ & 90^\circ \\ 90^\circ & 120^\circ \\ 90^\circ & 90^\circ \end{bmatrix}$$





$\beta = 15^\circ$ (sudut yang dibentuk oleh servo pinggang untuk memutar tubuh robot)

 = Arah depan robot

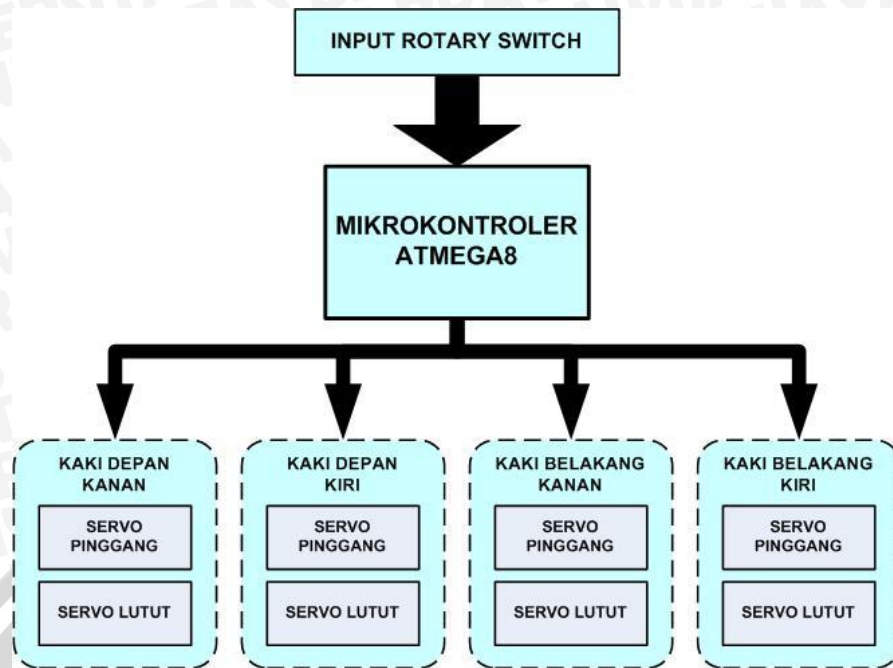
Dari tahap-tahap gerak berputar seperti dijelaskan dalam Tabel 4.4 di atas dapat diketahui bahwa robot berputar sebesar 30° dalam satu siklus gerakan. Untuk menghasilkan sudut 90° ke arah kiri, maka robot harus mengulangi siklus tersebut sebanyak 3 kali.

4.2.3.5 Gerak Berputar 180°

Robot Berkaki Empat dapat berputar sebesar 180° dengan menggunakan metode yang sama seperti metode hadap kanan atau hadap kiri. Dalam metode hadap kanan dan hadap kiri dijelaskan bahwa satu siklus gerak berputar menghasilkan perubahan sudut putar robot sebesar 30° . Maka untuk dapat berputar 180° robot harus mengulangi tahap-tahap tersebut sebanyak 6 kali.

4.3 Perancangan Sistem

Sistem Robot Berkaki Empat secara umum dibagi menjadi tiga bagian, yaitu blok input, blok mikrokontroler, dan blok motor DC servo seperti ditunjukkan dalam Gambar 4.12.



Gambar 4. 12. Diagram Blok Sistem Robot Berkaki Empat

Fungsi masing-masing blok dalam diagram blok diatas adalah sebagai berikut :

1). Blok Input

Blok ini berisi *rotary switch* yang digunakan sebagai pemilih mode program. Ada 8 mode yang digunakan, yaitu mode gerakan maju pelan, mode gerakan maju sedang, mode gerakan maju cepat, mode gerakan mundur, mode hadap kanan untuk memerintahkan robot berputar 90^0 ke kanan, mode hadap kiri untuk memerintahkan robot berputar 90^0 ke kiri, mode putar 180^0 untuk memerintahkan robot berputar 180^0 , dan mode gabungan untuk memerintahkan robot melakukan kombinasi beberapa gerakan secara berurutan.

2). Blok Mikrokontroler Kontrol Servo

Mikrokontroler yang digunakan adalah mikrokontroler ATmega8. Fungsi mikrokontroler ini adalah untuk mengontrol pergerakan motor DC servo. Karena jumlah motor servo yang dikendalikan cukup banyak, maka digunakan metode kontrol multi servo.

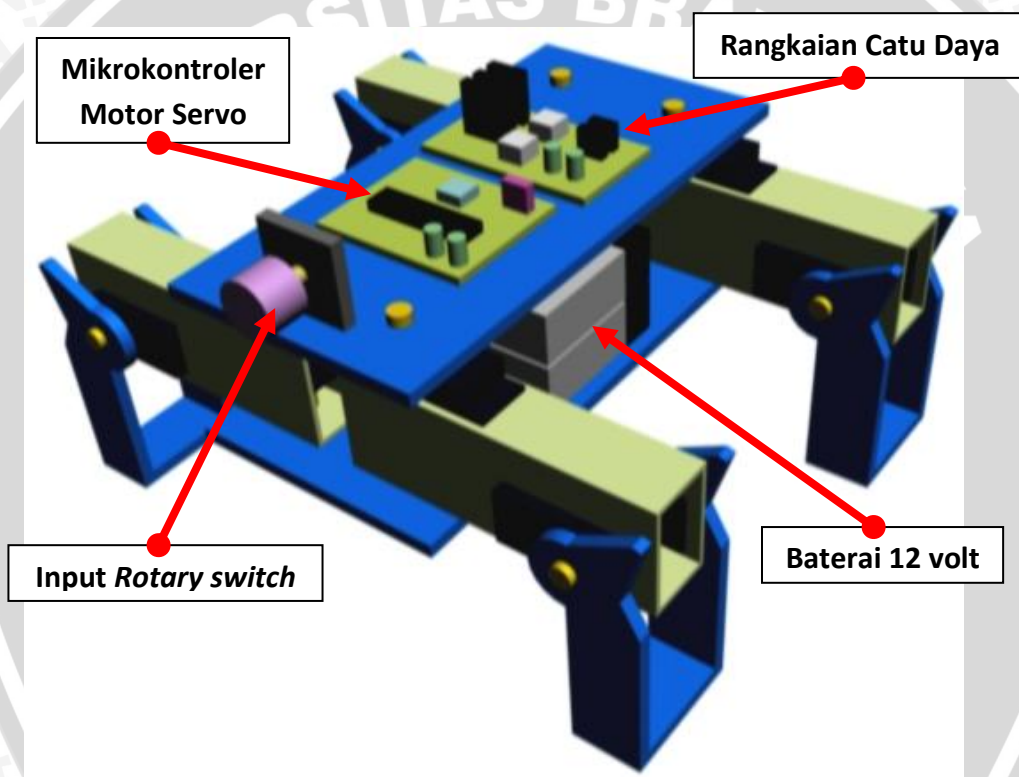
3). Blok Kaki

Robot ini memiliki empat buah kaki menyerupai bentuk kaki hewan. Pada masing-masing kaki terdapat dua buah motor DC servo yang berfungsi

sebagai sendi pergerakan kaki. Jadi secara keseluruhan terdapat delapan buah motor DC servo yang harus dikontrol. Untuk penjelasan letak dan fungsi masing-masing motor DC servo akan dijelaskan pada sub bab berikutnya.

4.4 Perancangan Perangkat Keras

Perangkat keras yang dibutuhkan untuk menggerakkan Robot Berkaki Empat terdiri dari beberapa rangkaian, yaitu rangkaian catu daya sistem, rangkaian mikrokontroler kontrol servo, rangkaian *rotary switch*, dan motor DC servo. Posisi perangkat keras tersebut pada robot ditunjukkan dalam Gambar 4.13.



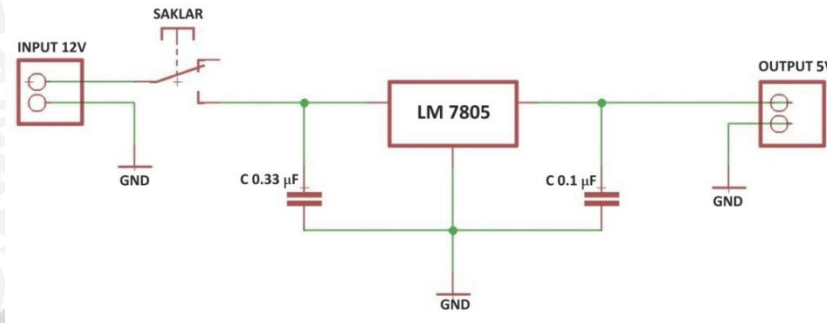
Gambar 4. 13. Peletakkan Perangkat Keras pada Robot Berkaki Empat

4.4.1 Catu Daya Sistem

Robot Berkaki Empat membutuhkan dua jenis catu daya yang berbeda, yaitu catu daya 5 V untuk rangkaian mikrokontroler ATmega8, dan catu daya 6 V untuk motor DC servo. Sumber catu daya yang dipakai adalah dua buah baterai SLA (*Sealed Lead Acid*) 12 V yang dirangkai paralel.

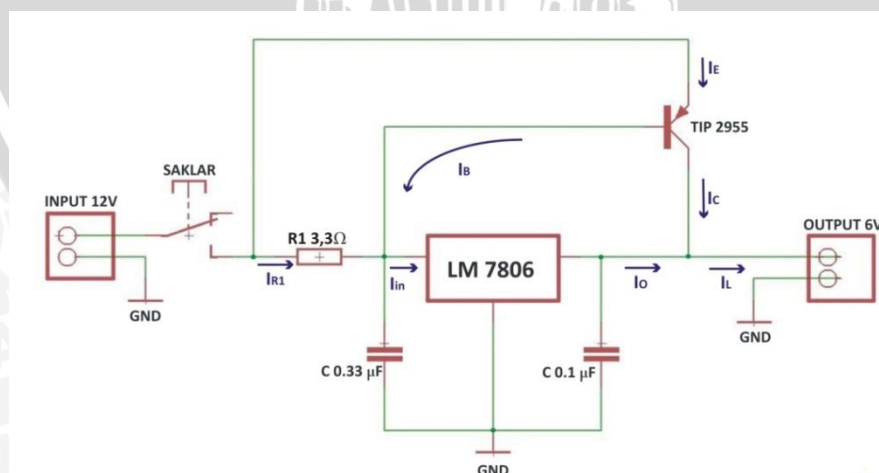
Mikrokontroler ATmega8 dapat bekerja jika diberi catu tegangan antara 4,8 V hingga 5,5 V sesuai dengan datasheet ATmega8. Pada perancangan digunakan catu

daya sebesar 5V yang diperoleh dari rangkaian *Fixed Output Regulator* pada datasheet LM78XX. Pada rangkaian digunakan regulator LM7805 agar diperoleh tegangan keluaran sebesar 5 V. Skema rangkaian catu daya mikrokontroler ATmega8 ditunjukkan dalam Gambar 4.14.



Gambar 4. 14. Rangkaian Catu Daya Mikrokontroler ATmega8

Sumber tegangan 6 V untuk motor DC servo diperoleh dari rangkaian *High Current Voltage Regulator* sesuai pada datasheet LM78XX yang digunakan untuk mencukupi kebutuhan arus yang besar dari motor DC servo. Regulator yang digunakan adalah tipe LM7806. Rangkaian ini dapat melewati arus yang lebih besar dari batas arus maksimal yang bisa dilewatkan oleh regulator LM7806 yang hanya 1 A. Berdasarkan hasil pengukuran, satu motor DC servo membutuhkan arus sebesar 300 mA untuk dapat bergerak, sementara motor DC servo yang digunakan sebanyak 8 buah. Jadi total arus yang dibutuhkan sekitar 2,4 A. Skema rangkaian catu daya motor DC servo ditunjukkan dalam Gambar 4.15.



Gambar 4. 15. Rangkaian Catu Daya Motor DC Servo

Pada perancangan rangkaian ini digunakan regulator dengan tipe LM7806 dengan arus maksimum yang bisa dilewatkan (I_O) adalah 1 A, sedangkan transistor yang digunakan adalah transistor jenis PNP dengan tipe TIP 2955, dengan spesifikasi sebagai berikut :

$$\beta (h_{FE}) = 20$$

$$V_{EB} = 1,8 \text{ V}$$

Untuk arus beban yang relatif kecil, jatuh tegangan di R_1 masih lebih kecil dari batas V_{EB} (1,8 V) dan masih belum mengaktifkan transistor. Jadi semua arus beban masih disuplai dari regulator itu sendiri. Sedangkan pada saat arus beban meningkat, jatuh tegangan di R_1 juga meningkat, pada saat V_{R1} melebihi 1,8 V, maka transistor akan mulai aktif. Pada perancangan digunakan R_1 sebesar $3,3 \Omega$, jadi untuk arus yang lebih besar dari 550 mA, maka V_{R1} menjadi $3,3 \Omega \times 550 \text{ mA} = 1,815 \text{ V}$, dan transistor akan terkonduksi. Kebutuhan arus yang melebihi batas arus pada regulator akan disuplai oleh transistor melalui penguatan I_B .

Persamaannya adalah sebagai berikut:

$$I_C = \beta \cdot I_B \quad (4.1)$$

$$I_L = I_C + I_O = \beta \cdot I_B + I_O \quad (4.2)$$

$$I_B = I_{in} - I_{R1} \quad (4.2)$$

$$I_{in} = I_O, \text{ maka :}$$

$$I_B = I_O - I_{R1}$$

sementara,

$$I_{R1} = \frac{V_{EB}}{R_1}, \quad (4.3)$$

maka:

$$I_B = I_O - \frac{V_{EB}}{R_1}$$

$$I_L = \beta \left[I_O - \frac{V_{EB}}{R_1} \right] + I_O$$

$$I_L = (1 + \beta) I_O - \beta \frac{V_{EB}}{R_1} \quad (4.4)$$

Dari persamaan 4.4 dapat diketahui bahwa besarnya arus yang bisa dilewatkan pada rangkaian ini tergantung pada nilai β dan V_{EB} dari transistor, dan juga nilai I_O (maksimum) dari regulator. Maka besar arus maksimum yang bisa dilewatkan oleh rangkaian ini, yaitu :

$$I_L = (1 + \beta) I_O - \beta \frac{V_{EB}}{R_1}$$



$$I_L = [(1 + 20)1 - 20 \frac{1,8}{3,3}] A$$

$$I_L = (21 - 4,242) A$$

$$I_L = 16,758 A$$

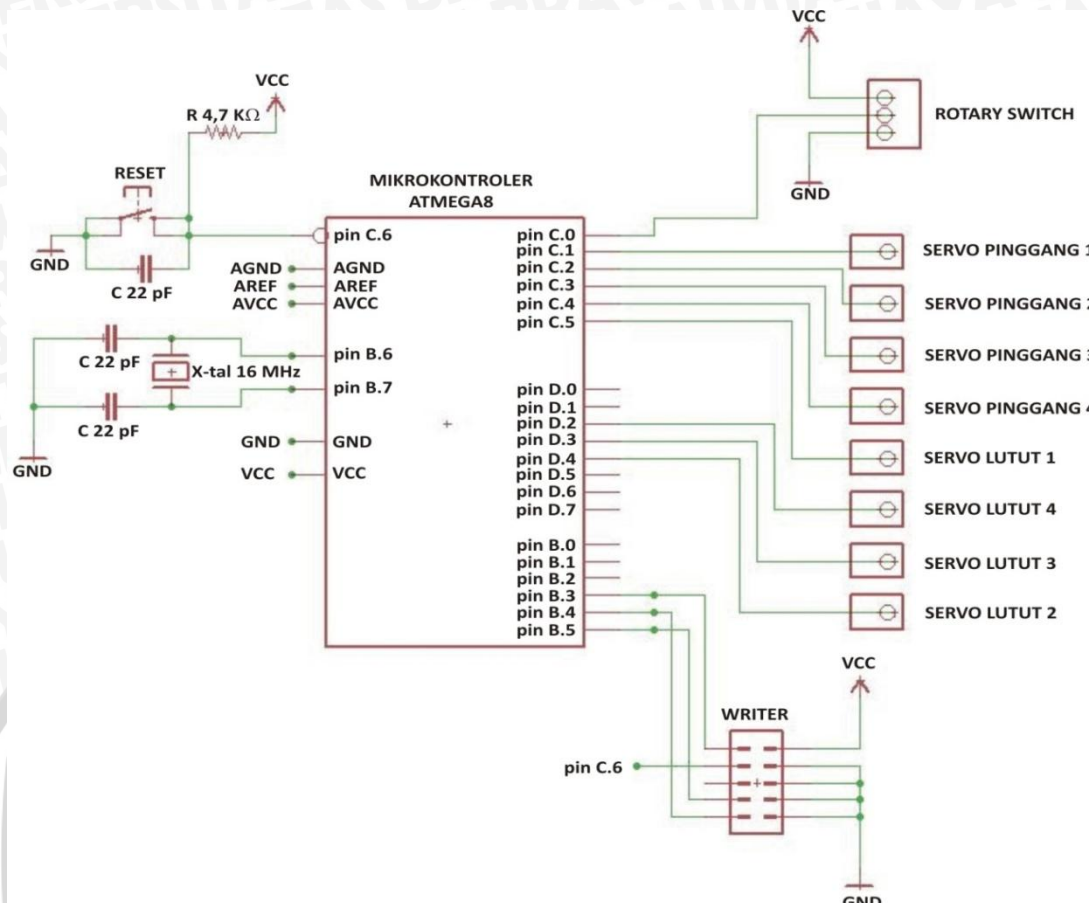
Nilai arus I_L ini (16,758 A) sudah cukup untuk memenuhi kebutuhan arus 8 buah motor DC servo yang mencapai 2,4 A.

4.4.2 Rangkaian Mikrokontroler Kontrol Servo

Motor DC servo pada Robot Berkaki Empat dikontrol menggunakan mikrokontroler ATmega8. Pin-pin pada mikrokontroler ini dihubungkan langsung dengan jalur sinyal pada motor DC servo untuk memberikan sinyal periodik sebesar 20ms. Pin-pin yang digunakan dalam perancangan rangkaian mikrokontroler kontrol servo adalah sebagai berikut :

- 1). Pin C.6 : Fungsi RESET pada Mikrokontroler
- 2). Pin C.0 : Sebagai Input Mode *Rotary switch*
- 3). Pin C.1 : Sebagai sinyal kontrol Servo Pinggang 1
- 4). Pin C.2 : Sebagai sinyal kontrol Servo Pinggang 2
- 5). Pin C.3 : Sebagai sinyal kontrol Servo Pinggang 3
- 6). Pin C.4 : Sebagai sinyal kontrol Servo Pinggang 4
- 7). Pin C.5 : Sebagai sinyal kontrol Servo Lutut 1
- 8). Pin D.4 : Sebagai sinyal kontrol Servo Lutut 2
- 9). Pin D.3 : Sebagai sinyal kontrol Servo Lutut 3
- 10). Pin D.2 : Sebagai sinyal kontrol Servo Lutut 4

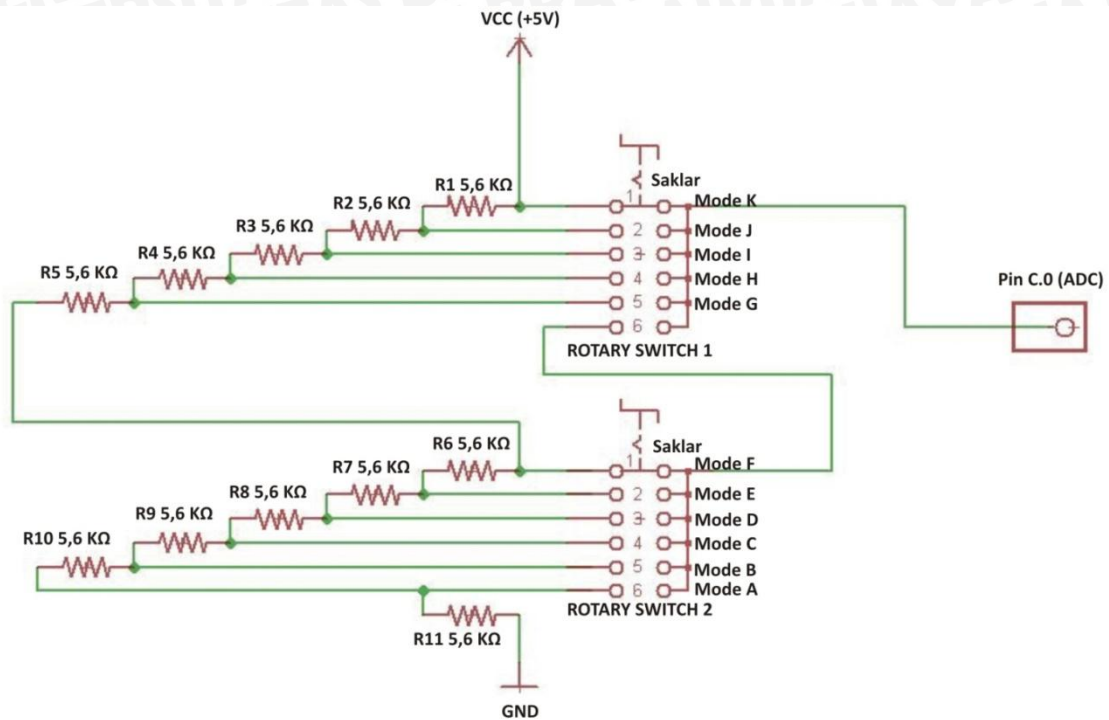
Mikrokontroler menggunakan kristal osilator eksternal sebesar 16 MHz yang dihubungkan dengan kapasitor sebesar 22 pF sesuai dengan datasheet ATmega8. Gambar skema rangkaian mikrokontroler kontrol servo ditunjukkan dalam Gambar 4.16.



Gambar 4. 16. Skema Rangkaian Mikrokontroler Kontrol Servo

4.4.3 Rangkaian *Rotary Switch*

Rangkaian *rotary switch* digunakan sebagai masukan untuk mengubah mode program pada Robot Berkaki Empat. Terdapat 8 mode program dalam perancangan ini, yaitu mode maju (kecepatan pelan), mode maju (kecepatan sedang), mode maju (kecepatan cepat), mode mundur, mode hadap kanan, mode hadap kiri, mode berputar 180°, dan mode gabungan semua gerakan. Dalam satu *rotary switch* terdapat 6 mode/saklar yang bisa digunakan. Karena terdapat 8 mode program yang akan digunakan dalam pengujian, maka dibutuhkan 2 *rotary switch*. Skema rangkaian *rotary switch* ditunjukkan dalam Gambar 4.17.



Gambar 4. 17. Skema Rangkaian *Rotary Switch*

Prinsip kerja rangkaian *rotary switch* adalah pemakaian fitur ADC (*Analog to Digital Converter*) dari mikrokontroler ATmega8. Konverter dari sinyal analog menjadi sinyal digital ini akan mengubah level tegangan analog menjadi sinyal digital 8 bit. Dengan batas konversi adalah sinyal digital 8 bit berarti rentang nilai digital yang dihasilkan adalah antara 0 – 255. Input level tegangan berasal dari rangkaian *rotary switch* yang terdiri dari 11 level tegangan berbeda. Level tegangan yang berbeda tersebut didapat dari prinsip pembagian tegangan menggunakan resistor. Perhitungan nilai masing-masing level tegangan dari rangkaian *rotary switch* adalah sebagai berikut:

$$V_{CC} = 5 \text{ volt}$$

$$R_1 = R_2 = R_3 = R_4 = R_5 = R_6 = R_7 = R_8 = R_9 = R_{10} = R_{11} = 5,6 \text{ k}\Omega$$

$$R_{TOTAL} = 61,6 \text{ k}\Omega$$

$$\text{Tegangan pada Mode A} = \frac{R_{11}}{R_{TOTAL}} \times V_{CC} = \frac{5,6 \Omega}{61,6 \Omega} \times 5 \text{ V} = 0,45 \text{ V}$$

$$\text{Tegangan pada Mode B} = \frac{R_{10} + R_{11}}{R_{TOTAL}} \times V_{CC} = \frac{11,2 \Omega}{61,6 \Omega} \times 5 \text{ V} = 0,90 \text{ V}$$

$$\text{Tegangan pada Mode C} = \frac{R_9 + R_{10} + R_{11}}{R_{TOTAL}} \times V_{CC} = \frac{16,8 \Omega}{61,6 \Omega} \times 5 \text{ V} = 1,36 \text{ V}$$

$$\text{Tegangan pada Mode D} = \frac{R_8 + R_9 + R_{10} + R_{11}}{R_{TOTAL}} \times V_{CC} = \frac{22,4 \Omega}{61,6 \Omega} \times 5 \text{ V} = 1,81 \text{ V}$$

$$\text{Tegangan pada Mode E} = \frac{R_7+R_8+R_9+R_{10}+R_{11}}{R_{TOTAL}} \times V_{CC} = \frac{28 \Omega}{61,6 \Omega} \times 5 \text{ V} = 2,27 \text{ V}$$

$$\text{Tegangan pada Mode F} = \frac{R_6+R_7+R_8+R_9+R_{10}+R_{11}}{R_{TOTAL}} \times V_{CC} = \frac{33,6 \Omega}{61,6 \Omega} \times 5 \text{ V} = 2,72 \text{ V}$$

$$\text{Tegangan pada Mode G} = \frac{R_5+R_6+R_7+R_8+R_9+R_{10}+R_{11}}{R_{TOTAL}} \times V_{CC} = \frac{39,2 \Omega}{61,6 \Omega} \times 5 \text{ V} = 3,18 \text{ V}$$

$$\text{Tegangan pada Mode H} = \frac{R_4+R_5+R_6+R_7+R_8+R_9+R_{10}+R_{11}}{R_{TOTAL}} \times V_{CC} = \frac{44,8 \Omega}{61,6 \Omega} \times 5 \text{ V} = 3,63 \text{ V}$$

$$\text{Tegangan pada Mode I} = \frac{R_3+R_4+R_5+R_6+R_7+R_8+R_9+R_{10}+R_{11}}{R_{TOTAL}} \times V_{CC} = \frac{50,4 \Omega}{61,6 \Omega} \times 5 \text{ V} = 4,09 \text{ V}$$

$$\text{Tegangan pada Mode J} = \frac{R_2+R_3+R_4+R_5+R_6+R_7+R_8+R_9+R_{10}+R_{11}}{R_{TOTAL}} \times V_{CC} = \frac{56 \Omega}{61,6 \Omega} \times 5 \text{ V} = 4,54 \text{ V}$$

$$\text{Tegangan pada Mode K} = \frac{R_{TOTAL}}{R_{TOTAL}} \times V_{CC} = \frac{61,6 \Omega}{61,6 \Omega} \times 5 \text{ V} = 5 \text{ V}$$

Dari perhitungan di atas diperoleh 11 level tegangan yang berbeda. Tegangan analog ini akan dikonversi menjadi sinyal digital 8 bit (0-255) menggunakan perhitungan konversi ADC seperti yang tercantum pada *datasheet* ATmega8 sebagai berikut :

$$ADC = \frac{V_{IN} \times 255}{V_{REF}} \quad (4.5)$$

dengan ADC = hasil konversi ADC

V_{IN} = tegangan masukan

V_{REF} = tegangan referensi

Pada perancangan ini tegangan referensi yang digunakan adalah tegangan dari Pin AREF pada mikrokontroler yang bernilai +5 V. Karena tipe data ADC adalah integer maka hasil akhir pengkonversian akan dibulatkan. Melalui program, hasil akhir konversi tersebut akan dibagi dengan suatu bilangan tertentu sehingga dihasilkan angka-angka yang urut dari 1 sampai 11 yang dapat memudahkan pemrogram. Bilangan pembagi yang dipilih pada perancangan ini adalah 22. Hasil konversi ADC dari 11 level tegangan tersebut ditunjukkan dalam Tabel 4.5.

Tabel 4. 5. Hasil Konversi Tegangan Analog ke Sinyal Digital

Mode	Tegangan Analog (dalam volt)	Hasil Konversi ADC	Hasil Konversi ADC (Pembulatan)	ADC / 22 (Pembulatan)
A	0,45	22,95	22	1
B	0,90	45,9	45	2
C	1,36	69,36	69	3
D	1,81	92,31	92	4
E	2,27	115,77	115	5
F	2,72	138,72	138	6
G	3,18	162,18	162	7
H	3,63	185,13	185	8
I	4,09	208,59	208	9
J	4,54	231,54	231	10
K	5,00	255	255	11

Nilai-nilai dari 1 sampai 11 dari hasil konversi di atas adalah nilai yang akan dideteksi oleh mikrokontroler sebagai masukan mode program melalui *rotary switch*, sehingga mikrokontroler akan menjalankan program sesuai dengan mode yang dipilih pada rangkaian *rotary switch*. Mode-mode program pada Robot Berkaki Empat ditunjukkan dalam Tabel 4.6.

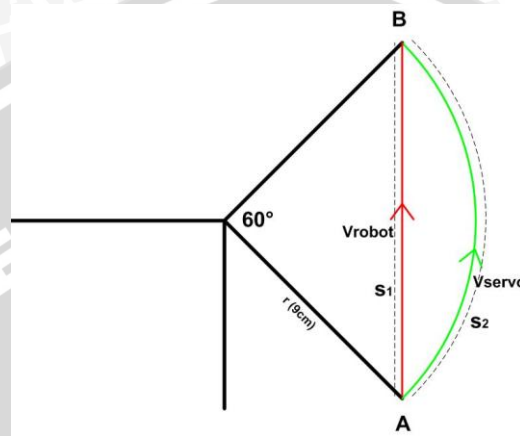
Tabel 4. 6. Mode Program pada *Rotary Switch*

Hasil Konversi ADC	Mode	Program
1	A	Maju pelan
2	B	Maju sedang
3	C	Maju cepat
4	D	Mundur
5	E	Hadap kanan
6	F	Hadap kiri
7	G	Putar 180°
8	H	Gerak Gabungan
9	I	-
10	J	-
11	K	-

4.4.4 Motor DC Servo

Motor DC servo dipakai untuk menggerakkan kaki pada Robot Berkaki Empat. Pada setiap kaki terdapat dua buah motor DC servo sebagai pemodelan dari sendi kaki. Jadi total ada 8 buah motor DC servo yang digunakan. Pada ruang lingkup telah disebutkan bahwa batas kecepatan maksimum robot pada saat berjalan adalah 20 cm/s. Dari batasan ini dapat ditentukan besar torsi motor DC servo yang dibutuhkan agar robot dapat berjalan dengan kecepatan yang diinginkan. Untuk menentukan torsi motor

DC servo, kecepatan robot yang merupakan kecepatan linier harus diubah menjadi kecepatan linier motor DC servo. Diketahui panjang lengan robot dari pusat motor servo adalah 9 cm. Diasumsikan massa total robot adalah 2 kg. Selain itu juga diasumsikan bahwa untuk mencapai kecepatan konstan, motor DC servo membutuhkan waktu 0,02 s. Hubungan kecepatan robot dengan kecepatan motor DC servo ditunjukkan dalam Gambar 4.18.



Gambar 4. 18. Hubungan Kecepatan Robot dengan Kecepatan Servo

Dari Gambar 4.18, besarnya nilai s_1 adalah:

$$s_1^2 = r^2 + r^2 - 2 \cdot r \cdot r \cdot \cos 60^\circ \quad (4.6)$$

$$s_1^2 = [9^2 + 9^2 - 2 \cdot 9 \cdot 9 \cdot \cos 60^\circ] \text{ cm}$$

$$s_1^2 = 81 \text{ cm}$$

$$s_1 = 9 \text{ cm}$$

Waktu yang dibutuhkan kaki robot (t) untuk bergerak dari titik A ke titik B adalah:

$$t = \frac{s_1}{V_{robot}} \quad (4.7)$$

$$t = \frac{9}{20} \text{ s}$$

$$t = 0,45 \text{ s}$$

Sedangkan besarnya nilai s_2 (jarak tempuh kaki robot) adalah:

$$s_2 = 2\pi r \cdot \frac{60^\circ}{360^\circ}$$

$$s_2 = [2 \cdot 3,14 \cdot 9 \cdot \frac{60^\circ}{360^\circ}] \text{ cm}$$

$$s_2 = 9,42 \text{ cm}$$

Maka V_{servo} (kecepatan linier motor DC servo) adalah:

$$V_{\text{servo}} = \frac{s_2}{t} \quad (4.8)$$

$$V_{\text{servo}} = \left[\frac{9,42}{0,45} \right] \text{ cm/s}$$

$$V_{\text{servo}} = 20,93 \text{ cm/s}$$

$$V_{\text{servo}} = 0,2093 \text{ m/s}$$

Maka kecepatan sudutnya adalah :

$$\omega = \frac{V_{\text{servo}}}{r} \quad (4.9)$$

$$\omega = \frac{0,2093}{0,09} \text{ rad/s}$$

$$\omega = 2,32 \text{ rad/s}$$

Sementara, waktu yang dibutuhkan servo untuk melakukan percepatan dari kecepatan 0 m/s menuju kecepatan konstan diasumsikan 0,02 s, maka percepatan sudutnya adalah :

$$\alpha = \frac{\omega}{t} \quad (4.10)$$

$$\alpha = \frac{2,32}{0,02} \text{ rad/s}^2$$

$$\alpha = 116 \text{ rad/s}^2$$

Torsi (τ) adalah perkalian antara momen inersia (I) dengan percepatan sudutnya (α), dimana:

$$I = m \cdot r^2 \quad (4.11)$$

Massa total robot diasumsikan 2kg, sementara pada perancangan gerakan dijelaskan bahwa pada saat berjalan selalu terdapat 2 kaki robot yang bertumpu pada permukaan secara bergantian. Maka massa robot yang membebani masing-masing kaki adalah 1 kg.

Maka torsinya adalah:

$$\tau = I \cdot \alpha \quad (4.12)$$

$$\tau = m r^2 \cdot \alpha$$

$$\tau = [1 (0,09)^2 \cdot 116] \text{ Nm}$$

$$\tau = [0,0081 \cdot 116] Nm$$

$$\tau = 0,93 Nm$$

$$\tau = 0,093 kg \cdot m$$

$$\tau = 9,3 kg \cdot cm$$

Dari hasil perhitungan dapat disimpulkan bahwa torsi motor DC servo yang dibutuhkan adalah 9,3 kg.cm. Maka motor DC servo yang digunakan adalah Tower Pro dengan tipe MG945R, yang memiliki torsi sebesar 10 kg.cm. Bentuk fisik motor DC servo ini ditunjukkan dalam Gambar 4.19.



Gambar 4. 19. Motor DC Servo Tower Pro MG945R

Spesifikasi motor DC servo Tower Pro MG945R adalah sebagai berikut :

- 1). Torsi: 10 kg-cm pada 4,8 V DC dan 12 kg-cm pada 6,0 V DC
- 2). Kecepatan: 0,23 detik/60° pada 4,8 V DC dan 0,2 detik/60° pada 6 V DC
- 3). Dimensi: 40,7 x 19,7 x 42,9 mm
- 4). Berat: 55 gr

Motor DC servo Tower Pro MG945R memiliki 3 jalur pengkabelan, yaitu V_{CC} , ground, dan sinyal. Jalur V_{CC} dihubungkan ke sumber tegangan 6 V, jalur ground dihubungkan ke ground mikrokontroler yang menjadi satu dengan ground baterai, dan jalur sinyal dihubungkan ke pin mikrokontroler ATmega8 sebagai pengontrol motor DC servo. Setiap motor DC servo terhubung ke mikrokontroler ATmega8 melalui pin tersendiri. Pin-pin mikrokontroler ATmega8 yang dihubungkan ke jalur sinyal motor DC servo adalah Pin C.1, Pin C.2, Pin C.3, Pin C.4, Pin C.5, Pin D.2, Pin D.3, dan Pin D.4.

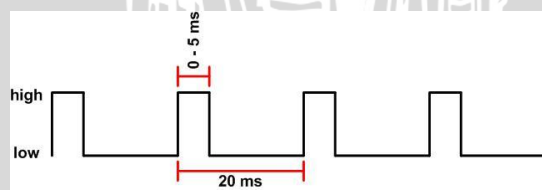
Motor DC servo ini memiliki periode sebesar 20 ms dan hanya dapat berputar dari sudut 0° sampai sudut 180° . Dengan mengubah kondisi logika *high* dari *duty cycle* akan dapat mengatur besarnya sudut putar motor servo. Dari hasil percobaan, untuk menghasilkan sudut 0° dibutuhkan logika *high* selama 600 μs , sedangkan untuk sudut 180° dibutuhkan logika *high* selama 2400 μs .

4.5 Perancangan Perangkat Lunak

Perancangan perangkat lunak bertujuan untuk menentukan metode yang dipakai untuk mengontrol motor DC servo. Selain itu juga untuk menentukan algoritma yang akan digunakan pada mikrokontroler ATmega8 sebagai pengontrol motor DC servo.

4.5.1 Kontrol Multi-Servo

Motor DC servo membutuhkan pulsa periodik agar dapat berputar sesuai dengan posisi yang diinginkan. Jika pulsa periodik tersebut hilang maka motor DC servo akan kehilangan posisinya dan motor seakan-akan mati. Matinya motor akan menyebabkan kinerja robot terganggu karena tiap sendi pada robot tidak dapat mempertahankan posisinya. Untuk menghindari hal tersebut, mikrokontroler sebagai kontroler motor DC servo harus dapat menyediakan pulsa periodik secara terus-menerus kepada motor DC servo. Pulsa periodik juga harus bersifat stabil, karena jika pulsa periodik tersebut lebarnya berubah-ubah akan menyebabkan posisi servo berubah-ubah. Pulsa periodik motor DC servo ditunjukkan dalam Gambar 4.20.

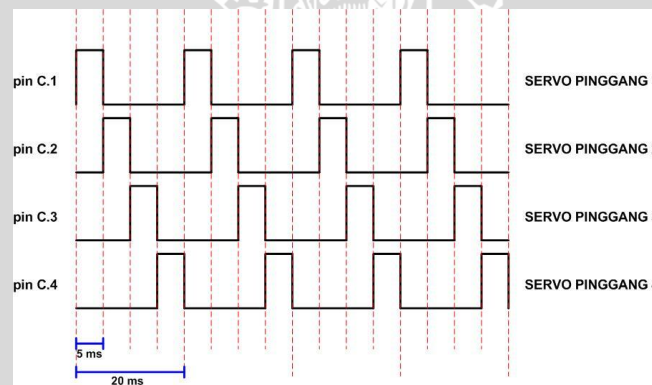


Gambar 4. 20. Sinyal Kontrol Motor DC Servo

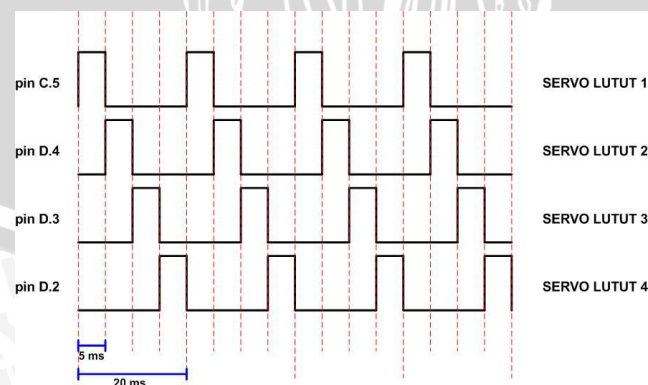
Sebuah mikrokontroler memiliki fungsi timer yang handal dan dapat digunakan untuk menciptakan pulsa periodik untuk kontrol servo. Akan tetapi pada ATmega8 hanya terdapat dua buah timer yaitu pin OCRA dan pin OCRB. Jika tiap-tiap timer digunakan untuk mengontrol sebuah motor servo, maka sebuah ATmega8 hanya bisa digunakan untuk mengontrol dua buah motor servo. Cara tersebut tidak bisa

digunakan untuk mengontrol Robot Berkaki Empat, karena diperlukan banyak mikrokontroler dan tidak efisien.

Pengontrolan multi-servo menggunakan satu ATmega8 dapat dilakukan dengan cara menggunakan pin-pin I/O biasa (selain pin timer) dan pemakaian timer secara bersama-sama. Karena pada mikrokontroler ATmega8 terdapat 2 pin timer yaitu OCRA dan OCRB, maka masing-masing pin timer ini digunakan untuk mengeluarkan pulsa periodik melalui 4 pin I/O biasa secara bergantian tetapi tetap dalam satu periode. Jika periode motor servo adalah 20 ms, maka interupsi yang dilakukan oleh masing-masing timer adalah setiap 5 ms. Interupsi dilakukan setiap 5ms karena setiap servo yang dikontrol pada dasarnya hanya membutuhkan logika tinggi pada rentang waktu 0 hingga 2500 μ s pada awal pulsa periodiknya, sisanya adalah logika rendah. Pulsa periodik yang dikeluarkan oleh pin I/O non-timer untuk melakukan pengontrolan multi-servo ditunjukkan dalam Gambar 4.21 dan Gambar 4.22.



Gambar 4.21. Pulsa Pengontrolan Multi Servo (OCRA)



Gambar 4. 22. Pulsa Pengontrolan Multi Servo (OCRB)

Garis putus-putus warna merah pada Gambar 4.21 dan Gambar 4.22 menunjukkan interupsi yang dilakukan oleh timer setiap 5 ms. Gambar 4.21 dan

Gambar 4.22 juga menunjukkan bagaimana tiap-tiap pin mikrokontroler secara bersamaan menghasilkan pulsa periodik. Yang membedakan hanya waktu awalan periode pulsa kontrolnya. Hal tersebut terlihat dari perbedaan posisi logika tinggi masing-masing pin mikrokontroler. Untuk menghasilkan kontrol posisi yang berbeda pada tiap-tiap pin mikrokontroler digunakan interupsi pada fitur *Timer/Counter Output Compare Match* pada timer 16 bit. Dengan metode seperti pada Gambar 4.21 dan Gambar 4.22, dapat dilakukan pengontrolan sebanyak 8 motor DC servo dengan syarat menggunakan timer 16 bit dan menggunakan interupsi pada OCRA dan OCRB (masing-masing empat kali).

Lebar sinyal *high* pada awal pulsa periodik diatur oleh *Output Compare Register* (OCRA dan OCRB) pada timer 16 bit. Perhitungan nilai pada register OCRA dan OCRB diperoleh dari datasheet ATmega8 sebagai berikut.

$$f_{OCPWM} = \frac{f_{clk_I/O}}{N \cdot (1 + TOP)} \quad (4.13)$$

$$TOP = \frac{f_{clk_I/O}}{N \cdot f_{OCPWM}} - 1 \quad (4.14)$$

dengan $f_{OCn \times PWM}$ = Frekuensi PWM pada mode *Output Compare*

$f_{clk_I/O}$ = Frekuensi clock

N = *Prescaler*

TOP = Nilai pada *Output Compare Register*

Pada perancangan digunakan clock dengan frekuensi 16 MHz dan *prescaler* 8x. Pada perancangan ini hanya beberapa sudut tertentu saja yang digunakan, menyesuaikan dengan medan kerja setiap sendi robot dan metode gerakannya. Lebar sinyal *high* tiap sudut servo dan nilai TOP berdasarkan rumus 4.7 ditunjukkan dalam Tabel 4.7.

Tabel 4. 7. Lebar Sinyal *High* Sudut Servo dan Nilai TOP

No.	Sudut Motor DC Servo (dalam Derajat)	Lebar Sinyal <i>High</i> (dalam μ s)	Nilai TOP
1.	0	600	1199
2.	60	1200	2399
3.	70	1300	2599
4.	75	1350	2699
5.	80	1400	2799
6.	90	1500	2999
7.	100	1600	3199
8.	105	1650	3299
9.	110	1700	3399
10.	120	1800	3599
11.	180	2400	4799

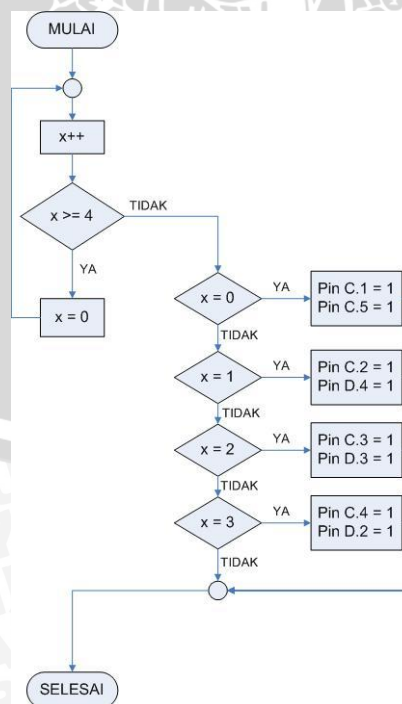
Nilai yang diperoleh dari perhitungan TOP diatas nantinya langsung dimasukkan ke dalam register OCRA dan OCRB.

4.5.2 Algoritma Mikrokontroler Kontrol Servo

Robot Berkaki Empat menggunakan mikrokontroler ATmega8 sebagai kontroler motor DC servo. mikrokontroler ini berfungsi untuk memberikan pulsa periodik kepada motor servo agar motor servo dapat membentuk sudut sesuai dengan perancangan yang diinginkan. Algoritma pada mikrokontroler kontrol servo secara umum dibagi menjadi tiga bagian, yaitu :

1). Interupsi *Timer1*

Timer yang digunakan adalah *Timer1-16 bit* dengan mode *Fast PWM* dan mode interupsi yang digunakan adalah *Timer1 Overflow Interrupt*. Tujuannya adalah untuk menghasilkan logika *HIGH* pada awal sinyal kontrol servo secara berurutan sesuai dengan servo yang sedang dikontrolnya. Dengan mengetahui bahwa periode sinyal kontrol servo adalah 20 ms, maka interupsi dilakukan setiap 5000 μ s sebanyak delapan kali. Diagram alir program Interupsi *Timer1* ditunjukkan dalam Gambar 4.23.

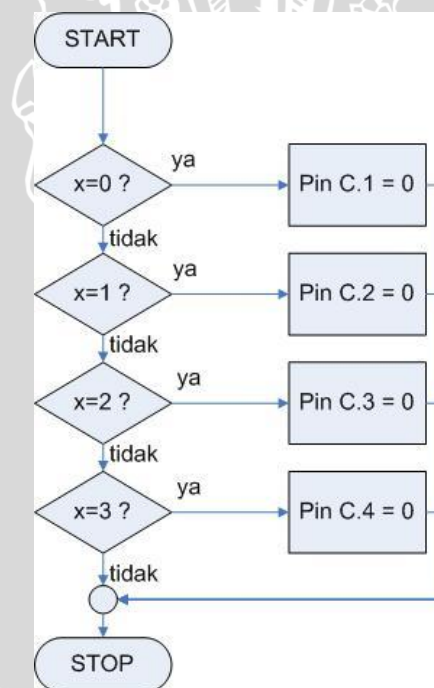


Gambar 4. 23. Diagram Alir Program Interupsi *Timer1*

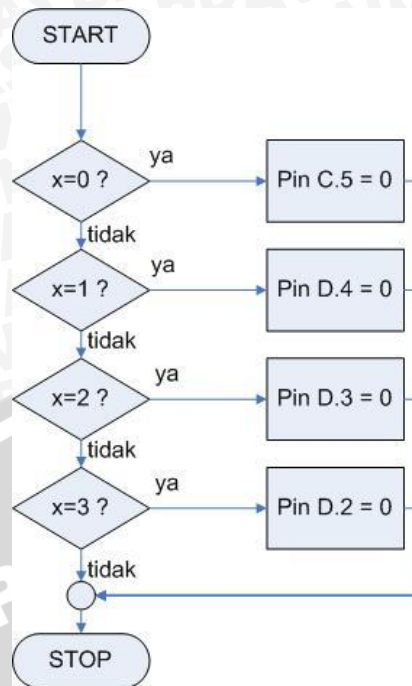
Nilai x pada awal program adalah 0. Pada saat subrutin interupsi aktif, nilai variabel x akan dicek apakah lebih dari 3, jika memenuhi maka nilai x akan dinolkan, tetapi jika tidak memenuhi maka akan masuk ke seleksi kondisi yang akan memberikan nilai 1 (logika *high*) pada pin-pin tertentu sesuai nilai x .

2). Interupsi *Timer Output Compare Match*

Interupsi *Timer Output Compare Match* bertujuan menentukan lebar sinyal kontrol *HIGH* pada awal sinyal kontrol servo. Isi program interupsi adalah memberi logika *low* pada pin yang mengontrol servo secara berurutan. Pada perancangan ini digunakan dua buah OCR pada timer 16 bit, yaitu OCRA dan OCRB. Masing-masing OCR mengontrol empat buah motor servo. OCRA untuk mengontrol empat servo pinggang, dan OCRB mengontrol empat servo lutut. Nilai OCRA dan OCRB mempengaruhi arah pergerakan motor servo. diagram alir program interupsi *Timer Output Compare Match* ditunjukkan dalam Gambar 4.24 dan Gambar 4.25.



Gambar 4. 24. Diagram Alir Program Interupsi Timer OCRA

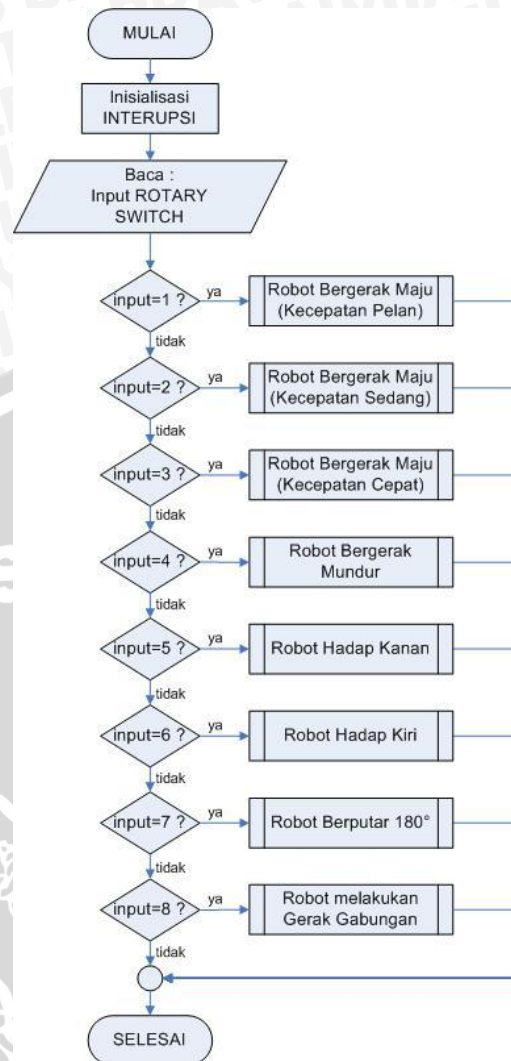


Gambar 4. 25. Diagram Alir Program Interupsi Timer OCRB

Program akan menyeleksi nilai x yang untuk selanjutnya akan memberikan nilai 0 (logika *low*) pada pin tertentu sesuai nilai x pada awal program. Logika *low* ini menunjukkan berakhirnya logika *high* yang diberikan pada motor DC servo.

3). Program Utama

Program utama mikrokontroler kontrol servo memiliki tujuan utama yaitu memberikan sinyal kontrol servo sesuai dengan perintah yang dikirim dari mikrokontroler utama, dan juga menerima masukan dari *rotary switch*. Terdapat 8 macam mode yang diproses pada program utama ini, yaitu gerak maju (kecepatan pelan), gerak maju (kecepatan sedang), gerak maju (kecepatan cepat), gerak mundur, gerak hadap kanan, gerak hadap kiri, gerak putar 180° , dan gerak gabungan. Masing-masing gerakan memiliki posisi sudut tertentu yang akan diberikan kepada servo sehingga secara keseluruhan dapat bergerak sesuai dengan tujuan. Diagram alir program utama pada mikrokontroler kontrol servo ditunjukkan dalam Gambar 4.26.



Gambar 4. 26. Diagram Alir Program Utama Mikrokontroler Kontrol Servo

Pada awal program dilakukan inisialisasi interupsi timer yang akan digunakan pada program selanjutnya. Kemudian program utama akan menyeleksi nilai variabel x yang didapat dari input *rotary switch*. Hasil seleksi nilai variabel x tersebut menentukan subfungsi program yang akan dijalankan. Terdapat 8 subfungsi program yang terdiri dari program gerak maju pelan, program gerak maju sedang, program gerak maju cepat, program gerak mundur, program gerak hadap kanan, program gerak hadap kiri, program gerak putar 180° , dan program gerak gabungan.

BAB V

PENGUJIAN DAN ANALISIS

Pengujian dan analisis dilakukan untuk mengetahui apakah sistem telah bekerja sesuai perancangan. Pengujian dilakukan per blok kemudian secara keseluruhan. Adapun pengujian yang perlu dilakukan sebagai berikut:

- 1). Pengujian Rangkaian Catu Daya
- 2). Pengujian Sinyal Kontrol Servo
 - a). Pengujian Sinyal *High* Servo Tunggal
 - b). Pengujian Sinyal Multi-Servo
- 3). Pengujian Motor DC Servo
- 4). Pengujian Rangkaian *Rotary Switch*
- 5). Pengujian Keseluruhan
 - a). Pengujian Posisi Dasar
 - b). Pengujian Gerak Maju
 - c). Pengujian Gerak Mundur
 - d). Pengujian Gerak Hadap Kanan
 - e). Pengujian Gerak Hadap Kiri
 - f). Pengujian Gerak Putar 180°
 - g). Pengujian Gerak Gabungan

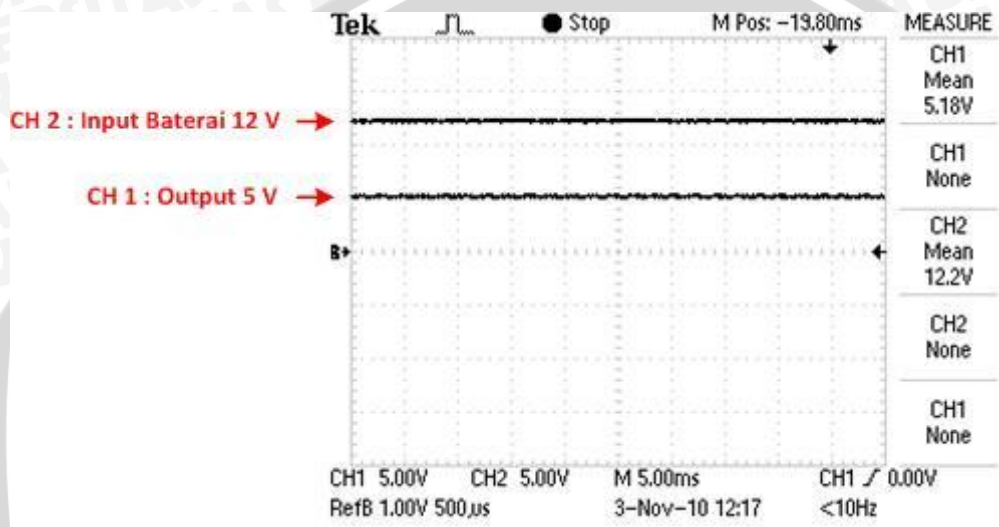
5.1 Pengujian Rangkaian Catu Daya

Pengujian rangkaian catu daya bertujuan untuk mengetahui kesesuaian tegangan keluaran dari catu daya. Pengujian ini diperlukan karena dari sumber tegangan baterai SLA 12 V akan dibagi menjadi 2 rangkaian catu daya yang berbeda, yaitu rangkaian catu daya 5 V dan rangkaian catu daya 6 V. Pengujian dilakukan dengan cara menghubungkan masukan dan keluaran dari rangkaian catu daya dengan osiloskop untuk mengetahui nilai tegangannya. Diagram blok pengujian rangkaian catu daya ditunjukkan dalam Gambar 5.1.



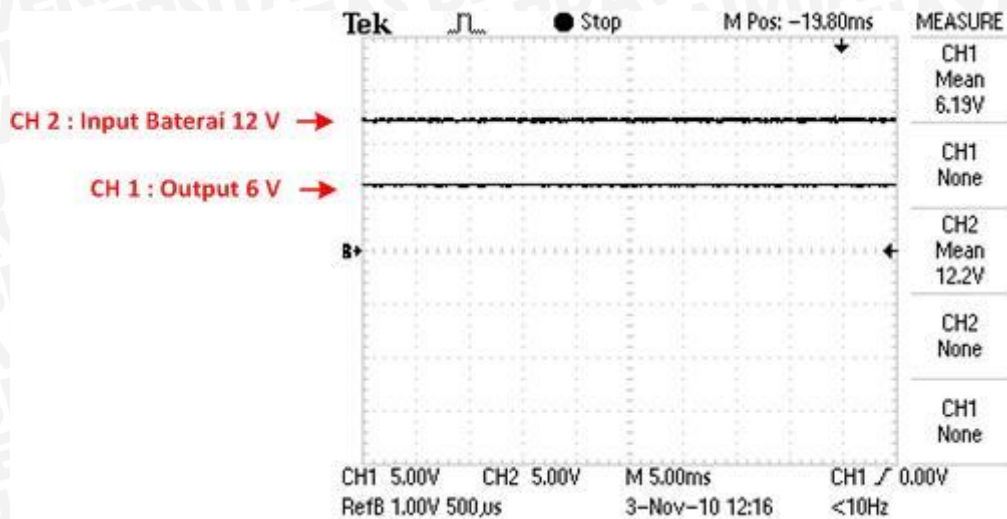
Gambar 5. 1. Diagram Blok Pengujian Rangkaian Catu Daya

Pengujian pertama dilakukan pada rangkaian catu daya 5 V. Pengujian dilakukan dengan menggunakan osiloskop TEKTRONIX TDS-1012B. Channel 1 osiloskop dihubungkan dengan baterai SLA 12 V, sedangkan channel 2 osiloskop dihubungkan dengan keluaran rangkaian regulator LM7805. Hasil pengujian tegangan masukan dan tegangan keluaran pada rangkaian catu daya 5 V ditunjukkan dalam Gambar 5.2.



Gambar 5. 2. Tegangan Masukan dan Keluaran pada Rangkaian Catu Daya 5 V

Dari hasil pengujian yang pertama diperoleh nilai tegangan keluaran sebesar 5,18 V. Nilai tersebut adalah nilai tegangan tanpa beban, yang berarti rangkaian catu daya 5 V dapat menghasilkan tegangan maksimal (tanpa beban) sebesar 5,18 V. Pengujian yang kedua dilakukan pada rangkaian catu daya 6 V menggunakan osiloskop TEKTRONIX TDS-1012B. Channel 1 osiloskop dihubungkan dengan baterai SLA 12 V, sedangkan channel 2 osiloskop dihubungkan dengan keluaran rangkaian regulator LM7806. Hasil pengujian tegangan masukan dan tegangan keluaran pada rangkaian catu daya 6 V ditunjukkan dalam Gambar 5.3.



Gambar 5. 3. Tegangan Masukan dan Keluaran pada Rangkaian Catu Daya 6 V

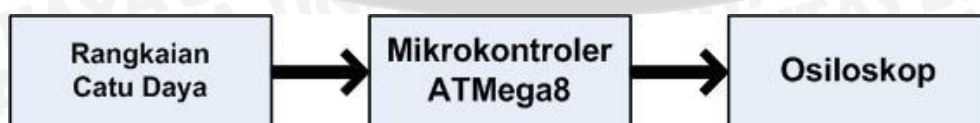
Dari hasil pengujian yang kedua diperoleh nilai tegangan keluaran sebesar 6,19 V. Sehingga bisa disimpulkan bahwa rangkaian catu daya 6 V dapat menghasilkan tegangan maksimal (tanpa beban) sebesar 6,19 V.

5.2 Pengujian Sinyal Kontrol Servo

Pengujian sinyal kontrol servo bertujuan untuk mengetahui bentuk pulsa periodik yang dihasilkan oleh mikrokontroler. Pengujian ini terdiri atas dua bagian yaitu pengujian sinyal *high* servo tunggal dan pengujian sinyal multi-servo.

5.2.1 Pengujian Sinyal *High* Servo Tunggal

Motor DC servo dapat bergerak membentuk sudut tertentu disebabkan oleh pulsa dengan periode 20 ms yang diberikan oleh mikrokontroler. Besarnya sudut yang dibentuk tergantung dari lebarnya sinyal *high* dari pulsa tersebut. Pengujian sinyal *high* servo bertujuan untuk mengetahui ketepatan lebar sinyal *high* yang dihasilkan oleh mikrokontroler. Kestabilan dan ketepatan dari sinyal *high* diperlukan untuk menjaga posisi servo agar sesuai dengan yang diinginkan. Blok diagram pengujian sinyal *high* servo tunggal ditunjukkan dalam Gambar 5.4.

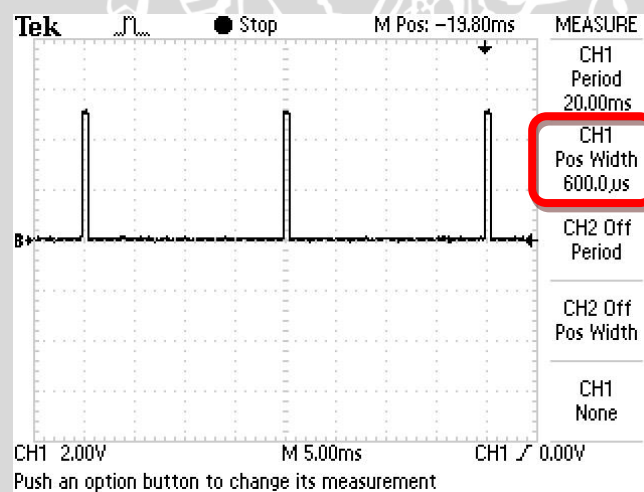


Gambar 5. 4. Diagram Blok Pengujian Sinyal *High* Servo Tunggal

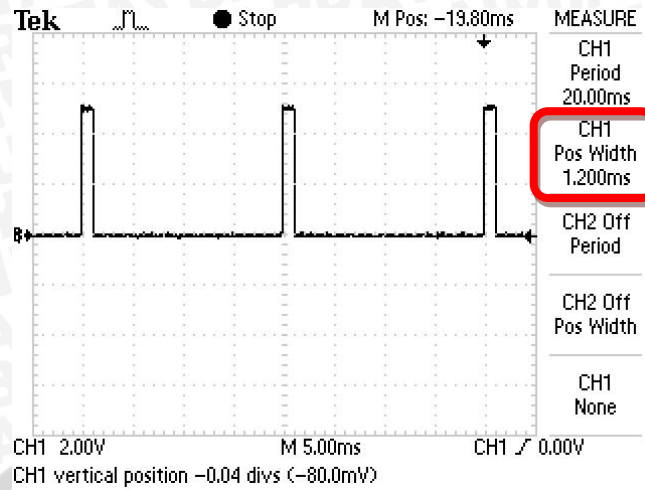
Pengujian sinyal *high* servo tunggal dilakukan dengan ketentuan sebagai berikut :

- 1). Mikrokontroler yang digunakan adalah ATmega8 sesuai dengan mikrokontroler yang digunakan sebagai kontrol servo.
- 2). Mikrokontroler menggunakan sumber clock eksternal sebesar 16 MHz.
- 3). TIMER 1 (timer 16 bit) diset pada mode *Fast PWM, TOP=ICR1*.
- 4). OCR1A dan OCR1B pada TIMER 1 diberi nilai sesuai dengan lebar sinyal kontrol *high* yang diuji. Perhitungan untuk nilai OCR1A dan OCR1B menggunakan persamaan 4.7.
- 5). Mikrokontroler ATmega8 dihubungkan dengan catu daya 5 V.
- 6). Osiloskop TEKTRONIX TDS-1012B digunakan untuk melihat periode sinyal dan lebar sinyal *high* yang dihasilkan oleh mikrokontroler.

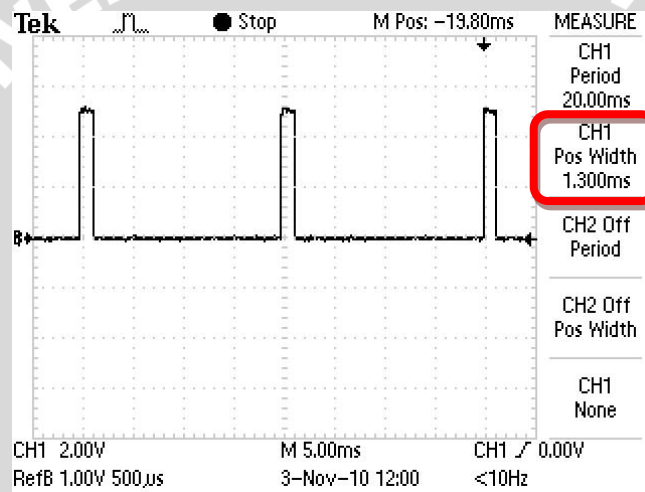
Robot Berkaki Empat hanya menggunakan beberapa nilai sudut tertentu dalam pergerakannya. Oleh karena itu pengujian sinyal *high* servo ini hanya dilakukan pada beberapa nilai sudut tertentu, seperti yang tercantum dalam Tabel 4.7. Hasil pengujian pada osiloskop untuk pengujian lebar sinyal *high* servo tunggal ditunjukkan dalam Gambar 5.5 sampai dengan Gambar 5.15.



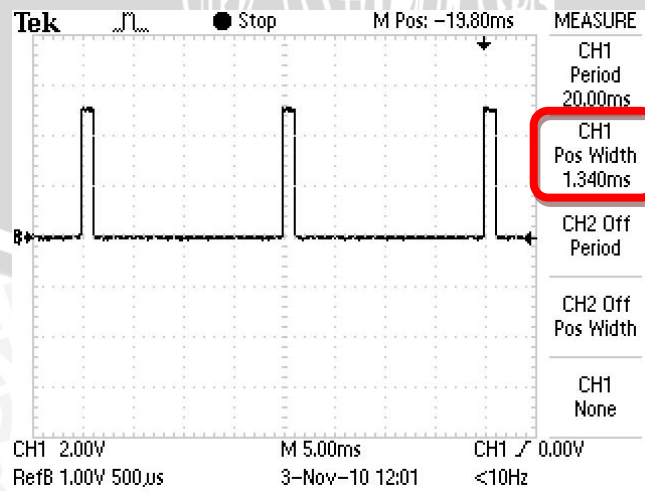
Gambar 5. 5. Sinyal Kontrol Servo dengan Lebar Sinyal *High* 600 μ s



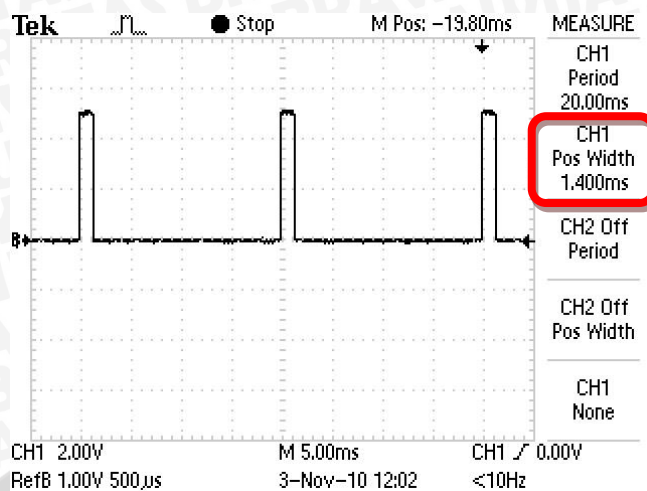
Gambar 5. 6. Sinyal Kontrol Servo dengan Lebar Sinyal High 1200 μ s



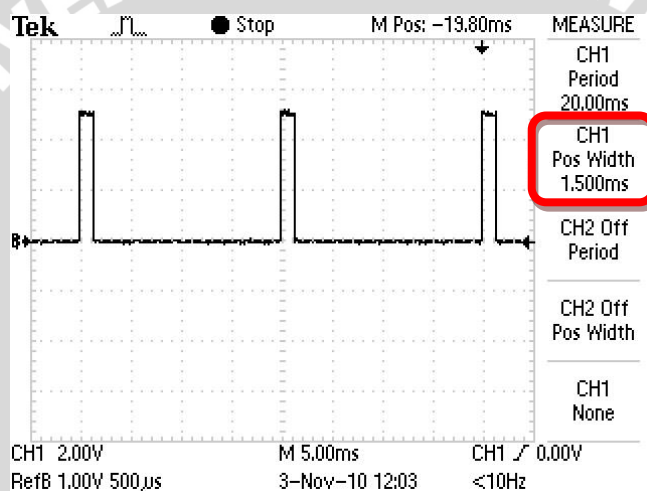
Gambar 5. 7. Sinyal Kontrol Servo dengan Lebar Sinyal High 1300 μ s



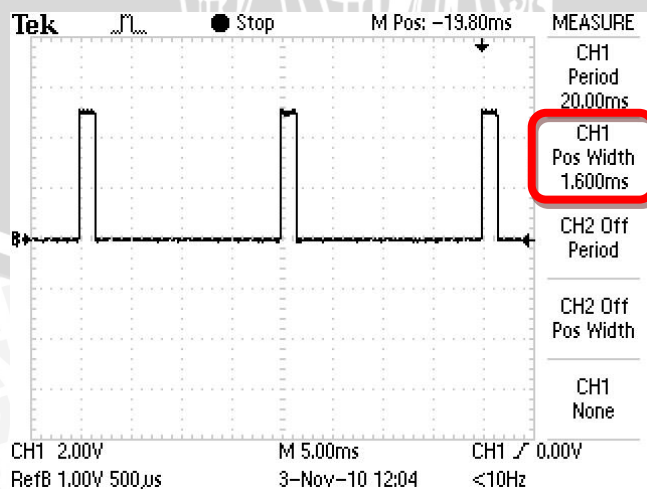
Gambar 5. 8. Sinyal Kontrol Servo dengan Lebar Sinyal High 1350 μ s



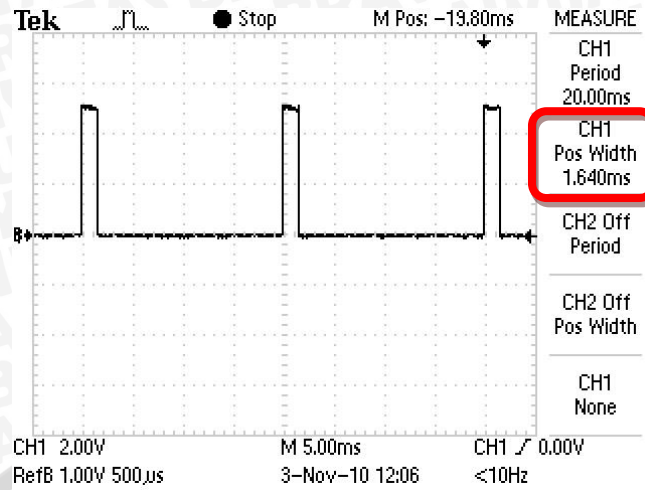
Gambar 5. 9. Sinyal Kontrol Servo dengan Lebar Sinyal High 1400 μs



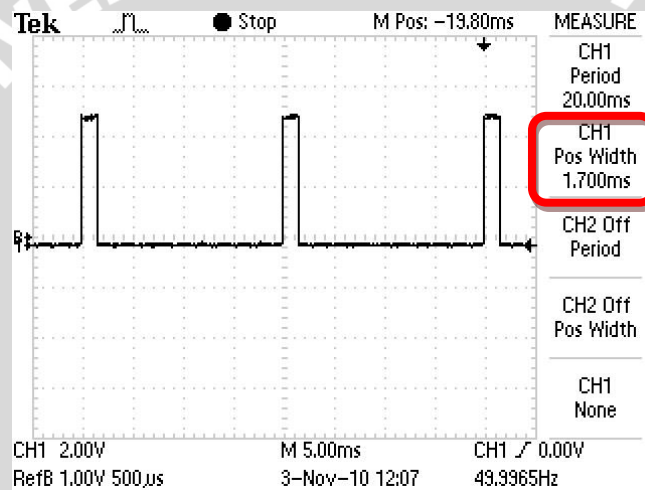
Gambar 5. 10. Sinyal Kontrol Servo dengan Lebar Sinyal High 1500 μs



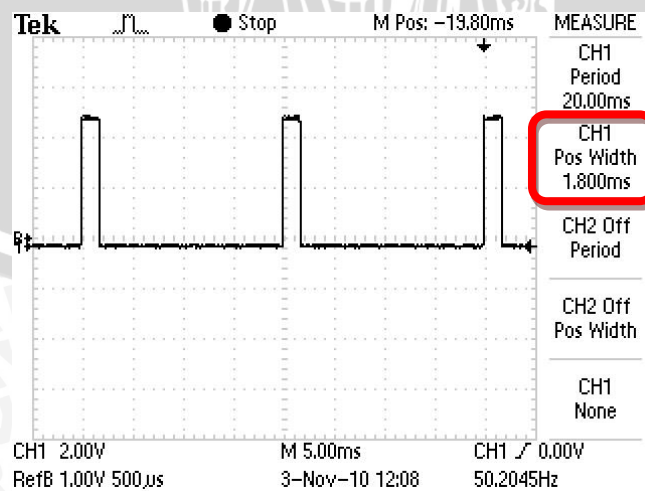
Gambar 5. 11. Sinyal Kontrol Servo dengan Lebar Sinyal High 1600 μs



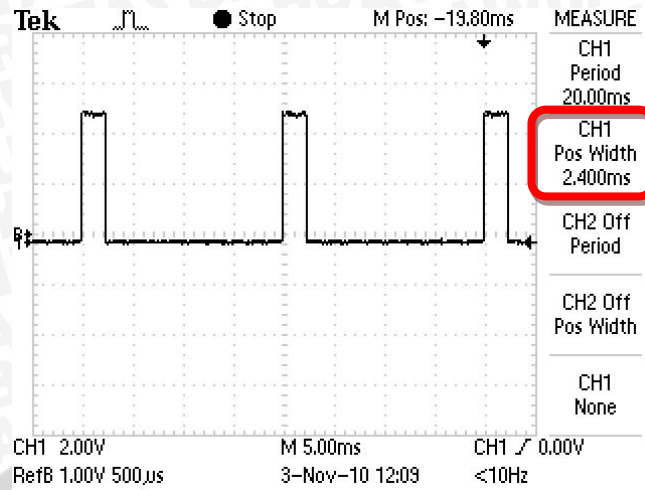
Gambar 5. 12. Sinyal Kontrol Servo dengan Lebar Sinyal High 1650 µs



Gambar 5. 13. Sinyal Kontrol Servo dengan Lebar Sinyal High 1700 µs



Gambar 5. 14. Sinyal Kontrol Servo dengan Lebar Sinyal High 1800 µs



Gambar 5. 15. Sinyal Kontrol Servo dengan Lebar Sinyal *High* 2400 μ s
 Selisih antara nilai yang diinginkan dan hasil pengujian sinyal *high* motor DC servo ditunjukkan dalam Tabel 5.1.

Tabel 5. 1. Hasil Pengujian Sinyal *High* Servo

Sinyal <i>High</i> pada Perancangan (dalam μ s)	Sinyal <i>High</i> pada Pengujian (dalam μ s)	Selisih Teori dan Praktik (dalam μ s)	Kesalahan (dalam %)
600	600	0	0
1200	1200	0	0
1300	1300	0	0
1350	1340	10	0,741
1400	1400	0	0
1500	1500	0	0
1600	1600	0	0
1650	1640	10	0,606
1700	1700	0	0
1800	1800	0	0
2400	2400	0	0
Kesalahan Rata-Rata			0,122

Persentase kesalahan pada Tabel 5.1 dihitung menggunakan rumus:

$$\text{Persentase kesalahan}(\%) = \frac{|\text{Selisih Teori dan Praktik}|}{\text{Sinyal High (teori)}} \times 100\% \quad (5.1)$$

Berdasarkan Tabel 5.1 diperoleh perbandingan antara lebar sinyal *high* yang diinginkan dengan lebar sinyal *high* hasil pengujian. Kesalahan rata-rata lebar sinyal *high* servo yang dihasilkan oleh mikrokontroler ATmega8 adalah 0,122%. Kesimpulan yang dapat diambil adalah bahwa mikrokontroler ATmega8 dapat menghasilkan pulsa periodik dengan baik, dengan catatan lebar sinyal *high* yang dihasilkan memiliki kesalahan rata-rata sebesar 0,122%.



5.2.2 Pengujian Sinyal Kontrol Multi-Servo

Pengujian sinyal kontrol multi-servo bertujuan untuk mengetahui kemampuan mikrokontroler kontrol servo dalam menghasilkan pulsa periodik. Pengujian ini juga bertujuan untuk mengetahui apakah metode multi-servo yang telah dirancang dapat diaplikasikan pada mikrokontroler ATmega8 yang nantinya akan mengontrol 8 buah servo. Sinyal kontrol multi servo diuji menggunakan *logic analyzer* ELAB-080. Diagram blok pengujian sinyal kontrol multi-servo ditunjukkan dalam Gambar 5.16.



Gambar 5. 16. Diagram Blok Pengujian Sinyal Kontrol Multi-Servo

Pengujian pertama dilakukan dengan memberikan nilai OCRA dan OCRB yang sama untuk semua pin I/O. Dalam pengujian ini, semua pin I/O yang dihubungkan pada *logic analyzer* CH00 – CH07 menghasilkan pulsa periodik dengan periode 20 ms, dengan sinyal *high* selebar 1500 μ s. Hasil pengujian sinyal kontrol multi-servo dengan sinyal *high* 1500 μ s ditunjukkan dalam Gambar 5.17.



Gambar 5. 17. Sinyal Kontrol Multi Servo dengan lebar pulsa *High* 1500 μ s

Pengujian kedua dilakukan dengan memberikan nilai OCRA dan OCRB yang berbeda untuk masing-masing pin I/O. Tujuan dari pengujian ini adalah untuk mengetahui apakah metode multi-servo ini dapat digunakan untuk mengontrol banyak servo dengan posisi yang berbeda-beda (sinyal *high* yang berbeda). Dalam pengujian ini

semua pin I/O yang terhubung pada *logic analyzer* CH00 – CH07 menghasilkan pulsa periodik selebar 20 ms dengan lebar sinyal *high* masing-masing sebagai berikut :

- 1). CH00 = 500 μ s
- 2). CH01 = 1000 μ s
- 3). CH02 = 1500 μ s
- 4). CH03 = 2000 μ s
- 5). CH04 = 2500 μ s
- 6). CH05 = 3000 μ s
- 7). CH06 = 3500 μ s
- 8). CH07 = 4000 μ s

Hasil pengujian sinyal kontrol multi-servo dengan lebar sinyal *high* berbeda-beda ditunjukkan dalam Gambar 5.18.



Gambar 5. 18. Sinyal Kontrol Multi-Servo dengan lebar pulsa *High* yang berbeda-beda

Dari kedua pengujian diperoleh kesimpulan bahwa metode kontrol multi-servo dapat diaplikasikan pada mikrokontroler ATmega8 untuk mengontrol 8 buah motor DC servo.

5.3 Pengujian Motor DC Servo

Pengujian motor DC servo terdiri dari 2 bagian, yaitu pengujian sudut motor DC servo dan pengujian arus motor DC servo

5.3.1 Pengujian Sudut Motor DC Servo

Pengujian sudut motor DC servo pada Robot Berkaki Empat bertujuan untuk mengetahui arah dan besar sudut putar motor DC servo, dan juga untuk mengetahui apakah sinyal kontrol mikrokontroler dapat mengontrol posisi motor DC servo. Pengujian dilakukan dengan menghubungkan kedelapan motor DC servo dengan mikrokontroler ATmega8 dan rangkaian catu daya. Untuk motor DC servo digunakan catu daya 6 V, sedangkan untuk mikrokontroler ATmega8 digunakan catu daya 5 V. Diagram blok pengujian motor DC servo ditunjukkan dalam Gambar 5.19.



Gambar 5. 19. Diagram Blok Pengujian Motor DC Servo

Pada pengujian ini motor DC servo diberi perintah untuk membentuk sudut awal 90° (tengah), 60° , dan 120° , sehingga diperoleh pulsa *high* yang dibutuhkan untuk membentuk sudut-sudut tersebut. Motor servo akan berada pada posisi tengah (90°) jika diberi sinyal *high* sebesar $1500 \mu\text{s}$. Akan tetapi karena kendala mekanik (penguncian motor servo, penempatan posisi kaki pada kepala servo yang berbeda-beda), untuk membentuk sudut 90° sinyal *high* tiap servo tidak selalu tepat $1500 \mu\text{s}$. Untuk itu pada awal pengujian motor DC servo diberi sinyal *high* tertentu sampai membentuk sudut 90° , kemudian sinyal *high* tersebut dikurangi atau ditambah agar dapat membentuk sudut 60° atau 120° . Hasil pengujian motor DC servo ditunjukkan dalam Tabel 5.2.

Tabel 5. 2. Hasil Pengujian Motor DC Servo

Motor DC Servo	Sudut 90°		Sudut 60°		Sudut 120°	
	Sinyal High	Arah Putar	Sinyal High	Arah Putar	Sinyal High	Arah Putar
Pinggang 1	1315 μ s	Tengah	965 μ s	CCW	1665 μ s	CW
Pinggang 2	1550 μ s	Tengah	1200 μ s	CCW	1900 μ s	CW
Pinggang 3	1430 μ s	Tengah	1080 μ s	CCW	1780 μ s	CW
Pinggang 4	1485 μ s	Tengah	1135 μ s	CCW	1835 μ s	CW
Lutut 1	1520 μ s	Tengah	1220 μ s	CCW	1820 μ s	CW
Lutut 2	1395 μ s	Tengah	1095 μ s	CCW	1695 μ s	CW
Lutut 3	1465 μ s	Tengah	1165 μ s	CCW	1765 μ s	CW
Lutut 4	1625 μ s	Tengah	1325 μ s	CCW	1925 μ s	CW

Keterangan:

1. CW : *clockwise*
2. CCW : *counter clockwise*

Dari hasil pengujian diketahui bahwa dengan memperbesar sinyal *high* servo akan menggerakkan servo searah jarum jam (*clockwise*), sedangkan mengurangi lebar sinyal *high* akan menggerakkan servo berlawanan arah jarum jam (*counter clockwise*). Berdasarkan Tabel 5.2, terdapat perbedaan selisih sinyal *high* yang dibutuhkan antara servo pinggang dan servo lutut untuk mengubah sudut dari sudut 90° ke sudut 60° maupun ke sudut 120°. Pada servo pinggang sinyal *high* servo harus dikurangi sebesar 350 μ s untuk mengubah sudut dari sudut 90° ke sudut 60°, sedangkan servo lutut membutuhkan pengurangan sinyal *high* sebesar 300 μ s. Perbedaan juga terjadi pada saat mengubah sudut servo dari sudut 90° ke sudut 120°. Pada servo pinggang dibutuhkan penambahan sinyal *high* sebesar 350 μ s, sedangkan pada servo lutut hanya dibutuhkan penambahan sebesar 300 μ s.

Perbedaan ini disebabkan servo pinggang memiliki beban yang lebih besar karena fungsinya sebagai penopang tubuh robot pada saat bergerak, sehingga dibutuhkan lebar sinyal *high* yang berbeda untuk membentuk sudut yang sama dengan servo lutut. Kesimpulan yang dapat diambil dari pengujian ini adalah bahwa pulsa periodik yang dihasilkan oleh mikrokontroler ATmega8 dapat digunakan untuk mengontrol posisi motor DC servo.

5.3.2 Pengujian Arus Motor DC Servo

Pengujian arus motor DC servo bertujuan untuk mengetahui besarnya arus yang dibutuhkan untuk dapat menggerakkan 8 buah motor DC servo. Selain itu juga bertujuan untuk membuktikan bahwa rangkaian catu daya 6 V dapat berfungsi dengan baik untuk menunjang penggunaan motor DC servo sebagai penggerak Robot Berkaki Empat. Diagram blok pengujian arus motor DC servo ditunjukkan dalam Gambar 5.20.



Gambar 5. 20. Diagram blok pengujian arus motor DC servo

Pengujian yang pertama dilakukan pada satu buah motor DC servo. Motor servo digerakkan secara terus menerus dalam kondisi bebas (tanpa beban) dan diukur arusnya. Pengukuran dilakukan menggunakan amperemeter tipe HC-3500T. Hasil pengukuran arus yang dibutuhkan pada servo tunggal ditunjukkan dalam Gambar 5.21.



Gambar 5. 21. Besar Arus untuk Motor DC Servo Tunggal

Dari Gambar 5.21 dapat diketahui bahwa arus yang dibutuhkan satu buah motor DC servo untuk bergerak adalah 300 mA. Pengujian kedua dilakukan pada kedelapan buah servo Robot Berkaki Empat secara bersama-sama. Robot digerakkan dalam kondisi berbeban (bergerak di atas tanah) dan diukur arusnya menggunakan amperemeter tipe HC-3500T. Hasil pengukuran terhadap arus yang dibutuhkan 8 buah motor DC servo pada saat bergerak ditunjukkan dalam Gambar 5.22.



Gambar 5. 22. Besar Arus yang Dibutuhkan 8 Motor DC Servo

Dari Gambar 5.22 dapat diketahui bahwa arus yang dibutuhkan untuk menggerakkan 8 buah motor DC servo secara bersama-sama adalah 2,16 A. Hasil pengukuran ini memiliki selisih 0,24 A dari hasil perhitungan pada perancangan yang besarnya 2,4 A. Kesimpulan yang dapat diambil adalah rangkaian catu daya 6 V dapat menyuplai kebutuhan arus motor DC servo Robot Berkaki Empat dengan baik.

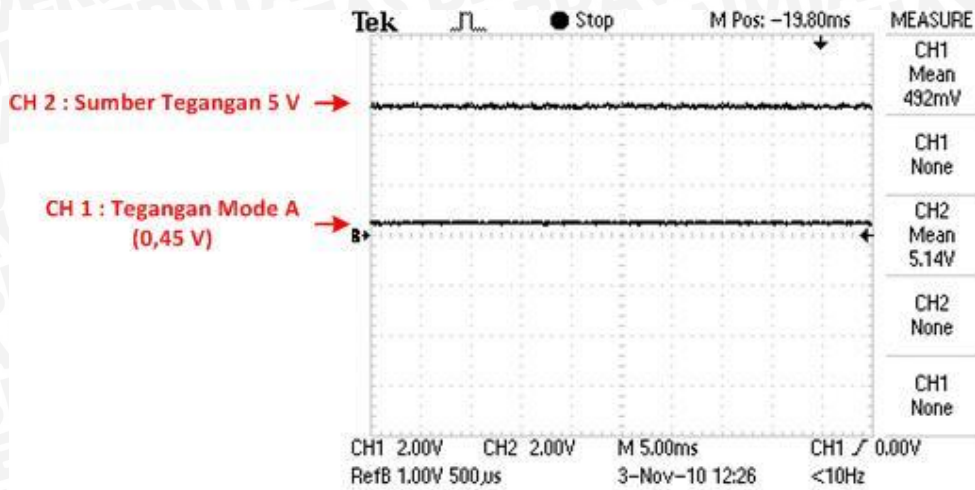
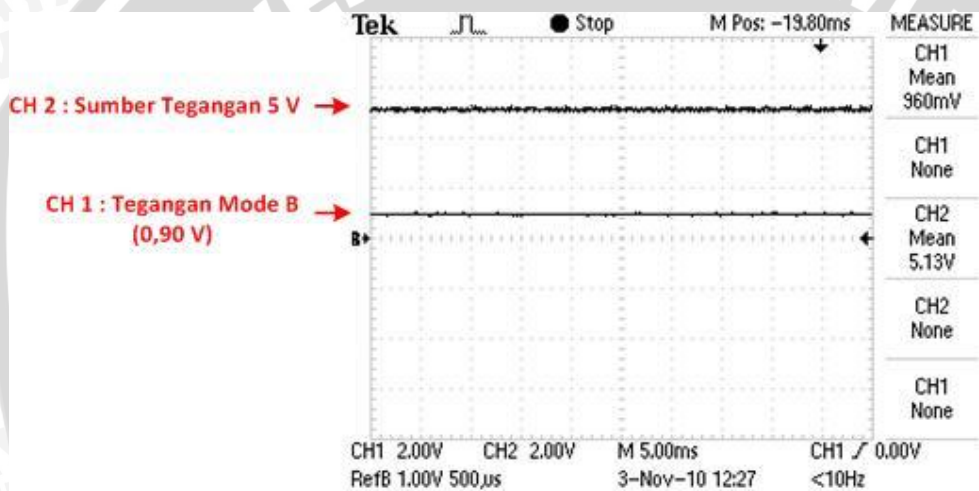
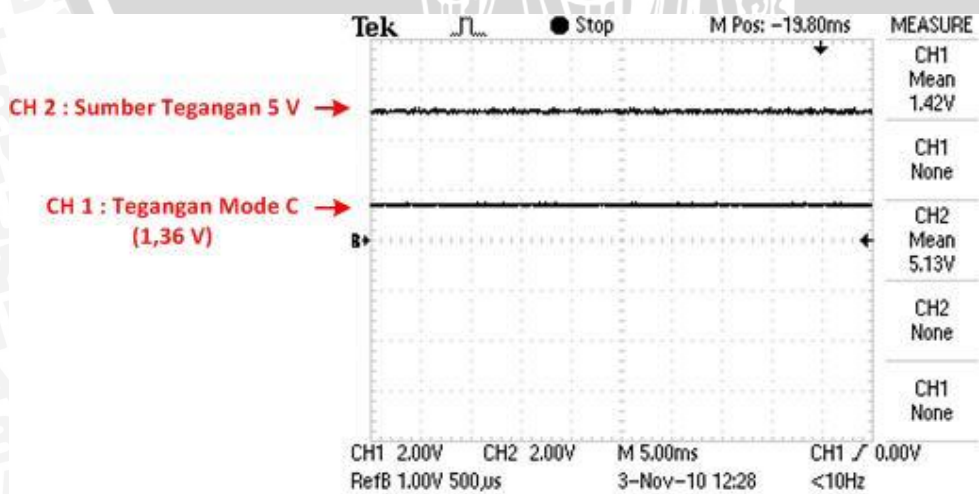
5.4 Pengujian Rangkaian *Rotary Switch*

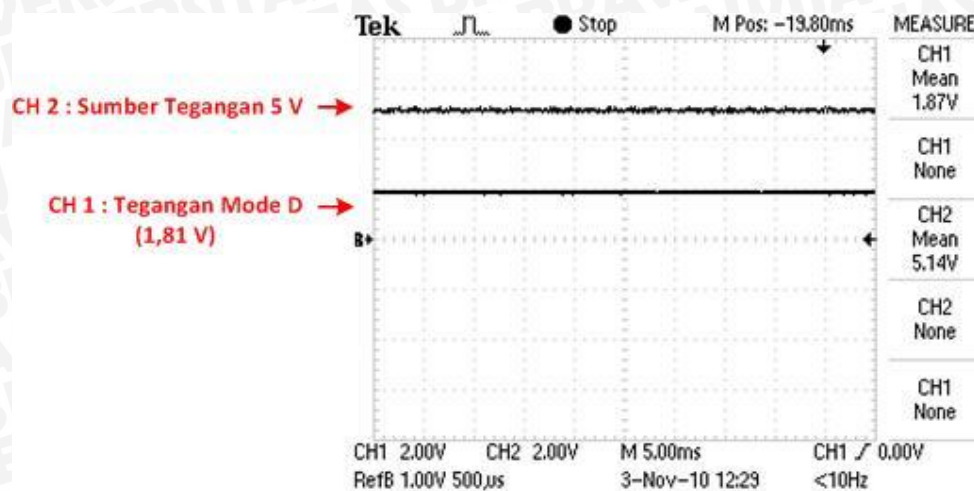
Pengujian rangkaian *rotary switch* bertujuan untuk mengetahui ketepatan nilai tegangan analog yang dihasilkan oleh rangkaian ini. Pada perancangan, *rotary switch* dirancang sebagai selektor mode program yang akan dijalankan pada Robot Berkaki Empat. 11 level tegangan yang dihasilkan oleh rangkaian ini akan diubah menjadi sinyal digital oleh ADC mikrokontroler. Ketepatan nilai tegangan output pada rangkaian *rotary switch* sangat dibutuhkan agar ADC mikrokontroler dapat menerjemahkan tegangan analog tersebut menjadi nilai digital yang benar dan sesuai dengan perancangan. Blok diagram pengujian rangkaian *rotary switch* ditunjukkan dalam Gambar 5.23.



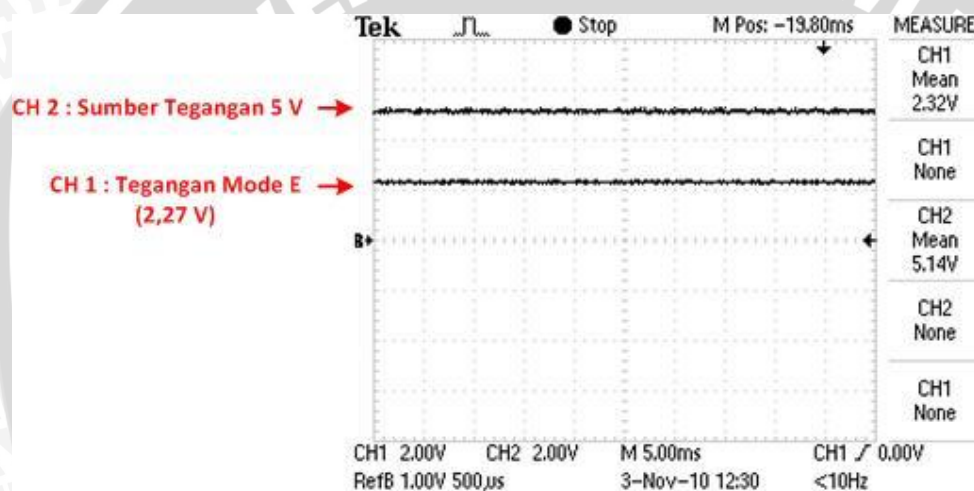
Gambar 5. 23. Diagram Blok Pengujian Rangkaian *Rotary Switch*

Pengujian rangkaian ini mengacu pada perancangan rangkaian *rotary switch*. Dari Tabel 4.5 didapatkan hasil perancangan 11 level tegangan yang berbeda dari rangkaian *rotary switch*. Pengujian dilakukan menggunakan osiloskop TEKTRONIX TDS-1012B. Hasil pengujian kesebelas level tegangan tersebut ditunjukkan dalam Gambar 5.24 sampai dengan Gambar 5.34.

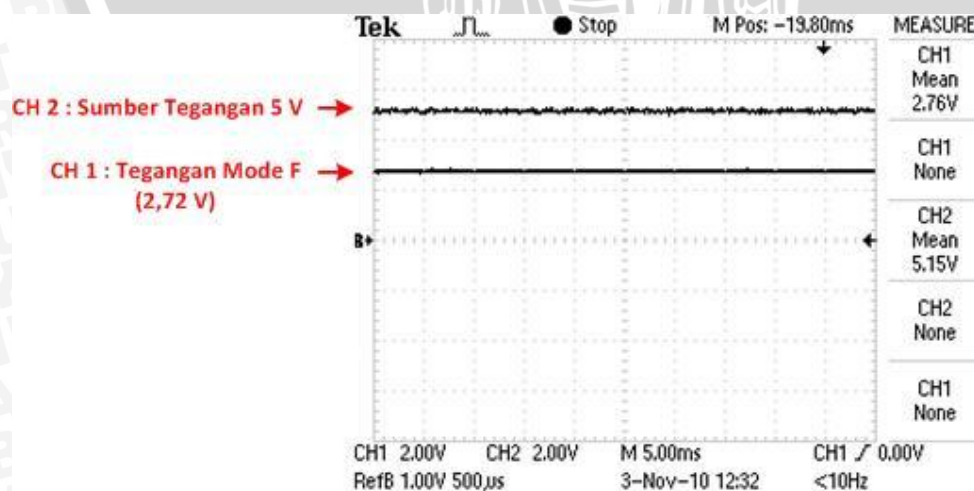
Gambar 5. 24. Nilai Tegangan Output Rangkaian *Rotary Switch* Mode A (0,45 V)Gambar 5. 25. Nilai Tegangan Output Rangkaian *Rotary Switch* Mode B (0,90 V)Gambar 5. 26. Nilai Tegangan Output Rangkaian *Rotary Switch* Mode C (1,36 V)



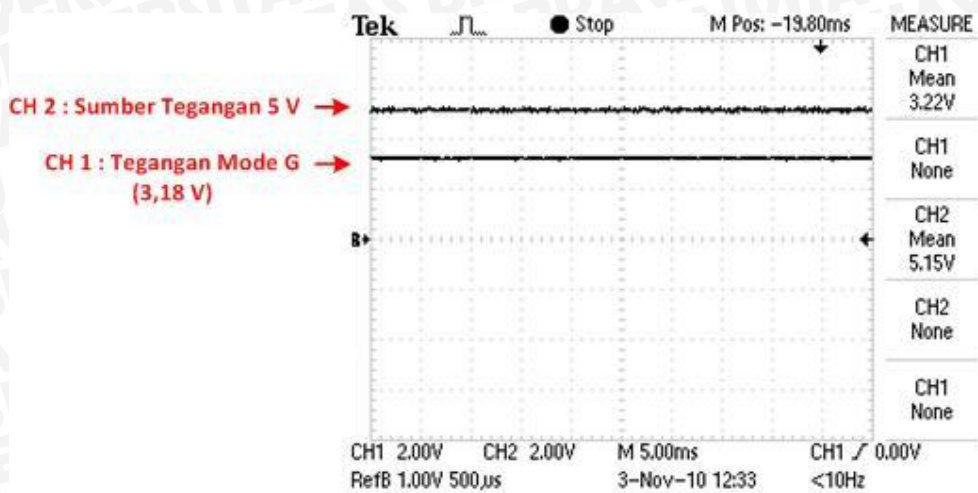
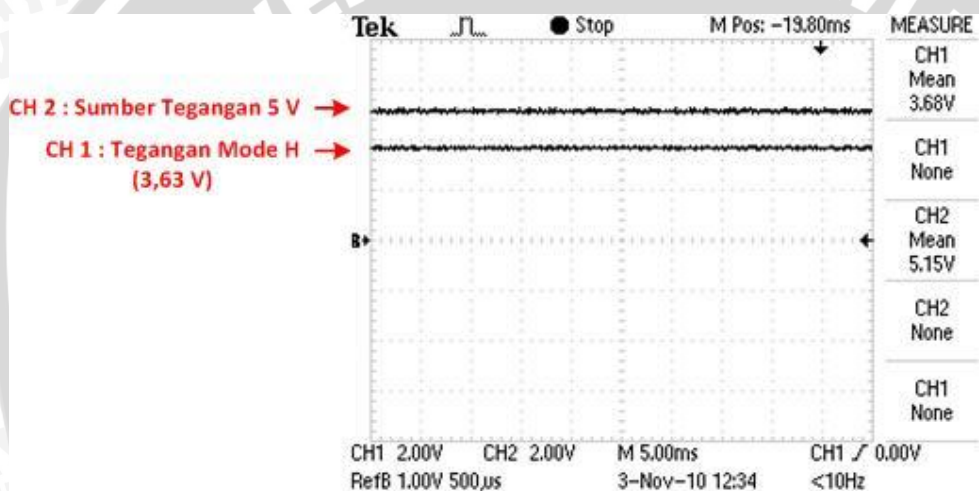
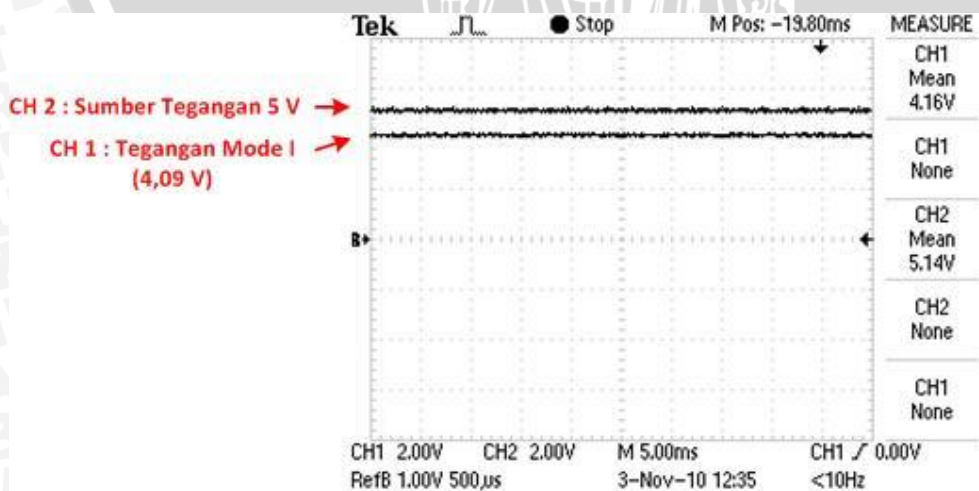
Gambar 5. 27. Nilai Tegangan Output Rangkaian Rotary Switch Mode D (1,81 V)

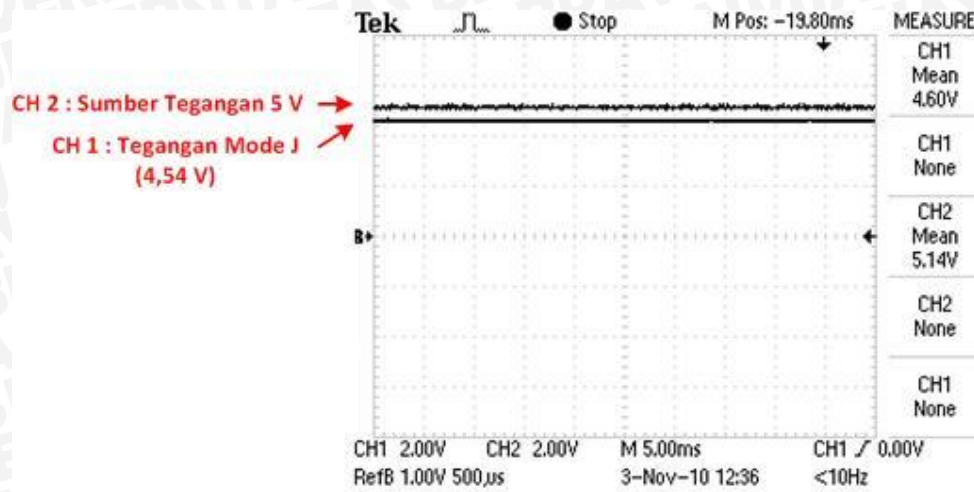


Gambar 5. 28. Nilai Tegangan Output Rangkaian Rotary Switch Mode E (2,27 V)

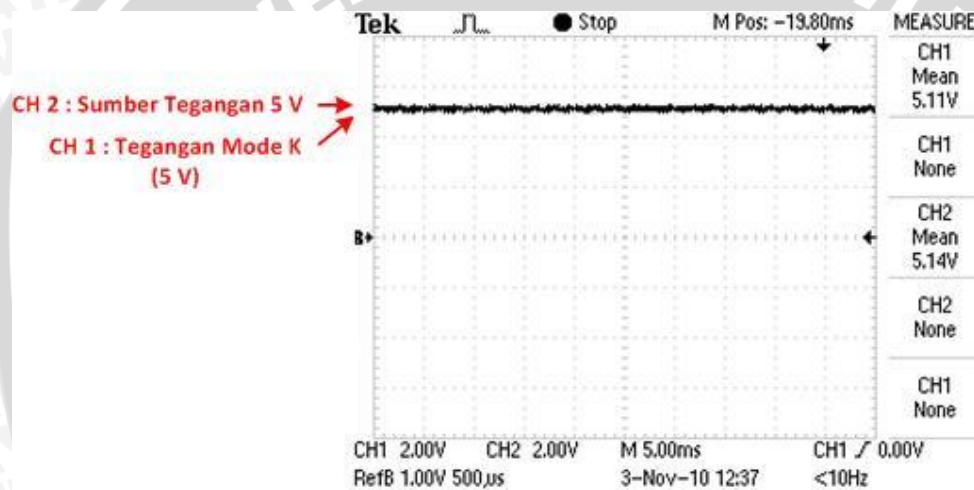


Gambar 5. 29. Nilai Tegangan Output Rangkaian Rotary Switch Mode F (2,72 V)

Gambar 5. 30. Nilai Tegangan Output Rangkaian *Rotary Switch* Mode G (3,18 V)Gambar 5. 31. Nilai Tegangan Output Rangkaian *Rotary Switch* Mode H (3,63 V)Gambar 5. 32. Nilai Tegangan Output Rangkaian *Rotary Switch* Mode I (4,09 V)



Gambar 5. 33. Nilai Tegangan Output Rangkaian *Rotary Switch* Mode J (4,54 V)



Gambar 5. 34. Nilai Tegangan Output Rangkaian *Rotary Switch* Mode K (5 V)

Selisih antara hasil perhitungan pada perancangan dan hasil pengujian rangkaian *rotary switch* ditunjukkan dalam Tabel 5.3.

Tabel 5. 3. Hasil Pengujian Rangkaian *Rotary Switch*

Tegangan Output Hasil Perhitungan (dalam volt)	Tegangan Output pada Pengujian (dalam volt)	Selisih Teori dan Praktik (dalam volt)	Kesalahan (dalam %)
0,45	0,492	0,042	9,333
0,90	0,960	0,06	6,667
1,36	1,42	0,06	4,412
1,81	1,87	0,06	3,315
2,27	2,32	0,05	2,203
2,72	2,76	0,04	1,471
3,18	3,22	0,04	1,258
3,63	3,68	0,05	1,377
4,09	4,16	0,07	1,711
4,54	4,60	0,06	1,322
5,00	5,11	0,11	2,2
Kesalahan Rata-Rata			3,206

Persentase kesalahan pada Tabel 5.3 dihitung menggunakan rumus:

$$\text{Persentase kesalahan}(\%) = \frac{|\text{Selisih Teori dan Praktik}|}{\text{Tegangan Output Hasil Perhitungan}} \times 100\% \quad (5.2)$$

Dari Tabel 5.3 dapat diketahui bahwa kesalahan rata-rata tegangan output *rotary switch* sebesar 3,206 %. Dari hasil ini dapat disimpulkan bahwa rangkaian *rotary switch* dapat bekerja dengan baik dengan catatan tegangan yang dihasilkan pada tiap mode memiliki kesalahan rata-rata sebesar 3,206 %.

5.5 Pengujian Keseluruhan

Pengujian keseluruhan bertujuan untuk mengetahui penerapan metode pergerakan Robot Berkaki Empat yang telah dirancang. Pengujian keseluruhan mencakup pengujian posisi dasar, pengujian gerak maju, pengujian gerak mundur, pengujian gerak hadap kanan, pengujian gerak hadap kiri, dan pengujian gerak putar 180°.

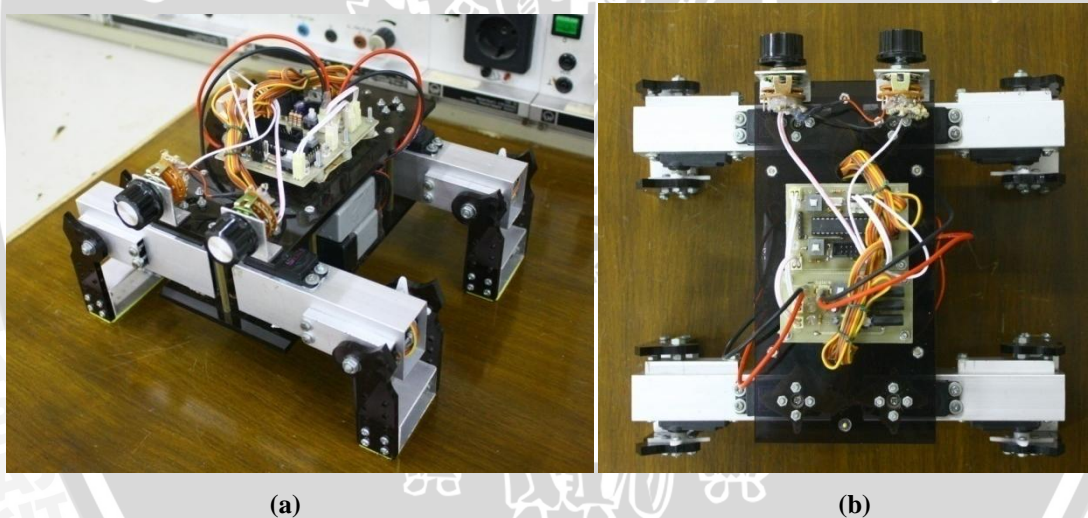
5.5.1 Pengujian Posisi Dasar

Pengujian posisi dasar bertujuan untuk mengetahui keseimbangan robot saat berada pada posisi dasar. Posisi dasar adalah dimana keempat kaki robot menginjak alas dan semua motor DC servo membentuk sudut 90°. Pengujian posisi dasar dengan 4 kaki dilakukan dengan memberikan sinyal kontrol *high* pada setiap motor DC servo seperti ditunjukkan dalam Tabel 5.4.

Tabel 5. 4. Hasil Pengujian Sinyal *High* untuk Posisi Dasar dengan 4 Kaki

Motor DC Servo	Lebar Sinyal <i>High</i> (dalam μs)	Nilai OCR (dalam Desimal)
Servo Pinggang 1	1315	2630
Servo Pinggang 2	1550	3100
Servo Pinggang 3	1430	2860
Servo Pinggang 4	1485	2970
Servo Lutut 1	1520	3040
Servo Lutut 2	1395	2790
Servo Lutut 3	1465	2930
Servo Lutut 4	1625	3250

Robot Berkaki Empat saat berada pada posisi dasar dengan 4 kaki ditunjukkan dalam Gambar 5.35.



Gambar 5. 35. Robot Berkaki Empat saat Posisi Dasar dengan 4 Kaki (a) Robot tampak Perspektif
(b) Robot tampak Atas

Dari hasil pengujian dapat disimpulkan bahwa robot berada dalam kondisi seimbang pada posisi dasar, dimana keempat kakinya menginjak alas. Posisi dasar akan selalu dijalankan pada saat robot baru diaktifkan, atau setelah robot mengakhiri gerakannya.

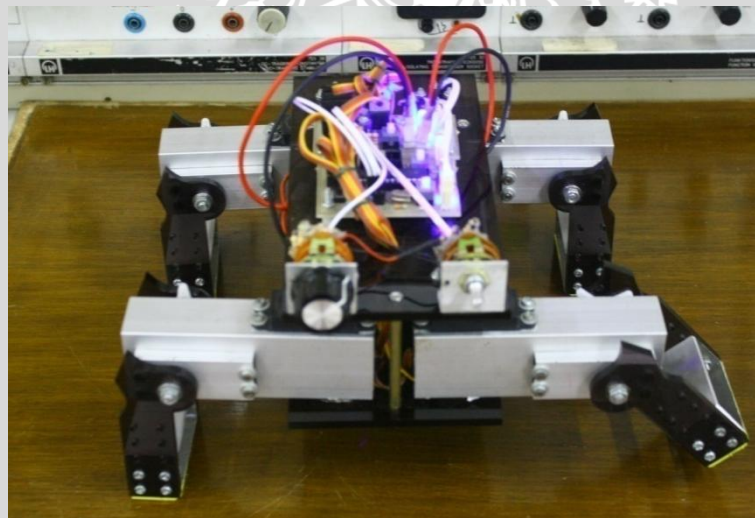
Selain posisi dasar seperti ditunjukkan dalam Gambar 5.35, pengujian juga dilakukan pada posisi dasar dimana hanya 3 kaki yang menapak di atas permukaan.

Pengujian ini dilakukan dengan memberikan sinyal kontrol *high* pada setiap motor DC servo seperti ditunjukkan dalam Tabel 5.5.

Tabel 5. 5. Hasil Pengujian Sinyal *High* untuk Posisi Dasar dengan 3 Kaki

Motor DC Servo	Lebar Sinyal <i>High</i> (dalam μs)	Nilai OCR (dalam Desimal)
Servo Pinggang 1	1315	2630
Servo Pinggang 2	1550	3100
Servo Pinggang 3	1430	2860
Servo Pinggang 4	1485	2970
Servo Lutut 1	1070	2140
Servo Lutut 2	1395	2790
Servo Lutut 3	1915	3830
Servo Lutut 4	1625	3250

Robot Berkaki Empat saat berada pada posisi dasar dengan 3 kaki ditunjukkan dalam Gambar 5.36.



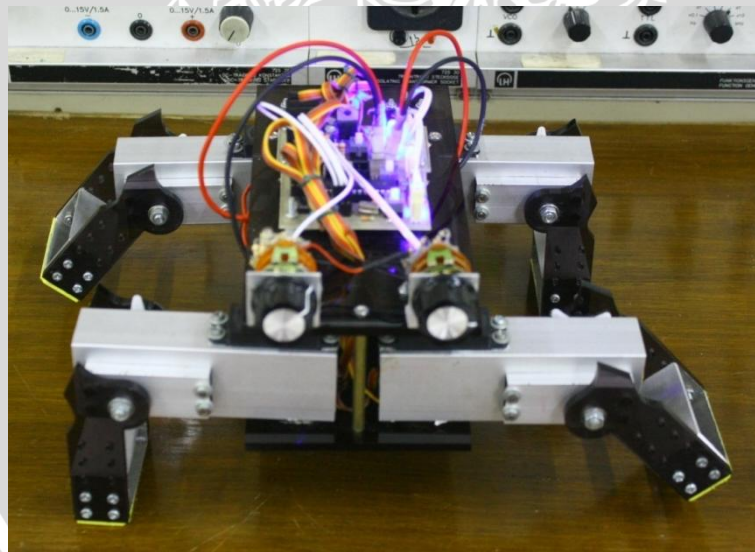
Gambar 5. 36. Robot Berkaki Empat saat Posisi Dasar dengan 3 Kaki

Pengujian selanjutnya adalah pengujian posisi dasar robot dengan hanya 2 kaki yang menginjak permukaan. Tujuan pengujian ini adalah untuk mengetahui apakah robot tetap dalam keadaan seimbang saat hanya 2 kaki saja yang menopang tubuhnya, dengan posisi kedua kaki saling bersilangan. Pengujian posisi dasar 2 kaki dilakukan dengan memberikan sinyal kontrol *high* pada setiap motor DC servo seperti ditunjukkan dalam Tabel 5.6.

Tabel 5. 6. Hasil Pengujian Sinyal *High* untuk Posisi Dasar dengan 2 Kaki

Motor DC Servo	Lebar Sinyal <i>High</i> (dalam μs)	Nilai OCR (dalam Desimal)
Servo Pinggang 1	1315	2630
Servo Pinggang 2	1550	3100
Servo Pinggang 3	1430	2860
Servo Pinggang 4	1485	2970
Servo Lutut 1	1070	2140
Servo Lutut 2	1395	2790
Servo Lutut 3	1465	2930
Servo Lutut 4	1625	3250

Robot Berkaki Empat saat berada pada posisi dasar dengan 2 kaki ditunjukkan dalam Gambar 5.37.



Gambar 5. 37. Robot Berkaki Empat saat Posisi Dasar dengan 2 Kaki

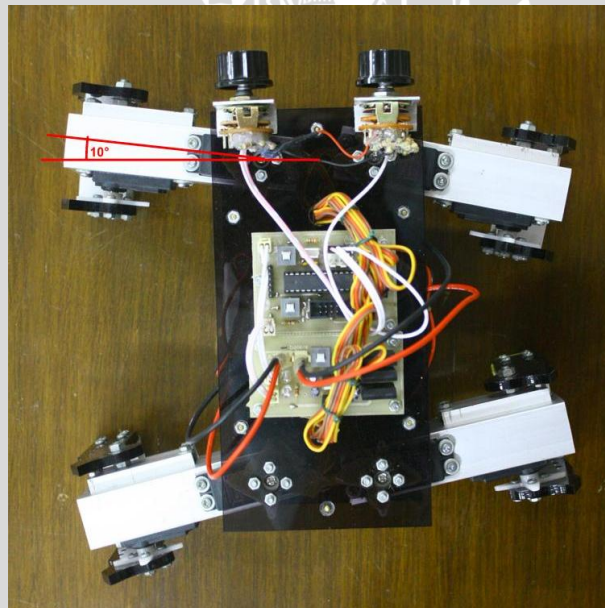
Dari Gambar 5.36 dan Gambar 5.37 dapat disimpulkan bahwa Robot Berkaki Empat tetap dalam keadaan seimbang meskipun hanya 3 kaki atau 2 kaki saja yang menopang tubuhnya.

5.5.2 Pengujian Gerak Maju

Pengujian gerak maju Robot Berkaki Empat bertujuan untuk mengetahui kemampuan robot dalam melakukan gerak maju. Pengujian dibagi menjadi 3 bagian, yaitu pengujian gerak maju dengan kecepatan pelan, pengujian gerak maju dengan kecepatan sedang, dan pengujian gerak maju dengan kecepatan cepat. Pengujian dilakukan dengan mengatur robot agar melakukan gerakan melangkah pada bidang datar sejauh 1 meter, kemudian dicatat waktunya dan dihitung kecepatan gerakannya.

5.5.2.1 Pengujian Gerak Maju Pelan

Gerak maju dengan kecepatan pelan adalah gerak maju dimana servo pinggang hanya berputar sebesar 10° pada setiap langkahnya. Akibatnya adalah perpindahan posisi kaki robot akan lebih sedikit daripada pada saat servo pinggang berputar dengan sudut yang lebih besar. Sudut yang dibentuk oleh servo pinggang pada gerak maju pelan ditunjukkan dalam Gambar 5.38.



Gambar 5. 38. Gerak Maju Pelan Robot Berkaki Empat dengan Sudut Servo Pinggang 10°

Jarak yang ditempuh robot dalam pengujian ini adalah 1 meter. Percobaan dilakukan sebanyak 10 kali. Hasil pengujian gerak maju pelan pada Robot Berkaki Empat ditunjukkan dalam Tabel 5.7.

Tabel 5. 7. Hasil Pengujian Gerak Maju Pelan pada Robot Berkaki Empat

Pengujian Ke-	Waktu Tempuh (dalam detik)	Kecepatan Gerak (dalam cm/detik)
1	14,04	7,12
2	14,46	6,92
3	14,81	6,75
4	14,37	6,96
5	14,01	7,14
6	14,77	6,77
7	14,46	6,92
8	14,20	7,04
9	14,30	6,99
10	13,93	7,18
Rata-rata	14,33	6,98

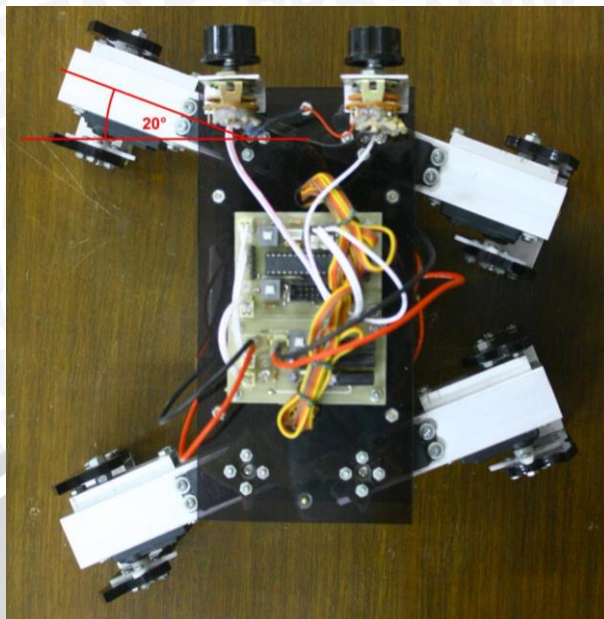
Kecepatan gerak untuk semua pengujian dihitung menggunakan rumus:

$$\text{Kecepatan(cm/detik)} = \frac{\text{Jarak tempuh (cm)}}{\text{Waktu tempuh (detik)}} \quad (5.3)$$

Dari Tabel 5.7 dapat diketahui bahwa Robot Berkaki Empat dapat menempuh jarak 1 m dengan waktu rata-rata 14,33 detik. Dari hasil pengujian ini dapat disimpulkan bahwa gerak maju dengan kecepatan pelan pada Robot Berkaki Empat memiliki kecepatan rata-rata 6,98 cm/detik.

5.5.2.2 Pengujian Gerak Maju Sedang

Gerak maju dengan kecepatan sedang memiliki sudut gerak servo pinggang lebih besar. Pada mode ini servo pinggang berputar dengan sudut 20° pada setiap langkahnya. Sudut yang dibentuk servo pinggang pada gerak maju dengan kecepatan sedang ditunjukkan dalam Gambar 5.39.



Gambar 5. 39. Gerak Maju Sedang Robot Berkaki Empat dengan Sudut Servo Pinggang 20°

Hasil pengujian gerak maju dengan kecepatan sedang pada Robot Berkaki Empat ditunjukkan dalam Tabel 5.8.

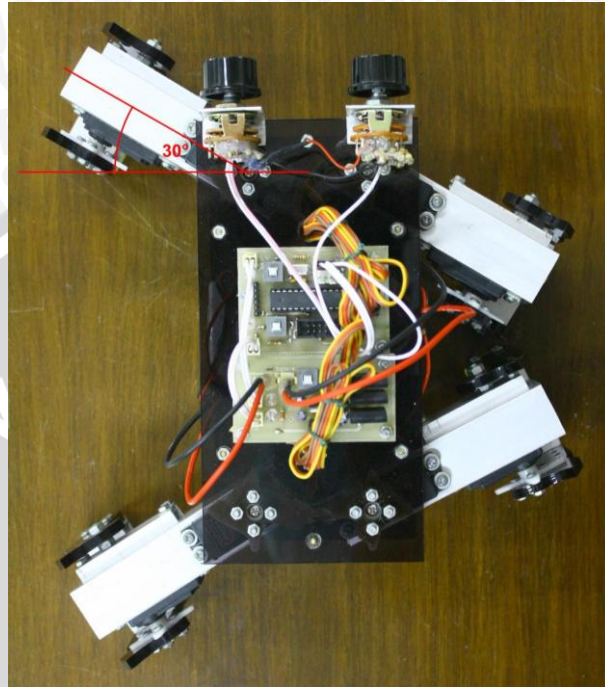
Tabel 5. 8. Hasil Pengujian Gerak Maju Sedang pada Robot Berkaki Empat

Pengujian Ke-	Waktu Tempuh (dalam detik)	Kecepatan Gerak (dalam cm/detik)
1	6,99	14,31
2	6,95	14,39
3	6,92	14,45
4	7,11	14,06
5	7,12	14,04
6	7,09	14,10
7	7,03	14,22
8	7,01	14,26
9	7,25	13,79
10	7,21	13,87
Rata-rata	7,07	14,15

Dari hasil pengujian dapat diambil kesimpulan bahwa Robot Berkaki Empat pada saat bergerak maju dengan kecepatan sedang memiliki kecepatan rata-rata 14,15 cm/detik.

5.5.2.3 Pengujian Gerak Maju Cepat

Gerak maju cepat pada Robot Berkaki Empat memiliki sudut gerak servo pinggang sebesar 30° pada setiap langkahnya. Sudut yang dibentuk servo pinggang pada gerak maju cepat ditunjukkan dalam Gambar 5.40.



Gambar 5. 40. Gerak Maju Cepat Robot Berkaki Empat dengan Sudut Servo Pinggang 30°

Hasil pengujian gerak maju cepat pada Robot Berkaki Empat ditunjukkan dalam Tabel 5.9.

Tabel 5. 9. Hasil Pengujian Gerak Maju Cepat pada Robot Berkaki Empat

Pengujian Ke-	Waktu Tempuh (dalam detik)	Kecepatan Gerak (dalam cm/detik)
1	6,16	16,23
2	5,93	16,86
3	5,77	17,33
4	5,84	17,12
5	5,67	17,63
6	5,61	17,82
7	5,82	17,18
8	5,57	17,95
9	5,63	17,76
10	5,84	17,12
Rata-rata	5,78	17,30

Dari hasil pengujian dapat diambil kesimpulan bahwa Robot Berkaki Empat pada saat bergerak maju cepat memiliki kecepatan rata-rata 17,30 cm/detik.

5.5.3 Pengujian Gerak Mundur

Pengujian gerak mundur pada Robot Berkaki Empat bertujuan untuk mengetahui kemampuan robot pada saat berjalan mundur atau berjalan melawan arah hadap robot. Pengujian dilakukan dengan mengatur robot agar berjalan mundur sejauh 1 meter. Untuk pengujian gerak mundur ini, besar sudut servo pinggang yang digunakan adalah 30° (mode gerak mundur cepat). Percobaan dilakukan sebanyak 10 kali. Hasil pengujian gerak mundur pada Robot Berkaki Empat ditunjukkan dalam Tabel 5.10.

Tabel 5. 10. Hasil Pengujian Gerak Mundur pada Robot Berkaki Empat

Pengujian Ke-	Waktu Tempuh (dalam detik)	Kecepatan Gerak (dalam cm/detik)
1	6,13	16,31
2	5,90	16,95
3	5,87	17,03
4	5,87	17,03
5	6,13	16,31
6	5,82	17,18
7	5,56	17,98
8	5,64	17,73
9	5,53	18,08
10	5,62	17,79
Rata-rata	5,81	17,24

Dari hasil pengujian dapat diketahui bahwa Robot Berkaki Empat mampu berjalan mundur dengan kecepatan rata-rata 17,24 cm/detik. Kesimpulan yang dapat diambil adalah bahwa Robot Berkaki Empat mampu melakukan gerak mundur dengan kecepatan rata-rata yang hampir sama dengan gerak majunya, hanya arahnya saja yang berlawanan.

5.5.4 Pengujian Gerak Hadap Kanan

Pengujian gerak hadap kanan bertujuan untuk mengetahui kemampuan robot dalam melakukan gerak hadap kanan. Pengujian dilakukan dengan memberikan perintah kepada robot untuk memutar tubuhnya sebesar 90° ke kanan, lalu dicatat

waktunya dan diukur penyimpangan sudutnya. Hasil pengujian untuk gerak hadap kanan pada Robot Berkaki Empat ditunjukkan dalam Tabel 5.11.

Tabel 5. 11. Hasil Pengujian Gerak Hadap Kanan pada Robot Berkaki Empat

Pengujian Ke.	Waktu yang ditempuh (dalam detik)	Sudut yang dibentuk (dalam derajat)	Penyimpangan Sudut (dalam derajat)	Kesalahan (dalam %)
1	2,64	98	+8	8,89
2	2,96	98	+8	8,89
3	2,57	97	+7	7,78
4	2,47	94,5	+4,5	5
5	2,49	94,5	+4,5	5
6	2,46	96	+6	6,67
7	2,65	96	+6	6,67
8	2,47	96,5	+6,5	7,22
9	2,54	96	+6	6,67
10	2,47	96	+6	6,67
Rata-rata	2,57	96,25	+6,25	6,95

Persentase kesalahan pada Tabel 5.11 dihitung menggunakan rumus:

$$\text{Persentase kesalahan}(\%) = \frac{|\text{Penyimpangan Sudut}|}{90^\circ} \times 100\% \quad (5.4)$$

Dari hasil pengujian gerak hadap kanan diperoleh rata-rata waktu yang dibutuhkan Robot Berkaki Empat dalam melakukan gerak hadap kanan adalah 2,57 detik dengan penyimpangan rata-rata sebesar 6,25° melebihi sudut 90°. Kesimpulan yang dapat diambil adalah bahwa Robot Berkaki Empat mampu melakukan gerak hadap kanan dengan baik dengan kesalahan rata-rata sebesar 6,95 %.

5.5.5 Pengujian Gerak Hadap Kiri

Pengujian gerak hadap kiri bertujuan untuk mengetahui kemampuan robot dalam melakukan gerak hadap kiri. Pengujian dilakukan dengan memberikan perintah kepada robot untuk memutar tubuhnya sebesar 90° ke kiri, lalu dicatat waktunya dan diukur penyimpangan sudutnya. Hasil pengujian untuk gerak hadap kiri pada Robot Berkaki Empat ditunjukkan dalam Tabel 5.12.

Tabel 5. 12. Hasil Pengujian Gerak Hadap Kiri pada Robot Berkaki Empat

Pengujian Ke.	Waktu yang ditempuh (dalam detik)	Sudut yang dibentuk (dalam derajat)	Penyimpangan Sudut (dalam derajat)	Kesalahan (dalam %)
1	2,57	89	-1	1,11
2	2,78	87	-3	3,33
3	2,56	90,5	+0,5	0,55
4	2,67	90	0	0
5	2,81	91,5	+1,5	1,67
6	2,66	90	0	0
7	2,68	86	-4	4,44
8	2,79	85	-5	5,55
9	2,71	85	-5	5,55
10	2,66	85	-5	5,55
Rata-rata	2,69	87,9	-2,1	2,77

Persentase kesalahan pada Tabel 5.12 dihitung menggunakan rumus:

$$\text{Persentase kesalahan}(\%) = \frac{|\text{Penyimpangan Sudut}|}{90^\circ} \times 100\% \quad (5.5)$$

Dari hasil pengujian gerak hadap kiri diperoleh rata-rata waktu yang dibutuhkan Robot Berkaki Empat dalam melakukan gerak hadap kiri adalah 2,69 detik dengan penyimpangan rata-rata sebesar 2,1° kurang dari 90°. Kesimpulan yang dapat diambil adalah bahwa Robot Berkaki Empat mampu melakukan gerak hadap kiri dengan baik dengan kesalahan rata-rata sebesar 2,77 %.

5.5.6 Pengujian Gerak Putar 180°

Pengujian gerak putar 180° bertujuan untuk mengetahui kemampuan robot untuk berputar sebesar 180° dari posisi awal, atau dengan kata lain arah hadap robot akan berkebalikan dari arah hadap awal. Untuk dapat berputar 180°, dapat ditempuh dengan dua cara, yaitu dengan menggunakan metode hadap kanan atau metode hadap kiri, hanya besar sudutnya saja yang diubah dari 90° menjadi 180°. Pada pengujian ini metode yang dipakai adalah metode hadap kiri, dengan mengatur robot untuk berputar sebesar 180° ke arah kiri. Percobaan dilakukan sebanyak 10 kali, dimana pada setiap percobaan dicatat waktu tempuh dan penyimpangan sudutnya. Hasil pengujian untuk gerak putar 180° ditunjukkan dalam Tabel 5.13.

Tabel 5. 13. Hasil Pengujian Gerak Putar 180° pada Robot Berkaki Empat

Pengujian Ke.	Waktu yang ditempuh (dalam detik)	Sudut yang dibentuk (dalam derajat)	Penyimpangan Sudut (dalam derajat)	Kesalahan (dalam %)
1	5,06	179	-1	0,55
2	5,08	179	-1	0,55
3	5,09	179	-1	0,55
4	4,94	177,5	-2,5	1,39
5	5,00	177	-3	1,67
6	5,08	177	-3	1,67
7	5,12	176	-4	2,22
8	5,03	176	-4	2,22
9	4,99	178	-2	1,11
10	5,05	176,5	-3,5	1,94
Rata-rata	5,04	177,5	-2,5	1,39

Persentase kesalahan pada Tabel 5.13 dihitung menggunakan rumus:

$$\text{Persentase kesalahan}(\%) = \frac{|\text{Penyimpangan Sudut}|}{180^\circ} \times 100\% \quad (5.6)$$

Dari hasil pengujian gerak putar 180 diperoleh rata-rata waktu yang dibutuhkan robot berkaki empat dalam melakukan gerak putar 180° adalah 5,04 detik dengan penyimpangan rata-rata sebesar 2,5° kurang dari sudut 180°. Kesimpulan yang dapat diambil adalah bahwa Robot Berkaki Empat mampu melakukan gerak putar 180° dengan baik dengan kesalahan rata-rata sebesar 1,39 %.

5.5.7 Pengujian Gerak Gabungan

Pengujian gerak gabungan pada Robot Berkaki Empat bertujuan untuk mengetahui kemampuan robot dalam melakukan berbagai macam gerakan secara bergantian. Pengujian ini diperlukan karena pada penerapannya robot tidak akan hanya melakukan satu jenis gerakan saja, tapi juga akan melakukan berbagai macam variasi gerakan dalam suatu waktu. Pengujian dilakukan dengan memprogram robot untuk melakukan variasi gerakan secara bergantian dengan urutan seperti ditunjukkan dalam Gambar 5.41.



Gambar 5. 41. Urutan Gerakan Gabungan Robot Berkaki Empat

Hasil pengujian gerak gabungan pada Robot Berkaki Empat ditunjukkan dalam Tabel 5.14.

Tabel 5. 14. Hasil Pengujian Gerak Gabungan pada Robot Berkaki Empat

Pengujian ke.	Waktu yang ditempuh (dalam detik)
1	20,36
2	20,10
3	20,27
4	20,33
5	20,53
6	20,50
7	20,56
8	20,66
9	20,46
10	20,32
Rata-rata	20,41

Dari Tabel 5.14 dapat diketahui bahwa rata-rata waktu yang dibutuhkan robot untuk melakukan gerak gabungan dengan urutan seperti ditunjukkan dalam Gambar 5.40 adalah 20,41 detik. Kesimpulan yang dapat diambil adalah bahwa Robot Berkaki Empat mampu melakukan variasi gerakan yang dilakukan secara berurutan dengan baik.

BAB VI

KESIMPULAN DAN SARAN

6.1 Kesimpulan

Dari hasil perancangan dan pengujian yang telah dilakukan, dapat disimpulkan beberapa hal sebagai berikut.

- 1). Sistem mekanik Robot Berkaki Empat dibangun menggunakan 8 buah motor DC servo yang digunakan sebagai pemodelan sendi pinggang dan sendi lutut pada masing-masing kaki.
- 2). Metode pergerakan Robot Berkaki Empat terdiri dari gerak maju, gerak mundur, gerak hadap kanan, gerak hadap kiri, dan gerak putar 180° , dengan kecepatan rata-rata gerak maju pelan 6,98 cm/detik, kecepatan rata-rata gerak maju sedang 14,15 cm/detik, kecepatan rata-rata gerak maju cepat 17,30 cm/detik, kecepatan rata-rata gerak mundur cepat 17,24 cm/detik, waktu yang dibutuhkan untuk gerak hadap kanan 2,57 detik dengan penyimpangan sudut rata-rata sebesar $+6,25^\circ$, waktu untuk hadap kiri 2,69 detik dengan penyimpangan sudut rata-rata $-2,1^\circ$, dan waktu untuk berputar 180° 5,04 detik dengan penyimpangan rata-rata $-2,5^\circ$.
- 3). Sistem elektronika Robot Berkaki Empat terdiri dari rangkaian mikrokontroler kontrol servo dan rangkaian catu daya, dimana rangkaian catu daya yang digunakan adalah rangkaian *High Currnt Voltage Regulator* yang berfungsi untuk menunjang penggunaan 8 buah motor DC servo.
- 4). Robot Berkaki Empat menggunakan mikrokontroler ATmega8 sebagai pengendali utama motor DC servo dengan metode kontrol multi servo.

6.2 Saran

Beberapa hal yang direkomendasikan untuk pengembangan Robot Berkaki Empat lebih lanjut adalah:

- 1). Disarankan untuk menambah jumlah derajat kebebasan pada tiap kaki robot menjadi 3 derajat kebebasan, karena dapat meningkatkan fleksibilitas robot saat bergerak dan memperluas daerah kerja kaki robot.
- 2). Disarankan untuk membuat desain mekanik robot yang lebih ringan agar kerja motor DC servo lebih optimal.

DAFTAR PUSTAKA

- Appin Knowledge Solutions. 2007. *Robotics*. New Delhi : Infinity Science Press LLC.
- Atmel. 2007. *8-bit AVR with 8K Bytes In-System Programmable Flash Atmega8, Atmega8L*. San Jose: Atmel.
- Braunl, Thomas. 2006. *Embedded Robotics Third Edition*. Jerman: Springer.
- DIKTI. 2009. *Panduan Kontes Robot Cerdas Indonesia 2010*. Jakarta: DIKTI.
- Firoozian, Riazollah. 2009. *Servo Motors and Industrial Control Theory*. USA: Springer.
- Gonzalez de Santos, P. 2006. *Quadrupedal Locomotion*. Spanyol: Springer.
- Kordic, Vedran, Aleksandar Lazinica, dan Munir Merdan. 2005. *Cutting Edge Robotics*. Jerman : pIV pro literatur Verlag Robert Mayer-Scholz.
- Nakamura, Masatoshi. 2004. *Mechatronic Servo System Control*. Jerman: Springer.
- Parallax. 2010. *Parallax Standard Servo*. California: Parallax.
- Pitowarno, Endra. 2006. *Robotika: Desain, Kontrol dan Kecerdasan Buatan*. Yogyakarta: Penerbit ANDI.
- Resh, Vincent H. 2009. *Encyclopedia of Insects*. USA: Academic Press.

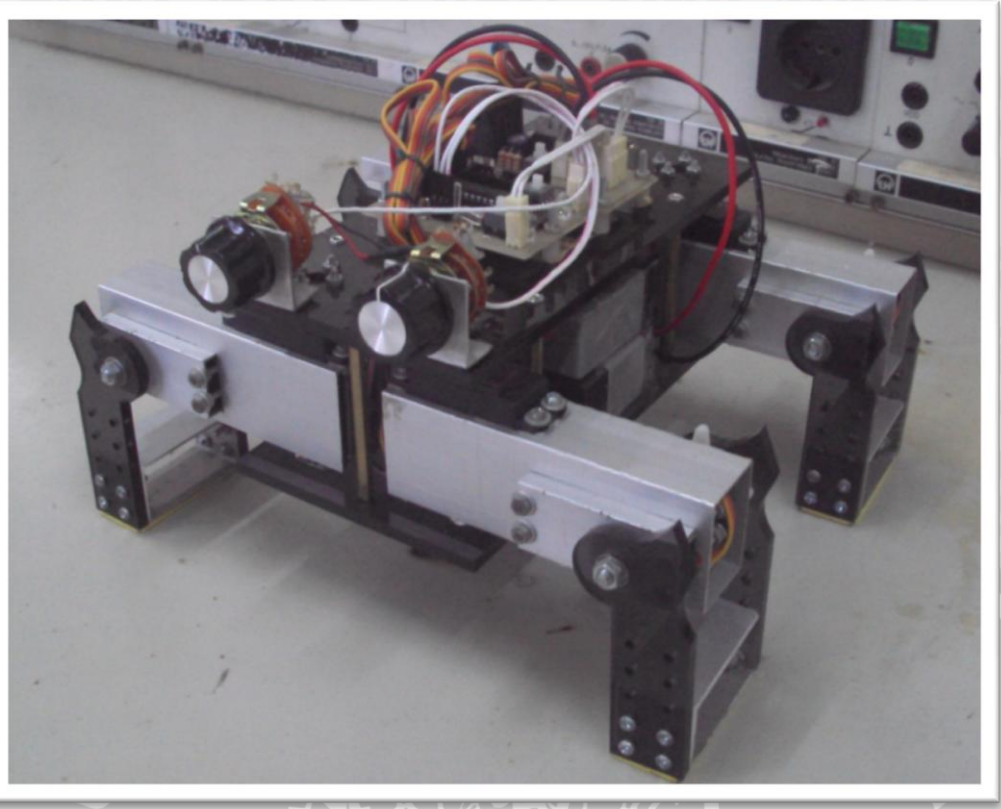
LAMPIRAN



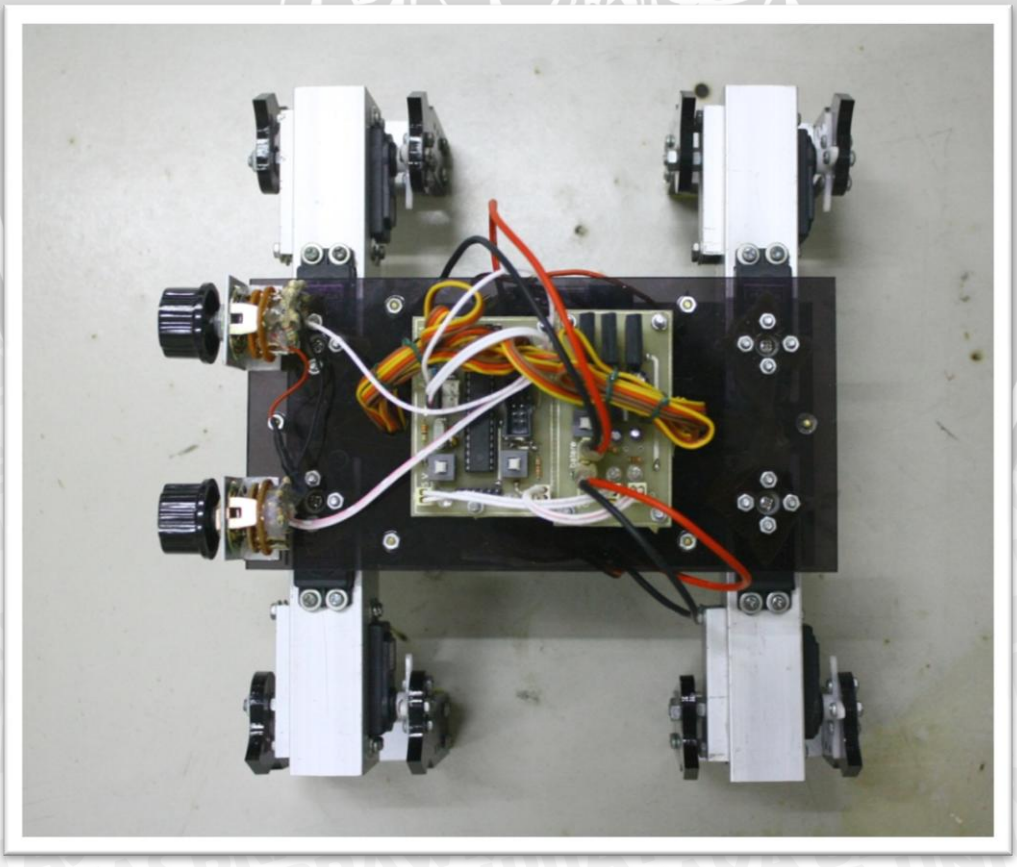
LAMPIRAN I

FOTO ALAT

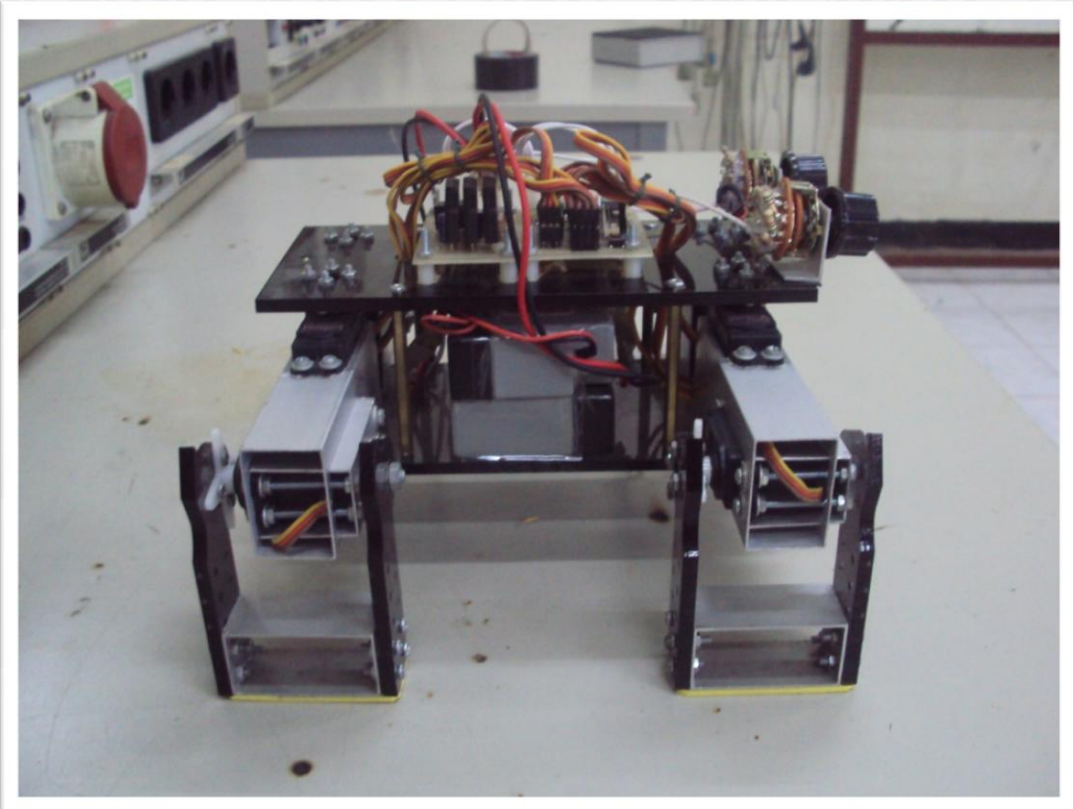




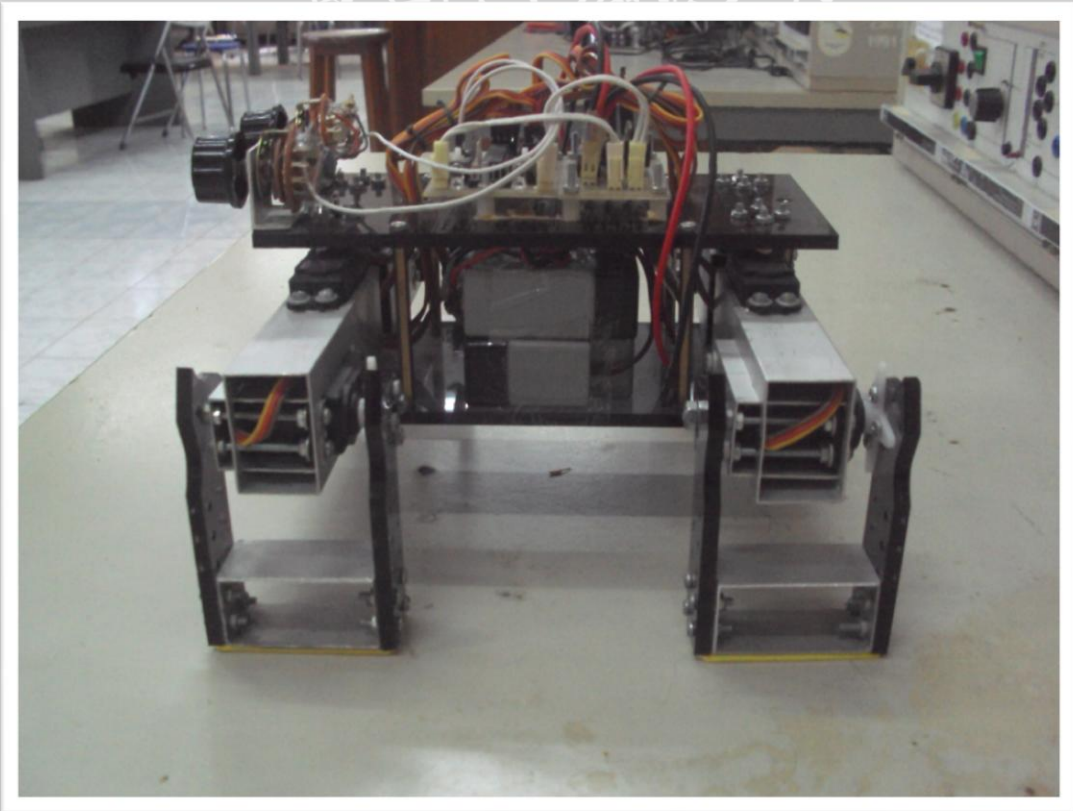
Gambar 1. Perspektif Robot Berkaki Empat



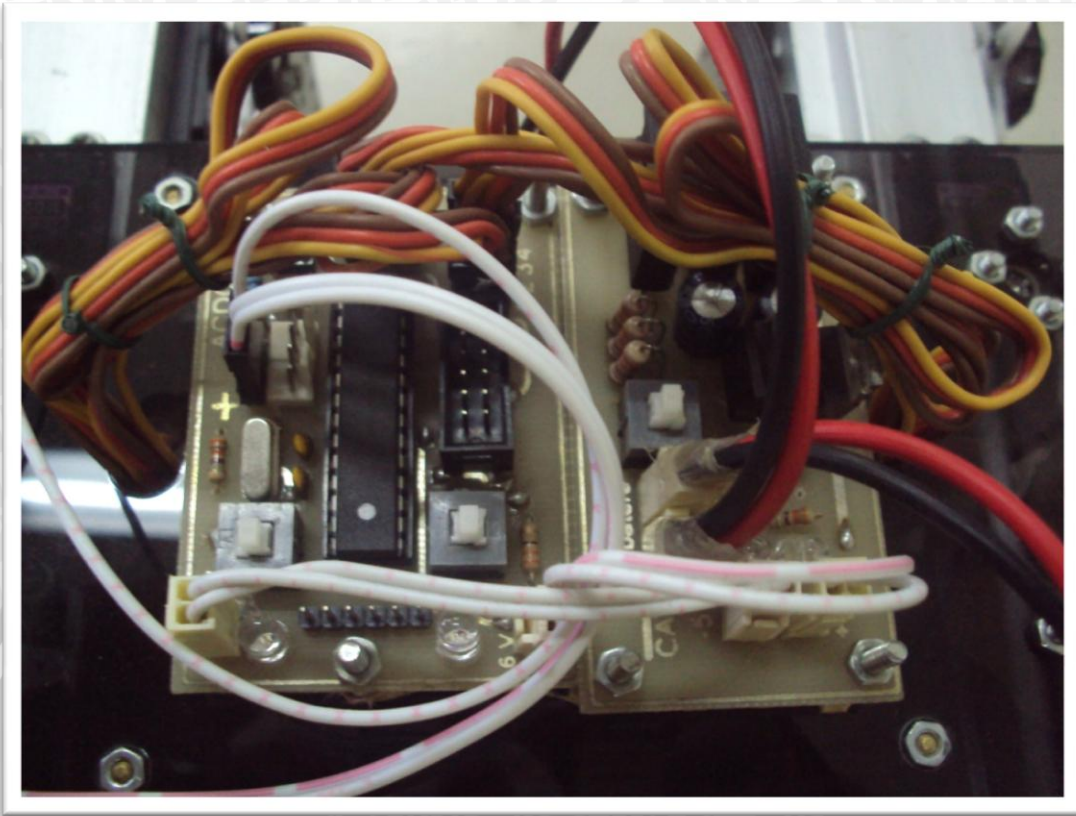
Gambar 2. Robot Berkaki Empat Tampak Atas



Gambar 3. Robot Berkaki Empat Tampak Samping Kanan



Gambar 4. Robot Berkaki Empat Tampak Samping Kiri



Gambar 5. Sistem Elektronika Robot Berkaki Empat



LAMPIRAN II

LISTING PROGRAM MIKROKONTROLER ATMEGA8




```

/*****
Project : ROBOT BERKAKI EMPAT
Version : 09.5.90
Date : 22/12/2010
Author : INDRAWAN RIZA F.
Company : TEUB
Comments:

```

```

Chip type : ATmega8
Program type : Application
Clock frequency : 16,000000 MHz
Memory model : Small
External SRAM size : 0
Data Stack size : 256
*****/

```

```

#include <mega8.h>
#include <delay.h>
#include <stdio.h>
#define ADC_VREF_TYPE 0x20

```

```

#define CH_1 PORTC.1
#define CH_2 PORTC.2
#define CH_3 PORTC.3
#define CH_4 PORTC.4
#define CH_5 PORTC.5
#define CH_6 PORTD.4
#define CH_7 PORTD.3
#define CH_8 PORTD.2

```

```

unsigned char index_servo=0,index_bantu=3;
unsigned int lama_waktu[8] = {
    2630,3100,2860,2970,
    3040,2790,2930,3250
};

```

```

void check_rot_switch();
void dasar();
void putar_kiri();
void putar_kanan();
void maju_pelan();
void maju_sedang();
void maju_cepat();
void mundur_cepat();
void putar_belakang();
void gabungan();

```

```

int i=0;
int p=0;
int r=0;
int j=0;
int k=0;
int l=0;
int m=0;

```

```

// Timer 1 overflow interrupt service routine
interrupt [TIM1_OVF] void timer1_ovf_isr(void) // 5ms
{
    if(++index_servo>=4){index_servo=0;}
    switch(index_servo)
    {
        case 0 : CH_1=CH_5=1; break;
        case 1 : CH_2=CH_6=1; break;
        case 2 : CH_3=CH_7=1; break;
        case 3 : CH_4=CH_8=1; break;
    }
}

```

```

if(index_servo>=3){index_bantu=0;}
else{index_bantu++;}
OCR1A = lama_waktu[index_bantu];
OCR1B = lama_waktu[index_bantu+4];
}

```

```

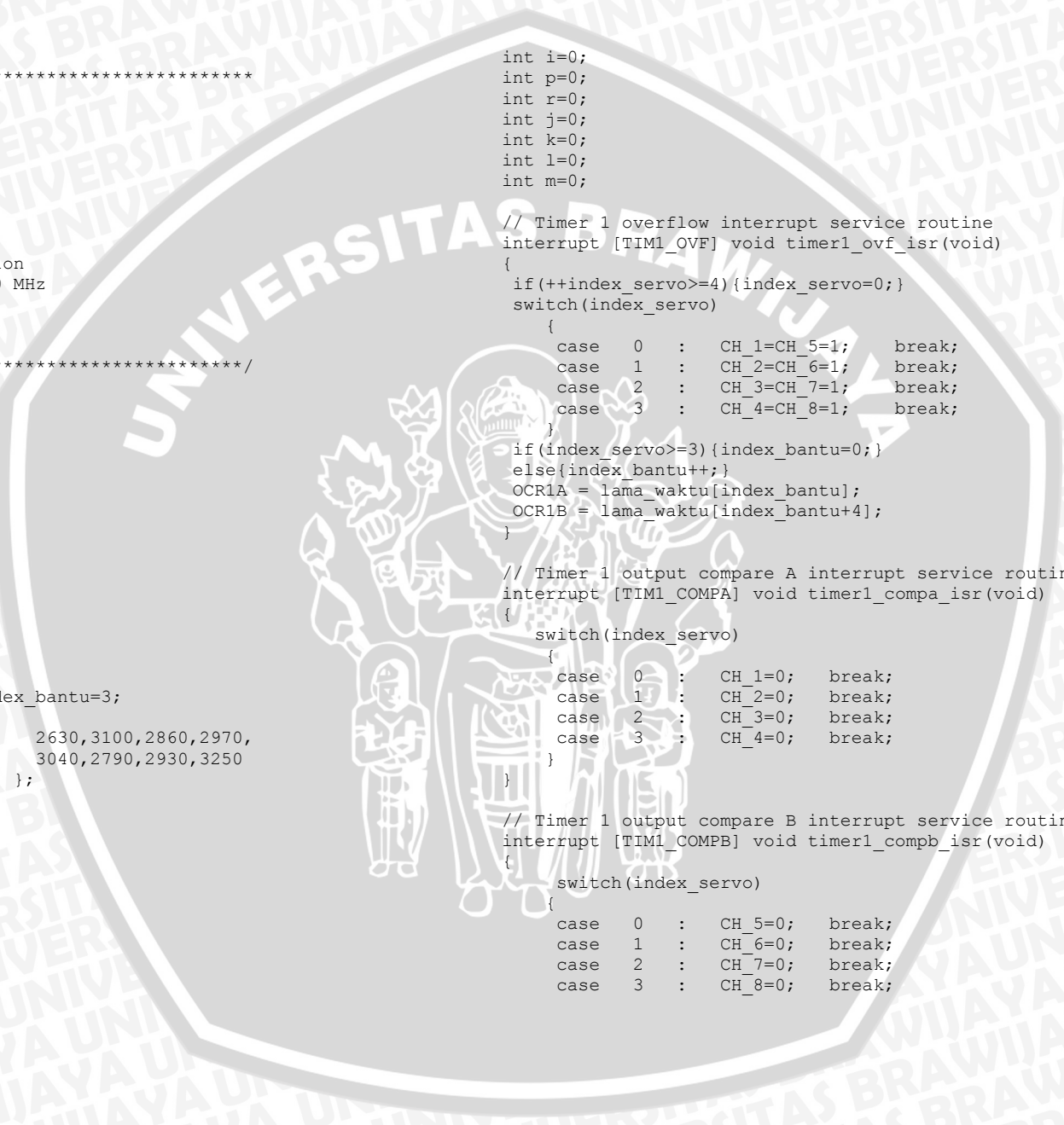
// Timer 1 output compare A interrupt service routine
interrupt [TIM1_COMPA] void timer1_compa_isr(void)
{
    switch(index_servo)
    {
        case 0 : CH_1=0; break;
        case 1 : CH_2=0; break;
        case 2 : CH_3=0; break;
        case 3 : CH_4=0; break;
    }
}

```

```

// Timer 1 output compare B interrupt service routine
interrupt [TIM1_COMPB] void timer1_compb_isr(void)
{
    switch(index_servo)
    {
        case 0 : CH_5=0; break;
        case 1 : CH_6=0; break;
        case 2 : CH_7=0; break;
        case 3 : CH_8=0; break;
    }
}

```



```
}
}

unsigned char read_adc(unsigned char adc_input)
{
    ADMUX=adc_input | (ADC_VREF_TYPE & 0xff);
    delay_us(10);
    ADCSRA|=0x40;
    while ((ADCSRA & 0x10)==0);
    ADCSRA|=0x10;
    return ADCH;
}

void main(void)
{
    // Port B initialization
    // Func7=In Func6=In Func5=In Func4=Out Func3=In Func2=In Func1=In
    Func0=In
    // State7=T State6=T State5=T State4=0 State3=T State2=T State1=T
    State0=T
    PORTB=0x00;
    DDRB=0x10;

    // Port C initialization
    // Func6=In Func5=Out Func4=Out Func3=Out Func2=Out Func1=Out
    Func0=In
    // State6=T State5=0 State4=0 State3=0 State2=0 State1=0 State0=T
    PORTC=0x00;
    DDRC=0x3E;

    // Port D initialization
    // Func7=In Func6=In Func5=In Func4=Out Func3=Out Func2=Out Func1=In
    Func0=In
    // State7=T State6=T State5=T State4=0 State3=0 State2=0 State1=T
    State0=T
    PORTD=0x00;
    DDRD=0x1C;

    // Timer/Counter 0 initialization
    // Clock source: System Clock
    // Clock value: Timer 0 Stopped
    TCCR0=0x00;
    TCNT0=0x00;

    // Timer/Counter 1 initialization
    // Clock source: System Clock

    // Clock value: 2000,000 kHz
    // Mode: Fast PWM top=ICR1
    // OC1A output: Discon.
    // OC1B output: Discon.
    // Noise Canceler: Off
    // Input Capture on Falling Edge
    // Timer 1 Overflow Interrupt: On
    // Input Capture Interrupt: Off
    // Compare A Match Interrupt: On
    // Compare B Match Interrupt: On
    TCCR1A=0x02;
    TCCR1B=0x1A;
    TCNT1H=0x00;
    TCNT1L=0x00;
    ICR1H=0x27;
    ICR1L=0x10;
    OCR1AH=0x07;
    OCR1AL=0xD0;
    OCR1BH=0x0F;
    OCR1BL=0xA0;

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

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

    // Timer(s)/Counter(s) Interrupt(s) initialization
    TIMSK=0x1C;

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

    // ADC initialization
}
```

```
// ADC Clock frequency: 1000,000 kHz
// ADC Voltage Reference: AREF pin
// Only the 8 most significant bits of
// the AD conversion result are used
ADMUX=ADC_VREF_TYPE & 0xff;
ADCSRA=0x84;
```

```
// Global enable interrupts
#asm("sei")

while (1)
{
    check_rot_switch();
};
}
```

```
///// MODE ROTARY SWITCH /////
```

```
void check_rot_switch()
{
    unsigned char data_adc;
    data_adc=(int) read_adc(0);
    data_adc/= 22;
```

```
switch(data_adc) {
    case 1 : dasar();
            break;
    case 2 : putar_kiri();
            break;
    case 3 : putar_kanan();
            break;
    case 4 : maju_pelan();
            break;
    case 5 : maju_sedang();
            break;
    case 6 : maju_cepat();
            break;
    case 7 : mundur_cepat();
            break;
    case 8 : putar_belakang();
            break;
    case 9 : gabungan();
            break;
    case 10 : break;
    case 11 : break;
```

```
}
}
///// PROGRAM GERAK /////
```

```
/// POSISI DASAR ///
```

```
void dasar(){
    //center
    lama_waktu[0]=2630;
    lama_waktu[1]=3100;
    lama_waktu[2]=2860;
    lama_waktu[3]=2970;
    lama_waktu[4]=3040;
    lama_waktu[5]=2790;
    lama_waktu[6]=2930;
    lama_waktu[7]=3250;
```

```
    delay_ms(200);
```

```
/// PUTAR KIRI ///
```

```
void putar_kiri(){
    //center
    lama_waktu[0]=2630;
    lama_waktu[1]=3100;
    lama_waktu[2]=2860;
    lama_waktu[3]=2970;
    lama_waktu[4]=3040;
    lama_waktu[5]=2790;
    lama_waktu[6]=2930;
    lama_waktu[7]=3250;
```

```
    delay_ms(200);
```

```
while (p<3)
```

```
{
    //gait1
```

```
    lama_waktu[0]=3330;
    lama_waktu[1]=2400;
    lama_waktu[2]=3560;
    lama_waktu[3]=2270;
    lama_waktu[4]=3040;
    lama_waktu[5]=3390;
    lama_waktu[6]=2930;
    lama_waktu[7]=2650;
```



```

delay_ms(200);

//gait2
lama_waktu[0]=3330;
lama_waktu[1]=2400;
lama_waktu[2]=3560;
lama_waktu[3]=2270;
lama_waktu[4]=3040;
lama_waktu[5]=2790;
lama_waktu[6]=2930;
lama_waktu[7]=3250;

delay_ms(200);

//gait3
lama_waktu[0]=2630;
lama_waktu[1]=3100;
lama_waktu[2]=2860;
lama_waktu[3]=2970;
lama_waktu[4]=2440;
lama_waktu[5]=2790;
lama_waktu[6]=3530;
lama_waktu[7]=3250;

delay_ms(200);

//gait4
lama_waktu[0]=2630;
lama_waktu[1]=3100;
lama_waktu[2]=2860;
lama_waktu[3]=2970;
lama_waktu[4]=3040;
lama_waktu[5]=2790;
lama_waktu[6]=2930;
lama_waktu[7]=3250;

delay_ms(200);

p++;
}
delay_ms(200);
//stop
lama_waktu[0]=2630;
lama_waktu[1]=3100;

lama_waktu[2]=2860;
lama_waktu[3]=2970;
lama_waktu[4]=3040;
lama_waktu[5]=2790;
lama_waktu[6]=2930;
lama_waktu[7]=3250;

delay_ms(200);

}

/// PUTAR KANAN ///
void putar_kanan() {
    //center
    lama_waktu[0]=2630;
    lama_waktu[1]=3100;
    lama_waktu[2]=2860;
    lama_waktu[3]=2970;
    lama_waktu[4]=3040;
    lama_waktu[5]=2790;
    lama_waktu[6]=2930;
    lama_waktu[7]=3250;

    delay_ms(200);

    while (r<3)
    {
        //gait1
        lama_waktu[0]=3330;
        lama_waktu[1]=2400;
        lama_waktu[2]=3560;
        lama_waktu[3]=2270;
        lama_waktu[4]=2440;
        lama_waktu[5]=2790;
        lama_waktu[6]=3530;
        lama_waktu[7]=3250;

        delay_ms(200);

        //gait2
        lama_waktu[0]=3330;
        lama_waktu[1]=2400;
        lama_waktu[2]=3560;
        lama_waktu[3]=2270;
        lama_waktu[4]=3040;
        lama_waktu[5]=2790;
        lama_waktu[6]=2930;
    }
}

```



```

lama_waktu[7]=3250;
delay_ms(200);

//gait3
lama_waktu[0]=2630;
lama_waktu[1]=3100;
lama_waktu[2]=2860;
lama_waktu[3]=2970;
lama_waktu[4]=3040;
lama_waktu[5]=3390;
lama_waktu[6]=2930;
lama_waktu[7]=2650;

delay_ms(200);

//gait4
lama_waktu[0]=2630;
lama_waktu[1]=3100;
lama_waktu[2]=2860;
lama_waktu[3]=2970;
lama_waktu[4]=3040;
lama_waktu[5]=2790;
lama_waktu[6]=2930;
lama_waktu[7]=3250;

delay_ms(200);

r++;
}
delay_ms(200);
//stop
lama_waktu[0]=2630;
lama_waktu[1]=3100;
lama_waktu[2]=2860;
lama_waktu[3]=2970;
lama_waktu[4]=3040;
lama_waktu[5]=2790;
lama_waktu[6]=2930;
lama_waktu[7]=3250;

delay_ms(200);
}

/// MAJU CEPAT ///

void maju_cepat(){
//center
lama_waktu[0]=2630;
lama_waktu[1]=3100;
lama_waktu[2]=2860;
lama_waktu[3]=2970;
lama_waktu[4]=3040;
lama_waktu[5]=2790;
lama_waktu[6]=2930;
lama_waktu[7]=3250;

delay_ms(200);

while (i<10)
{
//gait1
lama_waktu[0]=3330;
lama_waktu[1]=3800;
lama_waktu[2]=2160;
lama_waktu[3]=2270;
lama_waktu[4]=2440;
lama_waktu[5]=2790;
lama_waktu[6]=3530;
lama_waktu[7]=3250;

delay_ms(200);

//gait2
lama_waktu[0]=3330;
lama_waktu[1]=3800;
lama_waktu[2]=2160;
lama_waktu[3]=2270;
lama_waktu[4]=3040;
lama_waktu[5]=2790;
lama_waktu[6]=2930;
lama_waktu[7]=3250;

delay_ms(200);

//gait3
lama_waktu[0]=1930;
lama_waktu[1]=2400;
lama_waktu[2]=3560;
lama_waktu[3]=3670;
lama_waktu[4]=3040;
lama_waktu[5]=3390;

```

UNIVERSITAS BRAWIJAYA



```
lama_waktu[6]=2930;
lama_waktu[7]=2650;
```

```
delay_ms(200);
```

```
//gait4
lama_waktu[0]=1930;
lama_waktu[1]=2400;
lama_waktu[2]=3560;
lama_waktu[3]=3670;
lama_waktu[4]=3040;
lama_waktu[5]=2790;
lama_waktu[6]=2930;
lama_waktu[7]=3250;
```

```
delay_ms(200);
i++;
```

```
}
```

```
//gait5
lama_waktu[0]=2630;
lama_waktu[1]=3100;
lama_waktu[2]=2860;
lama_waktu[3]=2970;
lama_waktu[4]=2440;
lama_waktu[5]=2790;
lama_waktu[6]=3530;
lama_waktu[7]=3250;
```

```
delay_ms(200);
//stop
lama_waktu[0]=2630;
lama_waktu[1]=3100;
lama_waktu[2]=2860;
lama_waktu[3]=2970;
lama_waktu[4]=3040;
lama_waktu[5]=2790;
lama_waktu[6]=2930;
lama_waktu[7]=3250;
```

```
delay_ms(200);
while(1){};
```

```
}
```

```
/// MAJU SEDANG ///
```

```
void maju_sedang() {
```

```
//center
```

```
lama_waktu[0]=2630;
lama_waktu[1]=3100;
lama_waktu[2]=2860;
lama_waktu[3]=2970;
lama_waktu[4]=3040;
lama_waktu[5]=2790;
lama_waktu[6]=2930;
lama_waktu[7]=3250;
```

```
delay_ms(200);
```

```
while (j<15)
```

```
{
//gait1
```

```
lama_waktu[0]=3097;
lama_waktu[1]=3567;
lama_waktu[2]=2393;
lama_waktu[3]=2503;
lama_waktu[4]=2440;
lama_waktu[5]=2790;
lama_waktu[6]=3530;
lama_waktu[7]=3250;
```

```
delay_ms(200);
```

```
//gait2
```

```
lama_waktu[0]=3097;
lama_waktu[1]=3567;
lama_waktu[2]=2393;
lama_waktu[3]=2503;
lama_waktu[4]=3040;
lama_waktu[5]=2790;
lama_waktu[6]=2930;
lama_waktu[7]=3250;
```

```
delay_ms(200);
```

```
//gait3
```

```
lama_waktu[0]=2163;
lama_waktu[1]=2633;
lama_waktu[2]=3327;
lama_waktu[3]=3437;
lama_waktu[4]=3040;
lama_waktu[5]=3390;
```



```

lama_waktu[6]=2930;
lama_waktu[7]=2650;

delay_ms(200);

//gait4
lama_waktu[0]=2163;
lama_waktu[1]=2633;
lama_waktu[2]=3327;
lama_waktu[3]=3437;
lama_waktu[4]=3040;
lama_waktu[5]=2790;
lama_waktu[6]=2930;
lama_waktu[7]=3250;

delay_ms(200);
j++;
}

//gait5
lama_waktu[0]=2630;
lama_waktu[1]=3100;
lama_waktu[2]=2860;
lama_waktu[3]=2970;
lama_waktu[4]=2440;
lama_waktu[5]=2790;
lama_waktu[6]=3530;
lama_waktu[7]=3250;

delay_ms(200);

//stop
lama_waktu[0]=2630;
lama_waktu[1]=3100;
lama_waktu[2]=2860;
lama_waktu[3]=2970;
lama_waktu[4]=3040;
lama_waktu[5]=2790;
lama_waktu[6]=2930;
lama_waktu[7]=3250;

delay_ms(200);
while(1){};
}

// MAJU PELAN //
void maju_pelan(){
//center
lama_waktu[0]=2630;
lama_waktu[1]=3100;
lama_waktu[2]=2860;
lama_waktu[3]=2970;
lama_waktu[4]=3040;
lama_waktu[5]=2790;
lama_waktu[6]=2930;
lama_waktu[7]=3250;

delay_ms(200);

while (k<23)
{
//gait1
lama_waktu[0]=2863;
lama_waktu[1]=3333;
lama_waktu[2]=2627;
lama_waktu[3]=2737;
lama_waktu[4]=2440;
lama_waktu[5]=2790;
lama_waktu[6]=3530;
lama_waktu[7]=3250;

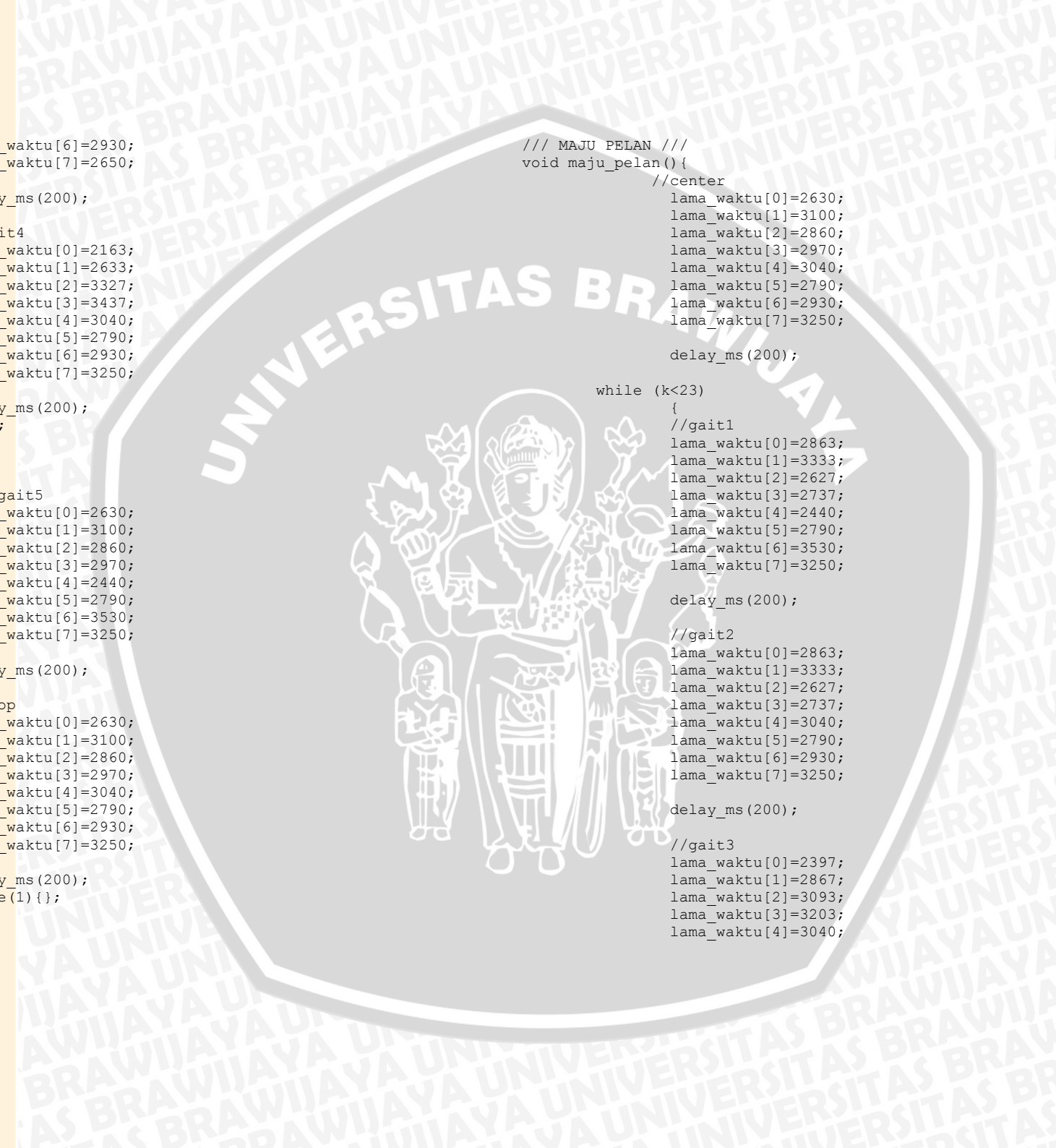
delay_ms(200);

//gait2
lama_waktu[0]=2863;
lama_waktu[1]=3333;
lama_waktu[2]=2627;
lama_waktu[3]=2737;
lama_waktu[4]=3040;
lama_waktu[5]=2790;
lama_waktu[6]=2930;
lama_waktu[7]=3250;

delay_ms(200);

//gait3
lama_waktu[0]=2397;
lama_waktu[1]=2867;
lama_waktu[2]=3093;
lama_waktu[3]=3203;
lama_waktu[4]=3040;

```



```

lama_waktu[5]=3390;
lama_waktu[6]=2930;
lama_waktu[7]=2650;

delay_ms(200);

//gait4
lama_waktu[0]=2397;
lama_waktu[1]=2867;
lama_waktu[2]=3093;
lama_waktu[3]=3203;
lama_waktu[4]=3040;
lama_waktu[5]=2790;
lama_waktu[6]=2930;
lama_waktu[7]=3250;

delay_ms(200);
k++;
};

//gait5
lama_waktu[0]=2630;
lama_waktu[1]=3100;
lama_waktu[2]=2860;
lama_waktu[3]=2970;
lama_waktu[4]=2440;
lama_waktu[5]=2790;
lama_waktu[6]=3530;
lama_waktu[7]=3250;

delay_ms(200);

//stop
lama_waktu[0]=2630;
lama_waktu[1]=3100;
lama_waktu[2]=2860;
lama_waktu[3]=2970;
lama_waktu[4]=3040;
lama_waktu[5]=2790;
lama_waktu[6]=2930;
lama_waktu[7]=3250;

delay_ms(200);
while(1){};
}

// MUNDUR ///
void mundur_cepat(){
    //center
    lama_waktu[0]=2630;
    lama_waktu[1]=3100;
    lama_waktu[2]=2860;
    lama_waktu[3]=2970;
    lama_waktu[4]=3040;
    lama_waktu[5]=2790;
    lama_waktu[6]=2930;
    lama_waktu[7]=3250;

    delay_ms(200);

    while (l<10)
    {
        //gait1
        lama_waktu[0]=3330;
        lama_waktu[1]=3800;
        lama_waktu[2]=2160;
        lama_waktu[3]=2270;
        lama_waktu[4]=3040;
        lama_waktu[5]=3390;
        lama_waktu[6]=2930;
        lama_waktu[7]=2650;

        delay_ms(200);

        //gait2
        lama_waktu[0]=3330;
        lama_waktu[1]=3800;
        lama_waktu[2]=2160;
        lama_waktu[3]=2270;
        lama_waktu[4]=3040;
        lama_waktu[5]=2790;
        lama_waktu[6]=2930;
        lama_waktu[7]=3250;

        delay_ms(200);

        //gait3
        lama_waktu[0]=1930;
        lama_waktu[1]=2400;
        lama_waktu[2]=3560;
        lama_waktu[3]=3670;

```

UNIVERSITAS BRAWIJAYA




```

lama_waktu[4]=2440;
lama_waktu[5]=2790;
lama_waktu[6]=3530;
lama_waktu[7]=3250;

delay_ms(200);

//gait4
lama_waktu[0]=1930;
lama_waktu[1]=2400;
lama_waktu[2]=3560;
lama_waktu[3]=3670;
lama_waktu[4]=3040;
lama_waktu[5]=2790;
lama_waktu[6]=2930;
lama_waktu[7]=3250;

delay_ms(200);
l++;
}

//gait5
lama_waktu[0]=2630;
lama_waktu[1]=3100;
lama_waktu[2]=2860;
lama_waktu[3]=2970;
lama_waktu[4]=3040;
lama_waktu[5]=3390;
lama_waktu[6]=2930;
lama_waktu[7]=2650;

delay_ms(200);
//stop
lama_waktu[0]=2630;
lama_waktu[1]=3100;
lama_waktu[2]=2860;
lama_waktu[3]=2970;
lama_waktu[4]=3040;
lama_waktu[5]=2790;
lama_waktu[6]=2930;
lama_waktu[7]=3250;

delay_ms(200);
while(1){};
}

}

// PUTAR BELAKANG //
void putar_belakang(){
//center
lama_waktu[0]=2630;
lama_waktu[1]=3100;
lama_waktu[2]=2860;
lama_waktu[3]=2970;
lama_waktu[4]=3040;
lama_waktu[5]=2790;
lama_waktu[6]=2930;
lama_waktu[7]=3250;

delay_ms(200);

while (m<6)
{
//gait1
lama_waktu[0]=3330;
lama_waktu[1]=2400;
lama_waktu[2]=3560;
lama_waktu[3]=2270;
lama_waktu[4]=3040;
lama_waktu[5]=3390;
lama_waktu[6]=2930;
lama_waktu[7]=2650;

delay_ms(200);

//gait2
lama_waktu[0]=3330;
lama_waktu[1]=2400;
lama_waktu[2]=3560;
lama_waktu[3]=2270;
lama_waktu[4]=3040;
lama_waktu[5]=2790;
lama_waktu[6]=2930;
lama_waktu[7]=3250;

delay_ms(200);

//gait3
lama_waktu[0]=2630;
lama_waktu[1]=3100;
lama_waktu[2]=2860;

```



```

lama_waktu[3]=2970;
lama_waktu[4]=2440;
lama_waktu[5]=2790;
lama_waktu[6]=3530;
lama_waktu[7]=3250;

```

```

delay_ms(200);

```

```

//gait4
lama_waktu[0]=2630;
lama_waktu[1]=3100;
lama_waktu[2]=2860;
lama_waktu[3]=2970;
lama_waktu[4]=3040;
lama_waktu[5]=2790;
lama_waktu[6]=2930;
lama_waktu[7]=3250;

```

```

delay_ms(200);

```

```

m++;

```

```

}
delay_ms(200);

```

```

//stop
lama_waktu[0]=2630;
lama_waktu[1]=3100;
lama_waktu[2]=2860;
lama_waktu[3]=2970;
lama_waktu[4]=3040;
lama_waktu[5]=2790;
lama_waktu[6]=2930;
lama_waktu[7]=3250;

```

```

delay_ms(200);

```

```

}

```

```

/// GERAK GABUNGAN ///
void gabungan(){
//maju5langkah

```

```

//center
lama_waktu[0]=2630;
lama_waktu[1]=3100;
lama_waktu[2]=2860;
lama_waktu[3]=2970;

```

```

lama_waktu[4]=3040;
lama_waktu[5]=2790;
lama_waktu[6]=2930;
lama_waktu[7]=3250;

```

```

delay_ms(200);

```

```

while (i<5)

```

```

{
//gait1
lama_waktu[0]=3330;
lama_waktu[1]=3800;
lama_waktu[2]=2160;
lama_waktu[3]=2270;
lama_waktu[4]=2440;
lama_waktu[5]=2790;
lama_waktu[6]=3530;
lama_waktu[7]=3250;

```

```

delay_ms(200);

```

```

//gait2

```

```

lama_waktu[0]=3330;
lama_waktu[1]=3800;
lama_waktu[2]=2160;
lama_waktu[3]=2270;
lama_waktu[4]=3040;
lama_waktu[5]=2790;
lama_waktu[6]=2930;
lama_waktu[7]=3250;

```

```

delay_ms(200);

```

```

//gait3

```

```

lama_waktu[0]=1930;
lama_waktu[1]=2400;
lama_waktu[2]=3560;
lama_waktu[3]=3670;
lama_waktu[4]=3040;
lama_waktu[5]=3390;
lama_waktu[6]=2930;
lama_waktu[7]=2650;

```

```

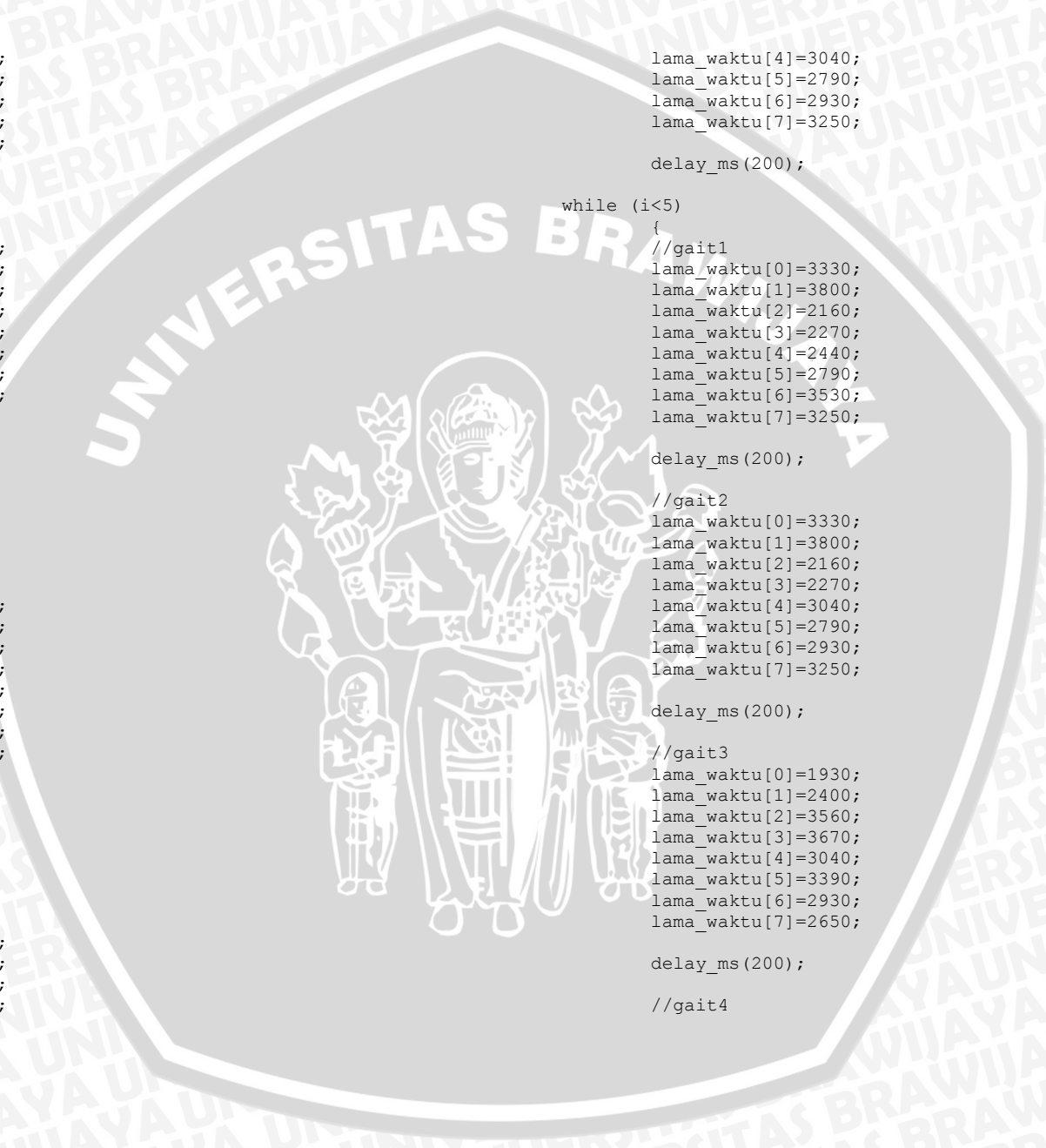
delay_ms(200);

```

```

//gait4

```



```

lama_waktu[0]=1930;
lama_waktu[1]=2400;
lama_waktu[2]=3560;
lama_waktu[3]=3670;
lama_waktu[4]=3040;
lama_waktu[5]=2790;
lama_waktu[6]=2930;
lama_waktu[7]=3250;

delay_ms(200);
i++;
}

//gait5
lama_waktu[0]=2630;
lama_waktu[1]=3100;
lama_waktu[2]=2860;
lama_waktu[3]=2970;
lama_waktu[4]=2440;
lama_waktu[5]=2790;
lama_waktu[6]=3530;
lama_waktu[7]=3250;

delay_ms(200);

//stop
lama_waktu[0]=2630;
lama_waktu[1]=3100;
lama_waktu[2]=2860;
lama_waktu[3]=2970;
lama_waktu[4]=3040;
lama_waktu[5]=2790;
lama_waktu[6]=2930;
lama_waktu[7]=3250;

delay_ms(200);

//kanan

//center
lama_waktu[0]=2630;
lama_waktu[1]=3100;
lama_waktu[2]=2860;
lama_waktu[3]=2970;
lama_waktu[4]=3040;
lama_waktu[5]=2790;

lama_waktu[6]=2930;
lama_waktu[7]=3250;

delay_ms(200);

while (r<3)
{
//gait1
lama_waktu[0]=3330;
lama_waktu[1]=2400;
lama_waktu[2]=3560;
lama_waktu[3]=2270;
lama_waktu[4]=2440;
lama_waktu[5]=2790;
lama_waktu[6]=3530;
lama_waktu[7]=3250;

delay_ms(200);

//gait2
lama_waktu[0]=3330;
lama_waktu[1]=2400;
lama_waktu[2]=3560;
lama_waktu[3]=2270;
lama_waktu[4]=3040;
lama_waktu[5]=2790;
lama_waktu[6]=2930;
lama_waktu[7]=3250;

delay_ms(200);

//gait3
lama_waktu[0]=2630;
lama_waktu[1]=3100;
lama_waktu[2]=2860;
lama_waktu[3]=2970;
lama_waktu[4]=3040;
lama_waktu[5]=3390;
lama_waktu[6]=2930;
lama_waktu[7]=2650;

delay_ms(200);

//gait4
lama_waktu[0]=2630;
lama_waktu[1]=3100;

```

UNIVERSITAS BRAWIJAYA



```

lama_waktu[2]=2860;
lama_waktu[3]=2970;
lama_waktu[4]=3040;
lama_waktu[5]=2790;
lama_waktu[6]=2930;
lama_waktu[7]=3250;

```

```

delay_ms(200);

```

```

r++;

```

```

}

```

```

delay_ms(200);

```

```

//stop

```

```

lama_waktu[0]=2630;
lama_waktu[1]=3100;
lama_waktu[2]=2860;
lama_waktu[3]=2970;
lama_waktu[4]=3040;
lama_waktu[5]=2790;
lama_waktu[6]=2930;
lama_waktu[7]=3250;

```

```

delay_ms(200);

```

```

//mundur5langkah

```

```

//center

```

```

lama_waktu[0]=2630;
lama_waktu[1]=3100;
lama_waktu[2]=2860;
lama_waktu[3]=2970;
lama_waktu[4]=3040;
lama_waktu[5]=2790;
lama_waktu[6]=2930;
lama_waktu[7]=3250;

```

```

delay_ms(200);

```

```

while (l<5)

```

```

{

```

```

//gait1

```

```

lama_waktu[0]=3330;
lama_waktu[1]=3800;
lama_waktu[2]=2160;
lama_waktu[3]=2270;
lama_waktu[4]=3040;

```

```

lama_waktu[5]=3390;
lama_waktu[6]=2930;
lama_waktu[7]=2650;

```

```

delay_ms(200);

```

```

//gait2

```

```

lama_waktu[0]=3330;
lama_waktu[1]=3800;
lama_waktu[2]=2160;
lama_waktu[3]=2270;
lama_waktu[4]=3040;
lama_waktu[5]=2790;
lama_waktu[6]=2930;
lama_waktu[7]=3250;

```

```

delay_ms(200);

```

```

//gait3

```

```

lama_waktu[0]=1930;
lama_waktu[1]=2400;
lama_waktu[2]=3560;
lama_waktu[3]=3670;
lama_waktu[4]=2440;
lama_waktu[5]=2790;
lama_waktu[6]=3530;
lama_waktu[7]=3250;

```

```

delay_ms(200);

```

```

//gait4

```

```

lama_waktu[0]=1930;
lama_waktu[1]=2400;
lama_waktu[2]=3560;
lama_waktu[3]=3670;
lama_waktu[4]=3040;
lama_waktu[5]=2790;
lama_waktu[6]=2930;
lama_waktu[7]=3250;

```

```

delay_ms(200);

```

```

l++;

```

```

}

```

```

//gait5

```



```

lama_waktu[0]=2630;
lama_waktu[1]=3100;
lama_waktu[2]=2860;
lama_waktu[3]=2970;
lama_waktu[4]=3040;
lama_waktu[5]=3390;
lama_waktu[6]=2930;
lama_waktu[7]=2650;

```

```

    delay_ms(200);
//stop
lama_waktu[0]=2630;
lama_waktu[1]=3100;
lama_waktu[2]=2860;
lama_waktu[3]=2970;
lama_waktu[4]=3040;
lama_waktu[5]=2790;
lama_waktu[6]=2930;
lama_waktu[7]=3250;

```

```

    delay_ms(200);

```

```

//kiri

```

```

//center
lama_waktu[0]=2630;
lama_waktu[1]=3100;
lama_waktu[2]=2860;
lama_waktu[3]=2970;
lama_waktu[4]=3040;
lama_waktu[5]=2790;
lama_waktu[6]=2930;
lama_waktu[7]=3250;

```

```

    delay_ms(200);

```

```

while (p<3)

```

```

{
//gait1
lama_waktu[0]=3330;
lama_waktu[1]=2400;
lama_waktu[2]=3560;
lama_waktu[3]=2270;
lama_waktu[4]=3040;
lama_waktu[5]=3390;
lama_waktu[6]=2930;

```

```

lama_waktu[7]=2650;

```

```

delay_ms(200);

```

```

//gait2

```

```

lama_waktu[0]=3330;
lama_waktu[1]=2400;
lama_waktu[2]=3560;
lama_waktu[3]=2270;
lama_waktu[4]=3040;
lama_waktu[5]=2790;
lama_waktu[6]=2930;
lama_waktu[7]=3250;

```

```

    delay_ms(200);

```

```

//gait3

```

```

lama_waktu[0]=2630;
lama_waktu[1]=3100;
lama_waktu[2]=2860;
lama_waktu[3]=2970;
lama_waktu[4]=2440;
lama_waktu[5]=2790;
lama_waktu[6]=3530;
lama_waktu[7]=3250;

```

```

    delay_ms(200);

```

```

//gait4

```

```

lama_waktu[0]=2630;
lama_waktu[1]=3100;
lama_waktu[2]=2860;
lama_waktu[3]=2970;
lama_waktu[4]=3040;
lama_waktu[5]=2790;
lama_waktu[6]=2930;
lama_waktu[7]=3250;

```

```

    delay_ms(200);

```

```

    p++;

```

```

}

```

```

    delay_ms(200);

```

```

//stop

```

```

lama_waktu[0]=2630;

```



```

lama_waktu[1]=3100;
lama_waktu[2]=2860;
lama_waktu[3]=2970;
lama_waktu[4]=3040;
lama_waktu[5]=2790;
lama_waktu[6]=2930;
lama_waktu[7]=3250;

```

```

delay_ms(200);

```

```

//putarbelakang

```

```

//center
lama_waktu[0]=2630;
lama_waktu[1]=3100;
lama_waktu[2]=2860;
lama_waktu[3]=2970;
lama_waktu[4]=3040;
lama_waktu[5]=2790;
lama_waktu[6]=2930;
lama_waktu[7]=3250;

```

```

delay_ms(200);

```

```

while (m<6)

```

```

{
//gait1
lama_waktu[0]=3330;
lama_waktu[1]=2400;
lama_waktu[2]=3560;
lama_waktu[3]=2270;
lama_waktu[4]=3040;
lama_waktu[5]=3390;
lama_waktu[6]=2930;
lama_waktu[7]=2650;

```

```

delay_ms(200);

```

```

//gait2
lama_waktu[0]=3330;
lama_waktu[1]=2400;
lama_waktu[2]=3560;
lama_waktu[3]=2270;
lama_waktu[4]=3040;
lama_waktu[5]=2790;
lama_waktu[6]=2930;
lama_waktu[7]=3250;

```

```

delay_ms(200);

```

```

//gait3

```

```

lama_waktu[0]=2630;
lama_waktu[1]=3100;
lama_waktu[2]=2860;
lama_waktu[3]=2970;
lama_waktu[4]=2440;
lama_waktu[5]=2790;
lama_waktu[6]=3530;
lama_waktu[7]=3250;

```

```

delay_ms(200);

```

```

//gait4

```

```

lama_waktu[0]=2630;
lama_waktu[1]=3100;
lama_waktu[2]=2860;
lama_waktu[3]=2970;
lama_waktu[4]=3040;
lama_waktu[5]=2790;
lama_waktu[6]=2930;
lama_waktu[7]=3250;

```

```

delay_ms(200);

```

```

m++;

```

```

}
delay_ms(200);

```

```

//stop

```

```

lama_waktu[0]=2630;
lama_waktu[1]=3100;
lama_waktu[2]=2860;
lama_waktu[3]=2970;
lama_waktu[4]=3040;
lama_waktu[5]=2790;
lama_waktu[6]=2930;
lama_waktu[7]=3250;

```

```

delay_ms(200);

```

```

while(1){};
}

```



UNIVERSITAS BRAWIJAYA

LAMPIRAN III

DATASHEET

