



TUGAS AKHIR - EE184801

**PERANCANGAN SISTEM NAVIGASI BERBASIS  
INS-GPS MENGGUNAKAN FILTER KALMAN  
UNTUK MENINGKATKAN PRESISI HASIL NAVIGASI**

Ananta Muhammad  
NRP 07111540000115

Dosen Pembimbing  
Ir. Rusdhianto Effendi A.K., MT.  
Mochammad Sahal, ST., M.Sc.

DEPARTEMEN TEKNIK ELEKTRO  
Fakultas Teknologi Elektro  
Institut Teknologi Sepuluh Nopember  
Surabaya 2019

*[Halaman ini sengaja dikosongkan]*



**TUGAS AKHIR - EE184801**

**PERANCANGAN SISTEM NAVIGASI BERBASIS  
INS-GPS MENGGUNAKAN FILTER KALMAN  
UNTUK MENINGKATKAN PRESISI HASIL NAVIGASI**

Ananta Muhammad  
NRP 07111540000115

Dosen Pembimbing  
Ir. Rusdhianto Effendi A.K., MT.  
Mochammad Sahal, ST., M.Sc.

DEPARTEMEN TEKNIK ELEKTRO  
Fakultas Teknologi Elektro  
Institut Teknologi Sepuluh Nopember  
Surabaya 2019

*[Halaman ini sengaja dikosongkan]*



**FINAL PROJECT - EE184801**

***INS-GPS BASED NAVIGATION SYSTEM DESIGN  
USING KALMAN FILTER FOR IMPROVING  
PRECISION OF NAVIGATION SOLUTIONS***

Ananta Muhammad  
NRP 07111540000115

*Supervisor*  
Ir. Rusdhianto Effendi A.K., MT.  
Mochammad Sahal, ST., M.Sc.

***DEPARTMENT OF ELECTRICAL ENGINEERING  
Faculty of Electrical Technology  
Institut Teknologi Sepuluh Nopember  
Surabaya 2019***

*[Halaman ini sengaja dikosongkan]*

## **PERNYATAAN KEASLIAN TUGAS AKHIR**

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya dengan judul

**“Perancangan Sistem Navigasi Berbasis INS-GPS Menggunakan Filter Kalman Untuk Meningkatkan Presisi Hasil Navigasi”**

adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka.

Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Surabaya, 30 Mei 2019



Ananta Muhammad  
Nrp: 07111540000115

*[Halaman ini sengaja dikosongkan]*



**PERANCANGAN SISTEM NAVIGASI BERBASIS  
INS-GPS MENGGUNAKAN FILTER KALMAN  
UNTUK MENINGKATKAN PRESISI HASIL  
NAVIGASI**

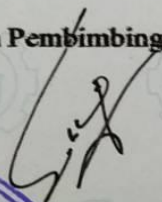
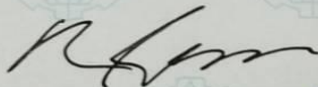
**TUGAS AKHIR**

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

Menyetujui:

Dosen Pembimbing I

Dosen Pembimbing II



Ir. Rusdhianto Effendi A.K., MT.

Mochammad Sahal, ST., M.Sc.

NIP. 19570424 198502 1001

NIP. 19701119 199802 1002



*[Halaman ini sengaja dikosongkan]*

# Perancangan Sistem Navigasi Berbasis INS-GPS Menggunakan Filter Kalman Untuk Meningkatkan Presisi Hasil Navigasi

Ananta Muhammad  
0711154000115

Pembimbing I : Ir. Rusdhianto Effendi A.K., MT.  
Pembimbing II : Mochammad Sahal, ST., M.Sc.

## ABSTRAK

Sistem navigasi menjadi bidang riset yang sangat menarik untuk dilakukan. Sekarang, sistem navigasi banyak dibutuhkan untuk keperluan kontrol *autopilot*, *guidance* dan *tracking*. Pada sistem navigasi, Sebelum wahana bergerak, sistem navigasi membutuhkan waktu untuk *alignment*. *Alignment* bertujuan untuk menentukan posisi, kecepatan dan *attitude* awal serta bias awal dari sensor yang digunakan pada sistem navigasi. Sistem navigasi dapat menggunakan *Inertial Navigation System* (INS) dan *Global Positioning Systems* (GPS). Di dalam INS terdapat dua subsistem yaitu sensor dan prosesor dimana sensor terdiri dari *accelero* dan *gyro*. Prosesor digunakan untuk mengolah sinyal dari sensor yaitu transformasi *frame* dari *body frame* ke *navigation frame* serta perhitungan integral sinyal sensor menjadi hasil navigasi berupa posisi, kecepatan dan sikap sudut (*attitude*) pada sumbu *North*, *East*, dan *Down* (NED). Proses pengolahan ini disebut mekanisasi. Bila hanya menggunakan proses mekanisasi nilai RMSE adalah  $[0.4904^\circ, 0.9215^\circ, 19.3315^\circ]$  untuk *attitude*,  $[1.7179, 4.8001, 0.2404]$ m/s untuk kecepatan dan  $[0.0009^\circ, 0.0012^\circ, 5.9941m]$  untuk posisi. RMSE tersebut relatif besar yang disebabkan oleh *noise* sehingga untuk mengatasi masalah tersebut pada Tugas Akhir ini akan diintegrasikan INS dan GPS untuk mendapatkan model pengukuran eror yang akan diestimasi oleh Filter Kalman untuk meningkatkan presisi hasil navigasi. setelah dilakukan simulasi dari hasil rancangan pada Tugas Akhir ini didapatkan nilai RMSE  $[0.2326, 0.5388, 2.7953]$  untuk *attitude*,  $[0.0638, 0.0586, 0.1430]$  untuk kecepatan dan  $[0.00002^\circ, 0.000007^\circ, 1.1485]$  untuk posisi.

**Kata Kunci:** INS, Navigasi, GPS, Kalman filter, Mekanisasi, *noise*

*[Halaman ini sengaja dikosongkan]*

# ***INS-GPS Based Navigation System Design Using Kalman Filter For Improving Precision of Navigation Solutions***

Ananta Muhammad  
07111540000115

*Supervisor I* : Ir. Rusdhianto Effendi A.K., MT.  
*Supervisor II* : Mochammad Sahal ST., M.Sc.

## ***ABSTRACT***

*Navigation system in recent years becoming one of interesting field on research. Nowadays, Navigation system is needed for controlling autopilot, guiding and tracking. On navigation system, before the vehicle start to navigate, navigation system takes time to align. The goal of Alignment is for determining the initial position, velocity, attitude and bias from sensor on navigation system. Navigation system can use Inertial Navigation System (INS) and Global Positioning Systems (GPS). In INS there are two subsystem that are sensor and processor. The Sensor is consisting of accelero and gyro. The Processor is used to manage signal from sensors. That is frame transformation from body frame to navigation frame as well to calculate integral from sensor signal to be position, velocity and attitude on North axis, East axis, and Down axis (NED). This Process is called mechazization. If only used mechanization process the value of RMSE is [0.4904°, 0.9215°, 19.3315°] for attitude, [1.7179, 4.8001, 0.2404]m/s for velocity and [0.0009°, 0.0012°, 5.9941m] for position. The RMSE from mechanization only is relatively large. It is caused by noise. so, for facing that issue on this final project is proposed INS and GPS integration with Kalman Filter for estimating noise error from INS in order to improve navigation solutions. On simulation using this proposed method, the RMSE value is [0.2326, 0.5388, 2.7953] for attitude, [0.0638, 0.0586, 0.1430] for velocity and [0.00002°, 0.000007°, 1.1485] for position.*

***Key Words:*** *INS, Navigation, GPS, Kalman filter, Mechanization, noise*

*[Halaman ini sengaja dikosongkan]*

## KATA PENGANTAR

Puji dan syukur atas kehadiran Allah SWT yang telah melimpahkan rahmat dan hidayah-Nya. Sehingga buku Tugas Akhir ini dapat diselesaikan oleh penulis dengan tepat waktu serta selawat dan salam yang selalu tercurah untuk baginda Rasulullah SAW yang telah menjadi teladan bagi penulis.

Buku Tugas Akhir ini disusun untuk melengkapi salah satu syarat memperoleh gelar Sarjana Teknik di departemen Teknik Elektro ITS. Buku yang berjudul “**Perancangan Sistem Navigasi Berbasis INS-GPS Menggunakan Filter Kalman Untuk Meningkatkan Presisi Hasil Navigasi**” dipersembahkan juga untuk kemajuan riset dan teknologi Indonesia khususnya untuk ITS, Fakultas Teknologi Elektro, Departemen Teknik Elektro dan bidang studi Teknik Sistem Pengaturan.

Penulis ingin mengucapkan terima kasih kepada Allah SWT, yang senantiasa melimpahkan rahmat, nikmat, karunia dan ridho-Nya pada penulis dalam menyelesaikan Tugas Akhir ini. Bapak Rusdianto Effendi AK MT., dan Bapak Mochammad Sahal ST., M.Sc selaku dosen pembimbing atas masukan, arahan dan ilmu yang disalurkan kepada penulis dalam menyelesaikan buku laporan ini. Kedua orang tua penulis Bapak Sahbudin dan Ibu Hariati yang selalu mengirimkan dukungan dan doa kepada penulis. Nur Amalia yang selalu mengingatkan dan menjadi teman berbagi keluh kesah serta menjadi pendorong untuk segera menyelesaikan Tugas Akhir ini. Ketiga adik penulis yaitu Fatifa Azzahra, Rafi Muhammad dan Firzan Muhammad yang telah mendoakan penulis. keluarga e55 yang telah menjadi teman menghilangkan kejenuhan serta membantu penulis selama empat tahun ini serta semua pihak yang telah membantu penulis yang tidak dapat disebutkan semua.

Penulis menyadari adanya kekurangan dalam pengerjaan Tugas Akhir ini sehingga penulis sangat menantikan kritik dan saran untuk memperbaiki kekurangan penulis. Akhir kata, semoga Tugas Akhir ini dapat bermanfaat bagi siapapun yang terlibat dan membacanya serta dapat menjadi acuan penelitian selanjutnya.

Surabaya, 25 Mei 2019

Penulis

*[Halaman ini sengaja dikosongkan]*



## DAFTAR ISI

HALAMAN JUDUL .....	i
HALAMAN JUDUL .....	iii
PERNYATAAN KEASLIAN TUGAS AKHIR .....	vii
LEMBAR PENGESAHAN .....	ix
ABSTRAK .....	xi
<i>ABSTRACT</i> .....	xiii
KATA PENGANTAR .....	xv
DAFTAR ISI .....	xvii
DAFTAR GAMBAR .....	xix
DAFTAR TABEL .....	xxiii
BAB 1 PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Perumusan Masalah .....	2
1.3 Batasan Masalah .....	2
1.4 Tujuan Penelitian .....	2
1.5 Metodologi .....	3
1.6 Sistematika Laporan Penelitian.....	4
BAB I Pendahuluan .....	4
BAB II Dasar Teori .....	4
BAB III Perancangan Sistem .....	4
BAB IV Hasil dan Analisa .....	5
BAB V Kesimpulan dan Saran .....	5
1.7 Relevansi .....	5
BAB 2 TINJAUAN PUSTAKA.....	7
2.1 Sistem Navigasi .....	7
2.1.1 <i>Dead Reckoning</i> .....	9
2.1.2 <i>Position Fixing</i> .....	10
2.1.3 Integrasi Sistem Navigasi .....	12
2.2 <i>Coordinate Frame</i> Pada Sistem Navigasi .....	13
2.2.1 <i>ECI (Earth-Center Inertial)</i> [3] .....	14
2.2.2 <i>ECEF (Earth-Centered Earth-Fixed Frame)</i> [3].....	16
2.2.3 <i>Local Navigation Frame</i> [3] .....	16
2.2.4 <i>Body Frame</i> [3].....	18
2.3 <i>Inertial Navigation System (INS)</i> .....	18

2.3.1	<i>Accelerometer</i> .....	20
2.3.2	<i>Gyroscope</i> .....	21
2.3.3	Eror pada IMU.....	22
2.4	<i>Global Positioning System (GPS)</i> .....	26
2.5	Model Eror INS [3].....	28
2.6	<i>Model Gravitasi</i> .....	30
2.7	Tahapan Pengolahan sinyal IMU.....	31
2.7.1	Proses <i>Alignment</i> .....	33
2.7.2	Proses Mekanisasi.....	33
2.7.3	Filter Kalman.....	38
BAB 3	PERANCANGAN SISTEM.....	45
3.1	Identifikasi Permasalahan.....	45
3.2	Model dan Parameter Sistem.....	46
3.3	Perancangan Pengolahan Sinyal Sensor.....	46
3.3.1	Inisialisasi Kondisi Awal.....	47
3.3.2	Perancangan Proses <i>Alignment</i> .....	47
3.3.3	Perancangan Proses Mekanisasi.....	51
3.3.4	Perancangan Filter Kalman.....	56
BAB 4	SIMULASI DAN ANALISA.....	59
4.1	Hasil Simulasi Proses <i>Alignment</i> .....	59
4.1.1	Hasil Simulasi <i>Coarse Alignment</i> .....	62
4.1.2	Hasil Simulasi <i>Fine Alignment</i> .....	64
4.2	Hasil Simulasi Saat Wahana Bergerak.....	72
4.2.1	Hasil Estimasi <i>Attitude</i> Saat Bergerak.....	72
4.2.2	Hasil Estimasi Kecepatan Saat Bergerak.....	75
4.2.3	Hasil Estimasi Posisi Saat Bergerak.....	76
4.2.4	Hasil Estimasi Bias Saat Kondisi Bergerak.....	79
4.2.5	Rekapitulasi RMSE Dari Kondisi Bergerak.....	80
BAB 5	KESIMPULAN DAN SARAN.....	83
5.1	Kesimpulan.....	83
5.2	Saran.....	84
DAFTAR	PUSTAKA.....	85
LAMPIRAN	.....	87
RIWAYAT	HIDUP PENULIS.....	97

## DAFTAR GAMBAR

Gambar 2.1	Penentuan posisi pada metode <i>dead reckoning</i> [1]. .....	8
Gambar 2.2	Skematik dari <i>inertial navigation system</i> (INS) ....	9
Gambar 2.3	Skematik penentuan posisi pada GNSS. ....	11
Gambar 2.4	Titik origin dan ketiga axis dari ECI [3].....	14
Gambar 2.5	Perbandingan <i>frame</i> ECI dan ECEF [4].....	15
Gambar 2.6	Koordinat dari <i>local navigation frame</i> [3] .....	17
Gambar 2.7	Koordinat dari <i>body frame</i> .....	18
Gambar 2.8	Skema <i>Inertial Measurement Unit</i> (IMU).....	19
Gambar 2.9	Pengaruh bias terhadap nilai sebenarnya yang diukur oleh sensor .....	23
Gambar 2.10	<i>Misalignment</i> sumbu sensor IMU terhadap <i>body frame</i> .....	24
Gambar 2.11	Bentuk sinyal yang merupakan <i>white noise</i> [1].	26
Gambar 2.12	Tahapan pengolahan sinyal IMU.....	32
Gambar 3.1	Skema proses mekanisasi pada <i>local navigation frame</i> .....	52
Gambar 4.1	Data pengukuran <i>accelero</i> saat mobil diam. ....	60
Gambar 4.2	Data pengukuran <i>gyro</i> saat mobil diam.....	60
Gambar 4.3	Data pengukuran <i>accelero</i> saat mobil dalam kondisi bergerak.....	61
Gambar 4.4	Data pengukuran <i>gyro</i> saat mobil dalam kondisi bergerak. ....	61
Gambar 4.5	Hasil estimasi <i>roll</i> dan <i>pitch</i> dari <i>coarse alignment</i> .....	62
Gambar 4.6	Hasil Estimasi <i>yaw</i> dari <i>coarse alignment</i> .....	63
Gambar 4.7	Error (estimasi-CNS) dari hasil Estimasi <i>Attitude coarse alignment</i> .....	63

Gambar 4.8	Hasil estimasi <i>roll</i> dan <i>pitch</i> dari <i>fine alignment</i> .	64
Gambar 4.9	Hasil estimasi <i>yaw</i> dari <i>fine alignment</i> .....	65
Gambar 4.10	Error (estimasi-CNS) estimasi <i>attitude</i> dari <i>fine alignment</i> .....	65
Gambar 4.11	Hasil estimasi kecepatan dari <i>fine alignment</i> .....	66
Gambar 4.12	Error hasil estimasi kecepatan pada <i>fine alignment</i> . .....	66
Gambar 4.13	Hasil estimasi posisi dari <i>fine alignment</i> .....	67
Gambar 4.14	Error hasil estimasi posisi dari <i>fine alignment</i> .....	67
Gambar 4.15	Hasil estimasi bias <i>accelero</i> dari Filter Kalman pada <i>fine alignment</i> .....	68
Gambar 4.16	Hasil estimasi bias <i>gyroscope</i> dari Filter Kalman pada <i>fine alignment</i> .....	68
Gambar 4.17	Hasil estimasi <i>Attitude (Roll-Pitch)</i> dari Filter Kalman ketika kondisi bergerak. ....	73
Gambar 4.18	Hasil estimasi <i>attitude</i> dari Filter Kalman pada kondisi bergerak.....	73
Gambar 4.19	Error dari estimasi <i>attitude</i> dari Filter Kalman saat bergerak .....	74
Gambar 4.20	Hasil estimasi kecepatan dari Filter Kalman ketika wahana bergerak. ....	75
Gambar 4.21	Error hasil estimasi kecepatan dari Filter Kalman saat bergerak .....	76
Gambar 4.22	Hasil estimasi posisi dari Filter Kalman saat bergerak .....	77
Gambar 4.23	Hasil estimasi posisi dengan menampilkan hasil mekanisme yang memiliki error relatif besar. ....	78
Gambar 4.24	Error estimasi posisi dari Filter Kalman saat kondisi bergerak.....	78

Gambar 4.25	Hasil estimasi bias pada <i>accelero</i> dari Filter Kalman.....	79
Gambar 4.26	Hasil estimasi bias <i>gyroscope</i> dari Filter Kalman... .....	80

*[Halaman ini sengaja dikosongkan]*

## DAFTAR TABEL

Tabel 3.1	Parameter sensor STIM300 .....	46
Tabel 3.2	Parameter WGS84 .....	53
Tabel 4.1	Parameter eror hasil <i>coarse alignment</i> .....	62
Tabel 4.2	Parameter eror hasil <i>fine alignment</i> .....	69
Tabel 4.3	Tabel perbandingan RMSE dan STD Filter Kalman dengan Mekanisasi .....	71
Tabel 4.4	Parameter eror dari estimasi hasil navigasi saat bergerak .....	74
Tabel 4.5	Tabel RMSE dan STD saat wahana bergerak .....	80

*[Halaman ini sengaja dikosongkan]*



# BAB 1

## PENDAHULUAN

Pada bab ini akan dijelaskan mengenai latar belakang pengambilan judul Tugas Akhir ini, perumusan masalah yang dihadapi, batasan masalah, tujuan penelitian dan metodologi yang digunakan pada Tugas Akhir ini serta sistematika penyusunan hingga relevansi Tugas Akhir ini.

### 1.1 Latar Belakang

Sistem navigasi menjadi bidang riset yang sangat menarik untuk dilakukan. Apalagi seiring berkembangnya teknologi pada bidang elektronik terutama setelah penemuan transistor sehingga pembuatan rangkaian terpadu semakin mudah dilakukan. Hal ini membuat instrumen yang ada pada sistem navigasi semakin hari semakin berukuran kecil, hingga hanya millimeter dalam bentuk *chip*. Semakin kecil dan murah sistem navigasi sehingga dapat digunakan pada wahana tanpa perlu menghabiskan banyak ruang pada wahana. Di sisi lain, Ukuran yang mengecil tidak membuat gangguan atau eror dari instrumen sistem navigasi menjadi berkurang. Hal ini masih menjadi hal yang menantang bagi para peneliti [1], [2].

Wahana baik di darat maupun di permukaan laut atau di udara dapat menggunakan INS atau GPS untuk keperluan kontrol *autopilot*, *guidance*, dan *tracking*. Namun bila hanya *Inertial Navigation System* (INS) yang digunakan, eror yang dihasilkan akan bertambah besar seiring bertambahnya waktu yang dikarenakan eror sistemik dan eror acak yang terakumulasi oleh proses integral dua kali pada *accelero* dan satu kali pada *gyro*. Oleh karena itu, dibutuhkan pengukuran dari sistem navigasi lain untuk membandingkan hasil navigasi INS sehingga didapatkan eror pada INS. Pada [9] diintegrasikan INS/GPS dengan menggunakan *Adaptive Fuzzy* sebagai estimator. Sedangkan, penelitian yang dilakukan pada [10] mengintegrasikan INS/GPS dengan *Neural Network* sebagai estimatornya. Sistem navigasi yang akan diintegrasikan dengan INS pada Tugas Akhir ini adalah *Global Positioning System* (GPS). Hasil navigasi dari GPS diselisihkan dengan hasil navigasi INS, kemudian hasil selisih tersebut digunakan untuk mengurangi bias atau eror sistemik dari INS. Namun eror dari integrasi INS-GPS tidak hilang begitu saja, masih terdapat eror acak yang dapat mengurangi presisi dari hasil navigasi

integrasi INS-GPS. Error acak tersebut merupakan error acak yang berasal dari INS dan GPS. Oleh karena itu, untuk mengatasi permasalahan error acak tersebut, pada Tugas Akhir ini akan dirancang sebuah filter yang dapat mengurangi error acak sehingga dapat meningkatkan presisi hasil navigasi. Presisi menyatakan seberapa dekat error iterasi dengan error iterasi yang lainnya. Sedangkan akurasi menyatakan seberapa dekat error estimasi dengan nilai sebenarnya. Dikarenakan kehandalan Filter Kalman dalam memfilter sinyal yang acak, dalam Tugas Akhir ini akan menggunakan Filter Kalman untuk mengestimasi error acak dari sistem navigasi berbasis INS-GPS.

## 1.2 Perumusan Masalah

Pada Tugas Akhir ini masalah yang akan dibahas yaitu:

1. Bagaimana cara mengurangi error dari keluaran INS yang meningkat hingga sangat besar seiring bertambahnya waktu sehingga menyebabkan menurunnya presisi dari keluaran INS.
2. Bagaimana mengestimasi error acak pada sinyal sensor *accelero* dan *gyro* yang menjadi satu dengan error integrasi INS-GPS di mana, hal tersebut dapat mengurangi presisi hasil navigasi.

## 1.3 Batasan Masalah

Dalam Tugas Akhir ini ada beberapa batasan yang perlu diperhatikan antara lain sebagai berikut:

1. Tugas Akhir ini menggunakan data yang diukur pada wahana darat yaitu menggunakan mobil. Pada mobil tersebut dipasang sensor navigasi untuk melakukan pengukuran.
2. Pengambilan data dilakukan di kota Surabaya, sehingga lintang dan bujur pada data yang digunakan pada Tugas Akhir ini hanya terbatas pada wilayah kota Surabaya.
3. Pada Tugas Akhir ini sinyal satelit dari GPS diasumsikan selalu tersedia yaitu berjumlah minimum empat buah.

## 1.4 Tujuan Penelitian

Tujuan dari Tugas Akhir ini adalah sebagai berikut:

1. Merancang Sistem Navigasi yang meminimalisir error yang meningkat seiring bertambahnya waktu pada INS.

2. Merancang Filter Kalman untuk mengestimasi eror acak yang ada pada sinyal pengukuran *accelero* dan *gyro* yang menjadi satu dengan eror integrasi INS-GPS.

## 1.5 Metodologi

Pada Tugas Akhir ini digunakan suatu tahapan sistematis sehingga tujuan dari Tugas Akhir ini dapat tercapai. Tahapan tersebut urutan yang diterapkan pada Tugas Akhir ini. Berikut urutan tersebut:

1. Studi literatur

Pada tahap awal penelitian ini, dilakukan studi literatur untuk mempelajari dan mengkaji teori-teori serta mengumpulkan data penelitian yang relevan dan terpercaya untuk keabsahan Tugas Akhir ini. Literatur yang digunakan dalam penelitian ini memiliki batasan yaitu literatur yang digunakan harus bersumber dari paper, jurnal, buku, maupun artikel yang berasal dari badan pemerintahan atau institusi akademik terpercaya.

2. Pemodelan dan perancangan sistem navigasi

Setelah melakukan studi literatur pada tahap awal dan telah mendapatkan dasar teori mengenai INS, GPS, Filter Kalman, dan proses-proses yang digunakan pada sistem navigasi maka setelah itu dilakukan perancangan sistem navigasi. Dalam sistem navigasi terdapat beberapa proses atau tahapan, di mana pada setiap tahapan akan dilakukan pengolahan sinyal dari sensor. Pada proses tertentu membutuhkan model sistem dari INS. Model yang digunakan adalah model eror dari INS, di mana model ini akan digunakan pada Filter Kalman.

3. Perancangan filter Kalman

Pada tahap ini setelah mendapatkan model eror dari INS, maka model eror hasil pemodelan dari eror INS dapat digunakan pada model sistem untuk filter Kalman. Filter Kalman merupakan filter yang membutuhkan model dari sistem.

4. Pengujian sistem navigasi

Setelah perancangan sistem navigasi serta memodelkan eror INS, langkah selanjutnya yaitu pengujian hasil rancangan tersebut. Pengujian dilakukan dengan mengambil data hasil pengukuran pada wahana darat. Keakuratan dari sistem diuji

dengan membandingkan dengan alat ukur navigasi lain, dalam Tugas Akhir ini yaitu KVH-CNS5000 yang diasumsikan ideal.

5. Analisa hasil dan evaluasi  
Melakukan analisa terhadap hasil yang didapatkan dari sistem navigasi yang dirancangan pada Tugas Akhir ini. Sehingga didapatkan hasil navigasi yang diinginkan dengan eror yang diizinkan dari sistem navigasi yang telah dirancang dan disimulasikan. Analisa dilakukan khususnya pada eror INS yang mengalami berbagai kondisi gangguan apakah sudah memenuhi kriteria atau toleransi yang diizinkan. Lalu dilakukan evaluasi serta revisi desain sistem navigasi apabila diperlukan.
6. Penyusunan Laporan  
Penyusunan laporan akan dilakukan bersamaan dengan tahap-tahap lainnya. Isinya berkaitan dengan Tugas Akhir yang dikerjakan, meliputi pendahuluan, studi literatur, perancangan model sistem dan gangguan, perancangan filter Kalman pada sistem navigasi, simulasi sistem, pengujian dan analisa serta kesimpulan.

## **1.6 Sistematika Laporan Penelitian**

Penulisan laporan Tugas Akhir ini menggunakan sistematika sebagai berikut:

### **BAB I Pendahuluan**

Pada bab ini, akan diuraikan mengenai latar belakang pengambilan judul Tugas Akhir, masalah yang akan dibahas pada Tugas Akhir, tujuan dari Tugas Akhir dan sistematika penulisan Tugas Akhir, serta relevansi dari Tugas Akhir ini.

### **BAB II Dasar Teori**

Sesuai judul bab 2 yaitu Dasar Teori, bab ini berisikan dasar teori dan landasan dari Tugas Akhir ini. Semua teori pada Tugas Akhir ini mengacu pada bab 2. Dasar teori dikumpulkan dari melakukan studi literatur.

### **BAB III Perancangan Sistem**

Pada bab ini, dilakukan perancangan sistem navigasi. bab 3 menguraikan langkah – langkah perancangan sistem navigasi, yaitu

perancangan tahap *alignment*, transformasi *frame*, model integrasi INS-GPS dan perancangan filter Kalman.

#### **BAB IV Hasil dan Analisa**

Pada bab ini ditampilkan hasil dari pengujian sistem navigasi yang dirancang pada bab 3. Data pengujian merupakan data yang diambil menggunakan sensor IMU STIM300 yang diletakan pada wahana darat dan hasil dari pengujian sistem tersebut dibandingkan dengan KVH-CNS5000 yang hasilnya dianggap data ideal kemudian dianalisa.

#### **BAB V Kesimpulan dan Saran**

Setelah melakukan pengujian dan analisa pada bab 4, didapatkan kesimpulan yang akan dijabarkan pada bab 5 yaitu kesimpulan dan saran serta pemberian saran untuk penelitian yang akan menggunakan topik serupa dengan Tugas Akhir ini.

### **1.7 Relevansi**

Pada Tugas Akhir ini diharapkan dapat dimanfaatkan anatara lain sebagai berikut:

1. Dapat digunakan untuk sistem *autonomous*, seperti *Unmanned Surface Vehicle (USV)* atau *Unmanned Aerial Vehicle (UAV)*.
2. Tugas akhir ini dapat dijadikan acuan untuk membuat *prototype* sistem navigasi yang dapat diterapkan pada berbagai wahana baik di permukaan darat atau laut serta di udara.

*[Halaman ini sengaja dikosongkan]*

## BAB 2 TINJAUAN PUSTAKA

Pada bab ini berisi dasar teori yang akan digunakan pada Tugas Akhir ini. Semua dasar teori pada bab ini merupakan hasil studi literatur yang telah dilakukan pada tahap awal dari Tugas Akhir ini. Hasil dari studi literatur ini kemudian akan digunakan sebagai dasar teori dalam tahap perancangan sistem.

### 2.1 Sistem Navigasi

Sistem navigasi awalnya sangat populer pada bidang pelayaran, hal ini dapat dilihat pada kata navigasi yang berasal dari kata “*navigare*” yang berarti berlayar. Kata *navigare* terdiri dari dua kata yaitu “*navis*” yang berarti kapal dan “*agere*” yang berarti menjalankan atau berlayar. Sehingga sistem navigasi pada awalnya dipergunakan untuk belayar dengan kapal [2]. Pada sistem navigasi sederhana terdapat dua hal penting yaitu prosesor dan sensor. Prosesor pada sistem navigasi berguna untuk mengolah data yang telah diukur oleh sensor termaksud transformasi *frame* pada sistem navigasi. prosesor akan menghasilkan keluaran navigasi atau disebut hasil navigasi. Sensor navigasi merupakan suatu devais yang mengukur besaran-besaran pada navigasi yang kemudian akan diolah oleh prosesor [1].

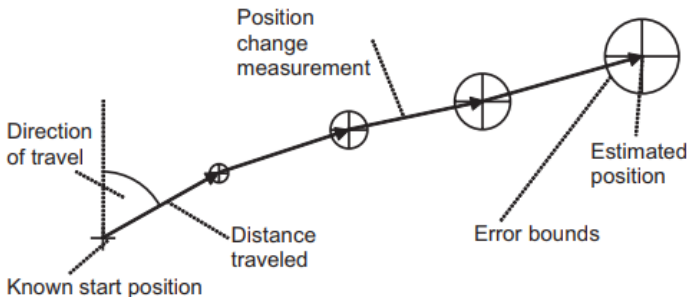
Sensor navigasi terdiri dari sensor yang tidak memerlukan bantuan sumber informasi eksternal dan sensor yang membutuhkan informasi eksternal. Jenis sensor yang pertama yaitu sensor yang secara mandiri tidak membutuhkan sumber informasi eksternal. Contoh dari sensor yang mandiri tersebut adalah sensor inersia yaitu *accelerometers*, *gyroscopes*, *odometry* dan lain lain. Jenis sensor kedua yaitu sensor yang membutuhkan sumber informasi eskternal. sumber eksternal tersebut dapat berupa sinyal radio atau keadaan lingkungan sistem navigasi. Contohnya adalah *Global Navigation Satellite* (GNSS) yang membutuhkan sinyal informasi yang dikirimkan oleh satelit. Pada Tugas Akhir ini akan digunakan kedua jenis sensor navigasi tersebut. Tujuannya yaitu meningkatkan presisi hasil navigasi dengan menggabungkan keuntungan dari masing-masing jenis sensor. Pada sensor inersia akan digunakan *Inertial Measurement Unit* (IMU) dan *Global Positioning*

System (GPS). Keduanya akan diintegrasikan agar dapat menghasilkan hasil navigasi yang lebih akurat [1].

Seiring berkembangnya zaman serta diiringi dengan perkembangan dibidang teknologi komputer menghasilkan sistem autonomus yang mengharuskan penggunaan sistem navigasi untuk keperluan *tracking* dan *control*. Penggunaan Sistem navigasi merambat pada hampir setiap bidang mulai dari transportasi dan pengaplikasiannya antara lain servis berdasarkan lokasi, *search and rescue*, perjalanan darat dan udara, manajemen armada kapal, *intelligence gathering*, olahraga, keselamatan umum, penilaian dan perencanaan lingkungan hingga penegakan hukum [1].

Kegunaan dari sistem navigasi makin dipermudah dengan harga dari peralatan navigasi yang semakin murah. Hal tersebut dikarenakan pesatnya perkembangan dalam bidang *microelectronics* yang dapat memperkecil rangkaian terpadu. Semakin mudah dalam membuat rangkaian terpadu maka hal tersebut dapat menurunkan biaya produksi dari peralatan sistem navigasi seperti sensor inersia, GPS *receivers*, dan komputer yang dapat melakukan perhitungan yang cepat.

Pada sistem navigasi terdapat dua metode dalam penentuan hasil navigasi yang pertama yaitu *dead reckoning* (DR). Metode DR menggunakan data pengukuran sebelumnya untuk mendapatkan hasil navigasi yaitu posisi, kecepatan dan *attitude*. Kedua adalah *Position Fixing* yang memerlukan referensi data dari obyek lain oleh sebab itu metode ini sering juga disebut *Reference-based Systems* [1].



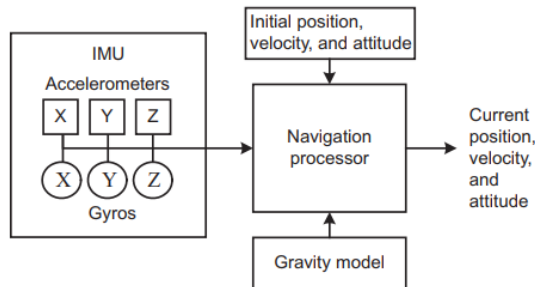
**Gambar 2.1** Penentuan posisi pada metode *dead reckoning* [1].



### 2.1.1 *Dead Reckoning*

*Dead reckoning* adalah salah satu metode menentukan posisi dengan mengukur perubahan posisi atau kecepatan kemudian dijumlahkan dengan nilai posisi sebelumnya sehingga didapatkan posisi saat itu. Pada metode *Dead reckoning* posisi atau kecepatan diukur pada *body frame* sehingga diperlukan devais untuk mengukur arah *heading* terhadap lingkungan yang kemudian disebut *yaw*. Pada kasus navigasi dua dimensi dibutuhkan cukup arah *yaw*, namun untuk kasus navigasi tiga dimensi dibutuhkan pengukuran sikap sudut wahana pada ketiga aksis yaitu *roll*, *pitch* dan *yaw*. Pada pengukuran perubahan posisi atau perubahan sudut, semakin kecil rentang waktu pengukuran dengan pengukran sebelumnya maka semakin akurat hasil navigasi yang dihasilkan. namun semakin kecil rentang waktu tersebut akan membutuhkan perhitungan yang semakin cepat pula pada sensor. Masalah tersebut dapat diselesaikan dengan menggunakan komputer terpisah dari sensor atau disebut prosesor untuk melakukan perhitungan.

Pengukuran sudut *yaw* dapat dilakukan dengan menggunakan kompas magnetik. Mungkin metode pengukuran tersebut telah kuno, namun telah tersedia kompas magnetik dengan pengukuran elektronik. Dalam bidang pelayaran penentuan sudut *yaw* dapat dilakukan dengan menggunakan *gyrocompass*. Penentuan sudut *yaw* dapat dilakukan dengan menggunakan jalur lintasan pada wahana darat, namun metode tersebut dapat dilakukan setelah wahana bergerak terlebih dahulu.



**Gambar 2.2** Skematik dari *inertial navigation system* (INS).

Salah satu sistem navigasi yang menggunakan metode *dead reckoning* adalah *inertial navigation system* (INS). INS merupakan sistem

navigasi tiga dimensi yang terdiri dari prosesor dan sensor-sensor inersia yang dikenal dengan sebutan *inertial measurement unit* (IMU) seperti yang ditampilkan pada Gambar 2.2. setiap sensor pada IMU terdiri dari tiga *accelero* yang saling tegak lurus satu sama lain dan tiga *gyroscope* yang diselaraskan dengan *accelero*. Fungsi dari prosesor adalah melakukan pengolahan sinyal dari sensor yaitu mengintegrasikan hasil dari sensor-sensor pada IMU dan melakukan transformasi *frame*, untuk mendapatkan hasil navigasi yaitu posisi, kecepatan dan sikap sudut wahana [3]. Proses tersebut disebut mekanisasi dan akan dijelaskan pada bagian tahapan pengolahan sinyal. Prozessor juga dapat melakukan proses pemfilteran sinyal.

### 2.1.2 *Position Fixing*

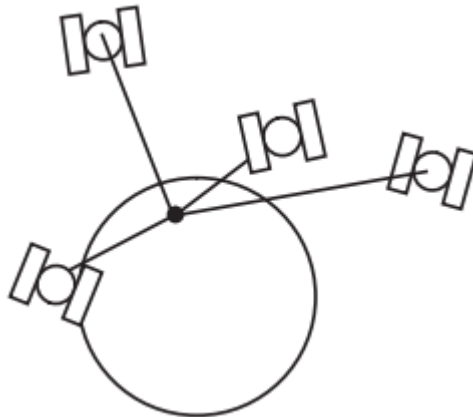
Pada metode *position fixing* sendiri terdiri dari beberapa metode antara lain yang sering digunakan dalam keperluan sistem navigasi pada sebuah wahana ialah *signal based positioning*. Metode tersebut menggunakan sinyal yang dipancarkan oleh *transmitter*. Sinyal tersebut dapat berupa sinyal radio yang mengandung informasi posisi dari *transmitter*. Radio pertama kali digunakan untuk keperluan navigasi pada 1920an, ketika gelombang radio 75-MHz digunakan untuk mencari arah. Contoh dari *signal based positioning* yang akan digunakan pada Tugas Akhir ini adalah GPS.

Dalam perkembangannya, sinyal radio terbagi atas dua kategori utama, yang pertama adalah penerus dari sistem navigasi dengan gelombang radio yang dikembangkan pada 1940an dan 1950an hingga dikembangkan sampai 1980an. Kategori ini digunakan untuk navigasi pada pesawat terbang. Pada beberapa negara kategori ini dikembangkan untuk jarak jauh yang digunakan dalam navigasi pelayaran. sistem ini biasa disebut Loran (*Long-range navigation*) *system* yang sinyal dari *transmiternya* dapat mencapai radius 3000 km.

Kategori kedua dikembangkan pada tahun 1990an dan 2000an yang awalnya bertujuan untuk memanfaatkan sinyal radio yang digunakan untuk keperluan komunikasi dan penyiaran agar dapat digunakan untuk *positioning*. Beberapa sinyal tersebut antara lain sinyal telepon genggam, Wi-Fi, Bluetooth Zigbee, RFID, komunikasi *ultrawideband* (UWB), sinyal televisi dan radio yang dipancarkan oleh stasiun televisi dan radio. Biasanya radius dari teknologi ini termasuk dalam *short-range* yang berkisar puluhan meter. Namun untuk sinyal

yang dipancarkan oleh stasiun televisi dan radio dapat mencapai radius 100 km dan sinyal dari telepon genggam dapat mencapai radius 35 km.

Dalam bidang militer, agar dapat melacak setiap armada kapal dan pesawat terbang serta peluru kendali jarak jauh maka dibutuhkan sistem navigasi yang dapat mencakup area secara global. Pada tahun 1978 Amerika Serikat meluncurkan satelit prototipe pertama yang akan digunakan untuk *Global Positioning System* (GPS). Setelah peluncuran pertama tersebut maka GPS dikatakan dapat beroperasi secara penuh pada 1993. GPS merupakan salah satu contoh dari GNSS di mana untuk setiap GNSS terdiri dari 24 satelit atau lebih. Hal tersebut bertujuan untuk menjamin ketersediaan sinyal satelit dengan banyak minimum sinyal satelit yang tersedia adalah empat satelit untuk posisi di manapun secara global. Dengan empat sinyal satelit yang tersedia, pengguna dapat mendapatkan posisi tiga dimensi seperti yang diilustrasikan pada



**Gambar 2.3** Skematik penentuan posisi pada GNSS.

Metode *position fixing* sangat bergantung dari sumber informasi eksternal. Pada *signal based positioning* sumber informasi tersebut ialah sinyal radio dari *transmitter*. Sinyal tersebut sangat rentan oleh gangguan dan pelemahan yang disebabkan hambatan seperti gedung-gedung, dedaunan pada pepohonan yang lebat dan gunung.

### 2.1.3 Integrasi Sistem Navigasi

Pada sistem navigasi terdapat dua metode yang sangat umum digunakan yaitu *dead reckoning* dan *position fixing* yang keduanya telah dijelaskan sebelumnya. Kedua metode tersebut dikombinasikan agar mendapatkan keuntungan dari kedua metode atau saling mengatasi kekurangan dari masing-masing metode. Dalam Tugas Akhir ini akan dikombinasikan metode *dead reckoning* dan *position fixing*. Pada metode *dead reckoning* menggunakan INS dan untuk *position fixing* akan menggunakan salah satu GNSS yaitu GPS. Pada INS, eror akan mengakibatkan hasil mekanisasi tidak akurat. GPS dibutuhkan untuk membatasi hasil navigasi yang menyimpang yang disebabkan oleh eror yang terakumulasi pada proses mekanisasi seiring bertambahnya waktu. Oleh karena itu diintegrasikan INS dan GPS agar didapatkan hasil navigasi yang akurat. Integrasi INS-GPS terdiri dari tiga jenis integrasi yaitu *loosely coupled*, *tightly coupled* dan *ultra-tightly (deeply) coupled*. Ketiga integrasi tersebut akan dibahas secara bergantian.

#### 2.1.3.1 Integrasi INS-GPS Loosely Coupled

Integrasi jenis ini membiarkan INS dan GPS memiliki hasil yang independen, di mana hasil navigasi dari INS dan GPS terpisah. Hasil navigasi dari GPS akan dikurangkan dengan hasil navigasi dari INS sehingga mengabaikan eror pada hasil pengukuran INS. Eror tersebut kemudian diestimasi dengan metode estimasi tertentu. Dalam Tuga Akhir ini metode estimasi yang digunakan adalah Kalman filter. Kelebihan dari integrasi jenis ini adalah mudah diimplementasikan dan memiliki sifat yang *robust*. Pada integrasi ini tidak dilakukan koreksi dari hasil GPS. Namun sinyal yang diterima oleh *receiver* pada pengguna GPS sangat rentan terhadap gangguan seperti degradasi sinyal dari satelit yang disebabkan sinyal dari satelit melewati atmosfer atau ketersediaan sinyal dari satelit yang kurang yang disebabkan sinyal dari satelit terhalang oleh dedaunan yang lebat pada daerah yang masih hijau atau gedung-gedung pada daerah perkotaan. Masalah tersebut dapat mengurangi presisi dari hasil navigasi integrasi *loosely coupled* [3].

#### 2.1.3.2 Integrasi INS-GPS Tightly Coupled

Pada integrasi ini membutuhkan data mentah dari GPS seperti *pseudo-range* dan *pseudo-range rate* dari satelit serta ofset *receiver clock* dan bias dari penerima atau pengguna GPS. Data mentah tersebut akan

digunakan pada estimasi eror dari INS. Hasil estimasi eror dari INS akan digunakan untuk mengkoreksi hasil navigasi dari INS. Integrasi ini dapat mengatasi masalah pada integrasi *loosely coupled* yaitu adanya pengukuran yang berkorelasi dan mampu mengatasi masalah ketersediaan sinyal dari satelit bahkan bila sinyal satelit yang tersedia kurang dari empat satelit. Hal ini sangat membantu mengatasi masalah yang terjadi ketika sinyal dari satelit terhalang oleh hambatan yang berupa dedaunan yang lebat pada daerah hutan atau gedung-gedung pada daerah perkotaan. Di sisi lain, integrasi jenis ini lebih rumit diimplementasikan dibandingkan dengan *loosely coupled* dan tidak semua GPS menyediakan keluaran berupa data mentah GPS [3].

### **2.1.3.3 Integrasi INS-GPS Ultra-Tight Coupled**

Integrasi jenis ini disebut juga *deep integration*, di mana terdapat dua hal yang sangat berbeda dari dua jenis integrasi sebelumnya. Pertama pada penerima sinyal GPS terdapat perubahan skema sehingga dapat digunakan dengan implementasi yang berbeda untuk *tracking loop* agar mendapatkan estimasi posisi. Kedua dibutuhkan informasi dari INS untuk digunakan sebagai bagian integral pada perhitungan yang terjadi di *receiver* GPS, hal ini menyebabkan INS dan GPS bukan lagi sistem navigasi yang independen satu sama lain. Tipe integrasi jenis ini juga menambah kompleksitas pada implementasinya, karena diharuskan untuk mengubah internal perangkat keras GPS *receiver*. Kelebihan dari sistem terintegrasi ini adalah tersedianya sebuah data GPS yang bebas dari gangguan, terutama *jamming*, dan bekerja pada SNR (*Signal to Noise Ratio*) yang rendah serta mampu menghasilkan solusi navigasi meskipun jumlah satelit yang tertangkap/ terbaca oleh receiver kurang dari empat [3].

## **2.2 Coordinate Frame Pada Sistem Navigasi**

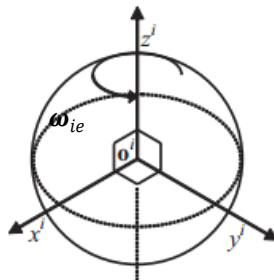
Dalam sistem navigasi ada hal yang perlu dipahami yaitu *Coordinate frame*. Sistem navigasi memberikan hasil berupa posisi, kecepatan dan sikap sudut (*attitude*) oleh sebuah objek. Namun hasil tersebut kurang berarti bila tanpa menggunakan sebuah acuan terhadap lingkungan dari objek. Oleh sebab itu diperlukan definisi *coordinate frame* acuan (*reference frame*). *Reference* digunakan untuk menggambarkan posisi, kecepatan dan sikap sudut suatu objek pada lingkungan tertentu, misalnya posisi pesawat pada *coordinate frame* bumi. Pada subbab ini akan

diuraikan beberapa *coordinate frame* (kerangka koordinat) yang sering digunakan pada sistem navigasi terutama yang akan digunakan dalam Tugas Akhir ini.

### 2.2.1 ECI (*Earth-Center Inertial*) [3]

Sensor *accelero* dan *gyro* mengukur gaya spesifik dan kecepatan sudut dari bodi IMU terhadap *frame* inersia. *Frame* inersia merupakan *frame* yang tidak berakselerasi atau berputar terhadap alam semesta. *Frame* ECI disimbolkan oleh  $i$ , yang merupakan singkatan dari *inertial*. *Frame* ECI bukan merupakan *frame* inersia karena mengalami percepatan atau akselerasi pada orbitnya yang mengelilingi matahari. Selain itu sumbu putar dari ECI (sumbu- $z$ ) berputar secara perlahan. Namun, efek dari fenomena tersebut relatif kecil dari pada eror yang terdapat pada sensor IMU.

Titik origin dari *frame* ECI berada pada pusat massa bumi. Sumbu- $z$  *frame* ECI selalu mengarah dari pusat massa bumi ke kutub utara (bukan kutub magnet). Sumbu- $z$  merupakan sumbu putar dari bumi. Ada beberapa cara dalam menentukan Sumbu- $x$  dan sumbu- $y$  dari *frame* ECI namun yang digunakan pada Tugas Akhir ini yaitu, Sumbu- $x$  dan sumbu- $y$  berada pada bidang ekuatorial bumi, di mana sumbu- $x$  berada pada *vernal equinox* atau memiliki arah dari pusat massa bumi mengarah ke matahari. Sedangkan sumbu- $y$  melengkapi kaidah tangan kanan yang berada pada  $90^\circ$  searah perputaran bumi atau tegak lurus dengan arah dari pusat massa bumi ke matahari. Gambar 2.4 memperlihatkan titik origin dan sumbu dari *frame* ECI. Versi *frame* ECI ini sering disebut *celestial coordinate*.

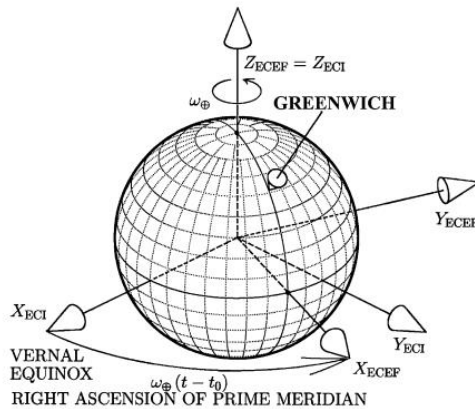


**Gambar 2.4** Titik origin dan ketiga axis dari ECI [3].

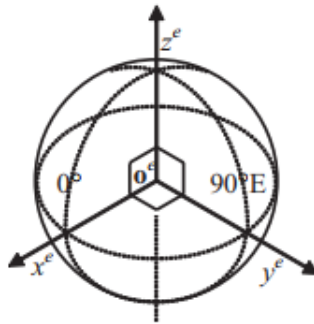
Pada penggunaan *frame* ECI terdapat beberapa yang perlu diperhatikan antara lain sebagai berikut:

1. Menentukan pusat massa dari bumi terhadap suatu titik yang diketahui pada permukaan bumi. Di mana titik origin dari *frame* ECI berada pada pusat dari bentuk *ellipsoidal* yang mewakili bentuk permukaan dari bumi. Pusat dari bentuk ini dekat dengan pusat massa bumi sebenarnya.
2. Sumbu putar dari bumi yaitu sumbu-z sebenarnya berputar dengan mengikuti jalur yang berbentuk lingkaran dengan jari-jari sekitar 15m. salah satu solusinya yaitu menggunakan IERS Reference Pole (IRP) atau *Conventional Terrestrial Pole* (CTP). Di mana posisi dari sumbu-z diamati pada 1900 dan 1905 dan kemudian menggunakan rata-rata dari hasil pengamatan tersebut.

*Frame* ECI dalam Tugas Akhir digunakan sebagai representasi dari *frame* inersia yang merupakan *frame* acuan dari pengukuran yang dilakukan oleh sensor inersia yaitu dalam Tugas Akhir *accelero* dan *gyroscope*. Kedua sensor tersebut mengukur percepatan  $f_{ib}^b$  dan kecepatan sudut  $\omega_{ib}^b$  dari *body* dengan mengacu pada *frame* inersia atau dalam Tugas Akhir ini *frame* ECI.



**Gambar 2.5** Perbandingan *frame* ECI dan ECEF [4].



**Gambar 2.6** Titik origin dari ECEF dan ketiga sumbunya.

### 2.2.2 ECEF (*Earth-Centered Earth-Fixed Frame*) [3]

ECEF atau biasa juga disingkat *Earth frame* memiliki kesamaan dengan ECI, hanya saja sumbu- $x$  dan sumbu- $y$  pada ECEF ikut berotasi dengan bumi Gambar 2.5 memperlihatkan perbandingan keduanya. ECEF memiliki titik origin yang berada pada pusat massa bumi. Sumbu- $x$  berada pada perpotongan garis katulistiwa dengan *conventional zero meridian* (CZM) di mana *longitude* memiliki nilai  $0^\circ$ . Sedangkan untuk sumbu- $y$  pada ECEF berada  $90^\circ$  dari sumbu- $x$  yang searah dengan rotasi bumi atau sumbu yang mengarah dari pusat massa bumi ke perpotongan garis katulistiwa dengan  $90^\circ$  *east meridian*. Simbol yang digunakan pada ECEF yaitu  $e$  yang merupakan singkatan dari *earth*. Ketiga sumbunya yaitu  $x^e$ ,  $y^e$  dan  $z^e$ .

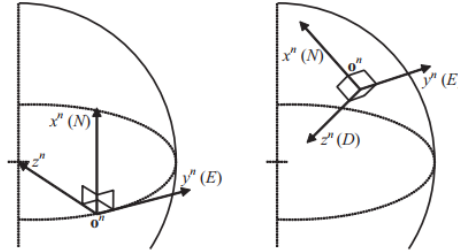
*Earth-Centered Earth-Fixed Frame* pada Tugas Akhir ini digunakan untuk menyatakan kecepatan rotasi bumi  $\omega_{ie}$  di mana  $\omega_{ie}$  bernilai sangat kecil yaitu  $7.292115 \times 10^{-5}$  *rad/s*.  $\omega_{ie}$  menyatakan kecepatan rotasi bumi (kecepatan rotasi  $e$ ) dengan mengacu pada *frame* inersia atau *frame* ECI (mengacu pada *frame*  $i$ ).

### 2.2.3 *Local Navigation Frame* [3]

*Frame* ini sering disebut juga dengan beberapa nama yaitu, *geodetic*, *geographic*, dan *topocentric frame*. Simbol dari *local navigation frame* adalah  $n$  (digunakan pada Tugas Akhir ini). Namun, ada



beberapa penulis yang menggunakan simbol  $g$  atau  $l$ . Titik origin dari *frame* ini berada pada objek (wahana) atau berada pada pusat massa dari objek suatu sistem navigasi. Titik origin dari *local navigation frame* dapat pula berada pada pengguna seperti pada pengguna *smartphone*. Gambar 2.6 menampilkan titik origin dan sumbu dari *local navigation frame*.



**Gambar 2.6** Koordinat dari *local navigation frame* [3]

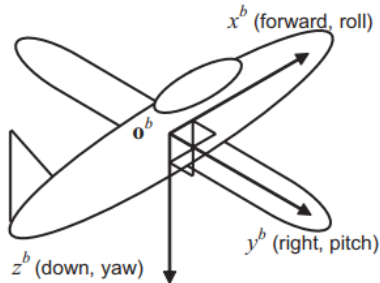
Sumbu dari *local navigation frame* sejajar dengan arah mata angin yaitu: utara (*north*), timur (*east*) dan vertikal. Dalam Tugas Akhir ini, kesepakatan yang digunakan untuk sumbu dari *local navigation frame* adalah *down* untuk sumbu- $z$  di mana sumbu ini mengarah dari wahana ke permukaan bumi yang direpresentasikan *ellipsoidal*. Model gravitasi searah dengan sumbu- $z$  pada *local navigation*. Sumbu- $x$  atau sumbu *north* pada *local navigation frame* merupakan proyeksi dari permukaan *ellipsoidal* bumi yang tegak lurus terhadap sumbu- $z$  dan mengarah dari objek atau wahana ke arah kutub utara. Sedangkan untuk sumbu- $y$  dari *local navigation frame* yaitu berasal dari objek ke arah timur. Ketiga sumbu *local navigation frame* dengan konvensi ini disingkat NED (*north* sebagai sumbu- $x$ , *east* sebagai sumbu- $y$ , *down* sebagai sumbu- $z$ ).

Konvensi NED merupakan konvensi yang sering digunakan dan akan digunakan pada Tugas Akhir ini. Namun, ada beberapa konvensi yang digunakan yaitu seperti ENU (*east* sebagai sumbu- $y$ , *north* sebagai sumbu- $x$ , *up* sebagai sumbu- $z$ ) [1]. Perlu diketahui bahwa *heading* dari NED adalah sumbu *north* atau sumbu  $x$  pada *body frame* sedangkan pada ENU adalah sumbu *east* atau sumbu  $y$  pada *body frame*.

Pada Tugas Akhir ini hasil navigasi dari wahana akan menggunakan *resolving axis* dari *local navigation frame*. *Resolving axis* merupakan sumbu yang digunakan untuk menyatakan nilai hasil navigasi dari suatu objek.

### 2.2.4 Body Frame [3]

Sumbu yang digunakan pada IMU adalah sumbu pada *body frame*. Walaupun mengukur gerakan mengacu terhadap *frame* inersia. Oleh karena itu *body frame* penting dalam navigasi. contoh besaran yang di representasikan pada sumbu *body frame* ialah  $\omega_{ib}^b$  dan  $f_{ib}^b$  yang keduanya merupakan kecepatan sudut dan gaya spesifik.



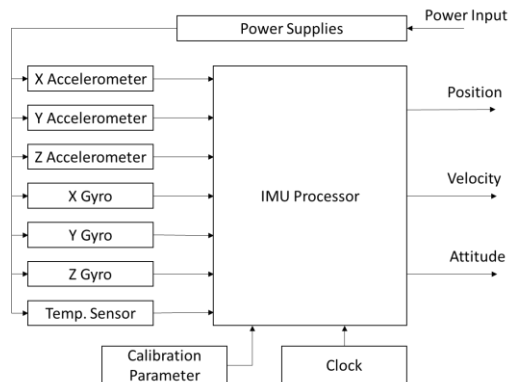
**Gambar 2.7** Koordinat dari *body frame*

Titik origin dari *body frame* adalah titik berat dari wahana atau objek. Titik origin dari *body frame* biasanya sama dengan *local navigation frame*. Sedangkan untuk sumbu-x memiliki arah dari titik berat wahana ke arah di mana wahana bergerak (*heading*) dan sumbu-y  $90^\circ$  searah putaran jam dari sumbu-x. sedangkan sumbu-z vertikal ke atas dan tegak lurus dari bidang x-y. ketiga sumbu tadi merupakan konvensi yang digunakan pada Tugas Akhir ini untuk *body frame*. Perbedaan *body frame* dan *Local Navigation frame* adalah pada *body frame* ketiga sumbu teletak tetap pada wahana sedangkan pada *local navigation frame* tetap pada arah mata angin *north* dan *east*. Gambar 2.7 menunjukkan letak titik origin dan ketiga sumbu dari *body frame*.

### 2.3 Inertial Navigation System (INS)

Sistem navigasi *Inertial Navigation System* atau disingkat INS merupakan sebuah sistem *autonomus*, di mana INS tidak membutuhkan sumber informasi eksternal untuk menghasilkan hasil navigasi seperti posisi, kecepatan, dan sikap sudut berdasarkan pengukuran yang dilakukan INS. Dalam implementasinya, INS menggunakan metode *Dead Reckoning* (DR) untuk mendapatkan hasil navigasi yang diinginkan. Seperti telah dijelaskan sebelumnya metode DR tidak membutuhkan

sumber informasi eksternal, melainkan membutuhkan hasil navigasi sebelumnya yang kemudian digunakan bersama hasil pengukuran sekarang sehingga menghasilkan hasil navigasi sekarang. Menurut literatur terdapat dua arsitektur yang digunakan pada sensor INS yaitu arsitektur bergimbal dan arsitektur *strapdown*. Pada sistem yang menggunakan gimbal IMU dipasang pada satu set gimbal yang terdiri dari gimbal luar dan gimbal dalam. Set gimbal tersebut diselaraskan dengan aksis INS.



**Gambar 2.8** Skema *Inertial Measurement Unit* (IMU)

INS terdiri dari sensor inersia yang sering disebut *inertial measurement unit* (IMU) dan prosesor. Sensor yang digunakan dalam IMU adalah *accelerometer* dan *gyroscope*. *Accelerometer* mengukur gaya spesifik dan *gyroscope* mengukur kecepatan sudut, keduanya tanpa membutuhkan sumber informasi dari luar. Setiap *accelerometer* mengukur gaya spesifik yang bekerja pada setiap aksis sensitif, sama halnya dengan *gyroscope* yang mengukur kecepatan sudut pada sebuah aksis sensitif. Dalam setiap IMU terdapat tiga buah *accelerometer* yang saling tegak lurus satu sama lain. Pada setiap aksis sensitif dari trio *accelerometer* tersebut disandingkan dengan sebuah *gyroscope*, sehingga dalam setiap IMU terdapat tiga *accelerometer* dan *gyroscope* yang saling tegak lurus satu sama lain untuk setiap aksis sensitif. Desain seperti itu menghasilkan hasil navigasi dalam tiga dimensi yang independen pada setiap aksis.

### 2.3.1 *Accelerometer*

*Accelerometer* merupakan sensor untuk mengukur percepatan yang dialami oleh suatu objek dengan mengacu pada *frame* inersia. *Accelerometer* memiliki banyak jenis salah satunya *piezo-electric crystal*. Namun *accelerometer* jenis ini sangat besar sedangkan kebutuhan mengharuskan memiliki ukuran yang kecil. Sehingga, dikembangkanlah *Micro Electronical-Mechanical Systems* (MEMS). Pada tahun 1979 *accelerometer* yang berukuran kecil berhasil dikembangkan oleh Stanford University. Namun, butuh waktu 15 tahun agar *accelerometer* ini baru dapat digunakan pada skala yang besar. Kehadiran *accelerometer* MEMS sangat mendukung perkembangan teknologi yang sangat luas dipergunakan, seperti *airbag* pada keselamatan otomotif. Keunggulan MEMS selain ukurannya yang kecil MEMS juga sedikit mengkonsumsi energi. MEMS *accelerometer* dapat digunakan juga untuk multi-sensor di mana dapat meningkatkan akurasi.

Skala operasi dari *accelerometer* biasanya dinyatakan dalam satuan  $g$ , di mana untuk  $1g = 9.80665 \text{ m/s}^2$ . Perlu diketahui bahwa percepatan yang disebabkan oleh gravitasi bumi bergantung pada posisi bumi. Pada umumnya tubuh manusia dapat menahan hanya beberapa  $g$  saja, namun ada beberapa bagian tubuh yang mengalami lebih dari  $10g$  tapi pada waktu singkat saja. Oleh karena itu sensor *accelerometer* yang digunakan pada sistem navigasi dapat beroperasi sekurang-kurangnya  $10g$ . skala yang lebih besar dibutuhkan untuk peluru kendali dan pesawat tanpa awak. Besar kecilnya skala pengoperasian *accelerometer* berbanding lurus dengan eror kuantisasi.

Percepatan merupakan gaya yang bekerja pada suatu massa dibagi massa itu sendiri. Gaya tersebut dapat berupa gaya dinamik yaitu gaya yang diakibatkan oleh pergerakan atau getaran yang dialami wahana. Gaya juga ada yang statik yaitu salah satu contohnya gaya yang menahan kita yang berlawanan dengan gaya gravitasi bumi. Perlu diketahui bahwa sensor *accelerometer* tidak mengukur gaya gravitasi yang disebabkan oleh bumi, melainkan gaya yang melawan gaya gravitasi. pengukuran dari ketiga sumbu sensor *accelerometer* dapat ditulis sebagai berikut:

$$\mathbf{f} = \mathbf{a} - \mathbf{g} \quad (2.1)$$

Di mana

$\mathbf{f}$  : merupakan vektor  $3 \times 1$  dari ketiga sumbu gaya spesifik,

$\mathbf{a}$  : merupakan percepatan *accelerometer* terhadap kerangka inersia yang merupakan vektor  $3 \times 1$  mewakili ketiga sumbu,

$\mathbf{g}$  : percepatan gravitasi yang dinyatakan dengan vektor  $3 \times 1$  dan memiliki nilai bergantung pada posisi di bumi,

Percepatan  $\mathbf{a}$  dapat dinyatakan dalam bentuk turunan kedua dari vektor posisi  $\mathbf{r}$  yaitu:

$$\mathbf{a} = \frac{d^2\mathbf{r}}{dt^2} = \ddot{\mathbf{r}} \quad (2.2)$$

### 2.3.2 Gyroscope

Pada sensor IMU selain mengukur gerakan translasi, agar dapat menggambarkan gerakan dalam tiga dimensi, maka dibutuhkan pengetahuan mengenai gerakan rotasi. Oleh karena itu, sensor *gyroscope* digunakan. Sensor *gyroscope* merupakan sensor yang mengukur kecepatan rotasi sudut. Sama seperti sensor inersia lainnya yaitu sensor *accelerometer*, *gyroscope* mengukur kecepatan rotasi sudut mengacu pada *frame* inersia. Nilai dari pengukuran *gyroscope* berada pada sumbu *body frame*. *Body frame* dari *gyroscope* adalah kotak tempat diletakkannya *gyroscope*. Bila *gyroscope* di letakan tepat di titik berat dari wahana maka *body frame* dari *gyroscope* ialah wahana tersebut.

Ada beberapa metode yang dapat digunakan untuk mengukur gerakan rotasi antara lain yaitu:

1. ***Spinning-mass gyros***, di mana metode ini menggunakan prinsip kekekalan momentul angular. Sebuah motor digunakan untuk memutar sebuah massa pada sebuah sumbu tertentu. metode ini kurang unggul dari pada metode lain yang akan dibahas setelah ini.
2. ***Optical gyros***, metode ini menggunakan kecepatan cahaya untuk dapat mendeteksi gerakan rotasi. Ada dua jenis desain dari metode ini yaitu, *Ring Laser Gyros* (RLG). RLG merupakan teknologi yang berperforma tinggi sehingga pengaplikasiannya relatif lebih mahal. Pada tahun 1970an dikembangkan *Interferometric Fiber-Optic Gyro* (IFOG) sebagai solusi harga yang lebih murah untuk metode *optical gyros*. Perkembangan dari IFOG menghasilkan *range* performa yang lebih beragam bahkan hingga *aviation grade*. Selain kedua disain tadi, ada beberapa metode *optical gyros*

yang dalam tahap pengembangan yaitu, *Resonant Fiber-Optic Gyros* (RFOG) dan *Micro-Optic Gyro* (MOG) [3].

3. **Vibratory gyro**, di mana pertama kali dikembangkan sekitar tahun 1980an. MEMS *gyro* menggunakan metode ini. *Vibratory gyro* memiliki sebuah bagian yang dibuat mengalami gerakan harmoik. Bagian itu dapat berupa sehelai senar, cincin, silinder atau sebuah *beam*. Bagian itu mendeteksi percepatan *coriolis* ketika *gyro* berotasi.

Selain ketiga metode menentukan sebuah rotasi di atas, terdapat sebuah metode yang disebut *cold-atom interferometric*. Metode tersebut memiliki akurasi yang sangat tinggi dibanding ketiga metode di atas.

Tiga *gyroscope* dibutuhkan untuk dapat mengamati gerakan dalam tiga dimensi. Pengukuran *gyroscope*  $\omega_{ib}^b$  merupakan vektor kolom yang terdiri dari tiga elemen mewakili ketiga sumbu *gyroscope*. *Gyroscope* mengukur rotasi *body* terhadap *local navigation frame*, rotasi *local navigation frame* terhadap *frame* bumi dan rotasi bumi terhadap *frame* inersia [2]. Sehingga dapat ditulis sebagai berikut.

$$\omega_{ib}^b = \omega_{ie}^b + \omega_{en}^b + \omega_{nb}^b \quad (2.3)$$

Di mana

$\omega_{ib}^b$  : kecepatan rotasi *body gyroscope* terhadap *frame* inersia,

$\omega_{nb}^b$  : kecepatan rotasi *body gyroscope* terhadap *local navigation frame*,

$\omega_{en}^b$  : kecepatan rotasi *local navigation frame* terhadap *frame* bumi,

$\omega_{ie}^b$  : kecepatan rotasi bumi terhadap *frame* inersia.

*Gyroscope* yang mampu mengukur kecepatan rotasi bumi minimal *aviation grade* atau yang lebih baik.

### 2.3.3 Error pada IMU

Sensor IMU tidak dapat dilepaskan dari eror yang menyertainya. Baik *accelerometer* atau *gyroscope* memiliki eror yang terdiri dari eror sistematik dan dinamik. Eror sistematik terdiri dari bias, *scale factor* dan *cross coupling eror*. Pengaruh suhu lingkungan merupakan eror sistematik juga. Eror sistematik dapat dihilangkan dengan melakukan proses *alignment* sebelum IMU digunakan. Ada beberapa eror sistematik yang dapat dikoreksi hanya dengan menggunakan data dari kalibrasi

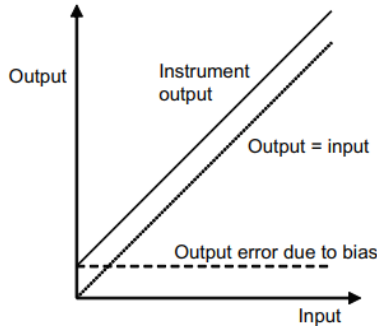
laboratorium yaitu eror pengaruh suhu. Sedangkan eror dinamik merupakan eror yang tidak memiliki nilai yang tetap selama sensor IMU beroperasi. Eror ini dimodelkan dengan *white noise* oleh karena itu digunakan Filter Kalman untuk mengkoreksi eror dinamik. Persamaan pengukuran pada sensor IMU dapat ditulis sebagai berikut:

$$\begin{aligned}\hat{f}_{ib}^b &= f_{ib}^b + M_a f_{ib}^b + b_a + w_a \\ \hat{\omega}_{ib}^b &= \omega_{ib}^b + M_g \omega_{ib}^b + b_g + G f_{ib}^b + w_g\end{aligned}\quad (2.4)$$

Dimana  $f_{ib}^b$  dan  $\omega_{ib}^b$  ialah gaya spesifik dan kecepatan sudut sebenarnya, sedangkan  $b_a$ ,  $b_g$ ,  $M_a$ ,  $M_g$ ,  $w_a$  dan  $w_g$  merupakan eror yang akan dijelaskan.  $G$  merupakan matriks gravitasi *dependent*.

### 2.3.3.1 Eror bias

Eror ini merupakan eror yang konstan selama sensor *accelerometer* dan *gyroscope* dioperasikan. Eror ini merupakan eror yang tidak bergantung dari gaya spesifik atau kecepatan sudut yang diukur sensor. Gambar 2.9 memperlihatkan efek dari eror berupa bias.

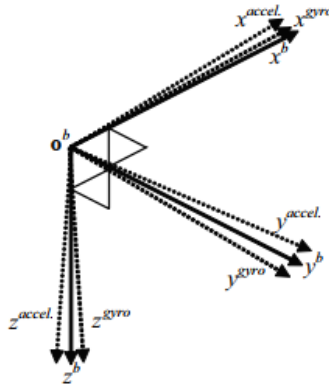


**Gambar 2.9** Pengaruh bias terhadap nilai sebenarnya yang diukur oleh sensor

Bias sendiri terdiri dari dua bentuk bias yaitu bias yang bersifat statis dan bias yang bersifat dinamis. Dapat ditulis sebagai berikut:

$$\begin{aligned}b_a &= b_{as} + b_{ad} \\ b_g &= b_{gs} + b_{gd}\end{aligned}\quad (2.5)$$

$b_{as}$  dan  $b_{gs}$  merupakan bias yang statis. Namun, akan berbeda-beda ketika IMU dihidupkan. Sedangkan  $b_{ad}$  dan  $b_{gd}$  adalah bias yang bersifat dinamis. Bias ini bersifat acak saat IMU dioperasikan. Besar nilai dari eror ini sekitar 10% dari bias statis.



**Gambar 2.10** Misalignment sumbu sensor IMU terhadap *body frame*

### 2.3.3.2 *Error Scale factor dan cross-coupling*

Error *scale factor* merupakan error yang bergantung dari nilai yang diukur oleh sensor. error ini berbanding lurus dengan hasil pengukuran sensor *accelerometer* dan *gyroscope* pada masing-masing sumbu. Persamaan error *scale factor* untuk sensor *accelerometer* dapat dirumuskan sebagai berikut:

$$\text{error scale factor} = \mathbf{s}_a \mathbf{f}_{ib}^b \quad (2.6)$$

Di mana  $\mathbf{s}_a$  merupakan vektor kolom dari *scale factor* untuk setiap sumbu *accelerometer* atau:

$$\mathbf{s}_a = \begin{bmatrix} s_{a,x} \\ s_{a,y} \\ s_{a,z} \end{bmatrix}$$

Sedangkan error scale factor untuk *gyroscope* sebagai berikut:

$$\text{error scale factor} = \mathbf{s}_g \boldsymbol{\omega}_{ib}^b \quad (2.7)$$

Di mana  $\mathbf{s}_g$  adalah vektor kolom yang mewakili *scale factor* untuk setiap sumbu pada *gyroscope* yang dapat ditulis sebagai berikut:

$$\mathbf{s}_g = \begin{bmatrix} s_{g,x} \\ s_{g,y} \\ s_{g,z} \end{bmatrix}$$



Eror *cross-coupling* pada sensor *accelerometer* dan *gyroscope* berasal dari sumbu sensitif yang tidak saling tegak lurus satu sama lain atau ada selisih beberapa mrad dari sumbu *body frame* yang ortogonal. Hal ini sering disebut *misalignment* sumbu sensor IMU terhadap sumbu *body frame*. Gambar 2.10 mengilustrasikan hal tersebut.

Hal ini menyebabkan salah satu sumbu *accelerometer* sensitif terhadap gaya spesifik pada sumbu lain yang ortogonal. Begitu pula pada *gyroscope* yang sensitif pada kecepatan sudut pada sumbu lain yang ortogonal. Notasi yang digunakan untuk eror *cross-coupling* yaitu  $m_{\alpha,\beta}$  yang mewakili koefisien untuk gaya spesifik pada sumbu  $\beta$  yang sensitif pula pada sumbu  $\alpha$ , sedangkan  $m_{g,\beta}$  mewakili koefisien untuk kecepatan sudut pada sumbu  $\beta$  yang sensitif juga pada sumbu  $\alpha$ . Pada *datasheet*, eror *cross-coupling* dinyatakan dalam eror *misalignment*.

Perlu diketahui bahwa *scale factor* dan *cross-coupling* merupakan koefisien yang tidak memiliki satuan. Namun pada umumnya dinyatakan dalam *parts per million* (PPM) atau dalam bentuk persentase. Kebanyakan sensor *accelerometer* dan *gyroscope* memiliki koefisien *scale factor* dan *cross-coupling* sekitar  $10^{-4}$  dan  $10^{-3}$  (100-1000 PPM). Namun pada beberapa *grade* sensor IMU seperti *consumer-grade* memiliki koefisien *scale factor* dan *cross-coupling* hingga 0.1 atau 100.000 PPM (10%).

*Scale factor* dan *cross-coupling* dapat ditulis dalam bentuk matriks seperti berikut:

$$\mathbf{M}_a = \begin{bmatrix} S_{a,x} & m_{a,xy} & m_{a,xz} \\ m_{a,yx} & S_{a,y} & m_{a,yz} \\ m_{a,zx} & m_{a,zy} & S_{a,z} \end{bmatrix} \quad (2.8)$$

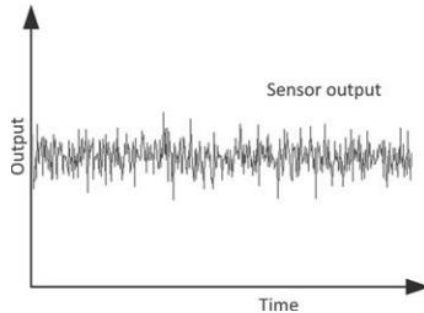
$$\mathbf{M}_g = \begin{bmatrix} S_{g,x} & m_{g,xy} & m_{g,xz} \\ m_{g,yx} & S_{g,y} & m_{g,yz} \\ m_{g,zx} & m_{g,zy} & S_{g,z} \end{bmatrix} \quad (2.9)$$

Sehingga total dari eror *scale factor* dan *cross-coupling* adalah  $\mathbf{M}_a \mathbf{f}_{ib}^b$  untuk eror pada gaya spesifik dan  $\mathbf{M}_g \boldsymbol{\omega}_{ib}^b$  untuk eror pada kecepatan sudut.

### 2.3.3.3 Error acak

Eror yang bersifat acak tidak dapat dilepaskan dari sensor *accelerometer* dan *gyroscope* yang disimbolkan dengan  $\mathbf{w}$ . Pada Tugas Akhir ini eror

acak tersebut didekati dengan model stokastik berupa *white noise*. Model *white noise* digunakan karena eror acak dari sensor IMU tidak saling berkorelasi. Eror acak ini dapat berasal dari sumber daya dari sensor atau berasal dari komponen elektronik pada MEMS IMU. Gambar 2.11 memperlihatkan bentuk dari sinyal yang berupa *white noise*.



**Gambar 2.11** Bentuk sinyal yang merupakan *white noise* [1].

Produsen dari sensor IMU menyatakan eror acak dari sensor yang mereka produksi dengan *velocity random walk* (VRW) untuk *accelero* dalam satuan  $\mu g/\sqrt{Hz}$  atau  $m/s/\sqrt{hr}$ . *angle random walk* (ARW) untuk sensor *gyroscope* dalam satuan  $deg/hr/\sqrt{Hz}$  atau  $deg/hr$ .

## 2.4 *Global Positioning System* (GPS)

GNSS dapat memberikan hasil navigasi tiga dimensi dalam hal ini posisi dengan akurasi beberapa meter, di mana sistem navigasi terestrial hanya terbatas pada hasil navigasi dua dimensi dikarenakan geometri dari titik pemancar sinyal. GNSS merupakan sistem navigasi yang lebih akurat dibandingkan dengan sistem navigasi terestrial kecuali UWB dan untuk sekarang GNSS merupakan teknologi dengan metode *position fixing*. Salah satu jenis GNSS adalah *Global Positioning System* (GPS).

Pada tahun 1970an mentri pertahanan US mengembangkan *Global Positioning Systems* untuk keperluan navigasi militer [3]. GPS bekerja dengan menggunakan bantuan sinyal dari satelit yang diketahui posisinya. Ada dua sinyal yang dikirimkan oleh satelit, yaitu *Pseudo-Random Noise* (PRN) dan informasi navigasi. PRN digunakan oleh *receiver* untuk

menentukan lamanya sinyal dari satelit merambat ke *receiver* ( $\Delta t$ ) yang kemudian dikalikan dengan kecepatan cahaya ( $c$ ) untuk mendapatkan *pseudo-range* ( $\rho$ ). Dapat ditulis sebagai berikut:

$$\rho = c \times \Delta t \quad (2.10)$$

Pada persamaan (2.10) digunakan oleh satelit untuk menentukan *Pseudo-range* namun GPS menggunakan paling sedikit 4 satelit yang tersedia agar dapat menghasilkan hasil navigasi, oleh karena itu dapat ditulis sebagai berikut:

$$\rho^m = r^m + c\delta t_r - c\delta t_s + I^m + \varepsilon_\rho^m \quad (2.11)$$

Di mana

- $\rho^m$  : *pseudo-range* dari satelit ke- $m$  dan *receiver* (meter)
- $r^m$  : jarak sebenarnya antara antenna *receiver* pada waktu  $t_r$  dan antena satelit pada waktu  $t_t$  (meter)
- $\delta t_r$  : *offset* jam pada *receiver* (detik)
- $\delta t_s$  : *offset* jam pada satelit (detik)
- $I^m$  : *delay* yang disebabkan oleh *ionospheric* (detik)
- $T^m$  : *delay* yang disebabkan oleh lapisan *tropospheric* (detik)
- $\varepsilon_\rho^m$  : eror yang disebabkan oleh banyak jenis eror yaitu gangguan pada *receiver*, *multipath*, dan sebagainya

GPS didesain untuk menyusun sebuah konstelasi yang terdiri atas 24 atau lebih satelit yang mengorbit bumi pada radius antara 25,000 hingga 30,000 km dari pusat bumi, untuk memastikan bahwa sinyal dari sedikitnya empat satelit dapat diterima *receiver* di permukaan bumi manapun. Pada prakteknya, jumlah satelit yang terpantau oleh *receiver* bisa lebih dari empat, hal ini membuat akurasi dari estimasi posisi dapat lebih baik dan konsisten saat sistem navigasi digunakan. Namun, jumlah satelit yang terlihat oleh *receiver* bisa juga kurang dari 4, hal ini dikarenakan beberapa hal, seperti gedung-gedung yang tinggi atau pepohonan yang lebat yang menghalangi sinyal transmisi sampai pada *receiver*. Kondisi ini merupakan salah satu gangguan yang terjadi pada GPS dan dapat menurunkan akurasi GPS dalam mengestimasi jarak dan posisi *receiver* di permukaan bumi.

## 2.5 Model Error INS [3]

Model yang digunakan pada Tugas Akhir ini merupakan model eror dari INS. Model eror dari INS merupakan pemodelan matematika dari eror INS. Pada Tugas Akhir ini digunakan 15 *state* yang merupakan eror dari INS yaitu:

$$\mathbf{x} = \begin{bmatrix} \delta\psi \\ \delta\mathbf{v}_{eb}^n \\ \delta\mathbf{p}_b \\ \mathbf{b}_a \\ \mathbf{b}_g \end{bmatrix} \quad (2.12)$$

Cara mendapatkan model eror INS yaitu dengan mengekspresikan selisih antara sinyal IMU yang terukur dengan sinyal IMU sebenarnya ( $\mathbf{f}_{ib}^b$  dan  $\boldsymbol{\omega}_{ib}^b$ ) yang direpresentasikan dengan persamaan berikut:

$$\dot{\mathbf{x}} = \mathbf{F}\mathbf{x} + \mathbf{w} \quad (2.13)$$

$\mathbf{w}$  merupakan eror yang bersifat acak dan  $\mathbf{F}$  dinyatakan dengan persamaan berikut:

$$\mathbf{F} = \begin{bmatrix} \mathbf{F}_{11} & \mathbf{F}_{12} & \mathbf{F}_{13} & \mathbf{0}_3 & \widehat{\mathbf{C}}_b^n \\ \mathbf{F}_{21} & \mathbf{F}_{22} & \mathbf{F}_{23} & \widehat{\mathbf{C}}_b^n & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{F}_{32} & \mathbf{F}_{33} & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \end{bmatrix} \quad (2.14)$$

Di mana

$$\mathbf{F}_{11} = -[\widehat{\boldsymbol{\omega}}_{in}^n] \quad (2.15)$$

$$\mathbf{F}_{12} = \begin{bmatrix} 0 & \frac{-1}{R_E(\widehat{L}_b) + \widehat{h}_b} & 0 \\ \frac{1}{R_N(\widehat{L}_b) + \widehat{h}_b} & 0 & 0 \\ 0 & \frac{\tan \widehat{L}_b}{R_E(\widehat{L}_b) + \widehat{h}_b} & 0 \end{bmatrix} \quad (2.16)$$

$$\mathbf{F}_{13} = \begin{bmatrix} \boldsymbol{\omega}_{ie} \sin \hat{L}_b & 0 & \frac{\hat{v}_{eb,E}^n}{(R_E(\hat{L}_b) + \hat{h}_b)^2} \\ 0 & 0 & \frac{-\hat{v}_{eb,N}^n}{(R_N(\hat{L}_b) + \hat{h}_b)^2} \\ \left( \begin{array}{c} \boldsymbol{\omega}_{ie} \cos \hat{L}_b \\ + \frac{\hat{v}_{eb,E}^n}{(R_E(\hat{L}_b) + \hat{h}_b)^2 \cos \hat{L}_b} \end{array} \right) & 0 & \frac{-\hat{v}_{eb,E}^n \tan \hat{L}_b}{(R_E(\hat{L}_b) + \hat{h}_b)^2} \end{bmatrix} \quad (2.17)$$

$$\mathbf{F}_{21} = -[(\hat{\mathbf{C}}_b^n \mathbf{f}_{ib}^b)^\wedge] \quad (2.18)$$

$$\mathbf{F}_{22} = \begin{bmatrix} \frac{\hat{v}_{eb,D}^n}{R_N(\hat{L}_b) + \hat{h}_b} & \left( \begin{array}{c} -\frac{2\hat{v}_{eb,E}^n \tan \hat{L}_b}{R_E(\hat{L}_b) + \hat{h}_b} \\ -2\boldsymbol{\omega}_{ie} \sin \hat{L}_b \end{array} \right) & \frac{\hat{v}_{eb,N}^n}{R_N(\hat{L}_b) + \hat{h}_b} \\ \left( \begin{array}{c} \frac{\hat{v}_{eb,E}^n \tan \hat{L}_b}{R_E(\hat{L}_b) + \hat{h}_b} \\ +2\boldsymbol{\omega}_{ie} \sin \hat{L}_b \end{array} \right) & \frac{\hat{v}_{eb,N}^n \tan \hat{L}_b + \hat{v}_{eb,D}^n}{R_E(\hat{L}_b) + \hat{h}_b} & \left( \begin{array}{c} \frac{\hat{v}_{eb,E}^n \tan \hat{L}_b}{R_E(\hat{L}_b) + \hat{h}_b} \\ +2\boldsymbol{\omega}_{ie} \sin \hat{L}_b \end{array} \right) \\ -\frac{2\hat{v}_{eb,N}^n}{R_N(\hat{L}_b) + \hat{h}_b} & \left( \begin{array}{c} -\frac{2\hat{v}_{eb,E}^n}{R_N(\hat{L}_b) + \hat{h}_b} \\ -2\boldsymbol{\omega}_{ie} \sin \hat{L}_b \end{array} \right) & 0 \end{bmatrix} \quad (2.19)$$

$$\mathbf{F}_{23} = \begin{bmatrix} \left( \begin{array}{c} -\frac{(\hat{v}_{eb,E}^n)^2 \sec^2 \hat{L}_b}{R_E(\hat{L}_b) + \hat{h}_b} \\ -2\hat{v}_{eb,E}^n \boldsymbol{\omega}_{ie} \cos \hat{L}_b \end{array} \right) & 0 & \left( \begin{array}{c} \frac{(\hat{v}_{eb,E}^n)^2 \tan \hat{L}_b}{(R_E(\hat{L}_b) + \hat{h}_b)^2} \\ -\frac{\hat{v}_{eb,N}^n \hat{v}_{eb,D}^n}{(R_N(\hat{L}_b) + \hat{h}_b)^2} \end{array} \right) \\ \left( \begin{array}{c} \frac{\hat{v}_{eb,N}^n \hat{v}_{eb,E}^n \sec^2 \hat{L}_b}{R_E(\hat{L}_b) + \hat{h}_b} \\ +2\hat{v}_{eb,N}^n \boldsymbol{\omega}_{ie} \cos \hat{L}_b \\ -2\hat{v}_{eb,D}^n \boldsymbol{\omega}_{ie} \sin \hat{L}_b \end{array} \right) & 0 & \frac{\hat{v}_{eb,N}^n \hat{v}_{eb,E}^n \tan \hat{L}_b + \hat{v}_{eb,E}^n \hat{v}_{eb,D}^n}{(R_E(\hat{L}_b) + \hat{h}_b)^2} \\ 2\hat{v}_{eb,E}^n \boldsymbol{\omega}_{ie} \sin \hat{L}_b & 0 & \left( \begin{array}{c} \frac{(\hat{v}_{eb,E}^n)^2}{(R_E(\hat{L}_b) + \hat{h}_b)^2} - \frac{2g_0(\hat{L}_b)}{\mathbf{r}_{eS}^e(\hat{L}_b)} \\ + \frac{(\hat{v}_{eb,N}^n)^2}{(R_N(\hat{L}_b) + \hat{h}_b)^2} \end{array} \right) \end{bmatrix} \quad (2.20)$$

$$\mathbf{F}_{32} = \begin{bmatrix} \frac{1}{R_N(\hat{L}_b) + \hat{h}_b} & 0 & 0 \\ 0 & \frac{1}{(R_E(\hat{L}_b) + \hat{h}_b) \cos \hat{L}_b} & 0 \\ 0 & 0 & -1 \end{bmatrix} \quad (2.21)$$

$$\mathbf{F}_{33} = \begin{bmatrix} 0 & 0 & -\frac{\hat{v}_{eb,N}^n}{(R_N(\hat{L}_b) + \hat{h}_b)^2} \\ \frac{\hat{v}_{eb,E}^n \sin \hat{L}_b}{(R_E(\hat{L}_b) + \hat{h}_b) \cos^2 \hat{L}_b} & 0 & -\frac{\hat{v}_{eb,E}^n}{(R_E(\hat{L}_b) + \hat{h}_b)^2 \cos \hat{L}_b} \\ 0 & 0 & 0 \end{bmatrix} \quad (2.22)$$

## 2.6 Model Gravitasi

Pada umumnya nilai gravitasi bumi pada permukaan bumi bernilai  $9.8 \text{ m/s}$ , namun nilai gravitasi bergantung pada posisi terhadap bumi. Konsepnya semakin tinggi dari permukaan bumi maka gravitasi bumi akan berkurang. Oleh karena itu, agar mendapatkan hasil navigasi yang lebih presisi, penentuan nilai gravitasi perlu dilakukan secara akurat. Sehingga model gravitasi diperlukan dalam hal ini.

Model gravitasi pada Tugas Akhir ini merupakan fungsi dari latitude ( $L_b$ ) dan height ( $h_b$ ) atau dapat ditulis

$$\mathbf{g}(L_b, h_b) = \begin{bmatrix} g_{b,N}^n \\ g_{b,E}^n \\ g_{b,D}^n \end{bmatrix} \quad (2.23)$$

Di mana masing-masing komponen dirumuskan adalah sebagai berikut:

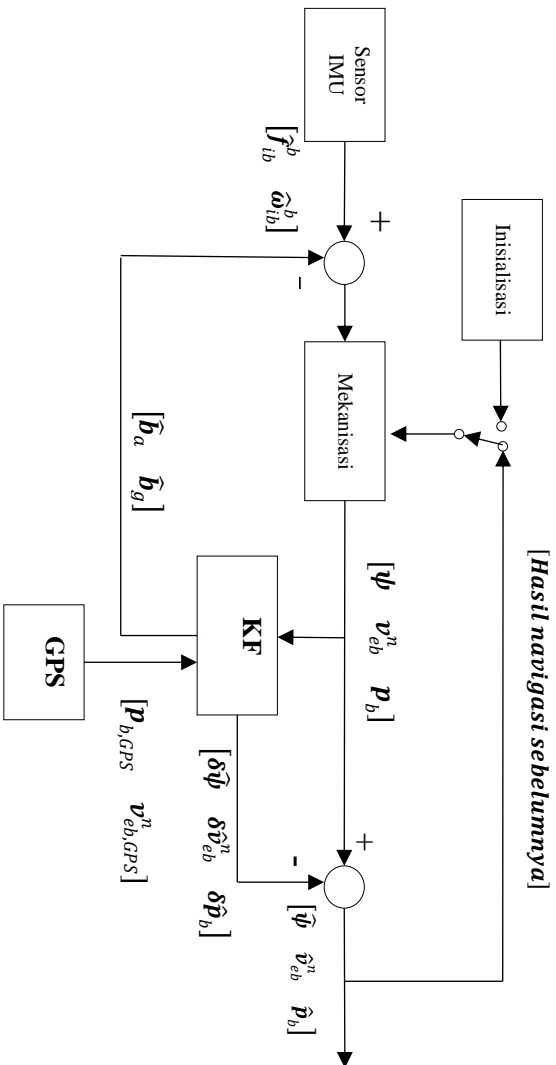
$$\begin{aligned} g_{b,N}^n &= -8.08 \times 10^{-9} h_b \sin 2L_b \\ g_{b,E}^n &= 0 \\ g_{b,D}^n &= g_0(L_b) \left\{ 1 - \frac{2}{R_0} \left[ \frac{1 + f(1 - 2 \sin^2 L_b)}{+ \frac{\omega_{ie}^2 R_0^2 R_p}{\mu}} \right] h_b + \frac{3}{R_0^2} h_b^2 \right\} \end{aligned} \quad (2.24)$$

Di mana  $g_0(L_b)$  dikenal sebagai model somigliana yang dirumuskan sebagai berikut:

$$g_0(L_b) = 9.7803253359 \frac{(1 + 0.001931853 \sin^2 L_b)}{\sqrt{1 - e^2 \sin^2 L_b}} \quad (2.25)$$

## 2.7 Tahapan Pengolahan sinyal IMU

Sinyal IMU tidak dalam bentuk hasil navigasi yang berupa posisi, kecepatan dan *attitude* melainkan percepatan yang diukur oleh *accelero* dan berupa kecepatan sudut yang diukur oleh *gyroscope*. Oleh karena itu, sinyal dari IMU agar dapat menghasilkan hasil navigasi maka perlu melewati tahapan pengolahan yang terdiri dari proses *alignment*, mekanisasi serta proses estimasi untuk mendapatkan hasil navigasi yang lebih presisi. Gambaran besar dari tahapan pengolahan sinyal IMU dapat dilihat pada Gambar 2.12.



Gambar 2.12 Tahapan pengolahan sinyal IMU.



Namun sebelum tahap-tahap sinyal IMU tersebut, terlebih dahulu dilakukan proses inisialisasi hasil navigasi mengingat metode IMU merupakan *dead reckoning* yang membutuhkan hasil navigasi sebelumnya untuk menentukan hasil navigasi sekarang. Selain proses inisialisasi, perlu pula mengidentifikasi parameter sistem.

### 2.7.1 Proses *Alignment*

Pada sistem navigasi inisialisasi kondisi awal dibutuhkan antara lain seperti posisi, kecepatan dan attitude awal. Pada inisialisasi posisi dapat menggunakan sumber informasi eksternal. Pada keadaan diam inisialisasi kecepatan dapat diketahui yaitu  $0\text{ m/s}$ . Sedangkan untuk inisialisasi attitude awal dapat menggunakan proses *leveling* dan *gyrocompassing*. Proses *leveling* digunakan untuk melakukan inisialisasi pada *roll* dan *pitch*. Pada keadaan benar-benar diam, *accelero* hanya akan mengukur percepatan yang disebabkan oleh gravitasi bumi, di mana dengan pengetahuan tersebut, *roll* dan *pitch* awal dapat diperoleh. Kemudian untuk mendapatkan *yaw* digunakan metode *gyrocompassing*, di mana prinsip dari metode ini adalah bila pada keadaan benar-benar diam (*absolutely stationary*), sensor *gyroscope* hanya mengukur kecepatan rotasi bumi. Akan tetapi, hanya *gyroscope* dengan *grade* tertentu yang dapat mengukur kecepatan bumi tersebut seperti *aviation-grade*. apalagi bila kecepatan bumi jauh lebih kecil dari gangguan yang berupa getaran yang dialami oleh sensor *gyroscope*. Proses inisialisasi attitude dengan menggunakan proses *leveling* dan *gyrocompassing* disebut juga sebagai *coarse alignment*.

Pada proses *coarse alignment* tidak dilakukan koreksi terhadap eror yang dialami baik pada sensor *accelero* dan *gyroscope*. Akan tetapi dengan nilai awal attitude dari hasil *coarse alignment* walaupun memiliki eror, dapat dilakukan *fine alignment*. Tujuan dari *fine alignment* adalah mengurangi eror pada attitude yang diperoleh dari *coarse alignment*. Proses ini menggunakan filter kalman untuk mengestimasi eror pada attitude. Hasil eror tersebut digunakan untuk mengkoreksi attitude.

### 2.7.2 Proses Mekanisasi

Pada bagian ini akan diuraikan mengenai proses mekanisasi [1]. Ada literatur yang menggunakan istilah *persamaan navigasi* untuk waktu kontinyu dan *mekanisasi* untuk waktu diskrit [3]. Dalam Tugas Akhir ini akan menggunakan sistem yang diskrit maka, untuk selanjutnya akan

menggunakan istilah mekanisasi. Mekanisasi merupakan proses yang mengubah hasil dari sensor IMU menjadi hasil navigasi di mana hasil tersebut berupa posisi, kecepatan dan *attitude*. Proses mekanisasi merupakan proses yang berulang sesuai dengan data dari sensor dengan iterasi sebanyak data IMU yang terukur dan dalam waktu interval periode data IMU. Pada proses mekanisasi, selain mengubah data percepatan dan kecepatan sudut yang diukur oleh sensor *accelero* dan *gyroscope* menjadi posisi, kecepatan dan *attitude*, proses mekanisasi juga melakukan transformasi *frame*. Seperti yang disinggung pada pembahasan sebelumnya bahwa pada Tugas Akhir ini hasil navigasi ditampilkan pada sumbu dari *local navigation frame* oleh karena itu keluaran dari proses mekanisasi merupakan posisi, kecepatan dan *attitude* pada wahana yang menggunakan *resolving axis* dari *local navigation frame*.

Dalam mengukur pergerakan tiga dimensi suatu benda dibutuhkan tiga serangkai sensor *accelero* yang saling tegak lurus antara satu dan yang lain dan tiga sensor *gyro* yang saling tegak lurus satu sama lain pula. Tiga sensor *accelero* dan tiga sensor *gyro* dibutuhkan karena untuk mengukur pergerakan suatu objek dalam tiga dimensi membutuhkan enam derajat kebebasan yaitu tiga sumbu untuk gerakan linear dan tiga sumbu untuk mengukur gerakan angular.

Pada bagian 2.2 telah dijelaskan bahwa sensor IMU yang ada pada INS mengukur gaya spesifik dan kecepatan sudut. Keluaran IMU tersebut memiliki interval  $\tau_i$  untuk setiap data yang diukur. Pada Tugas Akhir ini menggunakan IMU yang memiliki frekuensi 2000Hz atau  $\tau_i = 5 \times 10^{-4}$  detik. Interval ini memberikan waktu pada setiap iterasi pada proses mekanisasi untuk melakukan perhitungan serta proses estimasi yang dilakukan oleh filter kalman.

Mekanisasi melakukan perhitungan yang membutuhkan  $f_{ib}^b$  dan  $\omega_{ib}^b$  dari masing-masing sensor *accelero* dan *gyro*. Kedua hasil dari IMU tersebut digunakan untuk menghitung *attitude* terlebih dahulu kemudian mengubah sumbu gaya spesifik yang berada pada *body frame* menjadi sumbu pada *local navigation frame* dan kemudian menghitung kecepatan dan posisi pada saat sekarang dengan menggunakan gaya spesifik  $f_{ib}^n$  yang telah ditransformasikan sebelumnya.

### 2.7.2.1 Menghitung Attitude

Seperti yang telah diuraikan pada subbab 2.3 bahwa sensor *accelero* dan *gyroscope* mengukur gaya spesifik dan kecepatan sudut dari

*body* IMU yang mengacu pada inersia *frame* dan menggunakan *resolving axis* dari *frame body* di mana *body* tersebut adalah *body* dari IMU. Perlu diketahui bahwa sensor-sensor pada IMU tidak mengukur gerakan yang mengacu pada bumi melainkan mengacu pada *frame* inersia dalam Tugas Akhir ini *frame* inersia tersebut yaitu *frame* ECI.

Dari penjelasan awal tersebut maka menghitung *attitude* dilakukan pada awal iterasi proses mekanisasi karena akan digunakan untuk mentransformasi sumbu dari *body* IMU menjadi *local navigation frame*. *Attitude* direpresentasikan dengan matriks transformasi koordinat yaitu sebagai berikut:

$$\mathbf{C}_{b,k}^n = \mathbf{C}_{b,k-1}^n (\mathbf{I}_3 + \mathbf{\Omega}_{ib,k}^b \tau_i) - (\mathbf{\Omega}_{ie,k-1}^n + \mathbf{\Omega}_{en,k-1}^n) \mathbf{C}_{b,k-1}^n \tau_i \quad (2.26)$$

Di mana  $\mathbf{\Omega}_{ib}^b$  merupakan matriks *skew* dari vektor  $\boldsymbol{\omega}_{ib}^b$  yang dapat ditulis sebagai berikut:

$$\mathbf{\Omega}_{ib}^b = [\boldsymbol{\omega}_{ib}^b \wedge] = \begin{bmatrix} 0 & -\omega_{ib,z}^b & -\omega_{ib,y}^b \\ \omega_{ib,z}^b & 0 & -\omega_{ib,x}^b \\ \omega_{ib,y}^b & \omega_{ib,x}^b & 0 \end{bmatrix} \quad (2.27)$$

Sedangkan  $\mathbf{\Omega}_{ie,k-1}^n$  dan  $\mathbf{\Omega}_{en,k-1}^n$  merupakan matriks *skew* kecepatan rotasi bumi dan *transport rate*. *Transport rate* dihasilkan dari rotasi sumbu *local navigation frame* yang bergerak dengan mengacu pada bumi [3]. Ketika IMU berada pada keadaan diam, maka *gyro* akan hanya membaca kecepatan rotasi bumi dan *transport rate*, di mana hal ini harus dikoreksi agar *attitude* tidak berubah saat keadaan diam.

Kecepatan putar bumi pada *local navigation frame* diuraikan pada [3], sehingga matriks *skew*-nya dapat ditulis:

$$\mathbf{\Omega}_{ie}^n = \begin{bmatrix} 0 & \omega_{ie} \sin L_b & 0 \\ -\omega_{ie} \sin L_b & 0 & -\omega_{ie} \cos L_b \\ 0 & \omega_{ie} \cos L_b & 0 \end{bmatrix} \quad (2.28)$$

$\mathbf{\Omega}_{ie}^n$  merupakan fungsi dari  $L_b$ , sehingga untuk mendapatkan  $\mathbf{\Omega}_{ie,k-1}^n$  menggunakan  $L_b$  pada saat sebelumnya.

*Transport rate* dapat diperoleh dengan menguraikan persamaan berikut:

$$\dot{\mathbf{C}}_e^n = -\mathbf{\Omega}_{en}^n \mathbf{C}_e^n \quad (2.29)$$

Untuk matriks transformasi koordinat  $\mathbf{C}_e^n$  dapat dilihat pada [3]. Dengan melakukan diferensiasi pada  $\mathbf{C}_e^n$  terhadap waktu didapatkan:

$$\dot{\mathbf{C}}_e^n = \begin{bmatrix} -\dot{\lambda}_b \cos L_b \\ -\dot{L}_b \\ -\dot{\lambda}_b \sin L_b \end{bmatrix} \mathbf{C}_e^n \quad (2.30)$$

Dengan mensubstitusikan persamaan (2.30) ke  $\dot{\mathbf{C}}_e^n$  pada persamaan (2.29) didapatkan:

$$\mathbf{\Omega}_{en}^n = \begin{bmatrix} 0 & -\omega_{en,z}^n & \omega_{en,y}^n \\ \omega_{en,z}^n & 0 & -\omega_{en,x}^n \\ -\omega_{en,y}^n & \omega_{en,x}^n & 0 \end{bmatrix} \quad (2.31)$$

Di mana

$$\boldsymbol{\omega}_{en}^n = \begin{bmatrix} \frac{v_{eb,E}^n}{R_E(L_b) + h_b} \\ \frac{v_{eb,N}^n}{R_N(L_b) + h_b} \\ \frac{v_{eb,E}^n \tan L_b}{R_E(L_b) + h_b} \end{bmatrix} \quad (2.32)$$

$R_E$  merupakan fungsi dari  $L_b$  yang digunakan untuk menentukan jari-jari bumi dan  $\mathbf{\Omega}_{en}^n$  merupakan fungsi dari  $L_b$  dan  $v_{eb}^n$ , sehingga untuk menghitung  $\mathbf{\Omega}_{en}^n$  menggunakan  $L_b$  dan  $v_{eb}^n$  pada saat sebelumnya.

### 2.7.2.2 *Mentransformasikan gaya spesifik*

Setelah menghitung *attitude* yang didapatkan dari mengolah data *gyro*, kemudian *attitude* dinyatakan dalam bentuk matriks transformasi  $\mathbf{C}_b^n$ . Langkah selanjutnya yaitu melakukan transformasi gaya spesifik

yang diukur oleh *accelero* pada *resolving axis body frame* menjadi *resolving axis local navigation frame*. Hal ini dikarenakan pada Tugas Akhir ini hasil navigasi ingin direpresentasikan berada pada *local navigation frame*. Persamaan dari transformasi gaya spesifik dapat ditulis:

$$\mathbf{f}_{ib}^n = \mathbf{C}_b^n \mathbf{f}_{ib}^b \quad (2.33)$$

Di mana  $\mathbf{C}_b^n$  merupakan matriks transformasi koordinat dan  $\mathbf{f}_{ib}^b$  merupakan hasil pengukuran dari sensor *accelero*.

### 2.7.2.3 Update Kecepatan

Tahap selanjutnya setelah mendapatkan gaya spesifik dengan *resolving axis local navigation frame* adalah melakukan proses integral sehingga mendapatkan kecepatan dari wahana. Kecepatan yang didapat dari proses integral dari  $\mathbf{f}_{ib}^n$  akan memiliki *resolving axis* pada *local navigation frame*. Sehingga kecepatan pada *resolving axis local navigation frame* dapat ditulis:

$$\mathbf{v}_{eb}^n = \mathbf{C}_e^b \mathbf{v}_{eb}^e \quad (2.34)$$

Dengan melakukan diferensial pada persamaan tersebut menjadi

$$\dot{\mathbf{v}}_{eb}^n = \mathbf{C}_e^n \mathbf{v}_{eb}^n + \mathbf{C}_e^n \dot{\mathbf{v}}_{eb}^e \quad (2.35)$$

Dari [3] diketahui bahwa

$$\dot{\mathbf{C}}_e^n = -\mathbf{\Omega}_{en}^n \mathbf{C}_e^n, \quad \dot{\mathbf{v}}_{eb}^e = -\mathbf{\Omega}_{ie}^e \mathbf{\Omega}_{ie}^e \mathbf{r}_{eb}^e - 2\mathbf{\Omega}_{ie}^e \mathbf{v}_{eb}^e + \mathbf{a}_{ib}^e$$

Sehingga persamaan (2.35) dapat ditulis

$$\dot{\mathbf{v}}_{eb}^n = -\mathbf{\Omega}_{ie}^n \mathbf{\Omega}_{ie}^n \mathbf{r}_{eb}^n - (\mathbf{\Omega}_{en}^n + 2\mathbf{\Omega}_{ie}^n) \mathbf{v}_{eb}^n + \mathbf{a}_{ib}^n$$

Di mana

$$\mathbf{a}_{ib}^n = \mathbf{f}_{ib}^n + \mathbf{g}_b^n(L_b, h_b) + \mathbf{\Omega}_{ie}^n \mathbf{\Omega}_{ie}^n \mathbf{r}_{eb}^n \quad (2.36)$$

Maka persamaan untuk menentukan kecepatan dapat ditulis sebagai berikut:

$$\begin{aligned} \mathbf{v}_{eb}^n &= \int_0^t (\mathbf{f}_{ib}^n + \mathbf{g}_b^n(L_b, h_b) - (\boldsymbol{\Omega}_{en}^n + 2\boldsymbol{\Omega}_{ie}^n)\mathbf{v}_{eb}^n) dt \\ \mathbf{v}_{eb,k}^n &= \mathbf{v}_{eb,k-1}^n + (\mathbf{f}_{ib,k}^n + \mathbf{g}_b^n(L_{b,k-1}, h_{b,k-1}) \\ &\quad - (\boldsymbol{\Omega}_{en,k-1}^n + 2\boldsymbol{\Omega}_{ie,k-1}^n)\mathbf{v}_{eb,k-1}^n)\tau_i \end{aligned} \quad (2.37)$$

#### 2.7.2.4 Update Posisi

Setelah mendapatkan kecepatan pada *resolving axis local navigation frame*, untuk mendapatkan posisi maka dilakukan integral pada  $\mathbf{v}_{eb}^n$ . Namun karena posisi pada *resolving axis local navigation frame* dinyatakan dengan lintang dan bujur maka persamaan untuk mendapatkan posisi pada *resolving axis local navigation frame* dapat ditulis sebagai berikut:

$$\begin{aligned} L_{b,k} &= L_{b,k-1} + \int_t^{t+\tau_i} \frac{v_{eb,k,N}^n}{R_N(L_{b,k}) + h_{b,k}} dt \\ \lambda_{b,k} &= \lambda_{b,k-1} + \int_t^{t+\tau_i} \frac{v_{eb,k,E}^n}{(R_E(L_b) + h_b) \cos L_{b,k}} dt \\ h_{b,k} &= h_{b,k-1} + \int_t^{t+\tau_i} v_{eb,k,D}^n dt \end{aligned} \quad (2.38)$$

#### 2.7.3 Filter Kalman

Pada tahun 1960an R.E. Kalman menghadapi masalah memisahkan sinyal dari gangguan *noise* aditif. Pada *paper* yang ditulis di tahun yang sama Kalman menggunakan sistem waktu diskrit dari pada pada domain waktu kontinyu. Kalman juga menggunakan konfigurasi *state space* untuk masalah ini. Penggunaan *state space* memungkinkan skenario *multiple input/multiple output* (MIMO). Algoritma yang dikerjakan Kalman untuk memisahkan *noise* aditif ini kemudian disebut Filter Kalman. Para insinyur terutama pada bidang navigasi dengan cepat menggunakan Filter Kalman untuk memfilter *noise* dari sinyal sebenarnya. Perkembangan komputer yang pesat pada tahun 1960an memviralkan aplikasi Filter Kalman untuk memisahkan sinyal dari *noise*.

Salah satunya pada komputer *Apollo* dengan kecepatan *clock* di bawah 100kHz mampu mengaplikasikan Filter Kalman dengan baik [5].

Optimasi pada algoritma Filter Kalman yaitu dengan cara meminimumkan *mean-square* eror estimasi dari suatu random variabel  $x$ . Oleh karena itu Filter Kalman merupakan filter yang menggunakan metode *least square* eror. Filter Kalman memerlukan dua buah persamaan matematika, yaitu model sistem dan model pengukuran dalam domain waktu diskrit.

$$\mathbf{x}_{k+1} = \boldsymbol{\phi}_k \mathbf{x}_k + \mathbf{w}_k \quad (2.39)$$

$$\mathbf{z}_k = \mathbf{H}_k \mathbf{x}_k + \mathbf{v}_k \quad (2.40)$$

Di mana

- $\mathbf{x}_k$  :  $(n \times 1)$  vektor *state* proses saat waktu  $t_k$ ,
- $\boldsymbol{\phi}_k$  :  $(n \times n)$  matriks transisi *state* antara  $\mathbf{x}_k$  dengan  $\mathbf{x}_{k+1}$ ,
- $\mathbf{w}_k$  :  $(n \times 1)$  vektor *noise* sistem yang merupakan *white noise*,
- $\mathbf{z}_k$  :  $(m \times 1)$  vektor pengukuran pada waktu  $t_k$ ,
- $\mathbf{H}_k$  :  $(m \times n)$  Matriks transisi *state* antara pengukuran  $\mathbf{z}_k$  dengan vektor *state*  $\mathbf{x}_k$ ,
- $\mathbf{v}_k$  :  $(m \times 1)$  vektor *noise* pengukuran merupakan *white noise* dan tidak berkorelasi dengan  $\mathbf{w}_k$ .

Sedangkan untuk matriks kovarian dari  $\mathbf{w}_k$  dan  $\mathbf{v}_k$  dapat ditulis sebagai berikut:

$$E[\mathbf{w}_k \mathbf{w}_i^T] = \begin{cases} \mathbf{Q}_k, & i = k \\ 0, & i \neq k \end{cases} \quad (2.41)$$

$$E[\mathbf{v}_k \mathbf{v}_i^T] = \begin{cases} \mathbf{R}_k, & i = k \\ 0, & i \neq k \end{cases} \quad (2.42)$$

Dan berlaku juga  $E[\mathbf{w}_k \mathbf{v}_i^T] = 0$ , untuk semua  $k$  dan  $i$ .

Sebagai inialisasi, diasumsikan bahwa telah memiliki estimasi awal dari proses pada waktu ke  $t_k$ , dan bahwa estimasi ini didasarkan pada proses yang terjadi sebelum proses waktu  $t_k$  atau estimasi ini

merupakan propagasi dari estimasi pada waktu  $t_k$ . Estimasi sementara ini dinotasikan dengan  $\hat{\mathbf{x}}_k^-$  dan perlu diketahui bahwa estimasi sementara ini merupakan estimasi terbaik sebelum disesuaikan dengan data pengukuran pada waktu  $t_k$ . Estimasi error dapat didefinisikan sebagai berikut:

$$\mathbf{e}_k^- = \mathbf{x}_k - \hat{\mathbf{x}}_k^- \quad (2.43)$$

Dari persamaan error (2.43) maka, Matriks kovarian error dapat ditulis sebagai berikut:

$$\mathbf{P}_k^- = E[\mathbf{e}_k^- \mathbf{e}_k^{-T}] = E[(\mathbf{x}_k - \hat{\mathbf{x}}_k^-)(\mathbf{x}_k - \hat{\mathbf{x}}_k^-)^T] \quad (2.44)$$

Tahap selanjutnya Setelah selesai mendapatkan propagasi dari estimasi sementara  $\hat{\mathbf{x}}_k^-$  adalah meningkatkan estimasi pada saat  $t_k$  dengan menggunakan data pengukuran  $\mathbf{z}_k$ . Pada tahap ini, hubungan propagasi estimasi *state*  $\hat{\mathbf{x}}_k^-$  dan data pengukuran  $\mathbf{z}_k$  agar mendapatkan hasil estimasi yang optimal dinyatakan dalam persamaan berikut:

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k(\mathbf{z}_k - \mathbf{H}_k \hat{\mathbf{x}}_k^-) \quad (2.45)$$

Di mana

$\hat{\mathbf{x}}_k$  : Estimasi terbaru

$\mathbf{K}_k$  : Gain kalman

Pada persamaan (2.45), hal yang perlu dilakukan untuk dapat menghasilkan estimasi *state* yang optimal adalah penentuan gain Kalman  $\mathbf{K}_k$ . Sama halnya dengan filter Wiener, sebagai indeks performansi dalam penentuan  $\mathbf{K}_k$  digunakan minimum *mean-square* error (MSE). Langkah pertama yang dibutuhkan yaitu menyatakan matriks kovarian error yang dapat ditulis sebagai berikut:

$$\mathbf{P}_k = E[\mathbf{e}_k \mathbf{e}_k^T] = E[(\mathbf{x}_k - \hat{\mathbf{x}}_k)(\mathbf{x}_k - \hat{\mathbf{x}}_k)^T] \quad (2.46)$$

Selanjutnya, menyubtitusi persamaan (2.40) ke persamaan (2.45) dan kemudian substitusikan hasil tersebut sebagai ekspresi dari  $\hat{\mathbf{x}}_k$  ke dalam persamaan (2.46) sehingga diperoleh persamaan sebagai berikut:

$$\mathbf{P}_k = E \left\{ \begin{array}{l} [(\mathbf{x}_k - \hat{\mathbf{x}}_k^-) - \mathbf{K}_k(\mathbf{H}_k \mathbf{x}_k + \mathbf{v}_k - \mathbf{H}_k \hat{\mathbf{x}}_k^-)] \\ [(\mathbf{x}_k - \hat{\mathbf{x}}_k^-) - \mathbf{K}_k(\mathbf{H}_k \mathbf{x}_k + \mathbf{v}_k - \mathbf{H}_k \hat{\mathbf{x}}_k^-)]^T \end{array} \right\} \quad (2.47)$$



$(\mathbf{x}_k - \hat{\mathbf{x}}_k^-)$  merupakan eror estimasi sebelumnya yang sama sekali tidak berkorelasi dengan eror pengukuran saat ini,  $\mathbf{v}_k$ , sehingga dengan menyelesaikan persamaan (2.47), dapat ditulis sebagai berikut:

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_k^- (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k)^T + \mathbf{K}_k \mathbf{R}_k \mathbf{K}_k^T \quad (2.48)$$

Perlu diketahui bahwa persamaan (2.48) adalah persamaan umum yang menyatakan matriks kovarian eror pada waktu  $t_k$ , dan berlaku untuk nilai  $\mathbf{K}_k$  manapun, suboptimal atau tidak.

Berikutnya, bentuk umum  $\mathbf{P}_k$  dari persamaan (2.48) diuraikan sehingga dapat dinyatakan dengan persamaan berikut:

$$\mathbf{P}_k = \mathbf{P}_k^- - \mathbf{K}_k \mathbf{H}_k \mathbf{P}_k^- - \mathbf{P}_k^- \mathbf{H}_k^T \mathbf{K}_k^T + \mathbf{K}_k (\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k) \mathbf{K}_k^T \quad (2.49)$$

Kembali ke persoalan penentuan nilai  $\mathbf{K}_k$ . Maka nilai  $\mathbf{K}_k$  yang optimal adalah yang meminimumkan MSE, di mana jumlah dari setiap elemen diagonal dari  $\mathbf{P}_k$  ekuivalen dengan MSE. Maka untuk menyatakan jumlah dari elemen diagonal dari  $\mathbf{P}_k$  digunakan operasi matriks yaitu *trace*. Selanjutnya *trace* dari matriks  $\mathbf{P}_k$  diturunkan terhadap  $\mathbf{K}_k$ , sehingga menghasilkan persamaan berikut:

$$\frac{d(\text{trace } \mathbf{P}_k)}{d\mathbf{K}_k} = -2(\mathbf{H}_k \mathbf{P}_k^-)^T + 2\mathbf{K}_k (\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k) \quad (2.50)$$

Dengan menggunakan syarat perlu, penyelesaian untuk *gain* optimal  $\mathbf{K}_k$  adalah

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k)^{-1} \quad (2.51)$$

*Gain*  $\mathbf{K}_k$  ini, adalah *gain* yang meminimisasi *mean-square estimation* eror.

Sedangkan matriks kovarian eror yang berhubungan dengan estimasi optimal dapat dinyatakan menggunakan persamaan berikut setelah mensubstitusikan persamaan (2.51) ke persamaan (2.49):

$$\mathbf{P}_k = \mathbf{P}_k^- - \mathbf{P}_k^- \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k)^{-1} \mathbf{H}_k \mathbf{P}_k^- \quad (2.52)$$

Atau

$$\mathbf{P}_k = \mathbf{P}_k^- - \mathbf{K}_k(\mathbf{H}_k\mathbf{P}_k^-\mathbf{H}_k^T + \mathbf{R}_k)\mathbf{K}_k^T \quad (2.53)$$

Atau

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\mathbf{P}_k^- \quad (2.54)$$

Didapat empat persamaan yang mengekspresikan perhitungan  $\mathbf{P}_k$  pada waktu  $t_k$  dari nilai  $\mathbf{P}_k^-$  sebelumnya. Tiga diantaranya, yakni persamaan (2.52), (2.53), dan (2.54), hanya dapat digunakan untuk kondisi *gain*  $\mathbf{K}_k$  yang optimal. Sementara itu, persamaan (2.48) valid atau dapat digunakan untuk nilai  $\mathbf{K}_k$  manapun baik yang optimal ataupun sub optimal.

Pada Tugas Akhir ini, agar tidak membebani perhitungan pada prosesor maka, persamaan dari matriks kovarian eror yang akan digunakan adalah yang paling sederhana, yaitu persamaan (2.54).

Sekarang data pengukuran saat  $t_k$  dapat diasimilasikan menggunakan persamaan (2.45) dengan  $\mathbf{K}_k$  adalah Kalman *gain* berdasarkan persamaan (2.51). Perlu diperhatikan bahwa diperlukan nilai  $\hat{\mathbf{x}}_k^-$  dan  $\mathbf{P}_k^-$ , untuk mendapatkan nilai estimasi *state* tersebut. Propagasi dari estimasi  $\hat{\mathbf{x}}_k$ ,  $\hat{\mathbf{x}}_{k+1}^-$  didapatkan melalui matriks transisi. Karena  $w_k$  adalah *zero mean* dan tidak berkorelasi dengan vektor-vektor  $w_k$  yang lalu, maka  $w_k$  dapat diabaikan, sehingga:

$$\hat{\mathbf{x}}_{k+1}^- = \boldsymbol{\phi}_k \hat{\mathbf{x}}_k \quad (2.55)$$

Matriks kovarian eror yang berhubungan dengan  $\mathbf{x}_{k+1}^-$  didapatkan dengan pertama membentuk persamaan yang mengekspresikan eror sebelumnya:

$$\begin{aligned} \mathbf{e}_{k+1}^- &= \mathbf{x}_{k+1} - \hat{\mathbf{x}}_{k+1}^- = (\boldsymbol{\phi}_k \mathbf{x}_k + \mathbf{w}_k) - \boldsymbol{\phi}_k \hat{\mathbf{x}}_k \\ &= \boldsymbol{\phi}_k \mathbf{e}_k + \mathbf{w}_k \end{aligned} \quad (2.56)$$

$w_k$  dan  $\mathbf{e}_k$  memiliki *zero crosscorrelation*, dikarenakan  $w_k$  adalah *noise* proses untuk langkah selanjutnya dari  $t_k$ . Sehingga, persamaan untuk propagasi matriks kovarian eror,  $\mathbf{P}_{k+1}^-$  adalah

$$\mathbf{P}_{k+1}^- = E[\mathbf{e}_{k+1}^- \mathbf{e}_{k+1}^{-T}] \quad (2.57)$$

Dengan menyubtitusikan persamaan (2.56) diperoleh

$$\mathbf{P}_{k+1}^- = E[(\boldsymbol{\phi}_k \mathbf{e}_k + \mathbf{w}_k)(\boldsymbol{\phi}_k \mathbf{e}_k + \mathbf{w}_k)^T] \quad (2.58)$$

Maka

$$\mathbf{P}_{k+1}^- = \boldsymbol{\phi}_k \mathbf{P}_k \boldsymbol{\phi}_k^T + \mathbf{Q}_k \quad (2.59)$$

Persamaan (2.45), (2.51), (2.54), (2.55), dan (2.59) merupakan persamaan yang digunakan pada algoritma rekursif Kalman filter. Pada saat simulasi, *loop* akan berhenti ketika simulasi dihentikan atau ketika data terakhir telah diproses, namun dalam implementasi, *loop* Filter Kalman dihentikan dengan *software* atau perintah pada INS.

Pada poin ini dirangkum proses perhitungan rekursif dari Kalman filter dengan mengikuti algoritma berikut:

0. Inisialisasi, saat  $k = 0$ , nilai dari state estimasi  $\hat{\mathbf{x}}_0^-$  dan matriks kovarian *error*  $\mathbf{P}_0^-$  adalah nilai tertentu.
1. Hitung nilai Kalman *gain*  $\mathbf{K}_k$  menggunakan persamaan (2.51)
2. Hitung matriks kovarian *error*  $\mathbf{P}_k$  dengan menggunakan persamaan (2.54)
3. *Update* estimasi  $\hat{\mathbf{x}}_k$  dengan nilai pengukuran  $\mathbf{z}_k$  dengan menggunakan persamaan (2.45)
4. Menyelesaikan perhitungan untuk prediksi *state* ke  $k + 1$  dengan menggunakan persamaan (2.55)
5. Menghitung propagasi matriks kovarian *error*  $\mathbf{P}_{k+1}^-$  menggunakan persamaan (2.59)
6. Untuk  $k = k + 1$ , kembali ke proses nomor 1.

*[Halaman ini sengaja dikosongkan]*

## BAB 3

### PERANCANGAN SISTEM

Pada bab ini akan diuraikan mengenai tahapan perancangan sistem dari Tugas Akhir ini. Seperti yang telah dijelaskan pada bab 2, perancangan sistem terbagi menjadi beberapa tahapan yang terdiri dari perancangan sistem untuk tahap *coarse alignment*, *fine alignment*, dan saat bergerak. Pada bab ini juga akan dijelaskan perancangan filter kalman untuk tahap *fine alignment* dan saat bergerak. Namun, sebelum mengurai tahapan-tahapan tersebut terlebih dahulu dilakukan identifikasi permasalahan. Setelah masalah telah dipahami maka selanjutnya menguraikan tahap perancangan sistem.

#### 3.1 Identifikasi Permasalahan

Keperluan akan sistem navigasi yang presisi semakin meningkat setiap tahunnya. Hal tersebut salah satunya disebabkan oleh perkembangan teknologi pada sistem *autonomous* yang membutuhkan pengetahuan mengenai posisi atau arah perpindahan dari wahana. Pengetahuan tersebut akan digunakan oleh wahana *autonomous* untuk kontrol, *tracking* dan *guidance*.

Sistem navigasi tidak hanya dibutuhkan untuk sistem (wahana) *autonomous*, namun kendaraan transportasi seperti pesawat terbang dan kapal laut membutuhkan panduan (*guideance*) lintasan yang akan dilewatinya agar mencapai tujuan dengan selamat. Panduan lintasan tersebut diperoleh dari informasi yang disediakan oleh sistem navigasi berupa posisi atau arah pergerakan dari kendaraan tersebut.

Sistem navigasi yang dapat digunakan untuk kendaraan di atas dapat menggunakan *Inertial Navigation System (INS)* keunggulan dari INS adalah kemampuan sensor *accelero* dan *gyro* dalam mengukur percepatan dan kecepatan sudut dengan *bandwidth* lebih dari 200Hz, di mana data *rate* yang cepat dari sebuah sensor sangat dibutuhkan oleh pesawat terbang Selain itu, keunggulan dari INS ukurannya yang relatif kecil. Namun, dari beberapa keunggulan tersebut, INS memiliki beberapa sumber eror yang dapat mengurangi presisi hasil navigasi yang akan terakumulasi seiring bertambahnya waktu. Integrasi INS\GPS digunakan untuk mengatasi eror INS yang meningkat seiring bertambahnya waktu

kemudian dengan menggunakan Filter Kalman untuk mengestimasi eror pada INS tersebut.

### 3.2 Model dan Parameter Sistem

Pertama-tama dalam perancangan sistem, hal yang dibutuhkan adalah model dari sistem yang akan kita gunakan dan beberapa parameter yang diperlukan. Model sistem dan *state* yang digunakan telah diuraikan pada bagian 2.5 yaitu model eror dari INS. Pada Tugas Akhir ini, Alat ukur yang digunakan bersifat diskrit dengan frekuensi 2000 Hz, sehingga model eror perlu didiskritisasi sebagai berikut:

$$\phi = I + F\tau_i \quad (3.1)$$

Di mana  $I$  merupakan matriks identitas dan  $\tau_i$  adalah *time sampling* atau dapat dinyatakan dengan persamaan berikut:

$$\tau_i = \frac{1}{freq} \quad (3.2)$$

Pada Tugas Akhir ini *freq* atau frekuensi yang digunakan ialah sesuai dengan alat ukur yang dipakai yaitu 2000 Hz. Sedangkan untuk matriks  $F$  dapat dilihat pada persamaan (2.14)

Berikutnya, menentukan parameter sistem. Parameter tersebut dapat ditentukan pada proses *alignment* atau didapatkan dari spesifikasi peralatan yang digunakan pada Tugas Akhir ini. Parameter sensor *accelero* dan *gyroscope* yang akan digunakan untuk menentukan *power spectran density* (PSD) dapat dilihat pada tabel Tabel 3.1.

**Tabel 3.1** Parameter sensor STIM300

Parameter	Nilai	Unit	Simbol
Bias <i>instability gyro</i>	0.3	°/h	$\sigma_{bgd}$
<i>Angular Random Walk</i>	0.15	°/√hr	$\sigma_{rg}$
Bias <i>instability accelero</i>	0.05	mg	$\sigma_{bad}$
<i>Velocity Random Walk</i>	0.07	m/s/√hr	$\sigma_{ra}$

### 3.3 Perancangan Pengolahan Sinyal Sensor

Perancangan sistem navigasi berarti merancang suatu sistem yang mengolah sinyal yang didapatkan oleh sensor kemudian diolah menjadi

hasil navigasi yang diinginkan seperti posisi, kecepatan dan sikap sudut (*attitude*). Pengolahan tersebut terjadi di prosesor.

Sistem navigasi yang akan dirancang pada Tugas Akhir ini sebelum digunakan untuk bergerak, terlebih dahulu dilakukan *alignment*. Oleh karena itu akan dirancang tahap *alignment*. Pada bab 2 telah dijelaskan mengenai tahap *alignment*.

### **3.3.1 Inisialisasi Kondisi Awal**

Tahap inisialisasi hasil navigasi perlu dilakukan di awal, sebelum wahana beroperasi. Untuk inisialisasi posisi dapat menggunakan bantuan GPS atau menggunakan hasil dari *fine alignment*. Pada Tugas Akhir ini, inisialisasi posisi menggunakan hasil dari estimasi *fine alignment* dengan bantuan posisi dari GPS. Pada inisialisasi kecepatan, karena wahana berada dalam keadaan diam saat proses *alignment*, maka inisialisasi dari kecepatan adalah 0.

Sedangkan inisialisasi *attitude* menggunakan hasil dari *coarse alignment* yang dikoreksi lagi pada tahap *fine alignment* agar hasil dari estimasi *attitude* lebih presisi.

### **3.3.2 Perancangan Proses Alignment**

Seperti yang telah dijelaskan pada dasar teori bahwa sistem navigasi berbasis INS membutuhkan hasil navigasi sebelumnya pada setiap iterasinya, di mana hasil navigasi tersebut akan diintegrasikan dengan pengukuran sekarang untuk mendapatkan hasil navigasi sekarang. Oleh karena itu, diperlukan proses inisialisasi hasil navigasi sebelumnya sebelum sistem navigasi berbasis INS digunakan.

Untuk inisialisasi posisi dan kecepatan membutuhkan informasi eksternal, dalam Tugas Akhir ini menggunakan GPS sebagai sumber informasi eksternal. Sedangkan untuk inisialisasi sikap sudut (*attitude*) dapat dilakukan dengan proses *alignment*. Di mana pada proses *alignment* terdiri dari dua tahap yaitu *coarse* dan *fine alignment*. Perlu dicatat bahwa proses *alignment* pada Tugas Akhir ini dilakukan ketika wahana dalam keadaan diam.

#### **3.3.2.1 Coarse Alignment**

Tujuan dari *coarse alignment* adalah menentukan *attitude* awal dari wahana (sistem). Dengan menggunakan gaya spesifik yang diukur oleh sensor *accelero* pada keadaan diam atau tidak ada percepatan

melainkan percepatan yang merupakan rekasi dari gravitasi. dengan menggunakan reaksi dari gravitasi tersebut yang diukur oleh sensor *accelero*, maka *pitch* dan *roll* dari wahana dapat ditentukan. Teknik ini disebut proses *leveling*. Di mana dengan menurunkan persamaan berikut:

$$\mathbf{f}_{ib}^b = \mathbf{C}_n^b \mathbf{g}_b^n(L_b, h_b) \quad (3.3)$$

$$\begin{bmatrix} f_{ib,x}^b \\ f_{ib,y}^b \\ f_{ib,z}^b \end{bmatrix} = \begin{bmatrix} \sin \theta_{nb} \\ -\cos \theta_{nb} \sin \phi_{nb} \\ -\cos \theta_{nb} \cos \phi_{nb} \end{bmatrix} \mathbf{g}_{b,D}^n(L_b, h_b)$$

Didapatkan

$$\theta_{nb} = \arctan \left( \frac{f_{ib,x}^b}{\sqrt{f_{ib,y}^b{}^2 + f_{ib,z}^b{}^2}} \right), \quad (3.4)$$

$$\phi_{nb} = \arctan_2(-f_{ib,y}^b, -f_{ib,z}^b)$$

Pada keadaan dengan kondisi benar-benar diam (*absolutely stationary*), eror inialisasi *attitude* hanya dipengaruhi oleh eror dari sensor *accelero*. Namun adanya gangguan gerakan seperti getaran, efek angin, dan aktivitas manusia dapat mempengaruhi proses *leveling*. Eror tersebut dapat diatasi dengan cara merata-ratakan hasil pengukuran sensor *accelero* pada setiap iterasi.

Pada proses *leveling*, *attitude* yang diperoleh hanya *roll* ( $\phi$ ) dan *pitch* ( $\theta$ ). Proses inialisasi sudut *heading* dapat dilakukan dengan teknik yang disebut *gyrocompassing*. Prinsip dari *gyrocompassing* telah diuraikan pada bagian 2.3.2, di mana pada keadaan diam (*stationary*), sensor *gyro* hanya mengukur rotasi kecepatan putar bumi yang berada pada sumbu *z* dari *frame* ECEF. Dengan kecepatan putar bumi yang diukur langsung oleh sensor *gyro* tersebut, sehingga dengan menggunakan Teknik *direct gyrocompassing*, sudut *heading* dapat diperoleh dengan menurunkan persamaan berikut:



$$\boldsymbol{\omega}_{ib}^b = \mathbf{C}_n^b \mathbf{C}_e^n(L_b, \lambda_b) \begin{bmatrix} 0 \\ 0 \\ \boldsymbol{\omega}_{ie} \end{bmatrix} \quad (3.5)$$

$$\begin{bmatrix} \cos L_b \boldsymbol{\omega}_{ie} \\ 0 \\ -\sin L_b \boldsymbol{\omega}_{ie} \end{bmatrix} = \mathbf{C}_b^n \boldsymbol{\omega}_{ib}^b$$

Sehingga didapatkan persamaan berikut

$$\psi_{nb} = \arctan_2(\sin \psi_{nb}, \cos \psi_{nb}) \quad (3.6)$$

Di mana

$$\sin \psi_{nb} = -\omega_{ib,y}^b \cos \phi_{nb} + \omega_{ib,z}^b \sin \phi_{nb}$$

$$\cos \psi_{nb} = \omega_{ib,x}^b \cos \theta_{nb} + \omega_{ib,y}^b \sin \phi_{nb} \sin \theta_{nb} + \omega_{ib,z}^b \cos \phi_{nb} \sin \theta_{nb}$$

Proses *coarse alignment* menggunakan persamaan (3.9) dan (3.10) karena pada proses tersebut melakukan perhitungan rata-rata dari hasil pengukuran sensor *accelero* untuk *roll* dan *pitch*, sedangkan untuk *heading*, hasil pengukuran sensor *gyro* perlu dirata-ratakan pula untuk mengurangi gangguan berupa getaran mesin wahana atau aktifitas manusia. Perhitungan rata-rata menggunakan mean rekursif yang dapat dinyatakan pada persamaan berikut:

$$\bar{\boldsymbol{\omega}}_{ib,k}^b = \frac{(\boldsymbol{\omega}_{ib,k}^b + \bar{\boldsymbol{\omega}}_{ib,k-1}^b(k-1))}{k} \quad (3.7)$$

dan

$$\bar{\mathbf{f}}_{ib,k}^b = \frac{(\mathbf{f}_{ib,k}^b + \bar{\mathbf{f}}_{ib,k-1}^b(k-1))}{k} \quad (3.8)$$

sehingga persamaan (3.4) dan (3.6) menjadi

$$\bar{\theta}_{nb,k} = \arctan \left( \frac{\bar{f}_{ib,x,k}^b}{\sqrt{\bar{f}_{ib,y,k}^b{}^2 + \bar{f}_{ib,z,k}^b{}^2}} \right) \quad (3.9)$$

$$\bar{\phi}_{nb,k} = \arctan_2(-\bar{f}_{ib,y,k}^b, -\bar{f}_{ib,z,k}^b)$$

$$\bar{\psi}_{nb,k} = \arctan_2(\sin \psi_{nb,k}, \cos \psi_{nb,k}) \quad (3.10)$$

Di mana

$$\sin \psi_{nb,k} = -\bar{\omega}_{ib,y,k}^b \cos \phi_{nb,k} + \bar{\omega}_{ib,z,k}^b \sin \phi_{nb,k}$$

$$\begin{aligned} \cos \psi_{nb,k} = & \bar{\omega}_{ib,x,k}^b \cos \theta_{nb,k} + \bar{\omega}_{ib,y,k}^b \sin \phi_{nb,k} \sin \theta_{nb,k} \\ & + \bar{\omega}_{ib,z,k}^b \cos \phi_{nb,k} \sin \theta_{nb,k} \end{aligned}$$

$k$  merupakan variabel iterasi yang menyatakan iterasi ke berapa dan memiliki nilai 1 sampai sebanyak jumlah data pengukuran sensor IMU.

### 3.3.2.2 Fine Alignment

Seperti yang telah diuraikan pada bab 2, bahwa *fine alignment* bertujuan untuk mengurangi eror yang masih terkandung pada hasil *coarse alignment*. *Fine alignment* yang digunakan pada Tugas Akhir ini adalah *quasy stationary*, di mana pada teknik ini wahana diasumsikan berada pada keadaan diam. Sehingga perpindahan posisi dan kecepatan yang dialami oleh INS merupakan getaran mekanik yang merupakan eror.

*Fine alignment* dengan teknik *quasy stationary* dapat menggunakan proses mekanisasi yang digunakan pada saat bergerak jika *heading* diketahui dalam beberapa derajat. Pada kondisi ini pula *state* dan model sistem filter Kalman yang digunakan pada saat bergerak dapat digunakan untuk *fine alignment* dengan pengukuran sebagai berikut:

$$\delta \mathbf{z}_k^n = \begin{bmatrix} \hat{\mathbf{P}}_{b,k} - \mathbf{P}_{b,k}^{\text{GPS}} \\ \hat{\mathbf{v}}_{eb,k-1} - \mathbf{0}_3 \end{bmatrix} \quad (3.11)$$

Di mana

$\mathbf{P}_{b,k}^{\text{GPS}}$  : Posisi GPS saat  $k_0$  atau saat inisialisasi

$\hat{\mathbf{P}}_{b,k}$  : Posisi hasil proses mekanisasi  
 $\hat{\mathbf{v}}_{eb,k-1}$  : Kecepatan hasil estimasi filter Kalman sebelumnya  
 Serta matriks  $H$  yang digunakan adalah

$$H = \begin{bmatrix} \mathbf{0}_3 & \mathbf{0}_3 & -\mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & -\mathbf{I}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \end{bmatrix} \quad (3.12)$$

Matriks  $H$  berukuran  $6 \times 15$  dengan  $\mathbf{I}_3$  merupakan matriks identitas  $3 \times 3$  dan  $\mathbf{0}_3$  merupakan matriks yang semua elementnya bernilai 0 berukuran  $3 \times 3$ .

Karena mekanisasi membutuhkan hasil navigasi sebelumnya, maka untuk kondisi awal, pada *fine alignment* diperlukan inisialisasi posisi, kecepatan dan *attitude*. Pada kondisi awal posisi sebelumnya didapatkan dari GPS. Sedangkan kecepatan sebelumnya diset 0. Sedangkan untuk *attitude*, kondisi awal *roll* dan *pitch* menggunakan hasil akhir dari *coarse alignment* dan untuk *heading* menggunakan sumber eksternal yaitu *heading* dari CNS5000.

### 3.3.3 Perancangan Proses Mekanisasi

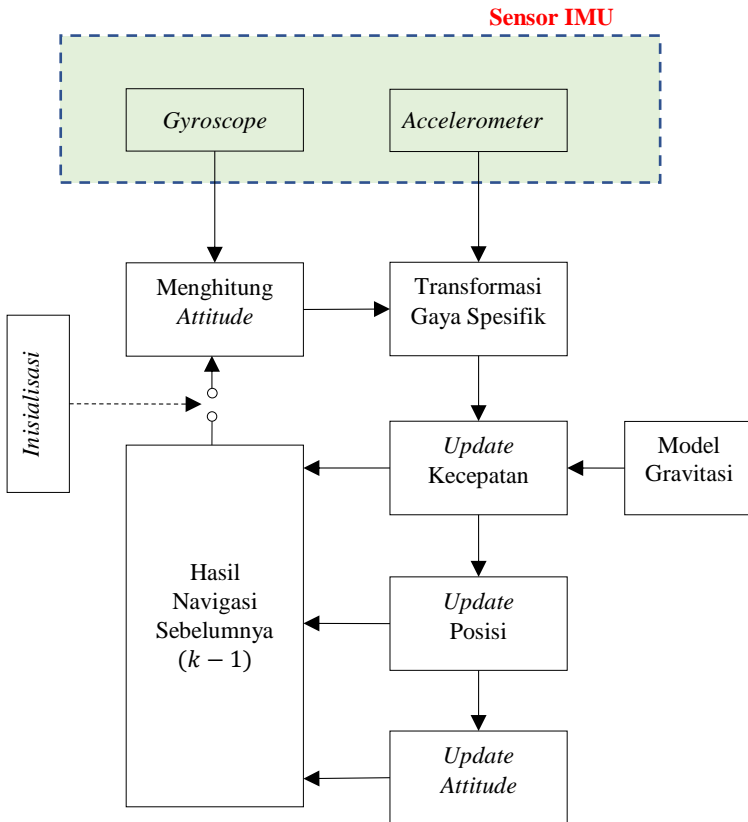
Pergerakan suatu objek dalam dalam tiga dimensi membutuhkan enam derajat kebebasan yang terdiri dari tiga derajat kebebasan yang linear dan tiga derajat kebebasan yang angular. Oleh karena itu dibutuhkan tiga *gyro* sensor untuk tiga sumbu dan tiga *accelero* sensor untuk tiga sumbu agar dapat mengukur pergerakan dalam tiga dimensi tersebut. kedua sensor IMU tersebut mengukur gaya spesifik dan kecepatan sudut yang masing-masing  $f_{ib}^b$  dan  $\omega_{ib}^b$  pada *body* dari IMU dengan mengacu pada *frame* inersia (ECI) dan menggunakan sumbu dari *body* IMU.

Perlu diketahui bahwa proses mekanisasi merupakan proses pengolahan keluaran sensor IMU yaitu  $f_{ib}^b$  dan  $\omega_{ib}^b$  menjadi hasil navigasi yang berupa posisi, kecepatan dan *attitude* dengan menggunakan sumbu pada *frame* acuan tertentu sesuai kebutuhan pengguna. Pada Tugas Akhir ini proses mekanisasi dirancang untuk mendapatkan hasil navigasi yang menggunakan sumbu dari *local navigation frame*. Tabel

Parancangan dari proses mekanisasi terdiri dari beberapa tahapan. Setiap tahapan akan dijelaskan proses perancangannya. Seperti yang diuraikan pada bab 2, Tahapan tersebut antara lain ialah:

1. Menghitung *attitude*
2. Mentransformasi *resolving axis* gaya spesifik  $f_{ib}^b$  dari *body frame* menjadi gaya spesifik  $f_{ib}^n$  pada *local navigation frame*.
3. Menghitung kecepatan dari hasil transformasi gaya spesifik  $f_{ib}^b$  menjadi  $f_{ib}^n$  dengan memperhitungkan model gravitasi.
4. Menghitung posisi dari kecepatan yang telah didapatkan sebelumnya.

Untuk dapat memahami gambaran proses tersebut perhatikan Gambar 3.1.



**Gambar 3.1** Skema proses mekanisasi pada *local navigation frame*

### 3.3.3.1 Menghitung Attitude

Pada bab 2 telah diuraikan bahwa hal yang pertama kali dilakukan adalah pada proses mekanisasi yaitu menghitung *attitude*, hal ini dikarenakan dibutuhkannya transformasi sumbu dari *body frame* menjadi sumbu dari *local navigation frame*. *Attitude* dibutuhkan pada matriks tranformasi koordinat, sehingga *attitude* direpresentasikan menjadi matriks transformasi koordinat  $C_b^n$ . Penentuan *attitude* sangat mempengaruhi keakuratan hasil navigasi apalagi bila digunakan untuk wahana yang gerakannya relatif cepat seperti UAV. Perubahan *attitude* yang terjadi pada interval proses pengolahan sinyal dapat mempengaruhi hasil  $C_b^n$  sehingga, agar menghasilkan  $C_b^n$  yang lebih presisi maka dilakukan rata-rata dari nilai  $C_b^n$  yaitu nilai saat awal interval dan saat akhir interval proses mekanisasi, di mana dapat ditulis sebagai berikut:

$$C_{b,k}^n = C_{b,k-1}^n C_b^{b-} + \frac{(\Omega_{ie,k-1}^n + \Omega_{en,k-1}^n) C_{b,k-1}^n \tau_i}{2} \quad (3.13)$$

$$C_b^{b-} = I_3 + \left( \frac{1 - \cos|\alpha_{ib}^b|}{|\alpha_{ib}^b|^2} |\alpha_{ib}^b| \wedge \right) + \frac{1}{|\alpha_{ib}^b|^2} \left( 1 - \frac{\sin|\alpha_{ib}^b|}{|\alpha_{ib}^b|} \right) |\alpha_{ib}^b| \wedge^2 \quad (3.14)$$

Di mana

$$\alpha_{ib}^b = \omega_{ib}^b \times \tau_i \quad (3.15)$$

Dan  $I_3$  merupakan matriks identitas dengan ukuran  $3 \times 3$

**Tabel 3.2** Parameter WGS84

Parameter	Nilai	Keterangan	Unit
$R_0$	6378137	Jari-jari equatorial	Meter [m]
$R_p$	6356752.31425	Jari jari polar	Meter [m]
$e$	0.0818191908425	<i>eccentricity</i>	Konstanta

$f$	$\frac{1}{298.257223563}$	<i>flattening</i>	Konstanta
$\mu$	3.986004418E14	<i>Earth Gravitational Constant</i>	$m^3/s^2$
$\omega_{ie}$	$7.292115 \times 10^{-5}$	<i>Kecepatan putar bumu</i>	$rad/s$

### 3.3.3.2 Transformasi Gaya Spesifik

Mentrasformasikan gaya spesifik yang diukur oleh sensor *accelero* dibutuhkan untuk langkah selanjutnya yaitu menghitung kecepatan dan posisi. Setelah  $C_{b,k}^n$  diperoleh maka langkah selanjutnya yaitu mentransformasikan gaya spesifik  $f_{ib}^b$  menjadi  $f_{ib}^n$  pada saat  $k$  yang dirumuskan sebagai berikut:

$$f_{ib,k}^n = C_{b,k}^n f_{ib,k}^b \quad (3.16)$$

### 3.3.3.3 Update Kecepatan

Tahap selanjutnya setelah menentukan  $f_{ib}^n$  pada saat  $k$  adalah mengintegrasikan  $f_{ib,k}^n$  sehingga didapatkan kecepatan pada saat  $k$ . Pada tahap menghitung kecepatan dilibatkan model gravitasi yang telah diuraikan pada bagian 2.5. persamaan untuk mendapatkan kecepatan  $v_{eb,k}^n$  dapat menggunakan persamaan (2.37), namun agar hasil yang diperoleh maksimal untuk wahana yang bergerak relatif cepat maka memperbarui  $\omega_{en}^n$  perlu dilakukan setelah menghitung  $v_{eb,k}^n$  menggunakan persamaan (2.37) kemudian menghitung  $v_{eb,k}^n$  yang baru. Berikut tahapan menghitung  $v_{eb,k}^n$

1. Menghitung  $v_{eb}^{n+}$  menggunakan persamaan (2.37)
2. Menghitung  $\Omega_{en,k-1}^{n+}$  menggunakan hasil dari  $v_{eb}^{n+}$  serta  $R_E$  dan  $R_N$  dari  $L_b$  dan  $\lambda_b$  sebelumnya dengan menggunakan persamaan (2.31) dan (2.32(2.32)).
3. Setelah mendapatkan  $v_{eb}^{n+}$  dan  $\Omega_{en,k-1}^{n+}$  maka langkah selanjutnya yaitu menghitung  $v_{eb,k}^n$  dengan menggunakan persamaan berikut:

$$\mathbf{v}_{eb,k}^n = \mathbf{v}_{eb,k-1}^n + \left\{ \begin{array}{l} \mathbf{f}_{ib,k}^n + \mathbf{g}_b^n(L_{b,k-1}, h_{b,k-1}) \\ - \left( 0.5(\boldsymbol{\Omega}_{en,k-1}^n + 2\boldsymbol{\Omega}_{ie,k-1}^n)\mathbf{v}_{eb,k-1}^n \right) \\ + 0.5(\boldsymbol{\Omega}_{en,k-1}^{n+} + 2\boldsymbol{\Omega}_{ie,k-1}^{n+})\mathbf{v}_{eb}^{n+} \end{array} \right\} \tau_i \quad (3.17)$$

### 3.3.3.4 Update Posisi

Penentuan posisi dilakukan dengan mengintegrasikan hasil dari kecepatan  $\mathbf{v}_{eb,k}^n$ . Karena hasil navigasi posisi pada Tugas Akhir ini menggunakan *resolving axis local navigation frame* yang direpresentasikan dengan *latitude* ( $L$ ), *longitude* ( $\lambda$ ) dan *height* ( $h$ ) maka penentuan posisi dapat menggunakan persamaan (2.38) atau menggunakan pendekatan diskrit sebagai berikut:

$$\begin{aligned} h_{b,k} &= h_{b,k-1} - \left( \mathbf{v}_{eb,D,k-1}^n + \mathbf{v}_{eb,D,k}^n \right) \frac{\tau_i}{2} \\ L_{b,k} &= L_{b,k-1} + \left( \frac{v_{eb,k-1,N}^n}{R_N(L_{b,k-1}) + h_{b,k-1}} + \frac{v_{eb,k,N}^n}{R_N(L_{b,k}) + h_{b,k}} \right) \frac{\tau_i}{2} \\ \lambda_{b,k} &= \lambda_{b,k-1} + \left( \frac{v_{eb,k,E}^n}{(R_E(L_{b,k-1}) + h_{b,k-1}) \cos L_{b,k-1}} + \frac{v_{eb,k,E}^n}{(R_E(L_{b,k}) + h_{b,k}) \cos L_{b,k}} \right) \frac{\tau_i}{2} \end{aligned} \quad (3.18)$$

Di mana  $R_E$  dan  $R_N$  dirumuskan dengan persamaan berikut:

$$\begin{aligned} R_N &= \frac{R_0(1 - e^2)}{\sqrt[2]{(1 - (e \sin L_{b,k})^2)^3}} \\ R_E &= \frac{R_0}{\sqrt{1 - (e \sin L_{b,k})^2}} \end{aligned} \quad (3.19)$$

### 3.3.3.5 Update Attitude

Seperti yang telah dibahas sebelumnya, bahwa untuk menentukan *attitude* menggunakan rata-rata dari *attitude* pada saat awal interval

mekanisasi dan *attitude* pada saat akhir mekanisasi. Pada tahap awal interval mekanisasi menggunakan persamaan (3.13), namun pada tahap akhir mekanisasi mengikuti langkah-langkah berikut:

1. Menghitung  $\Omega_{en}^n$  dengan persamaan (2.31) dan menggunakan  $v_{eb,k}^n$  dari persamaan (2.18) serta  $R_E$  dan  $R_N$  dari  $L_{b,k}$  dan  $\lambda_{b,k}$  pada saat sekarang
2. Menghitung  $C_{b+}^{b-}$  dengan persamaan berikut:

$$C_{b+}^{b-} = I_3 + \frac{\sin|\alpha_{ib}^b|}{|\alpha_{ib}^b|} [\alpha_{ib}^b \wedge] + \frac{1 - \cos|\alpha_{ib}^b|}{|\alpha_{ib}^b|^2} [\alpha_{ib}^b \wedge]^2 \quad (3.20)$$

3. Langkah terakhir yaitu memperbarui  $C_b^n$  dengan menggunakan persamaan berikut:

$$C_{b,k}^n = \left[ I_3 - \begin{pmatrix} \Omega_{ie,k-1}^e \\ +0.5\Omega_{en,k-1}^n \\ +0.5\Omega_{en,k}^n \end{pmatrix} \tau_i \right] C_{b,k-1}^n C_{b+}^{b-} \quad (3.21)$$

Proses mekanisasi secara lengkap dalam bentuk *script* MATLAB dapat dilihat pada lampiran.

### 3.3.4 Perancangan Filter Kalman

Pada bab 2 telah diuraikan konsep mengenai Filter Kalman bahwa dalam penggunaan Filter Kalman dibutuhkan model sistem dan model pengukuran. Oleh karena itu, perancangan Filter Kalman dilakukan dengan menentukan model yang direpresentasikan dalam bentuk *state space*. Berikut model sistem dan model pengukuran yang digunakan pada Filter Kalman pada Tugas Akhir ini:

$$\begin{aligned} \mathbf{x}_{k+1} &= \boldsymbol{\phi}_k \mathbf{x}_k + \mathbf{w}_k \\ \mathbf{y}_k &= \mathbf{H} \mathbf{x}_k + \mathbf{v}_k \end{aligned} \quad (3.22)$$

Di mana  $\boldsymbol{\phi}_k$  merupakan matriks transisi sistem dalam waktu diskrit yang telah ditulis pada persamaan (3.1).  $\mathbf{H}$  merupakan matriks pengukuran yang dapat dilihat pada persamaan (3.12). Pada Tugas Akhir ini, seperti yang telah disinggung pada bagian 3.3.2 bahwa data pengukuran merupakan error atau selisih dari posisi dan kecepatan INS dan GPS yang dapat dinyatakan dalam persamaan berikut:



$$\delta \mathbf{z}_k^n = \begin{bmatrix} \hat{\mathbf{p}}_{b,k} - \mathbf{p}_{b,k}^{\text{GPS}} \\ \hat{\mathbf{v}}_{eb,k} - \mathbf{v}_{eb,k}^{\text{GPS}} \end{bmatrix} \quad (3.23)$$

Sedangkan,  $\mathbf{w}_k$  dan  $\mathbf{v}_k$  masing-masing merupakan error acak pada sistem dan pengukuran yang memiliki matriks kovarian sebagai berikut:

$$E[\mathbf{w}_{k_1} \mathbf{w}_{k_2}^T] = \begin{cases} 0 & k_1 = k_2 \\ \mathbf{Q} & k_1 \neq k_2 \end{cases} \quad (3.24)$$

$$E[\mathbf{v}_{k_1} \mathbf{v}_{k_2}^T] = \begin{cases} 0 & k_1 = k_2 \\ \mathbf{R} & k_1 \neq k_2 \end{cases}$$

Di mana

$$\mathbf{Q} = \begin{bmatrix} S_{ra} & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & S_{rg} & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & S_{bad} & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & S_{bgd} \end{bmatrix}, \quad \mathbf{R} = \begin{bmatrix} \sigma_{P_b}^2 & \mathbf{0}_3 \\ \mathbf{0}_3 & \sigma_{v_{eb}}^2 \end{bmatrix} \quad (3.25)$$

Setelah menentukan model sistem dan pengukuran serta beberapa parameter yang akan digunakan pada Filter Kalman, Selanjutnya ialah merancang proses Filter Kalman. Proses dari Filter Kalman mengikuti algoritma yang telah dijabarkan pada bagian 2.7.3. Berikut tahapan pengolahan sinyal pada proses Filter Kalman:

0. Hitung  $\mathbf{Q}$  dan  $\mathbf{R}$  di awal,  
Sebelum proses iterasi Filter Kalman dimulai, agar tidak membebani proses penghitungan, matriks  $\mathbf{Q}$  dan  $\mathbf{R}$  ditentukan terlebih dahulu ketimbang ditentukan pada setiap iterasi.
1. Hitung matriks  $\boldsymbol{\phi}_k$  menggunakan persamaan (3.1),  
Pada persamaan (3.1), matriks  $\mathbf{F}$  membutuhkan hasil pengukuran dari IMU ( $\mathbf{f}_{ib,k}^b$  dan  $\boldsymbol{\omega}_{ib,k}^b$ ) dan beberapa hasil navigasi dari proses mekanisasi ( $L_{b,k}$ ,  $h_{b,k}$ ,  $\mathbf{v}_{eb,k}^n$  dan  $C_{b,k}^n$ ). Sehingga perhitungan  $\boldsymbol{\phi}_k$  harus dilakukan pada setiap iterasi,
2. Hitung propagasi matriks kovarian eror menggunakan persamaan (2.59),
3. Hitung Kalman *gain*  $\mathbf{K}_k$   
Setelah menentukan model sistem dan parameter sistem langkah selanjutnya ialah menghitung  $\mathbf{K}_k$  menggunakan persamaan (2.51),
4. Hitung selisih antara posisi dan kecepatan hasil mekanisasi INS dengan GPS,

5. Melakukan estimasi *state* menggunakan persamaan (2.45)  
Setelah data GPS tersedia dan telah diselisihkan dengan hasil mekanisasi yang kemudian menjadi pengukuran pada Filter Kalman, selanjutnya yaitu mengestimasi *state* eror  $\hat{\mathbf{x}}_k$ ,
6. Hitung matriks kovarian eror menggunakan persamaan (2.48)  
Pada implementasinya, menentukan matriks kovarian eror  $\mathbf{P}_k$  sebelum mengestimasi *state* dikarenakan menunggu tersedianya data pengukuran dari GPS.
7. Hitung prediksi *state*  $\hat{\mathbf{x}}_{k+1}^-$ ,  
Pada tahap ini, prediksi *state*  $\hat{\mathbf{x}}_{k+1}^- = 0$ . Hal tersebut dikarenakan pada Tugas Akhir ini menggunakan skema *close loop* sehingga prediksi terbaik dari *state* ialah tanpa eror,
8. Untuk  $k = k + 1$ , kembali ke langkah nomor 1.

## BAB 4 SIMULASI DAN ANALISA

Pada bab ini akan dibahas hasil simulasi dari sistem navigasi yang telah dirancang pada bab 3 sebelumnya. Hasil navigasi dari simulasi INS-GPS dengan Filter Kalman akan dibandingkan dengan hasil navigasi yang hanya menggunakan proses mekanisasi saja. Selain itu dilakukan juga analisa mengenai hasil simulasi tersebut. Namun, sebelum kedua hal tersebut, terlebih dahulu mempersiapkan data yang akan diujikan pada rancangan sistem Tugas Akhir ini.

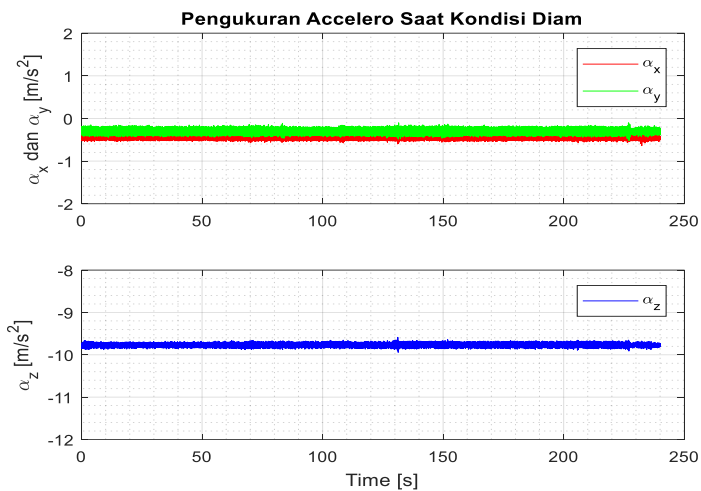
Data uji diambil dari pengukuran dengan menggunakan sensor STIM300 yang terdiri dari sensor *accelero* dan *gyro*. Pengukuran dilakukan di daerah kota Surabaya dengan menggunakan mobil yang telah terpasang sensor STIM300 dan GPS. Sebelum bergerak mobil akan diam selama 4 menit atau 240 detik untuk melakukan *alignment*. STIM300 memberikan data pengukuran dengan frekuensi 2000 Hz yang terdiri dari gaya spesifik  $f_{ib}^b$  dan kecepatan sudut  $\omega_{ib}^b$ . Sedangkan GPS memberikan data yang terdiri dari posisi dan kecepatan yang menggunakan koordinat *local navigation frame* dengan frekuensi 10 Hz. Data pengukuran dari STIM300 dapat dilihat pada Gambar 4.1-Gambar 4.4

Pada mobil dipasang juga sensor KVH-CNS5000 yang berguna sebagai data pembanding atau sebagai referensi. Sensor KVH-CNS5000 memberikan data yang terdiri dari posisi, kecepatan dan *attitude* yang menggunakan koordinat dari *local navigation frame*. Data pada sensor KVH-CNS5000 memiliki frekuensi 20 Hz. Pada Tugas Akhir ini sensor KVH-CNS5000 diasumsikan sebagai hasil navigasi sebenarnya.

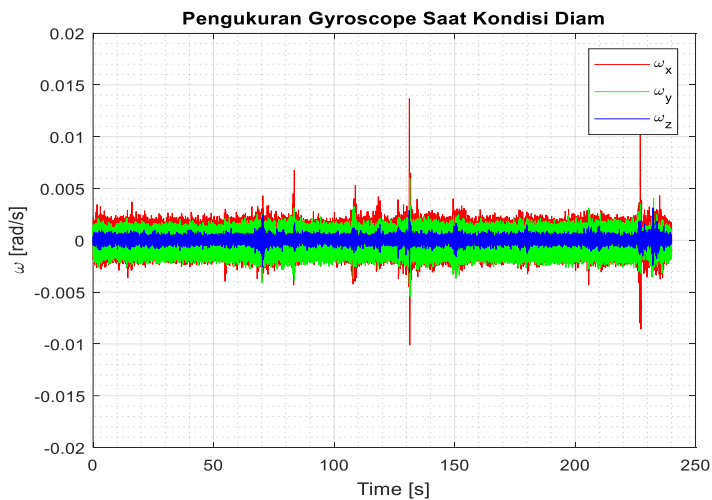
### 4.1 Hasil Simulasi Proses *Alignment*

Sesuai dengan tahapan pada rancangan sistem, sebelum wahana digunakan pada saat bergerak, terlebih dahulu dilakukan proses *alignment*. Proses *alignment* dilakukan selama 240 detik di mana pada durasi tersebut data yang berada pada 10 detik pertama akan diabaikan karena data pada saat tersebut, sensor *accelero* dan *gyroscope* masih berada pada keadaan transien yang menyebabkan data belum stabil sehingga memiliki eror yang relatif besar. Selanjutnya data pada 20 detik kemudian dialokasikan untuk proses *coarse alignment*. Setelah itu, durasi

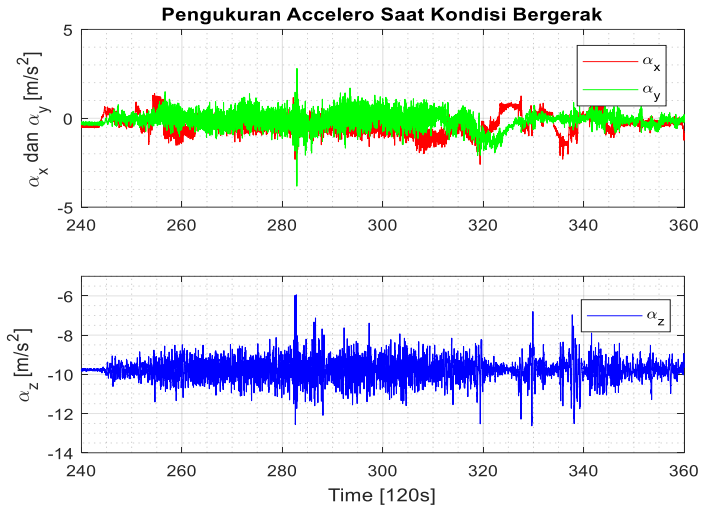
yang tersisa yaitu selama 210 detik, digunakan untuk proses *fine alignment*.



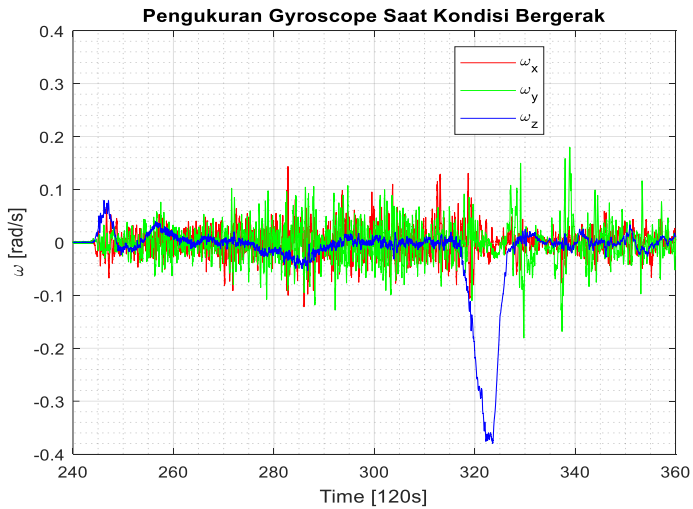
**Gambar 4.1** Data pengukuran *accelero* saat mobil diam.



**Gambar 4.2** Data pengukuran *gyro* saat mobil diam.



**Gambar 4.3** Data pengukuran *accelero* saat mobil dalam kondisi bergerak.



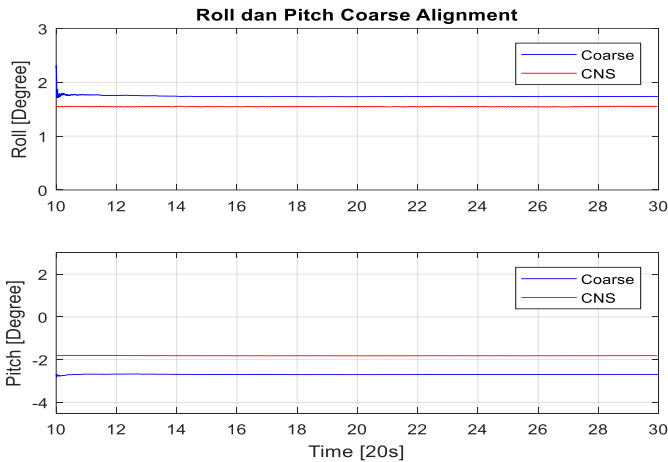
**Gambar 4.4** Data pengukuran *gyro* saat mobil dalam kondisi bergerak.

#### 4.1.1 Hasil Simulasi *Coarse Alignment*

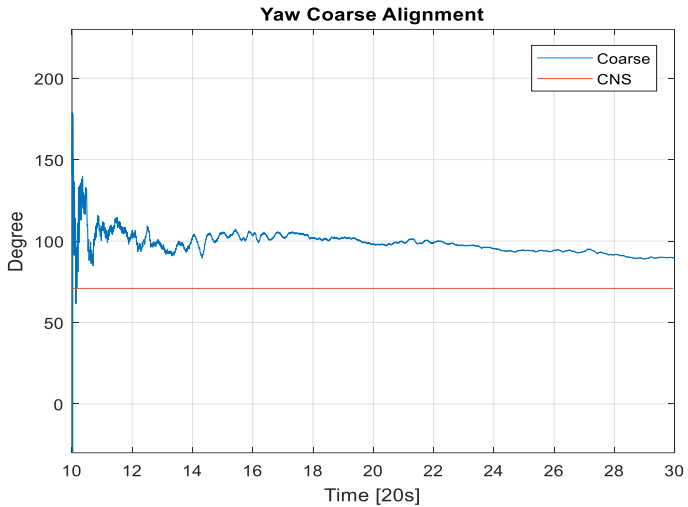
Pada *coarse alignment* penentuan *roll* dan *pitch* dilakukan terlebih dahulu sebelum menentukan *yaw* dari wahana. Hasil simulasi dari *coarse alignment* dapat dilihat pada Gambar 4.5 dan Gambar 4.6. Pada iterasi yang sama, setelah menentukan *roll* dan *pitch*, selanjutnya sudut *yaw* dapat ditentukan dengan hasil yang dapat dilihat pada Gambar 4.6. Secara sekilas hasil *coarse alignment* pada Gambar 4.5 masih terdapat selisih antara hasil estimasi *coarse alignment* dan sinyal sebenarnya (KVH-CNS5000). Secara numerik eror hasil dari *coarse alignment* dapat dilihat pada Tabel 4.1.

**Tabel 4.1** Parameter eror hasil *coarse alignment*

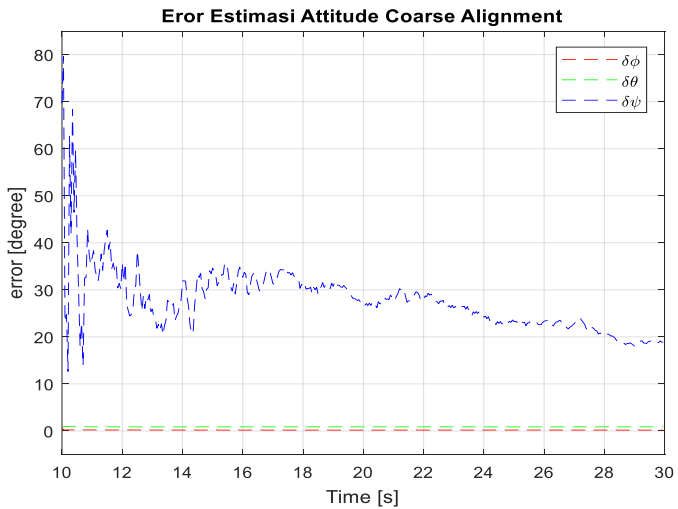
Eror Attitude (Estimasi-CNS)	$[\delta\phi \quad \delta\theta \quad \delta\psi]$ degree
Mean	[0.1942 0.8769 27.7649]
Max	[0.7180 0.9504 79.6794]
Min	[0.1779 0.8559 12.6239]
Standard Deviation	[0.0287 0.0097 6.9983]
RMSE	[0.1942 0.8769 27.7649]



**Gambar 4.5** Hasil estimasi *roll* dan *pitch* dari *coarse alignment*



**Gambar 4.6** Hasil Estimasi yaw dari coarse alignment



**Gambar 4.7** Error (estimasi-CNS) dari hasil Estimasi Attitude coarse alignment

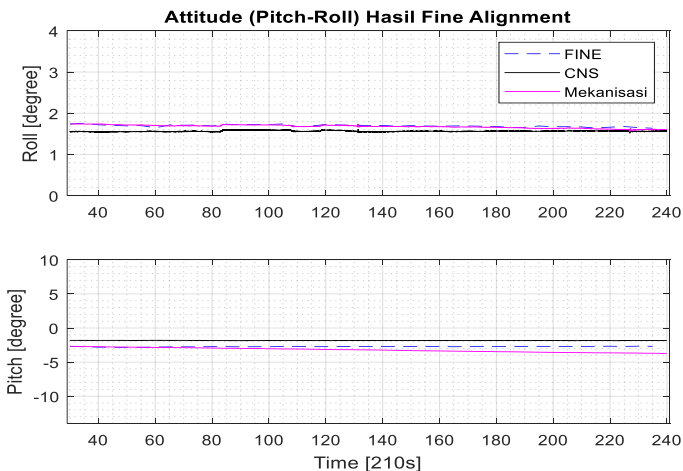
Dari Tabel 4.1 dan Gambar 4.7, hasil estimasi *roll* dan *pitch* relatif lebih kecil dengan nilai maksimal eror berada di bawah  $1^\circ$ . sedangkan untuk estimasi *yaw* eror masih relatif besar dengan nilai rata-rata lebih dari  $27^\circ$ . Hal ini dikarenakan pada proses *coarse alignment* tidak dilakukan koreksi dari sinyal sensor *accelero* dan *gyroscope* yang masih memiliki bias dan belum menggunakan Filter Kalman untuk meminimalkan gangguan yang berupa *white noise*.

#### 4.1.2 Hasil Simulasi *Fine Alignment*

Tahap selanjutnya setelah mengestimasi *attitude* awal pada *coarse alignment* adalah melakukan proses *fine alignment* yang akan mengestimasi kondisi awal dari wahana yang terdiri dari posisi, kecepatan dan *attitude* serta bias sensor *accelero* dan *gyroscope* yang akan digunakan untuk mengoreksi sinyal yang terukur oleh sensor.

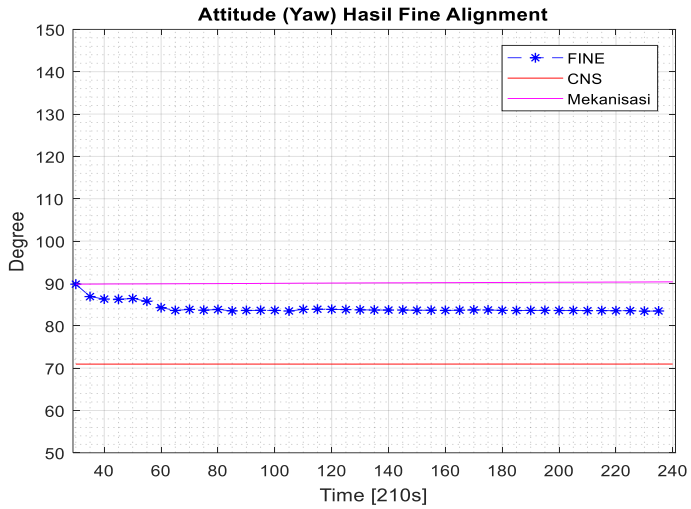
##### 4.1.2.1 Hasil Estimasi *Attitude Fine Alignment*

Pada *fine alignment* telah menggunakan Filter Kalman yang menggunakan model eror dari INS, sehingga dapat mengestimasi eror hasil navigasi serta dapat mengoreksi sinyal pengukuran dari sensor yang masih memiliki bias. Hasil *fine alignment* disajikan pada Gambar 4.8-Gambar 4.16.

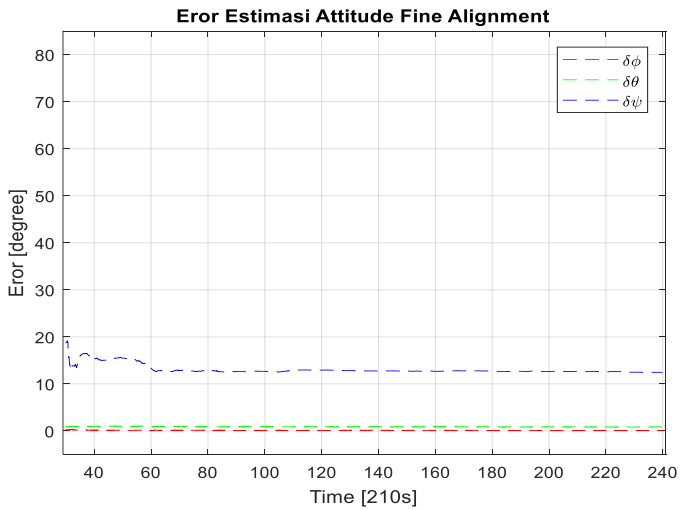


**Gambar 4.8** Hasil estimasi *roll* dan *pitch* dari *fine alignment*.

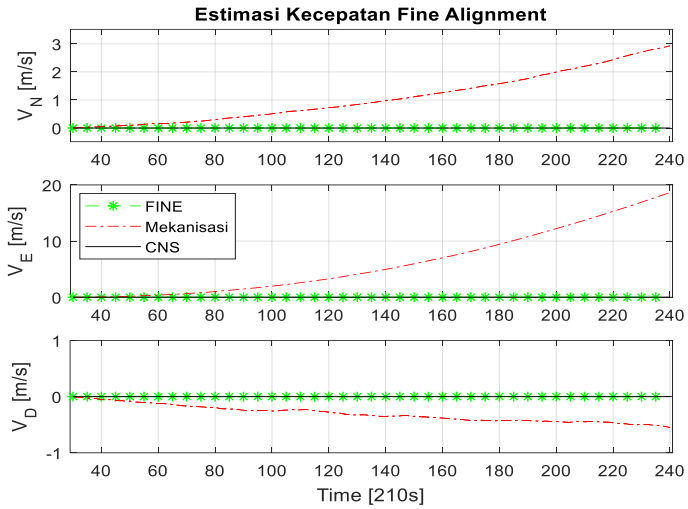




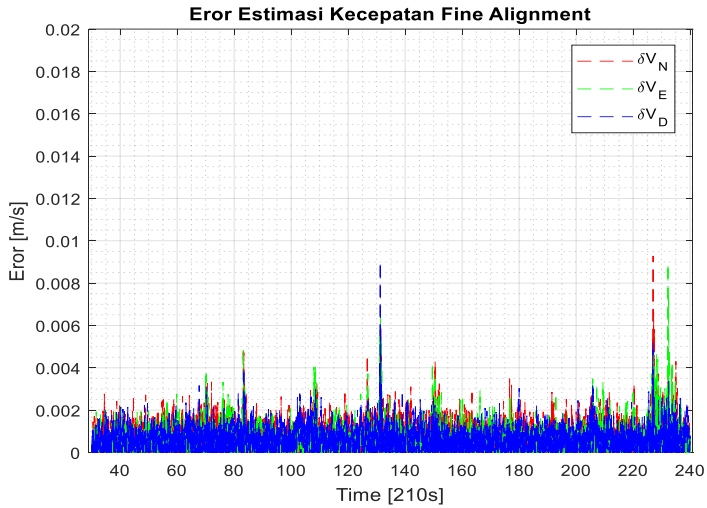
**Gambar 4.9** Hasil estimasi yaw dari *fine alignment*



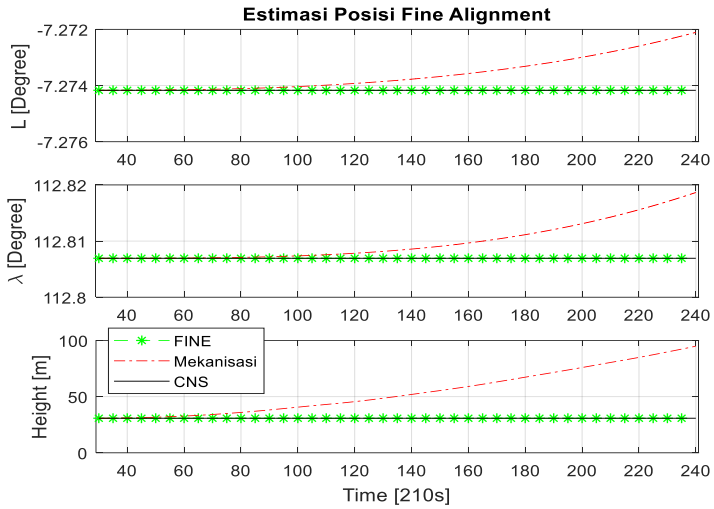
**Gambar 4.10** Error (estimasi-CNS) estimasi *attitude* dari *fine alignment*



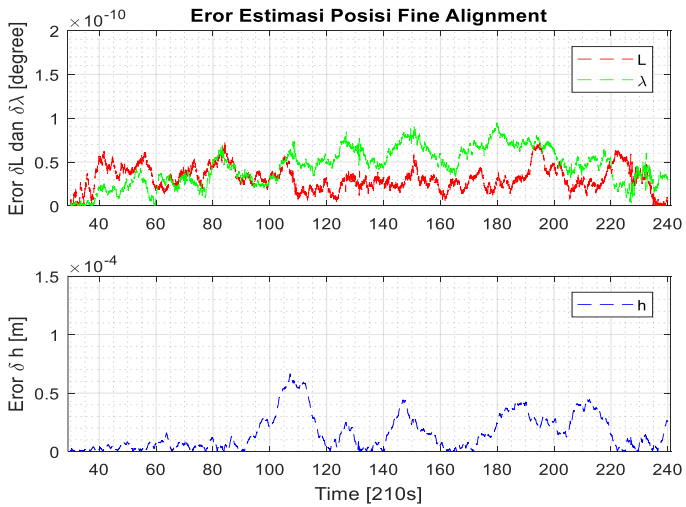
Gambar 4.11 Hasil estimasi kecepatan dari *fine alignment*



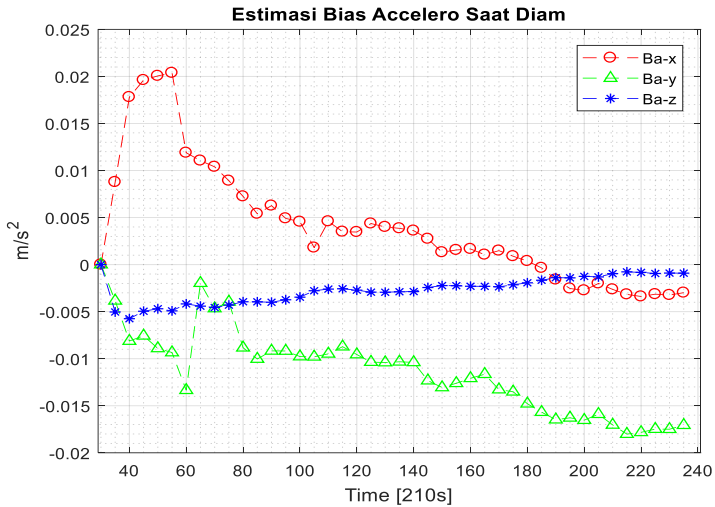
Gambar 4.12 Eros hasil estimasi kecepatan pada *fine alignment*



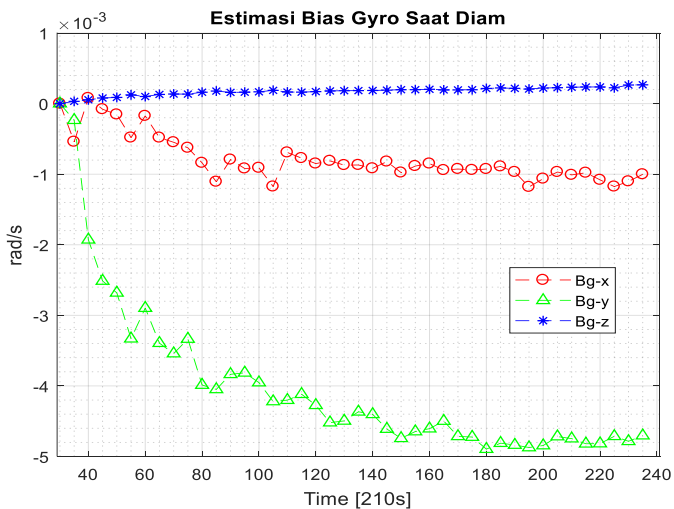
**Gambar 4.13** Hasil estimasi posisi dari *fine alignment*



**Gambar 4.14** Error hasil estimasi posisi dari *fine alignment*



**Gambar 4.15** Hasil estimasi bias *accelero* dari Filter Kalman pada *fine alignment*



**Gambar 4.16** Hasil estimasi bias *gyroscope* dari Filter Kalman pada *fine alignment*

Pada proses *fine alignment* terdapat Filter Kalman yang menggunakan parameter pada Tabel 3.1. Hasil dari estimasi *attitude* pada *fine alignment* mengalami peningkatan keakuratan pada nilai rata-rata eror hingga lebih dari 50% untuk sudut *yaw* dan lebih dari 34% untuk sudut *roll*, namun terjadi pengurangan nilai rata-rata sebesar 2.23% untuk sudut *pitch*. Hal ini dapat disebabkan oleh terjadinya gerakan yang diakibatkan manusia ketika pengukuran data *fine alignment* diambil dimana seperti yang telah diuraikan pada bab 2 bahwa proses *alignment* wahana diasumsikan berada dalam kondisi benar-benar diam. Parameter lain dari eror estimasi *attitude* mengalami peningkatan keakuratan walaupun pada beberapa parameter eror meningkat, namun hal tersebut masih dapat ditoleransi karena bernilai kurang dari  $2^\circ$ . Sedangkan untuk *yaw* masih memiliki eror yang cukup besar dengan nilai rata-rata mencapai  $13^\circ$ . Hal ini dikarenakan untuk menggunakan metode *gyrocompassing* membutuhkan sensor yang memiliki *aviation grade* atau yang lebih baik. Namun hasil estimasi *yaw* dari *fine alignment* hasil rancangan pada Tugas Akhir ini lebih baik dari hasil mekanisasi yang dapat dilihat pada Gambar 4.10 menampilkan eror estimasi *attitude* dari *fine alignment*.

**Tabel 4.2** Paramter eror hasil *fine alignment*

<b>Error Attitude (Estimasi-CNS)</b>	<b><math>[\delta\phi \ \delta\theta \ \delta\psi]</math> degree</b>
<i>Mean</i>	[0.1277 0.8992 13.0747]
<i>Max</i>	[0.3263 1.0465 19.1766]
<i>Min</i>	[0.0544 0.8160 12.4516]
<b>Error Kecepatan (Estimasi-CNS)</b>	<b><math>[\delta V_N \ \delta V_E \ \delta V_D]</math> m/s</b>
<i>Mean</i>	$[0.7331 \ 0.7190 \ 0.6756] \times 10^{-3}$
<i>Max</i>	[0.0093 0.0089 0.0090]
<i>Min</i>	$[0.1069 \ 0.0271 \ 0.0085] \times 10^{-7}$
<b>Error Posisi (Estimasi-CNS)</b>	<b><math>[\delta L \ \delta\lambda \ \delta h]</math></b>
<i>Mean</i>	$[0.0314^\circ n \ 0.0449^\circ n \ 0.013 \text{ mm}]$
<i>Max</i>	$[0.0736^\circ n \ 0.0941^\circ n \ 0.0666 \text{ mm}]$

<i>Min</i>	$[0^\circ \quad 0^\circ \quad 0.0003nm]$
<b>Estimasi bias Accelero</b>	$[b_{a,z} \quad b_{a,z} \quad b_{a,z}] \text{ m/s}^2$
<i>Mean</i>	$[0.0041 \quad -0.0114 \quad -0.0028]$
<i>Max</i>	$[0.0235 \quad 0.0167 \quad 0.0007]$
<i>Min</i>	$[-0.0038 \quad -0.0191 \quad -0.0104]$
<i>Standard Deviation</i>	$[0.0066 \quad 0.0048 \quad 0.0015]$
<b>Estimasi bias Gyroscope</b>	$[b_{g,z} \quad b_{g,z} \quad b_{g,z}] \text{ rad/s}$
<i>Mean</i>	$[-0.1375 \quad -0.7076 \quad 0.0307] \times 10^{-4}$
<i>Max</i>	$[0.8014 \quad 0.5483 \quad 0.5165] \times 10^{-5}$
<i>Min</i>	$[-0.2233 \quad -0.8586 \quad -0.0036] \times 10^{-4}$
<i>Standard Deviation</i>	$[0.0577 \quad 0.1865 \quad 0.0103] \times 10^{-4}$

#### 4.1.2.2 Hasil Estimasi Kecepatan Fine Alignment

Pada estimasi kecepatan dari *fine alignment* eror yang dimiliki relatif sangat kecil hingga kurang dari  $1 \text{ mm/s}$  untuk nilai rata-rata kecepatan pada ketiga sumbu dan kurang dari  $10 \text{ mm/s}$  untuk nilai tertingginya. Hal ini dapat menyatakan proses *fine alignment* yang menggunakan Filter Kalman dengan bantuan GPS dapat mengestimasi kecepatan dengan presisi dibanding hanya menggunakan mekanisasi tanpa Filter Kalman yang disajikan pada Gambar 4.11 dan secara numerik dapat dilihat pada Tabel 4.2. Secara sekilas kecepatan dari hasil mekanisasi terus meningkat seiring bertambahnya waktu. Hal tersebut disebabkan oleh bias yang tidak dikoreksi pada tahap mekanisasi sehingga terakumulasi seiring bertambahnya waktu.

#### 4.1.2.3 Hasil Estimasi Posisi Fine Alignment

Selanjutnya hasil estimasi posisi pada *fine alignment* disajikan pada Gambar 4.13, terlihat sekilas deviasi dari hasil mekanisasi terhadap nilai posisi sebenarnya (KVH-CNS5000) selalu bertambah besar seiring bertambahnya waktu. Hal tersebut dikarenakan tidak dilakukannya koreksi terhadap sinyal dari sensor yang mengakibatkan eror dari mekanisasi angkat meningkat secara kuadratik seiring bertambahnya waktu. Pada Gambar 4.13 menampilkan hasil estimasi posisi dengan menggunakan Filter

Kalman yang secara sekilas saling berhimpit dengan nilai posisi sebenarnya. Secara numerik dapat dilihat pada Tabel 4.2. eror dari estimasi posisi pada *fine alignment* relatif sangat kecil. Jika kita mendefinisikan  $1^\circ$  dari lintang dan bujur sama dengan  $111.322 \text{ km}$  maka eror dari *latitude* dan *longitude* berkisar antara  $0.1 \text{ mm}$ . Hal ini dapat menyatakan bahwa dengan menggunakan Filter Kalman pada proses *fine alignment* hasil navigasi yang berupa posisi dapat memiliki tingkat akurasi dan presisi yang tinggi ketimbang hanya menggunakan Mekanisasi dibuktikan pula dengan parameter eror pada Tabel 4.3.

#### 4.1.2.4 Hasil Estimasi Bias Fine Alignment

Pada *fine alignment* bias dari sensor *accelero* dan *gyroscope* dapat diestimasi menggunakan Filter Kalman. Hal tersebut dikarenakan model sistem dari Filter Kalman menggunakan model eror dari INS. Bias dari sensor *accelero* dapat dilihat pada Gambar 4.15. dan Gambar 4.16 untuk bias *gyroscope*. parameter dari eror bias dari sensor dapat dilihat pada Tabel 4.2.

#### 4.1.2.5 Rekapitulasi RMSE dari Fine Alignment

Pada bagian ini akan disajikan rekapitulasi perbandingan RMSE dari hasil *fine alignment* yang menggunakan Filter Kalman sebagai estimator eror dengan hasil dari mekanisasi yang dapat dilihat pada Tabel 4.3.

**Tabel 4.3** Perbandingan RMSE dan STD Filter Kalman dengan Mekanisasi

<b>ATTITUDE</b>				
<b>Hasil navigasi</b>	<b>Filter Kalman</b>		<b>Mekanisasi</b>	
	<b>RMSE</b>	<b>STD</b>	<b>RMSE</b>	<b>STD</b>
<i>Roll</i> ( $\phi$ )	0.1313	0.0304	0.1183	0.0398
<i>Pitch</i> ( $\theta$ )	0.9006	0.0486	1.4241	0.3006
<i>Yaw</i> ( $\psi$ )	13.1101	0.9632	19.1573	0.1546
<b>KECEPATAN</b>				
$V_N$	0.0010	0.001	1.3781	0.8480
$V_E$	0.0010	0.001	8.3050	5.5739
$V_D$	0.0009	0.0009	0.3365	0.1411

	POSISI			
<i>Latitude</i> ( <i>L</i> )	3.4254 $\times 10^{-11}$	1.4415 $\times$ $10^{-11}$	0.0008	5.8990 $\times 10^{-4}$
<i>Longitude</i> ( $\lambda$ )	4.9661 $\times 10^{-11}$	2.1310 $\times 10^{-11}$	0.0044	0.0033
<i>Height</i> ( <i>h</i> )	2.2084 $\times 10^{-5}$	1.7873 $\times 10^{-5}$	30.6265	19.4364

## 4.2 Hasil Simulasi Saat Wahana Bergerak

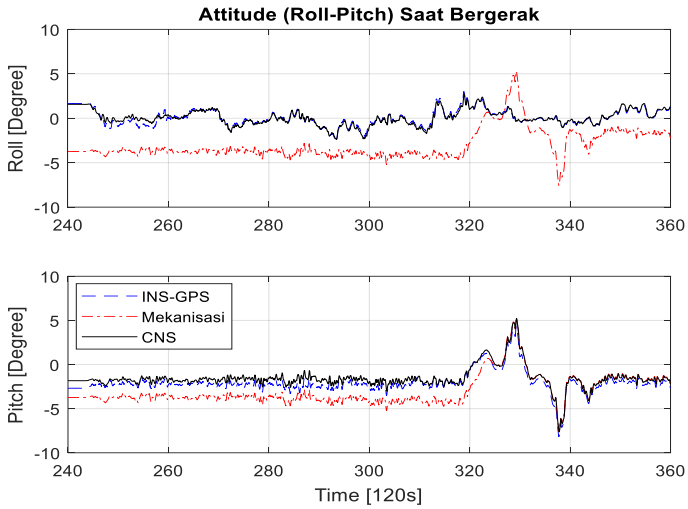
INS merupakan sistem navigasi yang membutuhkan inisialisasi awal sebelum digunakan oleh wahana. Inisialisasi awal tersebut berupa posisi, kecepatan dan *attitude* awal serta bias dari sensor *accelero* dan *gyro*. Inisialisasi tersebut dapat dilakukan pada proses *alignment*. Selanjutnya setelah dilakukan *alignment* maka sistem navigasi siap digunakan untuk bergerak. Data untuk simulasi bergerak diambil pada titik awal  $-7.27417004379^\circ$  untuk *latitude* dan  $112.80694047152^\circ$  untuk *longitude* dengan ketinggian 30.7645 meter. Nilai tersebut diambil menggunakan KVH-CNS5000 yang pada Tugas Akhir ini dianggap hasil navigasi sebenarnya.

### 4.2.1 Hasil Estimasi *Attitude* Saat Bergerak

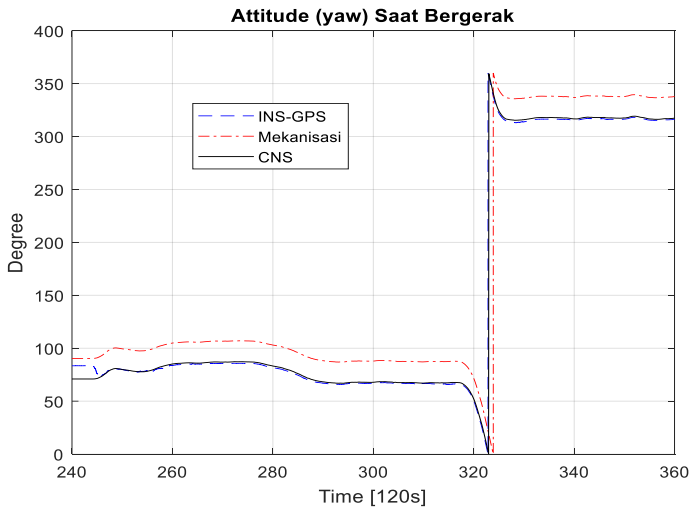
Nilai akhir dari *attitude* pada *fine alignment* akan menjadi nilai awal dari sistem navigasi saat bergerak. Hasil estimasi *attitude* saat bergerak dapat dilihat pada Gambar 4.17 dan Gambar 4.18. Secara sekilas hasil dari estimasi *attitude* berhimpit dengan nilai sebenarnya (KVH-CNS5000) dibandingkan dengan hasil mekanisasi. Secara numerik penyajian untuk melihat parameter eror dari estimasi *attitude* ketika bergerak dapat dilihat pada Tabel 4.4.

Nilai rata-rata eror dari hasil estimasi *attitude* pada keadaan bergerak berkisar antara  $1.7032^\circ$  sampai  $2.2618^\circ$  hal ini masih dapat ditoleransi karena pada saat melakukan pengukuran alat ukur tidak ditempatkan pada posisi yang sama pada mobil, sehingga *attitude* yang dialami berbeda. Oleh karena itu, sistem navigasi yang dirancang pada Tugas Akhir ini dapat mengestimasi *attitude* dengan akurat.

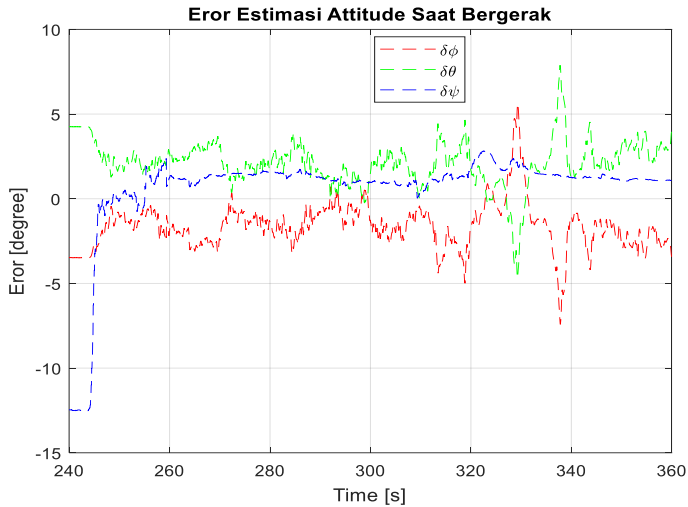




**Gambar 4.17** Hasil estimasi *Attitude (Roll-Pitch)* dari Filter Kalman ketika kondisi bergerak.



**Gambar 4.18** Hasil estimasi *attitude* dari Filter Kalman pada kondisi bergerak

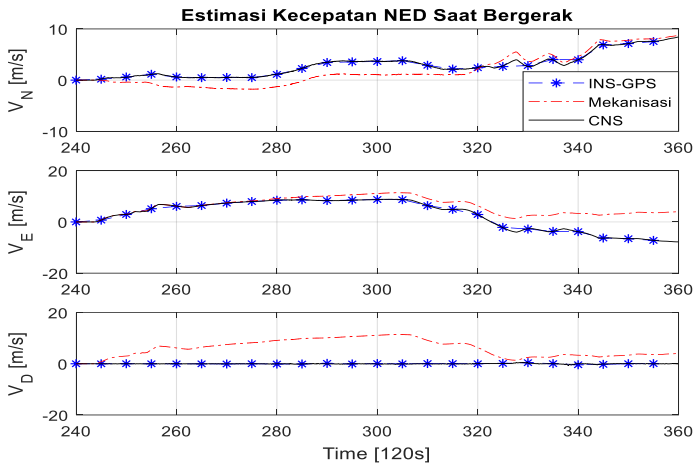


**Gambar 4.19** Eror dari estimasi *attitude* dari Filter Kalman saat bergerak

**Tabel 4.4** Parameter eror dari estimasi hasil navigasi saat bergerak

<b>Eror Attitude (Estimasi-CNS)</b>	<b><math>[\delta\phi \ \delta\theta \ \delta\psi]</math> degree</b>
<i>Mean</i>	[1.8495 2.2618 1.7032]
<i>Max</i>	[7.4159 7.8822 12.5222]
<i>Min</i>	[0.0013 0.0191 0.0024]
<b>Eror Kecepatan (Estimasi-CNS)</b>	<b><math>[\delta V_N \ \delta V_E \ \delta V_D]</math> m/s</b>
<i>Mean</i>	[0.0483 0.0445 0.0967]
<i>Max</i>	[0.2035 0.2270 0.9760]
<i>Min</i>	$[0.0999 \ 0.0723 \ 0.4922] \times 10^{-7}$
<b>Eror Posisi (Estimasi-CNS)</b>	<b><math>[\delta L \ \delta\lambda \ \delta h]</math></b>
<i>Mean</i>	[0.01920°m 0.0056°m 0.9171 meter]
<i>Max</i>	[0.0309°m 0.0113°m 2.9659 meter]

<i>Min</i>	[0.0067°m 0.7541°n 0.0037meter]
<b>Estimasi bias Accelero</b>	<b><math>[b_{a,z} \ b_{a,z} \ b_{a,z}] \text{ m/s}^2</math></b>
<i>Mean</i>	[-0.0769 -0.0366 -0.0004]
<i>Max</i>	[0.0297 0.0502 0.0082]
<i>Min</i>	[-0.1731 -0.1899 -0.0087]
<i>Standard Deviation</i>	[0.0313 0.0349 0.0026]
<b>Estimasi bias Gyroscope</b>	<b><math>[b_{g,z} \ b_{g,z} \ b_{g,z}] \text{ rad/s}</math></b>
<i>Mean</i>	$[-0.3539 \ -0.8654 \ 0.9344] \times 10^{-4}$
<i>Max</i>	$[-0.0165 \ -0.0697 \ 0.2436] \times 10^{-3}$
<i>Min</i>	$[-0.0504 \ -0.1159 \ 0.0029] \times 10^{-3}$
<i>Standard Deviation</i>	$[0.0618 \ 0.0990 \ 0.3748] \times 10^{-4}$

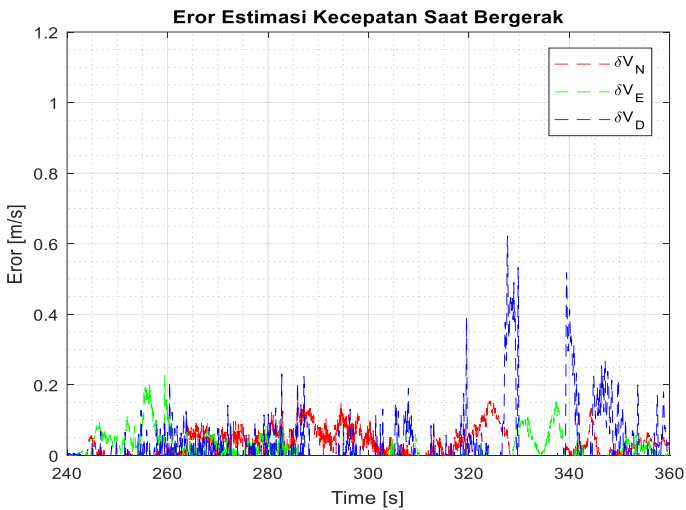


**Gambar 4.20** Hasil estimasi kecepatan dari Filter Kalman ketika wahana bergerak.

#### 4.2.2 Hasil Estimasi Kecepatan Saat Bergerak

Estimasi kecepatan pada kondisi bergerak relatif kecil dengan nilai rata 0.0483, 0.0445 dan 0.0967 meter dengan nilai maksimal 0.2035,

0.2270, 0.9760 meter untuk ketiga sumbunya. Angkat tersebut relatif kecil, dibandingkan dengan hasil mekanisasi yang secara sekilas tidak berhimpit dengan kurva kecepatan sebenarnya melainkan cenderung menjauh dari kurva kecepatan sebenarnya. Sehingga sistem navigasi yang dirancang pada Tugas Akhir ini lebih presisi dibandingkan hasil mekanisasi saja. Hal tersebut dibuktikan pula dengan STD kecepatan hasil estimasi Filter Kalman pada Tabel 4.5 lebih kecil dari hasil mekanisasi saja. Perlu diketahui bahwa nilai error tersebut tidak menyatakan bias kecepatan yang akan mengakibatkan error posisi meningkat seiring berjalannya waktu. Namun, error tersebut merupakan nilai mutlak dari rata-rata deviasi kecepatan yang diestimasi dengan kecepatan dari KVH-CNS5000. Sedangkan secara numerik hasil error hasil estimasi kecepatan Filter Kalman pada kondisi bergerak dapat dilihat pada Tabel 4.4.



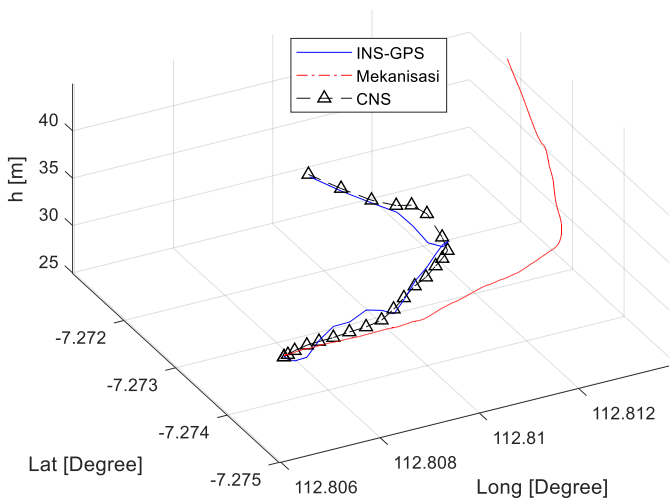
**Gambar 4.21** Error hasil estimasi kecepatan dari Filter Kalman saat bergerak

#### 4.2.3 Hasil Estimasi Posisi Saat Bergerak

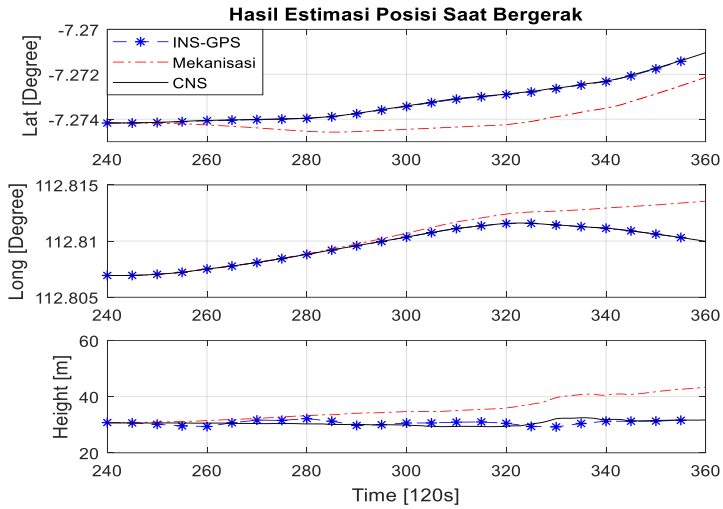
Pada estimasi posisi yang merupakan hasil integral sekali dari kecepatan yang dilakukan pada proses mekanisasi dan dikoreksi oleh Filter Kalman memiliki nilai error rata-rata  $0.01920 \times 10^{-3}$  pada

*latitude*,  $0.0056^\circ \times 10^{-3}$  dan 0.9171 meter untuk ketinggian. Bila kita mendefinisikan nilai  $1^\circ$  sama dengan 111.322 km maka eror *latitude* dan *longitude* adalah 2.1374 meter dan 0.6234 meter. Angka tersebut relatif kecil dibanding dengan eror hasil mekanisasi yang pada Gambar 4.22 terlihat secara sekilas hasil perhitungan dari mekanisasi menyimpang dari nilai sebenarnya. Pada Gambar 4.23 menampilkan hasil perhitungan mekanisasi dan hasil estimasi INS-GPS menggunakan Filter Kalman, hasil mekanisasi cenderung menyimpang dari nilai sebenarnya. Hal tersebut dikarenakan pada proses mekanisasi tidak dilakukan koreksi terhadap eror dari INS. Oleh karena itu, hasil estimasi menggunakan Filter Kalman relatif lebih akurat dan presisi dibuktikan dengan parameter eror yang ada pada Tabel 4.5. Hal tersebut dikarena Filter Kalman melakukan koreksi sinyal pengukuran dari sensor *accelero* dan *gyroscope* selain itu Filter Kalman melakukan estimasi eror posisi yang kemudian digunakan untuk mengkoreksi posisi hasil mekanisasi. Secara numerik perbandingan kepresisian dari estimasi posisi saat wahana bergerak dapat dilihat pada Tabel 4.5.

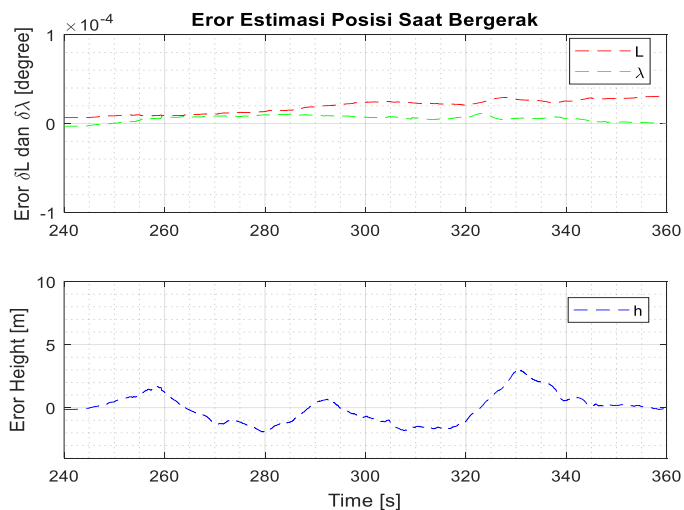
**Hasil Estimasi Posisi Saat Bergerak**



**Gambar 4.22** Hasil estimasi posisi dari Filter Kalman saat bergerak



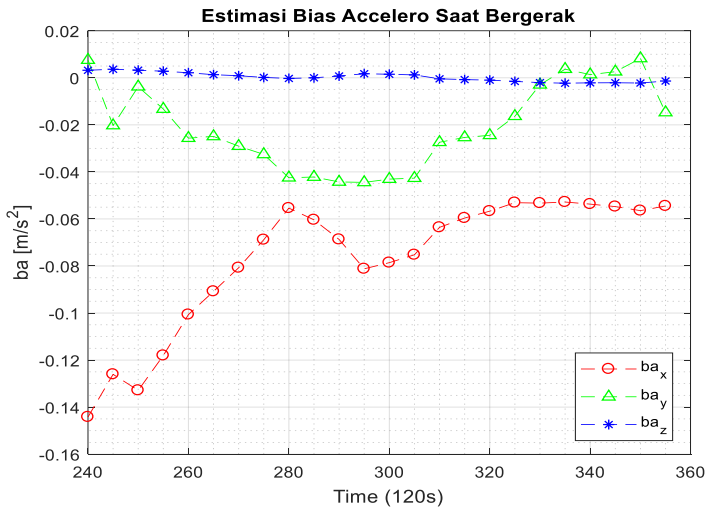
**Gambar 4.23** Hasil estimasi posisi dengan menampilkan setiap sumbu pada *local navigation frame*.



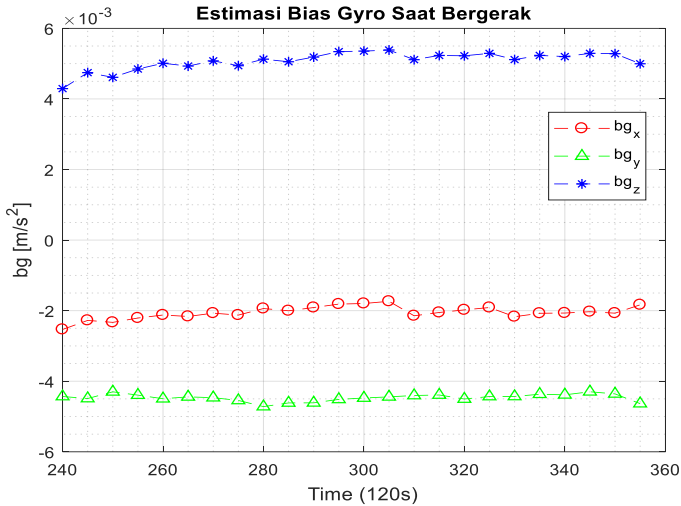
**Gambar 4.24** Error estimasi posisi dari Filter Kalman saat kondisi bergerak

#### 4.2.4 Hasil Estimasi Bias Saat Kondisi Bergerak

Pada kondisi bergerak Filter Kalman mengestimasi bias dari sensor *accelero* dan *gyroscope* yang disajikan pada Gambar 4.25 dan Gambar 4.26. Hasil tersebut kemudian menjadi korektor dari sinyal pengukuran sensor *accelero* dan *gyroscope* dengan cara dikurangkan pada sinyal pengukuran *accelero* dan *gyroscope*. Hasil dari estimasi bisa ini dapat dilihat dari perbandingan hasil navigasi menggunakan Filter Kalman dengan hasil mekanisasi saja. Pada perbandingan tersebut hasil mekanisasi saja memiliki eror yang cenderung meningkat dimana untuk kecepatan meningkat secara linear dan posisi meningkat secara kuadratik. Hal ini dapat disebabkan oleh tidak dilakukannya koreksi pada sinyal pengukuran yang mengandung bias. Oleh karena itu dapat disimpulkan bahwa hasil estimasi bias dengan menggunakan Filter Kalman dapat meningkatkan presisi dari hasil navigasi.



Gambar 4.25 Hasil estimasi bias pada *accelero* dari Filter Kalman



**Gambar 4.26** Hasil estimasi bias *gyroscope* dari Filter Kalman.

#### 4.2.5 Rekapitulasi RMSE Dari Kondisi Bergerak

Pada bagian ini akan disajikan perbandingan RMSE dari hasil estimasi Filter Kalman dengan hasil mekanisasi. Kemudian data RMSE ini dapat menjadi acuan dalam menarik kesimpulan. Penyajian dari komparasi RMSE tersebut dapat dilihat pada

**Tabel 4.5** Perbandingan RMSE dan STD saat wahana bergerak

<i>ATTITUDE</i>				
Hasil navigasi	Filter Kalman		Mekanisasi	
	RMSE	STD	RMSE	STD
<i>Roll</i> ( $\phi$ )	0.2326	0.2326	0.4904	0.3370
<i>Pitch</i> ( $\theta$ )	0.5388	0.2332	0.9215	0.4234
<i>Yaw</i> ( $\psi$ )	2.7953	2.7219	19.3315	1.9393
<i>KECEPATAN</i>				
$V_N$	0.0638	0.0616	1.7179	1.4003



$V_E$	0.0586	0.0586	4.8001	3.4867
$V_D$	0.1430	0.1406	0.2404	0.2232
<b>POSISI</b>				
<i>Latitude</i> ( $L$ )	2.0668 $\times 10^{-5}$	7.6489 $\times 10^{-6}$	9.4330 $\times 10^{-4}$	4.6621 $\times 10^{-4}$
<i>Longitude</i> ( $\lambda$ )	6.5847 $\times 10^{-6}$	3.4842 $\times 10^{-6}$	0.0012	9.6614 $\times 10^{-4}$
<i>Height</i> ( $h$ )	1.1485	1.1469	5.9941	3.3935

*[Halaman ini sengaja dikosongkan]*

## **BAB 5**

### **KESIMPULAN DAN SARAN**

Pada bab ini akan diuraikan kesimpulan dari Tugas Akhir ini berdasarkan hasil simulasi dari rancangan sistem navigasi yang telah dilakukan pada bab-bab sebelumnya. Selanjutnya akan diberikan saran untuk topik serupa.

#### **5.1 Kesimpulan**

Simulasi telah dilakukan pada bab 4 dan telah didapatkan hasil dari simulasi tersebut. Oleh karena itu, berdasarkan data hasil simulasi tersebut dapat disimpulkan beberapa hal dari Tugas Akhir ini yaitu:

- a. Hasil *coarse alignment* dari rancangan Tugas Akhir ini dapat digunakan untuk menentukan *attitude* awal dari sistem navigasi. Hal tersebut disimpulkan dari nilai rata-rata eror *roll* dan *pitch* kurang dari  $0.9^\circ$ . Namun beda halnya dengan estimasi *yaw* memiliki nilai rata-rata eror lebih dari  $27^\circ$ . Hal ini dikarenakan eror bias pada *coarse alignment* tidak diperhitungkan.
- b. *Fine alignment* dengan menggunakan Filter Kalman dapat mengurangi eror dari *yaw* hasil *coarse alignment*, sehingga estimasi *attitude* awal lebih akurat. Hal tersebut dapat dilihat dari hasil simulasi. Walaupun eror dari *yaw* masih relatif besar dengan RMSE 13.1101. Angka tersebut masih dapat ditoleransi karena hal tersebut disebabkan *grade* dari sensor tidak cukup baik digunakan untuk *gyrocompassing* yang merupakan *tactical grade*.
- c. Hasil simulasi menunjukkan estimasi posisi dan kecepatan menggunakan Filter Kalman pada *fine alignment* lebih presisi dan akurat dibandingkan hanya menggunakan proses mekanisasi. Hal tersebut dapat dilihat pada RMSE dan standar deviasi dengan menggunakan Filter Kalman relatif lebih kecil dari pada hanya menggunakan mekanisasi saja.
- d. Pada keadaan bergerak hasil estimasi menggunakan Filter Kalman lebih baik dari hasil proses mekanisasi. Hal tersebut dibuktikan dengan RMSE hasil Filter Kalman selalu lebih kecil dari hasil mekanisasi.

- e. Hasil estimasi bias pada Filter Kalman baik pada *fine alignment* dan saat bergerak mampu mengurangi eror yang meningkat seiring bertambahnya waktu yang dibuktikan dengan hasil estimasi posisi, kecepatan dan *attitude* tidak menyimpang seperti pada hasil mekanisasi.

## 5.2 Saran

Selain kesimpulan yang dijabarkan pada bab 5 ini, akan diberikan beberapa saran sebagai lanjutan dari Tugas Akhir ini yaitu antara lain:

- a. Sistem navigasi pada Tugas Akhir ini hanya disimulasikan pada komputer menggunakan *software* MATLAB. Sehingga untuk kedepannya diharapkan dapat diimplementasikan pada wahana tertentu.
- b. Proses *alignment* pada Tugas Akhir ini membutuhkan waktu 4 menit. Oleh karena itu, penelitian selanjutnya diharapkan dapat mengurangi waktu *alignment* tanpa mengurangi keakuratan hasil navigasi sebelumnya.

## DAFTAR PUSTAKA

- [1] A. Noureldin, T. B. Karamat dan J. Georgy, *Fundamentals of Inertial Navigation, Satellite-based Positioning and their Integration*, New York: Springer, 2013.
- [2] E. Bekir, *Introduction to Modern Navigation Systems*, Singapore: World Scientific, 2007.
- [3] P. D. Groves, *Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems*, Boston: Artech House, 2013.
- [4] M. S. Grewal, A. P. Andrews dan C. G. Bartone, *Global Navigation Satellite Systems, Inertial Navigation, and Integration*, Canada: Wiley, 2013.
- [5] R. G. Brown dan P. Y. C. Hwang, *Introduction to Random Signals and Applied Kalman Filtering with Matlab Exercises*, 2008.
- [6] Shang, Hong & Wang, ZHELONG & Jiang, Ming. (2011). Comparison of initial alignment methods for SINS. *Proceedings of the World Congress on Intelligent Control and Automation (WCICA)*. 10.1109/WCICA.2011.5970584.
- [7] Liu, Meng et al. "An Improved Alignment Method for the Strapdown Inertial Navigation System (SINS)." *Sensors (Basel, Switzerland)* vol. 16,5 621. 29 Apr. 2016, doi:10.3390/s16050621
- [8] L. Schimelevich and R. Naor, "New approach to coarse alignment," *Proceedings of Position, Location and Navigation Symposium - PLANS '96*, Atlanta, GA, USA, 1996, pp. 324-327. doi: 10.1109/PLANS.1996.509095

- [9] H. Nourmohammadi and J. Keighobadi, "Fuzzy adaptive integration scheme for low-cost SINS/GPS navigation system," *Mechanical Systems and Signal Processing*, vol. 99, pp. 434-449, 2018.
- [10] C. Xiyuan, S. Chong, Z. Wei-bin, T. Masayoshi, X. Yuan and C. Kuanlin, "Novel hybrid of strong tracking Kalman filter and wavelet neural network for GPS/INS during GPS outages," *Measurement*, vol. 46, no. 10, pp. 3847-3854, 2013.
- [11] Butterflygyro, 'Datasheet STIM300 *Inertial Measurement Unit*', Sensor, 2018

## LAMPIRAN

### *script Main.m*

```
tic
%% PERCOBAAN INS/GPS DATA SENSOR DI MOBIL 1 NOV 2018
% Skenario I
% Data : Rekam Data 181101 ('scenel.mat') NED
% main.m
clear;
close all;
clc;
%Load Data
load('..\..\BELAJAR MATLAB\DATA PENGUKURAN\data
scenes\scenel.mat');
addpath('fungsi grove');
%% ===== COARSE ALIGNMENT =====
% setting parameter...
dur_coarse = 20; % detik
f_stim = 2000; % Hz
f_cns = 20; % Hz
% jumlah data yang dibuang (detik)
ignore_data_time = 10;
data_buang = ignore_data_time*f_stim;
mulai_coarse = data_buang;
% allocating...
phiAngle=zeros(20*f_stim,1);
thetaAngle=zeros(20*f_stim,1);
psiAngle=zeros(20*f_stim,1);
% initializing...
prev_acc_mean=[0 0 0]';
prev_gyro_mean=[0 0 0]';
prev_acc_var = [0 0 0]';
prev_gyro_var = [0 0 0]';
init_L_b = deg2rad(cns01(1,1)); % mengambil dari GPS
init_lambda_b = deg2rad(cns01(1,2)); % mengambil dari GPS
init_h_b = cns01(1,3); % mengambil dari GPS
% ambil data cns sebagai pembanding
att_cns_coarse =
cns01(10*f_cns+1:10*f_cns+dur_coarse*f_cns,7:9);
gravity = 9.80665;
% banyak data coarse
jum_data_coarse = dur_coarse*f_stim;
% START!!!
for step=1:jum_data_coarse
accCoarse=ned01(mulai_coarse+step,4:6)*gravity;
gyroCoarse=ned01(mulai_coarse+step,1:3);
gyroCoarse=deg2rad(gyroCoarse);
```

```

if step == 1
    omega_ib_b_mean= gyroCoarse;
    var_gyro = prev_gyro_var;
    f_ib_b_mean = accCoarse;
    var_acc = prev_acc_var;
else
omega_ib_b_mean= (gyroCoarse+prev_gyro_mean*(step-1))/step;
var_gyro = ((step-1)*prev_gyro_var + (omega_ib_b_mean-
gyroCoarse).^2)/step;
f_ib_b_mean = (accCoarse+prev_acc_mean*(step-1))/step;
var_acc = ((step-1)*prev_acc_var + (f_ib_b_mean-
accCoarse).^2)/step;
end
temp =
f_ib_b_mean(1)/(sqrt(f_ib_b_mean(2)^2+f_ib_b_mean(3)^2));
theta=atan(temp);
phi=atan2(-f_ib_b_mean(2), -f_ib_b_mean(3));
psi=atan2(omega_ib_b_mean(3)*sin(phi)-
omega_ib_b_mean(2)*cos(phi),omega_ib_b_mean(1)*cos(theta)+om
ega_ib_b_mean(2)*sin(theta)*sin(phi)+omega_ib_b_mean(3)*sin(
theta)*cos(phi));
% saving result...
phiAngle(step,1)=rad2deg(phi);
thetaAngle(step,1)=rad2deg(theta);
psiAngle(step,1)=rad2deg(psi);
%updating...
prev_acc_mean=f_ib_b_mean;
prev_gyro_mean=omega_ib_b_mean;
prev_acc_var = var_acc;
prev_gyro_var = var_gyro;
end %END!!!
% hasil sitem pembanding (p,r,y)
theta_cns=deg2rad(cns01(1,8));
phi_cns=deg2rad(cns01(1,7));
psi_cns=deg2rad(cns01(1,9)); % dilihat dari inspva CNS-5000
% send to fine...
att_b_Coarse=[phi theta psi]';
gyro_std1 =
std(deg2rad(ned01(mulai_coarse+1:dur_coarse*f_stim,1:3)));
acc_std1 =
std((ned01(mulai_coarse+1:dur_coarse*f_stim,4:6))*gravity);
gyro_std = sqrt(var_gyro);
acc_std = sqrt(var_acc);
disp('Aligning...')
%% ===== FINE ALIGNMENT =====
% setting parameter...
mulai_fine = mulai_coarse + jum_data_coarse;
dur_fine = 210;

```



```

f_stim=2000;
f_cns;
jum_data_Fine = dur_fine*f_stim;
% ambil data...
acc=ned01(mulai_fine+1:mulai_fine+jum_data_Fine,4:6);
gyro=ned01(mulai_fine+1:mulai_fine+jum_data_Fine,1:3);
gyro=deg2rad(gyro);
progres = textprogressbar(jum_data_Fine);
% Constanta
deg_to_rad = 0.01745329252;
rad_to_deg = 1/deg_to_rad;
micro_g_to_meters_per_second_squared = 9.80665E-6;
init_param_P;
init_param_Q;
init_param_R;
% Posisi sebenarnya diambil dari CNS5000
true_L_b = init_L_b;
true_lambda_b = init_lambda_b;
true_h_b = init_h_b;
% Kecepatan sebenarnya yaitu sama dengan 0
true_v_b_n = zeros(3,1);
% attitude sebenarnya diambil dari CNS5000
true_att_nb = att_b_Coarse;% deg2rad([0 0 0]'); % DALAM
BENTUK VEKTOR MEWAKILI KETIGA SUMBU
true_C_b_n = att2C(true_att_nb); % DALAM BENTUK MATRIKS
TRANSFORM
tor_s = 5e-4;%1e-2;
tor_i = 5e-4; % period dari ins = stim
t_corr=5e-4;
% untuk model gravitasi awal
est_L_b = true_L_b;
est_lambda_b = true_lambda_b;
est_h_b = true_h_b;
% Inisialisasi nilai sebelumnya (untuk metode DR)
old_est_C_b_n = true_C_b_n;
old_est_v_b_n = true_v_b_n;
old_L_b = true_L_b;
old_lambda_b = true_lambda_b;
old_h_b = true_h_b;
% inisialisasi matriks P
P_matriks = P_matriks_init(param_KF);
% Inisialisasi nilai awal GPS
GNSS_L_b = true_L_b;
GNSS_lambda_b = true_lambda_b;
GNSS_h_b = true_h_b;
GNSS_v_b_n = true_v_b_n;
% Allocating...
est_IMU_bias = zeros(6,1);

```

```

out_IMU_bias_est=zeros(jum_data_Fine, 6);
out_IMU_bias_est_Corr=zeros(jum_data_Fine/(t_corr*f_stim),
6);
out_IMU_bias_est(1,1:6) = est_IMU_bias';
% Data Output (PVA)
Position=zeros(jum_data_Fine,3); % P
Velocity=zeros(jum_data_Fine,3); % V
Attitude=zeros(jum_data_Fine,3); % A
bias_fine =zeros(jum_data_Fine,3);
% Pengukuran f_ib_b dan omega_ib_b
asli_f_ib_b_fine = Position;
asli_omega_ib_b_fine = Position;
corr_f_ib_b_fine = Position;
corr_omega_ib_b_fine = Position;
% ambil data sebagai pembeding
pos_cns_fine =
cns01(30*f_cns+1:30*f_cns+dur_fine*f_cns,1:3);
vel_cns_fine =
cns01(30*f_cns+1:30*f_cns+dur_fine*f_cns,4:6);
att_cns_fine =
cns01(30*f_cns+1:30*f_cns+dur_fine*f_cns,7:9);
% indexing...
j=1;
% mencari matrix untuk Kalman Filter Q,R and H
Q_prime_matrix = zeros(15);
Q_prime_matrix(1,1) = param_KF.gyro_noise_PSDX * tor_s;
Q_prime_matrix(2,2) = param_KF.gyro_noise_PSDY * tor_s;
Q_prime_matrix(3,3) = param_KF.gyro_noise_PSDZ * tor_s;
Q_prime_matrix(4,4) = param_KF.accel_noise_PSDX * tor_s;
Q_prime_matrix(5,5) = param_KF.accel_noise_PSDY * tor_s;
Q_prime_matrix(6,6) = param_KF.accel_noise_PSDZ * tor_s;
Q_prime_matrix(10:12,10:12) = eye(3)*param_KF.accel_bias_PSD
* tor_s;
Q_prime_matrix(13:15,13:15) = eye(3)*param_KF.gyro_bias_PSD
* tor_s;
R_matrix(1:2,1:2) = eye(2) * param_KF.pos_meas_SDXY^2;
R_matrix(3,3) = param_KF.pos_meas_SDZ^2;
R_matrix(1:3,4:6) = zeros(3);
R_matrix(4:6,1:3) = zeros(3);
R_matrix(4:6,4:6) = eye(3) * param_KF.vel_meas_SD^2;
H_matrix = zeros(6,15);
H_matrix(1:3,7:9) = -eye(3);
H_matrix(4:6,4:6) = -eye(3);
for k=1:jum_data_Fine
meas_f_ib_b = acc(k,:) * gravity;
asli_f_ib_b_fine(k,:) = meas_f_ib_b;
meas_omega_ib_b = gyro(k,:)';
asli_omega_ib_b_fine(k,:) = meas_omega_ib_b;

```

```

% Correcting IMU errors
meas_f_ib_b = meas_f_ib_b - est_IMU_bias(1:3);
corr_f_ib_b_fine(k,:) = meas_f_ib_b;
meas_omega_ib_b = meas_omega_ib_b - est_IMU_bias(4:6);
corr_omega_ib_b_fine(k,:) = meas_omega_ib_b;
% MECHANIZATION
[est_L_b, est_lambda_b, est_h_b, est_v_b_n, est_C_b_n] =
Mekanisasi(tor_i,
old_L_b, old_lambda_b, old_h_b, old_est_v_b_n, old_est_C_b_n, mea
s_f_ib_b, meas_omega_ib_b);
% KALMAN FILTER
[est_C_b_n, est_v_b_n, est_L_b, est_lambda_b, est_h_b, est_IMU_bi
as, P_matrix]=Kalman_Filter(GNSS_L_b, GNSS_lambda_b, GNSS_h_b, G
NSS_v_b_n, tor_s, est_C_b_n, est_v_b_n, est_L_b, est_lambda_b, est
_h_b, est_IMU_bias, P_matrix, meas_f_ib_b, param_KF);
    out_IMU_bias_est(j,:) = est_IMU_bias';
    % SAVE OUTPUT
    Attitude(j,:) = rad2deg(C2att(est_C_b_n));
    Velocity(j,:) = est_v_b_n;
    Position(j,1) = rad2deg(est_L_b);
    Position(j,2) = rad2deg(est_lambda_b);
    Position(j,3) = est_h_b;
    j=j+1;
    % Updating...
    % Reset old values
    old_L_b = est_L_b;
    old_lambda_b = est_lambda_b;
    old_h_b = est_h_b;
    old_est_v_b_n = est_v_b_n;
    old_est_C_b_n = est_C_b_n;
    if mod(k,1000)==0
        progres(k);
    elseif k+1>jum_data_Fine
        progres(k);
    end
end
% get last result from alignment
old_L_bx = est_L_b;
old_lambda_bx = est_lambda_b;
old_h_bx = est_h_b;
old_est_v_bx_n = est_v_b_n;
old_est_C_bx_n = est_C_b_n;
disp('Navigating...')
%% ===== NAVIGATION (120s) =====
% setting parameter...
start_Nav = mulai_fine + jum_data_Fine;
dur_nav = 120;
dur_align = ignore_data_time+dur_coarse+dur_fine; % 4 menit

```

```

f_stim=2000;
f_GPS = 10;
start_GPS_Nav=dur_align*f_GPS+1; % setelah 4 menit
jum_data_Nav = dur_nav*f_stim;
progres = textprogressbar(jum_data_Nav);
% ambil data...
% data stim 2000Hz
accNAV=ned01(start_Nav+1:start_Nav+jum_data_Nav,4:6);
gyroNAV=ned01(start_Nav+1:start_Nav+jum_data_Nav,1:3);
gyroNAV=deg2rad(gyroNAV);
% data GPS 10Hz
dataGPSNAV=gps01(start_GPS_Nav:start_GPS_Nav+dur_nav*f_GPS, :
);
tau_s = 0.1;
% Inisialisasi
old_L_b = est_L_b;
old_lambda_b = est_lambda_b;
old_h_b = est_h_b;
old_est_v_b_n = est_v_b_n;
old_est_C_b_n = est_C_b_n;
est_IMU_bias_Corr = est_IMU_bias;
% Allocating...
AttitudeNav=zeros(jum_data_Nav,3);
% AttitudeNav2=zeros(nDataNav,3);
VelocityNav=zeros(jum_data_Nav,3);
PositionNav=zeros(jum_data_Nav,3);
asli_f_ib_b_nav = PositionNav;
asli_omega_ib_b_nav = asli_f_ib_b_nav;
corr_f_ib_b_nav = asli_f_ib_b_nav;
corr_omega_ib_b_nav = asli_f_ib_b_nav;
est_IMU_bias_Corrplot = zeros(f_GPS*dur_nav,6);
GPS_v_b_n = zeros(dur_nav*f_GPS,3);
% ambil data cns sebagai pembanding
pos_cns_nav =
cns01(240*f_cns+1:240*f_cns+dur_nav*f_cns,1:3);
vel_cns_nav =
cns01(240*f_cns+1:240*f_cns+dur_nav*f_cns,4:6);
att_cns_nav =
cns01(240*f_cns+1:240*f_cns+dur_nav*f_cns,7:9);
% indexing...
m=1;
for k=1:jum_data_Nav
meas_f_ib_b = accNAV(k,:) '*gravity;
asli_f_ib_b_nav(k,:) = meas_f_ib_b;
meas_omega_ib_b = gyroNAV(k,:)';
asli_omega_ib_b_nav(k,:) = meas_omega_ib_b;
% Correct IMU errors
meas_f_ib_b = meas_f_ib_b - est_IMU_bias_Corr(1:3);

```

```

corr_f_ib_b_nav(k,:) = meas_f_ib_b;
meas_omega_ib_b = meas_omega_ib_b - est_IMU_bias_Corr(4:6);
corr_omega_ib_b_nav(k,:)= meas_omega_ib_b;
% Update estimated navigation solution
[est_L_b, est_lambda_b, est_h_b,est_v_b_n,est_C_b_n] =
Mekanisasi(tor_i,old_L_b,old_lambda_b,old_h_b,old_est_v_b_n,
old_est_C_b_n,meas_f_ib_b,meas_omega_ib_b);
% Run Kalman filter 10Hz || menunggu sinyal GPS
if mod(k, (1/tor_i)/(1/tor_s)) == 0
GNSS_L_b = deg2rad(dataGPSNAV(m,2));
GNSS_lambda_b = deg2rad(dataGPSNAV(m,3));
GNSS_h_b = dataGPSNAV(m,4);
GNSS_hor_speed = dataGPSNAV(m,5);
GNSS_ground_track = deg2rad(dataGPSNAV(m,6));
GNSS_ver_speed = dataGPSNAV(m,7);
GNSS_v_north = GNSS_hor_speed * cos(GNSS_ground_track);
GNSS_v_east = GNSS_hor_speed * sin(GNSS_ground_track);
GNSS_v_b_n = [GNSS_v_north; GNSS_v_east; GNSS_ver_speed];
GPS_v_b_n(m,:) =GNSS_v_b_n';

[est_C_b_n,est_v_b_n,est_L_b,est_lambda_b,est_h_b,est_IMU_bi
as,P_matrix]=Filter_Kalman(GNSS_L_b,GNSS_lambda_b,GNSS_h_b,G
NSS_v_b_n,tau_s,est_C_b_n,est_v_b_n,est_L_b,est_lambda_b,est
_h_b,est_IMU_bias,P_matrix,meas_f_ib_b,param_KF);
est_IMU_bias_Corr = est_IMU_bias;
est_IMU_bias_Corrplot(m,:) = est_IMU_bias_Corr';
m=m+1;
end % end if Run kalman
% Updating and SAVE OUTPUT % Reset old values
old_L_b = est_L_b;
old_lambda_b = est_lambda_b;
old_h_b = est_h_b;
old_est_v_b_n = est_v_b_n;
old_est_C_b_n = est_C_b_n;
AttitudeNav(k,:) = rad2deg(C2att(est_C_b_n)');
if(AttitudeNav(k,3) < 0)
AttitudeNav(k,3) = AttitudeNav(k,3) + 360;
end
VelocityNav(k,:) = est_v_b_n;
PositionNav(k,1) = rad2deg(est_L_b);
PositionNav(k,2) = rad2deg(est_lambda_b);
PositionNav(k,3) = est_h_b;
if mod(k,1000) == 0
progres(k);
elseif k+1 > jum_data_Nav
progres(k);
end
end
end

```

## script fungsi Mekanisasi.m

```
function [L_b,lambda_b,h_b,v_eb_n,C_b_n] = Mekanisasi(tau_i,
old_L_b,old_lambda_b,old_h_b,old_v_eb_n,old_C_b_n,f_ib_b,ome
ga_ib_b)
omega_ie = 7.292115E-5; % kecepatan putar bumi (rad/s)
% hitung alpha
alpha_ib_b = omega_ib_b * tor_i;
mag_alpha = sqrt(alpha_ib_b' * alpha_ib_b);
skew_alpha_ib_b = Skew_symmetric(alpha_ib_b);
% hitung kecepatan rotasi bumi pada nav
omega_ie_n = omega_ie * [cos(old_L_b); 0; -sin(old_L_b)];
[old_R_N,old_R_E] = Radii_of_curvature(old_L_b);
old_omega_en_n = [old_v_eb_n(2) / (old_R_E + old_h_b);
                 -old_v_eb_n(1) / (old_R_N + old_h_b);
                 -old_v_eb_n(2) * tan(old_L_b) / (old_R_E + old_h_b)];
% hitung att
mean_C_b_n = old_C_b_n * (eye(3) + (1 - cos(mag_alpha)) /
mag_alpha^2 * skew_alpha_ib_b + (1 - sin(mag_alpha) /
mag_alpha) / mag_alpha^2 * skew_alpha_ib_b * skew_alpha_ib_b)
-0.5 * Skew_symmetric(old_omega_en_n + omega_ie_n)
*old_C_b_n;
% trans f
f_ib_n = mean_C_b_n * f_ib_b;
% update vel
v_eb_n = old_v_eb_n + tor_i * (f_ib_n +
cari_g_NED(old_L_b,old_h_b)-Skew_symmetric(old_omega_en_n +
2 * omega_ie_n) * old_v_eb_n);
new_omega_en_n = [v_eb_n(2) / (old_R_E + old_h_b);
                 -v_eb_n(1) / (old_R_N + old_h_b);
                 -v_eb_n(2) * tan(old_L_b) / (old_R_E + old_h_b)];
v_eb_n = old_v_eb_n + tor_i * (f_ib_n +
cari_g_NED(old_L_b,old_h_b)-0.5*Skew_symmetric(
old_omega_en_n + 2 * omega_ie_n) * old_v_eb_n-0.5*
Skew_symmetric(new_omega_en_n + 2 * omega_ie_n)*v_eb_n);
% update pos
h_b = old_h_b - 0.5 * tor_i * (old_v_eb_n(3) + v_eb_n(3));
L_b = old_L_b + 0.5 * tor_i * (old_v_eb_n(1) / (old_R_N +
old_h_b) +v_eb_n(1) / (old_R_N + h_b));
% hitung jari-jari bumi
[R_N,R_E]= Radii_of_curvature(L_b);
lambda_b = old_lambda_b + 0.5 * tor_i * (old_v_eb_n(2) /
((old_R_E +old_h_b) * cos(old_L_b)) + v_eb_n(2) / ((R_E +
h_b) * cos(L_b)));
% update omega en_n
omega_en_n = [v_eb_n(2) / (R_E + h_b);
             -v_eb_n(1) / (R_N + h_b);
             -v_eb_n(2) * tan(L_b) / (R_E + h_b)];
```

```

    C_new_old = eye(3) + sin(mag_alpha) / mag_alpha *
    skew_alpha_ib_b + (1 - cos(mag_alpha)) / mag_alpha^2 *
    skew_alpha_ib_b * skew_alpha_ib_b;
% update att
C_b_n = (eye(3) - Skew_symmetric(omega_ie_n + 0.5 *
omega_en_n + 0.5*old_omega_en_n) * tor_i) * old_C_b_n *
C_new_old;

```

### **Script fungsi Filter\_Kalman.m**

```

function[est_C_b_n_new,est_v_b_n_new,est_L_b_new,est_lambda_
b_new,est_h_b_new,est_IMU_bias_new,P_matrix_new]=
Filter_Kalman(GNSS_L_b,GNSS_lambda_b,GNSS_h_b,GNSS_v_b_n,
tor_s,est_C_b_n_old,est_v_b_n_old,est_L_b_old,est_lambda_b_o
ld,est_h_b_old,est_IMU_bias_old,P_matrix_old,meas_f_ib_b,par
am_KF)
[Omega_ie_n, omega_ie_n] = GetOmega_ie_n(est_L_b_old);
[Omega_en_n, omega_en_n] = GetOmega_en_n(est_L_b_old,
est_h_b_old, est_v_b_n_old);
Omega_in_n=Omega_en_n+Omega_ie_n;
% model eror INS
Phi_matrix =
GeneratePhiMatrixNED(tor_s,Omega_in_n,est_L_b_old,est_h_b_ol
d,est_v_b_n_old,est_C_b_n_old,meas_f_ib_b,1);
% state prediksi
x_est_predict(1:15,1) = 0;

% Prediksi matriks kovarian P
P_matrix_predict = Phi_matrix * P_matrix_old * Phi_matrix' +
1 * Q_prime_matrix;
% Hitung gain Kalman
K_matrix = P_matrix_propagated * H_matrix' * inv(H_matrix*
P_matrix_propagated * H_matrix' + R_matrix);
% pengukuran
delta_z(1,1) = GNSS_L_b -est_L_b_old;
delta_z(2,1) = GNSS_lambda_b -est_lambda_b_old;
delta_z(3,1) = GNSS_h_b -est_h_b_old;
delta_z(4:6,1) = GNSS_v_b_n -est_v_b_n_old;
% estimasi state
x_est_new = x_est_propagated + K_matrix * delta_z;
% estimasi matriks kovarian
P_matrix_new = (eye(15) - K_matrix * H_matrix) *
P_matrix_propagated*(eye(15) - K_matrix * H_matrix)' +
K_matrix*R_matrix*K_matrix';
% koreksi hasil mekanisasi
est_C_b_n_new = (eye(3) - Skew_symmetric(x_est_new(1:3)))*
est_C_b_n_old;

```

```
est_v_b_n_new = est_v_b_n_old - x_est_new(4:6);  
est_L_b_new = est_L_b_old - (x_est_new(7));  
est_lambda_b_new = est_lambda_b_old - (x_est_new(8));  
est_h_b_new = est_h_b_old - x_est_new(9);  
% estimasi bias  
est_IMU_bias_new = est_IMU_bias_old + x_est_new(10:15);  
end
```



## **RIWAYAT HIDUP PENULIS**



Ananta Muhammad lahir pada 22 Juli 1997 di kota Baubau, Sulawesi Tenggara, merupakan anak pertama dari empat bersaudara dari pasangan Sahbudin dan Hariati. Setelah menyelesaikan pendidikan sekolah dasar pada tahun 2009 di SDN 3 Nganganaumala, kemudian melanjutkan ke sekolah menengah SMPN 4 BAUBAU pada tahun 2012 dan sekolah menengah atas di SMAN 1 Baubau dan lulus pada tahun 2015. Setelah itu penulis merantau ke kota pahlwan dengan melanjutkan pendidikannya di Institut Teknologi

Sepuluh Nopember, Surabaya di departemen Teknik Elektro, Fakultas Teknik Elektro. Penulis memiliki minat pada kontrol dan sistem membuat penulis memilih Teknik Sistem Pengaturan sebagai bidang studi. Hingga saat ini, penulis terus berusaha agar menjadi seorang yang bermanfaat bagi orang lain, bangsa dan negara.

*[Halaman ini sengaja dikosongkan]*