

**IMPLEMENTASI INVERSE KINEMATIC  
PADA PERGERAKAN MOBILE ROBOT KRPAI  
DIVISI BERKAKI**

**SKRIPSI  
KONSENTRASI ELEKTRONIKA**

Diajukan Untuk Memenuhi Persyaratan  
Memperoleh Gelar Sarjana Teknik



Disusun Oleh :

**EKY PRASETYA  
NIM. 0910633047-63**

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

**LEMBAR PERSETUJUAN**

**IMPLEMENTASI INVERSE KINEMATIC  
PADA PERGERAKAN MOBILE ROBOT KRPAI  
DIVISI BERKAKI**

**SKRIPSI**

Diajukan untuk Memenuhi Persyaratan  
Memperoleh Gelar Sarjana Teknik



Disusun oleh:

**Eky Prasetya**

**NIM. 0910633047-63**

**Telah diperiksa dan disetujui oleh :**

**Dosen Pembimbing I**

**Dosen Pembimbing II**

**Waru Djuriatno, ST, MT.**  
**NIP. 19690725 1999702 1 001**

**Mochammad Rif'an, ST., MT.**  
**NIP. 19710301 200012 1 001**

**LEMBAR PENGESAHAN**

**IMPLEMENTASI INVERSE KINEMATIC  
PADA PERGERAKAN MOBILE ROBOT KRPAI  
DIVISI BERKAKI**

**SKRIPSI**

Disusun Oleh:

**EKY PRASETYA**

**NIM. 0910633047 – 63**

Skrripsi ini telah diuji dan dinyatakan lulus pada  
tanggal 21 Januari 2014

**DOSEN PENGUJI**

**Ir. Nanang Sulistyanto, MT.**  
**NIP. 19700113 199403 1 002**

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

**Akhmad Zainuri, ST., MT.**  
**NIP. 19840120 201212 1 003**

Mengetahui  
Ketua Jurusan Teknik Elektro

**M. Aziz Muslim, ST., MT., Ph.D**  
**NIP. 19741203 200012 1 001**



## PENGANTAR

Puji dan syukur penulis panjatkan kehadirat Tuhan YME yang telah memberikan rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan skripsi yang berjudul “Implentasi *Inverse Kinematic* pada Pergerakan Mobile Robot KRPAI Divisi Berkaki” dengan baik. Skripsi ini disusun sebagai salah satu syarat untuk mencapai gelar Sarjana Teknik dari jurusan Teknik Elektro Fakultas Teknik Universitas Brawijaya.

Penulis menyadari bahwa tanpa bantuan, bimbingan serta dorongan dari semua pihak, penyelesaian skripsi ini tidak mungkin bisa terwujud. Pada kesempatan ini penulis menyampaikan rasa terima kasih yang sebesar-besarnya kepada:

- Tuhan YME atas rahmat dan karunia yang telah diberikan,
- Ayah, Ibu dan kakak, R. Suhariono Broto, Sri Widayaningsih, dan Rosa Riza Apriliya. Terima kasih atas do'a, cinta, kasih sayang, kebaikan, nasihat, dan motivasi yang luar biasa, yang tak akan pernah tergantikan oleh suatu apapun,
- Bapak Aziz Muslim, ST.,MT.,Ph.D, MS selaku Ketua Jurusan Teknik Elektro Universitas Brawijaya,
- Bapak Hadi Suyono, ST.,MT.,Ph.D selaku Sekretaris Jurusan Teknik Elektro Universitas Brawijaya,
- Bapak Moch.Rif'an, ST.,MT selaku Ketua Prodi Strata Satu Jurusan Teknik Elektro Universitas Brawijaya, dan sebagai Dosen Pembimbing II atas segala bimbingan, pengarahan, saran dan kritik yang telah diberikan,
- Bapak R. Arief Setyawan, ST., MT selaku Dosen Pembimbing akademik atas segala bimbingan, nasehat, gagasan, ide, saran, motivasi dan masukan yang telah diberikan,
- Ibu Ir. Nurussa'adah., MT sebagai Ketua Kelompok Dosen Keahlian Elektronika Jurusan Teknik Elektro Universitas Brawijaya
- Bapak Waru Djuriatno, ST, MT. sebagai Dosen Pembimbing I atas segala bimbingan, pengarahan, ide, saran serta motivasi yang telah diberikan selama pengerjaan skripsi ini,
- Teman-teman Ampere angkatan 2009,

- Rekan seperjuangan dalam skripsi, Kharisma, Lintang, Vanti, Bona, Sam, Juang, Jatra, Ikhsan, Mbah, Wito, Somad, Gladi, dan Aka terima kasih atas segala bantuan yang telah diberikan,
- Fransisca Nining W yang setia menemani saat suka dan duka dalam setiap prosesnya,
- Seluruh Keluarga Besar Tim Robot UB Jurusan Teknik Elektro atas segala bantuan alat, bahan dan masukan-masukannya yang telah di berikan,
- Seluruh Keluarga KRPAI, mas mas yang banyak sekali, Anas, Lutfi, Manda, Guntur, Desta, Ricky, Reza, Sirojul, Sofyan,
- Sekuruh Keluarga Besar Dentech Tool, Rafi teman seperjuanganan, Latifah, Kiki, Yuli,
- Sekret KRPAI, sekret KRI, sekret KRSI, dan Laboratorium Dasar Elektronika yang selama ini telah menyediakan tempat bagi penulis dalam mengerjakan skripsi ini,
- Seluruh teman-teman serta semua pihak yang tidak mungkin untuk dicantumkan namanya satu-persatu, terima kasih banyak atas segala bentuk bantuan dan dukungannya.

Dalam penyusunan skripsi ini, penulis menyadari bahwa skripsi ini belumlah sempurna, karena keterbatasan ilmu dan kendala-kendala lain yang terjadi selama pengerjaan skripsi ini. Semoga tulisan ini dapat bermanfaat dan dapat digunakan untuk pengembangan lebih lanjut.

Malang, Januari 2014

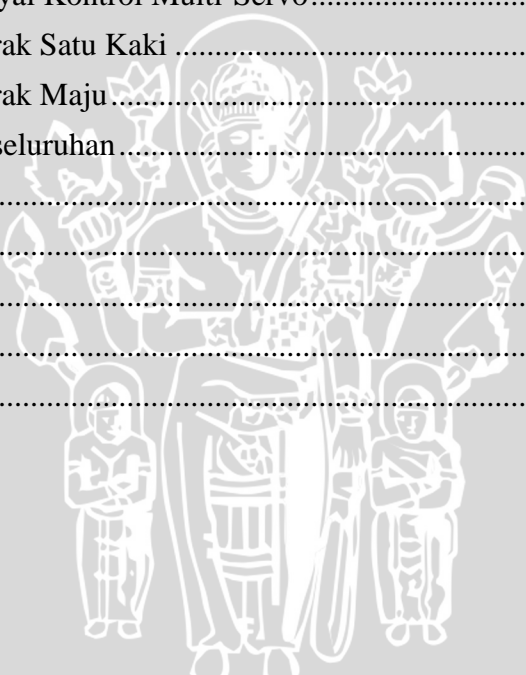
Penulis



## DAFTAR ISI

PENGANTAR .....	i
DAFTAR ISI.....	iii
DAFTAR GAMBAR .....	v
DAFTAR TABEL.....	vii
ABSTRAK .....	viii
PENDAHULUAN .....	1
1.1    LATAR BELAKANG.....	1
1.2    RUMUSAN MASALAH .....	2
1.3    RUANG LINGKUP .....	2
1.4    TUJUAN .....	3
1.5    SISTEMATIKA PEMBAHASAN.....	3
TINJAUAN PUSTAKA .....	5
2.1    Robot Berkaki.....	5
2.2    Kontes Robot Cerdas Indonesia Divisi Senior Berkaki .....	7
2.3    Keseimbangan Robot Berkaki.....	8
2.3.1    Keseimbangan Statis .....	9
2.3.2    Keseimbangan Dinamis .....	10
2.4    Kinematika dengan Persamaan Trigonometri .....	10
2.4.1    Kinematika Satu Sendi (1 DOF) .....	10
2.4.2    Kinematika Dua Sendi (2 DOF).....	11
2.4.3    Kinematika Tiga Sendi (3 DOF).....	13
2.5    Motor DC Servo .....	14
2.6    Mikrokontroler STM32F4.....	17
2.6.1    General Purpose Input/Output (GPIO) .....	17
2.6.2    Pengendali DMA.....	18
2.7    UBEC ( <i>Universal Battery Elimination Circuit</i> ) .....	18
METODE PENELITIAN.....	19
3.1    Penentuan Spesifikasi Alat.....	19
3.2    Perancangan dan Pembuatan Alat .....	19
3.2.1    Perancangan dan Pembuatan Perangkat Keras .....	19
3.2.2    Perancangan dan Penyusunan Perangkat Lunak .....	27
3.3    Pengujian Alat .....	28
3.3.1    Pengujian Tiap Blok.....	28

3.3.2	Pengujian Keseluruhan Sistem.....	28
	<b>PERANCANGAN DAN PEMBUATAN ALAT .....</b>	<b>29</b>
4.1	Penentuan Spesifikasi alat .....	29
4.2	Perancangan Mekanik Robot.....	30
4.3	Pola Langkah .....	36
4.4	Kontrol Multi Servo .....	39
4.4.1	Algoritma Mikrokontroler Kontrol Servo.....	42
4.5	Perancangan Perangkat Lunak .....	45
4.6	Perancangan Sistem.....	46
4.7	Perancangan Antarmuka Mikrokontroler.....	47
4.8	Perancangan Catu Daya Mikrokontroler.....	49
	<b>PENGUJIAN DAN ANALISIS .....</b>	<b>50</b>
5.1	Pengujian Sinyal Kontrol Multi-Servo.....	50
5.2	Pengujian Gerak Satu Kaki .....	52
5.3	Pengujian Gerak Maju.....	53
5.4	Pengujian Keseluruhan.....	56
	<b>PENUTUP.....</b>	<b>59</b>
6.1	Kesimpulan.....	59
6.2	Saran.....	59
	<b>DAFTAR PUSTAKA .....</b>	<b>60</b>
	<b>LAMPIRAN.....</b>	<b>61</b>



## DAFTAR GAMBAR

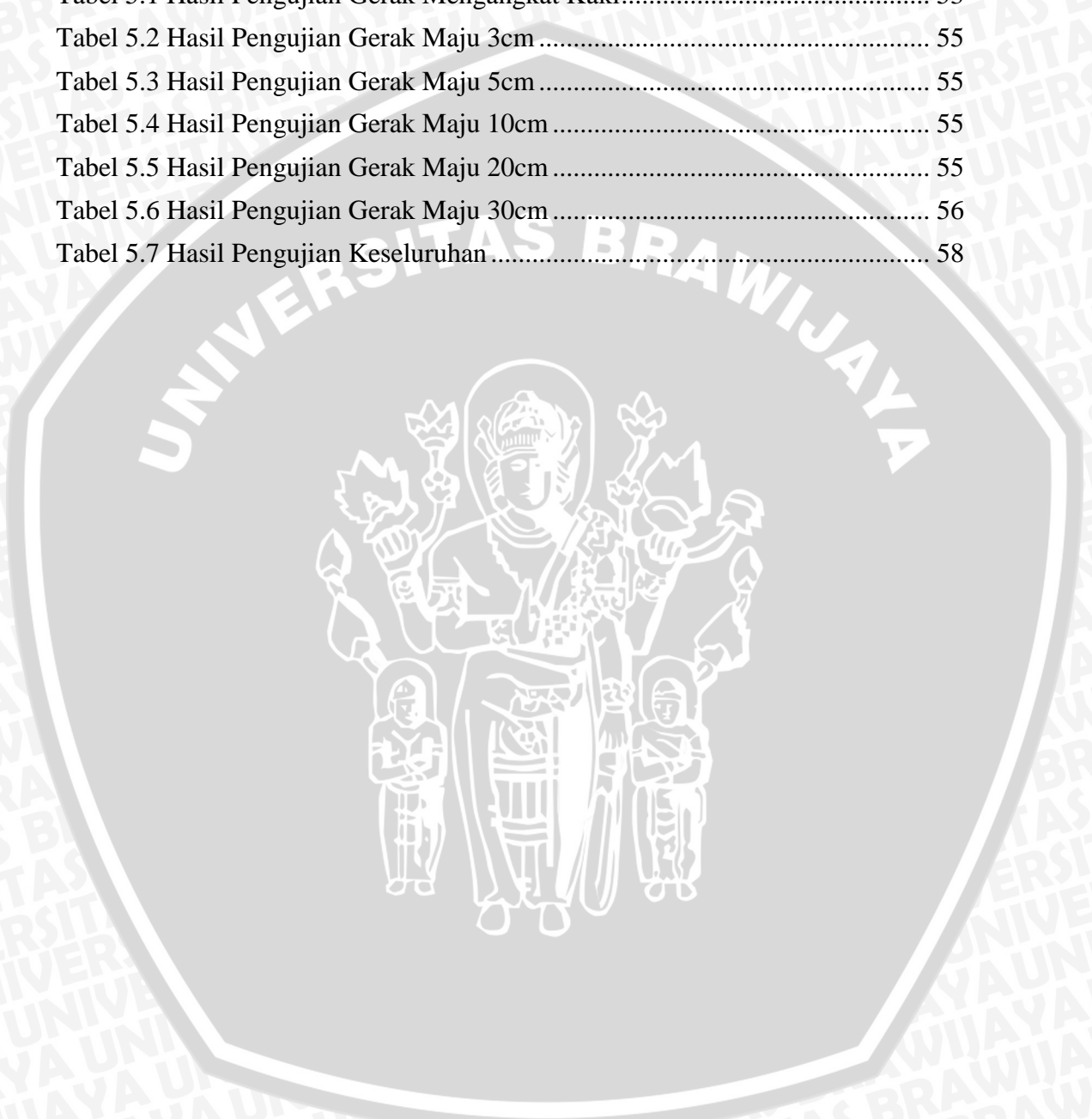
Gambar 2.1 Robot Kondo kmr m6.....	6
Gambar 2.2. Derajat Kebebasan pada Kaki Robot SILO4.....	6
Gambar 2.3. Arena KRCI Divisi Senior Berkaki.....	7
Gambar 2.4. Keseimbangan Robot Saat Mengangkat Satu Kakinya.....	8
Gambar 2.5. Keseimbangan Statis pada Robot Quadrupedal dan Robot Hexapodal .....	9
Gambar 2.6. Konfigurasi Robot Tangan dengan 1 DOF .....	10
Gambar 2.7. Konfigurasi Robot Tangan Planar dengan 2 DOF .....	11
Gambar 2.8. Konfigurasi Robot Tangan Planar dengan 3 DOF .....	13
Gambar 2.9. Blok Diagram Motor DC Servo .....	14
Gambar 2. 10. Motor Servo.....	15
Gambar 2.11. Konfigurasi Pin Pengkabelan Motor Servo.....	15
Gambar 2.12. Diagram waktu PWM motor servo .....	16
Gambar 2.13. Konfigurasi Pin STM32F4.....	17
Gambar 2.14, UBEC (Universal Battery Elimination Circuit ).....	18
Gambar 3.1. Diagram Blok Sistem Hexapod robot .....	19
Gambar 3.2. Perspektif Desain Mekanik <i>Hexapod robot</i> .....	20
Gambar 3.3 Bentuk Konstruksi Badan Robot.....	21
Gambar 3.4 Pergeseran titik pusat robot terhadap kaki belakang.....	21
Gambar 3.5 Pergeseran Titik Pusat.....	23
(a)terhadap Kaki Tengah, (b) terhadap Kaki Depan. ....	23
Gambar 3.6 Bentuk Konstruksi Kaki Robot .....	23
Gambar 3.7 Arah Putar Servo .....	25
Gambar 3.8 Gerak mengangkat kaki.....	25
Gambar 3.9 Diagram Gait .....	26
Gambar 4.1. Diagram Blok Sistem <i>Hexapod robot</i> .....	29
Gambar 4.2. Perspektif Desain Mekanik <i>Hexapod robot</i> .....	30
Gambar 4.3 Bentuk Konstruksi Badan Robot.....	31
Gambar 4.4 Pergeseran titik pusat robot terhadap kaki belakang.....	32
Gambar 4.5 Pergeseran Titik Pusat.....	32
(a)terhadap Kaki Tengah, (b) terhadap Kaki Depan. ....	32
Gambar 4.6 Bentuk Konstruksi Kaki Robot .....	34
Gambar 4.7 Arah Putar Servo .....	35



Gambar 4.8 Gerak mengangkat kaki.....	35
Gambar 4.9 Diagram Gait.....	36
Gambar 4.10 Diagram Gait.....	38
Gambar 4.11 Sinyal Kontrol Motor DC Servo .....	40
Gambar 4.12 Sinyal Kontrol Motor DC Servo .....	40
Gambar 4.13. Diagram Alir Program Interupsi <i>Timer3</i> .....	42
Gambar 4.14. Diagram Alir Program Interupsi Timer OCRA.....	43
Gambar 4.15. Diagram Alir Program Interupsi Timer OCRB.....	44
Gambar 4.16. Diagram Alir Program Interupsi Timer OCRC.....	44
Gambar 4.17. Diagram Alir Program Utama Mikrokontroler Kontrol Servo.....	45
Gambar 4.18. Skema Rangkaian Mikrokontroler STM32F4Discovery .....	47
Gambar 4.19. Konfigurasi I/O STM32F4Discovery.....	48
Gambar 4.20. Rangkaian Catu Mikrokontroler.....	49
Gambar 5. 1. Diagram Blok Pengujian Sinyal Kontrol Multi Servo .....	50
Gambar 5. 1. Pengujian Sinyal Kontrol Multi Servo .....	51
Gambar 5. 3. Sinyal Kontrol Multi Servo.....	51
Gambar 5.4. Diagram Blok Pengujian Gerak Satu Kaki .....	52
Gambar 5.5. Pengujian Gerak Satu Kaki (a) kaki kanan depan menapak (b) kaki kanan depan terangkat (c) kaki kiri depan menapak (d) kaki kiri depan terangkat .....	53
Gambar 5.6. Pengujian Gerak Maju (a) Posisi Robot sebelum bergerak (b) Posisi Robot bergerak 3cm(c) Posisi Robot bergerak 5cm (d) Posisi Robot bergerak 10cm (e) Posisi Robot bergerak 20cm (f) Posisi Robot bergerak 30cm .....	54
Gambar 5.7 Respon Pengujian Gerak Maju.....	56
Gambar 5.8 Lintasan Pengujian Kesluruhan.....	57
Gambar 5.9 Pengujian Kesluruhan (a) Posisi robot sebelum bergerak (b) Posisi robot setelah bergerak .....	57

## DAFTAR TABEL

Tabel 4.1 Konstanta Gerak Kaki.....	39
Tabel 5.1 Hasil Pengujian Gerak Mengangkat Kaki.....	53
Tabel 5.2 Hasil Pengujian Gerak Maju 3cm .....	55
Tabel 5.3 Hasil Pengujian Gerak Maju 5cm .....	55
Tabel 5.4 Hasil Pengujian Gerak Maju 10cm .....	55
Tabel 5.5 Hasil Pengujian Gerak Maju 20cm .....	55
Tabel 5.6 Hasil Pengujian Gerak Maju 30cm .....	56
Tabel 5.7 Hasil Pengujian Keseluruhan.....	58



## ABSTRAK

Kontes Robot Pemadam Api Indonesia (KRPAI) merupakan ajang perlombaan robotika nasional yang terdiri atas dua divisi yaitu Divisi Beroda dan Divisi Berkaki. Salah satu masalah yang sering dihadapi dalam perlombaan divisi berkaki adalah pergerakan untuk menyusuri arena dalam lomba. Tugas akhir ini merancang dan mengimplementasikan *inverse kinematic* sebagai sistem pergerakan pada robot berkaki. Pada perhitungan *inverse kinematic* akan dihasilkan sudut untuk masing-masing motor servo, agar robot dapat berjalan sesuai dengan yang diharapkan. Pola langkah digunakan untuk mengatur bagian dari kaki robot mana yang harus diangkat terlebih dahulu dan untuk kaki lain digeser. Dengan perhitungan *inverse kinematic* dapat diketahui rata-rata kesalahan gerak maju antara 3 cm hingga 30 cm yang dihasilkan 0,6 cm dengan kesalahan terbesar 2 cm pada jarak tempuh 30 cm. hasil gerak maju untuk 244 cm atau jalur terpanjang dalam lomba menunjukkan rata-rata kesalahan pembacaan sudut sebesar  $2,54^\circ$  dengan kesalahan terbesar  $3,51^\circ$ .

Kata Kunci : KRPAI divisi berkaki, *inverse kinematic*, pola langkah.



## BAB I PENDAHULUAN

### 1.1 LATAR BELAKANG

Robotika di Indonesia sudah pesat. Di bidang industri misalnya, sudah banyak digunakan robot-robot sebagai pengganti tugas manusia yang dapat mempermudah pekerjaan. Di bidang pendidikan, teknologi robotika juga sudah mulai diperkenalkan sejak tingkat Sekolah Dasar.

Salah satu wadah pengembangan teknologi robotika di bidang pendidikan adalah Kontes Robot yang diadakan oleh Direktorat Jenderal Pendidikan Tinggi (Dirjen DIKTI), yang terdiri dari 4 kategori, yaitu Kontes Robot Indonesia (KRI), Kontes Robot Pemadam Api Indonesia (KRPAI), Kontes *Robo Soccer Humanoid League*, dan Kontes Robot Seni Indonesia (KRSI) yang diselenggarakan setiap tahunnya.

Kontes Robot Pemadam Api Indonesia (KRPAI) dibagi menjadi dua divisi yaitu divisi beroda dan divisi berkaki. KRPAI divisi berkaki merupakan divisi yang mewajibkan robot untuk menggunakan kaki dalam pergerakannya. Tugas robot adalah untuk memadamkan api pada salah satu ruang dalam sebuah model rumah yang digunakan sebagai arena pertandingan.

Konfigurasi arena pertandingan yang digunakan dapat berubah-ubah sesuai dengan hasil undian. Robot harus mampu beradaptasi dan melaksanakan tugasnya sesuai dengan kondisi arena pertandingan dengan cara bergerak menyusuri arena. Agar dapat menyusuri arena tersebut maka *mobile robot* yang dirancang harus mampu mendeteksi keberadaan dinding dan lorong yang menjadi lintasan robot.

Untuk tujuan tersebut maka sebuah *mobile robot* harus dilengkapi dengan sensor yang dapat mengambil data. Data yang dihasilkan kemudian diolah oleh mikrokontroler. Salah satu sensor yang banyak digunakan adalah sensor ultrasonik (PING))) untuk mendeteksi jarak.

Permasalahan utama robot KRPAI divisi berkaki adalah merancang dan menentukan pergerakan *mobile robot* agar robot dapat bergerak menyusuri arena dan menyelesaikan tugasnya. Oleh karena itu, dibutuhkan suatu sistem kontrol

untuk menunjang cara tersebut yang dapat mengatasi kelemahan-kelemahan dalam pergerakan.

Skripsi ini akan dirancang suatu Sistem pergerakan Robot KRPAI Divisi Berkaki dengan menggunakan inverse kinematik. Sistem tersebut dirancang untuk mendapatkan respon yang fleksibel sehingga *mobile robot* otomatis diharapkan dapat dengan cepat menyelesaikan tugasnya.

Perancangan kinematika balik (*inverse kinematic*) berfungsi untuk menentukan besar sudut masing-masing sendi saat ujung kaki berada pada suatu koordinat tertentu, dan *gait pattern* untuk mengatur koordinasi masing-masing kaki untuk membentuk suatu pola pergerakan dasar dari robot. Sehingga robot berkaki ini memiliki pola pergerakan yang halus, presisi, dan cepat, untuk aplikasi robot berjalan dan lebih mudah dikembangkan untuk membentuk pola pergerakan lainnya.

## 1.2 RUMUSAN MASALAH

Sesuai dengan latar belakang, maka rumusan masalah dalam skripsi ini adalah:

- 1) Bagaimana merancang dan membuat sistem mekanik untuk Robot Berkaki dengan 3 sendi di setiap kakinya.
- 2) Bagaimana merancang metode pergerakan menggunakan inverse kinematic untuk Robot Berkaki Enam dengan 3 sendi di setiap kakinya.
- 3) Bagaimana merancang rangkaian atau sistem elektronika yang menunjang penggunaan motor servo sebagai penggerak pada Robot Berkaki Enam.
- 4) Bagaimana merancang dan membuat perangkat lunak sebagai pengolah data dan kendali utama sistem.

## 1.3 RUANG LINGKUP

Dengan perancangan untuk skripsi ini permasalahan dibatasi oleh hal-hal sebagai berikut :

- 1) *Mobile robot* yang dibuat memiliki 6 kaki dengan 3 derajat kebebasan (Degree of Freedom (DOF)) pada setiap kakinya.
- 2) Motor servo yang digunakan adalah tipe Hi-Tech.



- 3) Model lintasan yang digunakan lorong trepanjang dalam Kontes robot KRPAI . Robot bernavigasi lurus tanpa objek atau halangan apa pun.
- 4) Sistem bersifat *close loop*, hanya bernavigasi dan memantau posisi robot tanpa perhitungan waktu.
- 5) Tidak memperhitungkan konsumsi energy robot.

#### 1.4 TUJUAN

Tujuan dari penelitian ini adalah merancang dan membuat suatu sistem pergerakan menggunakan *inverse kinematic* yang mampu membuat robot bernavigasi untuk diterapkan pada robot KRPAI divisi berkaki. Serta lebih mudah dikembangkan untuk membentuk pola pergerakan lainnya.

#### 1.5 SISTEMATIKA PEMBAHASAN

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

##### **BAB I           Pendahuluan**

Membahas latar belakang, rumusan masalah, ruang lingkup, tujuan dan sistematika pembahasan.

##### **BAB II           Tinjauan Pustaka**

Membahas teori-teori yang mendukung dalam perancangan dan pembuatan alat, yang meliputi : teori Robot Berkaki, peraturan lomba KRCI Divisi Senior Berkaki, teori keseimbangan robot berkaki, teori Kinematika dengan Persamaan Trigonometri, motor DC Servo, dan mikrokontroler STM332F4.

##### **BAB III          Metodologi Penulisan**

Membahas metode penelitian dan perencanaan alat.

##### **BAB IV          Perancangan dan Pembuatan Alat**

Membahas desain dan perancangan Robot Berkaki Enam (*hexapod robot*), dan memodelkan sistem pergerakannya. Kemudian mengendalikan robot dengan mikrokontroler agar dapat bergerak lurus.

##### **BAB V          Pengujian dan Analisis**



Membahas hasil pengujian sistem Robot Berkaki Enam (*hexapod robot*) untuk tiap-tiap blok dan secara keseluruhan.

## **Bab VI Kesimpulan dan Saran**

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



## BAB II

### TINJAUAN PUSTAKA

Beberapa teori pendukung yang perlu dibahas dalam pembuatan system *Hexapod robot* ini meliputi :

- 1) Robot Berkaki
- 2) Kontes Robot Cerdas Indonesia Divisi Senior Berkaki.
- 3) Keseimbangan Robot Berkaki
- 4) Kinematika dengan Persamaan Trigonometri
- 5) Motor DC Servo
- 6) Mikrokontroler STM32F4
- 7) UBEC (*Universal Battery Elimination Circuit* )

#### 2.1 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.

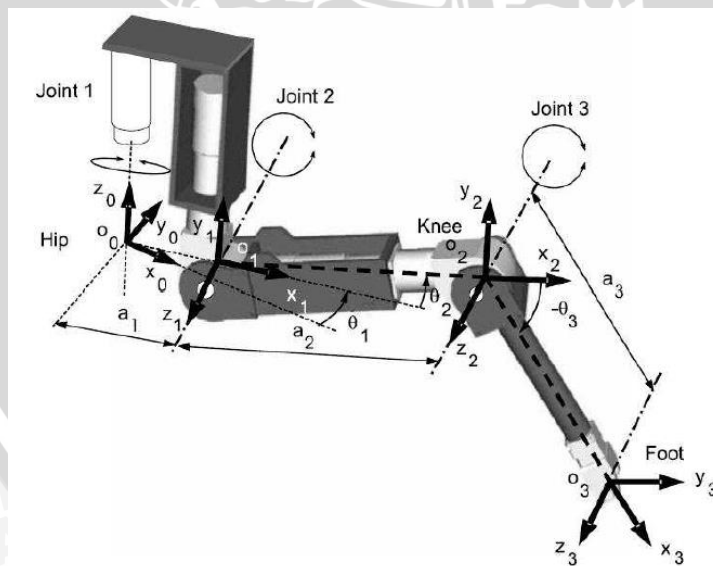
Robot berkaki enam atau lebih dikenal dengan istilah *hexapod robot* adalah robot yang menggunakan enam buah kaki untuk melakukan pergerakan. Robot jenis ini memiliki kestabilan yang lebih baik jika dibandingkan dengan robot berkaki dua atau berkaki empat. Pada saat bergerak, tiga kaki akan mengayun ke atas sedangkan tiga kaki lain masih menapak ke tanah. Inilah yang disebut dengan kestabilan statis, 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 enam ditunjukkan dalam gambar 2.1



Gambar 2.1 Robot Kondo kmr m6

Sumber : jogjarobotika, 2013

Robot berkaki enam 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 keenam kaki inilah yang dapat menggerakkan robot ke depan, belakang, atau berputar. Contoh persendian dari salah satu kaki robot SILO4 ditunjukkan dalam gambar 2.2.



Gambar 2.2. Derajat Kebebasan pada Kaki Robot SILO4

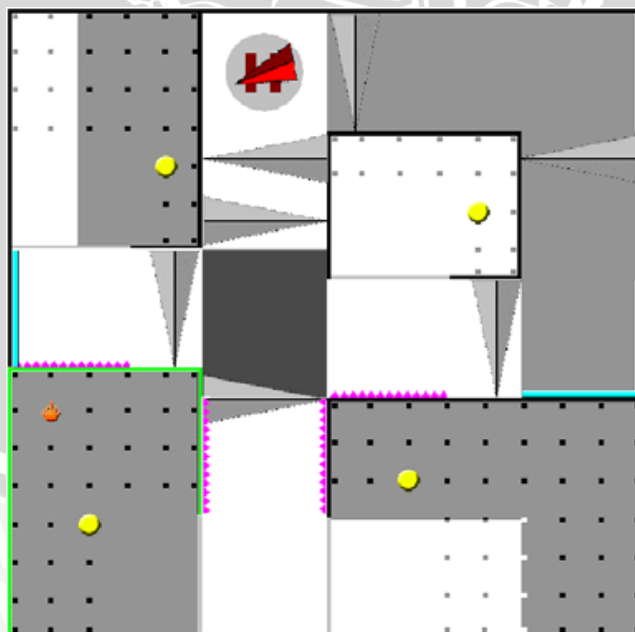
Sumber : Pablo Gonzalez de Santos, 2006 : 233



## 2.2 Kontes Robot Cerdas Indonesia Divisi Senior Berkaki

Kontes Robot Pemadam Api Indonesia (KRPAI) merupakan pertandingan robot tingkat nasional yang diadakan oleh Direktorat Jendral Pendidikan Tinggi (Dirjen DIKTI). Pertandingan ini diklasifikasikan ke dalam beberapa divisi yakni Divisi Beroda dan Divisi Berkaki. Untuk masing – masing divisi memiliki tema *fire fighting*, yaitu robot bergerak dari posisi *home* menelusuri arena untuk memadamkan api kemudian kembali lagi ke posisi *home*.

Divisi Senior Berkaki adalah salah satu Divisi KRCI yang memperlombakan robot-robot yang menggunakan kaki sebagai alat pergerakannya dengan misi mencari dan memadamkan api pada arena lapangan dengan tingkat kesulitan tertentu. Pada divisi ini yang diutamakan adalah kecepatan dan kemampuan robot dalam bernavigasi dan bermanuver dalam mencari dan memadamkan api di suatu arena dengan peta tertentu. Robot yang digunakan pada divisi ini 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 244 cm x 244 cm dan tinggi dinding 30 cm. Gambar 2.3 menunjukkan lapangan KRCI divisi berkaki.



Gambar 2.3. Arena KRCI Divisi Senior Berkaki

Sumber : DIKTI, 2012

Untuk tahun 2012, 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.. 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.

### 2.3 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 2.4.

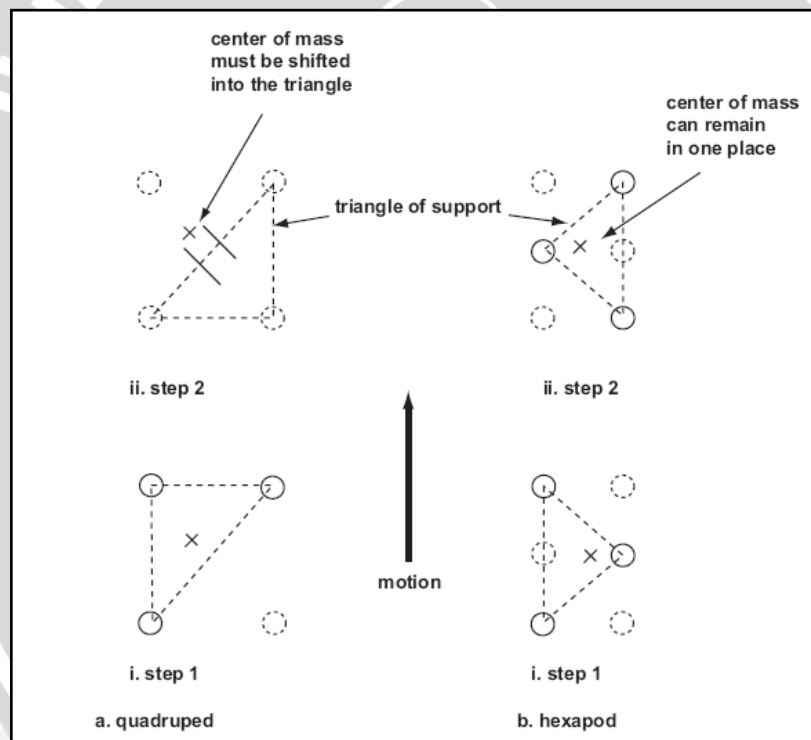


Gambar 2.4. Keseimbangan Robot Saat Mengangkat Satu Kakinya

Sumber : Cosmos magazine online, 2006 : 26

### 2.3.1 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. Gambar 2.5 menunjukkan bagaimana titik berat robot *quadrupedal* (empat kaki) dan *hexapodal* (enam kaki) saat robot berjalan.



Gambar 2.5. Keseimbangan Statis pada Robot Quadrupedal dan Robot Hexapodal

Sumber : Appin, 2007 : 294

Dalam Gambar 2.5 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, dan luasan telapak kaki harus besar.



### 2.3.2 Keseimbangan Dinamis

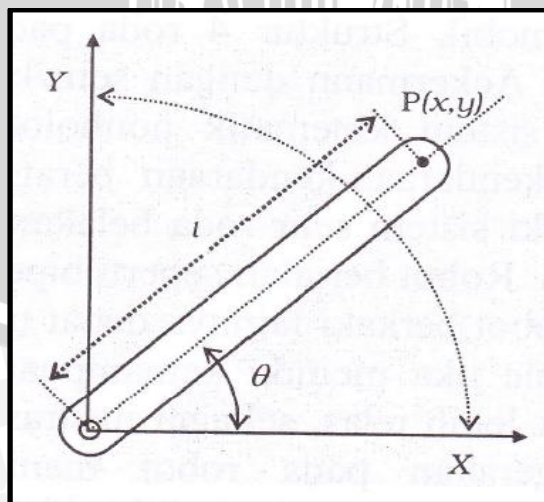
Keseimbangan dinamis adalah sebuah kondisi keseimbangan yang terbentuk dalam kondisi (pergerakan) apapun. Diperlukan analisa kestabilan sistem pada sebuah robot sebelum hal tersebut dapat diterapkan. Keseimbangan dinamis tidak harus diciptakan dari kombinasi posisi kaki, tetapi bisa diciptakan dari pergerakan 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.

### 2.4 Kinematika dengan Persamaan Trigonometri

Analisis persamaan kinematik dapat diselesaikan dengan cara yang paling dasar, yaitu menggunakan persamaan trigonometri. Setiap komponen dalam koordinat  $(x,y,z)$  dinyatakan sebagai transformasi dari tiap-tiap komponen ruang sendi  $(r,\theta)$ . Jari-jari  $r$  dalam persamaan sering ditulis sebagai panjang lengan  $(l)$ . Untuk koordinat 2 dimensi, komponen  $z$  dapat tidak dituliskan.

#### 2.4.1 Kinematika Satu Sendi (1 DOF)

Jika diketahui Robot Tangan Satu Sendi seperti dalam gambar 2.6.



Gambar 2.6. Konfigurasi Robot Tangan dengan 1 DOF

Sumber : Endra Pitowarno, 2006

Kedudukan titik  $P(x,y)$  dapat diperoleh dengan cara kinematik maju sebagai berikut,

$$x = l \cdot \cos(\theta) \text{ dan} \quad (2.1)$$

$$y = l \cdot \sin(\theta) \quad (2.2)$$

Jika  $(x,y)$  diketahui maka  $\theta$  dapat dihitung dengan cara,

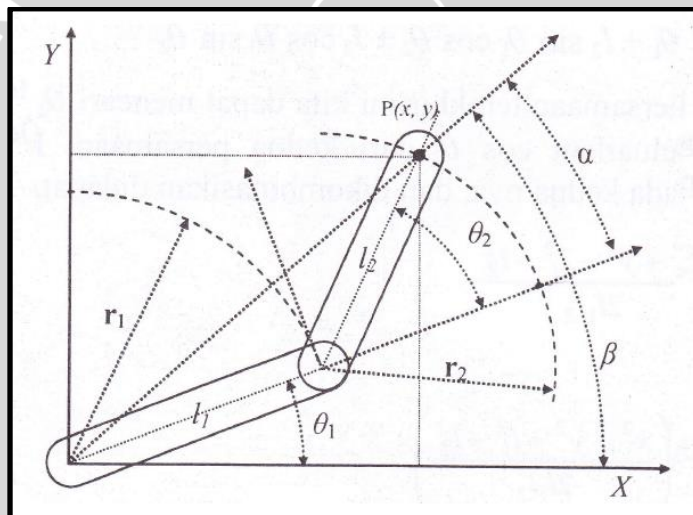
$$\theta = \arctan\left(\frac{y}{x}\right) \text{ atau}$$

$$\theta = \tan^{-1}\left(\frac{y}{x}\right) \quad (2.3)$$

Persamaan (2.3) adalah *Persamaan Kinematik Invers dari Robot Tangan Satu Sendi*.

#### 2.4.2 Kinematika Dua Sendi (2 DOF)

Jika diketahui Robot Tangan Planar Dua Sendi seperti dalam gambar 2.7.



Gambar 2.7. Konfigurasi Robot Tangan Planar dengan 2 DOF

Sumber : Endra Pitowarno, 2006

Kedudukan ujung lengan dinyatakan sebagai  $P(x,y)$ ,

$$P(x,y) = f(\theta_1, \theta_2) \quad (2.4)$$

Jika  $P$  diasumsikan sebagai vektor penjumlahan yang terdiri dari vektor  $\mathbf{r}_1$  lengan-1 dan vektor  $\mathbf{r}_2$  lengan-2, maka :

$$\mathbf{r}_1 = [l_1 \cos \theta_1, l_1 \sin \theta_1] \quad (2.5)$$

$$\mathbf{r}_2 = [l_2 \cos(\theta_1 + \theta_2), l_2 \sin(\theta_1 + \theta_2)] \quad (2.6)$$

maka

$$x = l_1 \cos \theta_1 + l_2 \cos(\theta_1 + \theta_2) \quad (2.7)$$

$$y = l_1 \sin \theta_1 + l_2 \sin(\theta_1 + \theta_2) \quad (2.8)$$



Persamaan (2.7) dan (2.8) adalah Persamaan Kinematik Maju dari Robot Tangan Planar Dua Sendi.

Kinematik invers robot dapat dijabarkan dengan menggunakan hukum identitas trigonometri sebagai berikut,

$$\cos(a + b) = \cos(a) \cos(b) - \sin(a) \sin(b) \quad \text{dan} \quad (2.9)$$

$$\sin(a + b) = \sin(a) \cos(b) + \sin(b) \cos(a) \quad (2.10)$$

Sehingga Persamaan (2.7) dan (2.8) dapat ditulis kembali sebagai berikut,

$$x = l_1 \cos \theta_1 + l_2 \cos \theta_1 \cos \theta_2 - l_2 \sin \theta_1 \sin \theta_2 \quad (2.11)$$

$$y = l_1 \sin \theta_1 + l_2 \sin \theta_1 \cos \theta_2 + l_2 \cos \theta_1 \sin \theta_2 \quad (2.12)$$

Dari persamaan diatas dapat diperoleh  $\theta_2$  dengan mengeluarkan  $\cos \theta_2$  dari kedua persamaan. Dengan operasi pangkat dua pada kesua persamaan tersebut diperoleh,

$$\cos \theta_2 = \frac{x^2 + y^2 - l_1^2 - l_2^2}{2 l_1 l_2} \quad (2.13)$$

sehingga

$$\theta_2 = \arccos \left( \frac{x^2 + y^2 - l_1^2 - l_2^2}{2 l_1 l_2} \right) \quad (2.14)$$

Untuk memperoleh nilai sudut  $\theta_1$  dapat digunakan analisa trigonometri dalam Gambar 16 sebagai berikut,

$$\tan \alpha = \frac{l_2 \sin \theta_2}{l_2 \cos \theta_2 + l_1} \quad (2.15)$$

$$\tan \beta = \frac{y}{x} \quad (2.16)$$

sedangkan

$$\theta_1 = \beta - \alpha \quad (2.17)$$

Dengan menggunakan hukum identitas trigonometri,

$$\tan (a - b) = \frac{\tan(a) - \tan(b)}{1 + \tan(a)\tan(b)} \quad (2.18)$$

diperoleh

$$\tan \theta_1 = \frac{y(l_1 + l_2 \cos \theta_2) - x \cdot l_2 \sin \theta_2}{x(l_1 + l_2 \cos \theta_2) + y \cdot l_2 \sin \theta_2} \quad (2.19)$$

Sehingga  $\theta_2$  dapat dihitung,

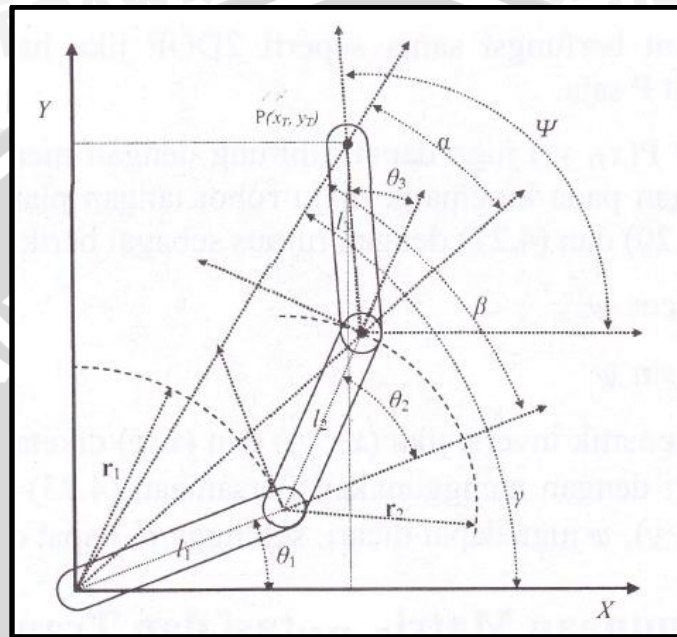
$$\theta_1 = \arctan \left( \frac{y(l_1 + l_2 \cos \theta_2) - x \cdot l_2 \sin \theta_2}{x(l_1 + l_2 \cos \theta_2) + y \cdot l_2 \sin \theta_2} \right) \quad (2.20)$$



Persamaan (2.14) dan (2.20) adalah Persamaan Kinematik Invers untuk Robot Tangan Planar Dua Sendi.

### 2.4.3 Kinematika Tiga Sendi (3 DOF)

Konfigurasi Robot Tangan Planar Tiga Sendi ditunjukkan dalam gambar 2.9.



Gambar 2.8. Konfigurasi Robot Tangan Planar dengan 3 DOF

Sumber : Endra Pitowarno, 2006

Dengan menggunakan analisis kinematik maju seperti pada *Kinematik Dua Sendi* diperoleh,

$$X_T = l_1 \cos \theta_1 + l_2 \cos (\theta_1 + \theta_2) + l_3 \cos (\theta_1 + \theta_2 + \theta_3) \quad (2.21)$$

$$Y_T = l_1 \sin \theta_1 + l_2 \sin (\theta_1 + \theta_2) + l_3 \sin (\theta_1 + \theta_2 + \theta_3) \quad (2.22)$$

dengan

$$\psi = (\theta_1 + \theta_2 + \theta_3) \quad (2.23)$$

$\psi$  adalah sudut arah hadap lengan-3 terhadap sumbu x.

Koordinat P dapat dicapai dalam lebih dari satu konfigurasi  $\psi = (\theta_1, \theta_2, \theta_3)$ . Dimisalkan ujung lengan pada posisi P ditahan dengan pada satu kedudukan yang tetap, kemudian sendi-2 dan sendi-3 dapat berubah-ubah posisinya tanpa merubah koordinat ujung lengan P. Hal tersebut menyebabkan lengan robot bersifat *redundant* (berlebih). Sifat *redundant* dapat mengurangi derajat

kebebasan robot. Sehingga lengan dengan 3 DOF seperti dalam Gambar 2.8 berfungsi sama seperti lengan 2 DOF jika hanya berorientasi pada koordinat P saja.

Koordinat  $P(X_T, Y_T)$  juga dapat dihitung dengan memanfaatkan hasil dari perhitungan pada kinematik maju robot tangan planar 2 DOF yaitu pada Persamaan (2.11) dan (2.12) sebagai berikut,

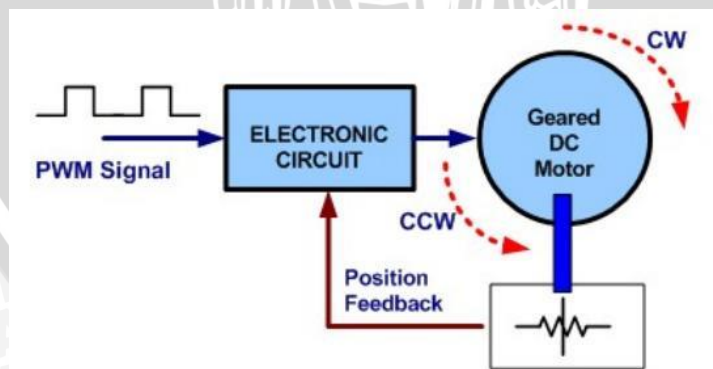
$$X_T = X - l_3 \cos \psi \quad (2.24)$$

$$Y_T = Y - l_3 \sin \psi \quad (2.25)$$

## 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.

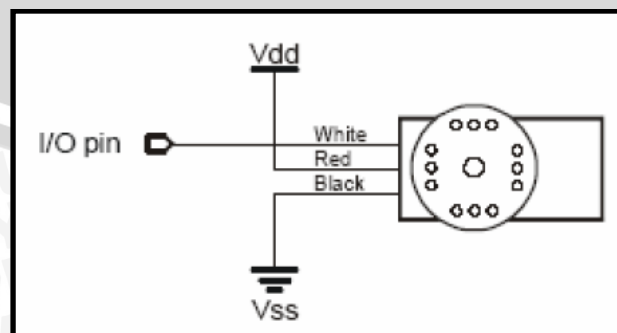


Gambar 2. 10. Motor Servo

Sumber : gwsus.com

Karakteristik motor servo adalah sebagai berikut :

- 1) Operasional dari servo motor dikendalikan oleh sebuah pulsa selebar  $\pm 20$  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 2.11
- 3) Pin Control untuk mengendalikan posisi.
- 4) Konstruksi didalamnya meliputi internal gear, potensiometer, dan feedback control.
- 5) 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 :

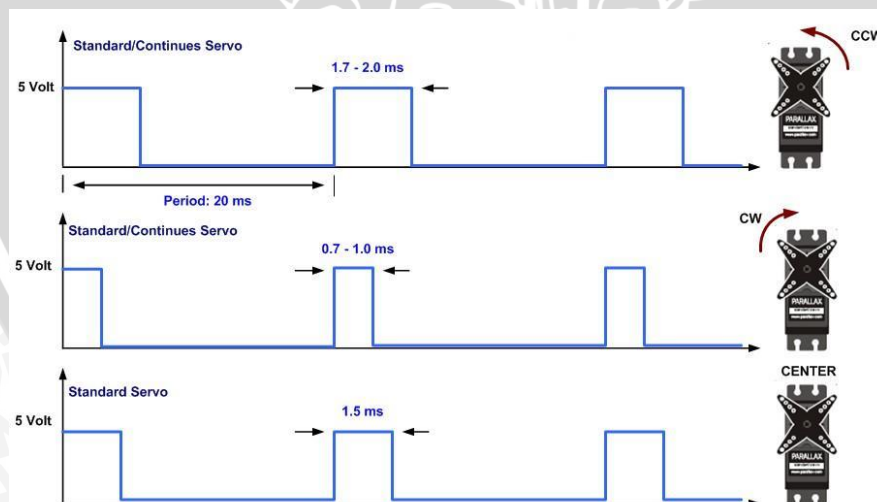
1) Motor Servo *Standar 180°*

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

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.

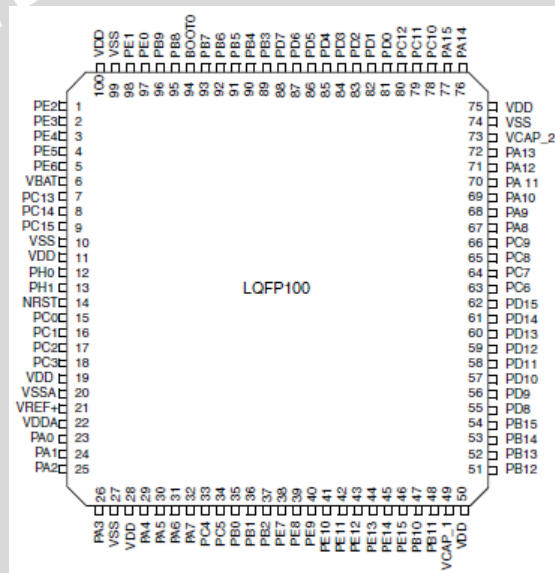


Gambar 2.12. Diagram waktu PWM motor servo

Sumber : Parallax, Inc.

## 2.6 Mikrokontroler STM32F4

Mikrokontroler STM32F4 adalah mikrokontroler CMOS 32-bit dengan arsitektur RISC buatan STMicroelectronics keluarga ARM Cortex-M4 dengan *clock* sampai dengan 168 MHz, mampu mengeksekusi perintah sampai dengan 210 MIPS (*Million Instruction per Second*). Mikrokontroler ini memiliki 1 MByte Flash PEROM (Flash Programmable and Eraseble Read Only Memory), 192 Kbyte SRAM, 100 pin, 5 buah port I/O yang mana setiap pin dalam masing masing port dapat diprogram tersendiri, memiliki dua belas buah *timer/counter* 16 bit dan dua buah *timer/counter* 32 bit, memiliki antarmuka kamera secara paralel 8-14 bit dengan kecepatan sampai 54 MByte/s, memiliki 24 *channel* ADC (*Analog Digital Converter*). gambar 2.13.



Gambar 2.13. Konfigurasi Pin STM32F4

Sumber : STMicroelectronics, 2012

### 2.6.1 General Purpose Input/Output (GPIO)

Masing – masing pin dari GPIO bisa dijadikan input, output dan fungsi alternatif. Bila pin dijadikan sebagai output ada beberapa konfigurasi yang dapat dilakukan secara software, yaitu jenis output *push pull* atau *open drain*, dengan atau tanpa *pull-up/pull-down*. Jika digunakan sebagai input maka dapat dikonfigurasi sebagai *floating*, *pull-up* dan *pull-down*. Fungsi alternatif setiap pin GPIO memiliki fungsi tersendiri, salah satunya yaitu sebagai pin untuk ADC. Kecepatan I/O untuk beresilasi maksimum sampai 84 Mhz.



### 2.6.2 Pengendali DMA

*Direct Memory Acces* (DMA) merupakan suatu teknik perpindahan data dari suatu alamat memori ke alamat memori yang lain tanpa mengganggu kerja dari mikroprocessor. Pada mikrokontroler STM32F4 terdapat 2 *port* DMA, yang masing masing port memiliki 8 aliran data yang sanggup menangani perpindahan dari memori ke memori, *peripheral* ke memory dan memori ke *peripheral*. DMA dapat diintegrasikan dengan *peripheral* :

- SPI (*Serial Peripheral Interface*)
- I2C (*Inter Integrated Circuit*)
- USART (*Universal Synchronous Asynchronous Receiver Transmitter*)
- *Timer*
- DAC (*Digital to Analog Converter*)
- SDIO (*Secure Digital Input/Output*)
- ADC (*Analog to Digital Converter*)

### 2.7 UBEC (*Universal Battery Elimination Circuit*)

*Universal Battery Elimination Circuit* adalah rangkaian elektronik yang mengambil daya dari battery pack atau sumber DC lainnya, dan menurunkannya ke level tegangan 5V atau 6V. Tegangan input maksimum tergantung pada spesifikasi UBEC. Digunakan untuk kebutuhan mencatu motor servo atau rangkaian lain yang bekerja pada tingkat tegangan 5V – 6V. UBEC biasanya digunakan pada aplikasi yang memerlukan arus lebih tinggi. Contoh UBEC ditunjukkan pada Gambar 2.14.



Gambar 2.14, UBEC (*Universal Battery Elimination Circuit*)

Sumber: Tjahyadi Christianto, 2013



## BAB III METODE PENELITIAN

Untuk menyelesaikan rumusan masalah dan merealisasikan tujuan penelitian yang terdapat di bab pendahuluan maka diperlukan metode untuk menyelesaikan masalah tersebut.

### 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:

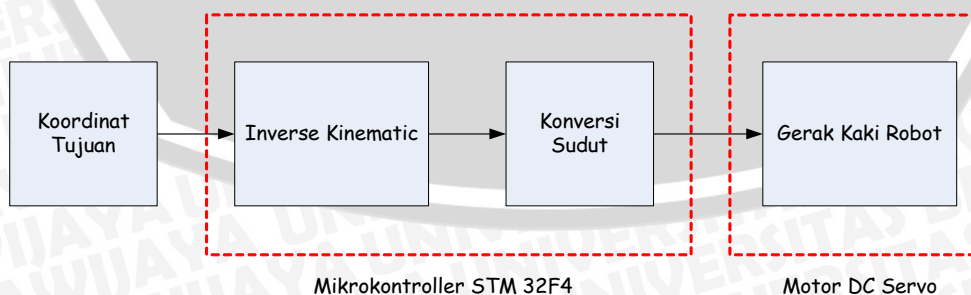
- 1) Robot berbahan dasar mika akrilik dan plat alumunium.
- 2) Robot memiliki 6 buah kaki.
- 3) Setiap kaki robot memiliki 3 derajat kebebasan (3 DOF) dengan menggunakan motor servo sebagai sendinya.
- 4) Motor servo yang digunakan adalah Hi-Tech Digital motor servo.
- 5) Dimensi maksimum robot adalah 46 cm x 31 cm x 27 cm (sesuai Peraturan Kontes Robot Pemadam Api Indonesia 2013).
- 6) Mikrokontroler STM32F4 discovery sebagai pengendali utama *mobile robot*.

### 3.2 Perancangan dan Pembuatan Alat

Perancangan dan pembuatan alat dalam skripsi ini dibagi menjadi beberapa bagian, yaitu perancangan system, inverse kinematic, control servo.

#### 3.2.1 Perancangan dan Pembuatan Perangkat Keras

- 1) Pembuatan blok diagram keseluruhan sistem



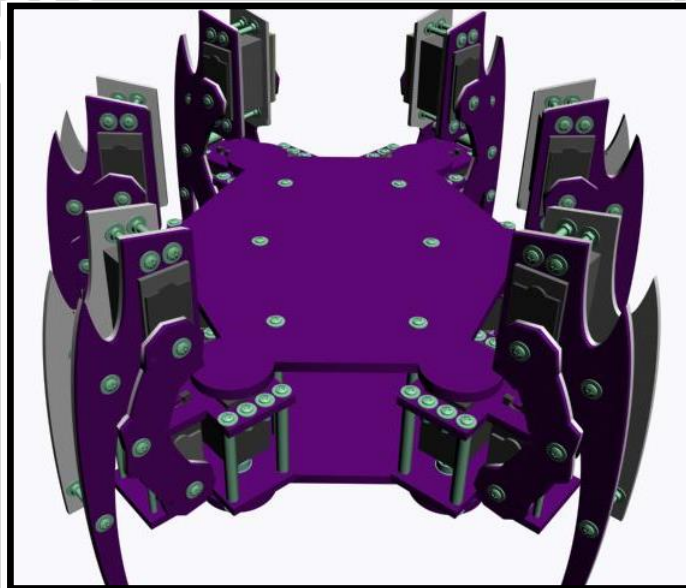
Gambar 3.1. Diagram Blok Sistem Hexapod robot

Fungsi bagian-bagian diagram blok dalam Gambar 3.1 adalah sebagai berikut:

- a. Mikrokontroler STM32F4 digunakan sebagai pengolah data yang diterima berupa koordinat tujuan (x,y,z) kemudian nilai koordinat akan diolah dan menghasilkan sudut untuk masing – masing motor servo. Setelah itu nilai sudut di konversikan menjadi sinyal PWM.
- b. Motor DC servo digunakan sebagai penggerak robot.

2) Pembuatan mekanik robot

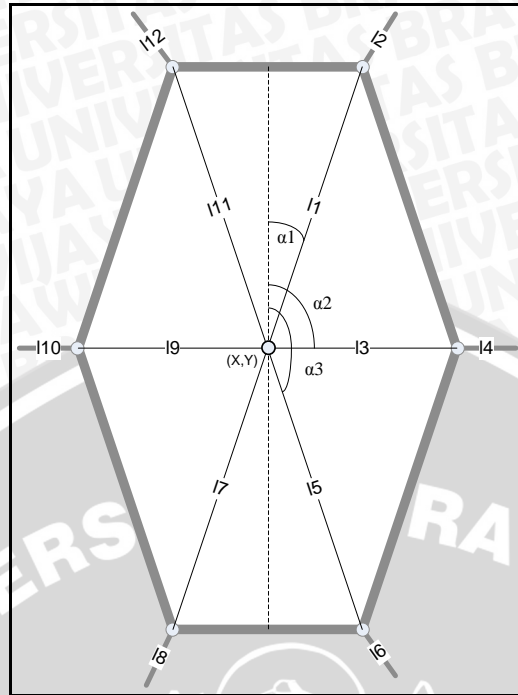
Sesuai dengan spesifikasi alat, robot hexapod memiliki 6 buah kaki, dan masing masing kaki memiliki 3 derajat kebebasan dengan menggunakan motor servo untuk setiap sendinya, ditunjukkan dalam gambar 3.2



Gambar 3.2. Perspektif Desain Mekanik *Hexapod robot*

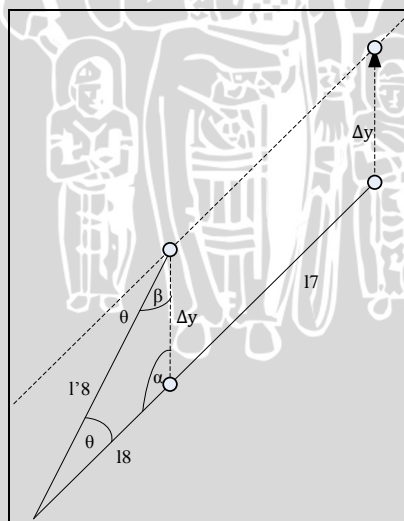
3) Perhitungan *inverse kinematic* yang akan digunakan

Dalam Gambar 3.3 menunjukkan sebuah robot berkaki enam yang digunakan dalam penelitian, terdiri dari badan robot berbentuk persegi panjang dan enam kaki yang didistribusi kedua sisi badan robot. Setiap kaki memiliki tiga derajat kebebasan.



Gambar 3.3 Bentuk Konstruksi Badan Robot

Gambar 3.3 merupakan tampak atas dari robot. Terdapat titik pusat robot yang terdapat di tengah badan robot dan lebar sudut dari masing-masing kaki terhadap titik pusat robot dengan kaki depan, kaki tengah dan kaki belakang.



Gambar 3.4 Pergeseran titik pusat robot terhadap kaki belakang

Pada gambar 3.4 yaitu pergeseran titik pusat robot terhadap kaki belakang, posisi awal robot adalah  $(x,y)$  dan posisi berikutnya setelah bergerak sejauh  $\Delta y$  adalah  $(x',y')$ , sehingga di dapatkan :



$$\frac{\Delta y}{\sin \theta_{1.4}} = \frac{l_8}{\sin \beta} \quad (3.1)$$

$$\sin \theta_{1.4} = \frac{\Delta y}{l_8} \sin \beta \quad (3.2)$$

Karena penjumlahan sudut dari bidang segitiga menghasilkan sudut  $180^\circ$  maka,

$$\beta = 180 - (\theta_{1.4} + \alpha) \text{ di substitusikan ke dalam persamaan menjadi,}$$

$$\sin \theta_{1.4} = \frac{\Delta y}{l_8} \sin(180 - (\theta_{1.4} + \alpha)) \text{ karena } \sin 180 \text{ adalah } 0 \text{ maka,}$$

$$\sin \theta_{1.4} = \frac{\Delta y}{l_8} \sin(\theta_{1.4} + \alpha) \quad (3.3)$$

dengan menggunakan hukum trigonometri dapat di ubah menjadi,

$$\sin \theta_{1.4} = \frac{\Delta y}{l_8} (\sin \theta_{1.4} \cos \alpha + \sin \alpha \cos \theta_{1.4}) \quad (3.4)$$

Untuk merubah salah satu sisi menjadi nilai satu dibagi dengan  $\sin \alpha$ ,

$$\frac{\sin \theta_{1.4}}{\sin \theta_{1.4}} = \frac{\Delta y}{l_8 \sin \theta_{1.4}} (\sin \theta_{1.4} \cos \alpha + \sin \alpha \cos \theta_{1.4}) \quad (3.5)$$

$$1 = \frac{\Delta y}{l_8} \left( \cos \alpha + \sin \alpha \frac{1}{\tan \theta_{1.4}} \right)$$

$$\sin \alpha \frac{1}{\tan \theta_{1.4}} = 1 - \frac{\Delta y}{l_8} \cos \alpha$$

$$\frac{1}{\tan \theta_{1.4}} = \frac{1}{\sin \alpha} - \frac{\Delta y}{l_8 \tan \alpha}$$

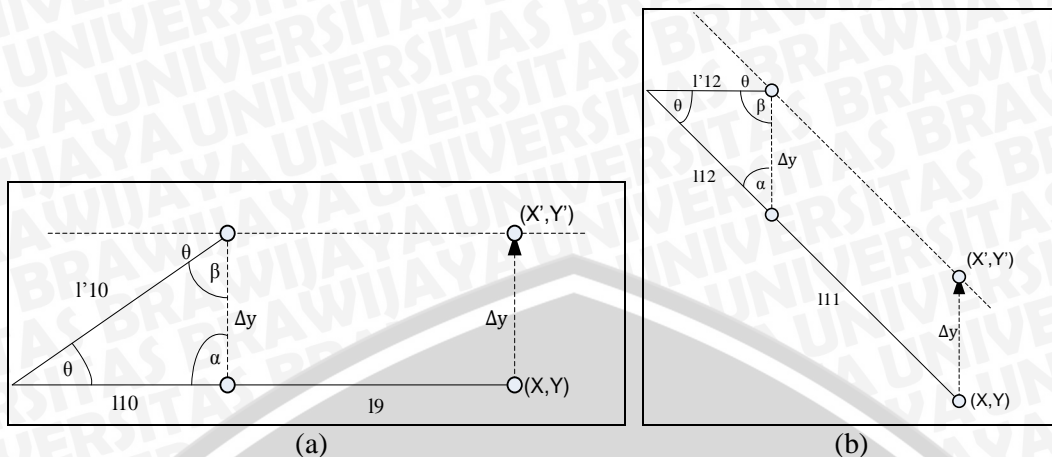
$$\tan \theta_{1.4} = \frac{\sin \alpha}{1 - \frac{\Delta y}{l_8} \cos \alpha}$$

$$\theta_{1.4} = \tan^{-1} \left( \frac{\Delta y * \sin \alpha}{l_8 - \Delta y * \cos \alpha} \right) \quad (3.6)$$

Dan untuk panjang kaki setelah bergerak di dapatkan

$$\frac{l'_8}{\sin \alpha} = \frac{\Delta y}{\sin \theta_{1.4}} \quad (3.7)$$

$$l'_8 = \frac{\Delta y * \sin \alpha}{\sin \theta_{1.4}} \quad (3.8)$$



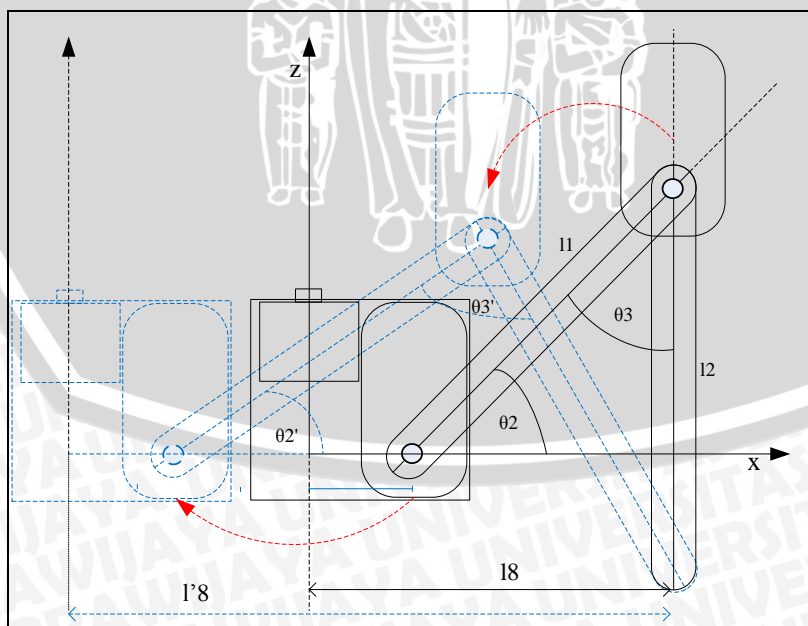
Gambar 3.5 Pergeseran Titik Pusat  
(a) terhadap Kaki Tengah, (b) terhadap Kaki Depan.

Karena konstruksi tiap kaki sama dan memiliki perbedaan dalam penempatan servo pangkal terhadap titik pusat yaitu pada \$(\alpha)\$ sehingga didapatkan untuk kaki tengah

$$\theta_{1.5} = \tan^{-1} \left( \frac{\Delta y \cdot \sin \alpha}{l_{10} - \Delta y \cdot \cos \alpha} \right) \quad (3.9)$$

Dan untuk kaki depan

$$\theta_{1.6} = \tan^{-1} \left( \frac{\Delta y \cdot \sin \alpha}{l_{12} - \Delta y \cdot \cos \alpha} \right) \quad (3.10)$$



Gambar 3.6 Bentuk Konstruksi Kaki Robot

Posisi awal tumpuan robot dapat di bentuk dengan dua sudut yaitu  $\theta_2$  dan  $\theta_3$ . Setelah di ketahui panjang untuk l'8 seperti perhitungan sebelumnya maka dapat menghitung sudut  $\theta_2$  dan  $\theta_3$ . Untuk menghitung sudut  $\theta_2$  dan  $\theta_3$  menggunakan persamaan sesuai dengan kinematika dua sendi yang telah dapatkan sebelumnya, untuk sendi  $\theta_2$  dan untuk kaki  $\theta_3$

$$\theta'_2 = \tan^{-1} \left( \frac{y(l_1+l_2\cos\theta_3)-x\cdot l_2\sin\theta_3}{x(l_1+l_2\cos\theta_3)+y\cdot l_2\sin\theta_3} \right) \quad (3.11)$$

$$\theta'_3 = \cos^{-1} \left( \frac{x^2+y^2-l_1^2-l_2^2}{2 l_1 l_2} \right) \quad (3.12)$$

Karena pada servo pinggang dan servo lutut terdapat jarak sebesar  $D = 2\text{cm}$  dan bidang XZ mengalami rotasi terhadap sumbu Z sebesar  $\theta_1$  maka untuk perhitungan  $\theta_2$  dan  $\theta_3$  berubah menjadi,

$$\theta'_2 = \arctan \left( \frac{z(l_1+l_2\cos\theta_2)-(|x'|-2)\cdot l_2\sin\theta_2}{(|x'|-2)(l_1+l_2\cos\theta_2)+z\cdot l_2\sin\theta_2} \right) \quad (3.13)$$

$$\theta'_3 = \arccos \left( \frac{(|x'|-2)^2+z^2-l_1^2-l_2^2}{2 l_1 l_2} \right) \quad (3.14)$$

dengan

$$x' = \frac{x}{\cos\theta_1} \quad (3.15)$$

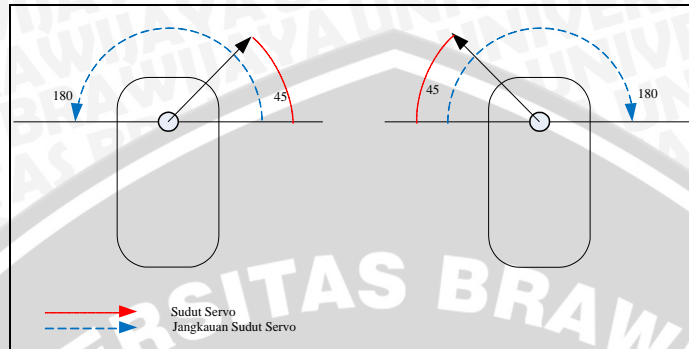
maka

$$\theta'_2 = \tan^{-1} \left( \frac{z(l_1+l_2\cos\theta_3)-\left(\frac{x}{\cos\theta_1}\right)-2\cdot l_2\sin\theta_3}{\left(\frac{x}{\cos\theta_1}\right)-2)(l_1+l_2\cos\theta_3)+z\cdot l_2\sin\theta_3} \right) \quad (3.16)$$

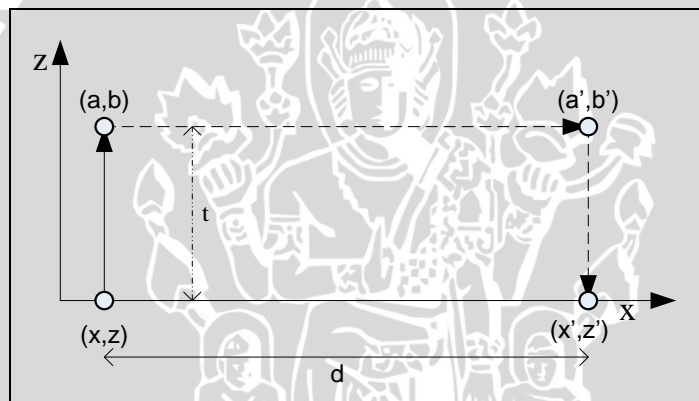
$$\theta'_3 = \cos^{-1} \left( \frac{\left(\frac{x}{\cos\theta_1}\right)-2)^2+z^2-l_1^2-l_2^2}{2 l_1 l_2} \right) \quad (3.17)$$



Karena posisi kaki pada sisi kanan merupakan pencerminan dari sisi kiri, maka hal serupa seperti di atas di lakukan kembali yaitu membalik arah putar dari servo, agar mendapatkan nilai yang sama dengan arah putar yang berbeda.



Gambar 3.7 Arah Putar Servo



Gambar 3.8 Gerak mengangkat kaki

Gambar 3.8 menunjukkan lintasan pergerakan kaki ketika di angkat, dengan ketentuan telah ditetapkan tinggi ( $t$ ) dan di ketahui titik awal ( $x, z$ ) dan titik tujuan ( $x', z'$ ), sehingga di dapatkan perhitungan

$$d = x' - x \quad (3.18)$$

Untuk mendapatkan titik ( $a, b$ )

$$a = x + t \quad (3.19)$$

$$b = z \quad (3.20)$$

Untuk mendapatkan titik ( $a', b'$ )

$$a' = a + d \quad \text{dimana,} \quad (3.21)$$

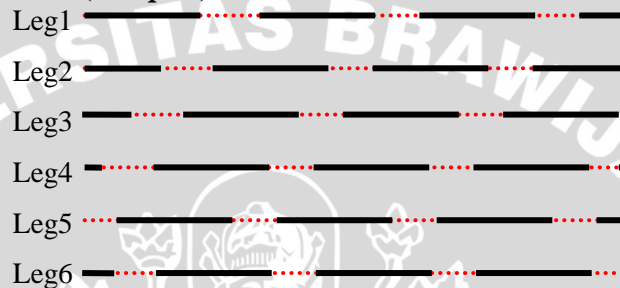
$$b' = z' \quad (3.22)$$

Kemudian kembali ke titik ( $x', z'$ ) yang sudah di tentukan sebelumnya

#### 4) Penentuan desain pergerakan robot

Perancangan pergerakan robot atau algoritma *gait* disini adalah semua gerakan masing – masing kaki adalah sama, yang berbeda adalah waktu tunda atau beda fase antar masing masing kaki (Woering R,2011). Terdapat tiga jenis *gait* umum yang dapat di aplikasikan, yaitu *gait pentapod*, *tetrapod* dan *tripod* sedangkan untuk penelitian ini di gunakan *gait tetrapod*. Urutan langkah kaki terdapat pada gambar 3.9.

##### Gait (*tetrapod*)

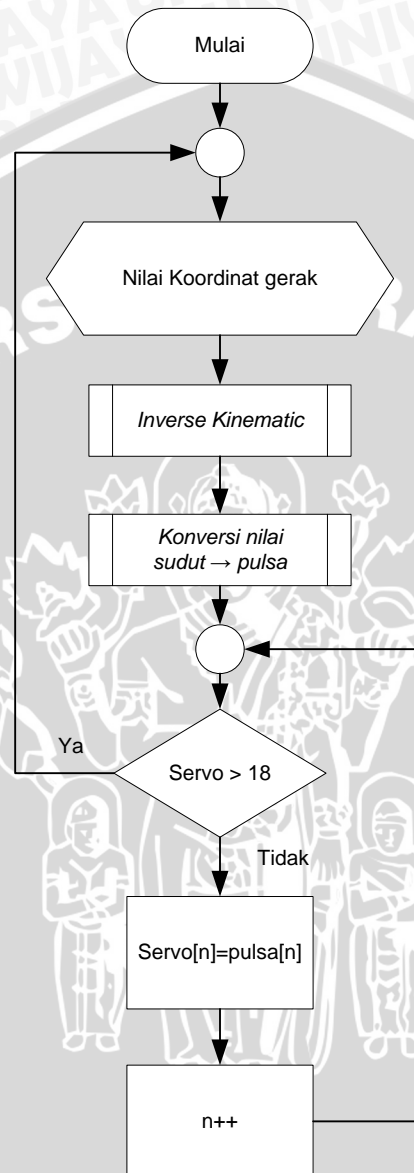


Gambar 3.9 Diagram Gait

Dalam gambar 3.9 dapat dilihat *gait tetrapod* menunjukkan semua kaki di gerakan menurut diagram gait yang telah di buat, dimana garis hitam tebal menunjukkan ketika fase kaki sedang menapak di atas alas, dan garis putus-putus merah merupakan fase kaki di angkat dan mengayun agar dapat kembali ke titik awal. Semua kaki tidak bekerja secara bersamaan tetapi bergantian, ada dua buah kaki yang diangkat sedang empat yang lainnya menapak pada alas. Hal ini dimaksudkan agar robot tetap memiliki keseimbangan pada saat bergerak. Ketika fase kedua kaki mengayun ke depan atau kembali ke titik awal, kaki yang sedang menapak pada alas tidak hanya diam, tetapi motor servo pada bagian bahu juga ikut bergeser mengikuti badan robot yang bergerak ke arah depan dan titik tumpu untuk setiap kaki tidak berubah. Hal ini di maksudkan agar robot dapat bergerak secara efisien.

### 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 STM32F4 dtunjuk dalam gambar 3.10.



Gambar 3.10. Diagram Alir Program Utama Mikrokontroler Kontrol Servo

Proses diawali dengan pemberian masukan yang berupa koordinat untuk pergerakan dari robot hexapod. Nilai koordinat tersebut kemudian diolah dengan rumus inverse kinematic yang telah di dapatkan dari perhitungan perancangan dan menghasilkan sudut untuk motor servo. Dari hasil sudut sudut yang telah



didapatkan tersebut akan dirubah menjadi sinyal PWM dan didistribusikan kepada motor servo.

### 3.3 Pengujian Alat

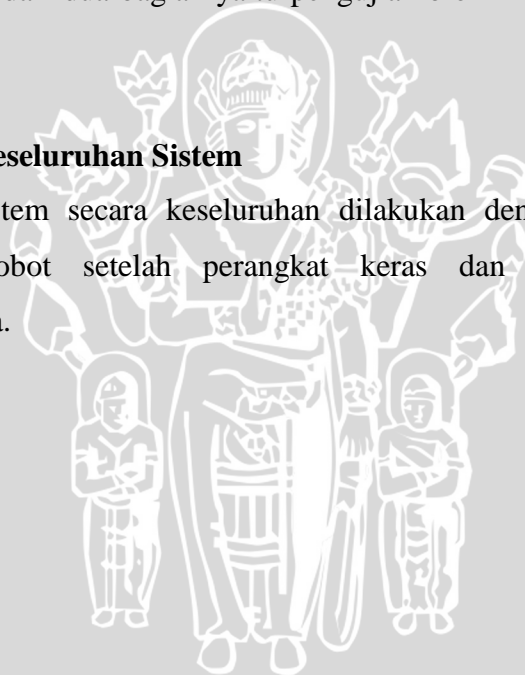
Untuk memastikan bahwa sistem ini berjalan sesuai yang direncanakan maka perlu dilakukan pengujian alat meliputi perangkat keras (*hardware*) yang dilakukan baik per blok maupun keseluruhan sistem.

#### 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 agar tidak terjadi kesalahan ketika keseluruhan sistem bekerja bersama. Pengujian blok terdiri dari dua bagian yaitu pengujian blok mikro kontroller dan pengujian satu kaki.

#### 3.3.2 Pengujian Keseluruhan Sistem

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

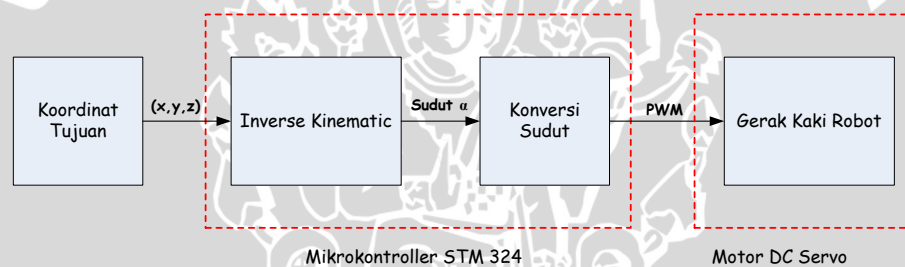


## BAB IV PERANCANGAN DAN PEMBUATAN ALAT

Bab ini menjelaskan tentang perancangan dan pembuatan *hexapod robot* mulai dari diagram blok sistem, desain mekanik robot, perancangan gerakan, perancangan perangkat keras, dan perancangan perangkat lunak. Perancangan dan pembuatan dilakukan secara bertahap dan sistematis, sehingga nantinya akan memudahkan dalam analisis sistem.

### 4.1 Penentuan Spesifikasi alat

Diagram blok perancangan system ini terdiri atas beberapa bagian, yaitu masukan berupa koordinat, Mikrokontroler STM 32F4 sebagai pengendali utama sistem, motor DC servo sebagai aktuator atau penggerak robot. Diagram blok alat ditunjukkan dalam Gambar 4.1.



Gambar 4.1. Diagram Blok Sistem *Hexapod robot*

Prinsip kerja sistem ini pertama pemberian input berupa titik koordinat  $(x,y,z)$  dimana  $y$  sebagai titik perubahan robot ke arah depan atau maju. Koordinat  $x$  dalam system ini tidak digunakan karena robot hanya bergerak maju, dan koordinat  $x$  mempengaruhi pergerakan kekanan atau kiri maupun menyerong. Sedangkan koordinat  $z$  mempengaruhi ketinggian dari body robot itu sendiri. Koordinat  $(x,y,z)$  kemudian dikirim ke mikrokontroler. Titik koordinat akan diolah menggunakan perhitungan inverse kinematic supaya didapatkan nilai sudut  $\alpha$  untuk masing motor servo. Tetapi nilai sudut tidak dapat langsung di kirimkan kepada motor servo karena masukan motor servo berupa pulsa PWM, sehingga nilai sudut  $\alpha$  akan di konversi oleh mikrokontroler menjadi PWM. Setelah sudut dapat di konversi menjadi PWM, sinyal PWM dari mikrokontroler

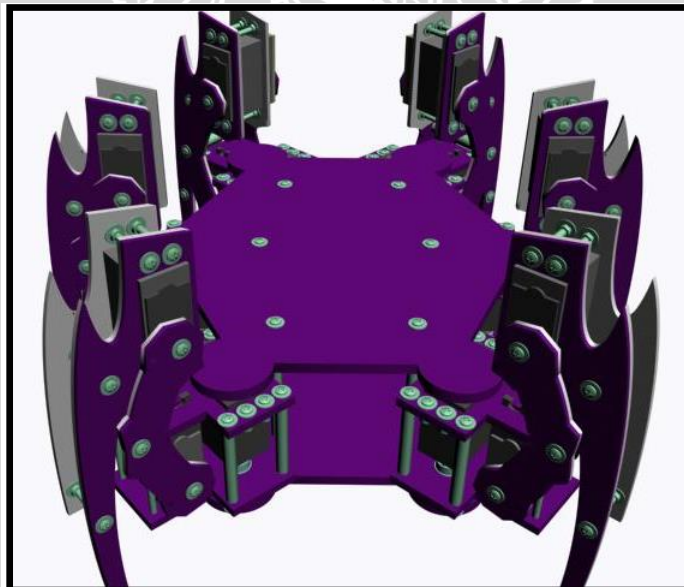
kemudian dikirim ke motor servo dan motor servo dapat bergerak sesuai dengan sudut yang di inginkan.

#### 4.2 Perancangan Mekanik Robot

Berdasarkan peraturan Kontes Robot Pemadam Api Indonesia (KRPAI) Divisi Berkaki tahun 2013, batasan dimensi robot baik saat posisi berhenti maupun saat posisi berjalan adalah sebagai berikut:

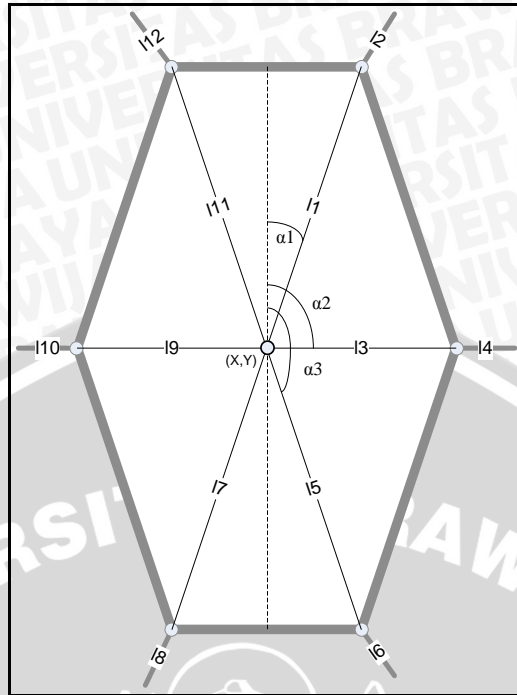
- Panjang maksimum : 46 cm
- Lebar maksimum : 31 cm
- Tinggi maksimum : 27 cm

Dengan adanya peraturan tersebut, maka desain robot berkaki enam ini harus dirancang agar tidak melebihi batas ukuran yang telah ditetapkan. Gambar desain mekanik robot berkaki enam ditunjukkan dalam Gambar 4.2.



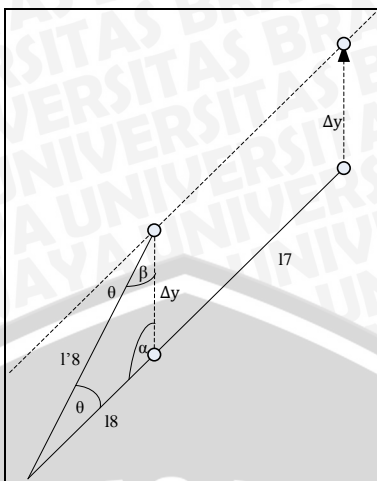
Gambar 4.2. Perspektif Desain Mekanik *Hexapod robot*





Gambar 4.3 Bentuk Konstruksi Badan Robot

Gambar 4.3 merupakan tampak atas dari robot. Terdapat titik pusat robot yang terdapat di tengah badan robot dan lebar sudut dari masing-masing kaki terhadap titik pusat robot dengan kaki depan sebesar  $30^0$ , kaki tengah sebesar  $90^0$  dan kaki belakang sebesar  $150^0$ . Sedangkan panjang titik pusat kaki dari titik pusat badan robot untuk kaki depan sepanjang  $9,057\text{cm}$ , kaki tengah sepanjang  $6,9\text{cm}$ , dan jarak kaki belakang sepanjang  $9,057\text{cm}$ . Dan panjang kaki dari titik pusat kaki hingga ujung kaki sepanjang  $9,5\text{cm}$ . Semua angka ini di gunakan sebagai variable konstan untuk perhitungan inverse kinematic. Karena pada sudut pangkal berubah terhadap titik pusat robot sehingga di dapatkan perhitungan



Gambar 4.4 Pergeseran titik pusat robot terhadap kaki belakang

Untuk gambar 4.4 telah didapatkan persamaan (3.6) dengan adanya variabel yg telah ditetapkan persamaan dapat disederhanakan menjadi

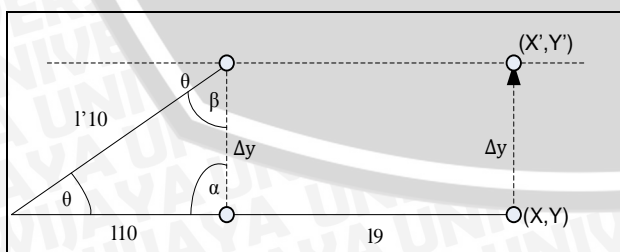
$$\theta_{1.4} = \tan^{-1} \left( \frac{\Delta y \cdot \sin \alpha}{l_8 - \Delta y \cdot \cos \alpha} \right)$$

$$\theta_{1.4} = \tan^{-1} \left( \frac{\Delta y \cdot 0,5}{9,5 + \Delta y \cdot 0,87} \right) \tag{4.1}$$

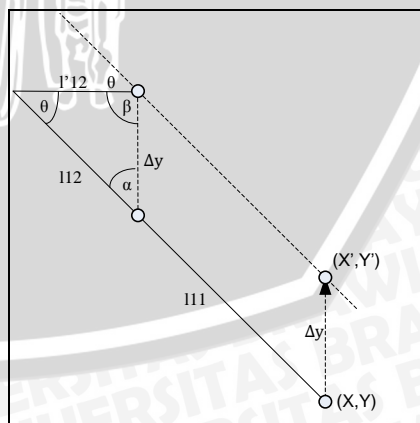
Dan untuk panjang kaki setelah bergerak sesuai dengan rumus (3.8) di dapatkan

$$l'_8 = \frac{\Delta y \cdot \sin \alpha}{\sin \theta_{1.4}}$$

$$l'_8 = \frac{\Delta y \cdot 0,5}{\sin \theta_{1.4}} \tag{4.2}$$



(a)



(b)

Gambar 4.5 Pergeseran Titik Pusat

(a) terhadap Kaki Tengah, (b) terhadap Kaki Depan.

Sedangkan untuk konstruksi pergerakan untuk kaki tengah didapatkan persamaan (3.9)

$$\theta_{1.5} = \tan^{-1} \left( \frac{\Delta y * \sin \alpha}{l_{10} - \Delta y * \cos \alpha} \right)$$

$$\theta_{1.5} = \tan^{-1} \left( \frac{\Delta y}{9,5} \right) \quad (4.3)$$

Dan untuk kaki depan di dapatkan (3.10)

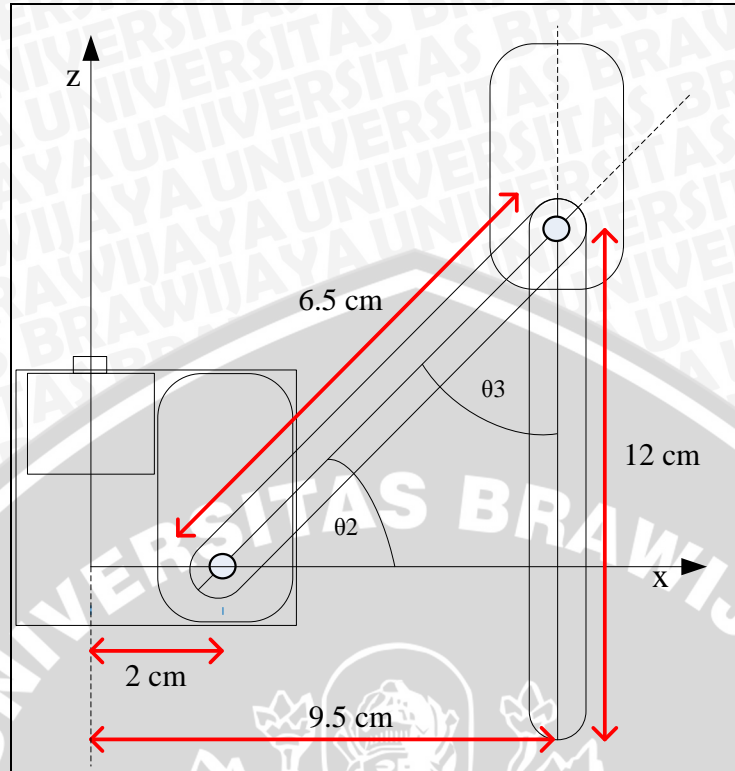
$$\theta_{1.6} = \tan^{-1} \left( \frac{\Delta y * \sin \alpha}{l_{12} - \Delta y * \cos \alpha} \right)$$

$$\theta_{1.6} = \tan^{-1} \left( \frac{\Delta y * 0,5}{9,5 - \Delta y * 0,87} \right) \quad (4.4)$$

Karena kaki pada sisi kiri merupakan pencerminan dari sisi kanan maka hasil sudut  $\theta_{1.4}$ ,  $\theta_{1.5}$ , dan  $\theta_{1.6}$  maka untuk sisi kiri adalah  $\theta_{1.1} = 180 - \theta_{1.6}$  untuk kaki depan,  $\theta_{1.2} = 180 - \theta_{1.5}$  untuk kaki tengah dan  $\theta_{1.3} = 180 - \theta_{1.4}$  untuk kaki belakang. Motor DC servo yang digunakan memiliki jangkauan sudut putar terbatas, yaitu antara  $0^\circ$  sampai  $180^\circ$ . Oleh karena itu untuk hasil sudut yang negatif, sudut putar servo harus dibalik yaitu dengan cara membalik arah putar servo dengan program servo.







Gambar 4.6 Bentuk Konstruksi Kaki Robot

Gambar 4.6 merupakan tampak konstruksi kaki robot. Terdapat panjang lengan yang menghubungkan servo lutut dengan servo kaki (11) dengan panjang 6,5cm, panjang kaki 12. Jarak kaki dari titik pusat kaki hingga ujung kaki sepanjang 9,5cm. dan jarak antara servo pinggang dan pusat servo lutut sebesar 2cm. Semua angka ini di gunakan sebagai variable konstan untuk perhitungan inverse kinematic untuk kaki-kaki. Untuk menghitung sudut  $\theta_2$  dan  $\theta_3$  menggunakan persamaan sesuai dengan kinematika dua sendi yang telah dapatkan sebelumnya pada persamaan (3.16) untuk sendi  $\theta_2$  dan (3.17) dan untuk kaki  $\theta_3$

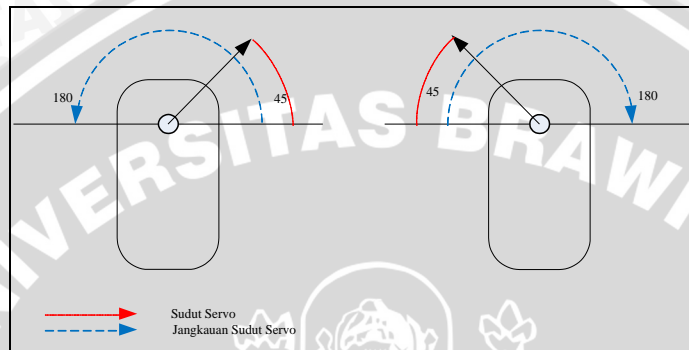
$$\theta'_2 = \tan^{-1} \left( \frac{z(l_1 + l_2 \cos \theta_3) - \left(\frac{x}{\cos \theta_1} - 2\right) \cdot l_2 \sin \theta_3}{\left(\frac{x}{\cos \theta_1} - 2\right)(l_1 + l_2 \cos \theta_3) + z \cdot l_2 \sin \theta_3} \right)$$

$$\theta'_2 = \tan^{-1} \left( \frac{z(6,5 + 12 \cos \theta_3) - \left(\frac{x}{\cos \theta_1} - 2\right) \cdot 12 \sin \theta_3}{\left(\frac{x}{\cos \theta_1} - 2\right)(6,5 + 12 \cos \theta_3) + z \cdot 12 \sin \theta_3} \right) \quad (4.5)$$

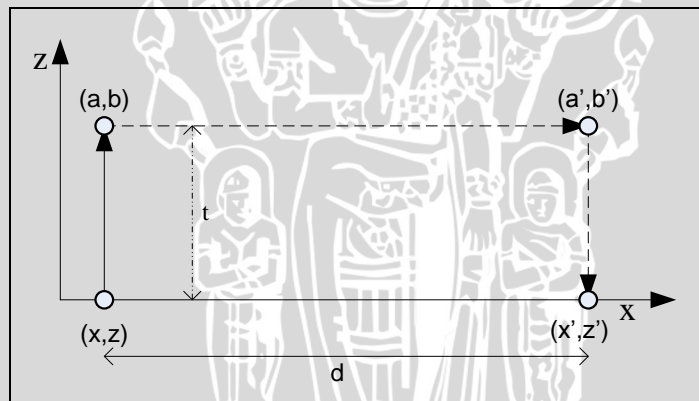
$$\theta'_3 = \cos^{-1} \left( \frac{\left(\frac{x}{\cos \theta_1} - 2\right)^2 + z^2 - l_1^2 - l_2^2}{2 l_1 l_2} \right)$$

$$\theta'_3 = \cos^{-1} \left( \frac{(\frac{x}{\cos \theta_1} - 2)^2 + z^2 - 42,25 - 144}{156} \right) \quad (4.6)$$

Karena posisi kaki pada sisi kanan merupakan pencerminan dari sisi kiri, maka hal serupa seperti di atas di lakukan kembali yaitu membalik arah putar dari servo, agar mendapatkan nilai yang sama dengan arah putar yang berbeda menggunakan HPP-21.



Gambar 4.7 Arah Putar Servo



Gambar 4.8 Gerak mengangkat kaki

Gambar 4.8 menunjukkan lintasan pergerakan kaki ketika di angkat, dengan ketentuan telah ditetapkan tinggi 2 cm dan di ketahui titik awal (x,z) dan titik tujuan (7,10), sehingga di dapatkan perhitungan sesuai persamaan (3.19),(3.20),(3.21), dan (3.22)

Untuk mendapatkan titik (a,b)

$$a = x + 2$$

$$b = z$$

Untuk mendapatkan titik (a',b')

$$a' = a + d \quad \text{dimana,}$$

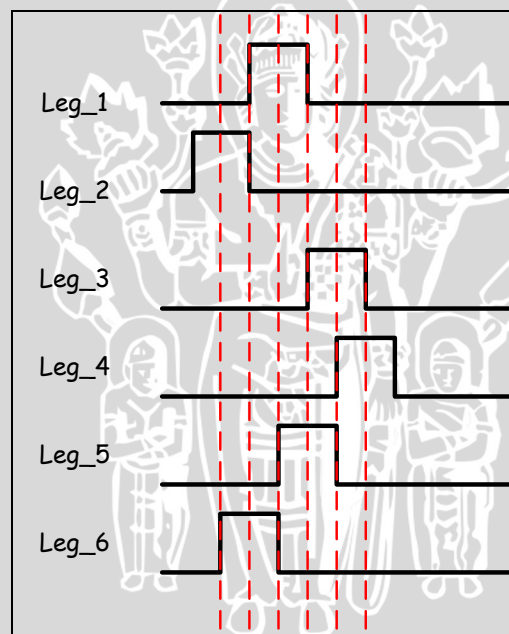
$$b' = z'$$

Kemudian kembali ke titik (x',z') yang sudah di tentukan sebelumnya.

### 4.3 Pola Langkah

Hasil perhitungan *inverse kinematic* pergerakan robot sudah dapat dibangkitkan, dan kaki robot sudah dapat melangkah. Pada bagian ini algoritma *gait* diperlukan untuk mengatur kapan sebuah kaki harus diangkat, digeser atau menapak. Urutan langkah kaki terdapat pada gambar 4.9

Gait (tetrapod)



Gambar 4.9 Diagram Gait

Pada penelitian untuk robot hexapod ini di gunakan pola langkah empat kaki menapak pada alas dan dua kaki di angkat. Penunjukkan untuk setiap leg adalah sebagai berikut, yaitu leg\_1 adalah kaki kanan depan, leg\_2 adalah kaki kanan tengah, leg\_3 adalah kaki kanan belakang, leg\_4 adalah kaki kiri depan, leg\_5 adalah kaki kiri tengah dan leg\_6 adalah kaki kiri belakang. Dalam diagram gait dapat dilihat membentuk seperti pulsa yang menunjukkan ketika berlogika tinggi berarti kaki harus diangkat dan ketika berlogika rendah kaki menapak pada

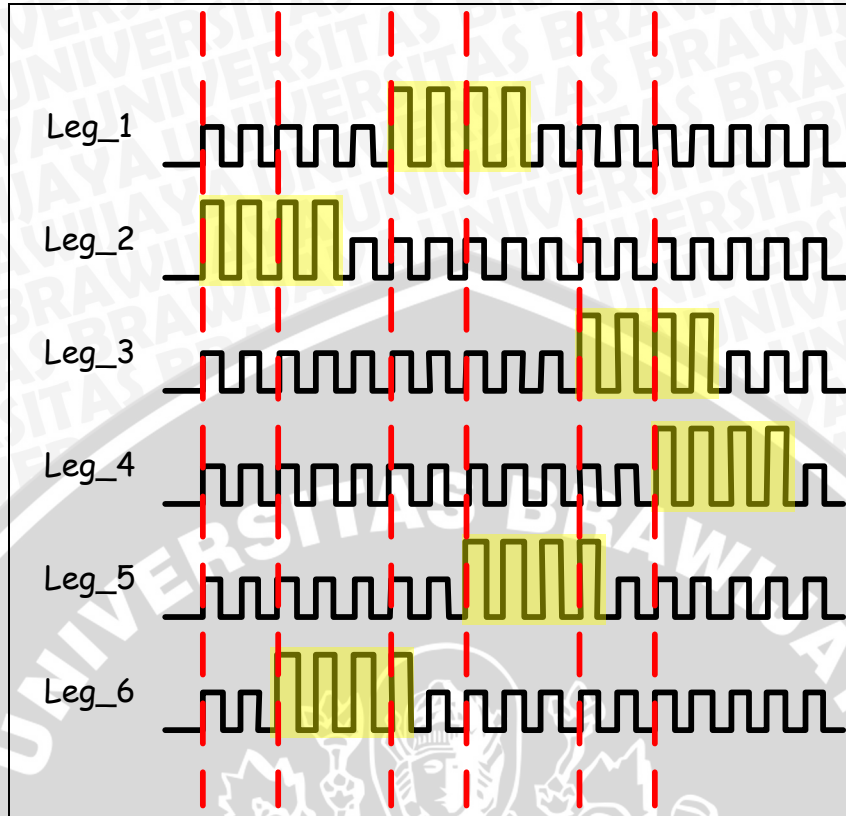


alas. Di dalam gambar juga di jelaskan bahwa gerak robot diawali dari leg\_2 mengangkat kaki sebelum proses mengangkat pada leg\_2 selesai segera disusul oleh leg\_6 untuk mengikat kaki, selama leg\_2 dan leg\_6 mengayun, kaki kaki lain yang menapak pada alas tetap bergerak sesuai arah gerak robot, tetapi hanya pada motor servo bagian bahu sedangkan yang lain tetap untuk mempertahankan titik tumpu pada masing masing kaki. Setelah proses leg\_2 selesai mengangkat kaki kemudian leg\_2 akan menapakkan kaki ke posisi awal titik tumpuan, setelah itu leg\_1 atau kaki kanan depan akan memasuki fase mengangkat kaki, setelah berjalan kaki kiri belakang atau leg\_6 akan menapakkan ujung kaki ke titik awal tumpuan. Proses ini akan berjalan hingga leg terakhir pada leg\_4 atau kaki kiri depan menapakkan pada alas. Dalam diagram gait adalah satu gerak secara keseluruhan. Untuk menghasilkan gerak maju yang harmonis, gait ini akan di kerjakan secara terus menerus.

Pada fase menapak dalam program dapat dituliskan sebagai berikut,

```
inv_kine(front_left, (beta[4]*0.231), (leg_next[4]*0.231), z);
inv_kine(front_right, (beta[1]*0.796), (leg_next[1]*0.796), z);
inv_kine(middle_left, (beta[5]*0.615), (leg_next[5]*0.615), z);
inv_kine(middle_right, beta[2], leg_next[2], z);
inv_kine(rear_left, beta[6], leg_next[6], z);
inv_kine(rear_right, beta[3]*0.385), (leg_next[3]*0.385), z);
```

Dalam satu sub gait yang berarti front\_left akan menggerakkan kaki kiri depan, beta[x] adalah sudut untuk motor servo bahu yang didapatkan dari persamaan (4.4), leg\_next[x] adalah panjang kaki setelah perubahan yang didapatkan dari persamaan (4.2) dan z adalah tinggi robot sesuai dari masukkan. Konstanta pengali (0.231, 0.796, 0.615, 0.385) yang terdapat di belakang beta[x] dan leg\_next[x] di maksudkan agar motor servo dapat bergerak perlahan untuk menuju titik yang akan dituju secara perlahan dan harmonis dengan kaki kaki yang lain.



Gambar 4.10 Diagram Gait

Konstanta didapatkan dari perancangan pola langkah dan didapatkan urutan sebanyak 17 kali gerakan seperti yang ditunjukkan dalam gambar 4.10, diantaranya 4 kali bergerak ketika posisi kaki diangkat dan 13 kali bergerak untuk kaki menapak dan bergeser. Untuk mendapatkan nilai konstanta pengali digunakan persamaan

$$konstanta_n = \frac{n}{13}$$

dengan  $n = \text{pulsa high / gerak}$

sehingga gerak ke 1 akan di dapatkan  $konstanta_1 = \frac{1}{13}$

$$konstanta_1 = 0,077$$

Dari 13 kali gerakan inilah kaki di gerakan setiap 1/13 untuk mendapatkan gerak yang harmonis antar kakinya, sehingga didiapatkan hasil konstanta pengali untuk setiap geraknya seperti dalam table 4.1,

Tabel 4.1 Konstanta Gerak Kaki

Gerak ke		Konstanta	Gerak ke		Konstanta
1	1/13	0,077	7	7/13	0,538
2	2/13	0,154	8	8/13	0,615
3	3/13	0,231	9	9/13	0,692
4	4/13	0,307	10	10/13	0,796
5	5/13	0,385	11	11/13	0,846
6	6/13	0,461	12	12/13	0,923

Sedangkan untuk memasukkan nilai konstanta pada pergerakan, dengan cara mengalikan dengan hasil beta yang sudah didapatkan sebelumnya, sehingga koordinat nilai x akan menjadi

$$x = (\text{beta}[\text{kaki}] * \text{konstanta}_1)$$

Pada fase mengangkat dapat dituliskan,

```
inv_kine(front_left, (beta[4]*0.231), (leg_next[4]*0.231), z);
inv_kine(front_right, (beta[1]*0.796), a', b');
inv_kine(middle_left, (beta[5]*0.615), (leg_next[5]*0.615), z);
inv_kine(middle_right, beta[2], leg_next[2], z);
inv_kine(rear_left, beta[6], a, b);
inv_kine(rear_right, beta[3]*0.385), (leg_next[3]*0.385), z);
```

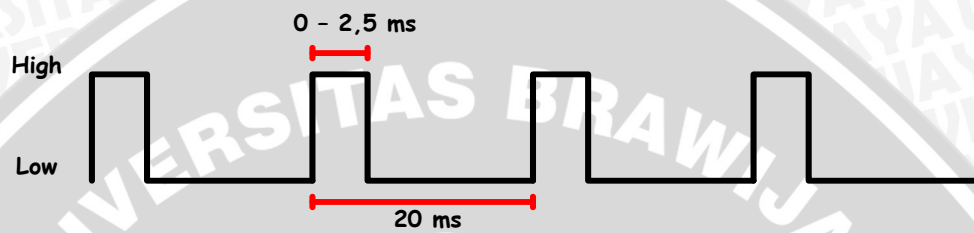
Dalam sub gait ini dapat terlihat pada kaki kanan depan atau front\_right terdapat koordinat nilai ((beta[1]\*0.796)a',b') dan pada kaki kiri belakang terdapat koordinat (beta[6],a,b). Nilai dari (a,b) dan (a',b') didapatkan dari persamaan (3.19),(3.20) dan(3.21),(3.22).

#### 4.4 Kontrol Multi Servo

Motor DC servo membutuhkan pulsa periodik agar dapat berputar sesuai dengan posisi yang diinginkan. Untuk mempertahankan posisi atau sudutnya, pulsa periodik ini harus diberikan secara berkala dan lebar pulsanya tetap. Apabila pulsa pemberian pulsa tidak secara berkala, motor Dc servo ini tidak akan berfungsi atau mati. Matinya motor akan menyebabkan kinerja robot terganggu

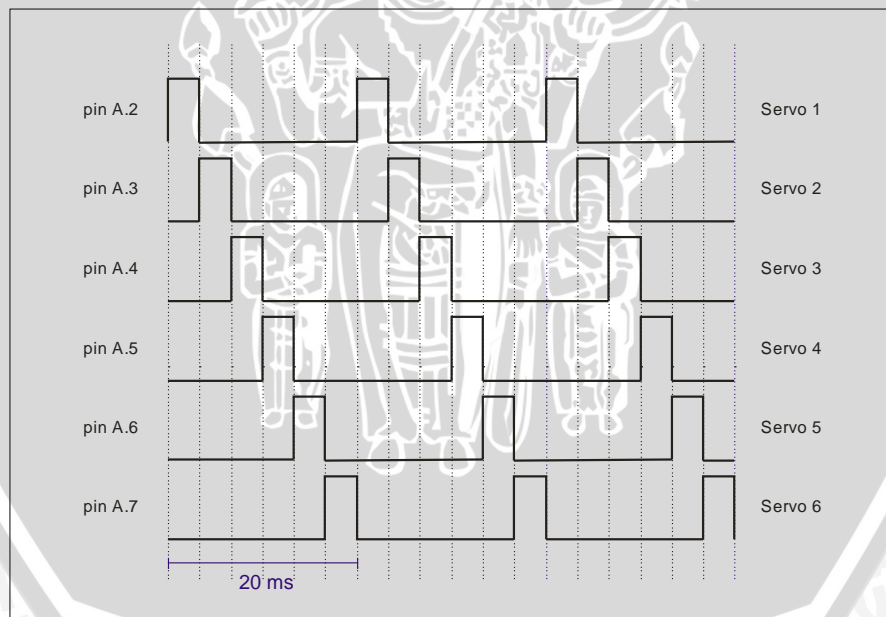


karena tiap sendi pada robot tidak dapat mempertahankan posisinya. Apabila lebar pulsa yang diberikan berubah-ubah maka posisi atau sudut motor DC servo yang dihsilkan juga berubah-ubah akibatnya gerakan robot akan tidak teratur. Untuk menghindari hal tersebut, mikrokontroler sebagai kontroler motor DC servo harus dapat menyediakan pulsa periodik secara terus-menerus dengan lebar yang sama untuk sudut tertentu kepada motor DC servo. Pulsa yang diberikan harus memiliki periode 20 ms. Pulsa periodik motor DC servo ditunjukkan dalam gambar 4.11.



Gambar 4.11 Sinyal Kontrol Motor DC Servo

Sebuah mikrokontroler memiliki fungsi timer yang dapat digunakan untuk menciptakan pulsa periodik untuk kontrol servo.



Gambar 4.12 Sinyal Kontrol Motor DC Servo

Garis putus-putus dalam gambar 4.12 menunjukkan interupsi yang dilakukan oleh timer setiap 5 ms. Gambar 4.12 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* pada *timer* TIM3 dengan 16 bit. Dengan metode seperti dalam gambar 4.13, dapat dilakukan pengontrolan sebanyak 18 motor DC servo.

Lebar sinyal *high* pada awal pulsa periodik diatur oleh *Output Compare* pada timer 16 bit. Perhitungan nilai pada *Output Compare* diperoleh dari datasheet sebagai berikut.

$$T = \frac{TIM\_Period}{N}$$

$$T = \frac{T_{oc}}{N}$$

dengan  $TIM\_Period$  = Bit Timer

$T$  = Periode pulsa servo

$N$  = *Prescaler*

$T_{oc}$  = Nilai pada *Output Compare*

Sedangkan untuk mengkonversi nilai sudut menjadi periode untuk lebar pulsa PWM digunakan rumus persamaan garis lurus melalui dua titik

$$\frac{y-y_1}{y_2-y_1} = \frac{x-x_1}{x_2-x_1} \quad (4.7)$$

Karena sudut  $35^\circ$  setara dengan pulsa 3615 dan sudut  $145^\circ$  setara dengan 6550 sehingga,

$$\frac{y-3616}{6550-3615} = \frac{x-35^\circ}{145^\circ-35^\circ} \quad (4.8)$$

$$x = 26,68y + 2681,14 \quad (4.9)$$

Dengan  $y$  = Periode PWM pada mode *Output Compare*

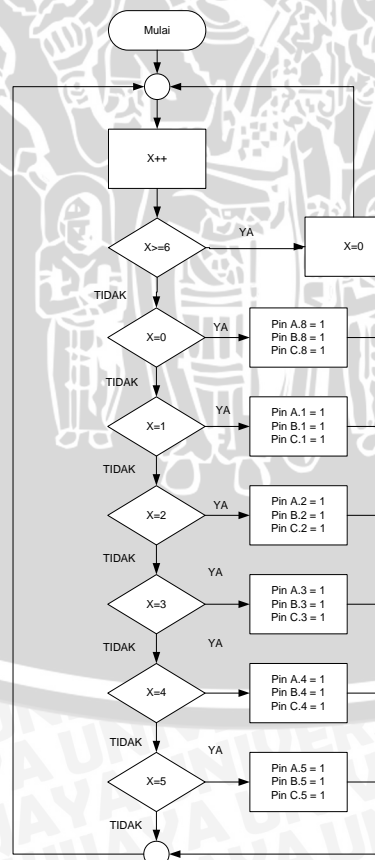
$x$  = Sudut pada motor DC servo

#### 4.4.1 Algoritma Mikrokontroler Kontrol Servo

Hexapod robot menggunakan mikrokontroler STM32F4 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 *Timer*

Timer yang digunakan adalah *Timer3–16 bit* dengan mode interupsi yang digunakan adalah *Timer3 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 3333  $\mu$ s. Diagram alir program Interupsi *Timer3* ditunjukkan dalam gambar 4.13.



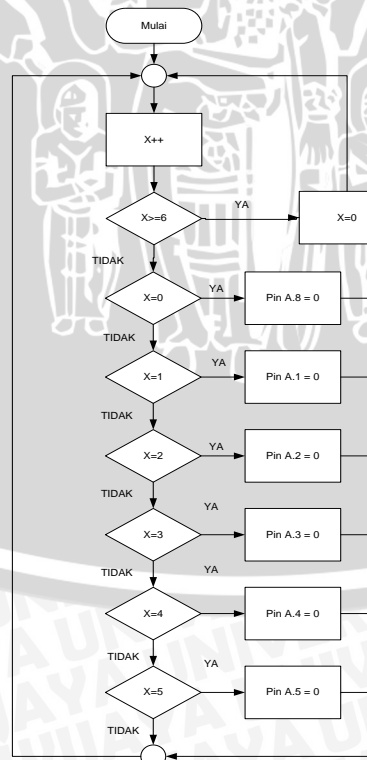
Gambar 4.13. Diagram Alir Program Interupsi *Timer3*



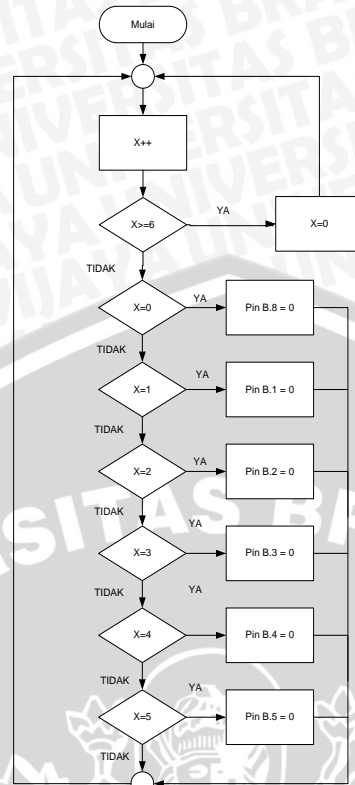
Nilai  $x$  pada awal program adalah 0. Pada saat subrutin interupsi aktif, nilai variabel  $x$  akan dicek apakah lebih dari 5, jika memenuhi maka nilai  $x$  akan dinolkan, tetapi jika tidak memenuhi maka akan masuk ke seleksi kondisi yang akan memberikan nilai 1 (logika *high*) pada pin-pin tertentu sesuai nilai  $x$ .

## 2) Interupsi *Timer Output Compare Match*

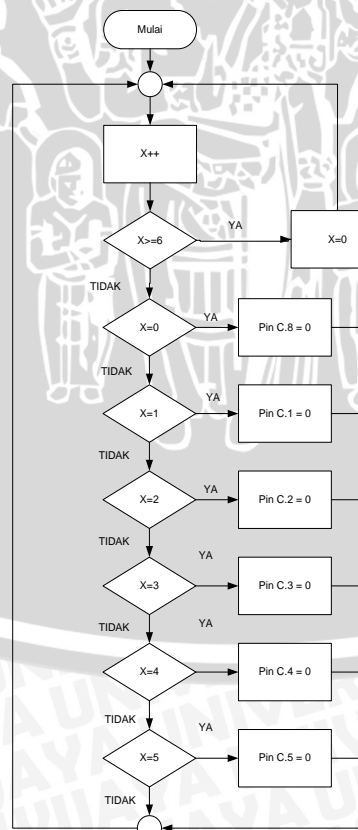
Interupsi *Timer Output Compare Match* bertujuan menentukan lebar sinyal kontrol *HIGH* pada awal sinyal kontrol servo. Isi program interupsi adalah memberi logika *low* pada pin yang mengontrol servo secara berurutan. Pada perancangan ini digunakan tiga buah OCR pada timer 16 bit, yaitu OCRA, OCRB dan OCRC. Masing-masing OCR mengontrol enam buah motor servo. Nilai OCRA, OCRB dan OCRC mempengaruhi arah pergerakan motor DC servo. diagram alir program interupsi *Timer Output Compare Match* ditunjukkan dalam gambar 4.14, 4.15. dan gambar 4.16.



Gambar 4.14. Diagram Alir Program Interupsi Timer OCRA



Gambar 4.15. Diagram Alir Program Interupsi Timer OCRB

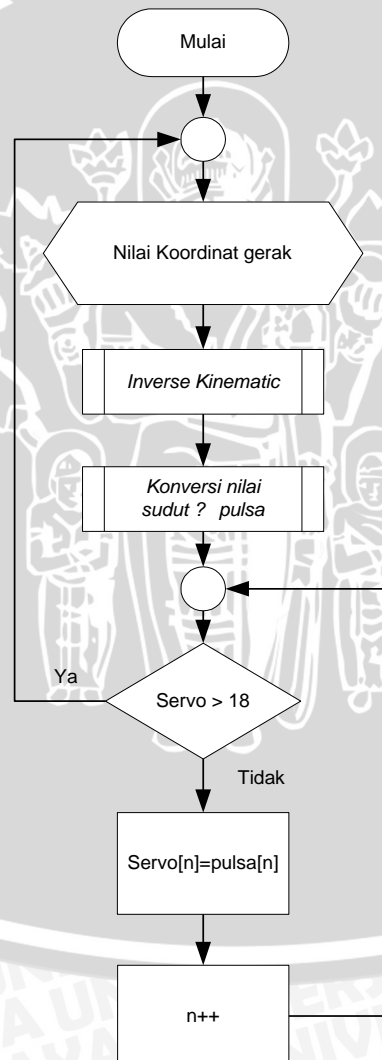


Gambar 4.16. Diagram Alir Program Interupsi Timer OCRC

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.

#### 4.5 Perancangan Perangkat Lunak

Perancangan algoritma perangkat lunak meliputi proses pengenalan port, perhitungan inverse kinematic, konversi nilai sudut menjadi pulsa atau PWM sebagai proses Bergeraknya hexapod robot dan. Algoritma program utama pada mikrokontroler kontrol servo ditunjukkan dalam gambar 4.10



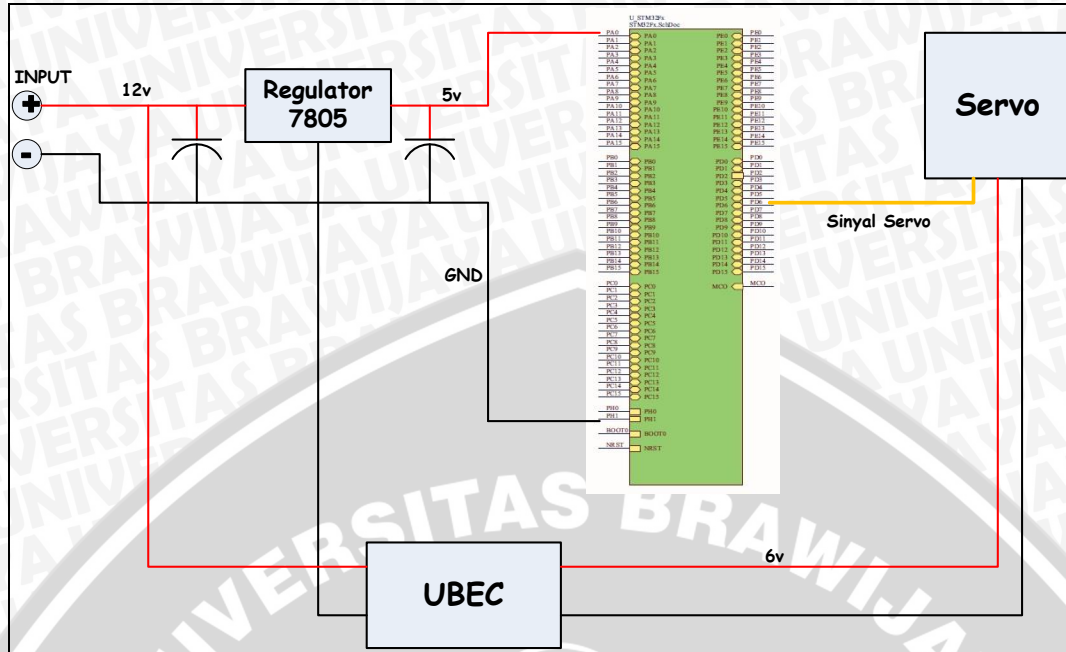
Gambar 4.17. Diagram Alir Program Utama Mikrokontroler Kontrol Servo



Pada program utama mikrokontroller, proses diawali dengan pemberian masukan yang berupa koordinat untuk pergerakan dari robot hexapod. Nilai koordinat tersebut kemudian diolah dengan rumus inverse kinematic yang telah di dapatkan dari perhitungan sebelumnya. Sesuai dengan hasil perhitungan (4.1), (4.3) dan (4.4) yang akan menghasilkan untuk sudut motor servo untuk bahu kaki depan, bahu kaki tengah dan bahu kaki belakang, sedang untuk perubahan panjang untuk setiap kaki didapatkan dari persamaan (4.2). Hasil dari perubahan panjang untuk setiap kaki akan didapatkan berupa nilai koordinat baru, dan nilai koordinat baru tersebut akan diselesaikan menggunakan persamaan (4.5) dan (4.6) dan menghasilkan sudut untuk motor servo pada lutut dan kaki. Dari hasil sudut sudut yang telah didapatkan tersebut akan didistribusikan kepada motor servo. Sebelum di distribusikan kepada motor servo, sudut harus di konversi menjadi pulsa PWM, ini dikarenakan masukan motor servo berupa PWM sesuai datasheet dari motor DC servo. Untuk merubah sudut menjadi PWM digunakan rumus persamaan garis lurus (4.9). Sinyal – sinyal PWM tersebut kemudian di distribusikan ke masing masing motor servo sesuai dengan alamat motor servo agar mendapatkan gerakan yang sesuai. Setelah sinyal PWM didistribusikan ke motor servo program akan memberikan koordinat baru sesuai dengan pola langkah yang telah di buat sebelumnya, dari pola langkah ini akan menunjukkan posisi kaki ketika kaki harus bergerak mengeser, mengangkat dan kembali kepada posisi semula. Dari pola langkah ini akan didapatkan sebuah koordinat untuk menciptakan gerakan yang baru, koordinat didap at kan dari perhitungan persamaan (3.19), (3.20) dan (3.21), (3.22) ,koordinat baru ini akan diolah dengan inverse kinematic dan menghasilkan sudut seperti penjelasan sebelumnya. Pada diagram alir program ini tidak terdapat selesai karena program berjalan terus, kecuali diberhentikan oleh user.

#### **4.6 Perancangan Sistem**

Perancangan system secara keseluruhan ini terdiri dari rangkaian catu daya sebagai suplay untuk mikrokontroler, mikrokontroler sebagai pengendali utama, modul ubec sebagai suplay untuk motor servo, dan motor servo sebagai penggerak dari robot.



Gambar 4.18. Skema Rangkaian Mikrokontroler STM32F4Discovery

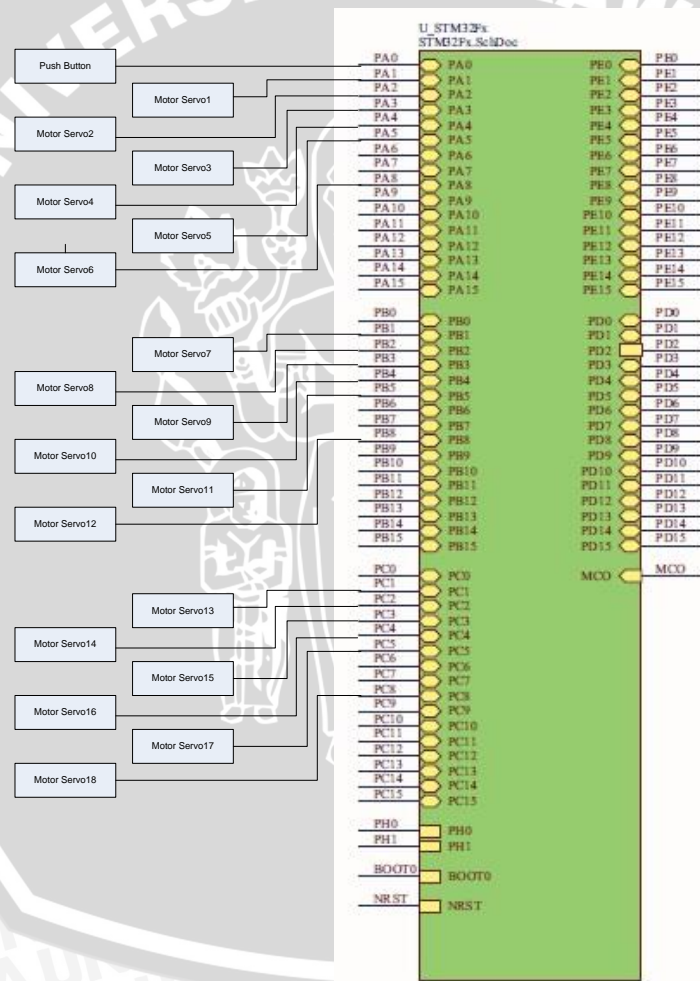
#### 4.7 Perancangan Antarmuka Mikrokontroler

Mikrokontroler STM32F4 memiliki tugas utama sebagai pengontrol motor DC servo. Pin-pin mikrokontroler ini dihubungkan langsung pada jalur sinyal pada motor servo. Tiap-tiap pin tersebut akan menghasilkan sinyal pulsa periodik sebesar 20 ms kepada servo. Pin-pin yang digunakan dalam perancangan rangkaian mikrontroller kontrol servo adalah :

- Pin A.0 : sebagai input tombol push botton untuk pemicu start
- Pin A.1 : Sebagai sinyal kontrol Servo Pinggang Kiri Depan
- Pin A.2 : Sebagai sinyal kontrol Servo Paha Kiri Depan
- Pin A.3 : Sebagai sinyal kontrol Servo Lutut Kiri Depan
- Pin A.4 : Sebagai sinyal kontrol Servo Pinggang Kiri Tengah
- Pin A.5 : Sebagai sinyal kontrol Servo Paha Kiri Tengah
- Pin A.8 : Sebagai sinyal kontrol Servo Lutut Kiri Tengah
- Pin B.1 : Sebagai sinyal kontrol Servo Pinggang Paha Kiri Belakang
- Pin B.2 : Sebagai sinyal kontrol Servo Paha Kiri Belakang
- Pin B.3 : Sebagai sinyal kontrol Servo Lutut Kiri Belakang
- Pin B.4 : Sebagai sinyal kontrol Servo Pinggang Kanan Depan



- Pin B.5 : Sebagai sinyal kontrol Servo Paha Kanan Depan
- Pin B.8 : Sebagai sinyal kontrol Servo Lutut Kanan Depan
- Pin C.1 : Sebagai sinyal kontrol Servo Pinggang Kanan Tengah
- Pin C.2 : Sebagai sinyal kontrol Servo Paha Kanan Tengah
- Pin C.3 : Sebagai sinyal kontrol Servo Lutut Kanan Tengah
- Pin C.4 : Sebagai sinyal kontrol Servo Pinggang Kanan Belakang
- Pin C.5 : Sebagai sinyal kontrol Servo Paha Kanan Belakang
- Pin C.8 : Sebagai sinyal kontrol Servo Lutut Kanan Belakang



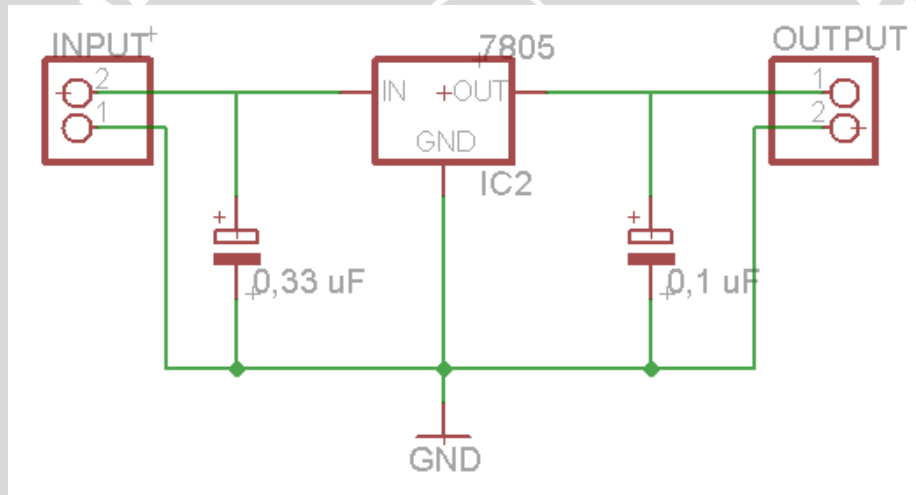
Gambar 4.19. Konfigurasi I/O STM32F4Discovery



#### 4.8 Perancangan Catu Daya Mikrokontroler

Robot ini menggunakan dua jenis catu daya, yaitu satu catu daya 5 V untuk rangkaian mikrokontroler utama. Sumber catu daya yang dipakai adalah dua buah baterai lipho (*lithium polimer*) 11,1 V.

Mikrokontroler STM32F4 bekerja pada tegangan 3 V hingga 3.3 V. Tetapi di dalam development board STM32F4Discovery sudah terdapat regulator untuk 3V sehingga hanya diperlukan satu jenis tegangan untuk menyuplai semua rangkaian. Pada perancangan digunakan catu daya sebesar 5V yang diperoleh dari rangkaian *Fixed Output Regulator* pada datasheet LM7805. Pada rangkaian digunakan regulator LM7805 agar diperoleh tegangan keluaran yang bisa diatur supaya nilai keluaran regulator bisa mendekati 5V. Skema rangkaian catu daya ditunjukkan dalam Gambar 4.18.



Gambar 4.20. Rangkaian Catu Mikrokontroler

## BAB V

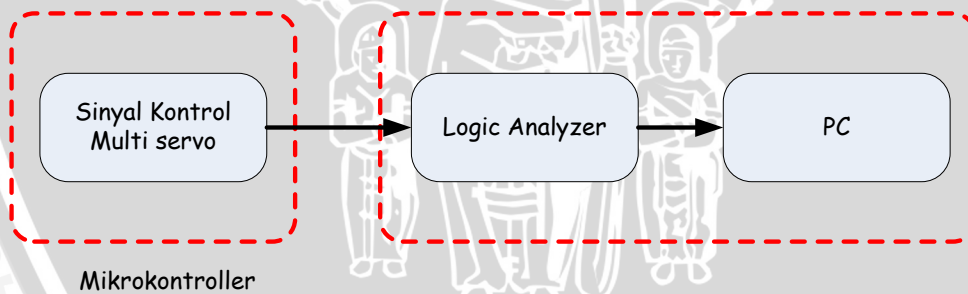
### PENGUJIAN DAN ANALISIS

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:

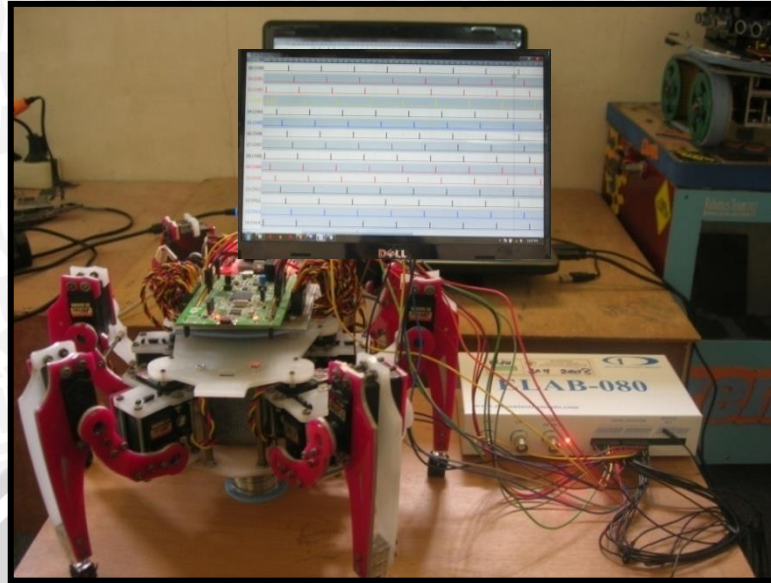
- 1) Pengujian sinyal kontrol motor servo
- 2) Pengujian gerak satu kaki
- 3) Pengujian gerak maju
- 4) Pengujian keseluruhan

#### 5.1 Pengujian Sinyal Kontrol Multi-Servo

Pengujian sinyal kontrol multi-servo bertujuan untuk mengetahui kemampuan mikrokontroler kontrol servo dalam menghasilkan pulsa periodik pada pin output. Gambar 5.1 menunjukkan diagram blok pengujian kontrol multi servo. Gambar 5.2 menunjukkan proses pengujian sinyal kontrol multi servo menggunakan *logic analyzer* ELAB-080 dan rangkaian pengujian Sinyal kontrol multi servo.

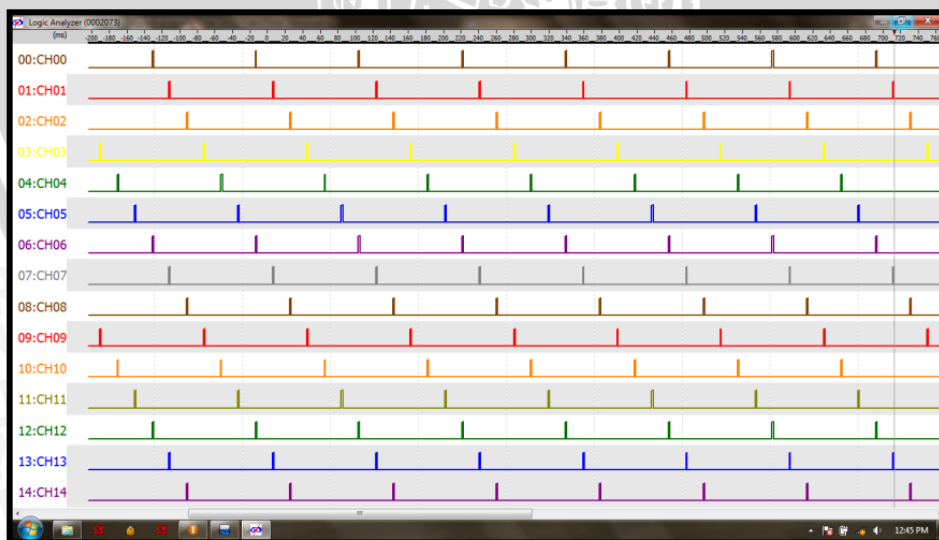


Gambar 5. 1. Diagram Blok Pengujian Sinyal Kontrol Multi Servo



Gambar 5. 1. Pengujian Sinyal Kontrol Multi Servo

Pengujian dilakukan pada saat robot sedang berdiri kemudian melihat sinyal yang dikirimkan dari mikrokontroler untuk delapan belas motor servo. Gambar 5.3 menunjukkan data hasil pengujian. Dalam pengujian ini, hanya 16 pin output yang dihubungkan pada *logic analyzer* CH00 – CH015 karena pada *logic analyzer* ELAB-080 hanya terdapat 16 pin, dan 16 pin output dapat menghasilkan pulsa periodik.

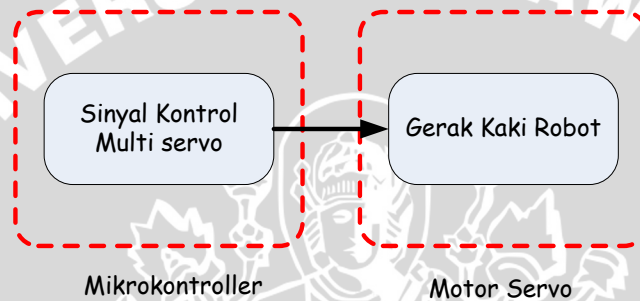


Gambar 5. 3. Sinyal Kontrol Multi Servo



## 5.2 Pengujian Gerak Satu Kaki

Pengujian gerak satu kaki bertujuan untuk mengetahui respon gerak untuk masing masing kaki yang digunakan pada robot berkaki enam , selain itu juga untuk mengetahui apakah sinyal yang dihasilkan mikrokontroler dapat mengontrol posisi motor servo. Pengujian dilakukan dengan menghubungkan motor servo dengan mikrokontroler dan rangkaian catu daya. Catu daya yang digunakan adalah 6 V untuk motor DC servo, dan 5 V untuk rangkaian mikrokontroler. Gambar 5.4 menunjukkan diagram blok pengujian motor DC servo.



Gambar 5.4. Diagram Blok Pengujian Gerak Satu Kaki

Tujuan akhir pengujian gerak satu kaki adalah untuk mengetahui respon masing masing kaki agar sesuai untuk mencapai gerak maju dari robot berkaki enam yang di tunjukkan dalam Gambar 5.5.



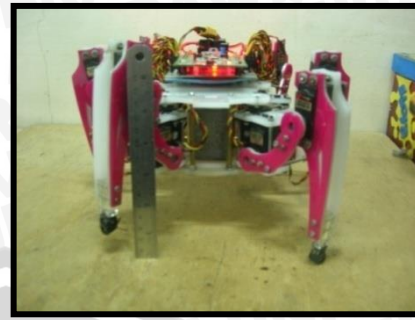
(a)



(b)



(c)



(d)

Gambar 5.5. Pengujian Gerak Satu Kaki (a) kaki kanan depan menapak (b) kaki kanan depan terangkat (c) kaki kiri depan menapak (d) kaki kiri depan terangkat

Pengujian pada gambar adalah kaki kiri depan dan kaki kanan depan diberi perintah untuk mengangkat, dan menapak.

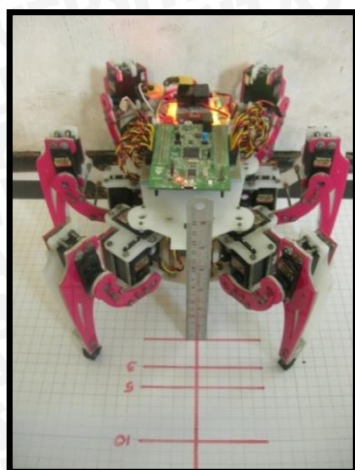
Tabel 5.1 Hasil Pengujian Gerak Mengangkat Kaki

Gerak Mengangkat Kaki		
No.	Kaki	Keterangan
1	Depan Kiri	Berhasil
2	Belakang Kanan	Berhasil
3	Tengah Kanan	Berhasil
4	Belakang Kiri	Berhasil
5	Depan Kanan	Berhasil
6	Tengah Kiri	Berhasil

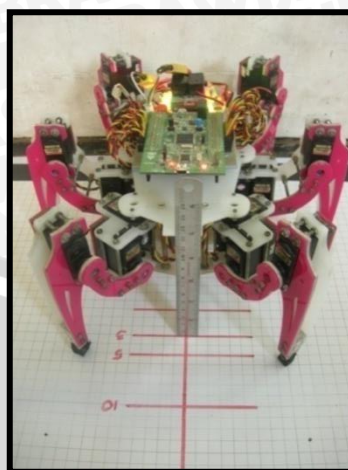
### 5.3 Pengujian Gerak Maju

Pengujian gerak maju bertujuan untuk mengetahui penerapan metode pergerakan menggunakan implentasi *invrse kinematic* pada robot berkaki enam yang telah dirancang, selain itu juga untuk mengetahui apakah robot dapat bergerak sejauh nilai koordinat yang diberikan. Prosedur pengujian dilakukan dengan meletakkan robot di area terbuka kemudian dijalankan, pada saat robot bergerak dan telah menyelesaikan satu kali gerak maju akan di ukur sejauh mana robot akan berhenti dan tidak bergeser dari garis yang sudah ditentukan. Pengujian dilakukan sebanyak lima kali untuk masing masing nilai koordinat yang diberikan. Nilai error akan didapatkan apa bila robot bergerak melebihi nilai yang ditentukan.

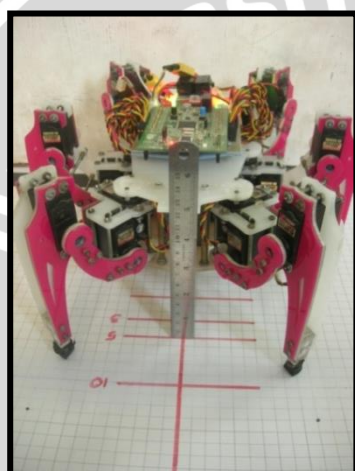




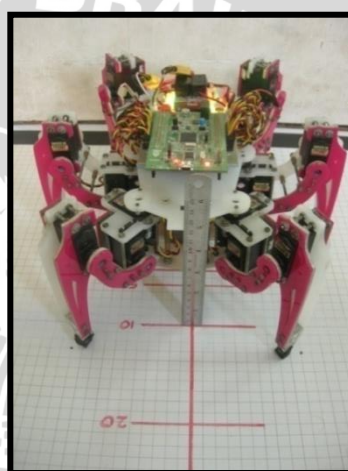
(a)



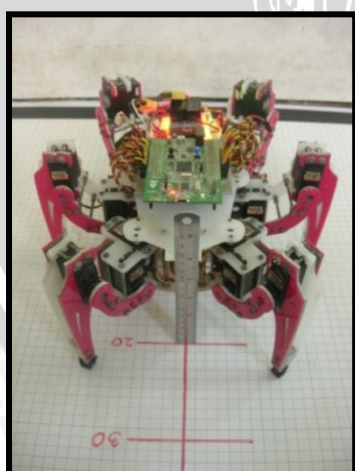
(b)



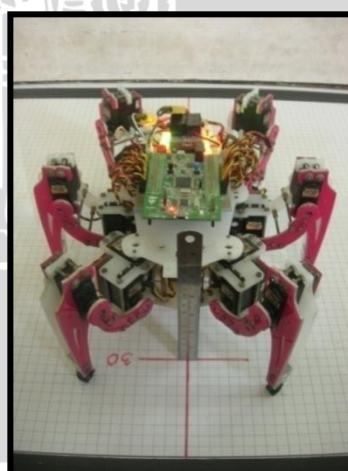
(c)



(d)



(e)



(f)

Gambar 5.6. Pengujian Gerak Maju (a) Posisi Robot sebelum bergerak (b) Posisi Robot bergerak 3cm (c) Posisi Robot bergerak 5cm (d) Posisi Robot bergerak 10cm (e) Posisi Robot bergerak 20cm (f) Posisi Robot bergerak 30cm



Tabel 5.2 Hasil Pengujian Gerak Maju 3cm

<b>Gerak Maju 3 cm</b>		
No.	Hasil yang didapatkan (cm)	Error (cm)
1	4	1
2	3	0
3	4	1
4	3	0
5	3.5	0,5
<b>Error rata-rata (cm)</b>		0,5
<b>Error rata-rata (%)</b>		16,67

Tabel 5.3 Hasil Pengujian Gerak Maju 5cm

<b>Gerak Maju 5 cm</b>		
No.	Hasil yang didapatkan (cm)	Error (cm)
1	6	1
2	5.5	0,5
3	4	1
4	5	0
5	5	0
<b>Error rata-rata (cm)</b>		0,5
<b>Error rata-rata (%)</b>		10

Tabel 5.4 Hasil Pengujian Gerak Maju 10cm

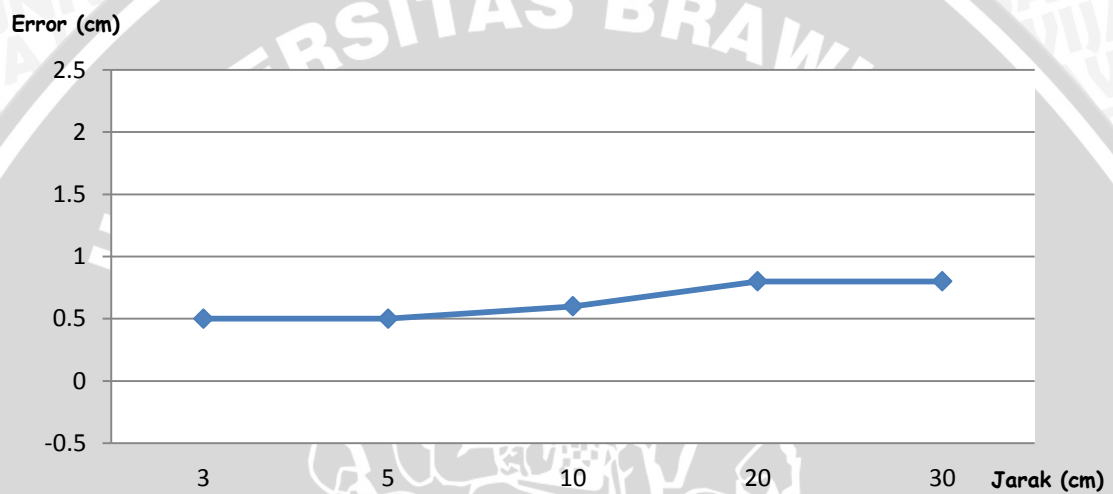
<b>Gerak Maju 10 cm</b>		
No.	Hasil yang didapatkan (cm)	Error (cm)
1	10	0
2	11	1
3	9	1
4	9.5	0,5
5	10.5	0,5
<b>Error rata-rata (cm)</b>		0,6
<b>Error rata-rata (%)</b>		6

Tabel 5.5 Hasil Pengujian Gerak Maju 20cm

<b>Gerak Maju 20 cm</b>		
No.	Hasil yang didapatkan (cm)	Error (cm)
1	21.5	1,5
2	19	1
3	21	1
4	20.5	0,5
5	19	1
<b>Error rata-rata (cm)</b>		0,8
<b>Error rata-rata (%)</b>		4

Tabel 5.6 Hasil Pengujian Gerak Maju 30cm

Gerak Maju 30 cm		
No.	Hasil yang didapatkan (cm)	Error (cm)
1	30	0
2	32	2
3	30	0
4	31	1
5	29	1
<b>Error rata-rata (cm)</b>		0,8
<b>Error rata-rata (%)</b>		2,67



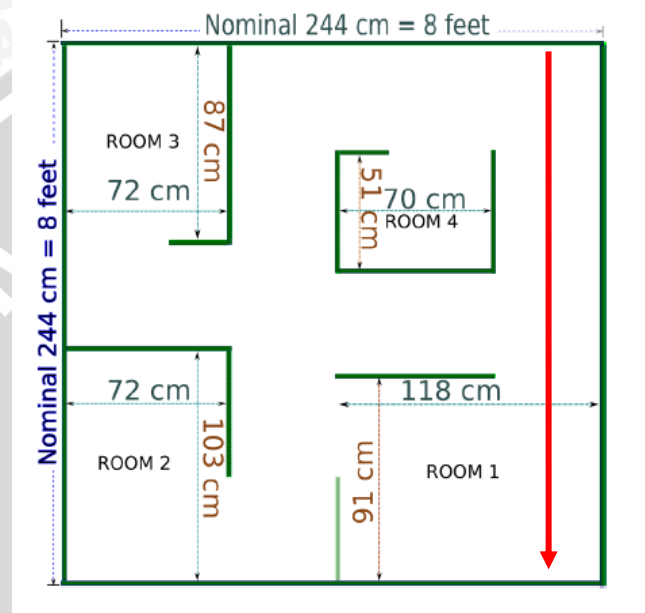
Gambar 5.7 Respon Pengujian Gerak Maju

Berdasarkan hasil Pengujian pada, dapat diketahui bahwa performa robot saat bergerak maju 3cm terdapat kesalahan rata-rata 16,67% atau 0.5cm dan saat bergerak maju 30cm terdapat kesalahan rata-rata 2,67% atau 0.8cm. Terjadinya kesalahan disebabkan pada saat robot berhenti, robot harus menyelesaikan satu program gerakan dan tidak adanya umpan balik pada robot dalam memperbaiki posisi, inilah yang menyebabkan robot tidak berhenti pada koordinat yang diinginkan.

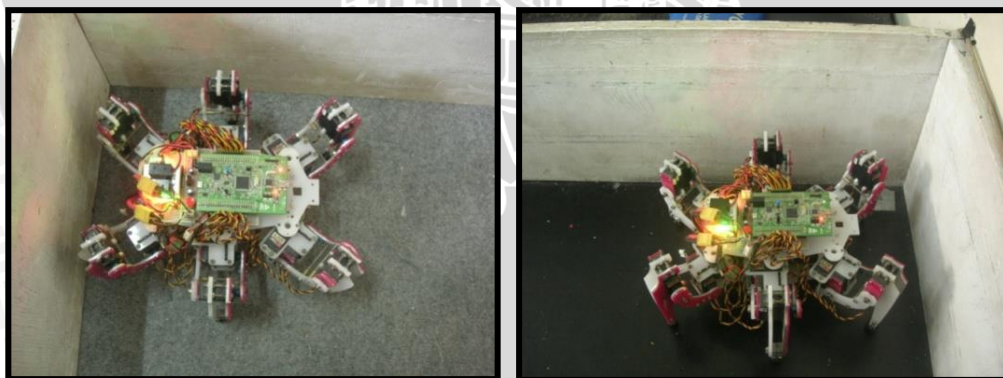
#### 5.4 Pengujian Keseluruhan

Pengujian keseluruhan bertujuan untuk mengetahui penerapan metode pergerakan menggunakan implentasi *inverse kinematic* pada robot berkaki enam yang telah dirancang, selain itu juga untuk mengetahui apakah robot dapat

bergerak lurus. Hasil dari pergerakan yang dilakukan berulang. Pengujian dilakukan dengan cara menjalakkan robot ditempat datar. Parameter keberhasilan system adalah jika alat dapat berjalan lurus pada jalur terjauh dalam arena lomba KRPAI yaitu 244 cm yang ditunjukkan dalam gambar 5.8. Nilai error akan didapatkan apa bila robot bergerak serong atau bergeser dari jalur yang semestinya dan dihitung besar sudut ketika robot bergeser.



Gambar 5.8 Lintasan Pengujian Kesluruhan



(a)

(b)

Gambar 5.9 Pengujian Kesluruhan (a) Posisi robot sebelum bergerak (b) Posisi robot setelah bergerak



Tabel 5.7 Hasil Pengujian Keseluruhan

No.	Gerak Maju 244cm		
	Selisih dari titik tengah (cm)	Error ( $^{\circ}$ )	Keterangan
1	14	3.51	Berhasil
2	10	2.35	Berhasil
3	-	-	Gagal, membentur dinding
4	5	1.17	Berhasil
5	-	-	Gagal, membentur dinding
6	13	3.15	Berhasil
<b>Error rata-rata (<math>^{\circ}</math>)</b>		2.54	

Berdasarkan hasil Pengujian pada table 5.7 dapat diketahui bahwa performa robot saat bergerak maju dapat terdapat kesalahan rata-rata  $2.54^{\circ}$ . Terjadinya error pada saat robot berjalan berulang adanya akurasi atau ketepatan motor servo dalam memberikan nilai sudut sesuai dari hasil perhitungan, maupun dari bentuk mekanik robot dalam bergerak maju dan tidak adanya umpan balik pada robot dalam memperbaiki posisi, inilah yang menyebabkan robot tidak berhenti pada koordinat akhir yang diinginkan.



## BAB VI PENUTUP

### 6.1 Kesimpulan

Berdasarkan pengujian dan analisis implementasi *inverse kinematic* pada pergerakan robot berkaki, dapat ditarik kesimpulan sebagai berikut :

- 1) Metode pergerakan robot berkaki enam menggunakan persamaan *inverse kinematic* mampu membuat robot bergerak maju 3cm terdapat kesalahan rata-rata 16,67% atau 0.5cm dan saat bergerak maju 30cm terdapat kesalahan rata-rata 2,67% atau 0.8cm.
- 2) Metode pergerakan robot berkaki enam menggunakan persamaan *inverse kinematic* mampu membuat robot bergerak maju sejauh 244cm, dan robot bergerak serong atau bergeser dari jarak yang semestinya dengan kesalahan rata-rata  $2.54^0$ .
- 3) Sistem elektronika robot berkaki enam terdiri dari rangkaian catu daya mikrokontroler dengan menggunakan rangkaian *fixed output regulator* untuk kontrol servo dan rangkaian catu daya menggunakan modul UBEC, dimana rangkaian catu daya yang digunakan dapat menunjang penggunaan 18 buah motor servo.
- 4) Robot Berkaki Enam menggunakan mikrokontroler STM32F4 Discovery sebagai pengendali utama motor servo dengan metode kontrol multi servo, dan dapat mengolah nilai koordinat menggunakan persamaan *inverse kinematic* menjadi sudut untuk tiap tiap motor servo.

### 6.2 Saran

Dalam perancangan dan pembuatan alat ini masih terdapat banyak kekurangan. Beberapa hal yang direkomendasikan untuk pengembangan lebih lanjut adalah sebagai berikut:

- 1) Disarankan untuk menggunakan motor DC servo yang dilengkapi dengan *position feedback* agar pencapaian koordinat setiap ujung kaki lebih akurat.
- 2) Penambahan sensor ultrasonic atau sejenisnya untuk digunakan pada ruang tertutup dan mendapatkan feedback berupa jarak agar robot biasa menyesuaikan gerakannya sendiri terhadap dinding.

## DAFTAR PUSTAKA

DIKTI. 2013. *Panduan Kontes Robot Pemadam Api Indonesia 2013*. Jakarta:

DIKTI.

Craig, John j. 2005. *Introduction to Robotics*. Pearson Prentice Hall.

Pitowarno, Endra. 2006. *Robotika: Desain, Kontrol dan Kecerdasan Buatan*.

Yogyakarta: Penerbit ANDI.

STMicroelectronic. 2012. *STM32F4DISCOVERY, STM32F4 high performance discovery board*.

Wardana, Wahyu Febrian Eko. 2011. *Penggunaan Sisitem Tiga Sendi Pada Robot KRCI Divisi Berkaki Untuk Mengatasi Objek Uneven Floor*. Malang: Skripsi Jurusan Teknik Elektro FT-UB.

Woering R. 2011. *Simulating the "first steps" of a walking hexapod robot*. University of Technology Eindhoven.



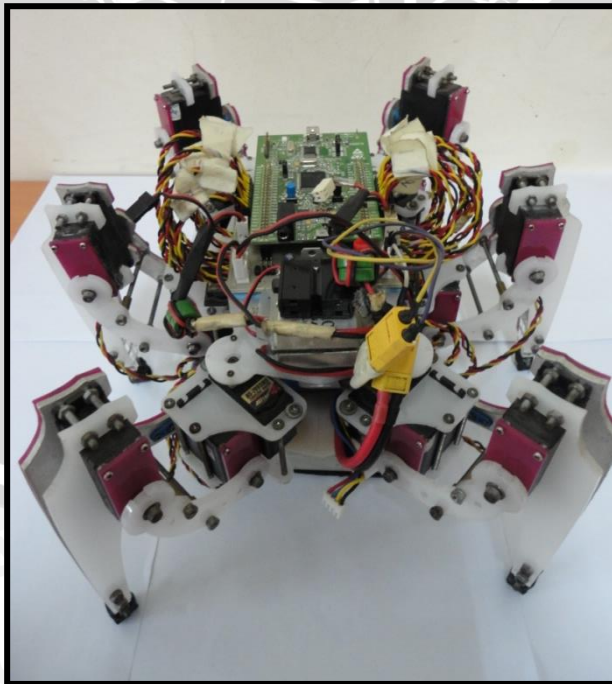
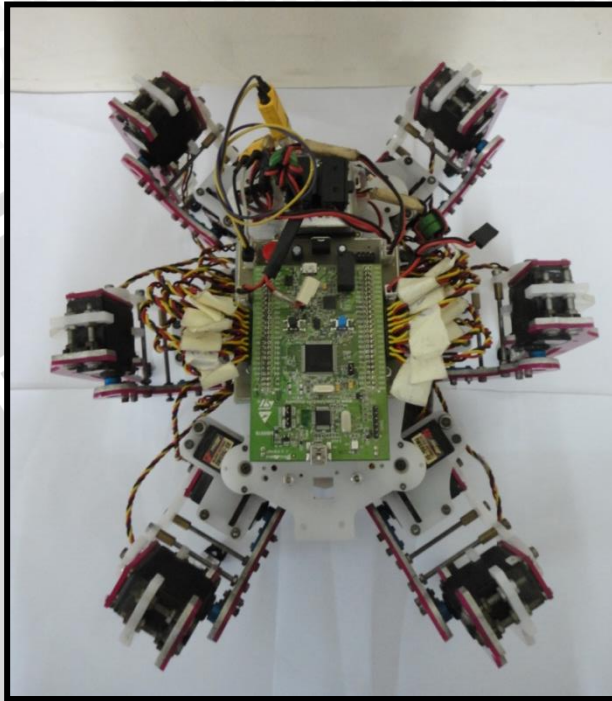
LAMPIRAN

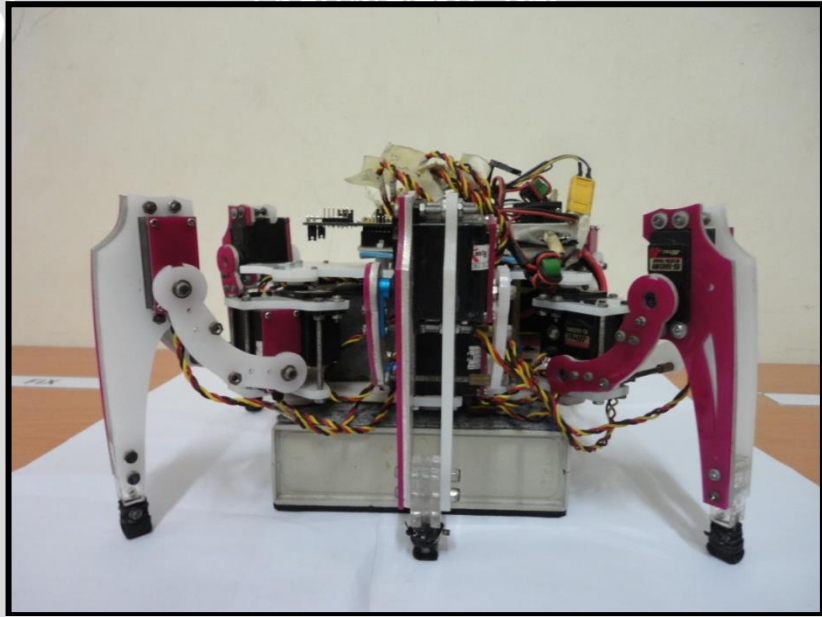
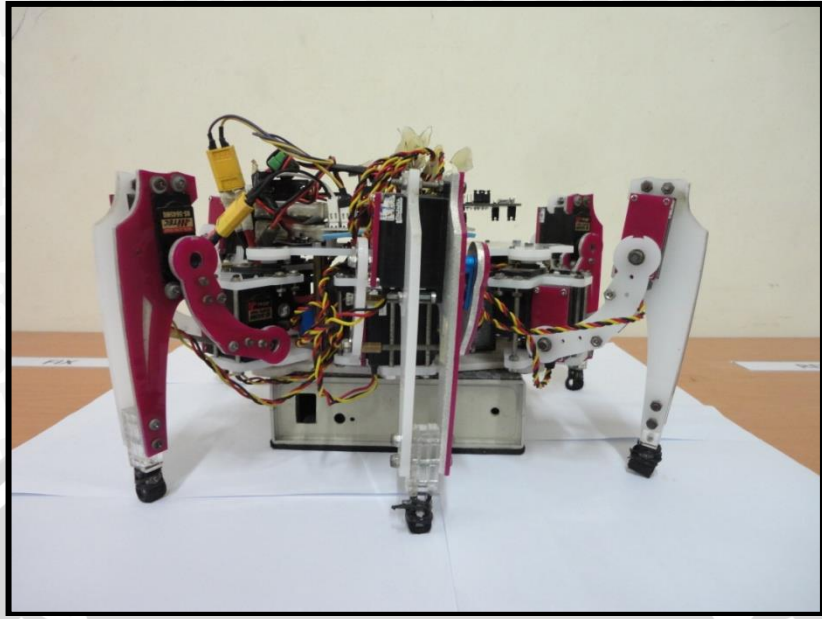
---



**FOTO ALAT**

---







## LISTING PROGRAM

```

#include "stm32f4xx.h"
#include "stm32f4xx_gpio.h"
#include "stm32f4xx_rcc.h"
#include "stm32f4xx_tim.h"
#include "misc.h"
#include "math.h"

GPIO_InitTypeDef      GPIO_InitStructure;
TIM_OCInitTypeDef     TIM_OCInitStructure;
TIM_TimeBaseInitTypeDef TIM_TimeBaseStructure;

int sudut_target[18], sudut_awal[18], x=0, z_A[18], pembagi, y_A[18],
sudut_tahap1[18], sudut_tahap2[18];

#define la 6.5           // besar La
#define lb 12           // besar Lb
#define lj 2.5          // besar L_junction
#define rad2deg 57.29

#define front_left 1
#define front_right 2
#define middle_left 3
#define middle_right 4
#define rear_left 5
#define rear_right 6

#define ch_0  GPIOA,GPIO_Pin_1
#define ch_1  GPIOA,GPIO_Pin_2

#define ch_2  GPIOA,GPIO_Pin_3
#define ch_3  GPIOA,GPIO_Pin_4
#define ch_4  GPIOA,GPIO_Pin_5
#define ch_5  GPIOA,GPIO_Pin_8
#define ch_6  GPIOB,GPIO_Pin_1
#define ch_7  GPIOB,GPIO_Pin_2
#define ch_8  GPIOB,GPIO_Pin_3
#define ch_9  GPIOB,GPIO_Pin_4
#define ch_10 GPIOB,GPIO_Pin_5
#define ch_11 GPIOB,GPIO_Pin_8
#define ch_12 GPIOC,GPIO_Pin_1
#define ch_13 GPIOC,GPIO_Pin_2
#define ch_14 GPIOC,GPIO_Pin_3
#define ch_15 GPIOC,GPIO_Pin_4
#define ch_16 GPIOC,GPIO_Pin_5
#define ch_17 GPIOC,GPIO_Pin_8

int leg;
int ww[6];
float beta[6];
float leg_next[6];
float A_body;           // translasi pusat badan
float A_leg[6];        // translasi setiap pangkal kaki
float P_body;          // jarak titik pusat robot ke titik imajiner
putar
int epsilon_body;     // besar sudut vektor hadap thdp vektor poros-
imajiner
float phi[6];
float lambpsi[6];     // lambda / psi temporal
float P_leg[6];       // jarak titik pusat kaki ke titik imajiner
putar
float epsilon[6];     // epsilon tiap N

```

```

unsigned int sudut[18]= {4915,4915,4915,4915,4915,4915,
                        4915,4915,4915,4915,4915,4915,
                        4915,4915,4915,4915,4915,4915};

unsigned char y,a,b,c,d,v,index_bantu,index_servo;
uint32_t PrescalerValue;

const float A_const[18]={26.68,26.68,26.68,26.68,26.68,26.68,
                        26.68,26.68,26.68,26.68,26.68,26.68,
                        26.68,26.68,26.68,26.68,26.68,26.68};

const int B_const[18] = {2681.14,2681.14,2681.14,2681.14,2681.14,2681.14,
                        2681.14,2681.14,2681.14,2681.14,2681.14,2681.14,
                        2681.14,2681.14,2681.14,2681.14,2681.14,2681.14};

const float correct[18] ={-12,95.5,-15, 0,103,-15, 12,89,-15, //left
                          -12,89,-15, 0,100,-20, 12,92,-5}; //right

const float leg_1[6]      = {9.0574,6.9,9.0574,9.0574,6.9,9.0574}; //L1
const float leg_norm[6]  = {9.5,9.5,9.5,9.5,9.5,9.5}; //L2
const float alpha[6]     = {30,90,150,30,90,150}; //alpha

struct {float teta1; float teta2; float teta3; }temp_angle;

float c1,c2;
float Lsem,c3,c4;
int i;
float temp_var,temp_var2;
void generate_movement_data(int deflection_angle, int translation, int high);
void inv_kine(char leg_pos,float x,float w,float z);
void servo_lin_func(int ch_num,float value);
void movement(int menu);
void awal();
void tahap();
void tahap1();

void tahap2();
void target();

void Delay(__IO uint32_t nCount);
void timer_config()
{
    PrescalerValue = (uint32_t) ((SystemCoreClock / 2) / 9830250) - 1;

    /* Time base configuration */
    TIM_TimeBaseStructure.TIM_Period = 65535;
    TIM_TimeBaseStructure.TIM_Prescaler = 0;
    TIM_TimeBaseStructure.TIM_ClockDivision = 0;
    TIM_TimeBaseStructure.TIM_CounterMode = TIM_CounterMode_Up;
    TIM_TimeBaseInit(TIM3, &TIM_TimeBaseStructure);
    /* Prescaler configuration */
    TIM_PrescalerConfig(TIM3, PrescalerValue,
    TIM_PSCReloadMode_Immediate);
}

/*=====
*setting timer
*=====*/

void interrupt_config()
{
    NVIC_InitTypeDef NVIC_InitStructure;

    /* TIM3 clock enable */
    RCC_APB1PeriphClockCmd(RCC_APB1Periph_TIM3, ENABLE);
    /* Enable the TIM3 global Interrupt */
    NVIC_InitStructure.NVIC_IRQChannel = TIM3_IRQn;
    NVIC_InitStructure.NVIC_IRQChannelPreemptionPriority = 0;
    NVIC_InitStructure.NVIC_IRQChannelSubPriority = 1;
    NVIC_InitStructure.NVIC_IRQChannelCmd = ENABLE;
    NVIC_Init(&NVIC_InitStructure);
    /* Output Compare Timing Mode configuration: Channel1 */
    TIM_OCInitStructure.TIM_OCMode = TIM_OCMode_Timing;
}

```

```

TIM_OCInitStructure.TIM_OutputState = TIM_OutputState_Enable;
TIM_OCInitStructure.TIM_Pulse = CCR1_Val;
TIM_OCInitStructure.TIM_OCPolarity = TIM_OCPolarity_High;
TIM_OC1Init(TIM3, &TIM_OCInitStructure);
TIM_OC1PreloadConfig(TIM3, TIM_OCPreload_Disable);
TIM_OCInitStructure.TIM_OutputState = TIM_OutputState_Enable;
TIM_OCInitStructure.TIM_Pulse = CCR2_Val;
TIM_OC2Init(TIM3, &TIM_OCInitStructure);
TIM_OC2PreloadConfig(TIM3, TIM_OCPreload_Disable);
TIM_OCInitStructure.TIM_OutputState = TIM_OutputState_Enable;
TIM_OCInitStructure.TIM_Pulse = CCR3_Val;
TIM_OC3Init(TIM3, &TIM_OCInitStructure);
TIM_OC3PreloadConfig(TIM3, TIM_OCPreload_Disable);

```

```

//TIM_ClearOC1Ref(TIM3, TIM_OCClear_Enable);
//TIM_ClearOC2Ref(TIM3, TIM_OCClear_Enable);
//TIM_ForcedOC1Config(TIM3, TIM_ForcedAction_Active);
/* TIM Interrupts enable */
TIM_ITConfig(TIM3, TIM_IT_Update, ENABLE);
TIM_ITConfig(TIM3, TIM_IT_CC1, ENABLE);
TIM_ITConfig(TIM3, TIM_IT_CC2, ENABLE);
TIM_ITConfig(TIM3, TIM_IT_CC3, ENABLE);
/* TIM3 enable counter */
TIM_Cmd(TIM3, ENABLE);
}

```

```

/*=====
*setting input/output
*=====*/

```

```

void gpioinit()
{
RCC_AHB1PeriphClockCmd(RCC_AHB1Periph_GPIOA, ENABLE);
GPIO_InitStructure.GPIO_Pin =
GPIO_Pin_1|GPIO_Pin_2|GPIO_Pin_3|GPIO_Pin_4|GPIO_Pin_5|GPIO_Pin_
8;
GPIO_InitStructure.GPIO_Mode = GPIO_Mode_OUT;

```

```

GPIO_InitStructure.GPIO_OType = GPIO_OType_PP;
GPIO_InitStructure.GPIO_Speed = GPIO_Speed_100MHz;
GPIO_InitStructure.GPIO_PuPd = GPIO_PuPd_NOPULL;
GPIO_Init(GPIOA, &GPIO_InitStructure);

```

```

RCC_AHB1PeriphClockCmd(RCC_AHB1Periph_GPIOB, ENABLE);
GPIO_InitStructure.GPIO_Pin =
GPIO_Pin_1|GPIO_Pin_2|GPIO_Pin_3|GPIO_Pin_4|GPIO_Pin_5|GPIO_Pin_
8;
GPIO_InitStructure.GPIO_Mode = GPIO_Mode_OUT;
GPIO_InitStructure.GPIO_OType = GPIO_OType_PP;
GPIO_InitStructure.GPIO_Speed = GPIO_Speed_100MHz;
GPIO_InitStructure.GPIO_PuPd = GPIO_PuPd_NOPULL;
GPIO_Init(GPIOB, &GPIO_InitStructure);

```

```

RCC_AHB1PeriphClockCmd(RCC_AHB1Periph_GPIOC, ENABLE);
GPIO_InitStructure.GPIO_Pin =
GPIO_Pin_1|GPIO_Pin_2|GPIO_Pin_3|GPIO_Pin_4|GPIO_Pin_5|GPIO_Pin_
8;
GPIO_InitStructure.GPIO_Mode = GPIO_Mode_OUT;
GPIO_InitStructure.GPIO_OType = GPIO_OType_PP;
GPIO_InitStructure.GPIO_Speed = GPIO_Speed_100MHz;
GPIO_InitStructure.GPIO_PuPd = GPIO_PuPd_NOPULL;
GPIO_Init(GPIOC, &GPIO_InitStructure);

```

```

RCC_AHB1PeriphClockCmd(RCC_AHB1Periph_GPIOD, ENABLE);
GPIO_InitStructure.GPIO_Pin = GPIO_Pin_12 | GPIO_Pin_13 | GPIO_Pin_14 |
GPIO_Pin_15;
GPIO_InitStructure.GPIO_Mode = GPIO_Mode_OUT;
GPIO_InitStructure.GPIO_OType = GPIO_OType_PP;
GPIO_InitStructure.GPIO_Speed = GPIO_Speed_100MHz;
GPIO_InitStructure.GPIO_PuPd = GPIO_PuPd_NOPULL;
GPIO_Init(GPIOD, &GPIO_InitStructure);

```

```

RCC_AHB1PeriphClockCmd(RCC_AHB1Periph_GPIOA, ENABLE);
GPIO_InitStructure.GPIO_Pin = GPIO_Pin_0;

```



```

GPIO_InitStructure.GPIO_Mode = GPIO_Mode_IN;
GPIO_InitStructure.GPIO_OType = GPIO_OType_PP;
GPIO_InitStructure.GPIO_Speed = GPIO_Speed_100MHz;
GPIO_InitStructure.GPIO_PuPd = GPIO_PuPd_DOWN;
GPIO_Init(GPIOA, &GPIO_InitStructure);
}

```

```

/*=====

```

```

*program utama

```

```

*=====*/

```

```

int main(void)

```

```

{
SystemInit();
gpioinit();
interrupt_config();
timer_config();

```

```

generate_movement_data(0,3,10);
Delay(150);

```

```

movement(1);

```

```

while(GPIO_ReadInputDataBit(GPIOA, GPIO_Pin_0)==0);

```

```

while(1)

```

```

{ /movement(1);

```

```

//--berjalan--//

```

```

GPIO_SetBits(GPIOD, GPIO_Pin_12);
GPIO_ResetBits(GPIOD, GPIO_Pin_15);
movement(2);

```

```

GPIO_ResetBits(GPIOD, GPIO_Pin_12);
GPIO_SetBits(GPIOD, GPIO_Pin_14);
movement(2);

```

```

GPIO_ResetBits(GPIOD, GPIO_Pin_14);

```

```

GPIO_SetBits(GPIOD, GPIO_Pin_13);
movement(2);

```

```

GPIO_ResetBits(GPIOD, GPIO_Pin_13);
GPIO_SetBits(GPIOD, GPIO_Pin_15);
movement(2);
}

```

```

/*=====

```

```

*setting port untuk output pulsa motor servo

```

```

*=====*/

```

```

void TIM3_IRQHandler(void)

```

```

{
if (TIM_GetITStatus(TIM3, TIM_IT_Update) != RESET)
{
if(++index_servo>=8) {index_servo=0;}
switch (index_servo)
{
case 0: TIM_ClearITPendingBit(TIM3, TIM_IT_Update);
GPIO_SetBits(ch_0); GPIO_SetBits(ch_6); GPIO_SetBits(ch_12);
break;
case 1: TIM_ClearITPendingBit(TIM3, TIM_IT_Update);
GPIO_SetBits(ch_1); GPIO_SetBits(ch_7); GPIO_SetBits(ch_13);
break;
case 2: TIM_ClearITPendingBit(TIM3, TIM_IT_Update);
GPIO_SetBits(ch_2); GPIO_SetBits(ch_8); GPIO_SetBits(ch_14);
break;
case 3: TIM_ClearITPendingBit(TIM3, TIM_IT_Update);
GPIO_SetBits(ch_3); GPIO_SetBits(ch_9); GPIO_SetBits(ch_15);
break;
case 4: TIM_ClearITPendingBit(TIM3, TIM_IT_Update);
GPIO_SetBits(ch_4); GPIO_SetBits(ch_10); GPIO_SetBits(ch_16);
break;
case 5: TIM_ClearITPendingBit(TIM3, TIM_IT_Update);
GPIO_SetBits(ch_5); GPIO_SetBits(ch_11); GPIO_SetBits(ch_17);
break;
}
}
}

```

```

if(index_servo>=5) {index_bantu=0;}
else{index_bantu++;}
TIM3 -> CCR1=sudut[index_bantu];
TIM3 -> CCR2=sudut[index_bantu+6];
TIM3 -> CCR3=sudut[index_bantu+12];
}
else if (TIM_GetITStatus(TIM3, TIM_IT_CC1) == SET)
{ TIM_ClearITPendingBit(TIM3, TIM_IT_CC1);
switch (index_servo) {
case 0: GPIO_ResetBits(ch_0);break;
case 1: GPIO_ResetBits(ch_1);break;
case 2: GPIO_ResetBits(ch_2);break;
case 3: GPIO_ResetBits(ch_3);break;
case 4: GPIO_ResetBits(ch_4);break;
case 5: GPIO_ResetBits(ch_5);break;
} }
else if (TIM_GetITStatus(TIM3, TIM_IT_CC2) == SET)
{ TIM_ClearITPendingBit(TIM3, TIM_IT_CC2);
switch (index_servo) {
case 0: GPIO_ResetBits(ch_6);break;
case 1: GPIO_ResetBits(ch_7);break;
case 2: GPIO_ResetBits(ch_8);break;
case 3: GPIO_ResetBits(ch_9);break;
case 4: GPIO_ResetBits(ch_10);break;
case 5: GPIO_ResetBits(ch_11);break;
} }
else if (TIM_GetITStatus(TIM3, TIM_IT_CC3) == SET)
{ TIM_ClearITPendingBit(TIM3, TIM_IT_CC3);
switch (index_servo) {
case 0: GPIO_ResetBits(ch_12);break;
case 1: GPIO_ResetBits(ch_13);break;
case 2: GPIO_ResetBits(ch_14);break;
case 3: GPIO_ResetBits(ch_15);break;
case 4: GPIO_ResetBits(ch_16);break;

```

```

case 5: GPIO_ResetBits(ch_17);break; } }
}

```

```

/*=====
*perhitungan inverse kinematic
*=====*/

```

```

void generate_movement_data(int deflection_angle, int translation, int high)
{
leg=high;
A_body = (float) translation;
for(i=0;i<6;i++)
{
temp_var = (A_body * cos(ww[i] / rad2deg)); //temp_var = Ab*cos(gamma)
temp_var = (leg_norm[i] - temp_var); //temp_var = L2 - temp_var
temp_var2 = (A_body * sin(ww[i] / rad2deg)); //temp_var2 = Ab*sin(gamma)
temp_var = (temp_var2 / temp_var); //var = temp_var2 / temp_var
temp_var = atan(temp_var); //temp_var = arctan (temp_var)
beta[i] = (temp_var*rad2deg); //beta[i] = temp_var

// hitung panjang kaki (L2')
temp_var = (float) sin(ww[i] / rad2deg); //temp_var = sin(gamma);
temp_var = (float) (A_body * temp_var); //temp_var = Ab * temp_var;
temp_var2 = (float) sin(beta[i] / rad2deg); //temp_var2 = sin(alpha);
temp_var = (float) (temp_var / temp_var2); //temp_var = temp_var/temp_var2
leg_next[i] = temp_var
}
}

```

```

void inv_kine(char leg_pos,float x,float w, float z)
{
temp_angle.teta1 = x;

```

```

Lsem = w/(cos(temp_angle.teta1));
Lsem -= lj;
c1 = (la*la) + (lb*lb);

```

```

c2 = 2*la*lb;
temp_angle.teta3 = acos((((Lsem*Lsem)+(z*z))-c1)/c2));
c3=la+(lb*cos(temp_angle.teta3));
c4=lb*sin(temp_angle.teta3);
temp_angle.teta2 = atan(((z*c3)-(Lsem*c4))/((Lsem*c3)+(z*c4)));
temp_angle.teta3 *= rad2deg;
temp_angle.teta2 *= rad2deg;

switch(leg_pos)
{
case front_left :
servo_lin_func(0, temp_angle.teta1); servo_lin_func(1, temp_angle.teta2);
servo_lin_func(2, temp_angle.teta3);
break;

case front_right :
servo_lin_func(9,temp_angle.teta1); servo_lin_func(10,temp_angle.teta2);
servo_lin_func(11,temp_angle.teta3);
break;

case middle_left :
servo_lin_func(3,temp_angle.teta1); servo_lin_func(4,temp_angle.teta2);
servo_lin_func(5,temp_angle.teta3);
break;

case middle_right :
servo_lin_func(12,temp_angle.teta1); servo_lin_func(13,temp_angle.teta2);
servo_lin_func(14,temp_angle.teta3);
break;

case rear_left :
servo_lin_func(6,temp_angle.teta1); servo_lin_func(7,temp_angle.teta2);
servo_lin_func(8,temp_angle.teta3);
break;

case rear_right :
servo_lin_func(15,temp_angle.teta1);servo_lin_func(16,temp_angle.teta2);
servo_lin_func(17,temp_angle.teta3);
break;
}

```

```

}
/*=====
*konversi nilai sudut -> pulsa
*=====*/
void servo_lin_func(int ch_num,float value)
{
sudut[ch_num] =(long int)((A_const[ch_num]*(value+correct[ch_num]))+
B_const[ch_num]);
//sudut[ch_num] = ((A_const[ch_num]*value)+B_const[ch_num]);
}
/*=====
*pola langkah
*=====*/
void movement(int menu)
{
switch (menu)
{
case 0:
inv_kine(front_left,90,6,leg-3);
inv_kine(front_right,90,6,leg-3);
inv_kine(middle_left,90,6,leg-3);
inv_kine(middle_right,90,6,leg-3);
inv_kine(rear_left,90,6,leg-3);
inv_kine(rear_right,90,6,leg-3);
break;

case 1:
inv_kine(front_left,90,7,leg);
inv_kine(front_right,90,7,leg);
inv_kine(middle_left,90,7,leg);
inv_kine(middle_right,90,7,leg);
inv_kine(rear_left,90,7,leg);
inv_kine(rear_right,90,7,leg);
}
}

```



break;

case 2:

```
inv_kine(front_left,75+(beta[0]*0.077),7+(leg_next[0]*0.077),leg);
inv_kine(front_right,95+(beta[3]*0.615),7+(leg_next[3]*0.615),leg);
inv_kine(middle_left,90+(beta[1]*0.461),7+(leg_next[1]*0.461),leg);
inv_kine(middle_right,90+beta[4],7+leg_next[4],leg);
inv_kine(rear_left,90+(beta[2]*0.846),7+(leg_next[2]*0.846),leg);
inv_kine(rear_right,90+(beta[5]*0.231),7+(leg_next[5]*0.231),leg);
Delay(17);
```

//tengah kanan

```
inv_kine(front_left,80+(beta[0]*0.154),7+(leg_next[0]*0.154),leg);
inv_kine(front_right,95+(beta[3]*0.692),7+(leg_next[3]*0.692),leg);
inv_kine(middle_left,90+(beta[1]*0.538),7+(leg_next[1]*0.538),leg);
inv_kine(middle_right,90+beta[4],7+leg_next[4],leg-3);
inv_kine(rear_left,90+(beta[2]*0.923),7+(leg_next[2]*0.923),leg);
inv_kine(rear_right,90+(beta[5]*0.307),7+(leg_next[5]*0.307),leg);
Delay(17);
```

```
inv_kine(front_left,80+(beta[0]*0.231),7+(leg_next[0]*0.231),leg);
inv_kine(front_right,100+(beta[3]*0.796),7+(leg_next[3]*0.796),leg);
inv_kine(middle_left,90+(beta[1]*0.615),7+(leg_next[1]*0.615),leg);
inv_kine(middle_right,90,7,leg-3);
inv_kine(rear_left,90+beta[2],7+leg_next[2],leg);
inv_kine(rear_right,90+(beta[5]*0.385),7+(leg_next[5]*0.385),leg);
Delay(17);
```

//belakang kiri

```
inv_kine(front_left,85+(beta[0]*0.307),7+(leg_next[0]*0.307),leg);
inv_kine(front_right,100+(beta[3]*0.846),7+(leg_next[3]*0.846),leg);
inv_kine(middle_left,90+(beta[1]*0.692),7+(leg_next[1]*0.692),leg);
inv_kine(middle_right,90,7,leg);
inv_kine(rear_left,90+beta[2],7+leg_next[2],leg-3);
inv_kine(rear_right,90+(beta[5]*0.461),7+(leg_next[5]*0.461),leg);
Delay(17);
```

```
inv_kine(front_left,85+(beta[0]*0.385),7+(leg_next[0]*0.385),leg);
inv_kine(front_right,105+(beta[3]*0.923),7+(leg_next[3]*0.923),leg);
inv_kine(middle_left,90+(beta[1]*0.796),7+(leg_next[1]*0.796),leg);
inv_kine(middle_right,90+(beta[4]*0.077),7+(leg_next[4]*0.077),leg);
inv_kine(rear_left,90,7,leg-3);
inv_kine(rear_right,90+(beta[5]*0.538),7+(leg_next[5]*0.538),leg);
Delay(17);
```

//menapak

```
inv_kine(front_left,90+(beta[0]*0.461),7+(leg_next[0]*0.461),leg);
inv_kine(front_right,105+beta[3],7+leg_next[3],leg);
inv_kine(middle_left,90+(beta[1]*0.846),7+(leg_next[1]*0.846),leg);
inv_kine(middle_right,90+(beta[4]*0.154),7+(leg_next[4]*0.154),leg);
inv_kine(rear_left,90,7,leg);
inv_kine(rear_right,90+(beta[5]*0.615),7+(leg_next[5]*0.615),leg);
Delay(17);
```

//depan kanan

```
inv_kine(front_left,90+(beta[0]*0.538),7+(leg_next[0]*0.538),leg);
inv_kine(front_right,105+beta[3],7+leg_next[3],leg-3);
inv_kine(middle_left,90+(beta[1]*0.923),7+(leg_next[1]*0.923),leg);
inv_kine(middle_right,90+(beta[4]*0.231),7+(leg_next[4]*0.231),leg);
inv_kine(rear_left,90+(beta[2]*0.077),7+(leg_next[2]*0.077),leg);
inv_kine(rear_right,90+(beta[5]*0.692),7+(leg_next[5]*0.692),leg);
Delay(17);
```

```
inv_kine(front_left,95+(beta[0]*0.615),7+(leg_next[0]*0.615),leg);
inv_kine(front_right,90,7,leg-3);
inv_kine(middle_left,90+beta[1],7+leg_next[1],leg);
inv_kine(middle_right,90+(beta[4]*0.307),7+(leg_next[4]*0.307),leg);
inv_kine(rear_left,90+(beta[2]*0.154),7+(leg_next[2]*0.154),leg);
inv_kine(rear_right,90+(beta[5]*0.796),7+(leg_next[5]*0.796),leg);
Delay(17);
```

```

//tengah kiri
inv_kine(front_left,95+(beta[0]*0.692),7+(leg_next[0]*0.692),leg);
inv_kine(front_right,75,7,leg);
inv_kine(middle_left,90+beta[1],7+leg_next[1],leg-3);
inv_kine(middle_right,90+(beta[4]*0.385),7+(leg_next[4]*0.385),leg);
inv_kine(rear_left,90+(beta[2]*0.231),7+(leg_next[2]*0.231),leg);
inv_kine(rear_right,90+(beta[5]*0.846),7+(leg_next[5]*0.846),leg);
Delay(17);

inv_kine(front_left,100+(beta[0]*0.796),7+(leg_next[0]*0.796),leg);
inv_kine(front_right,75+(beta[3]*0.077),7+(leg_next[3]*0.077),leg);
inv_kine(middle_left,90,7,leg-3);
inv_kine(middle_right,90+(beta[4]*0.461),7+(leg_next[4]*0.461),leg);
inv_kine(rear_left,90+(beta[2]*0.307),7+(leg_next[2]*0.307),leg);
inv_kine(rear_right,90+(beta[5]*0.923),7+(leg_next[5]*0.923),leg);
Delay(17);

//menapak
inv_kine(front_left,100+(beta[0]*0.846),7+(leg_next[0]*0.846),leg);
inv_kine(front_right,80+(beta[3]*0.154),7+(leg_next[3]*0.154),leg);
inv_kine(middle_left,90,7,leg);
inv_kine(middle_right,90+(beta[4]*0.538),7+(leg_next[4]*0.538),leg);
inv_kine(rear_left,90+(beta[2]*0.385),7+(leg_next[2]*0.385),leg);
inv_kine(rear_right,90+beta[5],7+leg_next[5],leg);
Delay(17);

//belakang kanan
inv_kine(front_left,105+(beta[0]*0.923),7+(leg_next[0]*0.923),leg);
inv_kine(front_right,80+(beta[3]*0.231),7+(leg_next[3]*0.231),leg);
inv_kine(middle_left,90+(beta[1]*0.077),7+(leg_next[1]*0.077),leg);
inv_kine(middle_right,90+(beta[4]*0.615),7+(leg_next[4]*0.615),leg);
inv_kine(rear_left,90+(beta[2]*0.461),7+(leg_next[2]*0.461),leg);
inv_kine(rear_right,90+beta[5],7+leg_next[5],leg-3);
Delay(17);

inv_kine(front_left,105+beta[0],7+(leg_next[0]),leg);
inv_kine(front_right,85+(beta[3]*0.307),7+(leg_next[3]*0.307),leg);
inv_kine(middle_left,90+(beta[1]*0.154),7+(leg_next[1]*0.154),leg);
inv_kine(middle_right,90+(beta[4]*0.692),7+(leg_next[4]*0.692),leg);
inv_kine(rear_left,90+(beta[2]*0.538),7+(leg_next[2]*0.538),leg);
inv_kine(rear_right,90,7,leg-3);
Delay(17);

//depan kiri
inv_kine(front_left,105+beta[0],7+(leg_next[0]),leg-3);
inv_kine(front_right,85+(beta[3]*0.385),7+(leg_next[3]*0.385),leg);
inv_kine(middle_left,90+(beta[1]*0.231),7+(leg_next[1]*0.231),leg);
inv_kine(middle_right,90+(beta[4]*0.796),7+(leg_next[4]*0.796),leg);
inv_kine(rear_left,90+(beta[2]*0.615),7+(leg_next[2]*0.615),leg);
inv_kine(rear_right,90,7,leg);
Delay(17);

inv_kine(front_left,90,7,leg-3);
inv_kine(front_right,90+(beta[3]*0.461),7+(leg_next[3]*0.461),leg);
inv_kine(middle_left,90+(beta[1]*0.307),7+(leg_next[1]*0.307),leg);
inv_kine(middle_right,90+(beta[4]*0.846),7+(leg_next[4]*0.846),leg);
inv_kine(rear_left,90+(beta[2]*0.692),7+(leg_next[2]*0.692),leg);
inv_kine(rear_right,90+(beta[5]*0.077),7+(leg_next[5]*0.077),leg);
Delay(17);

//menapak
inv_kine(front_left,75,7,leg);
inv_kine(front_right,90+(beta[3]*0.538),7+(leg_next[3]*0.538),leg);
inv_kine(middle_left,90+(beta[1]*0.385),7+(leg_next[1]*0.385),leg);
inv_kine(middle_right,90+(beta[4]*0.923),7+(leg_next[4]*0.923),leg);
inv_kine(rear_left,90+(beta[2]*0.796),7+(leg_next[2]*0.796),leg);
inv_kine(rear_right,90+(beta[5]*0.154),7+(leg_next[5]*0.154),leg);
Delay(17);
break;
}
}

```

**DATASHEET**

---





UNIVERSITAS BRAWIJAYA

