

SKRIPSI

**SISTEM KENDALI KESEIMBANGAN BERDIRI ROBOT HUMANOID
PADA PERMUKAAN BIDANG MIRING**

***HUMANOID ROBOT STANDING BALANCE CONTROL SYSTEM ON THE
INCLINED SURFACE***



DIMAS FAJRIAN NUGROHO
13/347550/PA/15300

**PROGRAM STUDI ELEKTRONIKA DAN INSTRUMENTASI
DEPARTEMEN ILMU KOMPUTER DAN ELEKTRONIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS GADJAH MADA
YOGYAKARTA
2017**

SKRIPSI

**SISTEM KENDALI KESEIMBANGAN BERDIRI ROBOT HUMANOID
PADA PERMUKAAN BIDANG MIRING**

***HUMANOID ROBOT STANDING BALANCE CONTROL SYSTEM ON THE
INCLINED SURFACE***

Diajukan untuk memenuhi salah satu syarat memperoleh derajat
Sarjana Sains Program Studi Elektronika dan Instrumentasi



DIMAS FAJRIAN NUGROHO
13/347550/PA/15300

**PROGRAM STUDI ELEKTRONIKA DAN INSTRUMENTASI
DEPARTEMEN ILMU KOMPUTER DAN ELEKTRONIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS GADJAH MADA
YOGYAKARTA
2017**

PERNYATAAN

Dengan ini saya menyatakan bahwa laporan skripsi ini tidak terdapat karya yang pernah diajukan untuk memperoleh gelar kesarjanaan di suatu Perguruan Tinggi, dan sepanjang pengetahuan saya juga tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali secara tertulis diacu dalam naskah ini dan disebutkan dalam daftar pustaka.

Yogyakarta, 7 Juni 2017

Dimas Fajrian Nugroho



HALAMAN PENGESAHAN

SKRIPSI

SISTEM KENDALI KESEIMBANGAN BERDIRI ROBOT HUMANOID PADA PERMUKAAN BIDANG MIRING

Telah dipersiapkan dan disusun oleh

DIMAS FAJRIAN NUGROHO
13/347550/PA/15300

Telah dipertahankan di depan Tim Penguji
pada tanggal 7 Juni 2017

Bakhtiar Alldino Ardi Sumbodo, S.Si., M.Cs.
Pembimbing I

Panggih Basuki, Drs., M.Si.
Penguji I

Andi Dharmawan, S.Si., M.Cs.
Pembimbing II

Abdul Ro'uf, Drs., M.Kom.
Penguji II

Mengetahui
d.n. Dekan FMIPA-UGM
Wakil Dekan Bidang Akademik dan
Kemahasiswaan



Haniat Aprilita, M.Si.
NIP. 197903071980317002

KATA PENGANTAR

Puji dan syukur kehadiran Tuhan Yang Maha Esa atas limpahan karunia dan bimbingan-Nya sehingga penulis dapat menyelesaikan skripsi yang berjudul “Sistem Kendali Keseimbangan Berdiri Robot Humanoid pada Permukaan Bidang Miring”.

Skripsi ini disusun sebagai salah satu syarat untuk memperoleh gelar sarjana S1 Program Studi Elektronika dan Instrumentasi, Departemen Ilmu Komputer dan Elektronika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Gadjah Mada Yogyakarta. Proses penyusunan skripsi ini tidak lepas dari dukungan beberapa pihak. Oleh karena itu, penulis ingin menyampaikan rasa terima kasih dan penghargaan yang sebesar-besarnya kepada:

1. Keluarga tercinta, Bapak Sapto Nugroho Handono Wareh, Ibu Sri Neni Trianawati, Rezky Destrio Nugroho, Ismira Nugrahningrum dan Anindya Zahra Nugrahningrum yang selalu mendoakan, mendoakan dan memberikan motivasi kepada penulis dalam menyelesaikan skripsi ini.
2. Bapak Andi Dharmawan, S.Si., M.Cs. dan Bapak Bakhtiar Alldino Ardi Sumbodo, S.Si., M.Cs. yang telah bersabar dan meluangkan waktu untuk membimbing, memberikan ide dan pemikiran serta saran kepada penulis sehingga dapat menyelesaikan skripsi ini dengan baik.
3. Dosen-dosen Program Studi Elektronika dan Instrumentasi yang telah membimbing dan memberikan ilmunya kepada penulis.
4. Mas Vian, Mas Widar, Mas Ridwan, Mas Rafi, Mas Rudi, Mas Hakim, Mas Haris, Mas Abha, Thundra, Fahmi, Auzan, Resa, Ela, Asedio, Hafizna, Ichsan, Rani, dan Ajeng, serta semua tim e-drones yang merupakan tempat penulis berbagi ilmu serta ide-ide untuk menyelesaikan skripsi ini.
5. Seluruh keluarga ELINS angkatan 2013 yang tidak bisa disebutkan satu persatu, terimakasih atas dukungan, bantuan dan kebersamaan dalam proses penyelesaian skripsi ini.

6. Tim ALFAROBI UGM yang telah memberikan motivasi, dukungan, semangat, dan pengalaman berharga yang menginspirasi untuk melakukan penelitian ini.
7. Tim KKN NTB-04 Sub-unit Bangle yang telah memberikan dukungan moral dan semangat selama proses penyelesaian skripsi.
8. Yusila Nur Wijayanti yang sudah memberikan saran, dukungan, dan bantuan.
9. Semua pihak yang memberikan bantuan, semangat dan motivasi dalam proses penyelesaian skripsi ini yang tidak dapat disebutkan satu per satu.

Penulis menyadari terdapat kekurangan dan kesalahan dalam penelitian skripsi ini dikarenakan keterbatas ilmu dan pengetahuan yang dimiliki penulis. Oleh karena itu, kritik dan saran yang bersifat membangun dari semua pihak sangat diharapkan oleh penulis.

Yogyakarta, 7 Juni 2017

Dimas Fajrian Nugroho

DAFTAR ISI

| | |
|-------------------------------------------------------|------|
| KATA PENGANTAR | v |
| DAFTAR ISI..... | vii |
| DAFTAR GAMBAR..... | viii |
| DAFTAR TABEL..... | xi |
| INTISARI | xii |
| BAB I PENDAHULUAN..... | 1 |
| 1.1. Latar Belakang | 1 |
| 1.2. Rumusan Masalah | 3 |
| 1.3. Batasan Masalah..... | 3 |
| 1.4. Tujuan Penelitian..... | 3 |
| 1.5. Manfaat Penelitian..... | 3 |
| 1.6. Metodologi Penelitian | 3 |
| 1.7. Sistematika Penulisan..... | 5 |
| BAB II TINJAUAN PUSTAKA | 6 |
| BAB III LANDASAN TEORI..... | 9 |
| 3.1. Robot Humanoid | 9 |
| 3.2. <i>Rotation Matriks</i> pada robot humanoid..... | 13 |
| 3.3. <i>Forward Kinematic</i> | 14 |
| 3.4. <i>Inverse Kinematic</i> | 15 |
| 3.5. Model Pendulum Terbalik Dua Dimensi | 17 |
| 3.6. <i>Linear Quadratic Regulator (LQR)</i> | 18 |
| 3.7. <i>Kalman Filter</i> | 19 |
| BAB IV ANALISIS DAN PERANCANGAN SISTEM..... | 24 |
| 4.1. Analisis Sistem..... | 24 |
| 4.2. Rancangan Pemodelan Sistem | 25 |
| 4.3. Rancangan Sistem Kendali..... | 32 |
| 4.4. Rancangan Arsitektur Sistem | 35 |
| 4.5. Rancangan Perangkat Lunak Sistem | 36 |
| 4.5.1. Rancangan Fungsi <i>Setup</i> | 37 |
| 4.5.2. Rancangan Fungsi <i>Sensor Fusion</i> | 38 |



| | | |
|----------------|-----------------------------------------------------------------------------------|----|
| 4.5.3. | Rancangan Fungsi Kendali LQR | 39 |
| 4.5.4. | Rancangan Fungsi Interupsi <i>Motion</i> | 40 |
| 4.6. | Pengujian Sistem | 40 |
| 4.6.1. | Pengujian <i>sensor fusion</i> | 41 |
| 4.6.2. | Pengujian keseimbangan berdiri robot pada sudut putar <i>pitch</i> | 41 |
| 4.6.3. | Pengujian keseimbangan berdiri robot pada sudut putar <i>roll</i> | 41 |
| BAB V | IMPLEMENTASI..... | 43 |
| 5.1. | Implementasi Perangkat Keras | 43 |
| 5.2. | Pengujian <i>Sensor Fusion</i> | 43 |
| 5.3. | Pengujian keseimbangan berdiri robot humanoid pada sudut <i>pitch</i> | 45 |
| 5.4. | Pengujian keseimbangan berdiri robot humanoid pada sudut <i>roll</i> | 47 |
| BAB VI | HASIL DAN PEMBAHASAN | 51 |
| 6.1. | Hasil Pengujian <i>Sensor Fusion</i> | 51 |
| 6.2. | Hasil Pengujian Keseimbangan Berdiri Robot Humanoid Pada Sudut <i>Pitch</i> | 57 |
| 6.3. | Hasil Pengujian Keseimbangan Berdiri Robot Humanoid Pada Sudut <i>Roll</i> | 61 |
| BAB VII | PENUTUP..... | 66 |
| 7.1. | Kesimpulan..... | 66 |
| 7.2. | Saran..... | 66 |
| DAFTAR PUSTAKA | | 67 |
| LAMPIRAN..... | | 69 |

DAFTAR GAMBAR

| | |
|-----------------------------------------------------------------------------------------------------------------------------------------|----|
| Gambar 3.1 Sistem Koordinat pada robot Humanoid (Kajita, 2014) (a) posisi dari r dan (b) posisi dari r' | 9 |
| Gambar 3.2 (a) Koordinat lokal lengan atas Σa (b) koordinat lokal lengan bawah Σb (Kajita, 2014) | 11 |
| Gambar 3.3 <i>Roll, pitch</i> , dan <i>yaw</i> pada sebuah segitiga (Kajita, 2014) | 14 |
| Gambar 3.4 Posisi, sikap, dan rotasi dari sebuah link (Kajita, 2014) | 14 |
| Gambar 3.5 Posisi dari dua <i>link</i> terhadap <i>world coordinate</i> (Kajita, 2014) | 15 |
| Gambar 3.6 <i>Inverse kinematic</i> dari kaki robot humanoid (Kajita, 2014) | 16 |
| Gambar 3.7 Model pendulum terbalik dua dimensi (Jazar, 2010)..... | 18 |
| Gambar 4.1 Model pendulum terbalik dua dimensi..... | 25 |
| Gambar 4.2 Momen inersia bangun ruang balok (Braun, 2012) | 27 |
| Gambar 4.3 Sistem keseimbangan berdiri robot humanoid pada sudut putar <i>pitch</i> | 28 |
| Gambar 4.4 Sistem keseimbangan berdiri robot humanoid pada sudut putar <i>roll</i> | 29 |
| Gambar 4.5 Sistem keseimbangan berdiri robot humanoid pada sudut putar <i>roll</i> dengan selisih nilai Z pada kedua kaki robot | 31 |
| Gambar 4.6 <i>Inverse Kinematic</i> robot humanoid dalam satu sumbu | 32 |
| Gambar 4.7 Diagram block kendali sistem keseimbangan berdiri robot humanoid pada permukaan bidang miring..... | 34 |
| Gambar 4.8 Arsitektur sistem | 35 |
| Gambar 4.9 Program utama | 37 |
| Gambar 4.10 Fungsi <i>setup</i> | 38 |
| Gambar 4.11 <i>Sensor fusion</i> | 38 |
| Gambar 4.12 Fungsi kendali LQR | 40 |
| Gambar 4.13 Fungsi interupsi <i>motion</i> | 40 |
| Gambar 5.1 Mekanik robot humanoid | 43 |
| Gambar 5.2 Kode program <i>kalman filter</i> | 44 |
| Gambar 5.3 Kode program <i>complementary filter</i> | 45 |
| Gambar 5.4 Kode program LQR..... | 45 |
| Gambar 5.5 Kode program perhitungan nilai masukkan proses..... | 46 |
| Gambar 5.6 Kode program persamaan hubungan torsi dengan sudut dan penggerak aktuator pada sumbu putar <i>pitch</i> | 46 |
| Gambar 5.7 Kode program persamaan hubungan torsi dengan sudut dan penggerak aktuator pada sumbu putar <i>roll</i> | 47 |
| Gambar 5.8 Kode program kinematika robot | 48 |
| Gambar 5.9 Kinematic simulator | 49 |
| Gambar 5.10 Kode program persamaan <i>inverse kinematic</i> pada simulator..... | 50 |
| Gambar 6.1 Kalibrator pengujian <i>sensor fusion</i> pada sudut <i>pitch</i> | 51 |
| Gambar 6.2 <i>Sensor fusion</i> dengan <i>complementary filter</i> pada sudut <i>pitch</i> | 51 |
| Gambar 6.3 <i>Sensor fusion</i> dengan <i>kalman filter</i> pada sudut <i>pitch</i> | 52 |
| Gambar 6.4 Kalibrator pengujian <i>sensor fusion</i> pada sudut <i>roll</i> | 54 |
| Gambar 6.5 <i>Sensor fusion</i> dengan <i>complementary filter</i> pada sudut <i>roll</i> | 55 |
| Gambar 6.6 <i>Sensor fusion</i> dengan <i>kalman filter</i> pada sudut <i>roll</i> | 55 |



| | |
|-----------------------------------------------------------------------------------------------------------------------------|----|
| Gambar 6.7 Grafik hasil penalaan nilai Q pada sistem kendali robot untuk sudut putar <i>pitch</i> | 57 |
| Gambar 6.8 Hasil <i>steady-state-error</i> dengan beberapa variasi sudut kemiringan permukaan pada sudut <i>pitch</i> | 59 |
| Gambar 6.9 Grafik hasil penalaan nilai Q pada sistem kendali robot untuk sudut putar <i>roll</i> | 61 |
| Gambar 6.10 <i>Support Polygon</i> dengan semua bagian telapak kaki menapak di atas permukaan..... | 62 |
| Gambar 6.11 <i>Support Polygon</i> dengan sebagian telapak kaki menapak di atas permukaan..... | 62 |
| Gambar 6.12 Hasil <i>steady-state-error</i> dengan beberapa variasi sudut kemiringan permukaan pada sudut <i>roll</i> | 64 |



DAFTAR TABEL

| | |
|----------------------------------------------------------------------------------------|----|
| Tabel 2.1 Tinjauan Pustaka | 7 |
| Tabel 4.1 Pengujian sistem | 42 |
| Tabel 6.1 Rata-rata nilai dan standar deviasi hasil pengujian sudut <i>pitch</i> | 52 |
| Tabel 6.2 Rata-rata nilai dan standar deviasi hasil pengujian sudut <i>roll</i> | 53 |
| Tabel 6.3 Hasil <i>steady-state error</i> pada sudut <i>pitch</i> | 60 |
| Tabel 6.4 Uji coba kecepatan perubahan kemiringan pada sudut <i>pitch</i> | 60 |
| Tabel 6.5 Hasil <i>steady-state error</i> pada sudut <i>roll</i> | 63 |
| Tabel 6.6 Uji coba kecepatan perubahan kemiringan pada sudut <i>roll</i> | 65 |

INTISARI

SISTEM KENDALI KESEIMBANGAN BERDIRI ROBOT HUMANOID PADA PERMUKAAN BIDANG MIRING

Oleh

Dimas Fajrian Nugroho

13/347550/PA/15300

Pada saat berdiri robot humanoid harus dapat mempertahankan keadaannya agar tidak terjatuh. Hal ini disebabkan tingkat kemiringan permukaan yang berbeda-beda dapat mempengaruhi keadaan robot saat berdiri. Oleh karena itu dibutuhkan suatu sistem kendali yang dapat mencegah robot humanoid terjatuh.

Robot humanoid menggunakan sistem MIMO. *Linear Quadratic Regulator* atau LQR merupakan kendali yang dapat diimplementasikan pada sistem MIMO. Oleh karena itu LQR merupakan salah satu sistem kendali yang dapat digunakan untuk mempertahankan keadaan robot agar dapat berdiri dengan stabil diatas permukaan bidang miring.

Nilai *gain K* didapatkan dengan simulasi MATLAB menggunakan metode LQR dengan memanfaatkan model sistem yang dinamis. Sistem yang dinamis tersebut didapatkan dengan menjabarkan persamaan dari pemodelan pendulum terbalik dua dimensi menggunakan pendekatan Newton-Euler.

Berdasarkan penelitian yang telah dilakukan, robot humanoid dapat mempertahankan keseimbangannya pada kemiringan sudut *pitch* dengan nilai *risetime* 0,53 detik, *settling time* 1,12 detik, *steady-state error* $0,38^0$, *overshoot* sebesar $0,66^0$. Robot dapat mempertahankan keseimbangannya pada kemiringan sudut *roll* dengan nilai *risetime* 0,84 detik, *settling time* 1,72 detik, *steady-state error* $0,26^0$, *overshoot* sebesar $0,39^0$.

Robot dapat mempertahankan keseimbangannya pada kemiringan sudut *roll* dengan kecepatan perubahan kemiringan 0,12 rad/s dan 0,07 rad/s pada kemiringan sudut *pitch* dengan maksimal kemiringan sebesar 40^0 pada kedua sudut rotasi.

Kata kunci: Humanoid, LQR, Pendulum

ABSTRACT

HUMANOID ROBOT STANDING BALANCE CONTROL SYSTEM ON THE INCLINED SURFACE

By

Dimas Fajrian Nugroho

13/347550/PA/15300

At the time when the humanoid robot stand, it must be in order for the situation so as not to fall. This brings different levels of surface tilt to bring up the robot state when standing. Therefore we need a control system that can save the humanoid robot from falling down.

Humanoid robot uses MIMO system. From many control systems, LQR is a control that can be implemented on a MIMO system. Therefore Linear Quadratic Regulator or LQR is one control system that can be used to maintain the robot state in order to stand on the inclined surface.

The gain value of K is obtained by simulation on MATLAB using LQR method by utilizing dynamic system model. The dynamic system is obtained by describing the equations of two-dimensional inverted pendulum modeling using the Newton-Euler approach.

Based on the research that has been done, humanoid robot can maintain its balance at pitch angle with 0.53 seconds risetime, settling time 1.12 seconds, steady-state error 0.38, overshoot of 0.66. In addition, the robot can maintain its balance at a roll angle with a risetime of 0.84 seconds, settling time 1.72 seconds, steady-state error 0.26, overshoot of 0.36.

The robot can maintain its balance at a roll angle with an angular velocity up to 0.12 rad / s and up to 0.07 rad / s at the pitch angle with a maximum slope of 40° at both angles of rotation.

Tags: LQR, Humanoid, Pendulum

BAB I

PENDAHULUAN

1.1. Latar Belakang

Robot humanoid adalah robot yang bentuknya menyerupai manusia. Robot humanoid banyak diceritakan dalam cerita dengan tema fiksi sains, sehingga robot humanoid dapat dikatakan sebagai bentuk yang umum dari robot bagi kebanyakan orang (Kajita et al., 2001). Namun, robot sendiri tidak selalu dalam bentuk humanoid yang dapat melakukan segala pekerjaan dalam dunia nyata karena benda-benda terbang buatan seperti pesawat maupun *drone* tidak berbentuk seperti burung atau semacamnya.

Selama bertahun-tahun lingkungan yang telah dibangun sebagian besar ditujukan untuk manusia. Mulai dari tangga, koridor, pintu, *handrail* tangga, dan berbagai desain bangunan yang lainnya dibuat untuk manusia. Dari hal tersebut dapat disimpulkan bahwa membuat robot yang sesuai dengan lingkungan saat ini merupakan pilihan yang lebih baik daripada membuat lingkungan yang disesuaikan untuk robot (Kajita et al., 2001). Oleh karena itu, dapat disimpulkan bahwa robot dengan bentuk menyerupai manusia adalah robot yang memiliki pergerakan dan bentuk yang paling sesuai untuk dikembangkan. Diharapkan dengan mengembangkan robot humanoid dapat mewujudkan impian untuk membuat robot yang dapat melakukan pekerjaan-pekerjaan yang dapat membantu manusia dalam kehidupan sehari-hari.

Tugas-tugas yang dapat dilakukan oleh robot humanoid terdiri dari berbagai hal yang dapat dilakukan oleh manusia. Dari tugas-tugas tersebut, pergerakan robot humanoid menjadi hal yang paling mendasar dan paling penting untuk dikembangkan. Dari sekian banyak pergerakan dari robot humanoid, mengoptimalkan keseimbangan pada robot humanoid merupakan hal yang penting dan mendasar. Keseimbangan berdiri dapat dikatakan sebagai hal yang paling mendasar dan penting untuk dipahami untuk membuat suatu sistem pada robot humanoid dalam melakukan tugas pada lingkungan yang kompleks (Deng-Peng dan Xu, 2011).



Pada keseimbangan robot humanoid, perilaku kaki adalah hal yang penting untuk diperhatikan karena pergerakan kaki tersebut menentukan keseimbangan dari robot humanoid baik ketika berjalan, berdiri, berlari dan lain-lain. Perilaku kaki tidak dapat dikendalikan secara langsung, tetapi secara tidak langsung. Hal ini dilakukan dengan cara memastikan dinamika yang sesuai pada mekanisme di atas kaki.

Terdapat tiga asumsi yang dapat digunakan untuk memodelkan sistem dinamis dari robot humanoid yang terdiri dari banyak derajat kebebasan, bagian-bagian kinematika, dan komponen-komponen lainnya. Pertama, semua massa bagian-bagian tubuh robot dipusatkan pada satu titik yang disebut *Center of Mass* (CoM). Kedua, robot diasumsikan memiliki kaki tanpa massa, yang ujungnya menyentuh tanah pada sebuah sendi berputar. Bagian yang ketiga adalah mempertimbangkan gerakan maju atau mundur dan gerakan naik atau turun dari robot, dengan mengabaikan gerak pada bidang koronal. Bidang koronal merupakan bidang vertikal yang melalui tubuh dan letaknya tegak lurus terhadap bidang median atau sagital yang membagi tubuh menjadi bagian depan (frontal) dan belakang (dorsal). Gerak robot dilihat dengan sudut pandang pada bidang sagital di mana bidang sagital adalah bidang yang sejajar dengan bidang median yang membagi tepat bodi menjadi dua bagian kanan dan kiri. Dengan asumsi tersebut maka robot humanoid dapat di modelkan sebagai pendulum terbalik dimensi 2 (Kajita et al., 2003).

Pada saat berdiri robot humanoid harus dapat mempertahankan keadaannya agar tidak terjatuh. Hal ini disebabkan tingkat kemiringan permukaan yang berbeda-beda dapat mempengaruhi keadaan robot saat berdiri, walaupun terdapat gaya gesek yang cukup untuk menahan robot agar tidak tergelincir. Oleh karena itu dibutuhkan suatu sistem kendali yang dapat mencegah robot humanoid terjatuh.

Robot humanoid menggunakan sistem *Multiple Input Multiple Output* (MIMO). Dari banyak sistem kendali, LQR merupakan kendali yang dapat diimplementasikan pada sistem MIMO. Oleh karena itu *Linear Quadratic Regulator* atau LQR merupakan salah satu sistem kendali yang dapat digunakan untuk mempertahankan keadaan robot agar dapat berdiri dengan stabil diatas

permukaan bidang miring. LQR merupakan kendali yang digunakan untuk mencari *gain K* pada pemodelan *state space*. Nilai *gain K* dihasilkan menggunakan persamaan Riccati dengan melakukan penalaan pada matriks *Q* dan *R* sehingga dihasilkan nilai masukan proses yang sesuai dengan sistem robot humanoid yang dinamis.

1.2. Rumusan Masalah

Dari latar belakang yang telah dijabarkan di atas, maka yang menjadi rumusan masalah pada penelitian ini adalah robot humanoid dapat terjatuh atau tidak bisa mengendalikan dirinya saat berdiri pada permukaan dengan tingkat kemiringan yang berbeda-beda.

1.3. Batasan Masalah

Penelitian ini memiliki beberapa batasan masalah yaitu:

1. Penelitian difokuskan untuk menangani keseimbangan berdiri robot humanoid pada bidang miring
2. Robot yang digunakan menggunakan platform BIOLOID yang dimodifikasi dengan jumlah *degree of freedom* yang sama dengan robot tersebut.
3. Penelitian dilakukan pada permukaan lantai yang tidak licin
4. Kemiringan diberlakukan pada bidang koronal dan sagital untuk menguji keseimbangan *roll* dan *pitch* pada robot.

1.4. Tujuan Penelitian

Tujuan dari penelitian ini adalah merancang dan mengimplementasikan suatu sistem keseimbangan pada robot humanoid agar tidak terjatuh saat berdiri diatas permukaan lantai yang miring menggunakan metode kendali LQR.

1.5. Manfaat Penelitian

Manfaat dari penelitian ini adalah agar robot humanoid dapat beroperasi dilingkungan dengan contour permukaan yang berbeda-beda.

1.6. Metodologi Penelitian

Metodologi penelitian beserta alur pelaksanaan yang digunakan adalah sebagai berikut:

1. Menentukan topik penelitian dengan cara konsultasi dengan dosen pembimbing untuk mengidentifikasi masalah yang ada saat ini yang kemudian dicari solusinya.
2. Permasalahan yang telah diidentifikasi adalah dasar dari pembuatan sistem keseimbangan berdiri robot humanoid pada permukaan bidang miring.
3. Pengumpulan data dilakukan dengan melakukan studi literatur untuk mempelajari tentang robot humanoid, konsep dari *Inverted pendulum*, dan kendali *Linear Quadratic Regulator (LQR)*. Selain itu, pengkajian dan pembelajaran lebih lanjut juga dilakukan dengan konsultasi dengan dosen pembimbing mengenai rancangan sistem yang dapat diterapkan.
4. Membuat rancangan sistem terdiri dari beberapa bagian seperti berikut:
 - a. Perancangan mekanika robot. Perancangan dilakukan dengan mengacu pada sistem kinematika beserta kerangka dari robot BIOLOID yang telah dimodifikasi.
 - b. Perancangan perangkat keras. Perancangan sistem elektronik melibatkan beberapa *hardware* seperti OpenCM9.04, sensor *Inertia Measurement Unit (IMU)*, dan aktuator servo *Dynamixel*.
 - c. Perancangan perangkat lunak. Perancangan perangkat lunak dilakukan dengan menggunakan Robotis IDE dan Robot Operating System.
 - d. Penerapan sistem yang telah dirancang adalah dengan cara penalaan yaitu uji coba dengan basis data dari pengalaman-pengalaman uji data sebelumnya. Hal ini meliputi proses *proof of concept*.
 - e. Analisis dan pengujian sistem yang telah dibuat yaitu dengan meletakkan robot pada permukaan bidang miring yang kemiringannya di ukur dengan sensor kemiringan. Berdasarkan proses tersebut dapat ditarik kesimpulan dari hasil penelitian skripsi yang telah dilakukan.



1.7. Sistematika Penulisan

Sistematika penulisan dalam penyusunan usulan penelitian S1 ini adalah sebagai berikut:

BAB I: PENDAHULUAN

Bab ini berisi tentang latar belakang penelitian, rumusan masalah beserta batasan masalahnya, tujuan dan manfaat dari penelitian, metodologi penelitian, serta sistematika dari penulisan penelitian.

BAB II: TINJAUAN PUSTAKA

Bab ini berisi tentang penelitian-penelitian terkait yang sudah pernah dilakukan.

BAB III: LANDASAN TEORI

Bab ini berisi tentang teori-teori dasar yang mendukung penelitian yang akan dilakukan. Adapun teori yang mendukung penelitian ini seperti definisi dan gambaran tentang robot humanoid, kinematika robot, konsep dari pendulum terbalik, dan kendali *Linear Quadratic Regulator*.

BAB IV: ANALISIS DAN PERANCANGAN SISTEM

Bab ini berisi tentang rancangan dari sistem yang akan digunakan dalam penelitian meliputi perangkat keras sistem, pemodelan sistem, sistem kendali, perangkat lunak sistem dan pengujian sistem.

BAB V: IMPLEMENTASI

Bab ini berisi tentang penerapan konsep dari sistem meliputi penerapan *sensor fusion*, penerapan sistem kendali, penerapan sistem keseimbangan robot dan Penerapan konsep kinematika sistem.

BAB VI: HASIL DAN PEMBAHASAN

Bab ini menjelaskan pembahasan dari hasil pengujian sistem keseimbangan robot humanoid menggunakan metode LQR.

BAB VII: PENUTUP

Bab ini berisi kesimpulan dan saran untuk pengembangan sistem kendali selanjutnya.

BAB II

TINJAUAN PUSTAKA

Penelitian mengenai keseimbangan pada robot humanoid sebelumnya pernah dilakukan. Penelitian yang dilakukan adalah tentang sistem kendali step motion robot humanoid berbasis BIOLOID dalam mengikuti suatu warna. Sensor giro ditanamkan untuk menguji kestabilan posisi siaga. Pengujian giro hanya dilakukan dengan memiringkan robot terhadap sumbu y (*pitch*) karena perubahan kecepatan sudut pada sumbu y yang paling berpengaruh dalam keseimbangan robot statis posisi siaga. Sedangkan perubahan kecepatan sudut pada sumbu x tidak mempengaruhi keseimbangan posisi siaga secara signifikan (Wicaksono et al., 2014).

Selain itu, telah dilakukan penelitian tentang sistem kendali keseimbangan statis pada robot humanoid menggunakan PID. Pengontrol Proporsional - Integral - Derivatif menghitung nilai eror sebagai perbedaan antara variabel proses terukur dan nilai yang diinginkan. Pengontrol akan mengurangi eror dengan mengatur masukan kontrol proses. Subsistem yang telah dibuat yaitu sistem kontrol keseimbangan robot humanoid Joko Klana berbasis Pengontrol PID menggunakan akselerometer ADXL202 (Iswara dan Putra, 2012).

Dalam mengatasi permasalahan keseimbangan pada robot humanoid, maka dibutuhkan suatu kendali yang dapat mengoptimalkan sistem keseimbangan robot. Deng-peng dan Xu (2011) telah melakukan penelitian tentang beragam strategi kendali saat robot berdiri dengan menggunakan pengendali LQR pada robot humanoid yang dimodelkan sebagai pendulum terbalik dengan jumlah *link* yang bervariasi.

Dalam memodelkan dinamika robot secara 3 dimensi, Kajita dkk. (2001) melakukan penelitian tentang dinamika robot humanoid menggunakan konsep 3D *Linear Inverted Pendulum Model*. Dalam penelitiannya, 3D LIPM diimplementasikan pada sistem berjalan robot humanoid.



Penelitian tentang kestabilan berdiri robot humanoid dengan menggunakan satu kaki telah dilakukan dengan menganalisa sistem kestabilan untuk mengetahui posisi-posisi bagian tubuh robot selama bergerak ketika diberi gangguan (Akhtaruzzaman dan Shafie, 2011).

Sebelumnya telah dilakukan penelitian tentang algoritma yang memanfaatkan FSR untuk mendeteksi keadaan robot pada saat terjatuh (Shi et al., 2016). Sebagai *feedback controller*, *Linear Quadratic Regulator* (LQR) digunakan untuk melacak secara efektif ZMP yang diinginkan menurut jejak kaki dari robot humanoid (Hong, 2008).

Penelitian tentang kestabilan robot humanoid saat berdiri dengan konsep LQR pun juga dilakukan dengan kendali non-linear yang hanya menggunakan sendi pinggang. Dorongan atau pemberlakuan gaya secara eksternal di demonstrasikan melalui simulator Webots dan robot humanoid NUSBIP-III ASLAN (Ahmed et al., 2013).

Dari penelitian-penelitian tersebut akan dilakukan penelitian mengenai sistem kendali yang digunakan untuk robot humanoid agar dapat berdiri dengan stabil pada permukaan bidang miring.

Hubungan antara penelitian ini dengan penelitian sebelumnya dapat dilihat pada Tabel 2.1.

Tabel 2.1 Tinjauan Pustaka

| No | Peneliti | Judul Penelitian | Keterangan |
|----|------------------|-------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------|
| 1. | Wicaksono (2014) | Sistem Kendali Stepmotion Robot Humanoid 20 Derajad Kebebasan dalam Mengikuti Suatu Warna | Merancang sistem kendali keseimbangan robot BIOLOID. |
| 2. | Iswara (2012) | Sistem Kontrol Keseimbangan Statis Robot Humanoid Joko Klana Berbasis Pengontrol PID | Merancang sistem kendali keseimbangan berdiri pada robot humanoid dengan menggunakan kendali PID. |



Tabel 2.1 (lanjutan)

| No. | Peneliti | Judul Penelitian | Keterangan |
|-----|-----------------------------|-------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------|
| 3. | Deng-Peng (2011) | <i>Multiple Balance Strategies for Humanoid Standing Control</i> | Merancang sistem kestabilan berdiri robot humanoid. |
| 4. | Kajita (2001) | <i>The 3D linear inverted pendulum mode: a simple modeling for a biped walking pattern generation</i> | Merancang pemodelan sistem berjalan robot humanoid. |
| 5. | Akhtaruzza man (2011) | <i>Geometrical Analysis on BIOLOID Humanoid System Standing on Single Leg</i> | Melakukan analisa sistem robot humanoid BIOLOID yang berdiri menggunakan satu kaki |
| 6. | Hong (2008) | <i>Walking Pattern Generation for Humanoid Robots with LQR and Feedforward Control Method</i> | Merancang metode pola berjalan pada robot humanoid menggunakan kendali LQR |
| 7. | Ahmed (2013) | <i>Standing Posture Modeling and Control for a Humanoid Robot</i> | Merancang sistem kestabilan berdiri pada robot humanoid dengan model pendulum terbalik dan menggunakan kendali LQR |
| 8. | Nugroho (2017) | <i>Sistem Keseimbangan Berdiri Robot Humanoid Faro pada Permukaan Bidang Miring</i> | Merancang dan mengimplementasikan sistem keseimbangan berdiri robot humanoid agar tidak terjatuh saat berdiri diatas permukaan lantai yang miring. |

BAB III

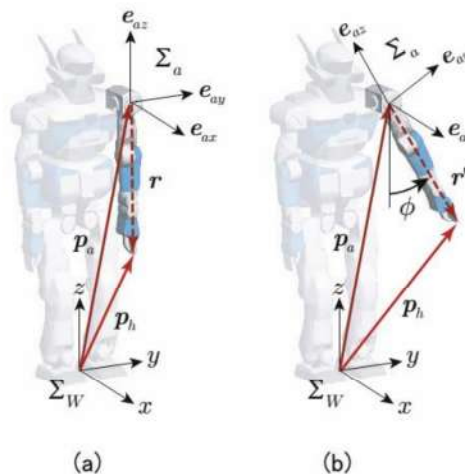
LANDASAN TEORI

3.1. Robot Humanoid

Robot humanoid merupakan suatu robot yang dibentuk menyerupai manusia. Secara umum robot humanoid terdiri dari kepala, badan, dua buah tangan dan dua buah kaki. Salah satu alasan penelitian tentang robot humanoid ini dilakukan adalah karena manusia memiliki bentuk yang lebih optimal dalam menyelesaikan permasalahan-permasalahan sehari-hari (Duindam dan Stramigioli, 2009).

Sendi-sendi pada robot humanoid terdiri dari aktuator-aktuator yang bergerak dalam satu sumbu yang dihubungkan oleh beberapa *link*. Sehingga membentuk sebuah robot humanoid.

Sistem transformasi koordinat digunakan untuk menentukan titik acuan utama dari sistem atau merupakan titik nol dari sistem koordinat yang berlaku pada robot (Kajita et al., 2014). Pada Gambar (3.1), Σ_W adalah titik acuan utama dari sistem yang disebut *World Coordinates*, dan Σ_a merupakan koordinat lokal atau titik yang bergerak secara relatif terhadap *link* yang terhubung.



Gambar 3.1 Sistem Koordinat pada robot Humanoid (Kajita, 2014) (a) posisi dari r dan (b) posisi dari r'



Titik absolut dari robot dapat di definisikan kedalam vektor 3 dimensi pada persamaan(3.1). Posisi dari suatu titik dapat di dapatkan dengan contoh menentukan posisi dari ujung tangan kiri robot. Berdasarkan pada Gambar 3.1 Sistem Koordinat pada robot Humanoid (Kajita, 2014) (a) posisi dari r dan (b) posisi dari r' , posisi dari ujung tangan kiri robot ditentukan dengan menggunakan persamaan (3.2) dengan r sebagai posisi dari tangan kiri robot terhadap koordinat lokal Σ_a , p_a posisi dari bahu kiri terhadap titik absolut Σ_w pada robot, dan p_h sebagai posisi ujung tangan kiri terhadap titik absolut robot.

$$p_h = \begin{bmatrix} p_{hx} \\ p_{hy} \\ p_{hz} \end{bmatrix} \quad (3.1)$$

$$p_h = p_a + r \quad (3.2)$$

Rotasi pada suatu joint seperti pada Gambar 3.1(a) menjadi Gambar 3.1(b), dapat dilakukan dengan menggunakan persamaan (3.3) yang diberlakukan pada vektor unit e_{ax}, e_{ay}, e_{az} yang berada pada koordinat lokal Σ_a .

$$e_{ax} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \quad e_{ay} = \begin{bmatrix} 0 \\ \cos\phi \\ \sin\phi \end{bmatrix}, \quad e_{az} = \begin{bmatrix} 0 \\ -\sin\phi \\ \cos\phi \end{bmatrix} \quad (3.3)$$

Penentuan r' dilakukan dengan cara vektor unit dari Σ_a digunakan kedalam matriks 3x3 pada Persamaan (3.4) untuk mendapatkan bentuk persamaan yang lebih sederhana, sehingga didapatkan Persamaan (3.5) untuk menentukan posisi dari r' .

$$R_a \equiv [e_{ax} \quad e_{ay} \quad e_{az}] \quad (3.4)$$

$$r' = R_a r \quad (3.5)$$

Berdasarkan Gambar 3.2 ${}^a p_h$ merupakan posisi dari ujung lengan yang berada dalam koordinat lokal Σ_a . Pada Gambar 3.2, ${}^a p_h = r$. Persamaan (3.6) menentukan posisi ujung lengan, dan dapat diubah kedalam bentuk Persamaan(3.7).

$$p_h = p_a + R_a {}^a p_h \quad (3.6)$$

$$\begin{bmatrix} p_h \\ 1 \end{bmatrix} = \begin{bmatrix} R_a & p_a \\ 0 & 1 \end{bmatrix} \begin{bmatrix} {}^a p_h \\ 1 \end{bmatrix} \quad (3.7)$$

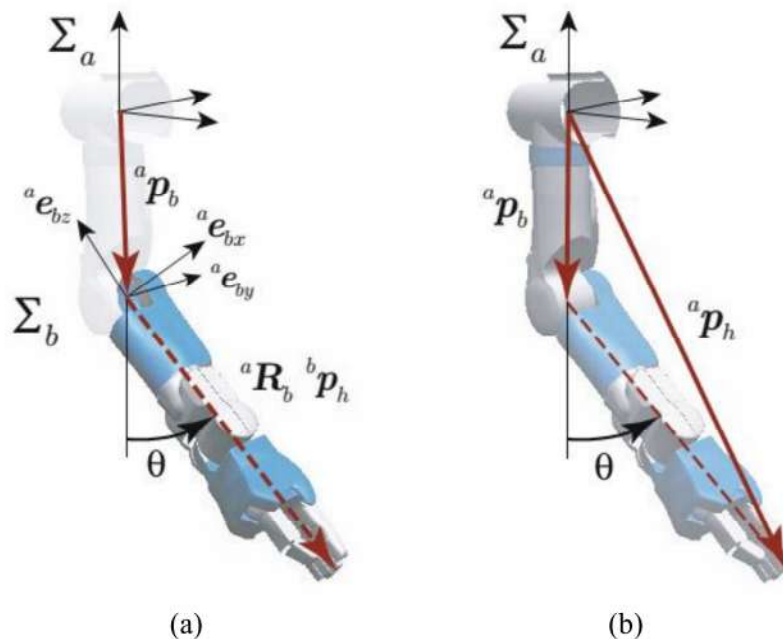
Seperti yang terdapat pada persamaan (3.7), 0 0 0 dan 1 pada matriks tersebut di tambahkan untuk menyesuaikan dimensi matriks agar dapat dioperasikan. Matriks 4x4 pada Persamaan (3.7), di ubah kedalam Persamaan (3.8),

$$T_a = \begin{bmatrix} R_a & p_a \\ 0 & 1 \end{bmatrix} \quad (3.8)$$

sehingga didapatkan persamaan (3.9) yang merupakan matriks dari transformasi homogen dengan T_a yang mengkonversikan titik yang dalam koordinat lokal kedalam *world coordinate*.

$$\begin{bmatrix} p \\ 1 \end{bmatrix} = T_a \begin{bmatrix} {}^a p \\ 1 \end{bmatrix} \quad (3.9)$$

Variabel ${}^a p$ dapat menjadi titik manapun pada lengan, sehingga bentuk lengan ditentukan dari sekumpulan komponen pembentuknya. Disamping itu, T_a menunjukkan konfigurasi dari lengan.



Gambar 3.2 (a) Koordinat lokal lengan atas Σ_a (b) koordinat lokal lengan bawah Σ_b (Kajita, 2014)



Hubungan antar koordinat lokal, dapat ditentukan dengan contoh pada Σ_b yang terpaut pada Σ_a dengan ketentuan sumbu-sumbu yang serupa. Σ_b memiliki unit vektor e_{bx} , e_{by} , dan e_{bz} seperti yang dimiliki Σ_a sebelumnya. Maka dari itu, persamaan yang digunakan untuk melakukan rotasi pada *joint* pada Σ_b ditunjukkan dalam persamaan (3.10).

$$e_{ax} = \begin{bmatrix} \cos\theta \\ 0 \\ \sin\theta \end{bmatrix}, \quad e_{ay} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \quad e_{az} = \begin{bmatrix} -\sin\theta \\ 0 \\ \cos\theta \end{bmatrix} \quad (3.10)$$

Ketiga vektor unit tersebut digabung menjadi aR_b sehingga didapatkan persamaan (3.11) dan konversi dari ${}^b p_h$ pada Σ_b , menjadi ${}^a p_h$ pada Σ_a ditunjukkan pada persamaan (3.12).

$${}^aR_b \equiv [e_{bx} \quad e_{by} \quad e_{bz}] \quad (3.11)$$

$$\begin{bmatrix} {}^a p_h \\ 1 \end{bmatrix} = {}^aT_b \begin{bmatrix} {}^b p_h \\ 1 \end{bmatrix} \quad (3.12)$$

Bentuk transformasi homogen aT_b terdapat pada persamaan (3.13).

$${}^aT_b = \begin{bmatrix} {}^aR_b & {}^a p_b \\ 0 \ 0 \ 0 & 1 \end{bmatrix} \quad (3.13)$$

Berdasarkan persamaan (3.12) dan (3.7), maka didapatkan persamaan yang mengubah titik tersebut dari koordinat lokal Σ_b kedalam *world coordinate*.

$$\begin{bmatrix} p_h \\ 1 \end{bmatrix} = T_a {}^aT_b \begin{bmatrix} {}^b p_h \\ 1 \end{bmatrix} \quad (3.14)$$

$T_a {}^aT_b$ pada persamaan (3.14), didefinisikan sebagai T_b sehingga didapatkan persamaan (3.15).

$$T_b = T_a {}^aT_b \quad (3.15)$$

Transformasi homogen T_b merupakan Σ_b dalam *world coordinate* yang juga menentukan posisi atau kedudukan dari lengan bawah robot. T_a adalah nilai dari bahu yang sudah berputar, aT_b berubah sesuai dengan perputaran bahu sehingga T_b menunjukkan perputaran pada titik Σ_a dan Σ_b .

Setelah itu persamaan yang dijelaskan diatas digeneralisasikan. Ketika sebuah sistem terhubung dari Σ_1 hingga Σ_N maka didapatkan (3.16) yang menggambarkan dua koordinat lokal Σ_i dan Σ_{i+1} yang saling terhubung.

$${}^i T_{i+1} \quad (3.16)$$

Berdasarkan persamaan (3.16) maka didapatkan persamaan (3.17) yang menunjukkan transformasi homogen yang menggambarkan posisi dan perilaku dari sendi ke-N. Metode untuk melakukan perkalian antar transformasi homogen demi menghitung koordinat ini disebut *Chain Rule*.

$$T_N = T_1 {}^1 T_2 {}^2 T_3 \dots {}^{N-1} T_N \quad (3.17)$$

3.2. Rotation Matriks pada robot humanoid

Prinsip perputaran terhadap suatu sumbu *roll*, *pitch*, dan *yaw* digunakan untuk menentukan perputaran dari sendi-sendi maupun bagian-bagian tubuh pada robot. Kali ini, sudut *roll* dilambangkan sebagai ϕ , sudut *pitch* sebagai θ , dan sudut *yaw* sebagai ψ . Persamaan (3.18), (3.19), dan (3.20) adalah matriks rotasi yang digunakan untuk menentukan perputaran dalam masing-masing sumbu *roll*, *pitch*, dan *yaw*.

$$R_x(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{bmatrix} \quad (3.18)$$

$$R_y(\theta) = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix} \quad (3.19)$$

$$R_z(\varphi) = \begin{bmatrix} \cos\varphi & -\sin\varphi & 0 \\ \sin\varphi & \cos\varphi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.20)$$

Roll, *pitch*, dan *yaw* diberlakukan dengan menggunakan persamaan (3.20) yang di persingkat menjadi persamaan (3.21). Persamaan (3.21) dapat di definisikan menjadi persamaan (3.22).

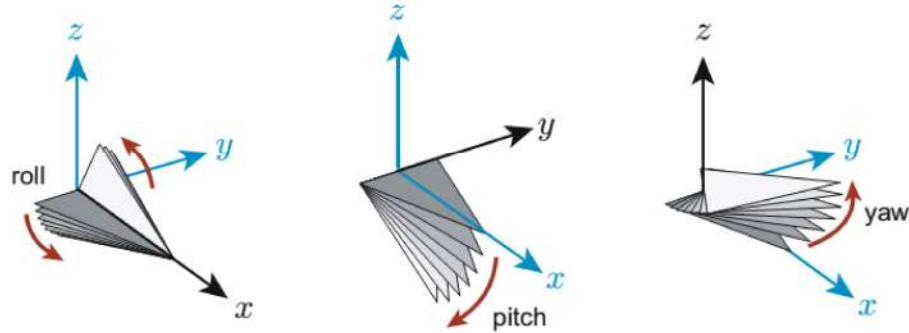
$$p' = R_x(\phi)R_y(\theta)R_z(\varphi)p \quad (3.21)$$

$$p' = R_{rpy}(\phi, \theta, \varphi)p \quad (3.22)$$

Matriks rotasi pada persamaan (3.23) yang dapat melakukan *roll*, *pitch*, dan *yaw* dalam 3 Dimensi didapatkan menggunakan ketentuan sudut Euler. Pada

Gambar 3.3 ditunjukkan *roll*, *pitch*, dan *yaw* yang diberlakukan pada sebuah segitiga.

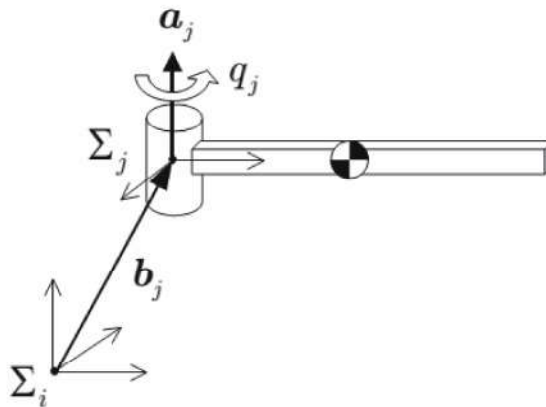
$$\begin{aligned}
 R_{rpy}(\phi, \theta, \varphi) &= R_x(\phi)R_y(\theta)R_z(\varphi) \\
 &\equiv \begin{bmatrix} C\phi C\theta & -S\phi C\theta + C\phi S\theta S\varphi & S\phi S\theta + C\phi S\theta C\varphi \\ S\phi C\theta & C\phi C\theta + S\phi S\theta S\varphi & -C\phi S\theta + S\phi S\theta C\varphi \\ -S\theta & C\theta S\varphi & C\theta C\varphi \end{bmatrix} \quad (3.23)
 \end{aligned}$$



Gambar 3.3 Roll, pitch, dan yaw pada sebuah segitiga (Kajita, 2014)

3.3. Forward Kinematic

Forward Kinematic digunakan untuk mendapatkan posisi *link* robot berdasarkan sudut dan struktur robot (Kajita et al., 2014). Perhitungan dapat dilakukan dengan menggunakan ketentuan transformasi homogen.



Gambar 3.4 Posisi, sikap, dan rotasi dari sebuah link (Kajita, 2014)

Dari Gambar 3.4, Σ_j merupakan koordinat lokal yang terletak pada sendi. \mathbf{a}_j merupakan vektor dari sendi yang berasal dari Σ_j , \mathbf{b}_j adalah vector yang menghubungkan dari Σ_i menuju Σ_j dan q_j merupakan sudut sendi yang berevolusi

terhadap vektor \mathbf{a}_j . Transformasi homogen didapatkan menggunakan persamaan (3.24).

$${}^i T_j = \begin{bmatrix} e^{a_j q_j} & \mathbf{b}_j \\ 0 & 1 \end{bmatrix} \quad (3.24)$$

Diasumsikan bahwa terdapat dua *link* seperti pada Gambar 3.5. Titik absolut dari $\mathbf{p}_i, \mathbf{R}_i$ adalah Σ_W sehingga transformasi homogen dari Σ_W ke Σ_i adalah sebagai berikut.

$$T_i = \begin{bmatrix} \mathbf{R}_i & \mathbf{p}_i \\ 0 & 1 \end{bmatrix} \quad (3.25)$$

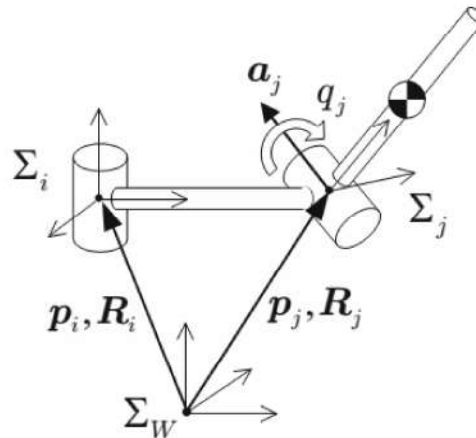
Transformasi homogen dari Σ_W ke Σ_j didapatkan dari persamaan

$$T_j = T_i {}^i T_j \quad (3.26)$$

\mathbf{p}_j yang merupakan posisi dari Σ_j dan \mathbf{R}_j sebagai *attitude* didapatkan dari persamaan (3.27) dan persamaan (3.28).

$$\mathbf{p}_j = \mathbf{p}_i + \mathbf{R}_i \mathbf{b}_j \quad (3.27)$$

$$\mathbf{R}_j = \mathbf{R}_i e^{a_j q_j} \quad (3.28)$$

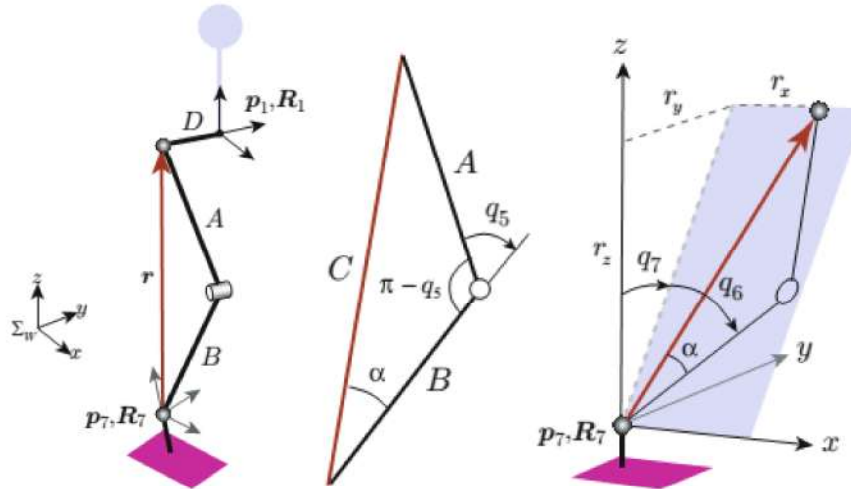


Gambar 3.5 Posisi dari dua *link* terhadap *world coordinate* (Kajita, 2014)

3.4. Inverse Kinematic

Jika dengan *forward kinematic* akan didapatkan posisi maupun *attitude* dari serangkaian *link* robot, maka yang dapat dilakukan dengan *inverse kinematic* adalah sebaliknya. *Inverse kinematic* digunakan untuk mendapatkan nilai sudut dari masing-masing sendi pada robot berdasarkan posisi dan panjang dari *link* robot

(Kajita, 2014). Gambar 3.6 menunjukkan penyelesaian *inverse kinematic* pada bagian kaki kanan secara analitis.



Gambar 3.6 Inverse kinematic dari kaki robot humanoid (Kajita, 2014)

Berdasarkan Gambar 3.6, posisi dari hip ditentukan oleh Persamaan (3.29)

$$p_2 = p_1 + R_1 \begin{bmatrix} 0 \\ D \\ 0 \end{bmatrix} \quad (3.29)$$

Nilai dari r ditentukan menggunakan Persamaan (3.30)

$$r = R_7^T + (p_2 - p_7) \equiv [r_x \quad r_y \quad r_z]^T \quad (3.30)$$

Jarak antara pinggang dengan pergelangan kaki didapatkan menggunakan Persamaan (3.31)

$$C = \sqrt{r_x^2 + r_y^2 + r_z^2} \quad (3.31)$$

Sudut dari q_5 didapatkan dengan persamaan (3.32)

$$q_5 = -\cos^{-1} \left(\frac{A^2 + B^2 - C^2}{2AB} \right) + \pi \quad (3.32)$$

Nilai dari sudut α pada Gambar 3.6 dapat ditemukan dengan persamaan (3.33)

$$\alpha = \sin^{-1} \left(\frac{A \sin(\pi - q_5)}{C} \right) \quad (3.33)$$

Nilai sudut q_7 didapatkan menggunakan persamaan (3.34) dan q_6 menggunakan persamaan (3.35).



$$q_7 = \text{atan2}(r_x, r_z) \quad (3.34)$$

$$q_6 = -\text{atan2} \left(r_x, \text{sign}(r_z) \sqrt{r_y^2 + r_z^2} \right) - \alpha \quad (3.35)$$

Dari persamaan-persamaan tersebut, $\mathbf{R}_7 =$

$\mathbf{R}_1 \mathbf{R}_z(q_2) \mathbf{R}_x(q_3) \mathbf{R}_y(q_4) \mathbf{R}_y(q_5 + q_6) \mathbf{R}_x(q_7)$, sehingga didapatkan persamaan

$$\mathbf{R}_z(q_2) \mathbf{R}_x(q_3) \mathbf{R}_y(q_4) = \mathbf{R}_1^T \mathbf{R}_7 \mathbf{R}_x(-q_7) \mathbf{R}_y(-q_5 - q_6) \quad (3.36)$$

, selanjutnya persamaan tersebut dijabarkan menjadi persamaan dengan $C_2 \equiv \cos q_2$, dan $s_2 \equiv \sin q_2$.

$$\begin{bmatrix} c_2 c_4 - s_2 s_3 s_4 & -s_2 c_3 & c_2 s_4 + s_2 s_3 s_4 \\ s_2 c_4 + c_2 s_3 s_4 & c_2 c_3 & s_2 s_4 - c_2 s_3 s_4 \\ -c_3 s_4 & s_3 & c_3 c_4 \end{bmatrix} = \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix} \quad (3.37)$$

Sudut-sudut sendi didapatkan dengan persamaan (3.38), (3.39), dan (3.40) (Kajita, 2014).

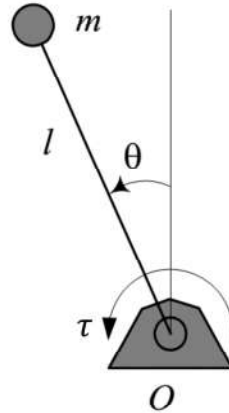
$$q_2 = \text{atan2}(-R_{12}, R_{22}) \quad (3.38)$$

$$q_3 = \text{atan2}(-R_{32}, -R_{12}s_2 + R_{22}c_2) \quad (3.39)$$

$$q_4 = \text{atan2}(-R_{31}, R_{33}) \quad (3.40)$$

3.5. Model Pendulum Terbalik Dua Dimensi

Terdapat tiga asumsi yang dapat digunakan untuk memodelkan sistem dinamis dari robot humanoid yang terdiri dari banyak derajat kebebasan, bagian-bagian kinematika, dan komponen-komponen lainnya. Pertama, semua massa bagian-bagian tubuh robot dipusatkan pada satu titik yang disebut *Center of Mass* (CoM). Kedua, robot diasumsikan memiliki kaki tanpa massa, yang ujungnya menyentuh tanah pada sebuah sendi berputar. Ketiga, kita hanya mempertimbangkan gerakan maju atau mundur dan gerakan naik atau turun dari robot, dengan mengabaikan gerak pada bidang koronal. Dengan kata lain, kita asumsikan gerak robot dibatasi pada bidang sagital. Berdasarkan asumsi tersebut maka robot humanoid dapat di modelkan sebagai pendulum terbalik dimensi 2 (Kajita et al., 2014).



Gambar 3.7 Model pendulum terbalik dua dimensi (Jazar, 2010)

Gambar 3.7 merupakan representasi dari model pendulum terbalik dimana Q sebagai torsi pendulum, θ adalah sudut simpangan pendulum, l panjang pendulum, dan m massa pendulum. Dari model tersebut didapatkan persamaan (3.41) yang merepresentasikan dinamika dari pendulum terbalik (Jazar, 2010).

$$ml^2\ddot{\theta} - mgl \sin \theta = \tau \quad (3.41)$$

3.6. Linear Quadratic Regulator (LQR)

LQR merupakan suatu kendali dengan dinamika sistem yang linier dan *cost function* yang kuadratik (Liberzon, 2012). *Cost function* merupakan simpangan *error* yang didapat oleh sistem. Keuntungan dari *quadratic optimal control* adalah, caranya yang sistematis dalam melakukan komputasi pada matriks *state feedback control gain* (Ogata, 2009).

Persamaan (3.42) menunjukkan index performansi dalam interval waktu awal t_0 hingga ∞

$$J = \int_{t_0}^{\infty} (\mathbf{x}^T \mathbf{Q} \mathbf{x} + \mathbf{u}^T \mathbf{R} \mathbf{u}) dt \quad (3.42)$$

dengan $\mathbf{Q} = \mathbf{Q}^T \geq 0$ dan $\mathbf{R} = \mathbf{R}^T > 0$. Keadaan dinamis dari permasalahan ini digambarkan pada persamaan *state space* dalam lingkup *linear-time-invariant*

$$\dot{\mathbf{x}} = \mathbf{A} \mathbf{x} + \mathbf{B} \mathbf{u} \quad \mathbf{A}, \mathbf{B} - \text{konstan} \quad \mathbf{x} \in \mathbf{R}^{n_x}, \mathbf{u} \in \mathbf{R}^{n_u} \quad (3.43)$$

dengan (\mathbf{A}, \mathbf{B}) *stabilizable* dan $(\mathbf{A}, \mathbf{Q}^{1/2})$ *detectable*, sehingga berdasarkan penyelesaian persamaan aljabar Riccati, maka didapatkan persamaan (3.44) dan



bentuk umpan balik pada persamaan (3.45) dengan $K \in R^{n_u \times n_s}$ merupakan matriks konstanta dari *gain* umpan balik dari LQR.

$$PA + A^T P - PBR^{-1}B^T P + Q = 0 \quad (3.44)$$

$$u = -R^{-1}B^T P x = -Kx \text{ dengan } K = R^{-1}B^T P \quad (3.45)$$

Dengan melakukan substitusi antara persamaan(3.43) dengan persamaan(3.45) maka sistem *close-loop* didefinisikan dalam persamaan(3.46).

$$\dot{x} = (A - BK)x = A_{cl}x \text{ dengan } A_{cl} = A - BK \quad (3.46)$$

LQR pada sistem *close-loop* yang dinamikanya digambarkan dalam matriks A_{cl} dapat digunakan untuk menjamin kestabilan dari sistem dengan *eigenvalue* yang terletak di sebelah kiri *complex plane* $\text{Re}(\lambda(A_{cl})) < 0$. Sistem di regulasikan menjadi nol ($x \rightarrow 0$ sebagai $t \rightarrow \infty$), sehingga $u \rightarrow 0$ sebagai $t \rightarrow \infty$. Jika u diturunkan, maka didapatkan persamaan(3.47), sehingga model sistem *close-loop* dapat disimulasikan pada persamaan (3.48) dengan $\dot{x} = -A_{cl}x$ dan $u = -Kx$.

$$\dot{u} = -Kx = -K(A - BK)x = -KA_{cl}x \quad (3.47)$$

$$y = \begin{bmatrix} x \\ u \\ \dot{u} \end{bmatrix} = \begin{bmatrix} I \\ -K \\ -KA_{cl} \end{bmatrix} x \quad (3.48)$$

Dalam penerapannya, pengaturan dari nilai K menjadi penting dan dapat mempengaruhi kestabilan dari sistem. Pada persamaan(3.45) nilai K akan bertambah besar jika P nilainya bertambah besar pula. Pada persamaan (3.44) nilai Q dan R akan mempengaruhi seberapa besar *gain*-nya

3.7. Kalman Filter

Kalman filter adalah suatu metode yang digunakan untuk melakukan estimasi terhadap data yang menghasilkan nilai yang fluktuatif. Kalman filter adalah estimator dari nilai data yang fluktuatif tersebut sehingga didapatkan nilai sebenarnya yang stabil dengan cara mengetahui nilai *error* pengukuran sensor, estimasi nilai sensor dan nilai masukkan sensor yang dihitung secara berulang-ulang sesuai dengan aliran data umpan balik yang diterima.

Kebanyakan sistem praktis memiliki beberapa tingkat nonlinearitas. Bergantung pada jenis proses dan lingkup prosesnya, beberapa proses dapat



dimodelkan secara linear dan kalman filter dapat digunakan untuk melakukan estimasi keadaan (Kandepu et al., 2008).

Dalam melakukan estimasi nilai sensor, maka diperlukan perhitungan untuk mencari nilai *gain*, nilai estimasi *error*, dan nilai estimasi sensor. Nilai-nilai tersebut dapat dihitung menggunakan persamaan (3.49), (3.50), dan (3.51).

$$KG = \frac{E_{EST}}{E_{EST} + E_{MEA}} \quad (3.49)$$

$$E_{EST_t} = \frac{E_{MEA} \cdot E_{EST_{t-1}}}{E_{MEA} + E_{EST_{t-1}}} = (1 - KG)(E_{EST_{t-1}}) \quad (3.50)$$

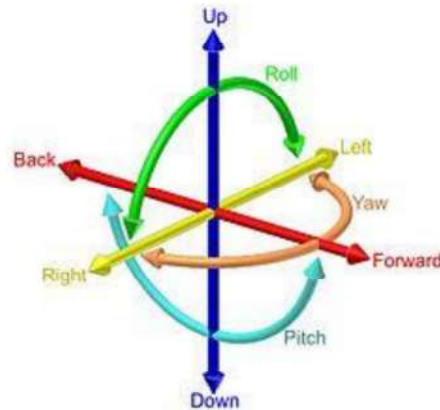
$$EST_t = EST_{t-1} + KG(MEA - EST_{t-1}) \quad (3.51)$$

Keterangan:

| | |
|-----------------|----------------------------------------|
| KG | = <i>gain</i> Kalman; |
| E_{EST} | = nilai estimasi kesalahan; |
| E_{MEA} | = nilai kesalahan pengukuran; |
| $E_{EST_{t-1}}$ | = nilai kesalahan estimasi sebelumnya; |
| E_{EST_t} | = nilai kesalahan estimasi; |
| MEA | = nilai terukur; |
| EST_{t-1} | = nilai estimasi sebelumnya; |

3.8. Inertial Measurement Unit (IMU)

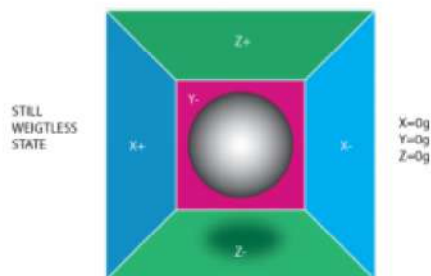
IMU adalah satu unit modul elektronik yang terdiri dari sensor akselerometer dan giroskop untuk mengukur percepatan sudut linier serta kecepatan sudut (Starlino, 2009). Kedua sensor tersebut memiliki tiga luaran analog yang mendiskripsikan tiga axis accelero dan tiga axis gyro yang disebut juga sebagai 6 DOF(Degree Of *Freedom*) seperti yang ditunjukkan pada Gambar 3.8.



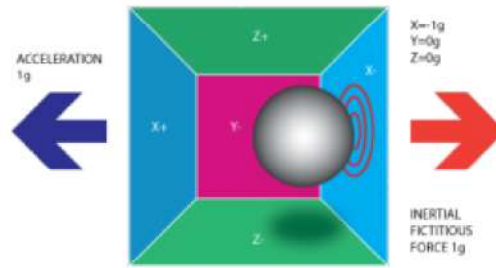
Gambar 3.8 Gambaran 6-DOF (Starlino, 2009)

3.8.1. Akselerometer

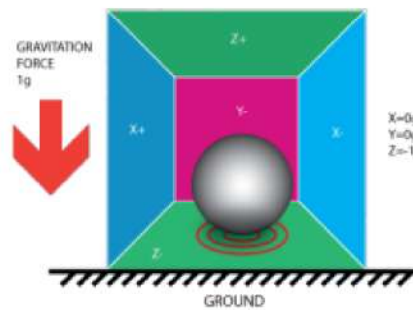
Akselerometer adalah sensor yang digunakan untuk mengukur akselerasi yang terjadi. Akselerasi tersebut diukur berdasarkan atau relatif pada gaya jatuh dari suatu benda. Penggambaran suatu sensor akselerometer adalah suatu kubus dengan bola pejal di dalamnya. Pada keadaan diam tanpa gravitasi bidang maka bola akan berada di tengah-tengah ruang kubus seperti melayang dengan penggambarannya yang ditunjukkan pada Gambar 3.9. Perilaku akselerometer yang lain seperti akselerometer pada saat +X ditunjukkan pada Gambar 3.10 Akselerometer saat kondisi +X (Starlino, 2009), akselerometer saat kondisi jatuh bebas ditunjukkan pada Gambar 3.11, dan akselerometer saat berotasi di sumbu Y ditunjukkan pada Gambar 3.12.



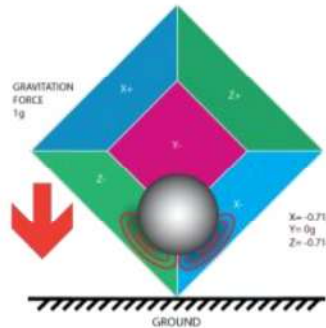
Gambar 3.9 Akselerometer saat kondisi normal (Starlino, 2009)



Gambar 3.10 Akselerometer saat kondisi +X (Starlino, 2009)

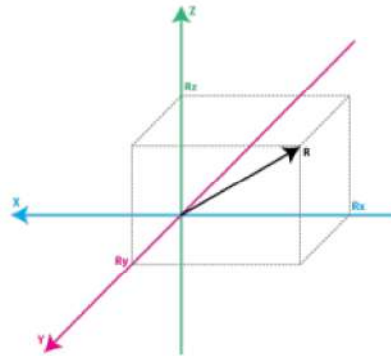


Gambar 3.11 Akselerometer saat kondisi -Z atau jatuh bebas (Starlino, 2009)



Gambar 3.12 Akselerometer saat kondisi berotasi di sumbu Y (Starlino, 2009)

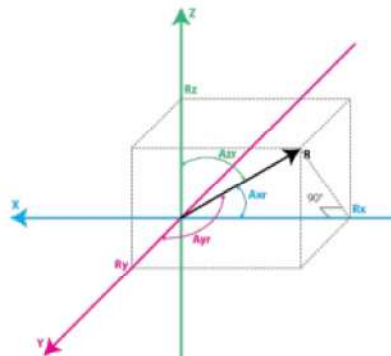
Gambar 3.13 menunjukkan penggambaran akselerometer dalam vektor bidang. Sensor akselerometer dapat digambarkan dari suatu keadaan posisi pada suatu titik di suatu bidang ruang dalam bentuk vektor. Posisi digambarkan dengan nilai vektor yang dimiliki pada sumbu dan. Tetapi akselerometer tidak dapat untuk merepresentasikan sudut yang dibentuk dari vektor yang ada atau dari berputaran vektor atau sumbu koordinat.



Gambar 3.13 Ruang vektor representasi akselerometer (Starlino, 2009)

3.8.2. Giroskop

Giroskop merupakan sensor untuk mengukur dan menjaga orientasi berdasarkan prinsip kerja momentum angular. Standar *MEMS* giroskop yang mengambil prinsip dari vibrasi suatu elemen. *MEMS (Micro Electro-Mechanical System)* adalah giroskop yang sering digunakan pada peralatan elektronik. Gambar 3.14 merupakan representasi penggambaran giroskop pada ruang vektor dan sudut yang dapat digambarkan pada setiap perputaran sumbu yang terjadi secara cepat ataupun perlahan pada waktu tertentu.



Gambar 3.14 Ruang vektor representasi giroskop (Starlino, 2009)

BAB IV

ANALISIS DAN PERANCANGAN SISTEM

4.1. Analisis Sistem

Robot humanoid digerakan menggunakan motor servo yang berperan sebagai sendi robot. Secara umum pada bagian kaki, robot humanoid memiliki 12 derajat kebebasan atau *degree of freedom* (DOF). Oleh karena itu, maka diperlukan 12 motor servo sebagai sendi-sendi robot yang harus dikendalikan agar robot dapat bergerak sesuai dengan pola yang akan diinginkan.

Teori untuk menganalisa mekanisme hubungan antara posisi dan sikap dari sudut sendi dan batang tubuh robot dapat dinyatakan sebagai kinematika. Kinematika merupakan dasar dari bentuk robot, yang juga digunakan untuk desain grafis. Teori kinematika menggunakan teori matematis dan algoritma yang merepresentasikan suatu objek bergerak dalam ruang 3 dimensi (Kajita et al., 2014).

Posisi kemiringan robot terhadap sumbu bumi diketahui dari sudut *pitch*, dan *roll*. Sudut orientasi robot yang mempengaruhi keseimbangan berdiri robot humanoid diketahui menggunakan pembacaan nilai dari sensor IMU. Sensor IMU yang digunakan terdiri dari sensor akselerometer dan giroskop yang nilai dari kedua sensor tersebut dipadukan untuk mendapatkan nilai *roll* dan *pitch*.

Batas nilai toleransi *steady-state error* untuk sistem linier orde 2 adalah 5% dari rentang nilai yang dikendalikan (Ogata, 2009). Rentang nilai yang dikendalikan pada sudut kemiringan permukaan *pitch* dan *roll* adalah sebesar 40° dikarenakan sudut yang lebih besar dari 40° merupakan kondisi dimana robot humanoid dapat terjatuh.

Terdapat beberapa parameter yang dapat dijadikan acuan untuk menentukan tingkat keberhasilan dari penelitian ini. Parameter keberhasilan dari sistem kendali robot humanoid ini dapat diketahui dari beberapa kondisi yang disebutkan sebagai berikut:

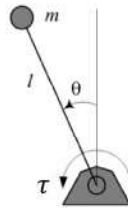
1. Nilai steady-state error dibawah batas toleransi 5% dari nilai maksimal kemiringan permukaan yaitu sebesar 40° untuk sudut *pitch* dan *roll*

yang dikendalikan, sehingga batas toleransi sudut adalah $\pm 2,25^{\circ}$ untuk sudut *pitch* dan *roll*.

2. *Risetime* yang dihasilkan untuk sudut *pitch* dan *roll* lebih cepat atau sama dengan 1 detik.
3. *Settling time* yang dihasilkan untuk sudut *pitch* dan *roll* yang dihasilkan lebih cepat atau sama dengan 2 detik.
4. *Maximum overshoot* untuk sudut *pitch* dan *roll* yang dihasilkan lebih kecil dari $\pm 2^{\circ}$.
5. Kecepatan perubahan kemiringan maksimum yang dapat diatasi adalah 0.1 rad/s untuk sudut *roll* dan sudut *pitch*.

4.2. Rancangan Pemodelan Sistem

Dalam sistem gerak robot humanoid yang terdiri susunan sendi-sendi yang kompleks, dibutuhkan suatu pemodelan yang dapat merepresentasikan dinamika sistem robot secara sederhana. Sistem robot merupakan sistem yang dinamis yang memanfaatkan pemodelan pendulum terbalik dua dimensi dengan persamaan yang dijabarkan menggunakan pendekatan Newton-Euler. Model tersebut merepresentasikan dinamika robot humanoid. Gambar 4.1 merupakan model pendulum terbalik dalam dua dimensi.



Gambar 4.1 Model pendulum terbalik dua dimensi (Jazar, 2010)

Dari Gambar 4.1 tersebut maka didapatkan persamaan (4.1) yang didapatkan dengan pendekatan konsep *Newton-Euler*.

$$I\ddot{\theta} - mgl \sin \theta = \tau \quad (4.1)$$

Keterangan:

- m = massa robot;
 l = panjang lengan pendulum;
 θ = simpangan sudut kemiringan;



- $\ddot{\theta}$ = percepatan sudut;
 I = inersia pendulum;
 τ = merupakan torsi pendulum.

$\sin(\theta)$ dapat dianggap θ dikarenakan sudut simpangan yang mempengaruhi keseimbangan robot humanoid cukup kecil. Dengan demikian didapatkan persamaan (4.2)

$$I\ddot{\theta} - mgl\theta = \tau \quad (4.2)$$

Walaupun model yang digunakan adalah model yang direpresentasikan dalam ruang dua dimensi, namun dalam implementasinya robot melakukan pergerakan dalam ruang tiga dimensi. Oleh karena itu, diperlukan perhitungan untuk mendapatkan nilai inersia dalam ruang tiga dimensi.

Nilai inersia I_{xx} dan I_{yy} didapatkan dengan persamaan (4.3) dan (4.4)

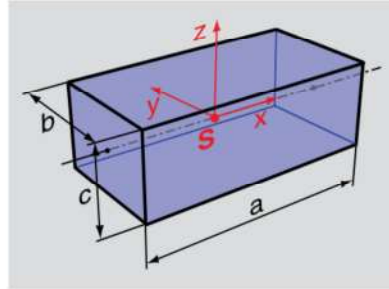
$$I_{xx} = \sum_{j=1}^n \left(I_{G_{xxj}} + m_j(y_j^2 + Z_j^2) \right) \quad (4.3)$$

$$I_{yy} = \sum_{j=1}^n \left(I_{G_{yyj}} + m_j(x_j^2 + Z_j^2) \right) \quad (4.4)$$

dengan $I_{G_{xxj}}$ dan $I_{G_{yyj}}$ adalah inersia pada sumbu x dan y bodi secara berurutan dari masing-masing komponen ke- j ; x_j dan y_j adalah jarak titik pusat massa masing-masing komponen ke- j ke poros putar pada sumbu x dan y ; dan m_j adalah massa dari masing-masing komponen ke- j (Hibbeler, 2016).

$I_{G_{xxj}}$ dan $I_{G_{yyj}}$ pada persamaan (4.3) dan (4.4) memiliki persamaan sesuai dengan bentuk komponen yang akan diukur. Terdapat dua bentuk dasar yang digunakan untuk menghitung inersia benda $I_{G_{xxj}}$ dan $I_{G_{yyj}}$ yaitu bentuk balok pada gambar Gambar 4.2 yang didapatkan menggunakan persamaan (4.5) dan (4.6).

Nilai inersia robot dianggap tetap walaupun robot memiliki pergerakan yang menyebabkan nilai inersia dapat berubah. Hal tersebut dikarenakan perubahan nilai inersia robot yang kecil, sehingga perubahan nilai inersia memberikan pengaruh yang kecil pula bagi sistem keseimbangan robot.



Gambar 4.2 Momen inersia bangun ruang balok (Braun, 2012)

$$I_{G_{xx}} = \sum_{j=1}^n \left(I_{G_{xxj}} + m_j (y_j^2 + z_j^2) \right) \quad (4.5)$$

$$I_{G_{yy}} = \frac{1}{12} m (b^2 + c^2) \quad (4.6)$$

Sendi-sendi robot humanoid digerakkan menggunakan motor servo yang dapat menghasilkan suatu pergerakan dengan mengatur variabel sudut dari motor. Dikarenakan nilai masukan dari model berupa torsi, maka diperlukan suatu persamaan yang menghubungkan antara torsi dengan sudut sendi robot.

Agar robot humanoid dapat mempertahankan keadaannya saat berdiri diatas permukaan bidang miring, maka torsi yang dibutuhkan untuk melawan torsi yang terjadi akibat gaya gravitasi harus sebanding ($\tau_{reaksi} = \tau_{aksi}$) dengan τ_{reaksi} merupakan torsi yang dihasilkan oleh sistem pergerakan robot (torsi pendulum), dan τ_{aksi} merupakan torsi yang terjadi akibat gaya gravitasi.

Torsi yang terjadi akibat gaya gravitasi memiliki nilai berdasarkan persamaan (4.7) dan (4.8)

$$\tau_{aksi} = mgl\theta \quad (4.7)$$

$$\tau_{reaksi} = mgl\theta \quad (4.8)$$

Dengan demikian, maka didapatkan persamaan yang menghasilkan nilai sudut pendulum pada persamaan (4.9)

$$\theta = \frac{\tau_{reaksi}}{mgl} \quad (4.9)$$

Berdasarkan persamaan tersebut dianggap bahwa $\frac{1}{mgl} = k$ dimana k adalah konstanta yang menghubungkan antara sudut pendulum dengan torsi pendulum.

$$k = \frac{1}{mgl} \quad (4.10)$$

Hubungan antara torsi pendulum dengan sudut pendulum dapat direpresentasikan kedalam matriks pada persamaan (4.11).

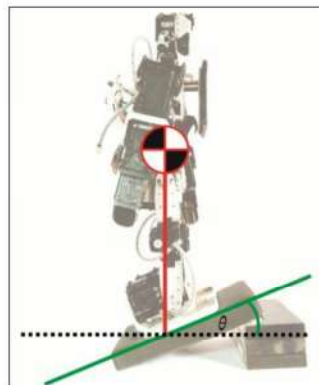
$$\begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} = \begin{bmatrix} k_1 & 0 \\ 0 & k_2 \end{bmatrix} \begin{bmatrix} \theta_x \\ \theta_y \end{bmatrix} = A_\tau \begin{bmatrix} \theta_x \\ \theta_y \end{bmatrix} \quad (4.11)$$

Berdasarkan persamaan (4.10) sudut pendulum didapatkan dengan menggunakan persamaan (4.12)

$$\begin{bmatrix} \theta_x \\ \theta_y \end{bmatrix} = A_\tau^{-1} \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} \quad (4.12)$$

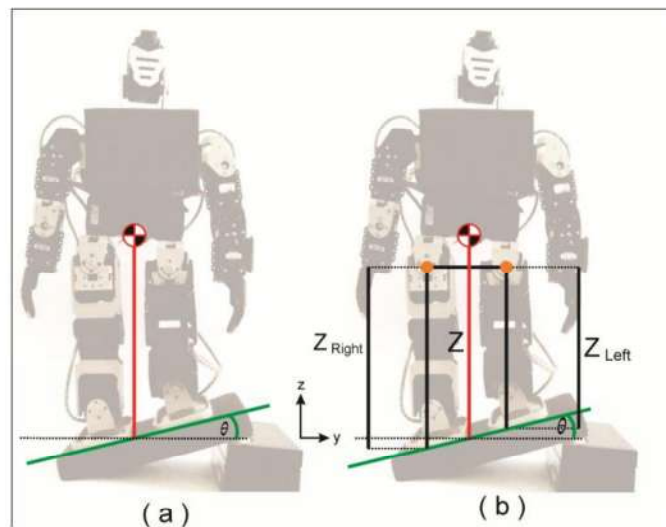
dengan θ_x merupakan sudut *pitch* dan θ_y merupakan sudut *roll*, τ_1 adalah torsi pada sudut putar *pitch* dan τ_2 merupakan torsi pada sudut putar *roll*, dan k_1 dan k_2 secara berurutan adalah konstanta hubungan torsi dan sudut pendulum pada sudut putar *pitch* dan *roll*. Sedangkan A_τ adalah matriks yang berisi konstanta k_1 dan k_2 yang menghubungkan torsi dengan sudut pendulum.

Sudut θ_x dan θ_y digunakan sebagai acuan untuk menentukan pola pergerakan robot dalam mempertahankan keadaannya agar tetap seimbang berdiri diatas permukaan bidang miring. Sudut θ_x , nilainya dapat dimasukkan secara langsung kedalam variabel masukkan sudut pada servo motor yang berada di bagian pergelangan kaki seperti pada . Sedangkan untuk θ_y , memerlukan suatu perlakuan khusus. Pergerakan robot saat mempertahankan keseimbangan berdiri pada sudut *pitch* ditunjukkan pada Gambar 4.3.



Gambar 4.3 Sistem keseimbangan berdiri robot humanoid pada sudut putar *pitch*

Dalam mempertahankan keseimbangan pada kemiringan dengan sudut putar *roll*, robot humanoid direpresentasikan dalam model pendulum terbalik. Namun, dalam keadaan yang sesungguhnya bentuk kaki robot yang menopang tubuhnya terdiri dari dua kaki yang tidak membantuk sebuah pendulum terbalik seperti yang digambarkan pada Gambar 4.4. Model pendulum terbalik disesuaikan dengan mekanisme gerak robot menggunakan *inverse kinematic* dan *forward kinematic*. *Forward Kinematic* digunakan untuk menentukan selisih jarak pada ΔZ_{Left} maupun ΔZ_{Right} pada Gambar 4.5. *Inverse kinematic* digunakan untuk menentukan nilai sudut-sudut pada masing-masing sendi yang mempengaruhi pergerakan robot saat mempertahankan keseimbangan saat berdiri sesuai yang digambarkan pada Gambar 4.6.



Gambar 4.4 Sistem keseimbangan berdiri robot humanoid pada sudut putar *roll*

Dalam Gambar 4.4 (a) digambarkan model pendulum terbalik yang sama dengan pendulum terbalik yang diimplementasikan untuk mengatasi kemiringan pada sudut putar *pitch*. Setelah itu pada Gambar 4.4 (b) digambarkan pendulum terbalik dengan panjang lengan pendulum Z dengan Z_{Right} sebagai panjang jarak antara pinggul kanan robot dengan pergelangan kaki robot dan Z_{Left} sebagai jarak antara pinggul kiri dengan pergelangan kaki kiri robot. Nilai Z dalam model ini memiliki nilai yang konstan. Agar panjang lengan pendulum tetap konstan pada



saat robot mempertahankan keseimbangannya, maka nilai Z_{Left} dan nilai Z_{Right} harus berubah. Perubahan panjang Z_{Left} dan Z_{Right} sendiri tergantung dari arah rotasi pendulum. Sesuai dengan Gambar 4.5, jika pendulum berotasi searah jarum jam, maka nilai Z_{Left} harus lebih kecil dari Z dan Z_{Right} harus lebih besar dari Z begitu pula jika pendulum berotasi berlawanan arah jarum jam nilai Z_{Right} lebih kecil dari Z dan Z_{Left} lebih kecil dari Z . Nilai ΔZ_{Right} dan ΔZ_{Left} merupakan selisih dari nilai Z dengan Z_{Left} maupun Z dengan Z_{Right} . Nilai ΔZ_{Left} dan ΔZ_{Right} ditentukan dari simpangan sudut yang berasal dari gangguan sistem (θ), dengan nilai L_h yang merupakan jarak pinggul bagian tengah hingga pinggul bagian tepi maka nilai ΔZ_{Left} dan ΔZ_{Right} dapat dihasilkan melalui persamaan (4.13) dan (4.14).

$$\Delta Z_{Right} = \tan\theta L_h \quad (4.13)$$

$$\Delta Z_{Left} = -(\tan\theta L_h) \quad (4.14)$$

Setelah itu maka nilai Z_{Right} dan nilai Z_{Left} didapatkan dengan persamaan (4.15) dan (4.16).

$$Z_{Right} = Z + \Delta Z_{Right} \quad (4.15)$$

$$Z_{Left} = Z + \Delta Z_{Left} \quad (4.16)$$

Nilai sudut dari masing-masing sendi kaki dapat ditemukan menggunakan persamaan (4.17), (4.18), (4.19), (4.20), (4.21) dan (4.22).

$$\theta_{1right} = \arccos\left(\frac{L1^2 + Z_{Right}^2 - L2^2}{2 \cdot L1 \cdot Z_{Right}}\right) \quad (4.17)$$

$$\theta_{1Left} = \arccos\left(\frac{L1^2 + Z_{Left}^2 - L2^2}{2 \cdot L1 \cdot Z_{Left}}\right) \quad (4.18)$$

$$\theta_{2right} = \frac{\pi}{2} - \arccos\left(\frac{L1^2 + L2^2 - Z_{Right}^2}{2 \cdot L1 \cdot L2}\right) \quad (4.19)$$

$$\theta_{2Left} = \frac{\pi}{2} - \arccos\left(\frac{L1^2 + L2^2 - Z_{Left}^2}{2 \cdot L1 \cdot L2}\right) \quad (4.20)$$

$$\theta_{3right} = \theta_{3a_{right}} = \arccos\left(\frac{L2^2 + Z_{Right}^2 - L2^2}{2 \cdot L2 \cdot Z_{Right}}\right) \quad (4.21)$$

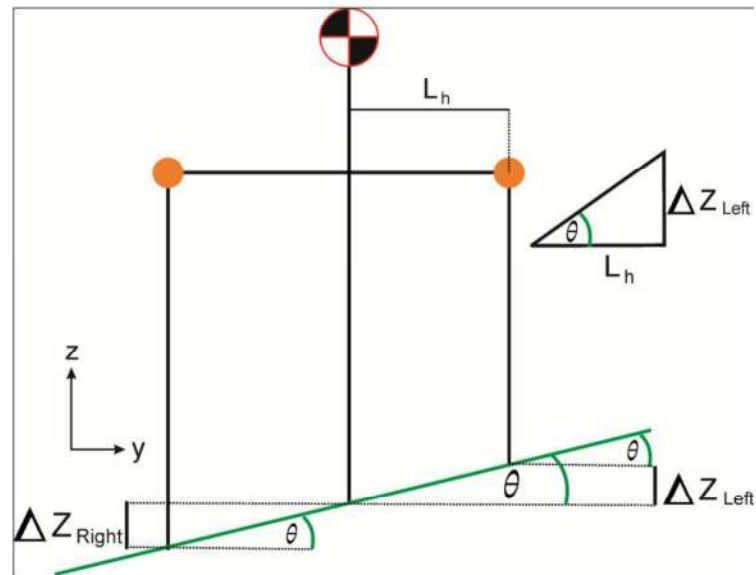


$$\theta_{3_{Left}} = \theta_{3a_{Left}} = \arccos\left(\frac{L2^2 + Z_{Left}^2 - L2^2}{2 \cdot L2 \cdot Z_{Left}}\right) \quad (4.22)$$

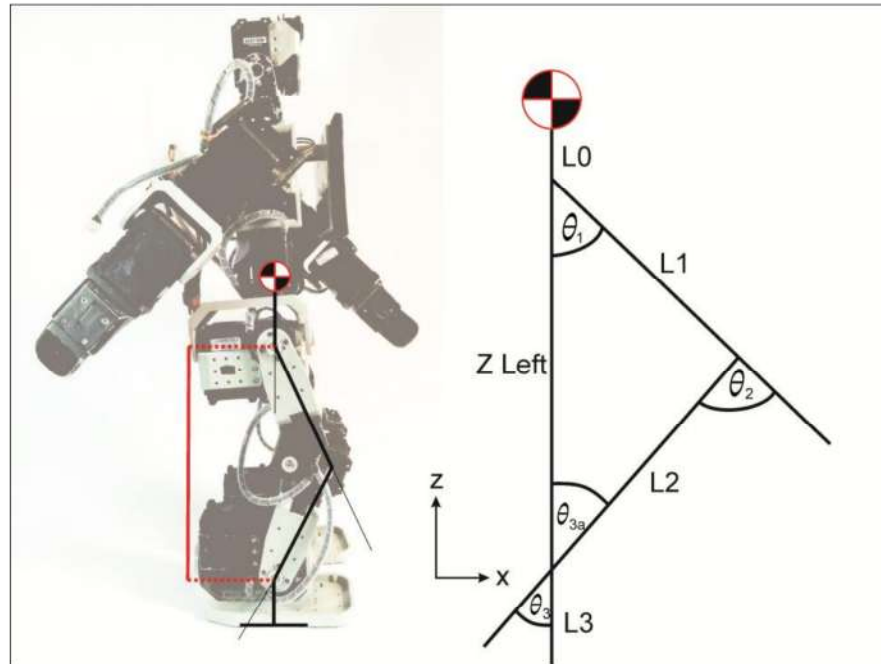
Keterangan:

- $\theta_{1_{Right}}$ = sudut *pitch* sendi pinggul kanan
- $\theta_{1_{Left}}$ = sudut *pitch* sendi pinggul kiri
- $\theta_{2_{Right}}$ = sudut *pitch* sendi lutut kanan
- $\theta_{2_{Left}}$ = sudut *pitch* sendi lutut kiri
- $\theta_{3_{Right}}$ = sudut *pitch* sendi pergelangan kaki kanan
- $\theta_{3_{Left}}$ = sudut *pitch* sendi pergelangan kaki kiri

Sedangkan sudut dari pergelangan kaki yang bergerak dalam sudut putar *roll* didapatkan dengan mengambil nilai masukkan hasil dari perhitungan menggunakan LQR.



Gambar 4.5 Sistem keseimbangan berdiri robot humanoid pada sudut putar *roll* dengan selisih nilai *Z* pada kedua kaki robot



Gambar 4.6 *Inverse Kinematic* robot humanoid dalam satu sumbu

4.3. Rancangan Sistem Kendali

Robot humanoid merupakan robot dengan sistem MIMO sehingga dibutuhkan sistem kendali yang merepresentasikan modelnya kedalam bentuk *state space*. LQR merupakan kendali yang mencari gain matriks K pada sistem *state space* secara sistematis. Adapun diagram kendali dari sistem robot humanoid terdapat pada Gambar 4.7.

Robot humanoid memiliki sistem MIMO yang memiliki dua masukan torsi dan dua keluaran sudut yaitu sudut *pitch* dan sudut *roll*. Persamaan tersebut direpresentasikan kedalam bentuk *State space* sehingga didapatkan persamaan (4.23) dan persamaan (4.24). Pada persamaan (4.23) dan (4.24) $\dot{\theta}_x$ merupakan kecepatan sudut pada sudut putar *pitch*, dan $\dot{\theta}_y$ merupakan kecepatan sudut pada sudut putar *roll*. u_1 dan u_2 merupakan masukan yang berupa torsi. Sedangkan y_1 dan y_2 merupakan keluaran dari sistem kendali. Inersia I_{xx} dan I_{yy} merupakan inersia robot masing-masing pada sudut putar *pitch* dan *roll*.

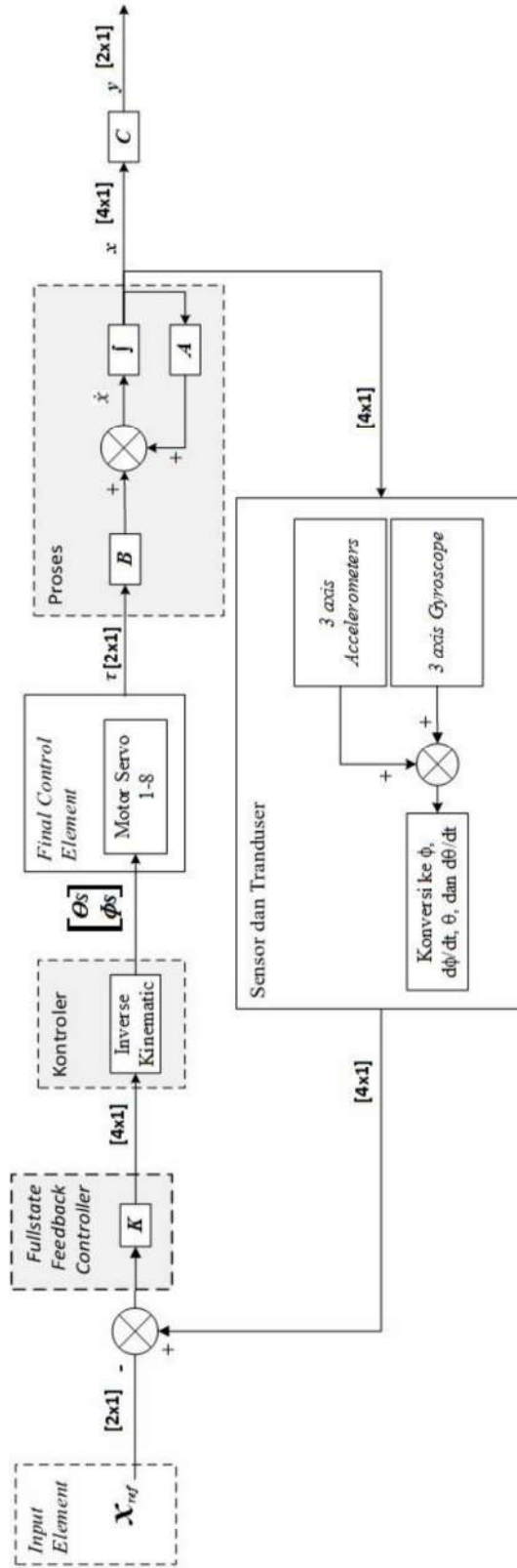


$$\begin{matrix} \begin{bmatrix} \dot{\theta}_x \\ \ddot{\theta}_x \\ \dot{\theta}_y \\ \ddot{\theta}_y \end{bmatrix} \\ \dot{\mathbf{x}} \\ \mathbf{A} \end{matrix} = \begin{matrix} \begin{bmatrix} 0 & 1 & 0 & 0 \\ \frac{m g}{I_{xx}} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & \frac{m g}{I_{xx}} & 0 \end{bmatrix} \\ \mathbf{A} \end{matrix} \begin{matrix} \begin{bmatrix} \theta_x \\ \dot{\theta}_x \\ \theta_y \\ \dot{\theta}_y \end{bmatrix} \\ \mathbf{x} \end{matrix} + \begin{matrix} \begin{bmatrix} 0 & 0 \\ \frac{1}{I_{xx}} & 0 \\ 0 & 0 \\ 0 & \frac{1}{I_{yy}} \end{bmatrix} \\ \mathbf{B} \end{matrix} \begin{matrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \\ \mathbf{u} \end{matrix} \quad (4.23)$$

$$\begin{matrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} \\ \mathbf{y} \end{matrix} = \begin{matrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \\ \mathbf{C} \end{matrix} \begin{matrix} \begin{bmatrix} \theta_x \\ \dot{\theta}_x \\ \theta_y \\ \dot{\theta}_y \end{bmatrix} \\ \mathbf{x} \end{matrix} + \begin{matrix} \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \\ \mathbf{D} \end{matrix} \begin{matrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \\ \mathbf{u} \end{matrix} \quad (4.24)$$

Dalam mempertahankan keadaannya agar tetap seimbang robot humanoid menerima masukan berupa sudut dan kecepatan sudut pada sudut putar *pitch* dan *roll* melalui IMU. Kondisi awal dari nilai sudut dan kecepatan sudut secara berturut-turut adalah 0^0 dan 0 rad/s. Disamping itu X_{ref} digunakan untuk mengatur sudut awal dari robot menyesuaikan dengan permukaan yang kemiringannya tidak sama antara satu tempat dengan tempat yang lainnya. Ketika terjadi simpangan sudut yang disebabkan oleh gangguan yang diberikan, maka nilai sudut dan kecepatan sudut tersebut dijadikan sebagai *state* dari sistem kendali. *State* tersebut kemudian dikalikan dengan matriks *gain* \mathbf{K} yang akan menghasilkan nilai masukan \mathbf{u} . Nilai *gain* \mathbf{K} didapatkan menggunakan konsep LQR yang menggunakan persamaan Riccati dengan melakukan penalaan terhadap nilai \mathbf{Q} . Nilai \mathbf{Q} yang ditentukan dalam melakukan penalaan merupakan matriks yang terdapat pada persamaan (4.25).

$$\begin{bmatrix} Q_{\theta_x} \\ Q_{\dot{\theta}_x} \\ Q_{\theta_y} \\ Q_{\dot{\theta}_y} \end{bmatrix} = \begin{bmatrix} Q_{1,1} & 0 & 0 & 0 \\ 0 & Q_{2,2} & 0 & 0 \\ 0 & 0 & Q_{3,3} & 0 \\ 0 & 0 & 0 & Q_{4,4} \end{bmatrix} \quad (4.25)$$



Gambar 4.7 Diagram block kendali sistem keseimbangan berdiri robot humanoid pada permukaan bidang miring

Dimana:

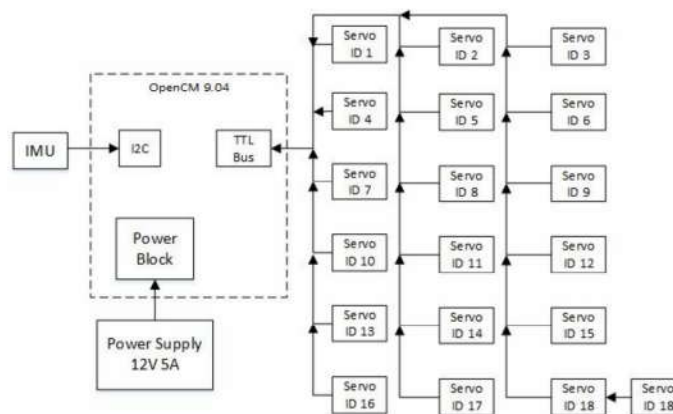
- θ_s : matriks sudut *pitch* yang berisi nilai sudut sendi pada bagian pergelangan kaki

- ϕ_s : matriks sudut *roll* yang berisi nilai sudut sendi pada bagian pinggang, lutut, dan pergelangan kaki pada kedua kaki

Nilai u yang berupa torsi yang digunakan untuk mengendalikan servo. Sebelum digunakan untuk mengendalikan servo, nilai torsi tersebut dikonversikan menjadi sudut menggunakan persamaan yang menghubungkan antara torsi dengan sudut. Setelah didapatkan nilai sudutnya, khusus untuk kendali keseimbangan pada sudut putar *roll*, nilai sudut tersebut akan digunakan untuk menentukan nilai sudut pada masing-masing sendi robot yang digerakkan oleh motor servo. Cara menentukan sudut-sudut sendi tersebut adalah menggunakan *inverse kinematic*. Nilai sudut dalam motor servo dapat dikendalikan dengan menggunakan fungsi khusus yang mengubah nilai sudut yang memiliki rentang sudut antara 0° hingga 300° menjadi nilai dalam satuan *byte* dengan rentang antara 0 hingga 1023. Nilai 0 hingga 1023 adalah nilai yang digunakan untuk mengatur posisi sudut dalam motor servo Dynamixel AX-12A.

4.4. Rancangan Arsitektur Sistem

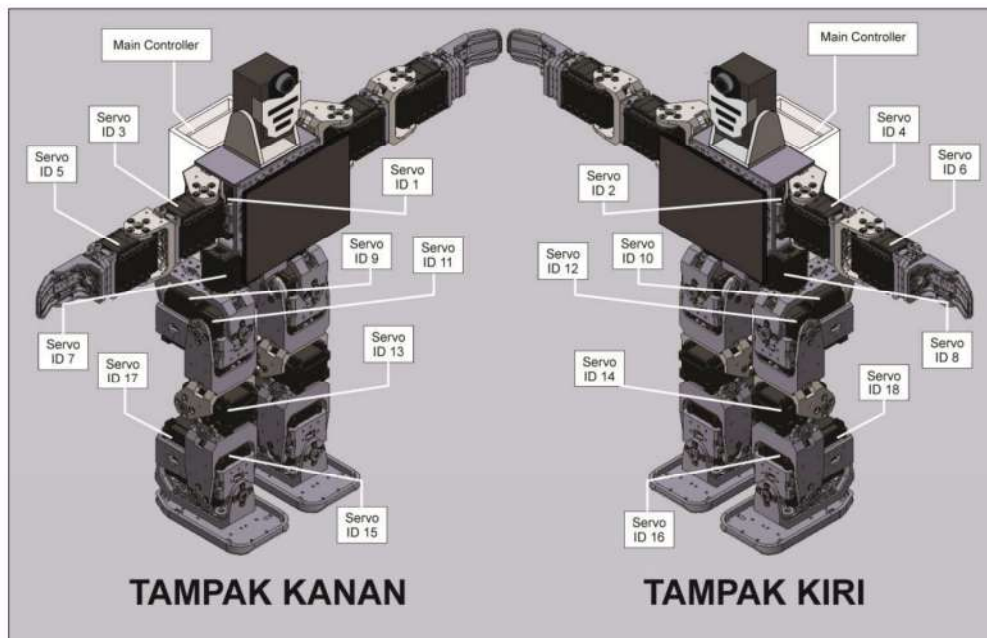
Arsitektur robot humanoid yang dirancang menggunakan 19 servo yang dihubungkan melalui jalur *TTL Bus* secara parallel. Robot humanoid tersebut mendapatkan sumber tegangan 12 volt dari *power supply* dengan arus sebesar 5A. Robot humanoid yang dirancang menggunakan sensor IMU yang menggunakan *serial bus* I2C untuk melakukan transfer data sensor dari IMU ke *main controller*. OpenCM 9.04 digunakan sebagai *main controller* atau “otak” dari sistem perangkat keras robot humanoid. Arsitektur sistem robot humanoid yang dirancang digambarkan pada Gambar 4.8 Arsitektur sistem.



Gambar 4.8 Arsitektur sistem

4.5. Rancangan Mekanik Sistem

Rancangan mekanik robot humanoid menggunakan *platform* robot BIOLOID yang dimodifikasi pada bagian tubuh bagian atas kecuali tangan, *main controller*, dan bagian kepala. Rancangan mekanik robot dapat dilihat pada Gambar 4.9. Konfigurasi ID servo yang digunakan robot ini sama dengan konfigurasi servo yang digunakan robot BIOLOID. Kerangka yang digunakan robot humanoid sama dengan kerangka robot BIOLOID. Pada bagian belakang robot terdapat kotak yang berisi perangkat keras robot yang digunakan untuk mengendalikan servo pada sendi-sendi robot.

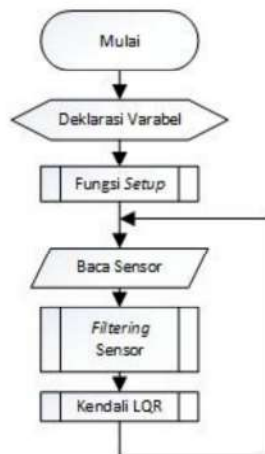


Gambar 4.9 Rancangan mekanik sistem

4.6. Rancangan Perangkat Lunak Sistem

Perangkat lunak sistem dibuat agar robot humanoid dapat bekerja sesuai dengan konsep dan metode-metode yang digunakan. Secara keseluruhan, algoritme robot humanoid terdiri deklarasi variabel, fungsi setup yang digunakan untuk melakukan inisialisasi, implementasi kendali LQR, *filtering* data sensor menggunakan *complementary filter* dan *kalman filter*, dan fungsi untuk menggerakkan aktuator.

Secara berurutan, algoritme diawali dengan melakukan deklarasi variabel, lalu melakukan inisialisasi *baudrate*, I/O, ID Servo dan sudut awalnya, dan komunikasi I2C. Setelah deklarasi variabel dan inisialisasi, maka nilai sensor didapatkan melalui IMU melalui komunikasi I2C. Kemudian nilai sensor tersebut dimasukkan kedalam fungsi *complementary filter* sehingga didapatkan nilai *pitch* dan *roll*. Setelah didapatkan nilai dari *pitch* dan *roll* maka untuk mengurangi hasil data sensor yang fluktuatif maka digunakanlah *kalman filter* sehingga didapatkan data sensor yang lebih stabil. Dari data sensor hasil perhitungan menggunakan *kalman filter* tersebut, maka nilai *pitch* dan *roll* digunakan sebagai *state* dalam kendali LQR yang selanjutnya dilakukan perhitungan untuk mendapatkan nilai inputan berupa torsi. Di dalam fungsi *setup* terdapat fungsi interupsi yang digunakan untuk menggerakkan aktuator dari robot humanoid sehingga dihasilkan pola pergerakan robot yang dapat berdiri mempertahankan keseimbangannya diatas permukaan bidang miring. Kode program utama dari sistem terdapat pada Gambar 4.10.

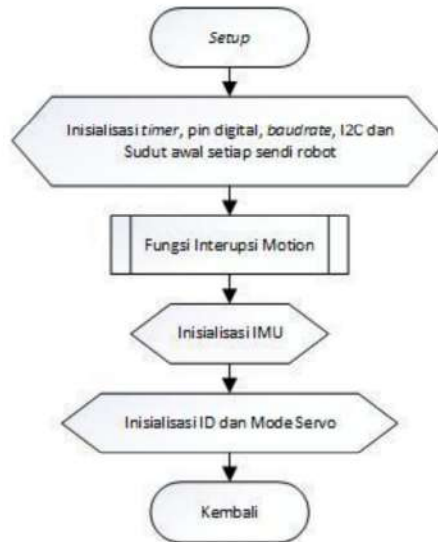


Gambar 4.10 Program utama

4.6.1. Rancangan Fungsi *Setup*

Fungsi *setup* pada Gambar 4.11 diawali dengan inisialisasi *timer interrupt*. Fungsi interupsi tersebut berisi fungsi yang digunakan untuk menggerakkan aktuator robot. Inisialisasi nilai *baudrate* dilakukan agar komunikasi antara *main controller* dengan servo dapat disinkronisasi. Setelah itu, dilakukan inisialisasi pin digital untuk menghidupkan led sebagai indikator, lalu inisialisasi komunikasi I2C agar

data dari IMU dapat ditransfer ke *main controller* yang kemudian dilanjutkan dengan inialisasi transfer data IMU dan inialisasi *mode* dan sudut awal dari setiap servo.



Gambar 4.11 Fungsi setup

4.6.2. Rancangan Fungsi *Sensor Fusion*

Filtering sensor dilakukan untuk mendapatkan nilai sudut *pitch* dan *roll* yang stabil dan akurat. Sebelum dilakukan *filtering*, data dari sensor dimasukkan kedalam fungsi *complementary filter* sehingga didapatkan nilai sudut putar *pitch* dan *roll*. Kemudian data *pitch* dan *roll* yang belum cukup stabil dimasukan kedalam fungsi *kalman filter* untuk mendapatkan data sensor yang stabil dan akurat sesuai kebutuhan sistem robot. Diagram alir fungsi *sensor fusion* ditunjukkan pada Gambar 4.12.



Gambar 4.12 Sensor fusion



4.6.3. Rancangan Fungsi Kendali LQR

Fungsi LQR pada Gambar 4.14 digunakan untuk menentukan nilai masukan u yang menentukan kemampuan robot dalam mempertahankan keseimbangannya ketika berdiri diatas permukaan bidang miring. Fungsi LQR mengolah nilai pembacaan sensor yang berupa *state* sudut dan *state* kecepatan sudut yang dikalikan dengan matriks K menjadi matriks u yang menghasilkan nilai masukan sistem berupa torsi. Dikarenakan nilai masukan yang dapat digunakan untuk menggerakkan servo adalah berupa sudut, maka nilai torsi tersebut dikonversikan menjadi sudut menggunakan persamaan yang menyatakan hubungan antara torsi dengan sudut. Nilai matriks K di dapatkan dengan menggunakan perhitungan yang dilakukan dengan perangkat lunak MATLAB. Diagram alir program untuk mendapatkan nilai matriks K terdapat pada Gambar 4.13.



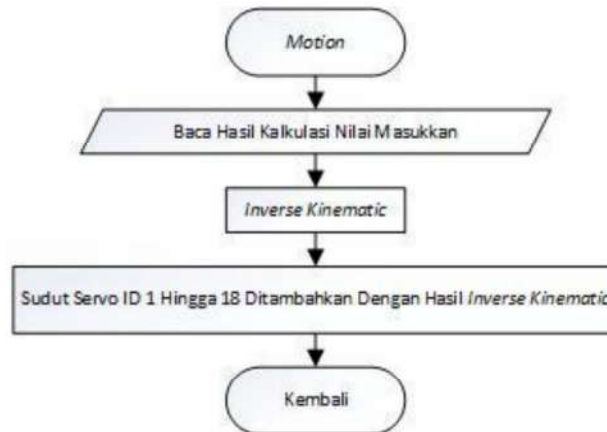
Gambar 4.13 Fungsi perhitungan matriks K



Gambar 4.14 Fungsi kendali LQR

4.6.4. Rancangan Fungsi *Motion*

Fungsi *motion* pada Gambar 4.15 digunakan untuk menggerakkan aktuator robot dengan menerima nilai masukan hasil perhitungan LQR yang kemudian dimasukkan kedalam fungsi *inverse kinematic* agar robot dapat bergerak sesuai dengan pola yang dibutuhkan untuk mempertahankan keseimbangannya. Setelah perhitungan menggunakan *inverse kinematic* dilakukan, maka nilai hasil perhitungan tersebut ditambahkan ke masing-masing servo.



Gambar 4.15 Fungsi *motion*

4.7. Pengujian Sistem

Pengujian dilakukan agar robot dapat mempertahankan keseimbangannya ketika berdiri diatas permukaan bidang miring. Adapun pengujian yang dilakukan adalah pengujian persamaan *inverse kinematic*, pengujian *filtering* sensor,

pengujian keseimbangan berdiri robot pada sudut *pitch* dengan kendali LQR, dan pengujian keseimbangan berdiri robot pada sudut *roll* dengan kendali LQR.

4.7.1. Pengujian *sensor fusion*

Pengujian ini dirancang untuk mengetahui estimator yang dirancang dengan *complementary filter* dan *kalman filter*. Tujuan dari penggunaan kedua *filter* tersebut adalah untuk mendapatkan nilai sudut putar *pitch* dan *roll*. Kemudian *Kalman filter* digunakan untuk mengurangi kesalahan nilai yang dihasilkan oleh sensor. Pengujian ini dilakukan dengan menggunakan konstanta Q_{θ} sebesar 0,001, $Q_{\dot{\theta}}$ sebesar 0,003 dan R sebesar 0,03. Sudut yang diuji adalah pembacaan sudut 0° hingga 90° dengan kenaikan 1° pada posisi diam selama 110 detik dengan waktu pengujian masing – masing sudut adalah 10 detik. Nilai *error* sudut *roll* dan *pitch* dari hasil *Complementary Filter* dengan *Kalman Filter* dan tanpa *Kalman Filter* akan dibandingkan.

4.7.2. Pengujian keseimbangan berdiri robot pada sudut putar *pitch*

Dalam pengujian keseimbangan berdiri hal pertama kali yang dilakukan adalah memilih permukaan bidang miring dengan gaya gesek statis yang besar sehingga faktor utama yang menyebabkan robot terjatuh bukanlah karena permukaan bidang yang licin melainkan karena sudut kemiringan. Setelah menentukan permukaan bidang miring yang sesuai, maka dilakukan penalaan nilai Q untuk mendapatkan nilai *gain* K yang paling optimal untuk menentukan kestabilan sistem robot humanoid.

Robot humanoid diletakkan diatas permukaan, kemudian sudut kemiringan permukaan divariasikan dengan kecepatan sudut yang konstan. Berdasarkan pengujian tersebut dapat diketahui maksimum sudut kemiringan dan kecepatan sudut kemiringan yang dapat diatasi oleh sistem kendali robot humanoid.

4.7.3. Pengujian keseimbangan berdiri robot pada sudut putar *roll*

Pengujian keseimbangan berdiri robot humanoid pada sudut putar *roll* dilakukan dengan tahapan yang sama seperti pengujian keseimbangan berdiri robot pada sudut putar *pitch*. Adapun tahapan-tahapan tersebut meliputi menentukan permukaan dengan gaya gesek yang besar, melakukan penalaan nilai Q untuk menghasilkan nilai *gain* K yang optimal. Setelah didapatkan nilai *gain* optimal,



maka keseimbangan robot diuji pada permukaan bidang miring dengan variasi sudut tertentu dan dengan variasi konstanta kecepatan sudut perubahan permukaan bidang miring. Pengujian yang telah diuraikan sebelumnya di rangkum dalam Tabel 4.1.

Tabel 4.1 Pengujian sistem

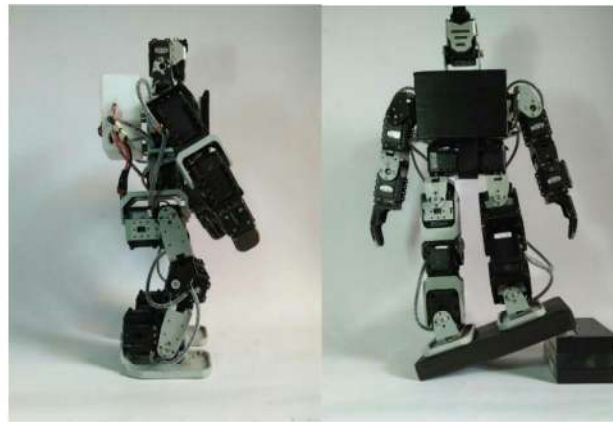
| No. | Pengujian | Tujuan Pengujian |
|-----|--------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1. | Pengujian <i>sensor fusion</i> | Mendapatkan nilai sudut putar <i>pitch</i> dan <i>roll</i> |
| 2. | Pengujian keseimbangan berdiri robot pada sudut putar <i>pitch</i> | Mendapatkan respon robot saat mempertahankan keseimbangan pada sudut <i>pitch</i> dengan kriteria respon yang terdiri dari <i>risetime</i> , <i>settling time</i> , <i>steady-state error</i> , <i>overshoot</i> maksimum, sudut kemiringan maksimum yang dapat diatasi, dan kecepatan sudut kemiringan permukaan yang dapat diatasi |
| 3. | Pengujian keseimbangan berdiri robot pada sudut putar <i>roll</i> | Mendapatkan respon robot saat mempertahankan keseimbangan pada sudut <i>roll</i> , dengan kriteria respon yang terdiri dari <i>rsetime</i> , <i>settling time</i> , <i>steady-state error</i> dan <i>overshoot</i> maksimum, yang dapat diatasi dan kecepatan maksimum kemiringan yang dapat diatasi |

BAB V

IMPLEMENTASI

5.1. Implementasi Perangkat Keras

Perangkat keras robot humanoid menggunakan 18 aktuator *dynamixel* dengan tipe AX-12A dan frame BIOLOID premium yang telah dimodifikasi. Sistem mekanik robot humanoid terdapat pada Gambar 5.1



Gambar 5.1 Mekanik robot humanoid

Robot humanoid memiliki *main controller* berupa *board* mikrokontroler OpenCM 9.04A. 18 servo yang telah ditentukan penomorannya dihubungkan langsung ke OpenCM 9.04A secara paralel melalui jalur TTL khusus untuk transfer data dari OpenCM 9.04A dengan semua servo.

Sensor IMU yang digunakan adalah GY-512 yang terdiri dari sensor akselerometer dan giroskop yang masing-masing memiliki 3 derajat kebebasan. GY-512 dihubungkan ke OpenCM 9.04A melalui jalur komunikasi serial I2C. Catu daya yang diberikan ke GY-512 adalah sebesar +3,3 volt.

Sebagai sumber daya robot humanoid, sistem robot menggunakan *power supply* dengan tegangan listrik DC sebesar 12 volt dengan arus sebesar 5A. *Power supply* dihubungkan ke *port* catu daya OpenCM 9.04A.

5.2. Pengujian *Sensor Fusion*

Pengujian hasil pengolahan data sensor IMU menggunakan *Complementary Filter* dan *Kalman Filter* dilakukan untuk menghasilkan data hasil olahan yang



stabil. Pengujian *filter* dilakukan pada sudut 0^0 hingga 60^0 . *Complementary filter* dalam program yang terdapat pada Gambar 5.3 memanfaatkan nilai sensor akselerometer dan giroskop untuk menghasilkan nilai *pitch* dan *roll* (*sensor fusion*). Nilai *pitch* dan *roll* dari perhitungan menggunakan *complementary filter* yang belum stabil kemudian diproses menggunakan *kalman filter* untuk menghasilkan nilai *pitch* dan *roll* yang lebih stabil. Kode program *kalman filter* terdapat pada Gambar 5.2.

```
1. ....
2. // The angle should be in degrees and the rate should be in degrees per
3. second and the delta time in seconds
4. float Kalman::getAngle(float newAngle, float newRate, float dt) {
5. // KasBot V2 - Kalman filter module - http://www.x-firm.com/?page_id=145
6. // Modified by Kristian Lauszus
7. // See my blog post for more information:
8. http://blog.tkjelectronics.dk/2012/09/a-practical-approach-to-kalman-
9. filter-and-how-to-implement-it
10. // Discrete Kalman filter time update equations - Time Update ("Predict")
11. // Update xhat - Project the state ahead
12. /* Step 1 */
13. rate = newRate - bias;
14. angle += dt * rate;
15. // Update estimation error covariance - Project the error covariance ahead
16. /* Step 2 */
17. P[0][0] += dt * (dt*P[1][1] - P[0][1] - P[1][0] + Q_angle);
18. P[0][1] -= dt * P[1][1];
19. P[1][0] -= dt * P[1][1];
20. P[1][1] += Q_bias * dt;
21.
22. // Discrete Kalman filter measurement update equations - Measurement
23. Update ("Correct")
24. // Calculate Kalman gain - Compute the Kalman gain
25. /* Step 4 */
26. float S = P[0][0] + R_measure; // Estimate error
27. /* Step 5 */
28. float K[2]; // Kalman gain - This is a 2x1 vector
29. K[0] = P[0][0] / S;
30. K[1] = P[1][0] / S;
31. // Calculate angle and bias - Update estimate with measurement zk
32. (newAngle)
33. /* Step 3 */
34. float y = newAngle - angle; // Angle difference
35. /* Step 6 */
36. angle += K[0] * y;
37. bias += K[1] * y;
38.
39. // Calculate estimation error covariance - Update the error covariance
40. /* Step 7 */
41. float P00_temp = P[0][0];
42. float P01_temp = P[0][1];
43.
44. P[0][0] -= K[0] * P00_temp;
45. P[0][1] -= K[0] * P01_temp;
46. P[1][0] -= K[1] * P00_temp;
47. P[1][1] -= K[1] * P01_temp;
48. return angle;
49. };
50. ....
```

Gambar 5.2 Kode program *kalman filter*



```
1. ....
2. //-----Complementary Filter-----
3.   dataAccel[0]=ax;
4.   dataAccel[1]=ay;
5.   dataAccel[2]=az;
6.   dataGiro[0]=gx;
7.   dataGiro[1]=gy;
8.   dataGiro[2]=gz;
9.
10.  float accelerationX = (signed int)(((signed int)dataAccel[0]) * 3.9);
11.  float accelerationY = (signed int)(((signed int)dataAccel[1]) * 3.9);
12.  float accelerationZ = (signed int)(((signed int)dataAccel[2]) * 3.9);
13.  double pitch_ = (180 *
14.  atan2(accelerationX,sqrt(accelerationY*accelerationY +
15.  accelerationZ*accelerationZ))/M_PI)+3;//-4
16.  double roll_ = (180 *
17.  atan2(accelerationY,sqrt(accelerationX*accelerationX +
18.  accelerationZ*accelerationZ))/M_PI)+5;
19.  ....
```

Gambar 5.3 Kode program *complementary filter*

5.3. Pengujian keseimbangan berdiri robot humanoid pada sudut *pitch*

Pengujian keseimbangan robot ketika berdiri di atas permukaan bidang miring pada sudut putar *pitch* terdiri dari beberapa tahap. Pertama model disimulasikan dengan kode program pada Gambar 5.4 yang dijalankan dalam perangkat lunak MATLAB untuk mendapatkan nilai *gain K* berdasarkan model *state space* yang telah ditentukan.

```
1. ....
2. %=====2D Inverted Pendulum with LQR=====
3. m=1.83; L=0.153; g=9.80; I=0.059171438
4. A=[ 0 1; -(L*m*g)/I 0 ]
5. B=[ 0; (1/I) ]
6. C=[ 1 0 ]
7. D=[ 0; ]
8. states = {'Theta' 'Theta_dot'};
9. inputs = {'u'};
10. outputs = {'Theta'};
11. Faro_sys_ss =
12. ss(A,B,C,D,'statename',states,'inputname',inputs,'outputname',outputs);
13. poles = eig(A)
14. Faro_sys_tf=tf(Faro_sys_ss)
15. co = ctrb(Faro_sys_ss)
16. controllability = rank(co)
17. %Q = C'*C;
18. Q=[1 0; 0 1]
19. R = 1;
20. K = lqr(A,B,Q,R)
21. Ac = [(A-B*K)];
22. Bc = [B];
23. Cc = [C];
24. Dc = [D];
25. states = {'Theta' 'Theta_dot'};
26. inputs = {'r'};
27. outputs = {'Theta'};
28. Faro_sys_cl =
29. ss(Ac,Bc,Cc,Dc,'statename',states,'inputname',inputs,'outputname',outputs);
30. ....
```

Gambar 5.4 Kode program LQR



Nilai *gain K* yang dihasilkan kemudian di masukkan ke dalam kode program pada Gambar 5.5 sehingga didapatkan nilai *u*. Nilai *u* yang merupakan hasil perhitungan kode program pada Gambar 5.6 digunakan sebagai masukan proses.

```
1. ....
2. //-----State space-Controller-----
3. #define K_Sudut_Pitch          0.4001 //gain X pitch
4. #define K_Kecepatan_Sudut_Pitch 1.5960 //gain X_dot pitch
5.
6. float Input_Torsi_Pitch(float Sudut_Pitch, float Kecepatan_Sudut_Pitch){
7.     float u;
8.     float X      = Sudut_Pitch;
9.     float X_dot  = Kecepatan_Sudut_Pitch;
10.    float K1     = K_Sudut_Pitch;
11.    float K2     = K_Kecepatan_Sudut_Pitch;
12.    u = (-1*(K1*X)) + (-1*(K2*X_dot));
13.    return u;
14. }
15. ....
```

Gambar 5.5 Kode program perhitungan nilai masukan proses

Setelah perhitungan untuk mendapatkan nilai masukan *u*, maka nilai *u* yang merupakan torsi dikonversikan menjadi nilai sudut menggunakan persamaan yang menyatakan hubungan antara torsi dengan sudut. Nilai sudut yang digunakan untuk menggerakkan masing-masing aktuator tersebut didapatkan menggunakan persamaan yang terdapat pada kode program.

```
1. ....
2. void handler_led(void) {
3.     outputTorsi=Input_Torsi_Pitch(radians(pitch), Kec_sudut_pitch);
4.     sudut_dari_u=degrees( asin( outputTorsi /
5.     ( Massa_Total * Panjang_Lengan_Pendulum * Percepatan_Gravitasi )
6.     ));
7.     increment_sudut +=sudut_dari_u;
8.     if(pitch>0){
9.         Dxl.writeWord(ID_NUM15, GOAL_POSITION, DegToBit(OFFSET_ID15-
10.    increment_sudut ));
11.         Dxl.writeWord(ID_NUM16, GOAL_POSITION,
12.    DegToBit(OFFSET_ID16+increment_sudut ));
13.     }
14.     else if(pitch<-0){
15.         Dxl.writeWord(ID_NUM15, GOAL_POSITION, DegToBit(OFFSET_ID15-
16.    increment_sudut ));
17.         Dxl.writeWord(ID_NUM16, GOAL_POSITION,
18.    DegToBit(OFFSET_ID16+increment_sudut ));
19.     }
20.     else{
21.         Dxl.goalPosition(ID_NUM15,Dxl.getPosition(ID_NUM15));
22.         Dxl.goalPosition(ID_NUM16,Dxl.getPosition(ID_NUM16));
23.     }
24. }
25. ....
```

Gambar 5.6 Kode program persamaan hubungan torsi dengan sudut dan penggerak aktuator pada sumbu putar *pitch*

5.4. Pengujian keseimbangan berdiri robot humanoid pada sudut *roll*

Pengujian keseimbangan berdiri robot ada sudut *roll* memiliki tahapan yang sama dalam menghasilkan nilai *gain K*, nilai masukkan process *u*, dan nilai sudut hasil perhitungan persamaan hubungan torsi dan sudut. Perbedaan di antara keduanya terdapat pada algoritme penggerak aktuator dari masing-masing sendi robot.

```

1. ....
2. void handler_led(void) {
3.   outputTorsi=Input_Torsi_Roll(radians(roll), Kec_sudut_roll);
4.   sudut_dari_u=degrees(asin(outputTorsi/( Massa_Total *
5.   Panjang_Lengan_Pendulum * Percepatan_Gravitasi )));
6.   increment_sudut +=sudut_dari_u;
7.   if(roll>0){
8.     delta_Z_Right=tan(radians(sudut_dari_u))*Hip_range;
9.     delta_Z_Left=tan(radians(sudut_dari_u))*Hip_range;/**(-1);
10.    sudut_ankle_R=atan2(delta_Z_Right,Hip_range)/2;
11.    sudut_ankle_L=atan2(delta_Z_Left,Hip_range)/2;
12.    increment_sudut_ankle_R+=sudut_ankle_R;
13.    increment_sudut_ankle_L+=sudut_ankle_L;
14.    Right_Leg_Z+=(delta_Z_Right); Left_Leg_Z-=(delta_Z_Left);
15.    ID_12_Pos = THETA12_1_AXIS(Right_Leg_Z);
16.    ID_11_Pos = THETA11_1_AXIS(Left_Leg_Z);
17.    ID_14_Pos = THETA14_1_AXIS(Right_Leg_Z);
18.    ID_13_Pos = THETA13_1_AXIS(Left_Leg_Z);
19.    ID_16_Pos = THETA16_1_AXIS(Right_Leg_Z);
20.    ID_15_Pos = THETA15_1_AXIS(Left_Leg_Z);
21.
22. Motion(OFFSET_ID1,OFFSET_ID2,OFFSET_ID3,OFFSET_ID4,OFFSET_ID5,OFFSET_ID6,0
23. FFSET_ID7,OFFSET_ID8,OFFSET_ID9,OFFSET_ID10,OFFSET_ID11+ID_11_Pos,OFFSET_I
24. D12+ID_12_Pos,OFFSET_ID13+ID_13_Pos,OFFSET_ID14+ID_14_Pos,OFFSET_ID15+ID_1
25. 5_Pos,OFFSET_ID16+ID_16_Pos,OFFSET_ID17+degrees(increment_sudut_ankle_R),
26. OFFSET_ID18+degrees(increment_sudut_ankle_L),1023);
27.   }
28.   else if(roll<-0){
29.     delta_Z_Right=tan(radians(sudut_dari_u))*Hip_range;
30.     delta_Z_Left=tan(radians(sudut_dari_u))*Hip_range;/**(-1);
31.     sudut_ankle_R=atan2(delta_Z_Right,Hip_range)/2;
32.     sudut_ankle_L=atan2(delta_Z_Left,Hip_range)/2;
33.     increment_sudut_ankle_R+=sudut_ankle_R;
34.     increment_sudut_ankle_L+=sudut_ankle_L;
35.     Right_Leg_Z+=(delta_Z_Right); Left_Leg_Z-=(delta_Z_Left);
36.     ID_12_Pos = THETA12_1_AXIS(Right_Leg_Z);
37.     ID_11_Pos = THETA11_1_AXIS(Left_Leg_Z);
38.     ID_14_Pos = THETA14_1_AXIS(Right_Leg_Z);
39.     ID_13_Pos = THETA13_1_AXIS(Left_Leg_Z);
40.     ID_16_Pos = THETA16_1_AXIS(Right_Leg_Z);
41.     ID_15_Pos = THETA15_1_AXIS(Left_Leg_Z);
42.
43. Motion(OFFSET_ID1,OFFSET_ID2,OFFSET_ID3,OFFSET_ID4,OFFSET_ID5,OFFSET_ID6,
44. OFFSET_ID7,OFFSET_ID8,OFFSET_ID9,OFFSET_ID10,OFFSET_ID11+ID_11_Pos,OFFSET_
45. ID12+ID_12_Pos,OFFSET_ID13+ID_13_Pos,OFFSET_ID14+ID_14_Pos,OFFSET_ID15+ID_
46. 15_Pos,OFFSET_ID16+ID_16_Pos,OFFSET_ID17+degrees(increment_sudut_ankle_R),
47. OFFSET_ID18+degrees(increment_sudut_ankle_L),1023);
48.   }
49.   }
50.   else{ }
51. }
52. ....

```

Gambar 5.7 Kode program persamaan hubungan torsi dengan sudut dan penggerak aktuator pada sumbu putar *roll*



Pada algoritme keseimbangan berdiri terhadap sumbu putar *pitch*, nilai sudut dimasukkan kedalam satu sendi putar yaitu pada bagian pergelangan kaki, sedangkan pada algoritme keseimbangan berdiri terhadap sumbu putar *roll* diperlukan persamaan *forward kinematic* dan *inverse kinematic* untuk melakukan analisa geometri robot sekaligus menghasilkan nilai-nilai sudut masing-masing sendi. Kode program untuk menggerakkan masing-masing sendi robot terdapat pada Gambar 5.7 sedangkan kode program kinematika robot humanoid terdapat pada Gambar 5.8.

```
1. ....
2. #ifndef _InverseKinematic_h_
3. #define _InverseKinematic_h_
4. float L13=77, L14=77, L15=77, L16=77;
5. float THETA12_1_AXIS(int LRL_Z){
6.     float theta12,a,u;
7.     a= ((LRL_Z*LRL_Z)+(L14*L14)-(L16*L16)) / (2*LRL_Z*L14);
8.     theta12=acos(a);
9.     u=degrees(theta12);
10.    return u;
11. }
12. float THETA14_1_AXIS(int LRL_Z){
13.     float theta14,theta14a,a,u;
14.     a=((L16*L16)+(L14*L14)-(LRL_Z*LRL_Z))/(2*L16*L14);
15.     theta14a=acos(a);
16.     theta14=(22/7)-theta14a;
17.     return u=degrees(theta14);
18. }
19. float THETA16_1_AXIS(int LRL_Z){
20.     float theta16,a,b,u;
21.     a=((LRL_Z*LRL_Z)+(L16*L16)-(L14*L14))/(2*LRL_Z*L16);
22.     theta16=acos(a);
23.     b=(-1)*theta16;
24.     return u=degrees(b);
25. }
26. float THETA11_1_AXIS(int LRL_Z){
27.     float theta11,a,b,u;
28.     a=((LRL_Z*LRL_Z)+(L13*L13)-(L15*L15))/(2*LRL_Z*L13);
29.     theta11=acos(a);
30.     b=(-1)*theta11;
31.     return u=degrees(b);
32. }
33. float THETA13_1_AXIS(int LRL_Z){
34.     float theta13,theta13a,a,u;
35.     a=((L15*L15)+(L13*L13)-(LRL_Z*LRL_Z))/(2*L16*L13);
36.     theta13a=acos(a);
37.     theta13=(-1)*((22/7)-theta13a);
38.     return u=degrees(theta13);
39. }
40. float THETA15_1_AXIS(int LRL_Z){
41.     float theta15,a,b,u;
42.     a=((LRL_Z*LRL_Z)+(L15*L15)-(L13*L13))/(2*LRL_Z*L15);
43.     theta15=acos(a);
44.     return u=degrees(theta15);
45. }
46. }
47. #endif
48. ....
```

Gambar 5.8 Kode program kinematika robot

Sebelum diimplementasikan kedalam sistem robot, persamaan *inverse kinematic* disimulasikan menggunakan program Processing IDE terdapat pada

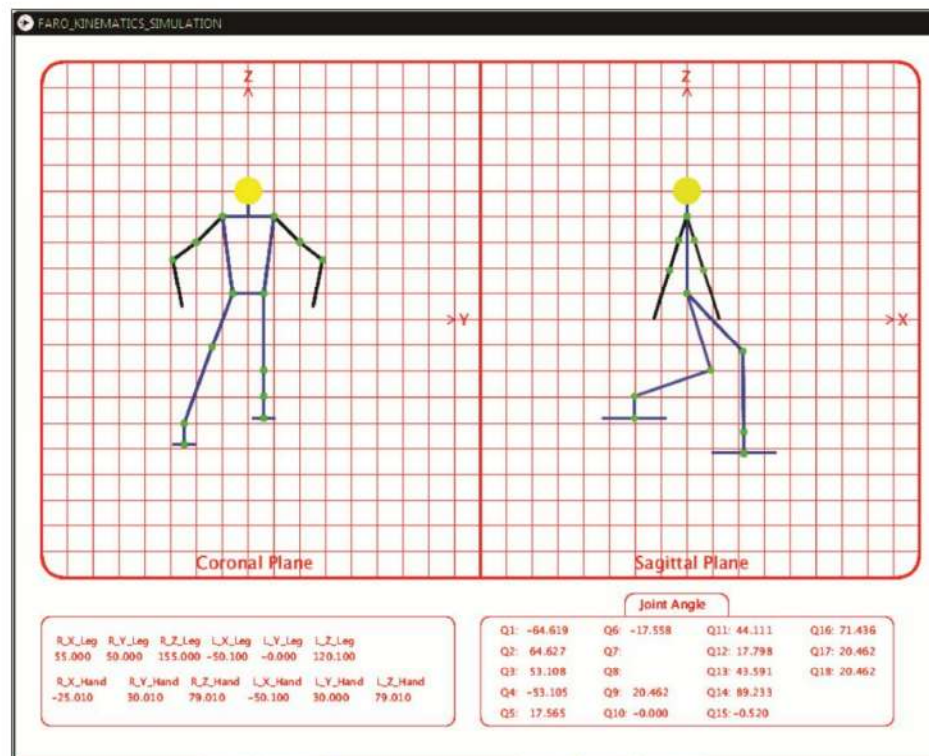
Gambar 5.9. Simulasi *inverse kinematic* dilakukan untuk memprediksikan

```

1. ....
2.   R_Z           = R_Z-21;
3.   R_XZ         = TRIANGLE_HYPOTENUSE(R_X,R_Z);
4.   R_THETA_1a  = TRIANGLE_ANGLE( R_XZ, R_L1, R_L2 );
5.   R_THETA_1b  = -atan2(R_Z,R_X);
6.   R_THETA_1    = (PI/2)+R_THETA_1a + R_THETA_1b;
7.   R_THETA_2a  = TRIANGLE_ANGLE( R_L2, R_L1, R_XZ );
8.   R_THETA_2    = PI-R_THETA_2a;
9.   R_THETA_3a  = TRIANGLE_ANGLE( R_L2, R_XZ, R_L1 );
10.  R_THETA_3b  = R_THETA_1b;
11.  R_THETA_3    = (R_THETA_3a - R_THETA_3b)-radians(90.0);
12.  R_THETA_4    = atan2(R_Y,R_Z);
13.  R_THETA_5    = R_THETA_4;
14.  //Roll_Pitch_Yaw Kaki kanan
15.  R_THETA_3    = R_THETA_3+radians(R_Pitch);
16.  R_THETA_5    = R_THETA_5+radians(R_Roll);
17.  ....

```

nilai-nilai sudut masing-masing sendi robot pada bagian kaki untuk meminimalisir kerusakan yang disebabkan oleh nilai sudut yang melebihi batas maksimal dan batas minimal dari masing-masing sendi robot. Gambar 5.10 merupakan kode program perhitungan *inverse kinematic* pada simulasi.



Gambar 5.9 Kinematic simulator



Gambar 5.10 Kode program persamaan *inverse kinematic* pada simulator

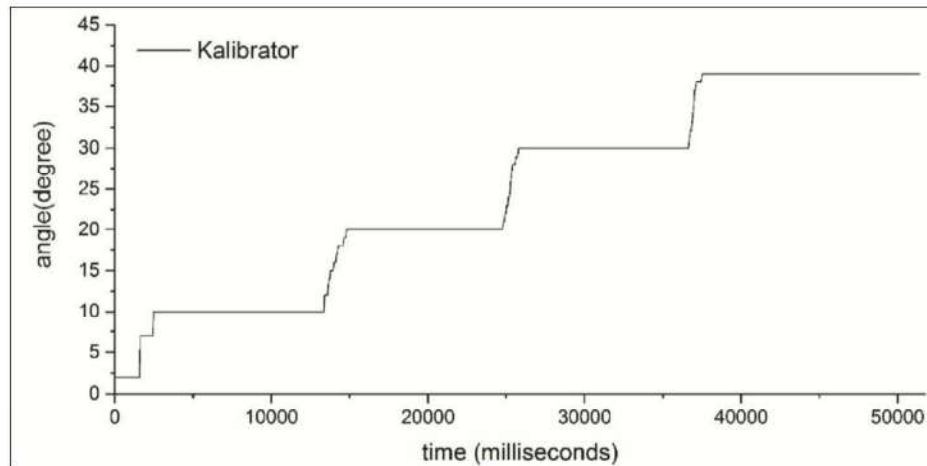
```
1. ....
2.   R_Z           = R_Z-21;
3.   R_XZ          = TRIANGLE_HYPOTENUSE(R_X,R_Z);
4.   R_THETA_1a   = TRIANGLE_ANGLE( R_XZ, R_L1, R_L2 );
5.   R_THETA_1b   = -atan2(R_Z,R_X);
6.   R_THETA_1    = (PI/2)+R_THETA_1a + R_THETA_1b;
7.   R_THETA_2a   = TRIANGLE_ANGLE( R_L2, R_L1, R_XZ );
8.   R_THETA_2    = PI-R_THETA_2a;
9.   R_THETA_3a   = TRIANGLE_ANGLE( R_L2, R_XZ, R_L1 );
10.  R_THETA_3b   = R_THETA_1b;
11.  R_THETA_3    = (R_THETA_3a - R_THETA_3b)-radians(90.0);
12.  R_THETA_4    = atan2(R_Y,R_Z);
13.  R_THETA_5    = R_THETA_4;
14.  //Roll_Pitch_Yaw_Kaki_kanan
15.  R_THETA_3     = R_THETA_3+radians(R_Pitch);
16.  R_THETA_5     = R_THETA_5+radians(R_Roll);
17.  ....
```

BAB VI

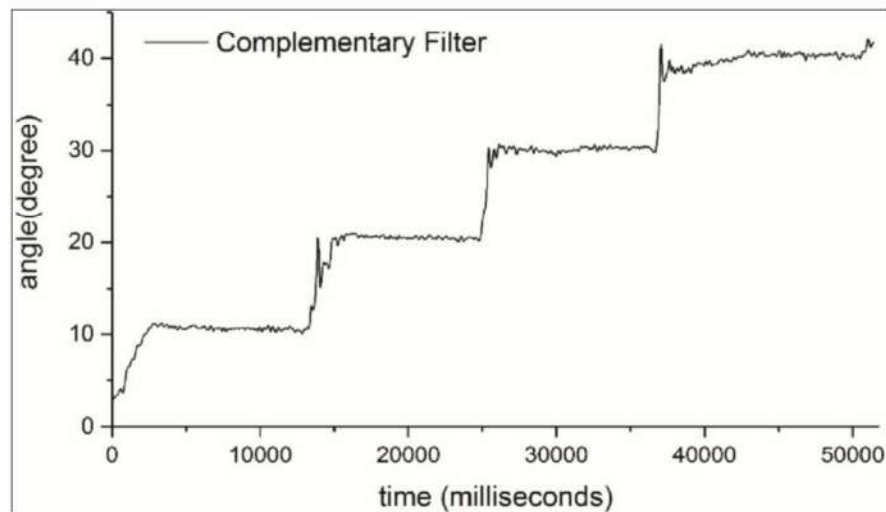
HASIL DAN PEMBAHASAN

6.1. Hasil Pengujian *Sensor Fusion*

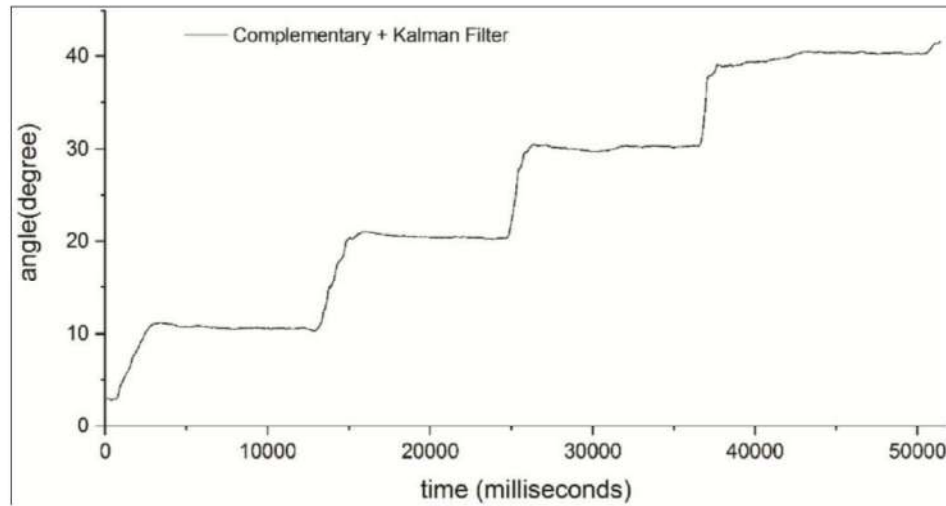
Pengujian *sensor fusion* terdiri dari pengujian hasil implementasi dengan menggunakan *Complementary Filter* dan *Complementary filter* yang digabungkan dengan *Kalman filter*. Pengujian *sensor fusion* dilakukan pada sudut *pitch* dan sudut *roll*. Sudut yang diuji memiliki rentang antara 0° hingga 40° dengan variasi sudut antara 10° , 20° , 30° dan 40° .



Gambar 6.1 Kalibrator pengujian *sensor fusion* pada sudut *pitch*



Gambar 6.2 *Sensor fusion* dengan *complementary filter* pada sudut *pitch*



Gambar 6.3 *Sensor fusion dengan kalman filter pada sudut pitch*

Gambar 6.1 Kalibrator pengujian *sensor fusion* pada sudut *pitch* menunjukkan nilai sudut kalibrator yang dijadikan acuan untuk mengetahui tingkat akurasi dari nilai pengukuran sudut yang dihasilkan oleh sensor. Nilai kalibrator dan hasil pengukuran sensor baik yang menggunakan algoritma *complementary filter* maupun *kalman filter* dicatat dalam waktu yang bersamaan.

Berdasarkan Gambar 6.3 dapat diketahui hasil dari pengukuran sensor menggunakan algoritma *complementary filter* dengan *kalman filter* menghasilkan nilai yang lebih akurat dibandingkan dengan hasil pengukuran sensor yang menggunakan *complementary filter* tanpa *kalman filter* seperti yang digambarkan pada Gambar 6.2.

Tabel 6.1 Rata-rata nilai dan standar deviasi hasil pengujian sudut *pitch*

| Kalibrator (derajat) | Standar deviasi sudut terukur (derajat) | | Rata-rata sudut terukur (derajat) | |
|-------------------------|--------------------------------------------|--------------------------------------|--------------------------------------|--------------------------------------|
| | <i>Complementary filter</i> | <i>Complementary + Kalman Filter</i> | <i>Complementary filter</i> | <i>Complementary + Kalman Filter</i> |
| 10 | 0,325 | 0,076 | 10,572 | 10,568 |
| 20 | 0,267 | 0,054 | 20,489 | 20,491 |
| 30 | 0,302 | 0,061 | 30,273 | 30,282 |
| 40 | 0,370 | 0,078 | 40,391 | 40,400 |



Tabel 6.1 menunjukkan nilai hasil pengujian *sensor fusion* dengan rata-rata nilai dan standar deviasi yang dihasilkan dari 100 sampel data pada masing-masing variasi sudut *roll* dari 10^0 , 20^0 , 30^0 , dan 40^0 . Pengukuran dengan algoritme *complementary filter* pada sudut 10^0 menghasilkan nilai sudut rata-rata $10,572^0 \pm 0,325^0$, sedangkan dengan algoritme *complementary filter* dan *kalman filter* pada sudut 10^0 menghasilkan $10,568^0 \pm 0,076^0$. Pada sudut 40^0 *sensor fusion* dengan *complementary filter* menghasilkan rata-rata nilai sudut $40,391^0 \pm 0,370^0$ dan *complementary filter* ditambahkan dengan *kalman filter* menghasilkan nilai $40,400^0 \pm 0,078^0$. Berdasarkan hasil pengukuran yang telah diuji pada sudut *pitch*, algoritme *complementary filter* tanpa *kalman filter* menghasilkan nilai standar deviasi pengukuran yang lebih besar dibandingkan dengan algoritme *complementary filter* dengan *kalman filter* sehingga algoritme *complementary filter* yang ditambahkan dengan *kalman filter* menghasilkan nilai pengukuran sudut yang lebih akurat.

Tabel 6.2 Rata-rata nilai dan standar deviasi hasil pengujian sudut *roll*

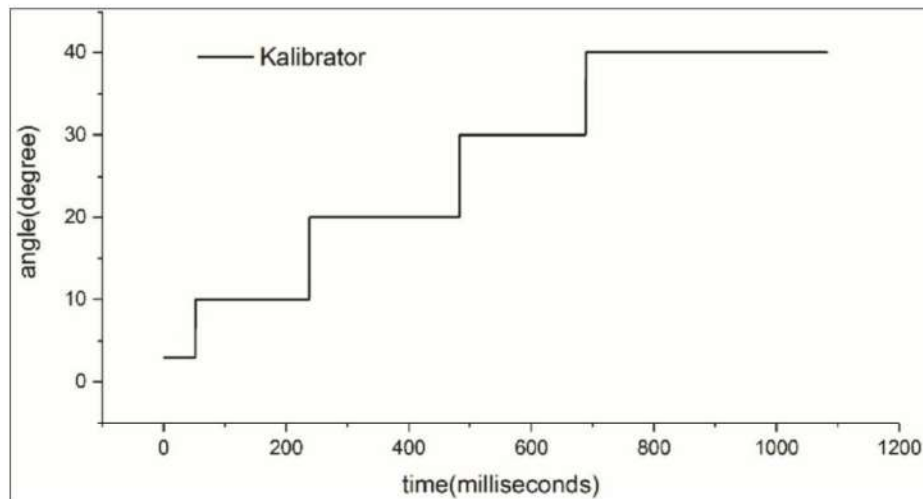
| Kalibrator (derajat) | Standar deviasi sudut terukur (derajat) | | Rata-rata sudut terukur (derajat) | |
|-------------------------|--------------------------------------------|--------------------------------------|--------------------------------------|--------------------------------------|
| | <i>Complementary filter</i> | <i>Complementary + Kalman Filter</i> | <i>Complementary filter</i> | <i>Complementary + Kalman Filter</i> |
| 10 | 0,640 | 0,156 | 9,594 | 9,610 |
| 20 | 0,686 | 0,161 | 19,804 | 19,832 |
| 30 | 0,616 | 0,053 | 30,211 | 30,223 |
| 40 | 0,541 | 0,083 | 40,511 | 40,515 |

Tabel 6.2 menunjukkan nilai hasil pengujian *sensor fusion* dengan rata-rata nilai dan standar deviasi yang dihasilkan dari 100 sampel data pada masing-masing variasi sudut *roll* dari 10^0 , 20^0 , 30^0 , dan 40^0 . Pengukuran dengan algoritme *complementary filter* pada sudut 30^0 menghasilkan nilai sudut rata-rata $30,211^0 \pm 0,616^0$, sedangkan dengan algoritme *complementary filter* dan *kalman*

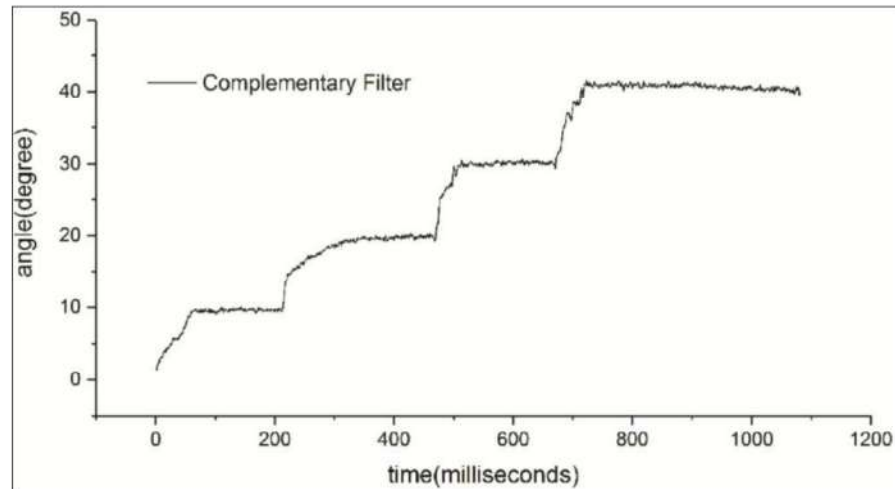


filter pada sudut 10^0 menghasilkan $30,223^0 \pm 0,053^0$. Pada sudut 40^0 *sensor fusion* dengan *complementary filter* menghasilkan rata-rata nilai sudut $40,511^0 \pm 0,541^0$ dan *complementary filter* ditambahkan dengan *kalman filter* menghasilkan nilai $40,515^0 \pm 0,083^0$. Berdasarkan hasil pengukuran yang telah diuji pada sudut *roll*, algoritme *complementary filter* tanpa *kalman filter* menghasilkan nilai standar deviasi pengukuran yang lebih besar dibandingkan dengan algoritme *complementary filter* dengan *kalman filter* sehingga algoritme *complementary filter* yang ditambahkan dengan *kalman filter* menghasilkan nilai pengukuran sudut yang lebih akurat.

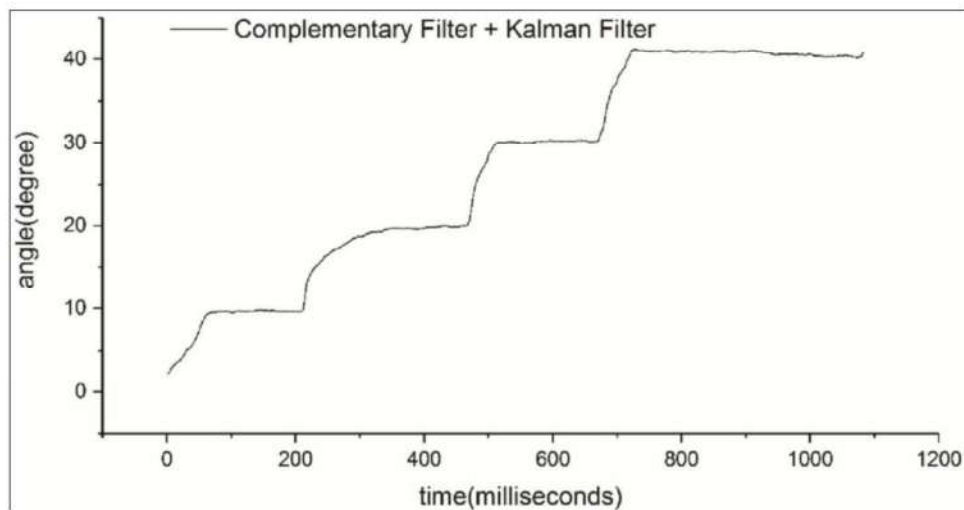
Hasil pengujian *sensor fusion* dapat dilihat pada grafik yang terdapat pada Gambar 6.4, Gambar 6.5 dan Gambar 6.6. Gambar 6.4 merupakan grafik dari kalibrator yang digunakan sebagai acuan untuk menentukan hasil pengukuran yang sebenarnya. Gambar 6.5 merupakan grafik hasil pengukuran sensor menggunakan algoritme *complementary filter*. Gambar 6.6 merupakan grafik hasil pengukuran sensor menggunakan algoritme *complementary filter* dan *kalman filter*. Berdasarkan ketiga gambar tersebut hasil pengukuran sensor dengan menggunakan algoritme *complementary filter* memiliki tingkat akurasi yang lebih rendah dibandingkan dengan hasil pengukuran sensor menggunakan algoritme *complementary filter* ditambahkan dengan *kalman filter*.



Gambar 6.4 Kalibrator pengujian *sensor fusion* pada sudut *roll*



Gambar 6.5 *Sensor fusion* dengan *complementary filter* pada sudut *roll*



Gambar 6.6 *Sensor fusion* dengan *kalman filter* pada sudut *roll*

Berdasarkan hasil pengujian *sensor fusion* pada sudut *pitch* dan sudut *roll*, *sensor fusion* dengan menggunakan *complementary filter* ditambahkan dengan *kalman filter* menghasilkan nilai yang lebih akurat. Hal tersebut dikarenakan *complementary filter* digunakan untuk menghasilkan nilai sudut *pitch* dan nilai sudut *roll* dengan mengkombinasikan hasil pengukuran akselerometer dan giroskop. Nilai *pitch* dan *roll* yang dihasilkan *complementary filter* memiliki tingkat akurasi yang belum sesuai dengan kebutuhan sistem, sehingga digunakanlah *kalman filter* sebagai estimator nilai pengukuran yang sebenarnya. *Kalman filter* sebagai estimator dalam hal ini berarti *kalman filter* dapat mengestimasi nilai



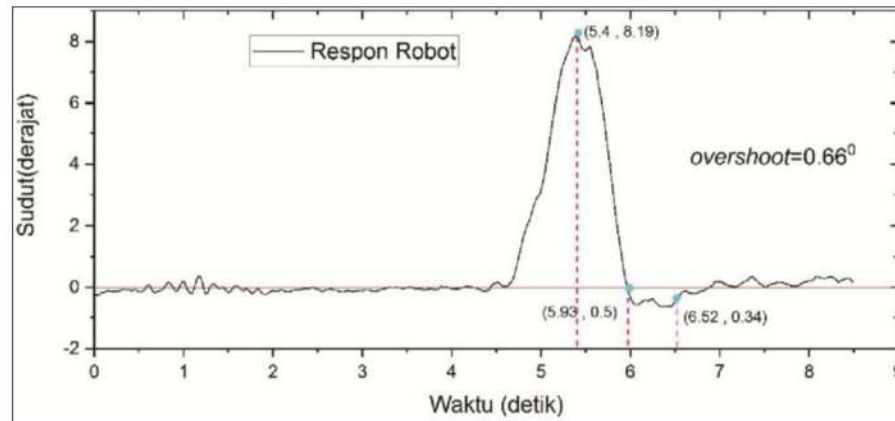
sesungguhnya dari pengukuran yang dilakukan oleh *complementary filter* dengan tingkat akurasi yang lebih baik. *Kalman filter* mengestimasi nilai pengukuran dengan cara mendapatkan nilai *gain*, nilai estimasi error, dan nilai estimasi sensor sesuai dengan persamaan yang terdapat pada sub bab 3.7. *Kalman Filter* sendiri mampu memberikan informasi yang mendekati keadaan sebenarnya ketika pengukuran bervariasi.

Kalman Filter dapat memperbaiki nilai estimasi dengan *Kalman gain*. Nilai *Kalman gain* ini berubah setiap waktu sesuai dengan nilai pengukuran dan nilai yang terprediksi. *Kalman gain* akan menaruh berat lebih pada nilai yang dapat dipercaya. Algoritme *Kalman Filter* tidak hanya mengandalkan informasi dari salah satu sensor saja sehingga algoritme ini dikatakan dapat mengisi kekurangan dari *sensor accelerometer* dengan kelebihan yang dimiliki *sensor gyroscope* begitu pula sebaliknya.

Oleh karena itu, jika dilihat dari segi kebutuhan sistem maka *complementary filter* ditambahkan dengan *kalman filter* lebih tepat untuk digunakan di dalam sistem dibandingkan dengan menggunakan *complementary filter* saja. Hal tersebut dikarenakan jika menggunakan *complementary filter* saja, nilai yang tidak akurat dan cenderung memiliki nilai kesalahan yang besar dapat memberikan dampak yang besar bagi robot walaupun pada kenyataannya robot dalam kondisi diam tanpa diberi gangguan. Berdasarkan hal tersebut, maka robot akan selalu merespon perubahan nilai sudut yang dihasilkan oleh *sensor fusion* menggunakan *complementary filter* yang dimana pada kenyataannya tidak ada gangguan yang diberikan pada robot. Berdasarkan permasalahan tersebut maka diperlukan nilai pengukuran hasil dari *sensor fusion* yang lebih stabil yang tidak memberikan nilai simpangan yang besar saat gangguan tidak diberikan. Dengan demikian, dapat disimpulkan bahwa hasil pengukuran menggunakan *sensor fusion* dengan *complementary filter* ditambahkan *kalman filter* lebih sesuai digunakan dibandingkan dengan menggunakan *complementary filter* saja.

6.2. Hasil Pengujian Keseimbangan Berdiri Robot Humanoid Pada Sudut *Pitch*

Pengujian keseimbangan terdiri dari beberapa tahap, meliputi pengujian penalaan nilai matriks Q untuk mendapatkan nilai *gain* K yang optimal, selanjutnya adalah menguji maksimum sudut kemiringan pada *pitch* dengan konstanta kecepatan sudut kemiringan yang divariaskan.



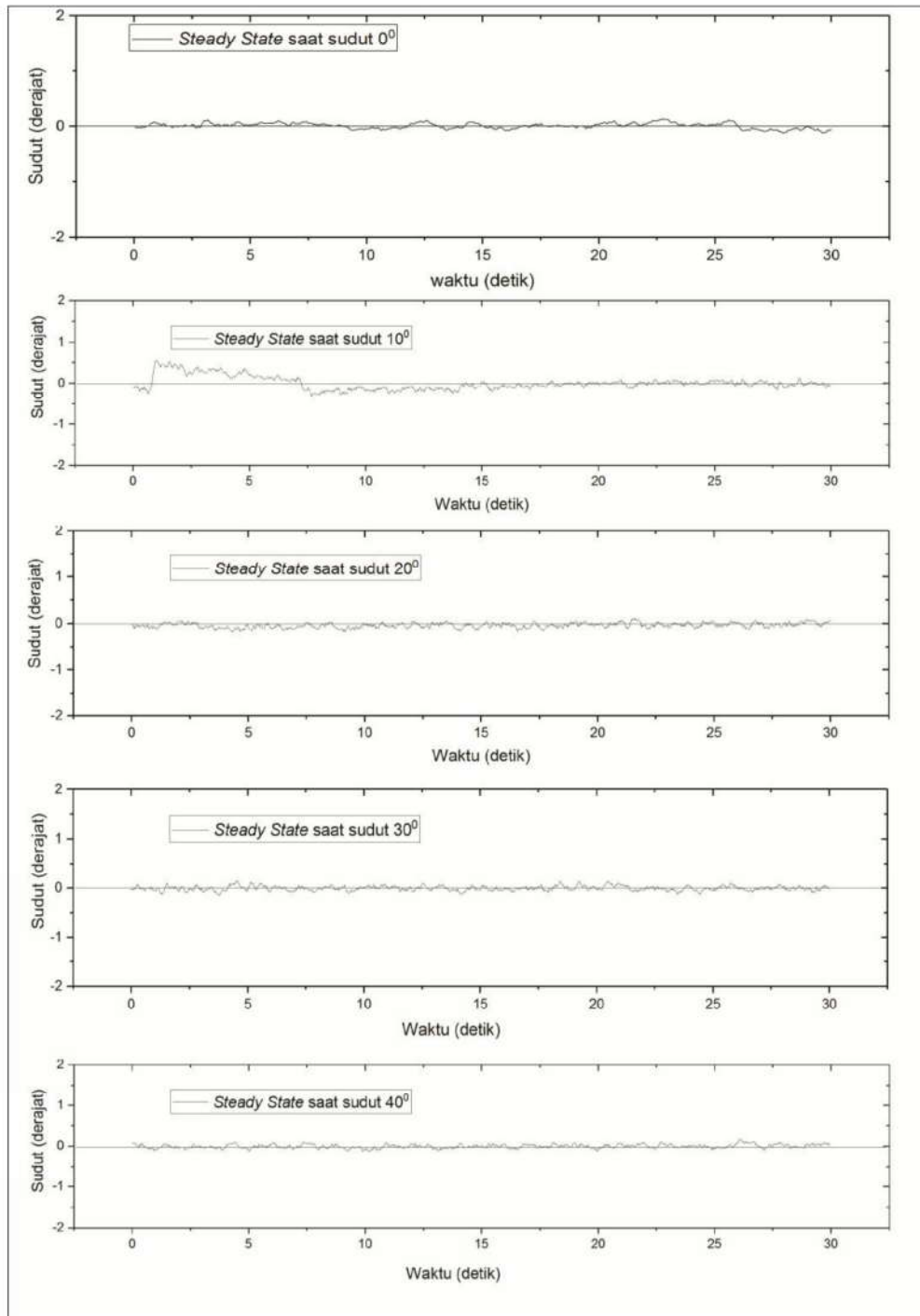
Gambar 6.7 Grafik hasil penalaan nilai Q pada sistem kendali robot untuk sudut putar *pitch*

Penalaan dilakukan terhadap matriks Q dengan $Q_{1,1}$ sebesar 2,3, $Q_{2,2}$ sebesar 1, $Q_{3,3}$ sebesar 1, $Q_{4,4}$ sebesar 1, dan R sebesar 1. Penalaan nilai Q menghasilkan nilai *gain* K yang kemudian menghasilkan respon optimal dari sistem seperti yang terdapat pada Gambar 6.7. Berdasarkan grafik yang terdapat pada Gambar 6.7, robot menghasilkan respon sistem dengan *risetime* 0,53 detik, *settling time* 1,12 detik, *steady-state error* $\pm 0,38^\circ$, dan *overshoot* sebesar $0,66^\circ$. Nilai $Q_{1,1}$ merupakan nilai yang mempengaruhi respon robot dalam mempertahankan keseimbangan robot pada sudut *pitch*. $Q_{1,1}$ menghasilkan nilai *gain* K yang dikalikan dengan *state* sudut *pitch*. Oleh karena itu, untuk mengoptimalkan respon robot dalam mempertahankan keseimbangan pada sudut *pitch* maka dilakukan penalaan pada $Q_{1,1}$ untuk menghasilkan *gain* K yang paling tepat. Hubungan antara nilai Q dan *gain* K terdapat pada persamaan Riccati pada konsep LQR yang terdapat pada sub bab 3.6.



Berdasarkan analisa sistem, karakteristik respon yang dihasilkan pada Gambar 6.7 menunjukkan hasil yang ideal dengan *risetime*, *settling time*, *overshoot*, dan *steady-state error* yang tidak melampaui batas dari kebutuhan sistem dalam merespon gangguan yang diberikan. Bahkan *risetime* yang dihasilkan hampir dua kali lebih cepat dari kebutuhan sistem. *Risetime* yang cepat dapat memberikan dampak *overshoot* yang besar, namun berdasarkan hasil yang didapatkan, sistem keseimbangan robot dapat mengatasi terjadinya *overshoot* yang besar dengan hasil yang masih jauh dibawah batas maksimal kebutuhan sistem berdasarkan pada Gambar 6.7. Hal tersebut dikarenakan kendali LQR yang diimplementasikan pada robot mampu meminimalkan *cost function* yang dibutuhkan untuk respon sistem sehingga robot dapat mempertahankan keseimbangannya saat berdiri diatas permukaan bidang miring.

Berdasarkan Gambar 6.7, *steady-state error* yang dihasilkan $\pm 0,38^{\circ}$ dimana nilai tersebut berada jauh dibawah batas maksimal kebutuhan sistem yaitu 2° . Hasil pengujian *steady-state-error* juga telah dilakukan dalam beberapa variasi sudut kemiringan permukaan seperti yang ditunjukkan pada Gambar 6.8 dan Tabel 6.3.



Gambar 6.8 Hasil *steady-state-error* dengan beberapa variasi sudut kemiringan permukaan pada sudut *pitch*

Tabel 6.3 Hasil *steady-state error* pada sudut *pitch*

| No. | Sudut kalibrator (derajat) | <i>steady-state error</i> (derajat) |
|-----|----------------------------|-------------------------------------|
| 1. | 0 | 0.13 |
| 2. | 10 | 0.55 |
| 3. | 20 | 0.17 |
| 4. | 30 | 0.15 |
| 5. | 40 | 0.18 |

Berdasarkan Tabel 6.3 dan Gambar 6.8 *steady-state error* yang dihasilkan oleh sistem kendali robot dalam semua sudut kemiringan memiliki nilai dibawah 0.20° kecuali pada sudut 10° . Berdasarkan kebutuhan sistem, *steady-state error* yang dihasilkan oleh sistem robot berada dibawah batas maksimal yaitu 2° dan jika nilai-nilai tersebut dirata-ratakan ($0,236^{\circ}$), maka dapat dikatakan bahwa *steady-state error* yang dihasilkan oleh sistem hampir 10 kali lipat lebih kecil daripada batas maksimal kebutuhan sistem.

Tabel 6.4 Uji coba kecepatan perubahan kemiringan pada sudut *pitch*

| No. | Kecepatan sudut (rad/s) | Sudut Maksimum 1 (derajat) | Sudut Maksimum 2 (derajat) | Sudut Maksimum 3 (derajat) | Sudut Maksimum 4 (derajat) | Sudut Maksimum 5 (derajat) | Rata-rata |
|-----|-------------------------|----------------------------|----------------------------|----------------------------|----------------------------|----------------------------|-----------|
| 1 | 0.04 | 44.99 | 44.04 | 47.91 | 45.05 | 47.9 | 45.978 |
| 2 | 0.07 | 39.52 | 44.47 | 46.33 | 36.67 | 38.33 | 41.064 |
| 3 | 0.08 | 38.4 | 39.23 | 34.12 | 40.01 | 37.36 | 37.824 |
| 4 | 0.11 | 29.73 | 28.72 | 35.72 | 32.66 | 32.27 | 31.82 |
| 5 | 0.12 | 31.22 | 30.74 | 30.74 | 27.8 | 30.31 | 30.162 |

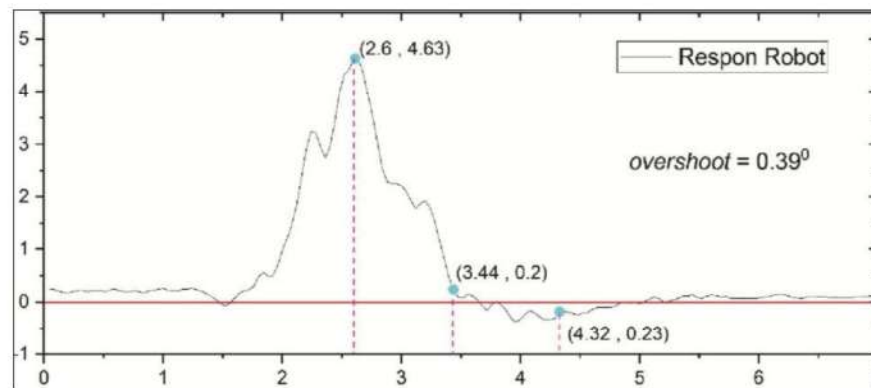
Faktor lain yang mempengaruhi keseimbangan robot dalam berdiri diatas permukaan miring adalah gangguan yang berupa kecepatan perubahan kemiringan permukaan. Oleh karena itu, maka dilakukan pengujian pada sistem keseimbangan robot dengan memberikan simpangan sudut kemiringan dalam kecepatan perubahan kemiringan yang konstan dan pengujian dilakukan sebanyak lima kali untuk masing-masing kecepatan perubahan sudut kemiringan.

Berdasarkan Tabel 6.4 robot dapat mempertahankan keseimbangannya terhadap kemiringan hingga sudut 40° pada saat kecepatan perubahan kemiringan 0.04 rad/s hingga 0.07 rad/s. Pada pengujian keempat dengan kecepatan perubahan kemiringan 0.08 rad/s robot dapat mempertahankan dirinya hingga kemiringan 40° , sehingga dapat dikatakan saat kecepatan 0.08 rad/s sistem robot *critically stable*.

Berdasarkan pengujian-pengujian respon yang dilakukan, dapat dikatakan bahwa robot humanoid memiliki sistem keseimbangan yang stabil dengan karakteristik yang tidak melampaui batas kebutuhan sistem.

6.3. Hasil Pengujian Keseimbangan Berdiri Robot Humanoid Pada Sudut *Roll*

Pengujian keseimbangan berdiri robot humanoid pada sudut *roll* memiliki tahapan yang sama dengan pengujian sudut *pitch*. Tahapan pengujian meliputi penalaan nilai matriks Q untuk menghasilkan respon sistem yang optimal dan pengujian kecepatan perubahan sudut kemiringan yang konstan untuk mengetahui batas kemampuan robot dalam mempertahankan keseimbangannya.

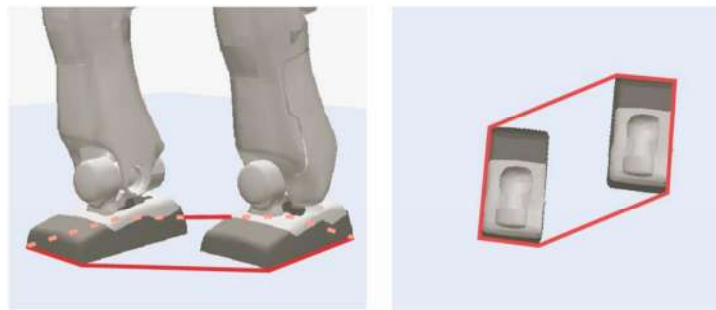


Gambar 6.9 Grafik hasil penalaan nilai Q pada sistem kendali robot untuk sudut putar *roll*

Penalaan dilakukan terhadap matriks Q dengan $Q_{1,1}$ sebesar 1, $Q_{2,2}$ sebesar 1, $Q_{3,3}$ sebesar 1,5, $Q_{4,4}$ sebesar 1 dan R sebesar 1. Kemudian didapatkan respon optimal dari sistem seperti yang terdapat pada Gambar 6.9. Berdasarkan grafik yang terdapat pada Gambar 6.9, robot menghasilkan respon sistem dengan *risetime* 0,84 detik, *settling time* 1,72 detik, *steady-state error* $\pm 0,26^0$, dan *overshoot* sebesar $0,39^0$. Seperti yang telah dilakukan pada sub bab 6.2, nilai $Q_{3,3}$ merupakan nilai yang mempengaruhi respon robot dalam mempertahankan keseimbangan robot pada sudut *roll*. $Q_{3,3}$ menghasilkan nilai *gain* K yang dikalikan dengan *state* sudut *roll*. Oleh karena itu, untuk mengoptimalkan respon robot dalam mempertahankan keseimbangan pada sudut *roll* maka dilakukan penalaan pada $Q_{3,3}$ untuk

menghasilkan *gain K* yang paling tepat. Hubungan antara nilai *Q* dan *gain K* terdapat pada persamaan Riccati pada konsep LQR yang terdapat pada sub bab 3.6.

Karakteristik respon yang dihasilkan pada Gambar 6.9 menunjukkan bahwa *risetime*, *settling time*, *steady-state error*, dan *overshoot* masih di dalam batas maksimal dari kebutuhan sistem. *Overshoot* yang dihasilkan lebih kecil dari seperempat maksimal *overshoot* yang dibutuhkan oleh sistem. Jika dibandingkan dengan sistem keseimbangan pada sudut *pitch*, sistem keseimbangan pada sudut *roll* memiliki *overshoot* yang lebih kecil. Hal tersebut disebabkan karena faktor redaman yang berasal dari kaki robot yang menopang tubuh robot jika dilihat dari bidang koronal membentuk *support polygon* yang lebih lebar dibandingkan saat robot dilihat dari bidang sagital. *Support polygon* adalah wilayah yang terbentuk dengan menyertakan semua titik kontak antara kaki robot dengan permukaan sehingga membentuk luasan yang menentukan keseimbangan robot (Kajita et al., 2014). *Support polygon* digambarkan pada Gambar 6.10 dan Gambar 6.11.



Gambar 6.10 *Support Polygon* dengan semua bagian telapak kaki menapak di atas permukaan (Kajita et al., 2014)



Gambar 6.11 *Support Polygon* dengan sebagian telapak kaki menapak di atas permukaan (Kajita et al., 2014)

Berdasarkan penjelasan tentang *support polygon* tersebut, dapat dikatakan bahwa saat robot mempertahankan keseimbangannya pada sudut *roll* faktor



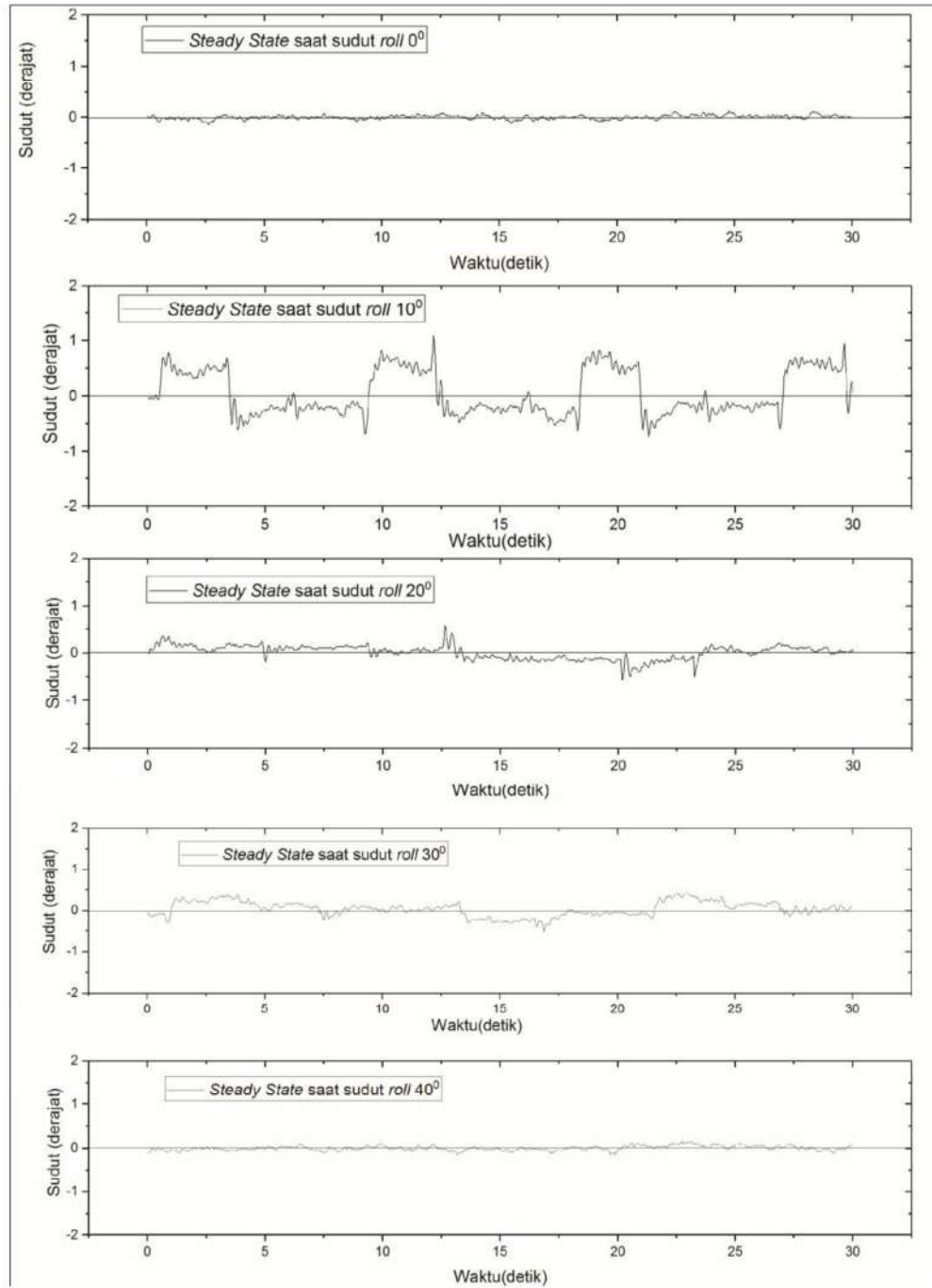
redaman yang berlaku lebih besar daripada saat robot mempertahankan keseimbangannya pada sudut *pitch*.

Berdasarkan Gambar 6.12 dan Tabel 6.5 Hasil *steady-state error* pada sudut *roll* dapat diketahui *steady-state error* yang dihasilkan oleh robot saat sudut kemiringan permukaan $0^0, 10^0, 20^0, 30^0$ dan 40^0 . Jika dibandingkan dengan respon robot saat mempertahankan keseimbangan pada sudut *pitch*, *steady-state error* yang dihasilkan saat robot mempertahankan keseimbangan pada sudut *roll* lebih besar. Hal tersebut dikarenakan aktuator yang digerakkan saat terjadi perubahan sudut permukaan lebih banyak daripada saat robot mempertahankan keseimbangan pada sudut *pitch*. Saat mempertahankan keseimbangan pada sudut *roll*, robot menggerakkan 8 sendi robot sedangkan saat mempertahankan keseimbangan pada sudut *pitch* sendi yang digerakkan ada 2. Perbedaan banyaknya aktuator yang digerakkan dikarenakan sistem pergerakan yang dilakukan robot disesuaikan dengan model yang digunakan.

Walaupun *steady-state error* yang dihasilkan oleh robot saat mempertahankan keseimbangan pada sudut *roll* lebih besar dari pada saat pada *pitch*, *steady-state error* yang dihasilkan masih di dalam batas kebutuhan sistem. Oleh karena itu, sistem keseimbangan robot dalam mempertahankan keadaan saat berdiri diatas permukaan miring pada sudut *roll* memiliki sistem yang stabil dengan rata-rata *steady-state error* $0,5^0$ dan nilai tertinggi $1,09^0$. Hal tersebut dikarenakan kendali LQR yang diimplementasikan pada robot mampu meminimalkan *cost function* yang dibutuhkan untuk respon sistem sehingga robot dapat mempertahankan keseimbangan berdiri diatas permukaan bidang miring. Tabel 6.5 Hasil *steady-state error* pada sudut *roll* dan Gambar 6.12 menunjukkan hasil *steady-state error* pada sudut *roll*.

Tabel 6.5 Hasil *steady-state error* pada sudut *roll*

| No. | Sudut kalibrator (derajat) | <i>steady-state error</i> (derajat) |
|-----|----------------------------|-------------------------------------|
| 1. | 0 | 0.14 |
| 2. | 10 | 1.09 |
| 3. | 20 | 0.57 |
| 4. | 30 | 0.54 |
| 5. | 40 | 0.16 |



Gambar 6.12 Hasil *steady-state error* dengan beberapa variasi sudut kemiringan permukaan pada sudut *roll*



Seperti yang telah dilakukan pada saat menguji sistem keseimbangan robot pada sudut *pitch*, pengujian keseimbangan robot terhadap kecepatan perubahan kemiringan dengan nilai kecepatan yang konstan dilakukan pada sudut *roll*. Hasil pengujian yang dilakukan terdapat pada Tabel 6.6 Uji coba kecepatan perubahan kemiringan pada sudut *roll*.

Tabel 6.6 Uji coba kecepatan perubahan kemiringan pada sudut *roll*

| No. | Kecepatan sudut (rad/s) | Sudut Maksimum 1 (derajat) | Sudut Maksimum 2 (derajat) | Sudut Maksimum 3 (derajat) | Sudut Maksimum 4 (derajat) | Sudut Maksimum 5 (derajat) | Rata-rata |
|-----|-------------------------|----------------------------|----------------------------|----------------------------|----------------------------|----------------------------|-----------|
| 1 | 0.04 | 40.22 | 46.04 | 44.56 | 43.81 | 41.97 | 43.32 |
| 2 | 0.07 | 40.29 | 45.09 | 46.8 | 45.75 | 40.8 | 43.746 |
| 3 | 0.08 | 41.72 | 46.57 | 46.72 | 45.07 | 40.71 | 44.158 |
| 4 | 0.11 | 45.73 | 48.35 | 45.68 | 49.94 | 42.58 | 46.456 |
| 5 | 0.12 | 45.3 | 46.03 | 45.22 | 40.91 | 45.79 | 44.65 |

Berdasarkan Tabel 6.6 dapat diketahui bahwa robot humanoid dapat mempertahankan keseimbangannya pada kecepatan perubahan kemiringan 0.04 rad/s hingga 0.12 rad/s. Hasil tersebut menunjukkan bahwa pada saat terjadi perubahan kemiringan pada sudut *roll*, robot dapat mempertahankan keseimbangannya terhadap kecepatan perubahan kemiringan yang lebih cepat daripada saat robot mempertahankan keseimbangannya pada sudut *pitch*. Hal tersebut juga disebabkan karena faktor redaman seperti yang telah digambarkan pada Gambar 6.10 Gambar 6.11, dimana kaki robot yang menopang tubuh robot jika dilihat dari bidang koronal membentuk *support polygon* yang lebih lebar dibandingkan saat robot dilihat dari bidang sagital.

BAB VII

PENUTUP

7.1. Kesimpulan

Berdasarkan penelitian tentang sistem kendali keseimbangan berdiri robot humanoid pada permukaan bidang miring yang telah dilakukan menggunakan kendali LQR, maka dapat disimpulkan bahwa:

1. Robot mampu mempertahankan keseimbangannya saat berdiri pada permukaan bidang miring menggunakan konsep LQR.
2. Robot dapat mempertahankan keseimbangannya pada sudut *pitch* dan sudut *roll* dengan nilai *risetime*, *settling time*, *steady-state error*, dan *overshoot* yang sesuai dengan kebutuhan spesifikasi sistem.
3. Robot dapat mempertahankan keseimbangannya pada sudut *roll* dengan kecepatan perubahan kemiringan 0,12 rad/s.
4. Robot dapat mempertahankan keseimbangannya pada sudut *pitch* dengan kecepatan perubahan kemiringan 0,07 rad/s.
5. Robot humanoid mampu berdiri diatas permukaan bidang miring dengan sudut kemiringan maksimal 40° pada sudut *roll* dan *pitch*.

7.2. Saran

Saran-saran yang dapat digunakan untuk menyempurnakan penelitian selanjutnya adalah sebagai berikut.

1. Diperlukan pemodelan sistem dinamis robot yang direpresentasikan dalam ruang tiga dimensi sehingga robot humanoid dapat mempertahankan keseimbangannya jika gangguan yang diberikan adalah gangguan pada sudut *pitch* dan *roll*.
2. Perlu ditambahkan pemodelan yang memungkinkan robot untuk menggerakkan tangan sehingga dapat membantu robot dalam merespon perubahan kemiringan dalam mempertahankan keseimbangannya.

DAFTAR PUSTAKA

- Ahmed, S.M., Chew, C.M. dan Tian, B., 2013, Standing Posture Modeling and Control for a Humanoid Robot, *International Conference on Intelligent Robots and Systems*, 4152–4157,
- Akhtaruzzaman, M. dan Shafie, A.A., 2011, Geometrical analysis on BIOLOID humanoid system standing on single leg, *2011 4th International Conference on Mechatronics: Integrated Engineering for Industrial and Societal Development, ICOM'11 - Conference Proceedings*, [Online] (May), 17–19, tersedia di DOI:10.1109/ICOM.2011.5937193.
- Braun, J., 2012, *Formulae Handbook*, [Online] tersedia di academy.maxonmotor.ch.
- Deng-Peng, X. dan Xu, L., 2011, Multiple Balance Strategies for Humanoid Standing Control, *Acta Automatica Sinica*, [Online] 37 (2), 228–233, tersedia di DOI:10.1016/S1874-1029(11)60204-8.
- Duindam, V. dan Stramigioli, S., 2009, *Modeling and Control for Efficient Bipedal Walking Robots*, [Online]. tersedia di DOI:10.1007/978-3-540-89918-1.
- Hibbeler, R.C., 2016, *Dynamics*.
- Hong, S., 2008, Walking pattern generation for Humanoid robots with LQR and feedforward control method, *2008 34th Annual Conference of IEEE Industrial Electronics*, [Online] 1698–1703, tersedia di DOI:10.1109/IECON.2008.4758209.
- Iswara, P.J. dan Putra, A.E., 2012, Sistem Kontrol Keseimbangan Statis Robot Humanoid Joko Klana Berbasis Pengontrol PID, *Indonesian Journal of Electronics and Instrumentation Systems*, 2 (1), 67–76,
- Jazar, R.N., 2010, *Theory of Applied Robotics: Kinematics, Dynamics, and Control*, [Online]. tersedia di DOI:10.1007/978-1-4419-1750-8.
- Kajita, S., Hirukawa, H., Harada, K. dan Yokoi, K., 2014, *Introduction to Humanoid Robotics*, Bruno Siciliano dan Khatib Oussama (ed.),

Springer Berlin Heidelberg, Heidelberg.

Kajita, S., Kanehiro, F., Kaneko, K. dan Fujiwara, K., 2003, *Biped Walking Pattern Generation by using Preview Control of Zero-Moment Point*, 1620–1626,

Kajita, S., Kanehiro, F., Kaneko, K., Yokoi, K. dan Hirukawa, H., 2001, The 3D linear inverted pendulum mode: a simple modeling for a biped walking pattern generation, *Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems. Expanding the Societal Role of Robotics in the the Next Millennium (Cat. No.01CH37180)*, [Online], 2001 IEEE., hal. 239–246, tersedia di DOI:10.1109/IROS.2001.973365.

Kandepu, R., Foss, B. dan Imsland, L., 2008, Applying the unscented Kalman filter for nonlinear state estimation, *Journal of Process Control*, [Online] 18 (7–8), 753–768, tersedia di DOI:10.1016/j.jprocont.2007.11.004.

Liberzon, D., 2012, *Calculus of Variations and Optimal Control Theory*, Princeton University Press, New Jersey.

Ogata, K., 2009, *Modern Control Engineering*, Fifth, Prentice Hall, New Jersey.

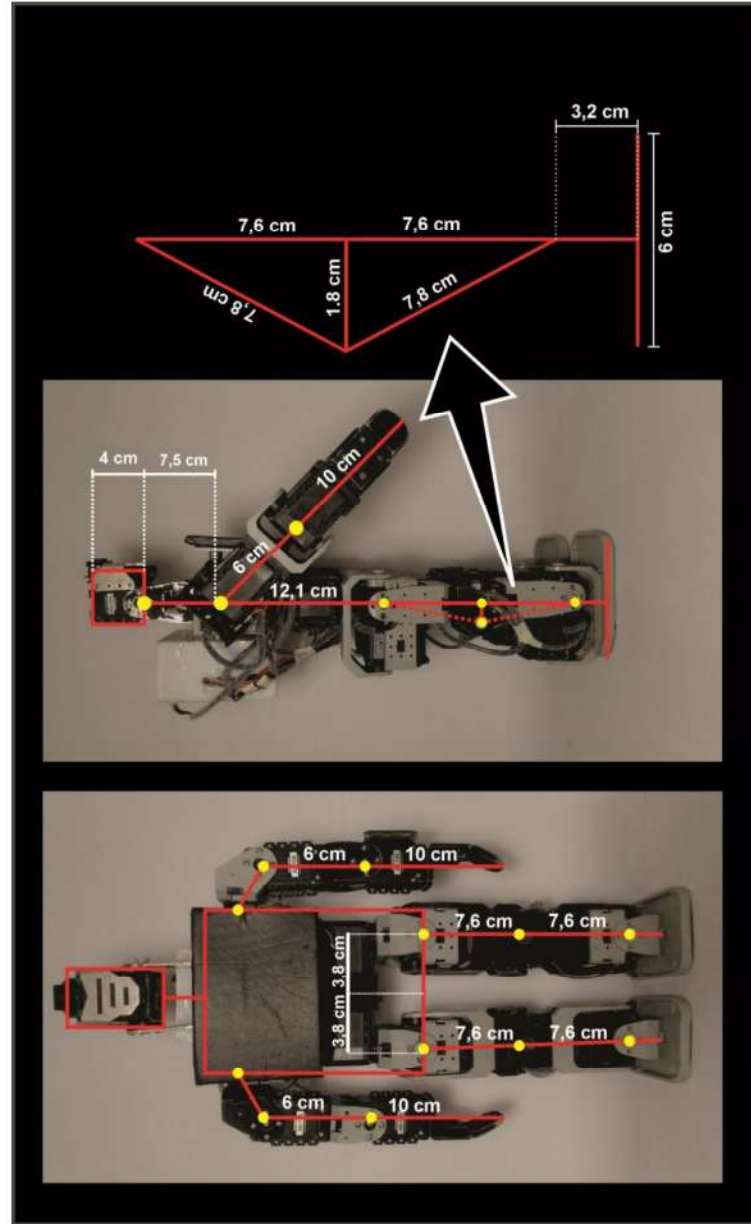
Shi, G., Dong, C., Zhang, T., Liu, H., Su, H., Wang, J. dan Wang, Z., 2016, Improvement of Human Postural Stability Criterion using ZMP Simplification and Optimization Algorithms, *IEEE International Conference on Cyber Technology in Automation, Control and Intelligent Systems*, 360–364,

Starlino, 2009, A Guide to Using IMU (Accelerometer and Gyroscope Devices) in Embedded Applications, [Online], tersedia di http://www.starlino.com/imu_guide.html, diakses 20 Mei 2017.

Wicaksono, R., Priyatmadi, Wahyunggoro, O., Majid, I. dan Purnama, D., 2014, Sistem Kendali Step-motion Robot Humanoid 20 Derajat Kebebasan Dalam Mengikuti Suatu Warna, *Indonesian Symposium on Robot Soccer Competition*, 66–73,

LAMPIRAN

Lampiran 1 Gambar dimensi robot



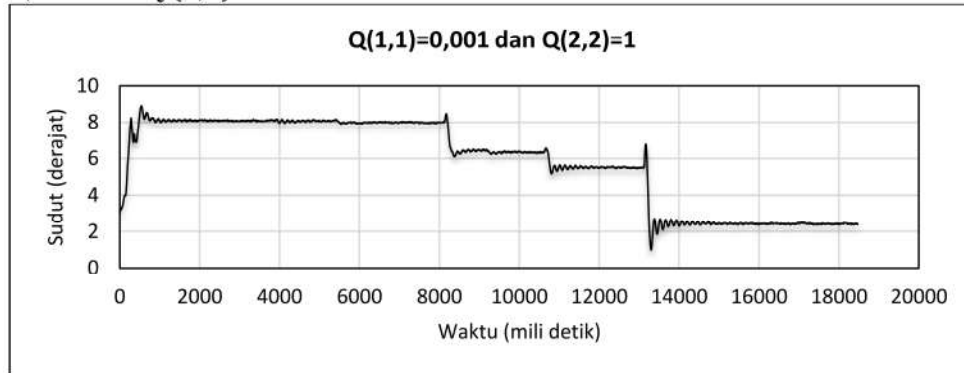


Lampiran 2 Data Pengukuran Dimensi dan Perhitungan Inersia

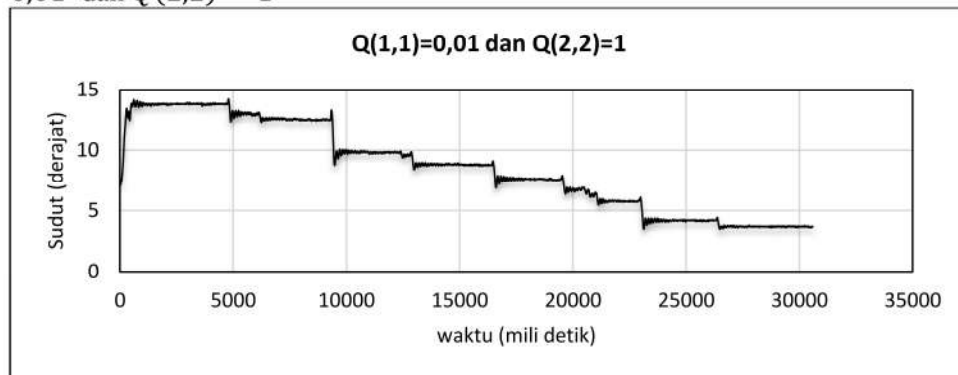
| Jumlah | Nama | Massa (Kg) | X | Y | Z | Panjang (meter) | Lebar (meter) | Tinggi (Meter) | Ixx (Kg ^m ²) | Iyy (Kg ^m ²) | Izz (Kg ^m ²) |
|--------|--------|------------|----------|----------|----------|-----------------|---------------|----------------|-------------------------------------|-------------------------------------|-------------------------------------|
| 2 | Tangan | 0.1620 | 0.019175 | 0.07425 | 0.023275 | 0.03835 | 0.03175 | 0.19185 | 0.002995 | 0.001316 | 0.001972 |
| 1 | Badan | 0.3630 | 0.015875 | 0 | 0.018675 | 0.0441 | 0.10985 | 0.10795 | 0.000538 | 0.000936 | 0.000515 |
| 2 | Angkle | 0.1720 | 0 | 0.0348 | 0.215945 | 0.0704 | 0.04025 | 0.04965 | 0.016671 | 0.016159 | 0.000605 |
| 1 | Kepala | 0.1020 | 0.031975 | 0.019 | 0.111925 | 0.04715 | 0.038 | 0.0499 | 0.001355 | 0.001415 | 0.000172 |
| 2 | betis | 0.1120 | 0.00875 | 0.035425 | 0.14342 | 0.0499 | 0.039 | 0.0422 | 0.004968 | 0.004686 | 0.000373 |
| 2 | Paha | 0.1570 | 0 | 0.036025 | 0.07445 | 0.0735 | 0.0378 | 0.04965 | 0.002354 | 0.001842 | 0.000586 |
| 1 | shield | 0.0140 | 0.066 | 0.04455 | 0.033275 | 0.002 | 0.0891 | 0.0665 | 4.85E-05 | 9.09E-05 | 9.8E-05 |
| 1 | Box | 0.1280 | 0.02955 | 0.033225 | 0.05645 | 0.0462 | 0.06645 | 0.089 | 0.000656 | 0.000651 | 0.000323 |
| 1 | power | 1.1280 | 0.0119 | 0.0525 | 0.0178 | 0.0238 | 0.105 | 0.0356 | 0.029586 | 0.027096 | 0.004645 |

Lampiran 3 Data Hasil Penalaan Matriks Q pada sudut *pitch*

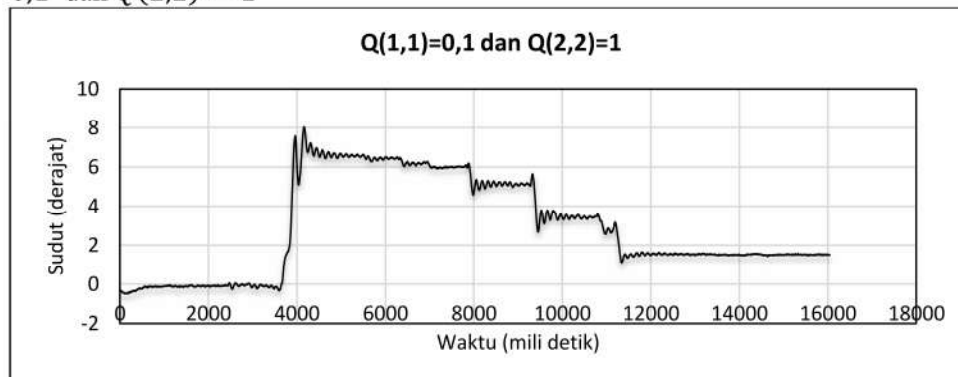
1. Grafik Respon robot pada sudut pitch dengan hasil penalaan $Q(1,1) = 0,001$ dan $Q(2,2) = 1$



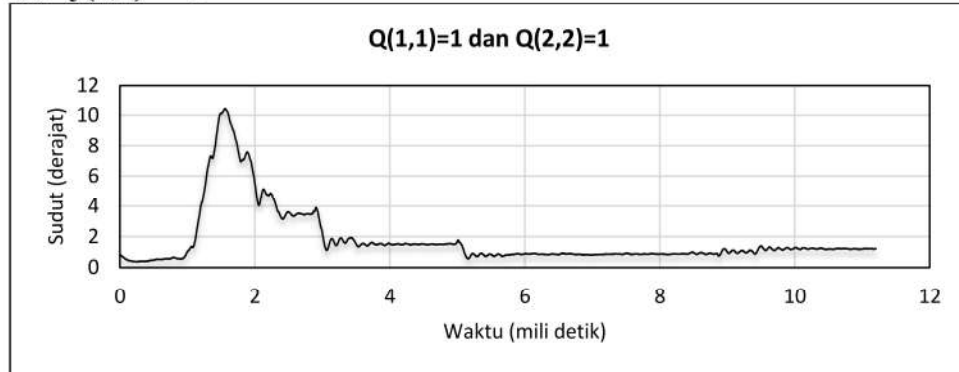
2. Grafik Respon robot pada sudut pitch dengan hasil penalaan $Q(1,1) = 0,01$ dan $Q(2,2) = 1$



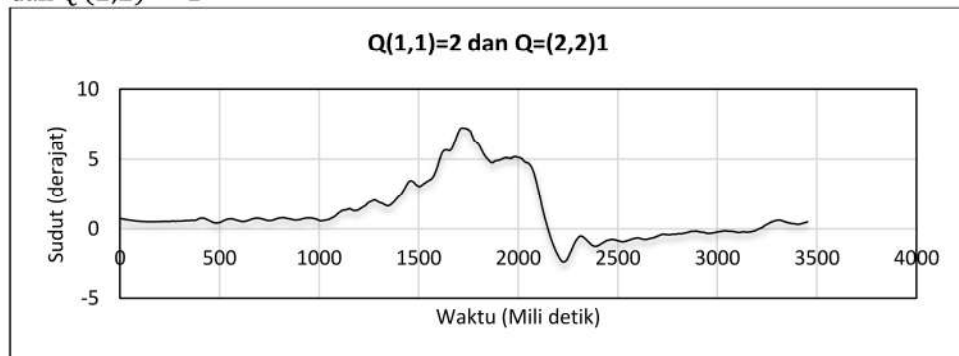
3. Grafik Respon robot pada sudut pitch dengan hasil penalaan $Q(1,1) = 0,1$ dan $Q(2,2) = 1$



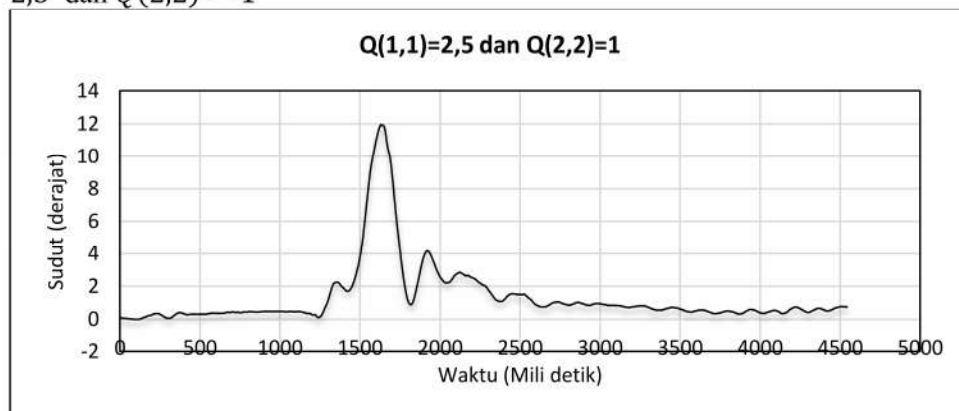
4. Grafik Respon robot pada sudut pitch dengan hasil penalaan $Q(1,1) = 1$ dan $Q(2,2) = 1$



5. Grafik Respon robot pada sudut pitch dengan hasil penalaan $Q(1,1) = 2$ dan $Q(2,2) = 1$

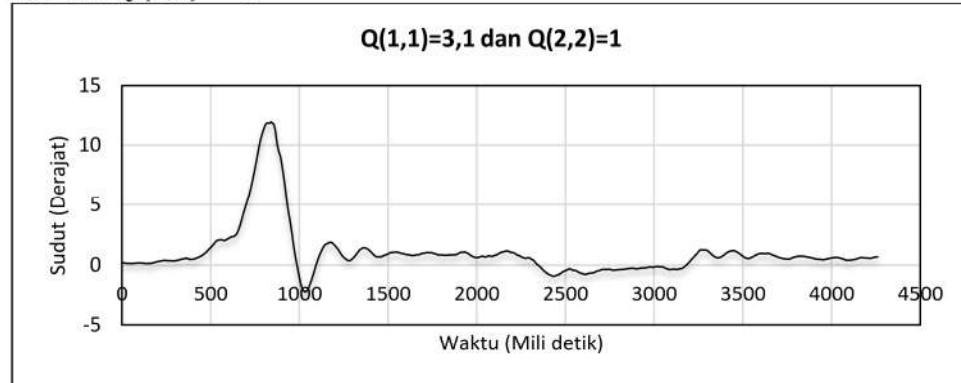


6. Grafik Respon robot pada sudut pitch dengan hasil penalaan $Q(1,1) = 2,5$ dan $Q(2,2) = 1$

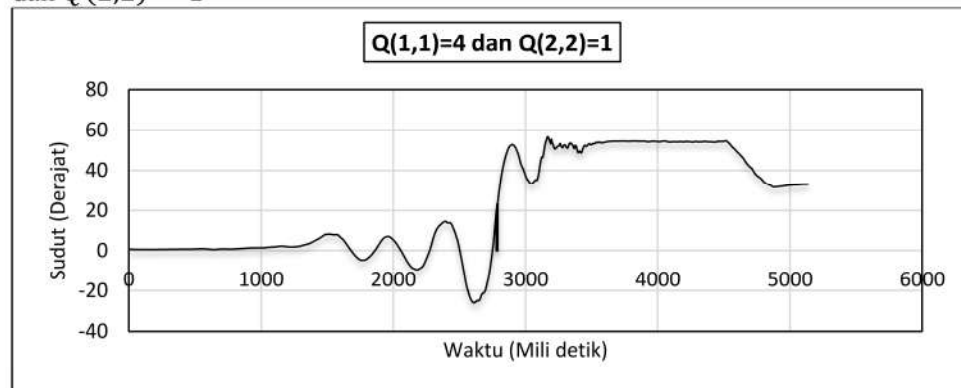




7. Grafik Respon robot pada sudut pitch dengan hasil penalaan $Q(1,1) = 3,1$ dan $Q(2,2) = 1$



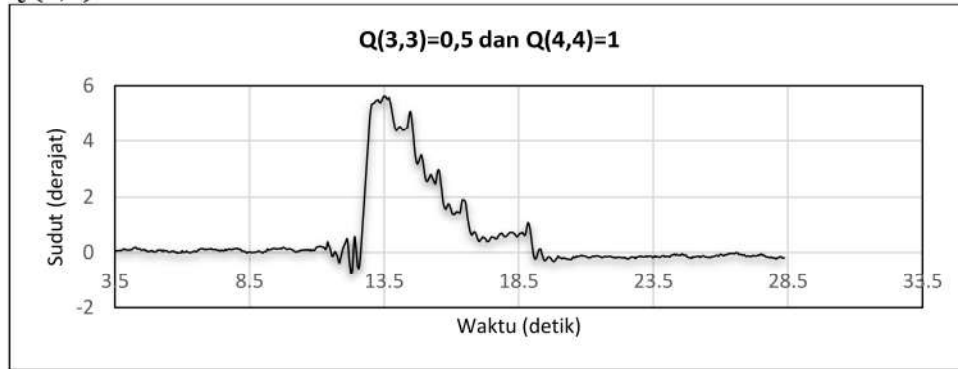
8. Grafik Respon robot pada sudut pitch dengan hasil penalaan $Q(1,1) = 4$ dan $Q(2,2) = 1$



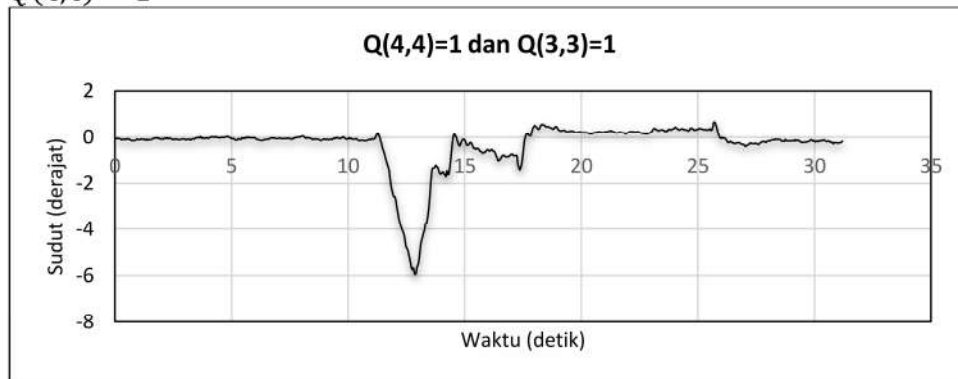


Lampiran 4 Data Hasil Penalaan Matriks Q pada sudut *roll*

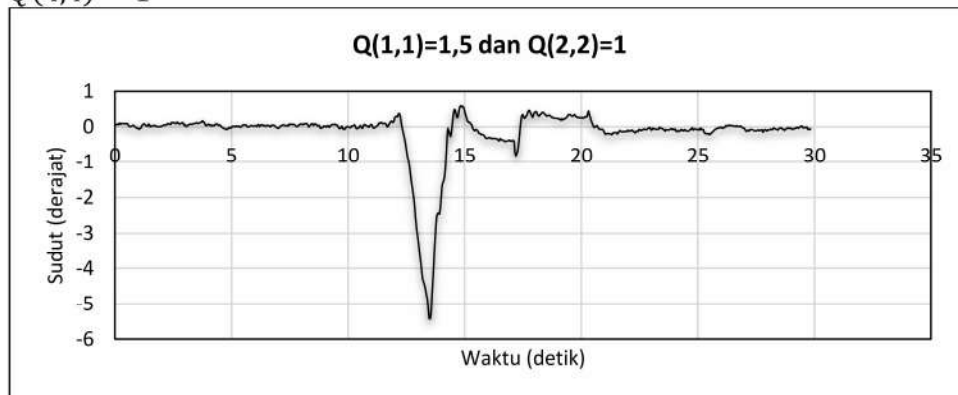
1. Grafik Respon robot pada sudut *roll* dengan hasil penalaan $Q(3,3) = 0,5$ dan $Q(4,4) = 1$



2. Grafik Respon robot pada sudut *roll* dengan hasil penalaan $Q(3,3) = 1$ dan $Q(4,4) = 1$

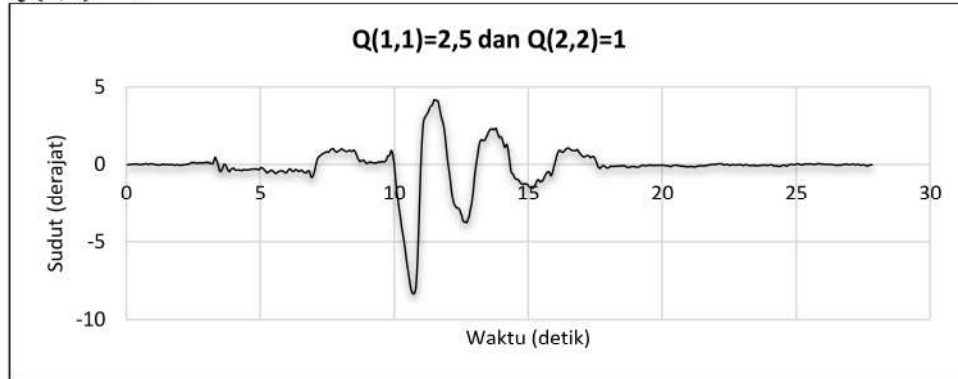


3. Grafik Respon robot pada sudut *roll* dengan hasil penalaan $Q(3,3) = 1,5$ dan $Q(4,4) = 1$

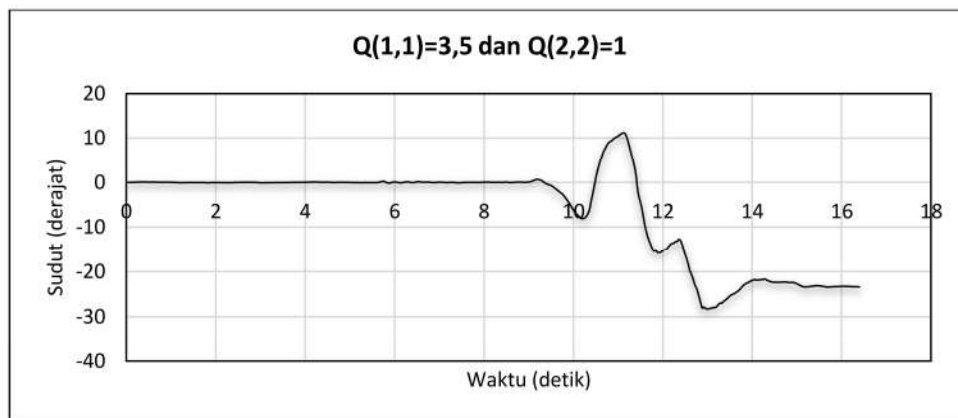




4. Grafik Respon robot pada sudut *roll* dengan hasil penalaan $Q(3,3) = 2,5$ dan $Q(4,4) = 1$



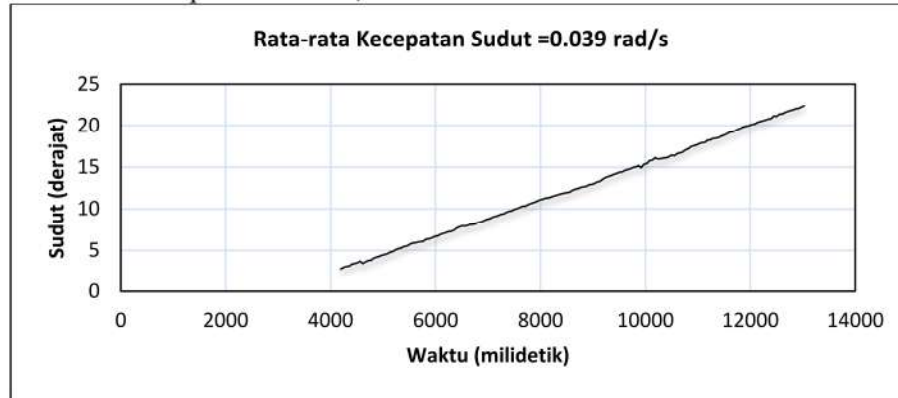
5. Grafik Respon robot pada sudut *roll* dengan hasil penalaan $Q(3,3) = 1$ dan $Q(4,4) = 1$



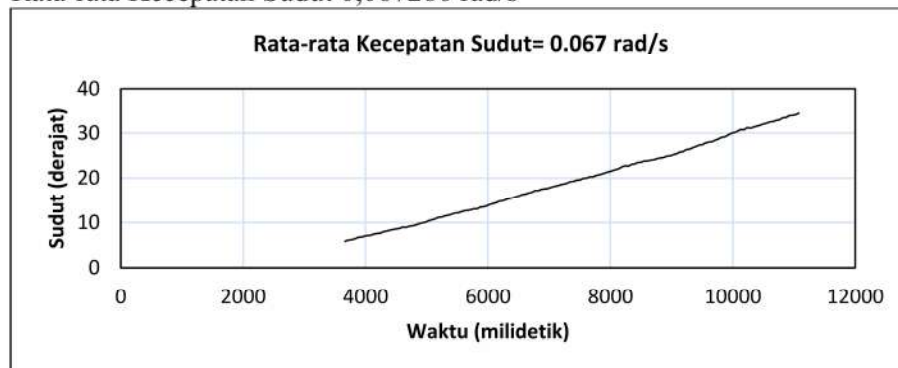


Lampiran 5 Grafik Hasil Uji Coba Kecepatan Perubahan Kemiringan

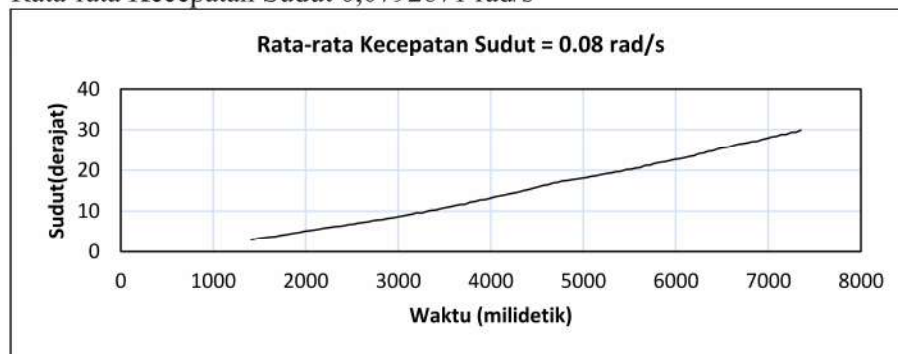
1. Grafik Perubahan Sudut Kemiringan Permukaan Terhadap Waktu Dengan Rata-rata Kecepatan Sudut 0,038826 rad/s



2. Grafik Perubahan Sudut Kemiringan Permukaan Terhadap Waktu Dengan Rata-rata Kecepatan Sudut 0,067286 rad/s

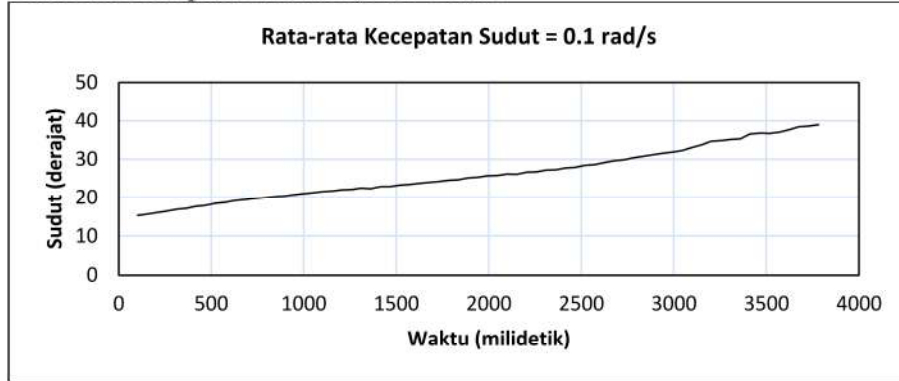


3. Grafik Perubahan Sudut Kemiringan Permukaan Terhadap Waktu Dengan Rata-rata Kecepatan Sudut 0,0792671 rad/s

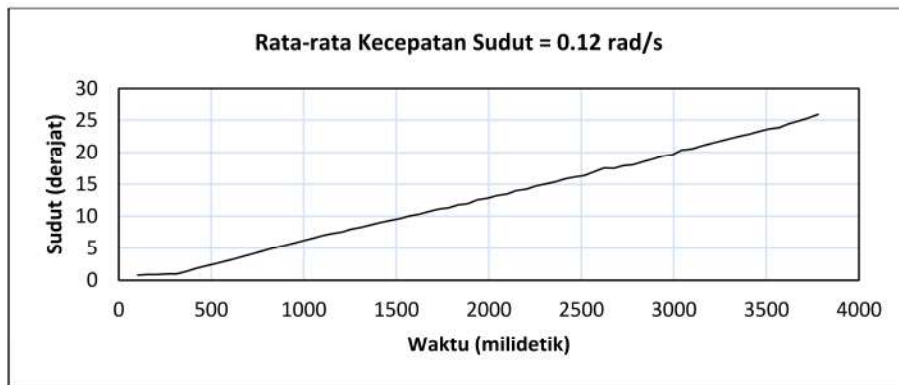




4. Grafik Perubahan Sudut Kemiringan Permukaan Terhadap Waktu Dengan Rata-rata Kecepatan Sudut 0,11197 rad/s



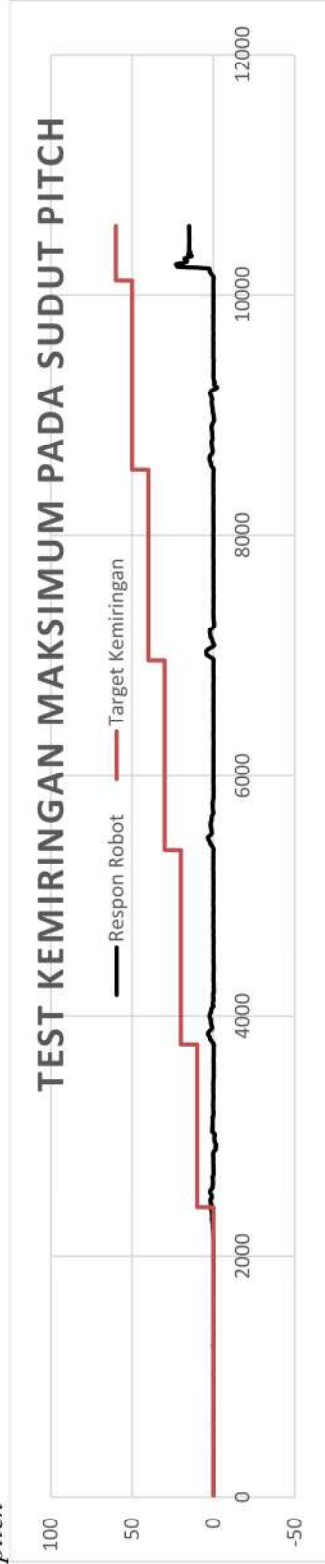
5. Grafik Perubahan Sudut Kemiringan Permukaan Terhadap Waktu Dengan Rata-rata Kecepatan Sudut 0,11905 rad/s





Lampiran 6 Uji respon robot dengan kemiringan yang diberikan secara bertahap

1. Grafik Respon Robot dengan gangguan yang diberikan secara bertahap untuk menguji sudut maksimal yang bisa ditangani pada sudut *pitch*



2. Grafik Respon Robot dengan gangguan yang diberikan secara bertahap untuk menguji sudut maksimal yang bisa ditangani pada sudut *roll*

