

Purwarupa Sistem Otomasi Terbang Landas dan Mendarat *Quadcopter*

Andi Dharmawan*¹, Irfan Nurudin Firdaus²

¹Jurusan Ilmu Komputer dan Elektronika, FMIPA, UGM, Yogyakarta

²Program Studi Elektronika dan Instrumentasi, JIKE, FMIPA, UGM, Yogyakarta

e-mail: *dharmawan.andi@gmail.com, irfan.elins@gmail.com

Abstrak

Telah dibuat sistem quadcopter yang dapat terbang dan mendarat otomatis dari satu titik ke titik lainnya menggunakan arduino mega sebagai kontrolernya. Sensor yang digunakan adalah sensor IMU GY-80 yang terdiri dari sensor accelerometer, gyroscope dan magnetometer. Keluaran dari sensor adalah angka dalam satuan derajat. Selain itu juga digunakan sensor ultrasonik untuk menentukan ketinggian dari quadcopter.

Sistem memiliki dua mode yaitu mode otomatis dan mode manual. Mode otomatis menggunakan data keluaran dari sensor IMU GY-80 yang akan diolah menggunakan PID controller sebagai proses pengatur keseimbangan. Quadcopter akan terbang pada ketinggian 1 meter dan bergerak maju lalu melakukan proses pendaratan. Untuk mempertahankan quadcopter berada pada ketinggian tertentu digunakan data dari hasil pembacaan sensor ultrasonik kemudian diolah menggunakan PID controller. Mode manual digunakan apabila pada mode otomatis terjadi masalah. Mode manual mendapat masukan dari remote kontrol.

Kata kunci—Quadcopter, PID, Otomatis

Abstract

Quadcopter system was created to fly and land automatically from one point to another using arduino mega as the controller. The sensor used is a GY-80 IMU sensor consists of a sensor accelerometer, gyroscope and magnetometer. The output of the sensor is a number in degrees. It is also used ultrasonic sensors to determine the height of the Quadcopter.

The system has two modes, namely automatic mode and manual mode. Automatic mode using the output data from the IMU sensor GY-80 that will be processed using the PID controller as the stabilizer. Quadcopter will fly at a height of 1 meter and move forward and make the process of landing. To maintain Quadcopter be a certain height to use the data from the ultrasonic sensor readings are then processed using a PID controller. Manual mode is used when there is a problem in the automatic mode. Manual mode gets input from the remote control.

Keywords—Quadcopter, PID, automatically

1. PENDAHULUAN

Dunia elektronika saat ini semakin berkembang. Perkembangan yang pesat ini juga berdampak pada perkembangan teknologi robotika yang berkaitan erat dengan dunia elektronika dan instrumentasi. Kebutuhan manusia akan alat bantu yang kuat, murah, dan efisien dapat disediakan oleh robot-robot yang dibuat untuk mendukung tugas-tugas yang umumnya dilakukan oleh manusia.

Salah satu jenis robot yang sedang dikembangkan adalah robot terbang. Robot terbang disebut pesawat tanpa awak atau (*Unmanned Aerial Vehicle*). Secara umum UAV (*Unmanned*

Aerial Vehicle) merupakan sebuah sistem pesawat tanpa awak yang memiliki kemampuan untuk melakukan berbagai jenis misi penginderaan jarak jauh berbasis video maupun foto/*still image*, baik untuk kegunaan sipil ataupun militer. Misi yang dapat dilakukan meliputi *surveillance, reconnaissance, monitoring*, patroli udara, foto udara resolusi tinggi dan lain sebagainya. UAV (*Unmanned Aerial Vehicle*) dikenal sebagai pesawat terbang tanpa awak atau dikenal juga dengan istilah UAS (*Unmanned Aircraft System*) di Amerika. UAV didefinisikan sebagai pesawat terbang tanpa pilot, menggunakan gaya aerodinamik untuk terbang, baik secara mandiri (otomatis) dengan bantuan *autopilot* atau dikemudikan jarak jauh dengan bantuan *remote control*, dan dapat membawa muatan senjata atau tidak.[1]

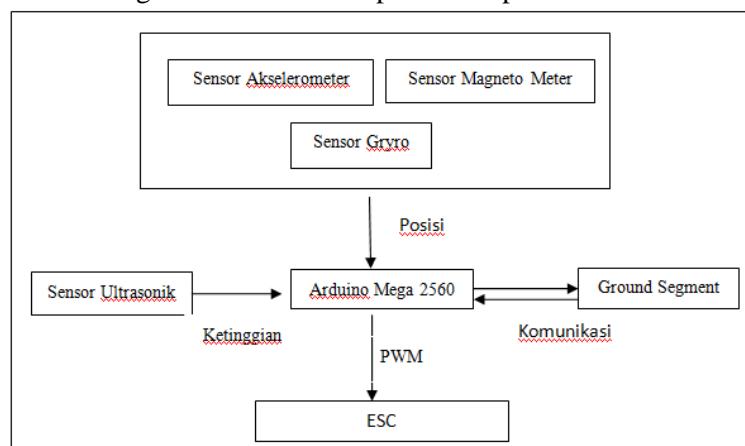
Salah satu jenis dari UAV (*Unmanned Aerial vehicle*) adalah *quadcopter* (atau kadang disebut *quadrotor*), merupakan *multicopter* yang memiliki empat rotor (balong-balong). *Quadcopter* ini dapat terbang dengan dua cara yaitu terbang dengan cara dikontrol dan terbang otomatis [2]. Namun demikian *quadcopter* yang dikembangkan di Indonesia masih banyak yang belum menggunakan sistem otomatis. Oleh karena itu, perlu juga dikembangkan sistem *quadcopter* yang dapat terbang secara otomatis tanpa harus dikendalikan.

2. METODE PENELITIAN

2.1 Analisis dan Perancangan Sistem

Rancangan dalam purwarupa sistem otomasi terbang landas dan mendarat *quadcopter* secara umum dibagi menjadi rancangan mekanik, rancangan perangkat keras (*hardware*) atau sistem elektronis yang berfungsi dalam kontrol *quadcopter*, rancangan perangkat lunak atau *software* dan rancangan kontrol keseimbangan *quadcopter*.

Sistem mekanik *quadcopter* dibuat sedemikian rupa sehingga sistem mekanik tidak mempengaruhi/menghambat kinerja terbang dari *quadcopter*. Rancangan perangkat keras merupakan sistem-sistem elektronis yang berperan dalam kontrol *quadcopter* yaitu terdiri dari sensor-sensor sebagai fungsi dari input, mikrokontroler sebagai pengolah data dan ESC (*Electronic Speed Controller*) sebagai fungsi keluaran. Dalam rancangan *software* dibagi menjadi dua bagian yaitu program pada arduino dan rancangan program untuk membuat *ground segment*. Kontrol *quadcopter* berfungsi untuk mengatur keseimbangan *quadcopter* dan untuk menjaga ketinggian *quadcopter* menggunakan kontrol PID (*PID controller*) dengan cara menggunakan konstanta PID yang sesuai, sehingga *quadcopter* dapat bekerja sesuai dengan yang kita inginkan. Rancangan secara umum dapat dilihat pada Gambar 1.

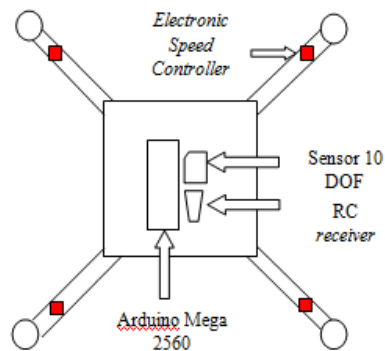


Gambar 1 Rancangan sistem keseluruhan

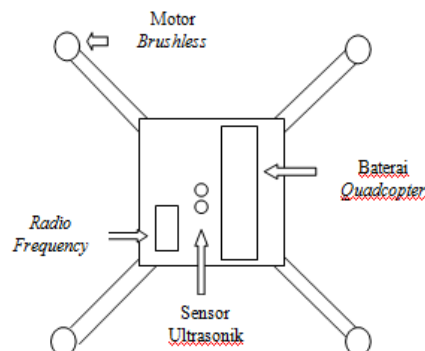
Gambar diatas menunjukkan hubungan antar masing-masing bagian dari sistem. Arduino mendapat data dari sensor akselerometer, gyro dan *magneto* sebagai data posisi dari *quadcopter*

serta data ketinggian dari sensor ultrasonik. Data-data tersebut diolah dan kemudian arduino mengirimkan sinyal PWM ke ESC untuk menggerakkan motor. Data-data yang diolah arduino juga dikirim ke *ground segment* agar dapat dilihat oleh *user*.

Quadcopter dirancang menggunakan bahan yang ringan yaitu aluminium sehingga akan didapatkan bobot yang minimal. Pemilihan bahan sangat penting karena mempengaruhi terbang dari *quadcopter* itu sendiri. Desain dari *quadcopter* itu sendiri yaitu menyerupai huruf X dengan masing-masing motor dan baling-baling di keempat sisinya. Perancangan mekanik secara utuh dapat dilihat pada Gambar 2 dan 3.



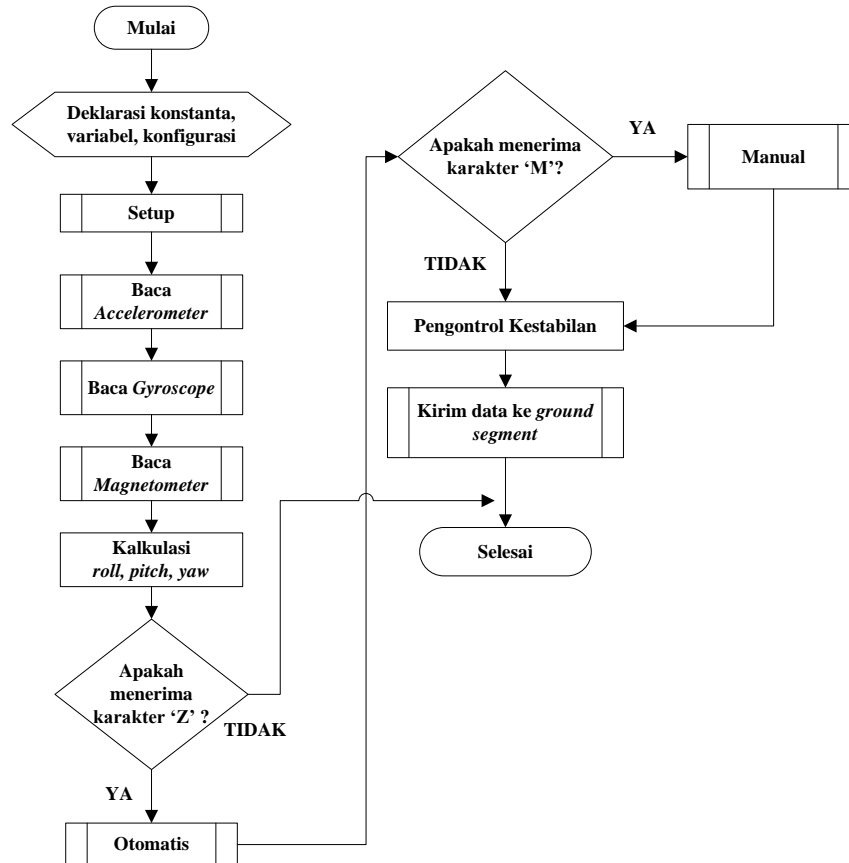
Gambar 2 Rancangan mekanik tampak atas



Gambar 3 Rancangan mekanik tampak bawah

Pada gambar 2 terlihat bahwa seluruh bagian elektronis *quadcopter* diletakkan ditengah, hal ini bertujuan agar pusat massa dari *quadcopter* berada ditengah badan, sehingga pemrograman keseimbangan dapat dilakukan lebih mudah. Gambar 3 menunjukkan bagian bawah *quadcopter*, dimana pada bagian bawah terdapat sensor jarak yaitu sensor ultrasonik untuk menentukan ketinggian *quadcopter*. Baterai dan modul RF (*Radio Frequency*) juga diletakkan pada bagian bawah *quadcopter*.

Perangkat lunak yang digunakan untuk menggunakan Arduino IDE (*Integrated Development Environment*) versi 1.0. Bahasa pemrograman yang digunakan arduino adalah bahasa pemrograman turunan C++ sehingga kebanyakan fungsi C dan C++ dapat dijalankan di arduino. Pemilihan bahasa pemrograman arduino sendiri karena arduino bersifat *open source* yang membuat banyak pengembang-pengembang membuat *library* sehingga memudahkan kita. Gambar 4 menunjukkan diagram alir cara kerja *quadcopter* secara umum.



Gambar 4 Diagram alir program utama

Dari diagram alir diatas dapat dilihat bahwa awal program dimulai dengan deklarasi variabel dll. Deklarasi tersebut menggunakan *library* standar yang sudah disediakan oleh arduino IDE seperti *wire.h* untuk mengakses jalur *i²c* pada arduino dan *servo.h* sebagai *library* yang digunakan untuk menginisialisasi motor *brushless*. Selain itu, dalam penelitian kali ini digunakan juga *library NewPing.h* yang digunakan untuk melakukan inisialisasi sensor ultrasonik sekaligus menjalankan sensor tersebut. Selanjutnya program akan berjalan ke dalam fungsi *setup* yang berfungsi untuk melakukan penyetingan awal *quadcopter*.

Setelah masuk program *setup*, program kemudian membaca keluaran dari sensor-sensor yaitu sensor *accelerometer*, sensor *gyroscope* dan sensor *magnetometer*. Setelah semua data selesai dibaca, program akan melakukan kalkulasi data-data tersebut menjadi sudut *roll*, *pitch* dan *yaw* kemudian jika ada penerimaan perintah untuk menjalankan program yaitu dengan adanya pengiriman karakter *Z* maka program akan masuk fungsi otomatis.

Setelah fungsi program otomatis dijalankan maka program akan melakukan kendali motor dengan cara mengirimkan sinyal PWM yang dapat memutar motor. Selama menjalankan program, data-data yang diolah akan dikirimkan ke *ground segment*. Dalam menjalankan program, jika ada karakter *M* yang diterima maka program akan masuk ke mode manual.

Perancangan *ground segment* dibuat untuk menampilkan data-data yang dikirimkan oleh *quadcopter* yang dapat dilihat oleh *user*. Antarmuka yang akan dibuat adalah tampilan data secara serial, tampilan data posisi *quadcopter* (posisi *x,y,z* dan ketinggian) baik secara data maupun visualisasi. Selain itu juga, *ground segment* berfungsi untuk mengirimkan perintah kepada *quadcopter* misalnya perintah untuk terbang atau perintah untuk berganti menjadi mode manual.

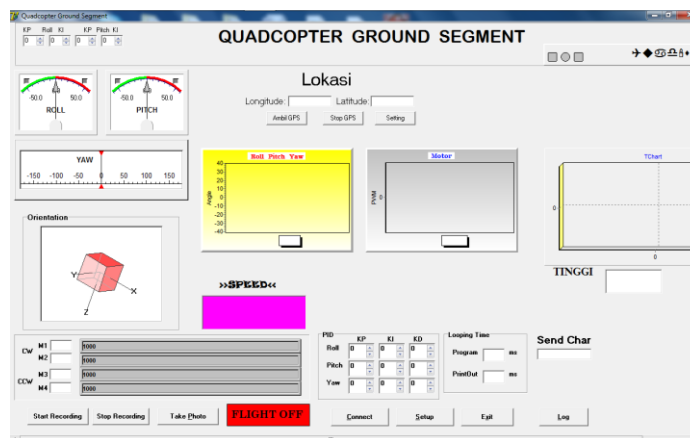
2.2 Implementasi Sistem

Quadcopter dibuat dengan massa 1,3 kg dan jari-jari 40 cm dari pusat. Material yang digunakan untuk rangka *quadcopter* adalah plat alumunium dengan ketebalan 1 cm. Pemilihan plat alumunium ini karena plat alumunium ukuran 1 cm cukup kuat untuk menopang beban dari *quadcopter* dan pat ini bersifat ringan sehingga tidak mempengaruhi terbang dari *quadcopter* itu sendiri. Pembuatan sistem mekanik ini secara umum masih bersifat manual seperti pemotongan, pembuatan lubang pada plat alumunium dan pengecatan. Sedangkan untuk menyambung plat alumunium digunakan teknik pengelasan yang khusus untuk mengelas alumunium. Gambar 5 menunjukkan bentuk mekanik dari *quadcopter* yang dibuat dalam penelitian ini.



Gambar 5 Implementasi mekanik

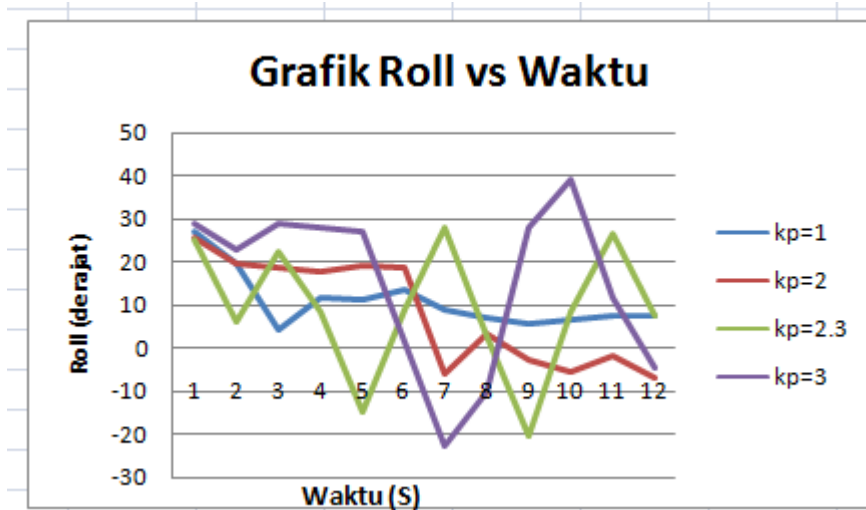
Antarmuka *ground segment* dibuat dengan menggunakan Delphi 7 untuk menampilkan informasi *roll*, *pitch*, *yaw* dan ketinggian dari *quadcopter*. *Ground segment* juga menampilkan visualisasi posisi dari *quadcopter* serta menampilkan juga informasi PID. Selain ditampilkan dalam bentuk data, data-data tersebut juga ada yang ditampilkan dalam bentuk grafik. Tidak hanya menerima data dari *quadcopter*, *ground segment* juga berfungsi untuk mengirimkan perintah ke *quadcopter*. Tampilan *ground segment* yang dibuat dapat dilihat pada gambar 6.



Gambar 6 Tampilan ground segment

3. HASIL DAN PEMBAHASAN

Pengujian pertama yang dilakukan adalah pengujian mencari nilai konstanta masing-masing sudut *roll*, *pitch* dan *yaw*. Nilai-nilai tersebut akan digunakan untuk melakukan kontrol keseimbangan terbang dari *quadcopter*. Pengujian dilakukan dengan tahap mencari nilai konstanta proporsional selanjutnya dengan nilai proporsional tersebut dicari nilai konstanta derivative selanjutnya dicari nilai konstanta integral. Berikut adalah hasil pengujian pada sudut *roll*.

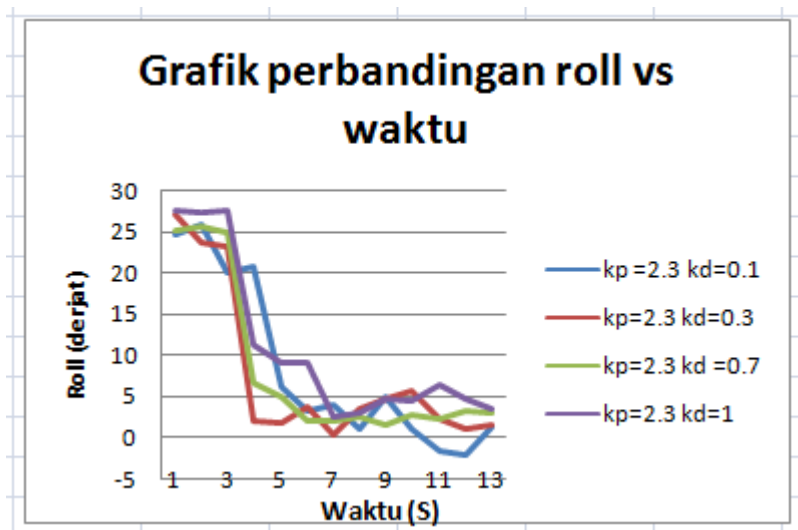


Gambar 7 grafik pengujian mencari konstanta proporsional pada sudut roll

Pada gambar terlihat bahwa nilai $K_p=1$ dan $K_p=2$ *quadcopter* belum mampu menstabilkan dirinya sendiri karena dengan nilai K_p tersebut motor belum mampu memberikan dorongan agar posisi *quadcopter* menjadi stabil. Nilai K_p 3,0 membuat *quadcopter* berosilasi sangat cepat pada titik 0° bahkan membuat *quadcopter* lepas kendali karena osilasinya yang makin lama makin besar. Berdasarkan hasil pengujian tersebut nilai K_p yang tepat adalah 2,3 sebab sesuai dengan grafik tersebut *quadcopter* mampu bergerak dengan cepat menuju titik 0° namun tidak menimbulkan osilasi yang berlebihan.

Konstanta proporsional berfungsi untuk membuat sistem *overshoot* yang berfungsi agar *quadcopter* dapat bergerak menuju titik set point. Namun demikian pemberian nilai yang terlalu besar akan membuat *overshoot* semakin besar pula. Untuk mengatasi *overshoot* perlu digunakan konstanta lain untuk mengimbangi proporsional sehingga *overshoot* yang terjadi dapat diredam. Konstanta yang dapat melakukan peredaman *overshoot* tersebut adalah konstanta derivative [3].

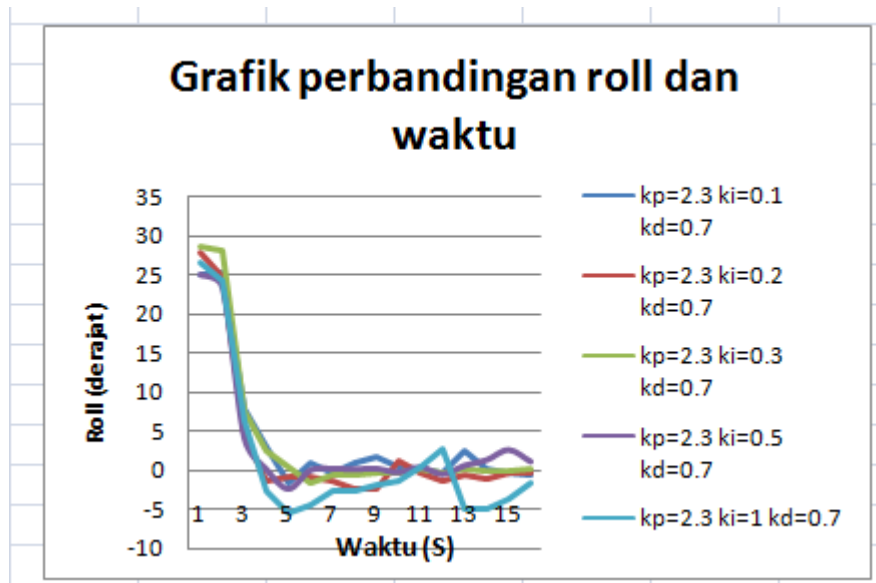
Pengujian mencari konstanta derivatif dilakukan untuk meredam *overshoot* yang terjadi akibat pengaruh konstanta *proporsional*. Pengujian dilakukan dengan melakukan variasi nilai kd dan hasil dari pengujian ditunjukkan pada Gambar 8.



Gambar 8 hasil pengujian mencari konstanta derivative

Berdasarkan gambar 8 terlihat bahwa untuk nilai $K_d=1$ belum dapat meredam osilasi yang dihasilkan oleh proporsional. Pada saat nilai $k_d=0.3$ *overshoot* yang terjadi masih belum dapat diredam secara sempurna dan masih terdapat *overshoot* yang cukup besar. Ketika diberikan nilai $K_d=0.7$ terlihat bahwa *overshoot* yang dihasilkan oleh proporsional sudah mampu diredam dengan penyimpangan sudut rata-rata kecil yaitu sekitar kurang dari 5^0 . Untuk $K_d=1$ tampak bahwa efek redaman yang diberikan terlalu besar yang akan membuat pergerakan *quadcopter* menjadi tersendat. Berdasarkan grafik tersebut, maka dapat dikatakan bahwa K_d yang cocok digunakan dalam sistem ini adalah dengan nilai 0,7.

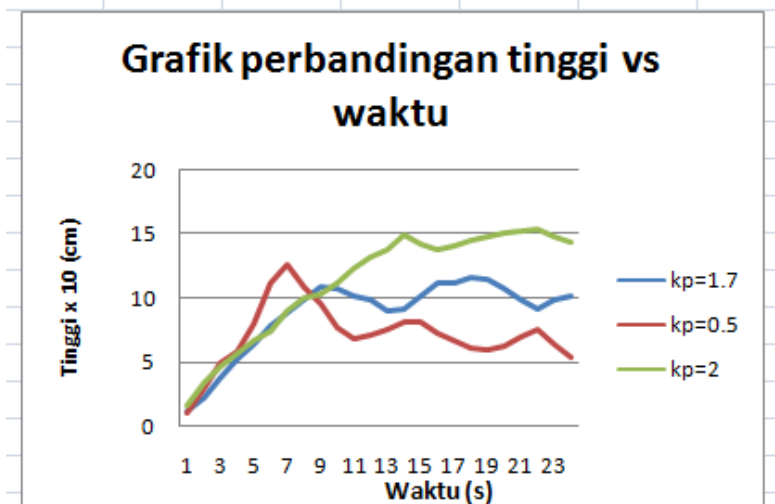
Pengujian berikutnya adalah mencari konstanta integral dan hasilnya ditunjukkan pada Gambar 9.



Gambar 9 Pengujian mencari konstanta integral sudut *roll*

Pada Gambar 9 terlihat ketika nilai K_i diberikan 0.1 grafik masih terjadi osilasi. Ketika nilai K_i 0,2 osilasi yang terjadi sudah berkurang dan grafik berada pada posisi set point. Ketika nilai diberikan nilai K_i sebesar 0,3 grafik menunjukkan pergerakan *quadcopter* yang sudah berada pada daerah set point dengan tingkat osilasi yang relatif kecil. Ketika nilai K_i diberikan lebih besar, pergerakan *quadcopter* kembali terjadi osilasi, hal tersebut terlihat pada gambar yakni ketika diberikan nilai K_i 0,5 dan 1. Dari data tersebut dapat dikatakan bahwa nilai K_i yang sesuai pada sistem ini adalah 0,3. Hal yang sama dilakukan untuk sudut *pitch* dan *yaw*.

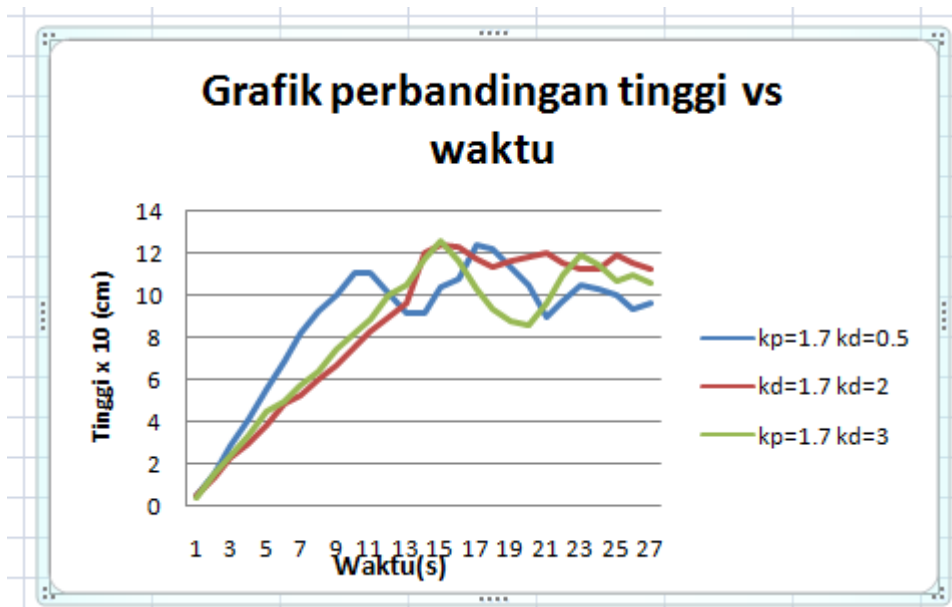
Untuk dapat menjaga ketinggian *quadcopter* pada ketinggian tertentu diperlukan juga pengontrol PID untuk menjaga ketinggian. Kontrol ini mendapatkan masukan dari sensor ultrasonic yang dipasang pada bagian bawah *quadcopter*. Pengujian dilakukan dengan cara mencari konstanta *proporsional*, *derivatif* dan *integral* untuk pengatur ketinggian. Pengujian dilakukan dengan cara memvariasikan nilai konstanta *proporsional*, *derivatif* dan *integral*.



Gambar 10 pengujian mencari konstanta proporsional

Pada saat pengujian terlihat pada grafik bahwa ketika pemberian nilai K_p 0,5 *quadcopter* sempat naik melebihi set point yaitu pada 100 cm akan tetapi setelah itu *quadcopter* bergerak turun dibawah set point dan cenderung berosilasi pada nilai dibawah set point. Hal ini menandakan bahwa nilai tersebut belum mampu mengangkat *quadcopter* pada set point yang diinginkan. Pada saat diberikan nilai 1,7 pergerakan *quadcopter* terlihat berosilasi di daerah dekat set point dengan jarak osilasi yang kecil. Ketika nilai K_p yang diberikan 2, *quadcopter* terus bergerak naik melebihi set point. Dari data tersebut dapat dikatakan bahwa nilai K_p yang sesuai adalah 1,7.

Pengujian berikutnya adalah mencari konstanta *derivative* dengan menggunakan konstanta *proporsional* yang tetap. Hasil pengujian ditunjukkan pada Gambar 11.

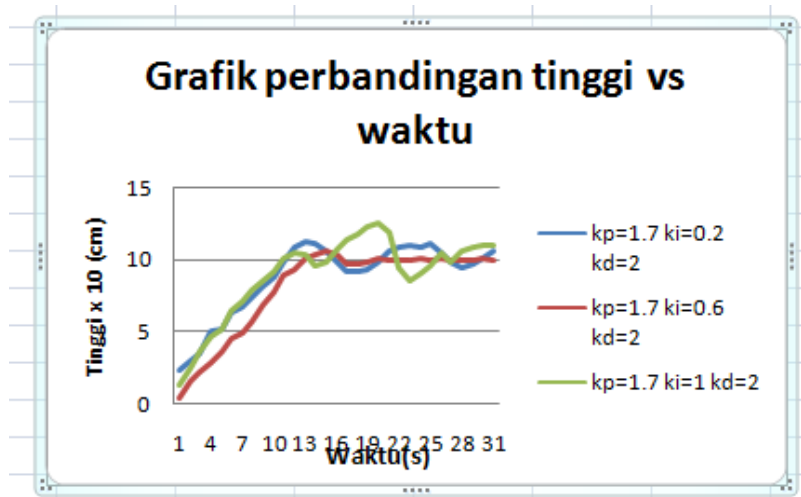


Gambar 11 hasil pengujian mencari konstanta derivatif

Pada pengujian mencari nilai konstanta derivatif terlihat pada grafik ketika diberikan nilai K_d 0,2 masih terjadi lonjakan osilasi yang menandakan pergerakan *quadcopter* masih naik turun. Saat nilai K_d diberikan 2 *quadcopter* cenderung bergerak stabil pada titik tertentu dengan nilai osilasi yang kecil. Pergerakan *quadcopter* kembali mengalami osilasi ketika nilai K_d yang

diberikan 3 atau lebih besar dari itu. Dari data tersebut dapat disimpulkan bahwa nilai K_d yang digunakan adalah 2 yang mampu membuat *quadcopter* bergerak stabil dititik tertentu namun masih di atas titik set point. Oleh karena itu diperlukan konstanta lain yaitu konstanta derivatif untuk memperbaiki pergerakan *quadcopter* sehingga berada di garis set point.

Konstanta integral diberikan dengan tujuan untuk memperbaiki pergerakan *quadcopter* agar berada di daerah *set point* yaitu pada jarak 100 cm. Pengujian mencari konstanta integral dilakukan dengan cara memberikan variasi nilai integral dimana nilai K_p dan K_d diset tetap. Hasil pengujian mencari konstanta integral ditunjukkan pada Gambar 12.



Gambar 12 Hasil pengujian mencari konstanta *integral*

Ketika diberikan nilai K_i 0,2 terlihat pada grafik bahwa *quadcopter* sudah bergerak didaerah set point akan tetapi masih sering terjadi osilasi didaerah tersebut. Saat nilai K_i diberikan 0,6 osilasi yang terjadi ketika diberikan nilai K_i 0,2 sudah dapat diredam dan *quadcopter* bergerak didaerah *set point*. Pergerakan *quadcopter* kembali mengalami osilasi ketika nilai K_i yang diberikan diatas 0,6 dengan kata lain nilai K_i yang sesuai digunakan sistem ini adalah 0,6.

4. KESIMPULAN

Dari hasil pengamatan, pengujian dan analisis pada hasil perancangan dapat diperoleh beberapa kesimpulan sebagai berikut :

1. Quadcopter mampu terbang dan mendarat secara otomatis dari satu titik ke titik lainnya.
2. Deteksi kemiringan sudut *quadcopter* dilakukukan dengan menggunakan sensor *gyroscope* L3G4200D, *accelerometer* ADXL345 dan *magnetometer* HMC5883L.
3. Terdapat 4 buah pengontrol PID yakni pengontrol PID *roll*, pengontrol PID *pitch*, pengontrol PID *yaw* dan pengontrol PID ketinggian.
4. Konstanta proporsional, derivatif dan integral yang digunakan pada:

Pengontrol PID <i>roll</i>	: $K_p=2.3$, $K_d=0.3$ dan $K_i=0.7$
Pengontrol PID <i>pitch</i>	: $K_p=2.3$, $K_d=0.2$ dan $K_i=0.7$
Pengontrol PID <i>yaw</i>	: $K_p=2.8$, $K_d=0.4$ dan $K_i=1$
Pengontrol PID ketinggian	: $K_p=1.7$, $K_d=0.6$ dan $K_i=2.0$

5. SARAN

Pada penelitian ini masih terdapat beberapa hal yang perlu disempurnakan. Berikut saran yang disampaikan untuk penelitian-penelitian selanjutnya yang sejenis.

1. Untuk pengembangan lebih lanjut, sebaiknya digunakan GPS untuk *quadcopter* otomatis ini.
2. Perlu dilakukan penelitian lebih lanjut mengenai algoritma PID adaptif supaya dapat meningkatkan stabilitas terbang *quadcopter*.
3. Perlu dilakukan penelitian lebih lanjut mengenai penggunaan Sistem Operasi untuk dapat meningkatkan kinerja *quadcopter*

DAFTAR PUSTAKA

- [1] Ajie, A.K.B., 2007. *Penerapan Mikrokontroler AVR dalam Pembuatan UAV Sebagai Sarana Fotografi Udara*. Jurusan Fisika, Fakultas Mipa, UNDIP, Semarang.
- [2] Hoffman, M., 2004. DIY Drones at Home. Tersedia di URL: <http://www.DIYdrones.com>. diakses tanggal 13 desember 2011.
- [3] Dorf, R.C., dan H. Bishop, R., 2010, *Modern Control System*, 12th Edition, Prentice Hall International, United Kingdom.