

## Purwarupa Sistem Integrasi *Quadcopter* dan *Mobile Robot*

Andi Dharmawan\*<sup>1</sup>, Christian Antonia Lusiarta Putera<sup>2</sup>

<sup>1</sup>Jurusan Ilmu Komputer dan Elektronika, FMIPA, UGM, Yogyakarta

<sup>2</sup>Program Studi Elektronika dan Instrumentasi, JIKE, FMIPA, UGM, Yogyakarta

e-mail: \*[dharmawan.andi@gmail.com](mailto:dharmawan.andi@gmail.com), [assael\\_07@yahoo.com](mailto:assael_07@yahoo.com)

### **Abstrak**

*Quadcopter, juga dikenal sebagai quadrotor, adalah helikopter dengan empat motor yang dilengkapi dengan empat buah propeller pada masing-masing motornya yang digunakan untuk terbang dan bermanuver. Pada quadcopter terdapat 3 variabel sudut yang menjadi element utama dalam pengendaliannya yaitu roll, pitch, dan yaw yang diperoleh dari penggabungan sensor accelerometer, gyroscope dan magnetometer dengan metode Direction Cosine Matrix. Pada sistem integrasi mobile robot dan quadcopter, digunakan kontroler jenis Proporsional-Integral-Derivatif (PID Controller) untuk meminimalisir nilai error dari sudut roll, pitch, dan yaw quadcopter dengan cara mengatur keluaran dari proses kontrol agar quadcopter dapat terbang stabil saat membawa mobile robot.*

*Pada sistem ini, quadcopter selain juga digunakan sebagai pengendali mobile robot, dimana quadcopter akan mengirim karakter dengan menggunakan komunikasi serial melalu RF YS1020 kemudian mobile robot membaca data yang dikirim oleh quadcopter yang masuk secara serial melalui RF Transceiver YS1020, kemudian menentukan kecepatan motor mobile robot dengan menggunakan cara membandingkan data berupa karakter ASCII yang masuk untuk melakukan mode gerak.*

**Kata kunci**—*Direction Cosine Matrix, Kontroler PID, integrasi, quadcopter, mobile robot*

### **Abstract**

*Quadcopter, also known as the quadrotor, helicopter with four motors are equipped with four propellers on each bike that used to fly and maneuver. In Quadcopter there are 3 variable angles to the main element in the control roll, pitch, and yaw that derived from combining sensor accelerometer, gyroscope and magnetometer methods Direction Cosine Matrix. In the mobile robot system integration and Quadcopter, use the Proportional-Integral controller type-Derivative (PID Controller) to minimize the error value from the point of roll, pitch, and yaw Quadcopter by regulating the output of the control process in order to fly Quadcopter stabilized when carrying a mobile robot.*

*In this system, as well Quadcopter used as a mobile robot controller, which sends the character Quadcopter using serial communication through mobile robot RF YS1020 then read the data sent by the incoming serial Quadcopter via RF Transceiver YS1020, and then determines the motor speed mobile robot with use by comparing the data in the form of ASCII characters that go to make a motion mode.*

**Keywords**—*Direction Cosine Matrix, PID controllers, integration, quadcopter, mobile robot*

## 1. PENDAHULUAN

Pesawat tanpa awak (UAV) adalah pesawat yang tidak menggunakan pilot untuk pengemudinya dan bergerak secara otomatis dengan bantuan alat elektronik. Awalnya, pembuatan pesawat tanpa awak untuk tujuan perang, tetapi sekarang banyak orang menggunakannya untuk keperluan lain. Seperti fotografi udara, survai daerah, pengintaian dari udara, penelitian kelautan.[1] Salah satu jenis pesawat tanpa awak yang digunakan dalam penginderaan jarak jauh adalah *quadcopter* atau *quadrotor*.

*Quadcopter*, juga dikenal sebagai *quadrotor*, adalah pesawat dengan empat motor yang dilengkapi empat buah *propeller* pada masing-masing motornya yang digunakan untuk terbang dan bermanuver. Motor-motor tersebut diarahkan ke atas dan ditempatkan dalam suatu bentuk formasi persegi dengan jarak yang sama dari pusat massa *quadcopter*. Suatu *quadcopter* dikendalikan dengan menyesuaikan kecepatan dari empat buah motor yang berputar pada masing-masing lengannya.[2]

*Quadcopter* saat ini menjadi suatu wahana yang sering digunakan dalam saat melakukan pemantauan wilayah pasca bencana alam. Salah satu kendala terbesar saat melakukan pemantauan visual daerah pasca bencana alam adalah adanya banyak reruntuhan yang menghalangi Tim SAR. Untuk itu digunakan *quadcopter* dan *mobile robot* untuk membantu melakukan pemantauan visual daerah pasca bencana alam. *Quadcopter* dapat digunakan untuk mengangkat *mobile robot* untuk menuju ke daerah yang sulit dijangkau sekaligus melakukan pemantauan citra visual daerah pasca bencana alam dari udara. Sedangkan disisi lain, untuk membantu Tim SAR melakukan pemantauan visual dari darat dapat digunakan *mobile robot*. Pada skripsi ini akan dibahas sistem integrasi *mobile robot* dan *quadcopter* sehingga dapat saling mendukung dalam melakukan pemantauan visual daerah pasca bencana alam. Dengan adanya sistem integrasi yang memungkinkan *mobile robot* untuk dibawa terbang didalam *quadcopter*, maka *mobile robot* dapat menjangkau daerah-daerah yang kurang dapat dijangkau *quadcopter*.

## 2. METODE PENELITIAN

### 2.1 Tinjauan Pustaka

Pesawat tanpa awak (UAV) adalah pesawat yang tidak menggunakan pilot untuk pengemudinya dan bergerak secara otomatis dengan bantuan alat elektronik. Awalnya, pembuatan pesawat tanpa awak untuk tujuan perang, tetapi sekarang banyak orang menggunakannya untuk keperluan lain. Seperti fotografi udara, survai daerah, pengintaian dari udara, penelitian kelautan.[1]

Pada *quadcopter* terdapat 3 variabel sudut yang menjadi element utama dalam pengendaliannya. Roll, pitch, dan yaw adalah ketiga sudut yang menjadi elemen utama pengendalian *quadcopter*. Posisi linear absolut dari *quadcopter* didefinisikan dalam kerangka inersia sumbu x, y, z dengan  $\xi$ . Sikapnya, yaitu posisi sudut, didefinisikan dalam kerangka inersia dengan tiga sudut Euler  $\eta$ . *Pitch* sudut  $\theta$  menentukan rotasi *quadcopter* sekitar sumbu y. *Roll* sudut  $\phi$  menentukan rotasi di sekitar sudut sumbu x dan *yaw* sekitar sumbu z. Vektor q berisi vektor posisi linier dan *angular*. Kecepatan sudut *rotor* akan menimbulkan gaya ke arah sumbu *rotor*. Kecepatan sudut dan percepatan *rotor* juga menciptakan torsi sekitar sumbu *rotor* dimana terdapat daya angkat konstan Gabungan gaya dari *rotor* membuat dorongan ke arah tubuh sumbu z. Torsi  $\tau_B$  terdiri dari torsi  $\tau_\phi$ ,  $\tau_\theta$  dan  $\tau_\psi$  ke arah sudut kerangka tubuh yang berhubungan [3]

Pada *quadcopter* digunakan *Direction Cosine Matrix (DCM) Filter* yang bertujuan untuk menentukan langkah berikutnya dalam fungsi stabilisasi dan kontrol dari *quadcopter* tanpa awak agar stabil dengan kontrol *elevator* dan kontrol kemudi. Metode DCM Filter ini digunakan untuk mengatasi masalah-masalah yang timbul saat *quadcopter* terbang yaitu

pencampuran dan percepatan. Pada *quadcopter* gerakan berputar diperoleh dengan mengurangi kecepatan rotor kedua dan meningkatkan kecepatan rotor keempat. Demikian pula, gerakan *pitch* diperoleh dengan mengurangi kecepatan rotor pertama dan meningkatkan kecepatan rotor ketiga. Gerakan *roll* diperoleh dengan meningkatkan dengan kecepatan sudut dua rotor berlawanan dan penurunan kecepatan dari dua lainnya. Kinematika terdiri dari hubungan antara inersia koordinat badan *quadcopter*  $\mathbf{x}_b$ ,  $\mathbf{y}_b$ , dan  $\mathbf{z}_b$  terhadap koordinat permukaan bumi  $\mathbf{x}_e$ ,  $\mathbf{y}_e$ , dan  $\mathbf{z}_e$  yang akan dihubungkan dalam bentuk matriks. Matriks rotasi tersebut menggambarkan orientasi satu sistem koordinat yang berhubungan dengan lain. Kolom matriks adalah vektor unit dalam satu sistem seperti terlihat pada sistem lainnya. Sebuah vektor dalam satu sistem dapat ditransformasikan ke dalam sistem lain dengan mengalikannya dengan matriks rotasi. Transformasi ke arah yang sebaliknya dilakukan dengan invers dari matriks rotasi, yang ternyata sama dengan transposnya dimana transpos yang digunakan hanya swap baris dan kolom. Vektor Satuan berguna dalam kontrol dan perhitungan navigasi, karena panjang keduanya sama. Oleh karena itu bentuk matriks diatas dapat digunakan dalam dot dan produk silang untuk mendapatkan sinus atau cosinus dari berbagai sudut.[4]

Menurut Kristiyanta [5] , mengatakan robot merupakan suatu perangkat mekanik yang mampu menjalankan tugas-tugas fisik, baik dibawah kendali dan atau dijalankan dengan serangkaian program yang telah didefinisikan terlebih dahulu (kecerdasan buatan). Robot bergerak membutuhkan navigasi dalam melakukan gerakan pada arah tertentu. Sehingga koordinasi antara piranti masukan-masukan berupa sensor posisi relatif terhadap titik acuan navigasi yang sudah dipetakan dengan piranti penggerak berupa motor.

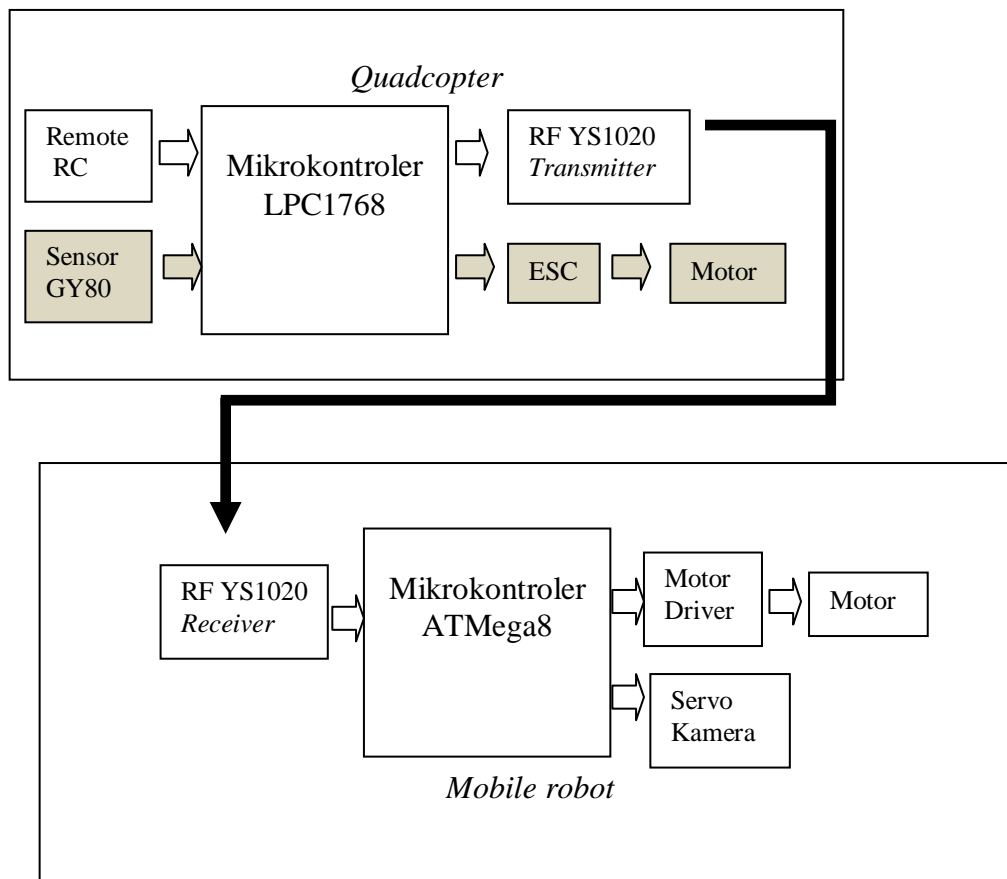
## 2.2 Analisis dan Perancangan Sistem

Rancangan sistem secara keseluruhan dibagi menjadi 2 buah bagian utama, yaitu perancangan sistem dari *quadcopter* sebagai transporter *mobile robot* dan rancangan sistem dari *mobile robot*. Rancangan sistem *quadcopter* secara keseluruhan dibagi menjadi rancangan mekanik *quadcopter*, rancangan perangkat keras (*hardware*) *quadcopter*, rancangan perangkat lunak (*software*) pada *quadcopter*, dan rancangan sistem kontrol keseimbangan *quadcopter* dengan menggunakan pengontrol PID (*PID Controller*). Sedangkan rancangan sistem *mobile robot* secara keseluruhan dibagi menjadi rancangan perangkat keras (*hardware*) *mobile robot*, rancangan perangkat lunak (*software*) pada *mobile robot*, dan rancangan sistem kontrol manual pada *mobile robot* dengan menggunakan komunikasi wireless antara *mobile robot* dan *quadcopter*.

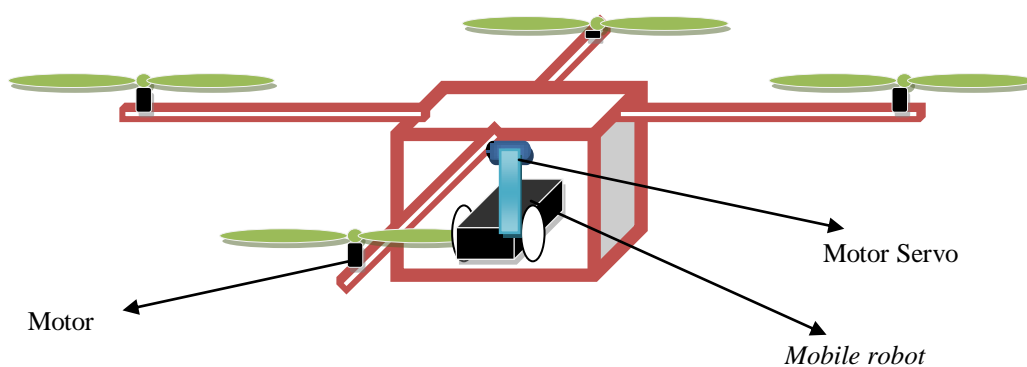
Sistem kontrol pada *mobile robot* menggunakan sistem kontrol manual dimana *mobile robot* akan mendapatkan perintah untuk melakukan mode gerak dari pengguna atau user melalui komunikasi wireless. Rancangan software meliputi pengaturan mode gerak dari *mobile robot* dengan menggunakan komunikasi serial secara wireless berbasis radio frequency YS1020.

Pada Gambar 1 dapat dilihat diagram blok rancangan integrasi sistem. Pada sistem integrasi antara *quadcopter* dan *mobile robot*, *quadcopter* akan difungsikan sebagai media transportasi *mobile robot* atau sebagai kapal induk. Sistem *quadcopter* digunakan untuk membawa *mobile robot* ke suatu area yang sulit dijangkau oleh *mobile robot* saat melakukan proses monitoring. Kemudian setelah *quadcopter* mendarat, *quadcopter* akan difungsikan sebagai pengendali *mobile robot* dengan cara memproses sinyal kendali masukan dari remote RC yang kemudian akan diolah oleh mikrokontroler MBED LPC1768 pada *quadcopter* untuk diubah menjadi karakter serial yang selanjutnya akan dikirim melalui RFYS1020 pada *quadcopter* ke *mobile robot*. *Quadcopter* akan meneruskan sinyal remote RC dari pengguna menjadi sinyal komunikasi serial wireless dengan menggunakan radio frequency YS1020 untuk mengontrol mode gerak dari *mobile robot*. Pada *mobile robot* akan dipasang kamera yang

dilengkapi dengan video sender yang berfungsi untuk mengambil data video dari area yang dilalui *mobile robot* dan mengirimkannya ke video receiver pengguna.



Gambar 1 Rancangan sistem keseluruhan



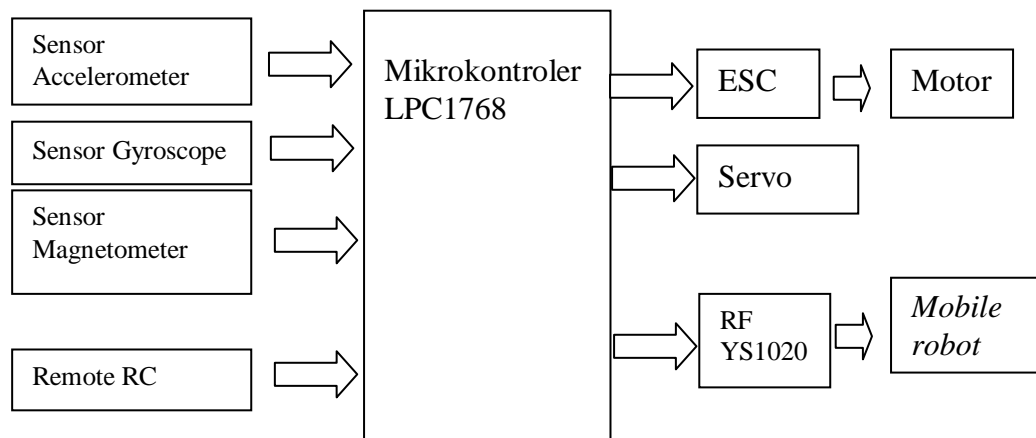
Gambar 2 Rancangan mekanik quadcopter tampak samping

Pada sistem elektronik quadcopter terdapat MBED LPC 1768 sebagai pengendali utama pada sistem *quadcopter* yang mendapatkan masukan terhubung dengan sensor *10DOF module nine axis attitude the instrument L3G4200D ADXL345 HMC5883L BMP085 module* dan receiver remote RC, serta keluaran yang terhubung pada *Electronic Speed Control (ESC)* untuk menggerakkan motor. Sistem kontrol pada *quadcopter* ini memperoleh masukan dari *10DOF module nine axis attitude the instrument L3G4200D ADXL345 HMC5883L BMP085 module*

yang didalam module tersebut terdapat sensor akselerometer ADXL345, sensor *gyroscope* L3G200D, dan sensor magnetometer HMC5883L berupa nilai posisi kemiringan dari *quadcopter*. Agar nilai kemiringan yang terbaca akurat, maka peletakan sensor-sensor tersebut harus sedekat mungkin dengan pusat massa atau *Center of Gravity (CoG) quadcopter*.

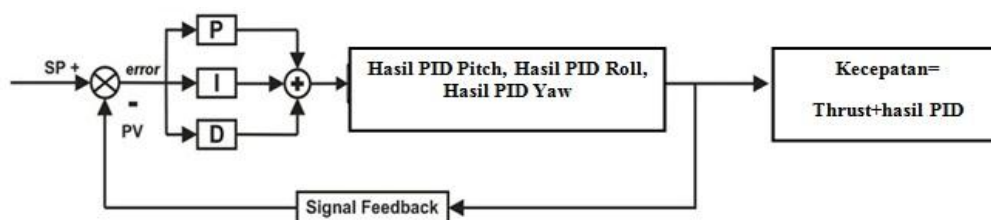
Keluaran utama dari sistem ini adalah informasi nilai *Pulse Width Modulation (PWM)* yang digunakan untuk menggerakkan motor brushless *quadcopter* melalui driver ESC dan nilai posisi sudut servo garasi untuk membuka tutup garasi. Nilai PWM yang diberikan ke ESC terdapat dari perhitungan pengontrol PID dari sudut *Yaw, Pitch, dan Roll* tetap stabil ketika terkena gangguan dari luar. Informasi posisi sudut ini akan diolah oleh mikrokontroler MBED LPC 1768 yang selanjutnya akan dikirimkan ke *Electronic Speed Controller* yang kemudian *Electronic Speed Controller* inilah yang akan mengatur kecepatan motor brushless pada masing-masing lengan *quadcopter*.

Garis besar jalannya program adalah membaca nilai kemiringan dari *quadcopter* dan membandingkannya dengan nilai kemiringan referensi (*desire state*), kemudian menentukan kecepatan motor brushless pada masing-masing lengan *quadcopter* dengan menggunakan pengontrol PID. Pengontrol PID adalah pengontrol utama yang digunakan pada sistem ini. Parameter masukan utama yang digunakan dalam pengontrol PID adalah nilai *error* atau selisih nilai yang luaran sistem (variabel proses) dengan nilai yang diharapkan (*set point*).



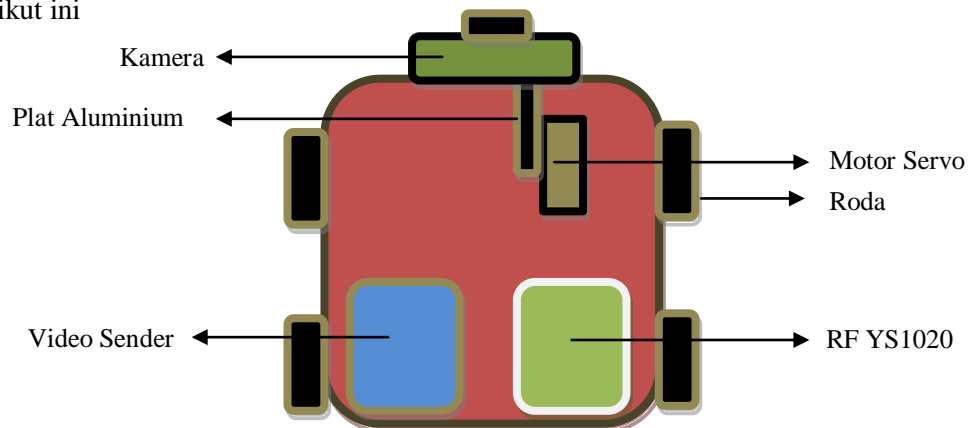
Gambar 3 Blok diagram rancangan perangkat keras

Nilai *desire state* sumbu *Yaw*, sumbu *Pitch* dan sumbu *Roll quadcopter* didapat dari pengambilan sampel data kemiringan awal *quadcopter* pada posisi tegak lurus permukaan bumi. Untuk memastikan posisi badan *quadcopter* tegak lurus dengan permukaan bumi digunakan waterpas. Sedangkan parameter *outputnya* adalah pulsa PWM dengan lebar tertentu yang akan digunakan untuk mengatur kecepatan putar pada motor brushless di masing-masing lengan *quadcopter*.



Gambar 4 Diagram blok aksi kontrol pengontrol PID yang digunakan

Rancangan bagian mekanik pada *mobile robot* secara umum dapat dilihat pada Gambar 5 berikut ini



Gambar 5. Rancangan mekanik *mobile robot* tampak atas

Pada Gambar 5 dapat dilihat rancangan mekanik *mobile robot* tampak atas. Pada bagian atas terdapat platform atas yang berisi kamera yang terhubung dengan servo untuk mengatur sudut kamera, RF YS1020 dan video sender sistem. Pada rancangan mekanik *mobile robot* bagian bawah. Pada bagian bawah *mobile robot* terdapat sistem minimum mikrokontroler dan driver motor *mobile robot*.

## 2.2. Implementasi Sistem

Pada implementasi sistem, badan *quadcopter* dibuat dari bahan pipa aluminium kotak yang disatukan dengan kayu. Gambar 6 dibawah ini menunjukkan bentuk mekanik *quadcopter*.



Gambar 6 Bentuk mekanik *quadcopter* keseluruhan

Pada Gambar 6 terlihat bentuk mekanik *quadcopter* secara keseluruhan. Pada bagian bawah diberi sebuah kotak sebagai tempat garasi *mobile robot*. Sebagai pengunci garasi agar *mobile robot* tidak jatuh saat terbang pada garasi diberi plat yang terpasang pada servo. Pusat massa (*CoG, Center of Gravity*) dari *quadcopter* diletakkan pada titik pusat pertemuan dari sumbu *Roll* dan sumbu *Pitch quadcopter*. Demikian juga pemasangan Sensor IMU GY80 yang diletakkan pada persimpangan semua lengan *quadcopter*, dengan tujuan sumbu x dan sumbu y searah dengan masing masing lengan *quadcopter*.

Pada Gambar 7 terlihat bentuk mekanik *mobile robot* tampak atas dimana pada bagian atas terdapat *video sender*, RF YS1020, kamera dan servo untuk mengatur posisi kamera. Pada mekanik *mobile robot* bagian bawah terdapat bagian elektronis *mobile robot* yang terhubung dengan motor DC dan sistem gearbox untuk menggerakkan *mobile robot*.

Gambar 7. Mekanik *mobile robot* tampak atas

### 3. HASIL DAN PEMBAHASAN

#### 3.1. Pengujian Sensor

Pada pengujian sistem *quadcopter* akan diuji data output dari sensor dan proses stabilisasi *quadcopter* dengan menggunakan metode pengontrolan PID sederhana. Penentuan konstanta-konstanta yang digunakan berdasarkan metode eksperimen. Pada pengujian sensor IMU GY-80 digunakan *Direction Cosine Matrix (DCM) Filter* yang bertujuan untuk menentukan langkah berikutnya dalam fungsi stabilisasi dan kontrol dari *quadcopter* tanpa awak agar stabil dengan kontrol *elevator* dan kontrol kemudi.

Cara kerja DCM Filter ialah dengan menggunakan sensor *gyroscope* sebagai sumber utama informasi orientasi dengan cara mengintegrasikan persamaan diferensial nonlinear kinematis yang berhubungan dengan laju perubahan waktu yang berorientasi dari *quadcopter* untuk menilai putaran, dan orientasi yang sekarang. Hal ini dilakukan pada frekuensi tingkat tinggi, (40 sampai 50 Hz) sehingga *update* data yang dihasilkan akan cukup untuk memberikan informasi pada motor *brushless* untuk pulsa PWM setiap yang dikirimkan ke motor *brushless*. Pada tabel-tabel berikut ini dapat dilihat data hasil pengujian yang didapatkan dari sensor IMU GY-80.

Tabel 1. Pengujian hasil output sensor sumbu *pitch* pada sudut  $0^{\circ}$ 

Data Ke-	Sudut keluaran ( $^{\circ}$ )	Keluaran Gyroscope (rad/s)	Keluaran DCM ( $^{\circ}$ )
1	0,4	0	0,3
2	0,4	0	0,2
3	0	0	0,3
4	0,2	0	0,2
5	0,4	0	0,2

Tabel 2. Pengujian Hasil Output Sensor Sumbu *Pitch* pada Sudut  $10^{\circ}$ 

Data Ke-	Sudut keluaran ( $^{\circ}$ )	Keluaran Gyroscope (rad/s)	Keluaran DCM ( $^{\circ}$ )
1	11	-4,2	11,3
2	11,4	-4,2	11,3
3	11,2	-4,2	11,2
4	11	-4,2	11,2
5	11,1	-4,2	11,3



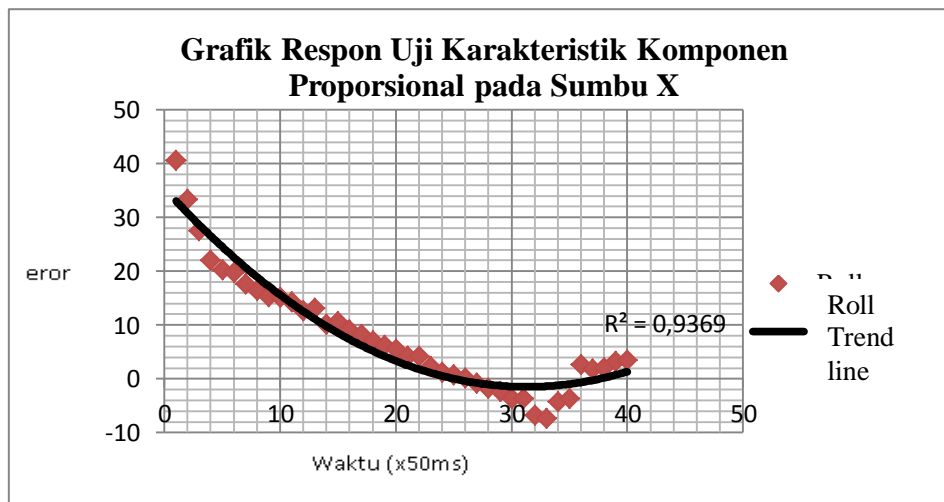
Pada Tabel 1 dan Tabel 2 diatas dapat dilihat hasil pengujian sensor IMU GY-80 pada sumbu *Pitch* yang dibandingkan dengan sudut pada penggaris busur. Untuk mendeteksi sudut pada sumbu *Pitch* dan sumbu *Roll* digunakan sensor *accelerometer* dan sensor *gyroscope* dikarenakan posisi sumbu *Pitch* dan sumbu *Roll* yang tegak lurus dengan bumi sehingga perubahan posisi sudut pada *quadcopter* akan menyebabkan perubahan percepatan gravitasi bumi yang dideteksi oleh *accelerometer*.

Perubahan posisi pada *quadcopter* terhadap sistem koordinat bumi akan menyebabkan adanya perubahan nilai percepatan gravitasi *accelerometer* terhadap permukaan bumi yang digunakan untuk mendeteksi sudut *Roll* dan *Pitch quadcopter*. Demikian juga perubahan posisi sistem koordinat *quadcopter* pada sudut *Yaw* akan mengakibatkan adanya perubahan nilai pada sensor *magnetometer* terhadap kutub magnetik bumi sehingga magnetomer digunakan sebagai referensi untuk penghitungan sudut *Yaw*. Untuk mengatasi kesalahan numerik, *gyroscope* akan mengarah dan mengimbangi secara bertahap akan menumpuk kesalahan dalam elemen DCM. Pada sistem digunakan vektor referensi untuk mendeteksi kesalahan, dan proportional integral (PI) yang memberikan umpan balik yang ideal antara kesalahan yang terdeteksi dan input gyro yang digunakan pada langkah awal. Data yang didapat pada Tabel 1 dan Tabel 2 mendekati sudut terukur (sudut referensi) dari busur dengan ralat ketelitian sekitar  $\pm 1^{\circ}$ .

### 3.2. Pengujian kontrol keseimbangan pada *quadcopter*

Pengujian selanjutnya ialah pengujian kontrol keseimbangan. Pengujian kontrol keseimbangan ini dilakukan dalam empat tahap. Tahap pertama adalah pengujian pengontrol pada sumbu X atau sumbu *Roll*, tahap kedua adalah pengujian pengontrol pada sumbu Y atau sumbu *Pitch*, tahap ketiga adalah pengujian pengontrol pada sumbu Z atau sumbu *Yaw*, dan tahap keempat adalah pengujian pengontrol PID pada sumbu X,Y dan Z pada saat *quadcopter* terbang.

Pada pengujian ini konstanta yang diberi nilai hanya komponen proporsional ( $K_p_{Roll}$ ). Pada pengujian ini divariasikan dengan dua nilai  $K_p_{Roll}$  yang berbeda. Hasil yang diperoleh dari pengujian ini ditampilkan pada Gambar 8



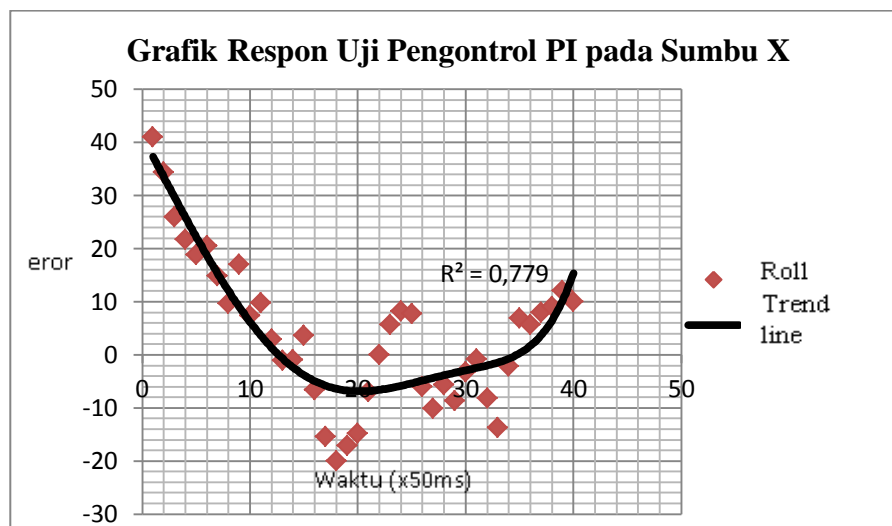
Gambar 7 Grafik respon uji keseimbangan *quadcopter* dengan nilai  $k_p_{roll} = 2,5$

Pada pengujian ini diuji penggunaan pengontrol Proporsional pada *quadcopter*. Dengan menggunakan rumus  $u(k) = K_p \cdot e_k$ . Nilai  $e_k$  didapatkan dari selisih nilai antara nilai *desire state* dengan nilai *present output* data *Roll*. Pada Gambar 8 dapat terlihat dengan adanya pengontrol Proporsional maka *present error* yang terjadi pada *quadcopter* dapat langsung dikoreksi. Tetapi *respon time* yang dibutuhkan sistem masih lambat. Pada pengontrol PID,



karakteristik komponen proporsional adalah langsung merespon nilai *error* yang terjadi dan meminimalisirnya secara perlahan. Dan semakin tinggi nilai *gain* proporsional ( $K_p_{Roll}$ ) yang diberikan, maka sistem akan semakin responsive terhadap *error* yang terjadi. Tetapi akan menyebabkan osilasi yang cukup besar.

Pengontrol selanjutnya yang diuji adalah pengontrol jenis PI, yaitu dengan menambahkan komponen integral pada pengontrol proporsional kemudian diamati *time respon* dari sistem *quadcopter*.



Gambar 9 Grafik respon keseimbangan *quadcopter* dengan nilai  $K_p_{Roll} = 4$ , dan  $K_I_{Roll} = 2$

Pada Gambar 9 menunjukkan grafik respon dari pengontrol jenis PI terhadap *quadcopter*. Penambahan komponen pengontrol integratif pada tersebut mengikuti persamaan 2 yaitu dengan mengambil komponen pengontrol proporsional dan integratif, sehingga dihasilkan persamaan berikut ini

$$u(k) = K_p \cdot e_k + K_I \cdot T \sum_0^k e_k. \quad (6)$$

Bentuk diskrit dari persamaan tersebut ialah

$$X = K_p \cdot error + K_i \cdot integralError \quad (7)$$

dimana

$$integralError = last\_integralError + error \cdot Ts;$$

$$last\_integralError = integralError;$$

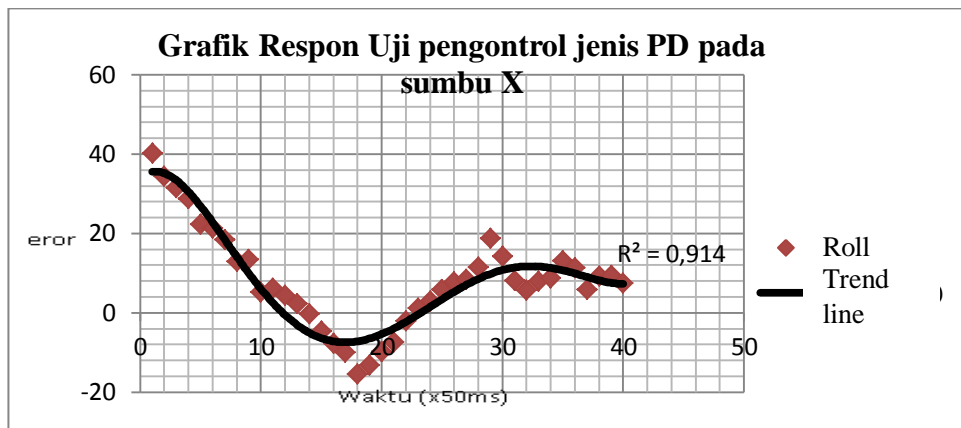
Konstanta  $T_s$  ialah *time sampling* dari sistem, yaitu 0,02 s. Pengaturan KI yang salah akan mengakibatkan *quadcopter* miring saat terbang. Dari Gambar 9 diatas, dapat dilihat bahwa jenis pengontrol PI kurang sesuai saat digunakan dalam sistem kontrol keseimbangan *quadcopter* pada sumbu X ini karena *quadcopter* akan terus menerus osilasi.

Pengujian sistem kontrol kesimbangan selanjutnya ialah dengan menggabungkan jenis pengontrol proporsional dan derivatif mengikuti persamaan 2, dengan cara menghilangkan komponen integratif pada persamaan tersebut, sehingga didapatkan persamaan

$$u(k) = K_p \cdot e_k + \frac{1}{T} K_D (e_k - e_{k-1}) \quad (8)$$

Bentuk diskrit dari persamaan 6.3 tersebut adalah

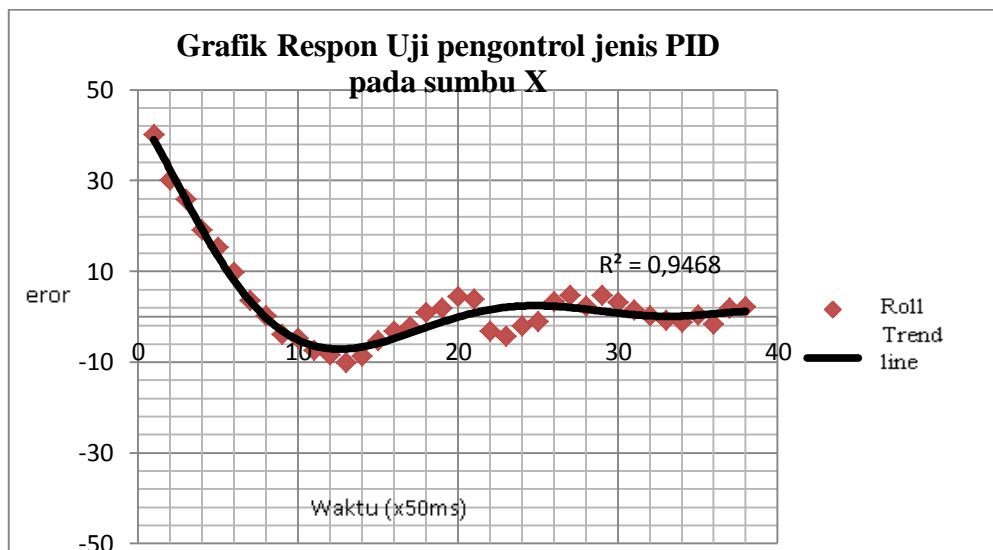
$$X_o = K_p \cdot error + K_d \cdot (e - last\_e); \quad (9)$$



Gambar 10 Grafik respon keseimbangan *quadcopter* dengan  $K_p\_Roll = 2,5$  dan  $KD\_Roll = 1$

Pada Gambar 10 dapat diamati bahwa pengaruh konstanta *derivative* adalah untuk menurunkan *overshoot* terjadi pada sistem. Atau dengan kata lain fungsi dari konstanta *derivative* adalah untuk meredam *overshoot* yang terjadi akibat dari kenaikan konstanta proporsional. Konstanta *derivative* hanya akan bereaksi jika terjadi perubahan error sehingga konstanta ini sama seperti konstanta proporsional yaitu tidak bisa digunakan sendirian atau harus diimbangi oleh konstanta lain. Dalam hal ini antara konstanta proporsional dan *derivative* bersifat saling melengkapi.

Pengontrol terakhir yang diuji adalah pengontrol jenis PID. Pengontrol PID adalah pengontrol yang paling kompleks jika dibandingkan dengan pengontrol jenis P, PI, dan PD karena pengontrol jenis PID ini menggunakan ketiga buah komponen kontrol (Proporsional, Integral, dan Derifatif). Gambar 11 berikut adalah hasil uji coba menggunakan pengontrol jenis PID.



Gambar 11 Grafik respon keseimbangan *quadcopter* dengan nilai  $K_p\_Roll = 2,5$ ,  $KI\_Roll = 0,8$  dan  $KD\_Roll = 1$

Pada pengujian pengontrol PID ini, digunakan persamaan 2, yaitu dengan cara menggabungkan komponen pengontrol proporsional, integratif, dan derivatif mengikuti persamaan:

$$u(k) = K_p \cdot e_k + K_I \cdot T \sum_0^k e_k + \frac{1}{T} K_D (e_k - e_{k-1}) \quad (10)$$

Pada penggunaan pengontrol jenis PD sebelumnya, saat *quadcopter* mendekati stabil, badan *quadcopter* masih mendeteksi eror yang mengakibatkan miring, saat ditambahkan atau dikurangi nilai konstanta KP dan KD masih terdapat satu simpangan kecil yang masih bisa dihilangkan. Kenaikan nilai atau penurunan nilai kedua konstanta KP dan KD jika hanya menggunakan pengontrol PD tersebut bukan akan mengoreksi kondisi *steady state error* tetapi hanya akan membuat sistem semakin osilasi karena pengaruh konstanta *derivative* adalah untuk menurunkan *overshoot* terjadi pada sistem atau untuk meredam *overshoot* yang terjadi akibat dari kenaikan konstanta proporsional.

Untuk menghilangkan *steady state error* kemiringan tersebut tersebut, maka ditambahkan komponen Integral pada sistem dengan nilai *gain* yang relatif kecil ( $K_i_{Roll} = 1$ ). Hasil pengujian penambahan komponen integral ini ternyata benar mampu menghilangkan simpangan kecil dari kondisi *desire state* yang sebelumnya muncul pada pengontrol jenis PD yang diujikan sebelumnya. Waktu yang dibutuhkan *quadcopter* untuk kembali ke posisi stabil juga meningkat menjadi hanya 0,8 detik saja. Berdasarkan pengujian jenis pengontrol diatas, dapat dilihat bahwa jenis pengontrol yang paling efektif untuk digunakan untuk proses stabilisasi *quadcopter* adalah jenis pengontrol PID.

Pada uji coba pengontrol jenis PID sudah didapatkan hasil yang memuaskan terutama pada pengaturan konstanta  $KP_{Roll} = 2,5$ ;  $KI_{Roll} = 0,8$ ; dan  $KD_{Roll} = 1$  didapatkan data yang memiliki noise sensor paling sedikit. Pada penambahan pengontrol I disini akan mengoreksi *error* saat *quadcopter* mendekati stabil agar *error* mendekati nilai 0.

Pengujian selanjutnya ialah pengujian karakter yang diterima oleh mobile robot dengan cara mengirimkan ulang karakter yang diterima oleh mobile robot dan menuju *receiver* YS1020 yang terdapat pada komputer pengguna. Pada tabel Tabel 3 berikut ini dapat dilihat data pengujian mode gerak yang terjadi pada mobile robot.

Tabel 3 Data pengujian mode gerak *mobile robot*

Posisi <i>Stick Remote RC</i>	Karakter pada <i>Hyperterminal</i>	Perubahan pada <i>Mobile robot</i>
Atas	w	maju
Bawah	s	mundur
Kiri	ao	Belok kiri, servo kamera naik
Kanan	bo	Belok kanan, servo kamera naik
<i>Throttle Up</i>	i	Servo kamera turun

Pada *hyperterminal* terlihat karakter saat posisi *stick* keatas ialah “w”. Karakter tersebut sudah sesuai dengan karakter yang akan digunakan oleh program pada *mobile robot* untuk menggerakkan *mobile robot* maju kedepan sehingga robot bergerak maju kedepan. Selanjutnya pada saat posisi *stick* remot kebawah, maka dikirim karakter “s” yang selanjutnya akan dibaca oleh *mobile robot* untuk bergerak mundur. Pada saat posisi *stick* kearah samping kanan atau kiri dikirim 2 buah karakter, yaitu “ao” dan “bo”, karakter “a” dan “b” digunakan untuk melakukan mode belok pada *mobile robot*, selanjutnya karakter “o” digunakan untuk menaikkan posisi kamera. Pengiriman karakter “o” pada saat mode belok digunakan agar setelah *mobile robot* keluar dari garasi *quadcopter* dan berbelok, kamera *mobile robot* mengangkat. Sedangkan saat remote RC diposisikan pada “*throttle up*” atau *quadcopter* mulai terbang, posisi servo kamera

diturunkan agar tidak menabrak bagian atas garasi *quadcopter* dan dapat dilakukan pengamatan video obyek dibawah *quadcopter* saat terbang.

#### 4. KESIMPULAN

Kesimpulan yang dapat diambil dari penelitian ini adalah sebagai berikut:

1. Proses integrasi sistem *quadcopter* dan *mobile robot* dilakukan dengan cara mengatur mode-mode gerak pada *mobile robot* oleh *quadcopter* agar *mobile robot* dapat dan *quadcopter* dapat saling mendukung.
2. Proses untuk mendapatkan sudut *roll*, *pitch*, dan *yaw* dilakukan dengan memanfaatkan masukan dari sensor *accelerometer*, *gyroscope*, dan *magnetometer*.
3. Pengontrol PID dapat digunakan sebagai metode kontrol untuk menjaga keseimbangan *quadcopter* dengan cara mengontrol kecepatan motor *brushless* pada lengan-lengan *quadcopter*.
4. Proses mengontrol mode gerak secara manual pada *mobile robot* melalui *quadcopter* dapat dilakukan dengan memanfaatkan masukan dari *receiver remote RC* yang selanjutnya dihasilkan keluaran data berupa karakter *ASCII* yang dikirim oleh RF YS1020 oleh sistem pengontrol *quadcopter* untuk mengontrol *mobile robot*.

#### 5. SARAN

Pada penelitian ini masih terdapat banyak hal yang harus disempurnakan. Berikut ini disampaikan saran - saran untuk menyempurnakan penelitian dan sistem yang dibuat.

1. Penggunaan sensor akselerometer yang lebih sensitif dengan data yang stabil akan lebih memaksimalkan performa dari sistem ini.
2. Untuk meminimalkan *error* pada sistem *quadcopter* dapat ditambahkan peredam antara badan *quadcopter* dan komponen elektronik sistem

#### DAFTAR PUSTAKA

- [1] Ajie, A. K. B. 2007. *Penerapan Mikrokontroler AVR dalam Pembuatan UAV sebagai Sarana Fotografi Udara*. Jurusan Fisika, Fakultas MIPA, Universitas Diponegoro, Semarang.
- [2] Pradel, Gilbert dkk. 2007. *Modelling and Development of a quadrotor UAV*. Toulouse.
- [3] Luukkonen. 2011. *Modelling and control of quadcopter*. Aalto University. Espoo.
- [4] Pramerlani, William dan Paul B. 2009. *Direction Cosine Matrix IMU: Theory*. <http://gentlenav.googlecode.com/files/MahonyPapers.zip>. diakses 3 Maret 2012.
- [5] Kristiyana, Samuel. 2008. *Desain dan Sistem Pengendalian Robot Beroda Pemadam Api*. Jurusan Teknik Elektro, Institut Sains dan Teknologi AKPRIND, Yogyakarta