

Pembuatan Model *Quadcopter* yang Dapat Mempertahankan Ketinggian Tertentu

Wili Kumara Juang, Lauw Lim Un Tung
Program Studi Teknik Elektro, Universitas Kristen Petra
Jl. Siwalankerto 121-131, Surabaya 60236, Indonesia
E-Mail: wilikumara@hotmail.com; tung@peter.petra.ac.id

Abstrak - *Quadcopter* adalah salah satu topik pembelajaran yang akan dilakukan oleh Laboratorium Elektronika Universitas Kristen Petra. Karakteristik kestabilan terbang pada *Quadcopter* serta pertahanan terbang yang dilakukannya menjadi bahan ajar yang baik saat ini.

Quadcopter menggunakan *PID controller* untuk melakukan keseimbangan terbang melalui pembacaan sudut oleh *sensor MPU6050*. Terdapat *sensor SRF02* sebagai pendeteksi ketinggian yang juga menggunakan *PID controller* dalam mempertahankan ketinggian. Selain itu *Quadcopter* juga menggunakan *remote control* untuk keamanan operasional.

Melalui pengujian *MPU6050* diperoleh *error* sudut oleh sumbu putar *x (roll)* yaitu 2° dan *y (pitch)* yaitu -3° yang dirata-rata dengan jumlah data dari *take off*, menjaga ketinggian, hingga *landing*. Pertahanan terbang mengalami osilasi namun tetap berada pada batas bawah toleransi. Pertambahan nilai *RPM propeller* dari kisaran 2000 *RPM* hingga 10.000 *RPM*.

Kata kunci :

Arduino, MPU6050, PID controller, Quadcopter, SRF02.

I. PENDAHULUAN

Quadcopter dengan istilah formal *unmanned aerial vehicle (UAV)* merupakan salah satu dari berbagai macam robot terbang yang telah dikembangkan oleh berbagai pihak. Sebuah *Quadcopter* juga disebut helikopter *quadmotor* yang merupakan helikopter yang diangkat dan didorong oleh empat motor.

Quadcopter membutuhkan kecepatan dan torsi yang sama pada masing-masing motor untuk memperoleh keseimbangan terbang. Akibat aerodinamika, keseimbangan terbang tidak didapatkan dengan cara yang sederhana, sehingga pada prosesnya dibutuhkan pendeteksi kemiringan sistem. Pengatur keseimbangan ini diperoleh dari pengontrolan perubahan sudut dari sumbu putar *x* dan *y* sedangkan sumbu putar *z* hanya menyesuaikan.

II. METODE PENELITIAN

Metodologi penelitian dan perancangan yang digunakan selama mengerjakan tugas akhir yaitu:

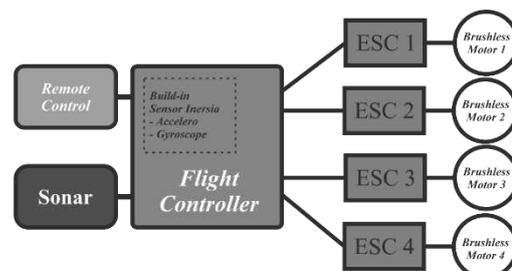
- Desain sistem terbang *Quadcopter*:
Mengumpulkan referensi untuk membuat *Quadcopter*, baik secara *hardware* maupun *software*.
- Pengujian *Quadcopter*:
 - Pengujian pembacaan sudut oleh *sensor MPU6050* dengan alat ukur busur yang bertujuan untuk memastikan kesesuaian besaran sudut yang bentuk oleh kemiringan terbang *Quadcopter*.

- Pengujian pembacaan *sensor* sonar SRF02 dengan alat ukur panjang yang bertujuan untuk memastikan kesesuaian jarak pembacaan *sensor* dan panjang aktual menggunakan alat ukur panjang (meteran).
 - Pengujian perilaku konstanta *PID controller* pada sumbu putar *x (roll)* dan *y (pitch)* terhadap kemiringan sudut dan keempat *brushless motor* untuk memperoleh grafik *PID controller* yang dibentuk.
 - Pengujian perilaku sinyal *brushless motor* dengan kecepatan putar baling-baling menggunakan *tachometer* untuk mengetahui rotasi per menit (*RPM*) pada masing-masing motor untuk dengan kecepatan yang ditentukan.
 - Pengujian terbang *Quadcopter*. Pengujian ini dilakukan untuk memperoleh data secara keseluruhan pada saat terbang, yang berguna untuk menyesuaikan dengan data uji yang sebelumnya. Pengujian ini dilakukan untuk mengetahui pertahanan ketinggian yang dibaca oleh SRF02.
3. Penyajian Laporan:
Hasil pengujian disajikan dalam bentuk laporan tugas akhir yang pada setiap tahap pembuatannya akan diperiksa oleh dosen pembimbing.

III. PERANCANGAN SISTEM

A. Gambaran Umum Sistem

Sistem yang berkerja pada penerbangan *Quadcopter* melingkup *input*, proses dan *output* yang bekerja bersamaan saat penerbangan berlangsung. *Input* pada sistem penerbangan *Quadcopter* adalah *sensor* inersia yaitu *MPU6050*, *remote control* 6 channel, dan *sensor* sonar SRF02. Pada proses terdapat *microcontroller* Arduino Pro Micro sedangkan pada *output* terdapat 4 *ESC* yang terhubung masing-masing pada *brushless motor*.

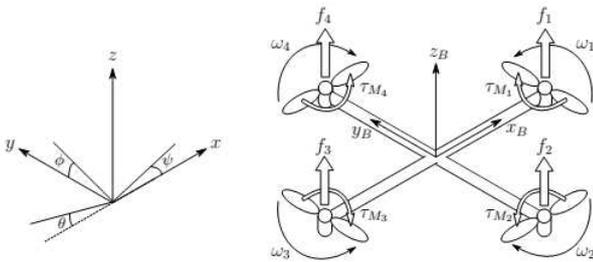


Gambar. 1. Blok Diagram Sistem

Input bertugas untuk memberikan masukan yang selanjutnya diproses oleh *microcontroller*. Data kemiringan sudut diberikan oleh *MPU6050*, data sinyal *PWM* yang diberikan oleh *remote control*, dan data

ketinggian yang diberikan oleh SFR02 yang kemudian diproses untuk melakukan suatu kerja.

Output diberikan perintah oleh *microprocessor* setelah membaca semua kondisi pada *input*. Sinyal yang diberikan pada ESC kemudian dieksekusi oleh *brushless motor* yang digunakan untuk memperoleh aerodinamika pada putaran *propeller*. Sinyal tersebut berupa sinyal PWM yang diterima oleh masing-masing ESC untuk menghasilkan 3 fasa sinyal pada *input brushless motor*.



Gambar. 2. Cara Kerja
Sumber: (Luukkonen, 2011)

Arah putar penghasil gaya 1 (f_1) dan gaya 3 (f_3) yaitu berlawanan arah putar jarum jam sedangkan arah putar penghasil gaya 2 (f_2) dan gaya 4 (f_4) searah putar jarum jam seperti pada gambar. Keempat gaya pada motor penggerak *propeller* tersebut menghasilkan gaya yang melawan arah gravitasi atau dengan kata lain gaya angkat. Salah satu metode untuk melakukan keseimbangan terbang adalah dengan menambahkan PID *controller* pada sistem. Rumusan sederhana untuk kesetabilan/keseimbangan terbang ini dapat dituliskan sebagai berikut:

$$e(t) = x_d(t) - x(t) \quad (1)$$

$$u(t) = K_p e(t) + K_I \int_0^t e(\tau) d\tau + K_D \frac{de(t)}{dt} \quad (2)$$

Dimana $u(t)$ merupakan *control input*, $e(t)$ adalah selisih antara kondisi yang diinginkan atau $x_d(t)$ dan kondisi sekarang atau $x(t)$. Nilai K_p , K_I , dan K_D merupakan konstanta elemen *proportional*, *integral* dan *derivative controller* itu sendiri [1].

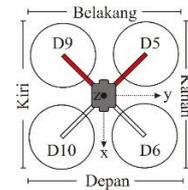
Masing-masing konstanta mengubah efektivitas stabilitas pada sistem. Umumnya ada tiga PID *loop* dengan konstanta PID masing-masing sumbu putar, sehingga harus mengatur nilai konstanta P, konstanta I dan konstanta D:

- *Proportional Control (P)* mempunyai efek untuk mengurangi *rise time* dan juga mempunyai efek untuk mengurangi *steady state error*.
- *Integral Control (I)* mempunyai efek untuk menghilangkan *steady state error* tetapi membuat *response transient* semakin buruk.
- *Derivative Control (D)* mempunyai efek untuk menambah kestabilan sistem, mengurangi *overshoot* dan memperbaiki *respon transient*.

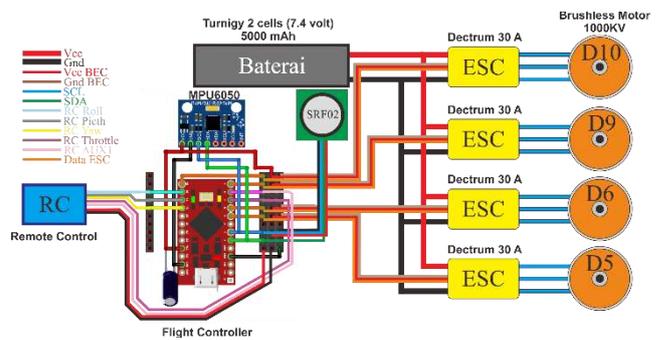
B. Perancangan Komponen Elektronika

Hubungan antar komponen dilakukan dengan menghubungkan *pin* pada *flight controller output* ESC yang terhubung pada *brushless motor* menggunakan *pin* PWM pada *pin* D10, D9, D6 dan D5. *Brushless motor*

depan kanan terhubung pada *pin* D10, *brushless motor* belakang kanan pada *pin* D9, *pin* D6 untuk *brushless motor* depan kiri dan *pin* D5 dihubungkan untuk *brushless motor* belakang kiri.



Gambar. 3. Konfigurasi Pin Brushless Motor



Gambar. 4. Rangkaian Elektronika



Gambar. 5. Remote Control

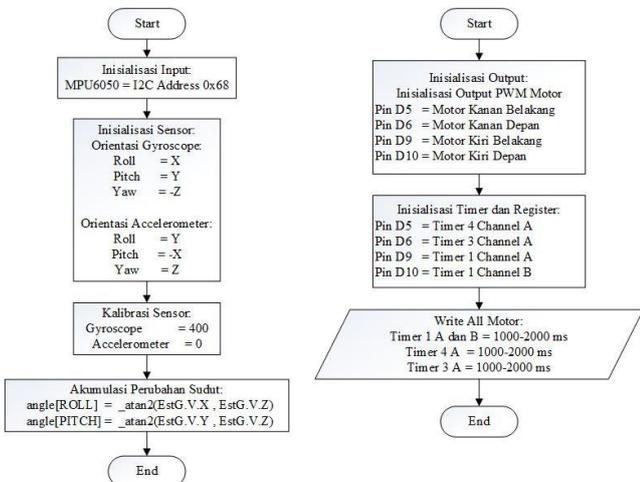
C. Desain Software

Penyelesaian *Quadcopter* ini tidak seutuhnya dibangun dari nol, karena ada pihak pengembang *flight controller* telah menyediakan kemudahan bagi setiap orang yang ingin belajar sistem penerbangan pada *Quadcopter*. Salah satu pengembang bernama Alexander Dubus menerbitkan *software* yang digunakan untuk model *multirotor* yang disebut dengan Multiwii [2]. Berikut tahapan *download firmware Flight Controller* serta pengaturan pada Arduino IDE:

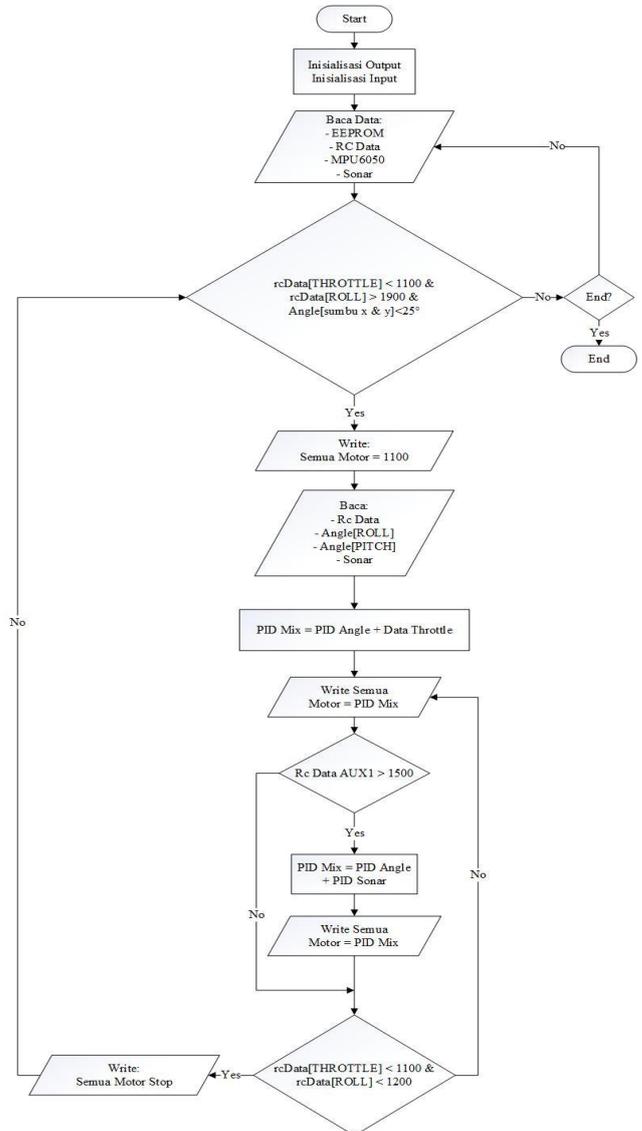
- Pastikan PC sudah ter-*install* Arduino IDE, atau sediakan dengan men-*download software* Arduino IDE pada <http://arduino.cc/en/main/software>.
- *Firmware flight controller* dapat diperoleh dengan membuka *link* dengan URL (*Uniform Resource Identifier*): <https://code.google.com/p/multiwii/>
- MultiWii adalah perusahaan pengembang *flight controller*. Versi yang digunakan pada pembuatan *Quadcopter* ini adalah 2.1, karena versi sebelumnya tidak menyediakan konfigurasi *board flight controller*.
- Buka hasil *download* dengan mengekstraksi semua data yang ada pada *compressed folder*.
- Buka *folder* yang telah diekstraksi, kemudian klik pada *folder* MultiWii_2_1 dan akhiri dengan membuka *sketch* Arduino dengan nama *file* MultiWii_2_1.
- Tahap berikutnya adalah melakukan pengaturan pada Arduino IDE dengan klik *Tools, setting board* dengan memilih *board* Leonardo karena konfigurasi *board flight controller* telah disesuaikan dengan penggunaan *board* Arduino Leonardo.

- Berikutnya adalah mengatur *port* yang digunakan oleh *flight controller* dengan cara melakukan pengaturan pada *Tools*, kemudian arahkan pada *port*, dan pilih *port* yang bertuliskan *port COM* (Leonardo).
- Tahap berikutnya adalah melakukan inisialisasi kode program untuk memperoleh kesesuaian dengan *flight controller* yang telah dibuat.

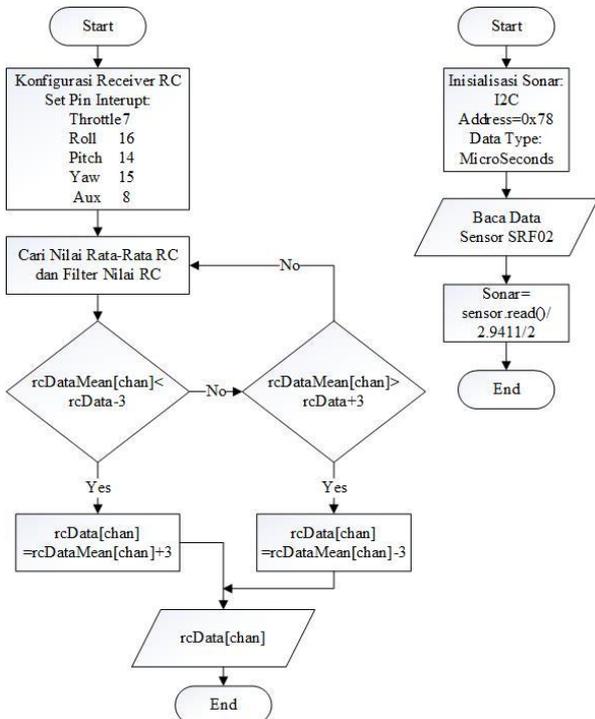
Garis besar sistem yang digunakan pada konfigurasi model *Quadx* dengan menggunakan *board Nanowii* ini dapat dipahami dengan mencari setiap bagian yang didefinisikan sebelumnya. Untuk memperoleh gambaran besar sistem, tahapan yang dilakukan adalah melakukan penelusuran pada setiap bagian kode program yang terdapat pada *tab Multiwii_2_1 (tab awal)*. Caranya adalah dengan melakukan penelusuran pada `void setup()` dan `void loop()`.



Gambar. 6. Flowchart Input MPU6050 dan Output PWM



Gambar. 8. Flowchart Sistem Keseluruhan



Gambar. 7. Flowchart Receiver RC dan Sonar

D. Perancangan Komponen Elektronik dan Mekanik

Model mekanik menggunakan *frame kit* milik Tarot FY450 harus dirakit terlebih dahulu hingga menjadi seperti pada gambar di bawah ini.

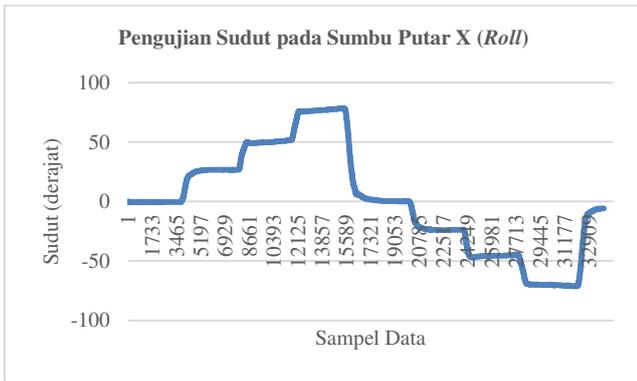


Gambar. 9. Quadcopter Keseluruhan

IV. PENGUJIAN SISTEM

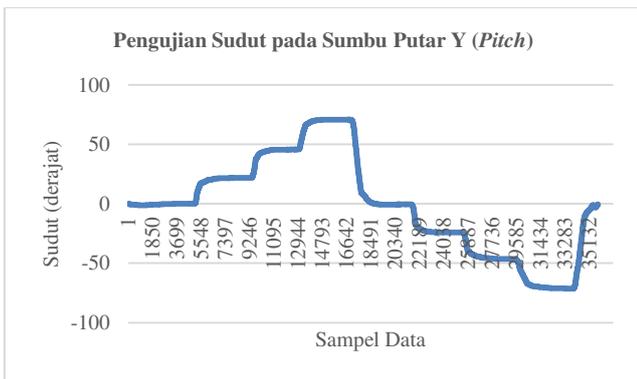
A. Pengujian Software

- Pembacaan Sudut Pada Sumbu Putar X dan Y
 Pengujian dilakukan untuk menguji kesesuaian sumbu pembacaan oleh MPU6050 dengan pembanding berupa alat ukur sudut.



Gambar 10. Grafik Pembacaan Sudut *Sensor* MPU6050 pada Sumbu Putar X

Arah putar pengujian pertama yaitu berlawanan dengan arah jarum. Pada sudut kemiringan 0°, *sensor* menunjukkan pembacaan dengan nilai rata-rata sebesar -0.4°. Kemiringan 20° diperoleh nilai rata-rata sebesar 25.9°. Pada pembacaan 40° *sensor* menunjukkan nilai rata-rata pembacaan 48.8°. Pengujian dengan kemiringan 60° pembacaan *sensor* menunjukkan nilai rata-rata sebesar 76.2°. Pembacaan *sensor* pada kemiringan -20° memiliki nilai rata-rata -21.6°. Kemiringan -40° *sensor* menunjukkan data dengan rata-rata nilai -45.7°. Pada pengujian kemiringan -60° pengamatan busur, *sensor* menunjukkan pembacaan dengan rata-rata nilai kemiringan sebesar -69.7°.



Gambar 11. Grafik Pembacaan Sudut *Sensor* MPU6050 pada Sumbu Putar Y

Pembacaan *sensor* 0° pada grafik menunjukkan kecocokan dengan nilai sesungguhnya pada pembacaan busur. Nilai rata-rata yang diperoleh pada pembacaan *sensor* sudut 0° adalah -0.4°. Pembacaan busur pada sudut 20° diperoleh pembacaan *sensor* dengan rata-rata 21.4°. Pada pembacaan *sensor* untuk kemiringan 40° *sensor* menunjukkan nilai dengan rata-rata data yang diperoleh sebesar 44.9°. Hasil pembacaan *sensor* untuk kemiringan 60° diperoleh rata-rata 69.7°. Pengamatan busur pada sudut -20° diperoleh pembacaan *sensor* dengan rata-rata -23.5°. Pembacaan *sensor* pada pengukuran -40° menghasilkan nilai rata-rata -45.4°. Pada sudut -60° pembacaan *sensor* menunjukkan nilai dengan rata-rata -69.5°.

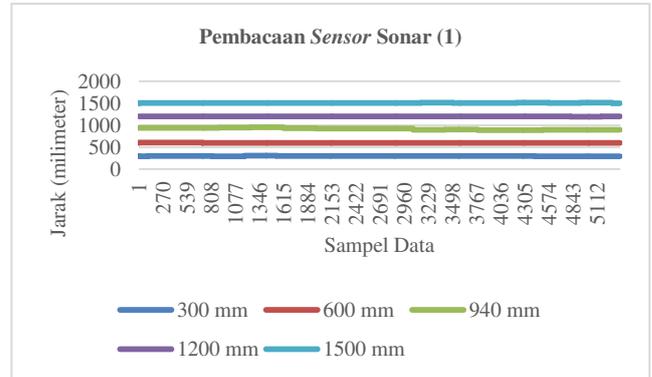
• Pengujian Pembacaan Jarak Ketinggian

Pengujian ini dilakukan untuk memastikan kesamaan ukur pembacaan *sensor* dengan pembanding alat ukur panjang (meteran). Pengambilan data dilakukan dengan metode seperti pada gambar di bawah ini:



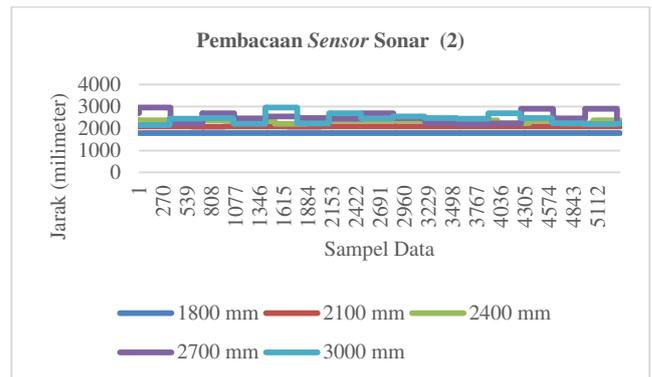
Gambar 12. Metode Pengujian Jarak Bacaan

Pembacaan nilai panjang oleh meteran diperoleh dengan cara menggeser *Quadcopter* pada jarak yang ditentukan.



Gambar 13. Pembacaan *Sensor* SRF02 Tahap 1

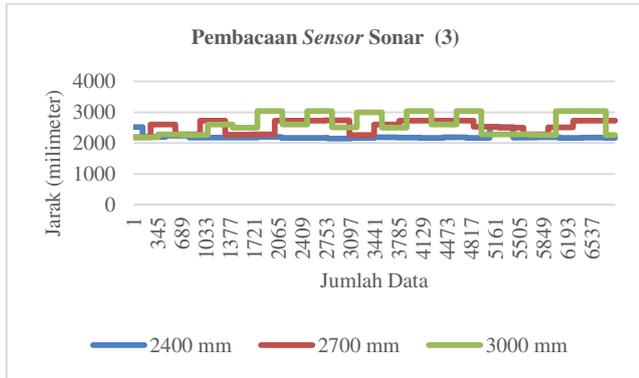
Pada jarak uji 300 mm, pembacaan *sensor* SRF02 memperlihatkan kestabilan data pada kisaran yang sama dengan rata-rata nilai 298 mm. Pada jarak uji 600 mm, diperoleh nilai rata-rata yang sesuai yaitu 600 mm yang terlihat jelas dari garis warna merah. Pada jarak 900 mm, pembacaan *sensor* mengalami penurunan yang cukup signifikan pada akhir data, namun memiliki rata-rata nilai yang cukup baik yaitu 919 mm. Jarak 1200 mm *sensor* menunjukkan konsistensi data yang cukup baik dengan nilai pembacaan rata-rata yang cocok yaitu 1200 mm. Jarak pengujian 1500 mm juga memiliki hasil yang cukup baik dengan nilai rata-rata 1508 mm.



Gambar 14. Pembacaan *Sensor* SRF02 Tahap 2

Pembacaan *sensor* tahap dua ini memiliki perbedaan dengan pembacaan pada tahap satu. Pada pembacaan 1800 mm dan 2100 mm memperlihatkan grafik yang cocok dengan pembacaan nilai rata-rata 1799 mm dan 2094 mm. Berbeda halnya dengan pengujian pada jarak ukur 2400 mm, 2700 mm dan 3000 mm, terdapat perubahan data yang cukup signifikan dan mempengaruhi bentuk grafik. Pada pembacaan *sensor* jarak 2400 mm, nilai rata-rata yang diperoleh sebesar 2344 mm, pada jarak 2700 mm

pembacaan *sensor* menunjukkan nilai rata-rata 2527 mm, dan pada jarak 3000 mm menunjukkan nilai rata-rata 2448 mm. Untuk memastikan pembacaan yang tidak sesuai karena data yang tidak stabil dilakukan pengujian ulang sebagai berikut:

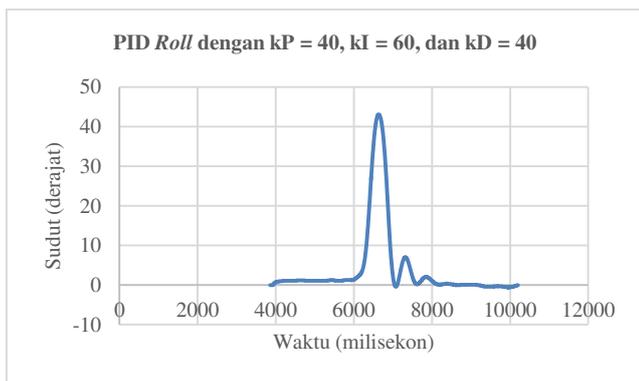


Gambar. 15. Pembacaan *Sensor* SRF02 Tahap 3

Pembacaan pada jarak 2400 mm diperoleh nilai rata-rata 2202 mm, pembacaan *sensor* pada jarak 2546 mm, sedangkan pada jarak 3000 mm diperoleh pembacaan rata-rata 2629 mm. Bentuk sinyal seperti ini tidak diijinkan pada sistem penerbangan, mengingat penerbangan ini membutuhkan kepresisian dan kestabilan. Pengujian terbang hanya dapat dilakukan pada *range* jarak minimal 150 mm hingga 2100 mm saja.

• Pengujian Konstanta *PID Controller*

Tujuan pengujian ini adalah memberikan keseimbangan terbang pada *Quadcopter* yaitu dengan mengembalikan posisi *Quadcopter* pada kemiringan 0° dengan menggunakan *PID controller*. Setelah dilakukan pengujian dengan beberapa kombinasi konstanta *PID*, ditentukan kombinasi konstanta *PID* terbaik yang tergambar pada grafik di bawah ini:



Gambar. 16. *PID Roll* dengan $k_P = 40$, $k_I = 60$, dan $k_D = 40$

Berdasarkan grafik, kombinasi konstanta *PID* ini menjadi kombinasi *PID* yang akan diimplementasikan langsung pada penerbangan. Kombinasi konstanta *PID* ini menunjukkan respon yang terbaik dari kombinasi konstanta sebelumnya. *PID controller* ini memiliki waktu untuk mencapai *steady state error* yang paling singkat sekitar 1000 milisekon meskipun dengan gangguan kemiringan mencapai 40° lebih.

Dalam menentukan konstanta *PID controller* pada sumbu putar *x*, langkah yang dilakukan sama seperti saat menentukan kombinasi *PID controller* pada sumbu putar *x*. Sehingga saat dilakukan penentuan nilai konstanta *PID*

dapat langsung menggunakan nilai konstanta yang juga digunakan pada sumbu putar *x*.



Gambar. 17. *PID Roll* dengan $k_P = 40$, $k_I = 60$, dan $k_D = 40$

Pada grafik *PID* ini, terlihat masih banyak amplitudo yang terbentuk. Pada pengujian ini, kombinasi konstanta yang diberikan kurang memberikan *PID* yang baik pada sistem. Kombinasi konstanta ini memiliki *settling time* yang masih terhitung lambat. Jumlah amplitudo dapat diredam dengan menambahkan nilai k_D . Pengujian pada Gambar 16 merupakan hasil dari penambahan k_D menjadi bernilai 60.



Gambar. 18. *PID Roll* dengan $k_P = 40$, $k_I = 60$, dan $k_D = 40$

Penentuan nilai konstanta *PID* ini tidak belum dikatakan berhasil jika belum dilakukan pengujian terbang. Harus dilakukan pengujian terbang untuk membuktikan kestabilannya.

B. Pengujian *Hardware*

Pengujian ini dilakukan untuk memperoleh perbandingan nilai kecepatan pada *controller* dengan pembacaan kecepatan RPM *propeller* menggunakan *tachometer*. Penentuan nilai perbandingan tersebut bertujuan untuk memprediksi kecepatan aktual *propeller* dalam RPM saat terbang.

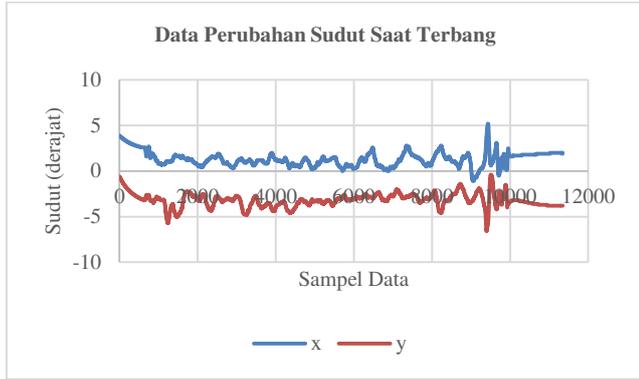
Tabel. 1. Nilai Rata-rata Pembacaan RPM *Propeller*

Variabel Uji	Propeller			
	D10	D9	D6	D5
1100	2639	2708	2864	2856
1150	3524	3460	3785	3660
1200	4314	4366	4625	4489
1250	4874	4861	5209	4946
1300	5481	5450	5792	5719
1350	5922	5916	6213	5990
1400	6746	6614	7009	6693

Variabel uji pada aplikasi *Quadcopter* seperti yang dijelaskan sebelumnya memiliki *range* dari 1000-2000 μ s. Grafik di atas memperlihatkan kenaikan variabel uji pada *range* 1100-1400 μ s yang menghasilkan perubahan RPM dari kisaran 2000 RPM hingga mencapai kisaran 6000 RPM. Peningkatan variabel uji total dari sebesar 300 μ s memperoleh peningkatan RPM sebesar 4000 RPM, berarti jika variabel uji 1700 μ s, tiap *propeller* akan memiliki RPM sebesar 10.000 RPM.

C. Pengujian Terbang

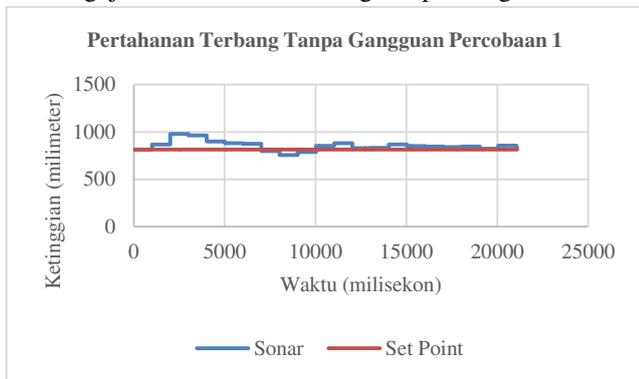
- Pengujian Keseimbangan Terbang



Gambar. 19. Pembacaan Nilai Sudut MPU6050 Saat Terbang

Grafik di atas menunjukkan *sampling* data dari kedua sudut dari sumbu putar x dan y. Diperoleh grafik yang sesuai dengan hasil pengujian pemilihan kontanta PID. Pada kondisi terbang ini terlihat bahwa terjadi fluktuasi pada sumbu putar x yaitu sekitar 1° sampai 5° dengan rata-rata 2° kemiringan. Pada sumbu putar y terdapat fluktuasi sebesar 1° sampai (-5°) dengan rata-rata -3° kemiringan. Hasil uji pada *set point* 0° tidak terpenuhi, namun menyimpang dengan kemiringan 2° dan -3°. Masalah penyimpangan *set point* ini dapat diatasi dengan menggunakan RC, yaitu mengatur *offset* masing-masing sumbu yang menyimpang. Jika penyimpangan terjadi pada sumbu putar x (*roll*), sebesar 3°, *offset roll* pada RC ditambahkan dari nilai tengahnya hingga penerbangan *Quadcopter* tidak cenderung ke arah tertentu lagi.

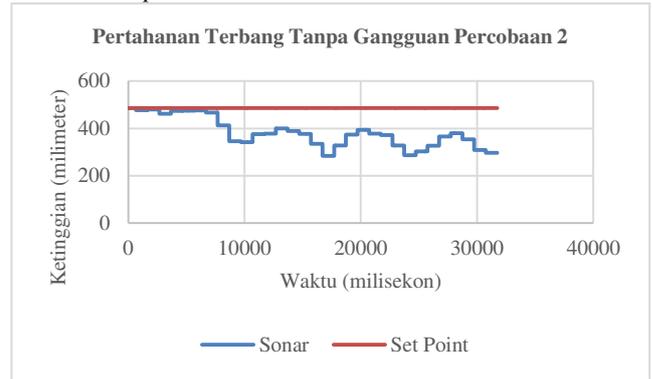
- Pengujian Pertahanan Terbang Tanpa Gangguan



Gambar. 20. Pembacaan SRF02 Saat Terbang dengan Gangguan (1)

Pada gambar 20 memperlihatkan bahwa saat *set point* atau ketinggian yang diinginkan yaitu 615 mm. Terjadi penurunan dari *set point* sebesar 58 mm karena osisilasi yang menurun tersebut. Data dari sonar kemudian menunjukkan lagi peningkatan dan penurunan ketinggian yang masih termasuk dalam toleransi ketinggian yang

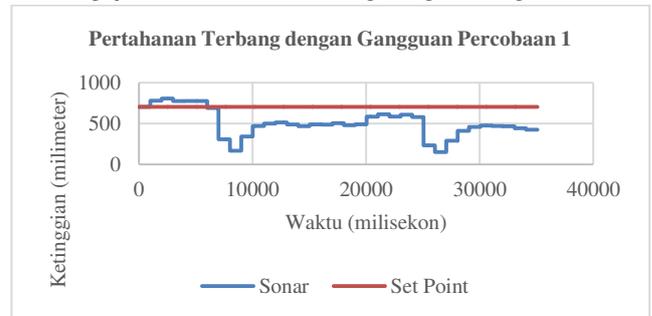
ditentukan yaitu 200 mm batas bawah, dan 200 mm batas atas dari *set point*.



Gambar. 21. Pembacaan SRF02 Saat Terbang Tanpa Gangguan (2)

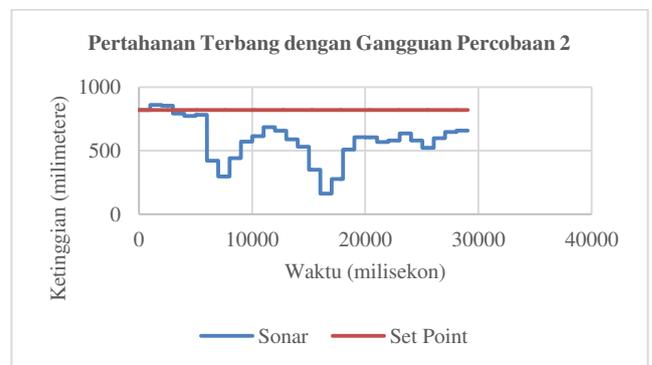
Pembacaan sonar pada percobaan ini mengalami osilasi yang cenderung menurun. Selain itu pembacaan sonar ini menunjukkan bahwa *set point* tidak dicapai seperti pada percobaan pertama. Hal ini ini membuktikan bahwa *Quadcopter* terbang lebih stabil saat jarak terbang mendekati mencapai 1 meter.

- Pengujian Pertahanan Terbang dengan Gangguan



Gambar. 22. Pembacaan SRF02 Saat Terbang dengan Gangguan (1)

Pada percobaan pertama pertahanan terbang ini, *set point* yang ditentukan adalah pada ketinggian 705 mm. Pada percobaan pertama pertahanan terbang ini, gangguan diberikan sebanyak dua kali. Gangguan pertama yang diberikan pada *Quadcopter* menyebabkan sonar memberikan pembacaan ketinggian 167 mm selama 1 detik. Gangguan kedua yang dilakukan menyebabkan *Quadcopter* memiliki ketinggian 151 mm.



Gambar. 23. Pembacaan SRF02 Saat Terbang dengan Gangguan (2)

Pada percobaan kedua pertahanan terbang dengan gangguan, *set point* yang ditentukan adalah 821 mm. Gangguan pertama menyebabkan *Quadcopter* memiliki ketinggian 298 mm selama kisaran 1 detik sedangkan gangguan kedua menyebabkan *Quadcopter* terbang rendah pada ketinggian 163 mm.

- Pengujian Kecepatan Rotasi Saat Terbang
 Pengambilan data RPM tidak dilakukan saat *Quadcopter* terbang. Pengujian ini dilakukan dengan memberikan kecepatan maksimal pada data RC *throttle*.

Tabel. 2. Pengujian RPM dengan Nilai Maksimal dan Minimal RC

Variabel Kecepatan	RPM Aktual <i>Propeller</i>				Hipotesis
	D10	D9	D6	D5	
1100	2639	2708	2864	2856	2000
1700	10033	10337	10808	10456	10000

Variabel 1100 μ s diperoleh dari pengujian sebelumnya, sedangkan 1700 μ s merupakan data rata-rata RC *throttle* saat terbang dipengujian sebelumnya. Tabel di atas digunakan untuk melakukan perkiraan pertambahan kecepatan hingga mencapai nilai maksimum 1000 μ s – 2000 μ s (skala data) pada *flight controller* dengan menggunakan baterai 2 *cells* (7.4 volt). Berdasarkan pengujian ini dapat disimpulkan bahwa hubungan peningkatan RPM dan variabel kecepatan saling keterkaitan. Pada ketinggian sekitar 1475 mm, menggunakan perbandingan kecepatan di atas diperoleh perkiraan RPM *propeller* sebesar 10.000 RPM.

V. KESIMPULAN

Berdasarkan pengerjaan tugas akhir ini, diperoleh beberapa kesimpulan diantaranya sebagai berikut:

1. *Quadcopter* dapat melakukan penerbangan yang stabil dengan menggunakan kombinasi PID *controller* sebagai berikut:
 - Konstanta PID *controller* pada sumbu putar x (*roll*) menggunakan:
 - Konstanta *Proportional* = 40
 - Konstanta *Integral* = 40
 - Konstanta *Derivative* = 60
 Pada kombinasi ini, *Quadcopter* memiliki *settling time* sekitar 1000 ms untuk mencapai *steady state error* meskipun dengan pemberian gangguan kemiringan lebih dari 40°. Pada pengujian terbang diperoleh grafik perubahan sudut yang berfluktuasi dari 1° sampai 5° dengan rata-rata 2° kemiringan.
 - Konstanta PID *controller* pada sumbu putar y (*pitch*) menggunakan:
 - Konstanta *Proportional* = 40
 - Konstanta *Integral* = 60
 - Konstanta *Derivative* = 60
 Pemberian gangguan lebih dari kemiringan 40°, kombinasi konstanta PID *controller* ini mampu meredam amplitudo penyimpangan sudut hampir 100% dengan *settling* yang sangat singkat. Pada saat terbang, terdapat fluktuasi sebesar 1° sampai (-5°) dengan rata-rata -3° kemiringan.
2. Dalam mempertahankan ketinggian terbang digunakan PID *controller* dengan konstanta PID dua area atas dan bawah.
 - Konstanta PID area bawah:
 - Konstanta *Proportional* = 9/10
 - Konstanta *Integral* = 1/10
 - Konstanta *Derivative* = 5
 - Konstanta PID area atas:
 - Konstanta *Proportional* = 1/00
 - Konstanta *Integral* = 1/1000
 - Konstanta *Derivative* = 10

PID *controller* untuk mempertahankan ketinggian 815 mm berhasil mempertahankan *set point*. Pada ketinggian 485 mm terjadi osilasi serta penyimpangan *set point* sebesar 200 mm namun masih termasuk batas toleransi ketinggian yang ditentukan. Pada saat pemberian gangguan, osilasi juga terjadi namun sistem menunjukkan keberhasilan karena kedua *set point* saat diberikan gangguan masih menuju batas bawah toleransi *set point*.

- Pada ketinggian sekitar 1475 mm, menggunakan perbandingan kecepatan antara variabel kecepatan dan kecepatan rotasi aktual diperoleh perkiraan RPM *propeller* sebesar 10.000 RPM. Jarak ketinggian tidak dapat menjadi acuan penentu dalam penambahan kecepatan motor akibat gaya berat yang menyebabkan penurunan jarak ketinggian. Hal tersebut terjadi karena pada kecepatan yang sama yaitu dengan kisaran 10.000 RPM, pembacaan *sensor* akhir menunjukkan ketinggian 1200 mm.

DAFTAR PUSTAKA

- [1] Luukkonen, T. (2011, August 22). Modelling and control of quadcopter. *School of Science*, 10.
- [2] Multiwii. (n.d.). *Multiwii*. Retrieved from multiwii.com: <http://www.multiwii.com>