

PERANCANGAN KONTROLER *SELF – TUNING* PI DENGAN LOGIKA FUZZY SEBAGAI KENDALI MOTOR SERVO PADA SISTEM TRACKING BOLA

Rizki Anggoro Kurniawan^{*)}, Wahyudi, dan Ajub Ajulian Zahra

Departemen Teknik Elektro, Universitas Diponegoro
Jl. Prof. Sudharto, SH, Kampus UNDIP Tembalang, Semarang 50275, Indonesia

^{*)}E-mail: rizkianggoro2@gmail.com

Abstrak

Perekaman pertandingan sepak bola saat ini masih menggunakan tenaga manusia untuk mengarahkan kamera. Pengarahan kamera dapat diotomasi Sistem self-targetting turret ini akan mengikuti gerak bola. Sistem kendali yang masih populer digunakan adalah kendali PI. Salah satu cara menentukan parameter Kp dan Ki pada kontrol PI adalah dengan metode fuzzy. Berdasarkan hal tersebut, pada Penelitian ini, diterapkan sistem kendali self-tuning PI dengan logika fuzzy. Pendeteksian bola pada Penelitian ini menggunakan Hough Transform pada citra yang telah di-threshold agar citra bola dengan citra latar belakang terpisah. Pengujian lainnya yaitu adalah pengujian sistem kendali self-tuning PI dengan metode fuzzy. Hasil pengujian kendali self-tuning PI pada servo tilt didapatkan t_r sebesar 70 ms, t_s sebesar 155 ms, M_p sebesar 22%, dan e_{ss} sebesar 1,4%, sedangkan pada servo pan diperoleh t_r sebesar 125 ms, t_s sebesar 140 ms, M_p sebesar 10,8%, dan e_{ss} sebesar 1,3%. Pengujian sistem keseluruhan pada target bergerak dengan pola tertentu didapatkan error rata-rata sebesar $-6,65^\circ$ untuk servo pan dan $-2,36^\circ$ pada servo tilt. Pengujian pada target bergerak secara bebas diperoleh rata-rata error sebesar $-1,8^\circ$ untuk servo tilt dan $0,67^\circ$ pada servo pan.

Kata Kunci: tracking bola, kontroler PI, logika fuzzy, self-tuning

Abstract

Currently, football tournament recording still used human to direct the camera. Directing the camera can be automated. The most popular control system were PI control. One way to tune the value of Kp and Ki in the PI control is with fuzzy logic. This Research design the self tuning PI control system with fuzzy logic. The ball tracking system in this Research use Hough transform method to detect the ball. The other test were self tuning PI control system test. The test result for the tilt servo obtained t_r as much as 70 ms, t_s as much as 155 ms, M_p as much as 22%, and e_{ss} as much as 1.4%, whereas the pan servo obtained t_r as much as 125 ms, t_s as much as 140 ms, M_p as much as 10.8%, and e_{ss} as much as 1.3%. The test of the whole system in a moving target with specific pattern resulted in average error -6.65° for the pan servo and -2.36° for the tilt servo. The test in a moving target with free pattern resulted in average error -1.8° for the tilt servo and 0.67° for the pan servo.

Keywords: ball tracking, PI control, fuzzy logic, self tuning

1. Pendahuluan

Pada perekaman pertandingan sepak bola saat ini masih menggunakan tenaga manusia untuk mengarahkan kamera. Penggunaan tenaga manusia untuk mengarahkan kamera dapat digantikan dengan sistem otomasi. Otomasi sistem dapat menggunakan konsep *self-targetting turret* yang sering digunakan pada dunia militer untuk mendeteksi dan mengikuti target[1].

Permasalahan utama pada sistem otomasi adalah adanya *error* dan respon sistem yang kurang baik. Penggunaan metode kontrol Proporsional Derifatif (PI)[2] dapat

mengatasi *error steady state* yang terjadi sedangkan logika fuzzy[3] dapat digunakan untuk mengatur parameter Kp dan Ki pada kontroler.

Mengacu pada penelitian [4] tentang kendali *turret coil gun* menggunakan kontroler *proporsional integral derivative* (PID) dapat mengarahkan senjata dengan cukup baik namun sistem memiliki *settling time* yang cukup lama. Respon yang kurang baik pada sistem kendali PID dikarenakan parameter Kp, Ki, dan Kd bersifat statis. Permasalahan ini dapat diatasi dengan metode *self tuning* PID menggunakan metode *fuzzy* [5] pada kendali sudut *yaw* dan *pitch* pada *hovercraft*. Penggunaan metode *self*

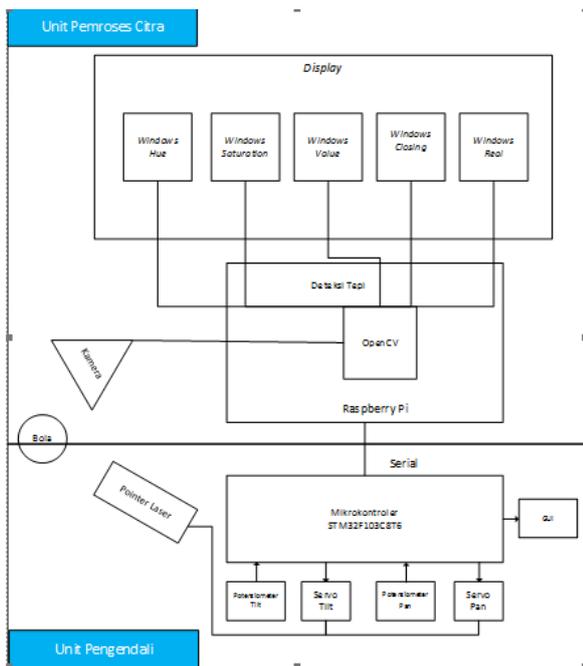
tuning PID dengan logika fuzzy dapat mengubah parameter Kp, Ki, dan Kd pada *plant* dengan membandingkan nilai *error* dan *delta error* sistem untuk menentukan parameter PID yang tepat sesuai dengan perubahan respon sistem. Pada Penelitian ini, dirancang alat *tracking* bola menggunakan metode self tuning PI dengan metode logika fuzzy untuk melacak bola.

2. Metode

Perancangan sistem alat penggerak *turret* terdiri atas Perancangan hardware, perancangan pendeteksian target, perancangan kontrol PI dan perancangan *self tuning* PI dengan metode *fuzzy*.

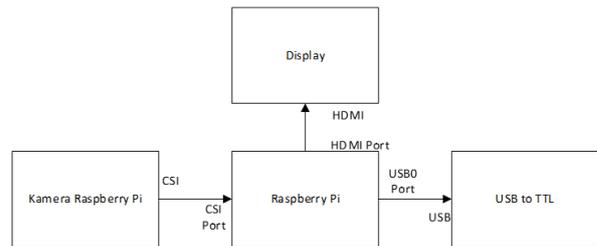
2.1. Perancangan Hardware.

Hardware memiliki dua bagian yaitu bagian pengolah citra dan bagian pengendali. Bagian pengolah citra menggunakan kamera raspberry pi untuk menangkap citra yang kemudian akan diolah untuk pendeteksian target pada raspberry pi[6]. Bagian pengendali menggunakan STM32F103c8t6[7] sebagai pengendali. Diagram blok keseluruhan sistem dapat dilihat pada Gambar 1.



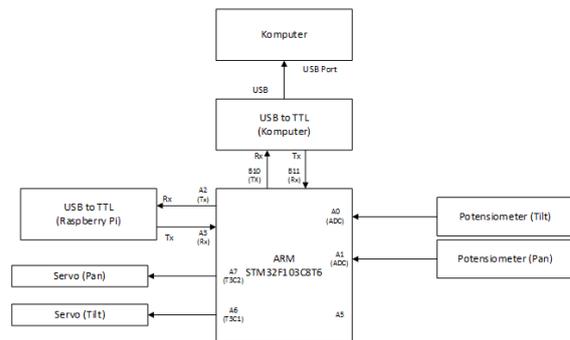
Gambar 1. Diagram blok perancangan sistem *tracking turret*.

Unit pemroses citra memiliki 4 komponen utama yaitu raspberry pi, kamera raspberry pi[8], USB to TTL module, dan *display*. Kamera dihubungkan dengan menggunakan kabel *Camera Serial Interface* (CSI) ke raspberry pi. Bagian *display* dihubungkan dengan menggunakan kabel *High Definition Media Interface* (HDMI) dari monitor ke raspberry pi. Diagram blok perancangan bagian pemroses citra dapat dilihat pada Gambar 2.



Gambar 2. Diagram blok pemroses citra.

Unit pengendali merupakan bagian pengendali aktuator sistem.



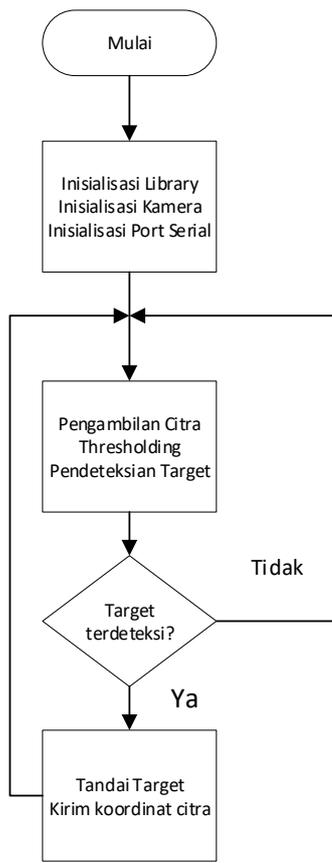
Gambar 3. Diagram blok pengendali.

Berdasarkan Gambar 3, *servo tilt* menggunakan *port* A6 pada mikrokontroler. *Servo pan* menggunakan *port* A7. *Potensiometer tilt* menggunakan *port* A0. *Potensiometer pan* menggunakan *port* A1. USB to TTL yang menghubungkan antara mikrokontroler dengan raspberry pi menggunakan *port* A2(Tx) dan A3(Rx) sedangkan, *port* B10(Tx) dan B11(Rx) digunakan untuk menghubungkan USB to TTL yang mentransmisikan data ke komputer.

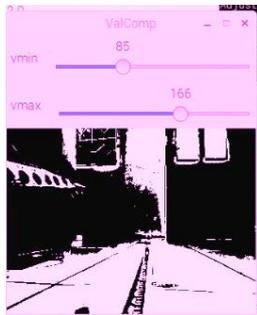
2.2. Perancangan Pendeteksian Target

Sistem pendeteksian target menggunakan *binary thresholding* untuk memisahkan citra target yang berupa bola ping-pong dengan citra latar, kemudian citra target dideteksi dengan menggunakan *Hough transform*[9]. Setelah target dideteksi maka dikirimkan koordinat objek yaitu berupa titik tengah objek melalui komunikasi serial ke mikrokontroler ARM. Algoritma pendeteksian target dapat dilihat pada Gambar 4.

Kalibrasi *threshold* dilakukan dengan *window* seperti Gambar 5 pada setiap bidang citra *Hue*, *Saturation*, and *Value* (HSV)[10]. *Thresholding* dilakukan untuk memisahkan citra target dengan latar pada setiap komponen citra HSV yaitu *hue*, *saturation*, dan *value*. *Window* citra *hue* digunakan untuk menentukan warna yang digunakan, *saturation* untuk menentukan kecerahan citra, dan *value* untuk menentukan kedalaman warna.



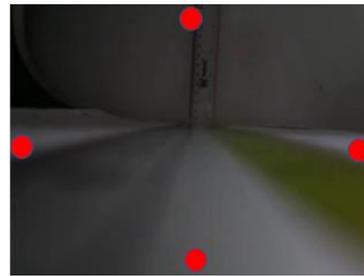
Gambar 4. Diagram algoritma pendeteksian target.



Gambar 5. Window dan trackbar kalibrasi citra.

2.3. Perancangan Jangkauan Gerak Servo

Penentuan masukan untuk motor servo menggunakan fungsi *scaling* dari koordinat citra ke sudut. Kalibrasi skala yaitu dengan mengaktifkan laser kemudian membaca nilai sudut dari *potensiometer* pada batas bawah, batas atas, batas kanan, dan batas kiri citra.



Gambar 6. Pengambilan nilai sudut untuk perbandingan skala.

Dari pengambilan sudut pada posisi pengambilan citra seperti yang ditunjukkan di Gambar 6, diperoleh sumbu x sudut 72 derajat hingga 130 derajat, sedangkan untuk sumbu y diperoleh sudut 50 hingga 98 derajat.

2.4. Perancangan Kontrol PI

Sistem kontrol PI digunakan untuk memperbaiki respon sistem. Kontrol PI dapat mempercepat respon sistem dan menghilangkan *error steady state*[2]. Perancangan Kp dan Ki menggunakan metode *try and error*. Hasil pengujian dapat dilihat pada Tabel 1 dan Tabel 3 untuk *servo tilt* dan Tabel 2 dan Tabel 4 untuk *servo pan*.

Tabel 1. Hasil uji coba nilai Kp pada Servo Tilt

| No. | Kp Tilt | | | | |
|-----|---------|-----------|-----------|-------|-------------|
| | Kp | $t_r(ms)$ | $t_s(ms)$ | Mp(%) | $e_{ss}(%)$ |
| 1 | 0,7 | 60 | 115 | 18,3 | 3,5 |
| 2 | 0,8 | 65 | 115 | 20,9 | 4,8 |
| 3 | 1 | 70 | 155 | 22,7 | 2,6 |
| 4 | 1,2 | 65 | 180 | 22,9 | 1,1 |
| 5 | 1,5 | 65 | 175 | 20,3 | 0,9 |
| 6 | 1,8 | 65 | 375 | 41,4 | 3,1 |

Tabel 2. Hasil uji coba nilai Kp pada Servo Pan

| No. | Kp Pan | | | | |
|-----|--------|-----------|-----------|-------|-------------|
| | Kp | $t_r(ms)$ | $t_s(ms)$ | Mp(%) | $e_{ss}(%)$ |
| 1 | 0,7 | 75 | 145 | 19,2 | 5,7 |
| 2 | 0,8 | 75 | 145 | 19,2 | 3,4 |
| 3 | 1 | 80 | 180 | 22,9 | 2,7 |
| 4 | 1,2 | 75 | 205 | 19,3 | 5,7 |
| 5 | 1,5 | 80 | 215 | 19 | 6 |
| 6 | 1,8 | 80 | 270 | 21,8 | 5 |

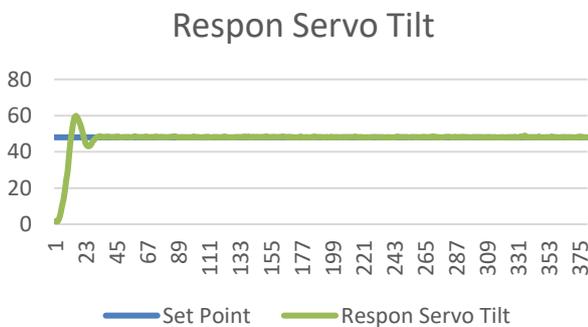
Tabel 3. Hasil uji coba nilai Ki pada Servo Tilt

| No. | Ki Tilt | | | | |
|-----|---------|-----------|-----------|-------|-------------|
| | Ki | $t_r(ms)$ | $t_s(ms)$ | Mp(%) | $e_{ss}(%)$ |
| 1 | 0,01 | 60 | 115 | 15,8 | 4 |
| 2 | 0,05 | 100 | 5020 | 12,7 | 4,5 |
| 3 | 0,1 | 100 | - | 11,5 | 10,4 |
| 4 | 0,15 | 95 | 5065 | 11,3 | 5 |
| 5 | 0,25 | 65 | 5860 | 3 | 4,5 |
| 6 | 0,5 | 6490 | 5640 | - | 1 |

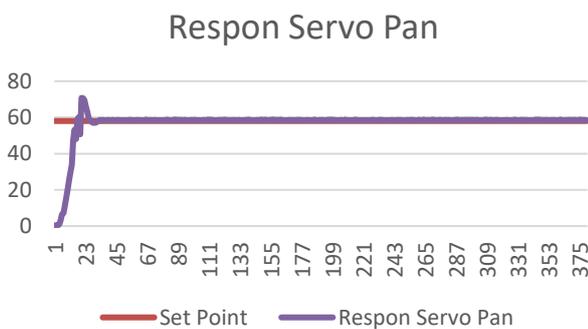
Tabel 4. Hasil uji coba nilai Ki pada Servo Pan

| No. | Ki Pan | | | | |
|-----|--------|------------|------------|-------|--------------|
| | Kp | t_r (ms) | t_s (ms) | Mp(%) | e_{ss} (%) |
| 1 | 0,01 | 135 | 270 | 24,1 | 1 |
| 2 | 0,05 | 75 | 250 | 13,5 | -1,1 |
| 3 | 0,1 | 80 | 170 | 21,9 | -4,3 |
| 4 | 0,15 | 155 | 6125 | 3 | 5 |
| 5 | 0,25 | 125 | 3840 | 9,3 | 2 |
| 6 | 0,5 | 4485 | 7565 | 20,8 | -1 |

Berdasarkan hasil pengujian diperoleh Kp sebesar 1 untuk servo tilt dan 0,8 untuk servo pan. Nilai Ki yang digunakan untuk servo tilt adalah 0,01, sedangkan untuk servo pan sebesar 0,05. Pemilihan nilai berdasarkan pertimbangan respon sistem pada setiap nilai Kp dan Ki yang telah diuji. Hasil respon sistem dengan Kp dan Ki dapat dilihat pada Gambar 7 untuk servo tilt dan Gambar 8 untuk servo pan.



Gambar 7. Respon sistem tilt dengan Kp =1 dan Ki=0,01.

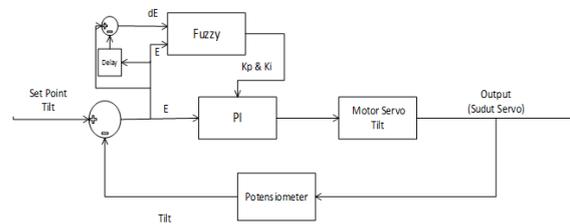


Gambar 8. Respon sistem pan dengan Kp=0,08 dan Ki=0,05.

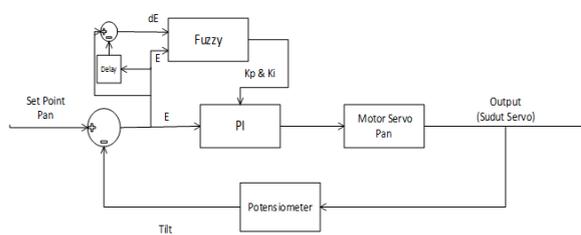
2.5. Perancangan Self Tuning PI Dengan Metode Fuzzy

Sistem dengan self-tuning PI dilakukan agar nilai Kp dan Ki sistem dapat berubah sesuai dengan perubahan respon. Metode fuzzy Sugeno digunakan sebagai sistem self tuning[3]. Error dan Δerror dipakai sebagai acuan untuk mengubah nilai Kp dan Ki secara adaptif. Diagram blok kontrol self-tuning PI dengan metode fuzzy dapat dilihat

pada Gambar 9 untuk servo tilt dan Gambar 10 untuk servo pan.

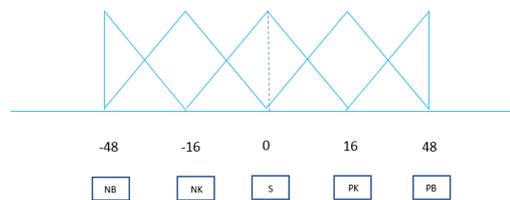


Gambar 9. Diagram blok Self Tuning PI dengan metode fuzzy pada servo tilt.

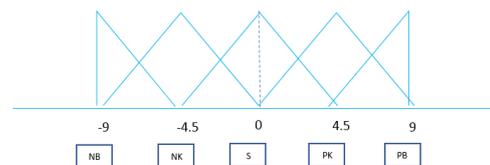


Gambar 10. Diagram blok Self Tuning PI dengan metode fuzzy pada servo pan.

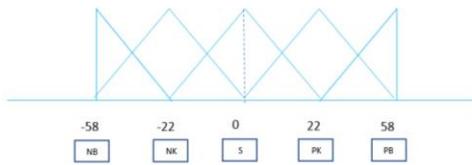
Perancangan membership function diperoleh dari observasi terhadap respon sistem. Membership function pada Penelitian ini menggunakan 5 fungsi segitiga dengan 5 label yaitu yaitu negatif besar (NB), negatif kecil (NK), sedang (S), positif kecil (PM) dan positif besar (PB). Pemilihan batas pada membership function error servo tilt dan pan yaitu berdasarkan jangkauan gerak servo yang digunakan, sedangkan pemilihan batsan pada Δerror ditentukan berdasarkan Δerror respon sistem. Membership function dapat dilihat pada Gambar 11, Gambar 12, Gambar 13, dan Gambar 14.



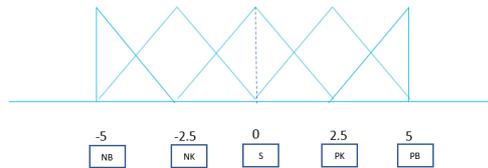
Gambar 11. Membership function error servo tilt.



Gambar 12. Membership function Δerror servo tilt.

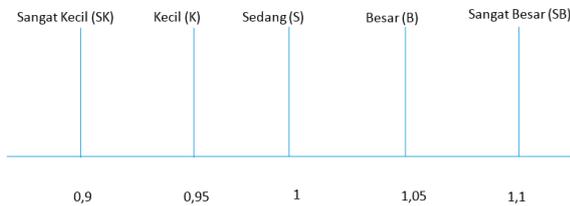


Gambar 13. Membership function error servo pan.

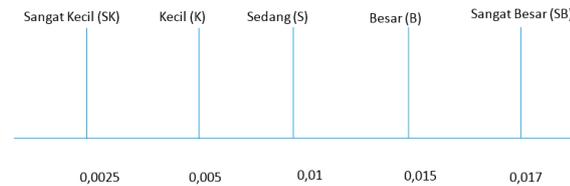


Gambar 14. Membership function Δ error servo pan.

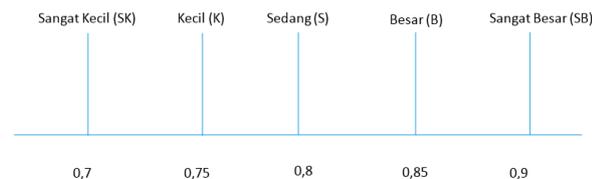
Singleton K_p dan K_i pada servo tilt dan servo pan menggunakan nilai K_p dan K_i pada perancangan kendali PI. Singleton K_p dan K_i pada perancangan Penelitian ini memiliki 5 label yaitu sangat kecil (SK), kecil (K), sedang (S), besar (B), dan sangat besar (SB). Perancangan singleton yang digunakan pada Penelitian ini dapat dilihat pada Gambar 15, Gambar 16, Gambar 17, dan Gambar 18.



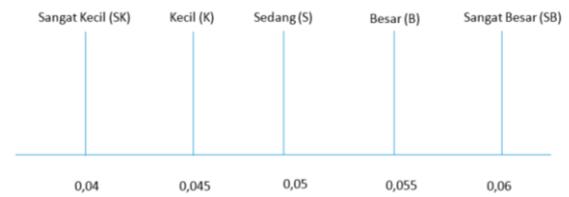
Gambar 15. Singleton K_p servo tilt.



Gambar 16. Singleton K_i servo tilt.



Gambar 17. Singleton K_p servo pan.



Gambar 18. Singleton K_i servo pan.

Penentuan rule base yang digunakan dimaksudkan agar nilai K_p dan K_i semakin kecil ketika respon mendekati setpoint. Nilai K_p dan K_i yang semakin mengecil dapat mengurangi overshoot dan waktu respon sistem. Penentuan rule base dapat dilihat pada Tabel 5, Tabel 6, Tabel 7, dan Tabel 8.

Tabel 5. Rule base fuzzy k_p tilt servo.

| E | Error Negatif Besar | Error Negatif Kecil | Error Sedang | Error Positif Kecil | Error Positif Besar | |
|-----------------------|---------------------|---------------------|--------------|---------------------|---------------------|----|
| Δ Error Besar | Negatif | SB | B | S | B | SB |
| Δ Error Kecil | Negatif | B | S | K | S | B |
| Δ Error Sedang | Negatif | S | K | SK | K | S |
| Δ Error Kecil | Positif | B | S | K | S | B |
| Δ Error Besar | Positif | SB | B | S | B | SB |

Tabel 6. Rule base fuzzy k_i tilt servo.

| E | Error Negatif Besar | Error Negatif Kecil | Error Sedang | Error Positif Kecil | Error Positif Besar | |
|-----------------------|---------------------|---------------------|--------------|---------------------|---------------------|----|
| Δ Error Besar | Negatif | SB | B | S | B | SB |
| Δ Error Kecil | Negatif | B | S | K | S | B |
| Δ Error Sedang | Negatif | S | K | SK | K | S |
| Δ Error Kecil | Positif | B | S | K | S | B |
| Δ Error Besar | Positif | SB | B | S | B | SB |

Tabel 7. Rule base fuzzy k_p tilt pan.

| E | Error Negatif Besar | Error Negatif Kecil | Error Sedang | Error Positif Kecil | Error Positif Besar | |
|-----------------------|---------------------|---------------------|--------------|---------------------|---------------------|----|
| Δ Error Besar | Negatif | SB | B | S | B | SB |
| Δ Error Kecil | Negatif | B | S | K | S | B |
| Δ Error Sedang | Negatif | S | K | SK | K | S |
| Δ Error Kecil | Positif | B | S | K | S | B |
| Δ Error Besar | Positif | SB | B | S | B | SB |

Tabel 8. Rule base fuzzy k_p tilt pan.

| E | Error Negatif Besar | Error Negatif Kecil | Error Sedang | Error Positif Kecil | Error Positif Besar | |
|-----------------------|---------------------|---------------------|--------------|---------------------|---------------------|----|
| Δ Error Besar | Negatif | SB | B | S | B | SB |
| Δ Error Kecil | Negatif | B | S | K | S | B |
| Δ Error Sedang | Negatif | S | K | SK | K | S |
| Δ Error Kecil | Positif | B | S | K | S | B |
| Δ Error Besar | Positif | SB | B | S | B | SB |

2.6. Perancangan Software Antarmuka

Perancangan *software* antarmuka dilakukan dengan menggunakan bahasa C#. *Software* menerima data yang akan ditampilkan dengan menggunakan *baudrate* sebesar 57600 bps. Tampilan *software* antarmuka dapat dilihat pada Gambar 19.



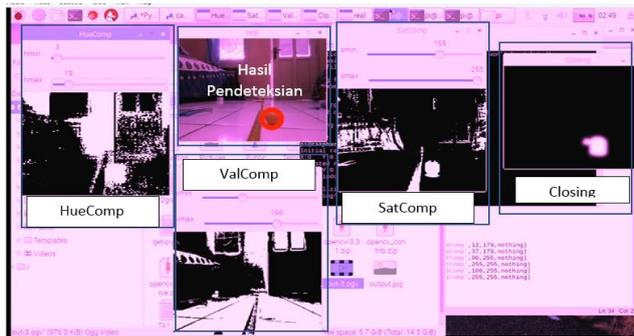
Gambar 19. Tampilan *software* antarmuka.

3. Hasil dan Analisa

Hasil dari penelitian ini diperoleh melalui serangkaian pengujian, pengujian yang dilakukan terdiri dari pengujian pendeteksian target, pengujian sistem kontrol, pengujian *tracking* dengan pola tertentu, dan pengujian *tracking* dengan gerak target bebas.

3.1. Pengujian Pendeteksian Target

Pengujian pendeteksian target dilakukan di dalam ruangan dengan menggunakan bola ping-pong sebagai target.

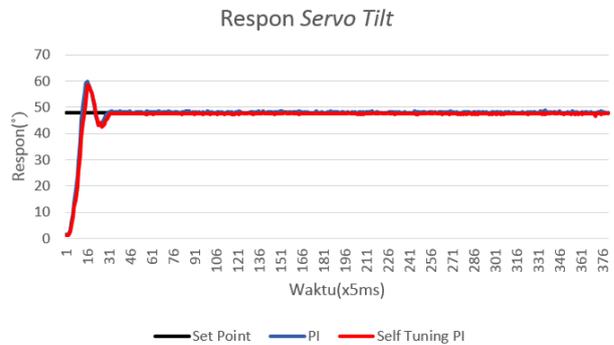


Gambar 20. Hasil Pengujian Pendeteksian Target.

Hasil pengujian pendeteksian target yang ditunjukkan pada Gambar 20, terlihat bahwa target dapat terdeteksi dengan baik. Setiap komponen citra dapat mengisolasi target dengan citra latarnya walaupun pada komponen citra *value*, citra latar tidak dapat dipisahkan secara seluruhnya dengan citra target *Thresholding* pada setiap komponen citra HSV dapat mengisolasi citra target dengan citra latarnya seperti yang ditunjukkan pada *windows* “closing”.

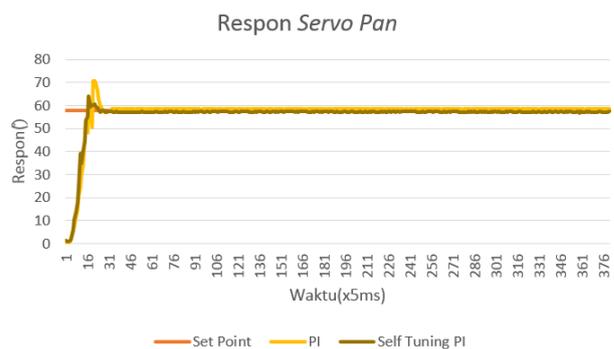
3.2. Pengujian Sistem Kendali Self-Tuning PI

Pengujian *self-tuning* PI dilakukan dengan menggerakkan *servo tilt* sebesar 48° dan *servo pan* sebesar 58°. Hasil pengujian dibandingkan dengan sistem kendali PI.



Gambar 21. Hasil pengujian sistem kontrol pada *servo tilt*..

Dari hasil pengujian yang dapat dilihat pada Gambar 21 didapatkan bahwa sistem kendali *self tuning* PI dengan metode *fuzzy* memiliki respon yang tidak jauh berbeda dengan kendali PI. Sistem *self tuning* PI dengan metode *fuzzy* memiliki respon t_r sebesar 70 ms, t_s sebesar 155 ms, M_p sebesar 22%, dan e_{ss} sebesar 1,4% . Sistem dengan kendali PI memiliki respon t_r sebesar 65 ms, t_s sebesar 150 ms, M_p sebesar 22,7%, dan e_{ss} sebesar -0,6%. Dari data parameter respon sistem dapat dilihat bahwa kontroler *self tuning* PI dengan metode *fuzzy* lebih baik dibandingkan dengan kontroler PI pada *servo tilt* pada maksimum *overshoot* sedangkan untuk respon lainnya tidak jauh berbeda satu sama lain.



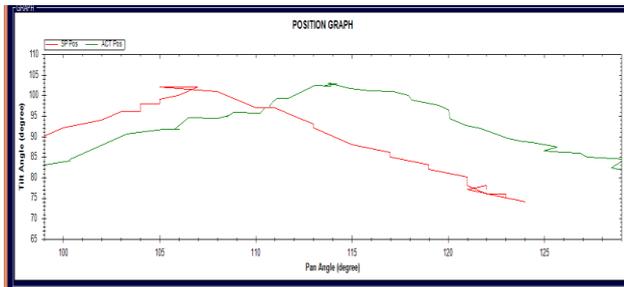
Gambar 22. Hasil pengujian sistem kontrol pada *servo pan* .

Dari hasil pengujian yang dapat dilihat pada Gambar 22 didapatkan bahwa sistem kendali *self tuning* PI dengan metode *fuzzy* memiliki respon yang tidak jauh berbeda dengan kendali PI. Sistem *self tuning* PI dengan metode *fuzzy* memiliki respon t_r sebesar 125 ms, t_s sebesar 140 ms, M_p sebesar 10,8%, dan e_{ss} sebesar 1,3% . Sistem dengan kendali PI memiliki respon t_r sebesar 130 ms, t_s sebesar 155 ms, M_p sebesar 19,8%, dan e_{ss} sebesar -1%.

Dari data parameter respon sistem dapat dilihat bahwa kontroler *self tuning* PI dengan metode *fuzzy* lebih baik dibandingkan dengan kontroler PI pada *servo pan* pada maksimum *overshoot*, t_r , dan t_s . sedangkan untuk e_{ss} tidak jauh berbeda namun pada sistem *self tuning* respon mengalami *undershoot* dan pada PI mengalami *overshoot*.

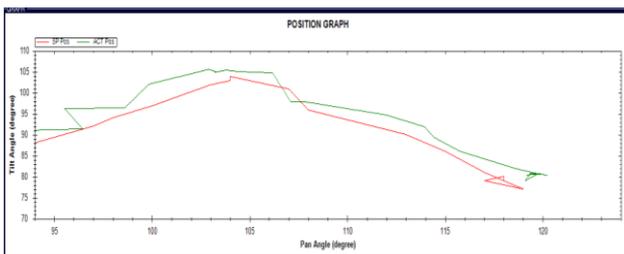
3.3. Pengujian Sistem Secara Keseluruhan

Pengujian sistem secara keseluruhan dilakukan dengan target bergerak dengan pola “V” dan bergerak bebas. Pada pengujian dengan gerak berpola sistem *self-tuning* PI dibandingkan dengan sistem dengan PI.



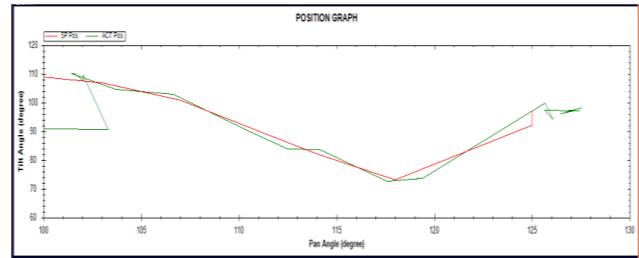
Gambar 23. Hasil Pengujian Targt Bergerak Berpola dengan kendalli *self tuning* PI..

Dari hasil pengujian yang ditunjukkan oleh Gambar 23, hasil pengujian gerak target berpola dengan menggunakan metode kendali *self tuning* PI dengan metode *fuzzy*. Dari pengujian didapatkan bahwa *turret* dapat mengikuti gerak bola dengan *error* rata-rata pada *servo tilt* sebesar $-2,36^\circ$ dan pada *servo pan* sebesar $-6,65^\circ$.



Gambar 24. Hasil pengujian target bergerak berpola dengan kendali PI..

Dari hasil pengujian yang ditunjukkan oleh Gambar 24, hasil pengujian gerak target berpola dengan menggunakan metode kendali PI. Dari pengujian didapatkan bahwa *turret* dapat mengikuti gerak bola dengan *error* rata-rata pada *servo tilt* sebesar $-1,86^\circ$ dan pada *servo pan* sebesar $-0,69^\circ$.



Gambar 25. Hasil pengujian target bergerak bebas dengan kendali *self tuning* PI dengan metode *fuzzy*..

Dari hasil pengujian yang ditunjukkan oleh Gambar 25, hasil pengujian gerak target bebas dengan menggunakan metode kendali *self tuning* PI dengan metode *fuzzy*. Dari pengujian didapatkan bahwa *turret* dapat mengikuti gerak bola dengan *error* rata-rata pada *servo tilt* sebesar $-1,8^\circ$ dan pada *servo pan* sebesar $0,67^\circ$.

4. Kesimpulan

Sistem *tracking* bola dengan kendali *self-tuning* PI memiliki respon yang lebih baik dibuktikan dengan M_p yang lebih kecil dibandingkan dengan sistem dengan PI. Sistem dengan kendali *self tuning* memiliki M_p sebesar 22% untuk *servo tilt* dibandingkan dengan 22,7% pada kendali PI, sedangkan pada *servo pan* menghasilkan M_p sebesar 10,8% dibandingkan dengan sistem dengan PI sebesar 19,8%. Pengujian sistem secara keseluruhan dengan target bergerak bebas menghasilkan sistem dengan rata-rata *error* sebesar $-1,8^\circ$ untuk *servo tilt* dan $0,67^\circ$ pada *servo pan*.

Referensi

- [1] E. Caponi, M. Lakus, A. Marar, and J. Thomas, *STATS Self-Targeting Autonomous Turret System*. Florida: University of Central Florida, 2014.
- [2] K. Ogata, *Modern Control Engineering*, 5th ed. New Jersey: Prentice Hall, 2010.
- [3] S. Kusumadewi and H. Purnomo, *Aplikasi Logika Fuzzy*, 2nd Editio. Yogyakarta: Graha Ilmu, 2013.
- [4]. D. S. Purnomo and I. A. Sulistijono, “Coil Gun Turret Control Using A Camera,” *Res. Gate*, no. January 2011, 2016.
- [5]. A. Rahman and I. Setiawan, “Perancangan Sistem Stabilisasi Roll Dan Yaw Pada Prototype Hovercraft Menggunakan Sensor Imu 6 Dof Dengan Metode Self Tuning Fuzzy PID,” *TRANSIENT*, vol. 6, no. 4, 2017.
- [6]. Raspberry, *Raspberry Pi 2 Model B Datasheet*.
- [7]. STMicroelectronics, *STM32F103x8 Datasheet*.
- [8]. Raspberry, *Raspberry Pi Camera v2 Datasheet*.
- [9]. R. Munir, *Pengolahan Citra Digital*. Informatika, 2005.
- [10]. L. Cotnoir, “Hue, Value, Saturation.” [Online]. Available: <http://learn.leighcotnoir.com/artspk/elements-color/hue-value-saturation/>. [Accessed: 08-Feb-2018].