

## Penentuan Koreksi Sudut *Attitude* pada *Quadrotor* Menggunakan Algoritma *Zero Acceleration Compensation*

Shandy Avisena, Freddy Kurniawan, Ndaru Atmi Purnami\*  
Program Studi Teknik Elektro, Fakultas Teknologi Industri  
Institut Teknologi Dirgantara Adisutjipto, Yogyakarta  
\*email: ndaru.atmi@gmail.com

### Abstract

*UAV orientation consists of attitude and heading angles. The attitude angle of a quadrotor UAV can be estimated from gyroscope and accelerometer data. It can be predicted from gyroscope data under static or dynamic conditions, but the predicted value has accumulated errors. Meanwhile, the attitude can also be calculated from accelerometer data, but only correct if the sensor is in a static state. To get a more precise attitude angle, the attitude predicted from the gyroscope data and the attitude calculated from the accelerometer data were fused using a Kalman filter. Determination of the condition of the sensor using a threshold value that is applied to moving-covariance of the acceleration data. In this study, the zero-acceleration compensation algorithm is used so that when the sensor is static, the attitude angle is calculated from the accelerometer. Meanwhile, when it is in dynamic condition, the attitude angle uses the predicted results from the gyroscope data. The use of this algorithm can increase the accuracy of the quadrotor orientation for roll angle to 96.84% and pitch angle to 98.91%.*

*Keywords – attitude, Kalman filter, moving-covariance, orientation, zero-acceleration compensation.*

### 1 Pendahuluan

Bidang sistem kendali dan navigasi mengalami perkembangan signifikan termasuk sensor yang diterapkan pada pesawat tanpa awak atau *Unmanned aerial vehicle* (UAV) untuk mendapat informasi mengenai orientasi yang akurat. Pesawat tanpa awak tersebut dilengkapi oleh sensor yang digunakan untuk menentukan orientasi menggunakan sensor *Inertial Measurement Unit* (IMU) [1].

Sensor IMU yang terdiri dari *accelerometer* dan *gyroscope* memiliki karakter berbeda pada tiap sensor. Sensor *accelerometer* dapat memberikan pengukuran sudut kemiringan yang akurat saat statis tetapi sensor sensitif terhadap getaran. Sensor *gyroscope* dapat memberikan pengukuran sudut kemiringan yang rendah akan *noise* dari getaran tetapi menghasilkan galat yang semakin besar dari waktu ke waktu [2].

*Kalman filter* merupakan *filter digital* yang berbasis *low pass filter* dan dapat digunakan untuk mengestimasi nilai orientasi. Proses perhitungan sudut orientasi dilakukan penggabungan data dengan mempropagasikan data kecepatan sudut dari sensor *gyroscope* menjadi data prediksi orientasi. Hasil perhitungan data sensor *gyroscope* dikoreksi dengan data sudut orientasi hasil perhitungan dari data percepatan [3].

Pada penelitian ini dikembangkan algoritma *zero Acceleration Compensation* (ZAC) untuk memberi kompensasi hasil perhitungan orientasi kepada data sensor *accelerometer*. Perhitungan menggunakan *Kalman filter* digunakan untuk menapis derau pada serangkaian pengukuran dan mengestimasi orientasi. Estimasi orientasi menggunakan algoritma ZAC untuk mendapatkan sudut *attitude* yang dihasilkan menjadi lebih akurat.

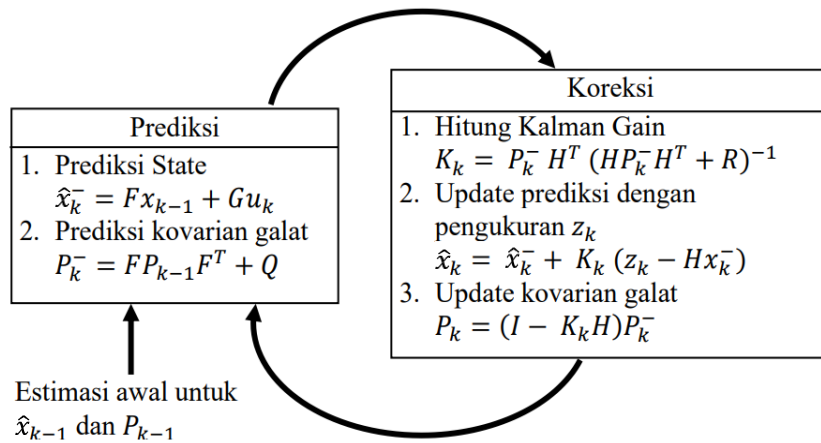
## 2. Metode Penelitian

### 2.1 Kalman Filter

*Kalman filter* merupakan *filter digital* yang berbasis *low pass filter* dengan komputasi yang ringan dan memiliki beberapa keunggulan. Keunggulan dari *Kalman filter* yaitu sebagai penggabungan data (*data fusion*), mengestimasi nilai orientasi, dan mengurangi derau data.

Dalam penelitian ini, *Kalman filter* digunakan untuk menggabungkan data dari sensor *gyroscope* dan *accelerometer* serta menangani derau. Algoritma *zero acceleration compensation* yang dirancang diimplementasikan pada salah satu proses di *Kalman filter*.

*Kalman filter* terdapat dua proses yaitu prediksi dan koreksi. Pada proses prediksi, *Kalman filter* terdapat dua tahapan yaitu prediksi *state* dan matriks kovarian galat. Tahapan yang pertama pada proses prediksi yaitu proses prediksi *state* kemudian dilanjutkan matriks kovarian galat. Tahap kedua yaitu dengan proses menghitung *Kalman gain*, selanjutnya dilakukan *update* prediksi dengan pengukuran  $z_k$  dan proses *update* kovarian galat. Proses ini dilakukan secara terus menerus sehingga mengalami pengulangan seperti Gambar 1.



Gambar 1. Proses di *Kalman Filter* [3]

Dalam mengestimasi suatu kondisi (*state*) dari kondisi awal ( $\hat{x}_k^-$ ) ditentukan pada Persamaan (1).

$$\hat{x}_k^- = \begin{bmatrix} \phi_k \\ \dot{\phi}_k \\ \theta_k \\ \dot{\theta}_k \\ \varphi_k \\ \dot{\varphi}_k \end{bmatrix}^- \quad (1)$$

Di mana  $\theta$ ,  $\phi$ , dan  $\varphi$  merupakan besar sudut sikap *roll*, *pitch* dan *yaw* serta  $\dot{\theta}$ ,  $\dot{\phi}$ , dan  $\dot{\varphi}$  merupakan kecepatan sudut pada sikap *roll*, *pitch* dan *yaw*. Pada Proses prediksi *state* terdapat matriks  $F$  dan Matriks  $G$ . Matriks  $F$  merupakan matriks transisi yang mengonversi *state* sebelumnya ke *state* saat ini. Matriks  $G$  merupakan matriks transisi masukan yang mengkompensasi masukan ke *state* saat ini dan  $dt$  merupakan periode cuplikan data dalam detik. Matriks  $F$  dan matriks  $G$  dapat ditunjukkan oleh Persamaan (2) dan (3).

$$F = \begin{bmatrix} 1 & dt & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & dt & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & dt \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (2)$$

$$G = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3)$$

Proses pertama pada *Kalman filter* yaitu menentukan prediksi *state* awal yang ditunjukkan pada Persamaan (4).

$$\hat{x}_k^- = F\hat{x}_{k-1} + Gu_k \quad (4)$$

$$\begin{bmatrix} \phi_k \\ \dot{\phi}_k \\ \theta_k \\ \dot{\theta}_k \\ \varphi_k \\ \dot{\varphi}_k \end{bmatrix}^- = \begin{bmatrix} 1 & dt & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & dt & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & dt \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \phi_{k-1} \\ \dot{\phi}_{k-1} \\ \theta_{k-1} \\ \dot{\theta}_{k-1} \\ \varphi_{k-1} \\ \dot{\varphi}_{k-1} \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{\phi}_k \\ \dot{\theta}_k \\ \dot{\varphi}_k \end{bmatrix}$$

Setelah menentukan prediksi *state* awal, kemudian menentukan matriks kovarian galat yang ditunjukkan pada Persamaan (5).

$$P_k^- = FP_{k-1}F^T + Q \quad (5)$$

Matriks  $P_k^-$  merupakan matriks kovarian galat pada waktu  $k$ . Matriks  $F$  merupakan model dari matriks transisi. Matriks  $P_{k-1}$  merupakan matriks kovarian galat pada waktu  $k - 1$ .

Matriks kovarian derau proses  $Q$  merupakan matriks yang merepresentasikan besar derau pada sistem dan pengaruh derau tersebut pada *state* yang satu dengan *state* yang lain. Matriks kovarian derau proses  $Q$  ditentukan agar *Kalman filter* dapat memperkirakan dan mengurangi dampak derau terhadap keluaran *Kalman filter*. Model matriks  $Q$  ditentukan dalam Persamaan (6).

$$Q = \begin{bmatrix} Q_\theta & Q_i & Q_i \\ Q_i & Q_\phi & Q_i \\ Q_i & Q_i & Q_\varphi \end{bmatrix} \quad (6)$$

Matriks  $Q_\theta$ ,  $Q_\phi$ , dan  $Q_\varphi$  merupakan matriks kovarian derau proses nilai dari *roll*, *pitch*, dan *yaw*. Matriks  $Q_i$  merupakan matriks nol  $2 \times 2$  dan dalam proses koreksi yaitu menentukan besar nilai *Kalman gain*. Matriks  $Q_i$  ditampilkan dalam Persamaan (7).

$$Q_i = \begin{bmatrix} \frac{dt^3}{3} & \frac{dt^2}{2} \\ \frac{dt^2}{2} & dt \end{bmatrix} \quad (7)$$

*Kalman gain* berfungsi untuk menentukan seberapa banyak bagian dari proses prediksi serta data dari pengukuran yang akan digunakan dan kemudian digabungkan untuk memperbarui *state* pada bagian koreksi yang dapat ditunjukkan dalam Persamaan (8).

$$K_k = P_k^- H^T (H P_k^- H^T + R)^{-1} \quad (8)$$

Di mana nilai  $K_k$  merupakan *Kalman gain* dalam indeks waktu  $k$ . Matriks  $P_k^-$  merupakan matriks kovarian galat pada waktu  $k$ . Matriks  $H$  merupakan model matriks transisi dalam Persamaan (9).

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \quad (9)$$

Matriks  $R$  adalah matriks kovarian derau pengukuran. Matriks  $R$  merepresentasikan distribusi derau pada masukan pengukuran yang diberikan oleh sensor. Matriks  $R$  ditunjukkan dalam Persamaan (10).

$$R = \text{diag} ([\text{cov}(\dot{\theta}) \text{cov}(\dot{\phi}) \text{cov}(\dot{\varphi})]) \quad (10)$$

Nilai  $R$  berisi diagonal dari kovarian derau sudut dari orientasi *pitch*, *roll*, *yaw* di mana  $\dot{\theta}$ ,  $\dot{\phi}$ , dan  $\dot{\varphi}$  merupakan data kecepatan sudut pada sikap *roll*, *pitch* dan *yaw*.

Langkah selanjutnya dalam proses koreksi yaitu *update* estimasi dengan pengukuran  $z_k$ . *State* ini sudah memuat data dari prediksi awal dan pengukuran sensor yang telah diperbarui oleh *Kalman gain*. Persamaan *state* ini dapat ditunjukkan dalam Persamaan (11).

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

$$\begin{bmatrix} \Phi_k \\ \dot{\Phi}_k \\ \theta_k \\ \dot{\theta}_k \\ \varphi_k \\ \dot{\varphi}_k \end{bmatrix} = \begin{bmatrix} \Phi_k^- \\ \dot{\Phi}_k^- \\ \theta_k^- \\ \dot{\theta}_k^- \\ \varphi_k^- \\ \dot{\varphi}_k^- \end{bmatrix} + \begin{bmatrix} K_\Phi & 0 & 0 \\ K_{\dot{\Phi}} & 0 & 0 \\ 0 & K_\theta & 0 \\ 0 & K_{\dot{\theta}} & 0 \\ 0 & 0 & K_\varphi \\ 0 & 0 & K_{\dot{\varphi}} \end{bmatrix} \left( \begin{bmatrix} \Phi_k^z \\ \theta_k^z \\ \varphi_k^z \end{bmatrix} - \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \Phi_k^- \\ \dot{\Phi}_k^- \\ \theta_k^- \\ \dot{\theta}_k^- \\ \varphi_k^- \\ \dot{\varphi}_k^- \end{bmatrix} \right)$$

Di mana nilai  $\hat{x}_k$  merupakan *state* dari sistem pada waktu  $k$ ,  $\hat{x}_k$  adalah prediksi *state* dari sistem pada waktu  $k$ ,  $K_k$  adalah nilai *Kalman gain*,  $z_k$  merupakan pengukuran data dari perhitungan rumus orientasi, dan matriks  $H$  adalah model matriks transisi.

Langkah yang terakhir dalam proses koreksi yaitu menghitung *update* galat kovarian akhir yang ditunjukkan dalam Persamaan (12).

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

Di mana  $P_k$  merupakan galat kovarian pada waktu  $k$ ,  $I$  merupakan matriks identitas  $6 \times 6$ ,  $K_k$  adalah *Kalman gain* dalam indeks waktu  $k$ ,  $H$  merupakan model matriks transisi, dan  $P_k^-$  adalah matriks kovarian galat pada waktu  $k$  [4-6].

## 2.2 Sudut Orientasi

Sudut orientasi merupakan besarnya sudut kemiringan suatu benda terhadap sudut referensi. Sudut orientasi terdiri dari keadaan (*attitude*) dan arah terbang (*heading*) dari UAV. Sudut *attitude* terdiri dari sikap *roll* (sumbu x) dan sikap *pitch* (sumbu y). Arah terbang (*heading*) terdiri dari sikap *yaw* (sumbu z).

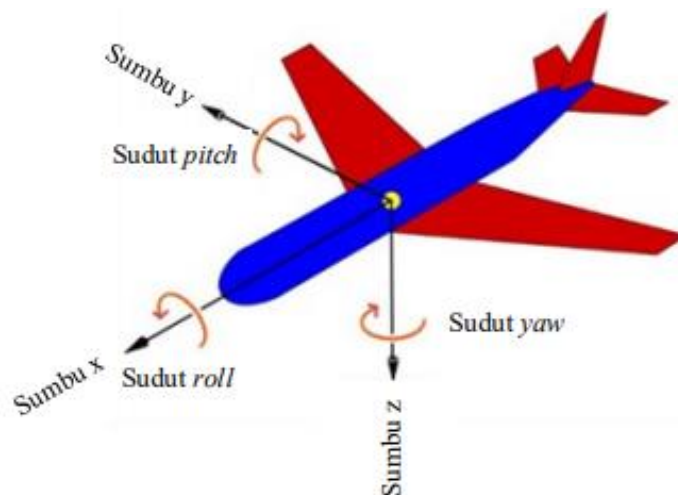
Perhitungan sudut orientasi dilakukan menggunakan sensor IMU yang terdiri dari *accelerometer* dan *gyroscope*. Setiap sensor memiliki kelebihan dan kekurangan yang tidak lepas dari gangguan dan derau yang akan mempengaruhi hasil pengukuran.

Sensor *gyroscope* memiliki kelebihan yaitu dalam perhitungan orientasi dapat digunakan saat statis dan dinamis. Proses Perhitungan sudut orientasi pada sensor *gyroscope* dilakukan dengan proses integral. Perhitungan dengan integral akan menjumlahkan galat yang didapat dan akan terus diakumulasi sampai akhir pengukuran sehingga semakin lama galat yang dihasilkan semakin besar.

Sensor *accelerometer* memiliki kelebihan handal saat statis dan memilki kelemahan dengan galat yang besar saat dinamis. Proses perhitungan sudut orientasi oleh sensor *accelerometer* dilakukan per detik sehingga perhitungan sudut pada detik berikutnya tidak menyebabkan kesalahan.

*Accelerometer* digunakan untuk pengukuran percepatan sebagai masukan dalam menentukan orientasi UAV. Percepatan merupakan suatu keadaan berubahnya kecepatan terhadap waktu. Bertambahnya suatu kecepatan dalam suatu rentang waktu disebut *acceleration*. Kecepatan yang semakin berkurang dari kecepatan sebelumnya disebut *deceleration*. Percepatan juga bergantung pada sudut orientasi karena kecepatan merupakan besaran vektor [7].

Sensor IMU akan menghitung sudut gerak pesawat terhadap ketiga sumbu tersebut untuk dapat mengetahui *attitude* dan *heading* UAV saat terbang. Penelitian ini berfokus pada sudut *attitude* di mana sikap ini terpengaruh langsung oleh percepatan sedangkan arah tidak dipengaruhi oleh percepatan karena gravitasi tegak lurus terhadap bidang horizontal. Sudut orientasi ditampilkan sebagaimana pada Gambar 2.



Gambar 2. Sudut Gerakan *Roll*, *Pitch*, dan *Yaw* [3]

Proses perhitungan sudut orientasi oleh sensor *accelerometer* dilakukan per detik sehingga perhitungan sudut pada detik berikutnya tidak menyebabkan kesalahan. Perhitungan sudut kemiringan untuk sikap *pitch*( $\theta$ ) dan sikap *roll*( $\Phi$ ) dalam Persamaan (13) dan (14) [8].

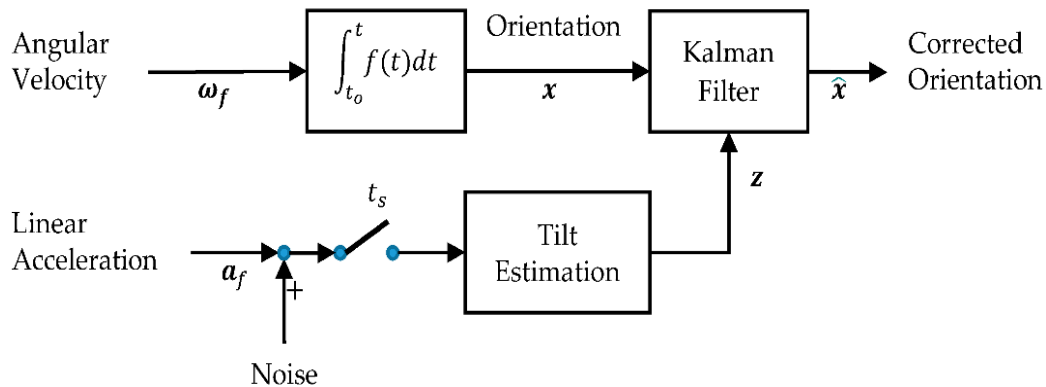
$$\text{roll}(\Phi) = \arctan\left(\frac{a_y}{\sqrt{a_x^2 + a_z^2}}\right) \quad (13)$$

$$\text{pitch}(\theta) = \arctan\left(\frac{a_x}{\sqrt{a_x^2 + a_z^2}}\right) \quad (14)$$

dengan:  $a_x$  = nilai percepatan dalam arah untuk sumbu x,  
 $a_y$  = nilai percepatan dalam arah untuk sumbu y,  
 $a_z$  = nilai percepatan dalam arah untuk sumbu z.

Sensor IMU terdiri dari kombinasi sensor *gyroscope* dan *accelerometer* yang digunakan bersama untuk menentukan sudut *attitude*. Sensor *gyroscope* berfungsi untuk mengukur kecepatan putar dari sudut *roll*, sudut *pitch*, dan sudut *yaw*. Kecepatan putar adalah perubahan sudut terhadap satuan waktu. Sensor *accelerometer* pada IMU berfungsi untuk mengkompensasi galat yang terjadi pada sudut *roll* dan sudut *pitch*. *Accelerometer* dapat dipakai untuk menentukan sudut *attitude* seperti sudut *roll* dan sudut *pitch*, meskipun secara langsung *accelerometer* mengukur gaya gravitasi [9].

IMU mengukur nilai percepatan linear ( $a_f = [a_x, a_y, a_z]$ ) dari *accelerometer* dan nilai kecepatan sudut ( $\omega_f = [\omega_x, \omega_y, \omega_z]$ ) dari *gyroscope*. Proses estimasi orientasi menggabungkan sensor dapat dilihat pada pemodelan dalam diagram blok Gambar 3.



Gambar 3. Diagram blok Estimasi Orientasi *Fusion Sensor* [10]

Kecepatan sudut ( $\omega_f$ ) diukur dengan *gyroscope* dan diintegrasikan satu kali di mana  $f$  merupakan fungsi kecepatan sudut orientasi yang diintegrasikan menjadi fungsi sudut dan  $dt$  adalah durasi waktu. Perhitungan dengan sensor *gyroscope* untuk mendapatkan estimasi orientasi ( $x$ ). Percepatan ( $a_f$ ) dari *accelerometer* digunakan untuk mendapatkan estimasi kemiringan ( $z$ ) untuk hasil perhitungan *roll* dan *pitch* selama periode stasioner ( $t_s$ ). *Kalman filter* menggabungkan prediksi orientasi ( $x$ ) berbasis *gyroscope* dan perhitungan kemiringan ( $z$ ) berbasis *accelerometer* untuk menetapkan orientasi terkoreksi  $\hat{x}$  [10].

Periode stasioner ( $t_s$ ) merupakan pengukuran menggunakan *accelerometer* dengan mengukur nilai gravitasi di setiap sumbu selama waktu tertentu. Dalam penentuan periode stasioner digunakan ambang batas pada besaran nilai data *accelerometer* untuk pendeteksian dalam kondisi statis sekaligus digunakan untuk mengurangi efek *noise*. Periode stasioner saat kondisi statis di mana fase keluaran *gyroscope* bernilai kecil dan besaran keluaran nilai *accelerometer* mendekati nilai gravitasi sehingga *switch* akan *on*.

### 2.3 Penentuan Kondisi Statis dan Dinamis

Dalam penentuan kondisi statis dan dinamis sensor dapat dilakukan dengan mengukur kovarian nilai percepatan. Kondisi dinamis ditandai dengan adanya percepatan linear yang bekerja pada sensor. Adanya percepatan linear yang bekerja pada sensor menjadikan nilai percepatan berubah-ubah.

Perhitungan besarnya perubahan nilai percepatan dilakukan dengan menghitung *moving covariance*. Perhitungan ini dapat dilakukan dengan cara menghitung nilai kovarian atas nilai data percepatan untuk beberapa data terakhir. Banyaknya data terakhir yang digunakan untuk menentukan nilai kovarian didefinisikan sebagai  $w$ . Nilai total *moving covariance* ketiga sumbu dapat dihitung menggunakan Persamaan (17).

$$\sigma_a = \sum cov (a_{i:(k-w) \sim k}) \quad (17)$$

dengan:  $\sigma_a$  = total kovarian nilai percepatan untuk ketiga sumbu

$a_i$  = data percepatan setiap sumbu

$k$  = indeks saat ini

$w$  = lebar jendela kovarian atau banyak data yang dihitung

Nilai kovarian tersebut dibandingkan dengan nilai ambang batas (*threshold*). Jika nilai kovarian kurang dari nilai *threshold* ( $\sigma_{th}$ ) maka sensor dinyatakan dalam keadaan statis. Hal ini berarti sensor tidak mengalami percepatan linear meskipun sensor ini dapat mengalami translasi dengan kecepatan konstan. Penentuan kondisi dinyatakan dalam kondisi statis jika memenuhi syarat pada Persamaan (18).

$$static = \sigma_a < \sigma_{th} \quad (18)$$

dengan:  $\sigma_a$  = nilai total kovarian nilai percepatan untuk ketiga sumbu

$\sigma_{th}$  = nilai ambang batas kovarian percepatan (*threshold*)

*Threshold* dirancang untuk dapat mendeteksi bahwa sensor dalam keadaan diam atau bergerak sehingga diperlukan suatu algoritma untuk menyatakan bahwa sensor sudah dalam keadaan diam. Proses koreksi kondisi statis dengan nilai *threshold* membatasi proses pengukuran pada nilai varian data. Nilai tersebut membatasi pengolahan data pada *Kalman filter* [11].

### 2.4 Algoritma Zero Acceleration Compensation

*Zero acceleration compensation* merupakan metode untuk memberikan kompensasi perhitungan *attitude* menggunakan data sensor *accelerometer* saat percepatan statis dan apabila percepatan dideteksi dinamis perhitungan *attitude* kembali menggunakan data sensor *gyroscope*. Kondisi statis pada pesawat diterapkan pada saat percepatan yang terukur bernilai nol. Penerapan kondisi dinamis pada pesawat terjadi saat percepatan positif (*acceleration*) yaitu saat proses *take off* dan percepatan negatif (*deceleration*) saat proses *landing*. Dalam mengatasi karakter dari sensor IMU dilakukan penentuan kondisi dalam untuk mendapatkan perhitungan orientasi yang lebih akurat.

Penentuan kondisi dilakukan dengan menghitung nilai kovarian dari sensor *accelerometer* di mana perhitungan *accelerometer* diterapkan fungsi *switch*. *Switch* akan *on* jika dalam kondisi tidak ada percepatan (*zero acceleration*) atau saat percepatan di bawah ambang batas (*threshold*). Saat *switch on* hasil perhitungan *gyroscope* dikoreksi menggunakan hasil perhitungan dari *accelerometer* sehingga hasil pengukuran keduanya digunakan sebagai estimasi orientasi. Saat *switch off* perhitungan *gyroscope* tidak dikoreksi

dari perhitungan *accelerometer* sehingga hanya hasil perhitungan *gyroscope* yang akan digunakan sebagai estimasi orientasi. Implementasi algoritma ZAC yang dirancang untuk dieksekusi di dalam *Kalman filter* dalam urutan proses sebagaimana Algoritma 1.

Algoritma 1. *Zero Acceleration Compensation*

Proses	Notasi
Baca data <i>accelerometer</i>	$(a_x, a_y, a_z)$
Baca data <i>gyroscope</i>	$(\dot{\theta}, \dot{\Phi}, \dot{\psi})$
Tentukan sudut <i>Pitch</i>	$(\theta) = \arctan \left( \frac{a_x}{\sqrt{a_x^2 + a_z^2}} \right)$
Tentukan sudut <i>Roll</i>	$(\Phi) = \arctan \left( \frac{a_y}{\sqrt{a_x^2 + a_z^2}} \right)$
Masukkan data <i>accelerometer</i>	$z_k = (\theta, \Phi)$
Masukkan data <i>gyroscope</i>	$u_k = (\dot{\theta}, \dot{\Phi}, \dot{\psi})$
Hitung varian dari data percepatan	$\text{acc\_var} = \text{var}(z_k)$
Hitung prediksi <i>state</i>	$\hat{x}_k^- = A\hat{x}_{k-1} + Bu_k$
Hitung prediksi kovarian	$P_k^- = AP_{k-1}A^T + Q$
Hitung <i>Kalman gain</i>	$K_k = P_k^- H^T (HP_k^- H^T + R)^{-1}$
Berikan kompensasi saat tidak ada percepatan	if $\text{acc\_var} < \sigma_{th}$ maka $K_k = 0$
Hitung update prediksi dengan pengukuran $z_k$	$\hat{x}_k = \hat{x}_k^- + K_k (z_k - H\hat{x}_k^-)$
Hitung update galat kovarian	$P_k = (I - K_k H)P_k^-$

Untuk mendapatkan prediksi *state* saat ini digunakan Persamaan (4) di mana hasil yang didapatkan merupakan hasil dari *prediksi state*. Perhitungan orientasi menggunakan data dari *gyroscope* ( $u_k$ ) yang mana data kecepatan sudut dipropagasikan menjadi sudut.

Proses selanjutnya yaitu menghitung galat kovarian ( $P_k^-$ ) dan perhitungan *Kalman gain* ( $K_k$ ). Pada proses *update* estimasi ( $\hat{x}_k$ ), nilai hasil pengukuran hanya menggunakan sensor *gyroscope* dan belum dikoreksi oleh sensor *accelerometer*. Selanjutnya, masukkan data dari sensor *accelerometer* ( $z_k$ ) merupakan hasil perhitungan sudut sikap *roll* pada Persamaan (13) dan sudut sikap *pitch* Persamaan (14).

Perhitungan data orientasi akan diberi kompensasi menggunakan data *accelerometer* apabila data percepatan kurang dari ambang batas maka nilai *Kalman gain* bernilai nol. Metode dalam menghitung data percepatan yaitu dengan menggunakan varian dengan menghitung ragam data. Data yang terekam adalah 6060 data selama 120 detik dan data yang dicuplik adalah 5 data tiap 0,1 detik. Ragam data varian per 5 data dilakukan perhitungan penjumlahan banyak data untuk mendapatkan rata-rata. Varian data dan nilai *threshold* digunakan untuk menentukan kondisi sensor dalam keadaan statis atau dinamis apabila rata-rata kurang dari *threshold* ( $\sigma_{th}$ ) dikatakan dalam kondisi statis.

Penentuan nilai *threshold* dengan cara proses *tuning*. Proses koreksi kondisi statis dengan *threshold* memiliki prinsip yaitu membatasi proses pengukuran pada nilai varian data. Proses *update* prediksi dengan pengukuran  $z_k$  dilakukan penggabungan data dari sensor *gyroscope* dan sensor *accelerometer*.

## 2.5 Galat

Galat atau *error* dalam metode numerik adalah selisih antara yang ditimbulkan antara nilai sebenarnya dengan nilai yang dihasilkan dengan metode numerik. Besarnya kesalahan atas suatu nilai taksiran dapat dinyatakan secara kuantitatif dan kualitatif. Besarnya kesalahan yang dinyatakan secara kuantitatif disebut galat *absolute*. Besarnya kesalahan yang



dinyatakan secara kualitatif disebut dengan galat relatif. Galat *absolute* dari pengukuran didefinisikan sebagai selisih antara nilai sebenarnya (*true value*) dengan nilai hasil pengukuran (*measured value*) dalam Persamaan (19) [12].

$$\text{Galat absolute} = |\text{Nilai Pengukuran} - \text{Nilai Sebenarnya}| \quad (19)$$

Rumus galat relatif dapat dilihat pada Persamaan (20).

$$\text{Galat Relative} = \left| \frac{\text{Galat absolute}}{\text{Nilai Sebenarnya}} \right| \times 100\% \quad (20)$$

Pada penelitian ini perekaman data dilakukan dengan memasang *software sensorstream* IMU+GPS untuk mendapatkan nilai orientasi pada perangkat HP. Orientasi HP merupakan perhitungan pada *software* HP yang dapat melakukan perhitungan orientasi secara langsung dan dapat digunakan sebagai validator atau nilai sebenarnya.

Dalam mencari besar perbedaan galat dapat dicari berdasarkan sebelum menerapkan algoritma ZAC dan setelah menerapkan algoritma ZAC. Sebelum menerapkan algoritma ZAC merupakan pemodelan perhitungan estimasi tanpa ditambah algoritma. Setelah menerapkan algoritma ZAC merupakan pemodelan perhitungan estimasi orientasi setelah ditambahkan algoritma ZAC. Perbedaan sebelum dan setelah ditambahkan algoritma *zero acceleration compensation* menggunakan Persamaan (21).

$$\text{Perbedaan} = \text{Sebelum diterapkan algoritma} - \text{setelah diterapkan algoritma} \quad (21)$$

Akurasi pengukuran digunakan untuk melihat seberapa akurat algoritma yang sudah dirancang. Akurasi pengukuran diperoleh dari perhitungan Persamaan (22).

$$\text{Akurasi} = 100\% - \text{galat}(\%) \quad (22)$$

### 3. Hasil dan Pembahasan

#### 3.1 Pengambilan Data UAV

Penelitian ini difokuskan pada penentuan sudut orientasi UAV berjenis *quadrotor*. Data yang direkam menggunakan data dari sensor *IMU* dengan menggunakan *smartphone*. Wahana yang digunakan adalah UAV DJI Mavic 2 Pro. Hasil data dari aplikasi IMU dalam *microsoft excel* berupa data dalam format csv kemudian mengubah format data menjadi *xlsx* dan dimasukkan serta diolah menggunakan *software* Matlab. Pengambilan data menggunakan *smartphone* yang diletakan di atas body UAV sebagaimana Gambar 4.



Gambar 4. Pengambilan Data Menggunakan UAV

*Quadrotor* bergerak dari titik awal yang telah ditentukan sebagai *home* bergerak membentuk lintasan persegi panjang dan kembali ke titik awal. *Track* drone ditampilkan pada Gambar 5.

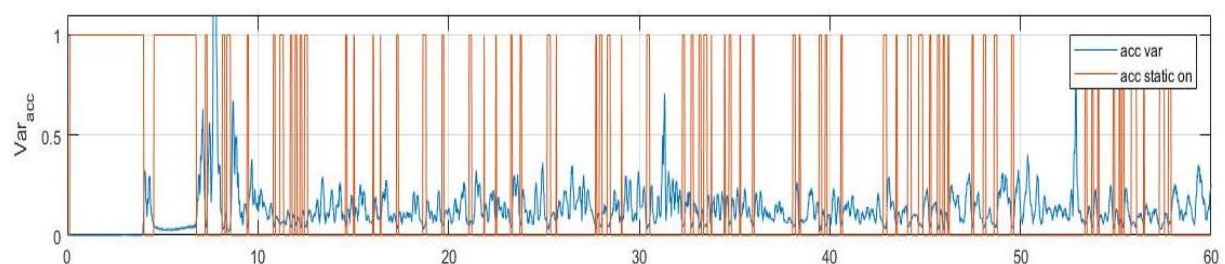


Gambar 5. Jalur lintasan UAV saat pengambilan data

### 3.2 Hasil Penentuan Kondisi

Dalam penggunaan algoritma *zero acceleration compensation* didapat hasil yang dapat dilihat pada proses penentuan kondisi. Proses penentuan kondisi menunjukkan seberapa banyak frekuensi sensor *accelerometer* saat statis digunakan dan frekuensi sensor *gyroscope* saat dinamis kembali digunakan.

Penelitian ini menampilkan penentuan kondisi statis dan dinamis dari varian data percepatan dimana nilai *threshold* untuk membatasi nilai varian data percepatan. Percepatan nol berarti nilai percepatan varian nol dan apabila grafik naik maka percepatan juga naik begitu juga sebaliknya. Nilai varian data dalam kondisi statis dan dinamis ditampilkan pada Gambar 6.

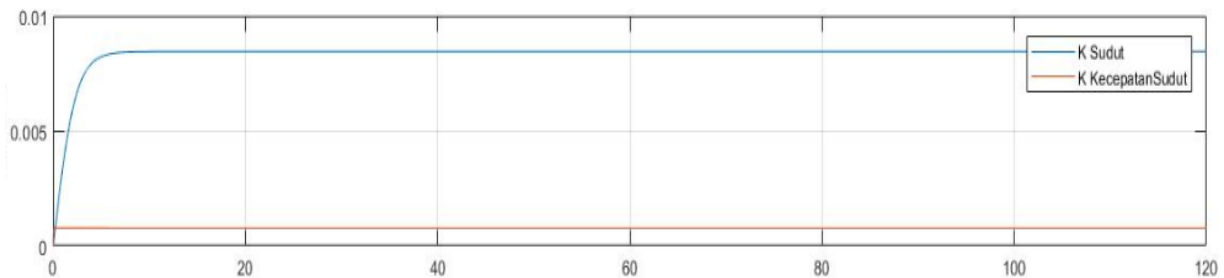


Gambar 6. Varian data kondisi statis dan dinamis

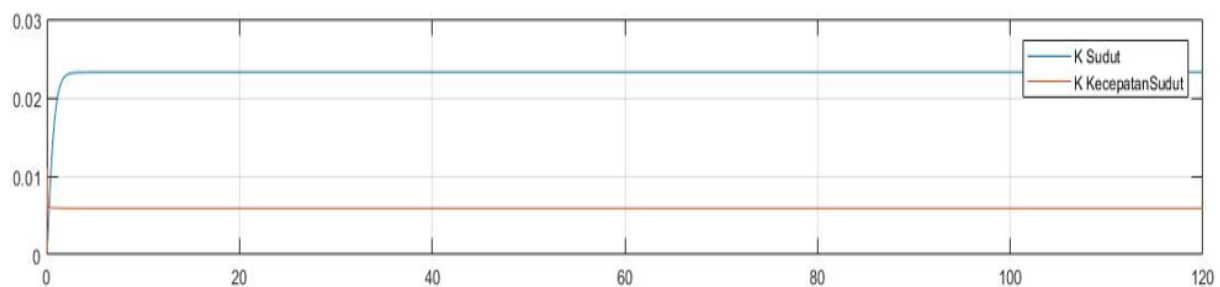
Acc var merupakan varian dari nilai percepatan yang diwakili oleh garis biru merupakan data nilai gabungan dari 3 sumbu varian percepatan. Acc static on merupakan penentuan kondisi statis dan dinamis diwakili dengan garis merah dengan menerangkan saat statis bernilai 1 dan dinamis bernilai 0.

### 3.3 Proses *Kalman Filter*

Pada penelitian ini ditampilkan hasil grafik sebelum menerapkan algoritma ZAC dan setelah menerapkan algoritma ZAC. Pemodelan *Kalman filter* sebelum menerapkan algoritma ditampilkan pada Gambar 7 dan Gambar 8.

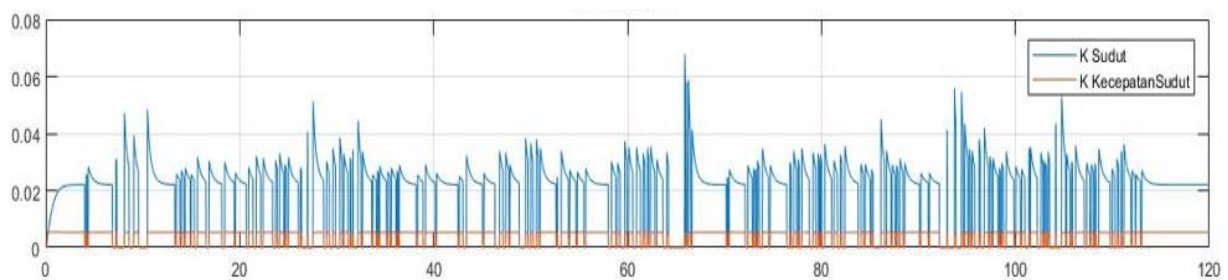


Gambar 7. *Kalman gain* sebelum diterapkan algoritma ZAC untuk sikap *roll*

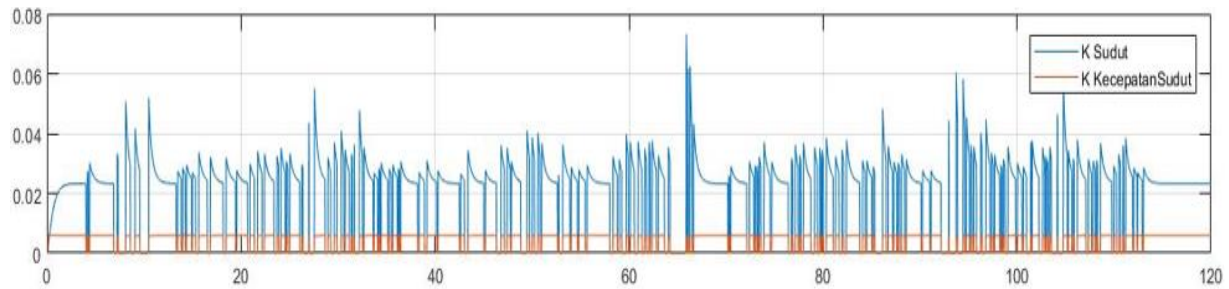


Gambar 8. *Kalman gain* sebelum diterapkan algoritma ZAC untuk sikap *pitch*

Hasil grafik pemodelan sebelum menerapkan algoritma pada sudut dan kecepatan sudut nilai *Kalman gain* yang terukur bernilai konstan. *Kalman gain* pada pemodelan sebelum algoritma merupakan pemodelan estimasi orientasi tidak diberi penambahan *threshold* dan pengukuran percepatan. Pemodelan sebelum algoritma tidak menampilkan penentuan kondisi.



Gambar 9. Nilai *Kalman gain* pada estimasi sudut *roll* setelah diterapkan Algoritma ZAC



Gambar 10. Nilai *Kalman gain* pada estimasi sudut *pitch* setelah diterapkan algoritma ZAC

Pada Gambar 9 dan Gambar 10 terlihat bahwa grafik yang ditampilkan mengalami kondisi yang berubah-ubah (*fluktuatif*) dan merupakan hasil setelah penambahan algoritma *zero acceleration compensation*. Pada saat tidak ada percepatan *Kalman gain* bernilai nol. Pada saat *Kalman gain* memiliki bobot yang tinggi hasil perhitungan banyak dikoreksi menggunakan data dari sensor *accelerometer*. Saat *Kalman gain* memiliki bobot yang rendah hasil perhitungan menggunakan data sensor *gyroscope*. Nilai *K* sudut merupakan besar nilai *Kalman gain* untuk sudut pergerakan benda. Nilai *K* Kecepatan sudut merupakan nilai *Kalman gain* untuk kecepatan sudut pergerakan benda.

### 3.4 Galat Hasil Sudut *Attitude*

#### 1. Galat Sebelum Menggunakan Algoritma *Zero Acceleration Compensation*.

Berikut merupakan galat sebelum algoritma *zero acceleration compensation* pada sikap *roll*.

$$\begin{aligned} \text{Error Absolute} &= |\text{Orientasi Kalman (Roll)} - \text{Orientasi HP (Roll)}| \\ &= |0.929^\circ - 0.214^\circ| \\ &= 0.715^\circ \end{aligned}$$

$$\begin{aligned} \text{Error Relative} &= \frac{\text{Error absolute}}{\text{Orientasi Hp (Roll)}} \times 100\% \\ &= \frac{0.715}{0.214} \times 100\% \\ &= 3.34 \times 100\% \\ &= 3.34 \% \end{aligned}$$

Berikut merupakan galat sebelum algoritma *zero acceleration compensation* pada sikap *pitch*.

$$\begin{aligned} \text{Error Absolute} &= |\text{Orientasi Kalman (Pitch)} - \text{Orientasi HP (Pitch)}| \\ &= |0.532^\circ - 0.223^\circ| \\ &= 0.309^\circ \end{aligned}$$

$$\begin{aligned} \text{Error Relative} &= \frac{\text{Error absolute}}{\text{Orientasi Hp (Pitch)}} \times 100\% \\ &= \frac{0.309^\circ}{0.223^\circ} \times 100\% \\ &= 1.387 \times 100\% \\ &= 1.38 \% \end{aligned}$$

## 2. Galat setelah menggunakan Algoritma *Zero Acceleration Compensation*.

Berikut merupakan galat setelah algoritma *zero acceleration compensation* pada sikap *roll*.

$$\begin{aligned} \text{Error Absolute} &= |\text{Orientasi Kalman (Roll)} - \text{Orientasi HP (Roll)}| \\ &= |0.891^\circ - 0.214^\circ| \\ &= 0.677^\circ \end{aligned}$$

$$\begin{aligned} \text{Error relative} &= \frac{\text{Error absolute}}{\text{Orientasi Hp (Roll)}} \times 100\% \\ &= \frac{0.677^\circ}{0.214^\circ} \times 100\% \\ &= 3.16 \times 100\% \\ &= 3.16\% \end{aligned}$$

Berikut merupakan galat setelah algoritma *zero acceleration compensation* pada sikap *pitch*.

$$\begin{aligned} \text{Error Absolute} &= |\text{Orientasi Kalman (Pitch)} - \text{Orientasi HP (Pitch)}| \\ &= |0.468^\circ - 0.223^\circ| \\ &= 0.245^\circ \end{aligned}$$

$$\begin{aligned} \text{Error relative} &= \frac{\text{Error absolute}}{\text{Orientasi Hp (Pitch)}} \times 100\% \\ &= \frac{0.245^\circ}{0.223^\circ} \times 100\% \\ &= 1.09 \times 100\% \\ &= 1.09\% \end{aligned}$$

## 3. Selisih dan Akurasi pada Sudut *Attitude*

Berikut merupakan selisih galat sebelum menerapkan algoritma ZAC dan setelah menerapkan algoritma ZAC pada sikap *roll*.

$$\begin{aligned} \text{Selisih} &= \text{Sebelum algoritma ZAC} - \text{setelah algoritma ZAC} \\ &= (3.34\% - 3.16\%) \\ &= 0.18\% \end{aligned}$$

Berikut merupakan hasil akurasi perbedaan sebelum menerapkan algoritma ZAC dan setelah menerapkan algoritma ZAC pada sikap *roll*.

$$\begin{aligned} \text{Akurasi (Roll)} &= 100\% - \text{sebelum algoritma ZAC} \\ &= 100\% - 3.34\% \\ &= 96.66\% \end{aligned}$$

$$\begin{aligned} \text{Akurasi (Roll)} &= 100\% - \text{setelah algoritma ZAC} \\ &= 100\% - 3.16\% \\ &= 96.84\% \end{aligned}$$

Berikut merupakan selisih galat sebelum menerapkan algoritma ZAC dan setelah menerapkan algoritma ZAC pada sikap *pitch*.

$$\begin{aligned} \text{Selisih} &= \text{Sebelum algoritma ZAC} - \text{setelah algoritma ZAC} \\ &= (1.38\% - 1.09\%) \\ &= 0.29\% \end{aligned}$$

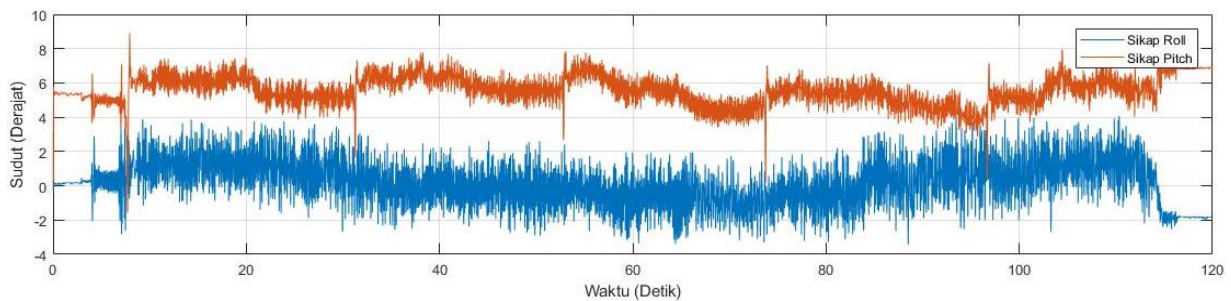
Berikut merupakan hasil akurasi perbedaan sebelum menerapkan algoritma ZAC dan setelah menerapkan algoritma ZAC pada sikap *pitch*.

$$\begin{aligned}\text{Akurasi (Pitch)} &= 100\% - \text{sebelum algoritma ZAC} \\ &= 100\% - 1.38\% \\ &= 98,62\%\end{aligned}$$

$$\begin{aligned}\text{Akurasi (Pitch)} &= 100\% - \text{setelah algoritma ZAC} \\ &= 100\% - 1.09\% \\ &= 98,91\%\end{aligned}$$

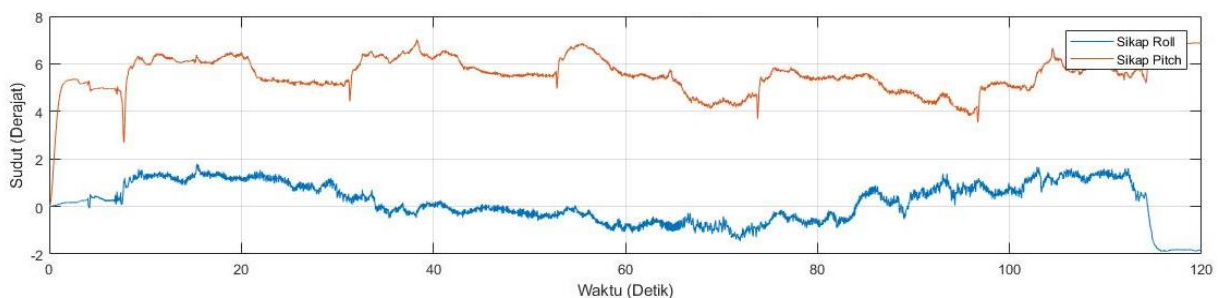
### 3.5 Perubahan Hasil Sudut *Attitude*

Berikut merupakan data hasil perhitungan sudut *attitude* dengan menampilkan grafik perekaman data menggunakan *software* HP. Sebelum penambahan algoritma yaitu pemodelan orientasi dan setelah penerapan algoritma yaitu mendapatkan orientasi yang akurat dengan implementasi algoritma *zero acceleration compensation*.



Gambar 11. Sudut *attitude* hasil hitungan *software* HP

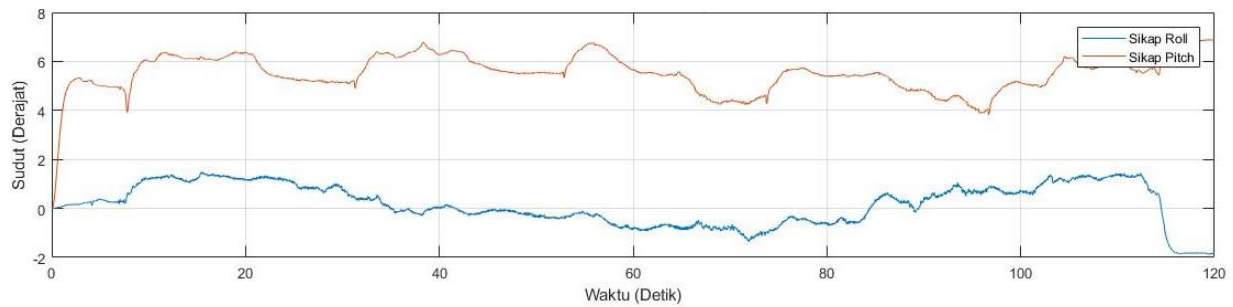
Pada Gambar 11 merupakan hasil perhitungan orientasi menggunakan HP pada sikap *roll* dan sikap *pitch* terdapat derau pada data. Pada grafik hasil orientasi HP dimana data yang ditampilkan merupakan nilai perekaman secara *real time* sehingga data yang disajikan belum dilakukan *filtering*.



Gambar 12. Sudut *attitude* sebelum diterapkan algoritma ZAC

Pada Gambar 12 merupakan data hasil perhitungan sudut *attitude* sebelum menerapkan algoritma ZAC. Hasil perhitungan orientasi sebelum menerapkan algoritma untuk mendapatkan sudut *attitude* sudah dapat dilihat dengan jelas pada grafik yang ditampilkan. Pada hasil sebelum menerapkan algoritma dilakukan proses *filtering* dengan *Kalman filter* tetapi hasil sudut *attitude* terdapat derau yang tidak teratasi dengan baik.





Gambar 13. Sudut *attitude* setelah diterapkan algoritma ZAC

Pada Gambar 13 merupakan data hasil perhitungan sudut *attitude* setelah menerapkan algoritma ZAC. Pada hasil orientasi sikap *roll* dan sikap *pitch* setelah menerapkan algoritma ZAC dapat mengatasi derau dengan baik. Pengaruh algoritma ZAC dapat mengkompensasi percepatan nol pada kondisi tertentu sehingga data sudut *attitude* yang dihasilkan dapat diperjelas. Hasil sudut *attitude* dengan algoritma ZAC dapat menurunkan galat, derau yang dihasilkan dapat diperkecil dan sudut *attitude* yang didapat lebih akurat.

#### 4. Kesimpulan

*Kalman filter* dapat digunakan untuk mengestimasi nilai orientasi UAV dengan melakukan penggabungan data dari sensor *accelerometer* dan *gyroscope*. Penentuan kondisi statis dan dinamis perlu dilakukan karena akan mempengaruhi keakuratan sudut orientasi. Dilakukan perancangan algoritma untuk mengatasi hal tersebut. Sebelum diterapkan algoritma ZAC, galat untuk sikap *roll* sebesar 3,34% dan untuk sikap *pitch* dengan galat sebesar 1,38%. Setelah diterapkannya algoritma ZAC, galat yang dihasilkan mengalami penurunan untuk sikap *roll* menjadi sebesar 3.16% dan untuk sikap *pitch* menjadi sebesar 1.09%. Akurasi pada sudut *attitude* sebelum menerapkan algoritma ZAC pada sikap *roll* sebesar 96,66% dan untuk sikap *pitch* sebesar 98,62%. Setelah diterapkan algoritma ZAC, dihasilkan akurasi yang lebih baik pada sikap *roll* menjadi sebesar 96,84% dan untuk sikap *pitch* menjadi sebesar 98,91%.

#### Daftar Pustaka

- [1] Sandi, B. Y., Kurniawan, F., & Lasmadi, L. (2020, December). Estimasi Sudut Orientasi Rigid Body dengan Menggunakan Sensor IMU (Inertial Measurement Unit) dan Magnetometer. In *Conference SENATIK STT Adisutjipto Yogyakarta* (Vol. 6, pp. 283-294).
- [2] Ardiantara, P. S., Sumiharto, R., & Wibowo, S. B. (2014). Purwarupa Kontrol Kestabilan Posisi dan Sikap pada Pesawat Tanpa Awak Menggunakan IMU dan Algoritma Fusion Sensor Kalman Filter. *IJEIS*, 4(1), 25-34.
- [3] Kurniawan, F., Nasution, M. R. E., Dinaryanto, O., & Lasmadi, L. (2021). Penentuan Orientasi dan Translasi Gerakan UAV Menggunakan Data Fusion Berbasis Kalman Filter. *AVITEC*, 3(2): 99 – 115.
- [4] Rhudy, M. B., Salguero, R. A., & Holappa, K. (2017). A Kalman filtering tutorial for undergraduate students. *International Journal of Computer Science & Engineering Survey*, 8(1), 1-9.
- [5] Lasmadi, Cahyadi, A., & Hidayat, R. (2016). Implementasi Kalman Filter untuk Navigasi Quadrotor Berbasis Sensor Accelerometer. *Prosiding SENIATI*, 242-B.

- [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] Alma'i, V. R., Wahyudi, W., & Setiawan, I. (2011). *Aplikasi Sensor Accelerometer Pada Deteksi Posisi* (Doctoral dissertation, Jurusan Teknik Elektro Fakultas Teknik).
- [8] Kimberly Tuck (2007). *Tilt Sensing Using Linear Accelerometers*, Tilt Sensing Using Linear Accelerometers, Application Note, AN3461, Rev 2, 06/2007.
- [9] Suryanti, D. I. (2017). Inertial Measurement Unit (IMU) Pada Sistem Pengendali Satelit. *Media Dirgantara*, 12(2): 7 – 9.
- [10] Ojeda, L. V., Zaferiou, A. M., Cain, S. M., Vitali, R. V., Davidson, S. P., Stirling, L. A., & Perkins, N. C. (2017). *Estimating Stair Running Performance Using Inertial Sensors*. *Sensors*, 17(11), 2647.
- [11] Wicaksono, M. A. R., Kurniawan, F., & Lasmadi, L. (2020). Kalman Filter untuk Mengurangi Derau Sensor Accelerometer pada IMU Guna Estimasi Jarak. *AVITEC*, 2(2), 145 – 160.
- [12] Ermawati, E., Rahayu, P., & Zuhairoh, F. (2017). Perbandingan Solusi Numerik Integral Lipat Dua pada Fungsi Aljabar dengan Metode Romberg dan Simulasi Monte Carlo. *Jurnal MSA*, 5(1) : 46 – 57.