Kalman Filter untuk Mengurangi Derau Sensor *Accelerometer* pada IMU Guna Estimasi Jarak

Muhammad Ari Roma Wicaksono, Freddy Kurniawan*, Lasmadi
Departemen Teknik Elektro, Sekolah Tinggi Teknologi Adisutjipto, Yogyakarta
* email: freddykurniawan@stta.ac.id

Abstract

This study aims to develop a Kalman filter algorithm in order to reduce the accelerometer sensor noise as effectively as possible. The accelerometer sensor is one part of the Inertial Measurement Unit (IMU) used to find the displacement distance of an object. The method used is modeling the system to model the accelerometer system to form mathematical equations. Then the state space method is used to change the system modeling to the form of matrix operations so that the process of the data calculating to the Kalman Filter algorithm is not too difficult. It also uses the threshold algorithm to detect the sensor's condition at rest. The present study had good results, which of the four experiments obtained with an average accuracy of 93%. The threshold algorithm successfully reduces measurement errors when the sensor is at rest or static so that the measurement results more accurate. The developed algorithm can also detect the sensor to move forward or backward.

Keywords — Accelerometer, IMU, Kalman Filter, Noise.

1. Pendahuluan

Dengan perkembangan teknologi saat ini banyak peralatan yang menggunakan konsep kendali jarak jauh bahkan kendali otomatis. Di mana peralatan yang menggunakan sistem kendali otomatis memerlukan beberapa sensor untuk mengetahui situasi lingkungan di sekitarnya. Salah satu alat yang memakai beberapa sensor adalah IMU (*Inertial Measurement Unit*). IMU merupakan suatu alat ukur percepatan, kecepatan, dan orientasi gerak.

IMU ini banyak digunakan dalam sistem navigasi yang kemudian dikombinasikan dengan GPS (*Global Positioning System*). Dengan digabungkannya sistem IMU dan GPS ini, diharapkan keluaran yang dihasilkan lebih akurat [1]. Karena sistem GPS diketahui kurang akurat untuk pengukuran jarak pendek karena faktor jarak antara satelit dan antena penerima, selain itu juga sinyal GPS yang terhalang rintangan seperti gedung dan pepohonan dapat mempengaruhi ketelitian pengukuran jarak berbasis GPS [2]. Sedangkan IMU kurang akurat untuk pengukuran jarak jauh karena adanya akumulasi *error* pada perhitungan [3].

Sistem navigasi pada dasarnya menggunakan suatu sistem yang disebut dengan *dead reckoning*. Berdasarkan sistem tersebut navigasi udara akan mendapatkan beberapa informasi, antara lain posisi dalam koordinat bumi, arah, waktu, dan kecepatan. Dengan memanfaatkan waktu dan kecepatan, maka akan didapatkan informasi jarak perpindahannya.

Setiap sensor memiliki derau atau gangguan yang dapat mempengaruhi hasilnya. Untuk menangani derau yang ada, dapat digunakan *Kalman Filter*. *Filter* ini mempunyai komputasi ringan dan kemampuan yang baik dalam menangani derau dalam beberapa kondisi [4]. Dalam makalah ini, dikembangkan metode *Kalman Filter* untuk menambahkan proses koreksi kondisi statis. Hal ini dilakukan karena pada saat keadaan sensor kembali diam, biasanya masih terdeteksi adanya kecepatan yang bekerja pada sensor.

Manuscript submitted 9 August 2020; revised 28 August 2020; accepted 28 August 2020.
Published 31 August 2020; http://dx.doi.org/10.28989/avitec.v2i2.752

2. Dasar Teori

2.1 *Dead Reckoning*

Dead Reckoning merupakan suatu sistem navigasi dasar untuk menentukan posisi dan mengarahkan pesawat terbang dengan berdasar data arah dan kecepatan dari posisi sebelumnya. Konsep ini adalah dasar untuk semua jenis navigasi udara dan tetap digunakan hingga saat ini. Navigasi adalah riwayat dan prediksi jalur penerbangan pesawat terbang. *Dead reckoning* terdiri empat komponen, yaitu posisi, arah, waktu, dan kecepatan. Posisi adalah koordinat yang menentukan lokasi spesifik pesawat di atas permukaan bumi. Arah adalah pengukuran sudut dari referensi, yang menentukan jalur penerbangan aktual dari titik awal yang diketahui. Sedangkan untuk kecepatan dikalikan waktu akan menghasilkan jarak ditempuh. Kombinasi keempat komponen ini akan memungkinkan kru pesawat menentukan posisi pesawat saat ini atau memperkirakan posisi ke depannya. Seperti halnya hubungan matematika, jika tiga dari empat komponen diketahui, yang keempat dapat ditentukan. [5]

2.2 *Inertial Measurement Unit* (IMU)

Inertial Measurement Unit (IMU) adalah suatu alat elektronik yang memanfaatkan pembacaan dari sensor *gyroscope* dan *accelerometer* untuk mendapatkan nilai perkiraan posisi relatif, kecepatan, serta akselerasi. IMU merupakan bagian dari sistem navigasi yang lebih dikenal dengan nama *Inertial Navigation System* (INS). IMU pertama kali didemonstarikan oleh C.S. Draper pada tahun 1949. IMU sering digunakan pada kendaraan udara untuk bermanuver termasuk UAV dan kendaraan luar angkasa seperti satelit, karena IMU bekerja dengan mendeteksi tingkat percepatan serta perubahan variabel rotasi, termasuk *pitch*, *roll*, dan *yaw*. *Pitch*, *roll*, dan *yaw* masing – masing merupakan rotasi dari ketiga sumbu yaitu sumbu x, sumbu y, dan sumbu z untuk koordinat global. Namun demikian, ketiganya juga dapat mewakili rotasi simbu y, sumbu x, sumbu z untuk koordinat lokal pada beberapa sistem seperti pada *handphone*. Lebih jauh lagi, sudut *roll*, *pitch*, dan *yaw* juga digunakan untuk menetukan orientasi (*attitude*) dari sebuah satelit di ruang angkasa terhadap kutub bumi. [6]

IMU sudah beredar luas secara komersil dengan tipe, ukuran, dan bentuk yang berbeda – beda. Berdasarkan jumlah *Degree of Freedom* (DOF), IMU dapat dibedakan menjadi empat, yaitu IMU dengan tiga buah DOF, IMU dengan lima buah DOF, IMU dengan enam buah DOF, dan IMU dengan sembilan buah DOF. Perbedaannya ada pada jumlah komponen yang digunakan pada IMU tersebut. IMU dengan tiga buah DOF memiliki konfigurasi sensor berupa dua buah *accelerometer* dan satu buah *gyroscope* yang mengukur *yaw*. IMU dengan lima buah DOF memiliki konfigurasi sensor berupa tiga buah *accelerometer* dan dua buah *gyroscope* mengukur *pitch* dan *roll*. IMU dengan enam buah DOF memiliki konfigurasi sensor berupa tiga buah *accelerometer* dan tiga buah *gyroscope* mengukur *pitch*, *roll*, dan *yaw*. IMU dengan Sembilan buah DOF memiliki konfigurasi sensor berupa tiga buah *accelerometer*, tiga buah *gyroscope* mengukur *pitch*, *roll*, dan *yaw*, serta tiga buah *magnetometer*. [6]

Pada sebuah IMU, *gyroscope* dan *accelerometer* digunakan bersama untuk menentukan perpindahan dan *atitude*. Sensor *gyro* berfungsi untuk mengukur kecepatan putar dari sudut *roll*, sudut *pitch*, dan sudut *yaw*. Kecepatan putar adalah perubahan sudut terhadap satuan waktu. *Accelerometer* berfungsi untuk mengukur percepatan dari sebuah benda yang bergerak, seperti satelit, pesawat terbang, atau UAV yang sedang bergerak dengan percepatan tertentu. Selain itu, *accelerometer* pada IMU berfungsi untuk mengkompensasi *drift / error* yang terjadi pada sudut *roll* dan sudut *pitch*, karena *accelerometer* dapat dipakai untuk menentukan sudut – sudut *attitude* seperti sudut *roll* dan sudut *pitch*, meskipun secara langsung *accelerometer* mengukur gaya gravitasi. [6]

2.3 Sensor *Accelerometer*

Accelerometer adalah sensor yang digunakan untuk mengukur percepatan suatu objek. *Accelerometer* mengukur percepatan *dynamic* dan *static*. Pengukuran *dynamic* adalah pengukuran percepatan pada objek bergerak, sedangkan pengukuran *static* adalah pengukuran gravitasi bumi dan ini dapat digunakan untuk mengukur sudut kemiringan [7]. Gaya yang terjadi dalam *accelerometer* dapat berupa gaya statis seperti gaya gravitasi konstan dan gaya dinamis yang disebabkan oleh pergerakan atau getaran dari *accelerometer* [6]. Percepatan merupakan suatu keadaan berubahnya kecepatan terhadap waktu. Bertambahnya suatu kecepatan dalam suatu rentang waktu disebut juga percepatan (*acceleration*). Jika kecepatan semakin berkurang dari pada kecepatan sebelumnya, disebut *deceleration*. Percepatan juga bergantung pada arah / orientasi karena merupakan penurunan kecepatan yang merupakan besaran vektor. Berubahnya arah pergerakan suatu benda akan menimbulkan percepatan pula. Untuk memperoleh data jarak dari sensor *accelerometer*, diperlukan proses integral ganda terhadap keluaran sensor.

$$\vec{s} = \int(\int(\vec{a})\,dt)dt \tag{1}$$

Dimana $\vec{s}$ merupakan vektor jarak yang dicari, dan $\vec{a}$ adalah vektor percepatan sebagai masukan yang diintegralkan sebanyak dua kali terhadap waktu. Proses penghitungan ini dipengaruhi oleh waktu cuplik data, sehingga jeda waktu cuplik data (*dt*) harus selalu konstan dan dibuat sekecil mungkin. [8]

Dalam pengaplikasian *Kalman Filter* diperlukan sebuah pemodelan sistem pada sensor *accelerometer*. Dimana sistem ini dimodelkan menjadi tiga variabel yaitu percepatan (a), kecepatan (v), dan posisi (D) yang dinyatakan dengan Persamaan (2), (3), dan (4).

$$a_k = a_{k-1} \tag{2}$$

$$v_k = v_{k-1} + (a_{k-1} - a_{bk-1})T \tag{3}$$

$$s_k = s_{k-1} + v_{k-1}T + (a_{k-1} - a_{bk-1})\frac{1}{2}T^2 \tag{4}$$

dimana, a_k merupakan percepatan linier dalam waktu k, a_{k-1} sebagai percepatan pada waktu *k-1* (waktu sebelum *k*), v_k adalah kecepatan pada waktu k, v_{k-1} sebagai kecepatan pada waktu *k-1*, a_b sebagai bias *accelerometer*, dan a_{bk-1} sebagai bias *accelerometer* pada waktu *k-1*, *T* merupakan periode sampling, s_k merupakan jarak pergerakan posisi pada waktu k, dan s_{k-1} adalah jarak pergerakan posisi pada waktu *k-1* [9].

Pada bagian ini, *Kalman Filter* dirancang sebagai estimator posisi dari data sensor *accelerometer*. Didasarkan pada Persamaan (2), (3) dan (4), vektor *state* setiap aksis-*j* ($j \in$ {x, y}) pada *accelerometer* berisi jarak pergerakan translasi, kecepatan, dan *bias accelerometer* sebagaimana Persamaan (5).

$$x_{ak} = [s_k \quad v_k \quad a_b]^T \tag{5}$$

Dengan x_k adalah vektor *state* pada waktu k untuk data sensor *accelerometer*, semuanya dalam koordinat sensor [6].

2.4 *Kalman Filter*

Kalman Filter membahas masalah umum untuk memperkirakan keadaan proses waktu diskrit terkontrol yang diatur oleh persamaan linier stokastik :

$$x_k = Cx_{k-1} + Du_k + w_k \quad (6)$$

pada waktu ke-k, dengan x_{k-1} menyatakan vektor keadaan dan u_k sebagai vektor input. Semenetara itu hasil pengukuran dimodelkan dalam z, dengan z merupakan bilangan *real*. [10]

$$z_k = Hx_k + v_k \quad (7)$$

Variabel acak w_k dan v_k menyatakan derau proses dan derau pengukuran [9]. Masing-masing diasumsikan independen dan memiliki probabilitas distribusi normal. Dalam praktiknya, matriks kovarian *noise* dapat berubah pada setiap kali pengukuran, namun dalam hal ini *noise* diasumsikan konstan. C, D, dan H masing – masing merupakan matriks – matriks yang elemennya adalah koefisien dari komponen masing-masing vektor.

Kalman Filter mengestimasi suatu proses dengan menggunakan suatu bentuk kontrol *feedback* yang berulang. Persamaan pada *Kalman Filter* dibagi dalam dua tahap yaitu persamaan pembaruan waktu (*time update*) dan persamaan pembaruan pengukuran (*measurement update*). Persamaan pembaharuan waktu yang ada akan digunakan untuk memproyeksikan (dalam waktu) keadaan saat ini dan estimasi kovarians erornya untuk mendapatkan estimasi *priori* untuk langkah selanjutnya. Selanjutnya persamaan pengukuran yang diperbarui akan digunakan untuk umpan balik, seperti halnya menggabungkan pengukuran baru ke dalam estimasi *apriori* untuk mendapatkan peningkatan estimasi priori. Persamaan pembaruan waktu juga dapat disebut sebagai prediksi, sedangkan persamaan pembaruan pengukuran sebagai koreksi. Persamaan *Kalman Filter* untuk tahapan prediksi disajikan dalam Persamaan (8) dan (9).

$$\hat{x_k}^- = C\hat{x}_{k-1} + Du_k \quad (8)$$

$$P_k^- = CP_{k-1} + C^T + Q \quad (9)$$

dimana $\hat{x_k}^-$ merupakan vektor prediksi dan P_k^- sebagai matriks kovarian prediksi.

Dari tahap prediksi, data hasil pengukuran awal akan digunakan untuk memprediksi variabel keadaan. Selanjutnya variabel keadaan yang diperoleh sebelumnya akan diperbaharui atau dikoreksi. Persamaan *Kalman Filter* untuk pembaruan pengukuran (koreksi) adalah sebagai berikut:

$$K_k = P_k^- H^T (HP_k^- H^T + R)^{-1} \quad (10)$$

Langkah pertama yang dilakukan adalah melakukan perhitungan *Kalman gain* K_k. Selanjutnya dilakukan pembaruan prediksi dengan pengukuran z_k:

$$\hat{x_k} = \hat{x_k}^- + K_k(z_k - H\hat{x_k}^-) \quad (11)$$

Kemudian dilakukan pembaruan kovarians *error*:

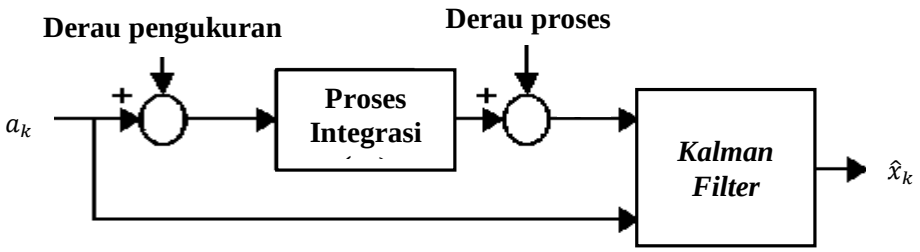
$$P_k = (I - K_k H)P_k^- \quad (12)$$

Variabel keadaan yang telah diperbaharui tersebut akan digunakan untuk prediksi pada kondisi selanjutnya. Proses tahapan tersebut akan dilakukan secara berulang sebanyak diskritisasi waktu yang diperlukan [10].

3. Metode

3.1 Blok Diagram Sistem

Blok diagram sistem Gambar 1 menjelaskan bagaimana proses yang terjadi pada sistem dari data *accelerometer* hingga memperoleh hasil akhir yang berupa informasi jarak. Pertama, data akselerasi dari sensor ditambah dengan derau pengukuran. Setelah itu data yang sudah ditambah derau diolah di tahap integrasi yang berisi algoritma dari x_k. Tetapi di proses tersebut terdapat derau yang dapat mempengaruhi hasil keluarannya, maka hasil dari proses integrasi difilter dengan menambahkan derau yang terjadi dalam proses integrasi dan juga dikoreksi kembali menggunakan beberapa parameter *filter* dan data pengukuran dari sensor. Proses *filter* ini dinyatakan dengan $\hat{x}_k$ [11].



Gambar 1. Blok Diagram Sistem.

3.2 Pengambilan Data Sensor *Accelerometer*

Proses pengambilan data keluaran sensor *accelerometer* dengan menggunakan aplikasi *Sensorstream* IMU+GPS. Data keluaran yang direkam aplikasi tersebut sudah dalam bentuk akselerasi atau percepatan (meter/detik^2). Frekuensi cupliknya sendiri sebesar 50 sampling/detik. Sensor *Accelerometer* BMI160 memiliki nilai resolusi sensor sebesar 16 bit [12].

Gambar 2. Tampilan Aplikasi Sensorstream IMU+GPS.

Langkah awal perekaman data dengan aplikasi Sensorstream IMU+GPS adalah mencentang pada pilihan "*Linear Accel*." seperti yang ditunjukkan Gambar 2 untuk merekam data linear *accelerometer* yang sudah dikurangi dengan nilai gravitasi bumi, dan pilihan "*Include User-Checked Sensor Data in Stream*" untuk menyimpan data yang sudah dicentang. Selanjutnya proses perekaman dimulai dengan memilih frekuensi pada pilihan "*Fast*". Untuk pilihan frekuensi ada beberapa pilihan, antara lain *slow*, *medium*, *fast*, dan *fastest*. Pada penelitian ini menggunakan frekuensi "*Fast*" karena data yang terbaik diperoleh di pilihan frekuensi "*Fast*". Setelah itu pilih mode perekaman menggunakan UDP (*User Data Protocol*) atau penyimpanan perangkat. Penelitian kali ini menggunakan penyimpanan perangkat, sehingga memilih "*SD-Card Stream*", lalu aktifkan perekaman dengan menyentuh fitur "*Switch Stream*". Setelah selesai perekaman sentuh kembali fitur "*Switch Stream*".

3.3 Pemodelan Sistem Sensor *Accelerometer*

Sensor *accelerometer* mengukuran percepatan terhadap tiga sumbu pada bodi. Percepatan setiap poros bodi berada pada sumbu x, y, dan z yang dilambangkan sebagai $\boldsymbol{a_x}$, $\boldsymbol{a_y}$, dan $\boldsymbol{a_z}$. Jika dituliskan dengan vektor $a_i = [a_x \quad a_y \quad a_z]^T$. Parameter penelitian ini dimodelkan dalam tiga variabel: percepatan ($\boldsymbol{a}$), kecepatan ($\boldsymbol{v}$), dan posisi ($\boldsymbol{p}$).

Sensor *accelerometer* secara teori digunakan untuk mengukur percepatan linear dari suatu benda. Tetapi penggunaan sensor *accelerometer* terpengaruh oleh percepatan gravitasi bumi, sehingga data yang diperoleh akan terpengaruh oleh percepatan gravitasi bumi secara vertikal. Maka, pengukuran pada sensor *accelerometer* dikurangi dengan percepatan gravitasi bumi yang mempengaruhi di setiap sumbunya, dan dimodelkan seperti Persamaan 13,

$$a_i = \tilde{a}_i - g \tag{13}$$

dimana a_i merupakan percepatan terukur pada bodi, $\tilde{a}_i$ melambangkan percepatan yang terukur

oleh sensor, dan g adalah percepatan gravitasi bumi yang mempengaruhi di setiap sumbu. Selain percepatan gravitasi bumi, sensor *accelerometer* juga terpengaruh derau dari sensor. Sehingga, persamaan percepatan menjadi

$$a_k = \tilde{a}_k - g + a_{\tilde{k}} \tag{14}$$

dimana, $a_{\tilde{k}}$ adalah derau pada sensor *accelerometer*.

Kecepatan merupakan integral dari percepatan, untuk memperoleh nilai kecepatan digunakan Persamaan 15.

$$v_k = v_{k-1} + a_{k-1}\Delta T + v_{\tilde{k}} \tag{15}$$

Kecepatan ($\boldsymbol{v_k}$) sama dengan kecepatan sebelumnya ($\boldsymbol{v_{k-1}}$) ditambah percepatannya ($\boldsymbol{a_{k-1}}$) yang dikalikan dengan waktu ($\boldsymbol{\Delta T}$) ditambah derau kecepatan yang berubah terhadap waktu ($\boldsymbol{v_{\tilde{k}}}$). Derau yang dimaksud karena pengaruh dari angin maupun gangguan lain yang mengakibatkan pengukuran tidak sesuai.

Jarak merupakan integral dari kecepatan, sehingga diperoleh persamaan jarak seperti pada Persamaan 16.

$$p_k = v_{k-1} + v_{k-1}\Delta T + a_{k-1}\frac{1}{2}\Delta T^2 + p_{\tilde{k}} \tag{16}$$

dimana $\boldsymbol{p_{\tilde{k}}}$ adalah derau jarak.

3.4 *State Space*

Metode *state space* digunakan untuk mengaplikasikan pemodelan sistem ke bentuk operasi matriks. Pada sistem sensor *accelerometer state* proses vektor x dinyatakan sebagai

$$x_k = [p_k \quad v_k \quad a_k]^T \tag{17}$$

sehingga *state* proses sensor *accelerometer* setiap sumbunya dinyatakan seperti Persamaan 18.

$$x_k = [x \quad y \quad z \quad \dot{x} \quad \dot{y} \quad \dot{z} \quad \ddot{x} \quad \ddot{y} \quad \ddot{z}]^T \tag{18}$$

dimana x, y, dan z adalah posisi pada sumbu x, sumbu y, dan sumbu z. Notasi $\dot{x}$, $\dot{y}$, dan $\dot{z}$ adalah kecepatan pada sumbu x, sumbu y, dan sumbu z, sedangkan $\ddot{x}$, $\ddot{y}$, dan $\ddot{z}$ adalah percepatan pada sumbu x, sumbu y, dan sumbu z.

3.5 *State* Matrik Transisi (F), Matrik Kontrol *Input* (G), dan Matriks Pengukuran (H)

State matriks **F** dan **G** digunakan untuk mengubah persamaan pada pemodelan ke bentuk operasi matriks supaya mudah dalam pengolahan data. *State* matriks transisi (**F**) tiga sumbu dalam metode ini dinyatakan sebagai,

$$F = \begin{bmatrix} 1 & dt & dt2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & dt & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & dt & dt2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & dt & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & dt & dt2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & dt \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (19)$$

State matriks kontrol *input* (**G**) tiga sumbu dinyatakan sebagai

$$G = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (20)$$

Sementara itu, untuk *state* matriks pengukuran (**H**) sebagai tiga sumbu dinyatakan sebagai.

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \quad (21)$$

3.6 Kovarian Derau Pengukuran (R)

Derau pengukuran pada penelitian ini diakibatkan dari sensor yang terlalu sensitif, sehingga saat kondisi benda diam terdapat sedikit getaran data tidak bernilai nol. Untuk mengurangi derau tersebut diperlukan matriks kovarian derau pengukuran. Matriks kovarian derau pengukuran ini digunakan dalam algoritma *Kalman Filter* untuk mengurangi dampak dari derau pengukuran. Matriks R pada tiga sumbu dinyatakan sebagai

$$R = cov(derau\ pengukuran) \quad (22)$$

dimana derau pengukuran pada penelitian ini dihitung dengan cara menentukan jumlah data derau pada saat awal sensor aktif hingga sensor mulai digerakkan. Semisal sensor mulai bergerak pada waktu ke 2,46 detik, maka nilai *R* merupakan kovarian dari data pertama hingga data pada waktu ke 2,46 detik.

3.7 Kovarian Derau Proses (Q)

Derau proses merupakan derau yang ada pada sistem. Sama halnya dengan kovarian derau pengukuran, kovarian derau proses juga digunakan untuk memperkirakan dan mengurangi dampak derau terhadap hasil pengukuran pada *Kalman Filter*. Sehingga dengan adanya matriks kovarian R dan Q akan menambah keakurasian sistem. Matriks kovarian derau proses (**Q**) dinyatakan sebagai

$$Q = \begin{bmatrix} Sub\_Q & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & Sub\_Q & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & Sub\_Q \end{bmatrix} \quad (23)$$

dimana Sub_Q dinyatakan sebagai,

$$Sub\_Q = \begin{bmatrix} \Delta t^5/20 & \Delta t^4/8 & \Delta t^3/3 \\ \Delta t^4/8 & \Delta t^3/3 & \Delta t^2/2 \\ \Delta t^3/6 & \Delta t^2/2 & \Delta t \end{bmatrix} \quad (24)$$

3.8 *Kalman Filter*

Kalman Filter dibagi menjadi dua proses, prediksi dan koreksi. Setiap proses prediksi dan koreksi tersebut terdapat beberapa langkah. Dimana dalam proses prediksi ada dua langkah, yaitu prediksi *state* dan prediksi kovarian *error*. Seperti pada Persamaan 25 dan 26, x_k adalah prediksi *state* dan P_k adalah prediksi kovarian *error*.

$$x_k = Fx_{k-1} + Ga_{k-1} \quad (25)$$

$$P_k = FP_{k-1}F^T + Q \quad (26)$$

Proses koreksi terdapat tiga langkah, yaitu perhitungan *kalman gain*, *update* estimasi, dan *update* kovarian *error*. Untuk algoritma dari ketiga langkah tersebut dinyatakan pada Persamaan 27, 28, dan 29, dimana *kalman gain* didefinisikan sebagai K_k, *update* estimasi sebagai $\hat{x}_k$, dan *update* kovarian derau sebagai $\hat{P}_k$.

$$K_k = P_k H^T (HP_k H^T + R)^{-1} \quad (27)$$

$$\hat{x}_k = x_k + K(z_k - Hx_k) \quad (28)$$

dimana z_k adalah pengukuran dari sensor [13].

$$\hat{P}_k = (1 - K_k H)P_k \quad (29)$$

3.9 Koreksi Kondisi Statis

Algoritma yang dirancang belum dapat mendeteksi bahwa sensor dalam keadaan diam setelah bergerak. Sehingga diperlukan suatu algoritma untuk menyatakan bahwa sensor sudah dalam keadaan diam. Dengan demikian pengukuran perpindahannya akan berhenti pada waktu sensor berhenti. Algoritma ini menggunakan sistem ambang batas. Jika tidak menggunakan algoritma ini, pada saat sensor berhenti, algoritma *dead reckoning* dapat mendeteksi bahwa sistem masih bergerak dengan konstan dan pengukuran perpindahan terus bertambah sesuai dengan kecepatan dan waktunya.

Proses koreksi kondisi statis dengan algoritma ambang batas ini memiliki prinsip akan menyatakan benda dalam keadaan diam jika nilai varian data akselerasi kurang dari ambang batas. Nilai ambang batas ditentukan dengan cara *tuning* manual. Nilai tersebut membatasi pengolahan data pada algoritma *Kalman Filter* yang dirancang. Pada penelitian ini nilai varian dihitung untuk data setiap 0,1 detik atau setiap 5 data. Selanjutnya varian data dan nilai ambang batas tersebut digunakan untuk menentukan kondisi sensor dalam keadaan bergerak atau diam.

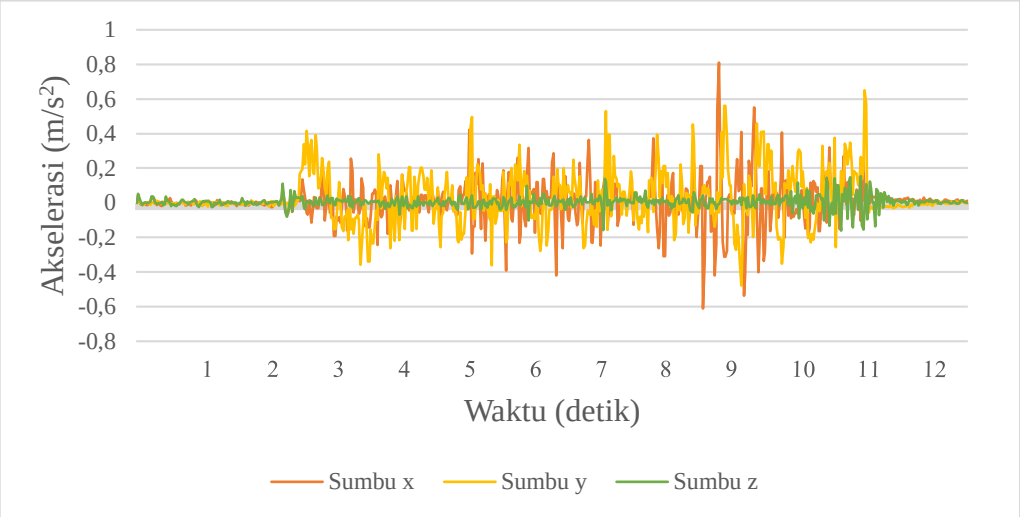
Apabila nilai varian data lebih besar dari nilai ambang batasnya, maka pengukuran

sebelumnya akan dilanjutkan. Sehingga, kecepatan akan terus berubah seiring dengan perubahan nilai percepatan berdasarkan waktunya. Tetapi, apabila nilai varian data kurang dari ambang batas, maka proses pengukuran dihentikan demikian pula proses penghitungan perpindahan. Pada saat itu, kecepatan akan dinyatakan bernilai nol m/s, sehingga tidak ada penambahan jarak perpindahan lagi. Dengan demikian, sensor dinyatakan telah berhenti/diam.

4. Hasil Dan Pembahasan

4.1 Parameter Kovarian Derau Pengukuran (R)

Parameter kovarian derau pengukuran (R) diperoleh dari nilai kovarian derau saat pengukuran. Pada contoh pengukuran jarak satu meter ini derau diambil dari data pertama sampai dengan data pada waktu ke 2,46 detik. Pengambilan data tersebut diambil saat kondisi sensor statis, yang seharusnya tidak ada akselerasi tetapi kenyataannya terdapat akselerasi walaupun kecil, dikarenakan karakter sensor *accelerometer* sensitif terhadap getaran.



Gambar 3. Grafik Data Akselerasi Jarak Satu Meter Sumbu z, y, dan z.

Jika dilihat pada grafik Gambar 3, dari waktu ke nol sampai dengan waktu ke 2,46 detik, nilai akselerasi yang terbaca sensor hanyalah derau pengukuran, karena pada saat itu sensor belum digerakkan. Begitu pula untuk data pada waktu 11,3 detik sampai data terakhir juga merupakan derau pengukuran, di posisi tersebut kondisi sensor statis. Karena saat memasukkan data derau ke algoritma menggunakan nilai data pertama sampai data ke-n, maka untuk mencari n perlu membagi waktu data derau terakhir tersebut dengan waktu cupliknya.

$$n = t/dt \tag{30}$$

Untuk nilai t = 2,46 dan waktu cupliknya 0,02 detik, maka n sama dengan 123. Sehingga nilai parameter **R** adalah kovarian dari data pertama hingga data ke 123 di setiap sumbunya. Untuk nilai **R** hanya perlu mencuplik salah satu rentang derau ketika kondisi statis saja. Rentang derau untuk perhitungan **R** di setiap percobaan ditampilkan pada Tabel 1.

Tabel 1. Jumlah Data Derau.

	Percobaan			
	1 meter	**2 meter**	**3 meter**	**4 meter**
sampai data ke-	123	89	46	121

Dari Tabel 1 diketahui untuk jumlah data derau pada pengukuran satu meter adalah 123 data awal. Untuk pengukuran dua meter jumlah data deraunya adalah 89 data awal. Untuk pengukuran tiga meter jumlah data deraunya adalah 46 data awal. Dan untuk pengukuran empat meter data deraunya adalalh 121 data awal.

4.2 Parameter Ambang Batas

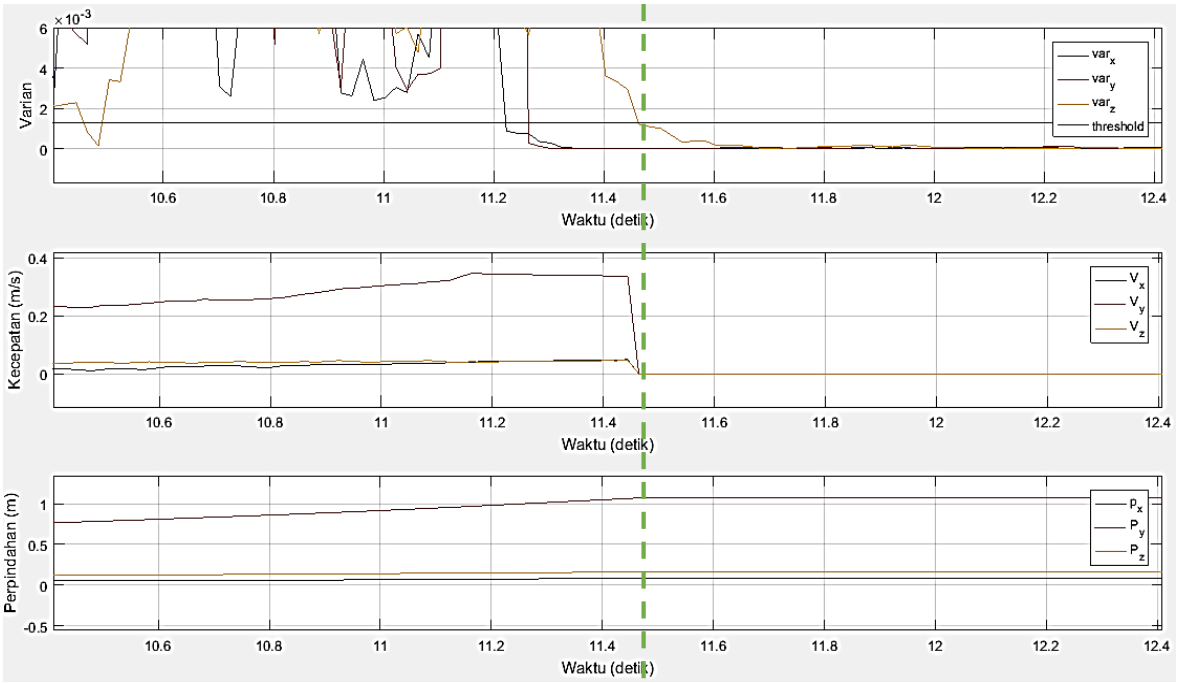
Pada percobaan pengukuran jarak satu meter, nilai ambang batas yang diberikan adalah 0,0013. Dengan nilai ambang batas tersebut diperoleh hasil yang paling efektif dari sistem. Yang mana saat menggunakan algoritma ambang batas, hasil perhitungan sebesar 1,07 meter, sedangkan tanpa algoritma tersebut hasilnya sebesar 1,39 meter, hanya berkurang 0,05 meter saja dari prediksinya sebesar 1,44 meter. Untuk nilai ambang batas setiap percobaan ditunjukkan pada Tabel 2.

Tabel 2. Nilai Ambang Batas di Setiap Percobaan.

	Percobaan			
	1 meter	**2 meter**	**3 meter**	**4 meter**
Ambang batas	0,0013	0,004	0,007	0,001

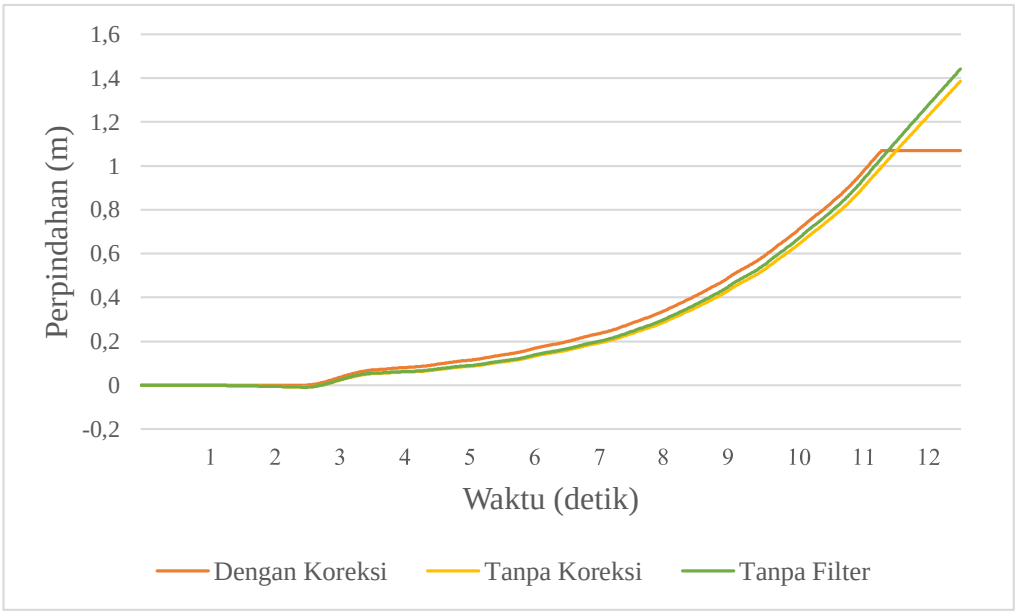
4.3 Proses Koreksi Kondisi Statis

Pada Gambar 4, ambang batas (*threshold*) ditunjukkan dengan garis warna ungu pada grafik varian. Dari grafik tersebut ambang batas membatasi varian derau dengan varian pengukuran. Varian derau adalah varian data derau saat statik, sehingga jika nilai varian data sensor *accelerometer* di bawah nilai ambang batas akan terdeteksi sebagai derau dan sensor dianggap statis atau diam. Dari Gambar 4 juga menunjukkan bahwa pada waktu 11,2 detik (garis hijau putus – putus) nilai varian data sensor *accelerometer* untuk sumbu x (var_x), sumbu y (var_y), dan sumbu z (var_z) berada di bawah nilai ambang batas. Pada saat itu, benda dinyatakan dalam keadaan diam, tidak ada perubahan perpindahan benda, dan grafik kecepatan mengalami penurunan ke arah nol. Dengan metode ini diperoleh hasil yang lebih akurat dibandingkan dengan metode *Kalman Filter* biasa.



Gambar 4. Grafik Pengaruh Ambang Batas terhadap Kecepatan dan Jarak.

Perbandingan hasil pengukuran antara sebelum proses filter, setelah *Kalman Filter*, dan penambahan algoritma ambang batas dalam *Kalman Filter* ditunjukkan pada Gambar 5. Terlihat jelas pengaruh dari proses koreksi dari *Kalman Filter* yang menggunakan algoritma ambang batas. Dari percobaan mengukur jarak satu meter diperoleh hasil tanpa *filter* sebesar 1,44 meter, dengan *Kalman Filter* tanpa koreksi 1,39 meter, dan setelah menerapkan algoritma ambang batas mendapatkan hasil 1,07 meter. Sehingga algoritma ini mampu meningkatkan akurasi dari 61% menjadi 93% pada jarak satu meter.



Gambar 5. Grafik Perbandingan Hasil Pengukuran.

Dari Gambar 5 terlihat perbedaan antara hasil integrasi tanpa *filter*, hasil *Kalman Filter* biasa, dan hasil *Kalman Filter* dengan penambahan koreksi kondisi statis. Hasil pengolahan data jika tanpa filter ditunjukkan oleh grafik warna hijau, dimana hasilnya bernilai 1,44 meter yang masih dipengaruhi oleh derau. Setelah dimasukkan ke algoritma *Kalman Filter* biasa grafik hasil berwarna kuning sedikit lebih turun menjadi 1,42 meter, karena derau yang tadinya mempengaruhi hasil integrasi sudah diredam, sehingga hasilnya menjadi lebih kecil dari hasil sebelumnya namun belum mendeteksi bahwa sensor dalam keadaan diam. Dan untuk hasil setelah ditambah dengan algoritma koreksi kondisi statis pada waktu 11,2 detik pergerakan mulai stabil di jarak 1,07 meter karena kecepatannya sudah dideteksi nol m/s. Selain itu, dengan algoritma koreksi kondisi statis juga mampu mengurangi derau, seperti pada waktu 1,84 detik hingga 2,74 detik dimana tanpa menggunakan algoritma koreksi kondisi statis baik tanpa *filter* maupun dengan *Kalman Filter* biasa sudah terdeteksi pergerakan dengan nilai minus. Sedangkan pada waktu yang sama hasil algoritma koreksi kondisi statis masih bernilai nol meter sehingga grafik untuk hasil koreksi lebih tinggi dibandingkan dengan hasil yang lainnya.

4.4 Galat dan Akurasi Pengukuran

Galat merupakan kesalahan pada pengukuran yang disebabkan oleh beberapa faktor. Dalam penelitian ini, galat dipengaruhi oleh derau saat pengukuran dan derau yang diakibatkan proses perhitungannya. *Kalman Filter* yang digunakan pada penelitian ini dirancang untuk mengurangi kesalahan tersebut dengan mendeteksi derau pengukuran dan derau pada proses perhitungan. Dimana pada saat kondisi sensor statis terdapat derau yang mengakibatkan hasil perhitungan tidak akurat. Nilai galat diperoleh dari Persamaan 31.

$$galat(\%) = \left|\frac{referensi - hasil\ perhitungan}{referensi}\right| x100\% \quad (31)$$

dari percobaan satu meter diperoleh hasil estimasi sebesar 1,44 meter untuk tanpa *filter*, dan 1,18 meter yang menggunakan *Kalman Filter*. Sehingga galat pengukuran sebesar

$$galat = \left|\frac{1-1{,}44}{1}\right| x100\% = 44\% \text{ untuk tanpa } \textit{filter}, \text{ dan} \quad (32)$$

$$galat = \left|\frac{1-1{,}07}{1}\right| x100\% = 7\% \text{ untuk galat setelah difilter.} \quad (33)$$

Akurasi pengukuran digunakan untuk melihat seberapa akurat algoritma yang sudah dirancang. Akurasi pengukuran sendiri diperoleh dari perhitungan Persamaan 34.

$$Akurasi = 100\% - galat(\%) \quad (34)$$

diketahui galat pada pengukuran satu meter adalah 7%, sehingga diperoleh nilai akurasi sebesar 93%. Hasil dari beberapa percobaan beserta galat dan akurasi setiap percobaan ditunjukkan pada Tabel 3.

Tabel 3. Hasil dan Akurasi Pengukuran.

	Percobaan							
	1 meter		**2 meter**		**3 meter**		**4 meter**	
	Tanpa KF	**Dengan KF**	**Tanpa KF**	**Dengan KF**	**Tanpa KF**	**Dengan KF**	**Tanpa KF**	**Dengan KF**
Hasil (meter)	1,44	1,07	3,32	1,85	3,92	2,81	4,99	4,27
Galat (%)	44,00	7,00	66,00	7,50	30,67	6,33	24,75	6,75
Akurasi (%)	56,00	93,00	34,00	92,50	69,33	93,67	75,25	93,25

Hasil percobaan pada penelitian ini untuk pengukuran satu meter setelah menggunakan *Kalman Filter* yang ditambah dengan algoritma koreksi kondisi statis sebesar 1,07 meter, dimana hasil tanpa filternya sebesar 1,44 meter. Sehingga nilai akurasinya meningkat dari 56% menjadi 93%. Selanjutnya, pada pengukuran jarak dua meter diperoleh hasil akhir sebesar 1,85 meter dengan hasil tanpa *filter* sebesar 3,32 meter. Dan nilai akurasinya naik dari 34% menjadi 92,5%. Untuk pengukuran tiga meter diperoleh hasil akhir sebesar 2,81 meter dengan hasil tanpa *filter* sebesar 3,92 meter. Sehingga nilai akurasinya naik dari 69,33% menjadi 93,67%. Dan percobaan terakhir, pengukuran jarak empat meter diperoleh hasil akhir sebesar 4,27 meter dengan hasil tanpa *filter* sebesar 4,99 meter dan nilai akurasinya naik dari 75,25% menjadi 93,25%.

Dari keempat hasil percobaan diatas diketahui bahwa tingkat kesalahan yang terjadi ketika proses integrasi maupun *Kalman Filter* biasa tanpa koreksi berbeda-beda setiap pengukuran. Hal tersebut diakibatkan karena durasi waktu berhenti setelah pengukuran di setiap percobaan berbeda-beda. Apabila waktu diam setelah melakukan pergerakan semakin lama, maka hasil tanpa *filter* dan hasil *Kalman Filter* tanpa proses koreksi kondisi statis akan semakin jauh dari jarak yang sebenarnya. Sebaliknya, apabila semakin cepat waktu diam setelah melakukan pergerakan, maka jarak yang terukur tanpa *filter* dan dengan *Kalman Filter* tanpa proses koreksi kondisi statis akan semakin mendekati jarak sebenarnya. Akan tetapi, dengan penambahan algoritma koreksi kondisi statis, durasi diam setelah melakukan pergerakan tidak mempengaruhi hasil akhirnya.

5. Kesimpulan

Dari hasil perancangan dan pengujian sistem, serta pembahasan, maka dapat ditarik kesimpulan sebagai berikut :

1. Pada penelitian ini menggunakan metode pemodelan sistem dan *state space* sebelum memasukkan data ke dalam algoritma *Kalman Filter*. Metode pemodelan digunakan untuk memodelkan sistem pada sensor *accelerometer* yang keluarannya sebagai data akselerasi ke bentuk persamaan matematis hingga keluar hasil berupa jarak perpindahan. Sementara itu, metode *state space* digunakan untuk mengubah bentuk persamaan dari pemodelan ke dalam operasi matriks. Setelah melalui dua tahap tersebut data diolah menggunakan algoritma dari *Kalman Filter*.
2. Algoritma yang dirancang juga menggunakan algoritma koreksi kondisi statis untuk membatasi proses pengukuran supaya saat benda statis tidak terbaca dalam kondisi bergerak konstan.
3. Dari empat percobaan yang dilakukan diperoleh tingkat akurasinya sebesar 93% pada pengukuran satu meter dengan pengurangan galat dari 44% menjadi 7%. Tingkat akurasi

pada pengukuran dua meter sebesar 92,5% dengan pengurangan galat dari 66% menjadi 7,5%. Tingkat akurasi untuk pengukuran tiga meter sebesar 93,67% dengan pengurangan galat dari 30,67% menjadi 6,33%. Dan tingkat akurasi untuk pengukuran empat meter sebesar 93,25% dengan kemampuan mengurangi galat dari 24,75% menjadi 6,75%.

Daftar Pustaka

[1] Nugroho, T. A., Hutagalung, M., Susantio, M. A., Jeremias, V., & Yonata, Y. (2018). Implementasi Sensor Fusion untuk Peningkatan Akurasi Sensor GPS. *JUPITER (JURNAL PENDIDIKAN TEKNIK ELEKTRO)*, *3*(1), 26-36

[2] Farida, A., & Rosalina, F. (2020). Pelatihan Dasar-Dasar Pengoperasian GPS Garmin Bagi Mahasiswa Fakultas Pertanian Universitas Muhammadiyah Sorong. *Abdimas: Papua Journal of Community Service*, *2*(1), 47-56

[3] Siciliano, B., & Khatib, O. (2008). *Springer Handbook of Robotics.* Heidelberg: Springer

[4] Lasmadi, Cahyadi, A., & Hidayat, R. (2016). Implementasi Kalman Filter untuk Navigasi Quadrotor Berbasis Sensor Accelerometer. *Prosiding SENIATI*, 242-B.

[5] Naval Aviation School Command. (2017). *Introduction to Air Navigation.* Florida: NAVAVSCOLSCOM-SG-200.

[6] Jonathan, N., & Rippun, F. (2016). Implementasi Filter Kalman Pada Sistem Sensor Inertial Measurement Unit (Imu) Quadcopter. *Jurnal Elektro Unika Atma Jaya, 9*(2), 99-110.

[7] Suryanti, D. I. (2017, Desember). Inertial Measurement Unit (IMU) pada Sistem Pengendali Satelit. *Media Dirgantara Vol.12 No.2 Desember 2017*, hal. 7-10.

[8] Alma'i, V. R., Wahyudi, W., & Setiawan, I. (2011). *Aplikasi Sensor Accelerometer Pada Deteksi Posisi* (Doctoral dissertation, Jurusan Teknik Elektro Fakultas Teknik).

[9] Lasmadi. (2019, April). Sistem Navigasi Quadrotor Berbasis IMU dengan Kalman Filter Tuning. *ELKHA, Vol. 11, No.1*, hal. 39-46.

[10] Syarifudin, A. N., Merdekawati, D. A., & Apriliani, E. (2018, Maret). Perbandingan Metode Kalman Filter, Extended Kalman Filter, dan Ensemble Kalman Filter pada Model Penyebaran Virus HIV/AIDS. *Limits: Journal of Mathematics and Its Applications Vol. 15, No. 1*, hal. 17-29.

[11] Becker, A. (2018). *Kalman Filter*. Dipetik Juli 1, 2020, dari KalmanFilter.NET: https://www.kalmanfilter.net/default.aspx

[12] Bosch Sensortec GmbH. (2018, Oktober). *Data Sheet BMI160 Small, Low Power Inertial Measurement Unit*. Reutlingen, Baden-Wuerttemberg, Jerman.

[13] Soebhakti, H., & Fatekha, R. A. (2014). Implementasi Kalman Filter Pada Sensor Jarak Berbasis Ultrasonik. *Jurnal Integrasi*, *6*(2).