

# Perancangan Sistem Stabilisasi Kamera Tiga Sumbu dengan Metode Kontrol *Fuzzy* untuk *Mobile Surveillance Robot*

Fahrezi Alwi Muhammad, Muhammad Rivai, dan Suwito

Jurusan Teknik Elektro, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember (ITS)

Jl. Arief Rahman Hakim, Surabaya

*e-mail:* muhammad\_rivai@ee.its.ac.id, masaji@elect-eng.its.ac.id, fahrezialwi@gmail.com

**Abstrak**—Robot pengintaian (*surveillance robot*) sering digunakan pada militer untuk melakukan tugas pengintaian sehingga tidak perlu membahayakan nyawa manusia karena dikendalikan dari jarak jauh. Robot pengintai memiliki kamera yang diletakkan di atas robot. Ketika melakukan pengamatan menggunakan kamera ini, seringkali pengamatan terganggu akibat guncangan-guncangan yang terjadi pada kamera. Guncangan ini disebabkan karena permukaan jalan yang dilalui oleh robot tidak rata.

Sistem stabilisasi kamera adalah perangkat yang digunakan untuk menghilangkan guncangan dan menjaga posisi kamera agar kamera dapat mengambil gambar dengan baik pada suatu sudut pandang tertentu. Pada tugas akhir ini, dibuat sebuah sistem stabilisasi untuk robot pengintai dengan dua buah sensor *gyroscope* MPU-6050 untuk mengetahui kecepatan sudut guncangan dan kecepatan sudut kamera, *mikrokontroler Arduino Mega* sebagai pusat kontrol dan tiga buah motor DC brushless sebagai aktuator. Metode kontrol yang ditanamkan pada sistem terdiri dari tiga sistem *Fuzzy* untuk menangani sumbu *pitch*, *roll*, dan *yaw*.

Pada tugas akhir ini diujikan dua metode. Metode pertama menggunakan satu sensor *gyroscope* sebagai nilai *feedback*, metode kedua menggunakan dua sensor *gyroscope* sebagai nilai *feedback* dan *set point*. Standar deviasi pada pengujian perekaman video tanpa kontrol untuk *pixel* x 40.57 dan *pixel* y 32.95. Standar deviasi dengan metode pertama untuk *pixel* x 24.73 adalah dan *pixel* y 21.73, Sedangkan standar deviasi metode kedua untuk *pixel* x 16.70 dan *pixel* y 22.44.

**Kata Kunci**—*Fuzzy*, gimbal, motor brushless DC, sensor *gyroscope*

## I. PENDAHULUAN

PERKEMBANGAN teknologi robot yang cepat saat ini menjadikan kualitas kehidupan manusia semakin baik. Robot adalah sebuah mesin yang dapat bekerja secara terus-menerus baik secara otomatis maupun terkendali [1]. Robot digunakan untuk membantu manusia dalam mengerjakan hal-hal yang sulit sehingga dapat menghemat waktu atau yang tidak bisa dilakukan manusia secara langsung. Sebagai contoh, robot digunakan dalam industri perakitan kendaraan, industri makanan, industri tekstil, hingga industri militer. Salah satu robot yang saat ini sering digunakan oleh militer adalah robot pengintai (*surveillance robot*). Dengan robot ini, para prajurit tidak perlu mengintai suatu lokasi yang berbahaya secara langsung karena robot ini dikendalikan dari jarak jauh. Robot ini digunakan untuk melakukan tugas pengintaian sehingga tidak perlu membahayakan nyawa manusia. Robot pengintai memiliki kamera yang diletakkan di atas robot. Kamera tersebut

akan mentransmisikan video secara langsung ke pusat pengendali sehingga pengguna dapat melakukan pengamatan di suatu tempat.

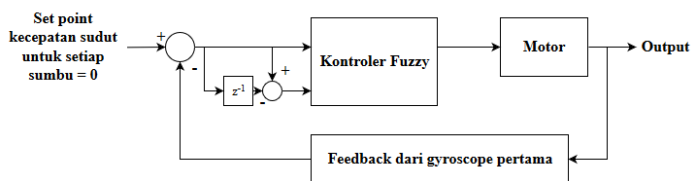
Ketika melakukan pengamatan menggunakan kamera jarak jauh ini, seringkali pengamatan terganggu akibat guncangan-guncangan yang terjadi pada kamera. Guncangan ini disebabkan karena permukaan jalan yang dilalui oleh robot tidak rata. Pada penelitian tugas akhir ini muncul suatu ide untuk membuat sistem stabilisasi kamera tiga sumbu yang menggunakan *Fuzzy* sebagai metode kontrolnya. Selama ini, metode pengontrolan pada aplikasi sistem stabilisasi kamera menggunakan kontrol PID. Penelitian ini bertujuan untuk mengetahui apakah metode *Fuzzy* cocok untuk diimplementasikan dalam sistem stabilisasi kamera. Sistem ini membaca dua buah sensor *gyroscope* untuk mengetahui kecepatan sudut guncangan dan kecepatan sudut kamera, dan tiga buah motor DC brushless sebagai aktuator.

## II. METODE PENELITIAN

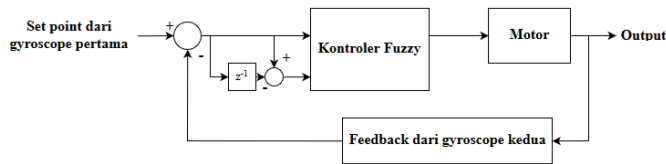
### A. Diagram Blok

Pada metode pertama, nilai *set point* akan dikurangkan dengan nilai *feedback* yang berasal dari *gyroscope* yang terpasang pada *body* bawah mobil RC. Hasil pengurangan ini akan menjadi sinyal *error*. Masukan dari *kontroler Fuzzy* yang dirancang adalah sinyal *error* dan sinyal *delta error*. Sinyal *delta error* merupakan sinyal *error* yang sekarang dikurangi sinyal *error* yang sebelumnya. Kemudian sinyal yang keluar dari *kontroler* adalah sembilan buah sinyal PWM yang berfungsi untuk menggerakkan ketiga motor.

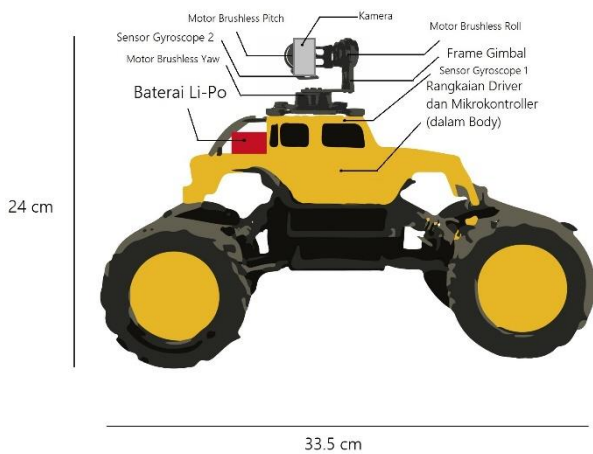
Sedangkan pada metode kedua, sensor *gyroscope* kedua dipasang di bawah dudukan kamera berfungsi memberi nilai *set point* yang memiliki besaran derajat/detik. Lalu nilai *set point* ini akan dikurangkan dengan nilai *feedback* yang berasal dari *gyroscope* pertama yang terpasang pada *body* bawah mobil RC. Hasil pengurangan ini akan menjadi sinyal *error*. Masukan dari *kontroler Fuzzy* yang dirancang adalah sinyal *error* dan sinyal *delta error*. Sinyal *delta error* merupakan sinyal *error* yang sekarang dikurangi sinyal *error* yang sebelumnya. Kemudian sinyal yang keluar dari *kontroler* adalah besar perubahan nilai PWM yang berfungsi untuk menggerakkan ketiga motor.



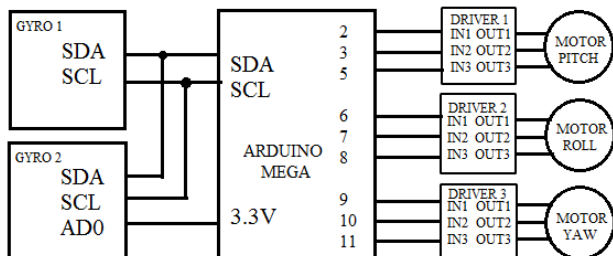
Gambar 1 Diagram Blok Sistem Metode Pertama



Gambar 2 Blok Diagram Sistem Kedua



Gambar 3 Ilustrasi Sistem



Gambar 4 Hubungan Kabel dalam Sistem

**B. Perancangan Perangkat Keras**

Perangkat keras dari sistem merupakan bagian yang terlihat pada sistem. Perangkat keras tersebut terdiri dari dua buah modul sensor gyroscope MPU-6050, mikrokontroler Arduino Mega, driver motor L298N. Selanjutnya mobil RC yang digunakan dalam sistem untuk dijadikan mobil robot adalah Rastar Off-roader Rock Crawler 1:18. Pada sistem ini, frame gimbal dan baterai Li-Po Tiger Power 2200mAh 2 cell 7.4V diletakkan di atas body mobil RC. Frame gimbal yang digunakan adalah frame yang telah jadi dan terdapat di pasaran. Frame ini dijual tanpa merk. Motor brushless DC yang digunakan sebagai aktuator adalah motor brushless gimbal GBM2804.

Sebagai catatan, kamera yang digunakan pada sistem adalah kamera aksi SJCAM SJ5000X Elite. Kamera ini memiliki stabilizer internal sehingga pada penelitian ini pengaturan

stabilizer dimatikan, Kemudian, pegas pada mobil RC yang berfungsi sebagai damper pada roda seluruhnya dilepas.

**C. Pengaturan Register**

Pada MPU-6050 dapat memakai sumber clock internal maupun sumber clock eksternal yang dapat digunakan. Pilihan untuk internal clock adalah MEMS oscillator dari sumbu X, sumbu Y, atau sumbu Z gyroscope. Sedangkan untuk eksternal clock dari 32.768 kHz square wave, dan 19.2 MHz square wave.

Pemilihan sumber clock ini tergantung dari mode operasi sistem MPU-6050 yang digunakan. Internal oscillator baik digunakan ketika menggunakan DMP (Digital Motion Processor) untuk pengolahan data accelerometer dan data gyroscope dimatikan. Sedangkan bila gyroscope aktif dianjurkan menggunakan pemilihan sumber clock dari gyroscope untuk keakuratan [2].

Pemilihan sumber clock untuk MPU-6050 dilakukan dengan mengatur CLKSEL (bit [2:0]) dalam register PWR\_MGMT\_1. Alamat register PWR\_MGMT\_1 pada MPU-6050 adalah 0x6B. Dalam sistem ini, sumber clock yang digunakan adalah sumber clock internal yang bernilai 8MHz, maka dari itu nilai yang diberikan pada register adalah 0x00.

Pengaturan yang dilakukan berkaitan dengan keluaran nilai gyroscope dapat diatur pada register GYRO\_CONFIG dengan alamat register 0x1B.

Opsi FS\_SEL difungsikan untuk memilih skala penuh yang digunakan pada gyroscope. ZG\_ST, YG\_ST, dan XG\_ST digunakan untuk melakukan tes performansi dari masing masing sumbu gyroscope. Dalam sistem yang dibuat pengaturan register hanya dilakukan pada FS\_SEL.

Terdapat empat pilihan skala penuh pada gyroscope. Skala penuh tersebut dapat dipilih dengan memberikan nilai bit ke-0 dan bit ke-1 dari FS\_SEL. Pada sistem yang dibuat, digunakan skala penuh ±250°. Oleh karena itu, register GYRO\_CONFIG diberikan nilai 0x00.

Pengaturan register pada MPU-6050 dilakukan oleh mikrokontroler dengan komunikasi I2C. Perintah yang digunakan adalah perintah tulis. Dalam pengaturan register MPU-6050 perlu diperhatikan alamat dari MPU-6050 tersebut dan alamat register yang akan diatur. Alamat default dari MPU-6050 adalah 0x68, alamat ini dijadikan alamat untuk MPU-6050 pertama sebagai pemberi nilai set point. Sedangkan alamat dari MPU-6050 kedua sebagai pemberi nilai feedback adalah 0x69. Alamat pada MPU-6050 dapat diubah menjadi 0x69 dengan cara memberi tegangan 3.3V pada pin AD0.

Pembacaan sensor gyroscope pada MPU-6050 disimpan pada register data. Perangkat luar dapat meminta data tersebut dengan menunjuk alamat dari register data tersebut. Data gyroscope memiliki lebar data 16-bit yang terdiri dari 8-bit low byte dan 8-bit high byte. Terdapat tiga buah sumbu pada gyroscope sehingga terdapat enam buah register data gyroscope.

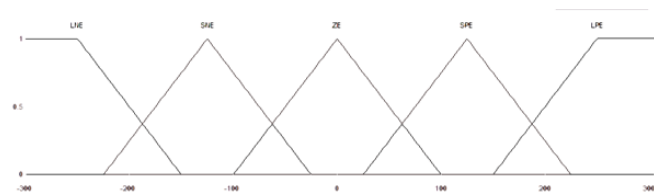
Karena masing-masing sumbu pada gyroscope register datanya terbagi menjadi dua maka untuk mendapatkan data yang valid dari masing-masing sumbu harus dilakukan penggabungan data dari dua buah register tersebut. Ukuran variabel yang disediakan pada program mikrokontroler harus 16-bit bertanda, karena data yang dibaca dapat bernilai positif maupun negatif. Proses penggabungan data dilakukan dengan

cara menggeser data *high byte* ke kiri sebesar 8-bit kemudian dijumlahkan dengan data *low byte*.

$$Data\ sensor = (High\ byte \ll 8) + Low\ byte \quad (1)$$

**D. Perancangan Kontrol Fuzzy**

Perancangan kontrol *Fuzzy* untuk sistem ini adalah menggunakan metode Sugeno. Metode ini dipilih karena ringan untuk dijalankan pada mikrokontroler, perbedaan mendasar dari metode Mamdani adalah, pada metode Sugeno proses *defuzzifikasi*-nya menggunakan *Weighted Average*. Hal inilah yang meringankan perhitungan dibandingkan menggunakan metode Mamdani yang *defuzzifikasi*-nya menggunakan *Center of Gravity/Centroid*, di mana pada proses perhitungannya memerlukan integral [3]. Input pertama pada per-ancangan kontrol menggunakan logika *Fuzzy* adalah nilai *error* dari penjumlahan nilai kedua sensor *gyroscope* di mana *range* nilainya adalah -2000 s.d. 2000.



Gambar 5 Membership Function Error

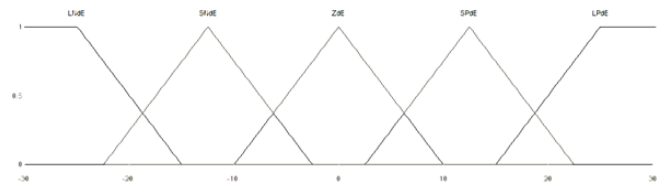
Tabel 1  
Rule Base Sistem Kontrol Fuzzy

	LNdE	SNdE	ZdE	SPdE	LPdE
LNE	LP	LP	MP	SP	SP
SNE	LP	MP	SP	ZM	ZM
ZE	SP	SP	ZM	SN	SN
SPE	ZM	ZM	SN	MN	LN
LPE	SN	SN	MN	LN	LN

Keterangan:

- LNE : Large Negative Error
- SNE : Small Negative Error
- ZE : Zero Error
- SPE : Small Positive Error
- LPE : Large Positive Error
- LNdE : Large Negative Delta Error
- SNdE : Small Negative Delta Error
- ZdE : Zero Delta Error
- SPdE : Small Positive Delta Error
- LPdE : Large Positive Delta Error
- LP : Large Positive
- MP : Medium Positive
- SP : Small Positive
- ZM : Zero Movement
- SN : Small Negative
- MN : Medium Positive
- LN : Large Negative

Sedangkan untuk input kedua adalah *delta error*, yakni nilai *error* sekarang dikurangi nilai *error* sebelumnya yang ditetapkan *range*-nya adalah -200 sampai 200.

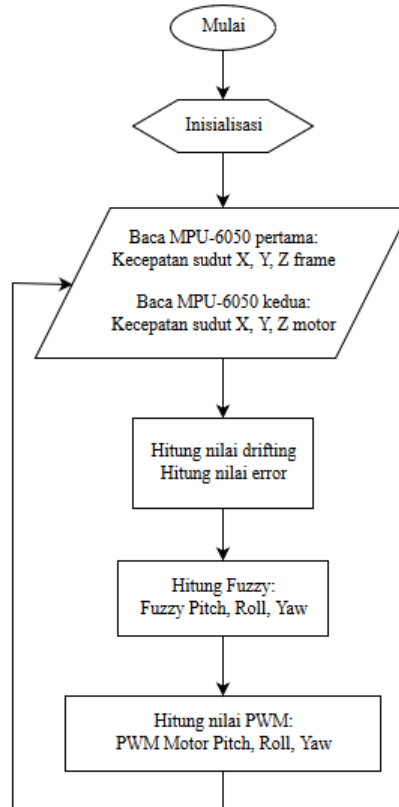


Gambar 6 Membership Function Delta Error

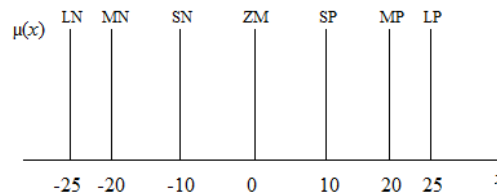
Untuk *defuzzifikasi* digunakan perhitungan *Weighted Average* [4].

$$Z = \frac{\sum_{i=1}^7 w_i \times \mu(w_i)}{\sum_{i=1}^7 \mu(w_i)} \quad (2)$$

**E. Diagram Alir Program Utama Mikrokontroler**



Gambar 7 Diagram Alir Program Utama Mikrokontroler



Gambar 8 Posisi Nilai x Anggota Output Singleton

**F. Perancangan Program untuk Pengujian Lapangan Menggunakan OpenCV**

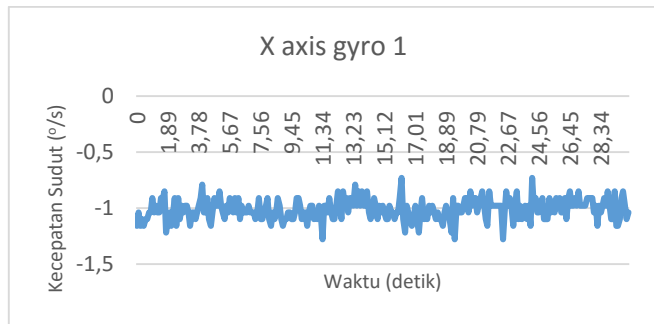
Pada proses pengujian sistem ini dilibatkan *tracking object* menggunakan *library* dari OpenCV [5]. Pengujian ini memakai cara merekam gambar lingkaran merah pada kertas yang dihadapkan lurus dengan kamera, lalu robot mulai dijalankan lurus. Hasil video kemudian diolah untuk men-*tracking* lingkaran merah tersebut apakah bergerak sedikit atau bergerak banyak.

Tahap pertama dalam *tracking* ini adalah menggunakan filter warna yang berdasarkan HSV (*Hue Saturation Value*). Pada pengujian ini akan memberi filter warna merah berdasarkan *range hue* warna merah yaitu dari nilai 160-179 dan warna oranye 0-10. Kemudian kedua nilai ini akan digabungkan dengan fungsi *addWeighted*. Tahap kedua adalah menandai bentuk warna-warna merah dalam video yaitu dengan fungsi *findCountour*. Lalu tahap ketiga adalah mencari nilai luasan kontur merah terbesar yang nantinya akan dipilih sebagai objek yang diamati menggunakan fungsi *contourArea*. Selanjutnya kontur tersebut diberi garis kotak di sekelilingnya dengan fungsi *boundingRect*, tahap terakhir yaitu mencari titik tengah berdasarkan garis kotak tersebut. Nilai titik tengah inilah yang nantinya akan diamati dan dibuat grafik, seberapa besar pergeseran objek lingkaran merah tersebut.

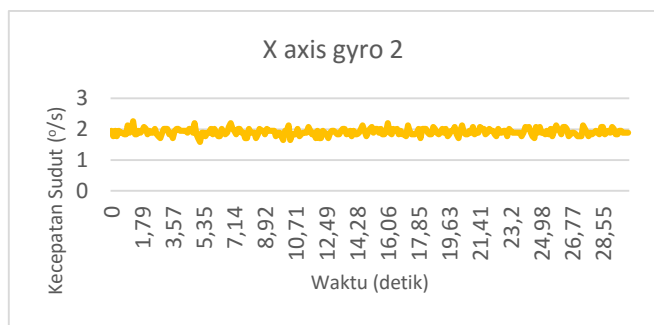
### III. PENGUJIAN DAN PEMBAHASAN

#### A. Hasil Pengujian Sensor Gyroscope

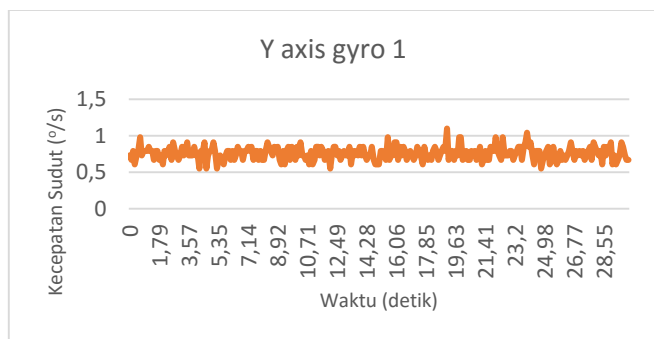
Pada bagian ini akan diketahui seberapa besar nilai *offset* karena *drifting* yang terjadi pada masing masing sumbu sensor *gyroscope* pertama dan kedua.



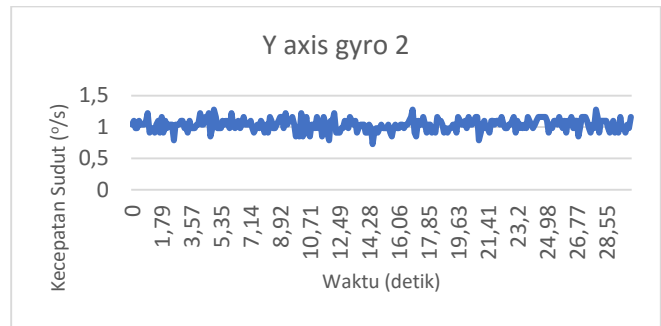
Gambar 9 Offset Sumbu X Sensor Gyroscope Pertama



Gambar 10 Offset Sumbu X Sensor Gyroscope Kedua

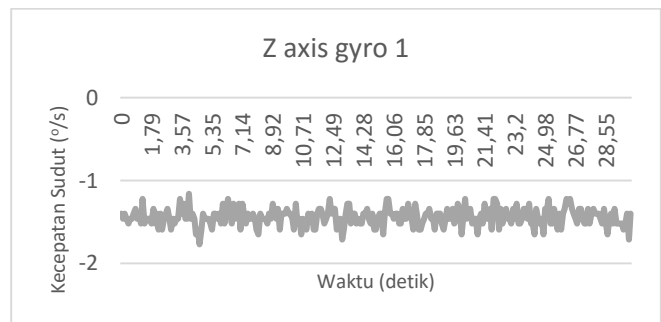


Gambar 11 Offset Sumbu Y Sensor Gyroscope Pertama

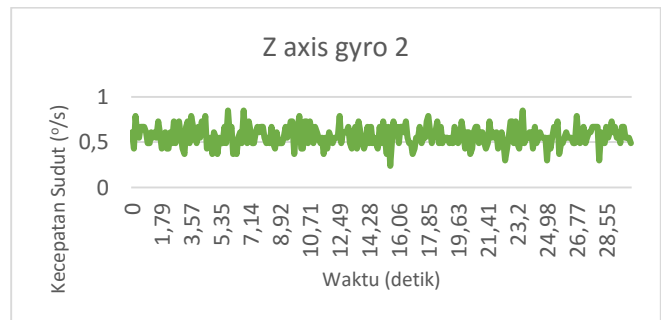


Gambar 12 Offset Sumbu Y Sensor Gyroscope Kedua

Dari pengujian selama 30 detik dapat diketahui nilai rata-rata *offset* untuk *gyroscope* pertama sumbu X adalah -1.1 derajat/detik. Dari pengujian selama 30 detik dapat diketahui nilai rata-rata *offset* untuk *gyroscope* kedua sumbu X adalah 1.92 derajat per detik. Dari pengujian selama 30 detik dapat diketahui nilai rata-rata *offset* untuk *gyroscope* pertama sumbu Y adalah 0.7 derajat per detik. Dari pengujian selama 30 detik dapat diketahui nilai rata-rata *offset* untuk *gyroscope* kedua sumbu Y adalah 1.1 derajat per detik.



Gambar 13 Offset Sumbu Z Sensor Gyroscope Pertama



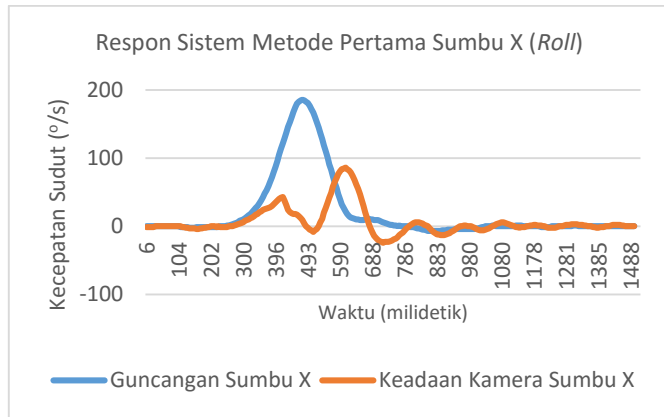
Gambar 14 Offset Sumbu Z Sensor Gyroscope Kedua

Dari pengujian selama 30 detik dapat diketahui nilai rata-rata *offset* untuk *gyroscope* pertama sumbu Z adalah -1.4 derajat per detik. Dari pengujian selama 30 detik dapat diketahui nilai rata-rata *offset* untuk *gyroscope* kedua sumbu Z adalah 0.55 derajat per detik.

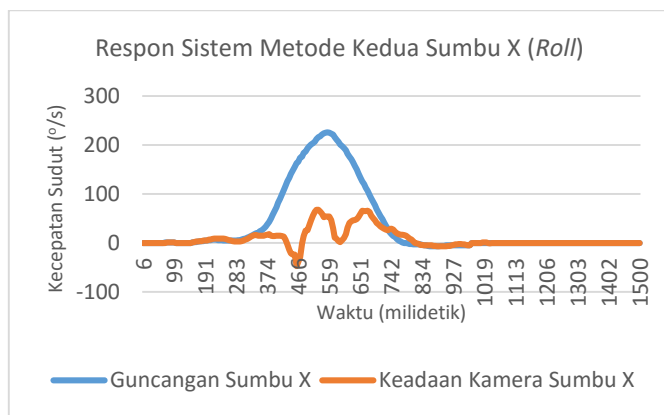
#### B. Pengujian Kestabilan Sistem dengan Guncangan Terukur

Dari pengujian selama 1500 milidetik dapat terlihat pada grafik bahwa ketika diberi guncangan pada sumbu X, dapat diketahui standar deviasi untuk metode pertama adalah sebesar 20.51. Dari grafik juga terlihat bahwa respon sistem memiliki *overshoot* dan *settling time* yang besar. Sementara ketika menggunakan metode kedua, standar deviasinya adalah 18.80.

Dari grafik juga terlihat bahwa *overshoot settling time* yang besar pada metode pertama dapat dihilangkan.

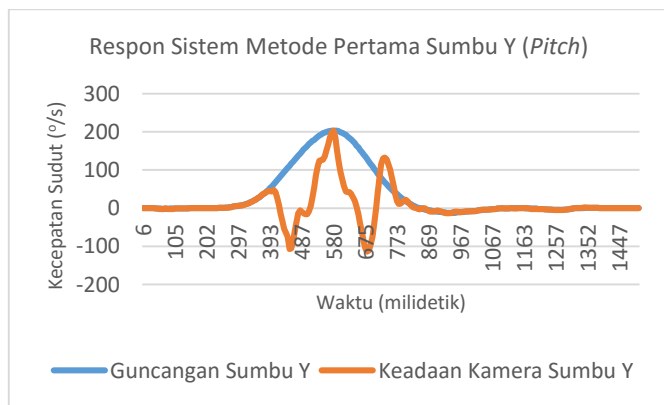


Gambar 15 Respon Sistem Sumbu X dengan Metode Pertama

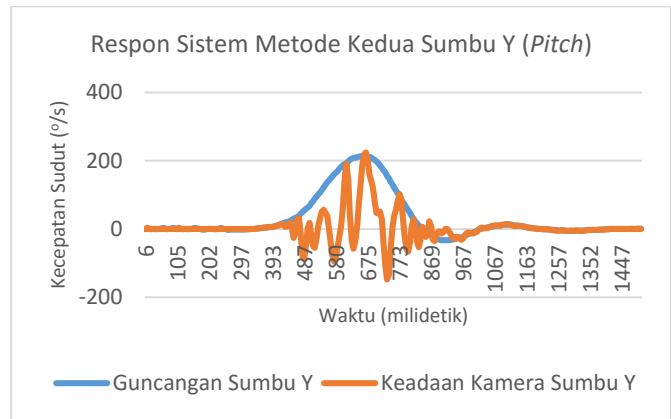


Gambar 16 Respon Sistem Sumbu X dengan Metode Kedua

Ketika diberi guncangan pada sumbu Y, dapat diketahui standar deviasi untuk metode pertama adalah sebesar 45.42. Dari grafik juga terlihat bahwa respon sistem memiliki *overshoot* yang besar. Sementara ketika menggunakan metode kedua, standar deviasinya adalah 43.82. Dari grafik juga terlihat bahwa *overshoot* yang terjadi pada metode pertama dapat dihilangkan.

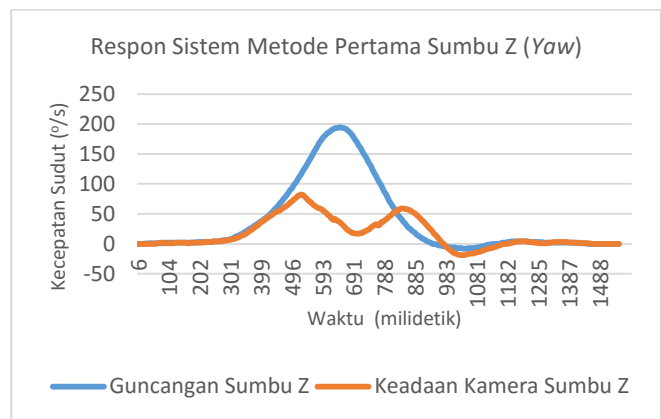


Gambar 17 Respon Sistem Sumbu Y dengan Metode Pertama

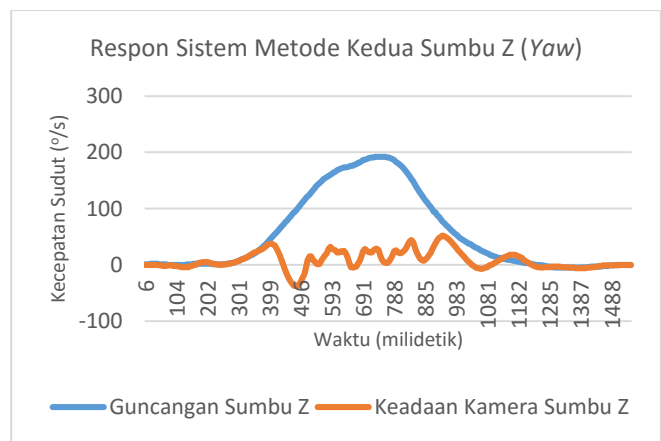


Gambar 18 Respon Sistem Sumbu Y dengan Metode Kedua

Ketika diberi guncangan pada sumbu Z, dapat diketahui standar deviasi untuk metode pertama adalah sebesar 25.16. Dari grafik juga terlihat bahwa respon sistem memiliki *overshoot* yang besar. Sementara ketika menggunakan metode kedua, standar deviasinya adalah 16.01. Dari grafik juga terlihat bahwa *overshoot* yang terjadi pada metode pertama dapat dihilangkan. Dapat disimpulkan bahwa pada sistem dengan *kontroler Fuzzy* ini, *overshoot* sistem dan *settling time* yang besar masih terjadi jika dibandingkan dengan *kontroler PID* ketika sama-sama menggunakan satu sensor *gyroscope* [6].



Gambar 19 Respon Sistem Sumbu Z dengan Metode Pertama



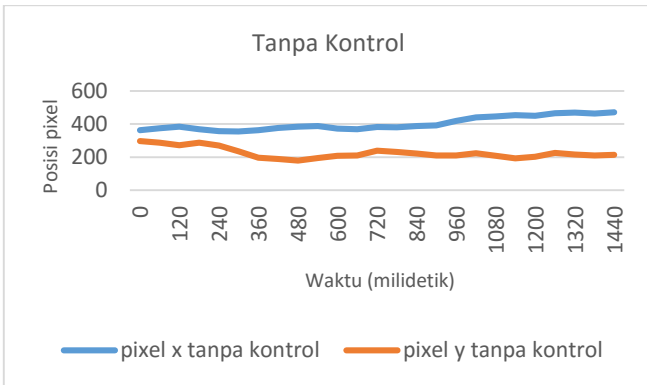
Gambar 20 Respon Sistem Sumbu Z dengan Metode Kedua

C. Pengujian Kestabilan Sistem pada Lapangan

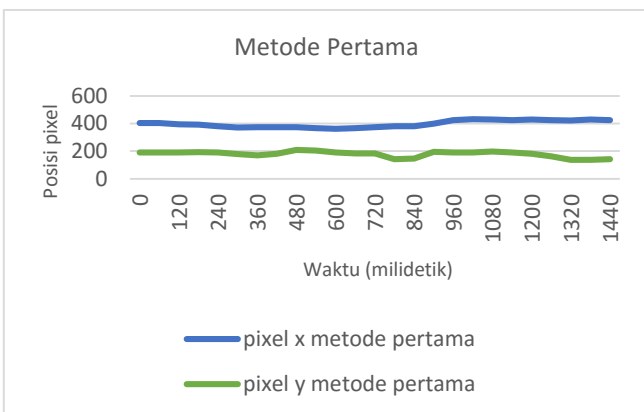
Pada bagian ini akan diketahui seberapa besar pergeseran suatu objek warna terhadap pergerakan *mobile robot*. Pengujian tanpa menggunakan sistem dilakukan dengan cara menjepit masing-masing motor pada *frame* menggunakan penjepit kertas. Hasil yang disertakan adalah titik pusat objek berupa nilai *pixel x* dan *y* dalam video selama 1.5 detik.



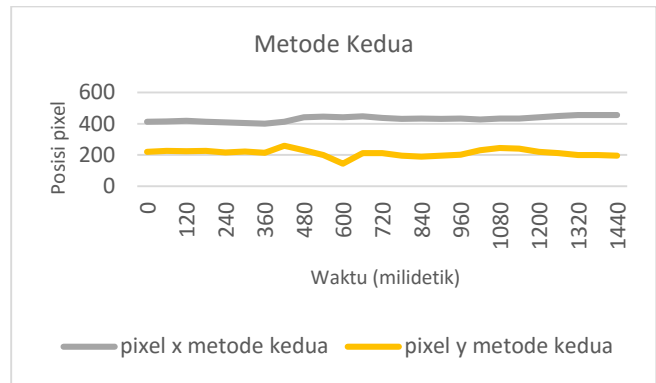
Gambar 18 Pengolahan Citra Menggunakan Open CV



Gambar 19 Posisi perubahan *pixel* objek yang diamati tanpa menggunakan kontrol



Gambar 20 Posisi perubahan *pixel* objek yang diamati dengan menggunakan metode pertama



Gambar 21 Posisi perubahan *pixel* objek yang diamati dengan menggunakan metode kedua

Dari data yang disajikan pada grafik di atas, sistem mampu meredam namun tidak cukup baik. Ketika tanpa kontrol perbedaan letak titik tengah objek pada sumbu *x* dan *y* memiliki perbedaan lebih dari 100 *pixel*, Hasil ketika tanpa kontrol, standar deviasi *pixel* sumbu *x* adalah 40.57 dan sumbu *y* adalah 32.95. Sementara ketika dengan kontrol, perbedaan *pixel* tidak terlalu jauh, sekitar 20 *pixel*. Ketika dengan menggunakan metode pertama, standar deviasi *pixel* sumbu *x* adalah 24.73 dan sumbu *y* adalah 21.73. Sementara ketika dengan menggunakan metode kedua, standar deviasi *pixel* sumbu *x* adalah 16.70 dan sumbu *y* adalah 22.44. Dari nilai standar deviasi, dapat disimpulkan bahwa metode kedua lebih stabil daripada metode pertama.

IV. KESIMPULAN

Dari seluruh nilai standar deviasi yang didapatkan pada pengujian, dapat disimpulkan bahwa metode kedua lebih stabil daripada metode pertama karena nilai standar deviasi metode kedua lebih kecil 1.71 s.d. 9.15 daripada metode pertama. Selisih standar deviasi pada pengujian perekaman video antara tanpa kontrol dan menggunakan metode pertama adalah 15.84 untuk *pixel x*, dan 11.22 untuk *pixel y*, dan selisih antara tanpa kontrol dan menggunakan metode kedua adalah 23.87 untuk *pixel x*, dan 10.55 untuk *pixel y*. Dapat disimpulkan bahwa sistem yang dibuat ini dengan *kontroler Fuzzy* ini, *overshoot* sistem dan *settling time* yang besar masih terjadi jika dibandingkan dengan *kontroler PID* ketika sama-sama menggunakan satu sensor *gyroscope*.

DAFTAR PUSTAKA

- [1] Shobakh, M. Nur., "Pengembangan Robot Pengikut Garis Berbasis Mikrokontroler Sebagai Meja Pengantar Makanan Otomatis", ITS Surabaya, 2012.
- [2] Datasheet MPU-6050 "MPU-6000 and MPU-6050 product specification", August 2013.
- [3] Jannus Maurits Nainggolan, "Logika Fuzzy (Fuzzy Logic): Teori dan Penerapan Pada Sistem Daya (Kajian Pengaruh Induksi Medan Magnet)", <URL: [http://member.unila.ac.id/~ftelektro/lab/ltp/dokumen/Fuzzy%20Logic%20Pa per.doc](http://member.unila.ac.id/~ftelektro/lab/ltp/dokumen/Fuzzy%20Logic%20Pa%20per.doc)> 23 Mei 2016.
- [4] <URL: [http://k12008.widyagama.ac.id/ai/diktatpdf/Logika\\_Fuzzy.pdf](http://k12008.widyagama.ac.id/ai/diktatpdf/Logika_Fuzzy.pdf)> 23 Mei 2016.
- [5] R. Laganiere, "Opencv 2 Computer Vision App Cookbook", [Packt] Publishing, Birmingham, 2011
- [6] Jodit Sulistyo, "Desain dan Implementasi Gimbal Menggunakan Brushless Motor untuk Stabilisator Posisi Kamera", ITS Surabaya, 2014.