

DESAIN DAN IMPLEMENTASI PENGONTROL PID PADA SISTEM 2 DERAJAT KEBEBASAN UNTUK *COLOUR OBJECT TRACKING*

DESIGN AND IMPLEMENTATION PID CONTROLLER OF 2 DEGREE OF FREEDOM SYSTEM FOR COLOUR OBJECT TRACKING

M Zakaria¹, Suwandi², Reza Fauzi I³

Prodi S1 Teknik Fisika, Fakultas Teknik Elektro, Universitas Telkom

[1muhammadzakaria181094@gmail.com](mailto:muhammadzakaria181094@gmail.com), [2suwandi.sains@gmail.com](mailto:suwandi.sains@gmail.com), [3rezafauzii@gmail.com](mailto:rezafauzii@gmail.com)

Abstrak

Colour object tracking (COT) merupakan salah satu topik yang sering dibahas pada bidang *computer vision*. Teknologi COT sangat bermanfaat dalam kehidupan sehari-hari diantaranya digunakan dalam sistem pengawasan. Namun dalam sistem COT sering ditemui masalah akibat perubahan dimensi 3D menjadi 2D. Maka dibutuhkan suatu teknik yang dapat memperbaiki respon sistem dalam keadaan *transien* dan tunak agar sistem dapat mengikuti gerak objek dengan baik. Pada penelitian ini digunakan kontrol PID untuk mengendalikan pergerakan dengan 2 derajat sistem kebebasan dengan menggunakan metode *tunning* Ziegler-Nichols. Sistem menghasilkan respon *overshoot* di bawah 20 % dengan nilai *rise time* dan *settling time* di bawah 1 detik ketika sistem didalam area deteksi kamera.

Kata kunci : *computer vision, colour object tracking, pengontrol PID, ziegler-nichols*

Abstract

Colour object tracking (COT) is one of discussed topic on *computer vision*. COT Technology very usefull for daily live including surveillance system . Then, a technique that can fix response system in transient and steady state is required, thus the system can follow the object movement. In this study PID control is used to control the movement with 2 degree of freedom with addition Ziegler Nichols *tunning* method. The system produce *overshoot* respond under 20% with the value of *rise time* and *settling time* under 1 second in the detection area of camera.

Keywords : *computer vision, colour object tracking, PID controller, ziegler-nichols*

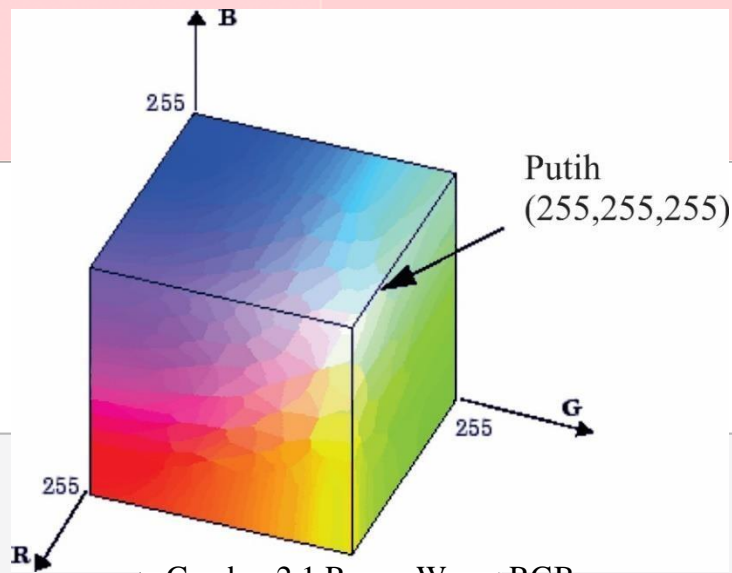
1. Pendahuluan

Saat ini perkembangan ilmu pengetahuan tumbuh dengan cepat. Dengan perkembangan ilmu pengetahuan diharapkan banyak teknologi-teknologi yang dihasilkan untuk membantu pekerjaan manusia. Saat ini ilmu pengetahuan selalu berkembang. Salah satu ilmu yang sedang berkembang saat ini adalah *computer vision*. *Computer vision* merupakan bidang pengetahuan yang berfokus pada bidang sistem kecerdasan buatan dan berhubungan dengan akuisisi dan pemrosesan citra [1]. *Colour object tracking* (COT) merupakan salah satu topik yang sering dibahas pada bidang *computer vision*, diantaranya untuk mengembangkan persepsi pandangan robot [3]. COT adalah teknologi yang mampu mendeteksi dan mengikuti objek sesuai dengan warnanya. Saat ini banyak sekali manfaat yang dihasilkan oleh teknologi COT. *Object tracking* memiliki peran yang sangat penting dalam aplikasi *computer vision* seperti pengawasan, *augmented reality*, *smart room*, *perceptual user interface*, dan *driver assistance* [4]. Namun teknologi *object tracking* sendiri memiliki beberapa masalah umum salah satunya adalah tidak bisa mendeteksi secara *real time* karena sistem kehilangan objek dan tidak mampu untuk menemukan objek secara cepat [5]. Maka dibutuhkan suatu teknik yang dapat memperbaiki respon sistem dalam keadaan *transien* dan tunak agar sistem dapat mengikuti gerak objek dengan baik.

2. Landasan Teori

2.1 Citra RGB

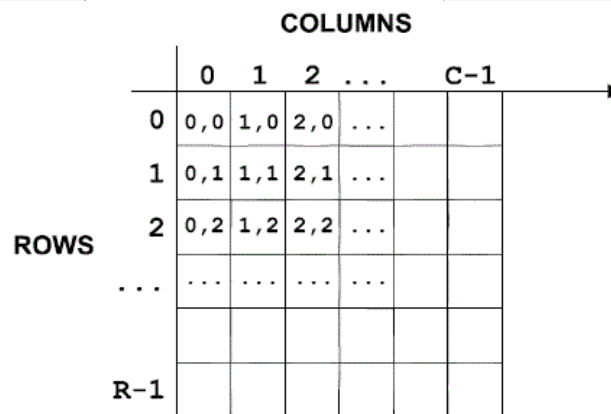
Citra RGB atau yang biasa disebut dengan citra *true colour*, disimpan dalam array berukuran $m \times n \times 3$ yang mendefinisikan warna merah, warna hijau dan warna biru untuk setiap *pixel*-nya. Warna pada setiap *pixel* ditentukan kombinasi merah, hijau dan biru. Format *file* citra menyimpan citra RGB sebagai citra 24 bit dengan disimpan masing-masing 8 bit ukurannya. Kombinasinya menjadi 16 juta warna, sehingga disebut *true colour* [7].



Gambar 2.1 Ruang Warna RGB

2.2 Sistem Koordinat Pksel

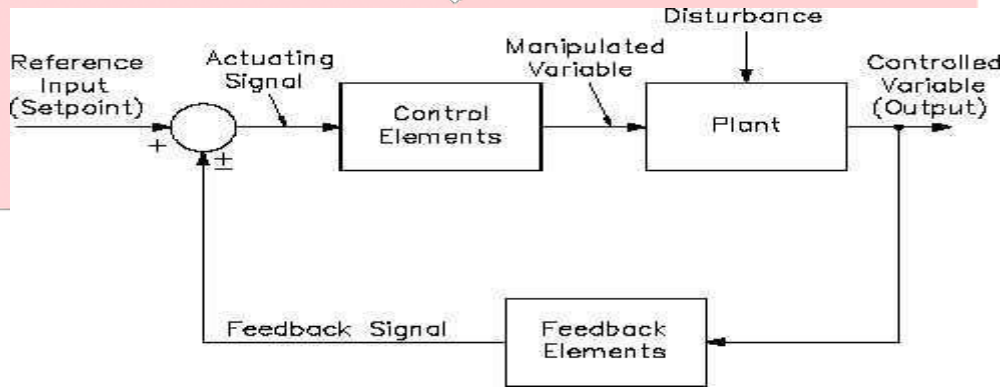
Secara umum, metode yang umum untuk mengekspresikan lokasi dari sebuah citra adalah menggunakan koordinat *pixel* [7]. Koordinat *pixel* ini dibutuhkan dalam menentukan posisi *object* target yang akan di-*tracking* nantinya.



Gambar 2.2 Ilustrasi Sistem Koordinat Pksel

2.3 Sistem Kontrol

Sistem kontrol suatu teknik untuk mengendalikan sebuah sistem dengan memanipulasi sinyal *error*, sehingga respon sistem (*output*) sama dengan yang diinginkan (*set point*). Pada sistem kontrol terdapat *sub* sistem dan proses yang ditujukan untuk menghasilkan keluaran sistem sesuai dengan yang diinginkan [8]. Jika keluaran dari sistem terdapat *error* maka *error* tersebut akan digunakan sebagai nilai pembanding dengan nilai referensi yang telah diberikan.



Gambar 2.3 Diagram Blok Sistem Kontrol Tertutup

2.4 Sistem Kontrol PID

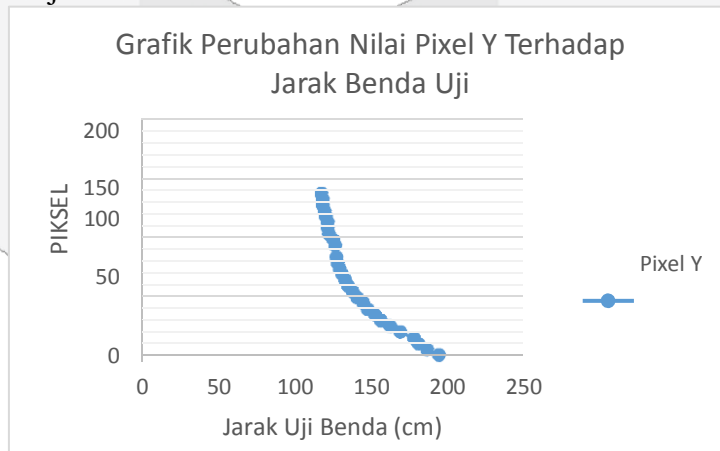
PID *controller* adalah bentuk sistem *feedback* yang paling umum digunakan.. Saat ini PID *controller* dapat ditemukan di semua area yang menggunakan sistem kontrol. Pada PID *controller* terdapat beberapa macam aksi kontrol yaitu kontrol proporsional, kontrol derivatif, dan kontrol integral. Pemilihan tipe kontrol yang digunakan tergantung kebutuhan dari *plant* yang akan menggunakan PID *controller*. Algoritma dari PID dapat dijelaskan dari persamaan 1.

$$e(t) = K_p e(t) + K_i \int e(t) dt + K_d \frac{de(t)}{dt} \tag{1}$$

3. Pembahasan

3.1 Uji Jarak Deteksi Sensor

Pengujian jarak deteksi sensor yaitu dengan meningkatkan jarak uji benda terhadap sistem COT untuk mengetahui jarak maksimum yang dapat dideteksi oleh sistem. Objek yang digunakan sebagai objek uji adalah bola berwarna merah dengan ukuran diameter 6 cm. Pada grafik 4.1 menunjukkan perubahan nilai piksel sumbu y terhadap nilai jarak benda uji.



Grafik 4.1 Perubahan Nilai Pixel Y Terhadap Jarak Benda Uji

Pada grafik di atas menunjukkan bahwa ukuran benda uji akan berkurang ketika jarak benda uji terhadap sistem dijauhkan. Perubahan nilai bersifat pada grafik bersifat kuadratik, ini menunjukkan bahwa ketika jarak uji bernilai disekitar 100 cm perubahan nilai piksel cukup besar namun ketika jarak uji bernilai disekitar 150 cm perubahan nilai piksel lebih kecil dan benda uji tidak terdeteksi ketika jarak uji di atas 200 cm dan jumlah piksel minimum yang dapat terdeteksi adalah 2.

3.2 Uji Sudut Deteksi Sensor

Pengukuran sudut deteksi sensor dilakukan untuk mengetahui seberapa besar area yang dapat dideteksi oleh sensor dan besar perubahan area objek ketika kamera dijauhkan dari area uji. Pengukuran dilakukan dengan meletakkan objek uji pada setiap tepi area yang dapat dideteksi kemudian diukur jarak antar tepi tersebut untuk mendapatkan sudut deteksi sensor.

Tabel 1 Hasil Pengukuran Sudut Deteksi Sensor

	Lebar (cm)	Panjang (cm)	Sudut X (°)	Sudut Y (°)
Jarak 60 cm	101	55	80.17	49.24
Jarak 100 cm	167	88	79.72	47.45
Rata-rata			79.94	48.345

Untuk sumbu x besar sudut yang dihasilkan adalah 79.94 derajat, sedangkan untuk sumbu y besar sudut yang dihasilkan 48.345 derajat.

3.3 Tuning Parameter PID

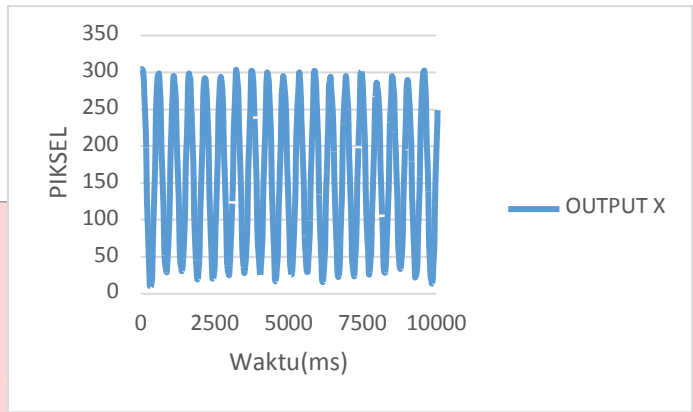
Tuning parameter PID yang digunakan menggunakan metode Ziegler-Nichols. Metode Ziegler-Nichols memiliki kelebihan dimana perancang tidak perlu membuat pemodelan sistem untuk melakukan *tuning*. Berikut adalah langkah-langkah yang dilakukan dalam melakukan *tuning* parameter PID menggunakan metode ZN.

- Rancang sistem *closed loop* dengan hanya menggunakan kontrol proporsional.
- Naikkan nilai K_p secara bertahap sampai sistem mendapatkan keadaan osilasi.
- Tentukan nilai K_p *ultimate* (K_{pu}) dan periode *ultimate* (P_u).

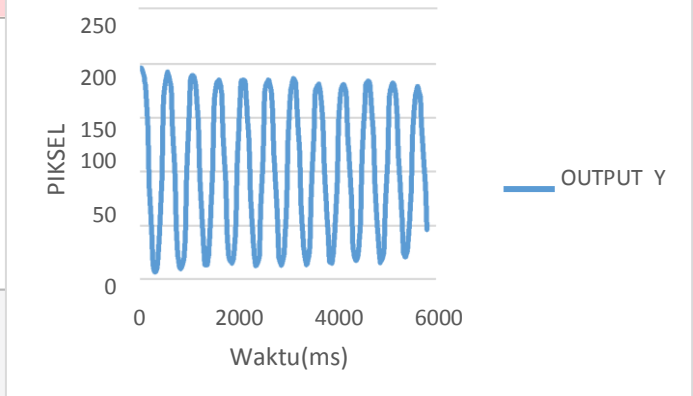
Setelah mendapatkan nilai K_{pu} dan P_u maka nilai K_p , T_i , dan T_d dapat ditentukan dengan menggunakan tabel 2.

Tabel 2 Persamaan *Tuning* PID Ziegler-Nichols

ZN-Method	K_p	T_i	T_d
P	$0.5K_{pu}$	-	-
PI	$0.45K_{pu}$	$P_u/1.2$	0
PID	$0.6K_{pu}$	$P_u/2$	$P_u/8$

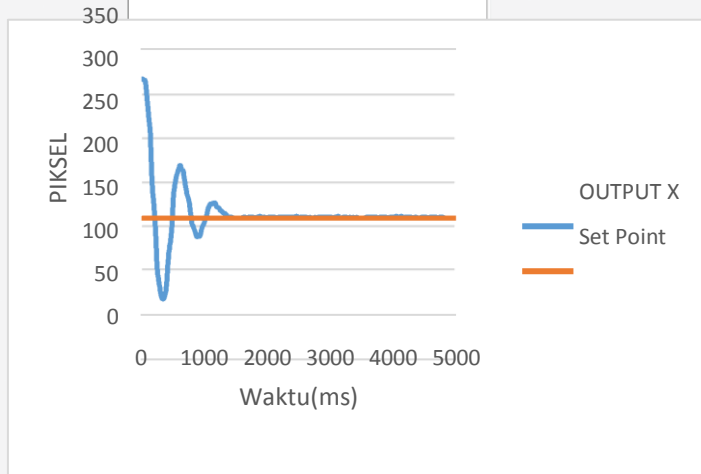


Grafik 4.2 Respon Sistem Sumbu X Dengan Kp 0.19

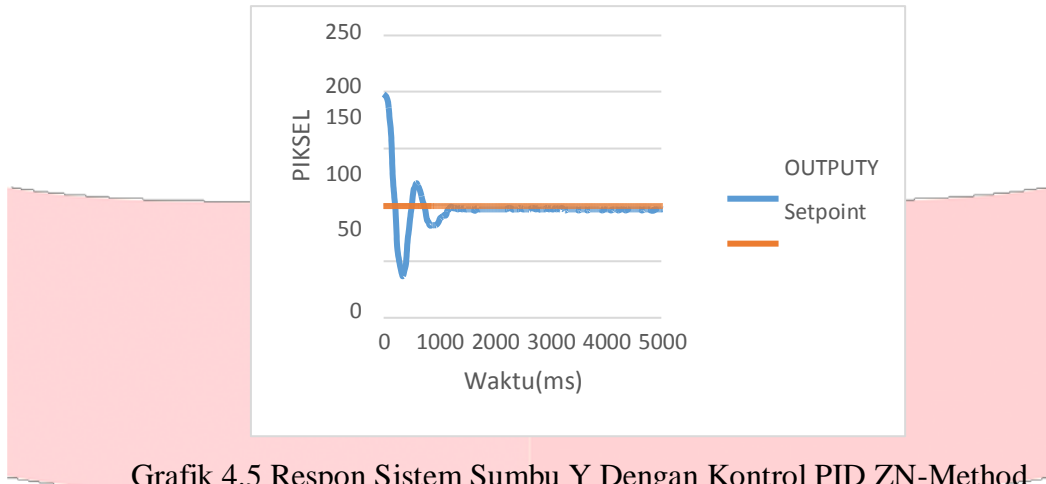


Grafik 4.3 Respon Sistem Sumbu Y Dengan Kp 0.19

Pada grafik di atas terlihat respon sistem dalam keadaan osilasi yang seragam sehingga dapat ditentukan bahwa nilai Kp saat 0.19 merupakan nilai Kpu. Kemudian sistem diuji respon hasil *tuning* menggunakan metode ZN.



Grafik 4.4 Respon Sistem Sumbu X Dengan Kontrol PID ZN-Method



Grafik 4.5 Respon Sistem Sumbu Y Dengan Kontrol PID ZN-Method

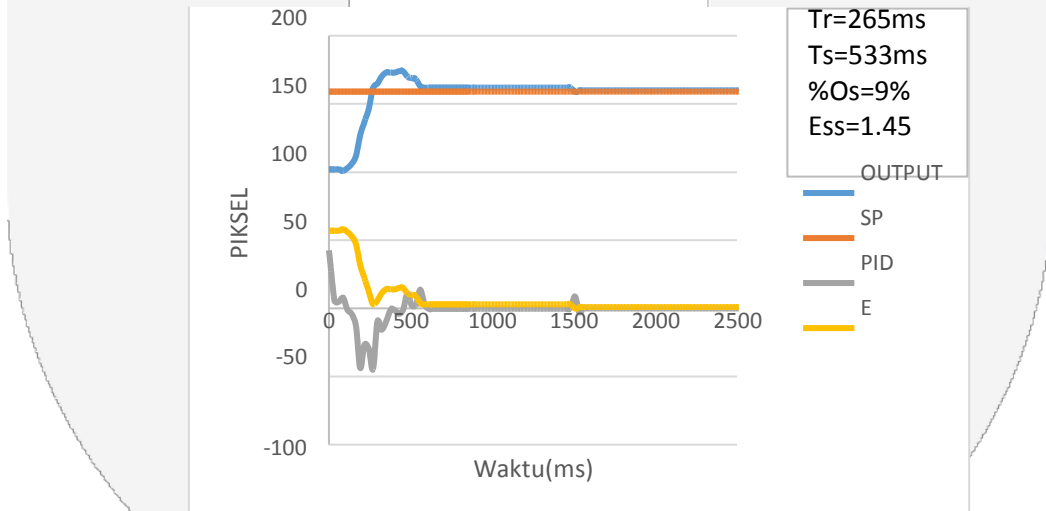
Pada grafik di atas dapat dilihat bahwa sistem memiliki nilai *overshoot* diatas 20% sehingga, untuk mendapatkan nilai *overshoot* dibawah 20% nilai *kp* dan *kd* harus diturunkan. Dengan melakukan *trial and error* maka persamaan yang digunakan adalah :

$$PI_{(s)} = (0.075 \frac{1}{s}) + (0.465 \int \frac{1}{s} ds) + (0.065 \frac{d}{dt}) \tag{4.1}$$

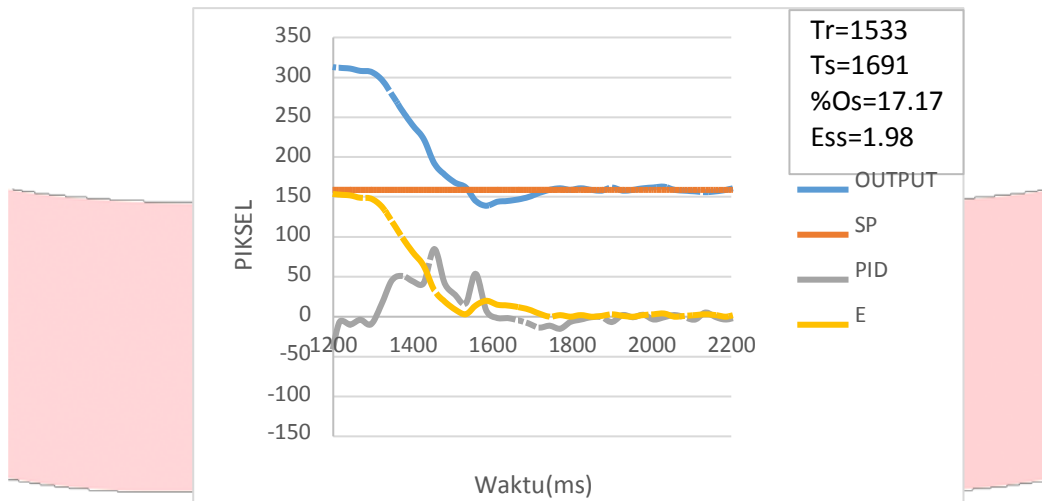
$$PI_{(s)} = (0.075 \frac{1}{s}) + (0.433 \int \frac{1}{s} ds) + (0.065 \frac{d}{dt}) \tag{4.2}$$

3.4 Uji Respon Kontrol

Uji coba sumbu x dilakukan dengan objek uji diletakkan pada area uji. Kemudian benda diletakkan dengan variasi perubahan sumbu x.



Grafik 4.1 Respon Sistem Sumbu X dengan Jarak Uji 60 cm dan Simpangan X 10 cm



Grafik 4.2 Respon Sistem Sumbu X dengan Jarak Uji 60 cm dan Simpangan X 60 cm

Nilai T_r merupakan nilai *rise time* yang menunjukkan waktu yang dibutuhkan sistem untuk memposisikan objek pada *set point* (titik tengah kamera). Sedangkan T_s merupakan nilai *settling time* yang menunjukkan waktu yang dibutuhkan sistem untuk stabil memposisikan objek (kondisi tunak) pada *set point*. Hasil pengujian menunjukkan bahwa ketika simpangan sumbu x diperbesar maka waktu yang dibutuhkan sistem mencapai titik tengah dan kondisi tunak semakin meningkat. Pada simpangan x 60 cm terjadi selisih nilai *rise time* 1268 ms dan *settling time* 1158 ms dari kondisi simpangan x 10 cm. Kedua nilai tersebut meningkat diakibatkan sistem membutuhkan putaran yang lebih jauh ketika nilai simpangan ditingkatkan. Pada simpangan sumbu x 60 cm sistem kehilangan objek sehingga sistem melakukan proses *scanning* terlebih dahulu, proses *scanning* tersebut membutuhkan waktu yang lebih lama untuk sistem bergerak menuju objek dan meningkatkan nilai *overshoot* menjadi 17.17%. Penyebab meningkatnya nilai *overshoot* adalah akibat keluaran PID yang maksimum dan sistem harus bergerak sebelum mendeteksi objek sehingga momen inersia sistem meningkat.

4. Kesimpulan

1. Sistem berhasil mendeteksi semua variasi warna merah dengan menjadikan 1 kategori warna merah
2. Sistem membutuhkan waktu 265 ms untuk mendapatkan titik tengah objek pada simpangan sumbu X sebesar 10 cm dengan jarak uji 60 cm
3. *Variable* respon sistem *rise time* turun ditingkatkan 6 ms, *settling time* turun 338 ms, dan *overshoot* turun 5.3 % ketika jarak uji ditingkatkan pada jarak 100 cm dan simpangan sumbu X sejauh 10 cm.

Daftar Pustaka

- [1] W. Budiharto and D. Purwanto, Robot Vision, Jakarta: CV ANDI, 2012.
- [2] D. H. Ballard and C. M. Brown, Computer Vision.
- [3] S. W. N. Rahman Arif, "Colored Ball Position Tracking Method for Goalkeeper Humanoid Robot Soccer," *TELKOMNIKA*, vol. 11, pp. 11-16, Maret 2013.
- [4] N. Endah S, A. Ali Husein, S. Ronny and K. F. Ade, "Design and Implementation of an Object Tracking System Control Using PID and Movement Prediction," *Industrial Electronics Seminar*, pp. 47-50, 2009.
- [5] K. Dinda, Embedded Visual Servoing, 2006.
- [6] C. S. Sherin Cherian, "REAL TIME IMPLEMENTATION OF OBJECT TRACKING THROUGH WEBCAM," *International Journal of Research in Engineering and Technology*, pp. 128-132, 2014.
- [7] M. C. Wijaya and A. Prijono, Pengolahan Citra Digital Menggunakan MATLAB, Bandung: Informatika Bandung, 2007.
- [8] N. S. Nise, Control System Engineering, California, 2003.