Indonesian Journal of Electronics and Instrumentation Systems (IJEIS)
Vol.x, No.x, April xxxx, pp. 1~5
ISSN (print): 2088-3714, ISSN (online): 2460-7681
DOI: 10.22146/ijeis.xxxx

Purwarupa *Multipurpose Tracking Camera* Menggunakan Metode *Object Tracking* CSR-DCF dan Kendali PID

Hafizh Zuumar Setiawan*[1], Bambang Nurcahyo Prastowo[2]
[1]Program Studi Elektronika dan Instrumentasi, FMIPA UGM, Yogyakarta, Indonesia
[2,3]Departemen Ilmu Komputer dan Elektronika, FMIPA UGM, Yogyakarta, Indonesia
e-mail: ***[1]hafizhzuumar@mail.ugm.ac.id** , [2]prastowo@mail.ugm.ac.id

Abstrak

Object tracking merupakan permasalahan menentukan lokasi, alur dan karakteristik dari objek yang ingin dideteksi menggunakan pengukuran dari sensor. Metode CSR-DCF mengabungkan antara algoritma discriminative correlation filter dan channel spatial reliability untuk dapat mendeteksi objek yang abstrak sehingga meningkatkan kemampuan deteksi objek. Tetapi permasalahan lain yang muncul adalah karena algoritma object tracking menerapkan prinsip online learning sehingga tidak dapat melakukan tracking apabila objek hilang dari frame.

Pada penelitian ini akan digunakan servo 2 sumbu sebagai penggerak pan-tilt camera agar sistem dapat mengikuti objek. Kendali yang digunakan pada sistem ini adalah PID dengan beberapa nilai kombinasi antara konstanta P, I dan D. Pengujian pada algoritma CSR-DCF juga dilakukan dengan mengubah nilai parameter terhadap objek yang berbeda. Dengan ini diharapkan akan didapat pengaruh dari parameter feature dari algoritma terhadap hasil tracking.

Dari hasil pengujian didapat bahwa nilai konstanta yang paling stabil yaitu nilai P = 1, I = 0.1 dan D = 60. Dengan konfigurasi ini sistem mengalami error paling sedikit. Kemudian pada pengujian parameter metode CSR-DCF, parameter use_color_names memiliki pengaruh yang besar terhadap objek yang dibuktikan dengan tingkat keberhasilan deteksi dengan penggunaan parameter tersebut.

Kata kunci*— CSR-DCF, PID, object tracking, motor servo, arduino*

Abstract

Object tracking is the problem of determining the location, path and characteristics of the object to be detected using measurements from sensors. The CSR-DCF method combines the discriminative correlation filter algorithm and channel spatial reliability to be able to detect abstract objects, thus increasing object detection capabilities. However, another problem that arises is because the object tracking algorithm applies online learning principles so it cannot track if the object disappears from the frame.

In this research, a 2-axis servo will be used to drive the pan-tilt camera so that the system can follow objects. The control used in this system is PID with several combination values between the constants P, I and D. Testing of the CSR-DCF algorithm is also carried out by changing parameter values for different objects. With this, it is hoped that the influence of the feature parameters of the algorithm on the tracking results will be obtained.

The test results showed that the most stable constant values were P = 1, I = 0.1 and D = 60. With this configuration the system experienced the fewest errors. Then, in testing the parameters of the CSR-DCF method, the use_color_names parameter has a huge influence on the object as evidenced by the success rate of detection using this parameter.

Keywords*—CSR-DCF, PID, object tracking, servo motor, arduino*

Received June 1st,2012; Revised June 25th, 2012; Accepted July 10th, 2012

1. PENDAHULUAN

Pada bidang *computer vision*, mendeteksi benda bergerak dari video yang berbasis *binary mask* pada setiap *frame*, merupakan isu penting dan menarik beberapa bidang pengguna *vision* seperti penerjemah gerakan, pengendalian lalu lintas, inspeksi industri, pengenalan prilaku manusia, dan pengawasan. Di banyak aplikasi dibidang tersebut, kamera yang bergerak terus – menerus digunakan. Sebagai contoh, pada beberapa kamera pengawasan cerdas, *pan-tilt-zoom* kamera digunakan agar lebih baik dalam mengikuti objek. Disini kamera dapat beroprasi dalam beberapa *degrees of movement* dan secara autonomi [1].

Object tracking merupakan salah satu masalah mendasar pada bidang *computer vision* dengan banyaknya aplikasi dari *object tracking* seperti pengawasan, interaksi manusia dengan komputer dan navigasi dari *autonomous vehicle* [2]. Definisi dari *Object tracking* adalah permasalahan menentukan lokasi, alur dan karakteristik dari objek yang ingin dideteksi menggunakan pengukuran dari sensor. Sensor dapat berupa alat apa saja yang dapat mengambil informasi seperti radar, sonar, ladar, kamera, sensor inframerah, microphone dan masih banyak lagi[3]. Proses *tracking* suatu objek diinisialisasi dengan menggambar *bounding box* pada objek yang akan di-*track* pada *frame* pertama, kemudian algoritma dari *object tracking* akan melakukan estimasi terhadap keadaan objek seperti lokasi dan ukuran objek pada *frame* selanjutnya [4].

Banyak tantangan yang dihadapi algoritma *object tracking*. Beberapa tantangan yang dihadapi oleh algoritma *object detection* yaitu perubahan bentuk dari objek (*deformation*), kecepatan objek yang berubah, perubahan pencahayaan, terhalangnya objek (*occlusion*), dan hilangnya target dari kamera [4]. Dalam kasus kamera bergerak, salah satu strategi untuk membedakan pergerakan objek dengan *background* ada 2 kategori utama. Yaitu berbasis *backgound modeling* dimana algoritma mencoba untuk membuat *background* pada setiap *frame* menggunakan *motion compensation method.* Yang kedua adalah *trajectory classification* dimana *trajectory* dihitung untuk mendapatkan *feature* dari objek [1].

Pemanfaatan dari *object tracking* dan kendali PID telah banyak dilakukan sebelumnya. Percobaan yang dilakukan oleh [5] membuat *following robot* yaitu robot beroda yang dapat mengikuti target dengan pemanfaatan *color filtering* dari target dan bergerak menggunakan kendali PID. Selanjutnya, [6] menggunakan algoritma *object tracking* TLD untuk mencari target dan kendali PID pada *drone* untuk dapat mengetahui posisi dari target pada saat tidak dapat terlihat oleh mata.

Untuk itu, maka dibuatlah prototipe kamera bergerak yang terdiri dari 2 *degree of freedom* (DOF) yaitu *pitch* dan *yaw*, dengan masing masing dari DOF dikendalikan dengan kendali PID yang berbeda. Nilai *set-point* diambil dari posisi tengah dari *bounding box* objek pada kamera. Nilai dari *current state* diambil dari nilai tengah kamera. Sehingga *current state* dari sistem akan berusaha untuk mendekati *set-point* untuk menghilangkan *error* pada sistem. Dengan itu maka kamera akan terus memperbaiki posisi selama *error* masih ada dalam sistem.

2. METODE PENELITIAN

Pada bab ini akan dijelaskan mengenai perancangan dan analisis sistem dari *tracking camera*. Mulai dari analisis dari sistem, rancangan perangkat keras dari sistem dan rancangan perangkat lunak dari sistem.

2.1 Analisis Sistem

Sistem yang dirancang pada penelitian ini berfungsi untuk dapat mengikuti objek yang di-*tracking*. Sehingga untuk mewujudkan tujuan tersebut, maka akan digunakan perangkat keras yang ditunjukkan pada Tabel 1.

Tabel 1 Perangkat keras

No.	Nama	Fungsi
1	Webcam Logitech	Melakukan pengangkapan gambar dari target.
2	Motor Servo MG90S	Sebagai aktuator untuk mengedalikan kamera bedasarkan input dari Arduino Uno
3	Arduino Uno	Sebagai mikrokontroler untuk meneggerakkan servo ke posisi *ser-point*
4	Pan/Tilt Bracket	Rangka *yaw* dan *pitch*
5	Laptop	Memproses metode CSR-DCF terhadap gambar kamera

Webcam digunakan untuk mengambil gambar dari objek yang akan terhubung ke laptop [7]. *Webcam* dan *servo* terpasang pada *pan/tilt bracket* agar masing – masing sumbu *yaw* dan *pitch*, dan pengambilan gambar dari *webcam* stabil. *Webcam* kemudian dihubungkan ke laptop dengan kabel USB untuk mengirim gambar [8]. Laptop kemudian digunakan untuk memproses algoritma *object tracking* dan perhitungan PID. Perhitungan kemudian dikirim dalam bentuk sudut kepada Arduino Uno agar dapat menggerakkan *servo*. Untuk mendukung perangkat keras, maka diperlukan perangkat lunak yang dapat dilihat pada Tabel 2.

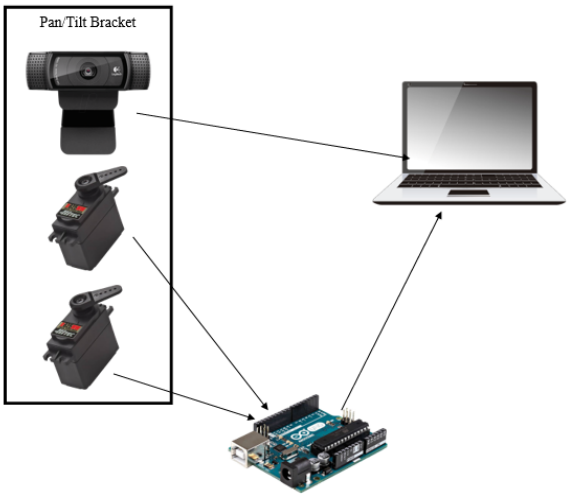
Tabel 2. Perangkat Lunak

No.	Nama	Fungsi
1	Python	Library Python untuk memanggil metode CSR-DCF, visualisasi data dan perhitungan PID
2	Arduino IDE	IDE untuk menggerakkan motor servo

Bahasa pemrograman Python memiliki banyak kegunaan pada perangkat lunak sistem. Python digunkaan untuk memanggil metode CSR-DCF dengan menggunakan *library* CV2. Python juga digunakan untuk melakukan perhitungan PID karena sangat mudah dan cepat dalam melakukan perhitungan PID dan juga dapat mengirim data tersebut kepada Arduino Uno. Visualisasi data menggunakan *library* dari python yaiu matplotlib untuk menggambar grafik yang nantinya dapat ditampilkan berdasarkan nilai x dan y yang dimasukkan kedalam fungsi. Arduino IDE digunakan untuk memprogram Arduino Uno sehingga dapat memproses data hasil perhitungan PID yang masuk kepada *servo* agar dapat bergerak mendekati *set-point.*

2.2 Rancangan Perangkat Keras

Pada Tabel 1, telah dijelaskan perangkat keras apa saja yang diperlukan oleh sistem, sehingga perlu dijelaskan bagaimana interkoneksi dari perangkat keras tersebut pada sistem. Interkoneksi perangkat keras dapat dilihat pada Gambar 1.

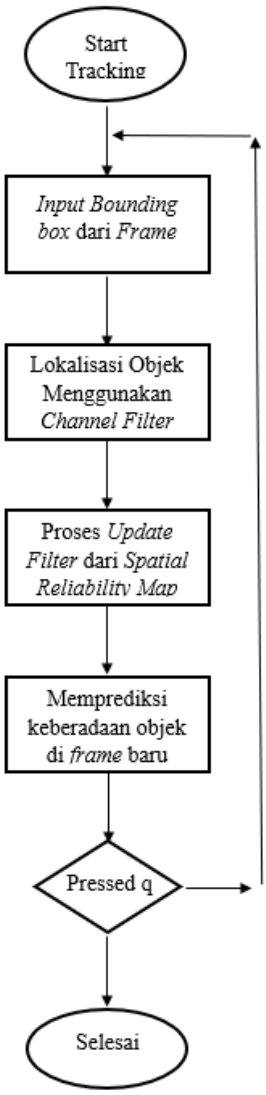


Gambar 1 Interkoneksi perangkat keras

Bracket berfungsi sebagai tempat dari *servo* dan *webcam* sehingga *axis* dari *servo* dapat stabil pada sumbu *yaw* dan *pitch*. *Webcam* juga terpasang pada *bracket* agar stabil dalam mengambil gambar target. Kabel dari *servo* terhubung pada arduino untuk mengirim data sudut. Laptop juga terhubung pada Arduino Uno untuk mengirim hasil perhitungan PID dan terhubung pada *webcam* untuk mendapatkan gambar dari objek.

2.3 Rancangan Perangkat Lunak

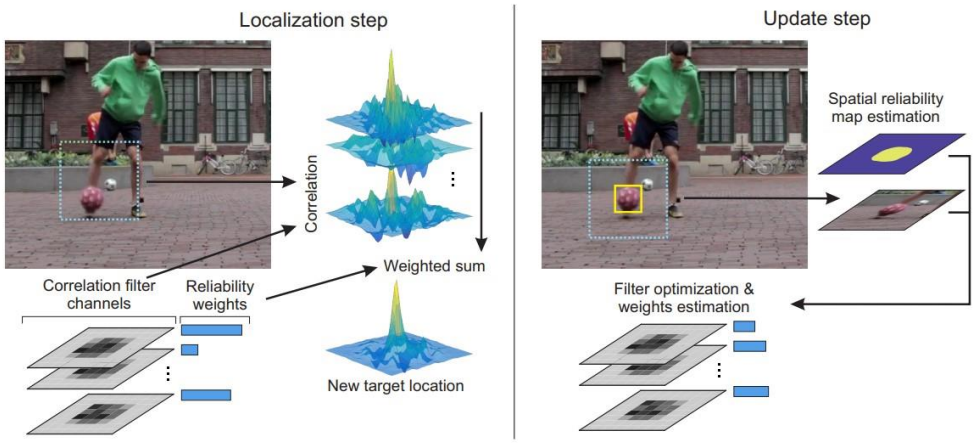
Perangkat lunak pada sistem digunakan untuk melakukan perhitungan terhadap algoritma CSR-DCF, perhitungan PID, visualisasi data dan juga mengerakkan *servo*. Perangkat lunak yang digunakan berdasarkan Tabel 2 adalah bahasa pemrograman Python dan Arduino IDE. Python memiliki banyak fungsi pada sistem ini salah satunya adalah proses *tracking*. Diagram dari proses *tracking* ditunjukkan pada Gambar 2.



Gambar 2 Diagram proses tracking algoritma CSR-DCF

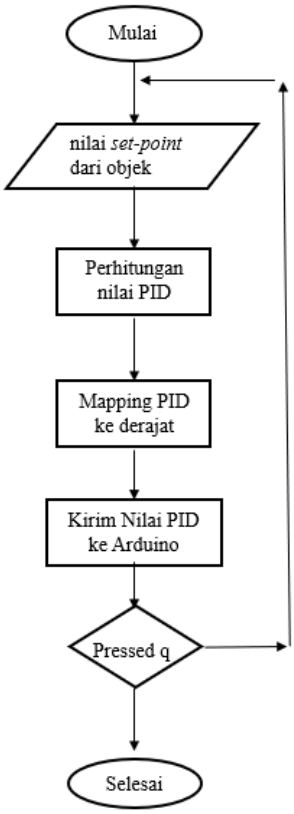
Proses *tracking* diawali dengan inisialisasi dari *bounding box* target yang kemudian algoritma akan belajar fitur – fitur dari target tersebut. Pada tahap lokalisasi, algoritma akan mencari maka yang kemungkinan besar merupakan objek didalam *bounding box.* Ini dilakukan dengan melakukan korelasi antara nilai fitur dari objek dan *correlation filter* [10]. Kemudian dihasilkan *correlation response* dari filter tersebut. Nilai respon dari korelasi tersebut menandakan bahwa objek kemungkinan besar ada pada area dengan nilai respon yang tinggi sehingga objek dapat dilokalisasi pada bagian tersebut.

Pada tahap *update,* objek yang sudah dilokalisasi akan dipisah antara objek tersebut dan *background* sehingga didapat murni bentuk objek itu sendiri. Kemudian *spatial reliability map* akan dibuat dari bentuk objek tersebut menggunakan teori bayes. Nilai dari filter juga dioptimasi dari *spatial reliability map* dan kemudian *weight* dari masing masing *channel feature* diperbarui. Proses dari algoritma CSR-DCF dapat dilihat pada Gambar 3.



Gambar 3 Proses lokasisasi dan update dari algoritma CSR-DCF

Proses selanjutnya adalah kendali PID. Kendali PID merupakan kendali yang mengimplementasikan feedback pada input sehingga dapat diketahui perbedaan antara nilai yang diinginkan (set point) dan nilai keluaran sebenarnya [11]. Nilai piksel yang telah didapat dari nilai tengah *bounding box* akan menjadi nilai *set-point* dari PID. Sehingga perhitungan *error* dapat dicari dengan membandingkan nilai *set-point* dengan nilai tengah dari *frame* kamera. Dengan itu maka komputasi PID dapat dilakukan. Setelah komputasi maka nilai piksel akan diubah ke derajat sudut menggunakan *mapping* nilai piksel ke sudut. Setelah *mapping* maka akan didapat nilai sudut yang kemudian dikirim ke Arduino Uno untuk kemudian diproses agar dapat menggerakkan *servo*. Diagram dari proses PID tersebut dapat dilihat pada Gambar 4.



Gambar 4 Diagram proses kendail PID

2.4 Rancangan Pengujian

Kemampuan *tracking* dari sistem akan diuji berdasarkan skenario-skenario pengujian. Skenario dari pengujian algoritma *tracking* dan kendali PID dibagi menjadi 2 skenario utama yaitu perubahan nilai konstanta PID dan skenario perubahan objek pada algoritma CSR-DCF. Pada skenario perubahan nilai konstanta PID, objek yang akan di-*tracking* adalah wajah dengan gerakan ke kanan sebesar 60 cm, ke kiri sebesar 60 cm, ke bawah sebesar 60 cm dan ke atas sebesar 20 cm. Setelah berada pada posisi tersebut, objek wajah akan diam selama 2 detik untuk melihat respon PID dari sistem. Pada skenario perubahan objek, 2 objek yang akan diuji adalah wajah dan kemasan *lotion.* Selain itu, algoritma akan diuji dari segi perubahan parameter dan juga terhalang objek lain dan perubahan bentuk dari objek tersebut. Skenario tersebut dapat dilihat pada Tabel 3.

Tabel 3 Skenario pengujian sistem

No.	Skenario pengujian	Aspek yang diuji
1	Perubahan Nilai Konstanta PID	Melihat kecepatan *settling time* dari kendali PID *yaw* dan *pitch* selama 2 detik
2	Perubahan parameter algoritma	Pengaruh parameter terhadap hasil *tracking* dari algoritma
3	Target terhalang dan berubah bentuk	Keberhasilan algoritma dalam melakukan *track* terhadap perubahan objek

3. HASIL DAN PEMBAHASAN

3.1 Format Pengujian Sistem

Berdasarkan rancangan pengujian yang dibahas pada bab sebelumnya. 2 skenario yang diujikan adalah perubahan nilai konstanta PID dan perubahan objek pada algoritma CSR-DCF. Pada pengujian skenario pertama, nilai dari konstanta PID telah ditentukan sebelumnya berdasarkan Tabel 4.

Tabel 4 Nilai konstanta PID

No	Konstanta P	Konstanta I	Konstanta D
1	0.7	0.1	30
2	0.8	0.1	40
3	0.9	0.1	50
4	1.0	0.1	60
5	1.1	0.1	70
6	0.9	0	50
7	1.0	0	60
8	1.1	0	70

Pengujian kedua menilai bagaimana performa dari algoritma CSR-DCF. Disini, objek wajah akan menguji kemampuan fitur *use_hog* dari algoritma dan kemasan *lotion* menguji fitur *use_color_names* karena kemasan memiliki warna yang kompleks. Sehingga 2 fitur tersebut dipilih untuk pengujian ditambah 1 fitur lagi berupa *window_function* untuk mengetes apakah

perbedaan *window function* berpengaruh pada hasil deteksi objek. Fitur tersebut akan dikombinasikan sesuai Tabel 5.

Tabel 5 Kombinasi pengujian parameter CSR-DCF

No	*use_hog*	*use_color_names*	*window_function*
1	*True*	True	Hann
2	True	False	Hann
3	False	True	Kaiser
4	False	False	Kaiser
5	*True*	True	Kaiser
6	True	False	Kaiser
7	False	True	Hann
8	False	False	Hann

3.2 Hasil Pengujian Perubahan Nilai Konstanta PID

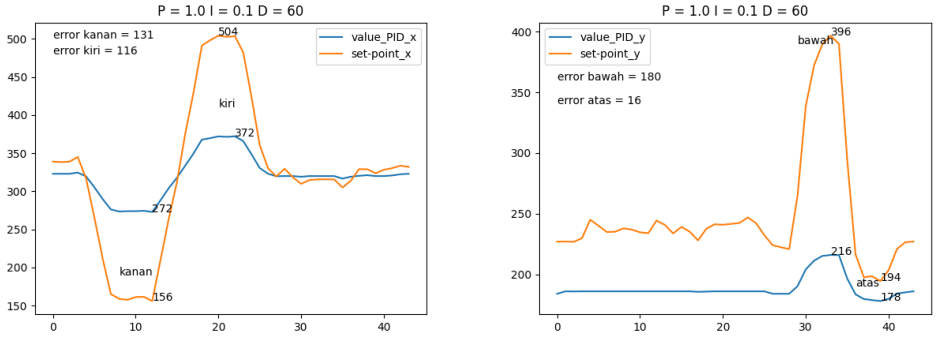
Pengujian dilakukan dengan memasukkan masing – masing nilai konstanta P, I dan D pada sistem sesuai dengan nilai pada Tabel 4. Setelah itu, respon PID akan divisualisasikan dalam bentuk grafik sehingga mudah untuk melakukan analisis. Dalam grafik tersebut, dapat dilihat nilai *error* dari sistem dan kestabilan sistem saat objek diam selama 2 detik sesuai dengan skenario pengujian. Hasil dari pengujian yaitu nilai *error* dan keberhasilan pengujian akan disimpan untuk analisis lebih lanjut.

Berdasarkan dari percobaan skenario pertama yaitu perubahan nilai konsanta PID, didapat hasil pada Tabel 6

Tabel 6 Hasil pengujian perubahan nilai konstanta PID

No	P	I	D	Kanan	Kiri	Bawah	Atas	Total	Hasil
1	0.7	0.1	30	143	157	212	1	513	Berhasil
2	0.8	0.1	40	151	129	201	8	489	Berhasil
3	0.9	0.1	50	148	118	204	11	481	Berhasil
4	1.0	0.1	60	131	116	180	16	443	Berhasil
5	1.1	0.1	70	110	121	178	136	545	Gagal
6	0.9	0	50	139	142	199	16	496	Berhasil
7	1.0	0	60	137	115	174	18	444	Berhasil
8	1.1	0	70	146	125	170	26	467	Berhasil

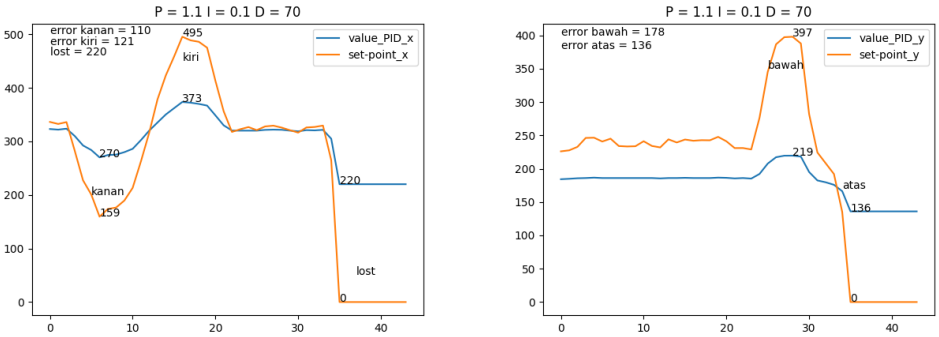
Dari Tabel 6 dapat dilihat bahwa nilai konstanta terbaik ada pada pengujian keempat dengan nilai total *error* adalah 443. Nilai tersebut merupakan nilai total dari akumulasi *error* terhadap gerakan dalam sumbu *yaw* yaitu ke kanan dan kiri, dan *error* terhadap sumbu *pitch* yaitu ke bawah dan atas. Pengujian keempat juga berhasil dalam mendeteksi objek secara keseluruhan dalam rentang waktu deteksi tanpa adanya kegagalan. Grafik dari pengujian keempat dapat dilihat pada Gambar 5.



Gambar 5 Hasil pengujian konstata PID keempat

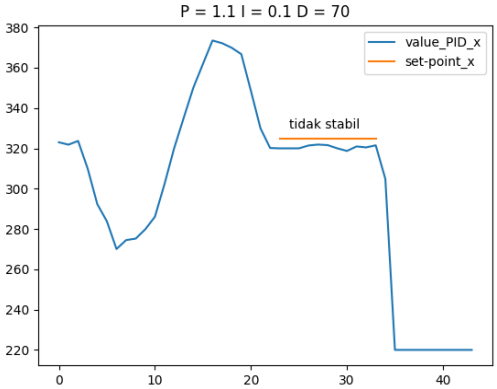
Dari grafik Gambar 5 tersebut dapat dilihat bahwa sistem secara keseluruhan stabil walaupun nilai *steady-state error* masih tinggi saat objek diam selama 2 detik. *Steady-state error* merupakan kondisi ketika respon dimasukkan dengan waktu yang sangat lama [9]. Tetapi disini waktu dibatasi sebesar 2 detik untuk melihat seberapa cepat kestabilan sistem dapat tercapai. Ini dapat diflihat pada grafik Gambar 5 dengan perbedaan nilai *set-point* dan nilai keluaran dari PID. Pada keluaran PID sumbu *pitch*, terlihat ada *steady-state error* dari awal program dijalankan sebesar kurang lebih 30 piksel. Ini disebabkan karena objek berada lebih atas dari sudut kamera yang membuat *error* bawah menjadi besar.

Berdasarkan Tabel 6, hanya ada 1 percobaan yang gagal yaitu percobaan kelima. Grafik dari percobaan kelima dapat dilihat pada Gambar 6.



Gambar 6 Hasil pengujian konstanta PID kelima

Dari grafik Gambar 6 dapat dilihat saat percobaan kelima, sistem menjadi tidak stabil saat objek naik pada sumbu *pitch*. Disini, sistem sudah tidak mengikuti objek lagi sehingga bisa dianggap *lost* atau kehilangan objek. Nilai *error lost* pada sumbu *yaw* adalah 220 piksel dan pada sumbu *pitch* adalah 136 piksel. Grafik dapat dianalisis lebih lanjut dengan melihar grafik Gambar 7.



Gambar 7 Grafik sumbu *yaw* pengujian kelima

Dari grafik pada Gambar 7, dapat dilihat bahwa nilai *yaw* tidak stabil sebelum sistem kehilangan objek. Dapat dilihat terdapat osilasi pada grafik sebelum kehilangan objek. Ketidakstabilan bisa jadi disebabkan oleh perubahan nilai konstanta P sebesar 0.1 dari yang sebelumnya pengujian keempat yang merupakan hasil pengujian terbaik.

3.2 Hasil Pengujian perubahan objek pada algoritma CSR-DCF

Pengujian perubahan objek algoritma CSR-DCF dilakukan berdasarkan kombinasi Tabel 5, dimana total ada 8 kombinasi percobaan. Parameter yang diuji yaitu *use_hog, use_color_names* dan *window_function.* HOG atau *histogram of oriented gradient* merupakan salah satu hasil dari ekstraksi fitur dari algoritma CSR-DCF, dimana HOG melihat perubahan piksel dengan piksel sekelilingnya untuk mendapatkan *gradient. Use_color_names* merupakan fitur warna dari algoritma. Total warna yang digunakan untuk ekstraksi fitur ada 16 warna. Respon dari warna ini akan disatukan dengan respon yang lain seperti HOG dan *greyscale* dalam satu histogram untuk melihat mana respon fitur yang paling berpengaruh dalam mendeteksi objek. Fungsi *window* diperlukan dalam perhitungan algoritma karena menggunakan *Fast Fourier Transform* (FFT) dalam menghitung *Discriminatif Correlation Filter* (DCF). Ini membuat adanya pilihan *window function* yang dapat digunakan saat menggunakan algoritma.

Dari hasil pengujian 8 kombinasi sesuai dengan Tabel 5, maka didapat hasil pengujian yang dapat dilihat pada Tabel 7.

Tabel 7 Hasil pengujian perubahan objek algoritma CSR-DCF

No	Wajah	Kemasan *lotion*	FPS
1	Berhasil	Berhasil	12
2	Berhasil	Gagal	7
3	Berhasil	Berhasil	9
4	Berhasil	Gagal	10
5	Berhasil	Berhasil	11
6	Berhasil	Gagal	10
7	Berhasil	Berhasil	9
8	Berhasil	Gagal	7

Dari tabel tersebut dapat dilihat bahwa *parameter use_color_names* sangat berpengaruh apabila mendeteksi benda berwarna. Ini dapat dibuktikan dari apabila kita tidak menggunakan parameter tersebut maka objek yang berwarna yaitu kemasan *lotion* gagal terdeteksi atau salah deteksi. Berbeda dengan wajah yang tidak memiliki warna yang kompleks, maka tetap dapat terdeteksi.

Dari segi fps algoritma, cenderung mengalami penurunan apabila algoritma gagal dalam mendeteksi objek. Ini mungkin disebabkan oleh *region of interest* dari algoritma yang menjadi besar ketika kehilangan objek sehingga berat untuk menghitung *feature* pada ROI tersebut.

Gambar 8 Kegagalan algoritma saat mendeteksi kemasan *lotion*

4. KESIMPULAN

Berdasarkan pengujian dan hasil yang diperoleh dalam 2 skenario pengujian tersebut, didapat kesimpulan bahwa masing masing pengujian berpengaruh terhadap sistem dalam 2 hal yang berbeda.

Pengujian perubahan nilai konstanta dari PID menguji kestabilan dari sistem terhadap terget yang dideteksi. Dari nilai nilai yang telah diuji pada percobaan, didapat nilai dari konstanta PID terbaik yaitu pada percobaan keempat dengan nilai konstanta P = 1, I = 0,1 dan D = 60. Ini dibuktikan dengan hasil *error* total paling sedikit dibanding percobaan lainnya. Secara keseluruhan semua percobaan sistem stabil kecuali pada percobaan kelima dikarenakan sistem mengalami osilasi.

Pada pengujian perubahan objek pada algoritma CSR-DCF, didapat hasil terbaik ketika *parameter use_hog* dan *use_color_names* digunakan. Pada saat parameter *use_color_names* tidak digunakan, maka sulit untuk CSR-DCF mendeteksi objek berwarna. Ini dibuktikan ketika pendeteksian kemasan *lotion.* Parameter *window function* tidak terlalu berpengaruh terhadap hasil dari *tracking,* hanya berpengaruh terhadapt nilai fps dari algoritma.

5. SARAN

Berdasarkan penelitian yang telah dilakukan, terdapat beberapa aspek yang perlu diperbaiki dan dikembangkan. Pertama, dengan membandingkan performa metode *object tracking* lain seperti *MIL Tracker*, *MOSSE Tracker*, *GOTURN Tracker,* dan masih banyak lagi. Kedua, menggunakan *servo* dengan resolusi sudut lebih kecil dari 1 derajat agar sistem menjadi lebih stabil. Ketiga, melakukan komputasi pada *edge* atau *cloud* agar alat menjadi lebih ringkas

DAFTAR PUSTAKA

[1] M. Yazdi, T. Bouwmans, "New trends on moving object in video images captured by a moving camera: A survey", *Computer Science Review*, vol 28, pp. 157-177, Mei. 2018 [Online]. Available: https://www.sciencedirect.com/science/article/abs/pii/S1574013716301794?via%3Dihub

[2] A. Yilmaz, dan O. Javed, "Object Tracking : a survey, ACM Computing Surveys", *ACM Computing Surveys*, No. 4, vol. 38, pp. 1-45, Des 2006 [Online]. Available: https://www.researchgate.net/publication/220566062_Object_tracking_a_survey_ACM_Comput_Surv.

[3] S. Challa, M. R. Morelande, D. Musicki dan R. J. Evans, "Fundamentals of Object Tracking", 2011, *Cambridge University Press, New York*.

[4] C. Ma, J. B. Huang, X. Yang dan M. H. Yang, "Adaptive Correlation Filter with Long-Term and Short-Term Memory for Object Tracking", *International Journal of Computer Vision*, vol. 126, pp. 771-796, Mar 2016 [Online]. Available: https://link.springer.com/article/10.1007/s11263-018-1076-4.

[5] C. S. Pati dan R. Kala, "Vision-Based Robot Following Using PID Control", *Technologies*, vol. 5, Jun 2017 [Online]. Avalilable: https://www.mdpi.com/2227-7080/5/2/34.

[6] V. Mariappan, M. Lee, M. Cho dan J. Cha, "OnBoard Vision Based Object Tracking Control Stabilization Using PID Controller", *International Journal of Advanced Culture Technology*, No. 4, vol, 4, pp. 81-86, Dec 2016 [Online]. Avaibale: https://www.researchgate.net/publication/315936252_OnBoard_Vision_Based_Object_Tracking_Control_Stabilization_Using_PID_Controller.

[7] M. Rouse, "What Does Webcam Mean?", Jun 2017 [Online]. Available: https://www.techopedia.com/definition/5333/webcam. [Accessed: 5-Oktober-2023]

[8] O. Brown, "What Is a Webcam? Here'a All You Need to Know", Aug 2023 [Online]. Available: https://www.obsbot.com/blog/webcam/what-is-webcam. [Accessed: 5-Oktober-2023]

[9] K. Ogata, "Modern Control Engineeering", 2010 *fifth edition, Pretice hall, New Jersey*.

[10] A. Lukezic, T. Vojir, C. L. Zajc, J. Matas dan M. Kristan, "Discriminative Correlation Filter Tracker with Channel and Spatial Reliability", *International Journal of Computer Vision*, vol. 126, pp. 671-688, Jan 2018 [Online]. Available: https://link.springer.com/article/10.1007/s11263-017-1061-3.

[11] K. J. Astrom dan T. Hagglund, "Advanced PID Control", 2006, *ISA – Instrumentation, Systems, and Automation Society, New York.*