



ITS
Institut
Teknologi
Sepuluh Nopember

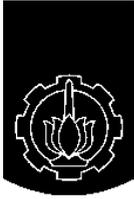
TUGAS AKHIR - TE141599

**SISTEM PENGATURAN KESEIMBANGAN ROBOT
HUMANOID UNTUK BERDIRI DAN BERJALAN PADA
BIDANG MIRING**

Mochamad Farid Mustofa
NRP 2211100031

Dosen Pembimbing
Ir. Rusdhianto Effendie A.K, M.T.

JURUSAN TEKNIK ELEKTRO
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember
Surabaya 2016



ITS
Institut
Teknologi
Sepuluh Nopember

FINAL PROJECT – TE141599

***BALANCE CONTROL SYSTEM OF HUMANOID ROBOT
FOR STANDING AND WALKING ON THE INCLINED
PLANE***

Mochamad Farid Mustofa
NRP 2211100031

Advisor

Ir. Rusdhianto Effendie A.K, M.T.

***ELECTRICAL ENGINEERING DEPARTMENT
Faculty of Industrial Technology
Institut Teknologi Sepuluh Nopember
Surabaya 2016***

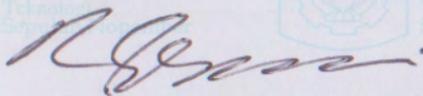
**SISTEM PENGATURAN ROBOT HUMANOID UNTUK BERDIRI
DAN BERJALAN PADA BIDANG MIRING**

TUGAS AKHIR

Diajukan Guna Memenuhi Sebagian Persyaratan
Untuk Memperoleh Gelar Sarjana Teknik
Pada
Bidang Studi Teknik Sistem Pengaturan
Jurusan Teknik Elektro
Institut Teknologi Sepuluh Nopember

Menyetujui :

Dosen Pembimbing



Ir. Rusdhianto Effendie A.K. M.T.

NIP : 195704241985021001



Sistem Pengaturan Robot *Humanoid* untuk Berdiri dan Berjalan pada Bidang Miring

Nama : Mochamad Farid Mustofa
Pembimbing : Ir. Rusdhianto Effendie A.K, M.T.

ABSTRAK

Robot *humanoid* adalah robot yang strukturnya menyerupai struktur tubuh manusia. Pada robot ini keseimbangan menjadi hal yang penting agar robot dapat berdiri dan berjalan dengan sempurna baik pada permukaan bidang datar maupun bidang miring. Untuk dapat melakukannya maka diperlukan suatu sistem yang mengatur keseimbangan robot. Salah satu skema yang digunakan untuk mengatur keseimbangan robot *humanoid* adalah dengan kontroler *upright pose* berbasis PID yang mengukur kemiringan tubuh robot terhadap lantai, kemudian tubuh bagian atas robot bergerak menyesuaikan kemiringan tersebut sehingga keseimbangan tubuh robot dapat tercapai.

Penerapan kontroler *upright pose* pada robot saat kondisi berdiri menghasilkan performa *overshoot* awal sebesar 89% dan mencapai *steady state* setelah 5,45 detik dengan nilai *error steady state* rata – rata sebesar $0,95^0$ ketika kemiringan lantai dinaikkan, dan sebesar $7,23^0$ ketika kemiringan lantai diturunkan pada kemiringan *pitch* dengan nilai parameter PID, $K_p = 0,5$; $K_i = 0,01$; $K_d = 0,01$. Pada gerakan *roll* dengan nilai $K_p = 0,5$; $K_i = 0,05$; dan $K_d = 0,01$, menghasilkan performa *overshoot* awal sebesar 98 % dan mencapai *steady state* setelah 19,35 detik dengan nilai *error steady state* rata – rata sebesar $0,31^0$ ketika kemiringan lantai dinaikkan, dan sebesar $0,52^0$ ketika kemiringan lantai diturunkan. Penerapan kontroler *upright pose* pada kondisi berjalan belum sempurna. Hal ini karena spesifikasi torsi motor servo yang kurang tinggi sehingga desain *walking gait* yang diterapkan menyesuaikan dengan kondisi tersebut serta kendala *delay* komunikasi *wireless* yang berakibat pada melambatnya gerakan robot.

Kata kunci : Robot *Humanoid*, *Inverse* Kinematik, Pola Gerakan Berjalan, Kontroler *Upright pose*

Balance Control System of Humanoid Robot for Standing and Walking on The Inclined Plane

Name : Mochamad Farid Mustofa
Advisor : Ir. Rusdhianto Effendie A.K, M.T.

ABSTRACT

Humanoid robot is a robot which structure is human-like. Balance is a main problem of this robot, especially for standing and walking in the inclined plane. It needs a system that maintain the balance of this robot. One way to maintain the balance of this robot is using upright pose controller based on PID by measure the inclination of the body which related to the floor. Then the torso move to maintain balance so it will not fall down. By combining this method in every step of walking gait, the robot will maintain its balance not only when standing but also walking in the inclined plane.

The result is by using $K_p=0.5$, $K_i=0.01$, and $K_d=0.01$ for roll, there is overshoot 98 % in the begining performance and reach steady state condition in 19.35 seconds. When the floor is inclined from 0^0 to 20^0 there is steady state error about 0.31^0 when the inclination increased and 0.52^0 when the inclination decreased. For pitch with PID parameters $K_p=0.5$, $K_i=0.05$, and $K_d=0.01$, there is 89 % overshoot in the beginning and reach steady state condition in 5.45 seconds but there is steady state error about 0.95^0 when the inclination increased and 7.23^0 when the inclination decreased. For walking control there are many problems in wireless communication and limited torque spesification of servo motor, so it can't well implemented in this systems.

Keyword :Humanoid Robot, Inverse Kinematic, Walking Gait, Upright pose Control

KATA PENGANTAR

Puji syukur atas kehadiran Allah atas segala rahmat, taufik, hidayah dan inayah-Nya sehingga Tugas akhir ini dapat diselesaikan dengan baik.

Tugas akhir ini disusun guna memenuhi sebagian persyaratan menyelesaikan pendidikan Strata-1 pada Bidang Studi Teknik Sistem Pengaturan, Jurusan Teknik Elektro, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember Surabaya, dengan judul “**SISTEM PENGATURAN ROBOT *HUMANOID* UNTUK BERDIRI DAN BERJALAN PADA BIDANG MIRING**”.

Penulis menyadari bahwa dalam penyelesaian pengerjaan Tugas akhir ini tidak terlepas dari bantuan dan dukungan dari berbagai pihak. Untuk itu dengan segala hormat, penulis mengucapkan terima kasih kepada semua pihak yang telah berperan serta dalam penyelesaian Tugas akhir ini, khususnya kepada dosen pembimbing, Ir. Rusdhianto Effendie A.K, M.T. yang selalu sabar membimbing dan memberikan kepercayaan dan motivasi, kedua orang tua yang senantiasa memberikan dukungan, dan semua pihak yang tidak dapat penulis sebutkan satu per satu.

Penulis menyadari bahwa masih banyak kekurangan dalam penelitian maupun dalam penulisan buku ini. Untuk itu, penulis memohon maaf atas segala kekurangan yang ada. Semoga Tugas akhir ini dapat bermanfaat untuk pengembangan ilmu pengetahuan dan teknologi selanjutnya.

Surabaya, Januari 2016

Mochamad Farid Mustofa

DAFTAR ISI

	HALAMAN
ABSTRAK.....	i
<i>ABSTRACT</i>	iii
KATA PENGANTAR.....	v
DAFTAR ISI.....	vii
DAFTAR GAMBAR.....	ix
DAFTAR TABEL.....	xiii
BAB 1 PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Perumusan Masalah.....	2
1.3 Batasan Masalah.....	2
1.4 Tujuan Penelitian.....	2
1.5 Sistematika Penulisan.....	2
1.6 Relevansi.....	3
BAB 2 TEORI DASAR.....	5
2.1 Robot.....	5
2.1.1 Pemodelan Robot Manipulator.....	6
2.1.2 Robot <i>Humanoid</i>	16
2.3 Sistem Pengaturan.....	19
2.4 Kontroler PID.....	21
2.5 Kontroler <i>Upright Pose</i>	22
BAB 3 PERANCANGAN SISTEM.....	25
3.1 Mekanik Robot Bioloid.....	25
3.1.1 Parameter DH Robot Bioloid.....	26
3.1.2 Kerangka Koordinat Robot Bioloid.....	27
3.1.3 <i>Inverse</i> Kinematik Kaki Robot Bioloid.....	28
3.1.4 Perancangan <i>Walking Gait</i> Robot.....	32
3.2 Perangkat Elektronik Robot Bioloid.....	33
3.2.1 Perancangan <i>Driver</i> Motor Servo Dynamixel.....	34
3.2.2 Perancangan Perangkat Kontroler Robot.....	35
3.2.3 <i>Inertial Measurement Unit</i> (IMU).....	36
3.2.4 Perangkat Komunikasi <i>Wireless</i>	37
3.2.5 Baterai Li-Po.....	38
3.2.6 Perangkat Mikrokontroler.....	39
3.3 Perancangan Kontroler Robot.....	42
3.3.1 Perancangan Kontroler <i>Upright Pose</i> untuk Berdiri.....	42

3.3.2 Perancangan Kontroler <i>Upright Pose</i> untuk Berjalan	42
BAB 4 PENGUJIAN DAN ANALISA	43
4.1 Pengujian Mekanik Robot	43
4.1.1 Pengujian Motor Servo Dynamixel AX-12A	43
4.1.2 Pengujian Gerakan Kaki Robot	46
4.2 Pengujian Perangkat Elektronik Robot	47
4.2.1 Pengujian IMU	47
4.3 Pengujian Kontroler	49
4.3.1 Pengujian Kondisi Berdiri Terhadap Gerakan <i>Roll</i>	49
4.3.2 Pengujian Kondisi Berdiri Terhadap Gerakan <i>Pitch</i>	60
4.3.3 Pengujian Kondisi Berjalan	70
BAB 5 PENUTUP	71
5.1 Kesimpulan	71
5.2 Saran	72
DAFTAR PUSTAKA	73
LAMPIRAN	75
RIWAYAT PENULIS	85

DAFTAR GAMBAR

	HALAMAN
Gambar 2.1 Robot ATLAS dari Boston Dynamics.....	5
Gambar 2.2 Robot Manipulator Pengelas Motoman MA1400	6
Gambar 2.3 Simbol Joint Robot Manipulator	8
Gambar 2.4 Konfigurasi Manipulator <i>Articulated</i>	9
Gambar 2.5 Konfigurasi Manipulator <i>Spherical</i>	9
Gambar 2.6 Konfigurasi Manipulator SCARA.....	10
Gambar 2.7 Konfigurasi Manipulator <i>Cylindrical</i>	10
Gambar 2.8 Konfigurasi Manipulator <i>Cartesian</i>	11
Gambar 2.9 <i>Trajectory Cubic</i> pada <i>Joint Robot</i>	16
Gambar 2.10 Robot Humanoid ASIMO buatan HONDA.....	17
Gambar 2.11 Robot Bioloid Premium buatan ROBOTIS	18
Gambar 2.12 <i>Walking Gait</i> Robot Humanoid.....	19
Gambar 2.13 Diagram Blok Sistem Pengaturan	19
Gambar 2.14 Diagram Blok Sistem Pengaturan Loop Terbuka.....	21
Gambar 2.15 Diagram Blok Sistem Pengaturan Loop Tertutup	21
Gambar 2.16 Kontroler <i>Upright Pose</i> pada Kemiringan Sudut <i>Pitch</i> ...	23
Gambar 2.17 Kontroler <i>Upright Pose</i> pada Kemiringan Sudut <i>Roll</i>	24
Gambar 3.1 Rancangan Sistem Pengaturan Keseimbangan Robot <i>Humanoid</i>	25
Gambar 3.2 Ukuran Bentangan Robot Bioloid	26
Gambar 3.3 Ukuran Kaki Robot Bioloid	26
Gambar 3.4 Kerangka Koordinat Kaki Robot Bioloid.....	28
Gambar 3.5 Model Kinematik Kaki Tampak Isometri.....	29
Gambar 3.6 Model Kinematik Kaki Tampak Atas.....	29
Gambar 3.7 Model Kinematik Kaki Tampak Depan	30
Gambar 3.8 Model Kinematik Kaki Tampak Samping.....	31
Gambar 3.9 <i>Walking Gait</i> Robot Tampak Samping.....	32
Gambar 3.10 <i>Walking Gait</i> Robot Tampak Depan	33
Gambar 3.11 Motor Servo Dynamixel AX-12A	34
Gambar 3.12 Rangkaian <i>Driver</i> Motor Dynamixel	35
Gambar 3.13 <i>Schematic Driver</i> Motor Dynamixel	35
Gambar 3.14 <i>Board Controller</i> Robot	36
Gambar 3.15 Sensor IMU MPU6050	36
Gambar 3.16 Modul Komunikasi Wireless Radio NRF24L01+	37
Gambar 3.17 Baterai Li-Po 3 Sel 11,1V 1000mAh	38
Gambar 3.18 Modul Arduino MEGA 2560	40

Gambar 3.19 Modul Arduino UNO	41
Gambar 3.20 Diagram Blok Sistem Pengaturan Keseimbangan Robot <i>Humanoid</i>	42
Gambar 4.1 Uji Linieritas Motor Servo Dynamixel AX-12A.....	45
Gambar 4.2 Uji Histerisis Motor Servo Dynamixel AX-12A.....	45
Gambar 4.3 Keseluruhan Respon Kemiringan Robot Ketika Gerakan <i>Roll</i>	49
Gambar 4.4 Respon Kemiringan Robot Terhadap Gerakan <i>Roll</i> Ketika Kemiringan 0^0	50
Gambar 4.5 Respon Kemiringan Robot Terhadap Gerakan <i>Roll</i> Ketika Terjadi Perubahan Kemiringan $0^0 - 5^0$	51
Gambar 4.6 Respon Kemiringan Robot Terhadap Gerakan <i>Roll</i> Ketika Kemiringan 5^0	52
Gambar 4.7 Respon Kemiringan Robot Terhadap Gerakan <i>Roll</i> Ketika Terjadi Perubahan Kemiringan $5^0 - 10^0$	53
Gambar 4.8 Respon Kemiringan Robot Terhadap Gerakan <i>Roll</i> Ketika Kemiringan 10^0	53
Gambar 4.9 Respon Kemiringan Robot Terhadap Gerakan <i>Roll</i> Ketika Terjadi Perubahan Kemiringan $10^0 - 15^0$	54
Gambar 4.10 Respon Kemiringan Robot Terhadap Gerakan <i>Roll</i> Ketika Kemiringan 15^0	54
Gambar 4.11 Respon Kemiringan Robot Terhadap Gerakan <i>Roll</i> Ketika Terjadi Perubahan Kemiringan $15^0 - 20^0$	55
Gambar 4.12 Respon Kemiringan Robot Terhadap Gerakan <i>Roll</i> Ketika Kemiringan 20^0	55
Gambar 4.13 Respon Kemiringan Robot Terhadap Gerakan <i>Roll</i> Ketika Terjadi Perubahan Kemiringan $20^0 - 15^0$	56
Gambar 4.14 Respon Kemiringan Robot Terhadap Gerakan <i>Roll</i> Ketika Kemiringan 15^0 Kondisi Kemiringan Turun	57
Gambar 4.15 Respon Kemiringan Robot Terhadap Gerakan <i>Roll</i> Ketika Terjadi Perubahan Kemiringan $15^0 - 10^0$	57
Gambar 4.16 Respon Kemiringan Robot Terhadap Gerakan <i>Roll</i> Ketika Kemiringan 10^0 Kondisi Kemiringan Turun	58
Gambar 4.17 Respon Kemiringan Robot Terhadap Gerakan <i>Roll</i> Ketika Terjadi Perubahan Kemiringan $10^0 - 5^0$	58
Gambar 4.18 Respon Kemiringan Robot Terhadap Gerakan <i>Roll</i> Ketika Kemiringan 5^0 Kondisi Kemiringan Turun	58
Gambar 4.19 Respon Kemiringan Robot Terhadap Gerakan <i>Roll</i> Ketika Terjadi Perubahan Kemiringan $5^0 - 0^0$	59

Gambar 4.20	Respon Kemiringan Robot Terhadap Gerakan <i>Roll</i> Ketika Kemiringan 50 Kondisi Kemiringan Turun.....	59
Gambar 4.21	Keseluruhan Respon Kemiringan Robot Ketika Gerakan <i>Pitch</i>	60
Gambar 4.22	Respon Kemiringan Robot Terhadap Gerakan <i>Pitch</i> Ketika Kemiringan 0 ⁰	61
Gambar 4.23	Respon Kemiringan Robot Terhadap Gerakan <i>Pitch</i> Ketika Terjadi Perubahan Kemiringan 0 ⁰ – 5 ⁰	62
Gambar 4.24	Respon Kemiringan Robot Terhadap Gerakan <i>Pitch</i> Ketika Kemiringan 5 ⁰	63
Gambar 4.25	Respon Kemiringan Robot Terhadap Gerakan <i>Pitch</i> Ketika Terjadi Perubahan Kemiringan 5 ⁰ - 10 ⁰	63
Gambar 4.26	Respon Kemiringan Robot Terhadap Gerakan <i>Pitch</i> Ketika Kemiringan 10 ⁰	64
Gambar 4.27	Respon Kemiringan Robot Terhadap Gerakan <i>Pitch</i> Ketika Terjadi Perubahan Kemiringan 10 ⁰ – 15 ⁰	64
Gambar 4.28	Respon Kemiringan Robot Terhadap Gerakan <i>Pitch</i> Ketika Kemiringan 15 ⁰	65
Gambar 4.29	Respon Kemiringan Robot Terhadap Gerakan <i>Pitch</i> Ketika Terjadi Perubahan Kemiringan 15 ⁰ - 20 ⁰	65
Gambar 4.30	Respon Kemiringan Robot Terhadap Gerakan <i>Pitch</i> Ketika Kemiringan 20 ⁰	66
Gambar 4.31	Respon Kemiringan Robot Terhadap Gerakan <i>Pitch</i> Ketika Terjadi Perubahan Kemiringan 20 ⁰ - 15 ⁰	66
Gambar 4.32	Respon Kemiringan Robot Terhadap Gerakan <i>Pitch</i> Ketika Kemiringan 15 ⁰ Kondisi Kemiringan Turun.....	67
Gambar 4.33	Respon Kemiringan Robot Terhadap Gerakan <i>Pitch</i> Ketika Terjadi Perubahan Kemiringan 15 ⁰ - 10 ⁰	67
Gambar 4.34	Respon Kemiringan Robot Terhadap Gerakan <i>Pitch</i> Ketika Kemiringan 10 ⁰ Kondisi Kemiringan Turun.....	68
Gambar 4.35	Respon Kemiringan Robot Terhadap Gerakan <i>Pitch</i> Ketika Terjadi Perubahan Kemiringan 10 ⁰ – 5 ⁰	68
Gambar 4.36	Respon Kemiringan Robot Terhadap Gerakan <i>Pitch</i> Ketika Kemiringan 5 ⁰ Kondisi Kemiringan Turun.....	69
Gambar 4.37	Respon Kemiringan Robot Terhadap Gerakan <i>Pitch</i> Ketika Terjadi Perubahan Kemiringan 5 ⁰ – 0 ⁰	69

DAFTAR TABEL

	HALAMAN
Tabel 3.1 Parameter DH Kaki Kiri Robot Bioloid.....	27
Tabel 3.2 Parameter DH Kaki Kanan Robot Bioloid.....	27
Tabel 3.3 Spesifikasi Sensor IMU MPU 6050.....	37
Tabel 3.4 Spesifikasi NRF24L01+.....	38
Tabel 3.5 Spesifikasi Baterai Li-Po.....	39
Tabel 3.6 Spesifikasi Arduino Mega2560.....	40
Tabel 3.7 Spesifikasi Arduino UNO.....	41
Tabel 4.1 Data Pengujian Motor Servo.....	44
Tabel 4.2 Data Pengujian <i>Inverse</i> Kinematik Posisi pada Sumbu X.....	46
Tabel 4.3 Data Pengujian <i>Inverse</i> Kinematik Posisi pada Sumbu Y.....	46
Tabel 4.4 Data Pengujian <i>Inverse</i> Kinematik Posisi pada Sumbu Z.....	47
Tabel 4.5 Data Pengujian Perangkat IMU pada Sudut <i>Roll</i>	48
Tabel 4.6 Data Pengujian Perangkat IMU pada Sudut <i>Pitch</i>	48

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Penelitian robot *humanoid* banyak dikembangkan di berbagai bidang meliputi kecerdasan tiruan, pengembangan *hardware*, realisasi gerakan berjalan, keseimbangan robot, serta interaksi dengan lingkungan sekitar. Hal ini dapat terjadi karena teknologi di bidang ini berkembang begitu pesat sehingga berdampak langsung pada penelitian di bidang ini.

Hal istimewa yang penting dari robot *humanoid* adalah bentuk dan gerakannya mirip dengan manusia. Jenis robot ini diharapkan bisa berjalan dan bergerak seperti manusia sungguhan. Selain itu juga bisa memanjat, berjalan pada bidang yang tidak rata, berlari, dan , menari.

Dari berbagai bidang penelitian tersebut, penelitian tentang keseimbangan robot ketika berdiri atau berjalan menjadi hal yang utama sebelum dapat melakukan gerakan – gerakan yang lainnya. Keseimbangan tersebut menjadi sangat penting ketika jalur yang dilalui berupa bidang miring.

Untuk dapat menyelesaikan masalah tersebut dibutuhkan suatu sistem pengaturan keseimbangan robot *humanoid* untuk bisa berdiri dan berjalan baik pada permukaan datar atau pada bidang miring. Pendekatan yang dilakukan adalah mengatur posisi tubuh bagian atas robot. Dengan pendekatan ini maka keseimbangan robot dapat tercapai pada saat posisinya berdiri ataupun berjalan dengan mengubah pose robot yakni dengan mengubah sudut – sudut *joint* yang mempengaruhi pose tersebut sehingga keseimbangan robot tetap terjaga.

Untuk merealisasikan metode tersebut maka digunakan perangkat kontroler dalam hal ini adalah Arduino Mega 2560, karena memiliki kemampuan yang cukup mumpuni untuk melakukan komputasi yang cukup rumit serta harga yang tidak terlalu mahal. Selain itu dibutuhkan pula metode kontrol yang sederhana dengan komputasi yang ringan namun mampu menangani permasalahan ini karena metode tersebut akan diterapkan pada perangkat kontroler. Metode kontrol yang diterapkan adalah kontroler *upright pose* berbasis PID karena memiliki struktur yang sederhana serta komputasi ringan namun mumpuni sehingga sesuai untuk perangkat mikrokontroler yang digunakan.

1.2 Perumusan Masalah

Keseimbangan robot merupakan hal yang penting pada robot *humanoid* agar tidak jatuh saat berdiri dan untuk dapat berdiri dan berjalan terutama pada bidang miring. Untuk itu diperlukan suatu metode kontrol yang mampu melakukan hal tersebut.

1.3 Batasan Masalah

Agar pengerjaan tugas akhir ini dapat terarah tanpa mengurangi maksud dan tujuannya, maka ditentukan batasan permasalahan sebagai berikut :

1. Melakukan pemodelan *inverse* kinematik hanya pada bagian kaki robot.
2. Bagian robot selain kaki dianggap sebagai benda tegar.
3. Menggunakan robot *humanoid* 6 DOF pada masing – masing kaki.
4. Data dari IMU yang digunakan sebagai sensor kemiringan robot adalah data *gyroscope*.
5. Perangkat kontroler yang digunakan adalah Arduino Mega 2560.
6. Kecepatan jalan robot 7 cm/s.
7. Kemiringan bidang miring antara 0^0 sampai dengan 20^0 .
8. Arah jalan maju dan menaiki bidang miring.

1.4 Tujuan Penelitian

Penelitian ini dilakukan untuk membuat sistem pengaturan keseimbangan dengan menggunakan informasi kemiringan yang diperoleh dari sensor *gyroscope* yang diterapkan pada robot *humanoid* sehingga robot tersebut dapat menjaga keseimbangannya pada saat berdiri atau berjalan pada bidang miring.

1.5 Sistematika Penulisan

Tugas akhir ini disusun sesuai dengan sistematika penulisan sebagai berikut:

BAB I : PENDAHULUAN

Bab ini berisi tentang pendahuluan yang terdiri dari latar belakang, perumusan masalah, batasan masalah, tujuan penelitian, sistematika penulisan, serta relevansi dari tugas akhir ini.

BAB II : TEORI DASAR

Bab ini membahas mengenai teori – teori yang berkaitan dengan topik tugas akhir yang diperoleh dari buku, jurnal, *datasheet*, artikel dan sumber – sumber lainnya.

BAB III : PERANCANGAN SISTEM

Bab ini membahas mengenai perancangan sistem, meliputi perancangan *hardware*, pemodelan kinematik, dan perancangan kontroler.

BAB IV : PENGUJIAN DAN ANALISA SISTEM

Bab ini membahas seluruh hasil dan analisa data dari implementasi sistem yang dirancang dalam tugas akhir ini.

BAB V : PENUTUP

Bab ini berisi kesimpulan hasil analisa dari implementasi sistem yang telah dirancang, serta saran untuk penyempurnaan sistem yang telah dibuat.

1.6 Relevansi

Hasil yang didapatkan dari tugas akhir ini diharapkan menjadi salah satu referensi yang digunakan oleh mahasiswa lain yang ingin menggunakan metode tersebut.

BAB 2 TEORI DASAR

Bab ini berisi materi – materi yang menjadi landasan teori dalam pengerjaan tugas akhir ini. Beberapa landasan teori yang digunakan antara lain mengenai robot, kinematika robot, dan kontroler yang digunakan.

2.1 Robot [1]

Istilah Robot berasal dari bahasa Ceko “*robota*” yang berarti pekerja atau kuli. Istilah ini pertama kali diperkenalkan oleh seorang penulis drama yang berasal dari Ceko bernama Karel Capek pada tahun 1920 pada pertunjukan drama *Rossum’s Universal Robots*. Sejak saat itu istilah ini berkembang dan diterapkan di berbagai macam perangkat mekanik, *teleoperators*, kendaraan dalam air, dan lainnya.



Gambar 2.1 Robot ATLAS dari Boston Dynamics [2]

Definisi robot menurut *Robot Institute of America* (RIA) adalah manipulator multifungsi yang dapat diprogram ulang yang dirancang untuk memindahkan material, benda, peralatan, atau piranti khusus melalui gerakan yang telah dirancang untuk melakukan berbagai macam tugas. Robot sesuai dengan yang telah di definisikan berawal dari perpaduan dua teknologi yaitu *teleoperators* dan *numerically controlled milling machines*. *Teleoperators* atau perangkat *master-slave* dikembangkan selama Perang Dunia II untuk menangani material radioaktif. Sementara *Computer Numerical Control* (CNC) dikembangkan pula pada saat itu karena kebutuhan terhadap tingkat

kepresisian yang tinggi dari beberapa piranti, seperti komponen pada pesawat terbang. Robot yang pertama merupakan penggabungan rangkaian mekanik dari *teleoperator* dengan mesin CNC yang dapat diprogram dan bekerja secara otomatis. Robot tersebut awalnya diterapkan pada mesin pemindah material, pengelas, dan pengepres.

Jenis robot dapat dikelompokkan menjadi dua berdasarkan kemampuan seluruh bagian mekaniknya untuk berpindah tempat, yakni robot *mobile* dan *fixed robot*. Pada perkembangannya banyak sekali jenis robot yang dibuat untuk memenuhi berbagai kebutuhan, salah satunya adalah robot *humanoid* yang ditunjukkan oleh Gambar 2.1.

2.1.1 Pemodelan Robot Manipulator

Robot manipulator tersusun atas rangkaian mekanik yang kompleks, seperti yang ditunjukkan pada Gambar 2.2, sehingga untuk dapat dilakukan suatu analisa maka dibutuhkan pemodelan robot tersebut. Pemodelan bisa secara dua dimensi atau tiga dimensi. Pemodelan robot meliputi identifikasi bagian – bagian mekanik robot, representasi simbol dari masing – masing elemen mekanik penyusun robot, kemudian dari elemen – elemen tersebut akan dirangkai membentuk konfigurasi tertentu. Setelah terbentuk konfigurasi, maka gerakan robot dapat dianalisa menggunakan transformasi homogen mengenai posisi dan orientasi dari ujung robot manipulator. Selain itu untuk analisa lebih dalam mengenai gerakan robot manipulator melibatkan peranan kinematika robot.



Gambar 2.2 Robot Manipulator Pengelas Motoman MA1400 [3]

2.1.1.1 Bagian – Bagian Mekanik Robot Manipulator

Salah satu sistem penyusun suatu robot manipulator adalah sistem mekanik. Sehingga sebagai suatu sistem, robot memiliki elemen – elemen penyusun. Bagian – bagian mekanik robot manipulator tersusun atas bagian yang dapat bergerak dan bagian yang diam, kemudian dari bagian yang bergerak dibagi menjadi elemen penggerak dan elemen yang digerakkan. Secara rinci, bagian – bagian mekanik robot manipulator meliputi :

a. *Base*

Base adalah bagian pondasi robot manipulator, atau tempat rangkaian robot bermula. Bagian ini termasuk dalam bagian mekanik yang diam serta merupakan titik referensi dari seluruh sumbu koordinat yang digunakan pada robot.

b. *Joint*

Joint adalah bagian robot manipulator yang bisa bergerak secara *independent* dan merupakan penggerak dari bagian yang terhubung dengannya. Berdasarkan gerakannya, secara umum terdapat dua jenis *joint* yaitu *revolute joint* yang gerakannya berputar atau gerak rotasi dan *prismatic joint* yang gerakannya bergeser atau berupa gerakan translasi. Besarnya sudut putar pada *revolute joint* atau pun geseran pada *prismatic joint* disebut dengan variabel *joint*.

c. *Link*

Link adalah bagian robot manipulator yang menghubungkan seluruh *joint* sehingga terbentuk sebuah rangkaian robot manipulator. Bagian ini termasuk dalam bagian mekanik yang tidak bisa bergerak secara *independent*, sehingga untuk menggerakkannya dibutuhkan suatu penggerak.

d. *End – Effector*

End - Effector adalah ujung dari untaian mekanik pada robot manipulator.

e. *Tools (Optional)*

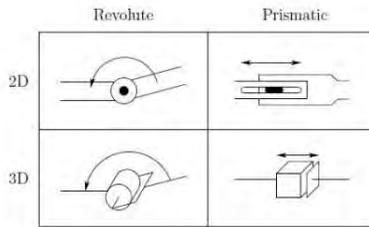
Bagian ini adalah peralatan tambahan yang disesuaikan dengan tujuan robot itu dibuat, atau tugas dari robot tersebut. Misalnya robot manipulator pengelas, maka *tools* nya berupa mata las beserta pakan las nya, begitu pula dengan yang lain disesuaikan dengan

kegunaannya. Bagian ini biasanya terletak pada ujung untaian mekanik dari robot.

2.1.1.2 Representasi Simbol Robot Manipulator

Simbol adalah lambang yang mewakili suatu informasi. Penyampaian informasi melalui simbol termasuk cara yang paling kuno namun efektif. Dalam penyampaian suatu informasi bisa saja terjadi suatu kesalahan dalam menangkap informasi tersebut. Salah satu penyebab terjadinya hal ini adalah adanya perbedaan dalam penggunaan simbol. Sehingga perlu dibuat suatu simbol yang baku beserta aturan penggunaannya untuk mengatasi hal tersebut.

Penggunaan simbol juga diberlakukan pada pemodelan robot manipulator yang melambangkan bagian – bagian mekanik dan penggunaannya, sehingga terdapat aturan baku mengenai penggunaan simbol tersebut. Simbol yang digunakan pada pemodelan robot manipulator khususnya untuk *joint* robot ditunjukkan oleh Gambar 2.3.



Gambar 2.3 Simbol *Joint* Robot Manipulator

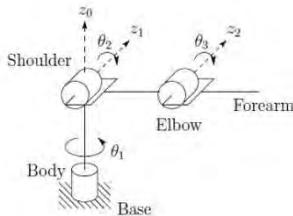
revolute joint digambarkan seperti lingkaran pada bidang 2D dan seperti tabung untuk bidang 3D, sedangkan *prismatic joint* digambarkan pada bidang 2D seperti dua buah persegi panjang yang terkait dan dapat bergeser dan seperti kubus yang salah satu tutupnya dapat memanjang pada bidang 3D. Untuk kemudahan penulisan *revolute joint* diwakili oleh huruf R dan *prismatic joint* diwakili oleh huruf P.

Jumlah *joint* pada robot menentukan tingkat kebebasan gerak dari robot atau yang disebut dengan *Degree of Freedom (DOF)*. Pada ruang tiga dimensi dibutuhkan minimal 6 DOF yaitu 3 DOF posisi dan 3 DOF orientasi agar robot dapat menjangkau semua area di ruang konfigurasinya.

2.1.1.3 Konfigurasi Umum Mekanik Robot Manipulator

Untuk dapat bekerja sesuai dengan tujuan, maka masing – masing bagian penyusun robot harus digabungkan membentuk suatu untaian mekanik. Dari macam – macam pilihan *joint* beserta bagian mekanik lainnya, dapat dibuat beberapa konfigurasi sehingga terbentuk sebuah robot manipulator yang memiliki perbedaan karakteristik gerakan. Beberapa konfigurasi mekanik yang umum pada robot manipulator ditunjukkan oleh Gambar 2.4 sampai Gambar 2.8.

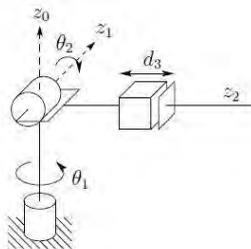
a. *Articulated* (RRR)



Gambar 2.4 Konfigurasi Manipulator *Articulated*

Konfigurasi ini terdiri atas 3 *revolute joint* yang disusun *serial* yang mirip dengan lengan manusia namun dengan derajat kebebasan yang lebih sedikit.

b. *Spherical* (RRP)

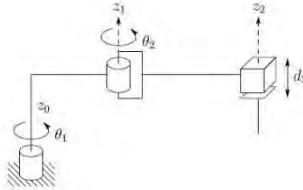


Gambar 2.5 Konfigurasi Manipulator *Spherical*

Konfigurasi ini tersusun atas dua buah *revolute joint* dan satu buah *prismatic joint*. Penamaan *spherical* disesuaikan dengan kenyataan bahwa koordinat *spherical* (bola) mendefinisikan posisi

dari *end-effector* terhadap *frame* yang pusatnya terletak pada perpotongan tiga sumbu z.

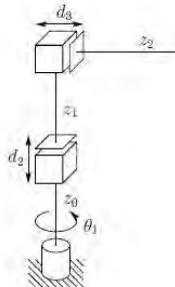
c. SCARA (RRP)



Gambar 2.6 Konfigurasi Manipulator SCARA

SCARA (*Selective Compliant Articulated Robot for Assembly*) merupakan salah satu manipulator yang terkenal dan namanya sesuai dengan tugasnya yaitu untuk pekerjaan perakitan. Meskipun sama – sama memiliki struktut RRP namun sedikit berbeda dengan manipulator *spherical* pada susunan *joint* nya dan penggunaannya. Berbeda dengan manipulator *spherical*, yang sumbu masing – masing sumbu z nya saling tegk lurus, pada SCARA sumbu z dari *joint* nya saling sejajar satu sama lain.

d. *Cylindrical* (RPP)

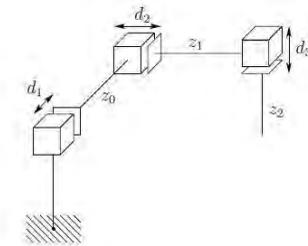


Gambar 2.7 Konfigurasi Manipulator *Cylindrical*

Konfigurasi ini tersusun atas satu *revolute joint* dan dua *prismatic joint*. Gerakan rotasi dibentuk oleh *joint* pertama yang

berupa *revolute joint*, sementara gerakan memanjang dari *joint* ke dua dan ke tiga. Sesuai dengan namanya, variabel *joint* akan membentuk koordinat tabung pada *end – effector* terhadap *base*.

e. *Cartesian* (PPP)



Gambar 2.8 Konfigurasi Manipulator *Cartesian*

Manipulator ini tersusun atas tiga buah *prismatic joint*. dari kombinasi variabel *joint* pada robot ini terbentuk koordinat *cartesian* pada *end-effector* terhadap *base*.

2.1.1.4 Transformasi Homogen

Sebuah konfigurasi robot akan menghasilkan gerakan tertentu yang berbeda dengan konfigurasi yang lain, sehingga diperlukan suatu cara untuk menganalisa gerakan dari robot tersebut. Cara yang umum digunakan untuk menganalisa gerakan robot adalah menggunakan pendekatan transformasi geometri pada sistem koordinat.

Transformasi pada sistem koordinat disesuaikan dengan perilaku dari masing – masing bagian mekanik penyusun robot terutama *joint* robot. Bagian ini adalah bagian yang bisa bergerak, dan memiliki dua macam gerakan sesuai dengan jenisnya yaitu gerakan memutar dan gerakan bergeser. Sehingga untuk merepresentasikan gerakan pada *joint* tersebut, maka pada sumbu koordinat yang terlibat dalam sistem koordinat robot juga akan diperlakukan seperti *joint* yaitu diputar atau digeser sesuai dengan jenis *joint* yang digunakan dan konfigurasi yang terbentuk.

Pada robot, ketika masing – masing *joint* bergerak, maka *end - effector* akan ikut bergerak. Konfigurasi yang terbentuk mengakibatkan gerakan *end - effector* menjadi kompleks, yaitu bisa berupa gerakan

bergeser, berputar, atau gabungan dari keduanya. Gerakan bergeser *end-effector* berpengaruh pada posisinya, sedangkan gerakan memutar berpengaruh pada orientasinya. Untuk memudahkan analisa, maka diperlukan suatu bentuk transformasi yang bisa memetakan posisi dan orientasi dari *end-effector* secara seketika. Suatu transformasi yang bisa melakukan hal tersebut adalah transformasi homogen. Transformasi homogen merupakan sebuah matriks berdimensi 4x4 yang tersusun atas empat submatriks, seperti pada Persamaan (2.1)

$$H = \begin{bmatrix} R_{3 \times 3} & d_{3 \times 1} \\ f_{1 \times 3} & s_{1 \times 1} \end{bmatrix} = \begin{bmatrix} \text{Rotation} & \text{Translation} \\ \text{Perspective} & \text{Scale Factor} \end{bmatrix} \quad (2.1)$$

Matriks transformasi homogen untuk rotasi terhadap sumbu x, y, z ditunjukkan oleh Persamaan (2.2) – (2.4) sedangkan untuk translasi pada sumbu x, y, z ditunjukkan oleh Persamaan (2.5) – (2.7).

$$Rot_{x,\theta} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \theta & -\sin \theta & 0 \\ 0 & \sin \theta & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.2)$$

$$Rot_{y,\theta} = \begin{bmatrix} \cos \theta & 0 & \sin \theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \theta & 0 & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.3)$$

$$Rot_{z,\theta} = \begin{bmatrix} \cos \theta & -\sin \theta & 0 & 0 \\ \sin \theta & \cos \theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.4)$$

$$Trans_{x,a} = \begin{bmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.5)$$

$$Trans_{y,b} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & b \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.6)$$

$$Trans_{z,c} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & c \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.7)$$

2.1.1.5 Kinematika Robot Manipulator

Kinematika robot manipulator merupakan analisa geometri mengenai gerak robot dengan mengacu pada referensi sistem koordinat yang tetap tanpa memperhatikan gaya – gaya yang menyebabkannya. Hal ini sangat diperlukan untuk analisa gerak pada robot. Sehingga untuk menggerakkan bagian *end – effector* ke posisi yang diinginkan dapat dilakukan dengan lebih mudah. Ada dua macam kinematika yang digunakan pada robot yaitu *forward* kinematik dan *inverse* kinematik.

2.1.1.5.1 Forward Kinematik

Forward kinematik merupakan analisa gerak ketika yang diketahui adalah nilai variabel *joint* kemudian yang dicari adalah posisi dan orientasi dari *end – effector*. Dengan menggunakan kinematika, pemodelan gerakan robot khususnya bagian *end – effector* bisa lebih terarah. Terdapat aturan umum yang baku terkait hal ini yaitu aturan *Denavit – Hartenberg* (DH). Aturan ini memberikan tuntunan yang sistematis untuk analisa kinematika robot meskipun sebenarnya dapat pula dilakukan analisis tanpa menggunakan aturan ini. Aturan ini menyatakan bahwa tiap matriks transformasi homogen dinyatakan sebagai perkalian dari empat transformasi dasar yang melibatkan parameter *link* dan *joint* robot. Selain mengatur tentang parameter *link* dan *joint*, aturan ini juga mengatur penetapan kerangka koordinat pada robot. Aturan DH mengenai parameter *joint* dan *link* adalah sebagai berikut :

a_i = jarak sepanjang x_i dari o_i ke perpotongan sumbu x_i dan z_{i-1} .

d_i = jarak sepanjang z_{i-1} dari o_{i-1} ke perpotongan sumbu x_i dan z_{i-1} . d_i berupa variabel jika *joint* i adalah *prismatic joint*.

α_i = sudut antara z_{i-1} dan z_i diukur terhadap x_i .

θ_i = sudut antara x_{i-1} dan x_i diukur terhadap z_i , θ_i berupa variabel jika *joint* i adalah *revolute joint*.

Langkah – langkah untuk memperoleh *forward* kinematik adalah sebagai berikut:

1. Tempatkan dan beri label z_0, \dots, z_{n-1} pada sumbu putar *joint* atau sumbu geser *joint*.
2. Tetapkan Base Frame. Tentukan titik *origin* pada sumbu z_0 . Sumbu z_0 . Sumbu x_0 dan y_0 dipilih sembarang sehingga membentuk *right-hand frame*.
3. Tempatkan *origin* o_i ke z_i dan z_{i-1} memotong z_i .
4. Tetapkan x_i sepanjang *common normal* antara z_{i-1} dan z_i melalui o_i .
5. Tetapkan y_i untuk melengkapi *right-hand frame*.
6. Tetapkan *end-effector frame* pada o_n, x_n, y_n, z_n .
7. Buat Tabel parameter *link* $a_i, d_i, \alpha_i, \theta_i$.
8. Bentuk matriks transformasi homogen A_n dengan melakukan substitusi parameter.
9. Bentuk matriks *forward* kinematik $T_0^n = A_1 \dots A_n$. Matriks ini memberikan keluaran berupa posisi dan orientasi dari *end-effector frame* dalam koordinat dasar.

Berdasarkan ketentuan DH, setiap transformasi homogen direpresentasikan sebagai sebuah hasil dari empat transformasi dasar seperti yang ditunjukkan pada Persamaan (2.8).

$$\begin{aligned}
 {}_{i-1}^i T &= Rot_{z, \theta_i} \cdot Trans_{z, d_i} \cdot Trans_{x, a_i} \cdot Rot_{x, \alpha_i} \\
 &= \begin{bmatrix} c_{\theta_i} & -s_{\theta_i} & 0 & 0 \\ s_{\theta_i} & c_{\theta_i} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
 &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c_{\alpha_i} & -s_{\alpha_i} & 0 \\ 0 & s_{\alpha_i} & c_{\alpha_i} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
 &= \begin{bmatrix} c_{\theta_i} & -s_{\theta_i} c_{\alpha_i} & s_{\theta_i} c_{\alpha_i} & a_i c_{\theta_i} \\ s_{\theta_i} & c_{\theta_i} c_{\alpha_i} & -c_{\theta_i} s_{\alpha_i} & a_i s_{\theta_i} \\ 0 & s_{\alpha_i} & c_{\alpha_i} & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}
 \end{aligned}$$

$$= \begin{bmatrix} R_i & p_i \\ 0 & 1 \end{bmatrix} \quad (2.8)$$

2.1.1.5.2 Inverse Kinematik

Inverse kinematik merupakan kebalikan dari *forward* kinematik yaitu analisa gerak ketika diketahui posisi dan orientasi *end – effector* kemudian yang dicari adalah nilai dari variabel *joint*. Penggunaan *inverse* kinematik lebih banyak digunakan pada robot, karena untuk menggerakkan *end – effector* tinggal diberikan koordinat yang ingin dituju dan fungsi *inverse* kinematik akan menghitung nilai variabel *joint*. Namun *inverse* kinematik lebih rumit dibandingkan dengan *forward* kinematik terutama mengenai solusinya yang bisa memiliki banyak solusi namun bisa juga tidak memiliki solusi, sehingga sulit untuk mencari rumusan yang general mengenai metode penyelesaian *inverse* kinematik.

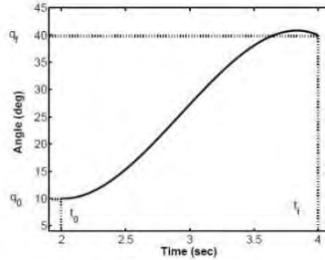
Salah satu metode yang digunakan untuk menyelesaikan masalah ini adalah metode kinematik *decoupling* yang memecah masalah tersebut menjadi dua bagian yaitu *inverse* posisi dan *inverse* orientasi. *Inverse* posisi kemudian diselesaikan dengan pendekatan geometri terutama trigonometri. Sedangkan *inverse* Orientasi diselesaikan dengan pendekatan sudut Euler pada matriks rotasi.

2.1.1.5.3 Trajectory Planning

Trajectory planning merupakan perencanaan jalur gerak robot yang terdiri dari kumpulan titik – titik yang akan dilalui oleh robot ketika bergerak. Dengan membuat perencanaan jalur maka gerakan robot dapat lebih terarah dan lebih halus saat bergerak.

2.1.1.5.3.1 Trajectory untuk gerakan titik ke titik

Fungsi dasar dari suatu robot adalah dapat bergerak dari satu titik menuju titik yang lain. Dalam perjalanannya mencapai titik tujuan, robot tidak seketika langsung sampai begitu saja. Untuk membuat gerakan tersebut menjadi lebih halus maka perlu dibuat perencanaan lintasan gerakannya yakni berupa titik – titik yang dilalui sampai mencapai titik tujuan. Salah satu cara untuk merancang jalur gerak yang halus adalah dengan menggunakan fungsi polinomial dengan empat koefisien yang dapat ditentukan yang disebut dengan *trajectory cubic*.



Gambar 2.9 *Trajectory Cubic* pada *Joint Robot*

Bentuk polinomial pada *trajectory cubic* ditunjukkan oleh Persamaan (2.9).

$$q(t) = a_0 + a_1t + a_2t^2 + a_3t^3 \quad (2.9)$$

Dengan menurunkan Persamaan (2.9) dapat diperoleh polinomial untuk *trajectory* kecepatannya yang ditunjukkan oleh Persamaan (2.10)

$$\dot{q}(t) = a_1 + 2a_2t + 3a_3t^2 \quad (2.10)$$

Koefisien dari polinomial tersebut dapat diperoleh dengan menggunakan matriks Vandermonde dan melakukan eliminasi Gauss pada matriks tersebut. Bentuk matriks Vandermonde ditunjukkan pada Persamaan (2.11).

$$\begin{bmatrix} 1 & x_0 & x_0^2 & \dots & x_0^{n-1} & x_0^n \\ 1 & x_1 & x_1^2 & \dots & x_1^{n-1} & x_1^n \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_{n-1} & x_{n-1}^2 & \dots & x_{n-1}^{n-1} & x_{n-1}^n \\ 1 & x_n & x_n^2 & \dots & x_n^{n-1} & x_n^n \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_{n-1} \\ a_n \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_{n-1} \\ y_n \end{bmatrix} \quad (2.11)$$

2.1.2 Robot *Humanoid* [4]

Robot *humanoid* adalah salah satu jenis robot yang penampilannya secara keseluruhan didasarkan pada bentuk manusia. Struktur robot ini secara keseluruhan memiliki dua kaki dengan tubuh utama bagian atas menghubungkan dua lengan dan leher dan kepala atau sebagai kombinasi dari beberapa manipulator yang dengan sendirinya dihubungkan bersama melalui sendi pinggang dan leher untuk meniru fungsi gerak pada

manusia. Salah satu robot *humanoid* yang terkenal adalah robot ASIMO buatan HONDA yang ditunjukkan oleh Gambar 2.10.



Gambar 2.10 Robot *Humanoid* ASIMO buatan HONDA [5]

Penelitian tentang robot *humanoid* seringkali berkaitan dengan pola gerakannya ketika berjalan, pengaturan keseimbangan, serta interaksi dengan lingkungan sekitar. Khusus untuk masalah pola gerakan dan pengaturan keseimbangan, keduanya saling terkait satu sama lain, karena robot ini meniru cara berjalan manusia untuk berpindah tempat yakni dengan berjalan tegak maka hal ini berbeda dengan model – model robot lainnya.

Beberapa metode yang digunakan dalam mengontrol gerakan robot *humanoid* adalah dengan menggunakan metode *iterative*, salah satunya adalah menggunakan matriks *Jacobian*. Namun penggunaan metode ini memiliki kekurangan terutama pada komputasinya yang cukup berat, karena metode ini tidak berdasarkan kecepatan pemrosesan, namun lebih mengutamakan ketepatan, sehingga adanya akumulasi dari *error* pada posisi robot dengan menggunakan metode *iterative* sangat mempengaruhi komputasi.

2.1.2.1 Robot Bioloid [6]

Robot Bioloid adalah salah satu kit robot buatan perusahaan Robotis asal Korea Selatan yang dibuat dengan tujuan untuk pembelajaran. Kit ini terdiri dari beberapa *part* yang dapat dirakit sesuai keinginan. Beberapa contoh model robot yang ingin dibuat juga disertakan pada kit ini salah satunya model Robot *humanoid*. Selain komponen mekanik, pada kit ini

dilengkapi pula dengan kontroler beserta *software* untuk memrogram gerakan robot yang telah dirakit. Serta beberapa modul sensor seperti sensor suara, sensor jarak, sensor *gyroscope* serta pengendali jarak jauh dengan *remote control*. Bentuk robot Bioloid yang dirangkai menjadi robot *humanoid* ditunjukkan oleh Gambar 2.11.

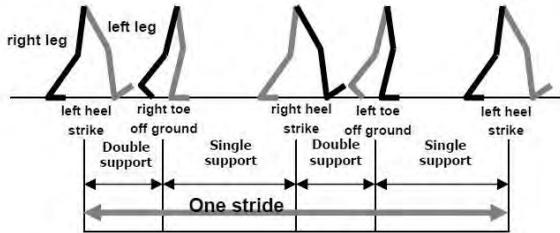


Gambar 2.11 Robot Bioloid Premium buatan ROBOTIS

Aktuator yang digunakan pada robot ini adalah motor DC servo Dynamixel tipe AX-12A. Motor servo Dynamixel sedikit berbeda dengan motor DC servo konvensional yang menggunakan PWM untuk mengontrol sudut putarnya. Untuk mengakses motor ini menggunakan komunikasi data dengan protokol khusus dengan komunikasi *serial*.

2.1.2.3 Walking Gait Robot Humanoid

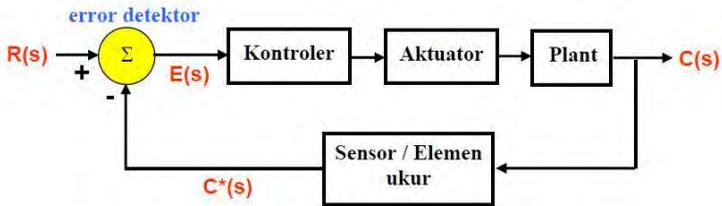
Robot *humanoid* memiliki cara berjalan yang unik meniru cara berjalan manusia meskipun tidak sama persis. Gerakan berjalan robot dapat dimodelkan sebagai suatu siklus yang berulang. Pola gerakan yang berulang ini sering disebut juga sebagai *gait*. Pola tersebut membentuk gerakan berjalan sehingga disebut dengan *walking gait*. Secara garis besar *walking gait* robot *humanoid* dibagi menjadi dua fase gerakan yaitu *Double Support Phase* (DSP) dan *Single Support Phase* (SSP). Fase DSP adalah ketika dua kaki menopang badan robot, sedangkan fase SSP adalah ketika hanya salah satu kaki yang menopang badan robot. *Walking gait* dari robot *humanoid* dapat ditunjukkan pada Gambar 2.12.



Gambar 2.12 *Walking Gait Robot Humanoid* [7]

2.3 Sistem Pengaturan [8]

Sistem merupakan sekumpulan elemen yang saling berhubungan dan bekerja sama untuk mencapai tujuan tertentu. Sistem tidak dibatasi hanya untuk sistem fisik saja. Konsep sistem dapat digunakan pada gejala abstrak dan dinamis seperti yang dijumpai dalam ekonomi. Sistem pengaturan merupakan suatu sistem yang komponen – komponennya telah dikonfigurasi untuk menghasilkan karakteristik sistem yang diinginkan. Salah satu penerapan dari sistem ini adalah pada bidang robotika, misalnya untuk mengatur robot agar berjalan mengikuti jalur tertentu yang sudah dibuat. Penggambaran sistem pengaturan dinyatakan dengan blok diagram seperti pada Gambar 2.13.



Gambar 2.13 Diagram Blok Sistem Pengaturan

2.3.1 Komponen Sistem Pengaturan

Sistem tersusun atas beberapa komponen yang memiliki tugas dan peran masing – masing sehingga dapat mencapai tujuan yang diinginkan, termasuk pada sistem pengaturan. Komponen penyusun sistem pengaturan adalah sebagai berikut:

- a. *Error* Detektor
Error detektor merupakan komponen pada sistem pengaturan yang digunakan untuk membandingkan sinyal keluaran sebenarnya atau sinyal keluaran terukur dengan sinyal masukan acuan (*set point*).
- b. Kontroler
Kontroler merupakan salah satu komponen sistem pengaturan yang berfungsi mengolah sinyal umpan balik dan sinyal masukan acuan (*set point*) atau sinyal *error* menjadi sinyal kontrol. Sinyal *error* dalam hal ini merupakan selisih antara sinyal umpan balik dengan sinyal masukan acuan (*set point*).
- c. Aktuator
Aktuator merupakan komponen penguat dan pengkonversi daya yang berfungsi untuk menguatkan sinyal kontrol yang berasal dari kontroler menjadi sinyal baru dengan daya yang besar sesuai dengan daya yang dibutuhkan oleh *plant*.
- d. *Plant*
Plant adalah seperangkat peralatan yang bekerja bersama – sama, yang digunakan untuk melakukan suatu operasi tertentu. Pada sistem pengaturan obyek fisik yang dikontrol disebut *plant*.
- e. Proses
Proses adalah operasi perkembangan alamiah yang berlangsung secara kontinyu yang ditandai oleh suatu deretan perubahan kecil yang berurutan dengan cara yang relatif tetap dan menuju suatu hasil atau keadaan tertentu. Pada umumnya setiap operasi yang dikontrol disebut proses.
- f. Sensor dan *Transducer*
Transducer merupakan perangkat fisik yang digunakan untuk mentransformasikan suatu sinyal dari bentuk energi yang satu menjadi bentuk energi yang lain atau dari besaran fisik yang satu menjadi besaran fisik yang lain. *Transducer* yang

mentransformasikan suatu besaran fisik menjadi besaran listrik disebut sensor.

2.3.2 Sistem Pengaturan Loop Terbuka

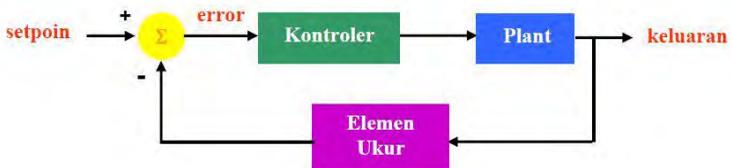
Sistem pengaturan *loop* terbuka merupakan suatu sistem pengaturan yang keluarannya tidak mempunyai pengaruh terhadap aksi kontrol. Pada sistem pengaturan *loop* terbuka keluarannya tidak dapat digunakan sebagai perbandingan umpan balik dengan masukan acuan (*set point*). Diagram blok sistem pengaturan *loop* terbuka ditunjukkan oleh Gambar 2.14.



Gambar 2.14 Diagram Blok Sistem Pengaturan *Loop* Terbuka

2.3.3 Sistem Pengaturan Loop Tertutup

Sistem pengaturan *loop* tertutup merupakan suatu sistem pengaturan dimana sinyal keluaran mempunyai pengaruh langsung terhadap aksi kontrol. Pada sistem pengaturan *loop* tertutup terdapat jaringan umpan balik dari sinyal keluaran yang akan dibandingkan dengan sinyal masukan acuan sehingga menghasilkan sinyal *error* yang akan menjadi masukan dari kontroler. Diagram blok sistem pengaturan *loop* tertutup ditunjukkan oleh Gambar 2.15.



Gambar 2.15 Diagram Blok Sistem Pengaturan *Loop* Tertutup

2.4 Kontroler PID [8]

Salah satu jenis kontroler berdasarkan aksi kontrolnya adalah kontroler proporsional ditambah integral ditambah differensial (PID). Kontroler PID merupakan kontroler yang aksi kontrolnya merupakan

kombinasi dari aksi kontrol proporsional, aksi kontrol integral dan aksi kontrol differensial.

Untuk kontroler PID, sinyal *error* menjadi masukan kontroler sedangkan keluaran kontroler adalah sinyal kontrol. Hubungan antara masukan kontroler dan keluaran kontroler ditunjukkan oleh Persamaan (2.12).

$$u(t) = K_p \left\{ e(t) + \frac{1}{\tau_i} \int_0^t e(t) dt + \tau_d \frac{de(t)}{dt} \right\} \quad (2.12)$$

K_p merupakan konstanta Proporsional, $e(t)$ merupakan sinyal *error*, τ_i adalah konstanta waktu integral, dan τ_d merupakan konstanta waktu differensial.

2.5 Kontroler *Upright Pose* [9]

Kontroler ini dirancang agar robot *humanoid* dapat berjalan secara stabil pada lantai yang miring. Apabila tidak ada kontroler maka ketika lantai miring maka robot akan miring sehingga jatuh. Kontroler ini digunakan untuk mencegah hal tersebut dengan menjaga robot tetap tegak meskipun lantainya miring

Untuk dapat menerapkan kontroler ini maka diperlukan sensor tambahan untuk mengukur kemiringan *body* robot. Sensor tersebut dapat berupa sensor *gyroscope 2 axis* yang akan digunakan untuk mengukur kemiringan pada sumbu x dan sumbu y robot. Salah satu contohnya adalah untuk mengatur kemiringan sudut *pitch* robot, kontroler ini memiliki struktur yang mirip dengan kontroler PI dengan tambahan nilai awal yang telah diberikan sebelumnya. Salah satu contohnya adalah ketika digunakan untuk mengatur keseimbangan terhadap kemiringan sumbu *roll*, kontroler ini menggabungkan kontroler PI dengan masukan *error pitch* $\theta_{err,p}^{torso}$ dengan sudut engkel $u_{ankle\ pitch}$ yang telah ditentukan sebelumnya sehingga diperoleh besar sudut yang harus diberikan ke engkel. Hal tersebut dapat dituliskan pada Persamaan (2.13).

$$u'_{ankle\ pitch} = u_{ankle\ pitch} + (K_p + \frac{K_i}{s}) \theta_{err,p}^{torso} \quad (2.13)$$

K_p dan K_i adalah Konstanta Proporsional dan Konstanta Integral. $u'_{ankle\ pitch}$ adalah kompensasi sudut engkel. Untuk keseimbangan pada sudut *roll* bentuknya sama dengan sudut *pitch*. Masukan dari kontroler PI

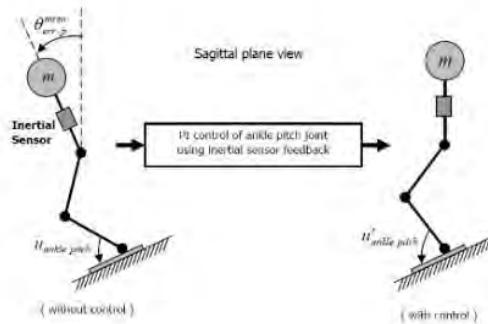
adalah *error roll* tubuh bagian $\theta_{err,r}^{torso}$ atas pada jarak antara *joint* pinggang dan *joint* engkel yang telah ditentukan sebelumnya yaitu l_R atau l_L . Untuk berjalan, maka kontroler ini diterapkan pada setiap fase gerakan. Kontroler *upright pose* pada sumbu *roll* ditunjukkan oleh Persamaan (2.14) sampai dengan (2.16).

$$l'_L = l_L + (K_p + \frac{K_I}{s})\theta_{err,r}^{torso} \quad (2.14)$$

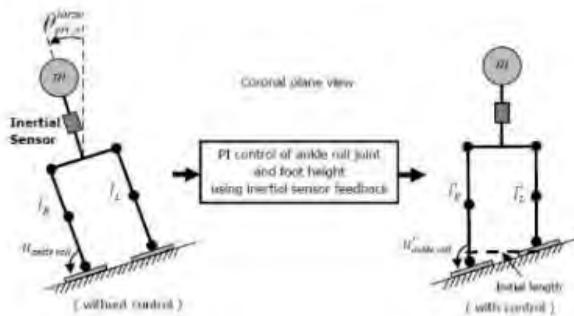
$$l'_R = l_R - (K_p + \frac{K_I}{s})\theta_{err,r}^{torso} \quad (2.15)$$

$$u'_{ankle\ roll} = u_{ankle\ roll} + \tan^{-1}\left(\frac{l'_R - l'_L}{\text{jarak antara joint engkel}}\right) \quad (2.16)$$

Ilustrasi dari kontroler *upright pose* pada robot *humanoid* ditunjukkan oleh Gambar 2.17 dan Gambar 2.18.



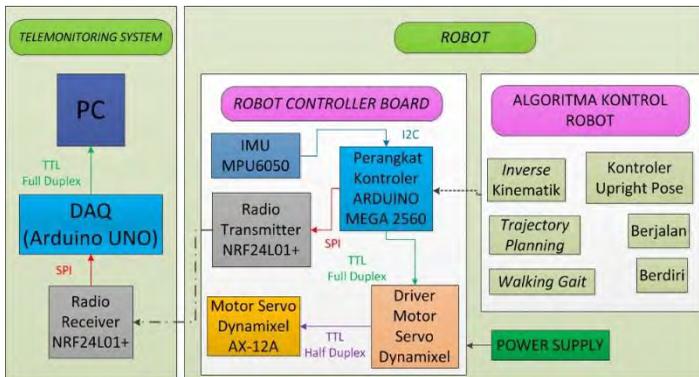
Gambar 2.16 Kontroler *Upright Pose* pada Kemiringan Sudut *Pitch*



Gambar 2.17 Kontroler *Upright Pose* pada Kemiringan Sudut *Roll*

BAB 3 PERANCANGAN SISTEM

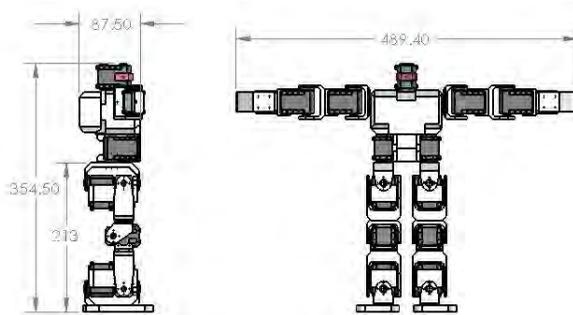
Bab ini menjelaskan tentang rancangan sistem yang digunakan pada tugas akhir ini yang terdiri dari bagian mekanik robot beserta parameter DH dan model kinematika nya termasuk model keseimbangan robot tersebut, bagian elektronik meliputi sensor, perangkat kontroler, perangkat *driver* motor, dan bagian *software* yang berisi algoritma kontrol serta program lain yang berkaitan. Masing – masing bagian tersebut dijelaskan pada tiap – tiap subbab yang terdapat pada bab ini. Rancangan keseluruhan sistem pada tugas akhir ini ditunjukkan pada Gambar 3.1.



Gambar 3.1 Rancangan Sistem Pengaturan Keseimbangan Robot *Humanoid*

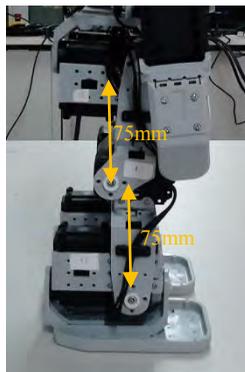
3.1 Mekanik Robot Bioloid

Robot *humanoid* yang digunakan adalah kit robot Bioloid Premium yang dirakit menjadi bentuk robot *humanoid*. Karena berupa robot kit pabrikan maka tingkat kepresisian masing – masing bagiannya tinggi. Robot ini terdiri dari 18 motor servo , masing – masing penggunaannya adalah 12 servo untuk kedua kaki sehingga masing – masing kaki tersusun atas 6 servo, dan 6 servo untuk kedua tangan sehingga 3 servo untuk masing – masing tangan. Untuk bagian *link* penghubung Ukuran bentangan robot Bioloid ditunjukkan oleh Gambar 3.2.



Gambar 3.2 Ukuran Bentangan Robot Bioloid

Pada tugas akhir ini, tubuh bagian atas robot dianggap sebagai benda tegar, dan yang menjadi pusat perhatian adalah kaki robot. Ukuran kaki robot Bioloid ditunjukkan oleh Gambar 3.3.



Gambar 3.3 Ukuran Kaki Robot Bioloid

3.1.1 Parameter DH Robot Bioloid

Sesuai dengan ukuran real dari masing – masing bagian mekanik robot Bioloid terutama kaki robot, maka selanjutnya dapat disusun tabel parameter DH kaki robot Bioloid seperti pada Tabel 3.1 dan Tabel 3.2.

Tabel 3.1 Parameter DH Kaki Kiri Robot Bioloid

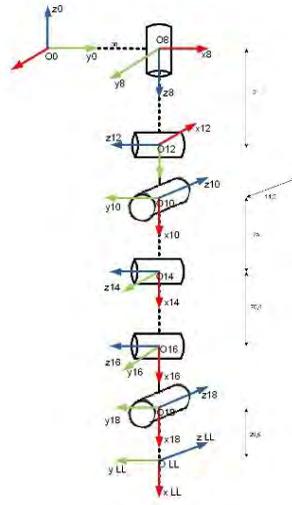
<i>Link ke – i</i>	$a_i(mm)$	$d_i(mm)$	$\theta_i(^{\circ})$	$\alpha_i(^{\circ})$
1	38	0	0	180
2	0	31	-58~30	-90
3	0	0	-36~0	-90
4	75	14,5	-124~21	90
5	75	0	-38~0	0
6	0	0	-16~26	-90
7	29,5	0	-10~21	0

Tabel 3.2 Parameter DH Kaki Kanan Robot Bioloid

<i>Link ke – i</i>	$a_i(mm)$	$d_i(mm)$	$\theta_i(^{\circ})$	$\alpha_i(^{\circ})$
1	38	0	0	180
2	0	31	-30~58	-90
3	0	0	0~36	-90
4	75	14,5	-21~124	90
5	75	0	0~38	0
6	0	0	-26~16	-90
7	29,5	0	-21~10	0

3.1.2 Kerangka Koordinat Robot Bioloid

Berdasarkan nilai dari parameter DH yang telah ditunjukkan oleh Tabel 3.1 dan Tabel 3.2, maka dapat digambarkan kerangka koordinat dari robot yang digunakan dengan mengikuti kaidah yang telah ditentukan oleh *Denavit – Hartenberg*. Kerangka yang digambar disesuaikan dengan kebutuhan yang akan dianalisa. Karena pada tugas akhir ini yang dianggap sebagai robot manipulator adalah kaki maka yang akan dibuat adalah kerangka kaki robot. Selain itu kaki kiri dan kanan memiliki struktur yang sama sehingga untuk kerangka koordinatnya dapat diwakili oleh satu buah kaki saja, dan dalam hal ini hanya diwakili oleh kaki kiri. Kerangka koordinat kaki kiri dari robot Bioloid ditunjukkan oleh Gambar 3.4

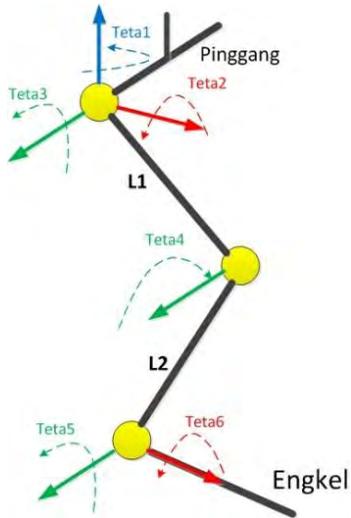


Gambar 3.4 Kerangka Koordinat Kaki Robot Bioloid

3.1.3 *Inverse Kinematik Kaki Robot Bioloid*

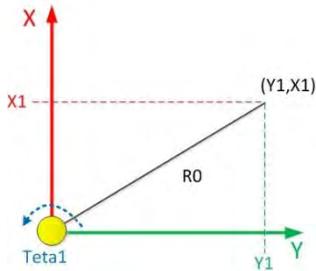
Untuk melakukan gerakan berjalan maka kaki robot harus bergerak pada posisi tertentu. *Inverse* kinematik diterapkan pada robot ini untuk memudahkan menggerakkan kaki untuk mencapai posisi tujuan dengan memberikan masukan posisi yang dituju dan langsung diperoleh nilai dari parameter *joint* kaki robot.

Perhitungan *inverse* kinematik posisi kaki robot ini dilakukan dengan pendekatan geometri dengan menerapkan rumus segitiga (trigonometri) pada kaki robot. Untuk memudahkan analisisnya maka bagian kaki robot dibagi menjadi beberapa segmen dan sudut pandang yakni dilihat dari depan, samping dan atas. Kaki robot apabila dilihat secara isometri ditunjukkan oleh Gambar 3.5.



Gambar 3.5 Model Kinematik Kaki Tampak Isometri

Kerangka kaki tampak atas adalah ditunjukkan oleh Gambar 3.6

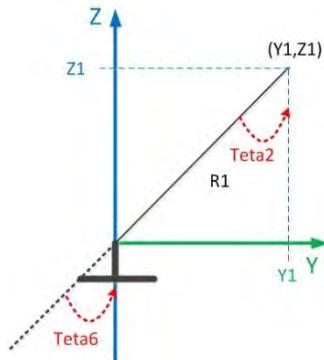


Gambar 3.6 Model Kinematik Kaki Tampak Atas

Berdasarkan Gambar 3.6 dapat diperoleh Persamaan posisi ujung kaki ketika dilihat dari atas yang dituliskan pada Persamaan (3.1).

$$R_0 = \sqrt{x_1^2 + y_1^2} \quad (3.1)$$

Selanjutnya sudut pandang diubah menjadi dari depan dan diperoleh model kaki tampak depan yang ditunjukkan oleh Gambar 3.7



Gambar 3.7 Model Kinematik Kaki Tampak Depan

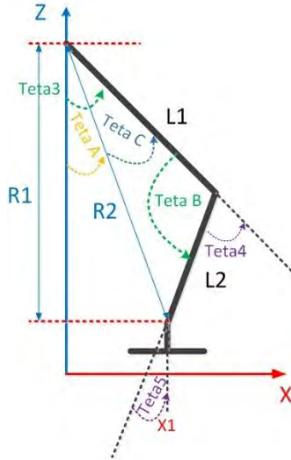
Berdasarkan Gambar 3.7 dapat diperoleh Persamaan posisi ujung kaki pada bidang YZ yang ditunjukkan oleh Persamaan (3.2) , (3.3), dan (3.4)

$$R_1 = \sqrt{y_1^2 + z_1^2} \quad (3.2)$$

$$\theta_2 = \tan^{-1}\left(\frac{y_1}{z_1}\right) \quad (3.3)$$

$$\theta_6 = -\theta_2 \quad (3.4)$$

Untuk melengkapi model kinematik kaki robot sudut pandang diubah menjadi dari samping dan diperoleh model kaki tampak samping yang ditunjukkan oleh Gambar 3.8



Gambar 3.8 Model Kinematik Kaki Tampak Samping

Berdasarkan Gambar 3.8 dapat diperoleh Persamaan posisi ujung kaki pada bidang XZ yang ditunjukkan oleh Persamaan (3.5) sampai dengan (3.14)

$$R_2 = \sqrt{R_1^2 + x_1^2} \quad (3.5)$$

$$\theta_A = \tan^{-1}\left(\frac{x_1}{R_1}\right) \quad (3.6)$$

$$\cos C = \frac{L_2^2 - (L_1^2 + R_2^2)}{2L_1R_2} \quad (3.7)$$

$$\sin C = \sqrt{1 - \cos^2 C} \quad (3.8)$$

$$\theta_C = \tan^{-1}\left(\frac{\sin C}{\cos C}\right) \quad (3.9)$$

$$\sin B = \sqrt{1 - \cos^2 B} \quad (3.10)$$

$$\theta_B = \tan^{-1}\left(\frac{\sin B}{\cos B}\right) \quad (3.11)$$

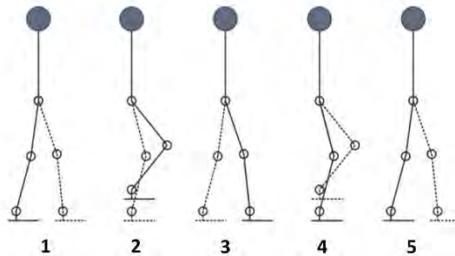
$$\theta_3 = \theta_A + \theta_C \quad (3.12)$$

$$\theta_4 = 180 - \theta_B \quad (3.13)$$

$$\theta_5 = 180 - \theta_B \quad (3.14)$$

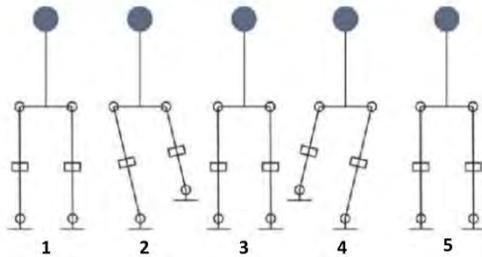
3.1.4 Perancangan *Walking Gait* Robot

Walking gait dari robot menjadi penentu berhasil tidaknya aksi menjaga keseimbangan dari robot ini, sehingga perlu dirancang dengan tepat. Untuk dapat berjalan maka kaki robot harus bergerak menuju titik – titik tertentu sehingga terbentuk suatu jalur gerakan yang dilalui oleh kaki robot yang disebut dengan *trajectory*. Pada robot ini, penentuan *trajectory* dari titik ke titik dilakukan dengan langsung memberikan titik koordinat yang akan dilalui. Titik – titik yang dilalui tersebut mewakili tiap fase gerakan berjalan robot mulai dari ketika kedua kaki berada di lantai atau fase DSP, kemudian salah satu kaki diangkat dan robot hanya ditopang oleh satu kaki atau fase SSP, hingga kedua kaki kembali berada di lantai atau fase DSP. Fase gerakan pada *walking gait* robot tampak samping ditunjukkan oleh Gambar 3.9



Gambar 3.9 *Walking Gait* Robot Tampak Samping

Kemudian *walking gait* robot jika dilihat dari depan ditunjukkan oleh Gambar 3.10



Gambar 3.10 *Walking Gait* Robot Tampak Depan

Berdasarkan Gambar 3.9 dan 3.10 *walking gait* robot *humanoid* dibagi menjadi 5 segmen gerakan yaitu:

1. Kedua kaki menapak lantai dan posisi kaki kanan berada di depan kaki kiri
2. Pinggang digeser ke kanan, kaki kanan menjadi tumpuan dan kaki kiri diangkat pada posisi puncak ketinggian
3. Pinggang digeser ke posisi semula, kedua kaki menapak lantai dengan posisi kaki kiri di depan kaki kanan
4. Pinggang digeser ke kiri, kaki kiri menjadi tumpuan dan kaki kanan diangkat pada posisi puncak ketinggian
5. Pinggang digeser ke posisi semula, kedua kaki menapak lantai dengan posisi kaki kanan di depan kaki kiri

Kelima segmen gerakan tersebut diulang terus menerus membentuk suatu siklus gerakan berjalan.

3.2 Perangkat Elektronik Robot Bioloid

Bagian penyusun robot selain mekanik adalah perangkat elektronik. Beberapa perangkat elektronik yang dibutuhkan agar sistem dapat bekerja adalah *driver* motor servo Dynamixel, *Inertial Measurement Unit*, perangkat komunikasi *wireless*, perangkat mikrokontroler dan baterai li-po sebagai sumber daya listrik dari semua perangkat elektronik robot ini.

Semua perangkat tersebut kemudian dirangkum dalam sebuah *main board controller*.

3.2.1 Perancangan *Driver Motor Servo Dynamixel*

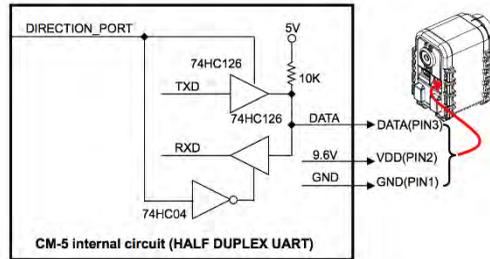
Motor servo Dynamixel adalah motor servo buatan Robotis. Motor dengan tipe AX-12 memiliki spesifikasi torsi sebesar 12 kgfcm dengan tegangan *supply* 7V dan 16,5 kgfcm dengan *supply* 10V. Kemudian terdapat sistem keamanan yang berupa alarm serta aksinya apabila terjadi sesuatu yang tidak sesuai dengan standar operasi motor ini diantaranya ketika temperatur motor, torsi beban atau tegangan yang melebihi batas, secara otomatis motor akan memberikan sinyal alarm melalui kedipan led, kemudian motor akan mati. Bentuk motor Dynamixel ditunjukkan oleh Gambar 3.11



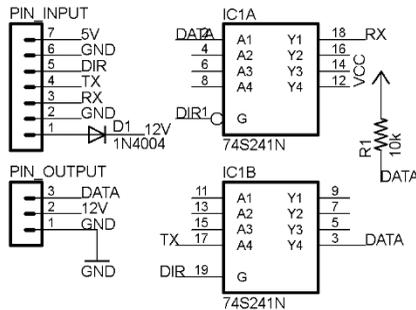
Gambar 3.11 Motor Servo Dynamixel AX-12A [10]

Dynamixel memiliki beberapa tipe dengan karakter masing – masing yang paling menonjol yakni karakter torsi dan kecepatan sehingga pengguna dapat menyesuaikan dengan kebutuhan. Beberapa tipe yang dimaksud diantaranya adalah tipe AX, MX, RX, dan EX. Motor servo Dynamixel yang digunakan pada robot ini memiliki tipe AX-12A.

Untuk bisa menggerakkan motor tersebut melalui perintah program dari mikrokontroler maka diperlukan rangkaian *driver* yang berupa rangkaian *serial* konverter *full duplex* ke *half duplex*. Hal ini karena jalur data *serial* yang terdapat pada motor hanya satu yang berfungsi sebagai jalur pengiriman dan penerimaan data sedangkan pada mikrokontroler terdapat dua jalur terpisah untuk pengiriman data dan penerimaan data. Rangkaian *driver* motor servo Dynamixel AX-12A dibuat dengan menggunakan IC 74LS241 dengan tambahan resistor *pull-up* pada jalur keluaran data. Rangkaian *driver* motor servo tersebut ditunjukkan oleh Gambar 3.12. Gambar *schematic* rangkaian *driver* motor servo Dynamixel AX-12A seperti pada Gambar 3.13.



Gambar 3.12 Rangkaian *Driver* Motor Dynamixel [11]



Gambar 3.13 *Schematic* *Driver* Motor Dynamixel

3.2.2 Perancangan Perangkat Kontroler Robot

Sebenarnya pada kit Robot Bioloid telah disertakan perangkat kontroler robot yaitu CM-530 dan beberapa modul sensor serta *software* untuk memrogram gerakan robot, namun perangkat ini memiliki keterbatasan dalam pemrograman gerakannya terutama pada bagian *software* nya. Pemrograman gerakan dilakukan dengan memberikan langsung data sudut servo sesuai dengan ID servo masing – masing dengan menggunakan aplikasi *RoboPlus Motion*. Untuk memanggil aksi – aksi dari gerakan yang telah disimpan digunakan aplikasi *RoboPlus Task*.

Kekurangan dari aplikasi ini yang paling menonjol adalah tidak dapat melakukan perhitungan matematis diluar aritmatika ($x : + -$), sedangkan untuk perhitungan *inverse* kinematik robot memerlukan perangkat kontroler yang dapat menghitung fungsi trigonometri. Akan

sangat merepotkan apabila harus memasukkan nilai sudut servo satu per satu untuk setiap sekuen gerakan yang diinginkan, sedangkan metode *inverse* kinematik dapat melakukan hal ini, untuk itulah perlu dirancang perangkat kontroler yang baru agar metode – metode yang telah dirancang sebelumnya dapat diterapkan langsung pada robot ini.



Gambar 3.14 *Board Controller Robot*

Board controller robot ini dibuat untuk mengintegrasikan semua komponen elektronik dalam sebuah modul yang kompak sehingga sistem pengaturan keseimbangan robot yang dibangun dapat berjalan dan menggantikan kontroler sebelumnya yaitu CM-530.

3.2.3 *Inertial Measurement Unit (IMU) [12]*

Inertial Measurement Unit adalah perangkat elektronik yang mampu memberikan informasi mengenai perubahan sudut dan percepatan pada tempat dimana perangkat ini dipasang. Pada robot ini perangkat ini digunakan untuk mengukur kemiringan robot terhadap lantai melalui sensor *gyroscope* yang tertanam di dalamnya. IMU yang digunakan adalah MPU6050 buatan Invensense Electronics yang memiliki spesifikasi yang ditunjukkan oleh Tabel 3.3.



Gambar 3.15 *Sensor IMU MPU6050*

Tabel 3.3 Spesifikasi Sensor IMU MPU 6050

<i>Full Scale Range ($^{\circ}/\text{sec}$)</i>	± 2000
<i>Gyro Sensitivity (LSB/$^{\circ}/\text{sec}$)</i>	16,4
<i>Gyro Rate Noise (dps/$\sqrt{\text{Hz}}$)</i>	16,4
<i>Accel Full Scale Range (g)</i>	± 16
<i>Accel Sensitivity (LSB/g)</i>	2048
<i>Digital Output</i>	I ² C
<i>Logic Supply Voltage (V)</i>	1,8 \pm 5% or VDD
<i>Operating Voltage Supply (V\pm5%)</i>	2,375V – 3,46V
<i>Package Size (mm)</i>	4x4x0,9

Pada sistem ini perangkat IMU berfungsi sebagai sensor kemiringan badan robot terhadap lantai dengan mengambil data kemiringan dari sensor *gyroscope* yang terdapat didalamnya. Informasi inilah yang akan diolah oleh kontroler untuk mengatur posisi tubuh bagian atas robot sehingga keseimbangan dapat tercapai.

3.2.4 Perangkat Komunikasi *Wireless* [13]

Pengambilan data dari robot untuk keperluan monitoring dilakukan melalui komunikasi *wireless*. Hal ini dipilih agar tidak ada kabel yang dapat mengganggu gerakan jalan robot. Perangkat komunikasi yang digunakan adalah modul komunikasi *wireless* NRF24L01+ yang merupakan modul komunikasi *wireless* radio 2,4 GHz . Perangkat fisik modul komunikasi *wireless* ditunjukkan oleh Gambar 3.16, dan spesifikasi modul komunikasi *wireless* NRF24L01+ ditunjukkan pada Tabel 3.4.



Gambar 3.16 Modul Komunikasi *Wireless* Radio NRF24L01+

Tabel 3.4 Spesifikasi NRF24L01+

<i>Supply Voltage (V)</i>	1,9 – 3,6
<i>Modulation</i>	GFSK
<i>Operating Frequency (MHz)</i>	2400 - 2525
<i>Crystal Frequency (MHz)</i>	16
<i>Frequency Deviation@ 1 Mbps (KHz)</i>	±160
<i>Average current during TX setting (mA)</i>	8,0
<i>Average current during RX setting (mA)</i>	8,9
<i>Operating Temperature (°C)</i>	+27

Pada sistem ini modul komunikasi *wireless* radio NRF24L01+ berfungsi sebagai media komunikasi *wireless* yang akan mengirimkan data kemiringan aktual robot, dan *set point* kemiringan pada komputer monitor untuk keperluan pengamatan dan analisa data.

3.2.5 Baterai Li-Po [14]

Sumber daya listrik yang digunakan pada perangkat elektronik robot ini adalah baterai Li-Po 3 sel yang setara dengan 11,1V 1000mAh. Baterai ini dipilih karena dapat menyimpan daya yang cukup besar dan memiliki daya tahan yang lama. Selain itu ukurannya yang cukup kecil dan ringan memungkinkan untuk dipasang pada *body* robot. Bentuk fisik baterai Li-Po 3 sel ditunjukkan oleh Gambar 3.17. Lebih detail mengenai spesifikasi baterai Li-Po yang digunakan pada robot ini dapat dilihat pada Tabel 3.5.



Gambar 3.17 Baterai Li-Po 3 Sel 11,1V 1000mAh

Tabel 3.5 Spesifikasi Baterai Li-Po

Kapasitas (mAh)	1000
Tegangan (V)	11,1
Konfigurasi (Cell)	3
<i>Discharge (C)</i>	20
Bobot (g)	87
<i>Peak Discharge (C)</i>	30
Jenis Konektor	JST-XH

3.2.6 Perangkat Mikrokontroler

Mikrokontroler merupakan sebuah sistem komputer yang dikemas dalam sebuah *chip*. Artinya dalam sebuah IC mikrokontroler sebetulnya sudah terdapat kebutuhan minimal agar mikroprosesor dapat bekerja, yaitu mikroprosesor, ROM, RAM, I/O, dan clock seperti halnya yang dimiliki sebuah komputer PC. Penggunaan mikrokontroler sangat luas di bidang elektronika, salah satunya adalah sebagai perangkat kontroler pada robot. Semua proses yang menyangkut pengolahan data pada robot dilakukan oleh perangkat ini.

Ada banyak sekali jenis dan keluarga mikrokontroler baik yang masih berupa *chip* atau yang sudah dalam bentuk *development board*. Salah satu *development board* yang terkenal adalah Arduino yang bersifat *open source*. Pada robot ini terdapat dua buah *board* Arduino yakni Arduino Mega 2560 yang digunakan sebagai kontroler pada robot dan Arduino UNO sebagai perangkat data akuisi pada komputer PC dan menangani komunikasi *wireless* antara komputer dengan robot.

3.2.6.1 Arduino Mega 2560 [15]

Arduino Mega 2560 adalah *board* mikrokontroler berbasis ATmega 2560. *Board* ini memiliki 54 pin digital I/O yang mana 14 pin diantaranya dapat memiliki fungsi PWM (*Pulse Width Modulation*), 16 pin *Analog Input*, 4 UART dan dilengkapi *oscillator crystal* 16 MHz. Bentuk fisik Arduino Mega 2560 ditunjukkan oleh Gambar 3.18 dan spesifikasi Arduino Mega 2560 dapat dilihat pada Tabel 3.6.



Gambar 3.18 Modul Arduino MEGA 2560

Tabel 3.6 Spesifikasi Arduino Mega2560

Mikrokontroler	Atmega2560
Catu daya	5V
Tegangan <i>input</i> (rekomendasi)	7-12V
Tegangan <i>input</i> (batasan)	6-20V
Pin I/O <i>digital</i>	54 pin dimana 15 diantaranya berupa PWM
Pin <i>input</i> Analog	16 pin
Arus DC per pin I/O	50 mA
<i>Flash memory</i>	256 KB (Atmega2560)
<i>Clock speed</i>	16 MHz

Pada sistem ini, Arduino Mega 2560 berfungsi sebagai kontroler utama robot yang mengatur semua gerakan robot, di dalamnya ditanamkan algoritma *inverse* kinematik, *walking gait*, pengolahan sensor kemiringan, pengaturan keseimbangan robot, dan komunikasi *wireless* dengan komputer.

3.2.6.2 Arduino UNO [16]

Arduino UNO adalah *board* mikrokontroler berbasis ATmega328P. *board* ini memiliki 14 pin digital I/O yang mana 6 pin diantaranya memiliki fungsi PWM, 6 *Analog Input*, dan dilengkapi dengan *oscillator crystal* 16 MHz. Bentuk fisik Arduino UNO ditunjukkan oleh Gambar 3.19 dan spesifikasi Arduino UNO dapat dilihat pada Tabel 3.7.



Gambar 3.19 Modul Arduino UNO

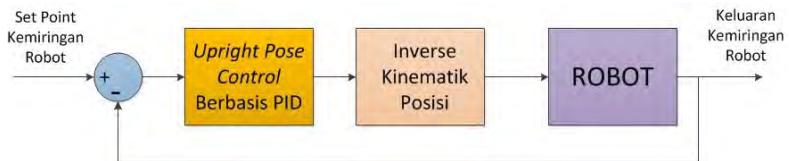
Tabel 3.7 Spesifikasi Arduino UNO

Mikrokontroler	Atmega328
Catu daya	5V
Tegangan <i>input</i> (rekomendasi)	7-12V
Tegangan <i>input</i> (batasan)	6-20V
Pin I/O <i>digital</i>	14 pin dimana 6 diantaranya berupa PWM
Pin <i>input</i> Analog	6 pin
Arus DC per pin I/O	40 mA
<i>Flash memory</i>	32 KB (Atmega328)
<i>Clock speed</i>	16 MHz

Pada sistem ini, Arduino UNO berfungsi sebagai perangkat data akuisisi yang menghubungkan antara komputer dengan robot untuk proses *telemonitoring*. Perangkat ini terhubung dengan modul komunikasi *wireless* pada sisi *receiver* dan didalamnya ditanamkan algoritma pengiriman data *wireless* serta komunikasi *serial* dengan komputer. Data tersebut kemudian ditampilkan pada *serial* monitor komputer sehingga kondisi kemiringan aktual robot dan data *set point* kemiringan robot dapat dipantau secara *real – time*. Untuk keperluan analisa data, maka data ini kemudian disimpan dan diolah sehingga bisa ditampilkan dalam bentuk grafik dengan menggunakan bantuan *software* seperti MATLAB atau Microsoft Excel sehingga dapat memudahkan analisa mengenai performa dari kontroler keseimbangan yang telah dirancang dan diterapkan pada robot.

3.3 Perancangan Kontroler Robot

Kontroler merupakan salah satu komponen penyusun sistem pengaturan. Sistem pengaturan keseimbangan robot *humanoid* dapat digambarkan pada diagram blok berikut ini:



Gambar 3.20 Diagram Blok Sistem Pengaturan Keseimbangan Robot *Humanoid*

Kontroler untuk keseimbangan robot ini dirancang setelah mengetahui perilaku keseimbangannya. Kontroler *upright pose* berbasis PID dipilih karena memiliki struktur yang simpel sehingga sesuai dengan kemampuan perangkat kontroler yang digunakan. Berdasarkan diagram blok tersebut, kontroler ini digunakan dalam dua kondisi yaitu ketika robot berdiri dan ketika robot berjalan.

3.3.1 Perancangan Kontroler *Upright Pose* untuk Berdiri

Pada saat berdiri robot akan mengambil titik acuan kemiringan dengan posisi tegak lurus terhadap lantai. Nilai ini akan digunakan sebagai nilai acuan kemiringan ketika robot berdiri pada bidang miring. Kemudian untuk aksinya kontrolnya digunakan algoritma kontrol PID yang parameter kontrolnya dicari dengan cara *manual tuning* dengan melihat perilaku dari robot sehingga diperoleh respon yang diinginkan.

3.3.2 Perancangan Kontroler *Upright Pose* untuk Berjalan

Kondisi robot ketika berjalan cenderung tidak seimbang bergantung pada desain *walking gait* nya terutama ketika salah satu menjadi penopang seluruh *body* robot dan kaki yang lain tidak menapak lantai atau fase *single support*. Perilaku yang berbeda ini membuat kontroler *upright pose* yang diterapkan pun berbeda dengan ketika berdiri. Ketika berjalan, kontroler ini diterapkan pada masing masing fase gerakan yang telah dirancang pada proses perancangan *walking gait*.

BAB 4 PENGUJIAN DAN ANALISA

Bab ini menjelaskan hasil pengujian dan analisa mengenai rancangan yang telah dibuat. Pengujian dan analisa yang dilakukan tersebut terkait dengan pengujian masing – masing komponen penyusun sistem sampai performa sistem secara keseluruhan.

4.1 Pengujian Mekanik Robot

Bagian mekanik robot ini merupakan kit bawaan dari pabrik sehingga seharusnya sudah lolos uji kualitas. Namun meskipun demikian perlu tetap dilakukan pengujian ulang, terutama pada bagian – bagian yang paling berpengaruh terhadap sistem keseimbangan yang telah dirancang.

4.1.1 Pengujian Motor Servo Dynamixel AX-12A

Pengujian motor servo Dynamixel dilakukan dengan memberikan tegangan *supply* dan memberikan data sudut ke motor. Kemudian sudut putaran *horn* servo tersebut diukur dengan menggunakan busur derajat. Data sudut terukur tersebut dibandingkan dengan data sudut masukan yang diberikan pada servo untuk mengukur tingkat keakuratan dan kepresisian motor servo Dynamixel AX-12A.

Dari data yang diperoleh kemudian dihitung nilai RMSE nya dengan cara yang ditunjukkan oleh Persamaan (4.1) untuk memperoleh nilai *error* rata – rata.

$$RMSE = \sqrt{\frac{\sum_{i=0}^k (y - y_1)^2}{k}} \quad (4.1)$$

Nilai ini digunakan sebagai nilai yang akan digunakan untuk menghitung nilai *Full Scale Error* (FSE). Nilai ini diperoleh dengan cara membagi nilai *error* pengukuran dengan nilai maksimal yang dapat diukur seperti ditunjukkan oleh Persamaan (4.2).

$$FSE = \frac{RMSE}{Fullscale} \times 100\% \quad (4.2)$$

Data hasil pengujian sudut motor servo Dynamixel AX-12A ditunjukkan oleh Tabel 4.1.

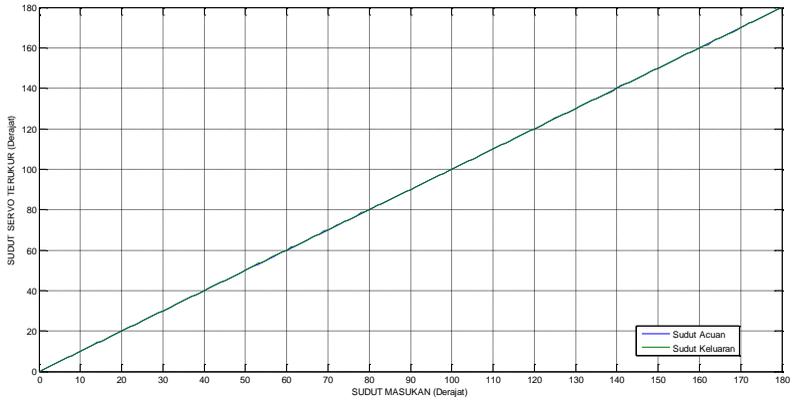
Tabel 4.1 Data Pengujian Motor Servo

Sudut Acuan ($^{\circ}$)	Sudut Terukur ($^{\circ}$)	Error ($^{\circ}$)
0	0	0
1	1	0
10	10	0
14	14,5	-0,5
17	17	0
25	25	0
30	30	0
45	45	0
55	55	0
57	57,5	-0,5
60	60	0
72	72,5	-0,5
90	90	0
135	135	0
150	150	0
155	155,5	-0,5
160	160	0
175	175	0
180	180	0

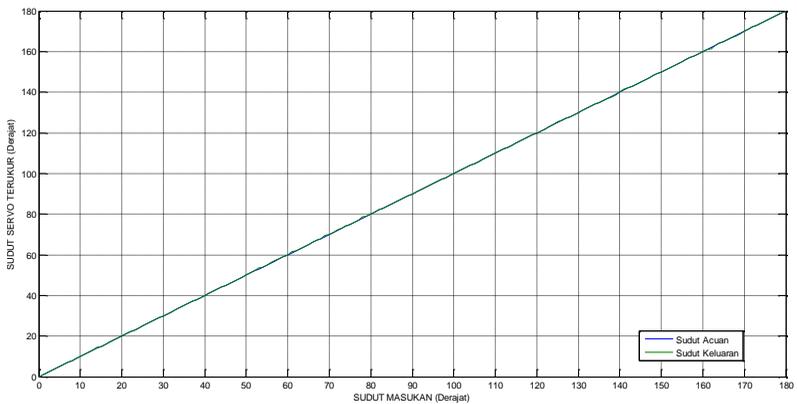
Berdasarkan data pada Tabel 4.1, nilai *error* diolah dengan menggunakan metode RMSE sehingga didapatkan tingkat kesalahan sebesar $0,229416^{\circ}$ dan FSE sebesar $0,076472\%$, sehingga memiliki tingkat keakuratan sebesar $99,92353\%$. Dari beberapa kali pengambilan data hasilnya tidak jauh berbeda sehingga diperoleh tingkat kepresisian yang tinggi. Dengan spesifikasi tersebut maka motor ini sesuai untuk digunakan sebagai aktuator dari robot ini.

Selain dilakukan uji akurasi dan kepresisian, dilakukan pula uji linieritas dan histerisis motor. Caranya dengan memberikan masukan sudut dari range 0° sampai dengan 180° dan dari 180° kembali lagi ke 0° dengan penambahan sebesar 1° .

Hasil uji linearitas dan histerisis motor ditunjukkan oleh Gambar 4.1 dan Gambar 4.2



Gambar 4.1 Uji Linieritas Motor Servo Dynamixel AX-12A



Gambar 4.2 Uji Histerisis Motor Servo Dynamixel AX-12A

Berdasarkan grafik tersebut diperoleh kesimpulan bahwa motor ini linier dan non histerisis sehingga sesuai untuk digunakan pada robot ini.

4.1.2 Pengujian Gerakan Kaki Robot

Pengujian gerakan robot merupakan kelanjutan dari pengujian motor servo Dynamixel. Motor yang berperan sebagai *joint* robot telah dirangkai dengan *frame* robot yang berperan sebagai *link*. Kemudian data sudut hasil perhitungan *inverse* kinematik dikirimkan ke motor sesuai dengan ID masing – masing.

4.1.2.1 Pengujian *Inverse Kinematik*

Pengujian *inverse* kinematik dilakukan dengan memberikan masukan posisi ujung kaki ke robot, kemudian setelah robot bergerak, dilakukan pengukuran posisi robot dari posisi awalnya (*Home*) dengan menggunakan penggaris. Data posisi hasil pengukuran dibandingkan dengan data posisi masukan *inverse* kinematik. Pengujian dilakukan pada tiga posisi yaitu posisi kaki terhadap sumbu X, posisi kaki terhadap sumbu Y, dan posisi kaki terhadap sumbu Z. Posisi kaki robot yang diinginkan diberikan pada program kemudian dilihat posisi aktual kaki robot setelah diberikan perintah posisi. Data hasil pengujian *inverse* kinematik posisi kaki robot ditunjukkan oleh Tabel 4.2 sampai dengan Tabel 4.4.

Tabel 4.2 Data Pengujian *Inverse* Kinematik Posisi pada Sumbu X

Posisi Masukan (mm)	Posisi Terukur (mm)	Error (mm)
0	0	0
10	8	2
20	17	3
50	46	4
70	67	3
80	76	4
100	96	4

Tabel 4.3 Data Pengujian *Inverse* Kinematik Posisi pada Sumbu Y

Posisi Masukan (mm)	Posisi Terukur (mm)	Error (mm)
0	0	0
10	7	3
20	16	4
50	47	3
70	68	2
80	76	4
100	97	3

Tabel 4.4 Data Pengujian *Inverse* Kinematik Posisi pada Sumbu Z

Posisi Masukan (mm)	Posisi Terukur (mm)	Error (mm)
150	150	0
140	145	-5
120	130	-10
110	126	-16
100	115	-15
90	110	-20
80	96	-16

Berdasarkan data hasil pengujian tersebut, terdapat *error* rata-rata posisi pada setiap gerakan sebesar 2,8 mm pada sumbu X, 2,7 mm pada sumbu Y, dan 11,71 mm pada sumbu Z hal ini terjadi karena terdapat offset pengukuran pada panjang *link* yang menjadi parameter pada perhitungan *inverse* kinematik sehingga nilai *error* tersebut akan terakumulasi menjadi *error* posisi total pada setiap gerakan. Kemudian efek pembulatan pada proses perhitungan juga berpengaruh pada posisi sudut servo yang akan digerakkan. Khusus pada Sumbu Z, distribusi beban pada motor servo tidak merata dan beban terberat terdapat pada servo pinggang dan engkel sehingga tingkat keakuratan motor juga ikut menurun akibat pembebanan. Hal inilah yang menyebabkan *error* posisi terbesar terdapat pada posisi sumbu Z.

4.2 Pengujian Perangkat Elektronik Robot

Pengujian perangkat elektronik dilakukan untuk mengetahui kemampuan kerja dari keseluruhan komponen yang terlibat di dalamnya. Pengujian ini melibatkan Pengujian sensor IMU.

4.2.1 Pengujian IMU

Pengujian perangkat IMU dibagi menjadi dua tahap pengujian yaitu pengujian *roll* dan *pitch*. Sebelum dilakukan pengujian, terlebih dahulu dilakukan inisialisasi awal yaitu dengan meletakkannya pada bidang datar kemudian mendiampkannya hingga proses inisialisasi selesai. Baru kemudian IMU tersebut diputar dan data sudut hasil bacaannya dibandingkan dengan data terukur dari busur derajat. Data hasil pengujian perangkat IMU ditunjukkan oleh Tabel 4.5 dan Tabel 4.6.

Tabel 4.5 Data Pengujian Perangkat IMU pada Sudut *Roll*

	Bacaan Sensor				
Pengambilan Data	1	2	3	4	5
Sudut Busur					
0	0,62	0,62	0,5	0,62	0,19
10	10,15	10,24	10,32	10,49	10,16
20	20,72	20,51	20,22	20,51	20,25
30	30,51	30,22	30,36	30,62	30,21
40	40,42	40,37	40,43	40,12	40,33
50	50,21	50,51	50,22	60,15	50,32
60	60,23	60,72	60,45	60,05	60,15
70	70,2	70,52	70,19	70,42	70,14
80	80,32	80,41	80,62	80,46	80,55
90	87,45	89,05	89,43	89,24	88,25

Berdasarkan data pada Tabel 4.5 , perangkat IMU MPU6050 memiliki rata – rata kesalahan bacaan sebesar 0,6626⁰ pada sudut *roll* dihitung dengan RMSE. Dan dari *trend data* yang ada sensor ini termasuk sensor yang linier.

Tabel 4.6 Data Pengujian Perangkat IMU pada Sudut *Pitch*

	Bacaan Sensor				
Pengambilan Data	1	2	3	4	5
Sudut Busur					
0	0,42	0,23	0,11	0,53	0,22
10	10,65	10,34	10,01	10,39	10,47
20	20,2	20,28	20,42	20,15	20,66
30	30,6	30,59	30,39	30,52	30,63
40	40,34	40,42	40,33	40,37	40,73
50	50,56	50,51	50,22	50,52	50,36
60	60,32	60,72	60,63	60,32	60,52
70	70,72	70,52	70,16	70,61	70,51
80	80,2	80,41	80,41	80,53	80,37
90	87,41	89,05	89,16	89,73	88,54

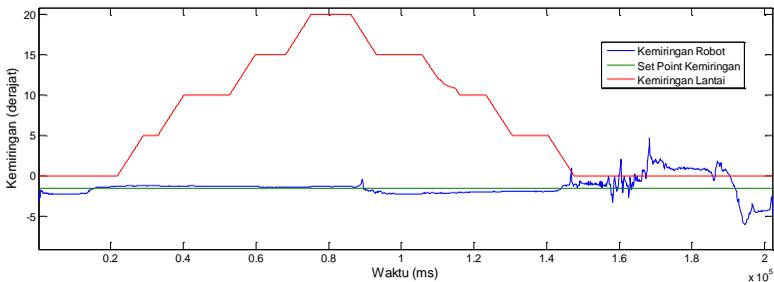
Berdasarkan data pada Tabel 4.3 , perangkat IMU MPU6050 memiliki rata – rata kesalahan bacaan sebesar 0,5046⁰ pada sudut *pitch* dihitung dengan RMSE. Dan dari *trend data* yang ada sensor ini termasuk

sensor yang linier. Sehingga dengan karakteristik tersebut sensor ini dapat digunakan pada robot.

4.3 Pengujian Kontroler

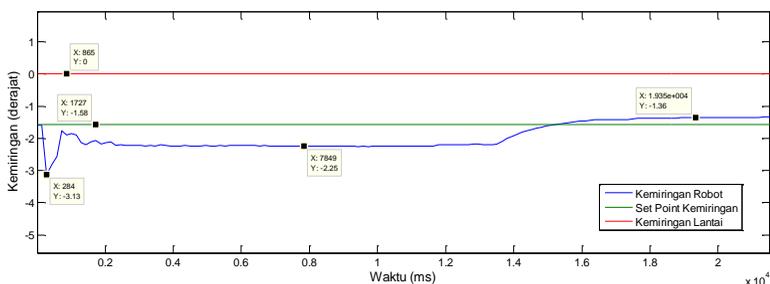
Pengujian kontroler *upright pose* hasil rancangan dilakukan Implementasi pada robot Bioloid. Kemudian diambil data pengukuran dengan *telemonitoring*. Data hasil pengukuran kemudian dianalisa. Hasil pengujian kontroler untuk kemiringan terhadap sudut *roll* robot ditunjukkan oleh Gambar 4.3.

4.3.1 Pengujian Kondisi Berdiri Terhadap Gerakan *Roll*



Gambar 4.3 Keseluruhan Respon Kemiringan Robot Ketika Gerakan *Roll*

Pengujian gerakan *roll* dilakukan dengan mengatur kemiringan lantai tempat pijakan robot dari kemiringan 0^0 naik sampai dengan 20^0 kemudian dari 20^0 turun hingga 0^0 . Pada setiap *interval* 5^0 kemiringan lantai ditahan selama beberapa detik untuk melihat respon robot secara bertahap. Keseluruhan respon kemiringan robot pada setiap *interval* kemiringan saat gerakan *roll* ditunjukkan oleh Gambar 4.3. Berdasarkan Gambar 4.3 terlihat respon robot beragam pada tiap – tiap *interval* namun sekilas terlihat mendekati nilai *set point* yang telah ditentukan, namun terjadi banyak osilasi pada pada *interval* kemiringan 5^0 turun menuju 0^0 hal ini disebabkan karena pendaratan lantai yang tidak mulus dan adanya gangguan gerakan pada saat penekanan tombol *stop* pada robot untuk mengakhiri proses perekaman data. Respon kemiringan terhadap robot pada kemiringan 0^0 ditunjukkan oleh Gambar 4.4

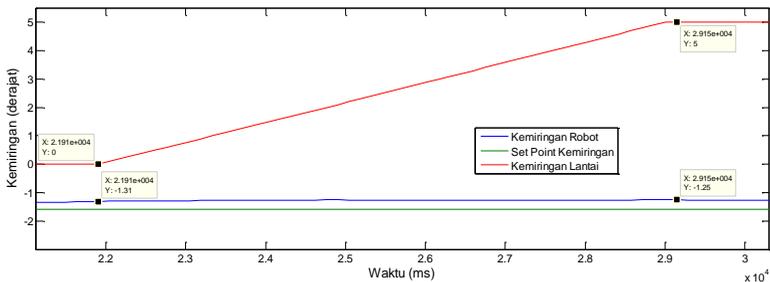


Gambar 4.4 Respon Kemiringan Robot Terhadap Gerakan *Roll* Ketika Kemiringan 0^0

Berdasarkan Gambar 4.4, setelah robot melakukan proses inialisasi, kontroler keseimbangan mulai bekerja pada robot. Hal yang dilakukan pertama adalah mengambil nilai *set point* kemiringan pada sudut *roll* dan didapatkan nilai *set point* sebesar $-1,58^0$. Nilai parameter PID yang digunakan pada sistem ini adalah $K_p = 0,5$; $K_i = 0,05$; dan $K_d = 0,01$. Pada awal setelah pengambilan *set point*, kontroler mulai bekerja. Kemiringan tubuh robot menyimpang dari *set point* dengan simpangan sebesar $1,58^0$. Hal ini terjadi karena efek proporsional dan integral yang digunakan pada kontroler. Konstanta proporsional dan integral mempercepat respon dengan memperbesar sinyal kontrol. Kondisi awal variabel *error* dan jumlah *error* perhitungan bernilai nol, setelah diberi diambil *set point* maka nilai *error* menjadi $1,58$ dan nilai jumlah *error* juga bernilai $1,58$. Kedua nilai ini masing – masing dikalikan dengan konstanta proporsional dan konstanta integral sehingga diperoleh sinyal kontrol sebesar $0,86$ konstanta differensial bernilai $0,001$ dan apabila dikalikan dengan nilai *delta error* awal yang juga sebesar $1,58$ akan menghasilkan nilai $0,00158$, nilai ini pada mikrokontroler akan dibulatkan menjadi 0 karena keterbatasan memori mikro dan kemampuan perhitungannya, sehingga efek differensial tidak begitu berpengaruh pada awal – awal aksi kontrol, sedangkan aksi proporsional dan integral sangat dominan. Nilai jumlah *error* untuk kontrol integral semakin lama semakin bertambah karena dihitung dari *error* sekarang ditambah dengan *error* sebelumnya sehingga pada satu waktu nilainya menjadi lebih besar dari nilai *error* itu sendiri meskipun sudah dikalikan dengan konstanta integral. Hal inilah yang mengakibatkan sinyal kontrol menjadi besar di awal – awal proses pengaturan keseimbangan dijalankan sehingga

mengakibatkan munculnya *overshoot*. Ketika *overshoot* muncul, maka nilai perubahan *error* menjadi besar. Hal ini memperbesar aksi kontrol differensial yang bersifat meredam adanya *overshoot* sehingga respon kemiringan robot kembali menuju *set point*. Karena besarnya konstanta differensial lebih kecil daripada konstanta integral atau pun konstanta proporsional maka efek redaman yang dihasilkan juga kecil. Hal ini mengakibatkan waktu yang dibutuhkan untuk mencapai kondisi *steady state* menjadi lebih lama. Pada sistem ini, robot dapat mencapai kondisi *steady state* pada kondisi kemiringan lantai 0° setelah 19,35 detik.

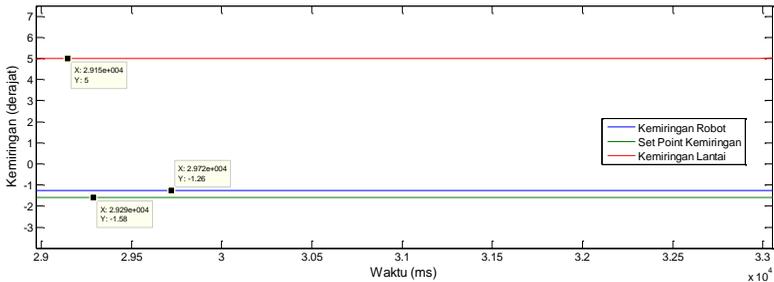
Setelah robot mencapai kondisi *steady state* pada kemiringan lantai 0° , lantai pelan – pelan dimiringkan hingga mencapai kemiringan 5° . Respon kemiringan robot ketika lantai dimiringkan dari 0° hingga 5° ditunjukkan oleh Gambar 4.5.



Gambar 4.5 Respon Kemiringan Robot Terhadap Gerakan *Roll* Ketika Terjadi Perubahan Kemiringan 0° - 5°

Perubahan kemiringan dilakukan dengan pelan – pelan untuk mencegah adanya guncangan yang diakibatkan oleh gerakan tangan atau pun guncangan yang muncul karena gerakan robot itu sendiri. Berdasarkan Gambar 4.5, robot bisa menjaga kemiringannya di sekitar *set point* meskipun perubahan kemiringannya cukup drastis. Hal ini menunjukkan kontroler *upright pose* berbasis PID yang diterapkan pada robot mulai bekerja dengan baik, namun masih terdapat *error steady state* sebesar $0,25^{\circ}$. Nilai ini berada di bawah resolusi dari motor servo yang digunakan yakni sebesar $0,3^{\circ}$ sehingga meskipun dapat diolah dalam proses perhitungan, nilai aksi kontrolnya tidak akan pernah bisa dilakukan karena keterbatasan kemampuan aktuator. Selain itu secara kasat mata robot akan terlihat berdiri tegak seimbang. Kemiringan kemudian dijaga

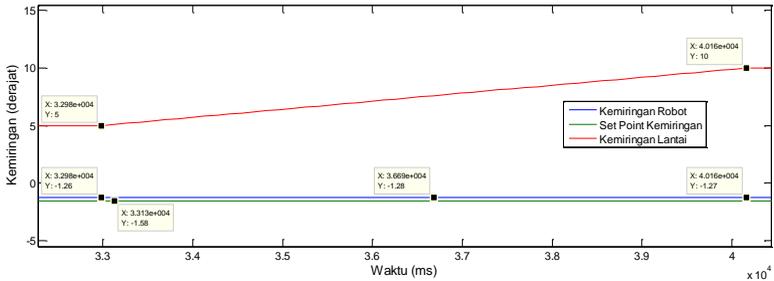
pada 5^0 selama 40 detik, dan diamati respon kemiringan robot terhadap kemiringan ini yang ditunjukkan oleh Gambar 4.6.



Gambar 4.6 Respon Kemiringan Robot Terhadap Gerakan *Roll* Ketika Kemiringan 5^0

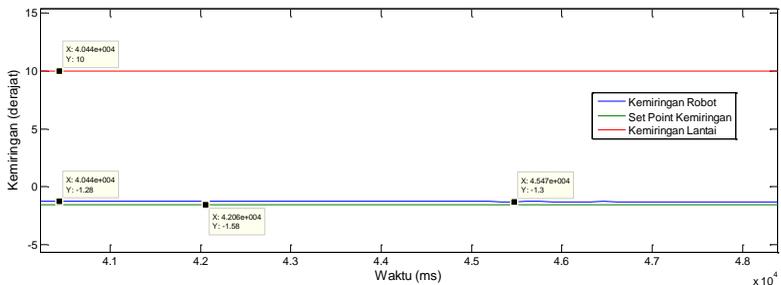
Pada kemiringan 5^0 robot dapat menjaga kemiringan tubuhnya di sekitar *set point* seperti halnya pada saat terjadi perubahan kemiringan dari $0^0 - 5^0$ sebelumnya dengan nilai *error steady state* sebesar $0,32^0$, terdapat perbedaan sebesar $0,07^0$ dari kondisi sebelumnya. Nilai *error steady state* ini sangat dekat dengan resolusi dari motor servo yang digunakan ketika tanpa beban, sehingga ketika motor tersebut sudah dirangkai menjadi rangkaian manipulator, tiap motor akan menanggung beban tertentu sesuai dengan posisinya pada rangkaian manipulator. Bertambahnya beban berakibat pada menurunnya resolusi motor servo yang digunakan sehingga resolusi motor servo yang sebelumnya $0,3^0$ berubah menjadi lebih besar. Hal inilah yang mengakibatkan *error steady state* sebesar $0,32^0$ tidak dapat diatasi oleh robot ini. Selain itu proses pembulatan pada perhitungan kontroler dan perhitungan *inverse kinematik* yang dilakukan oleh mikrokontroler juga berpengaruh pada semakin berkurangnya nilai sinyal kontrol yang akan diberikan pada motor servo yang berperan sebagai aktuator. Apabila nilai akhir hasil konversi data ke besaran data digital yang merepresentasikan sudut yang akan digerakkan oleh motor servo bernilai kurang dari 1 yang mewakili $0,3^0$, maka nilai tersebut tidak akan dieksekusi oleh motor servo atau dengan kata lain nilai tersebut diabaikan oleh sistem ini karena terlalu kecil dan berada di bawah resolusi dari aktuator yang digunakan.

Kemiringan robot kemudian ditambah secara pelan – pelan dari 5^0 menjadi 10^0 dan hasil respon kemiringan robot pada perubahan kemiringan ini ditunjukkan oleh Gambar 4.7



Gambar 4.7 Respon Kemiringan Robot Terhadap Gerakan *Roll* Ketika Terjadi Perubahan Kemiringan $5^0 - 10^0$

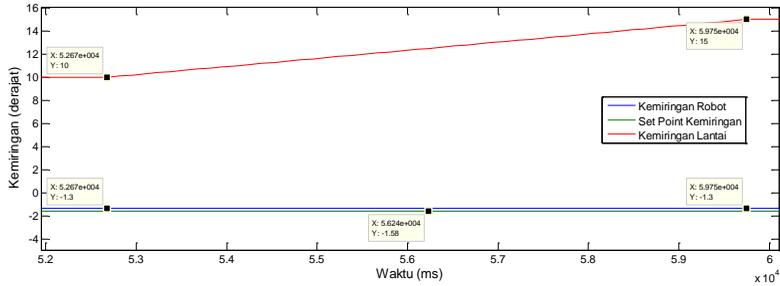
Berdasarkan Gambar 4.7, robot dapat menjaga kemiringannya berada di sekitar *set point* dengan nilai *error steady state* sebesar $0,31^0$. Nilai ini tidak akan bisa diatasi oleh aktuator karena berada di bawah resolusi dari motor servo yang digunakan. Hal ini sama dengan kondisi sebelumnya. Setelah dinaikkan, kemiringan tersebut dijaga selama 40 detik. Respon kemiringan robot ditunjukkan oleh Gambar 4.8



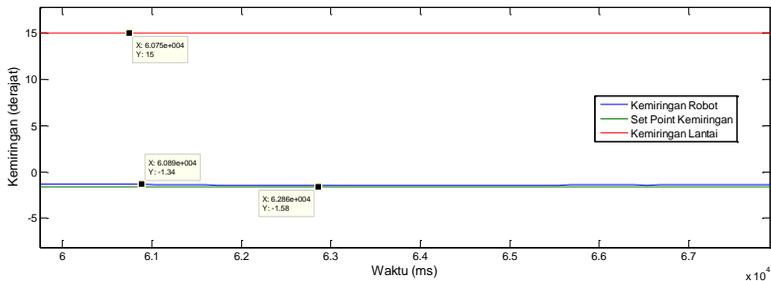
Gambar 4.8 Respon Kemiringan Robot Terhadap Gerakan *Roll* Ketika Kemiringan 10^0

Gambar 4.8 menunjukkan respon kemiringan robot terhadap gerakan *roll* saat kemiringan 10^0 yang mana tidak berbeda jauh dengan

kondisi sebelumnya dengan nilai *error steady state* yang berada di sekita $0,31^0$. Hasil yang sama ditunjukkan pada saat kemiringan lantai dinaikkan hingga 20^0 dan kemiringan tersebut dijaga selama 70 detik. seperti yang ditunjukkan oleh Gambar 4.9 sampai dengan Gambar 4.12.

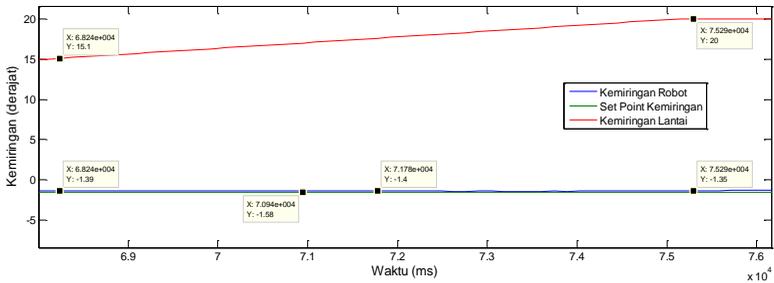


Gambar 4.9 Respon Kemiringan Robot Terhadap Gerakan *Roll* Ketika Terjadi Perubahan Kemiringan $10^0 - 15^0$

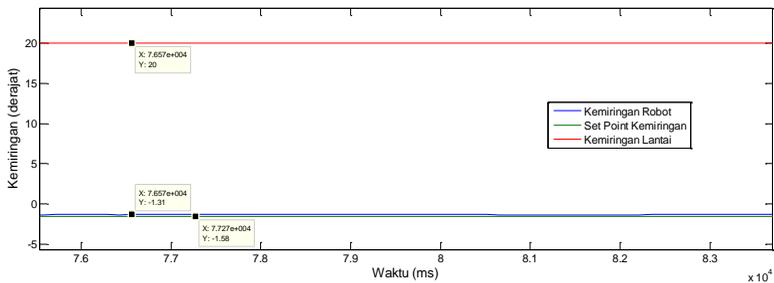


Gambar 4.10 Respon Kemiringan Robot Terhadap Gerakan *Roll* Ketika Kemiringan 15^0

Pada Gambar 4.9 dan Gambar 4.10 terlihat bahwa respon kemiringan robot tidak jauh berbeda meskipun terjadi perubahan kemiringan pada lantai. Perubahan kemiringan dilakukan dengan pelan – pelan karena kontroler ini tidak dirancang untuk menahan guncangan yang bisa muncul dari gerakan robot itu sendiri atau gerakan tangan yang menyebabkan lantai berguncang.



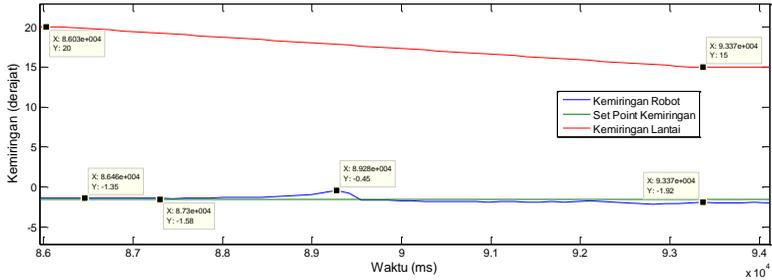
Gambar 4.11 Respon Kemiringan Robot Terhadap Gerakan *Roll* Ketika Terjadi Perubahan Kemiringan $15^0 - 20^0$



Gambar 4.12 Respon Kemiringan Robot Terhadap Gerakan *Roll* Ketika Kemiringan 20^0

Seperti halnya pada Gambar 4.9 dan 4.10, pada Gambar 4.11 dan 4.12 tidak terjadi perubahan yang signifikan pada respon kemiringan robot terhadap perubahan kemiringan lantai. Dari pengujian kemiringan tersebut ketika kemiringan bertambah dari 0^0 sampai dengan 20^0 robot dapat menyesuaikan kemiringannya dan menjaga kemiringannya di sekitar nilai *set point* dengan nilai *error steady state* rata – rata sebesar $0,31^0$. Nilai *error steady state* tersebut tidak dapat diatasi oleh aktuator yakni motor servo yang memiliki keterbatasan resolusi sebesar $0,3^0$ ketika tanpa beban. Keterbatasan ini dapat diatasi salah satunya dengan mengganti motor servo yang digunakan dengan motor servo dengan spesifikasi yang lebih tinggi pada spesifikasi torsi nya dan spesifikasi resolusinya sehingga bisa meningkatkan keakuratan aktuator ketika kondisi berbeban.

Perubahan respon terjadi ketika kemiringan diturunkan dari 20^0 sampai dengan 15^0 yang ditunjukkan oleh Gambar 4.13.

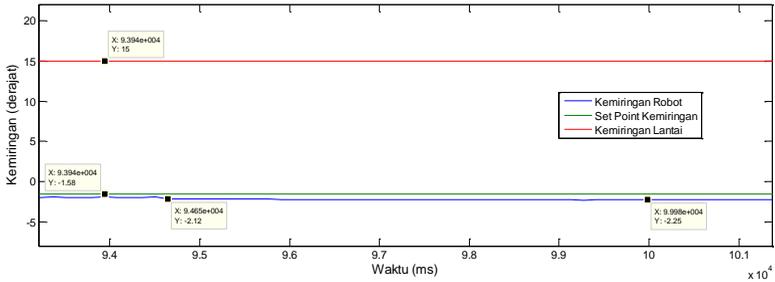


Gambar 4.13 Respon Kemiringan Robot Terhadap Gerakan *Roll* Ketika Terjadi Perubahan Kemiringan 20^0 - 15^0

Berdasarkan Gambar 4.13 terjadi simpangan kemiringan sebesar $0,45^0$ yang menyebabkan perubahan *error* menjadi $1,13^0$. Namun robot kembali berhasil menyesuaikan kemiringannya mendekati *set point* setelah 40 detik dengan *error steady state* sebesar $0,34^0$ yang tidak jauh dengan kondisi sebelumnya. Perubahan simpangan ini terjadi karena perubahan kecepatan penurunan kemiringan yang besar dalam sesaat sehingga muncul robot merespon dengan cepat dan terjadi guncangan pada badan robot akibat gerakan secara tiba – tiba. Perubahan kecepatan ini terjadi karena kemiringan lantai diatur dengan gerakan tangan secara *manual* dan alat ukur kemiringannya adalah busur derajat yang tidak dapat mengukur besarnya perubahan kecepatan perubahan sudut. Sementara sensor yang digunakan pada robot adalah IMU yang didalamnya terdapat *accelerometer* yang mempengaruhi bacaan sensor *gyroscope* ketika terdapat perubahan kecepatan secara tiba – tiba karena sensor ini terangkum dalam sebuah *Integrated Circuit* (IC) dan terhubung satu sama lain dalam sebuah sistem IMU sehingga sensor tersebut dapat membaca perubahan kemiringan secara tiba – tiba. Karena fitur *accelerometer* tidak digunakan pada sistem ini maka adanya perubahan kecepatan secara tiba – tiba tersebut berada di luar jangkauan kontroler yang telah dibuat.

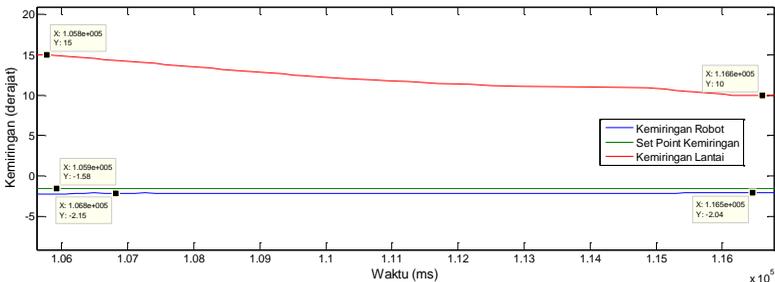
Setelah diturunkan pada kemiringan 15^0 , kemiringan tersebut dijaga selama 70 detik untuk diamati responnya dan respon kemiringan robot

terhadap kemiringan sebesar 15^0 pada saat kemiringan diturunkan ditunjukkan oleh Gambar 4.14

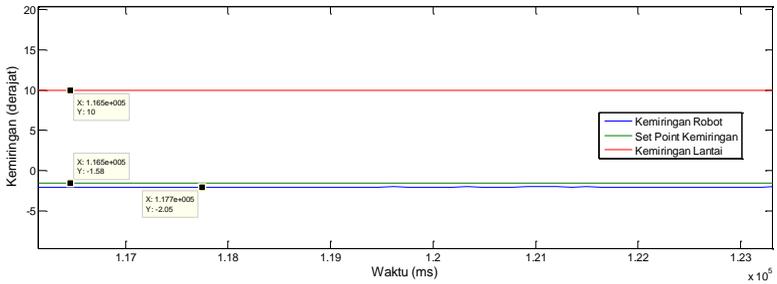


Gambar 4.14 Respon Kemiringan Robot Terhadap Gerakan *Roll* Ketika Kemiringan 15^0 Kondisi Kemiringan Turun

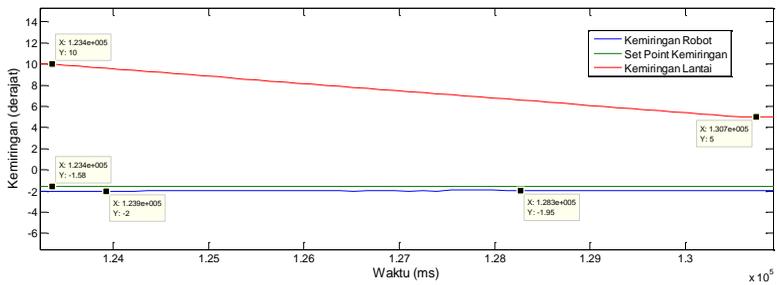
Berdasarkan Gambar 4.14 robot dapat menjaga kemiringannya mendekati *set point* dengan nilai *error steady state* $0,4^0$. Nilai ini berbeda sebesar $0,09^0$ dengan nilai *error steady state* pada kondisi kemiringan naik. Melihat respon kemiringan robot yang tidak dapat mencapai nilai *set point* hal ini mirip dengan situasi pada saat kemiringan dinaikkan sehingga nilai *error* tersebut masih terlalu kecil untuk diberikan pada aktuator akibat efek pembulatan pada proses perhitungan pada mikrokontroler dan efek beban pada motor servo yang digunakan. Respon serupa terjadi ketika kemiringan diturunkan dari 15^0 hingga mencapai kemiringan 5^0 yang ditunjukkan oleh Gambar 4.15 - Gambar 4.18



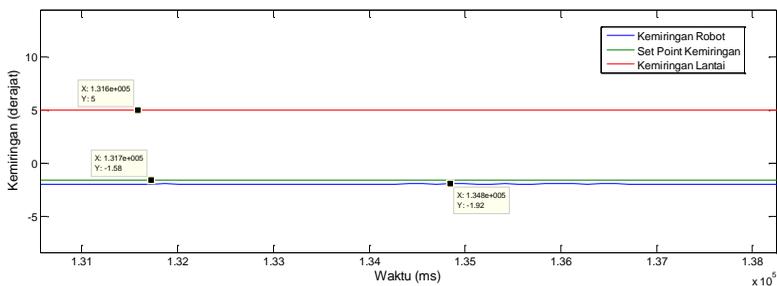
Gambar 4.15 Respon Kemiringan Robot Terhadap Gerakan *Roll* Ketika Terjadi Perubahan Kemiringan $15^0 - 10^0$



Gambar 4.16 Respon Kemiringan Robot Terhadap Gerakan *Roll* Ketika Kemiringan 10^0 Kondisi Kemiringan Turun

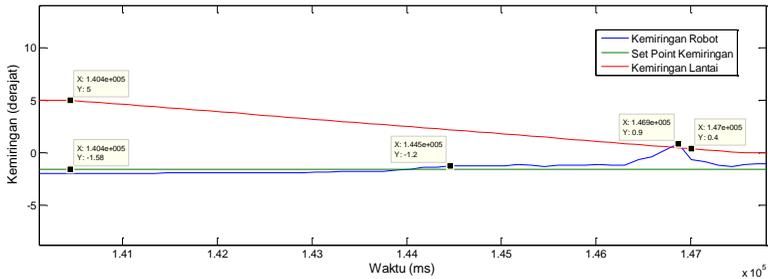


Gambar 4.17 Respon Kemiringan Robot Terhadap Gerakan *Roll* Ketika Terjadi Perubahan Kemiringan $10^0 - 5^0$

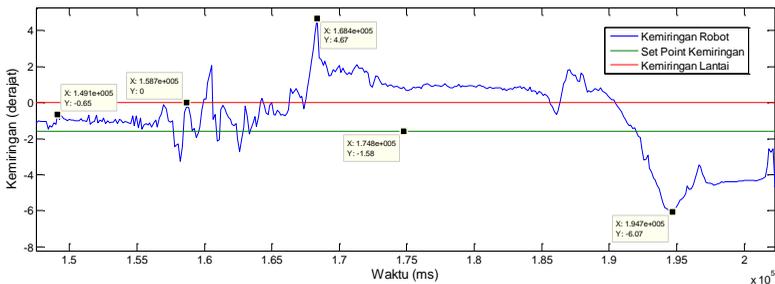


Gambar 4.18 Respon Kemiringan Robot Terhadap Gerakan *Roll* Ketika Kemiringan 5^0 Kondisi Kemiringan Turun

Respon yang berbeda terjadi ketika kemiringan diturunkan dari 5^0 menuju ke 0^0 . Pada akhir kemiringan ketika mencapai kemiringan 0^0 terjadi simpangan yang cukup signifikan akibat guncangan yang ditimbulkan ketika meletakkan papan lantai sehingga robot merespon dengan cepat perubahan tersebut. Simpangan yang terjadi sebesar $0,9^0$ dan robot dapat menyesuaikan kemiringannya selama 10 detik mendekati *set point*. Respon kemiringan robot pada kemiringan lantai sebesar 5^0 ditunjukkan oleh Gambar 4.19



Gambar 4.19 Respon Kemiringan Robot Terhadap Gerakan *Roll* Ketika Terjadi Perubahan Kemiringan $5^0 - 0^0$

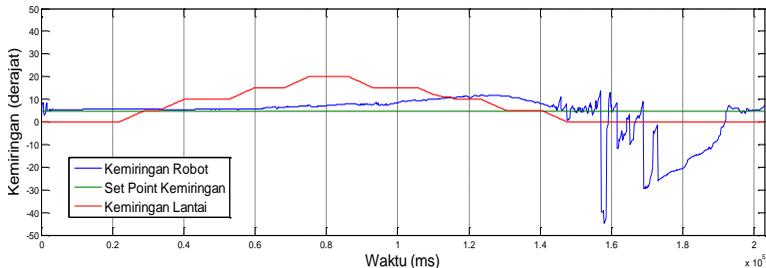


Gambar 4.20 Respon Kemiringan Robot Terhadap Gerakan *Roll* Ketika Kemiringan 50^0 Kondisi Kemiringan Turun

Pada kemiringan 0^0 ketika kemiringan turun, terdapat guncangan akibat pendaratan lantai yang tidak mulus seperti yang ditunjukkan oleh Gambar 4.20. Kemudian simpangan yang besar muncul ketika tombol *stop* pada robot ditekan karena hal ini seperti memberikan gangguan pada

robot, sehingga sebenarnya bagian yang memiliki simpangan besar bisa diabaikan karena terdapat variabel yang tidak dapat dikontrol yaitu adanya gangguan dari luar yang berasal dari penekanan tombol *stop*.

4.3.2 Pengujian Kondisi Berdiri Terhadap Gerakan *Pitch*

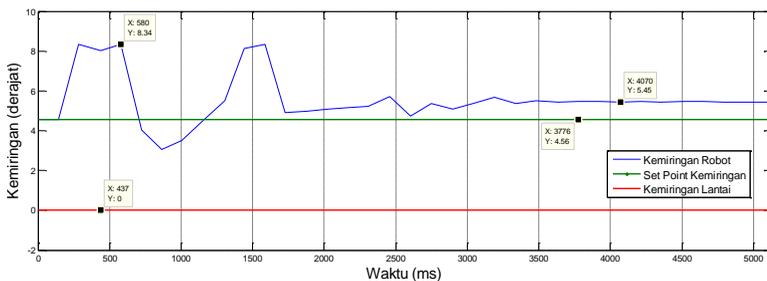


Gambar 4.21 Keseluruhan Respon Kemiringan Robot Ketika Gerakan *Pitch*

Pengujian respon robot terhadap gerakan *pitch* dilakukan seperti pada pengujian gerakan *roll* yang membedakan adalah arah gerakan putarnya yaitu lantai pada depan robot dinaikkan dan diturunkan pada *interval* kemiringan $0^{\circ} - 20^{\circ}$ naik dan $20^{\circ} - 0^{\circ}$ turun. Setiap *interval* naik dan turun dibagi ke dalam perubahan nilai sebesar 5° dan kemudian diamati respon kemiringan robot terhadap lantai yang secara keseluruhan ditunjukkan oleh Gambar 4.21. Berdasarkan Gambar 4.21 secara keseluruhan dapat dilihat bahwa robot dapat menjaga kemiringannya ketika terjadi kenaikan perubahan kemiringan, namun performanya menurun ketika kemiringan diturunkan. Banyak faktor yang menyebabkan terjadinya hal ini, terutama adalah kecepatan perubahan kemiringan yang ikut mempengaruhi respon robot terhadap kemiringan lantai.

Besarnya kecepatan perubahan kemiringan meningkatkan kecepatan respon robot terhadap kemiringan ditambah dengan penggunaan kontroler proporsional dan integral yang ikut mempengaruhi hal ini, sedangkan nilai konstanta differensial yang kecil mengakibatkan nilai redamannya menjadi kurang sehingga ketika terjadi kenaikan kecepatan perubahan kemiringan, robot ikut menaikkan sinyal kontrolnya namun tidak diiringi dengan redaman yang cukup sehingga ketika tiba – tiba perubahan

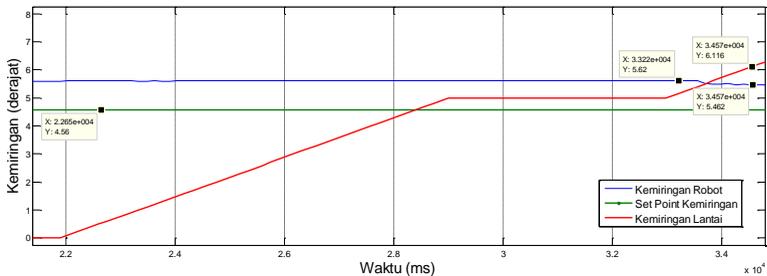
kemiringan tersebut berhenti nilai *error* justru menjadi besar karena gerakan robot yang tiba – tiba pada saat kecepatan perubahannya besar mengakibatkan posisinya menjauhi *set point*. Ketika hal ini terjadi, kontroler akan meningkatkan besarnya sinyal kontrol, sehingga kemiringan robot bisa kembali menuju *set point* dengan cepat, namun karena besarnya redaman yang kurang justru berakibat pada terjadinya osilasi di sekitar *set point* yang cukup lama. Hal ini memang berada di luar jangkauan kontroler yang dirancang hanya melihat kemiringan tanpa mengukur perubahan kemiringan yang terjadi sehingga munculnya percepatan tidak dapat diatasi oleh robot ini. Meskipun demikian, dengan menjaga kecepatan perubahan kemiringan sekecil mungkin dapat dilihat respon robot terhadap kemiringan lantai pijakannya. Respon kemiringan robot terhadap gerakan *pitch* pada kemiringan 0^0 ditunjukkan oleh Gambar 4.22.



Gambar 4.22 Respon Kemiringan Robot Terhadap Gerakan *Pitch* Ketika Kemiringan 0^0

Berdasarkan Gambar 4.22, pada waktu setelah kontroler dinyalakan, diambil nilai *set point* kemiringan robot sebesar $4,56^0$, kemudian muncul *overshoot* sebesar 82% pada waktu 580 milidetik dan secara pelan – pelan teredam hingga sampai pada waktu 4 detik dapat mencapai kondisi *steady state* dengan nilai *error steady state* sebesar $0,89^0$. Fenomena respon *transient* ini sama seperti pada pengujian respon robot terhadap gerakan *roll* dan penyebabnya pun juga sama karena kontroler yang digunakan pada gerakan *pitch* sama dengan yang digunakan pada gerakan *roll*. Pada pengujian gerakan *pitch* nilai parameter PID yang digunakan adalah $K_p = 0,5$; $K_i = 0,01$; $K_d = 0,01$. munculnya *overshoot* tersebut terjadi karena efek aksi kontrol proporsional dan integral seperti pada gerakan *roll* yang

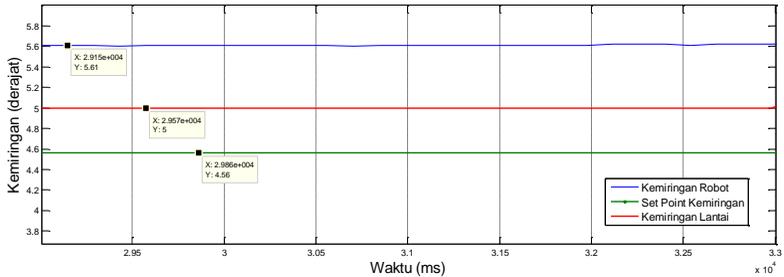
mana pada saat awal nilai semua variabel bernilai nol dan saat pertama membaca kemiringan, besarnya kemiringan tersebut dibandingkan dengan *set point* dan diperoleh nilai *error* besarnya *error* dikalikan diolah oleh kontroler PID untuk menjadi sinyal kontrol yang akan diberikan kepada aktuator yaitu motor servo. Kecilnya konstanta differensial yang berperan sebagai peredam menyebabkan nilai *overshoot* menjadi besar. Hal itu dapat diatasi dengan memperbesar nilai konstanta differensial melalui proses *tuning* ulang. Setelah mencapai keadaan *steady state* kemiringan secara perlahan – lahan dinaikkan sampai dengan 5^0 , kemudian diamati respon kemiringan robot terhadap lantai seperti yang ditunjukkan oleh Gambar 4.23



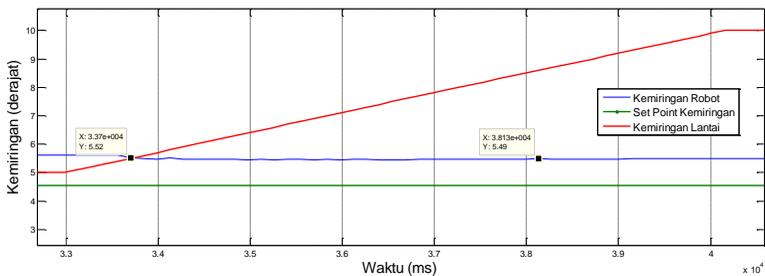
Gambar 4.23 Respon Kemiringan Robot Terhadap Gerakan *Pitch* Ketika Terjadi Perubahan Kemiringan $0^0 - 5^0$

Pada saat terjadi perubahan kemiringan, robot dapat menjaga kemiringannya di sekitar *set point* dengan nilai *error steady state* sebesar $1,06^0$ nilai ini terus bertahan bahkan sampai dengan ketika kemiringan 5^0 tersebut ditahan selama 30 detik. *Error steady state* yang kecil ini tidak dapat ditangani karena keterbatasan kemampuan aktuator ketika diberi beban yang mengakibatkan resolusinya menurun, sehingga *error steady state* diabaikan oleh aktuator. Perubahan kecil terjadi ketika kemiringan lantai dinaikkan dari 5^0 menuju 10^0 . Perubahan tersebut justru mengarah mendekati *set point*. Hal ini terjadi karena aksi kontrol integral yang meningkat seiring bertambahnya nilai jumlah *error*. Peningkatan aksi kontrol integral menyebabkan aktuator dapat beraksi sesuai dengan sinyal kontrol yang diberikan, namun ketika memasuki area di bawah spesifikasi resolusinya maka aktuator akan berhenti beraksi sehingga tetap muncul *error steady state* meskipun nilainya kecil dan tidak berpengaruh terhadap

keseimbangan robot secara keseluruhan. Perubahan respon tidak terlalu drastis hingga kemiringan robot mencapai kemiringan 15^0 seperti yang ditunjukkan oleh Gambar 4.24 sampai dengan Gambar 4.25



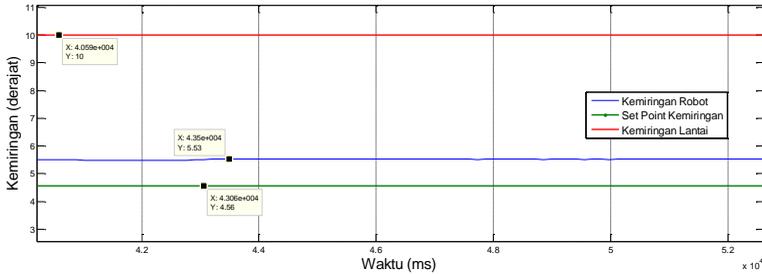
Gambar 4.24 Respon Kemiringan Robot Terhadap Gerakan *Pitch* Ketika Kemiringan 5^0



Gambar 4.25 Respon Kemiringan Robot Terhadap Gerakan *Pitch* Ketika Terjadi Perubahan Kemiringan $5^0 - 10^0$

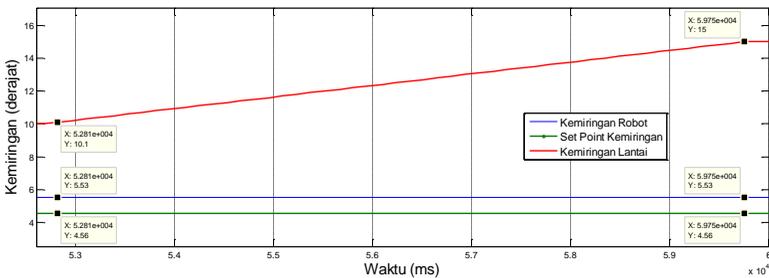
Berdasarkan Gambar 4.24 dan Gambar 4.25 tidak terjadi perubahan yang signifikan pada respon kemiringan robot terhadap gerakan *pitch* yang menyebabkan perubahan kemiringan lantai. Rata – rata nilai *error steady state* pun juga tidak terjadi banyak perubahan yaitu sebesar $0,945^0$ dan nilai ini masih diabaikan oleh aktuator karena efek pembulatan pada perhitungan serta efek beban pada aktuator yang mengakibatkan aksi kontrol tidak memberikan efek yang berarti bahkan terlihat tidak memberikan aksi kontrol ketika terjadi *error steady state*. Penambahan nilai konstanta integral dapat mempengaruhi hal ini namun akan menjadi

berbahaya ketika terjadi guncangan pada lantai atau robot karena percepatan gerakan robot tidak dapat diatasi oleh kontroler ini. Respon kemiringan robot ketika kemiringan lantai dijaga pada kemiringan 10^0 ditunjukkan oleh Gambar 4.26



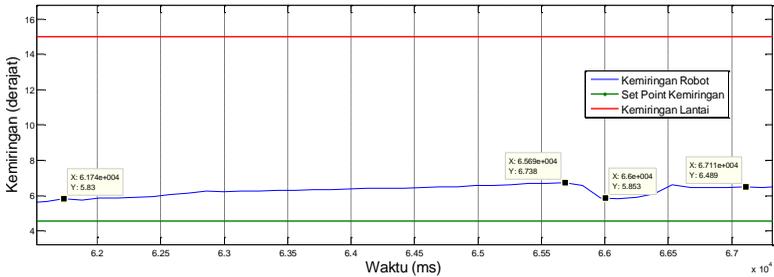
Gambar 4.26 Respon Kemiringan Robot Terhadap Gerakan *Pitch* Ketika Kemiringan 10^0

Berdasarkan Gambar 4.26, respon kemiringan robot tidak banyak berubah dibandingkan dengan kondisi – kondisi sebelumnya. Perubahan hanya sedikit pada kondisi *error steady state* yaitu sebesar $0,97^0$ dari sebelumnya sebesar $0,945^0$ sehingga terjadi perubahan sebesar $0,025^0$. Nilai sebesar ini apabila dilihat dengan kasat mata tidak akan terlihat ada perubahan. Hal yang sama ditunjukkan ketika kemiringan dinaikkan menuju 15^0 yang ditunjukkan oleh Gambar 4.27



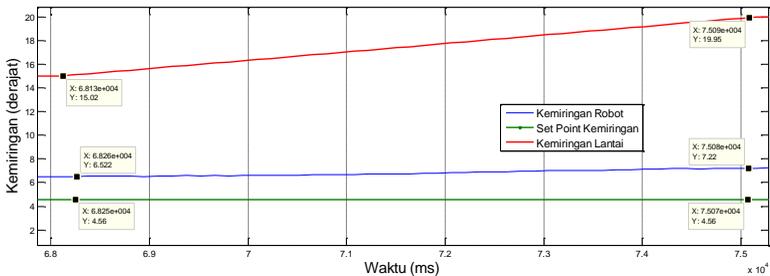
Gambar 4.27 Respon Kemiringan Robot Terhadap Gerakan *Pitch* Ketika Terjadi Perubahan Kemiringan $10^0 - 15^0$

Perubahan respon terjadi ketika kemiringan lantai dijaga pada kemiringan 15^0 dimana terdapat guncangan yang mengakibatkan sinyal kontrol menjauhi *set point* pada waktu 65 detik sampai dengan 66 detik kemudian kembali lagi pada kondisi *steady state* pada waktu 67 detik atau membutuhkan waktu 2 detik untuk kembali ke kondisi *steady state* seperti yang ditunjukkan oleh Gambar 4.28



Gambar 4.28 Respon Kemiringan Robot Terhadap Gerakan *Pitch* Ketika Kemiringan 15^0

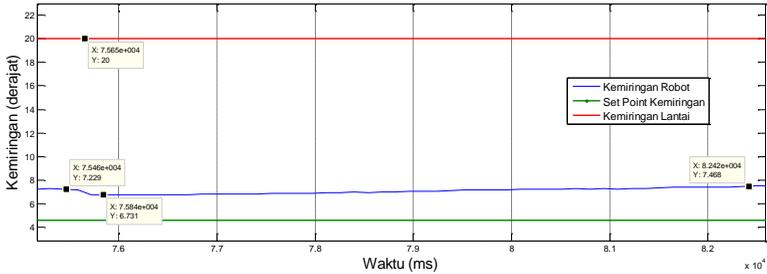
Kemiringan lantai kemudian dinaikkan hingga mencapai 20^0 dan dilihat respon kemiringan robot terhadap perubahan kemiringan ini seperti yang ditunjukkan oleh Gambar 4.29



Gambar 4.29 Respon Kemiringan Robot Terhadap Gerakan *Pitch* Ketika Terjadi Perubahan Kemiringan 15^0 - 20^0

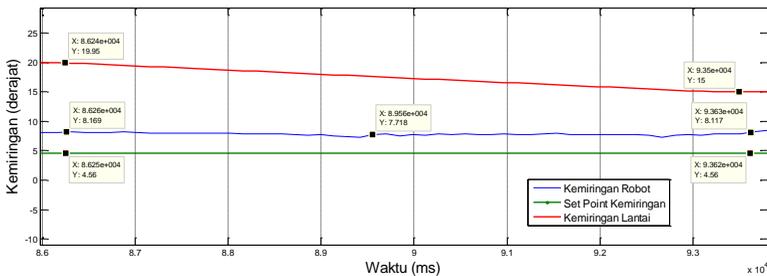
Berdasarkan Gambar 4.29 dapat dilihat bahwa tidak terjadi perubahan yang signifikan ketika terjadi perubahan kemiringan dari 15^0

menuju 20° . Hal yang sama ditunjukkan oleh Gambar 4.30 ketika kemiringan dijaga sebesar 20° .



Gambar 4.30 Respon Kemiringan Robot Terhadap Gerakan *Pitch* Ketika Kemiringan 20°

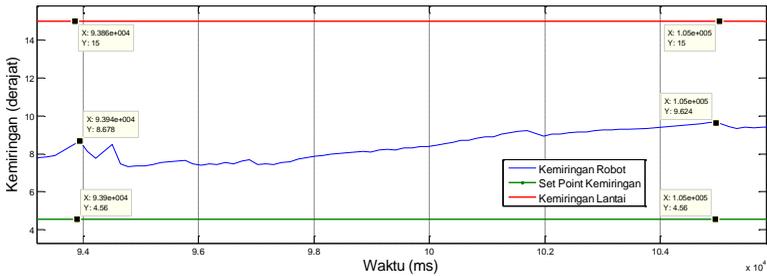
Setelah mencapai kemiringan 20° , lantai tempat robot berpijak diturunkan kemiringannya secara perlahan hingga kembali pada kemiringan awal sebesar 0° . Dalam perjalanannya menuju 0° kemiringan tersebut dibagi ke dalam *interval* perubahan sebesar 5° . Respon kemiringan robot ketika kemiringan diturunkan menuju 15° ditunjukkan oleh Gambar 4.31



Gambar 4.31 Respon Kemiringan Robot Terhadap Gerakan *Pitch* Ketika Terjadi Perubahan Kemiringan $20^{\circ} - 15^{\circ}$

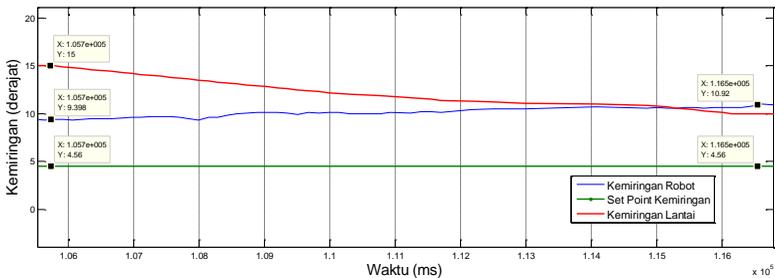
Berdasarkan Gambar 4.31, sempat terjadi perubahan respon ketika kemiringan diturunkan pada waktu 89 detik namun tidak terlalu besar yaitu sebesar $0,33^{\circ}$ dan secara kasat mata robot dalam keadaan seimbang.

Kemudian kemiringan lantai dijaga pada 15^0 untuk melihat respon kemiringan robot seperti yang ditunjukkan oleh Gambar 4.32



Gambar 4.32 Respon Kemiringan Robot Terhadap Gerakan *Pitch* Ketika Kemiringan 15^0 Kondisi Kemiringan Turun

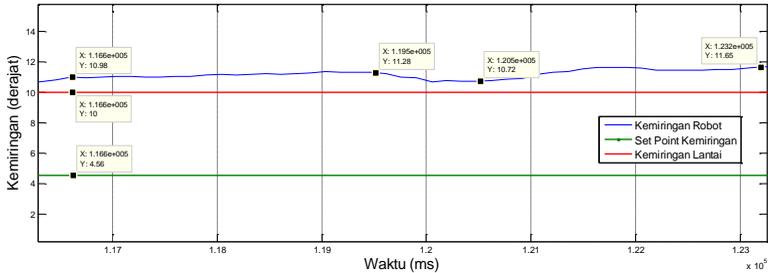
Berdasarkan Gambar 4.32 respon robot *transient* ketika kemiringan dijaga pada 15^0 . Hal ini muncul karena kecepatan perubahan kemiringan yang cukup besar yang terukur oleh perangkat IMU sehingga mengakibatkan respon sinyal kontrol yang cukup besar menyesuaikan dengan besarnya kecepatan perubahan kemiringan. Kemiringan kemudian diturunkan menuju 10^0 dan respon kemiringan robot ditunjukkan oleh Gambar 4.33



Gambar 4.33 Respon Kemiringan Robot Terhadap Gerakan *Pitch* Ketika Terjadi Perubahan Kemiringan $15^0 - 10^0$

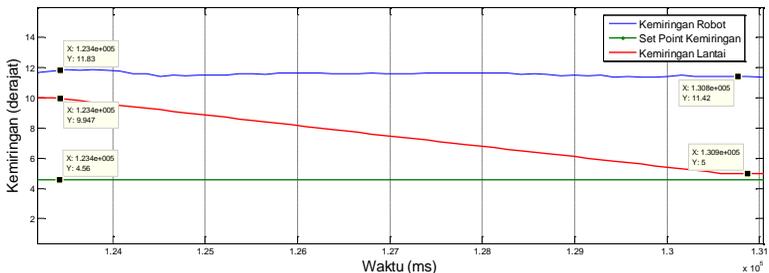
Berdasarkan Gambar 4.33 terlihat bahwa respon masih dalam keadaan *transient* dan semakin menjauhi *set point*. Dalam hal ini,

kecepatan perubahan menjadi hal yang berpengaruh namun berada di luar jangkauan kontroler yang telah dirancang. Setelah diturunkan, kemiringan lantai kemudian dijaga pada kemiringan 10^0 dan diamati respon kemiringannya seperti yang ditunjukkan oleh Gambar 4.34.



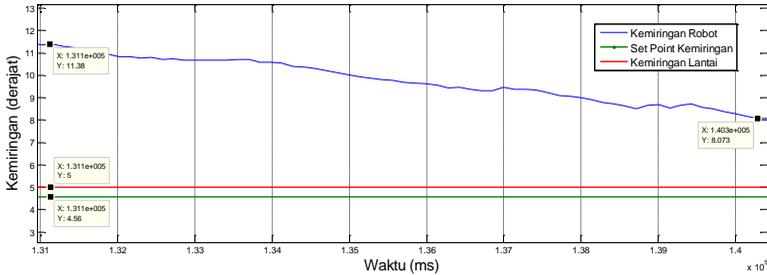
Gambar 4.34 Respon Kemiringan Robot Terhadap Gerakan *Pitch* Ketika Kemiringan 10^0 Kondisi Kemiringan Turun

Berdasarkan Gambar 4.34, respon robot masih dalam keadaan *transient* dengan nilai *error* rata – rata sebesar $7,09^0$. Aksi kontrol yang dilakukan tidak cukup cepat untuk mengembalikan kondisi kemiringan mendekati nilai *set point* namun apabila diperbesar, guncangan pada robot akan berakibat lebih buruk karena tidak ada pengatur percepatan gerakan robot. Kemiringan kemudian kembali diturunkan hingga menuju 5^0 dan diamati respon kemiringan robot seperti yang ditunjukkan oleh Gambar 4.35



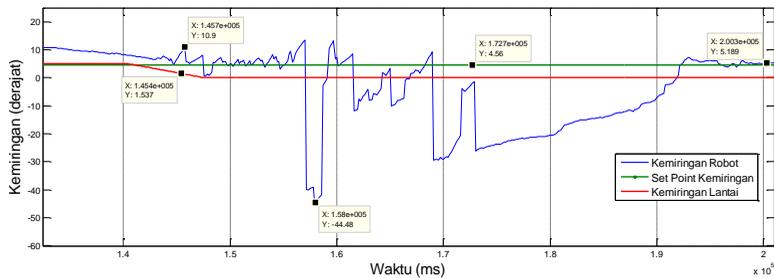
Gambar 4.35 Respon Kemiringan Robot Terhadap Gerakan *Pitch* Ketika Terjadi Perubahan Kemiringan $10^0 - 5^0$

Respon kemiringan robot masih berada dalam kondisi *transient* berdasarkan Gambar 4.35 dengan nilai *error* rata – rata sebesar $7,06^0$ sedikit lebih kecil dari kondisi sebelumnya, namun aksi kontrol masih belum begitu cepat membawa kemiringan mendekati nilai *set point*. Kemiringan pun dijaga pada 5^0 dan respon kemiringan robot pada saat kemiringan lantai dijaga pada 5^0 ditunjukkan oleh Gambar 5.36.



Gambar 4.36 Respon Kemiringan Robot Terhadap Gerakan *Pitch* Ketika Kemiringan 5^0 Kondisi Kemiringan Turun

Gambar 4.36 menunjukkan bahwa respon kemiringan robot mulai mendekati *set point* dengan nilai *error* rata – rata sebesar $5,34^0$ turun sekitar 2^0 dari kondisi sebelumnya. Kemiringan kemudian kembali diturunkan hingga 0^0 dan respon kemiringan robot ditunjukkan oleh Gambar 4.37



Gambar 4.37 Respon Kemiringan Robot Terhadap Gerakan *Pitch* Ketika Terjadi Perubahan Kemiringan $5^0 - 0^0$

Berdasarkan Gambar 4.37 respon kemiringan robot mencapai nilai *set point* pada waktu 109 detik dari awal pengujian dan terjadi osilasi di sekitar nilai *set point* dan karena pendaratan papan yang tidak mulus mengakibatkan terjadinya guncangan pada lantai yang berakibat guncangan pula pada robot bahkan terjadi simpangan maksimal sebesar $44,48^{\circ}$ akibat guncangan ini dan robot langsung merespon dan kembali ke kondisi *steady state* dalam waktu 40 detik. Dari pengujian ketika kondisi berdiri, secara umum respon kemiringan robot terhadap perubahan kemiringan lantai sangat dipengaruhi oleh besarnya kecepatan perubahan kemiringan lantai sehingga diperlukan modifikasi kontroler dengan melibatkan variabel percepatan yang ikut dikontrol agar respon kemiringan dapat lebih baik.

4.3.3 Pengujian Kondisi Berjalan

Penerapan kontroler *upright pose* pada kondisi berjalan belum sempurna. Hal ini karena desain *walking gait* yang belum seimbang. *Stable walking gait* tidak dapat diterapkan pada robot ini, karena terkendala spesifikasi motor servo pada engkel. Motor ini tidak kuat menerima seluruh beban robot pada fase gerakan *single support* atau berdiri dengan satu kaki. Sehingga dibuat *walking gait* yang lain yang mempercepat langkah kaki yang berpengaruh pada waktu fase berdiri satu kaki. Hasilnya robot dapat berjalan dengan kecepatan 7 cm/s pada bidang datar dan bidang miring dengan kemiringan maksimal 20° . Kendala *delay* komunikasi *wireless* sebesar 2 detik pada proses *telemonitoring* yang mengganggu gerakan robot sehingga gerakannya menjadi lambat. Hal ini terlalu berbahaya dan dapat berakibat kerusakan pada motor servo bagian engkel yang menjadi penumpu ketika gerakan robot berada pada fase *single support*, karena spesifikasi torsi motor servo pada engkel robot di bawah spesifikasi beban keseluruhan robot. Apabila *delay* komunikasi dihilangkan maka gerakan robot menjadi lebih cepat dan lebih aman bagi robot, tetapi kondisi kemiringan robot tidak dapat termonitor.

BAB 5

PENUTUP

5.1 Kesimpulan

1. Hasil analisa *inverse* kinematik bagian kaki berhasil diterapkan pada robot dengan *error* rata – rata sebesar 2,8 mm pada sumbu X, 2,7 mm pada sumbu Y, dan 11,71 mm pada sumbu Z. Metode ini memberikan kemudahan untuk menggerakkan kaki pada posisi tertentu.
2. Penerapan kontroler *upright pose* pada robot saat kondisi berdiri menghasilkan performa *overshoot* awal sebesar 89% dan mencapai *steady state* setelah 5,45 detik dengan nilai *error steady state* rata –rata sebesar 0,95⁰ ketika kemiringan lantai dinaikkan, dan sebesar 7,23⁰ ketika kemiringan lantai diturunkan pada kemiringan *pitch* dengan nilai parameter PID, $K_p = 0,5$; $K_i = 0,01$; $K_d = 0,01$. Pada gerakan *roll* dengan nilai $K_p = 0,5$; $K_i = 0,05$; dan $K_d = 0,01$, menghasilkan performa *overshoot* awal sebesar 98 % dan mencapai *steady state* setelah 19,35 detik dengan nilai *error steady state* rata –rata sebesar 0,31⁰ ketika kemiringan lantai dinaikkan, dan sebesar 0,52⁰ ketika kemiringan lantai diturunkan. Proses pendaratan lantai yang kurang halus dan kecepatan perubahan kemiringan yang signifikan mengakibatkan terjadinya osilasi yang mempengaruhi performa robot. Hal ini tidak dapat dijangkau oleh kontroler ini karena munculnya variabel percepatan tidak diatur.
3. Penerapan kontroler *upright pose* pada kondisi berjalan belum sempurna. Hal ini karena kondisi robot yang berguncang akibat hentakan kaki saat berjalan dan desain *walking gait* yang belum seimbang serta kendala *delay* komunikasi *wireless* sebesar 2 detik pada proses *telemonitoring* yang mengganggu gerakan robot sehingga gerakannya menjadi lambat. Hal ini terlalu berbahaya dan dapat berakibat kerusakan pada motor servo bagian engkel yang menjadi penumpu ketika gerakan robot berada pada fase *single support*, karena spesifikasi torsi motor servo pada engkel robot di bawah spesifikasi beban keseluruhan

robot. Apabila *delay* komunikasi dihilangkan maka gerakan robot menjadi lebih cepat dan lebih aman bagi robot, tetapi kondisi kemiringan robot tidak dapat termonitor.

5.2 Saran

1. Torsi pada *joint* pinggang depan pinggang belakang, engkel depan dan engkel belakang perlu dikuatkan agar *stable walking gait* dapat diterapkan pada robot ini.
2. *Walking gait* sebelum diberi kontroler perlu diperbaiki sehingga performa jalan robot dapat lebih stabil ketika berjalan pada bidang datar.
3. Diperlukan perangkat mikrokontroler dengan spesifikasi yang lebih tinggi atau dua buah mikrokontroler dengan konfigurasi *master – slave* untuk pemrosesan data pada sistem ini sehingga menjadi lebih cepat dan dapat mengurangi *delay* pengiriman pada komunikasi *wireless*.

DAFTAR PUSTAKA

- [1] I.-W. Park, J.-Y. Kim, J. Lee and J.-H. Oh, "Online Free Walking Trajectory Generation for Biped Humanoid Robot KHR-3(HUBO)," in *International Conference on Robotic and Automation*, Orlando, Florida, 2006.
- [2] M. W.Spong, S. Hutchinson and M.Vidyasagar, "*Robot Modeling and Control*", New York: John Wiley& Sons, Inc, 2006.
- [3], "*Atlas - The Agile Anthrhopomorphic Robot*",<URL: http://www.bostondynamics.com/robot_Atlas.html>,2015.
- [4], "*Motoman Arc Welding Robot Models*",<URL: <http://www.motoman.com/products/robots/arc-welding-robots.php>>, 2015.
- [5] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi and H. Hirukawa, "Biped Walking Pattern Generation by using Preview Control of Zero-Moment Point," in *International Conference on Robotics & Automation*, Taipei, 2003.
- [6], "*ASIMO The World's Most Advanced Humanoid Robot*", <URL: <http://asimo.honda.com/asimo-specs>>, 2015.
- [7], "*ROBOTIS e-manual v1.25.00 AX-12/AX-12+/AX-12A*", <URL:<http://support.robotis.com>>, 2015.
- [8] Gamayanti, Nurlita, "*Diktat Mata Kuliah Dasar Sistem Pengaturan*", Institut Teknologi Sepuluh Nopember, Surabaya, 2013.
- [9] J.-Y. Kim, I.-W. Park and J.-H. Oh, "*Walking Control Agorithm of Biped Humanoid Robot on Uneven and Inclined Floor*", Korea Advanced Institut of Science and Technology, Daejeon, 2007.
- [10] K. Fu, R. Gonzalez and C. Lee, "*ROBOTICS: Control, Sensing, Vision and Intelligence*", New York: McGraw-Hill Book Company, 1987.
- [11], "*explainthatstuff!*",<URL:www.explainthatstuff.com>, 2015.

- [12], "*THAOYU Electronics HAOYU Electronics MPU-6050*", <URL:<http://www.hotmcu.com/gy521-mpu6050-3axis-acceleration-gyroscope-6dof-module-p-83.html>>, 2015.
- [13], "*NRF24L01+ Wireless Transciever Module*", <URL:<http://www.amazon.co.uk/SainSmart-NRF24L01-Wireless-Transceiver-Module/dp/B006CHFUFU>>, 2015
- [14], "*OLX*", <URL: <http://olx.co.id/iklan/turnigy-lipo-battery-2s-1000mah-7-4v-20c-ID6u7e6.html>>, 2015.
- [15], "*ArduinoBoardMega2560*", <URL:<https://www.arduino.cc/en/Main/ArduinoBoardMega2560>>, 2015
- [16], "*ArduinoBoardUno*", <URL:<https://www.arduino.cc/en/Main/ArduinoBoardUno>>, 2015.