

Single Axis Quadcopter menggunakan Proporsional Integral Derivative (PID) berbasis Labview

Single Axis Quadcopter using Proportional Integral Derivative (PID) based Labview

Rivaldi Priansyah, Asep Miftahul M

Program Studi Teknik Elektro, Fakultas Teknik dan Ilmu Komputer
Universitas Komputer Indonesia Jl. Dipati ukur No 112, Bandung
Email : priansyahivaldi@email.unikom.ac.id

Abstrak - *Quadcopter* adalah sebuah *Miniature Aerial Vehicle* (MAV) yang mempunyai 4 buah baling-baling dengan kemampuan lepas landas, mendarat secara vertikal dan kemampuan menegosiasikan ruang sempit yang memungkinkan untuk digunakan di area dalam ruangan yang berantakan. Karena kemampuan inilah *quadcopter* baru-baru ini dipertimbangkan untuk berbagai aplikasi baik di bidang militer maupun sipil. Secara khusus, *quadcopter* telah dieksplorasi untuk aplikasi seperti pengawasan dan eksplorasi bencana. *Quadcopter* saat ini banyak digunakan karena mempunyai kemampuan untuk melakukan gerakan yang sulit dilakukan oleh pesawat jenis lain, serta kemampuan untuk terbang dan mendarat secara vertikal di tempat yang sempit dengan sistem yang lengkap mekanisme stabilitas rotor. Pada penelitian ini digunakan sebuah simulasi secara otomatis yang dilakukan pada sebuah laptop dengan software pemrograman berbasis Labview, simulasi pada penelitian ini hanya menggunakan satu sudut pada *quadcopter* sehingga hanya membutuhkan 2 buah motor sebagai penggerak baling-baling atau biasa disebut dengan *single axis quadcopter*. Pengendali *proportional integral derivative* (PID) untuk gerakan sudut *roll* pada *single axis quadcopter* ini diharapkan mampu mengembalikan posisi *roll* sesuai dengan *setpoint* dan dimaksudkan untuk mengoptimalkan stabilitas. Dibutuhkan sebuah sistem pengendali untuk mengatur gerakan sudut *roll*, sehingga dapat bergerak dengan stabil dan mampu mengatasi sebuah gangguan yang diberikan. Sebagai pengendali gerakan sudut *roll* digunakan pengendali PID, yang parameternya ditentukan dengan program aplikasi Labview. Parameter PID ini di program di dalam Labview begitupun simulasi percobaan. Berdasarkan hasil perhitungan di dapatkan nilai $K_p = 0,8$, $K_i = 25$, $K_d = 0,004$. Sistem pengendali mampu meredam osilasi dengan *error rate* 2 derajat dari input. Dengan koefisien PID sistem mampu mengembalikan kondisi sudut *roll* ketika diberi gangguan pada simulasi.

Kata kunci : *Single Axis Quadcopter, Quadcopter, PID, roll, Labview*

Abstract - *Quadcopter* is a *Miniature Aerial Vehicle* (MAV) that has 4 propellers with the ability to take off, land vertically and the ability to negotiate a narrow space that allows it to be used in a messy indoor area. Because of this ability the *quadcopter* has recently been considered for various applications both in the military and civilian fields. In particular, *quadcopter* has been explored for applications such as surveillance and disaster exploration. *Quadcopter* is currently widely used because it has the ability to make movements that are difficult for other types of aircraft, as well as the ability to fly and land vertically in a narrow place with a complete system of rotor stability mechanisms. In this study an automatic simulation was carried out on a laptop with Labview-based programming software, the simulation in this study only used one angle on the *quadcopter* so that it only needed 2 motors as propeller drives or commonly called *single axis quadcopter*. An *integral derivative* (PID) controller for roll angle movement on a single shaft *quadcopter* is expected to be able to restore the roll position according to the *setpoint* and support to optimize. A control system is needed to adjust roll angle movement, so that it can move stably and be able to overcome a given disturbance. As a roll angle movement controller PID controller is used, whose parameters are determined by the Labview application program. This PID parameter in the program in Labview as well as experiment simulation. Based on the calculation results, get the value $K_p = 0.8$, $K_i = 25$, $K_d = 0.004$. The control system is able to reduce oscillation with a 2 degree error rate from the input. With the PID coefficient the system is able to restore the roll angle condition when given a simulation error.

Keyword : *Single Axis Quadcopter, Quadcopter, PID, roll, Labview*

I. PENDAHULUAN

Quadcopter merupakan salah satu jenis pesawat tanpa awak yang memiliki empat baling-baling yang digerakan oleh empat buah motor. Keempat motor ini dapat dikendalikan kecepatannya secara terpisah, dua motor berputar searah jarum jam dan dua motor lain berputar berlawanan arah jarum jam. Helikopter merupakan salah satu alat transportasi udara yang menggunakan rotary wing untuk bergerak dan mampu untuk melayang di udara dalam posisi diam (*hovering*), namun memiliki kecepatan translasi yang lebih kecil. Helikopter konvensional dengan satu rotor utama dan satu rotor di ekor memiliki banyak sifat yang sama dengan *quadcopter*. *Quadcopter* tidak mempunyai bagian yang bergerak kecuali baling-baling yang digerakkan oleh motor, sedangkan helikopter konvensional membutuhkan hub kompleks untuk memutar sumbu motor agar dapat menginduksi gerakan translasi.

A. Latar Belakang

Quadcopter adalah salah satu desain paling populer untuk MAV karena kemampuan lepas landas dan mendarat secara vertikal, kesederhanaan konstruksi, kemampuan manuver, dan kemampuan untuk menegosiasikan ruang sempit yang memungkinkan untuk digunakan di dalam area ruangan yang berantakan. Karena kemampuan ini, *quadcopter* baru-baru ini dipertimbangkan untuk berbagai aplikasi baik di bidang militer maupun sipil. Secara khusus, *quadcopter* MAV telah dieksplorasi untuk aplikasi seperti pengawasan dan eksplorasi bencana (seperti kebakaran, gempa bumi, dan banjir), operasi pencarian dan penyelamatan, dan jaringan sensor seluler [1-2].

Balon, pesawat bersayap tetap, helikopter rotor tunggal, helikopter rotor ganda koaksial, quad-rotor, *quadcopter* rotor miring adalah contoh konfigurasi yang berbeda dan mekanisme propulsi yang telah dikembangkan untuk memungkinkan gerakan 3D di platform udara [3]. Masing masing memiliki kelebihan dan kekurangan.

Quadcopter mempunyai kelebihan pada mobilitas dan fleksibilitas untuk menjelajahi pada wilayah yang sempit [4]. *Quadcopter* mempunyai kemampuan melakukan manuver yang sulit dilakukan pesawat jenis lain, melayang di udara dalam posisi diam (*hovering*), serta terbang dan mendarat secara vertikal. Tidak seperti kendaraan di darat, pergerakan *quadcopter* memiliki sedikit

gesekan, oleh karena itu harus ada redaman untuk menghentikan gerakannya dan menjaga kestabilannya [5].

Perancangan pengendali untuk gerakan sudut *roll* dan *pitch* ini mengambil salah satu sudut saja yaitu sudut *roll* dimana sudut *roll* ini berada pada motor 2 dan motor 4. Untuk perancangan pengendali gerakan sudut *roll* pada *quadcopter* diharapkan mampu mengembalikan posisi *roll* sesuai dengan *setpoint* yang telah ditentukan, begitu juga saat diberi gangguan. Sudut *roll* dikendalikan dengan cara menambah atau mengurangi kecepatan salah satu motor kiri atau kanan. Karena *quadcopter* membutuhkan ketelitian dan *error actual* yang kecil, maka sistem ini menggunakan pengendali PID.

B. Tinjauan State of Art

Secara konvensional, sikap *quadrotor* dikendalikan dengan mengubah kecepatan rotasi setiap motor. Pasangan rotor depan dan rotor belakang berputar searah jarum jam, sedangkan pasangan rotor kanan dan kiri berputar dalam arah berlawanan arah jarum jam. Konfigurasi ini dirancang untuk menyeimbangkan mereka yang diciptakan oleh masing-masing pasangan motor. Pada dasarnya ada empat manuver yang dapat dilakukan dengan mengubah kecepatan empat rotor. Dengan mengubah kecepatan relatif rotor kanan dan kiri, sudut gulungan rotor dikendalikan. Demikian pula, sudut *pitch* dikontrol oleh variatif yang berbeda memiliki kecepatan rotor depan dan belakang, dengan memvariasikan kecepatan pasangan berputar searah jarum jam dan pasangan berputar berlawanan arah jarum jam. Menambah atau mengurangi kecepatan keempat rotor secara bersamaan mengontrol dorongan kolektif yang dihasilkan oleh robot [6-7].

Salah satu keterbatasan dasar dari desain *quadrotor* adalah bahwa dengan hanya memiliki 4 input kontrol independen. Kecepatan berputar 4 baling-baling, kontrol independen dari posisi enam dimensi dan orientasi *quadrotor* tidak dimungkinkan. Sebagai contoh, sebuah *quadrotor* dapat melayang di tempat saja dan jika hanya ketika horisontal ke bidang tanah atau perlu miring sepanjang arah gerakan yang diinginkan untuk dapat bergerak [8]. Memiringkan konsep rotor *quadcopter* telah berkembang untuk mengatasi keterbatasan dasar *quadrotor* ini. Sebagai contoh, dalam [9], konsep aktuasi novel untuk *quadrotor* di mana baling-baling diizinkan untuk memiringkan tentang sumbu yang menghubungkan mereka ke kerangka tubuh

utama, dengan demikian mewujudkan rotor quad [10].

Desain kemiringan membuat dinamika *quadcopter* lebih kompleks, dan memperkenalkan tantangan tambahan dalam desain kontrol. Namun, memiringkan rotor *quadcopter*, dirancang dengan menggunakan empat motor servo tambahan yang memungkinkan rotor untuk dimiringkan, adalah sistem yang terlalu digerakkan yang berpotensi dapat melacak lintasan sewenang-wenang dari waktu ke waktu [11]. Ini memberi kendali penuh atas posisi dan orientasi *quadrotor* yang memberikan kemungkinan melayang di konfigurasi miring.

C. Tujuan

Tujuan dari penelitian ini adalah membuat simulasi pada *quadcopter* dengan sistem kontrol berbasis PID menggunakan aplikasi labview. Simulasi yang dibuat hanya membuat satu buah sumbu yaitu pada sumbu Y.

D. Sistematika Pembahasan

Sistematika pada penulisan penelitian ini terdiri dari beberapa bagian. Pada bagian 1 menjelaskan mengenai latar belakang, tinjauan, tujuan dan sistematika pembahasan. Pada bagian 2 menjelaskan tentang perancangan sistem kendali yang digunakan, metode *tuning* dan diagram alir dari penelitian yang dilakukan. Pada bagian 3 akan memaparkan proses uji coba dan data yang didapat dari penelitian yang dilakukan mengenai cara kerja sistem, hasil pengujian, pengendalian PID, perbandingan pengujian dan *tuning* PID. Untuk Kesimpulan dan saran akan disajikan pada bagian 4.

II. METODOLOGI

Pada metodologi penelitian dibagi menjadi beberapa perancangan alat seperti, perancangan blok diagram sistem, perancangan sistem kendali, pengendalian PID, cara kerja *single axis Quadcopter*, bagian – bagian dari *plant* simulasi, listing program dan cara pengoprasian simulasi. Berikut adalah blok diagram secara komprehensif pada **Gambar 1**.



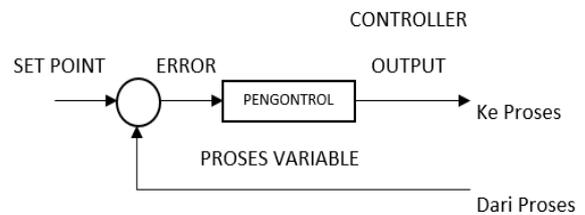
Gambar 1. Blok diagram sistem

Pada blok diagram **Gambar 1** terdapat tiga bagian utama yaitu input, proses dan output. Proses input digunakan untuk memasukan nilai

setpoint yang di inginkan. Sedangkan bagian proses menggunakan PID dan output akan menampilkan gerak dari *quadcopter*.

A. Pengontrol PID

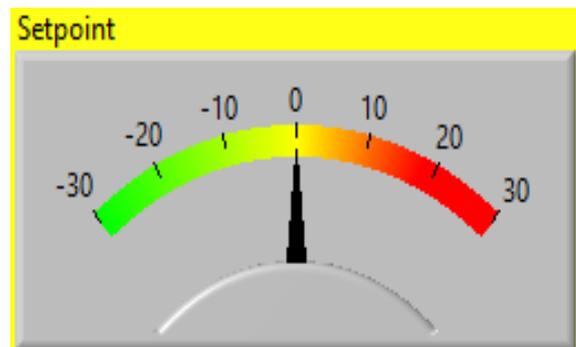
Sistem Kontrol PID merupakan kontroler untuk menentukan presisi suatu sistem instrumentasi dengan karakteristik adanya umpan balik pada sistem tesebut (*Feedback*). Sistem kontrol PID terdiri dari tiga buah cara pengaturan yaitu kontrol P (*Proportional*), D (*Derivative*) dan I (*Integral*), dengan masing-masing memiliki kelebihan dan kekurangan. Dalam implementasinya masing-masing cara dapat bekerja sendiri maupun gabungan diantaranya. Dalam perancangan sistem kontrol PID yang perlu dilakukan adalah mengatur parameter P, I atau D agar tanggapan sinyal keluaran sistem terhadap masukan tertentu sebagaimana yang diinginkan. Secara umum hubungan antara harga yang diinginkan atau disebut *setpoint*, harga proses yang terukur (*Process Variable*), sinyal kesalahan (*Error*) dan keluaran pengontrol (*Controller Output*) tampak seperti **Gambar 2**.



Gambar 2. Blok diagram pengontrol

B. Sistem Pengontrol

Sistem pengontrol yang di terapkan disini yaitu pengontrol PID yang di dalam nya terdapat 3 nilai yang harus di ketahui dengan cara tuning parameter pengontrol PID. Salah satunya yaitu dengan metode osilasi. Pada **Gambar 3** terdapat pengontrol *setpoint* yang berfungsi untuk mengatur sudut kemiringan dari *quadcopter*.



Gambar 3. Panel *Setpoint*

C. Tuning Parameter Pengontrol PID

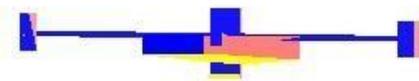
Agar diperoleh respon pengontrol yang terbaik, maka perlu dilakukan tuning parameter pengontrol PID. Dalam tuning, kita mencari nilai-nilai dari *Proportional Gain* (K_p), Waktu *Integral* (T_i) dan waktu *derivative* (T_d). Penalaan parameter *controller* PID selalu didasari atas tinjauan terhadap karakteristik yang diatur (*Plant*). Dengan demikian betapa rumitnya suatu *plant*, perilaku *plant* tersebut harus diketahui terlebih dahulu sebelum penalaan parameter PID itu dilakukan. Karena penyusunan model matematik *plant* tidak mudah, maka dikembangkan suatu metode eksperimental.

Metode ini didasarkan pada reaksi *plant* yang dikenai suatu perubahan. Dengan menggunakan metode itu model matematik perilaku *plant* tidak diperlukan lagi, karena dengan menggunakan data yang berupa kurva keluaran, penalaan kontroler PID telah dapat dilakukan. Penalaan bertujuan untuk mendapatkan kinerja sistem sesuai spesifikasi perancangan. Dua metode pendekatan eksperimen adalah Ziegler-Nichols.

D. Cara Kerja Single Axis Quadcopter

Pembuatan simulasi kontrol *Single Axis Quadcopter* ini menggunakan PID berbasis Labview. Meliputi proses pemrograman yang berfungsi saat setpoint (titik acuan) di atur, maka akan mengontrol 2 buah motor agar dapat bekerja sesuai dengan kontrol PID yang diinginkan, dan menampilkan data tersebut melalui sebuah simulasi pada Labview. Untuk tampilan simulasi program terdapat pada **Gambar 11**.

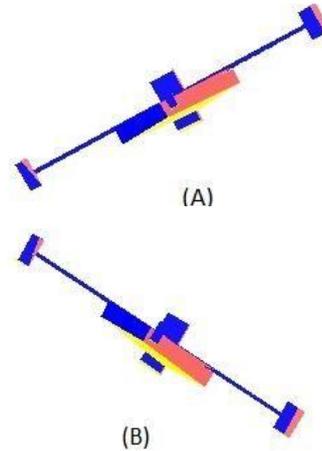
Cara kerja simulasi *Single Axis Quadcopter* ini memiliki 2 buah baling-baling yang digerakkan oleh dua buah motor yang berada pada posisi sumbu "Y" yang diperlihatkan pada **Gambar 4**.



Gambar 4. Sumbu "Y"

Gerakan yang dilakukan *Single Axis Quadcopter* ini yaitu gerakan *roll*. Konsep gerakan motor pada motor kanan dan motor kiri berputar searah. Konfigurasi ini dirancang untuk menyeimbangkan gaya tarik dan gaya dorong yang dihasilkan masing-masing pasangan motor. Untuk melakukan *hovering* atau posisi diam

kedua motor harus mempunyai kecepatan yang sama. Kombinasi kecepatan motor menghasilkan gerakan yang berbeda. Pada **Gambar 5** memperlihatkan gerakan dasar dari *Single Axis Quadcopter* dengan mengubah-ubah kecepatan dari kedua motornya.



Gambar 5. Gerakan Dasar *Single Axis Quadcopter*

Gerakan dasar *Single Axis Quadcopter* yaitu gerakan *roll* pada sumbu "y". Seperti pada **Gambar 4** gerakan ini dilakukan dengan menurunkan kecepatan satu motor kiri (A) dan menaikkan kecepatan satu motor kanan. Dengan pengaturan seperti ini, motor akan bergerak miring ke arah perputaran motor yang lebih lambat kekecepatannya begitupun sebaliknya apabila menurunkan kecepatan motor kanan (B) dan menaikkan kecepatan motor kiri maka akan bergerak miring ke arah perputaran motor yang lebih lambat kekecepatannya.

E. Bagian – Bagian Dari *Plant* Simulasi

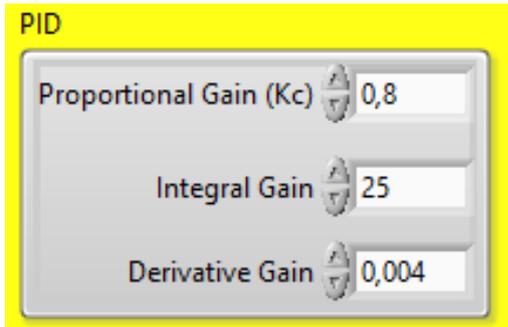
Ada beberapa bagian – bagian *plant* dan fungsi yang digunakan pada simulasi ini adalah sebagai berikut.

A. Panel *setpoint*

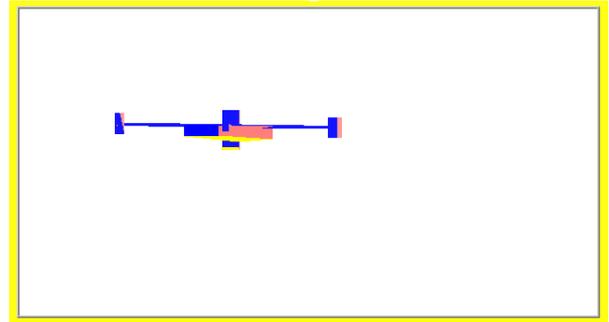
Panel *setpoint* digunakan untuk menentukan berapa *setpoint* yang di ingin kan dengan cara memindahkan jarum. Pada **Gambar 3** adalah panel *setpoint* yang digunakan memiliki rentang dari 30 s/d -30.

B. Parameter Pengontrol PID

Parameter pengontrol PID digunakan untuk mengontrol nilai P , I, dan D dengan cara memasukan nilai yang sudah diketahui ke dalam bagian pengontrol ini. **Gambar 6** adalah bagian dari pengontrol PID yang sudah dimasukan dengan nilai $P = 0,8$, $I = 25$, dan $D = 0,004$.



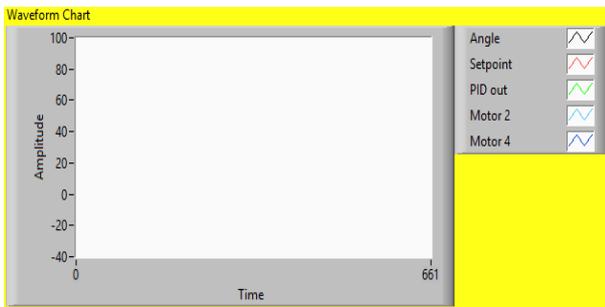
Gambar 6. Parameter Pengontrol PID



Gambar 9. Tampilan *Quadcopter*

C. *Waveform chart*

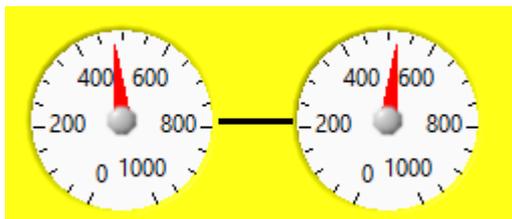
Waveform chart digunakan untuk mengetahui tampilan grafik performa sistem, pada **Gambar 7** adalah tampilan *waveform chart* yang digunakan dalam simulasi ini.



Gambar 7. Tampilan *Waveform Chart*

D. RPM Motor

RPM motor digunakan untuk mengetahui kecepatan motor. Pada **Gambar 8** adalah RPM motor yang digunakan dalam simulasi ini dengan RPM motor ini penulis dapat mengetahui kecepatan masing – masing motor.



Gambar 8. RPM Motor

E. Tampilan *Quadcopter*

Tampilan *quadcopter* digunakan untuk mengetahui hasil akhir dari pemrosesan dalam bentuk gambar bergerak. Pada **Gambar 9** adalah tampilan dari simulasi *Quadcopter*.

F. Tampilan Simulasi dan Program *Single Axis Quadcopter*

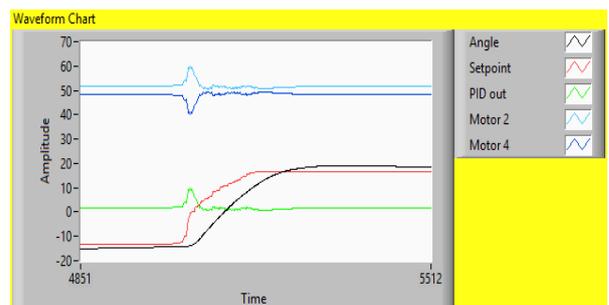
Pada **Gambar 11** adalah tampilan utama simulasi *single axis quadcopter* dan **Gambar 12** adalah program keseluruhan sistem dan **Gambar 13** adalah bagian kendali PID.

G. Cara Pengoperasian Simulasi

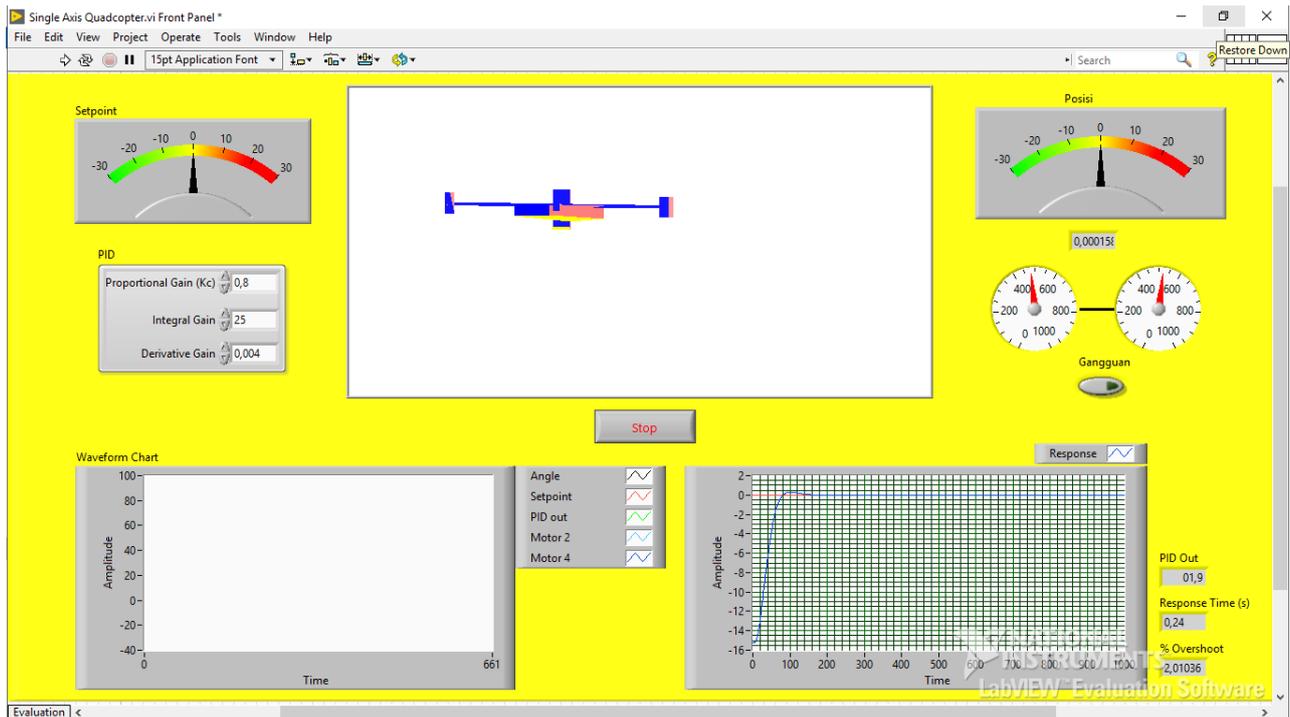
Untuk melakukan pengoperasian simulasi *single axis quadcopter* ini adalah sebagai berikut:

- Pertama download file simulasi pada halaman ini.
<http://bit.ly/2wiYM9H>
- Pada file tersebut terdapat 2 folder dan 1 file vi yang bernama *single axis quadcopter*.
- Buka file yang bernama *single axis quadcopter*.vi akan terlihat seperti pada **Gambar 11**.
- Kemudia sis nilai $P = 0,8$ $I = 25$ dan $D = 0,004$.
- Jalankan program.
- Atur *setpoint* untuk mengatur kemiringan dari *quadcopter*.
- Lihat grafik pada *waveform chart* akan tampil seperti **Gambar 10**.
- Tombol gangguan digunakan untuk menguji PID yang digunakan.

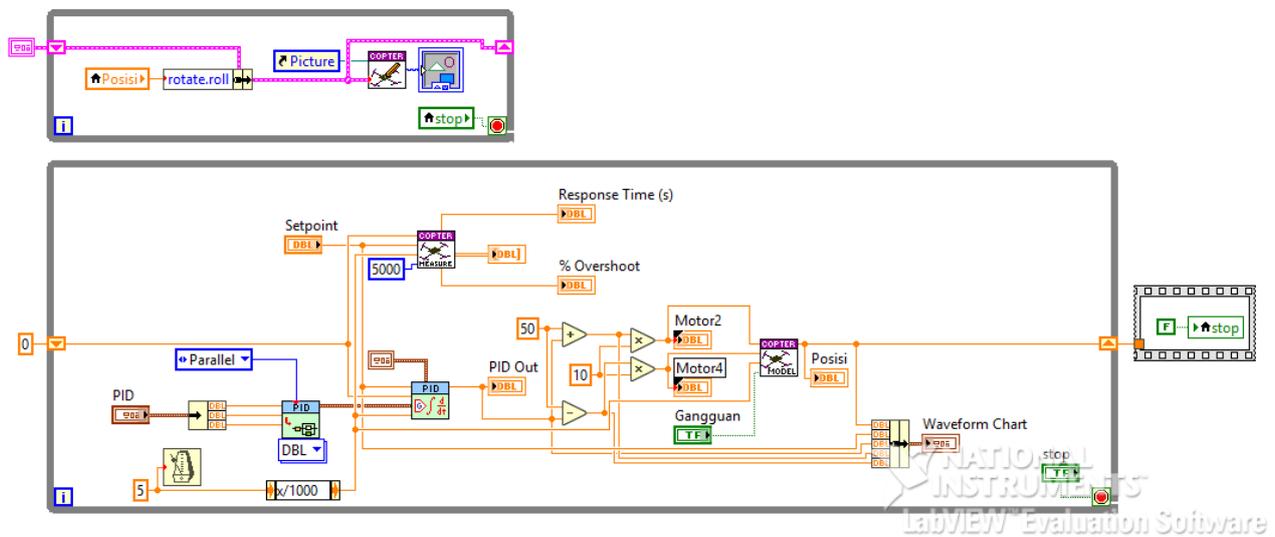
Untuk melihat pengoperasian lebih jelas dapat di akses videonya melalui halaman berikut ini.
<https://youtu.be/EM74rU42ZXM>



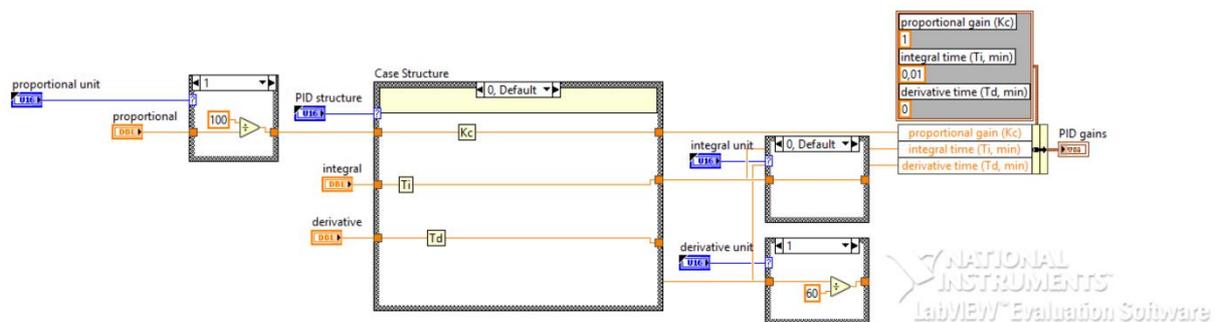
Gambar 10. Grafik *Waveform chart*



Gambar 11 . Tampilan Utama Program



Gambar 12. Program keseluruhan sistem



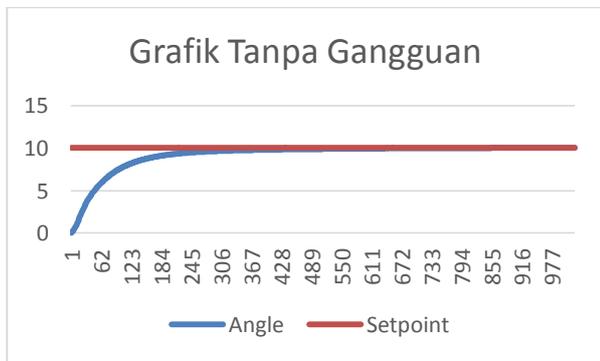
Gambar 13. Kendali PID

III. HASIL DAN PEMBAHASAN

Dalam merancang simulasi yang paling penting adalah pengujian dari simulasi tersebut, apakah simulasi memenuhi target yang di tentukan atau tidak. Dengan dicapainya tujuan yang telah ditargetkan maka simulasi tersebut dikatakan baik.

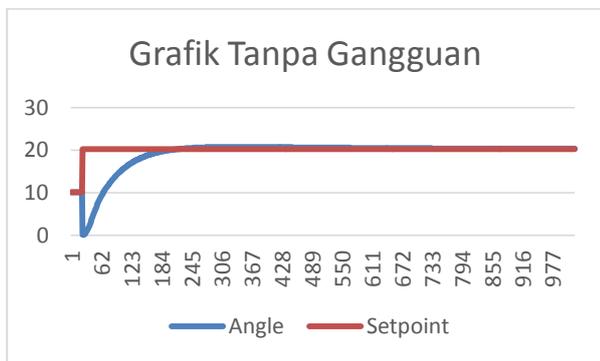
A. Pengujian Pengontrol PID Tanpa Gangguan

Pengujian pengontrol PID tanpa gangguan dilakukan guna untuk mengetahui hasil data grafik dari performansi sistem PID pada simulasi pengujian pengontrol PID berbasis Labview dengan menggunakan set point yang berbeda.



Gambar 14. Grafik Tanpa Gangguan

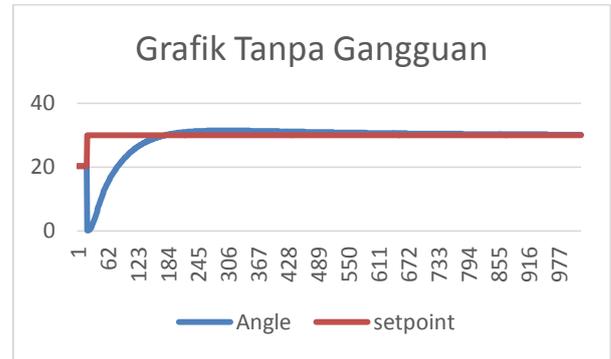
Pada Gambar 14. merupakan hasil grafik performansi sistem pengontrol PID tanpa gangguan dengan set point 10 dengan hasil sistem dapat bekerja dengan baik dan sistem dapat mengejar set point dengan nilai persentase overshoot 0,3 dan Respon waktu 0,4s. Lalu grafik dengan set point berbeda pada Gambar 15.



Gambar 15. Grafik Tanpa Gangguan

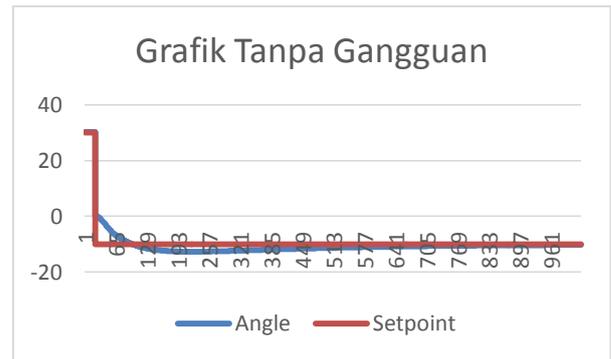
Pada Gambar 15. merupakan hasil grafik performansi sistem pengontrol PID tanpa gangguan dengan set point 20 dengan hasil sistem dapat bekerja dengan baik dan sistem dapat mengejar set point dengan nilai persentase

overshoot 4,0 dan Respon waktu 1,89 s. Lalu grafik dengan set point berbeda pada Gambar 16.



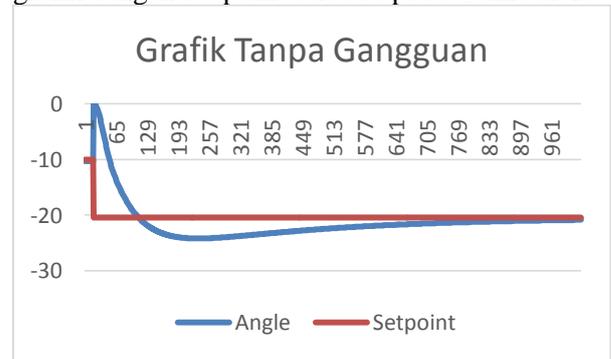
Gambar 16. Grafik Tanpa Gangguan

Pada Gambar 16. merupakan hasil grafik performansi sistem pengontrol PID tanpa gangguan dengan set point 30 dengan hasil sistem dapat bekerja dengan baik dan sistem dapat mengejar set point dengan nilai persentase overshoot 4,1 dan Respon waktu 1,89 s. Lalu grafik dengan set point berbeda pada Gambar 17.



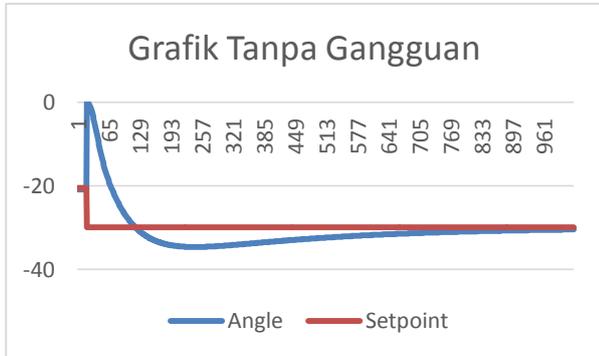
Gambar 17. Grafik Tanpa Gangguan

Pada Gambar 17. merupakan hasil grafik performansi sistem pengontrol PID tanpa gangguan dengan set point -10 dengan hasil sistem dapat bekerja dengan baik dan sistem dapat mengejar set point dengan nilai persentase overshoot 4,1 dan Respon waktu 1,89 s. Lalu grafik dengan set point berbeda pada Gambar 18.



Gambar 18. Grafik Tanpa Gangguan

Pada **Gambar 18**. merupakan hasil grafik performansi sistem pengontrol PID tanpa gangguan dengan set point -20 dengan hasil sistem dapat bekerja dengan baik dan sistem dapat mengejar set point dengan nilai persentase overshoot 4,1 dan Respon waktu 1,89 s. Lalu grafik dengan set point berbeda pada **Gambar 19**.

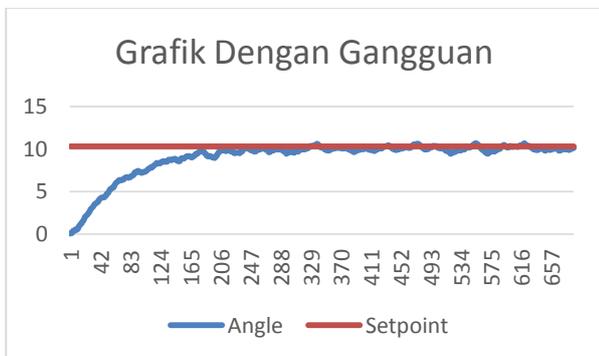


Gambar 19. Grafik Tanpa Gangguan

Pada **Gambar 19**. merupakan hasil grafik performansi sistem pengontrol PID tanpa gangguan dengan set point -30 dengan hasil sistem dapat bekerja dengan baik dan sistem dapat mengejar set point dengan nilai persentase overshoot 4,1 dan Respon waktu 1,89 s.

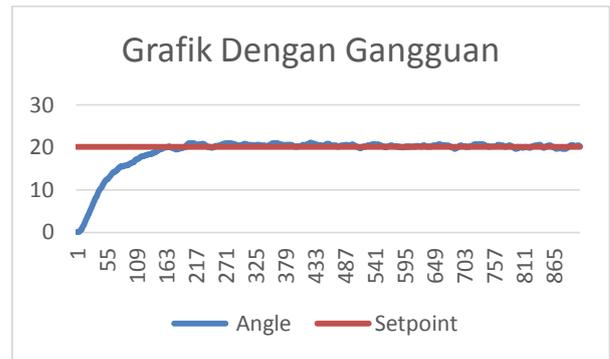
B. Pengujian Pengontrol PID Dengan Gangguan

Pengujian pengontrol PID dengan memberikan sebuah gangguan dilakukan guna untuk mengetahui hasil data grafik dari performansi sistem PID pada simulasi pengujian pengontrol PID berbasis Labview dengan menggunakan set point yang berbeda agar hasil dari pengujian pengontrol PID tanpa gangguan dan pengujian pengontrol PID dengan memberikan sebuah gangguan dapat dibedakan dengan menggunakan sebuah grafik.



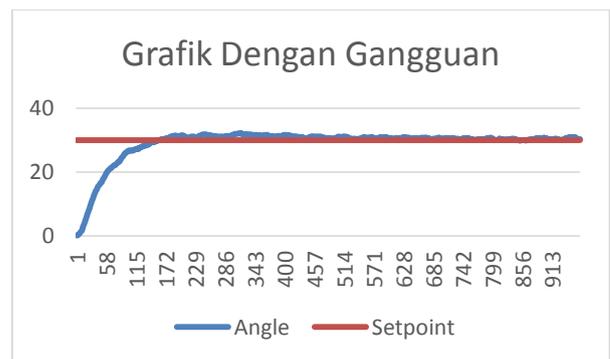
Gambar 20. Grafik Dengan Gangguan

Pada **Gambar 20**. merupakan hasil grafik performansi sistem pengontrol PID dengan memberikan sebuah gangguan dengan nilai set point 10 dengan hasil sistem dapat bekerja dengan baik dan sistem dapat mengejar set point dengan nilai persentase overshoot 79,1 dan Respon waktu 0,005 s. Lalu hasil data grafik dari performansi sistem PID pada simulasi dengan set point berbeda pada **Gambar 21**.



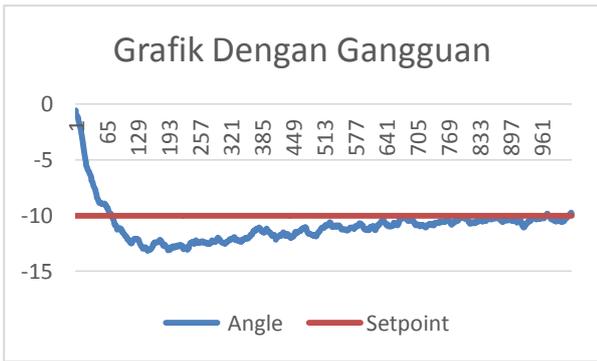
Gambar 21. Grafik Dengan Gangguan

Pada **Gambar 21**. merupakan hasil grafik performansi sistem pengontrol PID dengan memberikan sebuah gangguan dengan nilai set point 20 dengan hasil sistem dapat bekerja dengan baik dan sistem dapat mengejar set point dengan nilai persentase overshoot 92,7 dan Respon waktu 0,045 s. Lalu hasil data grafik dari performansi sistem PID pada simulasi dengan set point berbeda pada **Gambar 22**.



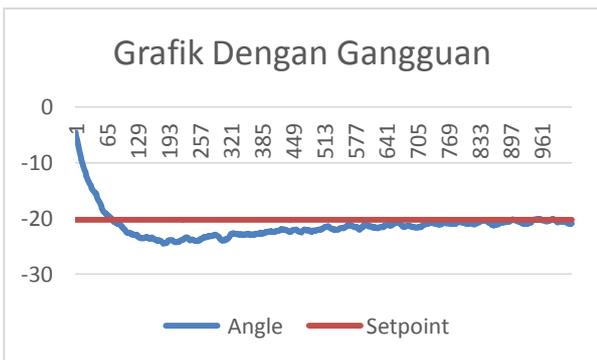
Gambar 22. Grafik Dengan Gangguan

Pada **Gambar 22**. merupakan hasil grafik performansi sistem pengontrol PID dengan memberikan sebuah gangguan dengan nilai set point 30 dengan hasil sistem dapat bekerja dengan baik dan sistem dapat mengejar set point dengan nilai persentase overshoot 87,8 dan Respon waktu 0,08 s. Lalu hasil data grafik dari performansi sistem PID pada simulasi dengan set point berbeda pada **Gambar 23**.



Gambar 23. Grafik Dengan Gangguan

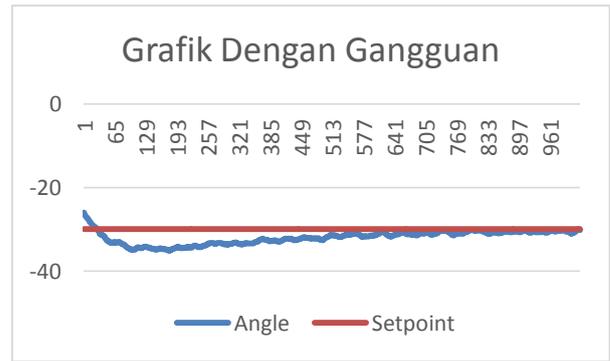
Pada **Gambar 23**, merupakan hasil grafik performansi sistem pengontrol PID dengan memberikan sebuah gangguan dengan nilai set point -10 dengan hasil sistem dapat bekerja dengan baik dan sistem dapat mengejar set point dengan nilai persentase overshoot 29,7 dan Respon waktu 0,25 s. Lalu hasil data grafik dari performansi sistem PID pada simulasi dengan set point berbeda pada **Gambar 24**.



Gambar 24. Grafik Dengan Gangguan

Pada **Gambar 24**, merupakan hasil grafik performansi sistem pengontrol PID dengan memberikan sebuah gangguan dengan nilai set point -20 dengan hasil sistem dapat bekerja dengan baik dan sistem dapat mengejar set point dengan nilai persentase overshoot 52,7 dan Respon waktu 0,005 s. Lalu hasil data grafik pada simulasi dengan set point berbeda pada **Gambar 25**.

Pada **Gambar 25**, merupakan hasil grafik performansi sistem pengontrol PID dengan memberikan sebuah gangguan dengan nilai set point -30 dengan hasil sistem dapat bekerja dengan baik dan sistem dapat mengejar set point dengan nilai persentase overshoot 43,9 dan Respon waktu 0,005 s.



Gambar 25. Grafik Dengan Gangguan

C. Penentuan Parameter PID

Langkah awal untuk pengujian pada simulasi ini yaitu dilakukan untuk mendapatkan nilai parameter Kp, Ki, dan Kd. Berdasarkan teori yang digunakan untuk mendapatkan nilai parameter tersebut dengan mencari nilai Kcr dan Pcr untuk mendapatkan nilai parameter Kp, Ki, dan Kd dengan menggunakan metode osilasi dan didapatkan hasil dari penentuan parameter PID, sebagai berikut:

1. Atur set point sesuai dengan yang diinginkan. Dalam percobaan di atur nilai set point 30.
2. Lalu set nilai Kp 1 lalu jalankan program, naikkan sedikit demi sedikit nilai Kp-nya hingga tercapai osilasi pada grafik, nilai Kp terakhir sama dengan nilai Kcr.
3. Hitung perioda dengan cara membaca grafiknya 1 Vpp berarti 1 perioda.
4. Hasil yang di dapatkan dari tuning yaitu:

$$Kcr = 1.5 \dots \dots \dots (1)$$

$$Pcr = 50 \dots \dots \dots (2)$$

5. Perhitungan untuk menentukan PID:

$$P = \frac{Kcr}{1.7}$$

$$P = \frac{1.5}{1.7}$$

$$P = 0.8 \dots \dots \dots (3)$$

$$I = \frac{Pcr}{2}$$

$$I = \frac{50}{2}$$

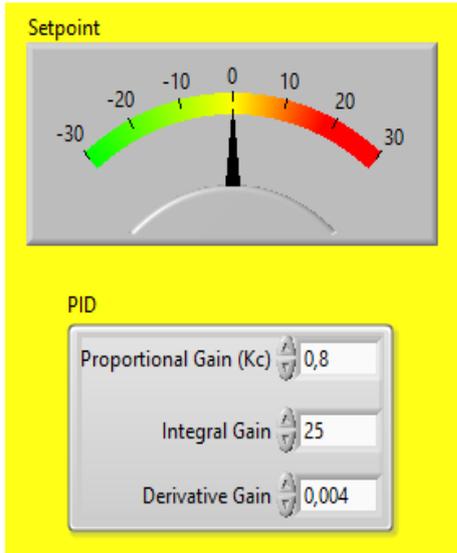
$$I = 25 \dots \dots \dots (4)$$

$$D = \frac{Pcr}{8}$$

$$D = \frac{0.032}{8}$$

$$D = 0.004 \dots \dots \dots (5)$$

Setelah didapatkan nilai PID, nilai-nilai tersebut akan dimasukkan ke dalam nilai pengontrol setiap parameter pada program yang telah dibuat pada **Gambar 26**.



Gambar 26. Bagian Pengontrol Simulasi

D. Hasil Pengujian

Setelah nilai keseluruhan Kp, Ki, dan Kd dimasukkan sesuai dengan perhitungan yang telah dilakukan maka pengontrol simulasi akan bekerja untuk mencapai nilai set point sesuai yang diinginkan. Hasil simulasi terlihat pada **Gambar 28** Pada simulasi sistem dilakukan tanpa menggunakan gangguan. Pada **Gambar 29** hasil simulasi sistem dilakukan menggunakan gangguan.

Dari percobaan yang telah dilakukan maka terdapat tingkat keberhasilan suatu sistem dapat dilihat pada **Tabel 1**. Pada pengujian sistem yang telah dilakukan maka dapat disimpulkan bahwa simulasi kerja sistem bekerja dengan sangat baik dari keadaan diam, stabil dan saat menerima gangguan.

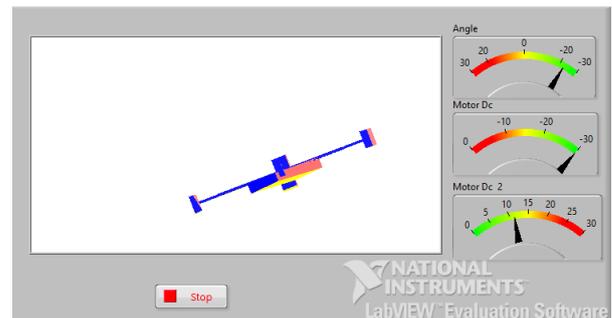
E. Perbandingan Pengujian PID

Pada percobaan ini dilakukan untuk membandingkan pengujian antara sistem percobaan menggunakan PID dan tanpa menggunakan PID. Pada percobaan tanpa menggunakan PID kedua motor tersebut berputar tidak stabil dikarenakan pengendalian motor diatur secara manual yaitu mengatur motor secara bergantian. Berbeda dengan program

menggunakan PID dimana motor berputar dengan stabil dikarenakan pengendalian motor diatur secara bersamaan oleh PID. Pada **Gambar 27** adalah bagian pengontrol secara manual tanpa PID.

Table 1. Tingkat Keberhasilan

Set Point	Tingkat Keberhasilan				
	Kanan	Diam	Kiri	Stabil	Gangguan
0	-	✓	-	✓	Aktif
10	✓	-	-	✓	Aktif
20	✓	-	-	✓	Aktif
30	✓	-	-	✓	Aktif
30 – 0	-	✓	-	✓	Aktif
-10	-	-	✓	✓	Aktif
-20	-	-	✓	✓	Aktif
-30	-	-	✓	✓	Aktif
(-30) - 0	-	✓	-	✓	Aktif



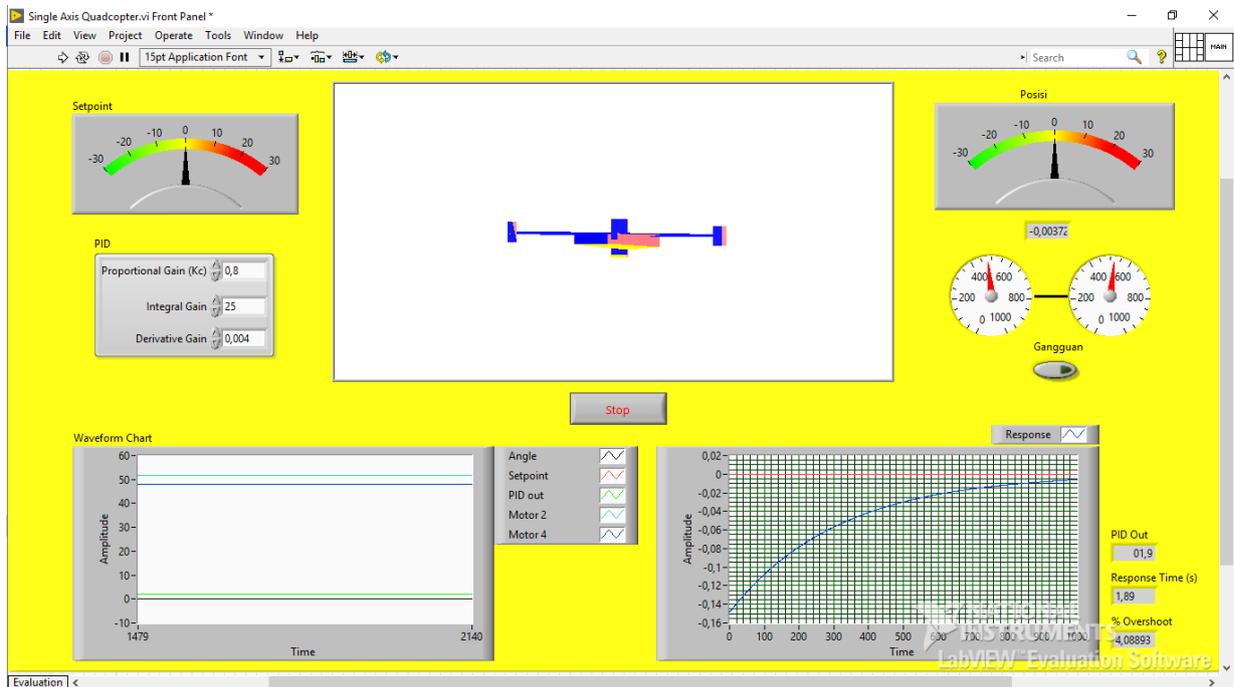
Gambar 27. Pengontrol Secara Manual

IV. KESIMPULAN DAN SARAN

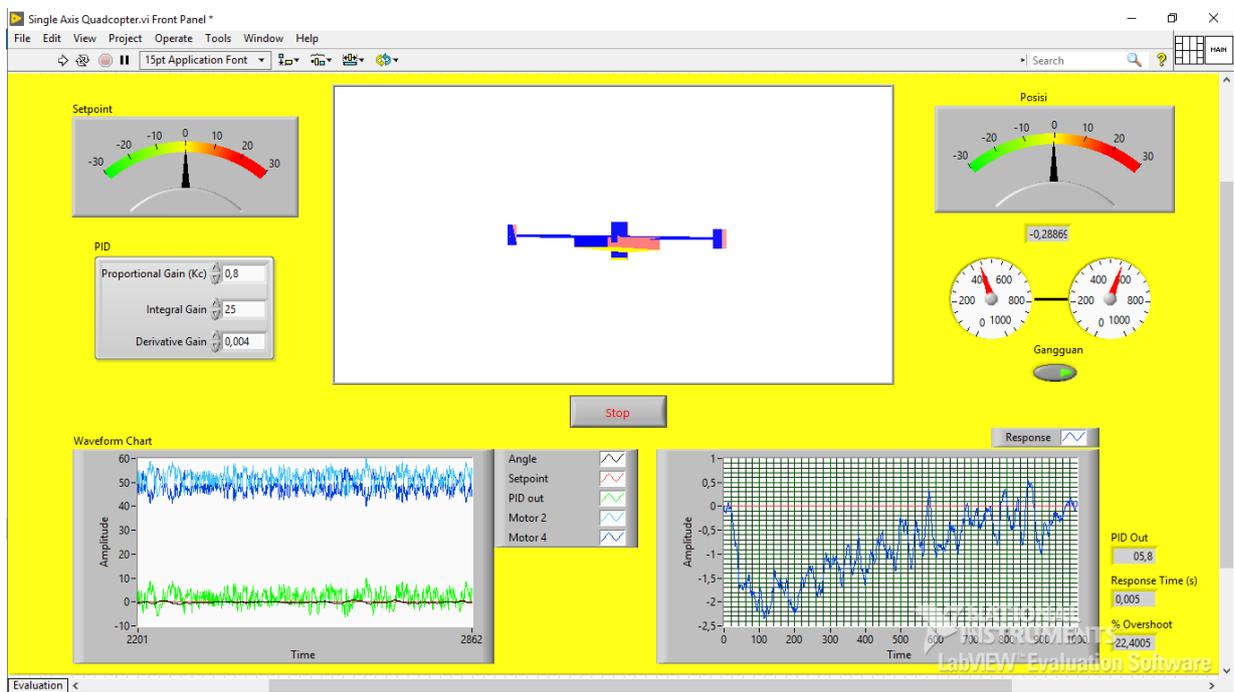
Setelah dilakukan pengujian pada sistem yang dirancang, dapat ditarik beberapa kesimpulan. Pada pengujian, sistem dapat merespon perubahan arah *setpoint* dengan tingkat keberhasilan 100%. Kemampuan sistem untuk mengubah posisi *quadcopter* dengan cepat mengikuti *setpoint* yang dipengaruhi oleh PID dalam sistem. Dengan koefisien PID Kp= 0,8, Ki= 25, Kd= 0,004 untuk mengendalikan gerakan *roll*, sistem pengendali *quadcopter* mampu meredam osilasi dan menghasilkan *error rate* 2 derajat dari input. Dengan koefisien PID tanpa menggunakan filter pada *Derivative*-nya, sistem pengendali *quadcopter* mampu meredam osilasi hingga menuju satu state tertentu. Koefisien pengendali PID yang dirancang tanpa filter mampu mengembalikan kondisi sudut *roll* ketika diberi gangguan pada simulasi.

Setelah menyelesaikan simulasi ini ada beberapa saran agar simulasi dapat dikembangkan menjadi lebih baik lagi. Sebaiknya simulasi ada

penambahan pengaturan kecepatan motor agar bisa di atur kecepatan motor nya secara manual.



Gambar 28. Hasil Simulasi Sitem Tanpa Menggunakan Gangguan



Gambar 29. Hasil Simulasi Sitem Menggunakan Gangguan

DAFTAR PUSTAKA

[1] H. Y. Chao, Y. C. Cao, and Y. Q. Chen, Autopilots for small unmanned aerial vehicles: a survey, International Journal of ontrl, Automation, and Systems, vol. 8, no. 1, pp. 36-44, 2010..

[2] D. Lee, I. Kaminer, V. Dobrokhodov, K. Jones, Autonomous feature following for visual surveillance using a small unmanned aerial vehicle with gimbaled camera system, International Journal of Control, Automation, and Systems, vol. 8, no. 5, pp. 957-966, 2010.

- [3] P. Pounds, R. Mahony, P. Corke, Modelling and Control of a QuadRotor Robot, In Proceedings of the Australasian conference on robotics and automation, December 2006.
- [4] Andreas Vikane Hystad, Joakim Brobakk Lehn. "Model, Design and Control of a Quadcopter." Master Thesis in Cybernetics and Robotics, Norwegian University of Science and Technology Department of Engineering Cybernetics, 2015.
- [5] Gembong Edhi Setyawan, Eko Setiawan, Wijaya Kurniawan. "Sistem Kendali Ketinggian Quadcopter Menggunakan PID". Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK), Vol. 2, No. 2, hlm. 125-131, Oktober 2015.
- [6] S. Bouabdallah, P. Murrieri, R. Siegwart, Towards Autonomous Indoor Micro VTOL, Autonomous Robots 18, 171-183, 2009.
- [7] S. Bouabdallah, P. Murrieri, R. Siegwart, Towards Autonomous Indoor Micro VTOL, Autonomous Robots 18, 171-183, 2009.
- [8] L. Gentili, R. Naldi, and L. Marconi, Modelling and control of VTOL UAVs interacting with the environment, IEEE Conf. on Decision and Control, pp. 1231-1236, 2008.
- [9] R. Naldi, L. Marconi, Modeling and control of the interaction between flying robots and the environment, Proc. of the 2010 IFAC NOLCOS, 2010.
- [10] M. Ryll, M. Heinrich, P. Robuffo Giordano, Modeling and Control of a Quadrotor UAV with Tilting Propellers, IEEE International Conference on Robotics and Automation RiverCentre, USA May 14-18, 2012.
- [11] F. Senkul, E. Altu G, Modeling and Control of a Novel Tilt Roll Rotor Quadrotor UAV, 2013 International Conference on Unmanned Aircraft Systems (ICUAS) May 28-31, 2013.