TUGAS AKHIR - EE 184801

OBSTACLE TRACKING PADA UNMANNED *SURFACE VEHICLE* MENGGUNAKAN FILTER KALMAN

Gaung Jagad
NRP 07111640000079

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

DEPARTEMEN TEKNIK ELEKTRO
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember
Surabaya 2020

Halaman ini sengaja dikosongkan

TUGAS AKHIR – EE 184801

OBSTACLE TRACKING PADA *UNMANNED SURFACE VEHICLE* MENGGUNAKAN FILTER KALMAN

Gaung Jagad
NRP 07111640000079

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

DEPARTEMEN TEKNIK ELEKTRO
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember
Surabaya 2020

Halaman ini sengaja dikosongkan

FINAL PROJECT – EE 184801

OBSTACLE TRACKING ON *UNMANNED SURFACE VEHICLE* USING KALMAN FILTER

Gaung Jagad
NRP 07111640000079

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

DEPARTMENT OF ELECTRICAL ENGINEERING
Faculty of Intelligent Electrical and Informatics Technology
Institut Teknologi Sepuluh Nopember
Surabaya 2020

Halaman ini sengaja dikosongkan

PERNYATAAN KEASLIAN
TUGAS AKHIR

Dengan ini menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya dengan judul "**Obstacle Tracking pada Unmanned Surface Vehicle Menggunakan Filter Kalman**" adalah hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diizinkan 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, Juli 2020

Gaung Jagad
NRP. 07111640000079

Halaman ini sengaja dikosongkan

OBSTACLE TRACKING* PADA U*NMANNED SURFACE VEHICLE
MENGGUNAKAN FILTER KALMAN

TUGAS AKHIR

Diajukan Guna Memenuhi Sebagian Persyaratan Untuk
Memperoleh Gelar Sarjana Teknik Elektro

Pada

Bidang Studi Teknik Sistem Pengaturan
Departemen Teknik Elektro
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember

Menyetujui:

Dosen Pembimbing I

Ir. Rusdhianto Effendi A.K., MT.
NIP. 195704241985021001

Surabaya,
Juli 2020

Halaman ini sengaja dikosongkan

OBSTACLE TRACKING* PADA U*NMANNED SURFACE VEHICLE
MENGGUNAKAN FILTER KALMAN

TUGAS AKHIR

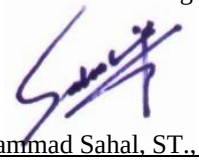
Diajukan Guna Memenuhi Sebagian Persyaratan Untuk
Memperoleh Gelar Sarjana Teknik Elektro

Pada

Bidang Studi Teknik Sistem Pengaturan
Departemen Teknik Elektro
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember

Menyetujui:

Dosen Pembimbing II



Mochammad Sahal, ST., M.Sc.

NIP. 197011191998021002

Surabaya,
Juli 2020

Halaman ini sengaja dikosongkan

***OBSTACLE TRACKING* PADA *UNMANNED SURFACE VEHICLE* MENGGUNAKAN FILTER KALMAN**

Gaung Jagad
07111640000079

Dosen Pembimbing : 1. Ir. Rusdhianto Effendi A.K., M.T.
2. Mochammad Sahal, ST., M.Sc.

ABSTRAK

Unmanned Surface Vehicle (USV) merupakan salah satu kendaraan tanpa awak yang beroperasi di atas permukaan air. USV dapat dikendalikan secara manual atau diberikan panduan dari jarak jauh atau beroperasi sendiri secara otonom. Agar dapat beroperasi secara otonom, USV memiliki serangkaian sistem pemanduan yang berfungsi untuk merencanakan lintasan agar kapal dapat mencapai titik tujuan. Kemampuan untuk mendeteksi halangan pada lintasan USV merupakan salah satu faktor yang krusial untuk merencanakan lintasan baru agar dapat menghindari halangan serta mencapai tujuan secara optimal. Sensor yang digunakan oleh USV untuk mendeteksi halangan dapat berupa kamera dan LiDar untuk mendeteksi posisi halangan pada lingkungan sekitar kapal. Pada penelitian ini dirancang sistem *tracking* halangan yang terdeteksi dengan mengintegrasikan sensor kamera dan LiDar untuk mengestimasi posisi relatif halangan terdeteksi dari kapal. Sistem yang dirancang juga mengestimasi lebar dari halangan yang terdeteksi. Informasi hasil *tracking* halangan pada lingkungan sekitar kapal selanjutnya dapat digunakan untuk menghindari halangan pada lintasan kapal. Untuk meningkatkan akurasi estimasi posisi relatif halangan dari kapal, digunakan filter Kalman untuk mengurangi *noise* pada pengukuran. Kemudian hasil rancangan sistem disimulasikan menggunakan software MATLAB sehingga dapat dianalisa performa dari sistem yang dirancang. Didapatkan hasil filter Kalman mengurangi noise pengukuran sebesar 12% dengan matriks kovarian $\boldsymbol{Q} = [2\ 0\ 0\ 0;\ 0\ 20\ 0\ 0;\ 0\ 0\ 2\ 0;\ 0\ 0\ 0\ 20]$ dan $\boldsymbol{R} = [1\ 0\ 0\ 0;\ 0\ 10\ 0\ 0;\ 0\ 0\ 1\ 0;\ 0\ 0\ 0\ 10]$.

Kata Kunci : *Unmanned Surface Vehicle*, *Obstacle Tracking*, filter Kalman.

Halaman ini sengaja dikosongkan

OBSTACLE TRACKING ON UNMANNED SURFACE VEHICLE USING KALMAN FILTER

Gaung Jagad
07111640000079

Supervisor : 1. Ir. Rusdhianto Effendi A.K., M.T.
2. Mochammad Sahal, ST., M.Sc.

ABSTRACT

Unmanned Surface Vehicles (USV) are one of the unmanned vehicles that operates on the water surface. USV can either be remote controlled or remote guided, or operating autonomously on their own. In order to be operated autonomously, USV has guidance system designed for path planning to reach its destination. The ability to detect obstacles in its paths is one of the important factors to plan a new path in order to avoid obstacles and reach its destination optimally. Sensors used by USV to detect obstacles consists of cameras and LiDars to locate the position of obstacles in the environment around the ship. This research will design an obstacle tracking system which is designed by integrating the sensors, camera and LiDar to estimate the relative position of the obstacle to the ship. The system designed will also estimate the width of the detected obstacle. Informations acquired from obstacle tracking on the environment around the ship can be used to avoid obstacles in its path. To improve the relative posisition estimation of the obstacles to the ship, a Kalman filter will be applied to reduce the measurements noises. Then the results of the system designed will be simulated using MATLAB software so that results can be analyzed to see the performance of the system designed. Results obtained using Kalman filter shows 12% noise reduction with covariance matrice $\boldsymbol{Q}$ = [2 0 0 0; 0 20 0 0; 0 0 2 0; 0 0 2 0] and $\boldsymbol{R} = [1\ 0\ 0\ 0;\ 0\ 10\ 0\ 0;\ 0\ 0\ 1\ 0;\ 0\ 0\ 0\ 10]$.

Keywords: Unmanned Surface Vehicle, Obstacle Tracking, Kalman filter.

Halaman ini sengaja dikosongkan

KATA PENGANTAR

Segala puji bagi Allah ﷻ. Shalawat serta salam semoga selalu dilimpahkan kepada Nabi Muhammad ﷺ, beserta keluarga, para sahabatnya, beserta semua orang yang mengikutinya dengan baik hingga akhir zaman. Dengan rahmat, taufiq, dan pertolongan Allah ﷻ, penulis dapat menyelesaikan tugas akhir yang berjudul "***Obstacle Tracking* pada U*nmanned Surface Vehicle* Menggunakan Filter Kalman**" untuk memenuhi syarat kelulusan pada Bidang Studi Teknik Sistem Pengaturan Departemen Teknik Elektro Institut Teknologi Sepuluh Nopember (ITS), Surabaya.

Penulis ingin mengucapkan terima kasih kepada kedua orangtua yang selalu sabar dan mendukung penulis untuk menyelesaikan pendidikan sarjana. Tidak lupa penulis ucapkan terima kasih kepada seluruh dosen yang telah membimbing penulis selama masa perkuliahan di Departemen Teknologi Elektro, terutama kepada Bapak Rushdianto Effendi A.K. dan Bapak Mochammad Sahal sebagai dosen pembimbing penulis. Tugas akhir ini juga dapat diselesaikan tidak lain karena bantuan dari teman-teman dari Bidang Studi Sistem pengaturan sehingga penulis dengan mengucapkan kasih yang sebesar-besarnya kepada semuanya terutama kepada teman-teman seperjuangan angkatan e56.

Penulis menyadari bahwa masih banyak kekurangan dari laporan tugas akhir ini baik dari segi materi maupun teknis penyajiannya, mengingat masih kurangnya pengetahuan dan pengalaman penulis. Akhir kata penulis terbuka untuk segala masukan dan penulis berharap tugas akhir ini dapat bermanfaat untuk penelitian lebih lanjut.

Surabaya, Juni 2020

Penulis

Halaman ini sengaja dikosongkan

DAFTAR ISI

PERNYATAAN KEASLIAN TUGAS AKHIR VII

ABSTRAK xi

ABSTRACT XV

KATA PENGANTAR XVII

DAFTAR ISI XIX

DAFTAR GAMBAR XXI

DAFTAR TABEL XXV

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 Metode Penelitian 2

1.6 Sistematika Penulisan 3

1.7 Relevansi 4

BAB 2 TEORI PENUNJANG 5

2.1 Unmanned Surface Vehicle (USV) 5

2.2 Kamera Digital 7

2.3 LiDar 7

2.4 *Object Detection* 8

2.5 *Artificial Neural Network* 8

2.5.1 *Convolutional Neural Network* 8

2.5.2 YOLO 9

2.6 Obstacle Tracking 11

2.7 Filter Kalman 11

2.8 Root Mean Square Error 13

BAB 3 SISTEMATIKA PENGOLAHAN DATA 15

3.1 Deteksi Halangan oleh Kamera 15

3.1.1 Pengambilan Data Kamera 15

3.1.2 Hasil Deteksi Halangan pada Kamera 16

3.2 Deteksi Halangan oleh LiDar 17

3.2.1 Penerimaan Data oleh LiDar 17

3.2.2 Pembacaan Data LiDar 18

3.3 Integrasi Kamera dan LiDar 19
3.3.1 Konsep Integrasi Kamera dan LiDar 19
3.3.2 Transformasi Perhitungan Sudut Dan Jarak 20
3.3.3 Mengukur Lebar Aktual Objek 21
3.4 Tracking Objek atau Halangan 23
3.4.1 Metode Tracking Objek atau Halangan 23
3.5 Perancangan Filter Kalman 24
BAB 4 HASIL DAN ANALISA DATA 27
4.1 Hasil Simulasi *Obstacle Tracking* 27
4.1.1 Hasil Simulasi dengan Satu Halangan 27
4.1.2 Hasil Simulasi dengan Dua Halangan 30
4.1.3 Hasil Simulasi dengan Tiga Halangan 34
4.1.4 Hasil Simulasi dengan Empat Halangan 38
4.1.5 Hasil Simulasi dengan Lima Halangan 44
4.2 51
4.3 Analisa Eror Jarak Halangan 51
4.3.1 Eror Jarak Halangan pada Simulasi dengan Satu Halangan 51
4.3.2 Eror Jarak Halangan pada Simulasi dengan Dua Halangan52
4.3.3 Eror Jarak Halangan pada Simulasi dengan Tiga Halangan 54
4.3.4 Eror Jarak Halangan pada Simulasi dengan Empat Halangan 56
4.3.5 Eror Jarak Halangan pada Simulasi dengan Lima Halangan 58
BAB 5 KESIMPULAN DAN SARAN 63
5.1 Kesimpulan 63
5.1 Saran 63
DAFTAR PUSTAKA 65
LAMPIRAN 66
RIWAYAT HIDUP PENULIS 76

DAFTAR GAMBAR

Gambar 2.1 Skema USV dalam gerak planar 6
Gambar 2.2 (a) Cahaya biasa (b) Cahaya koheren 7
Gambar 2.3 Penerapan Algoritma YOLO 10
Gambar 2.4 Bentuk Jaringan dari CNN pada metode YOLO 10
Gambar 2.5 Diagram blok *obstacle tracking* 11
Gambar 3.1 Struktur sensor kamera dan LiDar 15
Gambar 3.2 Spesifikasi output gambar dari sensor kamera 16
Gambar 3.3 Hasil pengolahan Data Gambar oleh YOLO 17
Gambar 3.4 Ilustrasi transformasi perhitungan jarak dan sudut 20
Gambar 3.5 Ilustrasi sudut pandang kamera 22
Gambar 3.6 Ilustrasi *Tracking* Objek 24
Gambar 4.1 Plot lintasan simulasi dengan satu halangan 28
Gambar 4.2 Plot jarak relatif halangan terhadap kapal 29
Gambar 4.3 Plot sudut relatif halangan terhadap kapal 29
Gambar 4.4 Plot lintasan simulasi dengan dua halangan 31
Gambar 4.5 Plot sudut relatif halangan terhadap kapal pada halangan satu 31
Gambar 4.6 Plot sudut relatif halangan terhadap kapal pada halangan satu 32
Gambar 4.7 Plot jarak relatif halangan terhadap kapal pada halangan dua 32
Gambar 4.8 Plot sudut relatif halangan terhadap kapal pada halangan dua 33
Gambar 4.9 Plot lintasan simulasi dengan tiga halangan 34
Gambar 4.10 Plot jarak relatif halangan terhadap kapal pada halangan satu 35
Gambar 4.11 Plot sudut relatif halangan terhadap kapal pada halangan satu 36
Gambar 4.12 Plot jarak relatif halangan terhadap kapal pada halangan dua 36
Gambar 4.13 Plot sudut relatif halangan terhadap kapal pada halangan dua 37
Gambar 4.14 Plot jarak relatif halangan terhadap kapal pada halangan tiga 37
Gambar 4.15 Plot sudut relatif halangan terhadap kapal pada halangan tiga 38

Gambar 4.16 Plot lintasan simulasi dengan empat halangan. 39
Gambar 4.17 Plot jarak relatif halangan terhadap kapal pada halangan satu 40
Gambar 4.18 Plot sudut relatif halangan terhadap kapal pada halangan satu 40
Gambar 4.19 Plot jarak relatif halangan terhadap kapal pada halangan dua 41
Gambar 4.20 Plot sudut relatif halangan terhadap kapal pada halangan dua 41
Gambar 4.21 Plot jarak relatif halangan terhadap kapal pada halangan tiga 42
Gambar 4.22 Plot sudut relatif halangan terhadap kapal pada halangan tiga..... 42
Gambar 4.23 Plot jarak relatif halangan terhadap kapal pada halangan empat..... 43
Gambar 4.24 Plot sudut relatif halangan terhadap kapal pada halangan empat..... 43
Gambar 4.25 Plot lintasan simulasi dengan lima halangan 45
Gambar 4.26 Plot jarak relatif halangan terhadap kapal pada halangan satu 46
Gambar 4.27 Plot sudut relatif halangan terhadap kapal pada halangan satu 46
Gambar 4.28 Plot jarak relatif halangan terhadap kapal pada halangan dua 47
Gambar 4.29 Plot sudut relatif halangan terhadap kapal pada halangan dua 47
Gambar 4.30 Plot jarak relatif halangan terhadap kapal pada halangan tiga 48
Gambar 4.31 Plot sudut relatif halangan terhadap kapal pada halangan tiga..... 48
Gambar 4.32 Plot jarak relatif halangan terhadap kapal pada halangan empat..... 49
Gambar 4.33 Plot sudut relatif halangan terhadap kapal pada halangan empat..... 49
Gambar 4.34 Plot jarak relatif halangan terhadap kapal pada halangan lima 50
Gambar 4.35 Plot sudut relatif halangan terhadap kapal pada halangan lima 50

Gambar 4.36 Plot perbandingan nilai eror jarak halangan pada halangan satu 52
Gambar 4.37 Plot perbandingan nilai eror jarak halangan pada halangan satu 53
Gambar 4.38 Plot perbandingan nilai eror jarak halangan pada halangan dua 53
Gambar 4.39 Plot perbandingan nilai eror jarak halangan pada halangan satu 54
Gambar 4.40 Plot perbandingan nilai eror jarak halangan pada halangan dua 55
Gambar 4.41 Plot perbandingan nilai eror jarak halangan pada halangan tiga 55
Gambar 4.42 Plot perbandingan nilai eror jarak halangan pada halangan satu 56
Gambar 4.43 Plot perbandingan nilai eror jarak halangan pada halangan dua 57
Gambar 4.44 Plot perbandingan nilai eror jarak halangan pada halangan tiga 57
Gambar 4.45 Plot perbandingan nilai eror jarak halangan pada halangan empat 58
Gambar 4.46 Plot perbandingan nilai eror jarak halangan pada halangan satu 59
Gambar 4.47 Plot perbandingan nilai eror jarak halangan pada halangan dua 59
Gambar 4.48 Plot perbandingan nilai eror jarak halangan pada halangan tiga 60
Gambar 4.49 Plot perbandingan nilai eror jarak halangan pada halangan empat 60
Gambar 4.50 Plot perbandingan nilai eror jarak halangan pada halangan lima 61

Halaman ini sengaja dikosongkan

DAFTAR TABEL

Tabel 3.1 Paket data hasil deteksi halangan pada kamera 16
Tabel 3.2 Deskripsi paket data LiDar 18
Tabel 3.3 Keterangan parameter dan variabel pada ilustrasi perhitungan jarak dan sudut 21
Tabel 4.1 Keterangan variasi nilai kovarians Q dan R 27
Tabel 4.2 Perbandingan RMSE hasil filter Kalman dan pengukuran pada simulasi dengan satu halangan 30
Tabel 4.3 Perbandingan RMSE hasil filter Kalman dan pengukuran pada simulasi dengan dua halangan 33
Tabel 4.4 Perbandingan RMSE hasil filter Kalman dan pengukuran pada simulasi dengan tiga halangan 38
Tabel 4.5 Perbandingan RMSE hasil filter Kalman dan pengukuran pada simulasi dengan empat halangan 44
Tabel 4.6 Perbandingan RMSE hasil filter Kalman dan pengukuran pada simulasi dengan lima halangan 51
Tabel 4.7 Perbandingan rata – rata eror jarak halangan pada simulasi dengan satu halangan 51
Tabel 4.8 Perbandingan rata – rata eror jarak halangan pada simulasi dengan dua halangan 52
Tabel 4.9 Perbandingan rata – rata eror jarak halangan pada simulasi dengan tiga halangan 54
Tabel 4.10 Perbandingan rata – rata eror jarak halangan pada simulasi dengan empat halangan 56
Tabel 4.11 Perbandingan rata – rata eror jarak halangan pada simulasi dengan lima halangan 58

Halaman ini sengaja dikosongkan

BAB 1
PENDAHULUAN

Dalam bab ini dipaparkan tentang latar belakang penelitian, permasalahan yang akan diselesaikan, tujuan dari penelitian, metodologi yang digunakan, sistematika penulisan serta relevansi penelitian ini terhadap penelitian di masa mendatang.

1.1 Latar Belakang

Teknologi terkini telah mendukung perkembangan teknologi dalam bidang *unmanned system* atau lebih umum dikenal dengan *drone*. Teknologi ini tersedia dalam berbagai macam jenis yang dapat digunakan untuk memenuhi berbagai macam keperluan, terutama dalam dunia militer. *Unmanned Surface Vehicle* (USV) adalah salah satu *drone* berbasis laut yang umumnya digunakan untuk mengambil data dan melakukan pengawasan di wilayah laut. USV dapat dioperasikan secara manual atau dipandu dari jarak jauh, atau beroperasi secara otomatis dengan sendirinya.

Sistem pemanduan pada USV memiliki kemampuan untuk merencanakan lintasan agar dapat mencapai titik tujuan. Salah satu unsur penting agar sebuah *unmanned surface vehicle* (USV) dapat mengambil tindakan yang dibutuhkan secara otomatis adalah dengan mengetahui rintangan yang dihadapi. Sensor pada USV memiliki peran untuk mendeteksi rintangan yang berada dalam jarak pandang USV. Rintangan yang terdeteksi oleh sensor menjadi acuan dari perubahan lintasan USV untuk menghindari tabrakan. Dengan bantuan deteksi halangan, sistem pemanduan USV akan merencanakan lintasan baru namun tetap menjaga efisiensi untuk mencapai titik tujuan. Karena USV memiliki kecepatan yang tinggi, keputusan dalam merencanakan lintasan baru untuk menghindari halangan harus diambil dengan cepat. Maka dibutuhkan sebuah metode estimasi dari data pengukuran untuk memperkirakan posisi halangan pada lintasan USV.

Pada tugas akhir ini dirancang sebuah filter yang dapat mengurangi eror acak sehingga dapat meningkatkan performa pengukuran. Filter Kalman dikenal sebagai filter yang handal dalam memfilter sinyal yang acak, dalam Tugas Akhir ini digunakan filter Kalman untuk mengestimasi eror acak dari sistem *tracking* menggunakan sensor kamera dan LiDar.

1.2 Perumusan Masalah

Masalah yang dibahas pada tugas akhir ini adalah kebutuhan USV terhadap sebuah pemodelan untuk mengestimasi posisi relatif halangan terhadap kapal pada lingkungan lintasan kapal. Sehingga didapatkan lintasan pergerakan setiap halangan terdeteksi. Hasil data pengukuran diasumsikan mengandung gangguan eksternal yang menyebabkan posisi relatif halangan terdeteksi dari kapal dapat mengalami kesalahan yang acak.

1.3 Batasan Masalah

Perlu diperhatikan batasan dalam melakukan penelitian ini, antara lain sebagai berikut:

a. Data yang digunakan adalah data simulasi.
b. Simulasi dijalankan menggunakan software MATLAB.
c. *USV* dan *Obstacle* dinamis memiliki kecepatan konstan.
d. Noise pada data simulasi adalah white noise.
e. Estimasi lebar halangan diasumsikan konstan.

1.4 Tujuan Penelitian

Pada penelitian ini, dirancang sebuah sistem untuk mengestimasi lintasan dari setiap halangan terdeteksi. informasi jarak, sudut, dan lebar dari halangan terdeteksi didapatkan dengan menggunakan kamera dan LiDar sebagai sensor. Filter Kalman digunakan untuk mengestimasi jarak dan sudut dengan mengeliminasi kesalahan acak dari data pengukuran.

1.5 Metode Penelitian

Penelitian ini dilaksanakan dengan melalui beberapa tahap proses yang telah dirancang sebagai berikut :

a. Studi Literatur
   Studi literatur dilakukan untuk mengumpulkan serta mempelajari teori, data dan penelitian yang dianggap relevan dan terpercaya untuk mendukung keabsahan tugas akhir ini. Literatur yang digunakan memiliki batasan-batasan tertentu. Yaitu, literatur yang digunakan harus bersumber dari paper, jurnal, buku, maupun artikel yang berasal dari badan pemerintahan atau institusi akademik terpercaya.

b. Perancangan Sistem Obstacle Tracking
   Merupakan tahap dimana dirancang sistem untuk Obstacle Tracking yang di dalamnya terdapat beberapa proses dan tahapan

untuk mengolah data pengukuran dari kamera dan LiDar. Pada proses tertentu didapatkan model pengukuran yang digunakan pada filter Kalman.

c. Perancangan Filter Kalman
Pada tahap ini setelah mendapatkan model sistem dari data pengukuran, maka pemodelan sistem hasil pengukuran dapat digunakan pada model sistem untuk filter Kalman. Filter Kalman merupakan filter yang membutuhkan model dari sistem.

d. Pengujian Sistem *Obstacle Tracking*
Setelah perancangan sistem obstacle tracking serta memodelkan sistem, selanjutnya dilakukan pengujian hasil rancangan model tersebut. Pengujian dilakukan dengan data simulasi yang dijalankan pada software MATLAB. Keakuratan dari sistem diuji dengan membandingkan RMSE nilai estimasi dan nilai pengukuran terhadap nilai sebenarnya.

e. Analisa Hasil dan Evaluasi
Pada tahap ini, dilakukan analisa terhadap hasil yang didapatkan dari sistem yang telah dirancang dan disimulasikan pada tugas akhir ini. Sehingga didapatkan hasil estimasi posisi relatif halangan terhadap kapal yang diinginkan dengan kesalahan sudah memenuhi toleransi yang diizinkan. Lalu dilakukan evaluasi serta revisi rancangan sistem obstacle tracking apabila diperlukan.

1.6 Sistematika Penulisan

BAB 1 PENDAHULUAN

Dalam pendahuluan menjelaskan mengenai latar belakang, permasalahan, batasan masalah, tujuan, metodologi, sistematika penulisan, serta relevansi dari penelitian.

BAB 2 TEORI PENUNJANG

Bab ini membahas teori-teori yang menunjang penelitian, diantaranya menjelaskan mengenai USV beserta sensor yang digunakan, algoritma untuk *obstacle tracking*, filter Kalman, dan perhitungan kesalahan dengan *Root Mean Square Error* (RMSE).

BAB 3 SISTEMATIKA PENGOLAHAN DATA

Bab ini menjelaskan mengenai perancangan sensor yang digunakan pada USV. Serta mengintegrasikan data yang didapatkan menggunakan sensor kamera dan LiDar. Kemudian

dijelaskan mengenai simulasi untuk menghasilkan bentuk data yang sama untuk diaplikasikan pada sistem hasil rancangan dengan filter Kalman.

BAB 4 HASIL DAN ANALISA DATA

Bab ini memuat hasil dan analisa data yang dilakukan dengan simulasi *obstacle tracking* menggunakan software MATLAB. Kemudian hasil estimasi filter Kalman dibandingkan dengan nilai pengukuran dan nilai sebenarnya.

BAB 5 KESIMPULAN DAN SARAN

Dalam Bab ini, dijelaskan kesimpulan terkait hasil simulasi yang telah dilakukan beserta saran yang dapat digunakan sebagai acuan pada penelitian terkait.

1.7 Relevansi

Sistem hasil rancangan pada tugas akhir ini diharapkan dapat diterapkan dan diaplikasikan pada USV untuk menghadapi permasalahan riil yang ada. Penelitian ini juga diharapkan dapat menjadi referensi bagi penelitian terkait *object detection* pada kendaraan tanpa awak pada umumnya maupun penelitian yang lebih rumit di masa depan.

BAB 2
TEORI PENUNJANG

Dalam bab ini dijelaskan mengenai teori dasar untuk membangun algoritma *obstacle tracking*. Beberapa teori pendukung lainnya dijelaskan dengan tujuan untuk memberikan pemahaman umum mengenai deteksi objek.

2.1 Unmanned Surface Vehicle (USV)

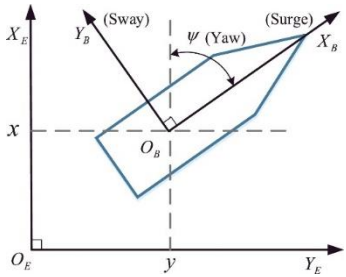
Unmanned surface vehicle (USV) adalah sebuah robot dalam bentuk kapal tanpa awak yang beroperasi di danau, pelabuhan, atau bahkan laut lepas. Kapal tanpa awak ini memiliki karakter ukurannya yang kecil, mobilitas yang tinggi, serta rendah biaya. Sistem NGC (*Navigation, Guidance, and Control*) pada USV mendukung kapal untuk beroperasi secara otomatis dengan mengendalikan kapal melalui lintasan yang telah direncanakan untuk mencapai lokasi tujuan, serta menghindari halangan yang terdapat pada jalurnya. Ketika navigasi, sensor yang digunakan untuk mendeteksi rintangan pada umumnya adalah kamera, LiDar, radar, atau kombinasi dengan sensor lainnya. Dalam hal ini, mendeteksi rintangan terbatas pada jarak puluhan hingga ratusan meter.

USV terintegrasi dengan berbagai macam sensor dan kontroler untuk menunjang kemampuan untuk beroperasi tanpa campur tangan manusia. USV juga dilengkapi dengan berbagai algoritma untuk banyak fungsi seperti *guidance*, *navigation*, dan *obstacle avoidance*. Namun bila dibutuhkan, kapal dapat dikendalikan secara manual dengan kendali jarak jauh.

Pemodelan USV dapat disederhanakan dalam tiga derajat kebebasan. Tujuan utamanya adalah untuk memastikan USV dapat mengikuti jalur yang diinginkan dengan akurat. Sehingga pemodelan USV dapat disederhanakan dengan hanya mempertimbangkan *surge*, *sway*, dan *heading*, dinamika *terkait roll*, *pitch*, dan *heave* dapat diabaikan untuk mempertahankan pemodelan yang sederhana. Pada Gambar 2.1, model kinematika USV tanpa gangguan dapat dinyatakan sebagai berikut :

$$\boldsymbol{\eta} = \boldsymbol{R}(\psi)\boldsymbol{\nu} \tag{2.1}$$

Dimana $\boldsymbol{\eta} = [x\ y\ \psi]^T$ menyatakan vektor posisi dan orientasi pada kerangka acuan bumi (earth-fixed) dan $\boldsymbol{\nu} = [u\ v\ r]^T$ menyatakan vektor kecepatan linear dan kecepatan sudut pada kerangka acuan badan kapal (*body-fixed).*



Gambar 2.1 Skema USV dalam gerak planar ($X_E O_E Y_E$ adalah kerangka acuan bumi, sedangkan $X_B O_B Y_B$ adalah kerangka acuan badan kapal)

$$\boldsymbol{R}(\psi) = \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.2}$$

$\boldsymbol{R}(\psi)$ menyatakan matriks transformasi antara kerangka acuan bumi dan kerangka acuan badan kapal. (x, y) koordinat kartesian dari titik tengah pusat massa USV dan ψ adalah orientasi (*yaw/heading angle)* dari USV pada kerangka acuan bumi, sedangkan u, v, dan r menyatakan kecepatan *surge*, *sway*, dan *yaw* pada kerangka acuan badan kapal.

Untuk model dinamika USV dapat dinyatakan dengan sederhana sebagai berikut :

$$\boldsymbol{M}\boldsymbol{v} + \boldsymbol{C}(\boldsymbol{v})\boldsymbol{v} + \boldsymbol{D}\boldsymbol{v} = \boldsymbol{\tau} \tag{2.3}$$

Dimana $\boldsymbol{\tau} = [\,\tau_u \; 0 \; \tau_r]^T$ menyatakan gaya dan momen yang bekerja pada kapal dengan τ_u dan τ_r menyatakan gaya *surge* dan momen *yaw* kapal, $\mathbf{M} = diag\{m_{11}\; m_{22}\; m_{33}\}$ menyatakan matriks sistem inertia dengan m_{11}, m_{22}, m_{33} sebagai nilai koefisien inersia dari USV, $\mathbf{D} = diag\{d_{11}\; d_{22}\; d_{33}\}$ dengan d_{11}, d_{22}, d_{33} menyatakan nilai koefisien damping dalam *surge*, *sway*, dan *yaw*.

$$\boldsymbol{C}(\boldsymbol{v}) = \begin{bmatrix} 0 & 0 & -m_{22}v \\ 0 & 0 & m_{11}u \\ m_{22}v & -m_{11}u & 0 \end{bmatrix} \tag{2.4}$$

$\boldsymbol{C}(\boldsymbol{v})$ menyatakan matriks *centripetal* dan *Coriolis*.

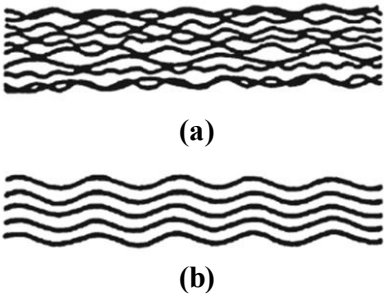
2.2 Kamera Digital

Kamera Digital adalah sebuah perangkat yang dapat menangkap cahaya menjadi sebuah gambar. Dilengkapi dengan sensor yang tersusun atas dioda foto, kamera digital mengubah partikel cahaya menjadi muatan listrik. Muatan listrik tersebut kemudian diubah menjadi tegangan yang selanjutnya dapat diproses menggunakan *analog to digital converter* (ADC) dari kamera. ADC adalah sebuah *chip* dalam kamera yang dapat mengklasifikasikan tegangan menjadi tingkatan kecerahan dan menetapkan setiap tingkatan kepada sebuah bilangan biner, yang terdiri dari satu dan nol. Sehingga menghasilkan gambar yang tersimpan dalam data digital.

2.3 LiDar

LiDar (*light detection and ranging*) adalah sebuah sensor yang bekerja seperti pencitraan radar. Dengan mentransmisikan energi dalam rentang frekuensi yang sempit, kemudian menerima kembali energi yang tersebar untuk mencitrakan lingkungan sekitar. LiDar adalah sensor aktif yang menyediakan sumber energinya sendiri. LiDar dapat membandingkan karakteristik dari energi yang ditrasnsmisikan dengan energi yang diterima, dalam hal panjang gelombang, sudut, dan lain-lain.

LiDar sebagai sensor yang berbasis laser, menggunakan cahaya yang koheren, yaitu cahaya yang tersusun atas panjang gelombang dengan pita yang sangat sempit. Dimana cahaya pada umumnya tersusun atas panjang gelombang yang beragam seperti ditunjukkan pada Gambar 2.2. Dengan kumpulan bentuk cahaya yang beragam, laser dapat menghasilkan cahaya yang koheren dan tersusun atas panjang gelombang dengan rentang yang sempit. Cahaya seperti ini dapat ditransmisikan dalam jarak yang jauh sebagai sinar yang sempit dengan penyimpangan yang kecil, dibandingkan dengan cahaya pada umumnya.



Gambar 2.2 (a) Cahaya biasa (b) Cahaya koheren

LiDar mengukur jarak ke objek pemantul dengan memancarkan pulsa sinar laser dan mengukur waktu antara emisi dan penerimaan sinar pantul. Interval waktu yang diukur kemudian dikonversi menjadi jarak. LiDar adalah alat yang ampuh untuk mengidentifikasi dan memetakan fitur-fitur dengan ekspresi topografi.

2.4 *Object Detection*

Object detection adalah sebuah teknologi yang berkaitan dengan disiplin ilmu *computer* vision untuk mengidentifikasi wilayah yang memiliki probabilitas adanya sebuah objek yang ada pada sebuah gambar. Algoritma *object detection* akan memberikan keluaran koordinat dari wilayah yang menjadi kandidat adanya sebuah objek pada gambar. Untuk dapat mendeteksi sebuah objek pada gambar, algoritma *object detection* dapat mendeteksi objek yang telah diketahui sebelumnya. Sehingga algoritma *object detection* memiliki dua kemampuan yaitu menentukan lokasi dari objek pada gambar serta mengklasifikasikan objek yang terdeteksi.

2.5 *Artificial Neural Network*

Artificial neural network adalah sebuah model elektronik yang mengikuti hipotesis struktur saraf otak manusia, yaitu belajar dari pengalaman. Model ini dikembangkan karena diketahui bahwa ada beberapa hal yang tidak dapat diselesaikan dengan tenaga yang ringan. Pemodelan otak atau sistem saraf tiruan ini juga dapat dikembangkan dengan cara yang lebih mudah walaupun memang membutuhkan waktu dan tenaga. Namun tentu saja sistem saraf tiruan ini masih sangat jauh dari aslinya, sehingga banyak ruang pengembangan untuk teknologi ini di masa yang akan datang.

2.5.1 *Convolutional Neural Network*

Convolutional Neural Network (ConvNet / CNN) adalah sebuah algoritma dari *neural network* yang terbukti efektif dalam *image processing*. CNN menerima gambar sebagai input, kemudian menetapkan nilai bobot dan bias untuk berbagai objek dalam gambar sehingga dapat membedakan satu objek dengan objek lain. CNN mengambil namanya dari operasi konvolusi, yaitu sebuah operasi matematis yang menghasilkan sebuah fungsi baru dari dua buah fungsi. Dalam hal ini CNN dapat dikatakan sebagai filter yang mendeteksi objek pada Gambar.

2.5.2 YOLO

You Only Look Once (YOLO) adalah salah satu algoritma *object detection* menggunakan *convolutional neural network.* Pada algoritma YOLO, satu buah *convolutional neural network* akan dijalankan pada sebuah Gambar, dalam serentak memberikan prediksi sejumlah *bounding boxes* dengan probabilitas klasifikasi dari objek yang berada dalam setiap *bounding box*. *Bounding box* adalah metode yang digunakan untuk merepresentasikan lokasi objek pada Gambar.

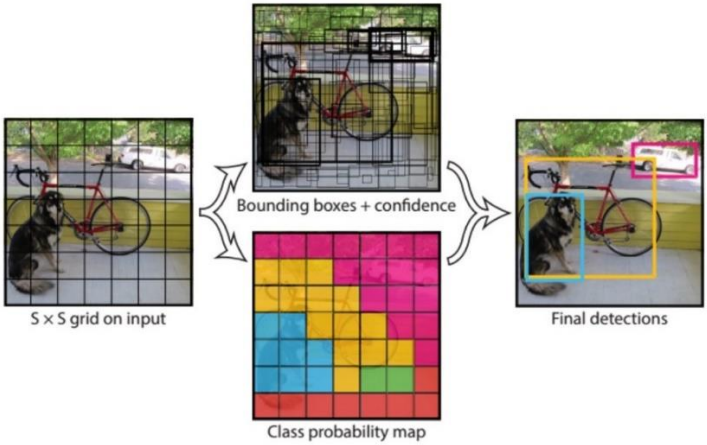
Algoritma *object detection* pada YOLO akan menghasilkan koordinat dari wilayah objek pada Gambar. Metode yang umum digunakan untuk menunjukan wilayah objek pada Gambar adalah dengan merepresentasikan wilayahnya dengan bounding box.

Bounding box dapat direpresentasikan dengan 4 elemen vektor dan biasanya memiliki keterangan confidence score yang menunjukan seberapa besar kemungkinannya mengandung sebuah objek. Sebuah bounding box menyimpan parameter sebagai berikut:

- X left: koordinat titik x pada bagian kiri dari bounding box
- Y top: koordinat titik y pada bagian atas dari bounding box
- X right: koordinat titik x pada bagian kanan dari bounding box
- Y bottom: koordinat titik y pada bagian bawah dari bounding box

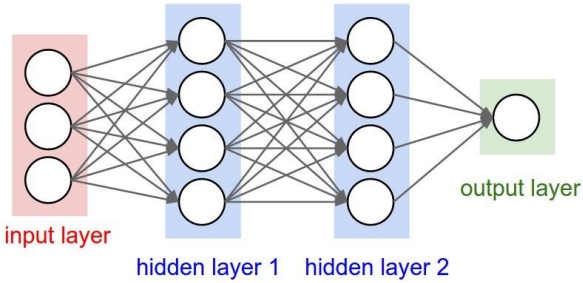
Algoritma YOLO membagi sebuah Gambar menjadi kotak dengan jumlah SxS, dimana setiap kotak memiliki sejumlah B bounding boxes. Untuk setiap bounding box, *neural network* akan memberikan keluaran probabillitas klasifikasi dari objek dalam bounding box. Bounding box yang memiliki probabilitas diatas nilai *threshold* akan dipilih menjadi wilayah suatu objek pada Gambar.

Dalam melakukan pendekatan object detection, langkah pertama adalah membuat model algoritma untuk menghasilkan wilayah yang diminati atau region proposal. Region proposals adalah kumpulan bounding box yang meliputi seluruh Gambar. Kemudian fitur visual diambil dari bounding box untuk ditentukan objek apa yang berada dalam kotak. Setelah itu bounding boxes yang tumpang tindih digabungkan menjadi satu bounding box, jika dinilai merupakan satu objek yang sama. Proses ini seperti ditunjukkan pada Gambar 2.3.



Gambar 2.3 Penerapan Algoritma YOLO

Convolutional neural network yang mendasari kemampuan untuk mendeteksi wilayah dan mengklasifikasikan semua objek yang terlihat pada Gambar. Agar algoritma object detection dapat mengetahui konten dari sebuah Gambar, diperlukan *training* pada algoritma untuk mempelajari perbedaan antar setiap kelas. Hasil training tersebut akan tersimpan dalam *hidden layer* sebagai tolok ukur CNN dalam mendeteksi objek pada Gambar seperti ditunjukkan pada Gambar 2.4. Dibutuhkan ratusan hingga ribuan Gambar yang mengandung objek X agar algoritma dapat mengenali objek X dengan baik. Algoritma *object detection* hanya dapat mengenali objek yang sudah dipelajarinya.



Gambar 2.4 Bentuk Jaringan dari CNN pada metode YOLO

2.6 Obstacle Tracking

Diagram blok dari skema *obstacle Tracking* ditunjukkan oleh Gambar 2.5. Tahapan pertama yang dilakukan pada *obstacle Tracking* adalah ekstraksi data dari sensor kamera dan LiDar. Kemudian akan didapatkan nilai koordinat *bounding box* untuk setiap halangan terdeteksi kamera dan jarak untuk setiap halangan terdeteksi LiDar. Karena hasil *bounding box* tidak memberikan informasi jarak halangan dengan kapal, maka jarak halangan untuk setiap *bounding box* diambil dengan mengintegrasikan jarak halangan yang terdeteksi oleh LiDar. Selanjutnya dibentuk matrix histori *tracking* halangan untuk setiap *time sampling*. Ketika matrix halangan sudah melebihi 2 *time sampling* maka posisi relatif halangan dapat diestimasi menggunakan filter Kalman. Sehingga diapatkan hasil estimasi posisi relatif halangan baru dengan noise yang lebih kecil.

Gambar 2.5 Diagram blok *obstacle tracking*

2.7 Filter Kalman

Filter Kalman adalah sebuah algoritma yang digunakan untuk memisahkan sinyal dari gangguan *noise* aditif. Dengan menggunakan konfigurasi state-space, filter Kalman memungkinkan skenario *multiple input/multiple output* (MIMO). Filter Kalman merupakan filter yang menggunakan metode *least square error* yang optimal dalam meminimumkan kovarian eror dari hasil estimasi. Filter Kalman memerlukan dua buah Persamaan matematika, yaitu model sistem dan model pengukuran dalam domain waktu diskrit.

$$\begin{aligned} \boldsymbol{x}_{k+1} &= \boldsymbol{\phi}_k \boldsymbol{x}_k + \boldsymbol{w}_k \\ \boldsymbol{z}_k &= \boldsymbol{H}_k \boldsymbol{x}_k + \boldsymbol{v}_k \end{aligned} \tag{2.5}$$

Di mana

$\boldsymbol{x}_k$: Vektor *state-space* dari proses pada waktu k

$\boldsymbol{\phi}_k$: Matriks transisi *state* antara $\boldsymbol{x}_k$ dengan $\boldsymbol{x}_{k+1}$

$\boldsymbol{w}_k$: Noise proses pada saat k yang diasumsikan *zero mean white noise*, dengan matriks kovarian

$$E[\boldsymbol{w}_k \boldsymbol{w}_i^T] = \begin{cases} 0 & i \neq k \\ \boldsymbol{Q}_k & i = k \end{cases} \tag{2.6}$$

$\boldsymbol{z}_k$: Vector output pengukuran pada saat k

$\boldsymbol{H}_k$: Matriks transisi *state* antara pengukuran $\boldsymbol{z}_k$ dengan vektor *state* $\boldsymbol{x}_k$,

$\boldsymbol{v}_k$: Noise pengukuran pada saat k yang diasumsikan zero mean white noise, dengan matriks kovarian

$$E[\boldsymbol{v}_k \boldsymbol{v}_i^T] = \begin{cases} 0 & i \neq k \\ \boldsymbol{R}_k & i = k \end{cases} \tag{2.7}$$

Sebagai inisialisasi, diasumsikan estimasi sementara dinotasikan dengan $\widehat{\boldsymbol{x}}_k^-$. sehingga estimasi eror dapat didefinisikan pada Persamaan (2.8).

$$\boldsymbol{e}_k^- = \boldsymbol{x}_k - \widehat{\boldsymbol{x}}_k^- \tag{2.8}$$

Dari Persamaan eror didapatkan matriks kovarian eror pada Persamaan (2.9).

$$\boldsymbol{P}_k^- = E[\boldsymbol{e}_k^- \boldsymbol{e}_k^{-T}] = E[(\boldsymbol{x}_k - \widehat{\boldsymbol{x}}_k^-)\,(\boldsymbol{x}_k - \widehat{\boldsymbol{x}}_k^-)^T] \tag{2.9}$$

Selanjutnya adalah menghubungkan propagasi estimasi *state* $\widehat{\boldsymbol{x}}_k^-$ dan data pengukuran $\boldsymbol{z}_k$ agar mendapatkan hasil estimasi yang optimal dengan Persamaan (2.10).

$$\widehat{\boldsymbol{x}}_k = \widehat{\boldsymbol{x}}_k^- + \boldsymbol{K}_k(\boldsymbol{z}_k - \boldsymbol{H}_k \widehat{\boldsymbol{x}}_k^-) \tag{2.10}$$

Di mana

$\widehat{\boldsymbol{x}}_k$: Estimasi terbaru

$\boldsymbol{K}_k$: Gain kalman

Gain Kalman digunakan untuk menghasilkan estimasi *state* yang optimal, nilai gain Kalman didapatkan dengan Persamaan (2.11).

$$\boldsymbol{K}_k = \boldsymbol{P}_k^- \boldsymbol{H}_k^T (\boldsymbol{H}_k \boldsymbol{P}_k^- \boldsymbol{H}_k^T + \boldsymbol{R}_k)^{-1} \tag{2.11}$$

Selanjutnya dicari nilai matriks kovarian eror terbaru dengan Persamaan sebagai berikut:

$$\boldsymbol{P}_k = (\boldsymbol{I} - \boldsymbol{K}_k \boldsymbol{H}_k)\boldsymbol{P}_k^- \tag{2.12}$$

Selanjutnya nilai estimasi *state* terbaru bisa langsung didapatkan dengan matriks transisi, karena $\boldsymbol{w}_k$ adalah *zero mean* dan tidak berkorelasi dengan vektor-vektor $\boldsymbol{w}_k$ yang lalu, maka $\boldsymbol{w}_k$ dapat diabaikan, Persamaan sebagai berikut:

$$\widehat{\boldsymbol{x}}_{k+1}^- = \boldsymbol{\phi}_k \widehat{\boldsymbol{x}}_k \tag{2.13}$$

Selanjutnya dihitung propagasi matriks kovarian eror terbaru dengan Persamaan sebagai berikut:

$$\boldsymbol{P}_{k+1}^- = \boldsymbol{\phi}_k \boldsymbol{P}_k \boldsymbol{\phi}_k^T + \boldsymbol{Q}_k \tag{2.14}$$

Sehingga didapatkan nilai yang dibutuhkan untuk mendapatkan nilai estimasi optimal pada waktu $k + 1$ dan pengukuran pada $\boldsymbol{z}_{k+1}$.

Bila dirangkum, maka proses perhitungan rekursif dari Kalman filter dengan mengikuti algoritma berikut:

1. Inisialisasi, saat $k = 0$, beri nilai awal pada state estimasi $\widehat{\boldsymbol{x}}_0^-$ dan matriks kovarian *eror* $\boldsymbol{P}_0^-$
2. Hitung nilai Kalman *gain* $\boldsymbol{K}_k$ dengan menggunakan Persamaan (2.11)
3. Perbarui estimasi $\widehat{\boldsymbol{x}}_k$ dengan nilai pengukuran $\boldsymbol{z}_k$ dengan menggunakan Persamaan (2.10)
4. Hitung matriks kovarian eror $\boldsymbol{P}_k$
5. Menyelesaikan perhitungan untuk prediksi *state* ke $k + 1$
6. Menghitung propagasi matriks kovarian eror $\boldsymbol{P}_{k+1}^-$
7. Untuk iterasi selanjutnya, mengulang kembali ke nomor 2.

2.8 Root Mean Square Error

Root Mean Square Error (RMSE) adalah sebuah metode untuk mengetahui standar deviasi dari nilai eror antara nilai estimasi dengan nilai sebenarnya. RMSE digunakan untuk menyatakan tingkat kemiripan antara nilai yang diinginkan dengan nilai sebenarnya. Semakin kecil nilai eror yang dihasilkan maka model sistem semakin mendekati dengan sistem sebenarnya. Karena selisih nilai estimasi dengan nilai sebenarnya

dapat berupa negatif maupun positif, maka nilai selisih dikuadratkan, kemudian dihitung nilai rata-ratanya, setelah itu diambil nilai akarnya sehingga didapatkan RMSE. Persamaan untuk mendapatkan nilai RMSE dinyatakan pada Persamaan (2.15).

$$RMSE = \sqrt{\left(\frac{\sum_{i=1}^{n} \hat{y}_i - y_i}{n}\right)} \tag{2.15}$$

di mana,
$\hat{y}_i$ = Nilai estimasi
y_i = Nilai sebenarnya
n = Banyaknya data

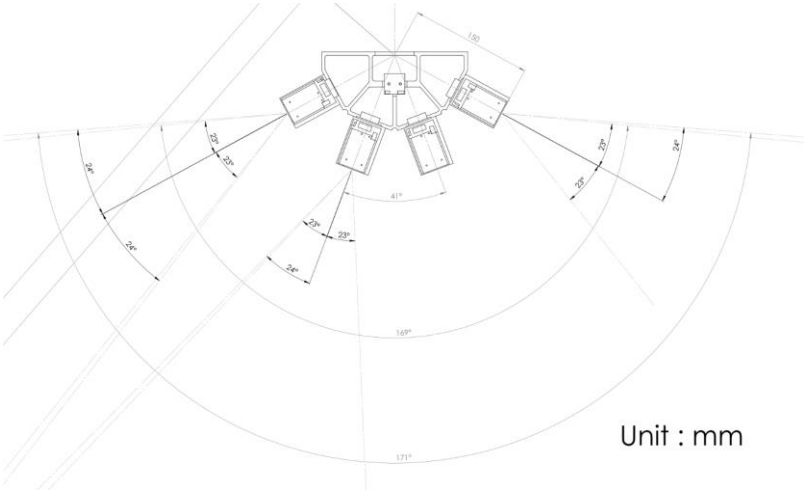
BAB 3
SISTEMATIKA PENGOLAHAN DATA

Bab ini menjelaskan mengenai sistem yang dibangun, seperti perancangan struktur sensor kamera dan LiDar, pengolahan integrasi data sensor kamera dan LiDar, Persamaan yang digunakan dalam sistem *tracking* halangan, serta estimasi menggunakan algoritma filter Kalman.

3.1 Deteksi Halangan oleh Kamera

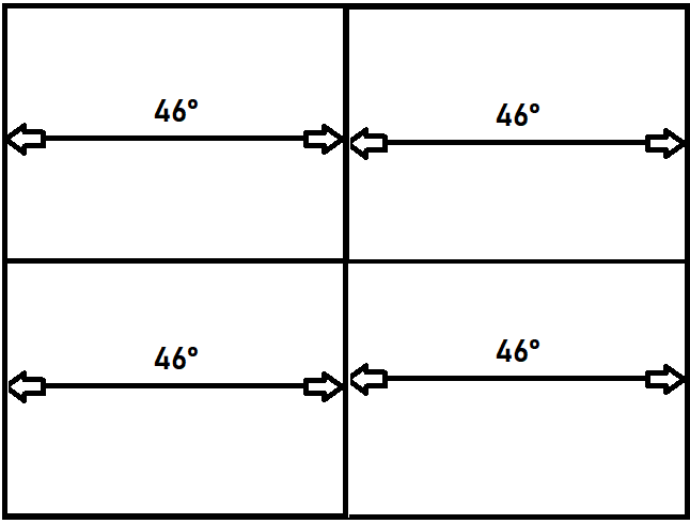
3.1.1 Pengambilan Data Kamera

Pengambilan Gambar dilakukan menggunakan 4 kamera sebagai sensor yang disusun membentuk jarak pandang selebar 169°, dengan spesifikasi setiap kamera memiliki jangkauan selebar 46°. Satu kamera dengan kamera sebelahnya terpisah sejauh 41° dari titik pusat. Ukuran Gambar yang dihasilkan kamera adalah 640 x 480 *pixel* untuk 4 Gambar, dengan ukuran setiap Gambar 320 x 240 *pixel*. Untuk ilustrasi struktur sensor kamera dan LiDar ditunjukkan pada Gambar 3.1



Gambar 3.1 Struktur sensor kamera dan LiDar

Spesifikasi Gambar yang diambil dari kamera yang digunakan memiliki jarak sebesar pandang 46°. Maka hasil data Gambar yang didapatkan ditunjukkan seperti pada Gambar 3.2. Dimana urutan kamera dari kanan ke kiri ditunjukkan dari kiri atas sampai kanan bawah.



Gambar 3.2 Spesifikasi output Gambar dari sensor kamera

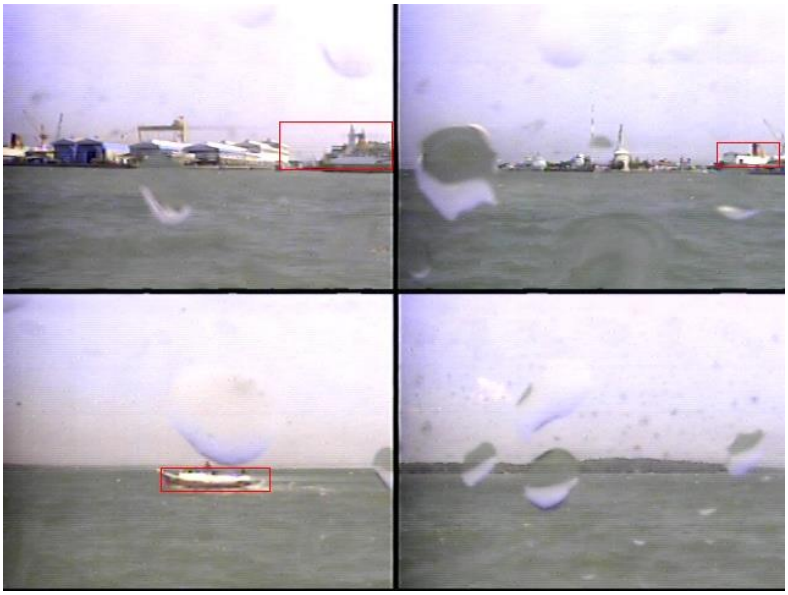
3.1.2 Hasil Deteksi Halangan pada Kamera

Output yang dihasilkan dari menggunakan algoritma YOLO adalah sebuah Gambar dengan objek yang ditandai dengan sejumlah *bounding box* sesuai dengan jumlah objek terdeteksi diatas *threshold* dan tabel data berisi informasi setiap *bounding box* objek terdeteksi dalam bentuk tabel. Kolom pada tabel data yang dihasilkan memberi informasi terkait timestamp, klasifikasi objek, *confidence score*, koordinat x kiri bawah bounding box, koordinat y atas bounding box, lebar bounding box, dan tinggi bounding box. Contoh data tabel yang didapatkan seperti yang ditunjukkan pada Tabel 3.1.

Tabel 3.1 Paket data hasil deteksi halangan pada kamera

Time stamp	Kelas	Confidence Score	Koordinat X kiri	Koordinat Y atas	Lebar	Tinggi
380268845216	14	0.334771	586	346	44	36
380367237216	0	0.345738	55	124	21	12

Adapun *output* berupa Gambar dari YOLO ditunjukkan pada Gambar 3.3



Gambar 3.3 Hasil pengolahan Data Gambar oleh YOLO

3.2 Deteksi Halangan oleh LiDar

3.2.1 Penerimaan Data oleh LiDar

Untuk mengumpulkan data, sensor LiDAR dipasang menumpuk pada sensor kamera untuk mendapatkan jarak pandang yang sama, walaupun LiDar yang digunakan memiliki sudut jarak pandang yang sedikit lebih besar daripada kamera yang digunakan. LiDar memancarkan pulsa laser inframerah menuju arah sisi depan kapal. LiDar akan memancarkan sinar laser dari *transmitter* dengan batas jarak pandang 80 meter dari kapal. Kemudian jika sinar dipantulkan dari objek terdeteksi, LiDar dapat mengetahui waktu yang telah berlalu sejak gelombang dipancarkan sampai kembali pada *receptor*. Sehingga dapat diketahui jarak yang ditempuh gelombang sejak dipancarkan hingga kembali dengan mengkonversikan waktu dengan kecepatan sinar yang dipancarkan.

3.2.2 Pembacaan Data LiDar

Data yang didapatkan dari LiDar diproses sehingga didapatkan data yang jelas dalam tabel dengan format csv, dimana spesifikasi data dalam tabel dapat dilihat pada Tabel 3.2,

Tabel 3.2 Deskripsi paket data LiDar

Kolom	Deskripsi
A	Timestamp
B	LiDar ke-1
C	Jumlah objek yang terdeteksi pada LiDar ke-1
D	*Beam* ke-1 pada LiDar ke-1
E	Jumlah objek yang terdeteksi oleh *beam* ke-1 pada LiDar ke-1
F, G, H, I, J	Jarak objek yang terdeteksi (jika tidak ada objek nilainya 0)
K	*Beam* ke-2 pada LiDar ke-1
L	Jumlah objek yang terdeteksi oleh *beam* ke-2 pada LiDar ke-1
M, N, O, P, Q	Jarak objek yang terdeteksi (jika tidak ada objek nilainya 0)
R	*Beam* ke-3 pada LiDar ke-1
S	Jumlah objek yang terdeteksi oleh *beam* ke-3 pada LiDar ke-1
T, U, V, W, X	Jarak objek yang terdeteksi (jika tidak ada objek nilainya 0)
Y	*Beam* ke-4 pada LiDar ke-1
Z	Jumlah objek yang terdeteksi oleh *beam* ke-4 pada LiDar ke-1
AA, AB, AC, AD, AE	Jarak objek yang terdeteksi (jika tidak ada objek nilainya 0)
AF	*Beam* ke-5 pada LiDar ke-1
AG	Jumlah objek yang terdeteksi oleh *beam* ke-5 pada LiDar ke-1
AH, AI, AJ, AK, AL	Jarak objek yang terdeteksi (jika tidak ada objek nilainya 0)
AM	*Beam* ke-6 pada LiDar ke-1

Kolom	Deskripsi
AN	Jumlah objek yang terdeteksi oleh *beam* ke-6 pada LiDar ke-1
AO, AP, AQ, AR, AS	Jarak objek yang terdeteksi (jika tidak ada objek nilainya 0)
AT	*Beam* ke-7 pada LiDar ke-1
AU	Jumlah objek yang terdeteksi oleh *beam* ke-7 pada LiDar ke-1
AV, AW, AX, AY, AZ	Jarak objek yang terdeteksi (jika tidak ada objek nilainya 0)
BA	*Beam* ke-8 pada LiDar ke-1
BB	Jumlah objek yang terdeteksi oleh *beam* ke-8 pada LiDar ke-1
BC,BD,BE,BF, BG	Jarak objek yang terdeteksi (jika tidak ada objek nilainya 0)
BH	LiDar ke-2
untuk selanjutnya mengikuti urutan yang sama seperti kolom C s.d BG hingga akhir LiDar ke-4	

3.3 Integrasi Kamera dan LiDar

3.3.1 Konsep Integrasi Kamera dan LiDar

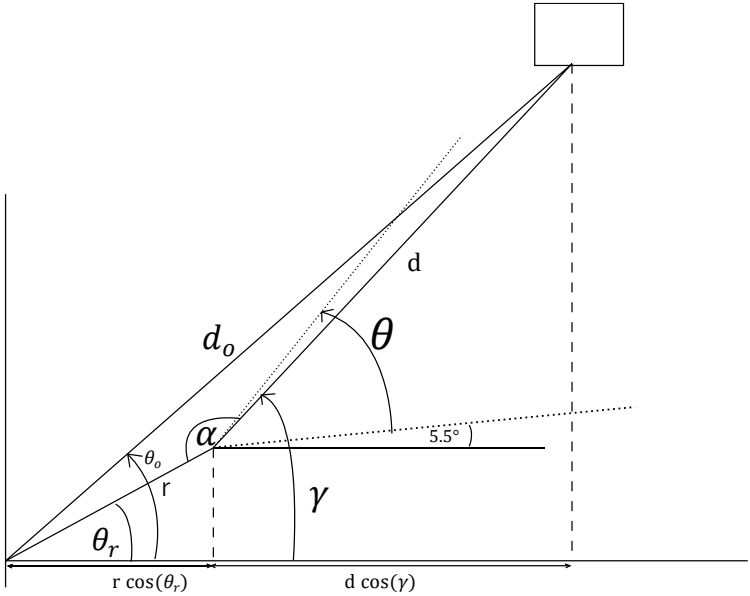
Pada data LiDar, diketahui pembacaan data setiap LiDar dibagi menjadi 8 segmen. Sehingga dilakukan pembagian wilayah pada data Gambar untuk setiap segmen, agar diketahui objek yang terdeteksi kamera terdapat pada segmen ke-berapa. Kemudian diambil jarak minimum pada segmen terkait untuk diasumsikan menjadi jarak objek terdeteksi terhadap kapal.

Wilayah koordinat pada Gambar dibagi menjadi 8, namun ternyata diketahui lebar wilayah setiap segmen pada LiDar adalah tidak sama. Wilayah segmen pada bagian tepi Gambar lebih lebar dari wilayah segmen pada bagian tengah. Dengan luas wilayah secara berurutan dari kiri ke kanan dalam pixel adalah 50, 42, 36, 32, 32, 36, 42, 50.

Dengan mengaplikasikan konsep ini, jarak objek terdeteksi kamera akan diketahui jika pada segmen terkait terdeteksi objek pada LiDar. Koordinat objek terdeteksi pada Gambar yang didapatkan dari *bounding box* dapat diubah menjadi sudut yang kemudian digunakan untuk menentukan posisi objek masuk dalam segmen terkait dari pembacaan sensor LiDar.

3.3.2 Transformasi Perhitungan Sudut Dan Jarak

Diketahui bahwa struktur sensor kamera terdiri dari 4 kamera yang tersusun membentuk sudut pandang 169°. Dengan sedikit perpotongan pada sisi pinggirnya. Sehingga untuk mendapatkan hasil posisi objek yang akurat, informasi nilai sudut setiap objek terdeteksi kamera akan digabungkan menjadi satu dihitung dari titik pusat untuk semua kamera. Dimisalkan Objek terdeteksi pada kamera 1 diilustrasikan seperti pada Gambar 3.4.



Gambar 3.4 Ilustrasi transformasi perhitungan jarak dan sudut

Dimana d adalah jarak objek terhadap kamera, d_o adalah jarak objek terhadap titik pusat sensor kamera, r adalah jarak dari titik pusat sampai titik tengah kamera, θ adalah sudut jangkauan kamera, θ_r adalah sudut dari titik pusat sampai titik tengah kamera pertama. Keterangan parameter dan variabel pada ilustrasi perhitungan jarak dan sudut dipaparkan pada Tabel 3.3.

Tabel 3.3 Keterangan parameter dan variabel pada ilustrasi perhitungan jarak dan sudut

Simbol	Nilai	Satuan
r	150	*mm*
θ_r	28.5	*deg*
θ	46	*deg*
γ	$\theta + 5.5$	*deg*
α	180 - γ + θ_r	*deg*

Berdasarkan ilustrasi pada Gambar 3.4 didapatkan Persamaan untuk mencari jarak objek menjadi satu terukur dari titik pusat dengan menggunakan Persamaan (3.2).

$$d_o^2 = r^2 + d^2 - 2.r.d.\cos(\alpha) \tag{3.1}$$

$$d_o = \sqrt{r^2 + d^2 - 2.r.d.cos(180° - \theta - 5.5° + \theta_r)} \tag{3.2}$$

Sedangkan untuk mencari sudut objek menjadi satu terukur dari titik pusat didapatkan Persamaan (3.4).

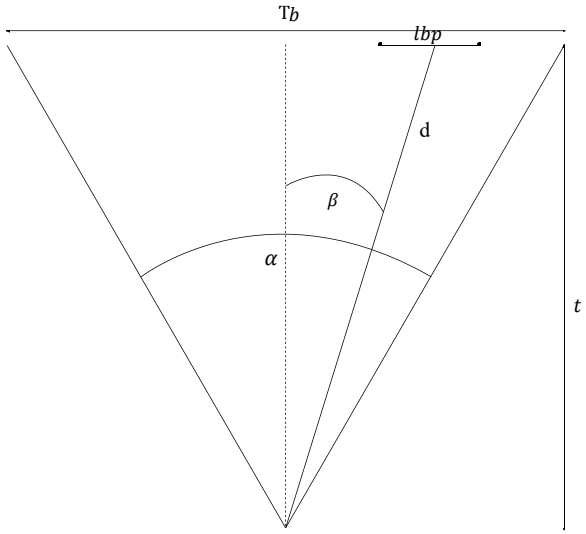
$$Cos\ (\theta_o) = \frac{(r.cos(\theta_r) + d.cos(\gamma))}{d_o} \tag{3.3}$$

$$\theta_o = \cos^{-1}\frac{(r.cos(\theta_r) + d.cos(\gamma))}{d_o} \tag{3.4}$$

Kemudian untuk mendapatkan sudut pada titik tengah = 0° nilai γ harus disesuaikan untuk setiap kamera. Dari kamera satu sampai empat, yaitu dari kanan sampai kiri nilai γ secara berurutan adalah $\theta + 38.5°, \theta - 2.5°, \theta - 43.5°,$ dan $\theta - 84.5°$.

3.3.3 Mengukur Lebar Aktual Objek

Sudut pandang kamera ketika mendeteksi sebuah objek dapat diilustrasikan seperti pada Gambar 3.5.



Gambar 3.5 Ilustrasi sudut pandang kamera

Dimana α adalah sudut jangkauan kamera, dengan nilai α = 46°, β, adalah sudut objek terhadap kapal, d adalah jarak objek terhadap kamera yang didapatkan dari integrasi informasi dengan sensor LiDar, lbp adalah lebar *bounding box* dalam *pixel*, T_b adalah lebar jangkauan kamera pada jarak t, sedangkan t adalah jarak objek terdeteksi tegak lurus terhadap kamera

Dengan nilai d didapatkan dengan integrasi data kamera dengan LiDar, dan sudut β didapatkan dari koordinat x *bounding box* dibandingan terhadap lebar *pixel* Gambar. Nilai β dan t didapatkan dengan Persamaan (3.5) dan (3.6).

$$\beta = \frac{x-160}{160} \times 23° \tag{3.5}$$

$$t = d \cos \beta \tag{3.6}$$

Kemudian dapat dicari nilai T_b dan dpp, dengan Persamaan (3.7) dan (3.8).

$$T_b = 2t\ \left(tan\left({}^{1}/_{2}\,\alpha\right)\right) \tag{3.7}$$

$$dpp = \frac{T_b}{320} \tag{3.8}$$

Dimana dpp adalah jarak aktual per *pixel* yang akan digunakan untuk mencari lebar aktual, dengan lbp adalah lebar bounding box dalam *pixel* maka lebar aktual objek (lb) dapat dicari dengan Persamaan (3.9).

$$lb = lbp\ X\ dpp \tag{3.9}$$

3.4 Tracking Objek atau Halangan

3.4.1 Metode Tracking Objek atau Halangan

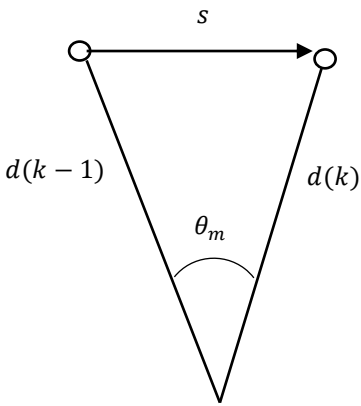
Kamera pada USV akan memberikan data berupa posisi objek atau halangan pada setiap *timestamp*, namun tidak diketahui apakah objek pada waktu sekarang adalah objek yang sama dengan objek pada waktu sebelumnya. Sehingga untuk mengetahui apakah objek yang sekarang adalah objek yang sama dengan objek pada waktu sebelumnya perlu dilakukan tracking atau penjejakan objek berdasarkan perpindahan jarak objek terhadap USV (s) untuk setiap objek yang terdeteksi pada tiap *timestamp*. Jarak perpindahan objek dihitung berdasarkan jarak objek terhadap USV dan sudut objek terhadap USV dengan menganggap jarak dan sudut objek dari USV pada setiap *timestamp* adalah vektor.

Jarak perpindahan untuk setiap objek, s, dapat dihitung dengan rumus *dot product* pada Persamaan (3.10)

$$s = \sqrt{d(k)^2 + d(k-1)^2 - 2.d(k).d(k-1)\cos\left(\theta_m\right)} \tag{3.10}$$

$$\theta_m = |\theta_0(k) - \theta_0(k-1)| \tag{3.11}$$

Dimana s adalah perpindahan jarak objek atau halangan, $d(k)$adalah jarak relatif halangan terhadap USV pada waktu k, θ_m adalah sudut yang dibentuk antara dua vektor jarak objek terhadap USV, dan $\theta_0(k)$ adalah sudut halangan terhadap USV pada saat k . Seperti ditunjukkan pada Gambar 3.6. Apabila jarak dan sudut perpindahan objek lebih rendah dari *threshold* maka dapat dinyatakan bahwa objek pada waktu $t = k$ dan $t = k - 1$ adalah objek yang sama.



Gambar 3.6 Ilustrasi *Tracking* Objek

3.5 Perancangan Filter Kalman

Konsep 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*. Model sistem dan model pengukuran yang digunakan pada aplikasi filter Kalman pada USV ditunjukkan pada Persamaan (3.12).

$$\begin{aligned} \boldsymbol{x}_k &= \boldsymbol{A}\boldsymbol{x}_{k-1} + \boldsymbol{B}\boldsymbol{u}_k + \boldsymbol{w}_k \\ \boldsymbol{z}_k &= \boldsymbol{H}_k\boldsymbol{x}_k + \boldsymbol{v}_k \end{aligned} \tag{3.12}$$

Dimana $\boldsymbol{A}$ adalah matriks transisi sistem dalam waktu diskrit yang didefinisikan pada Persamaan (3.13), $\boldsymbol{B}$ adalah matriks transisi variabel kontrol yang didefinisikan pada Persamaan (3.14), $\boldsymbol{H}_k$ merupakan matriks transisi state pengukuran dengan state $\boldsymbol{x}_k$, sedangkan $\boldsymbol{x}_k$ adalah vektor state space, dengan 4 state meliputi posisi, kecepatan, sudut, dan kecepatan sudut halangan terdeteksi relatif terhadap kapal pada waktu k yang dinyatakan dalam x_1, x_2, x_3, dan x_4 pada (3.15), sedangkan $\boldsymbol{u}_k$ adalah matriks variabel kontrol dengan nilai a adalah percepatan dan α adalah percepatan sudut seperti didefinikasn pada (3.16). Adapun $\boldsymbol{w}_k$ dan $\boldsymbol{v}_k$ masing-masing merupakan eror acak pada sistem dan pengukuran yang memiliki matriks kovarian pada Persamaan (3.16) dan (3.17). Nilai **Q** divariasikan pada simulasi untuk perbandingan performa sistem.

$$\boldsymbol{A} = \begin{bmatrix} 1 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \Delta t \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.13}$$

$$\boldsymbol{B} = \begin{bmatrix} \frac{1}{2}\Delta t^2 & 0 \\ \Delta t & 0 \\ 0 & \frac{1}{2}\Delta t^2 \\ 0 & \Delta t \end{bmatrix} \tag{3.14}$$

$$\boldsymbol{x}_k = \begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \\ x_4(k) \end{bmatrix} \tag{3.15}$$

$$\boldsymbol{u}_k = \begin{bmatrix} a(k) \\ \alpha(k) \end{bmatrix} \tag{3.16}$$

$$E[\boldsymbol{w}_k \boldsymbol{w}_i^T] = \begin{cases} 0 & i \neq k \\ \boldsymbol{Q} & i = k \end{cases} \tag{3.17}$$

$$E[\boldsymbol{v}_k \boldsymbol{v}_i^T] = \begin{cases} 0 & i \neq k \\ \boldsymbol{R} & i = k \end{cases} \tag{3.18}$$

Sebelumnya telah dilakukan *obstacle tracking* sehingga didapatkan matriks histori halangan yang menyimpan data jarak dan sudut relatif halangan terhadap kapal untuk setiap halangan terdeteksi pada waktu $\mathrm{t} = \mathrm{k}$ sampai $\mathrm{t} = \mathrm{k} - 2$. Data pengukuran ini digunakan untuk estimasi awal untuk mendapatkan nilai kecepatan, percepatan, kecepatan sudut, dan percepatan sudut dengan menggunakan Persamaan (3.19) sampai (3.22).

$$x_2(k-1) = \frac{x_1(k-1) - x_1(k-2)}{\Delta t} \tag{3.19}$$

$$a(k) = \frac{x_1(k) - 2x_1(k-1) + x_1(k-2)}{\Delta t^2} \tag{3.20}$$

$$x_4(k-1) = \frac{x_3(k-1) - x_3(k-2)}{\Delta t} \tag{3.21}$$

$$\alpha(k) = \frac{x_3(k) - 2x_3(k-1) + x_3(k-2)}{\Delta t^2} \tag{3.22}$$

Sehingga dapat dicari nilai estimasi awal pada waktu k dengan Persamaan (3.12). Kemudian dapat dihitung nilai awal matriks kovarians pada filter Kalman dengan Persamaan (2.9). Sehingga proses algoritma filter kalman dapat dijalankan sesuai dengan yang telah dijelaskan pada bab 2.

BAB 4
HASIL DAN ANALISA DATA

Bab ini membahas mengenai simulasi dan analisa dari perancangan sistem *obstacle tracking* menggunakan filter Kalman. Simulasi dilakukan menggunakan MATLAB dengan direncanakan lintasan sejumlah halangan beserta USV untuk mengetahui performa dari sistem yang dirancang dengan menggunakan beberapa parameter yang telah ditentukan. Simulasi dilakukan dengan melakukan perbandingan pada perubahan nilai kovarians **Q** pada filter Kalman untuk mendapatkan hasil estimasi yang optimal. Simulasi pertama dijalankan dengan nilai nilai kovarians **Q** lebih besar dari nilai kovarians **R**, sedangkan simulasi kedua dijalankan dengan nilai kovarians **Q** lebih kecil dari nilai kovarians **R**. Dengan nilai kovarians **Q** dan **R** yang digunakan dipaparkan pada Tabel 4.1.

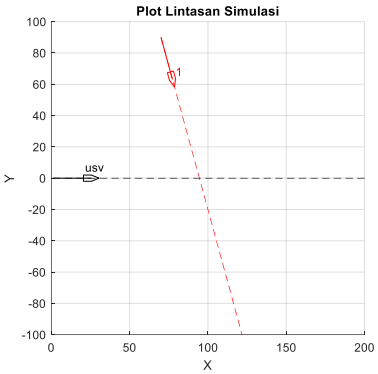
Tabel 4.1 Keterangan variasi nilai kovarians Q dan R

Simulasi	Q	R
Simulasi 1	$\begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 20 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 20 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 10 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 10 \end{bmatrix}$
Simulasi 2	$\begin{bmatrix} 0.5 & 0 & 0 & 0 \\ 0 & 5 & 0 & 0 \\ 0 & 0 & 0.5 & 0 \\ 0 & 0 & 0 & 5 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 10 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 10 \end{bmatrix}$

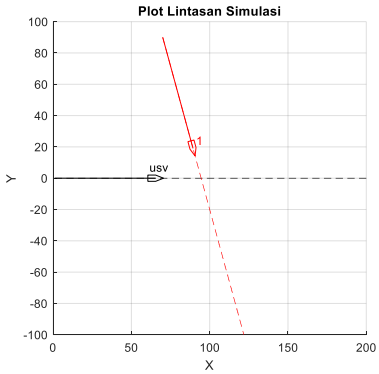
4.1 Hasil Simulasi *Obstacle Tracking*

4.1.1 Hasil Simulasi dengan Satu Halangan

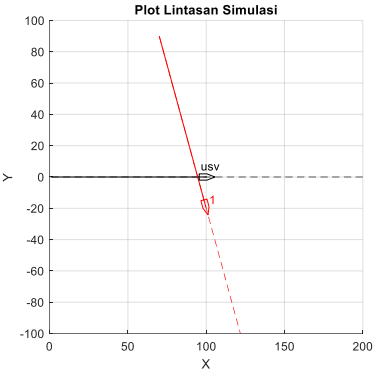
Pada simulasi dengan satu halangan, direncanakan lintasan halangan terdeteksi bergerak dari sisi kiri menuju sisi kanan pada sudut pandang kapal seperti ditunjukkan pada Gambar 4.1. Kemudian hasil plot jarak dan sudut relatif halangan terhadap kapan ditunjukkan pada Gambar 4.2 dan Gambar 4.3.



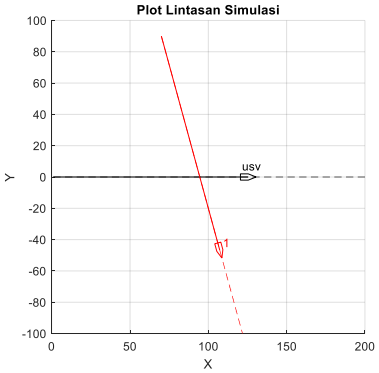
(a) Detik ke-5



(b) Detik ke-13

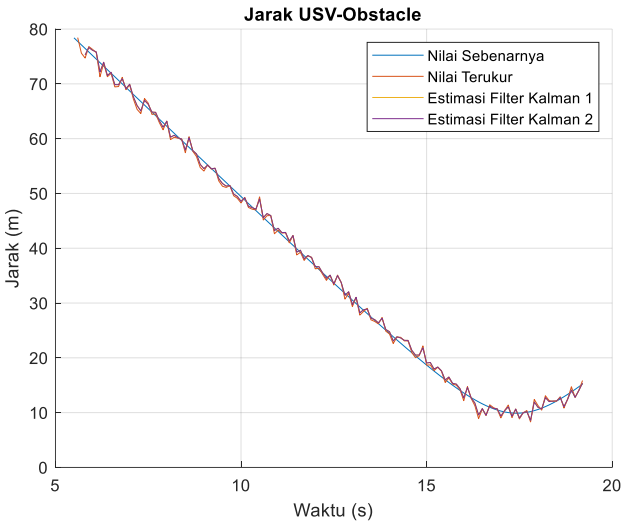


(c) Detik ke-20

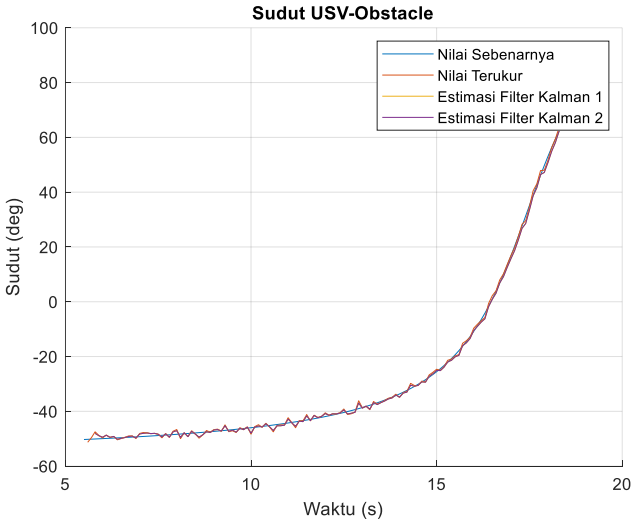


(d) Detik ke-25

Gambar 4.1 Plot lintasan simulasi dengan satu halangan



Gambar 4.2 Plot jarak relatif halangan terhadap kapal



Gambar 4.3 Plot sudut relatif halangan terhadap kapal

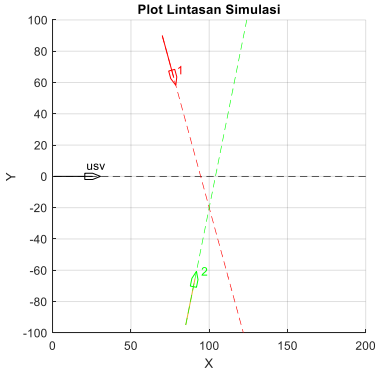
Perbandingan RMSE hasil filter Kalman dan pengukuran pada simulasi dengan satu halangan dengan hasil filter Kalman 2 menunjukkan RMSE nilai estimasi sudut yang lebih besar daripada pengukuran, seperti ditunjukkan pada Tabel 4.2

Tabel 4.2 Perbandingan RMSE hasil filter Kalman dan pengukuran pada simulasi dengan satu halangan

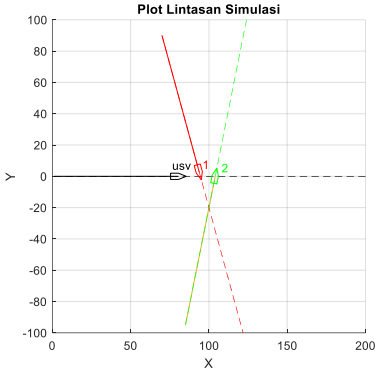
Halangan		RMSE Pengukuran	RMSE filter Kalman 1	RMSE filter Kalman 2
Halangan 1	Jarak	1.1264	0.9480	0.8960
	Sudut	1.0002	0.9236	1.4712

4.1.2 Hasil Simulasi dengan Dua Halangan

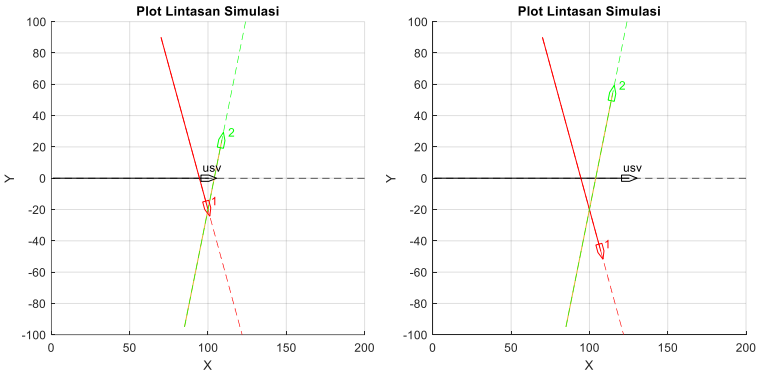
Pada simulasi dengan dua halangan, direncanakan halangan tambahan dengan lintasan bergerak dari sisi kanan menuju sisi kiri pada sudut pandang kapal. Pada satu titik halangan kedua akan terhalang oleh halangan pertama, dengan lintasan halangan pertama sama seperti pada simulasi sebelumnya. Lintasan simulasi dengan dua halangan ditunjukkan pada Gambar 4.4. Kemudian plot jarak dan sudut relatif halangan terhadap kapal ditunjukkan pada Gambar 4.5 sampai Gambar 4.8.



(a) Detik ke-5

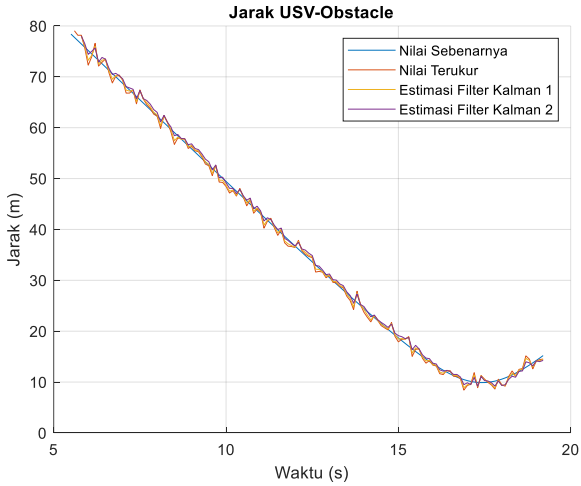


(b) Detik ke-16

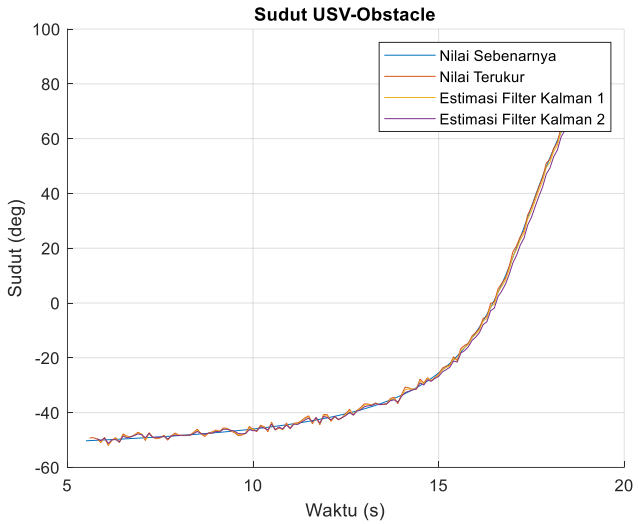


(c) Detik ke-20 **(d)** Detik ke-25

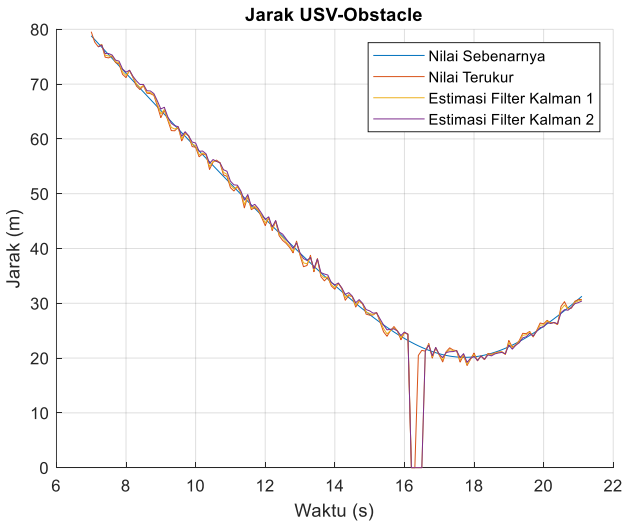
Gambar 4.4 Plot lintasan simulasi dengan dua halangan



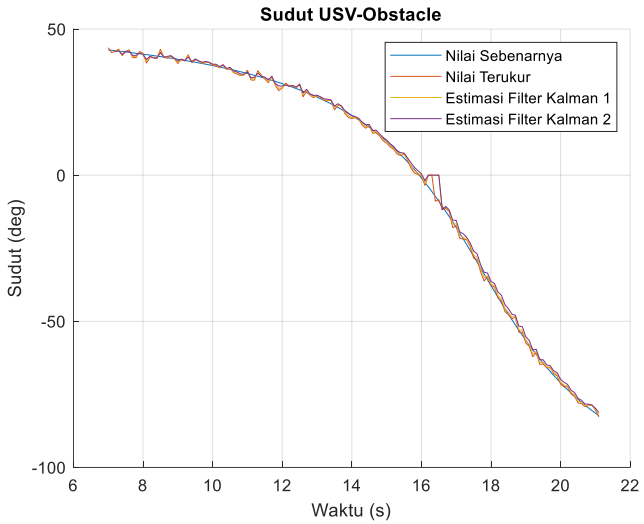
Gambar 4.5 Plot sudut relatif halangan terhadap kapal pada halangan satu



Gambar 4.6 Plot sudut relatif halangan terhadap kapal pada halangan satu



Gambar 4.7 Plot jarak relatif halangan terhadap kapal pada halangan dua



Gambar 4.8 Plot sudut relatif halangan terhadap kapal pada halangan dua

Plot jarak relatif halangan dua menunjukkan hilangnya nilai pengukuran pada detik 16 seperti ditunjukkan pada Gambar 4.7, hal ini dikarenakan halangan dua terhalang oleh halangan satu.

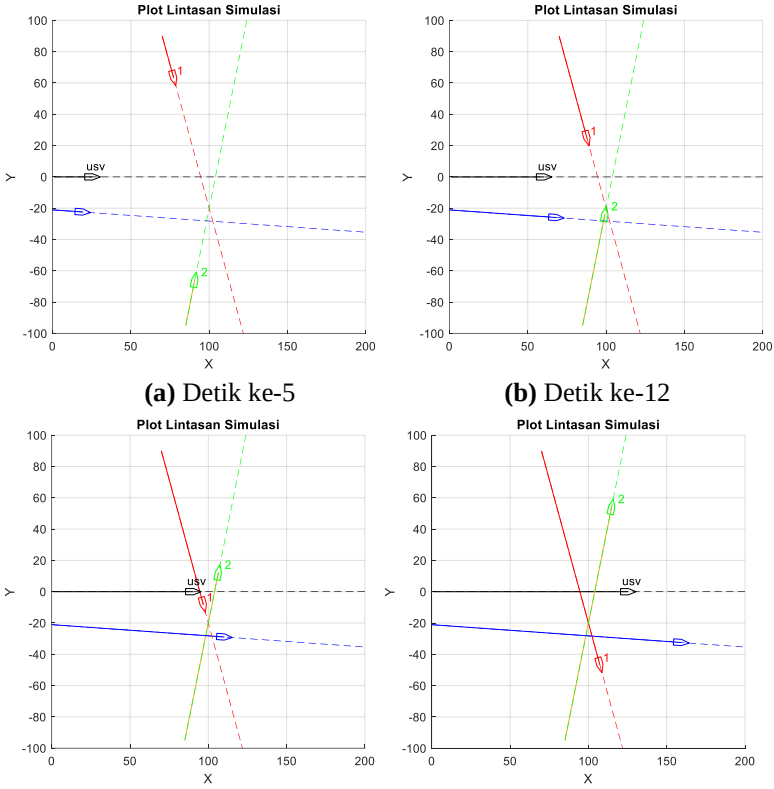
Perbandingan RMSE hasil filter Kalman dan pengukuran pada simulasi dengan dua halangan menunjukkan RMSE nilai estimasi sudut pada filter Kalman 2 lebih besar daripada RMSE pengukuran seperti ditunjukkan pada Tabel 4.3.

Tabel 4.3 Perbandingan RMSE hasil filter Kalman dan pengukuran pada simulasi dengan dua halangan

Halangan		RMSE Pengukuran	RMSE filter Kalman 1	RMSE filter Kalman 2
Halangan 1	Jarak	1.0300	0.8234	0.7561
	Sudut	1.0681	0.9673	1.4773
Halangan 2	Jarak	0.9027	0.7469	0.7358
	Sudut	1.0913	0.9604	1.1924

4.1.3 Hasil Simulasi dengan Tiga Halangan

Pada simulasi dengan tiga halangan, direncanakan halangan tambahan dengan lintasan halangan menyalip kapal dan semakin menjauhi kapal pada sisi kanan dari sudut pandang kapal. Dengan lintasan halangan sebelumnya sama seperti pada simulasi sebelumnya. Pada suatu titik halangan ketiga akan terhalang oleh halangan pertama. Lintasan simulasi dengan tiga halangan seperti ditunjukkan pada Gambar 4.9.



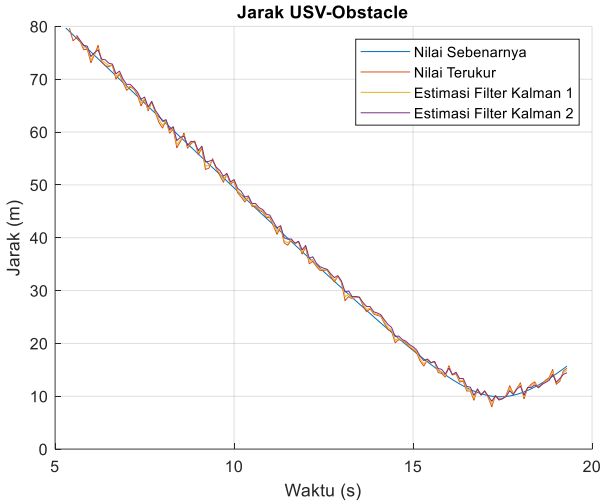
(a) Detik ke-5 **(b)** Detik ke-12

(c) Detik ke-18 **(d)** Detik ke-25

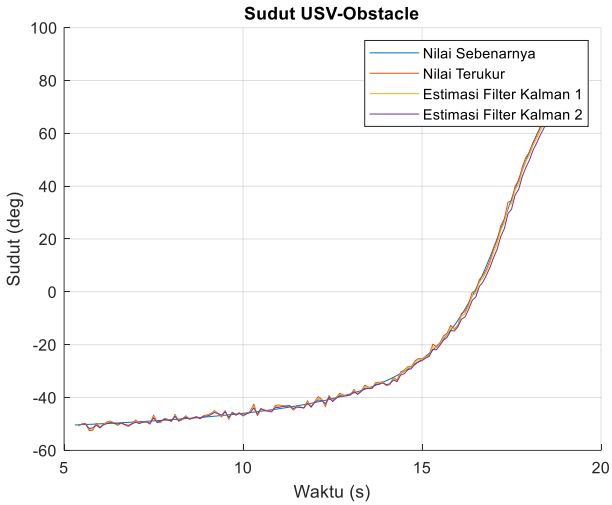
Gambar 4.9 Plot lintasan simulasi dengan tiga halangan

Plot jarak dan sudut relatif halangan terhadap kapal ditunjukkan pada Gambar 4.10 sampai Gambar 4.15. Perbandingan RMSE hasil filter Kalman dan pengukuran pada simulasi dengan dua halangan menunjukkan RMSE nilai estimasi sudut dengan filter Kalman 2 pada halangan 1 dan 2 lebih besar daripada RMSE pengukuran seperti ditunjukkan pada Tabel 4.4 **Perbandingan RMSE hasil filter Kalman dan pengukuran pada simulasi dengan tiga halangan** Tabel 4.4.

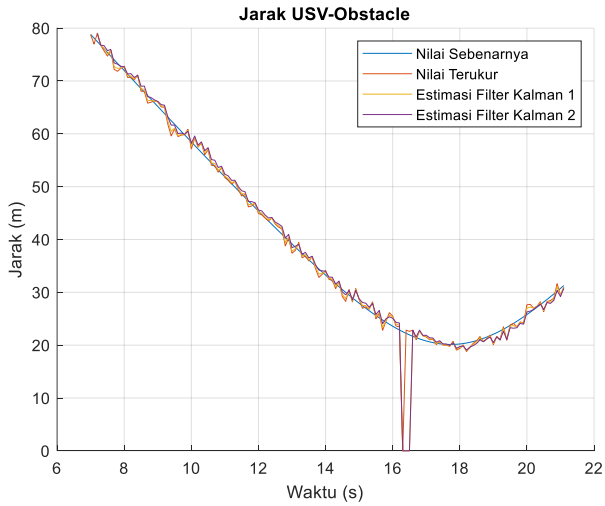
Pada Gambar 4.14, terlihat nilai pengukuran hilang sesaat sebelum detik ke-20. Hal ini dikarenakan halangan tiga terhalang oleh halangan satu. Seperti yang terlihat pada Gambar 4.9.



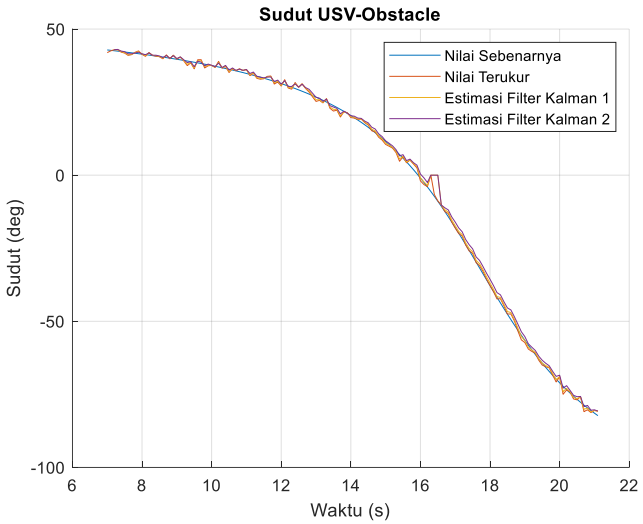
Gambar 4.10 Plot jarak relatif halangan terhadap kapal pada halangan satu



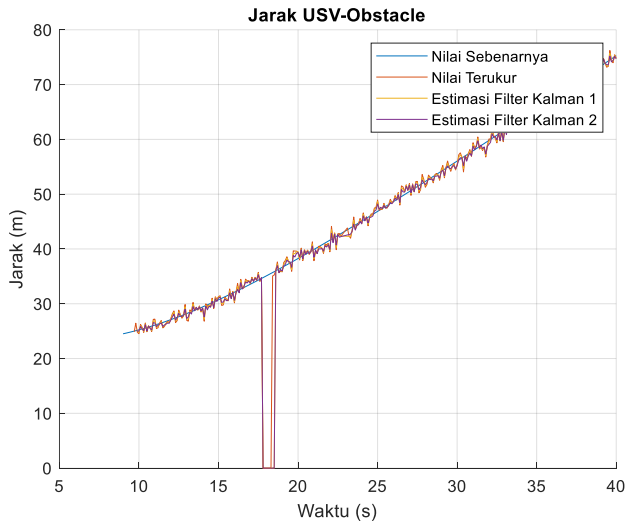
Gambar 4.11 Plot sudut relatif halangan terhadap kapal pada halangan satu



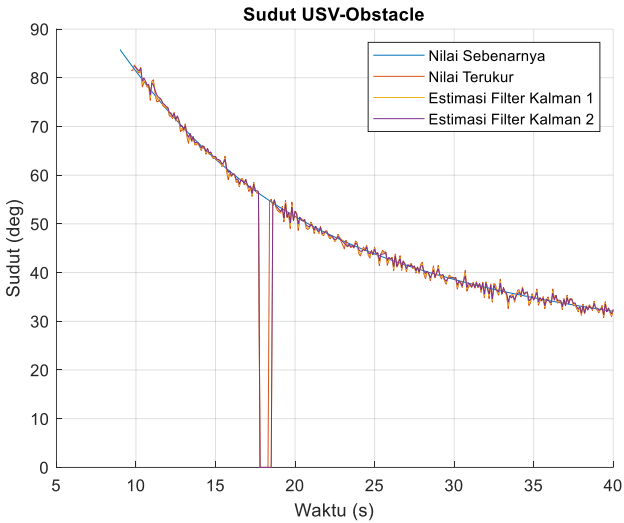
Gambar 4.12 Plot jarak relatif halangan terhadap kapal pada halangan dua



Gambar 4.13 Plot sudut relatif halangan terhadap kapal pada halangan dua



Gambar 4.14 Plot jarak relatif halangan terhadap kapal pada halangan tiga



Gambar 4.15 Plot sudut relatif halangan terhadap kapal pada halangan tiga

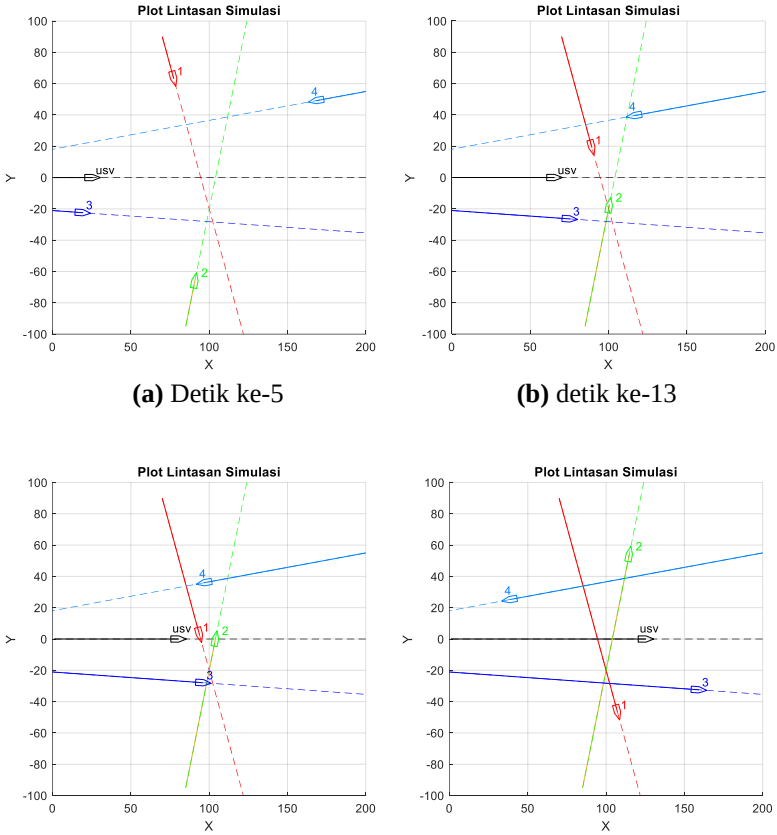
Tabel 4.4 Perbandingan RMSE hasil filter Kalman dan pengukuran pada simulasi dengan tiga halangan

Halangan		RMSE Pengukuran	RMSE filter Kalman 1	RMSE filter Kalman 2
Halangan 1	Jarak	0.9674	0.8438	0.8747
	Sudut	1.1028	1.0069	1.5085
Halangan 2	Jarak	1.0275	0.9137	0.9289
	Sudut	1.0222	0.9799	1.2886
Halangan 3	Jarak	1.0030	0.8314	0.6819
	Sudut	1.0315	0.8465	0.6798

4.1.4 Hasil Simulasi dengan Empat Halangan

Pada simulasi dengan empat halangan, ditambahkan halangan baru dengan lintasan halangan mendekati kapal pada sisi kiri dari sudut pandang kapal. Dengan lintasan halangan sebelumnya sama seperti pada simulasi sebelumnya. Pada suatu titik halangan keempat akan terhalang oleh halangan pertama. Seperti ditunjukkan pada Gambar 4.16. Plot jarak

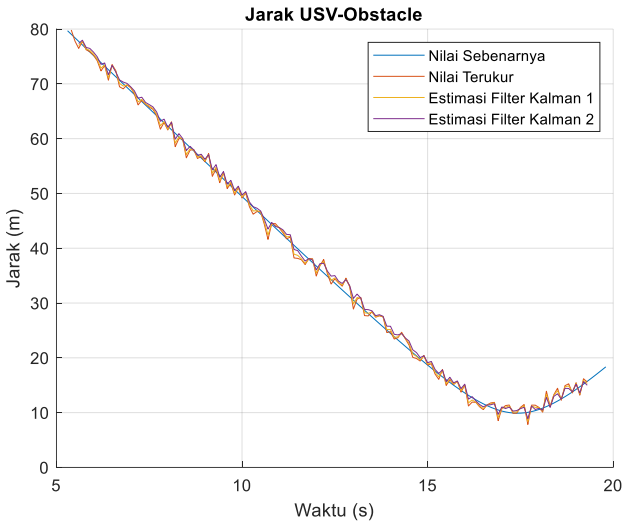
dan sudut relatif nilai sebenarnya, hasil pengukuran, dan filter kalman ditunjukkan pada Gambar 4.17 sampai Gambar 4.24.



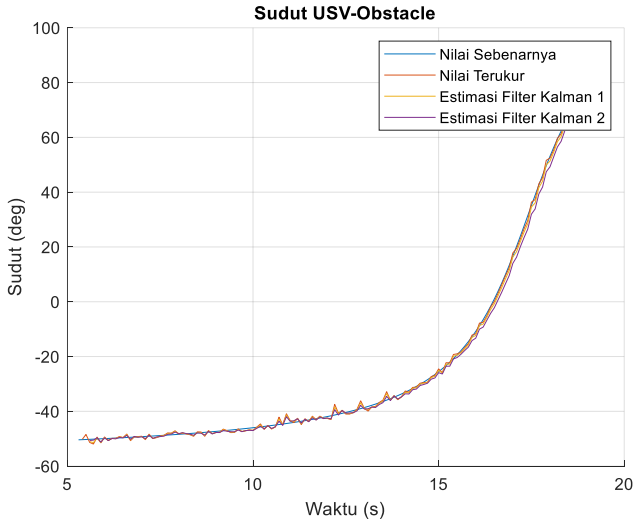
(a) Detik ke-5 **(b)** detik ke-13

(c) Detik ke-20 **(d)** Detik ke-25

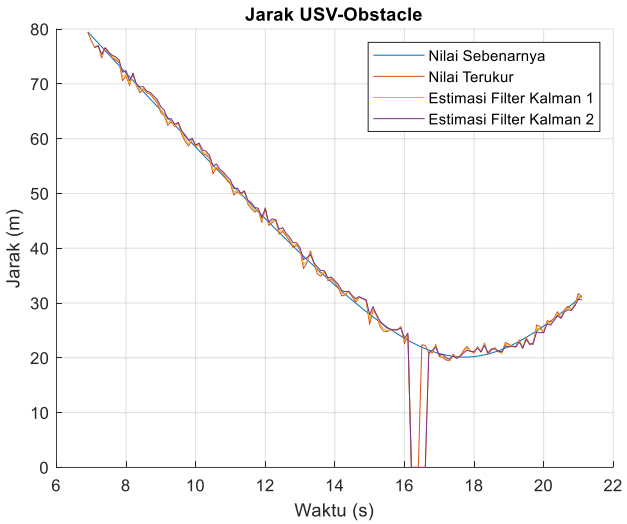
Gambar 4.16 Plot lintasan simulasi dengan empat halangan.



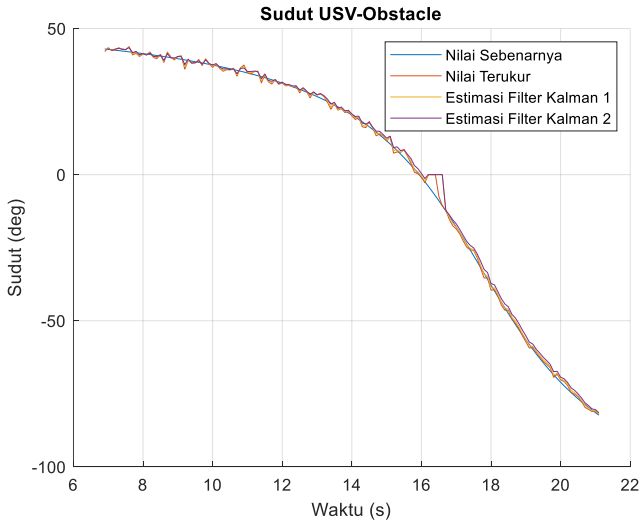
Gambar 4.17 Plot jarak relatif halangan terhadap kapal pada halangan satu



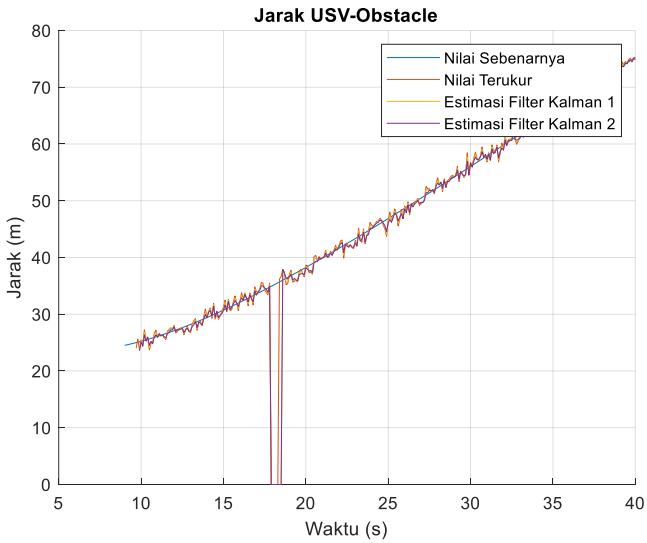
Gambar 4.18 Plot sudut relatif halangan terhadap kapal pada halangan satu



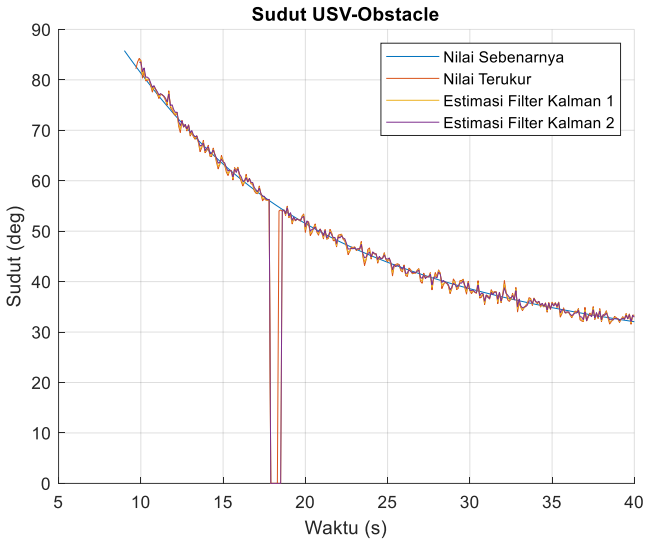
Gambar 4.19 Plot jarak relatif halangan terhadap kapal pada halangan dua



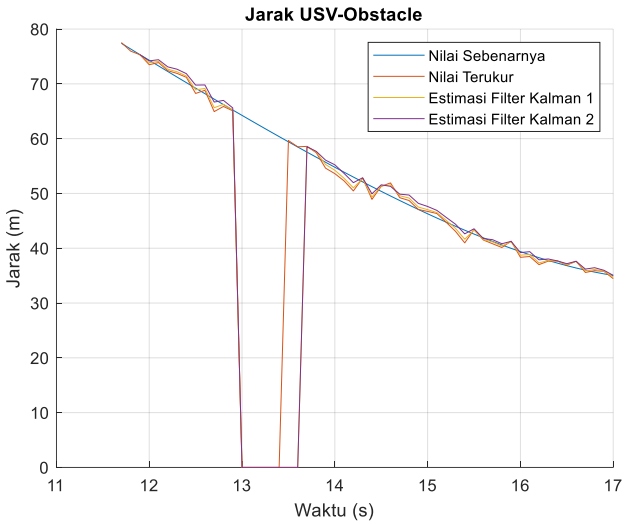
Gambar 4.20 Plot sudut relatif halangan terhadap kapal pada halangan dua



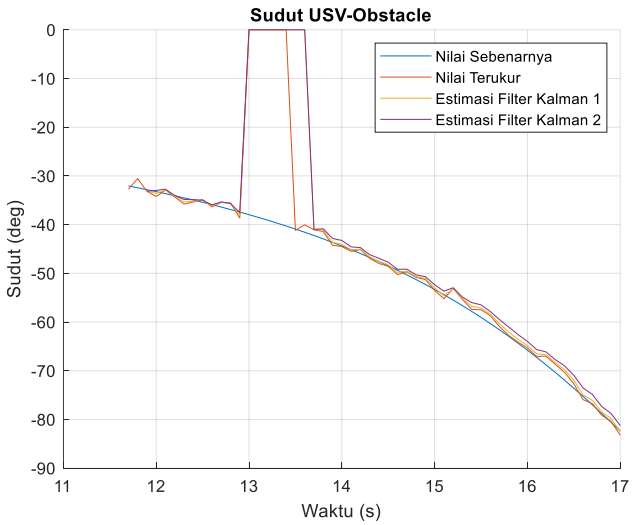
Gambar 4.21 Plot jarak relatif halangan terhadap kapal pada halangan tiga



Gambar 4.22 Plot sudut relatif halangan terhadap kapal pada halangan tiga



Gambar 4.23 Plot jarak relatif halangan terhadap kapal pada halangan empat



Gambar 4.24 Plot sudut relatif halangan terhadap kapal pada halangan empat

Hasil plot pada Gambar 4.23 menunjukkan nilai pengukuran hilang pada detik ke-13. Hal ini dikarenakan halangan empat terhalang oleh halangan satu, seperti yang terlihat pada plot lintasan pada Gambar 4.16.

Perbandingan RMSE hasil filter Kalman dan pengukuran menunjukkan beberapa nilai estimasi filter Kalman 2 lebih besar daripada nilai pengukuran seperti ditunjukkan pada Tabel 4.5.

Tabel 4.5 Perbandingan RMSE hasil filter Kalman dan pengukuran pada simulasi dengan empat halangan

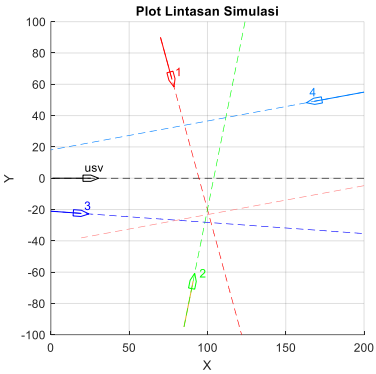
Halangan		RMSE Pengukuran	RMSE filter Kalman 1	RMSE filter Kalman 2
Halangan 1	Jarak	1.0608	0.9307	0.9519
	Sudut	0.9178	0.8654	1.4588
Halangan 2	Jarak	1.0779	0.9082	0.8460
	Sudut	1.0025	0.8488	1.0962
Halangan 3	Jarak	0.9686	0.8171	0.6778
	Sudut	0.9764	0.8081	0.6668
Halangan 4	Jarak	0.6460	0.6276	0.8819
	Sudut	1.0675	0.9276	1.1646

4.1.5 Hasil Simulasi dengan Lima Halangan

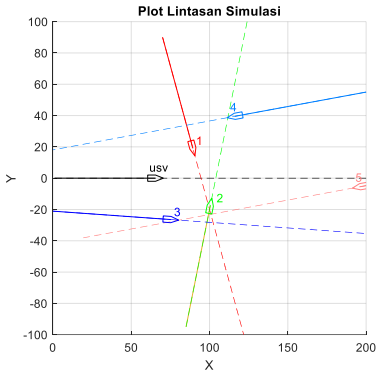
Pada Simulasi dengan lima halangan, ditambahkan satu lintasan untuk halangan baru dengan lintasan yang semakin mendekat kepada kapal dan menghalangi halangan 3, seperti ditunjukkan pada Gambar 4.25. Plot jarak dan sudut relatif halangan terhadap kapal ditunjukkan pada Gambar 4.26 sampai Gambar 4.35.

Plot pada Gambar Gambar 4.30 menunjukkan nilai pengukuran yang hilang sesaat sebelum detik ke-25. Hal ini dikarenakan halangan tiga terhalang oleh halangan 5 seperti yang telah dijelaskan sebelumnya. Halangan 5 menghalangi halangan 3 terlihat pada Gambar 4.25.

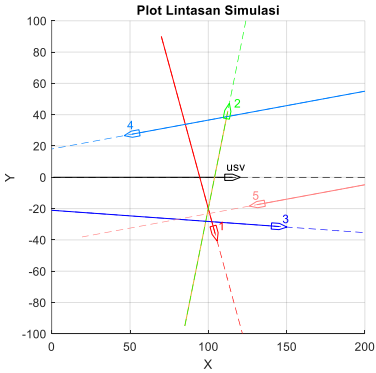
Perbandingan RMSE hasil filter Kalman dan pengukuran menunjukkan beberapa RMSE filter Kalman 2 menunjukkan nilai yang lebih besar daripada RMSE pengukuran, sedangkan RMSE dengan filter Kalman 1 menunjukkan nilai RMSE yang lebih kecil, seperti ditunjukkan pada Tabel 4.6.



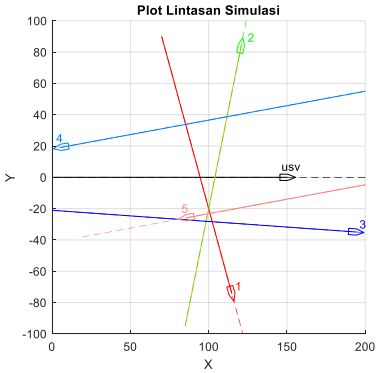
(a) Detik ke-5



(b) Detik ke-13

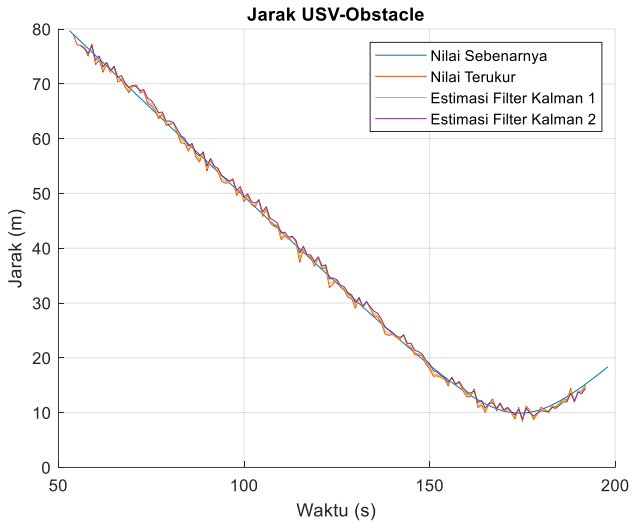


(c) Detik ke-23

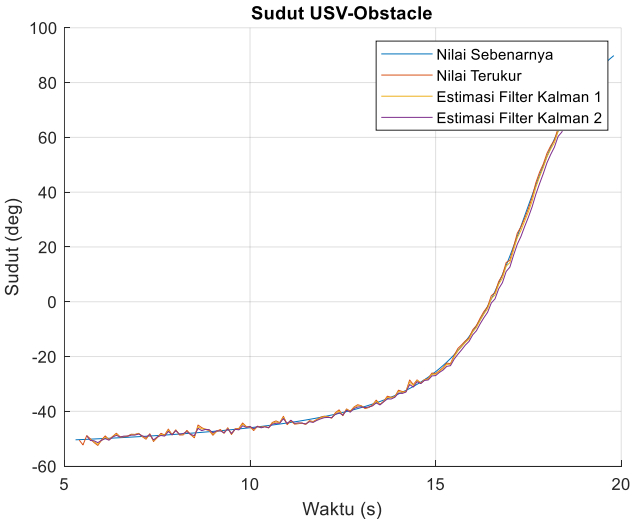


(d) Detik ke-30

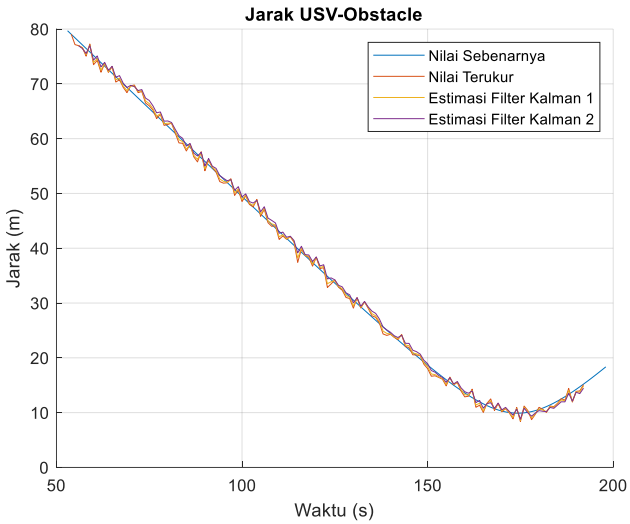
Gambar 4.25 Plot lintasan simulasi dengan lima halangan



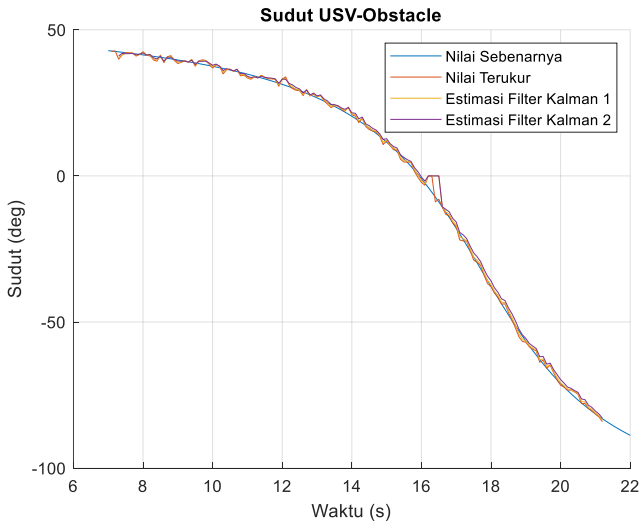
Gambar 4.26 Plot jarak relatif halangan terhadap kapal pada halangan satu



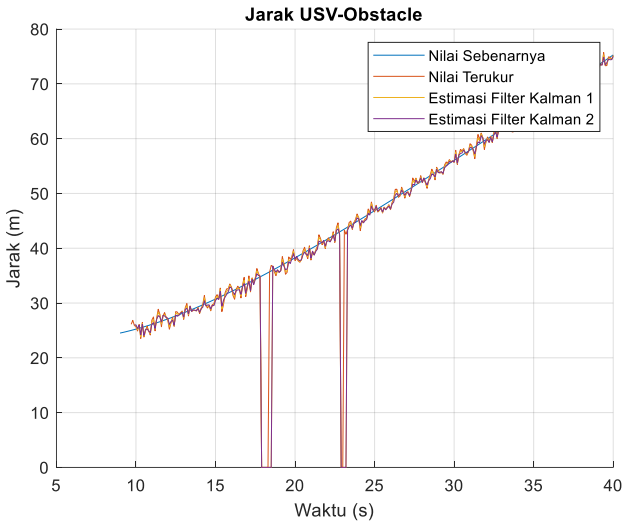
Gambar 4.27 Plot sudut relatif halangan terhadap kapal pada halangan satu



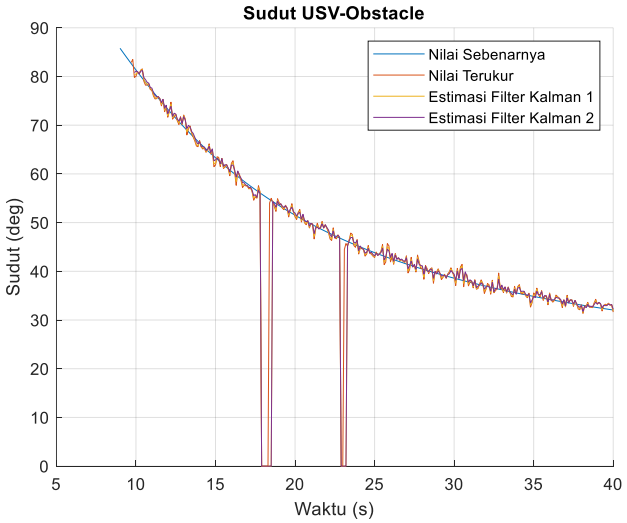
Gambar 4.28 Plot jarak relatif halangan terhadap kapal pada halangan dua



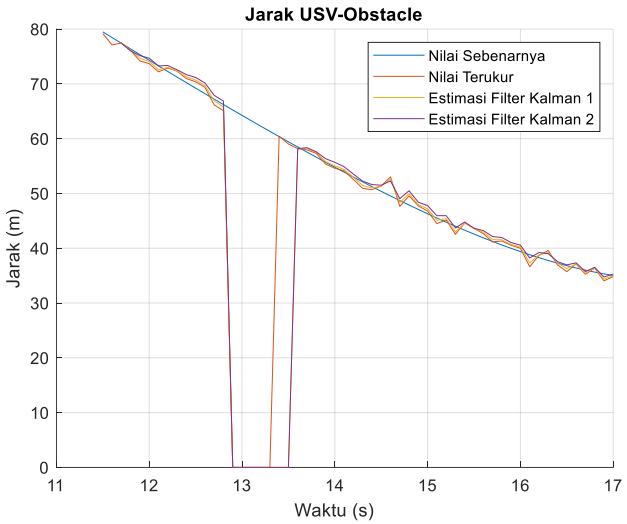
Gambar 4.29 Plot sudut relatif halangan terhadap kapal pada halangan dua



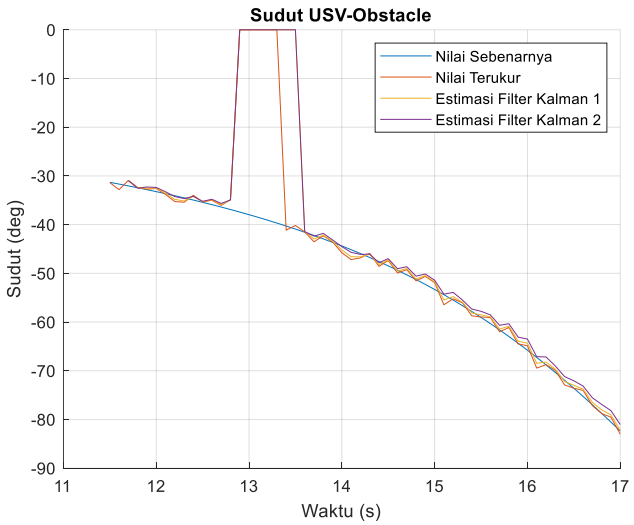
Gambar 4.30 Plot jarak relatif halangan terhadap kapal pada halangan tiga



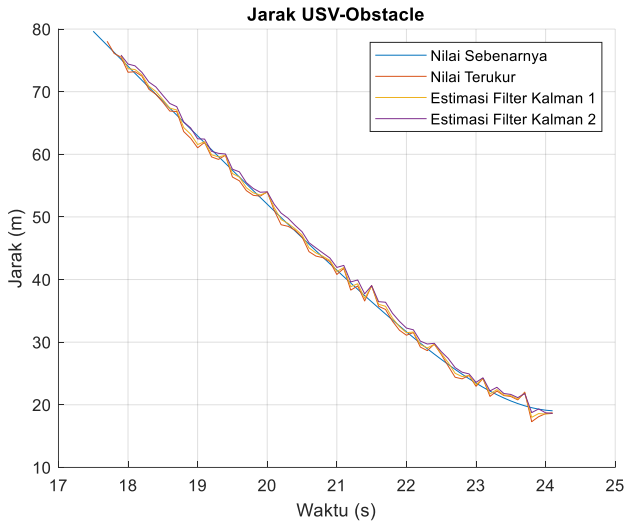
Gambar 4.31 Plot sudut relatif halangan terhadap kapal pada halangan tiga



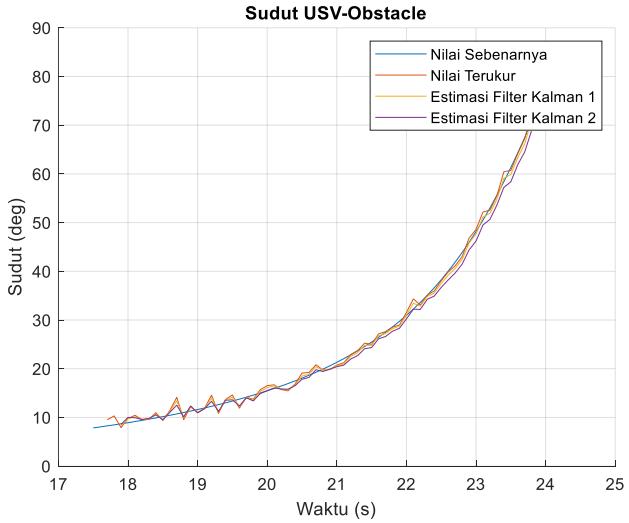
Gambar 4.32 Plot jarak relatif halangan terhadap kapal pada halangan empat



Gambar 4.33 Plot sudut relatif halangan terhadap kapal pada halangan empat



Gambar 4.34 Plot jarak relatif halangan terhadap kapal pada halangan lima



Gambar 4.35 Plot sudut relatif halangan terhadap kapal pada halangan lima

Tabel 4.6 Perbandingan RMSE hasil filter Kalman dan pengukuran pada simulasi dengan lima halangan

Halangan		RMSE Pengukuran	RMSE filter Kalman 1	RMSE filter Kalman 2
Halangan 1	Jarak	0.9356	0.8122	0.8290
	Sudut	1.0042	0.9652	1.5251
Halangan 2	Jarak	1.0297	0.8885	0.8666
	Sudut	0.9833	0.9205	1.2379
Halangan 3	Jarak	1.0984	0.8508	0.6961
	Sudut	1.0326	0.8838	0.7340
Halangan 4	Jarak	1.0694	0.9730	1.1124
	Sudut	0.9904	0.9144	1.2416
Halangan 5	Jarak	0.9745	0.8273	0.9908
	Sudut	1.0418	0.8868	1.3513

4.2 Analisa Eror Jarak Halangan

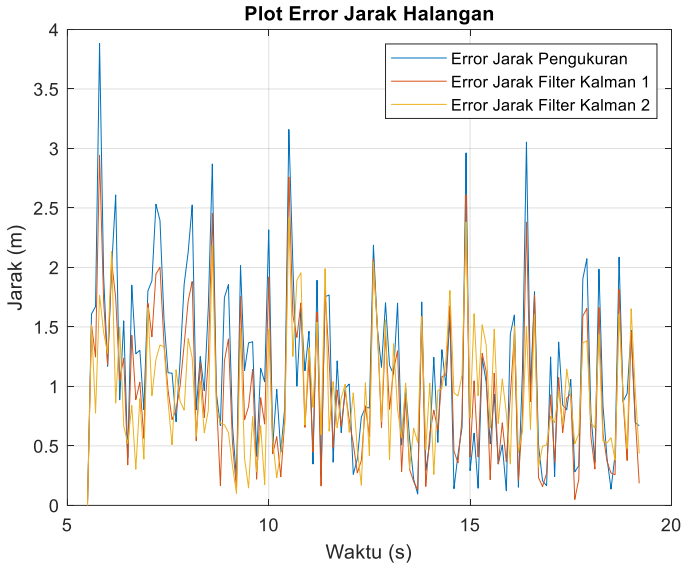
Selanjutnya dihitung nilai eror jarak antara titik koordinat sebenarnya dengan titik koordinat pengukuran, serta titik koordinat hasil estimasi filter Kalman pada kedua simulasi yang telah dilakukan pada subbab sebelumnya. Nilai eror jarak ini didapatkan menggunakan Persamaan (3.10) untuk menghitung jarak perpindahan antar titik koordinat dengan informasi jarak dan sudut relatif halangan terhadap kapal.

4.2.1 Eror Jarak Halangan pada Simulasi dengan Satu Halangan

Hasil perbandingan eror jarak halangan terdeteksi pada simulasi dengan satu halangan ditunjukkan pada Gambar 4.36. Dengan perbandingan nilai eror jarak halangan ditunjukkan pada Tabel 4.7.

Tabel 4.7 Perbandingan rata – rata eror jarak halangan pada simulasi dengan satu halangan

Halangan	Rata – rata eror jarak pengukuran	Rata- rata eror jarak filter Kalman 1	Rata – rata eror jarak filter Kalman 2
Halangan 1	1.1571	0.9768	0.9496



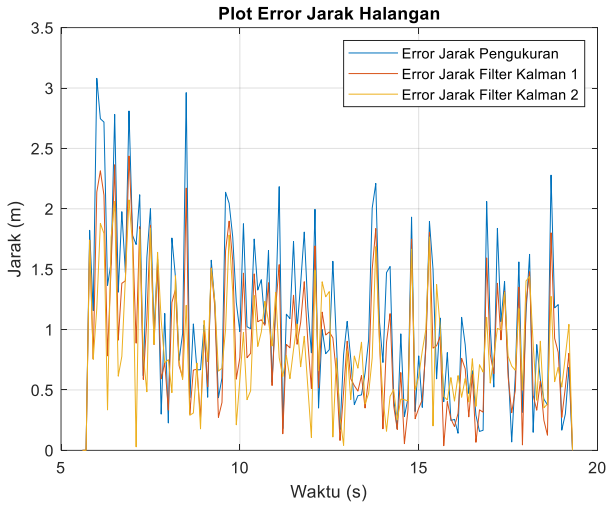
Gambar 4.36 Plot perbandingan nilai eror jarak halangan pada halangan satu

4.2.2 Eror Jarak Halangan pada Simulasi dengan Dua Halangan

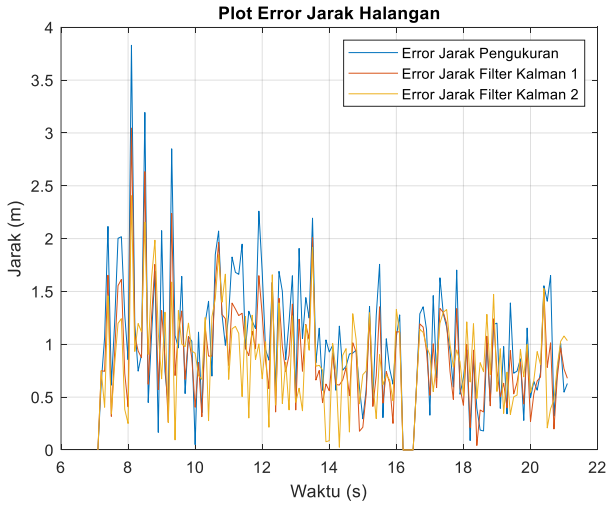
Hasil perbandingan eror jarak halangan terdeteksi pada simulasi dengan dua halangan ditunjukkan pada Gambar 4.37 dan Gambar 4.38. Dengan perbandingan nilai eror jarak halangan ditunjukkan pada Tabel 4.8. Nilai rata – rata eror jarak pada filter Kalman 2 lebih kecil daripada nilai rata – rata eror menggunakan filter Kalman 1, seperti ditunjukkan pada Tabel 4.8.

Tabel 4.8 Perbandingan rata – rata eror jarak halangan pada simulasi dengan dua halangan

Halangan	Rata – rata eror posisi pengukuran	Rata- rata eror jarak filter Kalman 1	Rata – rata eror jarak filter Kalman 2
Halangan 1	1.1093	0.9068	0.8519
Halangan 2	1.0740	0.9044	0.8933



Gambar 4.37 Plot perbandingan nilai eror jarak halangan pada halangan satu



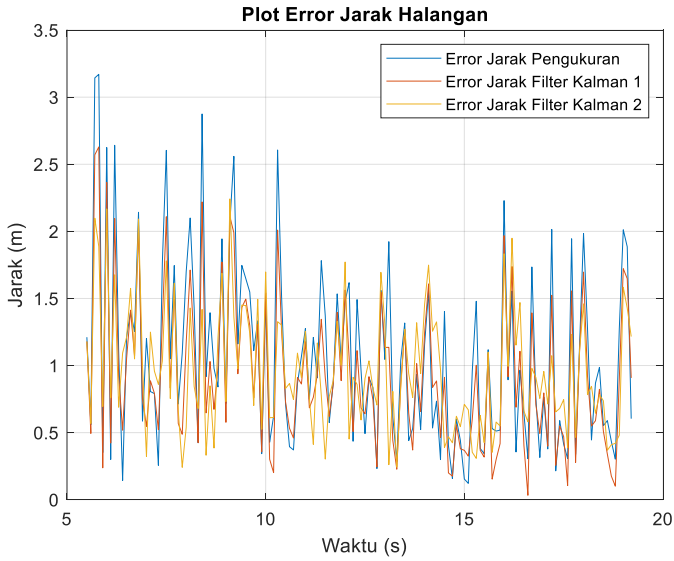
Gambar 4.38 Plot perbandingan nilai eror jarak halangan pada halangan dua

4.2.3 Eror Jarak Halangan pada Simulasi dengan Tiga Halangan

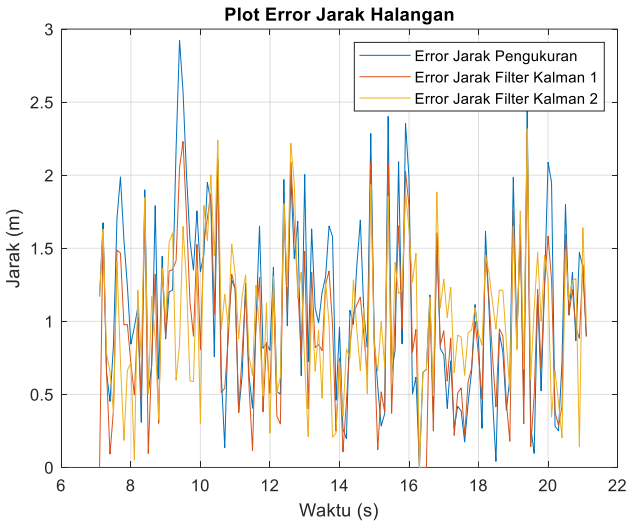
Hasil perbandingan eror jarak halangan terdeteksi pada simulasi dengan tiga halangan ditunjukkan pada Gambar 4.37 sampai Gambar 4.41. Nilai rata – rata eror jarak pada filter Kalman 1 selalu dibawah nilai rata – rata eror jarak pengukuran, sedangkan nilai rata – rata eror jarak pada halangan 2 menggunakan filter Kalman 2 lebih besar daripada nilai rata – rata eror jarak pengukuran., seperti ditunjukkan pada Tabel 4.9.

Tabel 4.9 Perbandingan rata – rata eror jarak halangan pada simulasi dengan tiga halangan

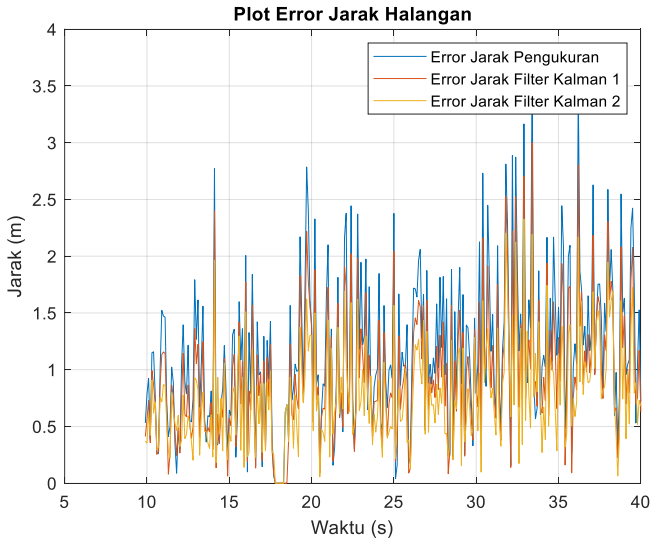
Halangan	Rata – rata eror posisi pengukuran	Rata- rata eror posisi filter Kalman 1	Rata – rata eror posisi filter Kalman 2
Halangan 1	1.0597	0.9280	0.9677
Halangan 2	1.0895	0.9772	1.0410
Halangan 3	1.1731	0.9746	0.7939



Gambar 4.39 Plot perbandingan nilai eror jarak halangan pada halangan satu



Gambar 4.40 Plot perbandingan nilai eror jarak halangan pada halangan dua



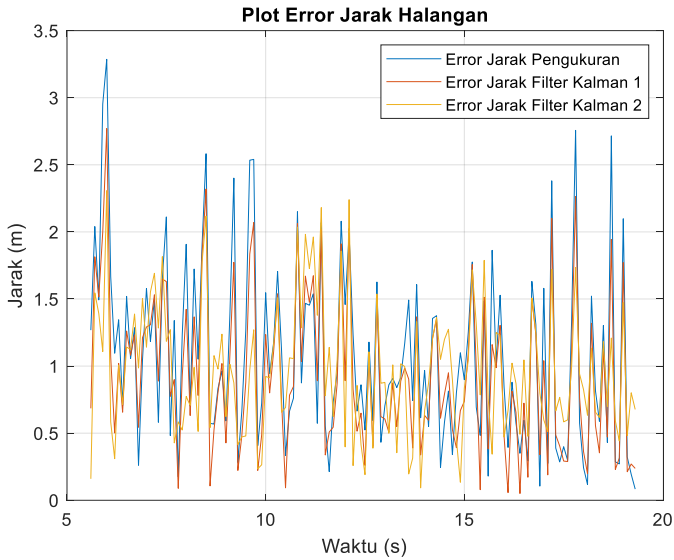
Gambar 4.41 Plot perbandingan nilai eror jarak halangan pada halangan tiga

4.2.4 Eror Jarak Halangan pada Simulasi dengan Empat Halangan

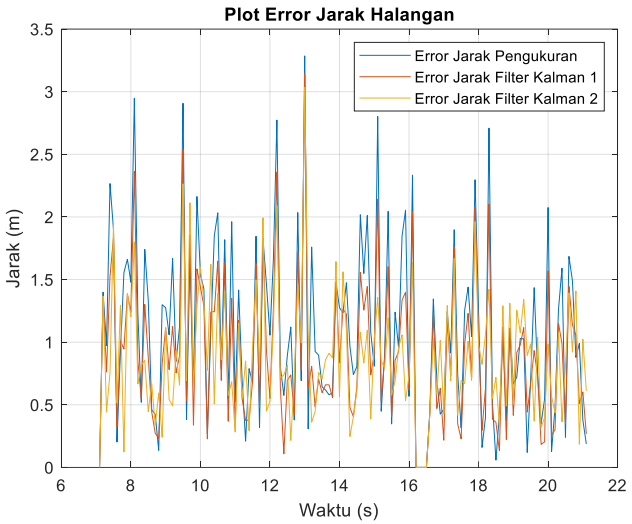
Hasil perbandingan eror jarak halangan terdeteksi pada simulasi dengan empat halangan ditunjukkan pada Gambar 4.42 sampai Gambar 4.45. Dengan perbandingan nilai eror jarak halangan ditunjukkan pada Tabel 4.10. Nilai rata – rata eror jarak pada filter Kalman 1 selalu dibawah nilai rata – rata eror jarak pengukuran, sedangkan nilai rata – rata eror jarak pada halangan 4 menggunakan filter Kalman 2 lebih besar daripada nilai rata – rata eror jarak pengukuran.

Tabel 4.10 Perbandingan rata – rata eror jarak halangan pada simulasi dengan empat halangan

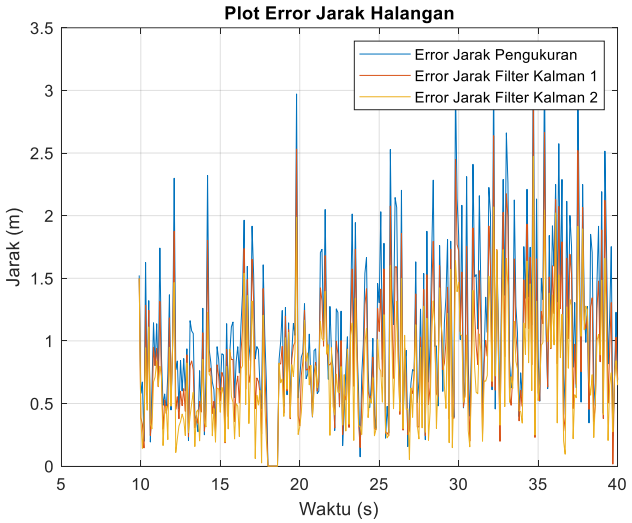
Halangan	Rata – rata eror jarak pengukuran	Rata – rata eror jarak filter Kalman 1	Rata – rata eror jarak filter Kalman 2
Halangan 1	1.0786	0.9332	0.9709
Halangan 2	1.1250	0.9373	0.9178
Halangan 3	1.1184	0.9287	0.7599
Halangan 4	1.0473	0.9423	1.2466



Gambar 4.42 Plot perbandingan nilai eror jarak halangan pada halangan satu



Gambar 4.43 Plot perbandingan nilai eror jarak halangan pada halangan dua



Gambar 4.44 Plot perbandingan nilai eror jarak halangan pada halangan tiga

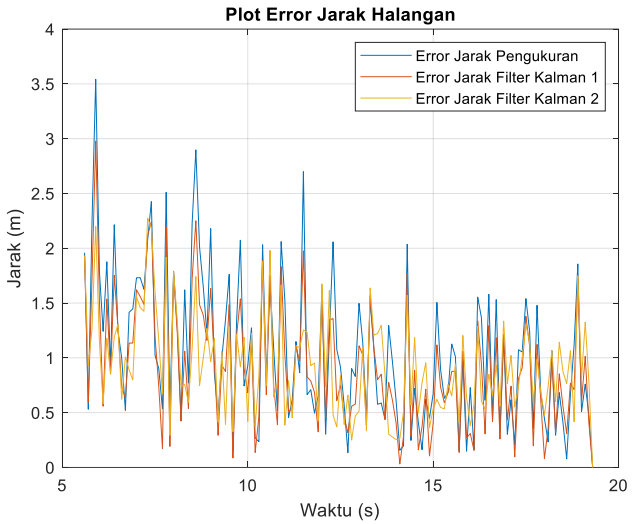
Gambar 4.45 Plot perbandingan nilai eror jarak halangan pada halangan empat

4.2.5 Eror Jarak Halangan pada Simulasi dengan Lima Halangan

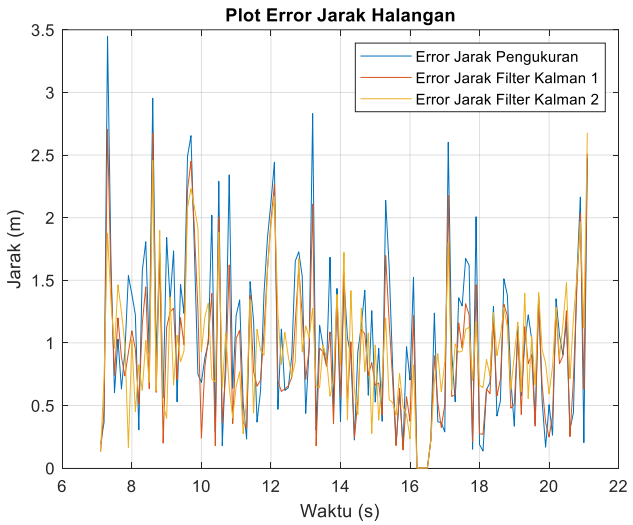
Hasil perbandingan eror jarak halangan terdeteksi pada simulasi dengan empat halangan ditunjukkan pada Gambar 4.46 sampai Gambar 4.50. Perbandingan nilai eror jarak halangan menunjukkan nilai rata – rata eror jarak pada halangan 4 menggunakan filter Kalman 2 lebih besar daripada nilai rata – rata eror jarak pengukuran, hal ini ditunjukkan pada Tabel 4.11.

Tabel 4.11 Perbandingan rata – rata eror jarak halangan pada simulasi dengan lima halangan

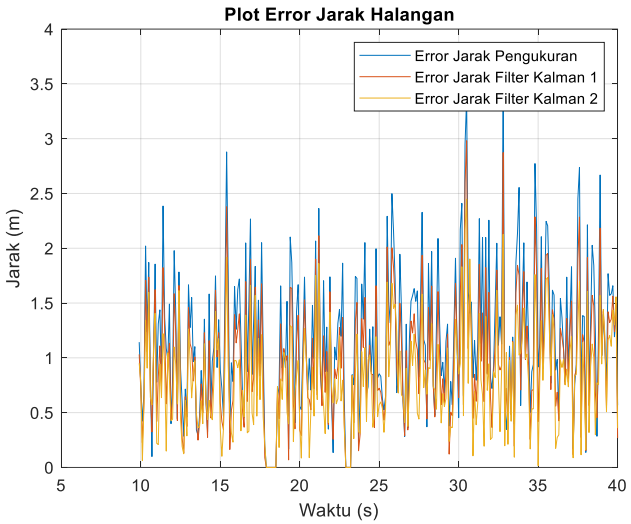
Halangan	Rata – rata eror jarak Pengukuran	Rata – rata eror jarak filter Kalman 1	Rata – rata eror jarak filter Kalman 2
Halangan 1	1.0567	0.9136	0.9336
Halangan 2	1.0749	0.9459	0.9936
Halangan 3	1.2281	1.0201	0.8373
Halangan 4	1.2450	1.1294	1.3531
Halangan 5	1.1932	0.9903	1.1888



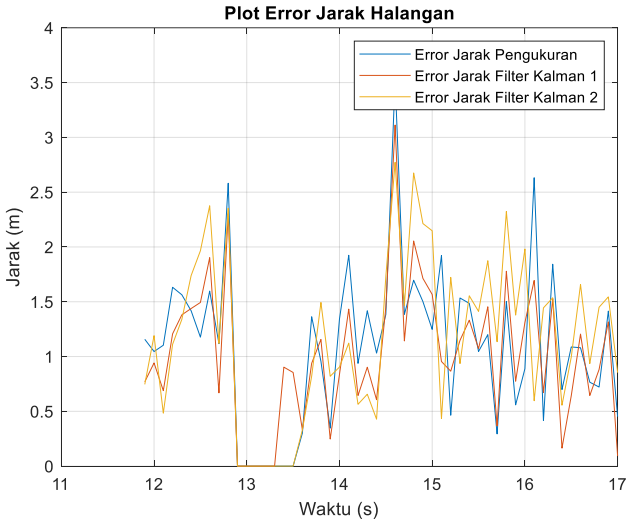
Gambar 4.46 Plot perbandingan nilai eror jarak halangan pada halangan satu



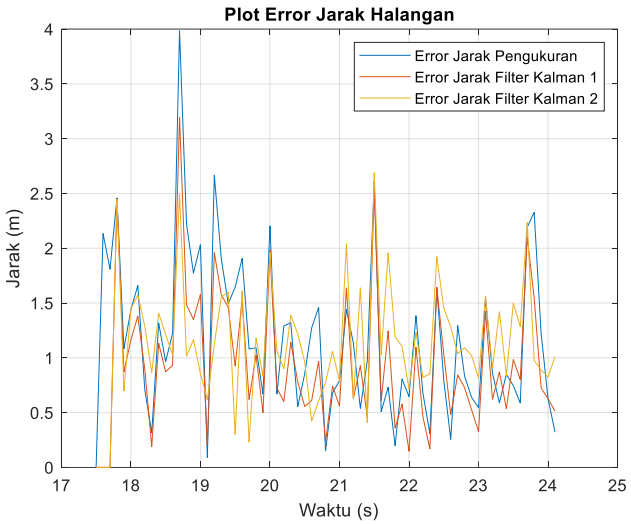
Gambar 4.47 Plot perbandingan nilai eror jarak halangan pada halangan dua



Gambar 4.48 Plot perbandingan nilai eror jarak halangan pada halangan tiga



Gambar 4.49 Plot perbandingan nilai eror jarak halangan pada halangan empat



Gambar 4.50 Plot perbandingan nilai eror jarak halangan pada halangan lima

Halaman ini sengaja dikosongkan

BAB 5
KESIMPULAN DAN SARAN

Bab ini membahas kesimpulan yang didapatkan berdasarkan hasil simulasi dan analisa dari perbandingan hasil filter Kalman, dan *obstacle tracking*.

5.1 Kesimpulan

Dari hasil simulasi *obstacle tracking* dengan menggunakan filter Kalman, dapat diambil beberapa kesimpulan sebagai berikut:

a. Algoritma *obstacle tracking* dengan integrasi data sensor kamera dan LiDar berjalan dengan baik. Hal ini dibuktikan dengan halangan yang tetap terdeteksi sebagai objek yang sama walaupun terjadi persimpangan antar halangan.
b. Estimasi jarak dan sudut relatif halangan terhadap kapal dengan menggunakan filter Kalman menunjukkan rata-rata nilai RMSE 12% lebih kecil dari nilai RMSE pengukuran**,** sehingga masih lebih mendekati nilai pengukuran daripada nilai sebenarnya.
c. Hasil estimasi filter Kalman menunjukkan nilai estimasi yang lebih baik dengan matriks kovarian $\boldsymbol{Q}$ =[2 0 0 0; 0 20 0 0; 0 0 2 0; 0 0 0 20] dan $\boldsymbol{R} = [1\ 0\ 0\ 0;\ 0\ 10\ 0\ 0;\ 0\ 0\ 1\ 0;\ 0\ 0\ 0\ 10]$. Hal ini menunjukkan nilai pengukuran lebih mendekati nilai sebenarnya dari nilai estimasi. Karena nilai kovarians $\boldsymbol{Q}$ akan mempengaruhi perubahan nilai kovarians $\boldsymbol{P}$, sedangkan hubungan nilai kovarians $\boldsymbol{P}$ dengan nilai kovarians $\boldsymbol{R}$ akan mempengaruhi nilai gain Kalman pada setiap iterasi. Hal ini dapat dilihat dari perbandingan RMSE filter Kalman.
d. Didapatkan hasil estimasi filter Kalman terbaik pada halangan tiga, dengan lintasan halangan semakin menjauh dari kapal. Hasil estimasi filter Kalman menunjukkan rata-rata nilai RMSE 22% lebih kecil dari nilai RMSE pengukuran.

5.1 Saran

Beberapa saran yang dapat diberikan untuk penelitian lanjutan dari penelitian ini antara lain :

a. Menggunakan varians noise real dari peralatan sensor pengukuran.
b. Sistem *obstacle tracking* pada tugas akhir ini hanya disimulasikan menggunakan software MATLAB. Sehingga untuk kedepannya dapat diimplementasikan pada wahana tertentu dengan kemampuan sensor yang mendukung.

c. Merumuskan model sistem yang baru untuk mendapatkan nilai estimasi yang lebih baik, dengan menambahkan *state-space* sistem.
d. Menggunakan filter lain untuk meningkatkan presisi nilai estimasi jarak dan sudut relatif halangan terhadap kapal.

DAFTAR PUSTAKA

[1] M. Hossein, “Developing a navigation, guidance and obstacle avoidance algorithm for an Unmanned Surface Vehicle (USV) by algorithms fusion,” 2018, hlm. 56–65.

[2] L. Zhixiang, "Unmanned surface vehicles: An overview of developments and challenges," 2016, hlm 71-93

[3] W. Lei, "Trajectory tracking control of underactuated USV based on modified backstepping approach," 2015, hlm 817-832

[4] C. James B, “Introduction to remote sensing,” 2011, hlm. 243-244.

[5] R. Joseph, “You Only Look Once: Unified, Real-Time Object Detection,” 2015.

[6] R. Joseph, “YOLOv3: An Incremental Improvement,” 2018.

[7] Y. Singh, R. Mehra. “Relative Study of Measurement Noise Covariance R and Process Noise Covariance Q of the Kalman Filter in Estimation,” 2015.

[8] B. Robert G. “Introduction to Random Signal Analysis and Kalman Filtering,” 2012.

LAMPIRAN

Script **main.m**

```
%menentukan objek masuk beam ke-berapa dan
mengambil jarak minimal sebagai jarak objek
j=initLed;
j=initLed;
d=[];
last_timestamp=timestamp(init);
for i=init:data
        if last_timestamp > timestampLed(j)
            j=j+1;
        end
        if (x_center(i,1)<320 &&
y_center(i,1)<240)
            if deg(i,1)<=5;
                d(i,1) = min(ledd(j,55:59));
            elseif deg(i,1)>5 && deg(i,1)<=11;
                d(i,1) = min(ledd(j,48:52));
                elseif deg(i,1)>11 &&
deg(i,1)<=17;
                d(i,1) = min(ledd(j,41:45));
                elseif deg(i,1)>17 &&
deg(i,1)<=23;
                d(i,1) = min(ledd(j,34:38));
                elseif deg(i,1)>23 &&
deg(i,1)<=29;
                d(i,1) = min(ledd(j,27:31));
                elseif deg(i,1)>29 &&
deg(i,1)<=35;
                d(i,1) = min(ledd(j,20:24));
                elseif deg(i,1)>35 &&
deg(i,1)<=41;
                d(i,1) = min(ledd(j,13:17));
                elseif deg(i,1)>41 &&
deg(i,1)<=46;
                d(i,1) = min(ledd(j,6:10));
            end
        elseif (x_center(i,1) > 320 &&
y_center(i,1) <240)
```

```
            if deg(i,1)<=5
                d(i,1) = min(ledd(j,113:117));
            elseif deg(i,1)>5 && deg(i,1)<=11
                d(i,1) = min(ledd(j,106:110));
                elseif deg(i,1)>11 &&
deg(i,1)<=17
                d(i,1) = min(ledd(j,99:103));
                elseif deg(i,1)>17 &&
deg(i,1)<=23
                d(i,1) = min(ledd(j,92:96));
                elseif deg(i,1)>23 &&
deg(i,1)<=29
                d(i,1) = min(ledd(j,85:89));
                elseif deg(i,1)>29 &&
deg(i,1)<=35
                d(i,1) = min(ledd(j,78:82));
                elseif deg(i,1)>35 &&
deg(i,1)<=41
                d(i,1) = min(ledd(j,71:75));
                elseif deg(i,1)>41 &&
deg(i,1)<=46
                d(i,1) = min(ledd(j,64:68));
            end

        elseif (x_center(i,1)< 320 &&
y_center(i,1)>240)
            if deg(i,1)<=5
                d(i,1) = min(ledd(j,171:175));
            elseif deg(i,1)>5 && deg(i,1)<=11
                d(i,1) = min(ledd(j,164:168));
                elseif deg(i,1)>11 &&
deg(i,1)<=17
                d(i,1) = min(ledd(j,157:161));
                elseif deg(i,1)>17 &&
deg(i,1)<=23
                d(i,1) = min(ledd(j,150:154));
                elseif deg(i,1)>23 &&
deg(i,1)<=29
                d(i,1) = min(ledd(j,143:147));
```

```
                elseif deg(i,1)>29 &&
deg(i,1)<=35
                d(i,1) = min(ledd(j,136:140));
                elseif deg(i,1)>35 &&
deg(i,1)<=41
                d(i,1) = min(ledd(j,129:133));
                elseif deg(i,1)>41 &&
deg(i,1)<=46
                d(i,1) = min(ledd(j,122:126));
            end
        elseif (x_center(i,1)>320 &&
y_center(i,1)>240)
            if deg(i,1)<=5
                d(i,1) = min(ledd(j,229:233));
            elseif deg(i,1)>5 && deg(i,1)<=11
                d(i,1) = min(ledd(j,222:226));
                elseif deg(i,1)>11 &&
deg(i,1)<=17
                d(i,1) = min(ledd(j,215:219));
                elseif deg(i,1)>17 &&
deg(i,1)<=23
                d(i,1) = min(ledd(j,208:212));
                elseif deg(i,1)>23 &&
deg(i,1)<=29
                d(i,1) = min(ledd(j,201:205));
                elseif deg(i,1)>29 &&
deg(i,1)<=35
                d(i,1) = min(ledd(j,194:198));
                elseif deg(i,1)>35 &&
deg(i,1)<=41
                d(i,1) = min(ledd(j,187:191));
                elseif deg(i,1)>41 &&
deg(i,1)<=46
                d(i,1) = min(ledd(j,180:184));
            end
        end
       last_timestamp=timestamp(i);
end
%mengubah sudut dan jarak menjadi global
degref= 28.5;
```

```
r=0.15;
for i=1:data
    d_global(i,1)=sqrt(r^2+d(i,1)^2-
2*r*d(i,1)*cosd(180-(deg(i,1)-5.5+28.5)));
    if d_global(i,1)<5
        d_global(i,1)=nan;
    end
end
for i=1:data
    if (x_center(i,1)<320 && y_center(i,1)<240)
    deg_global(i,1) =
acosd(((0.15*cosd(degref))+(d(i,1)*cosd(deg(i,1)
+128.5)))/d_global(i,1));

deg_global(isnan(deg_global))=deg(i,1)+128.5;
    elseif (x_center(i,1) > 320 && y_center(i,1)
<240)
    deg_global(i,1)=
acosd(((0.15*cosd(degref+41))+(d(i,1)*cosd(deg(i
,1)+87.5)))/d_global(i,1));
    deg_global(isnan(deg_global))=deg(i,1)+87.5;
    elseif (x_center(i,1) < 320 &&
y_center(i,1) >240)
    deg_global(i,1)=
acosd(((0.15*cosd(degref+82))+(d(i,1)*cosd(deg(i
,1)+46.5)))/d_global(i,1));
    deg_global(isnan(deg_global))=deg(i,1)+46.5;
    elseif (x_center(i,1) > 320 &&
y_center(i,1) >240)
    deg_global(i,1)=
acosd(((0.15*cosd(degref+123))+(d(i,1)*cosd(deg(
i,1)+5.5)))/d_global(i,1));
    deg_global(isnan(deg_global))=deg(i,1)+5.5;
    end

end
%mengukur lebar aktual objek
for i=1:data
    time_stamp(i,1)=(timestamp(i)-
timestamp(1))/1e+9;
```

```
    deg_global(i,1) = deg_global(i,1) -90;
    bearing(i,1)= deg_global(i,1)*pi/180;
    t(i,1) = d(i,1)*cosd(deg(i,1));
    Tb(i,1)=2*t(i,1)*tand(46/2);
    dpp(i,1)=Tb(i,1)/320;
    lebar(i,1) =width(i,1)*dpp(i,1);
end

%membuat matrix objek terdeteksi setiap frame
% time_stamp=timestamp;
last_timestamp=time_stamp(1);
j=1;
m=1;
matrix_d=[];
matrix_deg=[];
for i=1:size(time_stamp)
   if last_timestamp ~= time_stamp(i)
            j=j+1;
            m=1;
        end
    matrix_d(m,j)= d_global(i,1);
    matrix_deg(m,j)= deg_global(i,1);
    matrix_width(m,j)= lebar;
    matrix_timestamp(m,j)=time_stamp(i,1);
    last_timestamp=time_stamp(i);
    id(m,j)=i;
    m=m+1;
end

%%
matrix_d(isnan(matrix_d))=0;
matrix_width(isnan(matrix_width))=0;
obj=[];
obj_deg=[];
obj_width=[];
obj_timestamp=[];
%membuat matrix awal untuk tracking objek
for j=1:5;
    for i=1:size(matrix_d,1)
    obj(i,j)=matrix_d(i,j);
```

```
    obj_deg(i,j)=matrix_deg(i,j);
    obj_width(i,j)=matrix_width(i,j);
%     obj_bearing(i,j)=matrix_bearing(i,j);
    obj_timestamp(i,j)=matrix_timestamp(i,j);
    obj_id(i,j)=id(i,j);
    end
end
%membuat matrix tracking objek
for j = 5:size(matrix_d,2)+1
    for i=1:size(matrix_d,1)
        m=size(matrix_d,1);
        num=0;
        for k=1:size(obj,1)
        if ((sqrt(((matrix_d(i,j-1))^2+(obj(k,j-
2))^2)-(2*(matrix_d(i,j-1))*(obj(k,j-2))*...
                (abs(cosd(obj_deg(k,j-2)-
matrix_deg(i,j-1))))))))/3<2 &&...
                abs(obj_deg(k,j-2)-
matrix_deg(i,j-1))<7
        obj(k,j-1)=matrix_d(i,j-1);
        obj_deg(k,j-1)=matrix_deg(i,j-1);
        obj_width(k,j-1)=matrix_width(i,j-1);
%         obj_bearing(k,j-1)=matrix_bearing(i,j-
1);
        obj_timestamp(k,j-
1)=matrix_timestamp(i,j-1);
        obj_id(k,j-1)=id(i,j-1);

        break
        else
            num = num+1;
        end
        if num == size(obj,1) &&  matrix_d(i,j-
1)~=0
%objek baru, jarak melebihi threshold dengan
semua objek pada waktu sebelumnya
            if i+m<=size(obj,1) && j-
1<=size(obj,2)
                if obj(i+m,j-1)~=0
                m=m+1
```

```
                end
            end
               obj(i+m,j-1)=matrix_d(i,j-1);
            obj_deg(i+m,j-1)=matrix_deg(i,j-1);
            obj_width(i+m,j-1)=matrix_width(i,j-
1);
            obj_timestamp(i+m,j-
1)=matrix_timestamp(i,j-1);

        end
        end
    end
end
for j=1:size(obj,1)
    for i=3:size(obj,2)
        if obj(j,i)==0
            P_matrix=[];
        end
        if (obj(j,i-2:i))~=0
            if isempty(P_matrix)
            elseif i>8 && obj(j,i-3)==0
d1=obj(j,i);
d2=obj(j,i-1);
d3=obj(j,i-2);
d4=obj(j,i-3);
[x_est,v_est] = estimasi(d1,d2,d3);
x_true=d1;
v_true=(d1-d2)/ts;
s1=obj_deg(j,i);
s2=obj_deg(j,i-1);
s3=obj_deg(j,i-2);
s4=obj_deg(j,i-3);
[s_est,w_est] = estimasi(s1,s2,s3);
s_true=s1;
w_true=(s1-s2)/ts;
P_matrix=zeros(4,4);
P_matrix=([x_true;v_true;s_true;w_true]-
[x_est;v_est;s_est;w_est])*([x_true;v_true;s_tru
e;w_true]-[x_est;v_est;s_est;w_est])';
                end
```

```
            end

end
            if ~isempty(P_matrix)

x_true=obj(j,i);
v_true=obj(j,i)-obj(j,i-1);
s_true=obj_deg(j,i);
w_true=obj_deg(j,i)-obj_deg(j,i-1);

d1=obj(j,i);
d2=obj(j,i-1);
d3=obj(j,i-2);
[a] = estimasi2(d1,d2,d3);
x_est=x_est+((0.5*ts^2)*a);
v_est=v_est+(ts*a);
s1=obj_deg(j,i);
s2=obj_deg(j,i-1);
s3=obj_deg(j,i-2);
[as] = estimasi2(s1,s2,s3);
s_est=s_est+((0.5*ts^2)*as);
w_est=w_est+(ts*as);
if i>3 && obj(j,i-3)==0
P_matrix=(([x_true;v_true;s_true;w_true]-
[x_est;v_est;s_est;w_est])*([x_true;v_true;s_tru
e;w_true]-[x_est;v_est;s_est;w_est])');
end
[P_matrix,x_new,v_new,s_new,w_new,K,P]=KalmanFil
ter(P_matrix,x_true,v_true,s_true,w_true,...
x_est,v_est,s_est,w_est);

%save output
kalman_d(j,i)=x_new;
kalman_v(j,i)=v_new;
kalman_s(j,i)=s_new;
kalman_w(j,i)=w_new;

%update
x_est=x_new;
v_est=v_new;
```

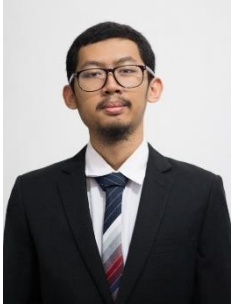
```
s_est=s_new;
w_est=w_new;
            end
        end
    end
```

Script **KalmanFIlter.m**

```
function
[P_new,x_est,v_est,s_est,w_est,K_matrix,P_old] =
KalmanObs2_1(P_old,true_x,true_v,true_s,true_w,e
st_x,...
    est_v,est_s,est_w)

ts=0.1;
Phi_matrix=eye(4);
Phi_matrix(1,2)=ts;
Phi_matrix(3,4)=ts;
% 2. Determine approximate system noise
covariance matrix using (12.82)
Q_matrix = zeros(4);
Q_matrix(2,2)=20;
Q_matrix(4,4)=20;
Q_matrix(1,1)=2;
Q_matrix(3,3)=2;

% 3. Propagate state estimates using (3.14)
noting that all states are zero
% due to closed-loop correction.
x_est_propagated = [est_x;est_v;est_s;est_w];

% 4. Propagate state estimation error covariance
matrix using (3.46)
P_matrix = P_old;
% Phi_matrix * (P_old) * Phi_matrix' + Q_matrix;

% MEASUREMENT UPDATE PHASE

% 5. Set-up measurement matrix using (12.115)
% H_matrix = zeros(3,15);
```

```
H_matrix = eye(4);

% 6. Set-up measurement noise covariance matrix
R_matrix = zeros(4);
R_matrix(1,1) = 1;
R_matrix(3,3) = 1;
R_matrix(2,2) = 10;
R_matrix(4,4) = 10;


% 7. Calculate Kalman gain using (3.21)
K_matrix = P_matrix * H_matrix' * inv(H_matrix *
P_matrix * H_matrix' + R_matrix);

% 8. Formulate measurement innovations
 delta_z(1,1) = true_x - est_x;
 delta_z(2,1) = true_v - est_v;
 delta_z(3,1) = true_s - est_s;
 delta_z(4,1) = true_w - est_w;

% 9. Update state estimates using (3.24)
x_est_new = x_est_propagated + (K_matrix *
delta_z);

% 10. Update state estimation error covariance
matrix
P_new = (eye(4) - K_matrix * H_matrix) *
P_matrix;
P_new = Phi_matrix * (P_new) * Phi_matrix' +
Q_matrix;

% CLOSED-LOOP CORRECTION

% Correction
x_est = x_est_new(1);
v_est = x_est_new(2);
s_est = x_est_new(3);
w_est = x_est_new(4);

end
```

RIWAYAT HIDUP PENULIS



Gaung Jagad lahir pada tanggal 11 Maret 1998 di Blitar. Penulis merupakan mahasiswa Departemen Teknik Elektro, Fakultas Teknologi Elektro dan Informatika Cerdas, Institut Teknologi Sepuluh Nopember, Surabaya. Sebelumnya penulis telah menempuh pendidikan di SD Pertiwi Bogor, SMP Negeri 1 Bogor, dan SMA Negeri 1 Bogor. Selama masa perkuliahan, penulis aktif dalam beberapa organisasi mahasiswa seperti HIMATEKTRO ITS, JMMI ITS, serta Kajian Islam Elektro ITS. Selain itu penulis juga aktif sebagai asisten dari Lab Sistem dan Sibernetika pada bidang studi Teknik Sistem Pengaturan. Untuk menghubungi penulis dapat melalui alamat email gaungj@yahoo.com.