

Sistem Navigasi Otonom Robot Pembersih Kaca untuk Gedung Bertingkat

Mohammad Wafi Nabil Karim

Fakultas Teknik Elektro

Telkom University

Bandung, Indonesia

wafinabil@student.telkomuniversity.ac.id

Sony Sumaryo

Fakultas Teknik Elektro

Telkom University

Bandung, Indonesia

sonysumaryo@telkomuniversity.ac.id

Basuki Rahmat

Fakultas Teknik Elektro

Telkom University

Bandung, Indonesia

basukir@telkomuniversity.ac.id

Abstrak — Pembersihan jendela pada gedung bertingkat merupakan pekerjaan berisiko tinggi yang menuntut efisiensi dan keamanan. Untuk mengatasi tantangan tersebut, penelitian ini bertujuan mengembangkan sistem navigasi otonom yang andal untuk robot pembersih jendela. Fokus utama penelitian adalah merancang sistem yang mampu bergerak secara mandiri di permukaan kaca vertikal yang datar, dengan batasan operasional pada kondisi cuaca yang mendukung untuk menjamin kinerja optimal. Metodologi penelitian ini diterapkan secara sistematis, diawali dengan studi literatur mendalam untuk mengkaji teknologi navigasi dan fusi sensor yang relevan. Tahap selanjutnya adalah perancangan arsitektur sistem, yang mencakup pengembangan mekanisme navigasi presisi serta sistem kontrol gerak. Rancangan tersebut kemudian divalidasi melalui simulasi sebelum diimplementasikan pada perangkat keras. Kunci dari implementasi ini adalah integrasi strategis antara Complementary Filter dan Kalman Filter. Kombinasi kedua filter ini sangat krusial untuk mengolah data sensor secara akurat, memastikan robot dapat menjaga stabilitas orientasi dan keakuratan posisi selama menjalankan tugas pembersihan. Tahap akhir penelitian meliputi analisis hasil untuk mengevaluasi performa dan keandalan sistem navigasi yang telah dikembangkan.

Kata kunci— Robot Pembersih Jendela, Navigasi Otonom, IMU, Ultrasonik, Kontrol Filter.

I. PENDAHULUAN

Di masa lalu, pembersihan jendela gedung tinggi menimbulkan tantangan besar karena melibatkan tenaga manusia yang menghadapi risiko tinggi dan ketidaknyamanan. Untuk mengatasi tantangan ini, berbagai alat mekanis seperti tangga dan alat pengangkat telah dikembangkan. Namun, meski alat ini memberikan kemudahan, penggunaannya tetap membutuhkan keterlibatan langsung manusia dan tidak sepenuhnya menghilangkan risiko yang ada.

Dalam dekade terakhir, berbagai teknologi telah dikembangkan dan diuji untuk mencapai tujuan ini, termasuk robot yang dirancang khusus untuk merangkak pada permukaan vertikal. Robot-robot ini telah digunakan untuk berbagai aplikasi, termasuk pembersihan jendela gedung, pemeliharaan jembatan, pemeliharaan pipa, dan pemeliharaan jaringan listrik. Pembersihan dan pemeliharaan bangunan modern, terutama pada bagian jendela, telah menjadi tantangan signifikan dalam dunia konstruksi. Pekerjaan di ketinggian yang melekat pada tugas ini membawa risiko tinggi bagi pekerja manusia. Untuk

meningkatkan kemampuan robot dalam merangkak di permukaan vertikal, teknologi penghisap motor diterapkan sebagai metode utama untuk menjaga robot tetap melekat pada permukaan jendela. Mekanisme ini menggunakan kipas berkecepatan tinggi untuk menciptakan area bertekanan rendah di dalam ruang suction.

II. KAJIAN TEORI

A. KONTROL PID

Kontrol PID merupakan salah satu metode pengendalian sistem yang paling umum digunakan karena kesederhanaan dan efektivitasnya dalam berbagai aplikasi teknik. Pengontrol PID terdiri dari tiga komponen utama, yaitu:

- Proporsional (P): Memberikan sinyal kontrol yang sebanding dengan kesalahan saat ini. Komponen ini merespons secara langsung terhadap besarnya error, membantu mempercepat pencapaian nilai referensi.
- Integral (I): Mengakumulasi kesalahan dari waktu ke waktu, berfungsi menghilangkan kesalahan steady-state yang tidak dapat diatasi oleh kontrol proporsional saja.
- Derivatif (D): Memberikan respons terhadap perubahan laju kesalahan, membantu mengurangi osilasi sistem dan mempercepat kestabilan respons.

B. COMPLEMENTARY FILTER

Filter complementary adalah metode estimasi orientasi yang menggabungkan keunggulan akselerometer (stabilitas jangka panjang) dan giroskop (respons jangka pendek). Dengan menggunakan filter low-pass pada akselerometer dan high-pass pada giroskop, diperoleh estimasi orientasi yang halus dan akurat. Filter ini cocok untuk sistem robotik yang bekerja di lingkungan dinamis, seperti permukaan vertikal kaca. Prinsip kerja CF dapat dinyatakan dalam persamaan matematis:

$$\theta[k] = \alpha(\theta_{gyro}[k-1] + \omega_z[k]\Delta t) + (1-\alpha)\theta_{acc}[k] \quad (1)$$

Keterangan:

- $\theta[k]$: estimasi sudut (yaw) pada sampel ke-k (derajat).
- $\theta_{gyro}[k-1]$: sudut hasil integrasi giroskop hingga sampel sebelumnya (k-1) (derajat).
- $\omega_z[k]$: laju putar sumbu-z dari giroskop pada sampel k (derajat/detik).

- Δt : periode sampel (detik).
- $\theta_{acc}[k]$: sudut referensi dari accelerometer pada sampel k (derajat).
- $\alpha \in (0,1)$: koefisien filter; makin besar $\alpha \Rightarrow$ kontribusi gyro makin dominan, makin kecil $\alpha \Rightarrow$ referensi accelerometer makin kuat.

Koefisien α menentukan keseimbangan antara respons cepat (gyro) dan kestabilan jangka panjang (accel). Nilai α yang besar membuat sistem lebih responsif namun lebih rawan drift, sedangkan nilai kecil membuat estimasi lebih stabil namun lebih lambat merespons perubahan. Dengan menggunakan complementary filter, sistem dapat mengurangi efek drift dari giroskop tanpa mengorbankan kecepatan respon.

C. Persamaan Hypsometric

Persamaan *hypsometric* merupakan hasil penggabungan keseimbangan hidrostatik dengan hukum gas ideal untuk memetakan perubahan tekanan menjadi perubahan ketinggian. Dalam praktik yang digunakan pada program, persamaan ini dipakai untuk menghitung ketinggian relatif terhadap suatu kondisi referensi, dengan mengasumsikan percepatan gravitasi konstan dan memakai suhu udara sebagai pendekatan suhu virtual. Intuisi fisiknya sederhana: semakin tinggi suatu titik, tekanan udaranya semakin rendah; karena itu ketinggian relatif dapat diperoleh dari logaritma rasio tekanan referensi terhadap tekanan sekarang, diskalakan oleh konstanta gas dan suhu, serta dibagi percepatan gravitasi. Implementasi yang dipakai menegaskan bentuk logaritmik ini dan memastikan konsistensi satuan sehingga hasil akhirnya berada dalam meter.

$$z = \left(\frac{R_d T}{g_0} \right) \cdot \ln \left(\frac{p_0}{p} \right) \quad (2)$$

Keterangan:

- z : Ketinggian relatif terhadap kondisi referensi (m)
- p_0 : Tekanan referensi (Pa)
- p : Tekanan saat ini (Pa)
- T : Suhu (K) yang dipakai sebagai aproksimasi suhu virtual
- R_d : Konstanta gas udara kering = $287.05 \text{ J (kg}^{-1} \text{ K}^{-1})$
- g_0 : Percepatan gravitasi = 9.80665 ms^{-2}

Secara operasional, besaran pada persamaan di atas dipersiapkan agar sesuai dengan kebutuhan komputasi. Tekanan dari sensor yang umumnya dinyatakan dalam hPa dikonversi terlebih dahulu ke Pa agar konsisten dengan konstanta fisik. Tekanan referensi diperoleh dari pengambilan beberapa sampel awal kemudian dirata-ratakan, sehingga ($z = 0$) pada lokasi kalibrasi. Suhu yang digunakan adalah suhu dalam Kelvin yang di-*low-pass* untuk meredam fluktuasi cepat, sehingga faktor penskalaan $\left(\frac{R_d T}{g_0} \right)$ lebih stabil dari waktu ke waktu. Karena bentuknya bersifat relatif

terhadap p_0 , nilai z yang dihasilkan merepresentasikan ketinggian relatif, bukan ketinggian absolut terhadap permukaan laut.

D. KALMAN FILTER

Kalman Filter merupakan algoritma optimasi statistik yang digunakan untuk memperkirakan keadaan sistem berdasarkan model dan pengukuran sensor. Kalman Filter terdiri dari dua proses utama: prediksi dan pembaruan. Dua pendekatan utama digunakan. Pada tahap prediksi, estimasi posisi dihitung berdasarkan data sebelumnya. Di tahap pembaruan, estimasi ini dikoreksi berdasarkan data sensor terbaru. Koreksi ini dilakukan dengan menghitung Kalman Gain (K), yaitu bobot yang menentukan seberapa besar pengaruh pengukuran baru terhadap hasil estimasi akhir. Jika pengukuran dianggap sangat akurat dalam artian dapat noise kecil, maka nilai Kalman Gain akan besar dan estimasi akan lebih dipengaruhi oleh pengukuran. Sebaliknya, jika pengukuran penuh noise, maka prediksi model akan lebih dominan.

Model ruang-keadaan (state-space)

$$x_k = F_k x_{k-1} + B_k u_{k-1} + w_{k-1} \quad (3)$$

$$z_k = H_k x_k + v_k \quad (4)$$

Keterangan :

- k : indeks waktu diskret (langkah ke- k).
- x_k : vektor keadaan/state pada waktu k (mis. komponen posisi, kecepatan, bias, dsb.).
- F_k : matriks transisi yang memetakan state dari waktu $k-1$ ke k sesuai model dinamika.
- u_{k-1} : vektor masukan/kontrol yang diberikan pada interval ($k-1 \rightarrow k$)
- B_k : matriks kontrol yang memetakan pengaruh u_{k-1} terhadap perubahan state.
- w_{k-1} : noise proses (gangguan/ketidakpastian model) yang umumnya dimodelkan Gaussian nol-mean, dengan varians Q .
- z_k : vektor pengukuran pada waktu k (output sensor).
- H_k : matriks observasi yang memetakan state ke ruang pengukuran.
- v_k : noise pengukuran (kebisingan sensor) yang umumnya dimodelkan Gaussian nol-mean, dengan varians R .

Dalam kerangka *Kalman Filter*, sebuah sistem dinamis diskret biasanya dimodelkan dengan dua persamaan inti: persamaan keadaan (state/transition) dan persamaan pengukuran (measurement/observation). Dua persamaan di atas menyusun kerangka matematis standar untuk pemodelan sistem dinamis linier dalam *Kalman Filter*. Persamaan keadaan menyatakan evolusi vektor keadaan dari langkah $k-1$ menuju k melalui pemetaan linier oleh matriks transisi

F_k , dengan kemungkinan adanya pengaruh masukan eksternal yang dimodelkan oleh $B_k u_{k-1}$. Ketidaktepatan model, gangguan luar, serta dinamika yang tidak seluruhnya terwakili ditangkap oleh noise proses w_{k-1} yang dicirikan oleh kovarians Q_k ; besaran ini mengatur tingkat kepercayaan terhadap model dinamika. Persamaan pengukuran menjelaskan bagaimana sensor menghasilkan observasi z_k sebagai proyeksi linier dari keadaan melalui H_k , yang kemudian terkontaminasi noise pengukuran v_k dengan kovarians R_k ; besaran ini mengatur tingkat kepercayaan terhadap data sensor. Dengan asumsi linieritas dan noise Gaussian nol-mean yang saling tidak berkorelasi, pasangan persamaan tersebut memungkinkan perancangan estimator rekursif yang optimal dalam arti galat kuadrat rata-rata: tahap prediksi menggunakan persamaan keadaan untuk menghasilkan taksiran *a priori* dan kovariansnya, sedangkan tahap pembaruan menggunakan persamaan pengukuran untuk mengoreksi taksiran tersebut sesuai kualitas relatif model (melalui Q_k) dan sensor (melalui R_k). Formulasi ini menjadi landasan bagi pengembangan semua rumus operasional *Kalman Filter* (inovasi, kovarians inovasi, Kalman gain, serta pembaruan keadaan dan kovarians).

Tahap prediksi.

$$\hat{x}_{k|k-1} = F_k \hat{x}_{k-1|k-1} + B_k u_{k-1} \quad (5)$$

$$P_{k|k-1} = F_k P_{k-1|k-1} F_k^T + Q_k \quad (6)$$

Keterangan :

- $\hat{x}_{k|k-1}$: prediksi vektor keadaan pada waktu k sebelum melihat pengukuran (*a priori*).
- $\hat{x}_{k-1|k-1}$: estimasi keadaan pada waktu $k-1$ sesudah pembaruan (*a posteriori*).
- F_k : matriks transisi keadaan (diskret, bergantung Δt) yang mewujudkan kinematika dengan bias konstan:

$$F_k = \begin{bmatrix} 1 & \Delta t & -\frac{1}{2}\Delta t^2 \\ 0 & 1 & -\Delta t \\ 0 & 0 & 1 \end{bmatrix} \quad (7)$$

- u_{k-1} : masukan kontrol pada interval $(k-1 \rightarrow k)$
- B_k : matriks kontrol yang memetakan a ke perubahan state,

$$B_k = \begin{bmatrix} \frac{1}{2}\Delta t^2 \\ \Delta t \\ 0 \end{bmatrix} \quad (8)$$

- $P_{k|k-1}$: prediksi kovarians kesalahan estimasi pada waktu k (*a priori*).
- $P_{k-1|k-1}$: kovarians kesalahan estimasi pada waktu $k-1$ (*a posteriori*).
- Q_k : kovarians noise proses yang merepresentasikan ketidakpastian dinamika.

$$Q_k = \begin{bmatrix} \frac{\sigma_a^2 \Delta t^4}{4} & \frac{\sigma_a^2 \Delta t^3}{2} & 0 \\ \frac{\sigma_a^2 \Delta t^3}{2} & \sigma_a^2 \Delta t^2 & 0 \\ 0 & 0 & \sigma_b^2 \Delta t \end{bmatrix} \quad (9)$$

Kedua persamaan tersebut membentuk tahap prediksi dalam *Kalman Filter*. Persamaan pertama memproyeksikan taksiran keadaan dari langkah sebelumnya ke langkah saat ini melalui pemetaan linier oleh matriks transisi F_k dan, bila tersedia, kontribusi masukan yang dipetakan oleh B_k . Hasilnya adalah taksiran (*a priori*) $\hat{x}_{k|k-1}$, yakni perkiraan keadaan berdasarkan model semata sebelum bukti pengukuran pada waktu k digunakan. Persamaan kedua mempropagasikan ketidakpastian taksiran: kovarians kesalahan dari langkah sebelumnya ditransformasikan oleh dinamika $F_k P_{k-1|k-1} F_k^T$, kemudian ditambah Q_k untuk memasukkan pengaruh noise proses yang meliputi dinamika tak termodelkan, variasi parameter, atau gangguan luar. Semakin besar Q_k , semakin besar pula $P_{k|k-1}$, menandakan bahwa prediksi diakui lebih tidak pasti dan nantinya cenderung memberi bobot lebih besar pada informasi pengukuran saat tahap pembaruan. Dengan demikian, tahap prediksi menyediakan pasangan $(\hat{x}_{k|k-1}, P_{k|k-1})$ sebagai basis awal yang kemudian diseimbangkan oleh data sensor pada tahap pembaruan.

Tahap pembaruan.

$$y_k = z_k - H_k \hat{x}_{k|k-1} \quad (10)$$

$$S_k = H_k P_{k|k-1} H_k^T + R_k \quad (11)$$

$$K_k = P_{k|k-1}^{-1} H_k^T S_k^{-1} \quad (12)$$

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k y_k \quad (13)$$

$$P_{k|k} = (I - K_k H_k) P_{k|k-1} \quad (14)$$

Keterangan :

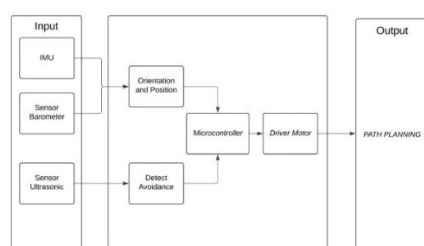
- z_k : vektor pengukuran pada waktu k .
- H_k : matriks observasi yang memetakan state ke ruang pengukuran.
- $\hat{x}_{k|k-1}$: taksiran *a priori* state pada waktu k (sebelum pembaruan).
- R_k : kovarians noise pengukuran.
- $P_{k|k-1}$: kovarians kesalahan taksiran *a priori*.
- y_k : inovasi/residu (selisih antara pengukuran dan prediksi pengukuran).
- S_k : kovarians inovasi.
- K_k : Kalman gain (bobot koreksi).
- $\hat{x}_{k|k}$: taksiran *a posteriori* state (sesudah pembaruan).
- $P_{k|k}$: kovarians kesalahan taksiran *a posteriori*.
- I : matriks identitas.

Tahap pembaruan menggabungkan informasi pengukuran untuk mengoreksi prediksi. Besaran y_k disebut inovasi atau residu dan menyatakan seberapa jauh pengukuran aktual z_k dari prediksi pengukuran $H_k\hat{x}_k|_{k-1}$ nilai ini menjadi sinyal koreksi. Kovariansnya, S_k menjumlahkan ketidakpastian prediksi yang diproyeksikan ke ruang pengukuran $H_kP_k|_{k-1}H_k^T$ dengan ketidakpastian sensor R_k sehingga S_k merepresentasikan seberapa “terpercaya” inovasi tersebut. Kalman gain K_k dipilih agar meminimalkan kovarians kesalahan *a posteriori* (optimal dalam arti kuadrat-rata untuk model linier-Gaussian) dan memberikan bobot yang menyeimbangkan kepercayaan antara model dan data jika R_k kecil (sensor akurat), maka S_k mengecil dan K_k membesar sehingga koreksi lebih dipengaruhi pengukuran; jika $P_k|_{k-1}$ besar (prediksi tidak pasti), K_k juga cenderung membesar. State kemudian diperbarui dengan menambahkan koreksi $K_k y_k$, menghasilkan taksiran *a posteriori* $\hat{x}_k|_k$. Kovarians kesalahan diperbarui melalui $(I - K_k H_k)P_k|_{k-1}$, yang secara ideal menurunkan ketidakpastian karena informasi baru telah diintegrasikan.

III. METODE

Navigasi robot pembersih jendela ini berbasis sistem kontrol otonom yang menggabungkan berbagai sensor dan aktuator. Robot ini menggunakan barometer untuk melacak posisi dan jarak tempuh, sensor ultrasonik untuk mendeteksi batasan, dan IMU untuk menjaga orientasi selama pergerakan. Komponen-komponen ini dikendalikan oleh mikrokontroler yang memproses data sensor secara real-time dan menjalankan algoritma kontrol C++/C yang terprogram untuk memastikan robot dapat bergerak pada lintasan yang diinginkan.

Diagram Blok



GAMBAR 1

(Blok Diagram Umum Sistem Navigasi.)

Diagram blok sistem pada robot pembersih jendela memberikan gambaran umum tentang bagaimana setiap komponen saling terhubung dan berfungsi dalam proses navigasi dan pembersihan. Sistem ini dibagi menjadi beberapa bagian utama, yaitu:

1. Sensor dan Input:

- Sensor Ultrasonik: Mendeteksi jarak ke rintangan dan batas kaca.
- IMU: Mengukur sudut dan percepatan untuk menjaga orientasi robot.

- Barometer: Memantau tekanan udara untuk mengukur ketinggian.

2. Unit Pemrosesan:

- Mikrokontroler ESP32: Mengolah data dari semua sensor dan menjalankan algoritma kontrol navigasi.

3. Aktuator:

- Driver Motor DC: Menggerakkan roda untuk pergerakan vertikal.

Fungsi dan Fitur

Robot pembersih jendela ini dirancang dengan fitur dan fungsi yang mendukung navigasi otonom. Fitur utama meliputi:

1. Navigasi Otonom Berbasis Sensor

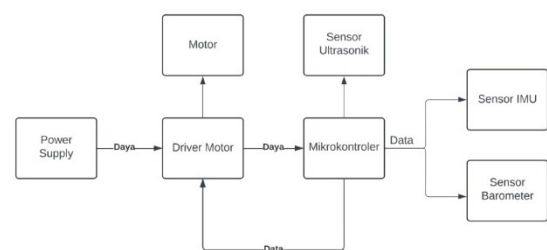
Robot menggunakan data dari sensor Ultrasonik, dan IMU untuk menentukan posisi dan lintasan gerak. Algoritma kontrol PID (Proportional-Integral-Derivative) digunakan untuk menjaga kestabilan dan mengoreksi penyimpangan posisi selama operasi.

2. Deteksi dan Penghindaran Rintangan

Sensor Ultrasonik mendeteksi jarak ke rintangan di sekitar robot. Jika terdeteksi hambatan, robot secara otomatis mengoreksi jalurnya untuk menghindari tabrakan.

Desain Perangkat Keras

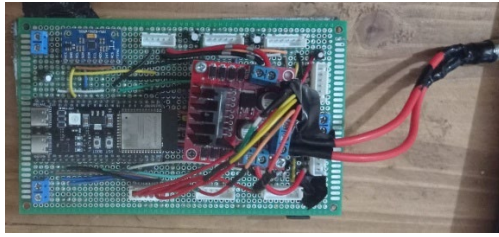
Desain perangkat keras mencakup integrasi komponen elektronik dan mekanik untuk mendukung sistem navigasi dan kontrol.



GAMBAR 2
(Skema Hardware)

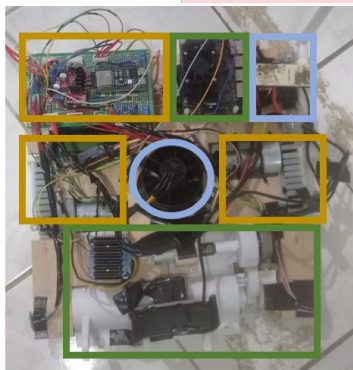
Diagram tersebut menunjukkan alur kerja sistem yang terdiri dari beberapa komponen utama yaitu power supply, driver motor, mikrokontroler, motor, serta sensor-sensor (ultrasonik, IMU, dan barometer). Power supply berfungsi sebagai sumber energi listrik yang mengalirkan daya ke driver motor dan mikrokontroler. Driver motor kemudian mengatur distribusi daya untuk menggerakkan motor berdasarkan sinyal data yang diterimanya dari mikrokontroler. Mikrokontroler menjadi pusat kendali yang memproses data dari berbagai sensor. Sensor ultrasonik digunakan untuk mendeteksi jarak atau hambatan di sekitar sistem, sensor IMU (Inertial Measurement Unit) berfungsi untuk mengukur orientasi dan percepatan, sedangkan sensor barometer digunakan untuk mengetahui tekanan udara yang dapat dikaitkan dengan ketinggian. Data dari sensor-sensor

ini diolah oleh mikrokontroler untuk mengendalikan pergerakan motor secara tepat melalui driver motor.



GAMBAR 3
(Board Sistem)

Board sistem pada Gambar 3.3 menampung beberapa komponen diantaranya adalah ESP32S3 (Mikrokontroler), MPU6500 (IMU), MS5611 (Barometer), L298n (Motor Driver) dan socket untuk HCSR04 (Ultrasonik) juga Motor DC.



GAMBAR 4
(Robot Pembersih Jendela)

Gambar (4) menunjukkan rancangan robot yang telah dirakit menggunakan platform yang telah di *laser cutting* dan memiliki beberapa komponen diantaranya untuk penghisap, navigasi, dan pembersih. Bagian penghisap (biru) memakai komponen Motor EDF sebagai aktuator utama, lalu untuk sensor digunakan sensor arus PZEM-016 dan dikontrol lewat mikrokontroler ESP32 sedangkan Bagian pembersih (hijau) menggunakan Motor DC untuk memutar *brush* pembersih dan kamera sebagai pendeteksi kotoran pada jendela dengan kontrol dari mikrokomputer Jetson. Bagian navigasi (oranye) seperti pada Gambar 3.4 melingkupi dari Motor DC, Motor Driver, mikrokontroler ESP32S3, Barometer, Ultrasonik, dan IMU.

IV. HASIL DAN PEMBAHASAN

E. KALMAN FILTER UNTUK ESTIMASI POSISI ROBOT

Dalam sistem robot pembersih jendela ini, estimasi posisi yang akurat sangat penting untuk memastikan navigasi berjalan dengan tepat. Oleh karena itu, digunakan metode Kalman Filter untuk menggabungkan data dari encoder dan sensor IMU agar menghasilkan estimasi posisi yang lebih halus dan bebas noise. Masing-masing sensor memiliki kelemahan inheren, data barometer meskipun relatif stabil terhadap drift jangka panjang, rentan terhadap noise berfrekuensi tinggi yang berasal dari turbulensi udara atau

getaran mekanik, sedangkan data IMU bersifat responsif terhadap perubahan cepat namun mengalami akumulasi kesalahan (drift) seiring integrasi waktu. *Kalman Filter* memanfaatkan sifat saling melengkapi ini untuk menghasilkan estimasi ketinggian yang lebih akurat dan stabil. Secara matematis, model state yang digunakan terdiri atas tiga variabel: ketinggian z (cm), kecepatan vertikal v (cm/s), dan bias percepatan b (cm/s²).

Vektor keadaan (*state*) dan masukan.

$$X_k = \begin{bmatrix} z_k \\ v_k \\ b_k \end{bmatrix}, \quad u_{k-1} = a_{k-1} \quad (15)$$

dengan z = posisi/tinggi (m), v = kecepatan vertikal (m/s), b = bias percepatan (m/s²), dan a = percepatan vertikal (m/s²).

Pada langkah prediksi, keadaan sistem diproyeksikan ke waktu berikutnya. Seiring dengan proyeksi keadaan (Persamaan 5), ketidakpastian atau kovarians kesalahan dari estimasi juga diperbarui menggunakan persamaan propagasi kovarians kesalahan (6). Persamaan ini pada dasarnya menyatakan bahwa kovarians kesalahan yang baru (disebut *a priori*) adalah hasil dari kovarians kesalahan sebelumnya yang ditransformasikan oleh dinamika sistem, ditambah dengan ketidakpastian dari model proses itu sendiri. Ketidakpastian proses ini didefinisikan oleh matriks kovarians proses Q_k (9), yang dalam kasus ini memodelkan akselerasi sebagai *white-noise* dan komponen lain sebagai *random-walk* untuk merepresentasikan gangguan acak pada sistem.

Selanjutnya, pada langkah pembaruan, kita menggunakan pengukuran baru untuk mengoreksi prediksi. Persamaan pengukuran (15) mendefinisikan bagaimana pengukuran (z_k) yang kita peroleh berhubungan dengan keadaan sistem (x_k), dengan mempertimbangkan adanya *noise* pengukuran (v_k) yang memiliki varians R .

Pengukuran tinggi

Pengukuran tinggi relatif diperoleh dari persamaan hypsometric dipakai untuk membentuk $z_k^{(meas)}$:

$$z_k^{(meas)} = \left(\frac{R_d T}{g_0} \right) \cdot \ln \left(\frac{p_0}{p} \right) \quad (16)$$

Keterangan:

- z : Ketinggian relatif terhadap kondisi referensi (m)
- p_0 : Tekanan referensi (Pa)
- p : Tekanan saat ini (Pa)
- T : Suhu (K) yang dipakai sebagai aproksimasi suhu virtual
- R_d : Konstanta gas udara kering = 287.05 J (kg⁻¹ K⁻¹)
- g_0 : Percepatan gravitasi = 9.80665 ms⁻²

Hasil $z_k^{(meas)}$ akan digunakan sebagai pada (2.11).

Perhitungan Data dan Parameter yang Digunakan

- Satu pengukuran tinggi relative ($z_k^{(meas)}$) : 0.61 m.
- Interval sampel : $\Delta t = 0.02$ s (typical 50 Hz).

- Parameter penalaan : $\sigma_z = 0.8 \text{ m} \rightarrow R_k = \sigma_z^2 = 0.64 \text{ m}^2$; $\sigma_a = 0.25 \frac{\text{m}}{\text{s}^2}$, $\sigma_b = 0.02 \frac{\text{m}}{\text{s}^2} / \sqrt{s}$
- Keadaan awal (*a posteriori* langkah sebelumnya): $\hat{x}_{k-1|k-1} = [0 \ 0 \ 0]^T$
- Masukan percepatan pada interval ini tidak tersedia dari dataset; untuk ilustrasi numerik diambil $a_{k-1} = 0$.

Prediksi keadaan

Matriks

$$F_k = \begin{bmatrix} 1 & 0.02 & -0.0002 \\ 0 & 1 & -0.02 \\ 0 & 0 & 1 \end{bmatrix}$$

$$B_k = \begin{bmatrix} 0.0002 \\ 0.02 \\ 0 \end{bmatrix}$$

Dengan $a_{k-1} = 0$:

$$\hat{x}_{k|k-1} = F_k \hat{x}_{k-1|k-1} + B_k a = [0, 0, 0]^T \text{ (karena } a = 0 \text{ dan state awal nol)}$$

Prediksi kovarians

Hitung Q_k :

$$\Delta t^2 = 0.0004, \quad \Delta t^3 = 0.000008, \quad \Delta t^4 = 0.00000016, \quad \sigma_a^2 = 0.0625, \quad \sigma_b^2 = 0.0004$$

$$Q_{00} = \sigma_a^2 \cdot \Delta t^4 / 4 = 0.0625 \cdot (1.6 \times 10^{-7}) / 4 = 2.5 \times 10^{-9}$$

$$Q_{01} = \sigma_a^2 \cdot \Delta t^3 / 2 = 0.0625 \cdot (8 \times 10^{-6}) / 2 = 2.5 \times 10^{-7}$$

$$Q_{11} = \sigma_a^2 \cdot \Delta t^2 = 0.0625 \cdot 4 \times 10^{-4} = 2.5 \times 10^{-5}$$

$$Q_{22} = \sigma_b^2 \cdot \Delta t = 4 \times 10^{-4} \cdot 2 \times 10^{-2} = 8 \times 10^{-6}$$

$$Q_k = \begin{bmatrix} 2.5 \times 10^{-9} & 2.5 \times 10^{-7} & 0 \\ 2.5 \times 10^{-7} & 2.5 \times 10^{-5} & 0 \\ 0 & 0 & 8 \times 10^{-6} \end{bmatrix}$$

Lalu

$$\begin{aligned} P_{00} &= 1^2 \cdot 1 + (0.02)^2 \cdot 1 + (-0.0002)^2 \cdot 100 + Q_{00} \\ &= 1 + 0.0004 + 0.000004 + 0.0000000025 \\ &= 1.0004040025 \end{aligned}$$

$$\begin{aligned} P_{01} &= (1 \cdot 0) \cdot 1 + (0.02 \cdot 1) \cdot 1 + (-0.0002 \cdot -0.02) \cdot 100 + Q_{01} \\ &= 0 + 0.02 + 0.0004 + 0.000000025 \\ &= 0.02040025 \end{aligned}$$

$$\begin{aligned} P_{02} &= (1 \cdot 0) \cdot 1 + (0.02 \cdot 0) \cdot 1 + (-0.0002 \cdot 1) \cdot 100 \\ &\quad + 0 \\ &= -0.02 \end{aligned}$$

$$\begin{aligned} P_{11} &= 0^2 \cdot 1 + 1^2 \cdot 1 + (-0.02)^2 \cdot 100 + Q_{11} \\ &= 0 + 1 + 0.04 + 0.000025 \\ &= 1.040025 \end{aligned}$$

$$\begin{aligned} P_{12} &= (0 \cdot 0) \cdot 1 + (1 \cdot 0) \cdot 1 + (-0.02 \cdot 1) \cdot 100 + 0 \\ &= -2.0 \end{aligned}$$

$$\begin{aligned} P_{22} &= 0^2 \cdot 1 + 0^2 \cdot 1 + 1^2 \cdot 100 + Q_{22} \\ &= 100 + 0.000008 \\ &= 100.000008 \end{aligned}$$

$$P_{k|k-1} = \begin{bmatrix} 1.0004040025 & 0.02040025 & -0.02 \\ 0.02040025 & 1.040025 & -2.0 \\ -0.02 & -2.0 & 100.000008 \end{bmatrix}$$

Inovasi dan kovarians Inovasi

$$y_k = z_k(\text{meas}) - H \hat{x}_{k|k-1} = 0.61 - 0 = 0.61 \text{ m}$$

$$\begin{aligned} S_k &= H P_{k|k-1} H^T + R = P_{00} + R \\ &= 1.0004040025 + 0.64 \\ &= 1.6404040025 \text{ m}^2 \end{aligned}$$

Kalman gain

$$\text{Karena } H_k = [1 \ 0 \ 0],$$

$$\begin{aligned} K_k &= (1/S_k) \cdot [P_{00} \ P_{10} \ P_{20}]^T \\ &= (1/1.6404040025) \\ &\quad \cdot [1.0004040025 \ 0.02040025 \\ &\quad -0.02]^T \end{aligned}$$

$$K_z = 1.0004040025 / 1.6404040025 \approx 0.60985221$$

$$K_v = 0.02040025 / 1.6404040025 \approx 0.01243611$$

$$K_b = (-0.02) / 1.6404040025 \approx -0.01219212$$

$$K_k = \begin{bmatrix} 0.60985221 \\ 0.01243611 \\ -0.01219212 \end{bmatrix}$$

Pembaruan keadaan

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k y_k$$

$$\begin{aligned}
&= [0, 0, 0]^T \\
&+ [0.60985221, 0.01243611, -0.01219212]^T \cdot 0.61 \\
&= \begin{bmatrix} 0.60985221 \cdot 0.61 \\ 0.01243611 \cdot 0.61 \\ -0.01219212 \cdot 0.61 \end{bmatrix} = \begin{bmatrix} 0.37200985 \\ 0.00758603 \\ -0.00743719 \end{bmatrix}
\end{aligned}$$

Artinya, setelah mengamati ($z_k^{(meas)}$) = 0.61 m, estimasi ($a_{posteriori}$) menjadi $z \approx 0.3720$ m, $v \approx 7.59 \times 10^{-3}$ m/s dan bias percepatan $\approx -7.44 \times 10^{-3}$ m/s²

Pembaruan kovarians

Dengan bentuk $(1-KH)P$ untuk $H_k = [1 \ 0 \ 0]$, komponen-komponen utama:

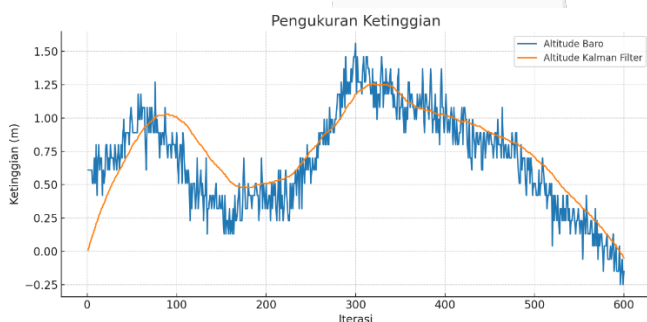
$$\begin{aligned}
P_{00|k} &= (1 - K_z) P_{00} \\
&= (1 - 0.60985221) \cdot 1.0004040025 \\
&= 0.390305413 \dots
\end{aligned}$$

$$\begin{aligned}
P_{10|k} &= P_{10} - K_v P_{00} \\
&= 0.02040025 - 0.01243611 \\
&\quad \cdot 1.0004040025 = 0.007959113
\end{aligned}$$

$$\begin{aligned}
P_{20|k} &= P_{20} - K_b P_{00} \\
&= -0.02 - (-0.01219212) \\
&\quad \cdot 1.0004040025 = -0.007802956
\end{aligned}$$

$$\begin{aligned}
P_{k|k} &= \begin{bmatrix} 0.390305413 & 0.007959113 & -0.007802956 \\ 0.007959113 & 1.039771299 & -1.999751281 \\ -0.007802956 & -1.999751281 & 99.99976416 \end{bmatrix}
\end{aligned}$$

dan seterusnya (entri lain diperbarui). ketidakpastian posisi menurun dari ≈ 1.0004 menjadi ≈ 0.3903 m²; kecepatan dan bias ikut membaik melalui kovarians silang, walau hanya posisi yang terukur.



GAMBAR 5
(Hasil Estimasi Posisi Kalman Filter)

F. COMPLEMENTARY FILTER UNTUK ESTIMASI ORIENTASI

Complementary Filter (CF) adalah salah satu teknik fusi sensor yang memadukan dua sumber data berbeda dengan karakteristik saling melengkapi. Dalam konteks estimasi orientasi, *gyroscope* digunakan karena mampu memberikan respon cepat terhadap perubahan sudut dan mendeteksi gerakan rotasi secara langsung. Namun, *gyroscope* memiliki kelemahan utama, yaitu akumulasi kesalahan (*drift*) yang

muncul akibat proses integrasi dari kecepatan sudut menjadi sudut orientasi. Untuk mendapatkan estimasi orientasi yang akurat dan andal, sistem ini menggabungkan dua jenis sensor yang memiliki kelebihan yang saling melengkapi: sensor referensi absolut (*accelerometer* dan *magnetometer*) dan sensor gerak relatif (*gyroscope*).

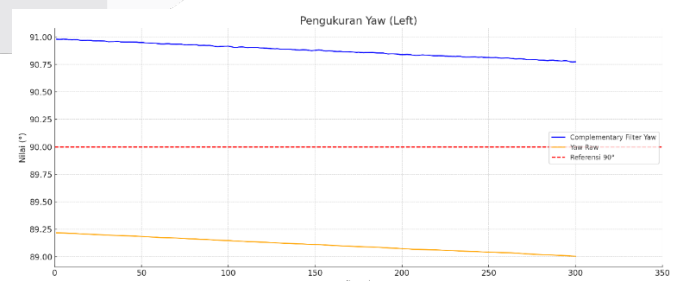
Sensor referensi absolut, seperti *accelerometer* dan *magnetometer*, memberikan informasi orientasi yang stabil dalam jangka panjang. *Accelerometer* sangat baik untuk menentukan sudut pitch (kemiringan ke depan/belakang) dan roll (kemiringan ke samping) dengan mendeteksi arah gaya gravitasi bumi. Sementara itu, *magnetometer* berfungsi seperti kompas digital untuk menentukan arah yaw (putaran horizontal) dengan mengacu pada medan magnet bumi. Kelemahan utama sensor-sensor ini adalah responsnya yang lambat dan rentan terhadap gangguan dari percepatan eksternal (untuk *accelerometer*) atau gangguan magnetik (untuk *magnetometer*). Untuk mengatasi kelemahan masing-masing sensor, digunakan metode fusi data yang dijelaskan oleh Persamaan (1), yang dikenal sebagai *Complementary Filter*. Ide dasarnya adalah melengkapi satu sama lain, data *gyroscope* yang cepat digunakan untuk estimasi jangka pendek, sementara data dari *accelerometer* yang stabil digunakan untuk mengoreksi *drift gyroscope* dalam jangka panjang.

Pengujian dilakukan dengan membandingkan nilai *Yaw Raw* dan *Yaw* hasil *Complementary Filter* terhadap nilai referensi pada empat orientasi: *Left* (90°), *Down* (180°), *Right* (270°), dan *Up* (360°).

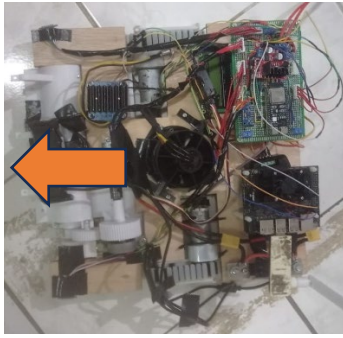
TABEL 1.
(Data Perbandingan Nilai *Yaw Raw* dan *CF Yaw*)

Orientasi	Referensi (°)	Yaw Raw (°)	Bias Raw (°)	CF Yaw (°)	Bias CF (°)
Left (90°)	90.0	89.1	-0.9	90.85	0.85
Down (180°)	180.0	173.4	-6.6	180.9	0.9
Right (270°)	270.0	268.5	-1.5	269.65	-0.35
Up (360°)	360.0	353.5	-6.5	352.9	-7.1

Orientasi Left (90°)



GAMBAR 6
(Orientasi Kiri *Complementary Filter*)



GAMBAR 7
(Arah Orientasi Robot (Kiri))

Data ($k = 150$)

$$\theta_{raw}[k-1] = 89.1114^\circ, \quad \theta_{raw}[k] = 89.1110^\circ$$

$$\theta[k-1] = 90.8811^\circ, \quad \theta[k] = 90.8827^\circ$$

Langkah 1 – Laju putar gyro

$$\Delta\theta_{gyro} = -0.0004^\circ \rightarrow \omega_z = -0.04^\circ/s$$

Langkah 2 – Sudut accel yang diimplikasikan

$$\begin{aligned} \theta_{acc}[k] &= \frac{90.8827 - 0.96(90.8811 - 0.04 \cdot 0.010)}{0.04} \\ &= 90.9307^\circ \end{aligned}$$

Langkah 3 – PID ($\theta = 90.8811^\circ$)

$$e[k] = 90.8811 - 90.8827 = -0.0016^\circ,$$

$$I[k] = 0 + 0.010(-0.0016) = -1.6 \times 10^{-5},$$

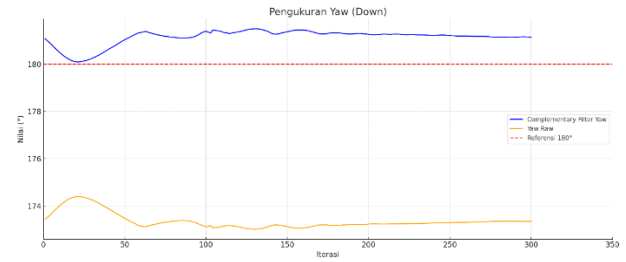
$$D[k] = \frac{-0.0016 - 0}{0.010} = -0.16,$$

$$\begin{aligned} u[k] &= 1.2(-0.0016) + 0.01(-1.6 \times 10^{-5}) \\ &\quad + 0.05(-0.16) = -0.00992016 \end{aligned}$$

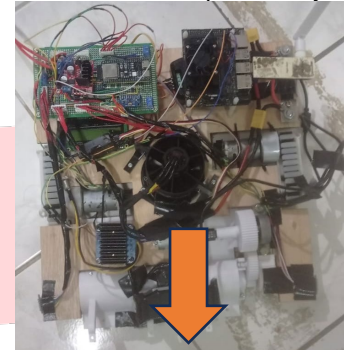
$$\begin{aligned} U_l &= 230 - 0.00992016 = 229.9901, \quad U_R \\ &= 230 + 0.00992016 = 230.0099 \end{aligned}$$

Grafik pada orientasi Left (90°) menunjukkan bahwa kurva *Complementary Filter* (garis biru) berada pada kisaran $90,8^\circ$ hingga 91° , yang sangat dekat dengan nilai referensi 90° . Sementara itu, kurva *Raw* (garis oranye) berada di kisaran $89,0^\circ$ hingga $89,2^\circ$, menunjukkan adanya bias negatif sekitar -1° . Perbedaan ini mengindikasikan bahwa CF mampu mengoreksi kesalahan pembacaan yang terdapat pada data *Raw*, sehingga bias yang tersisa menjadi hanya sekitar $+0,8^\circ$. Selain itu, grafik memperlihatkan bahwa fluktuasi nilai pada CF jauh lebih kecil dibanding *Raw*, menandakan peredaman noise yang efektif. Stabilitas kurva CF menunjukkan bahwa filter bekerja optimal pada kondisi ini, dengan respon yang halus dan tanpa gejala overshoot yang signifikan.

Orientasi Down (180°)



GAMBAR 8
(Orientasi Bawah *Complementary Filter*)



GAMBAR 9
(Arah Orientasi Robot (Bawah))

Data ($k = 150$)

$$\theta_{raw}[k-1] = 173.1152^\circ, \quad \theta_{raw}[k] = 173.1017^\circ$$

$$\theta[k-1] = 181.3892^\circ, \quad \theta[k] = 181.4029^\circ$$

Langkah 1 – Laju putar gyro

$$\Delta\theta_{gyro} = -0.0135^\circ \rightarrow \omega_z = \frac{-0.0135}{0.010} = -0.04^\circ/s$$

Langkah 2 – Sudut accel yang diimplikasikan

$$\begin{aligned} \theta_{acc}[k] &= \frac{181.40259 - 0.96(181.3892 - (-1.35) \cdot 0.010)}{0.04} \\ &= 182.0557^\circ \end{aligned}$$

Langkah 3 – PID ($\theta = 181.3892^\circ$)

$$e[k] = 181.3892 - 181.4029 = -0.0137^\circ,$$

$$I[k] = 0 + 0.010(-0.0137) = -1.37 \times 10^{-4},$$

$$D[k] = \frac{-0.0137 - 0}{0.010} = -1.37,$$

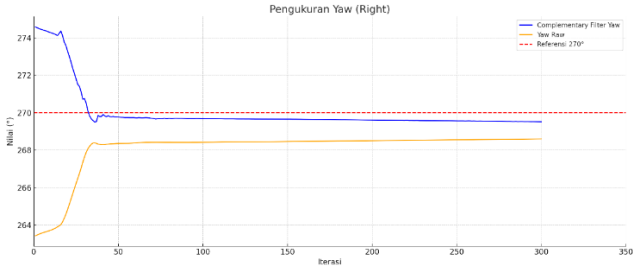
$$\begin{aligned} u[k] &= 1.2(-0.0137) + 0.01(-1.37 \times 10^{-4}) \\ &\quad + 0.05(-1.37) = -0.08494 \end{aligned}$$

$$\begin{aligned} U_l &= 230 - 0.08494 = 229.9151, \quad U_R = 230 + 0.08494 \\ &= 230.08494 \end{aligned}$$

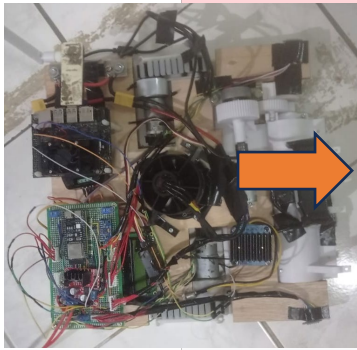
Grafik pada orientasi Down (180°) memperlihatkan perbedaan yang mencolok antara kurva CF dan *Raw*. Nilai CF stabil di kisaran $180,8^\circ$ hingga $181,0^\circ$, dengan bias positif hanya sekitar $+1^\circ$ dari nilai referensi. Sebaliknya, kurva *Raw* berada di kisaran $173,3^\circ$ hingga $173,5^\circ$, yang menunjukkan bias negatif cukup besar, sekitar $-6,5^\circ$. Hal ini mengindikasikan bahwa CF mampu memberikan koreksi signifikan terhadap kesalahan pembacaan yaw mentah.

Grafik juga menunjukkan bahwa nilai CF lebih halus dan minim fluktuasi dibanding Raw, yang terkadang mengalami perubahan kecil yang tidak diinginkan.

Orientasi Right (270°)



GAMBAR 10
(Orientasi Kanan Complementary Filter)



GAMBAR 11
(Arah Orientasi Robot (Kanan))

Data (k = 150)

$$\theta_{raw}[k-1] = 268.4654^\circ, \quad \theta_{raw}[k] = 268.4666^\circ$$

$$\theta[k-1] = 269.6533^\circ, \quad \theta[k] = 269.6514^\circ$$

Langkah 1 – Laju putar gyro

$$\Delta\theta_{gyro} = -0.0012^\circ \rightarrow \omega_z = \frac{-0.0012}{0.010} = -0.12^\circ/s$$

Langkah 2 – Sudut accel yang diimplikasikan

$$\theta_{acc}[k] = \frac{269.6514 - 0.96(269.6533 - 0.12 \cdot 0.010)}{0.04} = 269.577^\circ$$

Langkah 3 – PID ($\theta = 269.6533^\circ$)

$$e[k] = 269.6514 - 269.6533 = -0.0019^\circ,$$

$$I[k] = 0 + 0.010(-0.0019) = 1.9 \times 10^{-5},$$

$$D[k] = \frac{0.0019 - 0}{0.010} = 0.19,$$

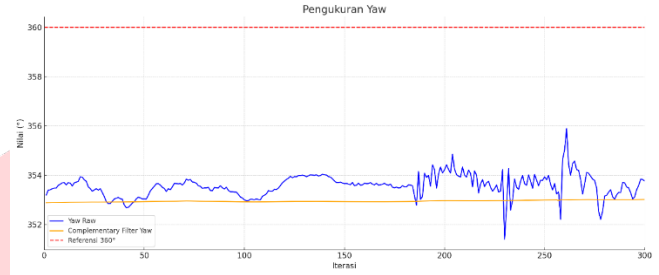
$$u[k] = 1.2(0.0019) + 0.01(1.9 \times 10^{-5}) + 0.05(0.19) = 0.01178019.$$

$$U_l = 230 + 0.01178019 = 230.0118, \quad U_R = 230 + 0.01178019 = 229.9882$$

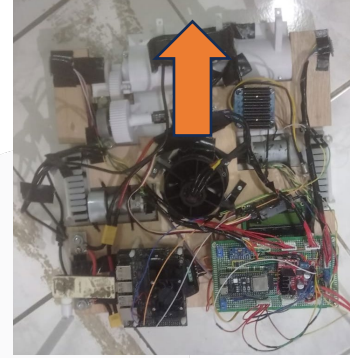
Grafik pada orientasi Right (270°) menunjukkan kinerja CF yang sangat baik. Kurva CF berada pada kisaran 269,6° hingga 269,7°, dengan bias negatif kecil, sekitar -0,3°.

Sebaliknya, kurva Raw berada di kisaran 268,4° hingga 268,6°, dengan bias sekitar -1,4°. Perbedaan yang relatif kecil antara CF dan referensi menunjukkan bahwa kesalahan referensi dari accelerometer pada sudut ini minimal, sehingga CF mampu bekerja hampir tanpa perlu melakukan koreksi besar. Grafik juga menunjukkan kestabilan yang tinggi pada CF, tanpa adanya lonjakan atau osilasi yang signifikan, yang menandakan bahwa filter beroperasi pada kondisi optimal.

Orientasi Up (360°)



GAMBAR 12
(Orientasi Atas Complementary Filter)



GAMBAR 13
(Arah Orientasi Robot (Atas))

Data (k = 150)

$$\theta_{raw}[k-1] = 352.9291^\circ, \quad \theta_{raw}[k] = 352.9287^\circ$$

$$\theta[k-1] = 353.6694^\circ, \quad \theta[k] = 353.6413^\circ$$

Langkah 1 – Laju putar gyro

$$\Delta\theta_{gyro} = -0.0004^\circ \rightarrow \omega_z = \frac{-0.0004}{0.010} = -0.04^\circ/s$$

Langkah 2 – Sudut accel yang diimplikasikan

$$\theta_{acc}[k] = \frac{353.6413 - 0.96(353.6694 - 0.04 \cdot 0.010)}{0.04} = 352.9765^\circ$$

Langkah 3 – PID ($\theta = 353.6694^\circ$)

$$e[k] = 353.6694 - 353.6413 = 0.0281^\circ,$$

$$I[k] = 0 + 0.010(0.0281) = 2.81 \times 10^{-4},$$

$$D[k] = \frac{0.0281 - 0}{0.010} = 2.81,$$

$$u[k] = 1.2(0.0281) + 0.01(2.81 \times 10^{-4}) + 0.05(2.81) = 0.17422281.$$

$$U_l = 230 + 0.17422281 = 230.1742, \quad U_R \\ = 230 + 0.17422281 = 229.8258$$

Grafik pada pengujian rotasi U_p (360°) menunjukkan bahwa meskipun kurva CF memiliki kestabilan yang baik dengan nilai berada di kisaran $352,8^\circ$ hingga $353,0^\circ$, terdapat bias yang cukup besar sekitar -7° dari nilai referensi. Kurva Raw berada di kisaran 353° hingga 354° , dengan bias $-6,5^\circ$. Hal ini menunjukkan bahwa meskipun CF efektif dalam meredam noise dan mempertahankan kestabilan pembacaan, filter tidak mampu mengoreksi offset besar pada yaw absolut jika tidak ada sensor magnetometer yang digunakan. Grafik ini menegaskan keterbatasan CF ketika hanya mengandalkan data gyroscope dan accelerometer untuk estimasi yaw, di mana bias jangka panjang tetap bertahan.

Dari hasil pengujian, *Complementary Filter* efektif dalam meredam noise dan mengurangi error pada raw yaw. Filter juga berhasil mengurangi drift jangka pendek akibat integrasi gyroscope. Namun, akurasi absolut yaw masih menyisakan bias bervariasi ($0,3^\circ$ hingga 7°) tergantung orientasi. Akar permasalahan terletak pada penggunaan accelerometer sebagai referensi yaw, yang secara fisik tidak menyediakan informasi heading absolut.

Untuk mendapatkan yaw absolut yang presisi, diperlukan sensor magnetometer dengan kompensasi tilt. Dengan $\alpha = 0.96$, koreksi drift berjalan cepat ($\sim 0,25$ detik konstanta waktu efektif). Namun, pada skenario tanpa magnetometer, nilai α yang tinggi akan membuat sistem mempertahankan bias tersebut secara konsisten.

V. KESIMPULAN

Penelitian ini telah menghasilkan suatu rancangan sistem navigasi otonom untuk robot pembersih kaca gedung bertingkat dengan integrasi berbagai sensor seperti IMU, sensor Ultrasonik, dan barometer, *Kalman Filter*, dan *Complementary Filter*. Dari hasil pengujian, perlu diketahui bahwa, Sistem mampu memberikan estimasi orientasi dan posisi dengan tingkat presisi yang cukup, namun masih terdapat sejumlah keterbatasan. Sensor IMU menunjukkan deviasi kecil dalam pengukuran pitch dan roll, tetapi pada orientasi yaw 90° terdapat error yang cukup besar, mencapai lebih dari 3 derajat. Sensor Ultrasonik memiliki error sistemik hingga ± 9 mm pada jarak tertentu, sementara sensor barometer mengalami drift kumulatif yang cukup signifikan dalam jangka waktu panjang.

REFERENSI

- [1] W. T. Higgins, "A Comparison of Complementary and Kalman Filtering," *IEEE Transactions on Aerospace and Electronic Systems*, vol. AES-11, no. 3, pp. 321–325, May 1975.
- [2] T. Kim, K. Seo, J. Kim, and H. S. Kim, "Adaptive Impedance Control of a Cleaning Unit for a Novel Wall-Climbing Mobile Robotic Platform (ROPE RIDE)," in *Proc. 2014 IEEE/ASME Int. Conf. on Advanced Intelligent Mechatronics (AIM)*, Besançon, France, Jul. 8–11, 2014, pp. 994–999.
- [3] Y. Jiali and Z. Jihong, "An Angular Acceleration Estimation Method Based on the Complementary Filter Theory," in *Proc. of the 35th Chinese Control Conference*, Chengdu, China, 2016, pp. 7442–7447.
- [4] Y. Kanayama, et al., "A Stable Tracking Control Method for an Autonomous Mobile Robot," in *Proc. of the 1990 IEEE Int. Conf. on Robotics and Automation*, Cincinnati, OH, USA, May 1990, pp. 384–389.
- [5] M. Vega-Heredia, R. E. Mohan, T. Y. Wen, J. S. 'Aisyah, A. Vengadesh, S. Ghanta, and S. Vinu, "Design and Modelling of a Modular Window Cleaning Robot," *Automation in Construction*, vol. 103, pp. 268–278, Jan. 2019.
- [6] S. Yu and Z. Jiang, "Design of the Navigation System through the Fusion of IMU and Wheeled Encoders," *Computer Communications*, vol. 160, pp. 730–737, 2020.
- [7] Y. Katsuki, T. Ikeda, and M. Yamamoto, "Development of a High Efficiency and High Reliable Glass Cleaning Robot with a Dirt Detect Sensor," in *Proc. of the 2011 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, San Francisco, CA, USA, Sep. 2011, pp. 5133–5138.
- [8] A. Sofwan, H. Afrisal, H. R. Mulyana, and A. Goni, "Development of Omni-Wheeled Mobile Robot Based-on Inverse Kinematics and Odometry," in *Proc. of the 2019 6th Int. Conf. on Information Technology, Computer, and Electrical Engineering (ICITACEE)*, Semarang, Indonesia, Sep. 2019, pp. 1–6.
- [9] T. Miyake, H. Ishihara, R. Shoji, and S. Yoshida, "Development of Small-Size Window Cleaning Robot: A Traveling Direction Control on Vertical Surface Using Accelerometer," in *Proc. of the 2006 IEEE Int. Conf. on Mechatronics and Automation (ICMA)*, Luoyang, China, Jun. 2006, pp. 1302–1307.
- [10] H. Zhang, J. Zhang, and G. Zong, "Realization of a Service Climbing Robot for Glass-wall Cleaning," in *Proc. of the 2004 IEEE Int. Conf. on Robotics and Biomimetics*, Shenyang, China, Aug. 2004, pp. 395–400.
- [11] M. Salunke and A. R. Mahajan, "Glass Cleaning Robot for High Rise Buildings," *International Research Journal of Engineering and Technology (IRJET)*, vol. 6, no. 7, pp. 2426–2431, July 2019.
- [12] T. Miyake and H. Ishihara, "Mechanisms and Basic Properties of Window Cleaning Robot," in *Proc. of the 2003 IEEE/ASME Int. Conf. on Advanced Intelligent Mechatronics*, Kobe, Japan, July 2003, pp. 1372–1377.
- [13] A. Pratama, A. H. Pramono, and R. Setiawan, "Movement Control of Two Wheels Balancing Robot Using Cascaded PID Controller," in *Proc. of the 2015 Int. Electronics Symposium (IES)*, Surabaya, Indonesia, Sep. 2015, pp. 48–53.
- [14] M. Vega-Heredia, I. Muhammad, S. Ghanta, V. Ayyalusami, S. Aisyah, and M. R. Elara, "Multi-Sensor Orientation Tracking for a Façade-Cleaning Robot," *Sensors*, vol. 20, no. 5, 1483, Mar. 2020. [Online]. Available: <https://doi.org/10.3390/s20051483>
- [15] ESPP Project Documentation, "Complementary Filter – ESP32 — ESPP main documentation." [Online]. Available: <https://esp-cpp.github.io/espp/filters/complementary.html>.
- [16] PLCDROID, "Pengertian Open Loop dan Close Loop," PLCDROID, diperbarui 06-Jul-2020. [Online]. Tersedia: <https://www.plcdroid.com/2015/04/pengertian-open-loop-dan-close-loop.html>.
- [17] AHRS Developers, "Complementary Filter — AHRS 0.4.0 documentation." [Online]. Available:

<https://ahrs.readthedocs.io/en/latest/filters/complementary.html>.

[18] MIT OpenCourseWare, “16.333: Lecture #15 – Complementary filter examples (Aircraft Stability and Control, Fall 2004).” [Online]. Available: https://ocw.mit.edu/courses/16-333-aircraft-stability-and-control-fall-2004/9bc94307e01f85dbdfce05e87851361a_lecture_15.pdf.

[19] F. Haugen, “Derivation of a Discrete-Time Lowpass Filter,” TechTeach, Mar. 21, 2008. [Online]. Available: https://techteach.no/simview/lowpass_filter/doc/filter_algorithm.pdf

[20] Control Systems Academy, “First-Order Low-Pass Filter Discretization.” [Online]. Available: <https://controlsacademy.com/0020/0020.html>.

[21] C. S. Evans, “Thickness and the Hypsometric Equation,” lecture notes, Dept. Atmos. Sci., Univ. of Wisconsin–Milwaukee. [Online]. Available: PDF..

[22] American Meteorological Society, “Glossary of Meteorology—Hypsometry/Hypsometric Equation.” [Online]. ([American Meteorological Society](#))

[23] G. Welch and G. Bishop, “An Introduction to the Kalman Filter,” Univ. of North Carolina at Chapel Hill, Chapel Hill, NC, USA, Tech. Rep., 2006.

[24] M. S. Grewal and A. P. Andrews, *Kalman Filtering: Theory and Practice with MATLAB*, 4th ed. Wiley, 2015.

[25] S. Särkkä, *Bayesian Filtering and Smoothing*. Cambridge University Press, 2013.