



**UNIVERSITAS INDONESIA**

**RANCANG BANGUN SISTEM KENDALI QUADROTOR  
UNTUK KESETIMBANGAN POSISI DENGAN PID**

**SKRIPSI**

**NUR HIDAYAT  
04 05 03 0621**

**FAKULTAS TEKNIK  
PROGRAM STUDI TEKNIK ELEKTRO  
DEPOK  
JUNI 2009**



**UNIVERSITAS INDONESIA**

**RANCANG BANGUN SISTEM KENDALI QUADROTOR  
UNTUK KESETIMBANGAN POSISI DENGAN PID**

**SKRIPSI**

Diajukan Untuk Melengkapi Sebagian Persyaratan Menjadi Sarjana  
Teknik

**NUR HIDAYAT  
04 05 03 0621**

**HALAMAN PERNYATAAN ORISINALITAS**

**Skripsi ini adalah hasil karya saya sendiri,  
dan semua sumber baik yang dikutip maupun dirujuk  
telah saya nyatakan dengan benar.**

**Nama : Nur Hidayat**

**NPM : 0405030621**

**Tanda Tangan :**

**Tanggal : 17 Juni 2009**

## HALAMAN PENGESAHAN

Skripsi ini diajukan oleh :

Nama : Nur Hidayat

NPM : 0405030621

Program Studi : Teknik Elektro

Judul Skripsi : Rancang Bangun Sistem Kendali Quadrotor Untuk  
Keseimbangan Posisi Dengan PID

**Telah berhasil dipertahankan di hadapan Dewan Penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Sarjana Teknik pada Program Studi Teknik Elektro, Fakultas Teknik, Universitas Indonesia**

### DEWAN PENGUJI

Pembimbing : Dr. Abdul Muis ST, MSc. ( )

Penguji : Ir. Wahidin Wahab Msc, PhD. ( )

Penguji : Dr. Ir. Ridwan Gunawan MT ( )

Ditetapkan di : Depok

Tanggal : 13 Juli 2009

## UCAPAN TERIMAKASIH

Puji syukur saya panjatkan kepada Allah SWT, karena atas berkat dan rahmat-Nya, saya dapat menyelesaikan skripsi ini. Penyusunan skripsi ini dilakukan dalam rangka memenuhi salah satu syarat untuk mencapai gelar Sarjana Teknik Departemen Teknik Elektro pada Fakultas Teknik Universitas Indonesia. Saya menyadari bahwa tanpa bantuan dan bimbingan dari berbagai pihak, baik dari masa perkuliahan sampai pada penyusunan skripsi ini sangatlah sulit bagi saya untuk menyelesaikan skripsi ini. Untuk karena itu, saya mengucapkan terima kasih kepada :

- (1) Dr. Abdul Muis, ST, M.Eng, selaku dosen pembimbing yang telah menyediakan waktu, tenaga dan pikiran didalam mengarahkan saya dalam penyusunan skripsi ini;
- (2) orangtua dan keluarga saya yang telah memberikan bantuan dukungan material maupun moril; dan
- (3) sahabat yang telah banyak membantu penulis dalam menyelesaikan skripsi ini.

Akhir kata, saya berharap Allah SWT berkenan membalas segala kebaikan semua pihak yang telah membantu. Semoga skripsi ini membawa manfaat bagi pengembangan ilmu.

Depok, 17 Juni 2009

Penulis

**HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI  
TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS**

---

Sebagai sivitas akademik Universitas Indonesia, saya yang bertanda tangan di bawah ini:

Nama : Nur Hidayat  
NPM : 0405030621  
Program Studi : Teknik Elektro  
Departemen : Teknik Elektro  
Fakultas : Teknik  
Jenis karya : Skripsi

demikian demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Indonesia **Hak Bebas Royalti Noneksklusif (*Non-exclusive Royalty-Free Right*)** atas karya ilmiah saya yang berjudul :

Rancang Bangun Sistem Kendali Quadrotor  
Untuk Keseimbangan Posisi Dengan PID

beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Noneksklusif ini Universitas Indonesia berhak menyimpan, mengalihmedia/formatkan, mengelolanya dalam bentuk pangkalan data (*database*), merawat, dan memublikasikan tugas akhir saya tanpa perlu meminta ijin dari saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Depok  
Pada tanggal : 17 Juni 2009  
Yang menyatakan

( Nur Hidayat )

## ABSTRAK

Nama : Nur Hidayat  
Program studi : Teknik Elektro  
Judul Skripsi : Rancang Bangun Sistem Kendali Quadrotor Untuk Keseimbangan Posisi Dengan PID

Robot merupakan salah satu ciptaan manusia yang diharapkan dapat mengambil keputusan sendiri dalam batasan tertentu untuk menyelesaikan permasalahan – permasalahan yang berbahaya bagi manusia. Salah satu bentuk robot yang dapat membantu manusia adalah *unmanned aerial vehicle* (UAV). Kelebihan UAV adalah bentuk yang kecil, hemat energi, and kemampuan untuk bekerja tanpa bantuan manusia. Salah satu bentuk UAV adalah quadrotor yang merupakan bentuk lain dari helikopter dengan empat baling - baling. Dengan kemampuan untuk melakukan manuver yang sulit dilakukan oleh wahana lain, serta kemampuan untuk terbang dan mendarat secara vertikal, quadrotor merupakan pilihan UAV yang semakin menarik.

Pembahasan akan difokuskan pada 3 hal, yaitu sistem navigasi, pemodelan, dan pengendalian quadrotor. Sistem navigasi menggunakan sensor *accelerometer*, *gyroscope* serta kompas digital yang dipadukan dengan *kalman filter* untuk menghilangkan derau. Sistem dimodelkan menggunakan metode *least-square*, selanjutnya quadrotor dikendalikan oleh PID. Berdasarkan hasil yang didapat, gabungan sistem navigasi dan pengendali yang diusulkan mampu mengendalikan quadrotor sehingga sudut sistem dapat dikendalikan.

Kata kunci: Quadrotor, PID, Kalman Filter, Least-Square.

## ABSTRACT

Name : Nur Hidayat  
Study Programme : Electrical Engineering  
Title : Design of Quadrotor Control System For Angle Stabilization  
Using PID Controller

Robot is one of human creation that is expected to take its own decision in a certain limitation for tasks which are too dangerous for humans. From many kind of robot, unmanned aerial vehicle (UAV) has received tremendous interest. The advantages of a UAV are their small body, energy efficient, and the ability to work without human assistance. One type of UAV is quadrotor which is another form of helicopter with four propellers. With the ability to do difficult maneuver performed by another vehicle, and the ability to do vertical take-off and landing, quadrotor is one of the most promising UAV.

This work will focus on three topics, namely navigation systems, modeling, and control of the quadrotor. The navigation system using an accelerometer, a gyroscope and a digital compass combined with a Kalman filter to remove noise. Modeling system using the method of least-square, and using the PID controller to control the quadrotor. Based on the results obtained, the combined navigation and control system proposed is able to control the quadrotor so that the system angle can be controlled.

Keywords: Quadrotor, PID, Kalman Filter, Least-Square.

## DAFTAR ISI

HALAMAN JUDUL .....	i
HALAMAN PERNYATAAN ORISINALITAS .....	ii
HALAMAN PENGESAHAN .....	iii
UCAPAN TERIMAKASIH .....	iv
HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI .....	v
ABSTRAK .....	vi
ABSTRACT .....	vii
DAFTAR ISI .....	viii
DAFTAR TABEL .....	ix
DAFTAR GAMBAR.....	x
DAFTAR NOTASI .....	xii
DAFTAR SINGKATAN.....	xiii
DAFTAR LAMPIRAN .....	xiv
<b>BAB 1</b> .....	<b>1</b>
1.1 Latar Belakang.....	1
1.2 Tujuan Penulisan .....	1
1.3 Pembatasan Masalah .....	1
1.4 Metode Penelitian.....	2
1.5 Sistematika Penulisan.....	2
<b>BAB 2</b> .....	<b>4</b>
2.1 Quadrotor.....	4
2.2 Identifikasi Sistem Dengan <i>Least-Square</i> .....	8
2.3 Observer .....	11
2.3.1 Observer <i>Luenberger</i> .....	11
2.3.1 Kalman Filter.....	13
<b>BAB 3</b> .....	<b>16</b>
3.1 Skema Pengendalian Quadrotor .....	16
3.2 Sistem pengendali sudut .....	16
3.3 Sistem Navigasi .....	21
3.4 Implementasi Perangkat Keras .....	30
<b>BAB 4</b> .....	<b>33</b>
4.1 Pengujian Sistem Navigasi .....	33
<b>BAB 5</b> .....	<b>44</b>
<b>DAFTAR PUSTAKA</b> .....	<b>45</b>
<b>LAMPIRAN</b> .....	<b>46</b>
Lampiran 1. Foto quadrotor yang digunakan .....	46
Lampiran 2. Sistem pengetesan robot.....	47

## DAFTAR TABEL

Tabel 3.1. Daftar nilai variabel dan konstanta yang dipakai dalam kalman filter....17



## DAFTAR GAMBAR

Gambar 2.1. Skema umum quadrotor dilihat dari atas.....	4
Gambar 2.2. Pengendalian manuver quadrotor. ....	5
Gambar 2.3. Penetapan koordinat quadrotor. ....	6
Gambar 2.4. Sudut-sudut <i>Tait–Bryan</i> quadrotor. ....	6
Gambar 2.5. Pengendalian manuver quadrotor. ....	7
Gambar 2.6. Kombinasi gerakan quadrotor .....	7
Gambar 2.7. Blok diagram persamaan ruang keadaan. ....	12
Gambar 2.8. Gambaran lengkap observer dan sistem yang diobservasi .....	13
Gambar 3.1. Gambaran umum sistem pengendali robot.. ....	16
Gambar 3.2. Skema pengendali sudut. ....	18
Gambar 3.3. Grafik daya angkat terhadap tegangan pada dua motor.....	20
Gambar 3.4. Daerah kerja robot.. ....	20
Gambar 3.5. Skema perubahan data <i>accelerometer</i> ke sudut kemiringan.....	22
Gambar 3.6. Sebaran pembacaan $\theta_{acc}$ .....	23
Gambar 3.7. Sebaran pembacaan $\dot{\theta}_{gyro}$ .....	24
Gambar 3.8. Gambaran lengkap sistem navigasi .....	25
Gambar 3.9. Sebaran bacaan sensor kompas.....	29
Gambar 3.10. Skema lengkap sistem pengendali.....	30
Gambar 3.11. Skema umum perangkat keras.....	30
Gambar 3.12. Modul CMPS03.....	31
Gambar 3.13. Modul IX001. ....	31
Gambar 3.14. Gambaran lengkap implementasi sistem. ....	32
Gambar 4.1. Perbandingan data $\theta_{acc}$ pada sudut 0 derajat. ....	33
Gambar 4.2. Perbandingan perkiraan sudut $\varphi$ sebelum dan sesudah blok <i>kalman filter</i> . ...	34
Gambar 4.3. Efek perubahan nilai $P$ pada data yang dipakai gambar 4.2.....	35
Gambar 4.4. Respon statik untuk sistem $\varphi$ terhadap $S_2$ .....	36
Gambar 4.5. Respon statik untuk sistem $\theta$ terhadap $S_1$ .....	36
Gambar 4.6. Hasil identifikasi <i>least-square</i> untuk sistem $\theta$ terhadap $S_1$ .....	37
Gambar 4.7. Hasil identifikasi <i>least-square</i> untuk sistem $\psi$ terhadap $S_3$ .....	38
Gambar 4.8. Plot letak kedudukan akar untuk sistem $\theta$ terhadap $S_1$ .....	38
Gambar 4.9. Respon loop tertutup untuk <i>set point</i> 1 untuk sistem $\theta$ terhadap $S_1$ .....	39
Gambar 4.10. Desain pengendali PID untuk $\theta$ terhadap $S_1$ .....	39
Gambar 4.11. Respon simulasi sistem $\theta$ dan $\varphi$ berpengendali terhadap <i>set point</i> 1.....	40
Gambar 4.12. Plot letak kedudukan akar untuk sistem $\psi$ terhadap $S_3$ .....	40
Gambar 4.13. Plot letak kedudukan akar untuk sistem $\psi$ dalam <i>range unit circle</i> .....	41
Gambar 4.14. Respon loop tertutup untuk <i>set point</i> 1 untuk sistem $\psi$ .....	41
Gambar 4.15. Respon simulasi sistem $\psi$ berpengendali terhadap <i>set point</i> 1.....	42
Gambar 4.16. Implementasi pengendali pada sistem $\theta$ terhadap $S_1$ .....	43

Gambar 4.17. Implementasi pengendali pada sistem  $\varphi$  terhadap  $S_2$  .....43

Gambar 4.18. Implementasi pengendali pada sistem  $\psi$  terhadap  $S_3$  .....43



## DAFTAR NOTASI

$y(k)$	Keluaran sistem SISO waktu ke $k$ .
$u(k)$	Masukan sistem SISO waktu ke $k$ .
$\Phi$	Matriks data sistem SISO pada algoritma <i>least-square</i> .
$\Theta$	Vektor parameter sistem SISO pada algoritma <i>least-square</i> .
$\theta$	Sudut <i>pitch</i> robot.
$\varphi$	Sudut <i>roll</i> robot.
$\psi$	Sudut <i>yaw</i> robot.
$\theta^*$	Sudut <i>pitch referensi</i> robot.
$\varphi^*$	Sudut <i>roll referensi</i> robot.
$\psi^*$	Sudut <i>yaw referensi</i> robot.
$\ddot{x}_{acc}$	Data percepatan searah sumbu $x$ <i>accelerometer</i> .
$\ddot{y}_{acc}$	Data percepatan searah sumbu $y$ <i>accelerometer</i> .
$\theta_{acc}$	Sudut <i>pitch</i> perkiraan <i>accelerometer</i> .
$\varphi_{acc}$	Sudut <i>roll</i> perkiraan <i>accelerometer</i> .
$\dot{\theta}_{gyro}$	Data kecepatan sudut <i>pitch</i> perkiraan <i>gyroscope</i> .
$\dot{\varphi}_{gyro}$	Data kecepatan sudut <i>roll</i> perkiraan <i>gyroscope</i> .
$\psi_{kompas}$	Sudut <i>yaw</i> perkiraan <i>kompas</i> .
$A$	Matriks transisi persamaan ruang keadaan pada <i>kalman filter</i> .
$B$	Matriks masukan persamaan ruang keadaan pada <i>kalman filter</i> .
$C$	Matriks keluaran persamaan ruang keadaan pada <i>kalman filter</i> .
$Q$	Matriks kovarian dari derau pada sistem <i>kalman filter</i> .
$R$	Matriks kovarian dari derau pada keluaran sistem <i>kalman filter</i> .
$X(k)$	Vektor variabel ruang keadaan pada sistem <i>kalman filter</i> .
$P(k)$	Matriks kovarian sistem <i>kalman filter</i> .
$S_n, n = 1, \dots, 3$	Masukan sistem ke $n$ pada sistem pengendalian robot.
$V_{mn}, n = 1, \dots, 4$	Tegangan motor ke $n$ .

## DAFTAR SINGKATAN

UAV	<i>Unmanned Aerial Vehicle</i>
PWM	<i>Pulse Width Modulation</i>
BLDC	<i>Brushless Direct Current (Motor)</i>
DC	<i>Duty Cycle</i>
BEMF	<i>Back – EMF</i>
MIMO	<i>Multiple-Input (and) Multiple-Output</i>
SISO	<i>Single-Input (and) Single-Output</i>
PID	<i>Propotional Integral Derivative (Controller)</i>
uC	<i>Microcontroller</i>
IMU	<i>Inertial Measurement Unit</i>
I <sup>2</sup> C	<i>Inter-integrated Circuit</i>
SPI	<i>Serial Peripheral Interface</i>
UART	<i>Universal Asynchronous Receiver/Transmitter</i>

## DAFTAR LAMPIRAN

Lampiran 1. Foto quadrotor yang digunakan .....	37
Lampiran 2. Sistem pengetesan robot.....	37



# BAB 1

## PENDAHULUAN

### 1.1 Latar Belakang

Robot merupakan salah satu ciptaan manusia yang diharapkan dapat mengambil keputusan sendiri dalam batasan tertentu untuk menyelesaikan permasalahan – permasalahan yang berbahaya bagi manusia. Salah satu bentuk robot yang dapat membantu manusia adalah *unmanned aerial vehicle* (UAV). Kelebihan UAV adalah bentuk yang kecil, hemat energi, and kemampuan untuk bekerja tanpa bantuan manusia. Quadrotor merupakan salah satu jenis UAV yang memiliki 4 baling - baling. Dengan kemampuan untuk melakukan manuver yang sulit dilakukan oleh wahana lain, serta kemampuan untuk terbang diam pada posisinya, quadrotor merupakan pilihan UAV yang semakin menarik.

### 1.2 Tujuan Penulisan

Skripsi ini bertujuan untuk menjabarkan sistem pengendalian sudut quadrotor. Pembahasan meliputi sistem navigasi, pemodelan, dan pengendalian sudut.

### 1.3 Pembatasan Masalah

Penulisan akan dibatasi pada bagian berikut:

1. Perancangan sistem pengendali meliputi sistem navigasi, pemodelan, dan pengendalian sudut. Sistem kendali motor BLDC yang merupakan aktuator sistem tidak dibahas.
2. Sistem navigasi menggunakan data sensor IMU dalam modul IX001 dan kompas digital dalam modul CMPS03. Data kedua sensor digabungkan dengan *kalman filter* untuk mendapatkan hasil yang optimal.
3. Pemodelan sistem menggunakan metode *least-square* untuk memodelkan hubungan antara sudut quadrotor dengan tegangan motor.
4. Pengendalian sudut menggunakan teknik letak kedudukan akar sehingga didapatkan pengendali yang bisa mengendalikan sudut robot.

#### 1.4 Metode Penelitian

Metode penelitian yang digunakan pada pekerjaan ini meliputi:

1. Tinjauan pustaka melalui studi literatur dari buku - buku pustaka, *datasheet*, atau *manual book* serta *reference book* dari suatu piranti.
2. Pendekatan diskusi dengan pembimbing skripsi yang berkaitan dengan topik bahasan skripsi.
3. Perancangan perangkat keras dan perangkat lunak dari sistem quadrotor dan sistem pengendalian quadrotor tersebut.
4. Simulasi pengendalian dan sistem - sistem yang terkait.
5. Implementasi dan ujicoba sistem yang telah didesain ke dalam sistem riil.

#### 1.5 Sistematika Penulisan

Untuk mempermudah penulisan dan agar pembahasan yang disajikan lebih sistematis, maka laporan ini dibagi kedalam empat bab. Isi masing – masing bab diuraikan secara singkat dibawah ini :

##### **BAB 1           PENDAHULUAN**

Bab ini menjelaskan tentang latar belakang masalah, tujuan, batasan masalah, metode penulisan, dan sistematika penulisan seminar.

##### **BAB 2           DASAR TEORI**

Dalam bab ini, pembahasan akan difokuskan pada dasar teori metode – metode yang dipakai dalam bab 3. Pembahasan meliputi dasar teori quadrotor, *kalman filter*, serta identifikasi sistem dengan menggunakan *least-square*.

##### **BAB 3           PERANCANGAN PENGENDALI SUDUT**

Bab ini menjelaskan perancangan sistem navigasi dengan menggunakan *kalman filter*, cara identifikasi sistem, dan pengendalian dengan menggunakan PID.

##### **BAB 4           PENGUJIAN DAN ANALISA**

Bab ini difokuskan pada pengujian sistem yang telah didesain pada bab 3 ke dalam sistem riil. Hasil yang didapat kemudian dianalisa.

**BAB 5 KESIMPULAN**

Bab terakhir ini berisikan hasil – hasil yang didapatkan dari pengujian dan analisa yang dibahas pada bab 4.

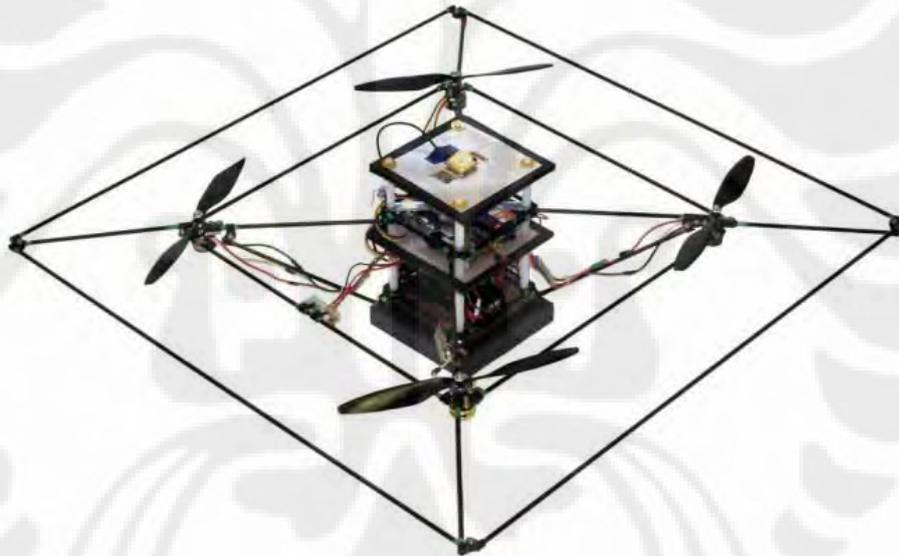


## BAB 2

### DASAR TEORI

#### 2.1 Quadrotor

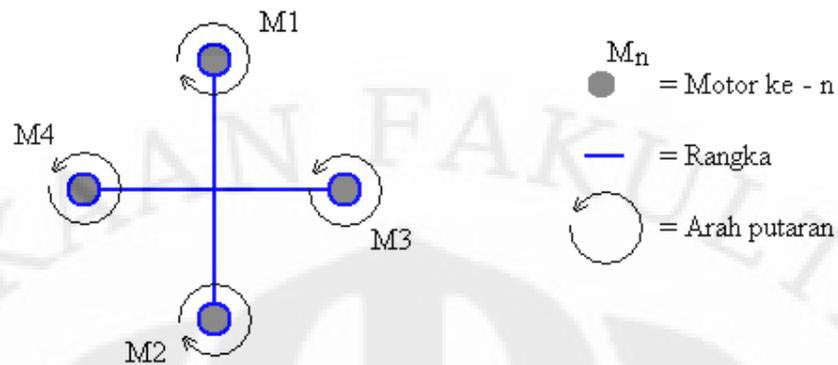
Quadrotor adalah salah satu tipe wahana dengan kemampuan untuk memulai keberangkatan dan mendarat secara vertikal. Kemampuan ini didapatkan dengan menggunakan 4 buah motor dan baling – baling sebagai aktuator. Keempat motor diletakkan sedemikian rupa sehingga quadrotor membentuk segiempat sama sisi dengan masing – masing motor berada di ujung titik sudutnya. Rangka quadrotor berada pada garis diagonal seperti yang dapat dilihat pada gambar 2.1.



Gambar 2.1. Contoh quadrotor.

STARMAC II desain universitas Stanford.

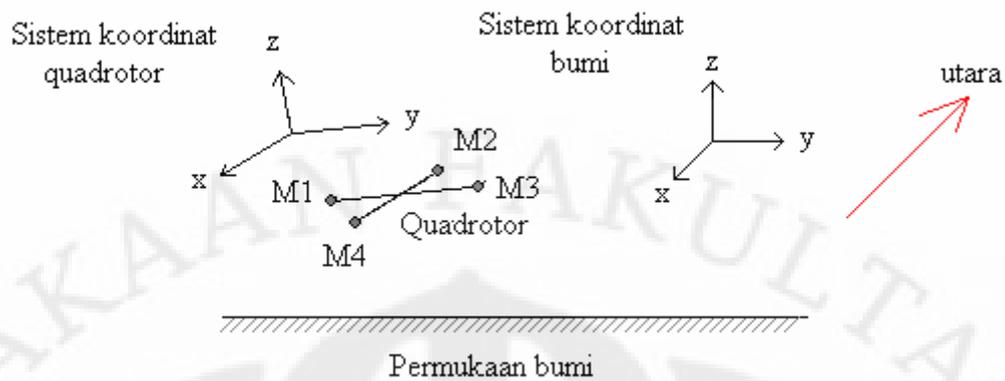
Skema umum quadrotor dapat digambarkan secara sederhana sebagai rangka berbentuk tanda ”+” dengan motor diletakkan pada ujung – ujung rangka tersebut seperti yang dapat dilihat pada gambar 2.2.



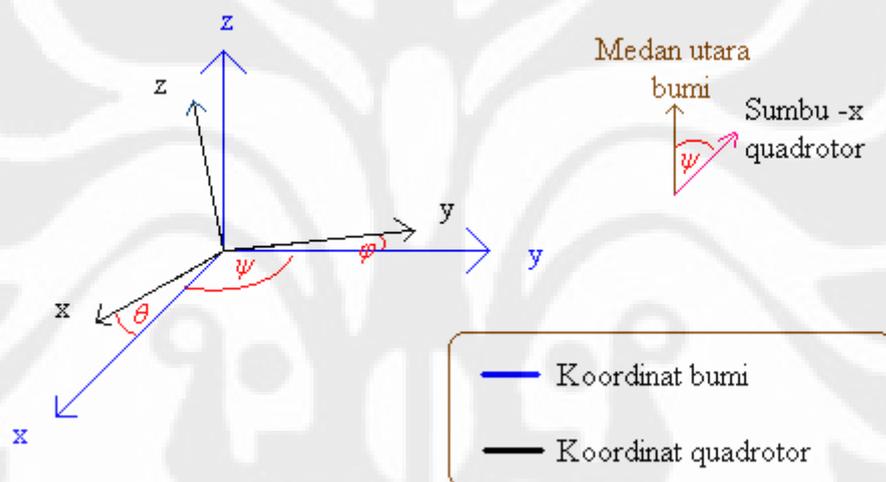
Gambar 2.2. Skema umum quadrotor dilihat dari atas.

Gambar 2.2 menunjukkan bahwa 2 buah baling – baling berputar searah jarum jam, sedangkan 2 baling – baling lainnya berputar berlawanan jarum jam. Pada skema quadrotor yang dipakai, penamaan motor dilakukan searah jarum jam, dengan motor 1 dan 3 berputar searah jarum jam, sedangkan motor 2 dan 4 berputar berlawanan jarum jam. Perbedaan arah putaran tersebut menyebabkan torsi balik dari motor 1 dan motor 2 akan dihilangkan oleh torsi balik dari motor 4 dan motor 3. Konfigurasi ini memungkinkan dihilangkannya *tail-rotor* yang diperlukan dalam konfigurasi helikopter tradisional.

Sistem koordinat yang digunakan dalam pembahasan ke depan melibatkan 2 buah sistem koordinat terpisah. Sistem koordinat pertama adalah sistem koordinat bumi, dimana sistem koordinat ini merupakan sistem koordinat kartesian relatif terhadap permukaan bumi. Sistem koordinat kedua adalah sistem koordinat quadrotor, dimana sistem koordinat ini merupakan sistem koordinat yang menempel pada rangka quadrotor. Dengan menetapkan kedua sistem koordinat ini, maka sudut euler dapat ditentukan, dimana sudut euler yang digunakan mencakup 3 sudut-sudut *Tait–Bryan*, yaitu sudut *pitch* ( $\theta$ ), *roll* ( $\varphi$ ), dan *yaw* ( $\psi$ ). Gambar berikut menggambarkan perbedaan kedua sistem koordinat tersebut dan sudut euler yang digunakan.

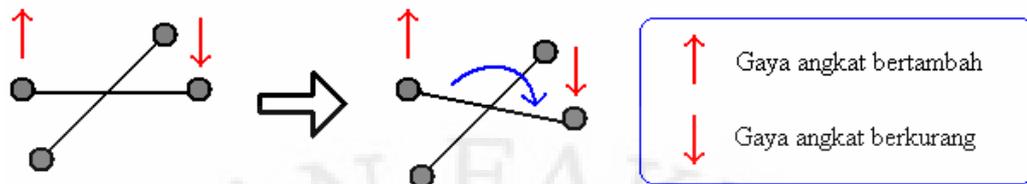


Gambar 2.3. Penetapan koordinat quadrotor.



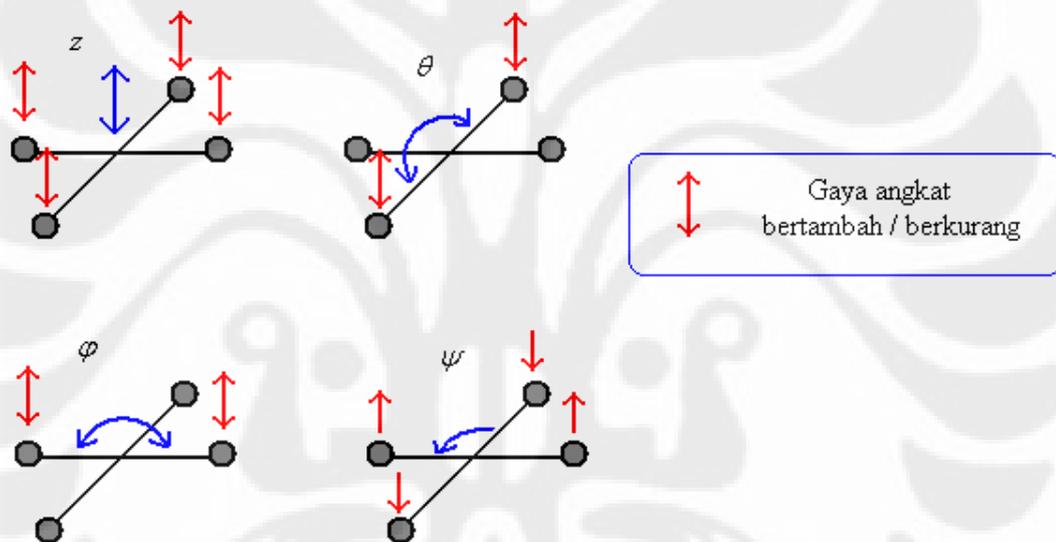
Gambar 2.4. Sudut-sudut *Tait-Bryan* quadrotor.

Quadrotor bergerak dengan memanfaatkan perbedaan gaya angkat tiap motor. Bila sebuah quadrotor ingin melakukan manuver ke kanan, maka quadrotor tersebut hanya perlu menaikkan gaya angkat motor di kiri dan menurunkan gaya angkat motor di kanannya. Dengan melakukan hal tersebut, maka gaya angkat keseluruhan dari quadrotor akan menghasilkan torsi yang memutar quadrotor ke kanan seperti yang dijelaskan pada gambar berikut.



Gambar 2.5. Pengendalian manuver quadrotor.

Dengan mengatur gaya angkat yang diberikan tiap motor, ada 4 buah kombinasi gerakan yang bisa dilakukan oleh quadrotor. Keempat gerakan tersebut adalah 3 gerakan berputar pada sudut-sudut *Tait–Bryan*, ditambah gerakan linear searah dengan sumbu  $z$  koordinat robot.



Gambar 2.6. Kombinasi gerakan quadrotor.

Karena dinamika sistem yang sangat tinggi, quadrotor sangat sulit untuk dikendalikan oleh manusia secara manual. Oleh karena itu, walaupun quadrotor telah ditemukan hampir satu abad yang lalu<sup>1</sup>, implementasi hampir tidak ada secara praktis. Perkembangan teknologi dan murahnya harga semikonduktor menjadikan wahana tipe ini mulai diminati sebagai wahana tanpa awak atau *unmanned aerial vehicle* (UAV) yang dikendalikan melalui mikrokontroler.

<sup>1</sup> <http://en.wikipedia.org/wiki/Quadrotor>.

## 2.2 Identifikasi Sistem Dengan *Least-Square*

Sebelum dapat mendesain pengendali, diperlukan informasi sistem yang akan dikendalikan. Salah satu metode yang dapat digunakan untuk mengidentifikasi fungsi alih sistem adalah *least-square*. Algoritma *least-square* bekerja pada sistem persamaan dengan solusi persamaan banyak, dan kemudian menyelesaikan persamaan tersebut sedemikian rupa sehingga nilai kuadrat kesalahan dapat diminimalkan.

Permasalahan identifikasi adalah bagaimana mengetahui parameter sistem. Sistem yang ingin diidentifikasi diasumsikan adalah sistem *single input single output* (SISO) diskrit yang dijabarkan oleh fungsi alih orde  $-n$  berikut:

$$\frac{Y(z^{-1})}{U(z^{-1})} = \frac{z^{-1}(b_0 + b_1z^{-1} + \dots + b_{n-1}z^{-(n-1)})}{1 + a_1z^{-1} + a_2z^{-2} + \dots + a_nz^{-n}} \quad (2.1)$$

Bila pada persamaan di atas dilakukan perkalian silang antara sisi kanan dan sisi kiri, didapatkan:

$$y + ya_1z^{-1} + ya_2z^{-2} + \dots + ya_nz^{-n} = z^{-1}(ub_0 + ub_1z^{-1} + \dots + ub_{n-1}z^{-(n-1)}) \quad (2.2)$$

Persamaan (2.2) dapat dinyatakan dalam domain waktu cuplik sebagai:

$$\begin{aligned} y(k) &= b_0u(k-1) + b_1u(k-2) + \dots + b_{n-1}u(k-n) \\ &\quad - a_1y(k-1) - a_2y(k-2) - \dots - a_ny(k-n) \end{aligned} \quad (2.3)$$

Bila persamaan (2.3) dinyatakan dalam bentuk vektor, didapatkan:

$$y(k) = [u(k-1) \quad u(k-2) \quad \dots \quad u(k-n) \quad y(k-1) \quad y(k-2) \quad \dots \quad y(k-n)] \cdot \begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_{n-1} \\ -a_1 \\ -a_2 \\ \vdots \\ -a_n \end{bmatrix} \quad (2.4)$$

$$y(k) = \phi_k \cdot \Theta$$

Dengan mengambil  $m$  data, dimana nilai  $m$  dianggap cukup besar, maka persamaan (2.4) dapat ditulis ulang dalam bentuk matrisk sebagai:

$$\begin{bmatrix} y(n+1) \\ y(n+2) \\ \vdots \\ y(m) \end{bmatrix} = \begin{bmatrix} u(n) & u(n-1) & \dots & u(2) & u(1) & y(n) & y(n-1) & \dots & y(2) & y(1) \\ u(n+1) & u(n) & \ddots & u(3) & u(2) & y(n+1) & y(n) & \ddots & y(3) & y(2) \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ u(m-1) & u(m-2) & \dots & u(m-n-1) & u(m-n) & y(m-1) & y(m-2) & \dots & y(m-n-1) & y(m-n) \end{bmatrix} \begin{bmatrix} -a_1 \\ -a_2 \\ \vdots \\ -a_n \\ b_0 \\ b_1 \\ \vdots \\ b_{n-1} \end{bmatrix} \quad (2.5)$$

Atau:

$$\underline{Y} = \underline{\Phi} \cdot \underline{\Theta} \quad (2.6)$$

Permasalahan yang dinyatakan dalam persamaan (2.6) adalah mencari nilai matriks  $\underline{\Theta} = [b_0 \ b_1 \ \dots \ b_{n-1} \ a_1 \ a_2 \ \dots \ a_n]^T$  agar nilai parameter sistem diketahui dan fungsi alih persamaan (2.1) dapat diselesaikan. Untuk itu, digunakan sebuah parameter yang merupakan nilai dari fungsi biaya yang merupakan kuadrat kesalahan, dimana kesalahan didefinisikan sebagai perbedaan antara solusi sistem yang sebenarnya ( $y_k$ ), dengan solusi sistem yang diperkirakan ( $\phi_k \cdot \underline{\Theta}$ ), atau:

$$\varepsilon_k = y_k - \phi_k \cdot \underline{\Theta} \quad (2.7)$$

dengan  $k = 1, \dots, m$ .

Persamaan (2.6) diselesaikan dengan meminimumkan fungsi biaya  $J$  terhadap parameter sistem  $\underline{\Theta}$  yang merupakan jumlah dari kuadrat kesalahan, atau:

$$\begin{aligned} \arg \min_{\underline{\Theta}} J &= \arg \min_{\underline{\Theta}} \sum_{k=1}^m \varepsilon_k^2 \\ &= \arg \min_{\underline{\Theta}} \sum_{k=1}^m (y_k - \phi_k \cdot \underline{\Theta})^2 \end{aligned} \quad (2.8)$$

Persamaan di atas dapat diselesaikan dengan mencari turunan fungsi  $J$  terhadap parameter  $\underline{\Theta}$  dan menyamakan dengan nol, atau:

$$\begin{aligned}
0 &= \frac{\partial J}{\partial \Theta} \\
0 &= \frac{\partial \sum_{k=1}^m (y_i - \phi_k \cdot \Theta)^2}{\partial \Theta} \\
0 &= \frac{\partial \left( (\underline{Y} - \underline{\Phi} \cdot \underline{\Theta})^T (\underline{Y} - \underline{\Phi} \cdot \underline{\Theta}) \right)}{\partial \Theta} \quad (2.9) \\
0 &= \frac{\partial \left( \underline{Y}^T \underline{Y} - \underline{Y}^T (\underline{\Phi} \cdot \underline{\Theta}) - (\underline{\Phi} \cdot \underline{\Theta})^T \underline{Y} + (\underline{\Phi} \cdot \underline{\Theta})^T (\underline{\Phi} \cdot \underline{\Theta}) \right)}{\partial \Theta} \\
0 &= -2 \underline{\Phi}^T \underline{Y} + 2 (\underline{\Phi}^T \underline{\Phi}) \underline{\Theta}
\end{aligned}$$

Nilai parameter persamaan di atas dapat dicari dengan persamaan berikut:

$$\underline{\Theta} \leftarrow (\underline{\Phi}^T \underline{\Phi})^{-1} \underline{\Phi}^T \underline{Y} \quad (2.10)$$

Dari persamaan (2.10), persamaan (2.1) dapat diselesaikan dengan meletakkan nilai parameter  $\underline{\Theta} = [b_0 \ b_1 \ \dots \ b_{n-1} \ a_1 \ a_2 \ \dots \ a_n]^T$  ke dalam persamaan (2.1). Algoritma *least-square* untuk sistem orde  $-n$  dapat dijabarkan dalam langkah-langkah berikut:

1. Beri masukan  $u(k)$  pada sistem *open-loop* sebanyak  $m - n$  masukan.

2. Simpan keluaran  $y(k)$  dari proses 1.

$$\Phi \leftarrow \begin{bmatrix} u(n) & u(n-1) & \dots & u(2) & u(1) & y(n) & y(n-1) & \dots & y(2) & y(1) \\ u(n+1) & u(n) & \ddots & u(3) & u(2) & y(n+1) & y(n) & \ddots & y(3) & y(2) \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ u(m-1) & u(m-2) & \dots & u(m-n-1) & u(m-n) & y(m-1) & y(m-2) & \dots & y(m-n-1) & y(m-n) \end{bmatrix}$$

4.  $\underline{Y} \leftarrow [y(n+1) \ y(n+2) \ \dots \ y(m)]^T$ .

5. Definisikan vektor  $\underline{\Theta} \leftarrow [b_0 \ b_1 \ \dots \ b_{n-1} \ a_1 \ a_2 \ \dots \ a_n]^T$ .

6.  $\underline{\Theta} \leftarrow (\underline{\Phi}^T \underline{\Phi})^{-1} \underline{\Phi}^T \underline{Y}$ .

7. Bentuk fungsi alih  $\frac{Y(z^{-1})}{U(z^{-1})}$  berdasarkan nilai  $\underline{\Theta}$ .

## 2.3 Observer

### 2.3.1 Observer *Luenberger*

Dalam menganalisa sistem, terkadang beberapa variabel keadaan pada sistem tidak dapat diketahui melalui sensor. Permasalahan ini muncul ketika sensor yang dibutuhkan untuk mengetahui nilai dari variabel keadaan memiliki harga yang sangat mahal, terkadang sensor yang dibutuhkan membuat kinerja sistem berkurang (sensor berdimensi terlalu besar, terlalu berat, dll), dan terkadang variabel keadaan memang tidak dapat diamati secara fisik. Untuk mengatasi permasalahan tersebut, digunakan sebuah skema yang dapat mengetahui variabel – variabel keadaan yang tidak dapat diketahui secara langsung. Skema dengan menggunakan observer bertujuan untuk mengetahui variabel – variabel keadaan dengan cara mengestimasi variabel – variabel tersebut berdasarkan variabel keadaan lainnya yang dapat diamati dengan sensor.

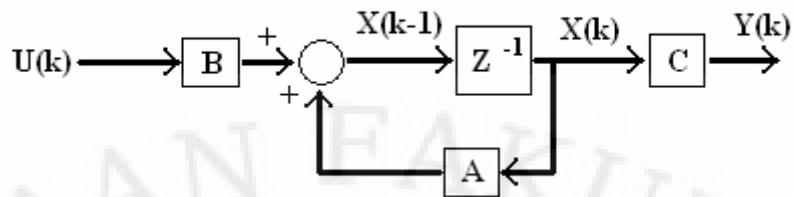
Observer adalah sebuah sistem yang dapat mengestimasi variabel dari sebuah sistem yang dinyatakan dalam bentuk ruang keadaan. Observer bekerja dengan cara meniru sistem yang sebenarnya, dan kemudian melakukan estimasi terhadap variabel keadaan yang tidak dapat diamati dengan sensor berdasarkan variabel lain yang dapat diamati dengan sensor.

Misalkan suatu sistem diskrit dapat dinyatakan dalam persamaan ruang keadaan dengan sebagai berikut:

$$\begin{aligned} X(k+1) &= A.X(k) + B.U(k) \\ Y(k) &= C.X(k) \end{aligned} \quad (2.11)$$

$X \in R^n$  merupakan vektor variabel keadaan,  $Y \in R^m$  merupakan vektor keluaran sistem, dan  $U \in R^r$  merupakan vektor masukan sistem. Matriks  $A$ ,  $B$ , dan  $C$  merupakan matriks yang merepresentasikan sistem yang memiliki sifat *linear* dan *time-invariant*.

Persamaan ruang keadaan di atas dapat dinyatakan dalam blok – blok diagram seperti gambar di bawah.



Gambar 2.7. Blok diagram persamaan ruang keadaan.

Vektor variabel keadaan  $X(k)$  merupakan gabungan antara variabel – variabel yang dapat diukur dan variabel – variabel tak terukur. Bila sistem di atas memiliki sifat *observable*, maka variabel – variabel yang tidak terukur dapat diketahui. Pertama – tama, definisikan matriks *observability*:

$$O = \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{n-1} \end{bmatrix} \quad (2.12)$$

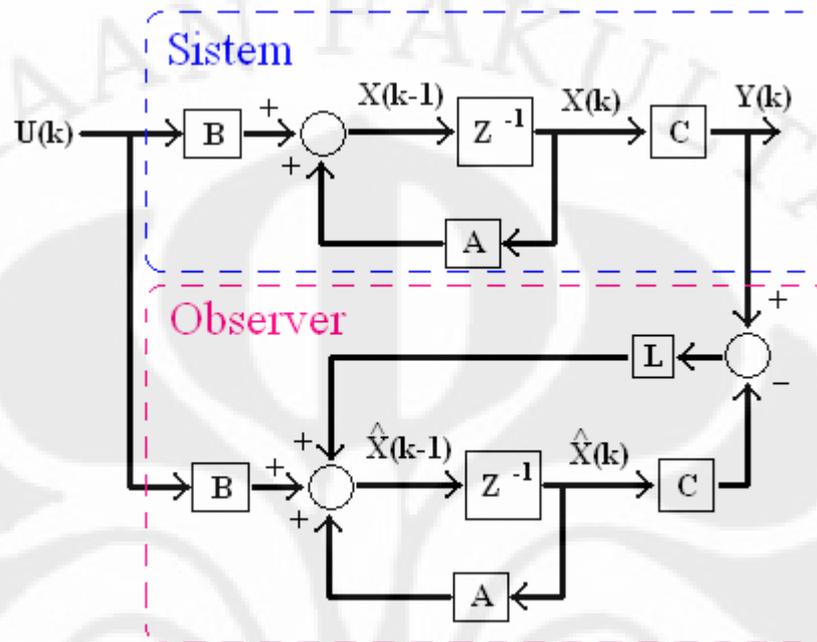
Suatu observer dari sistem dapat dibuat jika sistem tersebut memiliki sifat *observable*, atau dengan kata lain matriks *observability* dari sistem merupakan matriks dengan rank  $- n$ .

Suatu observer bekerja dengan meniru kerja sistem yang sebenarnya. Dengan memanfaatkan informasi berupa masukan dan keluaran sistem yang terukur, maka variabel keadaan dari sistem yang sebenarnya dapat diketahui. Observer dari sistem dengan persamaan ruang keadaan (2.11) dapat dinyatakan sebagai:

$$\begin{aligned} \hat{X}(k+1) &= A.\hat{X}(k) + B.U(k) + L(Y(k) - \hat{Y}(k)) \\ \hat{Y}(k) &= C.\hat{X}(k) \end{aligned} \quad (2.13)$$

Vektor  $\hat{X}$  adalah variabel keadaan dan  $\hat{Y}$  adalah keluaran sistem yang diestimasi oleh observer. Matriks sistem  $A$ ,  $B$ , dan  $C$  merupakan matriks yang sama dengan matriks sistem asli pada persamaan (2.11). Matriks  $L$  merupakan matriks konstan penguat observer yang dipilih sedemikian sehingga matriks  $(A - LC)$  memiliki pole atau nilai eigen di dalam *unit circle* dan memiliki pole

dengan waktu respon jauh lebih cepat dari sistem yang diobservasi. Observer yang didesain dengan cara ini dinamakan observer *Luenberger*.



Gambar 2.8. Gambaran lengkap observer dan sistem yang diobservasi.

### 2.3.1 Kalman Filter

Dalam mengimplementasikan teori kendali ke dalam aplikasi riil, keberadaan derau tidak dapat disepelekan. Derau yang merupakan sebuah sinyal yang tidak dapat dikendalikan dan tidak diketahui nilainya dapat membuat skema pengendalian yang dirancang tidak dapat bekerja dengan optimal, bahkan bisa saja tidak bekerja sama sekali. Dalam sistem yang tidak bebas derau, keberadaan derau dapat dimunculkan oleh beberapa sebab, seperti ketidaksempurnaan struktur fisik sistem, keterbatasan keakuratan penghitungan mikrokontroler yang digunakan, keterbatasan aktuator, ataupun keterbatasan sensor. Dari semua alasan yang telah disebutkan tersebut, umumnya keberadaan derau terbesar datang dari derau pada sensor.

Untuk mengatasi keberadaan derau, diperlukan sebuah skema *filtering* ke dalam desain sistem observer. *Kalman filter* adalah skema observer yang mengasumsikan adanya derau yang terjadi pada sistem linear, dimana derau

terdistribusi secara gaussian, dan memiliki rata – rata nol. Walaupun dalam implementasinya derau seringkali tidak berlaku demikian, eksperimen yang dijelaskan di bagian selanjutnya akan membuktikan bahwa asumsi ini terbukti cukup ampuh dalam mengatasi derau yang datang.

Pada sistem yang dinyatakan dalam persamaan (2.11), derau yang muncul berasal dari derau pada keluaran ( $v$ ) yang memiliki kovarian  $R$  dan derau pada proses sistem ( $w$ ) yang memiliki kovarian  $Q$ . Dengan memasukkan efek derau tersebut ke dalam persamaan (2.11), didapatkan:

$$\begin{aligned} X(k+1) &= A.X(k) + B.u(k) + w \\ Y(k) &= C.X(k) + v \end{aligned} \quad , \text{ atau}$$

$$\begin{aligned} X(k) &= A.X(k-1) + B.u(k-1) + w \\ Y(k-1) &= C.X(k-1) + v \end{aligned} \quad (2.14)$$

*Kalman filter* akan mengestimasi nilai variabel keadaan  $X(k)$  dengan menggunakan penguat observer khusus yang dinamakan penguat kalman. Dalam perhitungan yang dilakukan secara sekuensial, vektor estimasi variabel keadaan dibagi 2 menjadi  $\hat{X}(k|k)$  yang merupakan vektor estimasi variabel keadaan pada waktu ke  $k$  berdasarkan data pada waktu ke  $k$ , dan  $\hat{X}(k|k-1)$  yang merupakan vektor estimasi variabel keadaan pada waktu ke  $k$  berdasarkan data pada waktu ke  $k-1$ .

Bila matriks  $L$  pada observer *Luenberger* merupakan matriks konstan, matriks penguat kalman  $L_k$  akan berubah sesuai dengan variabel keadaan pada waktu lampau. *Kalman filter* menggunakan matriks kovarian  $P$  yang didefinisikan sebagai kovarian dari nilai estimasi  $\hat{X}$  dengan nilai yang sebenarnya  $X$ , atau:

$$\begin{aligned} P(k|k) &= \text{cov}\left(X(k) - \hat{X}(k|k)\right) \\ P(k|k-1) &= \text{cov}\left(X(k) - \hat{X}(k|k-1)\right) \end{aligned} \quad (2.15)$$

Algoritma *kalman filter* bekerja secara rekursif dalam dua tahap, yaitu tahap prediksi dan tahap koreksi. Algoritma umum *kalman filter* dapat dijabarkan dalam langkah-langkah berikut:

Untuk waktu cuplik ke -  $k$ :

1. Prediksi:

$$1.1. \hat{X}(k|k-1) \leftarrow A.X(k-1|k-1) + B.U(k-1)$$

$$1.2. \hat{P}(k|k-1) \leftarrow A.P(k-1|k-1).A^T + Q$$

2. Koreksi:

$$2.1. L_k \leftarrow \frac{\hat{P}(k|k-1).C^T}{C.\hat{P}(k|k-1).C^T + R}$$

$$2.2. \hat{X}(k|k) \leftarrow \hat{X}(k|k-1) + L_k.(Y(k) - C.\hat{X}(k|k-1))$$

$$2.3. \hat{P}(k|k) \leftarrow \hat{P}(k|k-1) - K.C.\hat{P}(k|k-1)$$

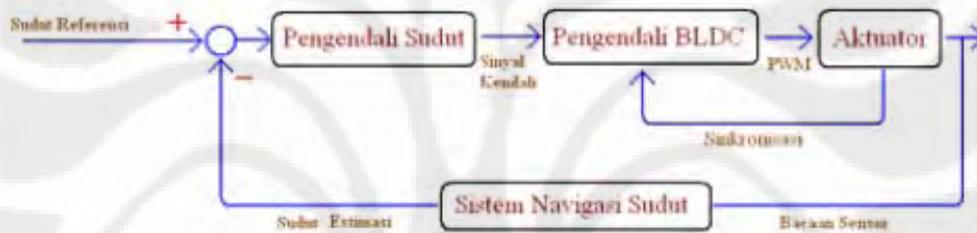
3. Tunggu waktu cuplik berikutnya.

## BAB 3

### PERANCANGAN PENGENDALI SUDUT ROBOT

#### 3.1 Skema Pengendalian Quadrotor

Agar robot dapat bekerja mengikuti *set point* / referensi yang diberikan, robot tersebut harus memiliki suatu skema pengendali yang dapat mempengaruhi robot tersebut agar sesuai dengan permintaan pengguna. Aktuator yang digunakan oleh robot berupa motor *brushless* DC (BLDC) dan baling – baling yang ditempelkan pada bagian rotor motor. Skema umum pengendali UAV yang dipakai dapat dilihat pada gambar berikut.



Gambar 3.1. Gambaran umum sistem pengendali robot.

Berdasarkan gambar 3.1, sistem pengendali robot dapat dibagi menjadi tiga bagian utama, yaitu bagian sistem navigasi, bagian pengendali sudut, dan bagian pengendali motor BLDC. Motor BLDC merupakan motor sinkron 3 fasa dengan rotor berupa medan magnet permanen. Pembahasan pengendali motor BLDC telah dibahas pada pekerjaan sebelumnya, sehingga pada bagian selanjutnya pembahasan akan terfokus pada bagian sistem navigasi dan pengendali sudut.

#### 3.2 Sistem pengendali sudut

Sistem pengendali sudut robot menggunakan pengendali PID. Desain pengendali didasarkan pada hubungan statik antara gaya angkat tiap motor dengan sudut yang diberikan.

$$\begin{aligned}
\theta &= c_1 (F_3 - F_1) \\
\phi &= c_2 (F_4 - F_2) \\
\psi &= c_3 (F_3 + F_1) - (F_4 + F_2) \\
z &= c_4 (F_1 + F_2 + F_3 + F_4)
\end{aligned} \tag{3.1}$$

dengan  $F_n; n=1, \dots, 4$  merupakan gaya angkat motor ke  $n$  dan  $c_n; n=1, \dots, 4$  adalah konstanta pada sistem ke  $n$ . Karena pengendalian meliputi pengendalian sudut saja, maka hanya 3 persamaan awal dari persamaan (3.1) yang dipakai dalam pengendalian. Persamaan keempat digunakan untuk menentukan daerah kerja dari pemodelan sistem. Pada sistem, diasumsikan hubungan antara gaya angkat dengan tegangan motor linear sehingga persamaan (3.1) menjadi:

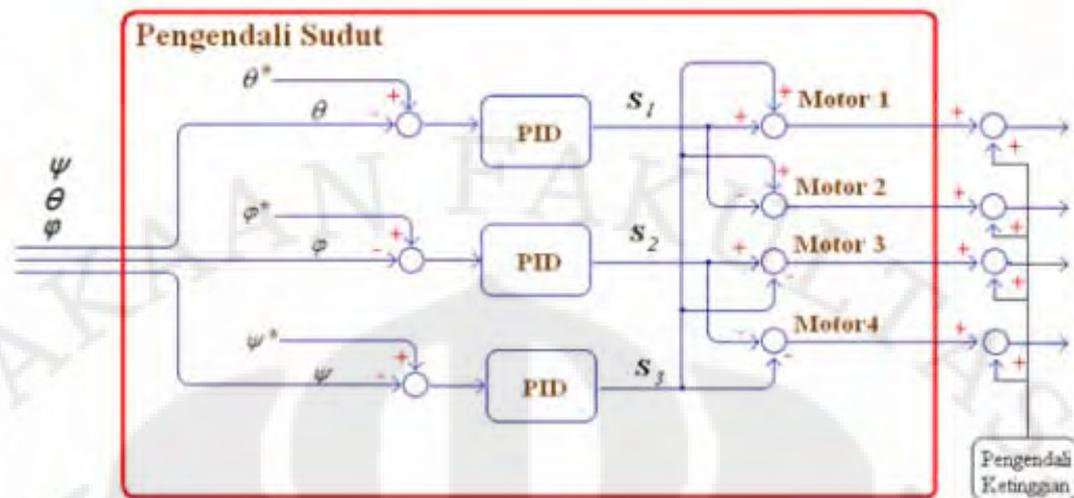
$$\begin{aligned}
\theta &= c_1 (V_{m3} - V_{m1}) \\
\phi &= c_2 (V_{m4} - V_{m2}) \\
\psi &= c_3 (V_{m3} + V_{m1}) - (V_{m4} + V_{m2}) \\
z &= c_4 (V_{m1} + V_{m2} + V_{m3} + V_{m4})
\end{aligned} \tag{3.2}$$

Sinyal kendali sistem dinyatakan sebagai fungsi linear dari masing – masing tegangan BLDC seperti yang dinyatakan dalam persamaan berikut:

$$\begin{aligned}
S_1 &= V_{m1} - V_{m2} \\
S_2 &= V_{m3} - V_{m4} \\
S_3 &= (V_{m1} + V_{m2}) - (V_{m3} + V_{m4})
\end{aligned} \tag{3.1}$$

dengan  $V_{m_n}, n=1, \dots, 4$  adalah tegangan motor ke  $n$  dan  $S_n, n=1, \dots, 3$  adalah sinyal kendali menuju pengendali sudut.

Karena sistem robot ini bersifat multivariabel dengan tiga masukan (sudut *roll* ( $\theta$ ), *pitch* ( $\phi$ ), dan *yaw* ( $\psi$ )) dan empat keluaran (tegangan untuk motor 1, 2, 3, dan 4), maka pengendali didesain menjadi 3 pengendali PID yang terpisah satu sama lain. Ketiga pengendali menerima masukan berupa sudut robot ( $\theta$ ,  $\phi$  dan  $\psi$ ) estimasi sistem navigasi, membandingkannya dengan nilai referensi ( $\theta^*$ ,  $\phi^*$  dan  $\psi^*$ ) yang diberikan *user*, dan kemudian memberikan sinyal kendali yang sesuai kepada pengendali motor BLDC sehingga sudut robot mengikuti nilai referensi yang telah ditentukan. Skema pengendali dapat dilihat pada gambar 3.2.



Gambar 3.2. Skema pengendali sudut.

Berdasarkan gambar 3.2, pengendali *pitch*, *roll*, dan *yaw* merupakan pengendali yang berdiri sendiri. Efek *coupling* berada pada sinyal kendali yang menuju ke pengendali BLDC. Proses desain pengendali sudut dapat dibagi menjadi 2 tahap utama untuk masing – masing pengendali, yaitu tahap pemodelan dan tahap desain pengendali.

### 3.2.1 Pemodelan sistem

Berdasarkan skema pengendali yang ada di gambar 3.2, pemodelan dilakukan menjadi tiga proses yang mirip. Proses pertama ditunjukkan untuk melihat hubungan antara beda kecepatan motor 1 dan motor 2 (motor 1 – motor 2) dan sudut *pitch* yang dibaca sensor. Sedangkan proses kedua ditunjukkan untuk melihat perbedaan kecepatan motor 3 dan motor 4 (motor 3 – motor 4) dengan sudut *roll*. Sedangkan proses ketiga ditunjukkan untuk melihat hubungan beda kecepatan keempat motor (motor 1 + motor 2 – motor 3 – motor 4) dengan sudut *yaw*.

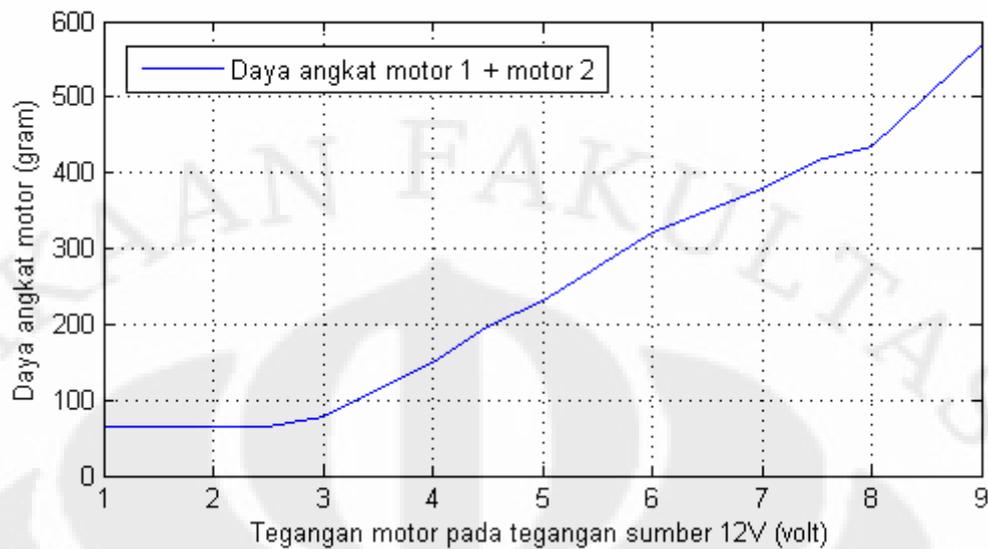
Berdasarkan gambar 3.2, hubungan antara sinyal kendali  $S_n$  dengan bacaan sensor  $\theta$ ,  $\varphi$ , dan  $\psi$  dapat dicari, dimana hubungan sistem sudut quadrotor  $\theta$  dipengaruhi oleh  $S_1$ ,  $\varphi$  dipengaruhi oleh  $S_2$ , dan  $\psi$  dipengaruhi  $S_3$ . Dengan demikian, sistem yang akan dikendalikan terdiri dari tiga sistem *SISO* (*single*

*input – single output*) yang berdiri sendiri terhadap sinyal kendali  $S_n$ , dan metode identifikasi sistem *SISO* yang sudah umum dipakai seperti *least-square* dapat digunakan.

Pada sistem pengendali sudut yang digunakan, identifikasi sistem dilakukan 3 kali untuk mencari hubungan antara sinyal kendali  $\theta$ ,  $\varphi$ , dan  $\psi$  terhadap  $S_1$ ,  $S_2$ , dan  $S_3$ . Karena berdasarkan konsep, ketiga proses identifikasi adalah sama, maka pembahasan selanjutnya hanya akan membahas identifikasi salah satu sistem, yaitu hubungan antara  $\theta$  terhadap  $S_1$ .

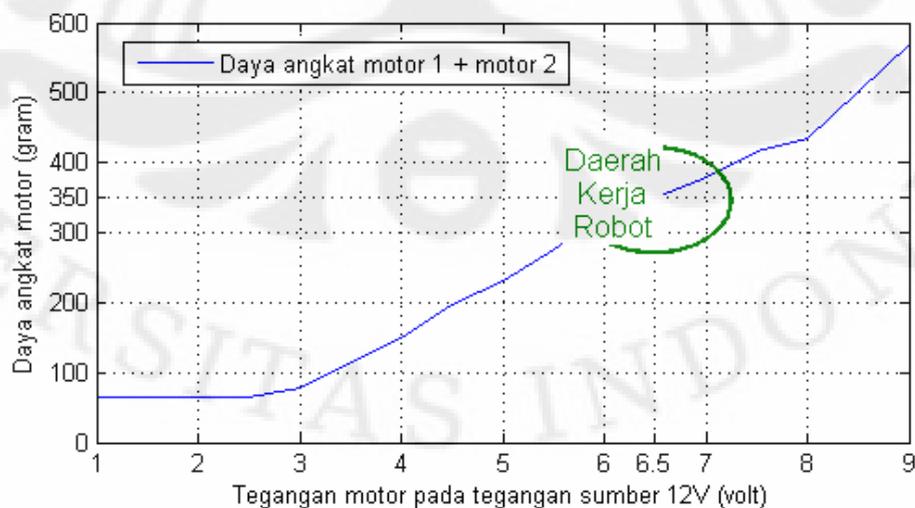
Identifikasi sistem  $\theta$  terhadap  $S_1$  menggunakan pendekatan *black box* dimana diasumsikan sistem yang ada adalah sistem berorde 2. Pemilihan orde 2 karena sistem orde 2 dianggap sudah mewakili sistem secara keseluruhan dan memakan lebih sedikit memori dibandingkan orde 3. Proses identifikasi dilakukan dengan cara memberikan sinyal kendali acak ke dalam sistem yang telah diberi pengaman secara manual. Sinyal kendali hanya diubah ketika sistem sudah mulai terlihat memasuki keadaan waktu tunak. Hal ini dilakukan agar data yang diperoleh juga mengandung informasi sistem saat keadaan waktu tunak.

Sebelum menggunakan metode identifikasi pada sistem, diperlukan informasi berupa daerah kerja dari sistem tersebut. Pada sistem quadrotor ini, daerah kerja ditentukan dari grafik antara daya angkat dengan tegangan yang diberikan pada motor seperti yang dapat dilihat pada gambar 3.3. Grafik ini memperlihatkan bahwa sistem memiliki saturasi pada tegangan rendah dan linear pada tegangan menengah. Data diambil hanya pada dua motor untuk mempermudah pengambilan data. Dengan mengasumsikan bahwa robot memiliki sifat simetris, maka daya angkat total adalah dua kali daya angkat pada grafik gambar 3.3.



Gambar 3.3. Grafik daya angkat terhadap tegangan pada dua motor.

Beban robot adalah 650 gram, dengan asumsi bahwa beban tambahan sebesar 50 gram, maka beban yang harus diangkat robot adalah 700 gram. Bila daya angkat robot melebihi 700 gram, robot akan terbang naik dan bila kurang maka robot akan turun. Sedangkan daya angkat tepat 700 gram akan membuat robot melayang di udara. Karena robot simetris, maka beban yang harus diangkat oleh kedua motor adalah setengah beban total, atau 350 gram. Dari asumsi tersebut, maka daerah kerja robot dapat ditentukan dari grafik gambar 3.9, yaitu pada tegangan motor sekitar 6.5 volt seperti yang dapat dilihat pada gambar 3.4.



Gambar 3.4. Daerah kerja robot.

Setelah menentukan daerah kerja, maka dilakukan identifikasi dengan *least-square*. Karena metode yang digunakan adalah *linear least-square*, maka identifikasi sama saja dengan melakukan linearisasi robot pada daerah kerja tersebut. Selanjutnya identifikasi sistem dapat dijabarkan oleh langkah – langkah berikut:

1. Definisikan matriks sistem  $\underline{\Phi} \leftarrow \begin{bmatrix} u(2) & u(1) & y(2) & y(3) \\ u(3) & u(2) & y(3) & y(4) \\ \vdots & \vdots & \vdots & \vdots \\ u(k-1) & u(k-2) & y(k-1) & y(k-2) \end{bmatrix}$
2. Definisikan vektor keluaran sistem  $\underline{Y} \leftarrow [y(3) \quad y(4) \quad \dots \quad y(k)]^T$
3. Definisikan vektor parameter sistem  $\underline{\Theta} \leftarrow [b_0 \quad b_1 \quad a_1 \quad a_2]^T$
4.  $\underline{\Theta} \leftarrow (\underline{\Phi}^T \underline{\Phi})^{-1} \underline{\Phi}^T \underline{Y}$
5. Bentuk fungsi alih  $\frac{Y(z^{-1})}{U(z^{-1})}$  berdasarkan nilai  $\underline{\Theta}$

### 3.2.2 Desain Pengendali

Desain pengendali dilakukan dengan menggunakan teknik letak kedudukan akar. Pemilihan teknik kedudukan akar dikarenakan penggunaan teknik ini lebih intuitif dan mudah dalam pengaplikasiannya.

Desain pengendali mengikuti sistem yang dimodelkan dan dijabarkan lebih lengkap pada bab 4.

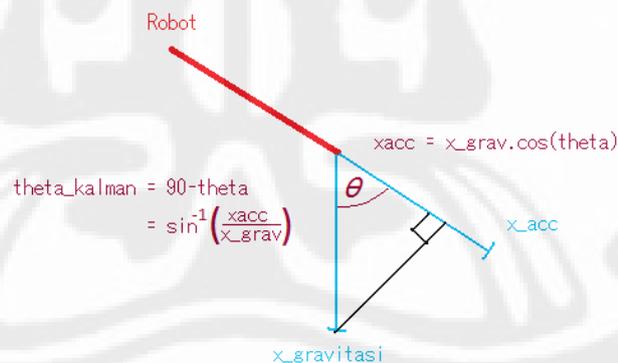
### 3.3 Sistem Navigasi

Tujuan utama keberadaan sistem navigasi digunakan untuk menerjemahkan data-data yang berasal dari pembacaan sensor ke dalam data dengan format yang dibutuhkan pada bagian pengendali sudut, dimana format data tersebut adalah sudut *roll*, *pitch*, dan *yaw* dalam derajat. Data yang akan diterjemahkan berasal dari dua buah sensor, yaitu sensor kompas dalam bentuk kompas digital dan sensor inersia dalam bentuk *inertial measurement unit* (IMU).

Data yang berasal dari IMU akan diolah menjadi data sudut *roll* dan *pitch*, sedangkan data yang berasal dari kompas digital akan diolah menjadi sudut *yaw*.

Penerjemahan data dari kompas digital menjadi sudut *yaw* dapat dilakukan secara sederhana. Data yang berasal dari kompas adalah data digital dengan panjang 2 *byte* dengan nilai yang berkisar antara 0 – 3599, dimana nilai tersebut merepresentasikan sudut kompas dari utara dengan pengali 10. Berdasarkan eksperimen, data yang datang juga tidak memiliki derau yang besar.

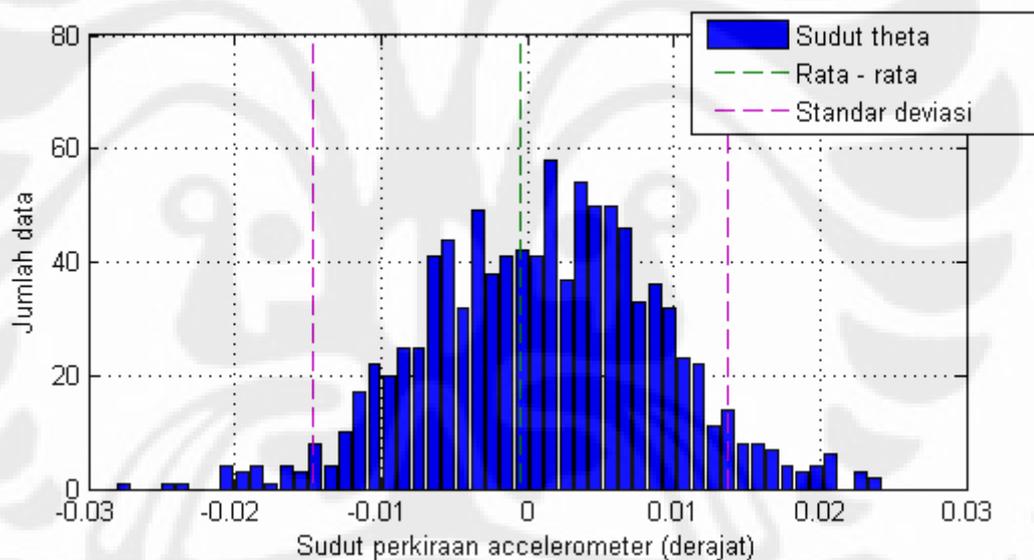
Pembacaan data dari IMU menjadi sudut *pitch* dan *roll* tidak dapat dilakukan secara langsung sebagaimana dibahas pada akhir subbab ini. IMU yang dipakai adalah modul dengan tipe VS-IX001. Modul ini merupakan gabungan dari lima buah sensor yang berbeda, yaitu 3 buah sensor percepatan (*accelerometer*) dan sensor kecepatan sudut (*gyroscope*). *Accelerometer* memberikan informasi percepatan robot dalam 3 dimensi, yaitu percepatan pada sumbu x, y, dan z relatif terhadap frame referensi. *Gyroscope* memberikan informasi percepatan sudut *roll* dan *pitch*. Sistem navigasi menggunakan *accelerometer* untuk mendeteksi sudut dengan menggunakan skema gambar 3.5.



Gambar 3.5. Skema perubahan data *accelerometer* ke sudut kemiringan.

Skema perubahan data pada gambar 3.5 menunjukkan bahwa robot tidak hanya mendapatkan percepatan yang berasal dari sistem aktuatornya sendiri dan gangguan luar, namun robot juga mendapatkan percepatan gravitasi yang berasal dari bumi. Dengan mengasumsikan bahwa percepatan yang diberikan pada sumbu x dan sumbu y mendekati nol, maka sudut robot relatif terhadap bidang yang

tegak lurus dari arah gaya gravitasi dapat diketahui. Asumsi ini dianggap cukup masuk akal, karena pada sistem quadrotor, satu – satunya percepatan yang diberikan oleh aktuator adalah percepatan yang searah sumbu z dari kerangka koordinat robot. Hal ini membuat satu – satunya sumber percepatan searah sumbu x dan y dari kerangka koordinat robot selain percepatan gravitasi adalah derau yang berasal dari gaya luar atau derau sensor, yang diasumsikan kecil dan mendekati nol. Tentu saja, pada kenyataannya derau yang didapatkan besar sehingga data sudut dari *accelerometer* tidak dapat diandalkan sepenuhnya. Gambar 3.6 memperlihatkan hasil perkiraan sudut oleh *accelerometer* pada sudut  $\theta_{acc}$  nol derajat. Terlihat bahwa derau memiliki sebaran gaussian namun memiliki beberapa nilai *bursting*. Derau berupa *bursting* akan sangat mempengaruhi kinerja dan kestabilan sistem.



Gambar 3.6. Sebaran pembacaan  $\theta_{acc}$ .

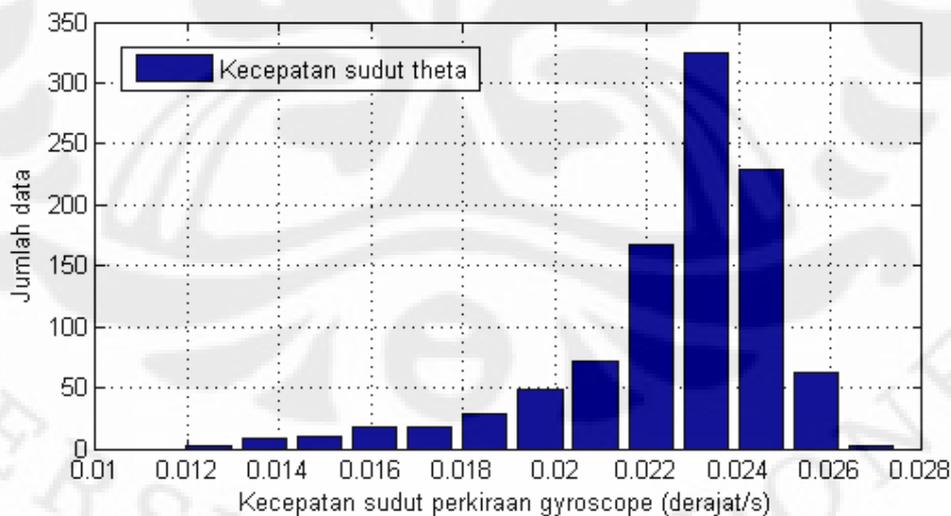
Dengan demikian, skema perubahan bacaan percepatan pada sumbu – x dan y oleh *accelerometer* ( $\ddot{x}_{acc}$  dan  $\ddot{y}_{acc}$ ) ke dalam sudut perkiraan *accelerometer*  $\theta_{acc}$  dan  $\varphi_{acc}$  dapat dinyatakan dalam persamaan berikut:

$$\begin{aligned}\theta_{acc}(t) &= \sin^{-1}\left(\frac{\ddot{x}_{acc}(t)}{g}\right) \\ \varphi_{acc}(t) &= \sin^{-1}\left(\frac{\ddot{y}_{acc}(t)}{g}\right)\end{aligned}\tag{3.2}$$

Sudut  $\theta$  dan  $\varphi$  juga bisa didapatkan dengan melakukan operasi integral pada kecepatan sudut *gyroscope* seperti yang dijabarkan pada persamaan berikut:

$$\begin{aligned}\theta_{gyro}(t) &= \int_0^t \dot{\theta}_{gyro}(t) dt \\ \varphi_{gyro}(t) &= \int_0^t \dot{\varphi}_{gyro}(t) dt\end{aligned}\quad (3.3)$$

Berdasarkan data yang diberikan *gyroscope*, sudut *roll* dan *pitch* dapat diperoleh dengan cara melakukan operasi integral pada data yang dibaca. Namun pada kenyataannya, titik nol pada *gyroscope* yang dibaca oleh sensor bergeser, dimana nilai pergeseran ini adalah *offset* dari data bacaan sensor. Nilai *offset* yang sebenarnya tidak dapat diketahui kecuali dengan membandingkannya dengan sebuah nilai referensi yang lebih akurat. Sebaran nilai  $\dot{\theta}_{gyro}$  pada kecepatan nol derajat per detik dapat dilihat pada gambar 3.7. Terlihat bahwa sebaran nilai bukan berbentuk sebaran gaussian. Terlebih rata-rata sebaran tidak berada di nilai nol seperti yang seharusnya walaupun terlihat bahwa deviasi nilai bacaan tidak sebesar deviasi *accelerometer*. Hal ini disebabkan adanya nilai *offset* yang berubah-ubah nilainya seperti yang dijabarkan pada bagian sebelumnya.



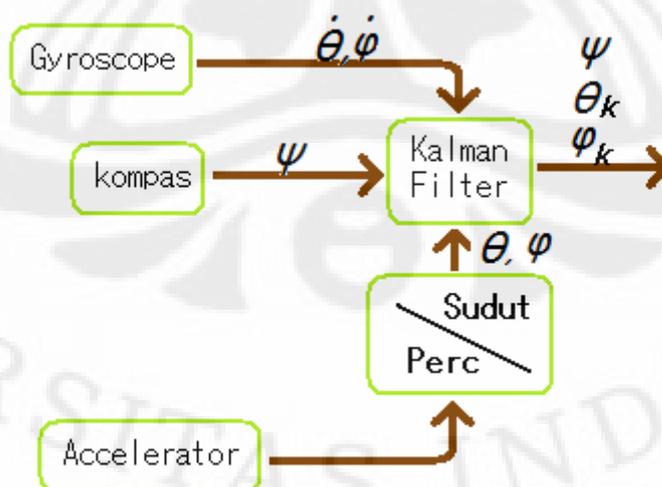
Gambar 3.7. Sebaran pembacaan  $\dot{\theta}_{gyro}$ .

Permasalahan lain muncul karena kecepatan sudut yang diberikan oleh *gyroscope* memiliki derau. Derau yang muncul pada saat pembacaan akan terus

diintegrasikan dan membuat nilai sudut bergeser, pergeseran ini dinamakan sebagai kesalahan *drift*. Selain *drift*, data yang masuk juga dipengaruhi oleh nilai *offset* yang sangat sulit untuk diketahui. Bila nilai *drift* dan *offset* digabungkan menjadi sebuah nilai bias serta memasukkannya ke dalam persamaan (3.3), didapatkan:

$$\begin{aligned}\theta_{gyro}(t) &= \int_0^t \dot{\theta}_{gyro}(t) dt - \int_0^t b_{\theta}(t) dt \\ \varphi_{gyro}(t) &= \int_0^t \dot{\varphi}_{gyro}(t) dt - \int_0^t b_{\varphi}(t) dt\end{aligned}\quad (3.4)$$

Dengan menggunakan persamaan – persamaan estimasi sudut seperti yang dijabarkan di atas, didapatkan 4 buah data bacaan IMU, yaitu 2 untuk sudut *pitch* dan 2 untuk sudut *roll*. Pada pembacaan sudut *pitch*, data pertama datang dari *gyroscope* dengan permasalahan berupa kesalahan *bias* yang bertambah terus menerus, sedangkan data kedua datang dari *accelerometer* dengan permasalahan derau yang besar. Untuk menggabungkan kedua buah sumber data ini, digunakan filter digital berupa *Kalman Filter* untuk mendapatkan nilai yang optimal. Gambaran lengkap dari sistem navigasi dapat dilihat pada gambar 3.8. Sudut robot akan dirujuk dengan notasi  $\theta$  untuk *roll*,  $\varphi$  untuk *pitch*, dan  $\psi$  untuk *yaw*.



Gambar 3.8. Gambaran lengkap sistem navigasi.

Bila persamaan (3.4) diimplementasikan dalam sistem diskrit, maka akan didapatkan persamaan berikut:

$$\begin{aligned}\theta_{gyro}(k) &= \theta_{gyro}(k-1) + \dot{\theta}_{gyro}(k-1) \cdot \Delta t - b_{\theta}(k-1) \cdot \Delta t \\ \varphi_{gyro}(k) &= \varphi_{gyro}(k-1) + \dot{\varphi}_{gyro}(k-1) \cdot \Delta t - b_{\varphi}(k-1) \cdot \Delta t\end{aligned}\quad (3.5)$$

Dimana  $\Delta t$  merupakan konstanta cuplik dengan nilai 0.02. Nilai ini didasarkan pada frekuensi cuplik maksimal sensor yang sebesar 50 Hz. Untuk mengetahui nilai bias dari tiap sudut dari persamaan (3.5), digunakan persamaan ruang keadaan sebagai berikut:

$$\begin{bmatrix} \theta(k+1) \\ e_{\theta}(k+1) \end{bmatrix} = \begin{bmatrix} 1 & -\Delta t \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \theta(k) \\ b_{\theta}(k) \end{bmatrix} + \begin{bmatrix} \Delta t \\ 0 \end{bmatrix} \cdot \dot{\theta}_{gyro}(k)\quad (3.6)$$

$$\theta_{acc}(k) = \begin{bmatrix} 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} \theta(k) \\ b_{\theta}(k) \end{bmatrix}$$

$$\begin{bmatrix} \varphi(k+1) \\ e_{\varphi}(k+1) \end{bmatrix} = \begin{bmatrix} 1 & -\Delta t \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \varphi(k) \\ b_{\varphi}(k) \end{bmatrix} + \begin{bmatrix} \Delta t \\ 0 \end{bmatrix} \cdot \dot{\varphi}_{gyro}(k)\quad (3.7)$$

$$\varphi_{acc}(k) = \begin{bmatrix} 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} \varphi(k) \\ b_{\varphi}(k) \end{bmatrix}$$

Dengan menerapkan *kalman filter* pada persamaan (3.6) dan (3.7), nilai  $\theta$ ,  $\varphi$ ,  $b_{\theta}$ , dan  $b_{\varphi}$  yang sebenarnya dapat dianggap sebagai variabel keadaan ( $X_{\theta}(k) = [\theta(k) \ b_{\theta}(k)]^T$  dan  $X_{\varphi}(k) = [\varphi(k) \ b_{\varphi}(k)]^T$ ) yang nilainya dapat diestimasi berdasarkan masukan sistem  $\dot{\theta}_{gyro}(t)$  dan  $\dot{\varphi}_{gyro}(t)$ , serta berdasarkan keluaran sistem  $y_{\theta}(k) = \theta_{acc}(k)$  dan  $y_{\varphi}(k) = \varphi_{acc}(k)$ .

Seperti yang dapat dilihat pada gambar 3.7, sebaran nilai  $\dot{\theta}_{gyro}$  pada kecepatan nol derajat per detik bukan berbentuk sebaran gaussian. Terlebih rata-rata sebaran tidak berada di nilai nol seperti yang seharusnya walaupun terlihat

bahwa deviasi nilai bacaan tidak sebesar deviasi *accelerometer*. Hal ini disebabkan adanya nilai bias yang berubah-ubah nilainya seperti yang dijabarkan pada bagian sebelumnya. Sebaran bukan berbentuk gaussian menandakan bahwa nilai variansi pembacaan sensor berubah, sedangkan nilai rata-rata bukan nol menandakan rata – rata derau sistem berubah. Kedua hal ini berlawanan dengan seperti asumsi kalman filter. Namun dengan pemilihan parameter filter yang tepat, derau dapat dihilangkan dengan kinerja yang cukup baik.

Sebelum memulai perhitungan, ada beberapa variabel *kalman filter* yang harus diinisiasi terlebih dahulu. Variabel tersebut adalah variabel keadaan  $X_\theta(0)$  dan  $X_\phi(0)$ , matriks kovarian  $P_\theta$  dan  $P_\phi$ , matriks kovarian derau pada variabel keadaan  $Q$  dan matriks kovarian derau pada variabel keluaran  $R$ .

Nilai inisial variabel keadaan  $X_\theta(0)$  dan  $X_\phi(0)$  dipilih  $[0 \ 0]^T$  karena diasumsikan bahwa robot memulai terbang dari bidang yang datar. Sedangkan nilai inisial matriks kovarian  $P_\theta$  dan  $P_\phi$  dipilih  $\begin{bmatrix} 10000 & 0 \\ 0 & 10000 \end{bmatrix}$  untuk mengantisipasi jika bidang terbang robot tidak terlalu datar. Nilai variabel  $Q$  dipilih  $\begin{bmatrix} 0.4 & 0 \\ 0 & 0 \end{bmatrix}$  karena nilai kecepatan sudut yang didapat dari sensor *gyroscope* memiliki error dengan deviasi sangat kecil, sedangkan nilai variabel  $R$  dipilih 100 karena berdasarkan percobaan, derau sensor *accelerometer* sangat besar dan sangat dipengaruhi oleh gangguan gaya luar. Penentuan nilai matriks kovarian  $P_\theta$  dan  $P_\phi$ , varians derau pada variabel keadaan  $Q$  dan varian derau pada variabel keluaran  $R$  dipilih berdasarkan penalaan secara manual selama masa percobaan.

Ringkasan parameter *kalman filter* yang dipakai dalam sistem navigasi dapat dilihat pada tabel 3.2.

Tabel 3.1. Daftar nilai variabel dan konstanta yang dipakai dalam kalman filter.

Variabel / Konstanta	Nilai	Keterangan
$A$	$\begin{bmatrix} 1 & -dt \\ 0 & 1 \end{bmatrix}$	Matriks transisi sistem navigasi.
$B$	$\begin{bmatrix} dt \\ 0 \end{bmatrix}$	Vektor masukan sistem.
$C$	$[1 \ 0]$	Vektor keluaran sistem.
$Q$	$\begin{bmatrix} 0.4 & 0 \\ 0 & 0 \end{bmatrix}$	Matriks varian derau sistem.
$R$	100	Varian derau keluaran sistem.
$X_\theta(0)$ dan $X_\varphi(0)$	$\begin{bmatrix} 0 \\ 0 \end{bmatrix}$	Nilai inisial variabel keadaan
$P_\theta$ dan $P_\varphi$	$\begin{bmatrix} 10000 & 0 \\ 0 & 10000 \end{bmatrix}$	Nilai inisiasi matriks kovarian sistem.
$\Delta t$	0.02	Waktu cuplik.

Berdasarkan persamaan (3.6) dan (3.7), maka terdapat dua macam *kalman filter* untuk mengestimasi nilai  $\theta$  dan  $\varphi$  yang dijalankan dalam satu loop. Karena kedua sistem sama, maka matriks dan vektor transisi persamaan ruang keadaan dijadikan satu. Penggabungan sistem *kalman filter*  $\theta$  dan  $\varphi$  ditunjukkan untuk menghemat ruang penyimpanan pada mikrokontroler. Sistem navigasi dapat dijabarkan dalam langkah - langkah berikut ini:

1.  $y_\theta \leftarrow \sin^{-1} \left( \frac{\ddot{x}_{acc}(t)}{g} \right)$
2.  $y_\varphi \leftarrow \sin^{-1} \left( \frac{\ddot{y}_{acc}(t)}{g} \right)$

3. Untuk tiap  $i \leftarrow (\theta \text{ atau } \varphi)$ :

Prediksi:

$$5.1. \hat{X}_i(k) \leftarrow A.X_i(k-1) + B.u_i(k-1)$$

$$5.2. \hat{P}(k) \leftarrow A.P(k-1).A^T + Q$$

Koreksi:

$$5.3. K_i \leftarrow \frac{\hat{P}_i(k).C^T}{C.\hat{P}_i(k).C^T + R}$$

$$5.4. X_i(k) \leftarrow \hat{X}_i(k) + K_i.(y_i(k) - C.\hat{X}_i(k)).$$

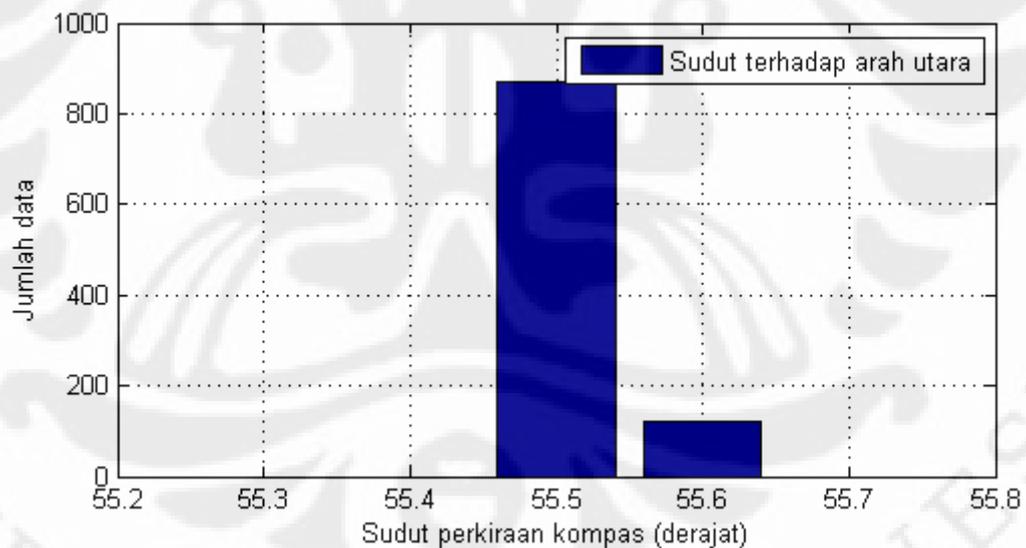
$$5.5. P_i(k) \leftarrow \hat{P}_i(k) - K_i.C.\hat{P}_i(k)$$

4.  $\theta \leftarrow X_\theta(1)$

5.  $\varphi \leftarrow X_\varphi(1)$

6.  $\psi \leftarrow \psi_{kompas}$

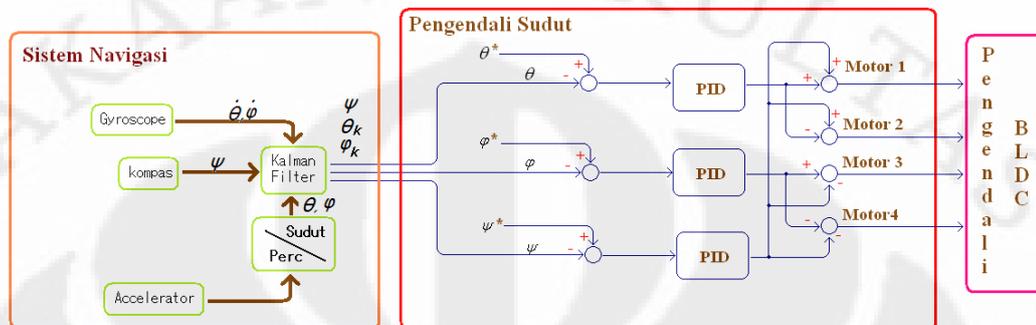
Kalman filter tidak dilakukan pada parameter  $\psi$  karena derau yang kecil dan pengendalian  $\psi$  tidak membutuhkan ketepatan karena tidak terlalu mempengaruhi kestabilan sistem. Sebaran nilai bacaan kompas yang kecil dapat dilihat pada gambar 3.9.



Gambar 3.9. Sebaran bacaan sensor kompas.

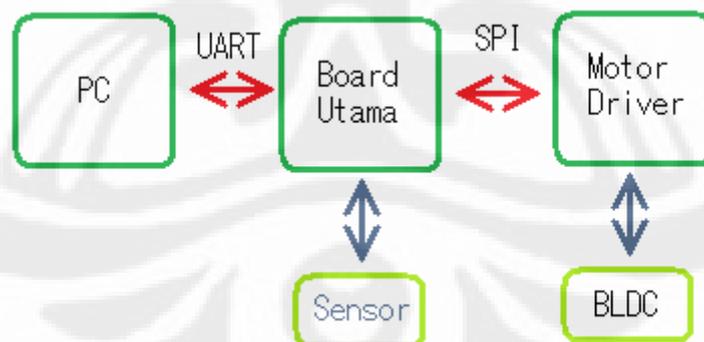
### 3.4 Implementasi Perangkat Keras

Gambaran lengkap dari sistem pengendali keseluruhan dapat dilihat pada gambar 3.10.



Gambar 3.10. Skema lengkap sistem pengendali.

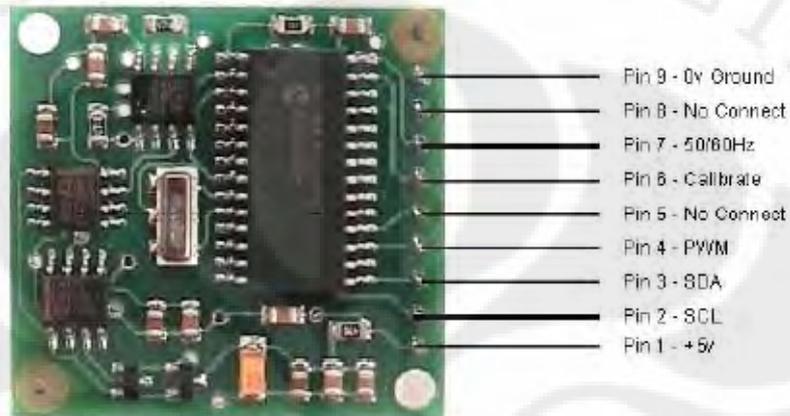
Implementasi pengendali dilakukan pada sistem mikrokontroler yang terdiri dari 5 buah mikrokontroler. 4 buah mikrokontroler digunakan untuk mengendalikan BLDC, sedangkan 1 mikrokontroler digunakan untuk menjalankan sistem navigasi, pengendalian sudut, dan komunikasi dengan komputer pusat yang berada di tanah (*ground station*).



Gambar 3.11. Skema umum perangkat keras.

Komunikasi antar mikrokontroler menggunakan komunikasi serial dengan protokol *serial peripheral interface* (SPI). SPI dipakai dengan alasan kecepatan transfer yang tinggi dan kemudahan implementasi perangkat lunak serta perangkat

keras. Sedangkan komunikasi antara mikrokontroller pusat dengan sensor menggunakan *Inter-Integrated Circuit* (I<sup>2</sup>C) yang merupakan standar komunikasi yang didukung oleh sensor – sensor yang dipakai, yaitu VS-IX001 untuk IMU dan CMPS03 untuk kompas digital.



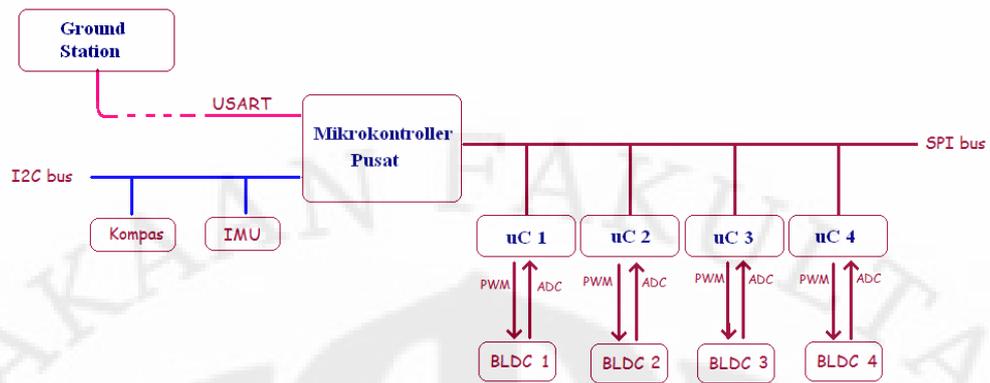
Gambar 3.12. Modul CMPS03.



Gambar 3.13 Modul IX001.

Masing – masing sensor dicuplik dalam rentang waktu 0.02 s. Waktu cuplik ini memberikan frekuensi kerja robot sebesar 50 Hz. Frekuensi ini dipilih berdasarkan frekuensi cuplik sensor yang paling tinggi, yaitu sensor IMU IX001 dengan frekuensi cuplik maksimal 50 Hz.

Skematik lengkap dari implementasi perangkat keras dapat dilihat pada gambar 3.15.



Gambar 3.14. Gambaran lengkap implementasi sistem.

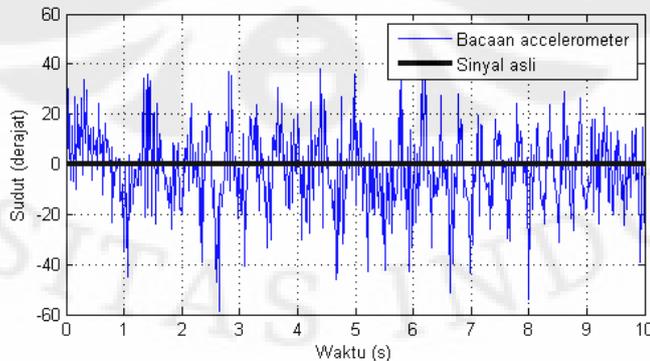
## BAB 4

### PENGUJIAN DAN ANALISA

#### 4.1 Pengujian Sistem Navigasi

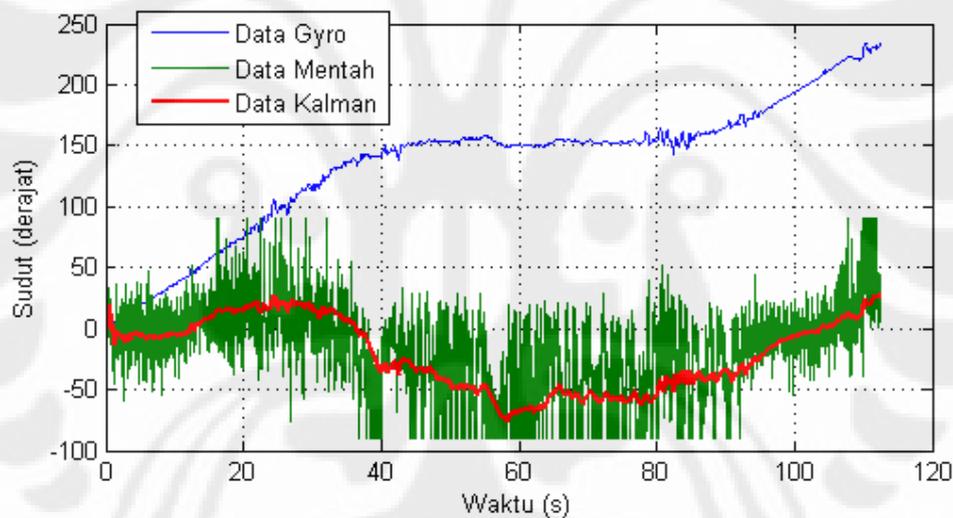
Parameter *kalman filter* yang perlu diperhatikan adalah parameter derau masukan  $Q$  dan derau keluaran  $R$ . Parameter  $Q$  ditentukan sebagai matriks  $\begin{bmatrix} 0.4 & 0 \\ 0 & 0 \end{bmatrix}$  karena hasil pengukuran menunjukkan bahwa masukan sistem ( $\dot{\theta}_{gyro}$  atau  $\dot{\phi}_{gyro}$ ) merupakan data yang cukup akurat dengan standar deviasi sekitar 0.02. Penentuan angka 0.4 didasarkan pada eksperimen, sedangkan penentuan nilai 0 bagi data  $Q$  yang lain dikarenakan derau pada masukan hanya mempengaruhi secara langsung variabel keadaan pertama, yaitu  $\theta$  atau  $\phi$ . Pada pembahasan kalman filter, pembahasan akan difokuskan pada satu sudut saja, yaitu  $\phi$ , karena karakteristik sinyal  $\theta$  atau  $\phi$  yang sama.

Pemilihan nilai  $R$  yang besar ( $R = 100$ ) juga didasarkan pada percobaan. Percobaan pengambilan data untuk sudut robot  $\theta = 0$  menunjukkan bahwa derau yang terjadi pada pembacaan sensor *accelerator*  $\theta_{acc}$  sangat besar. Hasil percobaan ini dapat dilihat pada gambar 4.1. Sebaran data percobaan ketika diimplementasikan ke dalam sistem lebih besar dari sebaran data sensor pada pembahasan bab 3. Hal ini dikarenakan pada percobaan ini ada gaya dorong angin yang mengakibatkan asumsi tidak adanya gaya luar selain gaya gravitasi tidak berlaku lagi.



Gambar 4.1. Perbandingan data  $\theta_{acc}$  pada sudut 0 derajat.

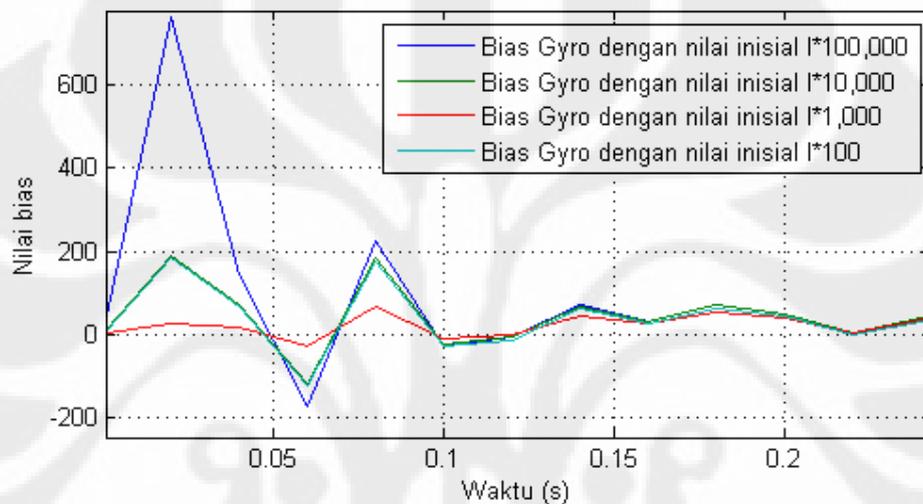
Gambar 4.2 menampilkan hasil percobaan untuk mengestimasi nilai  $\varphi$  dari data IMU dengan menggunakan parameter tabel 3.1. Terlihat bahwa hanya mengandalkan estimasi  $\varphi$  oleh *gyroscope* ( $\varphi_{gyro}$ ) saja tidak cukup, karena nilai  $\varphi_{gyro}$  bergeser seiring waktu dikarenakan adanya error yang diintegrasikan ke dalam nilai  $\varphi_{gyro}$ . Nilai estimasi  $\varphi$  oleh *accelerometer* ( $\varphi_{acc}$ ) juga tidak dapat begitu diandalkan karena derau yang sangat besar. Selain itu, pada detik ke 36 – 40 serta detik ke 45 – 60, sudut  $\varphi_{acc}$  sempat melewati batas minimum nilai sin, atau perbandingan percepatan robot searah sumbu – y terhadap gravitasi lebih kecil dari -1. Hal ini dikarenakan adanya derau yang besar sehingga asumsi tidak adanya percepatan non-gravitasi searah sumbu – y tidak begitu benar lagi.



Gambar 4.2. Perbandingan perkiraan sudut  $\varphi$  sebelum dan sesudah blok *kalman filter*.

Pemilihan nilai inisial matriks kovarian  $P_\theta$  dan  $P_\varphi$  sangat mempengaruhi kinerja sistem. Pemilihan matriks kovarian yang besar menyebabkan variabel keadaan cepat beradaptasi namun beresilasi pada waktu-waktu awal. Sedangkan pemilihan matriks kovarian yang kecil menyebabkan sistem lebih stabil namun lambat beradaptasi. Jika nilai matriks kovarian terlalu besar, maka kestabilan sistem pengendali secara keseluruhan dapat berkurang, namun nilai kovarian yang

terlalu kecil menyebabkan pengendali tidak responsif dan menyebabkan kestabilan sistem juga berkurang. Oleh karena itu dilakukan ujicoba beberapa nilai inisial, dan kemudian diambil yang terbaik. Grafik pengujian nilai inisial tersebut dapat dilihat pada 4.3. Berdasarkan percobaan, nilai inisial  $P_\theta$  dan  $P_\phi$  sebesar  $\begin{bmatrix} 10000 & 0 \\ 0 & 10000 \end{bmatrix}$  dipilih karena nilai tersebut terbukti mampu membuat sistem stabil namun dengan adaptasi yang cepat.



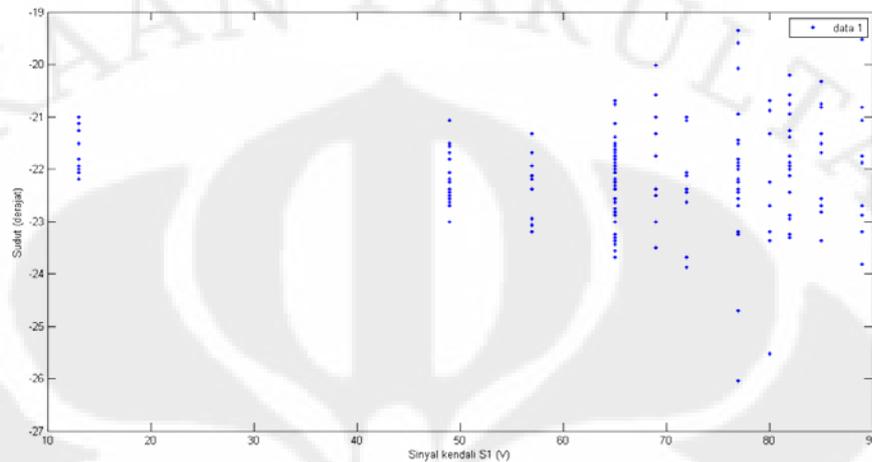
Gambar 4.3. Efek perubahan nilai  $P$  pada data yang dipakai gambar 4.2.

## 4.2 Analisa Identifikasi Sistem

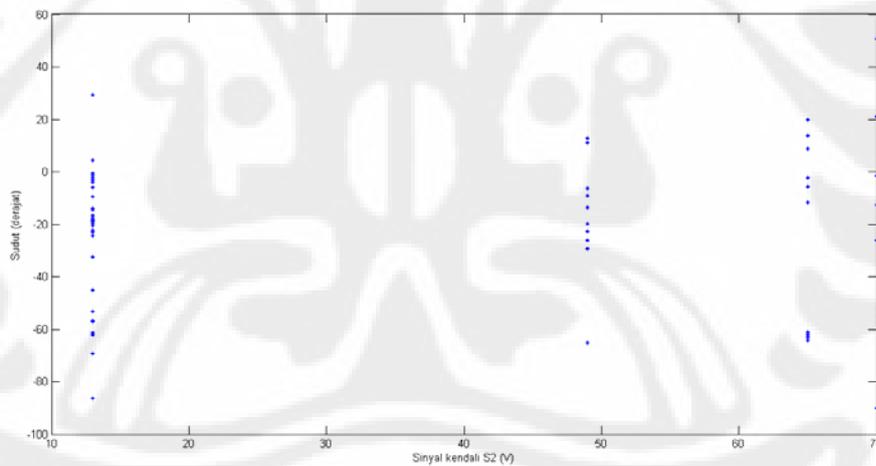
Sebelum mengidentifikasi sistem dengan menggunakan *least-square*, informasi pertama yang perlu diketahui dari sistem adalah analisa statik sistem. Analisa statik diperlukan untuk menentukan daerah kerja dari sistem. Respon statik sistem  $\theta$  terhadap  $S_1$  dapat dilihat pada gambar 4.4, sedangkan respon statik  $\phi$  terhadap  $S_2$  dapat dilihat pada gambar 4.5. Respon statik sistem didapatkan berdasarkan data estimasi *kalman filter* sesuai dengan sinyal kendali yang diberikan kepada sistem.

Kestabilan sistem dapat dilakukan dengan menganalisa keadaan statik sistem. Dengan mengasumsikan waktu keadaan tunak adalah 3 detik, maka data statik dapat dicari. Hubungan antara nilai sinyal kendali terhadap sudut pada

waktu tunak dapat dilihat pada gambar 4.4 dan 4.5. Nilai keadaan tunak yang tersebar luas pada nilai sinyal kendali besar menyatakan sistem tidak stabil pada nilai sinyal kendali besar.



Gambar 4.4. Respon statik untuk sistem  $\varphi$  terhadap  $S_2$ .

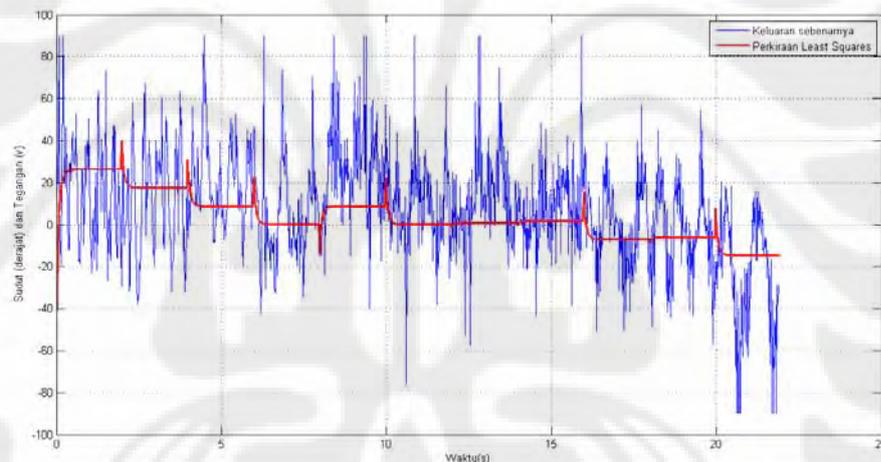


Gambar 4.5. Respon statik untuk sistem  $\theta$  terhadap  $S_1$ .

Algoritma *least – square* digunakan untuk mencari hubungan  $\theta$  terhadap  $S_1$  pada daerah kerja 8 volt sesuai dengan algoritma yang dijabarkan pada bab 3, hasil identifikasi memberikan fungsi alih sebagai berikut:

$$\frac{\theta}{S_1} = \frac{z^{-1}(1.361 + 1.65z^{-1})}{1 - 0.3557z^{-1} - 0.313z^{-2}} \quad (4.1)$$

Perbandingan antara hasil identifikasi dan keluaran sistem asli dapat dilihat pada gambar 4.6. Dari gambar tersebut, terlihat bahwa bacaan sensor memiliki derau yang sangat besar, dan *least-square* bekerja seperti filter dari data bacaan sensor. Hal ini dikarenakan sifat algoritma *least-square* yang berusaha mengecilkan kuadrat terkecil. Estimasi dilakukan berdasarkan bacaan sensor *accelerometer*, atau dengan kata lain pada identifikasi dengan *least-square*,  $\theta$  diasumsikan adalah  $\theta_{acc}$ . Asumsi ini didasarkan karena data  $\theta$  yang diambil hanya berkisar pada nilai yang kecil. Pada kenyataannya, data  $\theta_{acc}$  tetap memberikan derau yang cukup besar seperti yang bisa dilihat pada gambar 4.6, namun dianggap fungsi alih yang didapat sudah cukup mewakili sistem.



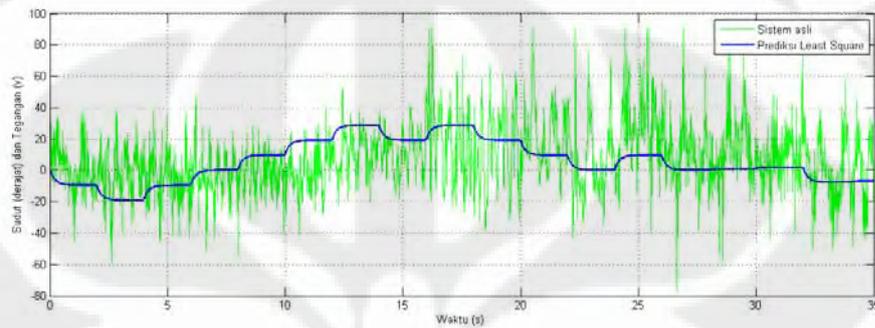
Gambar 4.6. Hasil identifikasi *least-square* untuk sistem  $\theta$  terhadap  $S_1$ .

Karena sistem  $\theta$  terhadap  $S_1$  memiliki sifat yang sama persis dengan sistem  $\varphi$  terhadap  $S_2$ , maka fungsi alih sistem  $\varphi$  terhadap  $S_2$  juga dianggap sama, atau

$$\frac{\varphi}{S_2} = \frac{z^{-1}(1.361 + 1.65z^{-1})}{1 - 0.3557z^{-1} - 0.313z^{-2}} \quad (4.2)$$

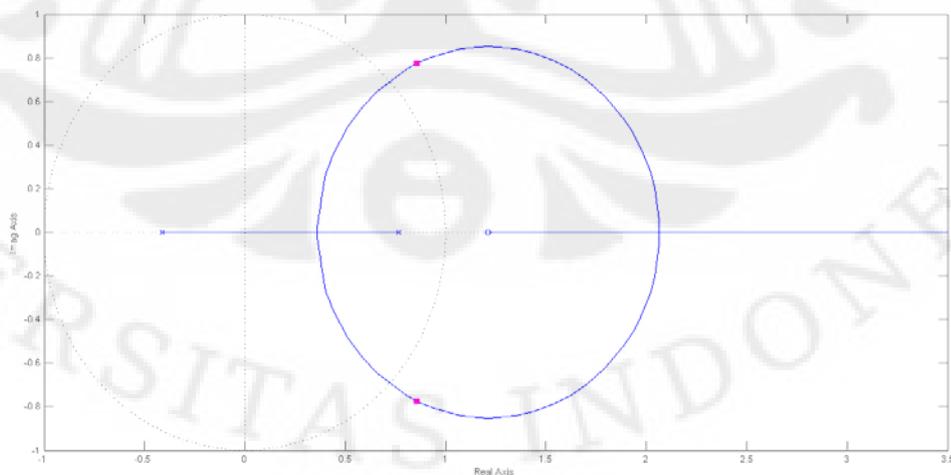
Analisa *least-square* pada sistem  $\psi$  terhadap  $S_2$  dapat dilihat pada gambar 4.7. Sedangkan persamaan sistem yang diperkirakan oleh *least-square* adalah:

$$\frac{\psi}{S_3} = \frac{z^{-1}(0.000115 + 0.1702z^{-1})}{1 - 0.4422z^{-1} - 0.3794z^{-2}} \quad (4.3)$$

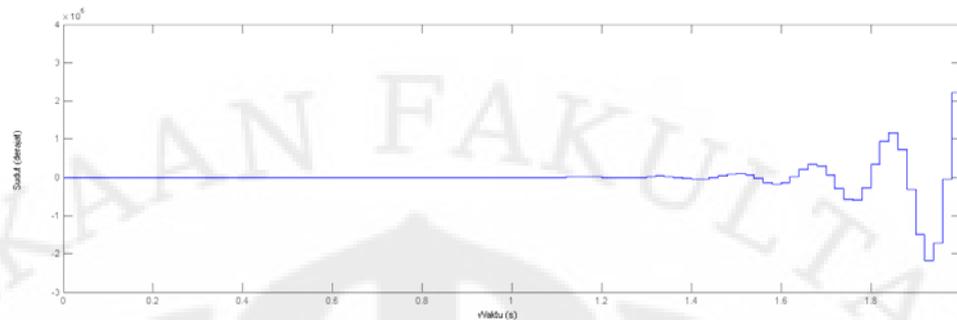


Gambar 4.7. Hasil identifikasi *least-square* untuk sistem  $\psi$  terhadap  $S_3$ .

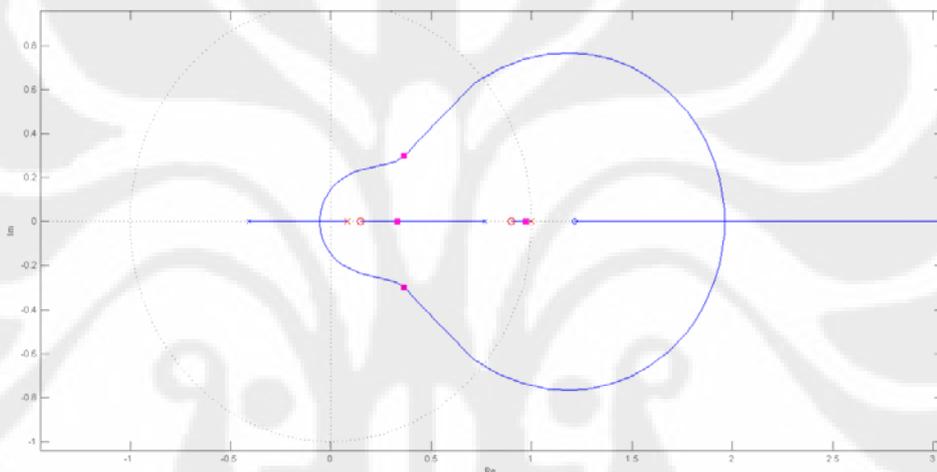
Gambar 4.8 dan 4.9 memperlihatkan analisa letak kedudukan akar dan respon dari sistem  $\theta$  terhadap  $S_1$ . Terlihat bahwa sistem memiliki sifat tidak stabil. Desain pengendali PID ditunjukkan untuk menstabilkan sistem, mempercepat respon sistem, dan sekaligus menghilangkan kesalahan keadaan tunak dari sistem.



Gambar 4.8. Plot letak kedudukan akar untuk sistem  $\theta$  terhadap  $S_1$ .



Gambar 4.9. Respon loop tertutup untuk *set point* 1 untuk sistem  $\theta$  terhadap  $S_1$ .

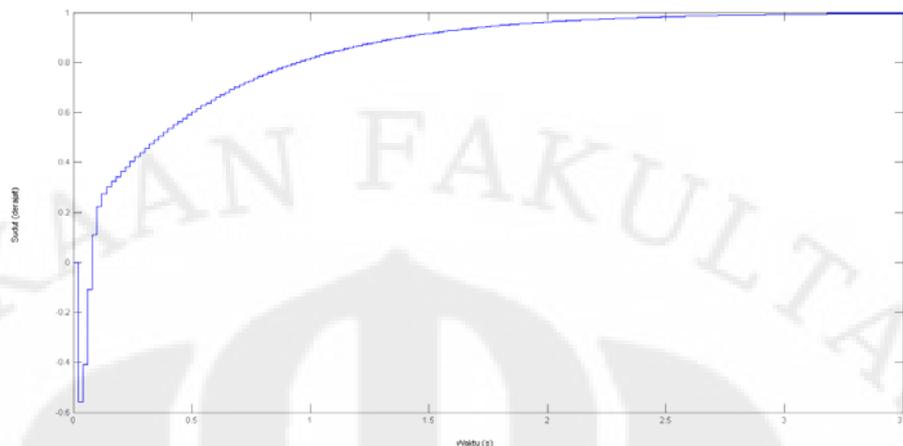


Gambar 4.10. Desain pengendali PID untuk  $\theta$  terhadap  $S_1$ .

Pengendali yang didesain untuk sistem di atas dapat dilihat pada gambar 4.10. Fungsi alih pengendali  $\theta$  terhadap  $S_1$  dan  $\varphi$  terhadap  $S_2$  adalah:

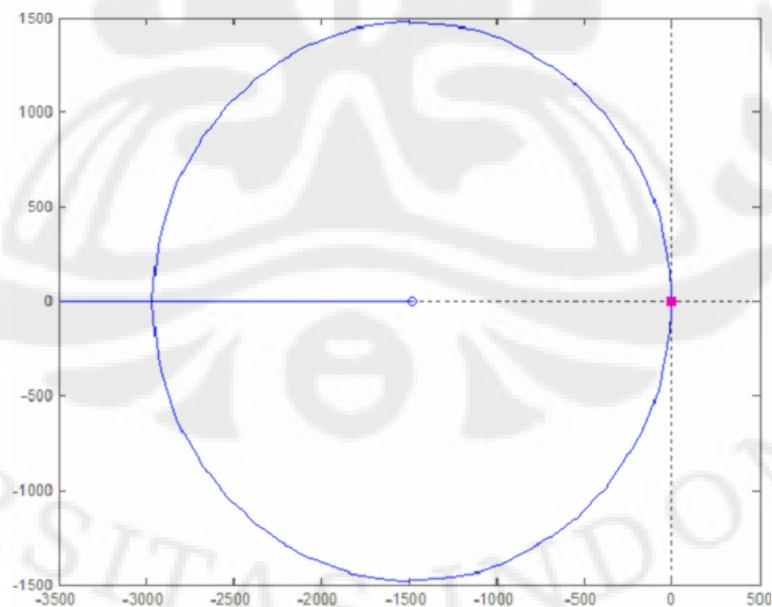
$$C = \frac{0.43442 \cdot (1 - 0.9z^{-1})(1 - 0.15z^{-1})}{(1 - z^{-1})(1 - 0.08z^{-1})} \quad (4.4)$$

Simulasi sstem berpengendali dalam loop tertutup dapat dilihat pada gambar 4.11. Pengendali dapat membawa sistem menuju keadaan tunak dalam waktu sekitar 1.5 detik, dan dengan kesalahan keadaan tunak nol. Terdapat undershoot karena dinamika sistem seperti yang terlihat pada gambar 4.6.

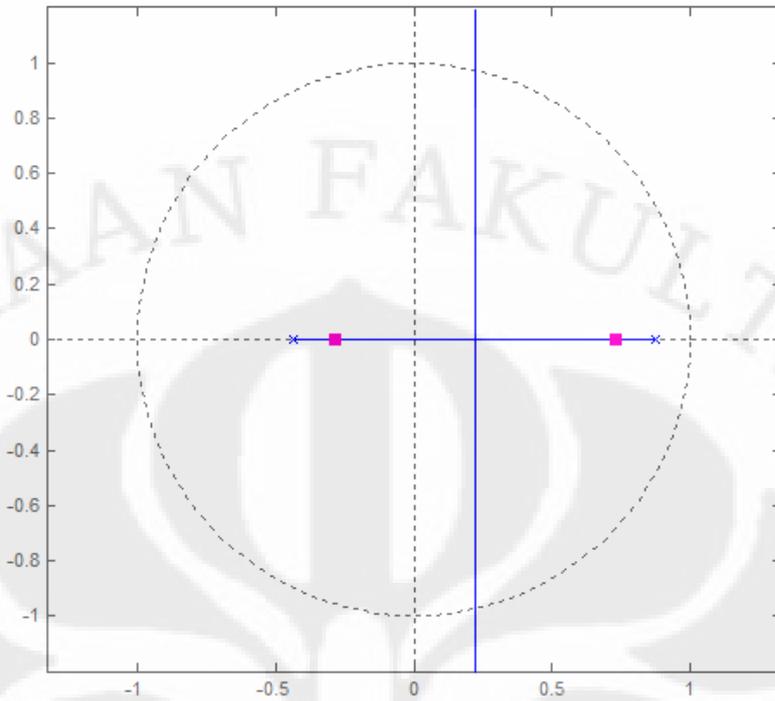


Gambar 4.11. Respon simulasi sistem  $\theta$  dan  $\varphi$  berpengendali terhadap *set point* 1.

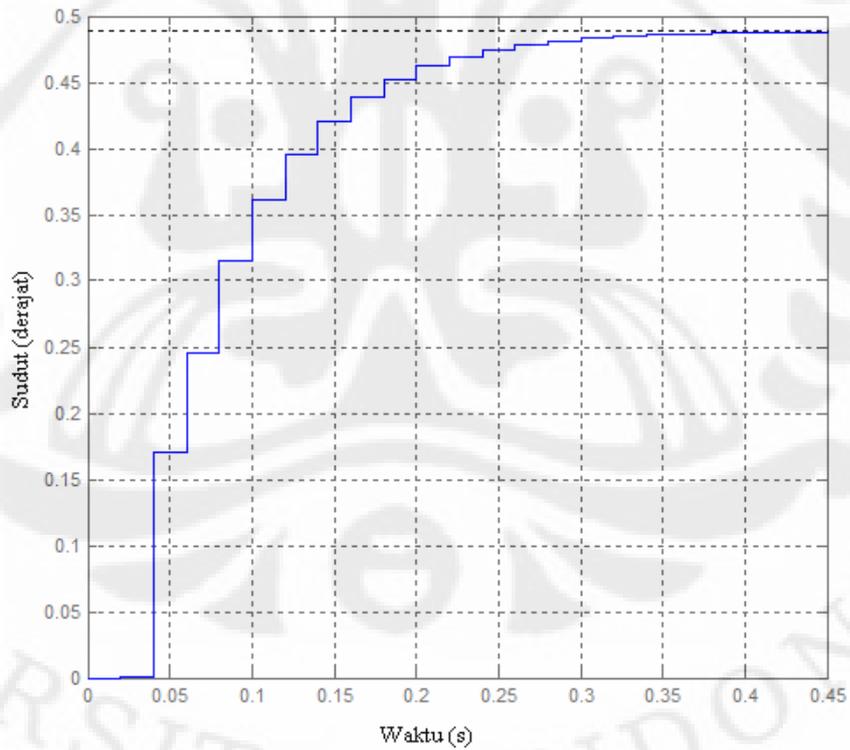
Karakteristik sistem  $\psi$  terhadap  $S_3$  didapat dengan menganalisa gambar 4.12, 4.13, dan 4.14. Gambar 4.13 menunjukkan bahwa sistem memiliki sifat stabil karena semua akar persamaan karakteristik sistem berada dalam *unit circle*, namun berdasarkan gambar 4.14, error keadaan tunak dari sistem terlalu besar, oleh karena itu, dirancang pengendali yang dapat menghilangkan error keadaan tunak.



Gambar 4.12. Plot letak kedudukan akar untuk sistem  $\psi$  terhadap  $S_3$ .



Gambar 4.13. Plot letak kedudukan akar untuk sistem  $\psi$  dalam *range unit circle*.

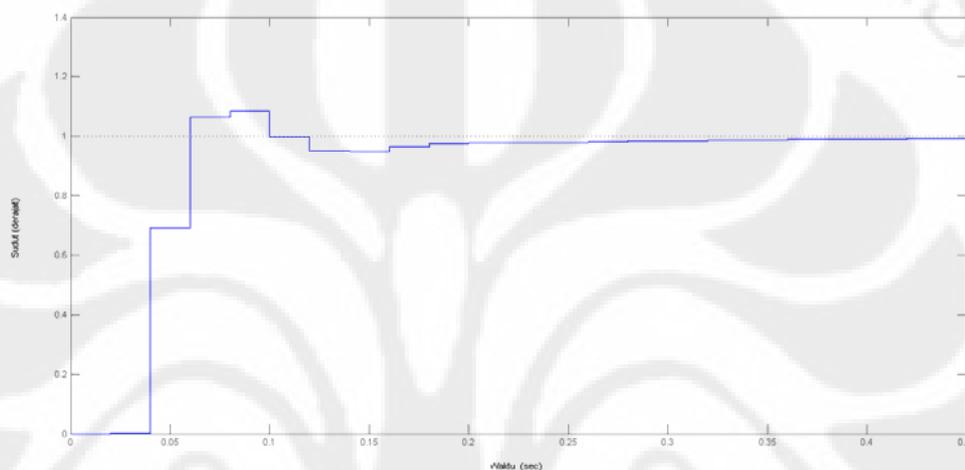


Gambar 4.14. Respon loop tertutup untuk *set point* 1 untuk sistem  $\psi$ .

Karena respon sistem  $\psi$  terhadap  $S_3$  yang terlihat pada gambar 4.14 cepat, maka pengendali untuk sistem ini menggunakan pengendali PI, yaitu:

$$C = \frac{4.058(1-0.9z^{-1})}{(1-z^{-1})} \quad (4.5)$$

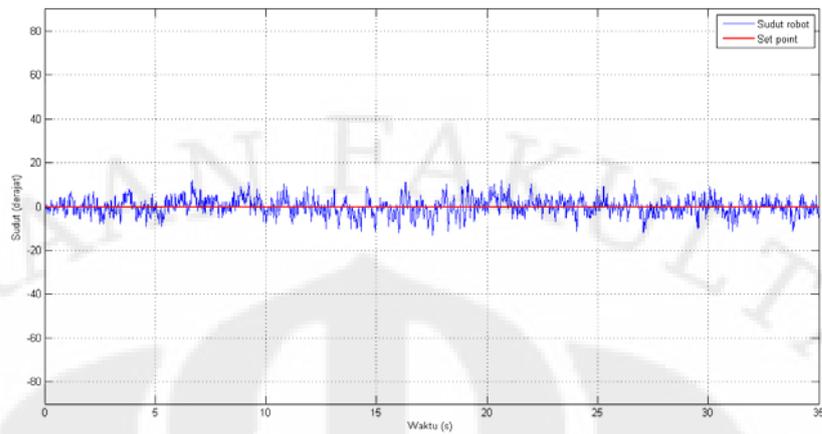
Simulasi sstem berpengendali dalam loop tertutup dapat dilihat pada gambar 4.15.



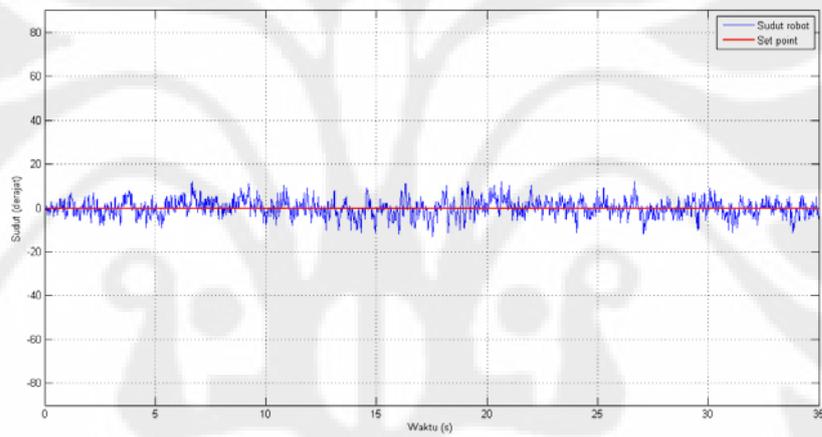
Gambar 4.15. Respon simulasi sistem  $\psi$  berpengendali terhadap *set point* 1.

Hasil implementasi pengendali ke dalam sistem nyata dapat dilihat pada gambar 4.16. Hasil yang didapat cukup memuaskan karena pengendali dapat membawa robot mengikuti sinyal referensi 0 derajat dengan kesalahan yang tidak melebihi 20 derajat, walaupun sensor yang dipakai mengandung derau yang sangat besar.

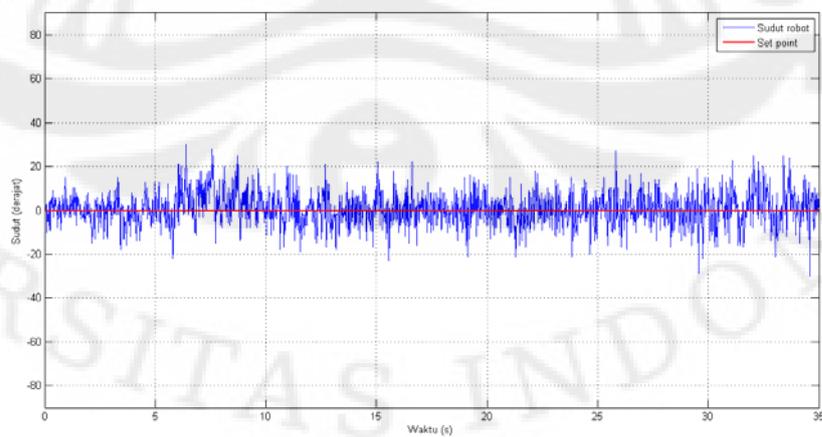
Walaupun hasil pengendali  $\psi$  lebih buruk daripada kedua pengendali yang lain, karena pengendali  $\psi$  tidak mempengaruhi kestabilan sistem secara signifikan maka hasil yang didapat cukup memuaskan. Hasil yang buruk ini kemungkinan karena waktu *settling time* pengendali yang lebih lambat daripada kedua pengendali yang lain.



Gambar 4.16. Implementasi pengendali pada sistem  $\theta$  terhadap  $S_1$ .



Gambar 4.17. Implementasi pengendali pada sistem  $\varphi$  terhadap  $S_2$ .



Gambar 4.18. Implementasi pengendali pada sistem  $\psi$  terhadap  $S_3$ .

## BAB 5

### KESIMPULAN

Dari keseluruhan pembahasan dalam skripsi ini dapat disimpulkan beberapa hal, yaitu :

1. Pengendalian sudut quadrotor yang merupakan sistem MIMO dengan 3 masukan dan 4 keluaran dapat direalisasikan dengan menggunakan pendekatan SISO.
2. Pemodelan *least-square* dapat digunakan untuk memodelkan quadrotor.
3. Sistem navigasi quadrotor yang memiliki sifat derau yang sangat tinggi dapat diatasi dengan *Kalman filter* dan dapat menghasilkan hasil yang cukup memuaskan.
4. Skema pengendali PID dapat mengendalikan sudut quadrotor tanpa bantuan manusia dengan hasil yang cukup memuaskan. Dimana sudut  $\theta$  dan  $\varphi$  dapat dikendalikan dengan kesalahan 5 derajat, dan sudut  $\psi$  dapat dikendalikan dengan kesalahan 20 derajat dari referensi.

**DAFTAR PUSTAKA**

Fisher, Michael. (2006). ATTITUDE STABILISATION OF A QUADROTOR AIRCRAFT. *Faculty of Engineering and Physical Systems*. Central Queensland University.

Welch, G & Bishop, G. (2004). *An Introduction to the Kalman Filter*. Department of Computer Sciences, University of North Carolina, Chappel Hill.

Bouabdallah1, Samir & Siegwart, Roland. (2007). *Towards Intelligent Miniature Flying Robots*. Autonomous Systems Lab, EPFL.

Hoffmann, Huang, et. Al. (2007). *Quadrotor Helicopter Flight Dynamics and Control: Theory and Experiment*. American Institute of Aeronautics and Astronautics , Inc.

**LAMPIRAN****Lampiran 1. Foto quadrotor yang digunakan**

**Lampiran 2. Sistem pengetesan robot.**

