

OBJECT TRACKING BERBASIS KEYPOINT MENGGUNAKAN ALGORITMA ORB (ORIENTED FAST AND ROTATED BRIEF) PADA RASPBERRY PI

Mahmudin^{1*}, Nurul Mega Saraswati², Sorikhi³

¹²³Program Studi Informatika, Universitas Peradaban, Indonesia

Article Info

Article history:

Submitted March 2, 2023

Accepted April 10, 2023

Published Oktober 1, 2023

Keywords:

Object tracking,
Algoritma ORB,
Algoritma SIFT,
Algoritma SURF,
Keypoint,
Raspberry Pi.

*Object tracking,
Algoritma ORB,
Algoritma SIFT,
Algoritma SURF,
Keypoint,
Raspberry Pi.*

ABSTRACT

Perkembangan teknologi saat ini, penggunaan gambar *digital* sudah menjadi kebutuhan untuk berbagai keperluan. *Citra digital* berperan untuk memberikan informasi tentang apa yang ada pada suatu gambar *digital*. *Object tracking* merupakan salah satu penelitian yang menarik dalam *computer vision* dalam pencitraan *digital*. *Object tracking* digunakan melalui beberapa metode, salah satunya menggunakan *keypoint* dengan algoritma ORB sebagai dasarnya. Algoritma ORB memiliki kecepatan yang sangat tinggi, akan tetapi apakah bisa dijalankan di *Raspberry Pi*. bagaimana unjuk kerja algoritma tersebut bila dibandingkan dengan algoritma SIFT dan SURF. Setelah dilakukan pengujian, dihasilkan bahwa untuk unjuk kerja algoritma ORB lebih cepat dibandingkan dengan algoritma SURF dan SIFT. Kekurangan dari algoritma tersebut masih banyak kehilangan *keypoint*, masih terpengaruh dengan lingkungan sekitar objek, serta *Raspberry Pi* yang belum bisa secara maksimal untuk digunakan sebagai *object tracking*.

Today's technological development has made the use of digital images a necessity for various purposes. The purpose of the digital image is to provide information about what is in a digital image. Object tracking is one of the interesting researches in computer vision in digital imaging. Object tracking is used through several methods, one of which uses a key point with the ORB algorithm as a basis. The ORB algorithm has very high speed, but can it run on a Raspberry Pi? How does the algorithm compare to the SIFT and SURF algorithms? After testing, it was found that the performance of the ORB algorithm is faster than that of the SURF and SIFT algorithms. The disadvantages of this algorithm are that there are still many keypoint losses, these are still influenced by the environment around the object and the Raspberry Pi cannot be optimally used for object tracking..



Corresponding Author:

Mahmudin,

Program Studi Informatika, Universitas Peradaban,

Jl. Raya Pagojangan Km 3 Paguyangan, Brebes, Jawa Tengah 52276

Email: freelance.mahmudin@gmail.com

1. PENDAHULUAN

Perkembangan teknologi saat ini, penggunaan gambar digital sudah menjadi kebutuhan untuk berbagai keperluan. Citra digital berperan untuk memberikan informasi tentang apa yang ada pada suatu gambar digital. *Object tracking* merupakan salah satu penelitian yang menarik dalam *computer vision* dalam pencitraan digital. *Object tracking* sebagai langkah pertama dalam berbagai aplikasi di bidang robotika, keamanan, pengawasan, pencitraan medis, dan olahraga, dan lain-lainnya. *Object tracking* berfungsi untuk mendeteksi keberadaan objek yang ditangkap oleh kamera, menangkap target, menandai dan mengikuti objek yang bergerak. Berbagai metode digunakan dalam *object tracking*, salah satu metode yang digunakan adalah penggunaan *keypoint* dalam *object tracking* [1].

Keypoint merupakan titik-titik penting yang terdapat dalam suatu citra digital yang digunakan sebagai penanda dalam sebuah gambar. *Keypoint* tidak selalu berupa warna atau intensitas saja, titik-titik penting seperti sudut, lengkungan garis, garis yang berpotongan dan lain-lainnya juga dianggap sebagai *keypoint*. *Keypoint* yang terdapat dalam suatu objek akan menjadi sumber data untuk analisa gambar, sehingga dapat dijelaskan menggunakan bentuk numerik yang menjadi tanda pada sebuah gambar. Penggunaan dalam *object tracking*

hanya mengambil titik-titik penting saja dalam suatu objek, sehingga tidak memerlukan pengecekan keseluruhan objek yang membutuhkan waktu lama. Keypoint memiliki beberapa algoritma yang digunakan untuk melakukan *object tracking* seperti SIFT, SURF, dan ORB [2].

SIFT (*Scale Invariant Feature Transform*) digunakan untuk mengekstrak *keypoint* dan membangun deskriptor fitur yang sesuai. Pencarian *keypoint* potensial dengan cara membangun piramida dengan menurunkan kualitas contoh gambar dan mengambil perbedaan *gaussian*. Algoritma tersebut dapat digunakan dalam *object tracking* yang menghasilkan *frame rate* sebesar 1.8 fps sampai 2.8 fps. Kekurangan dari algoritma tersebut yaitu akurasi yang kurang dikarenakan *keypoint* yang dianggap penting lebih sedikit, kemudian *frame rate* yang dihasilkan rendah sehingga kurang dalam akurasinya. Algoritma yang serupa dengan SIFT yaitu SURF (*Speeded Up Robust Features*) merupakan deteksi *keypoint* dengan algoritma *descriptor* yang kuat dan dapat diulang seperti SIFT, sehingga dua kali lebih cepat dari SIFT. Algoritma tersebut menggunakan pendekatan bilangan bulat diskritisasi persegi panjang untuk fungsi yang kompleks, dan menggunakan maxima dari determinan matriks *Hessian*. Penerapan pada *object tracking* menghasilkan tingkat akurasi yang tinggi serta memperoleh *frame rates* sebesar 6 fps. Penggunaan algoritma SIFT dan SURF masih kurang efektif ketika digunakan, karena algoritma tersebut membutuhkan waktu untuk bisa mengenali sebuah objek. Algoritma yang memungkinkan cocok digunakan untuk *object tracking* yaitu algoritma ORB, algoritma tersebut lebih cepat dari pada SIFT dan SURF [3].

ORB (*Oriented FAST and Rotated BRIEF*) merupakan teknik deteksi dan *descriptor keypoint* yang semakin banyak digunakan bersamaan dengan SIFT dan SURF. ORB merupakan skala dan rotasi invarian, kuat terhadap *noise* dan transformasi *affine*, dan masih mampu memberikan *frame rate* 25 fps. Algoritma tersebut merupakan kombinasi dari *keypoint* FAST (*Features from Accelerated Segment Test*) dan algoritma *descriptor keypoint* BRIEF (*Binary Robust Independent Elementary Features*) yang dimodifikasi untuk menangani *keypoint* berorientasi. Algoritma ORB memiliki keakuratan yang tinggi dalam *object tracking* dan mampu memberikan *frame rate* 28 fps [3]. Penggunaan algoritma SIFT, SURF dan ORB menggunakan *keypoint* untuk *object tracking* computer vision dapat digunakan melalui beberapa *platform* seperti Windows, Linux dan Raspberry Pi.

Raspberry Pi adalah serangkaian SBC berbiaya rendah dan berukuran kartu kredit. Tujuan pengembangan Raspberry Pi adalah untuk mempromosikan pengajaran keterampilan komputer dasar dan pemrograman di sekolah-sekolah. Berbagai bahasa pemrograman seperti Python, C, C++, dan Java dapat digunakan dengan Raspberry Pi. Raspberry Pi juga tersedia dalam bentuk yang lebih fleksibel yang ditujukan untuk aplikasi industri dan embedded, serta modul komputasi yang memiliki banyak literasi [4].

Berdasarkan beberapa penelitian dan literasi yang dibaca mengenai *object tracking* maka algoritma yang akan digunakan penulis untuk melakukan penelitian menggunakan algoritma ORB karena algoritma tersebut merupakan perpaduan dari algoritma FAST dan BRIEF sehingga proses pengenalan objeknya menjadi lebih cepat dari pada algoritma SIFT dan SURF, serta menggunakan *keypoint* untuk proses *object tracking*nya. Berdasarkan alasan tersebut, penulis bermaksud melakukan penelitian yang berjudul *Object tracking Berbasis Keypoint Menggunakan Algoritma ORB Pada Raspberry Pi*. Penelitian yang akan dilakukan mencocokkan gambar training dengan gambar testing dalam pengujiannya, sehingga dapat diketahui kinerja dari algoritma ORB jika dibandingkan dengan algoritma SIFT dan SURF, serta mengetahui bagaimana Raspberry Pi bekerja.

2. METODE PENELITIAN

Image Processing

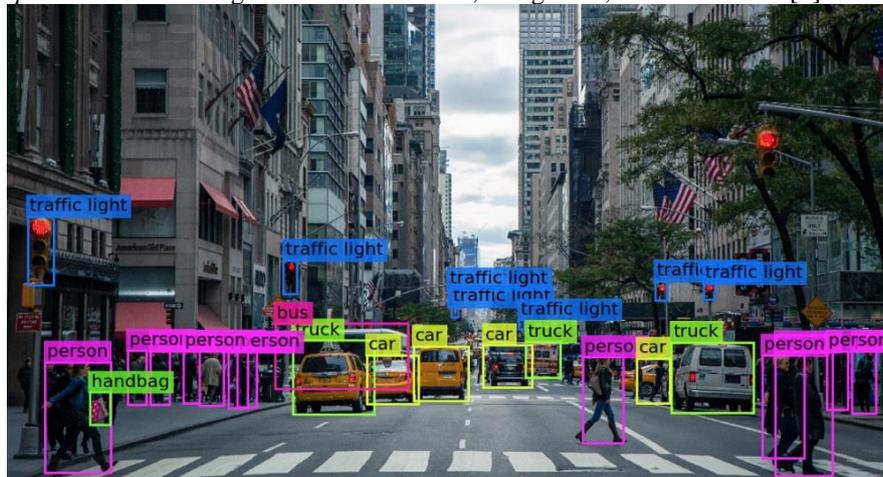
Gambar adalah representasi visual dari suatu objek, orang, atau pemandangan. Gambar digital adalah fungsi dua dimensi $f(x,y)$ yang merupakan proyeksi pemandangan tiga dimensi ke bidang proyeksi dua dimensi, di mana x,y mewakili posisi elemen gambar atau piksel dan mengandung intensitas nilai. Jika nilai x, y , dan intensitas bersifat diskrit, maka bayangan tersebut dikatakan sebagai citra digital. Secara matematis, citra digital adalah representasi matriks dari citra dua dimensi dengan menggunakan elemen sel titik dalam jumlah tak terhingga, yang biasa disebut piksel. Setiap piksel diwakili oleh nilai numerik, untuk gambar skala abu-abu, satu nilai yang mewakili intensitas piksel sudah cukup, untuk gambar berwarna, tiga nilai (mewakili jumlah merah (R), hijau (G), dan biru (B)). Ketika sebuah gambar hanya memiliki dua intensitas, gambar tersebut disebut sebagai gambar biner [5].

Sistem Penglihatan Manusia

Sistem penglihatan manusia terdiri dari dua bagian mata dan otak. Mata manusia bertindak sebagai reseptor untuk gambar, menangkap cahaya dan mengubahnya menjadi sinyal. Sinyal-sinyal ini kemudian dikirim ke otak untuk dianalisis lebih lanjut. Mata dan otak bekerja sama untuk menciptakan sebuah gambar. Manusia memiliki dua jenis sel peka cahaya di otak yaitu batang dan kerucut. Otak manusia memiliki sekitar 120 juta sel batang dan 6 juta sel kerucut. Sel batang memberikan visibilitas dalam cahaya yang sangat rendah dan sangat sensitif. Sel kerucut memberikan penglihatan siang hari dan bekerja paling baik dalam kondisi cahaya normal [6].

Computer vision

Computer vision adalah kecerdasan buatan yang memungkinkan komputer dan sistem memperoleh informasi yang berarti dari gambar *digital*, *video*, dan *input visual* lainnya dan mengambil tindakan atau membuat rekomendasi berdasarkan informasi tersebut. Kecerdasan buatan memungkinkan komputer untuk berpikir, sedangkan *computer vision* memungkinkan untuk melihat, mengamati, dan memahami [3].



Gambar 1. Contoh Penerapan *Computer vision*

Opencv

Opencv (*Open Source Computer vision Library*) adalah *open source computer vision* dan perpustakaan perangkat lunak pembelajaran mesin. *Opencv* dibangun untuk menyediakan infrastruktur umum untuk aplikasi visi komputer dan untuk mempercepat penggunaan persepsi mesin dalam produk komersial. Menjadi produk berlisensi BSD, *Opencv* memudahkan bisnis untuk memanfaatkan dan memodifikasi kode. Pustaka ini memiliki lebih dari 2500 algoritme yang dioptimalkan, yang mencakup serangkaian lengkap visi komputer klasik dan canggih serta algoritme pembelajaran mesin. Algoritme ini dapat digunakan untuk mendeteksi dan mengenali wajah, mengidentifikasi objek, mengklasifikasikan tindakan manusia dalam video, melacak pergerakan kamera, melacak objek bergerak, mengekstrak model objek 3D, menghasilkan awan titik 3D dari kamera stereo, menggabungkan gambar untuk menghasilkan resolusi tinggi, gambar seluruh pemandangan, temukan gambar serupa dari database gambar, hapus mata merah dari gambar yang diambil menggunakan flash, ikuti gerakan mata, kenali pemandangan dan buat penanda untuk melapisinya dengan augmented reality. *Opencv* memiliki lebih dari 47 ribu orang pengguna komunitas dan perkiraan jumlah unduhan melebihi 18 juta. Perpustakaan digunakan secara luas di perusahaan, kelompok penelitian dan oleh badan pemerintah [7].

Object Tracking

Object tracking merupakan arah penelitian yang penting dalam computer vision dengan berbagai aplikasi seperti video surveillance, interaksi manusia dan komputer, dan masih banyak lainnya. Fungsi dari object tracking yaitu untuk mengidentifikasi kembali karakteristik target[8]. Object tracking merupakan bagian dari banyak aplikasi computer vision yang memproses dari video kamera. Object tracking digunakan untuk membangun, menyebarkan dan menskalakan sistem pelacakan objek secara mendalam [9].

Keypoint

Keypoint adalah titik-titik penting yang terdapat pada suatu citra *digital* yang digunakan untuk mengkarakterisasi gambar tertentu. *Keypoint* tidak selalu mengacu pada warna atau intensitas, titik-titik yang dianggap sebagai *keypoint* bisa apa saja seperti sudut, lengkungan garis, garis yang berpotongan dan lain-lainnya. *Keypoint* yang terdapat dalam suatu objek akan menjadi sumber data untuk analisa gambar, sehingga *keypoint* akan direpresentasikan menggunakan bentuk numerik dan kemudian dijadikan tanda pada gambar. *Keypoint* digunakan untuk *object tracking* dikarenakan *keypoint* hanya mengambil titik-titik penting saja dalam suatu objek, sehingga tidak memerlukan pengecekan keseluruhan objek yang membutuhkan waktu lama. *Keypoint* memiliki beberapa algoritma yang digunakan untuk melakukan *object tracking* seperti SIFT, SURF, dan ORB [5].

SIFT Keypoint and Descriptor

SIFT (*Scale Invariant Feature Transform*) merupakan salah satu algoritma yang paling populer di semua *computer vision*. Penggunaan algoritma SIFT untuk mengekstrak *keypoint* dan membangun deskriptor fitur yang sesuai. Pengidentifikasi *keypoint* potensial, SIFT membangun piramida dengan menurunkan sampel gambar dan mengambil perbedaan *gaussian*. *Keypoint* bisa menjadi sesuatu yang sederhana seperti sudut. Algoritma SIFT memerlukan *keypoint* untuk memiliki skala dan orientasi yang terkait. Setiap *keypoint* yang menggunakan SIFT memiliki 128 elemen *descriptor* yang terkait. Elemen tersebut dibagi menjadi 16 blok, untuk setiap blok orientasi gradien dihistogramkan menjadi delapan biner [3].

SURF Keypoint and Descriptor

SURF (*Speeded Up Robust Features*) merupakan deteksi *keypoint* dan *algorithm descriptor* yang kuat dan dapat diulang seperti SIFT, algoritma tersebut dua kali lebih cepat dari SIFT. SURF menggunakan pendekatan bilangan bulat terdiskritisasi persegi panjang untuk fungsi bernilai nyata yang rumit, dan SURF menggunakan *maxima* dari determinan matriks *Hessian* [5].

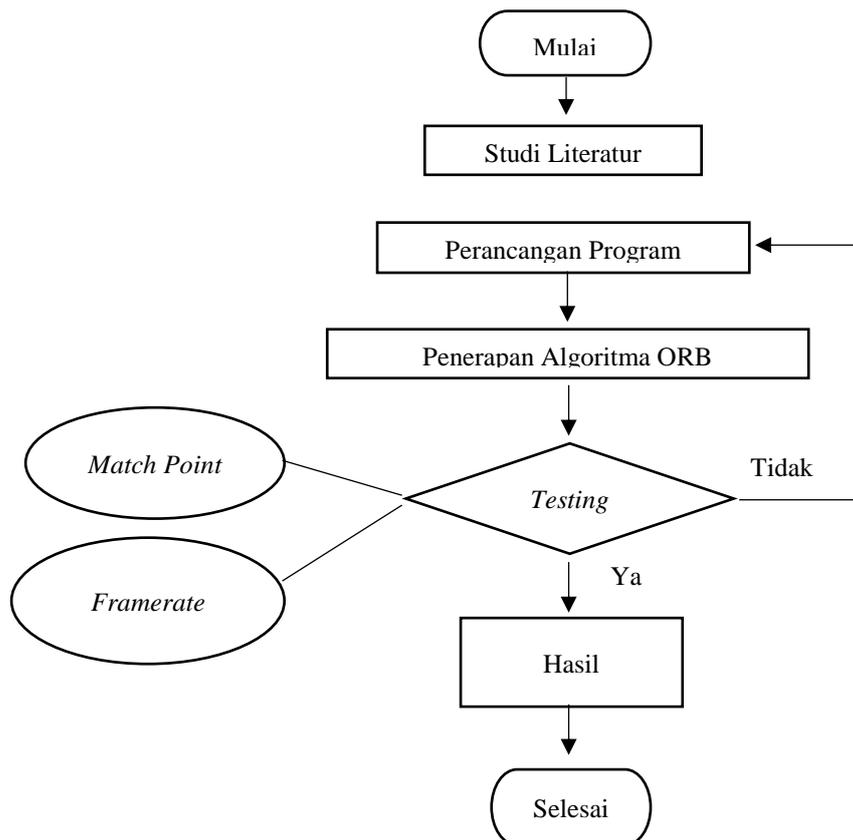
ORB (*Oriented FAST and Rotated BRIEF*)

ORB (*Oriented FAST and Rotated BRIEF*) adalah teknik deteksi dan deskripsi *Keypoint* yang semakin populer dengan SIFT dan SURF. ORB merupakan skala dan rotasi invarian, kuat terhadap *noise* dan transformasi *affine*, dan masih mampu memberikan frame rate 25 fps. Algoritma ORB sebenarnya adalah kombinasi dari *keypoint FAST (Features from Accelerated Segment Test)* deteksi dengan orientasi ditambahkan ke algoritma, dan algoritma deskriptor *keypoint BRIEF (Binary Robust Independent Elementary Features)* dimodifikasi untuk menangani titik kunci yang berorientasi. Detektor *keypoint FAST* menguji 16 piksel dalam lingkaran di sekitar piksel. Apabila piksel pusat lebih gelap atau lebih terang dari jumlah piksel ambang dari 16, itu ditentukan sebagai sudut. Pembuatan prosedur yang lebih cepat, pendekatan pembelajaran mesin digunakan untuk memutuskan urutan pemeriksaan 16 piksel yang efisien. Adaptasi FAST di ORB mendeteksi sudut pada beberapa skala dengan membuat piramida skala gambar, dan menambahkan orientasi ke sudut-sudut ini dengan menemukan pusat intensitas [3].

Raspberry Pi

Raspberry Pi adalah serangkaian SBC berbiaya rendah dan berukuran kartu kredit. Tujuan pengembangan *Raspberry Pi* adalah untuk mempromosikan pengajaran keterampilan komputer dasar dan pemrograman di sekolah-sekolah. Berbagai bahasa pemrograman seperti *Python*, *C*, *C++*, dan *Java* dapat digunakan dengan *Raspberry Pi*. *Raspberry Pi* juga tersedia dalam bentuk yang lebih fleksibel yang ditujukan untuk aplikasi industri dan *embedded*, serta modul komputasi yang memiliki banyak iterasi [4].

Penelitian yang dilakukan menggunakan metode eksperimen dalam mencapai tujuan. Tahapan penelitian yang digunakan untuk mengetahui unjuk kerja algoritma ORB untuk *object tracking* pada *Raspberry Pi*.



Gambar 2. Metode Penelitian

Studi Literatur

Studi literatur adalah proses pengumpulan teori berdasarkan sumber tertulis seperti buku, jurnal, tesis, skripsi maupun pemikiran dari pakar yang berkaitan dengan penelitian yang akan dilakukan. Pengumpulan teori-teori tersebut digunakan untuk memperkuat secara teori bagi penulis untuk melakukan penelitian.

Perancangan Program

Perancangan program merupakan pemaparan gambaran alur pembuatan program yang akan dibuat. Proses ini dilakukan untuk mempermudah dalam perancangan, menentukan kebutuhan sistem dan alur program dari penelitian yang akan dilakukan. Tahapan ini menggunakan algoritma ORB yang diterapkan kedalam *Raspberry Pi* untuk mengetahui bagaimana unjuk kerja algoritma tersebut untuk digunakan sebagai *object tracking*.

Penerapan Algoritma ORB

Proses penerapan merupakan implementasi dari algoritma ORB untuk *object tracking* menggunakan metode *keypoint* pada *Raspberry Pi*. Algoritma ORB berfungsi untuk membuat *keypoint* pada gambar referensi, kemudian akan dicocokkan dengan video yang ada pada input sehingga menghasilkan tingkat kecocokan pada suatu objek.

Pengujian Eksperimen

Setelah tahapan perancangan dan penerapan algoritma selesai, kemudian dilakukan tahapan pengujian untuk mengetahui bagaimana unjuk kerja algoritma ORB ketika digunakan untuk *object tracking* menggunakan *Raspberry Pi*. Pengujian pada penelitian ini memiliki dua parameter yaitu *frame rate* dan jumlah *match point* yang akan dibandingkan dengan algoritma *SIFT* dan *SURF*. Selain dari parameter tersebut pada tahap ini akan menguji kinerja *Raspberry Pi* untuk menjalankan program baik secara *GUI* maupun *console*.

Hasil

Tahap ini merupakan tahap terakhir setelah semua tahapan penelitian telah dilakukan mencapai hasil yang diharapkan, yang kemudian dapat ditarik kesimpulan dan saran dari semua pembahasan tahapan penelitian.

3. HASIL DAN PEMBAHASAN

Sesuai dengan hasil penelitian yang dihasilkan, berikut merupakan pembahasan dari penelitian yang sudah dilakukan.

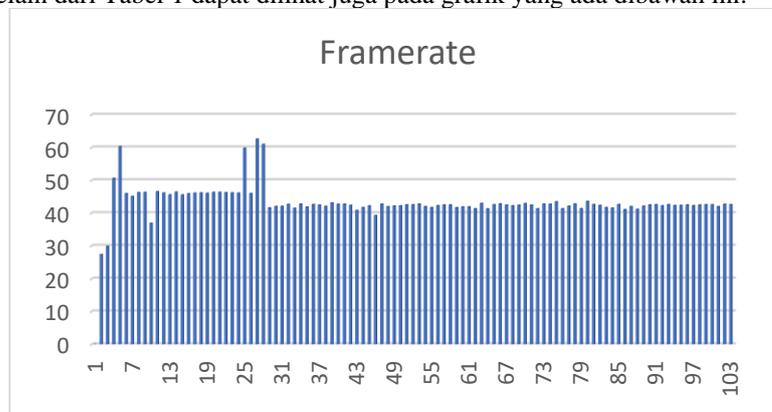
Tabel 1. Hasil rata-rata setiap algoritma

No	Nama Algoritma	Indikator				Hasil Kecepatan
		Framerate	Keypoint	Match Point (inlayer)	Match Point (outlayer)	
1	ORB	42	683	27	4	***
2	SURF	22	208	51	9	**
3	SIFT	7	302	104	12	*

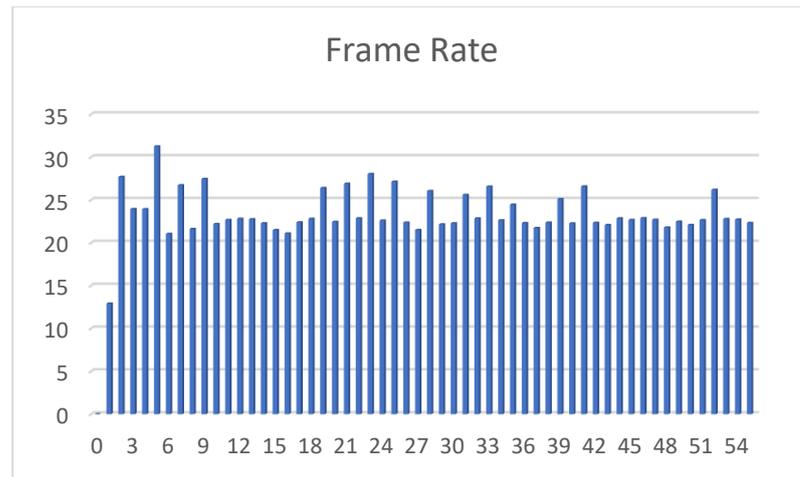
Keterangan:

- *** = Sangat Cepat
- ** = Cepat
- * = Lambat

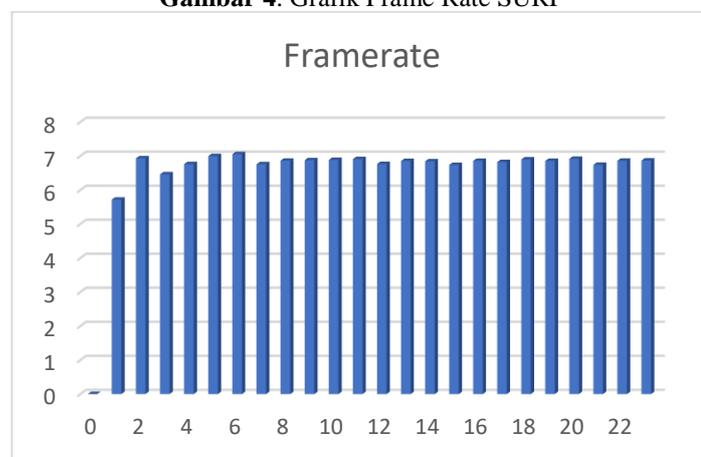
Berdasarkan Tabel 1 algoritma ORB memiliki kecepatan lebih bagus dari pada algoritma pembandingnya. Algoritma ORB menghasilkan framerate lebih besar sehingga dalam pengujiannya algoritma ORB menjadi lebih cepat dari pada dengan algoritma SURF dan SIFT. Dari hasil percobaan tersebut algoritma SIFT merupakan yang paling lambat dari ketiganya, tetapi juga paling akurat karena rata-rata match point yang dihasilkan tinggi. Hasil percobaan untuk algoritma SURF menghasilkan bahwa lagoritma SURF berada diantara algoritma SIFT dan ORB. Algoritma SURF mengutamakan akurasi dari pada dengan kecepatan, sehingga setiap percobaan algoritma SURF yang memiliki paling sedikit keypoint. Unjuk kerja algoritma ORB dari segi komputasi dan kecepatan lebih tinggi dari pada algoritma pembandingnya, akan tetapi masih terlalu banyak kehilangan keypoint dalam pengujiannya. Selain dari Tabel 1 dapat dilihat juga pada grafik yang ada dibawah ini:



Gambar 3. Grafik *Frame Rate* Algoritma ORB

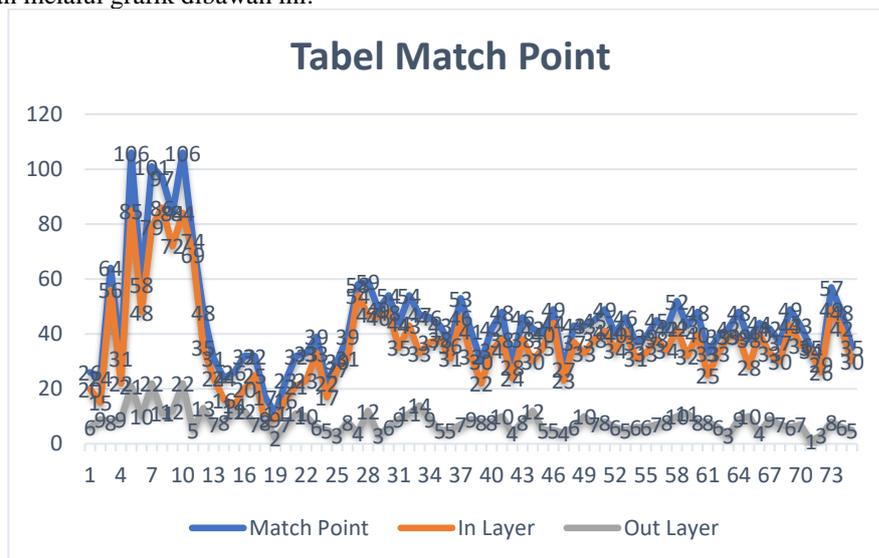


Gambar 4. Grafik Frame Rate SURF

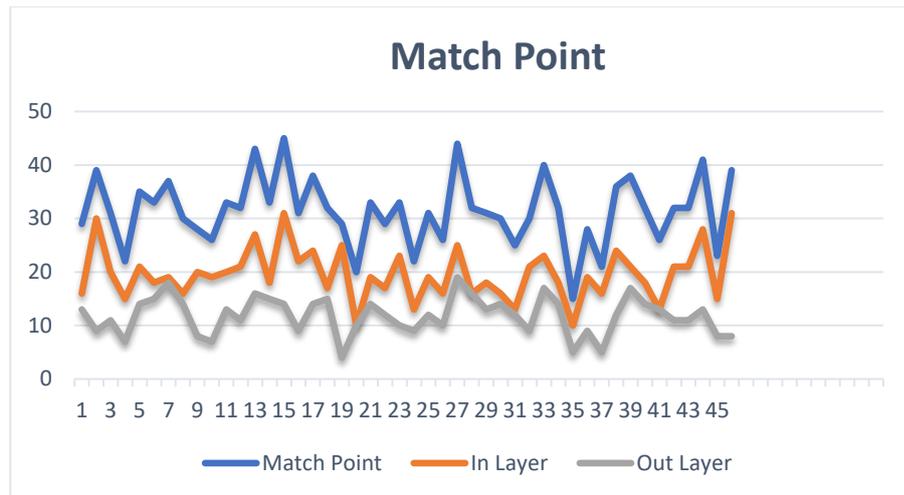


Gambar 5. Grafik Frame Rate SIFT

Berdasarkan grafik dari Gambar 3 – Gambar 5 didapatkan bahwa algoritma ORB memiliki *Frame Rate* lebih tinggi dibandingkan algoritma lainnya. Selain dari *framerate indicator* lainnya yaitu menggunakan *match point* yang akan disajikan melalui grafik dibawah ini:



Gambar 6. Match Point Algoritma ORB



Gambar 7. Match Point Algoritma SURF



Gambar 8. Match Point Algoritma SIFT

Berdasarkan match point yang diperoleh algoritma ORB memiliki match point yang lebih stabil dan lebih sedikit sehingga dalam proses pencocokannya algoritma ORB memiliki kecepatan yang lebih tinggi dari pada dengan algoritma SURF dan SIFT.

4. KESIMPULAN

Kesimpulan yang didapatkan pada penelitian mengenai *object tracking* berbasis *keypoint* menggunakan algoritma ORB pada *Raspberry Pi* yaitu kinerja algoritma ORB dari segi komputasi dan kecepatan lebih baik dari pada algoritma SURF dan SIFT. Berdasarkan *frame rate* dan *match point* yang terbentuk algoritma ORB memiliki kecepatan pencocokan lebih tinggi dari pada algoritma SURF dan SIFT. Algoritma ORB juga masih memiliki kelemahan yaitu masih besarnya jumlah *keypoint* yang terbentuk sehingga mempengaruhi dari tingkat akurasi, serta masih kehilangan beberapa *keypoint*. Selain dari kinerja algoritma ORB, kemampuan *Raspberry Pi* saat digunakan untuk *object tracking* dapat berjalan dengan baik, *Raspberry Pi* juga mendukung untuk pembuatan aplikasi berbasis GUI sehingga *Raspberry Pi* dapat direkomendasikan untuk pembelajaran *computer vision* maupun untuk pembuatan *project robotic* berskala kecil. Penelitian yang sudah dilakukan untuk nilai dari *match point* dan jumlah *keypoint* selalu berubah-ubah karena pemrosesan gambar nilainya tidak tetap, dan masih dipengaruhi dengan lingkungan dari gambar itu sendiri. Kondisi lingkungan objek tersebut meliputi gelap terangnya pencahayaan, ada tidaknya penghalang pada objek dan lain sebagainya, dengan kondisi tersebut yang menjadikan sebuah kelemahan dalam penelitian ini.

REFERENSI

- [1] "Implementasi objek detection dan tracking menggunakan deep learning untuk pengolahan citra digital | Asri | Konferensi Nasional Sistem Informasi (KNSI) 2018." <http://jurnal.atmaluhur.ac.id/index.php/knsi2018/article/view/439> (accessed May 28, 2022).
- [2] P. Joshi, *OpenCV with Python By Example : Build real-world computer vision applications and develop cool demos using OpenCV fo Python*. Birmingham B3 2PB, UK: Packt Publishing, 2015.
- [3] S. Brahmabhatt, *Practical OpenCV : HANDS ON PROJECTS FOR COMPUTER VISION ON THE WINDOWS, LINUX, AND RASPBERRY PI PLATFORMS*, 1st ed. Apress, 2013.

- [4] Ashwin. Pajankar, *Raspberry Pi Computer Vision Programming - Second Edition*. 2020.
- [5] Dr. A. Rosebrock, "Practical Python and OpenCV: An Introductory, Example Driven Guide to Image Processing and Computer Vision," pp. 1–23, 2016.
- [6] V. Tyagi, *Understanding Digital Image Processing*. 2018. doi: 10.1201/9781315123905.
- [7] J. Chen, Z. Xi, C. Wei, J. Lu, Y. Niu, and Z. Li, "Multiple Object Tracking Using Edge Multi-Channel Gradient Model with ORB Feature," *IEEE Access*, vol. 9, pp. 2294–2309, 2021, doi: 10.1109/ACCESS.2020.3046763.
- [8] Y. Zhang, Z. Chen, and B. Wei, "A Sport Athlete Object Tracking Based on Deep Sort and Yolo V4 in Case of Camera Movement," *2020 IEEE 6th International Conference on Computer and Communications, ICC3 2020*, pp. 1312–1316, Dec. 2020, doi: 10.1109/ICCC51575.2020.9345010.
- [9] A. Moghimi, T. Celik, A. Mohammadzadeh, and H. Kusetogullari, "Comparison of Keypoint Detectors and Descriptors for Relative Radiometric Normalization of Bitemporal Remote Sensing Images," *IEEE J Sel Top Appl Earth Obs Remote Sens*, vol. 14, pp. 4063–4073, 2021, doi: 10.1109/JSTARS.2021.3069919.