Laman web jurnal: https://ejournal.unama.ac.id/index.php/processor

Processor: Jurnal Ilmiah Sistem Informasi, Teknologi Informasi dan Sistem Komputer

P-ISSN: 1907-6738 | E-ISSN: 2528-0082

Implementasi Algoritma *SIFT (Scale-Invariant Feature Transform)* dan Algoritma *Kalman Filter* dalam Mendeteksi Objek Bola

M. Irwan Bustami[1], Chindra Saputra[2], Desi Kisbianty[3], Arjuna Panji Prakarsa[4]

[1,2,3,4]Fakultas Ilmu Komputer, Program Studi Sistem Komputer,Universitas Dinamika Bangsa, Jl. Jendral Sudirman, Thehok, Jambi 36138, Indonesia

ABSTRACT

The Indonesian Football Robot Contest (KRSBI) is a division of the Indonesian Robot Contest (KRI) which is held annually by RISTEKDIKTI and KEMENDIKBUD. In a soccer robot contest, the robot is required to detect the ball and then lead it to the opponent's goal so that a goal can be created. a good object detection system must be fast, light, and of course it must have good accuracy. Currently, the object detection system that has been applied to robots is in the form of color filtering which is considered quite good in detecting an object. However, if you only rely on this method, it is still lacking when viewed from the point of view of object tracking. The Kalman Filter algorithm serves as an estimator that can be used to predict the direction of movement of an object based on the object's status from the previous frame. This allows the robot to move 1 (one) frame faster than the object to be tracked. The SIFT algorithm can compare two images through the features of the image and produce whether the images are similar or not. This algorithm is useful for ascertaining whether the object detected by the robot is a ball or not. This research combines the SIFT and Kalmal Fiter algorithms to produce a better detection system than before. It is hoped that the results of this research can be used by robots in participating in the Indonesian soccer robot contest.

Keywords: OpenCV, Python, Object Detection, Kalman Filter, Computer Vision.

ABSTRAK

Kontes Robot Sepak Bola Indonesia (KRSBI) merupakan salah satu divisi dari Kontes Robot Indonesia (KRI) yang diadakan oleh RISTEKDIKTI dan KEMENDIKBUD setiap tahunnya. Dalam kontes robot sepak bola, robot di tuntut untuk bisa mendeteksi bola dan kemudian menggiringnya ke gawang lawan agar dapat tercipta sebuah gol. istem pendeteksian objek yang baik haruslah cepat, ringan, dan tentu saja harus memiliki akurasi yang baik. Saat ini, sistem pendeteksian objek yang telah diterapkan pada robot yaitu berupa color filtering (penyaringan warna) yang dinilai cukup baik dalam mendeteksi sebuah objek. Namun jika hanya mengandalkan metode ini dirasa masih kurang jika di lihat dari segi tracking objek. Algoritma Kalman Filter berfungsi sebagai estimator yang dapat digunakan untuk memprediksi arah pergerakan dari suatu objek berdasarkan status objek dari frame sebelumnya. Hal ini membuat robot dapat bergerak lebih cepat 1 (satu) frame dari pada objek yang akan di tracking. Algoritma SIFT dapat membandingkan dua citra melalui feature yang dimiliki citra tersebut dan menghasilkan apakah citra tersebut memiliki kemiripan atau tidak, algoritma ini berguna untuk memastikan apakah objek yang dideteksi oleh robot itu bola atau bukan. Penelitian ini menggabungkan algoritma SIFTdan Kalmal Fiter untuk menghasilkan sistem pendeteksian yang lebih baik dari sebelumnya. Diharapkan agar hasil dari penelitian ini dapat dipergunakan oleh robot dalam mengikuti kontes robot sepak bola Indonesia.

Kata Kunci: OpenCV, Python, Object Detection, Kalman Filter, Computer Vision.

1. PENDAHULUAN

Teknologi computer vision merupakan penerapan disiplin ilmu yang saat ini banyak digunakan sebagai objek penelitian dalam duniarobotika[1][2]. Dalam pertandingan robot sepak bola beroda, robot dituntut agar dapat mendeteksi bola dan gawang serta dapat mengetahui komponen lain yang ada dalam lapangan pertandingan sehingga robot dapat memainkan sepak bola seperti layaknya manusia bermain sepak bola. Untuk itu robot harus memiliki kemampuan dasar dalam bermain sepak bola seperti mengenali bola, mengenali gawang, mengenali musuh, menggiring bola dan menendang bola. Sampai saat ini telah banyak dikembangkan metode pendeteksian objek. Bernagai metode deteksi objek bola banyak diimplementasikan oleh tim robot dari berbagai perguruan tinggi yang mengikuti kontes robot Indonesia (KRI). Sistem pendeteksian objek yang baik haruslah cepat, ringan, dan tentu saja harus memiliki akurasi yang baik. Penelitian sebelumnya dilakukan oleh [3]. tentang sistem pendeteksian gambar termanipulasi menggunakan algoritma SIFT (*Scale-Invariant Feature Transform*) yang dimana metode ini dapat mendeteksi gambar yang disalin dan masih dapat mencocokan walaupun ketika ada perubahan. Kalman Filter atau dikenal juga dengan Linear Quadratic Estimation (LQE), adalah algoritma yang menggunakan serangkaian pengukuran yang diamati dari waktu ke waktu dan menghasilkan estimasi variabel tidak diketahui yang cenderung lebih banyak dan dapat meningkatkan akurasi dibandingkan jika hanya mengandalkan satu pengukuran saja[4]. Kalman Filter dapat juga digunakan untuk menstabilkan posis dan sikap pesawat terbang tanpa awak[5]. Berdasarkan latar belakang tersebut, peneliti menggabungkan kedua algoritma dan mengimplementasikannya kedalam sistem pendeteksian.

2. TINJAUAN PUSTAKA

2.1. Implementasi

Implementasi merupakan suatu proses mendapatkan suatu hasil yang sesuai dengan tujuan atau sasaran kebijakan itu sendiri. Dimana pelaksana kebijakan melakukan suatu aktivitas atau kegiatan[6].

2.2. Computer Vision

Computer Vision (visi komputer) dapat didefinisikan dengan pengertian pengolahan citra yang dikaitkan dengan akuisisi citra, pemrosesan, klasifikasi, penganan, dan pencakupan keseluruhan, pengambilan keputusan yang diikuti pengidentifikasian citra. Inti dari teknologi *Computer Vision* adalah untuk menduplikasi kemampuan penglihatan manusia ke dalam benda elektronik sehingga benda elektronik dapat memahami dan mengerti arti dari gambar yang dimasukkan[7].

2.3. Pengolahan Citra

Pengolahan citra digital adalah manipulasi dan interprestasi *digital* dari citra dengan bantuan komputer. *Input* dari pengolahan citra adalah citra, sedangkan *output*-nya adalah citra hasil pengolahan Istilah pengolahan citra *digital* secara umum didefinisikan sebagai pemrosesan citra dua dimensi dengan komputer. Dalam definisi yang lebih luas, pengolahan citra *digital* juga mencakup semua data dua dimensi. Citra *digital* adalah barisan bilangan nyata maupun kompleks yang diwakili oleh *bit-bit* tertentu[7].

2.4. Color Filtering

Proses ini dilakukan untuk melakukan pengambilan piksel pada suatu *image* yang memiliki itensitas warna tertentu. Setelah didapatkan objek dengan warna tertentu saja, langkah selanjutnya ialah proses *thresholding* yang bertujuan untuk mengubah citra berderajat keabuan menjadi citra *biner* atau hitam putih sehingga dapat diketahui daerah mana yang termasuk objek dan *background* dari citra secara jelas. Agar memudahkan proses pelacakan bentuk pada sebuah benda pada citra *digital* digunakan proses *edge detector*. Proses ini bertujuan untuk melakukan pengubahan format gambar hanya menampilkan bagian tepi saja[8].

2.5. OpenCV

Open CV adalah sebuah*Open Source*dan perpustakaan visi komputer yang sangat populer untuk pemrosesan dan menganalisis gambar[9].OpenCV (*Open Source Computer Vision Library*) merupakan salah satu software pustaka yang ditujukan untuk pengolahan citra dinamis secara real-time, yang dibuat oleh Intel, dan sekarang didukung oleh Willow Garage dan Itseez. OpenCV dirilis dibawah lisensi permisif BSD yang lebih bebas dari pada GPL, dan memberikan kebebasan sepenuhnya untuk dimanfaatkan secara komersil tanpa perlu mengungkapkan kode sumbernya. Ia juga memiliki antar muka yang mendukung bahasa pemrograman C++, C, Python dan Java, termasuk untuk sistem operasi Windows, Linux, Mac OS, iOS dan Android. OpenCV didisain untuk efisiensi dalam komputasi dan difokuskan pada aplikasi real-time[10].

2.6. Python

Phyton adalah bahasa pemograman yang fleksibel dan sederhana yang didefinisikan dalam dokumen-dokumenya. Bahasa pemograman yang dinamis sering digunakan dalam mengembangan aplikasi pada berbagai domain[11].Python dapat digunakan untuk berbagai keperluan pengembangan perangkat lunak dan dapat berjalan di berbagai platform sistem operasi. Saat ini kode python dapat dijalankan di berbagai platform sistem operasi, beberapa diantaranya adalah[12]:

- Linux/Unix
- Windows
- Mac OS X
- Java Virtual Machine
- Amiga
- Palm
- Symbian (untuk produk-produk Nokia)

2.7. Algoritma SIFT

Scale Invariant Feature Transform (SIFT) ialah sebuah algoritma dalam *computer vision* yang digunakan untuk mendeteksi dan mendeskripsikan fitur lokal dalam gambar. Algoritma *SIFT* dipublikasikan oleh *David G. Lowe* pada tahun 1999. Prinsip kerja dari *SIFT* yaitu suatu citra akan di ubah menjadi *vector* fitur local yang kemudian digunakan sebagai pendekatan dalam mendeteksi maupun mengenali objek yang dimaksud melalui titik titik point atau *keypoint*[13]. Algoritma *SIFT* terdiri atas beberapa bagian diantaranya adalaha sebagai berikut :

2.7.1. Gaussian dan Difference of Gaussian Scale Space

Dalam algoritma *SIFT*, fungsi *Gaussian* digunakan untuk membuat *scale space* sehingga disebut *Gausian Scale Space*. Untuk mendapatkan *Gaussian scale space* dapat dilihat seperti pada Persamaan (1)

(1)

$$G(x,y,\sigma) = \frac{1}{2\pi\sigma} e^{-(x^2-y^2)/2\sigma^2}$$

Keterangan:
- e = 2,72 (nilai ketetapan).
- σ = parameter (skala) jumlah kabur.
- π = 3,14

Setelah mendapatkan *Gaussian scale space*, kemudian menentukan *L(x , y , σ)* yang merupakan konvolusi citra asli *I (x , y)* dengan *Gaussian filterG(x , y , σ)* pada skala *σ* seperti pada Persamaan (2).

(2)

$$L(x,y,\sigma) = G(x,y,\sigma) \times I(x,y)$$

Sehingga *Difference of Gaussian (DoG)*pada skala kdapat dicari dengan menggunakan persamaan (3).

(3)

$$D(x,y,\sigma) = (G(x,y,k\sigma) - G(x,y,\sigma)) \times I(x,y)$$
$$D(x,y,\sigma) = L(x,y,k\sigma) - L(x,y,\sigma)$$

Dimana :
- *σ* adalah *parameter* (skala) jumlah kabur. Semakin tinggi nilainya, semakin besar *blur*-nya.
- *x , y* adalah koordinat lokasi titik dalam citra.
- *L(x , y , k σ)* adalah konvolusi dari citra asli *I (x , y)* dengan *Gaussian filter G(x , y , σ)* pada skala *k σ* dengan nilai *k* = √ *2*.

2.7.2. Deteksi Ekstremum

Deteksi *ekstremum* (nilai maksimum dan minimum) dilakukan dengan cara membandingkan nilai setiap piksel pada*DoG Scale space* dengan delapan piksel yang berada di sekelilingnya dan 9 piksel yang bersesuaian pada citra *DoG* sebelum dan setelahnya. Sebuah titik akan terpilih sebagai kandidat *keypoint*apabila nilai titik tersebut lebih kecil atau lebih besar dari nilai semua tetangganya yang sejumlah 26[13].

Setelah didapatkan kandidat-kandidat *keypoint* pada proses sebelumnya, tahapan selanjutnya adalah mengeliminasi kandidat *keypoint* yang tidak stabil. Biasanya *keypoint*yang tidak stabil itu adalah *keypoint* yang nilainya berbeda dari nilai sekitarnya pada matriks yang mungkin disebabkan oleh adanya *noise*. Selain itu, kandidat yang tidak stabil tersebut dapat juga berupa kandidat *keypoint* yang memiliki kontras yang rendah dan posisinya berada pada tepian gambar. Proses penghilangan *keypoint* yang memiliki kontras rendah dilakukan dengan melihat area maksimal dan area minimal, jika nilai kurang dari thresholdmaka titik tersebut tidak menjadi *keypoint*[13].

2.7.3. Penentuan Orientasi Keypoint

Orientasi *keypoint* ditentuka dengan meggunakan citra *Gaussian smooth L* yang mempunyai nilai mendekati skala *keypoint*. Setiap citra sampel *L(x,y) magnitude m(x,y)* dan orientasi *θ(x,y)* dihitung menggunakan perbedaan piksel seperti pada Persamaan (4)[13].

(4)

$$m(x,y) = \sqrt{(L(x+1,y) - L(x-1,y)^2 + (L(x,y+1) - L(x,y-1))^2)}$$

$$\theta(x,y) = \arctan\left(\frac{L(x,y+1) - L(x,y+1)}{L(x+1,y) - L(x-1,y)}\right)$$

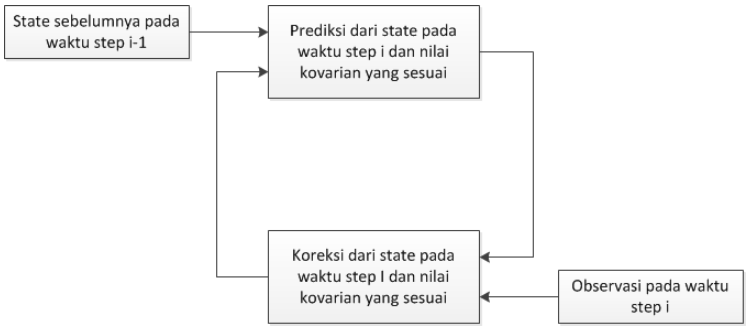
Keterangan:
- *m(x,y)* = magnitude terhadap koordinat lokasi x dan y
- *θ(x,y)* = orientasi terhadap koordinat lokasi x dan y

2.7.4. Penentuan Deskriptor Lokal

Langkah terakhir yaitu menghitung vektor *deskriptor*. Dimana vektor *deskriptor* dihitung menentukan masing-masing *keypoint*, langkah ini dilakukan terhadap gambar yang paling dekat dengan skala *keypoint*. Pertama membuat orientasi dengan 4x4 piksel dengan 8 *bit* pada setiap *keypoint*. Hasil histrogramdari penetapan orientasi dihitung *magnitude* dan nilai orientasi untuk sampel pada wilayah 16x16 disekitar *keypoint*. Fungsi *Gaussian* digunakan untuk menghitung *magnitude* dengan H^T sama dengan satu setengah lebar *deskriptor*. *Deskriptor*menjadi vektor dari semua nilai histogram. Karena 4x4 =16 histogram dengan masing-masing mempunyai 8 *bit*, sehingga vektor mempunya 128 elemen[13].

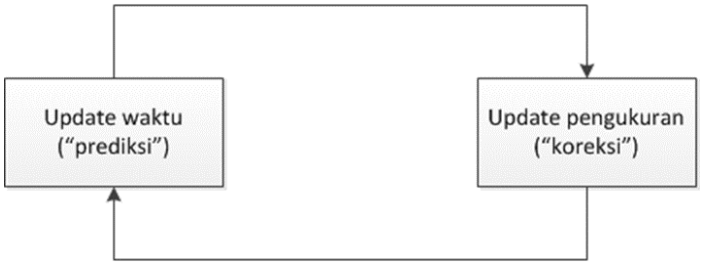
2.8. Algoritma Kalman Filter

Dwi Elmik Winata Putra Menjelaskan bahwa "*kalman filter* merupakan *estimator* rekursif. Untuk menggunakan *kalman filter*, pergerakan objek antarframe diasumsikan konstan. Status dari objek yang dilacak dapat dinyatakan dengan atribut-atribut seperti posisi, kecepatan, atau ukuran objek tersebut"[14]. Metode kalman filter menggunakan informasi dari objek yang terdeteksi di suatu frame dan status objek dari frame sebelumnya untuk mendapatkan status yang baru dari objek tersebut.
Komponen dasar dari kalman filter adalah vektor state, model dinamis dan model observasi, yang ditunjukkan pada gambar 1.



Gambar 1.Komponen Dasar Kalman Filter[14]

Vektor state menggambarkan state dari sistem dinamis dan menunjukkan derajat kebebasan. Variabel di vektor state tidak dapat diukur secara langsung, tetapi dapat disimpulkan dari nilai-nilai yang terukur. Elemen dari vektor state dapat diposisikan, kecepatan, orientasi, sudut, dan lain-lain. Vektor state memiliki dua nilai pada saat yang sama, salah satunya adalah nilai yang diprediksi sebelum di update dan nilai posterior setelah di update.Proses Kalman filter merupakan proses yang menggunakan bentuk kontrol umpan balik. Filter memperkirakan state proses pada beberapa waktu dan kemudian memperoleh umpan balik dalam bentuk pengukuran. Seperti terlihat pada gambar 2.



Gambar 2. Proses Kalman Filter[14]

Dengan demikian, persamaan untuk filter Kalman terbagi dalam dua kelompok:

1. Persamaan update waktu
   Persamaan ini berfungsi untuk memproyeksikan kedepan (dalam waktu) state saat ini dan kovarian error berfungsi memperkirakan waktu berikutnya. Persamaan update waktu bisa juga dianggap sebagai persamaan prediktor. Persamaan khusus untuk update waktu ditunjukkan pada persamaan (5) dan (6).

(5)

$$P_{(k)}^{-1} = AP_{(k-1)}A^T + Q$$

(6)

$$X_{(k)} = Ax_{(k-1)} + Bu_{(k-1)}$$

Keterangan:
- $X_{(k)}$ = Nilai diskrit state
- $Ax_{(k-1)}$ = Matriks transisi state
- $Bu_{(k-1)}$ = Kovarian error
- $P_{(k)}$ = Persamaan predictor
- $AP_{(k-1)}$ = Matriks A awal
- A^T = Matriks A transpose
- Q = Nilai kovarian

2. Persamaan update pengukuran.
   Persamaan ini merupakan proses umpan balik untuk menggabungkan pengukuran baru kedalam perkiraan yang mengandung perkiraan posterior yang telah diperbaiki. Persamaan update pengukuran dapat dianggap sebagai persamaan corrector. Persamaan untuk update pengukuran dapat dilihat pada persamaan (7) dan (8).

(7)

$$P_{(k)} = \left(I - K_{(k)}H\right)P_{(k)}^{-1}$$

(8)

$$K_{(k)} = P_{(k)}^{-1}H^T + P_{(k)}H + R^{-1}$$

Keterangan:

- $P_{(k)}$ = Persamaan predictor
- $K_{(k)}$ = Kalman gain
- I = Matriks I
- H = Matriks H
- $P^{-1}{}_{(k)}$= Persamaan predictor inverse
- R^{-1} = Matriks R inverse
- H^T = Matriks H transpose

Persamaan update waktu juga dapat dianggap sebagai persamaan prediktor, sementara persamaan update pengukuran dapat dianggap sebagai persamaan korektor.
Kalman filter digunakan untuk memperkirakan state dari suatu sistem linear dimana state diasumsikan didistribusikan oleh Gaussian. Kalman filter adalah filter prediksi rekursif yang didasarkan pada penggunaan teknik state-space dan algortima rekursif. State diperkirakan dari suatu sistem yang dinamik. Sistem yang dinamik ini dapat didistribusikan oleh beberapa noise, sebagian besar diasumsikan sebagai noise putih. Untuk meningkatkan estimasi stateKalman filter menggunakan pengukuran yang berkaitan dengan state, tetapi terganggu juga. Filtering Kalman terdiri dari 2 langkah, yaitu prediksi dan koreksi.

3. METODOLOGI PENELITIAN

3.1. Analisis Kebutuhan Sistem

Prinsip kerja sistem pendeteksian objek ini pertama-tama, kamera berada pada posisi default, kemudian kamera akan melakukan pemrosesan gambar untuk memfilter objek dengan benda-benda disekitar, setelah dilakukan pemrosesan. Jika pada frame ditemukan objek berwarna orange, sistem akan melakukan scanning dan membandingkan objek dengan sampel yang ada di database menggunakan algoritma SIFT. Jika objek tersebut berhasil di identifikasi sebagai bola maka sistem akan melakukan tracking dengan menggunakan algoritma kalman filter untuk memprediksi arah pergerakan objek. Adapun kebutuhan sistem pendeteksian objek bola ini yaitu:

3.1.1. Webcam

Pada sistem pendeteksian objek ini dibutuhkan kamera dengan resolusi tinggi agar gambar yang di hasilkan maksimal dan dapat dengan baik di proses oleh sistem.Webcam (singkatan dari webcamera) adalah sebutan bagi kamerareal-time (bermakna keadaan pada saat ini juga) yang gambarnya bisa diakses atau dilihat melalui World Wide Web, program instant messaging, atau aplikasi video call [15]. Berikut bentuk fisik kamera webcam Logitech C922. Terlihat pada gambar 3 :

Gambar 3. Webcam Logitech C922
(Sumber: logitech.com)

3.1.2. Servo Dynamixel AX-12A

Dalam penelitian sistem pendeteksian objek ini tidak langsung menggunakan robot, namun hanya sebatas kameranya saja oleh karena itu servo dynamixel AX-12A digunakan untuk melihat reaksi dari sistem saat melakukan tracking. Bentuk fisiknya dapat dilihat pada gambar 4 :

Gambar 4. Servo Dynamixel AX-12A
(Sumber: robotis.com)

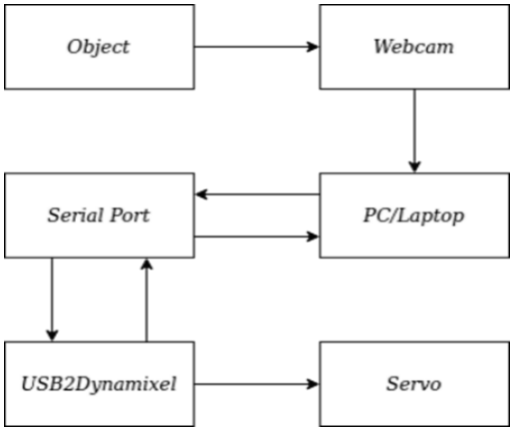
3.1.3. USB2Dynamixel

USB2Dynamixel dalam penelitian ini digunakan untuk mengontrol servo AX-12A agar kamera dapat mengikuti objek. Bentuk fisiknya dapat dilihat pada gambar 5 :

Gambar 5. USB2Dynamixel
(Sumber: robotis.com)

3.1.4. Blok Diagram

Blok diagram merupakan sistem yang terintegrasi, karena sistem tersebut tidak dapat berkerja apabila salah satu perangkat tidak ada. Isi dari sistem ini adalah Komputer/Laptop sebagai pusat pengendali utama dengan perangkat lunak (software) sebagai instruksi yang dilakukan oleh rangkaian (input), Block diagram dapat dilihat pada gambar 6:

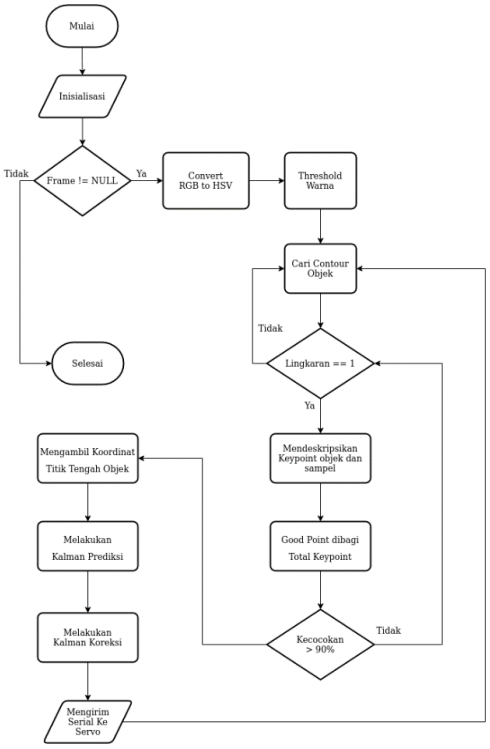


Gambar 6. Blok Diagram

Dari gambar 3.4 dapat dilihat sistem ini mempunyai beberapa bagian diantaranya:

1. Objek
   Objek yang ditangkap oleh kamera yaitu Bola futsal standar FIFA dengan warna orange.
2. Webcam
   Webcam yang berfungsi menangkap citra digital dan dikirimkan ke PC / Laptop untuk di olah menggunakan program yang telah dibuat.
3. PC/Laptop
   PC/Laptop berfungsi untuk mengolah citra digital yang telah ditangkap oleh kamera dan mengirimkan perintah berupa komuniksi serial.
4. Serial Port
   Setelah berhasil mendeteksi objek, perintah dikirimkan melalui serial port untuk menggerakkan servo.
5. Servo Dynamixel AX-12A
   Servo Dynamixel AX-12A berfungsi untuk menggerakkan webcam mengikuti arah pergerakan dari objek.

3.1.5. Flowchart Program



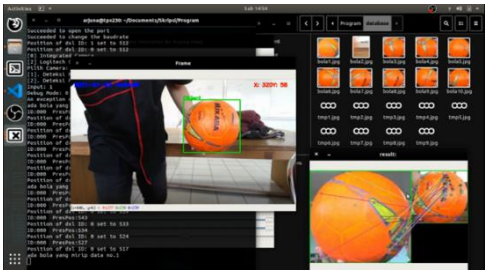
Gambar 7. Flowchart Program

1. Menjalankan program dengan memilih mode kamera dan mode pendeteksian yang dilakukan oleh user.
2. Melakukan inisialiasi berdasarkan user input.
3. Jika frame tidak kosong, maka program melakukan konversi RGB ke HSV pada frame. Jika frame kosong maka program berhenti / selesai.
4. Setelah melakukan konversi warna, selanjutnya sistem melakukan thresholding.
5. Lalu mencari area contour yang terbesar untuk dinyatakan sebagai objek.
6. Jika area contour sesuai dengan nilai yang ditentukan maka objek adalah bola, namun jika tidak maka objek bukan bola.
7. Jika bola terdeteksi, selanjutnya menggunakan algoritma SIFT (Scale-Invariant Feature Transform) untuk mendeskripsikan keypoint pada objek dan sample.
8. Kemudian keypoint yang cocok dibagi dengan total keypoint yang telah terdeskripsi.
9. Jika tingkat kecocokan lebih dari 90% sistem akan menandai titik tengah objek kemudian membuat kalman prediction berdasarkan informasi objek dari frame sebelumnya dan kemudian melakukan kalman correction untuk meminimalisir nilai error.
10. Selanjutnya sistem mengirim nilai koordinat objek kepada servo melalui serial port.
11. Setelah sistem mengirim perintah serial kepada servo, sistem akan otomatis mengulang pendeteksian karena sistem ini bersifat real time.

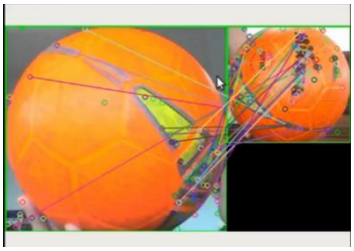
4. HASIL DAN PEMBAHASAN

4.1. Implementasi Algoritma SIFT

Algoritma SIFT bekerja dengan cara menandai feature lokal dalam matriks citra yang disebut keypoint dan kemudian membandingkan apakah terdapat kemiripan keypoint dengan sampel yang telah disiapkan. Berikut hasil implementasi algoritma SIFT dapat dilihat pada gambar 8 dan 9.



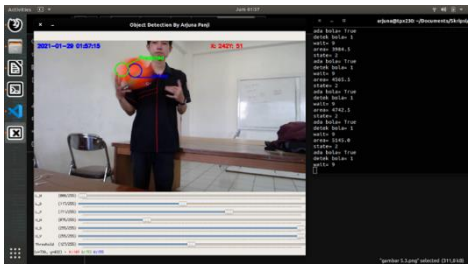
Gambar 8. Pendeteksian Menggunakan Algoritma SIFT



Gambar 9. Membandingkan Keypoint Dari Kedua Citra

4.2. Implementasi Algoritma Kalman Filter

Algoritma kalman filter berfungsi untuk memprediksi state objek berdasarkan state sebelumnya. Untuk hasil implementasi algoritma kalman filter dapat dilihat pada gambar 10 berikut:



Gambar 10. Pendeteksian Menggunakan Algoritma Kalman Filter

4.3. Pengujian Deteksi Menggunakan Algoritma SIFT

Gambar 11. Pengujian Pendeteksian Menggunakan SIFT

Pengujian ini dilakukan dengan tujuan untuk melihat sejauh mana kamera webcam dapat mendeteksi objek dengan baik saat menggunakan algoritma SIFT. Hasil pengujian dapat dilihat pada tabel 1.

Tabel 1. Pengujian Jarak Deteksi Menggunakan Algoritma SIFT

Percobaan	Jarak	Koordinat(x,y)	Kesimpulan
1	30 cm	(120,42)	Berhasil
2	50 cm	(321,44)	Berhasil
3	70 cm	(294,84)	Berhasil
4	90 cm	(275,65)	Berhasil
5	110 cm	(282,132)	Berhasil
6	130 cm	(190,87)	Berhasil
7	150 cm	(287,68)	Berhasil
8	170 cm	(291,121)	Berhasil
9	190 cm	(342,144)	Berhasil
10	210 cm	(-,-)	Tidak Berhasil

Dari tabel 1 merupakan hasil dari pengujian jarak deteksi menggunakan algoritma SIFT. Pada percobaan pertama pengujian dilakukan dengan jarak 30 cm dengan koordinat (x,y) adalah (120,42) dengan kesimpulan berhasil, artinya objek terdeteksi oleh kamera. Percobaan dilakukan hingga jarak 190 cm dengan koordinat (x,y) adalah (342,144) dengan kesimpulan berhasil terbaca. Percobaan terakhir di lakukan pada jarak 210 cm dengan koorniat (xy) adalah (-,-) dengan kesimpulan tidak berhasil, artinya pada jarak ini objek tidak terdeteksi oleh kamera.

4.4. Pengujian Deteksi Menggunakan Algoritma Kalman Filter

Gambar 12. Pengujian Pendeteksian Menggunakan Kalman Filter

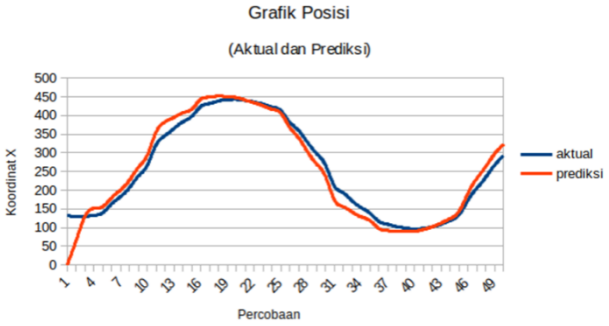
Gambar 13. Pengujian Jarak Deteksi Menggunakan Kalman Filter

Pengujian ini dilakukan untuk melihat hasil dari pengaplikasian algoritma Kalman Filter dalam sistem pendeteksian bola. Hasil pengujian dapat dilihat pada tabel 2.

Tabel 2. Pengujian Jarak Deteksi Menggunakan Algoritma Kalman Filter

Percobaan	Jarak	Koordinat(x,y)	Kesimpulan
1	30 cm	(154,44)	Berhasil
2	50 cm	(186,33)	Berhasil
3	70 cm	(222,40)	Berhasil
4	90 cm	(255,56)	Berhasil
5	110 cm	(294,73)	Berhasil
6	130 cm	(174,93)	Berhasil
7	150 cm	(390,84)	Berhasil
8	170 cm	(318,145)	Berhasil
9	190 cm	(217,112)	Berhasil
10	210 cm	(-,-)	Tidak Berhasil

Dari tabel 2 merupakan hasil dari pengujian jarak deteksi menggunakan algoritma kalman filter. Pada pecobaan pertama dilakukan dengan jarak 30 cm dengan koordinat (x,y) adalah (154,44) dengan kesimpulan berhasil, yang artinya dengan menggunakan algoritma kalman filter objek terbaca oleh kamera. Percobaan dilakukan hingga jarak 190 cm dengan koordinat (x,y) adalah (217,112) dengan keimpulan berhasil. Percobaan selanjutnya dengan jarak 210 cm koornidat (x,y) adalah (-,-) dengan kesimpulan tidak berhasil, yang artinya pada jarak ini objek sudah tidak terbaca lagi oelh kamera.
Berdasarkan hasil pengujian pendeteksian objek menggunakan kalman filter maka diperoleh hasil seperti dapat dilihat pada gambar 14.



Gambar 14. Grafik Posisi (Aktual dan Prediksi)

5. KESIMPULAN

Berdasarkan hasil pengujian dalam penelitian ini berhasil melakukan deteksi menggunakan algoritma SIFT dan algoritma Kalman Filter dengan sempurna pada jarak tertentu. Tracking yang digunakan dalam penelitian ini adalah algoritma kalman filter. Koordinat y akan bertambah kecil jika ibjek bergerak ke atas pada frame dan bertambah besar jika obek bergerak ke bawah frame. Sedangkan pada koordinat x akan bertambah kecil jika objek bergerak ke arah kiri frame, jika ke arah kanan frame maka akan bertambah besar.Pengaruh perpindahan posisi terhadap nilai prediksi yang dihasilkan Kalman filter terdapat rata-rata error pengukuran untuk koordinat x sebesar 1.06 dan 7.34 untuk koordinat y.

DAFTAR PUSTAKA

[1] S. Arifin and E. T. Wijaya, "Implementasi Teknologi Computer Vision Sebagai Pengendali Mobile Robot Berbasis Kamera Web," *Jouticla*, vol. 2, no. 2, pp. 75–80, 2017, doi: 10.30736/jti.v2i2.72.

[2] F. B. Setiawan, O. J. Aldo Wijaya, L. H. Pratomo, and S. Riyadi, "Sistem Navigasi Automated Guided Vehicle Berbasis Computer Vision dan Implementasi pada Raspberry Pi," *J. Rekayasa Elektr.*, vol. 17, no. 1, pp. 7–14, 2021, doi: 10.17529/jre.v17i1.18087.

[3] R. Lionnie and M. Alaydrus, "Sistem Pendeteksi Gambar Termanipulasi Menggunakan Metode SIFT," *Techné J. Ilm. Elektrotek.*, vol. 16, no. 02, pp. 133–140, 2017, doi: 10.31358/techne.v16i02.166.

[4] H. Vazirani, A. Kautsar, J. Fisika, F. Sains, and U. Diponegoro, "IMPLEMENTASI OBJECT TRACKING UNTUK MENDETEKSI DAN MENGGUNAKAN METODE KALMAN FILTER DAN GAUSSIAN MIXTURE MODEL," vol. 5, no. 1, 2016.

[5] P. Tanpa, A. Menggunakan, P. S. Ardiantara, R. Sumiharto, and S. B. Wibowo, "Purwarupa Kontrol Kestabilan Posisi dan Sikap pada Pesawat Tanpa Awak Menggunakan IMU dan Algoritma Fusion Sensor Kalman Filter," vol. 4, no. 1, pp. 25–34, 2014.

[6] M. D. Irawan and S. A. Simargolang, "Implementasi E-Arsip Pada Program Studi Teknik Informatika," vol. 2, no. 1, 2018.

[7] D. A. Prabowo and D. Abdullah, "Deteksi dan Perhitungan Objek Berdasarkan Warna Menggunakan Color Object Tracking," *Pseudocode*, vol. 5, no. 2, pp. 85–91, 2018, doi: 10.33369/pseudocode.5.2.85-91.

[8] Ikhsan and P. Ayomi, "Implementasi Raspebrry PI Pada ARM Robot Penyortir Benda Berdasarkan Warna dan Bnetuk," vol. 6, no. 2, pp. 176–182, 2019.

[9] M. Fuadi, U. Darusalam, and A. K. Whardana, "FACE RECOGNITION MENGGUNAKAN OPENCV DENGAN BAHASA PEMOGRAMAN PYTHON OOP UNTUK SISTEM," vol. 2, no. 3, pp. 218–224, 2021.

[10] P. Dlib, "Pendeteksian Kantuk Secara Real Time Menggunakan Pustaka OPENCV dan DLIB PYTHON Real Time Sleepiness Detection Using OPENCV Library and PYTHON DLIB," vol. 28, no. 2, pp. 22–26, 2018.

[11] D. Ayu and B. Utami, "Perancangan Sistem Login Pada Aplikasi Berbasis GUI Menggunakan QTDesigner Python," vol. 4, no. 2, pp. 92–100, 2021.

[12] A. N. Syahrudin and T. Kurniawan, "Input dan output pada bahasa pemrograman python," no. January, 2020.

[13] A. T. Khomaeni, "PENERAPAN METODE SCALE INVARIANT FEATURE TRANSFORM (SIFT) PADA AUGMENTED REALITY DALAM PENGENALAN GEDUNG UNIVERSITAS ISLAM NEGERI MALANG Oleh," 2020.

[14] R. Y. Amrullah, "Pengenalan Benda di Jalan Raya dengan Metode Kalman Filter," 2015.

[15] J. Ali, "SISTEM SECURITY WEBCAM DENGAN MENGGUNAKAN MICROSOFT VISUAL BASIC (6.0)," vol. 1, no. 2, pp. 48–60, 2016.