Jurnal Engineering

Volume 5 Nomor 1, Februari 2023
P-ISSN : 2461-0526
E-ISSN : 2623-1522

Prototype Sistem Pendeteksi Objek Mobil pada Lampu Lalu Lintas Menggunakan Metode *Blob Detection*

Ursi Aulia[1], Samratul Fuady[1], Yosi Riduas Hais[1]
[1]Program Studi Teknik Elektro, Fakultas Sains dan Teknologi, Universitas Jambi
Email: ursiaulia20@gmail.com, sfuady@unja.ac.id, yosi.riduas@unja.ac.id

Info Artikel

Diterima: 1 Februari 2023
Disetujui: 20 Februari 2023
Dipublikasikan: 28 Februari 2023

Alamat Korespondensi:
ursiaulia20@gmail.com

Abstrak

Pengolahan citra merupakan salah satu solusi yang dapat membantu dalam mengatasi masalah lalu lintas. Salah satunya dapat mengatur durasi lampu lalu lintas secara tepat sesuai dengan jumlah kendaraan yang terdeteksi. Penelitian ini dilakukan untuk mendeteksi jumlah kendaraan yang ada di lampu lalu lintas. *Input* objek dari *webcam* diolah dengan beberapa tahapan yaitu proses pengubahan ukuran, pengubahan citra menjadi skala abu-abu hingga mendapatkan *keypoint* dari objek. Hasil pengujian terhadap jarak menunjukkan bahwa objek dapat terdeteksi ketika berjarak 15-45 cm. Pada pengujian terhadap cahaya, objek dapat terdeteksi ketika diuji dengan intensitas cahaya diatas 35 *lux*. Pengujian secara *realtime* memiliki *accuracy* 89%, *precision* 95%, *recall 92%* dan F1-*Score* sebesar 93%.

Kata Kunci: Pengolahan citra, pendeteksian objek, Metode *Blob Detection*

Abstract **:**

Image processing is one solution that can help in overcoming future traffic problems. One of them is to set the duration of traffic lights appropriately according to the number of detected vehicles. This research was conducted to detect the number of vehicles at traffic lights. The object input from the webcam is processed in several stages, namely resizing, converting the image to gray scale to get the keypoint of the object. The test results for the distance show that objects can be detected when they are 15-45 cm away. For the light testing, the object can be detected when tested with a light intensity above 35 lux. Realtime testing has 89% accuracy, 95% precision, 92% recall and 93% F1-Score.

Keywords: *Image processing, object detection, Blob Detection Method*

1. Pendahuluan

Saat ini terdapat berbagai permasalahan lalu lintas karena meningkatnya jumlah kendaraan di Indonesia. Dengan bertambahnya jumlah kendaraan, fasilitas jalan saat ini tidak dapat mengimbangi sehingga menyebabkan masalah lalu lintas, salah satunya adalah kemacetan. Pengolahan citra dapat membantu dalam mengatasi masalah lalu lintas dengan sebuah sistem yang dapat mendeteksi kendaraan menggunakan kamera.

Pengolahan citra adalah cara untuk membuat gambar terlihat lebih baik sesuai dengan kebutuhan. Gambar input berkualitas buruk, buram, dan penuh *noise*. Namun, dengan menerapkan langkah pemrosesan, gambar dapat dibuat jauh lebih jelas dan mudah dibaca (Dalimunthe, 2020). Pemrosesan gambar adalah istilah luas yang mengacu pada berbagai teknik untuk memanipulasi dan memodifikasi gambar dengan mengenali pola.

Penelitian ini dilakukan untuk mendeteksi jumlah kendaraan di lampu lalu lintas. Data jumlah kendaraan yang terdeteksi kedepannya dapat digunakan untuk mengatasi berbagai permasalahan lalu lintas. Pengolahan citra digunakan untuk membuat sebuah sistem yang dapat mendeteksi objek menggunakan *webcam*.

Penelitian terkait yang menngunakan metode *blob detection* sudah dilakukan sebelumnya seperti penelitian oleh (Fitri Yana, 2020) dengan judul Implementasi Pengolahan Citra Digital pada Perhitungan Anak Burung Puyuh Menerapkan Metode *Blob*. Penelitian ini membuat sistem yang dapat mendeteksi dan menghitung jumlah anak brung puyuh dengan pengolahan citra. Dalam hal ini citra digital akan dapat mendeteksi jumlah anak burung puyuh melalui beberapa proses sebelum melakukan pendeteksian. Dengan menggunakan morfologi citra *dilasi* dan *erosi* maka dapat dilakukan pemisahan anak burung puyuh yang saling tumpang tindih atau berdekatan. Pendeteksian anak burung puyuh menggunakan metode *blob*, dalam hal ini citra digital akan dapat mendeteksi sebuah objek baik yang bergerak maupun tidak. Metode *blob* atau *blob detection* yaitu mendeteksi suatu kumpulan titik-titik piksel yang memiliki warna berbeda (lebih terang atau gelap).

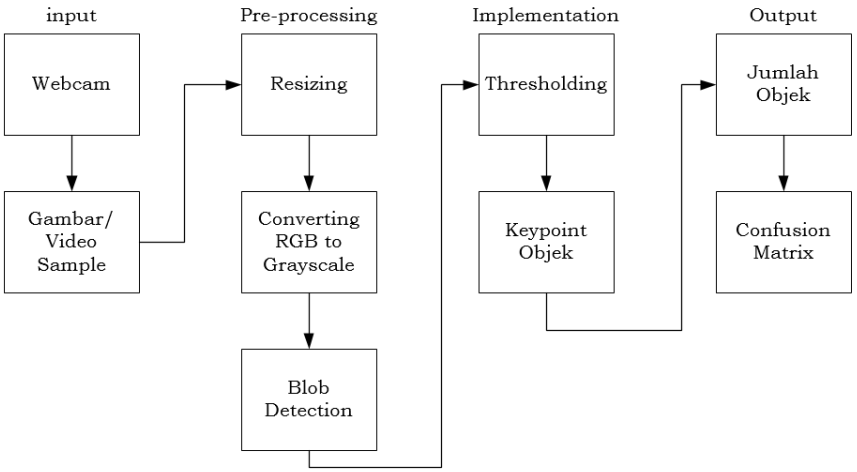
Berdasarkan permasalahan yang telah dipaparkan sebelumnya dan didukung oleh beberapa penelitian yang telah dilakukan maka pada penelitian ini akan membangun sistem untuk mendeteksi objek menggunakan metode *blob detection*. Sistem Ini diharapakan dapat membantu dalam menangani permasalahan lalu lintas kedepannya. Pada penelitian yang akan dilakukan yaitu deteksi citra objek mobil dari *webcam*. Sistem ini akan dibuat menggunakan *software Visual Studio Code* menggunakan bahasa pemrograman *python* dan hanya di implementasikan dengan *prototype* pada 1 jalur jalan dengan menerapkan metode *Blob Detection*.

2. Metode Penelitian

Gambar 1 menunjukkan gambaran umum dari *prototype* sistem pendeteksi objek dalam bentuk diagram blok. Terdapat 4 bagian yaitu *input, pre-processing, implementation* dan *output*.

a. *Input*

Input atau masukan dari sistem ini yaitu gambar atau video *sample* objek yang diperoleh dari *webcam*. *Input* digunakan untuk diproses ketahap berikutnya.



Gambar 1. Diagram blok sistem

b. *Pre-processing*

Beberapa tahapan pengolahan citra dilakukan pada bagian ini. Tahapan yang pertama yaitu *resizing* yaitu mengubah ukuran resolusi dari *input*. Setelah pengubahan ukuran selesai, langkah selanjutnya adalah mengubah citra berwarna RGB menjadi citra *grayscale* atau berwarna abu-abu untuk mengurangi kompleksitas gambar. Selanjutnya yaitu ketahap implementasi metode yang digunakan yaitu *blob detection*. Pada metode blob detection menggunakan teknik Thresholding atau teknik ambang batas yang mengeksploitasi informasi yang berbeda seperti bentuk, pengelompokan, entropi dan atribut objek.

c. *Implementation*

Pada tahap ini yaitu proses *thresholding* dan keypoint objek. *Thresholding* atau ambang batas dilakukan dengan kalibrasi secara berkala dengan mengubah nilai maksimal dan minimal yang tepat agar objek dapat diketahui. Selanjutnya yaitu agar sistem dapat mendeteksi objek secara tepat maka dilakukan penentuan *keypoint* dari objek.

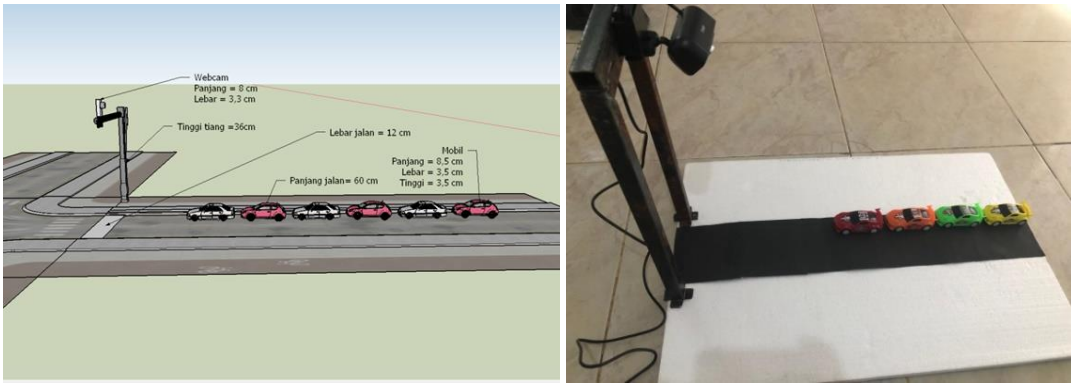
d. *Output*

Setelah tahap *pre-processing* dan *implementation* dilakukan maka dilakukan beberapa pengujian untuk mengetahui kinerja dari sistem. *Output* yang dihasilkan yaitu pengujian jarak, pengujian terhadap cahaya lingkungan serta pengujian secara *realtime*. Sistem juga dapat menampilkan jumlah objek yang terdeteksi secara *realtime*. Setelah dilakukan beberapa pengujian maka dapat diukur tingkat keakurasian dari sistem

3. Hasil Penelitian dan Pembahasan

a. Implementasi *Prototype*

Prototype sistem pendeteksi objek yang dibuat terdiri dari beberapa komponen, dimana posisinya diatur sesuai kebutuhan agar dapat bekerja sesuai dengan yang diharapkan. Salah satu hal yang perlu diperhatikan adalah posisi dan ketinggian *webcam* yang mempengaruhi pendeteksian objek.



Gambar 2. Perbandingan Rancangan dan Hasil Jadi Prototype

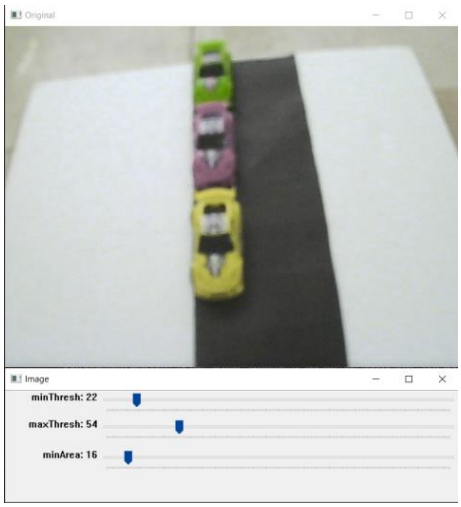
Pada Gambar 2 terlihat pebandingan dari rancangan dan *prototype* sistem pendeteksi objek yang telah jadi. Hasil jadi dari *prototype* yang dibuat hampir sama dengan rancangan. Alas dari *prototype* menggunakan *styrofoam* yang berukuran 60 x 40 cm. Sedangkan tiang penyangga *webcam* terbuat dari besi yang kokoh sehingga *webcam* tidak akan bergeser. Posisi *webcam* yang stabil dapat membantu dalam proses pengujian sistem. Tinggi tiang berbeda dengan rancangan yaitu dengan tinggi 38 cm. Adapun jalan pada *prototype* menggunakan kertas karton yang telah dipotong sesuai ukuran rancangan yaitu dengan panjang 60 cm dan lebar 40 cm. Objek mobil mainan yang digunakan memiliki jumlah yang berbeda dari rancangan yaitu hanya menggunakan 4 objek mobil mainan dengan ukuran 8 x 3.5 cm. Pada saat pengujian objek hanya bisa tertangkap dengan *full* sebanyak 4 objek mobil..

b. *Preprocessing*

Preprocessing memiliki beberapa tahapan yaitu pengumpulan data, *resizing* dan *convert to grayscale*. Tahap pertama yang dilakukan adalah mengumpulkan data objek sebagai input untuk diproses nantinya. Data objek berupa video didapatkan dari *webcam* yang sudah terpasang dengan *Personal Computer*/Laptop. Data yang digunakan dalam penelitian ini yaitu pada saat kondisi 1 objek, 2 objek dan 3 objek. Selanjutnya proses *resizing* yang merupakan proses mengubah ukuran pada *input* yaitu video. Pada penelitian ini ukuran *input* video dari *webcam* memiliki ukuran yang terlalu besar dengan *frame width* 2560 dan *frame height* 1440 (QHD/Quad HD). Oleh karena itu dilakukan pengecilan ukuran resolusi yaitu dengan *frame width* 640 dan *frame height* 360. Langkah selanjutnya setelah proses pengubahan ukuran yaitu proses skala abu-abu, proses *Grayscale* sendiri yaitu mengubah atau membuat warna gambar RGB menjadi gambar hitam putih abu-abu untuk memudahkan langkah selanjutnya.

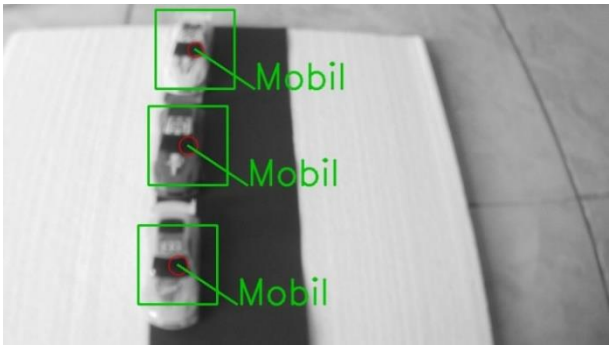
c. Implementasi *Blob Detection*

Implementasi *blob detection* memiliki beberapa bagian yaitu proses *threshold* adalah proses penentuan ambang batas objek pada citra. Threshold dilakukan dengan mengkalibrasi nilai ambang batas. Dimana nilai ambang batas terdiri dari nilai *minThresh*, *maxThresh*, dan *minArea*. Proses penentuan ambang batas dengan mengubah nilai parameter berulang kali pada *trackbar* untuk menentukan nilai yang tepat. Berikut tampilan beberapa proses pengkalibrasian.



Gambar 3. Kalibrasi parameter

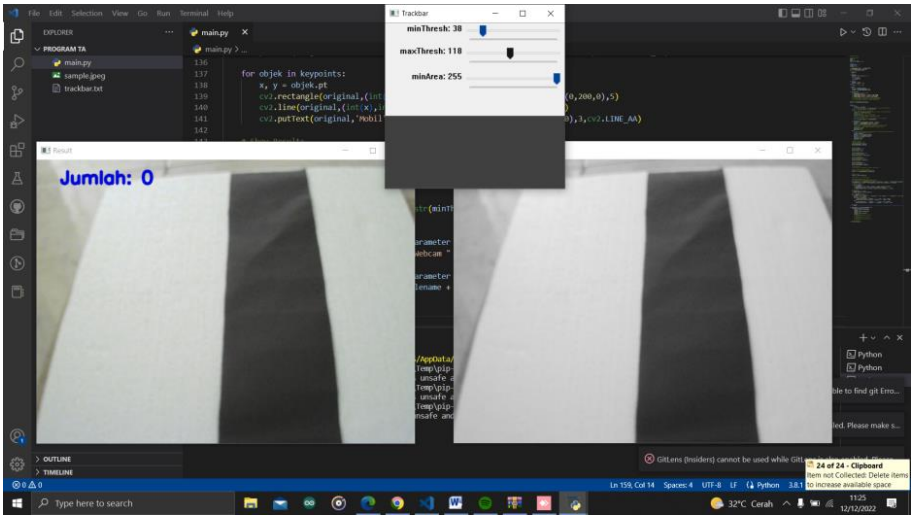
Parameter yang telah di dapatkan dan tersimpan kedalam *variable detector* akan digunakan untuk mendapatkan *keypoint* objek. *Keypoint* didapatkan setelah dilakukan penyaringan gumpalan. *Filtering* atau penyaringan dilakukan dari area, *circularity, convexity dan inertia.*



Gambar 4. *Keypoint* objek

d. Pengujian *webcam*

Pengujian dilakukan dengan menghubungkan *webcam* ke perangkat laptop/PC melalui kabel USB. Setelah *webcam* terhubung ke laptop, uji citra objek mobil *webcam* dan menggunakan *Microsoft Visual Code.*



Gambar 5. *Webcam terhubung*

e. Pengujian Jarak

Pengujian dilakukan dari tiang *webcam* terhadap objek dengan jarak 10 cm hingga 50 cm dengan parameter yang sama. Hasil pengujian yaitu ketika jarak objek 10 cm sistem tidak dapat mendeteksi objek karena *keypoint* dari objek tidak terdeteksi. Selanjutnya yaitu pengujian pada jarak 15 cm, 20 cm, 25 cm, 30 cm, 35 cm, 40 cm, 45 cm. Pengujian terakhir yaitu dengan jarak 50 cm, objek tidak dapat terdeteksi karena sistem tidak mendapatkan *keypoint* dari objek. Pada jarak 10 cm dan 50 cm objek hanya tertangkap sedikit oleh *webcam* yang menyebabkan sistem tidak dapat mendeteksi sebagai objek.

Tabel 1. Pengujian Jarak

Jarak (cm)	Keterangan
10 cm	Objek Tidak Terdeteksi
15 cm	Objek Terdeteksi
20 cm	Objek Terdeteksi
25 cm	Objek Terdeteksi
30 cm	Objek Terdeteksi
35 cm	Objek Terdeteksi
40 cm	Objek Terdeteksi
45 cm	Objek Terdeteksi
50 cm	Objek Tidak Terdeteksi

f. Pengujian Cahaya Lingkungan

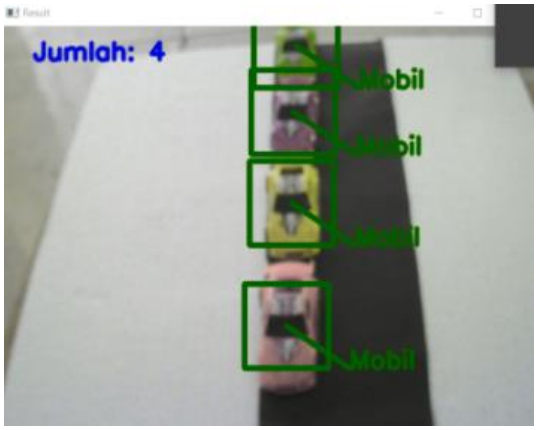
Berdasarkan pengujian yang telah dilakukan, objek diuji dari gelap hingga terang. Pengujian pertama yaitu dengan 0 *lux*, yang diuji didalam ruangan pada sore hari. Pada kondisi ini tidak adanya penerangan tambahan seperti lampu, hasil pendeteksian sistem pada kondisi ini yaitu objek tidak terdeteksi karena sistem tidak mendapatkan *keypoint* dari objek karena terlalu gelap. Sistem tidak dapat membedakan antara objek dan *background*. Selanjutnya yaitu pengujian dengan intensitas cahaya 35 *lux*, pengujian dilakukan didalam ruangan dengan adanya penerangan lampu pada pagi hari, hasil pendeteksian yaitu objek dapat terdeteksi. Pengujian selanjutnya yaitu dengan intensitas cahya 195 *lux*, pengujian ini dilakukan didalam ruangan pada siang hari dengan adanya penerangan lampu. Selanjutnya yaitu pengujian diluar ruangan pada pagi hari dengan intensitas cahaya 301 *lux*, hasil pengujian yaitu objek dapat terdeteksi. Pengujian berikutnya juga dilakukan diluar ruangan pada sore hari, dengan intensitas cahaya sebesar 668 *lux*, pada kondisi ini sistem dapat mendeteksi objek. Pengujian yang terakhir yaitu dilakukan diluar ruangan pada siang hari. Pada kondisi ini merupakan intensitas tertinggi diantara pengujian lainnya yaitu 6654 *lux*. Hasil pendeteksian menunjukkan bahwa sistem dapat mendeteksi objek dengan baik.

Tabel 2. Pengujian Cahaya Lingkungan

Intensitas Cahaya (Lux)	Keterangan
0 Lux	Objek Tidak Terdeteksi
35 *Lux*	Objek Terdeteksi
195 *Lux*	Objek Terdeteksi
301 *Lux*	Objek Terdeteksi
668 *Lux*	Objek Terdeteksi
6654 *Lux*	Objek Terdeteksi

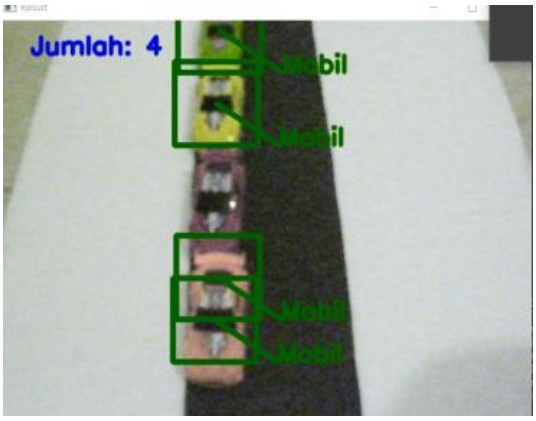
g. Pengujian *Realtime*

Pengujian secara realtime dilakukan sebanyak 4 kali pengujian disetiap kondisi dan dengan total pengujian 20 kali pengujian. Pengujian dimulai dari kondisi tanpa objek, 1 objek, 2 objek, 3 objek dan 4 objek. Setelah *webcam* dihubungkan ke personal *computer/laptop*, lakukan *run* program maka *output* akan terampil ke layar. Waktu pendeteksian sangat cepat dibawah 1 detik ketika parameternya sudah tepat.



Gambar 6. Pengujian realtime

Sistem dapat mendeteksi objek sesuai dengan jumlah sebenarnya seperti pada Gambar 6. Pengujian dilakukan dengan parameter yang sama yaitu dengan nilai *minThresh* 59, nilai *maxThresh* 78 dan nilai *minArea* 255. Selanjutnya dengan jumlah objek yang sama dan parameter yang sama dilakukan pengujian yang memiliki 3 kondisi yaitu jumlah objek yang terdeteksi ada 3, jumlah objek yang tidak terdeteksi 1 dan jumlah tidak ada objek tetapi sistem mendeteksi ada objek 1. Terlihat pada gambar bahwa sistem mendeteksi 2 objek yang jumlah sebenarnya adalah 1 objek atau terdeteksi *double*. Sistem mendeteksi 2 keypoint pada objek yang sama, penyebabnya karena nilai parameter yang tidak pas sehingga *keypoint* juga tidak akan akurat karena *keypoint* didapat berdasarkan parameter. Selanjutnya untuk kondisi objek yang tidak terdeteksi, terlihat bahwa sistem tidak dapat mendeteksi objek karena sistem tidak mendapatkan *keypoint* dari objek. Ketika tingkat warna objek hampir sama dengan *background*, sistem tidak akan mendeteksi secara akurat. Maka diperlukan cahaya tambahan atau menggunakan kamera *infrared* untuk kontras warna objek yang baik.



Gambar 7. Pengujian realtime

Untuk pengujian menyeluruh memiliki hasil yang berbeda-beda. Oleh karena itu digunakan *confusion matrix* untuk mengevaluasi pendeteksian ini. Nilai yang diambil yaitu *True Positif* (TP) artinya ada objek mobil dan sistem mengenali nya sebagai objek mobil, *True Negative* (TN) artinya tidak ada objek mobil dan sistem tidak mendeteksi adanya objek mobil, *False Positif* (FP) artinya tidak ada objek mobil tapi sistem menganggap ada objek mobil, *False Negatif* (FN) artinya ada objek mobil dan sistem tidak mendeteksi adanya objek.

Tabel 3. Evaluasi Sistem

n=40	Aktual : Positif	Aktual : Negatif
Prediksi : Positif	TP : 37	FP : 2
Prediksi Negatif	FN : 3	TN : 4

Berikut adalah perhitungan dari *confusion matrix*

1. *Accuracy*

Untuk mendapatkan nilai akurasi dari sistem digunakan rumus sebagai berikut.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \times 100\%$$
$$= \frac{37+4}{37+4+2+3} \times 100\%$$
$$= 89\%$$

Dari hasil perhitungan dapat dilihat bahwa sistem yang dirancang memiliki keakuratan sebesar 89%

2. *Precision*

Untuk menentukan nilai presisi dari sistem digunakan rumus sebagai berikut.

$$Precision = \frac{TP}{TP+FP} = \frac{37}{37+2} = 0.95 \text{ atau } 95\%$$

Dari hasil perhitungan dapat dilihat bahwa sistem yang dirancang memiliki presisi sebesar 0.95 atau 95%.

3. *Recall*

Untuk menentukan nilai *recall* dari sistem digunakan rumus sebagai berikut.

$$Recall = \frac{TP}{TP+FN} = \frac{37}{37+3} = 0.92 \text{ atau } 92\%$$

Dari hasil perhitungan dapat dilihat bahwa sistem yang dirancang memiliki *recall* sebesar 0.92 atau 92%.

4. *F1-Score*

Untuk menentukan nilai F1-*Score* dari sistem digunakan rumus sebagai berikut.

$$\text{F1-}Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} = 2 \times \frac{0.95 \times 0.92}{0.95+0.92} = 0.93 \; atau \; 93\%$$

Dari hasil evaluasi sistem yaitu dengan tingkat akurasi 89%, presisi sebesar 0.95, *recall* sebesar 0.92 dan F1-*Score* sebesar 0.93. Dari hasil tersebut sistem dapat dikategorikan sudah cukup baik dalam mendeteksi objek mobil sesuai yang diharapkan. Dari beberapa pengujian yang telah dilakukan seperti terhadap jarak dan cahaya, objek dapat terdeteksi. Namun untuk pendeteksian jumlah objek yang banyak harus memperhatikan metode yang digunakan dan kamera yang digunakan.

Kesimpulan

Berdasarkan hasil penelitian maka dapat diambil beberapa kesimpulan yaitu sistem pendeteksi objek dengan metode *blob detection* dirancang dengan meggunakan *input* video sebagai data utama untuk diproses ketahap selanjutnya. *Input* diproses dengan beberapa tahapan dintaranya, tahap *resizing*, konversi citra RGB ke citra *grayscale*, *thresholding* dan *filtering* objek. Setelah *input* diolah maka didapatkan *keypoint* dari objek sehingga objek dapat dideteksi secara *realtime*. Berdasarkan pengujian yang telah dilakukan, objek dapat terdeteksi dengan berbagai jarak yang masih dalam cakupan *webcam*. Sistem juga dapat mendeteksi dengan berbagai tingkat *intensitas* cahaya. *Intensitas* cahaya yang tinggi dapat mempermudah pendeteksian karena objek terlihat jelas. Pengujian secara *realtime* juga dilakukan dan memiliki tingkat akurasi sebesar 89%, nilai presisi 95%, nilai *recall* 92% dan nilai F1-*Score* sebesar 93%.

Daftar Pustaka

[1] A. Kaspers. 2011. "Blob detection," English, Image Science Institute, UMC Utrecht, Tech. Rep.

[2] Basuki, A., Palandi, J., & Fatchurrochman. 2005. Pengolahan citra digital menggunakan Visual Basic. Yogyakarta: Graha Ilmu.

[3] Dalimunthe, Ahmad Houlan. 2020. Segmentasi Citra MRI dengan Menggunakan Metode BLOB. Vol. 1, No. 2, 103-109.

[4] Fitri Yana Ade. 2020. "Implementasi Pengolahan Citra Digital Pada Perhitungan Anak Burung Puyuh Menerapkan Metode *Blob*". Vol. 1, No. 4, 237-245.

[5] M. Sezgin and B. Sankur. 2004. "Survey over image thresholding techniques and quantitative performance evaluation," J. Electronic Imaging, vol. 13, pp. 146–168.

[6] Miftahul, Ulum, et al. 2022. "Design and Build a Vaname Shrimp Sorting System Based on Image Processing". Vol. 6, No. 2.

[7] Munir, R. 2004. Pengolahan Citra Digital dengan Pendekatan Algoritmik. Bandung: Informatika.

[8] Putra, Darma. 2010. Pengolahan Citra Digital. Yogyakarta: Penertbit Andi.

[9] R. D. Kusumanto and A. N. Tompunu. 2011. "Pengolahan Citra Digital Untuk Mendeteksi Obyek Menggunakan Pengolahan Warna Model Normalisasi Rgb". Semantik : Palembang.

[10] Triwibowo, et al. 2020. "Analisis *Blob Detection* Pada Pendeteksian Dan Perhitungan Kendaraan di Jalan Tol". Jurnal Teknologi Informasi, vol. 10, no. 1, pp. 1-8.

[11] W. S. Qureshi, A. Payne, K. B. Walsh, R. Linker, O. Cohen, and M. N. Dailey. 2017. "Machine vision for counting fruit on mango tree canopies," en, Precision Agriculture, vol. 18, no. 2, pp. 224–244.

[12] Y. Li, S. Wang, Q. Tian, and X. Ding. 2015. "A survey of recent advances in visual feature detection," Neurocomputing, vol. 149, pp. 736–751.