



Implementasi Algoritma Correlation Tracker Untuk Sistem Pencarian Objek Menggunakan Library DLIB

Fajar Baskoro Ag

Program Studi Teknik Elektro Universitas Panca Budi, Medan, Indonesia

Email : Fajarbaskoroag@gmail.com

Abstract

Object tracking is a process to determine and follow the position of a desired moving object using a camera in real time. There are various algorithms in making object tracking systems, one of which is the correlation tracker algorithm in the DLIB library. This object tracking system uses Raspberry Pi 4 which contains a python program for object tracking. Pi Camera is used as a realtime video image input. The results of using the correlation tracker algorithm in the DLIB library, namely the object tracking process can run well. There are 5 objects that are tracked and as a result the tracking box is always around the object, whether the object is moving left, right, forward, and backward. As the object moves back and forth, the scale on the tracking box may change according to the object's size. Performance on the Raspberry Pi can be seen from the video framerate of 10.5 fps.

Keywords: *Correlation Tracker Algorithm, Dlib, Object Tracking, Raspberry PI*

Abstrak - Pelacakan objek merupakan suatu proses untuk menentukan dan mengikuti posisi dari sebuah objek bergerak yang diinginkan menggunakan kamera secara realtime. Ada berbagai macam algoritma dalam pembuatan sistem pelacakan objek, salah satunya adalah algoritma correlation tracker pada pustaka DLIB. Sistem pelacakan objek ini menggunakan Raspberry Pi 4 yang berisi program python untuk pelacakan objek. Pi Camera digunakan sebagai input gambar video realtime. Hasil penggunaan algoritma correlation tracker yang ada pada pustaka DLIB, yaitu proses pelacakan objek dapat berjalan dengan baik. Terdapat 5 objek yang dilacak dan hasilnya kotak pelacakan selalu berada di sekitar objek, baik saat objek bergerak ke kiri, ke kanan, maju dan mundur. Apabila objek bergerak maju dan mundur, skala pada kotak pelacakan dapat berubah mengikuti ukuran objek. Performa pada Raspberry Pi dapat dilihat dari framerate video sebesar 10,5 fps.

Kata kunci: Algoritma Correlation Tracker, Dlib, Pelacakan Objek, Raspberry PI

1. PENDAHULUAN

Penelitian di bidang *Computer Vision* telah berkembang cukup pesat. Terdapat berbagai topik penelitian di dalamnya, diantaranya pengolahan citra, citra geometri, pengenalan pola, rekonstruksi *3D*, dan sebagainya[1]. Pelacakan objek juga menjadi salah satu topik yang populer di bidang *computer vision*[2]. Pelacakan objek merupakan suatu proses untuk menentukan dan mengikuti posisi dari sebuah objek bergerak yang diinginkan menggunakan kamera[1][3]. Terdapat berbagai permasalahan pengolahan citra yang dapat diselesaikan dengan sistem pelacakan objek, misalnya pelacakan pergerakan bola dalam permainan sepak bola, *traffic monitoring*, *multiple tracking* manusia pada CCTV, sistem navigasi robot, dan lain sebagainya[3][4].

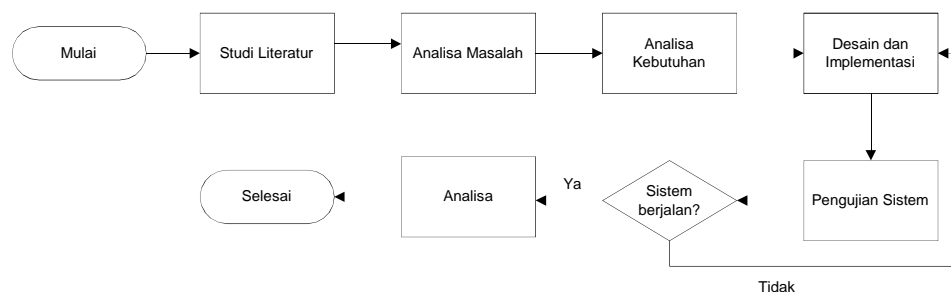
Ada berbagai algoritma untuk membuat sistem pelacakan objek. Pada penelitian yang dilakukan oleh Gigih Samudra, dkk [5] yaitu membuat sistem yang dapat menemukan objek dan menuju ke posisi objek, dalam hal ini objek berbentuk lingkaran dan kotak. Metode yang digunakan adalah dengan menemukan titik tengah objek dan dibandingkan dengan titik pusat *frame*. Penelitian yang dilakukan oleh Muhammad Izzuddin Mahali dkk [3] yaitu menggunakan algoritma *meanshift* dalam proses pelacakan benda. Penelitian Dian Rosdiana [6] yaitu dilakukan pemisahan antara latar belakang dan objek dengan menggunakan metode *K-Means* yang dibutuhkan untuk melakukan pelacak gerakan. Penelitian Ahmad Fahriannur dkk [1] yaitu membuat sistem menggunakan komputer, *webcam*, *servo*, dan *arduino*. Metode pelacakan yang digunakan yaitu *optical flow* dan *canny edge detection*. Penelitian yang dilakukan oleh Ravindra

R. Patil dkk [7] yaitu membandingkan *API* dari *OpenCV* yang terdiri dari *BOOSTING*, *TLD*, *MEDIANFLOW*, *MIL*, *MOSSE*, *CSRT*, *KCF*, dan *GOTURN*. *Hardware* yang digunakan adalah *Raspberry Pi 3* model *B+* dan *Processor Intel ULV* untuk menjalankan program *OpenCV* serta *webcam* untuk menangkap *video realtime*.

Penelitian yang dilakukan saat ini yaitu menggunakan algoritma *correlation tracker* yang terdapat pada pustaka *DLIB* dan menggunakan pustaka *OpenCV* sebagai pendukung program yang dibuat untuk mengolah *video*. *Hardware* yang digunakan yaitu *Raspberry Pi 4* model *B* sebagai alat untuk menjalankan program pelacakan objek. *Raspberry Pi* merupakan komputer kecil [8][9] tetapi mampu menjalankan tugas yang sama dengan komputer desktop.

2. Metode Penelitian

Metode yang digunakan dalam penelitian ini adalah metode eksperimen yang disusun dalam diagram alir penelitian seperti pada Gambar 1.



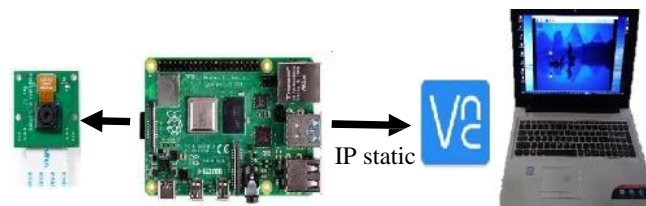
Gambar 1. Diagram alir penelitian

Tahap pertama dalam penelitian ini yaitu studi literatur mengenai pokok bahasan dalam penelitian yaitu sistem pelacakan objek. Tahap selanjutnya yaitu menentukan masalah serta menganalisa kebutuhan baik dari segi *hardware* maupun *software*. Berdasarkan analisa kebutuhan kemudian dilakukan perancangan *hardware* dan perancangan *software*, selanjutnya dilakukan penerapan atau implementasi dari hasil perancangan sistem. Sistem yang telah dibuat kemudian diuji untuk mengetahui apakah sistem sudah mencapai kriteria yang diinginkan. Tahap terakhir yaitu menganalisis untuk memperoleh kesimpulan akhir.

3. Perancangan Sistem

Perancangan sistem untuk pelacakan objek menggunakan algoritma *correlation tracker* pada pustaka *DLIB* terdiri dari perancangan *hardware* dan perancangan *software*.

3.1. Perancangan Hardware

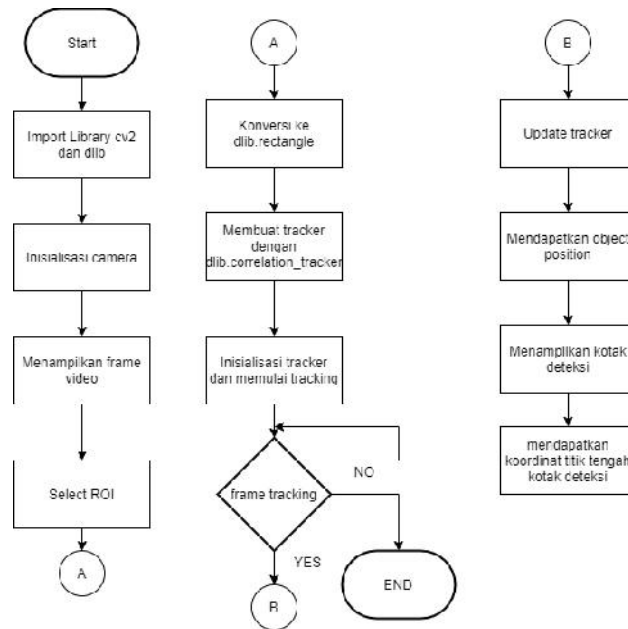


Gambar 2. Skema Hardware

Perancangan *hardware* untuk sistem pelacakan objek ini, dapat dilihat pada Gambar 2 yaitu menggunakan *Raspberry Pi 4* yang berisi program *python* untuk pelacakan objek dan *Pi Camera* sebagai *input* gambar *video realtime*. *Pi Camera* dapat digunakan pada *Raspberry PI*, tetapi harus mengaktifkan *interface camera* pada *Raspberry Configuration* terlebih dahulu. *Raspberry PI* dapat diakses dan dikontrol dengan menggunakan *VNC viewer* pada laptop, sehingga tidak perlu menggunakan *monitor*, *keyboard*, dan *mouse*.

3.2. Perancangan Software

Diagram alir program untuk sistem pelacakan objek ini dapat dilihat pada Gambar 3. Pustaka *OpenCV* digunakan sebagai pendukung program yang dibuat untuk mengolah *video*.



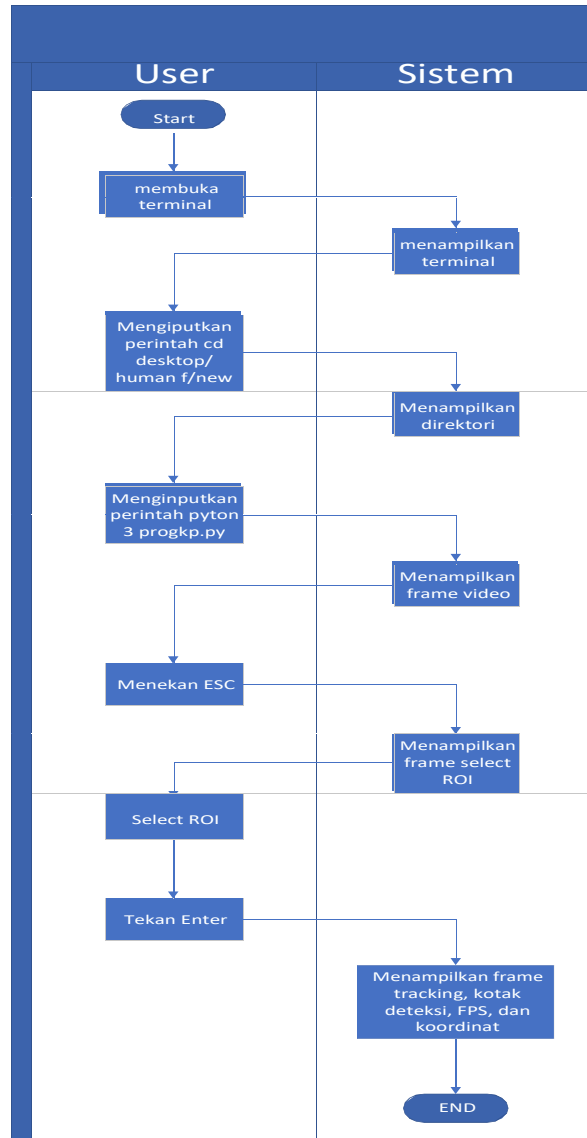
Gambar 3. Diagram Alir Program

Import cv2 pada pustaka *OpenCV* digunakan untuk menginisialisasikan kamera dan menampilkan *frame video streaming*. Secara normal, *frame* kamera berukuran $640 \times 480 \text{ px}$. Pemilihan objek yang dilacak dapat dilakukan dengan menggunakan fungsi *ROI* untuk menyeleksi daerah dari objek yang dideteksi. Hasil dari fungsi *ROI* didapatkan nilai yang dikonversi untuk menjadi nilai parameter pada fungsi *dlib.rectangle*. Pustaka *DLIB* memiliki fungsi *dlib.correlation_tracker* yang dapat digunakan sebagai algoritma pelacakan objek. Fungsi ini diinisialisasikan sebagai *tracker*. Inisialisasi *tracker* tersebut berisi parameter *frame* kamera dan titik *rectangle*. Setelah selesai menginisialisasikan *tracker* maka sistem menampilkan *frame* baru.

Frame baru yang muncul disebut *frame tracking*. *Frame tracking* ini memiliki fungsi untuk meng-*update tracker* sehingga didapatkan posisi objek pada setiap *frame*. Di dalam *frame tracking* ditampilkan kotak pelacakan yang berada di sekitar objek yang dilacak. Hasil yang didapatkan saat melacak objek yaitu koordinat yang didapatkan dari titik tengah kotak pelacakan. Selama *frame tracking* muncul, pelacakan objek terus dilakukan. Menghentikan pelacakan objek dapat dilakukan dengan cara keluar dari *frame tracking*.

3.3. Sistem Kerja Keseluruhan

Secara keseluruhan, sitem kerja pada pelacakan objek ini dapat dilihat pada Gambar 4.



Gambar 4. Diagram Alir Sistem Kerja

Langkah pertama yaitu pengguna mengakses terminal pada *raspberry pi* terlebih dahulu. Langkah berikutnya yaitu membuka *folder* yang terdapat program *python*. Program *python* berada di dalam *folder* 'HumanF' yang ada di *Desktop*. Pengguna mengetikkan perintah *python3* diikuti nama *file* dan ekstensi *file python* untuk menjalankan program. Sistem pertama kali bekerja menampilkan *frame video realtime*. Pengguna menekan tombol 'esc' untuk *capture video* yang nantinya ditampilkan pada *frame select ROI*. Pengguna membuat kotak seleksi untuk objek

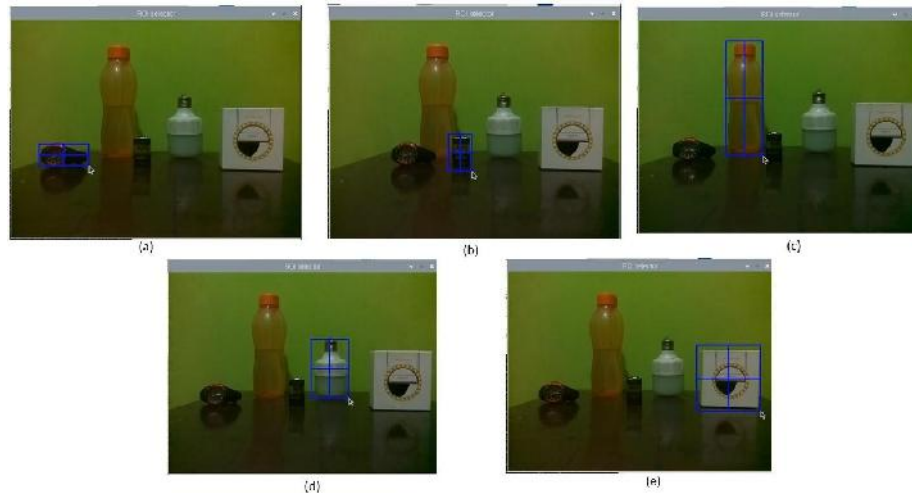
yang dipilih untuk dilacak menggunakan *mouse*. Setelah selesai membuat *ROI*, maka selanjutnya pengguna menekan tombol 'enter' untuk menampilkan *frame tracking*, sehingga proses pelacakan di mulai.

4. Hasil dan Pembahasan

Pengujian dilakukan untuk mengetahui apakah objek dapat dilacak dengan menggunakan algoritma *correlation tracker* dari pustaka *DLIB*. Terdapat beberapa proses yang dilakukan yaitu menampilkan hasil *streaming video* secara *realtime*, pemilihan objek, dan proses pelacakan objek.

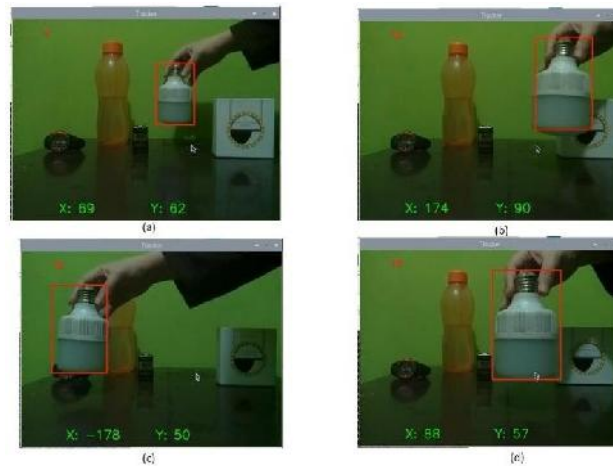


Gambar 5. *Frame video realtime*



Gambar 6. (a) Pemilihan ROI jam tangan, (b) Pemilihan ROI baterai 9v, (c) Pemilihan ROI botol, (d) Pemilihan ROI lampu LED, (e) Pemilihan ROI kotak ukuran 10 cm

Frame hasil dari *streaming video* dapat dilihat pada Gambar 5. *Frame* tersebut menampilkan berbagai objek yang berbeda. Pemilihan objek yang dilacak memanfaatkan fitur *ROI* yang di pilih oleh pengguna seperti pada Gambar 6. Setelah objek dipilih oleh pengguna, sistem menampilkan kotak pelacakan yang berada di sekitar objek dan melakukan *update tracker* di setiap *frame* sehingga kotak pelacakan selalu berada pada objek yang sudah dipilih.



Gambar 7. (a) Objek bergerak mundur, (b) Objek bergerak maju, (c) Objek bergerak ke kiri, (d) Objek bergerak ke kanan

Sebagaimana pada Gambar 7, di dalam *frame* terdapat koordinat x dan y yang didapatkan dari titik tengah kotak *tracking* terhadap *frame pixel video*. Titik tengah pada *frame* bernilai $0,0$. Sumbu x maksimal bernilai 320 dan minimal bernilai -320 , sedangkan pada sumbu y maksimal bernilai 240 dan minimal bernilai -240 .

Pengujian pelacakan objek juga dilakukan pada objek yang berbeda baik itu dari ukuran, bentuk dan warna. Pengujian ini dilakukan di dalam ruangan agar mendapatkan penerangan yang stabil dari lampu. Hasil pengujian dari berbagai objek yang berbeda dapat dilihat pada Tabel 1.

Tabel 1. Tracking objek

Objek	Hasil			
	Kiri	Kanan	Maju	Mundur
Jam Tangan				
Baterai 9V				
Kotak ukuran 10 cm				
Lampu LED				
Botol Minum				

Terdapat 5 objek yang diuji, yaitu jam tangan, baterai 9v, kotak ukuran 10cm, lampu LED, dan botol minum. Selanjutnya objek bergerak ke kiri, ke kanan, maju, dan mundur. Pergerakan objek saat ke kiri dan ke kanan, kotak pelacakan tetap berada di sekitar objek. Selanjutnya saat objek bergerak menjauhi kamera, maka ukuran objek semakin kecil sehingga ukuran kotak pelacakan pun berubah mengecil mengikuti ukuran objek. Apabila objek bergerak mendekati kamera, maka ukuran objek semakin besar sehingga ukuran kotak pelacakan pun berubah menjadi besar mengikuti ukuran objek.

Performa dari *Raspberry PI* dapat dilihat saat program *python* mulai dijalankan. Terdapat program untuk mengetahui *framerate* pada *video* dalam satu detik. Nilai *framerate* tersebut memiliki satuan *fps* (*frame per second*) dan ditampilkan pada *frame tracking* dengan *resolusi 480 x 640 pixel* sehingga dapat diamati secara langsung.

Tabel 2. Rata-rata framerate pada raspberry pi

Nilai	Nilai <i>framerate</i> (fps)
<i>Framerate</i> terkecil	7
<i>Framerate</i> terbesar	14
Rata-rata <i>framerate</i>	10,5

Berdasarkan Tabel 2, nilai *framerate* terkecil sebesar 7 fps, yang didapatkan saat *frame video* mulai diproses. Nilai *framerate* terbesar sebesar 14 fps. Nilai *framerate* rata-rata diambil saat menjalankan sistem pelacakan selama 1 menit. Rata-rata nilai *framerate* sebesar 10,5 fps.

5. Kesimpulan

Algoritma *correlation tracker* yang ada pada pustaka *DLIB* dapat digunakan untuk melakukan pelacakan objek secara *realtime*. Hasil dari pelacakan obyek yaitu kotak pelacakan selalu berada di sekitar objek, baik saat objek bergerak ke kiri, ke kanan, maju, dan mundur. Skala kotak pelacakan dapat berubah mengikuti ukuran objek. Saat program berjalan pada *Raspberry Pi*, rata-rata *framerate video realtime* sebesar 10,5 fps sehingga kualitas *video* cukup stabil tanpa adanya *delay*.

Referensi

- A. Fahriannur and M. Siswanto, "Sistem Tracking Objek Berbasis Algoritma Optical Flow Menggunakan Kamera Pan-Tilt," in *Seminar Nasional Hasil Penelitian*, 2017, pp. 175–178.
- Hidayatullah P. and Konik H., "CAMSHIFT Improvement on Multi-Hue and Multi-Object Tracking," in *International Conference on Electrical Engineering and Informatics*, 2011, pp. 1–6.
- M. I. Mahali and A. Harjoko, "Pelacakan Benda Bergerak Menggunakan Metode Mean-Shift dengan Perubahan Skala dan Orientasi," *Berkala MIPA*, vol. 24, no. 2, pp. 167–180, 2014.
- I. Supriana, "Multiple Object Tracking dan Estimasi Posisi untuk Menunjang Sistem Keamanan Rumah," *Jurnal Sekolah Teknik Elektro Informatika. Institut Teknologi Bandung*, 2015.
- G. Samudera, ; Jirio, ; Okky, and I. H. Kartowisatro, "OBJECT TRACKING," 2012.
- D. Rosdiana, "TRACKING OBJEK BERGERAK DENGAN ALGORITMA K-MEANS," *TEDC*, vol. 11, no. 1, pp. 31–36, 2017.
- R. R. Patil, O. S. Vaidya, G. M. Phade, and S. T. Gandhe, "Qualified Scrutiny for Real-Time Object Tracking Framework," *International Journal on Emerging Technologies*, vol. 11, no. 3, pp. 313–319, 2020, [Online]. Available: www.researchtrend.net
- F. A. Nugraha *et al.*, "Implementasi Sistem SCADA Pada Proses Koagulasi Water Treatment Plant Berbasis Raspberry Pi The Implementation SCADA System of Coagulation Process of Water Treatment Plant By Using Raspberry Pi," in *Seminar Nasional Teknik Elektro*, 2019, pp. 592–600.
- M. Ridlo Effendi, E. Ahmad, Z. Hamidi, and A. A. Suhardi, "Rancang Bangun Sistem Pendeteksi Manusia Pada Ruangan Menggunakan Raspberry Pi 3 Type B Dan Internet," in *Seminar Nasional Teknik Elektro*, 2017, pp. 271–279.