Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer e-ISSN: 2548-964X
Vol. 2, No. 9, September 2018, hlm. 3244-3250 http://j-ptiik.ub.ac.id

Stabilizer Kamera 2-*Axis* Dengan Pid *Control* Berdasarkan *Setpoint* pada Atmega 328

Adryan Chiko Pratama[1], Dahnial Syauqy[2], Mochammad Hannats Hanafi Ichsan[3]

Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Brawijaya
Email: [1]adryanchiko21@gmail.com, [2]dahnial87@ub.ac.id, [3]hanas.hanafi@ub.ac.id

Abstrak

Kemajuan perkembangan teknologi saat ini berpengaruh pada kehidupan manusia, salah satunya teknologi kamera. Saat ini kamera memiliki teknologi yang berfungsi untuk mengurangi getaran saat pengambilan gambar yang disebut *Image Stabilization*, tetapi teknologi ini kurang menghasilkan gambar yang presisi oleh karena itu dibutuhkan *stabilizer*. Diharapkan dengan adanya penelitian ini dapat membantu masalah kemiringan dalam pengambilan gambar dari kamera *smartphone*. Pada penelitian ini digunakan sebuah sensor *gyroscope* dan *accelerometer* yang berada pada satu unit MPU-6050 yang dapat dibaca nilainya menggunakan mikrokontroler Arduino Nano. Nilai sensor masuk dalam perhitungan PID *Control* yang variabel Kp, Ki dan Kd-nya dicari menggunakan metode *trial and error.* Berdasarkan pengujian yang telah dilakukan, penelitian menghasilkan kesimpulan yaitu tingkat akurasi nilai sensor MPU-6050 terhadap busur rata-rata selisih *error* sistem 0,4° dan presentase *error* sistem adalah 0,25%. Selisih atau nilai *error* yang dihasilkan karena akurasi pengukuran menggunakan busur derajat adalah dibawah 1°. Pada pengujian ketepatan sudut servo terhadap *setpoint* didapatkan rata-rata presentase *error* kedua sudut servo mencapai 0,17% dengan besar *error* derajat 0,24°. Kemudian pada pengujian kecepatan respon *stabilizer* mencapai stabil (*settling time*) didapati rata-rata kecepatan mencapai stabil sistem yaitu 0,92 detik.

Kata kunci: *stabilizer*, kamera, PID *control*, MPU-6050

Abstract

Advances in technological developments currently affect human life, one of which is camera technology. Currently the camera has a technology that serves to reduce vibration when shooting called Image Stabilization, but this technology produces less precise images therefore required stabilizer. It is expected that with this research can help the problem of tilted in shooting from smartphone camera. In this research used a sensor gyroscope and accelerometer located on one unit of MPU-6050 which can be read value using Arduino Nano microcontroller. The sensor value include in the calculation of PID Control which variables Kp, Ki and Kd are searched using trial and error method. Based on the test that has been done, the research have the conclusion that the accuracy level of MPU-6050 sensor value to the average arc of system error difference 0,4 ° and percentage of system error is 0,25%. The difference or the resulting error value due to measurement accuracy using a protractor is below 1 °. In testing the accuracy of the servo angle to the setpoint obtained the average percentage error of both servo angles reached 0.17% with a large error of degree 0.24 °. Then on testing the stabilizer response speed to achieve stable (settling time) found the average speed reached the stable system that is 0.92 seconds.

Keywords: *stabilizer, camera, PID control, MPU-6050*

1. PENDAHULUAN

Segala aspek kehidupan manusia saat ini dipengaruhi oleh perkembangan teknologi yang pesat dan kamera saat ini merupakan salah satu teknologi yang kian maju. Teknologi kamera pun juga sangat maju salah satunya terdapat *Optical Image Stabilizer (OIS)* adalah sebuah solusi efektif yang mengurangi getaran saat pengambilan gambar, dan merupakan ide yang telah ada sejak 30 tahun yang lalu (Sachs, Nasiri dan Goehl, 2010). Akan tetapi teknologi *IS* pun masih kurang cukup untuk menghasilkan video

yang presisi atau tidak miring oleh karena itu diperlukan sebuah *stabilizer* untuk mengatasinya.

UAV dan robotika kemajuan teknologinya juga kian maju seiring dengan dibutuhkannya kedua teknologi tersebut dalam masa kini. UAV pada umumnya digunakan untuk monitoring atau memantau suatu tempat pada ketinggian tertentu baik diremote atau otomatis. Robotika digunakan untuk menggantikan kerja manusia seperti robot lengan yang beroperasi secara otomatis atau manual. Kedua teknologi ini memiliki persamaan yaitu membutuhkan motor servo sebagai aktuator. Indikator keberhasilannya ditentukan pada akurasi dan minimnya error pergerakan motor servo. Untuk membantu meningkatkan akurasi dan meminimalkan error dapat menggunakan kontroler PID. PID biasa disebut pengendali tiga tingkat yang terdiri dari 3 koefisien dasar yaitu *proportional*, *integral* dan *derivative* (Johnson dan Moradi, 2005).

Dalam menentukan nilai PID bergantung pada jenis plant yang digunakan, dimana plant merupakan perangkat keras yang dikontrol. Misal pada motor servo, perbedaan jenis motor servo juga mempengaruhi nilai PID untuk mendapatkan hasil akurasi yang terbaik dan meminimalkan error. Salah satu kekuatan PID adalah untuk jenis *plant* sederhana ada korelasi langsung antara respon *plant*, penggunaan dan penyesuaian dari tiga istilah kontroler (Johnson dan Moradi, 2005). Salah satu cara yang sering digunakan untuk mendapatkan nilai PID terbaik adalah menggunakan metode *trial and error* dengan mengubah-ubah nilai PID pada *coding* dan dicoba pada *hardware* hingga mendapatkan hasil yang terbaik. Dari berbagai riset mengenai PID ini yang mempengaruhi nilai PID adalah jenis plant. Selain jenis plant yang digunakan, mikrokontroler juga dapat mempengaruhi nilai PID dengan mengesampingkan berbagai macam kebutuhan sistem terhadap mikrokontroler. Tiap mikrokontroler memiliki kecepatan proses, cara memproses *coding* dan juga pin-pin yang berbeda.

Pada tugas akhir ini dilakukan perancangan hingga analisis *stabilizer* kamera 2-*axis* menggunakan PID *control* berdasarkan *setpoint* pada ATMEGA 328. Untuk mendapatkan nilai kemiringan alat digunakan sensor *gyroscope* dan *accelerometer*. Terdapat motor servo yang digunakan untuk mengubah sudut dari masing-masing *axis* serta *potentiometer* yang digunakan untuk menentukan *setpoint* dari masing-masing *axis*. Untuk pemroses yang digunakan adalah *board* arduino nano. Selain lebih mudah digunakan dalam melakukan pemrograman pada ATMEGA 328 juga dikarenakan *board* ini sering digunakan dalam penerapan *embedded system.*

2. PERHITUNGAN PID *CONTROL*

Perhitungan PID *control* dibagi menjadi 3 bagian yaitu perhitungan variabel P, I dan D yang kemudian hasil dari ketiganya dijumlahkan menjadikan nilai *output* PID.

2.1 Metode PID *Control*

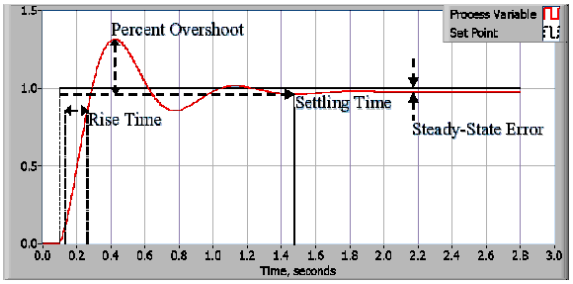
PID (*Proportional*, *Integral* dan *Derivative*) merupakan salah satu metode yang digunakan untuk memaksimalkan hasil output dari pengolahan sinyal (Instruments, 2011). Pada metode ini terdapat tiga aksi kontrol yaitu aksi kontrol *proportional* dapat meningkatkan kinerja secara maksimum dalam waktu yang cepat. Aksi kontrol *integral* memiliki kemampuan untuk memperkecil terjadinya error dari aksi *proportional*. Aksi kontrol *derivative* memiliki keunggulan untuk memperkecil error dan mengurangi terjadinya *overshoot.* Aksi kontrol PID beserta fungsinya dapat dilihat pada Tabel 1 dan Untuk grafik idel PID dapat dilihat pada Gambar 1.

Tabel 1. Fungsi Aksi Kontrol pada PID

	Rise Time	Overshoot	Settling Time	SSE
Kp	Berkurang	Bertambah	Sedikit bertambah	Menurun
Ki	Sedikit Berkurang	Bertambah	Bertambah	Menurun
Kd	Sedikit Berkurang	Bertambah	Berkurang	Menurun

Pada Tabel 1 SSE kepanjangan dari Steady-State Error. Dalam bentuk idealnya, parameter tiap aksi kontrol yang nampak pada Tabel 1 diatas dapat dijelaskan sebagai berikut :

- **Rise time** : waktu yang diperlukan respon dari 0 hingga mencapai 100% (set point)
- **Overshoot** : lonjakan maksimum yang dihasilkan oleh respon
- **Settling time** : waktu yang diperlukan respon untuk stabil antara 95% - 98% dari set point



Gambar 1. Grafik Ideal dari PID

Proportional

Proposional menghasilkan nilai yang berbanding lurus dengan nilai kesalahan. Repsonsenya dapat diatur dengan mengalikan kesalahan (error) dengan konstanta Kp disebut konstanta gain proposional. Proposional dirumuskan:

$$Pout = Kp\ e(t) \quad (1)$$

Integral

Integral berbanding lurus dengan besar dan lamanya error. Integral dalam kontroler PID adalah jumlahan error setiap waktu dan mengakumulasi offset yang sebelumnya telah dikoreksi. Error terakumulasi dikalikan dengan gain integral dan menjadi keluaran kontroler. Integral dirumuskan sebagai berikut :

$$Iout = Ki \int_0^t e\,(t)dt \quad (2)$$

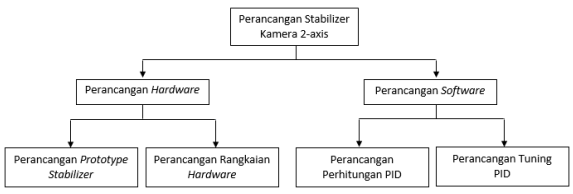
Derivative

Derivative ini memprediksi perilaku sistem dan kemudian memperbaiki waktu tinggal dan stabilitas sistem. Derivative bergantung pada perubahan error yang terjadi pada sistem jadi ketika ada statis error maka derivative tidak akan bekerja. Derivative dirumuskan sebagai berikut :

$$Dout = Kd\ \frac{de(t)}{dt} \quad (3)$$

3. PERANCANGAN DAN IMPLEMENTASI

Perancangan pada penelitian ini dibagi menjadi 2 bagian yaitu perancangan perangkat keras dan perancangan perangkat lunak. Bagan perancangan dapat dilihat pada Gambar 2.

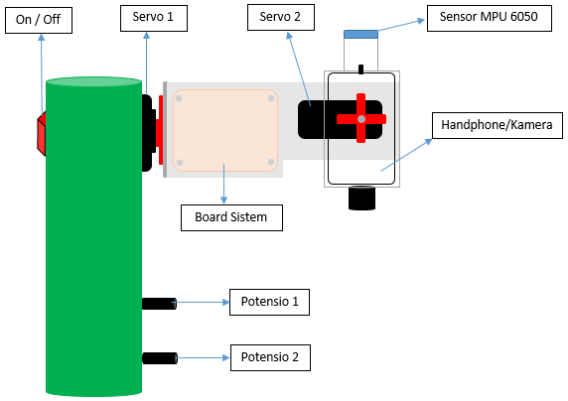


Gambar 2. Perancangan *Stabilizer* Kamera

Pada gambar 2 terlihat bahwa dalam perancangan *hardware* terbagi menjadi 2 yaitu perancangan *prototype* dan perancangan rangkaian *hardware*, sedangkan pada perancangan *software* dibagi menjadi 2 yaitu perancangan perhitungan PID dan perancangan *tuning* PID.

3.1 Perancangan *Prototype Stabilizer*

Dalam Perancangan *prototype stabilizer* terbagi menjadi 2 yaitu untuk pegangan pengguna dan bagian perangkat penggerak. Untuk bagian pegangan menggunakan bahan plastik karena ringan dan nyaman dipegang sedangkan untuk bagian perangkat penggerak digunakan bahan akrilik dengan tebal 0.4mm. Penggunaan bahan akrilik jenis ini karena ringan dan kuat. perancangan *prototype stabilizer* dapat dilihat pada Gambar 3.



Gambar 3. Perancangan *Prototype Stabilizer*

Pada Gambar 3 dapat dilihat potensio diletakkan pada kanan bawah pegangan agar tidak pengguna nyaman memegang *stabilizer*. Sensor MPU6050 diletakkan pada bagian penggerak *horizontal* ini bertujuan agar sensor dapat membaca nilai perputaran sudut *pitch* dan *roll* bersamaan.

3.2 Perancangan Rangkaian *Hardware*

Pada perancangan rangkaian *hardware stabilizer* ini terdapat 2 buah servo sebagai penggerak sudut *pitch* dan *roll*. Terdapat juga 2 buah potensio untuk mengubah *setpoint* masing-masing sudut sesuai keinginan. Dimana semua *hardware* terhubung ke satu *board* utama yaitu arduino nano dengan sumber daya 9 Volt. Untuk konfigurasi pin yang digunakan dapat dilihat pada Tabel 2, 3, 4, 5 dan 6.

Tabel 2. Konfigurasi Sensor MPU-6050 dengan Arduino Nano

MPU 6050	Arduino Nano
VCC	3.3 Volt
GND	GND
SCL	A5
SDA	A4
INT	D2

Pada Tabel 2 merupakan konfigurasi sensor MPU6050 dengan arduino nano dimana sumber tegangan MPU6050 didapat dari output tegangan 3.3 Volt arduino nano. SCL terhubung dengan pin Analog 5 arduino. SDA terhubung pin Analog 4 Arduino dan INT terhubung dengan pin Digital 2 arduino.

Tabel 3. Konfigurasi Servo *Roll* dengan Arduino Nano

Servo Roll	Arduino Nano
VCC	5 Volt
GND	GND
DATA	D9

Pada Tabel 3 merupakan konfigurasi servo penggerak sudut *Roll* dengan Arduino nano. Sumber tegangan servo didapat dari output tegangan 5 Volt arduino nano. Data servo terhubung dengan pin Digital 9 arduino.

Tabel 4. Konfigurasi Servo *Pitch* dengan Arduino Nano

Servo Pitch	Arduino Nano
VCC	5 Volt
GND	GND
DATA	D10

Pada Tabel 4 merupakan konfigurasi servo penggerak sudut *Pitch* dengan Arduino nano. Sumber tegangan servo didapat dari output tegangan 5 Volt arduino nano. Data servo terhubung dengan pin Digital 10 arduino.

Tabel 5. Konfigurasi Potensio *Roll* dengan Arduino Nano

Potensio Roll	Arduino Nano
VCC	5 Volt
GND	GND
DATA	A0

Pada Tabel 5 merupakan konfigurasi pin potensio yang mengatur *setpoint* sudut *Roll* dengan Arduino nano. Sumber tegangan potensio didapat dari output tegangan 5 Volt arduino nano. Data servo terhubung dengan pin Analog 0 arduino.

Tabel 6. Konfigurasi Potensio *Pitch* dengan Arduino Nano

Potensio Pitch	Arduino Nano
VCC	5 Volt
GND	GND
DATA	A1

Pada Tabel 6 merupakan konfigurasi pin potensio yang mengatur *setpoint* sudut *Pitch* dengan Arduino nano. Sumber tegangan potensio didapat dari output tegangan 5 Volt arduino nano. Data servo terhubung dengan pin Analog 1 arduino.

3.3 Perancangan Perhitungan PID

Perhitungan PID telah dibahas pada bagian 2 beserta tabel sifat masing-masing P, I dan D yang hasil akhirnya atau outputnya adalah menjumlahkan hasil dari masing-masing variabel tersebut. PID membutuhkan 6 variabel antara lain Kp, Ki, Kd, *Input*, *Setpoint* dan *Output*. Untuk Input didapat dari nilai pitch dan roll dari sensor MPU-6050. *Setpoint* didapat dari nilai potensio *roll* dan *pitch*. Untuk pencarian nilai Kp, Ki dan Kd terdapat beberapa metode yang ada tetapi peneliti menggunakan metode - *trial and error* dimana mencoba nilai pernilai hingga didapat hasil yang sesuai.

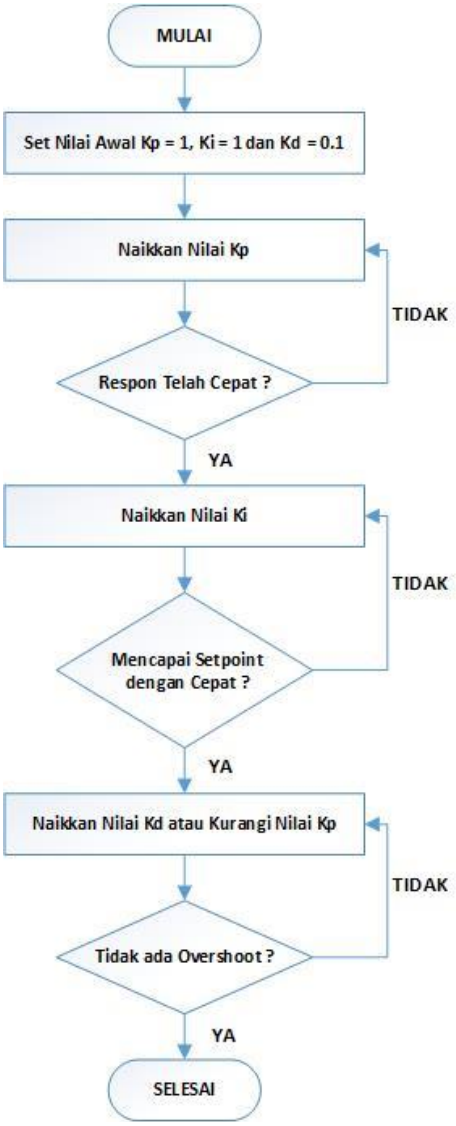
Perhitungan PID didapat dengan mencari dahulu perhitungan tiap variabel P (lihat Rumus 1), I (lihat Rumus 2), dan D (lihat Rumus 3). Setelah itu didapat nilai *output* PID yaitu penjumlahan variabel P + I + D.

3.4 Perancangan *Tuning* PID

Pada perancangan *tuning* PID peneliti menggunakan metode *trial and error*. Untuk urutan pengerjaan *tuning* menggunakan metode ini dapat dilihat pada Gambar 4.

Dalam Gambar 4 dapat dijelaskan alur untuk melakukan *tuning* PID pada nilai Kp, Ki dan Kd. Pada dasarnya penjelasan mengenai 3 variabel tersebut dapat dilihat pada bab 2. Pertama set nilai awal Kp = 1, Ki = 1 dan Kd = 0,1 setelah itu upload coding dan lihat grafik apakah respon sudah cepat atau sesuai yang diinginkan, jika tidak maka tambahkan nilai Kp jika telah sesuai maka lanjut step berikutnya. Menaikkan nilai Kp yang berlebihan dapat membuat sistem tidka stabil. Kedua naikkan nilai Ki hingga perubahan nilai menuju setpoint

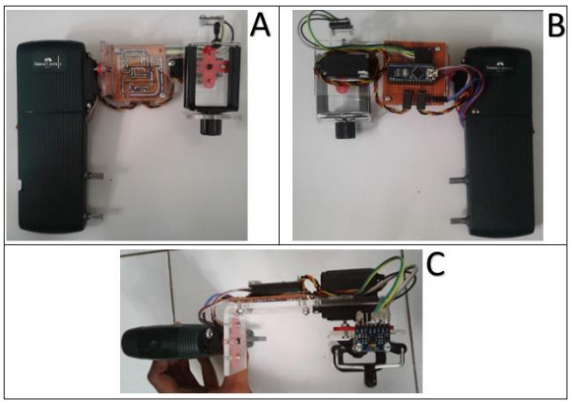
lebih cepat. Apabila kecepatan menuju setpoint belum sesuai yang diinginkan maka naikkan lagi nilai Ki, jika telah sesuai maka lanjut ke step berikutnya. Menaikkan nilai Ki dapat menimbulkan *overshoot* pada sistem. Ketiga menaikkan nilai Kd untuk mengurangi *overshoot.* Naikkan nilai Kd hingga *overshoot* berkurang hingga mendekati *setpoint*, apabila nilai Kd telah dinaikkan semakin membuat sistem tidak stabil maka kurangi nilai Kp hingga *overshoot* mendekati *setpoint.*



Gambar 4. *Flowchart* Metode *Trial and Error*

3.5 Implementasi *Stabilizer* Kamera 2-axis

Pada perancangan *stabilizer* menggunakan bahan plastik sebagai pegangan dan akrilik sebagai tempat penggerak 2 buah servo, arduino nano dan 2 buah potensio dengan dimensi yang telah diperhitungkan. Sesuai dengan perancangan dalam implementasi keseluruhan sama dengan apa yang ada dalam perancangan. Hasil dari implementasi *stabilizer* kamera yang telah dilakukan dapat dilihat pada Gambar 5.



Gambar 5. Implementasi *Stabilizer* Kamera

Dapat dilihat pada Gambar 5 bagian A merupakan implementasi *stabilizer* tampak depan sedangkan bagian B merupakan bagian belakangnya. Untuk membedakan bagian depan dan belakang dapat dilihat penjepit dari *handphone*/kamera menghadap kearah pengguna. Pada bagian C *stabilizer* tampak atas dimana sensor MPU-6050 letaknya juga diatas.

4. PENGUJIAN DAN ANALISIS

Pengujian dan analisis pada penelitian ini dibagi menjadi 3 variabel yaitu ketepatan sudut sensor MPU-6050, ketepatan sudut servo berdasarkan setpoint dan kecepatan sistem mencapai stabil di *setpoint* (*settling time*).

4.1 Pengujian Sudut Sensor MPU-6050

Pengujian ini dilakukan untuk mengetahui tingkat akurasi ketepatan sudut sensor pada sumbu *roll* dan *pitch.* Sebagai acuan menggunakan sudut terukur dengan busur dan sensor harus memiliki nilai yang sama atau mendekati nilai sudut terukur untuk mendapat akurasi yang tinggi. Skenario untuk pengujian sudut sensor MPU-6050 dilakukan dengan membandingkan hasil dari dua nilai yaitu *output* nilai sudut sensor pada masing-masing sumbu dengan nilai sudut terukur menggunakan busur derajat. Pengujian dilakukan sebanyak 7 kali pada masing-masing sumbu. Hasil pengujian dapat dilihat pada Tabel 7 dan 8.

Dengan hasil pengujian pada Tabel 7 dan 8 maka didapati rata-rata selisih *error* sistem 0,4° dan presentase *error* sistem adalah 0,25% yang didapat dengan menjumlahkan seluruh presentase *error* kemudian dibagi dengan banyaknya jumlah data. Selisih atau nilai *error*

yang dihasilkan karena akurasi pengukuran menggunakan busur derajat adalah dibawah 1°.

Tabel 7. Hasil Pengujian Sudut Sensor MPU-6050 pada Sumbu *Roll*

Sudut Terukur	Sudut *Roll*	Selisih	Presentase *Error*
0° atau 360°	0,03°	0,03°	0,01%
30°	30,02°	0,02°	0,07%
60°	60,27°	0,27°	0,45%
90°	88,67°	0,33°	0,37%
270°	270,83°	0,83°	0,31%
300°	299,5°	0,5°	0,17%
330°	328,8°	1,2°	0,36%

Tabel 8. Hasil Pengujian Sudut Sensor MPU-6050 pada Sumbu *Pitch*

Sudut Terukur	Sudut *Roll*	Selisih	Presentase *Error*
0° atau 360°	0,1°	0,1°	0,03%
30°	30,1°	0,1°	0,33%
60°	60,35°	0,35°	0,58%
90°	89,73°	0,27°	0,30%
270°	270,28°	0,28°	0,1%
300°	300,71°	0,71°	0,24%
330°	329,33°	0,67°	0,2%

4.2 Pengujian Ketepatan Sudut Servo Berdasarkan *Setpoint*

Dalam pengujian ketepatan sudut servo berdasarkan *setpoint* dilakukan dengan memutar *potensiometer* untuk mengubah nilai sudut pada masing-masing sumbu kemudian hasinya dilihat pada *serial monitor* arduino dan dibandingkan dengan nilai *output* sensor pada masing-masing sumbu. Dari nilai *setpoint* yang diberikan akan membuat sistem merubah sudutnya menuju setpoint. Nilai dari sudut yang terbaca oleh sensor harus sama atau mendekati dengan nilai sudut *setpoint*. Dalam pengujian ini dilakukan sebanyak 7 kali pada masing-masing sumbu, untuk hasilnya dapat dilihat pada Tabel 9 dan 10.

Dari Tabel 9 dan 10 didapatkan rata-rata presentase *error* kedua sudut servo mencapai *setpoint* mencapai 0,17% dengan besar *error* derajat 0,24°. Nilai rata-rata ini didapat dengan mengabaikan sudut yang memiliki presentase *error* diatas 1% ini dikarenakan *error* yang dihasilkan sudut tersebut bukan karena ketidakmampuan sensor dan servo tapi dikarenakan desain dari *stabilizer*.

Tabel 9. Hasil Ketepatan Sudut Servo Berdasarkan *Setpoint* pada Sumbu *Roll*

Setpoint	Sudut *Roll*	Selisih	Presentase *Error*
0° atau 360°	0,05°	0,05°	0,01%
30°	30,07°	0,07°	0,23%
60°	60,04°	0,04°	0,07%
90°	89,44°	0,56°	0,62%
270°	281,88°	11,88°	4,4%
300°	299,5°	0,5°	0,17%
330°	329,8°	0,2°	0,06%

Tabel 10. Hasil Ketepatan Sudut Servo Berdasarkan *Setpoint* pada Sumbu *Pitch*

Setpoint	Sudut *Roll*	Selisih	Presentase *Error*
0° atau 360°	0,05°	0,05°	0,01%
30°	30,03°	0,03°	0,1%
60°	60,24°	0,24°	0,4%
90°	89,55°	0,45°	0,5%
270°	281,77°	11,77°	4,36%
300°	299,88°	0,12°	0,04%
330°	329,5°	0,5°	0,15%

4.3 Pengujian Kecepatan Sistem Mencapai Stabil di *Setpoint* (*settling time)*

Pengujian kecepatan sistem mencapai stabil di setpoint dilakukan dengan melihat grafik perubahan sudut masing-masing servo saat pertama kali sistem dinyalakan. Kecepatan sistem mencapai stabil ini biasa disebut kecepatan *settling time*. Stabil atau *settling time* dalam hal ini hanya selisih 2% dari nilai teratas sumbu pada grafik dengan *setpoint.* Stabil atau *settling time* dalam hal ini hanya selisih 2% dari nilai teratas sumbu pada grafik dengan *setpoint*. Untuk mendapatkan nilai kecepatan dalam waktu dengan satuan miliseconds ini digunakan *serial monitor*. Settling time didapat dari selisih waktu akhir mencapai minimal 2% dengan waktu awal sistem mengeksekusi *code* program baris *code* pengolahan data. Percobaan bagian ini dilakukan sebanyak 10 kali pada masing-masing sumbu untuk dapat ditarik kesimpulan.

Tabel 11. Hasil Pengujian *Settling Time* Sumbu *Roll*

Percobaan	Millis Awal (ms)	Millis Akhir (ms)	*Settling Time* (ms)
1	2447	3367	920
2	2446	3337	891

3	2446	3367	921
4	2445	3342	897
5	2446	3352	906
6	2446	3458	1012
7	2446	3472	1027
8	2443	3328	895
9	2446	3382	936
10	2446	3367	921
	Rata-rata		932,6

Tabel 12. Hasil Pengujian *Settling Time* Sumbu *Pitch*

Percobaan	Millis Awal (ms)	Millis Akhir (ms)	*Settling Time* (ms)
1	2433	3357	924
2	2433	3342	909
3	2446	3397	951
4	2446	3337	891
5	2433	3373	940
6	2433	3312	879
7	2446	3322	876
9	2446	3412	966
10	2446	3427	981
	Rata-rata		924,1

Dari hasil pengujian kecepatan sistem mencapai stabil di *setpoint* (*settling time)* didapati rata-rata kecepatan mencapai stabil sistem yaitu 0,92 detik. Waktu mulai sistem masuk dalam fungsi utama juga berbeda antara 2443 ms sampai 2447, waktu tersebut merupakan waktu program memproses *setup* dan waktu sistem menunggu sensor MPU-6050 stabil.

5. KESIMPULAN

Berdasarkan pada tahap perancangan, implementasi serta analisis dari pengujian yang telah dilakukan dapat ditarik kesimpulan antara lain :

1. Perancangan *stabilizer* kamera 2-*axis* dengan PID berdasarkan *setpoint* dilakukan perhitungan PID secara terstruktur sesuai yang ada pada landasan teori yang dimana masukkannya berupa nilai sensor dan nilai *tuning* Kp, Ki dan Kd dengan metode *trial and error*, kemudian setelah didapat nilai tuning yang baik maka diimplementasikan dalam bentuk kode pemrograman C arduino dan di unggah ke arduino nano.
2. Berdasarkan analisis pada hasil pengujian keakuratan sudut sensor MPU-6050 yang dilakukan dapat diketahui bahwa dari 7 kali percobaan dari sudut 270° sampai 90° dengan kelipatan 30° pada kedua sumbu yaitu *roll* dan *pitch* didapatkan presentase *error* sebesar 0,25% atau kurang dari 0,5% sehingga tingkat keakuratan sudut sensor MPU-6050 mencapai 99,5%. Dengan akurasi yang tinggi sebesar 99,5% maka sensor dapat menghasilkan sudut yang sesuai dengan pengukuran nyata.
3. Berdasarkan analisis pada hasil pengujian keakuratan sudut servo berdasarkan *setpoint* yang dilakukan dapat diketahui bahwa dari 7 kali percobaan dari sudut 270° sampai 90° dengan kelipatan 30° pada kedua sumbu yaitu *roll* dan *pitch* didapatkan presentase *error* sebesar 0,17% atau kurang dari 0,5% sehingga tingkat keakuratan sudut servo terhadap *setpoint* mencapai 99,5%. Ketika pada sudut diatas 85° dan dibawah 285° maka akan terjadi *error* sudut diatas 1° ini dikarenakan desain pemasangan servo yang kurang tegak lurus. Tetapi selain sudut tertentu itu dapat dikatakan tingkat akurasi antara sudut servo dengan *setpoint* sangat tinggi yaitu mencapai 99,5%.
4. Berdasarkan analisis pada hasil pengujian kecepatan sistem mencapai stabil atau biasa disebut *settling time* yang telah dilakukan dapat diketahui bahwa dari 10 kali percobaan didapatkan kecepatan awal sistem mencapai *settling time* yaitu rata-rata waktu kedua sumbu yaitu 0,92 detik atau dibawah 1 detik. Dengan kecepatan stabil mencapai dibawah 1 detik maka sistem dapat dikatakan memiliki repson yang cepat dan kecepatan yang tinggi.

DAFTAR PUSTAKA

Al-Mashhadani, M. A., 2013. Optimal and PID Controller for Controlling Camera's Position in Unmanned Aerial Vehicles. Volume I.

Arduino, 2017. *store.arduino.cc*. [Online] Available at: https://store.arduino.cc/usa/arduino-nano [Diakses 10 November 2017].

Astrom, K. J. & Hagglund, T., 1995. PID Controllers: Theory, Design, and Tuning. Dalam: *PID Controllers: Theory, Design, and Tuning.* North Carolina: Instrument Society of America.

Elektroindonesia., 2017. *Blok Diagram PID Controller*. [Online] Available at: http://www.elektroindonesia.com/elektro/tutor12.html [Diakses 10 November 2017]

Instruments, N., 2011. *NI.com.* [Online] Available at: http://www.ni.com/white-paper/3782/en/ [Diakses 10 November 2017].

Johnson, M. A. & Moradi, M. H., 2005. *PID Control*. London: Springer Verlag.

Pitowarno, E., 2005. *Mikroprosesor & Interfacing.* Yogyakarta: ANDI Yogyakarta.

Priyambodo, T. K., 2017. Implementasi Sistem Kendali PID pada Gimbal Kamera 2-sumbu dengan Aktuator Motor Brushless. *Implementasi Sistem Kendali PID pada Gimbal Kamera 2-sumbu dengan Aktuator Motor Brushless,* Volume VII, pp. 117-126.

Sachs, D., Nasiri, S. & Goehl, D., t.thn. Image Stabilization Technology Overview.

Setiawan, I., 2008. *Kontrol PID untuk Proses Industri.* Jakarta: PT Elex Media Komputindo.