

Optimasi Komputer Vision Pada Turtlebot Sebagai Pendeteksi Objek Benda Kotak

Ach Tohir¹, Anang Habibi², Taqiyyuddin Alawiy³

^{1,2,3}) Universitas Islam Malang

tohirt350@gmail.com, ananghabibi@unisma.ac.id, taqijuddin.alawiy@unisma.ac.id

Abstract

The ability of a robot to detect box-shaped objects is crucial for applications that require interaction with the environment, such as navigation systems and automated object retrieval. This research implements and optimizes box-shaped object detection on a TurtleBot equipped with a Raspberry Pi camera module and the YOLO (You Only Look Once) algorithm. The object detection process includes stages of image acquisition, image processing, and the application of the YOLO algorithm to recognize box-shaped objects. Testing was conducted under three different lighting conditions: dark (0-50 lumens), dim (50-200 lumens), and bright (200-500 lumens), with distance variations ranging from 20 to 70 cm. The test results indicate that optimizing computer vision on the TurtleBot improves detection accuracy, especially under bright lighting conditions. Optimizing camera parameters, such as exposure adjustment and light sensitivity, plays a crucial role in enhancing detection performance across various lighting conditions, although accuracy decreases in dim and dark conditions. In conclusion, optimizing the computer vision system on the TurtleBot significantly enhances its ability to detect box-shaped objects. Adjusting camera parameters and the YOLO algorithm enables the TurtleBot to detect objects more accurately and efficiently, particularly in optimal lighting conditions. This research emphasizes that computer vision optimization is key to developing reliable and adaptive robots for various environmental interaction applications.

Keywords — computer vision optimization, object detection, TurtleBot, Raspberry Pi camera, YOLO.

Abstraksi

Kemampuan robot dalam mendeteksi objek berbentuk kotak sangat penting untuk aplikasi yang membutuhkan interaksi robot dengan lingkungan, seperti dalam sistem navigasi dan pengambilan objek secara otomatis. Penelitian ini mengimplementasikan dan mengoptimalkan deteksi objek berbentuk kotak pada TurtleBot yang dilengkapi modul kamera Raspberry Pi dan algoritma YOLO (You Only Look Once). Proses deteksi objek mencakup tahapan akuisisi gambar, pemrosesan citra, dan penerapan algoritma YOLO untuk mengenali objek berbentuk kotak. Pengujian dilakukan dalam tiga kondisi pencahayaan yang berbeda: gelap (0-50 lumen), redup (50-200 lumen), dan terang (200-500 lumen), dengan variasi jarak antara 20 hingga 70 cm. Hasil pengujian menunjukkan bahwa optimasi computer vision pada TurtleBot meningkatkan akurasi deteksi, terutama pada pencahayaan terang. Optimalisasi parameter kamera, seperti penyesuaian eksposur dan sensitivitas cahaya, berperan penting dalam meningkatkan performa deteksi pada berbagai kondisi pencahayaan, meskipun akurasi menurun pada kondisi redup dan gelap. Dengan kesimpulan, optimasi sistem computer vision pada TurtleBot secara signifikan meningkatkan kemampuan deteksi objek berbentuk kotak. Penyesuaian parameter kamera dan algoritma YOLO memungkinkan TurtleBot untuk mendeteksi objek dengan lebih akurat dan efisien, terutama dalam kondisi pencahayaan optimal. Penelitian ini menegaskan bahwa optimasi computer vision adalah kunci untuk mengembangkan robot yang andal dan adaptif dalam berbagai aplikasi interaksi dengan lingkungan.

Kata Kunci — optimasi computer vision, deteksi objek, TurtleBot, kamera Raspberry Pi, YOLO.

I. PENDAHULUAN

Kemajuan teknologi telah memudahkan pencapaian berbagai tujuan dengan lebih efisien dan praktis. Salah satu teknologi yang berkembang pesat adalah robotika, yang berakar dari kata "robota" dalam bahasa Ceko, yang berarti "kerja cepat." Istilah ini diperkenalkan oleh Karel Čapek dalam sandiwara "R.U.R. (Rossum's Universal Robots)" pada tahun 1920. Perkembangan robotika tidak terlepas dari kemajuan teknologi lain, seperti kecerdasan buatan

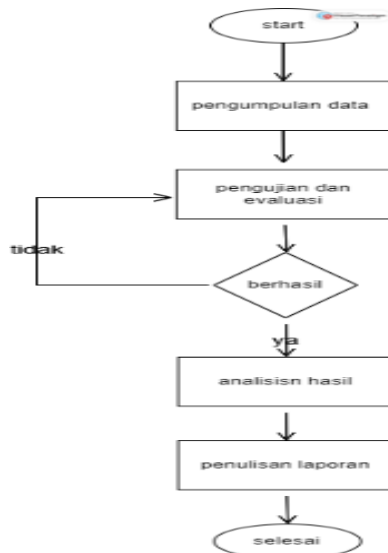
(AI), pembelajaran mesin (machine learning), dan teknologi sensor. AI memungkinkan robot belajar dari pengalaman, sementara machine learning memfasilitasi robot untuk mengenali pola dalam data dan meningkatkan kinerjanya tanpa perlu pemrograman ulang secara ekstensif. Teknologi sensor, seperti kamera, lidar, dan sensor ultrasonik, memungkinkan robot berinteraksi lebih akurat dan responsif dengan lingkungan di sekitarnya. Sistem kontrol robot mencakup tiga komponen utama: masukan, pemroses, dan keluaran. Untuk mendeteksi objek tertentu,

biasanya robot menggunakan sensor berupa kamera, yang menangkap gambar dan mengubahnya menjadi citra digital yang dapat dianalisis lebih lanjut. Computer vision menjadi elemen penting dalam robotika, mencakup akuisisi, pemrosesan, pengenalan, dan pengambilan keputusan berbasis citra. Tahapan pertama dalam visi komputer adalah akuisisi citra, diikuti dengan pemrosesan untuk ekstraksi dan analisis fitur-fitur gambar, yang digunakan robot untuk mengambil keputusan. Analisis citra ini terbagi menjadi manipulasi citra untuk perbaikan dan analisis citra untuk pengenalan objek. Penelitian ini bertujuan mengoptimalkan algoritma deteksi objek berbentuk kotak pada TurtleBot menggunakan teknologi computer vision, khususnya dengan menerapkan algoritma YOLO (You Only Look Once) pada kamera Raspberry Pi. Fokus penelitian ini adalah meningkatkan akurasi dan efisiensi deteksi dalam berbagai kondisi pencahayaan. Dengan demikian, sistem ini diharapkan dapat meningkatkan kemampuan TurtleBot dalam navigasi dan interaksi dengan lingkungan sekitar, dengan mendeteksi objek berbentuk kotak secara efisien dan akurat.

II. METODE PENELITIAN

A. Diagram Alir Penelitian

Pada bagian ini ditampilkan langkah – langkah penelitian yang akan dilakukan, tahap penelitian diawali dengan studi literatur, kemudian dilanjutkan dengan tahap simulasi, apabila simulasi sudah sesuai maka dilanjutkan dengan implementasi perancangan hardware dan software. Dalam tahapan penelitian juga dilakukan uji coba sistem dan analisa hasil serta kesimpulan dari hasil uji coba sesuai dengan Gambar 1.

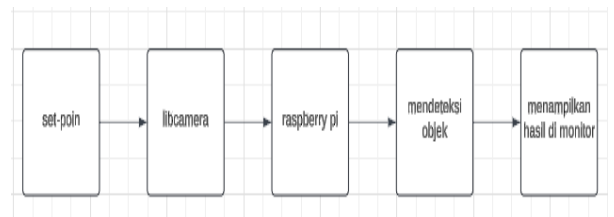


Gambar 1. Diagram Alir Penelitian

Pada Diagram Alir tersebut menunjukkan proses penelitian yaitu :

1. Pada bagian ini ditampilkan langkah–langkah penelitian yang akan dilakukan, tahap penelitian diawali dengan Tahap pertama dalam penelitian ini adalah **pengumpulan data**. Pada tahap ini, peneliti mengumpulkan data yang diperlukan untuk penelitian. Data ini bisa berupa gambar atau video yang mengandung objek kotak yang akan digunakan untuk melatih dan menguji model deteksi objek.
2. Selanjutnya, data yang telah dikumpulkan akan digunakan dalam tahap **pengujian dan evaluasi**. Model deteksi objek yang telah dibangun akan diuji menggunakan data tersebut, dan evaluasi dilakukan untuk menilai kinerja model berdasarkan metrik yang relevan seperti akurasi, presisi, dan recall.
3. Tahap pengujian dan evaluasi ini diikuti oleh **decision point** yang menentukan apakah hasil pengujian dan evaluasi model sudah memenuhi kriteria keberhasilan yang ditetapkan. Jika model belum berhasil atau tidak memenuhi kriteria, proses penelitian kembali ke tahap pengumpulan data untuk melakukan perbaikan atau mengumpulkan data tambahan. Proses ini bersifat iteratif dan memungkinkan penyesuaian dan perbaikan terus-menerus hingga model mencapai kriteria keberhasilan yang diinginkan.
4. Jika model berhasil atau telah memenuhi kriteria keberhasilan, langkah selanjutnya adalah **analisis hasil**. Pada tahap ini, hasil pengujian dan evaluasi dianalisis untuk mendapatkan pemahaman yang lebih mendalam tentang performa model dan implikasinya. Analisis ini membantu peneliti dalam memahami kekuatan dan kelemahan model serta potensi perbaikan di masa depan.
5. Setelah analisis hasil, peneliti masuk ke tahap **penulisan laporan**. Laporan penelitian mencakup semua tahapan, metode, hasil, analisis, dan kesimpulan dari penelitian. Laporan ini menjadi dokumen akhir yang menyajikan seluruh proses dan temuan penelitian secara sistematis dan terstruktur.
6. Akhirnya, proses penelitian mencapai titik **selesai**, menandakan bahwa semua tahapan telah dilalui dan laporan akhir telah disusun. Proses ini memberikan gambaran yang jelas tentang alur kerja penelitian dari awal hingga akhir, dengan siklus iteratif yang memungkinkan peningkatan dan optimasi terus-menerus.

B. Blok Diagram Sistem



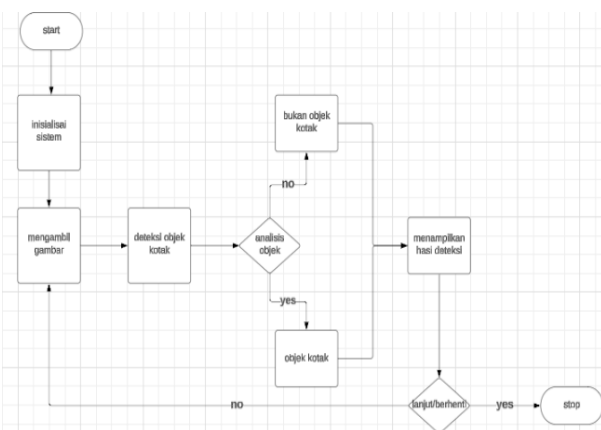
Gambar 2. Blok Diagram Alir Sistem

Blok Diagram Sistem diatas menggambarkan alur proses deteksi objek menggunakan Raspberry Pi dan kamera. Pertama, pengguna menentukan set-point, yaitu pengaturan awal yang mencakup parameter seperti resolusi gambar, fokus kamera, dan objek yang akan dideteksi. Kemudian, kamera yang terhubung ke Raspberry Pi menangkap gambar atau video menggunakan perangkat lunak libcamera, yang berfungsi untuk mengontrol dan mengelola pengambilan gambar dari kamera.

Setelah gambar ditangkap, Raspberry Pi memproses data gambar tersebut dengan menjalankan skrip atau algoritma deteksi objek, misalnya menggunakan model YOLO atau OpenCV. Pada tahap mendeteksi objek, Raspberry Pi menganalisis gambar dan mencari objek yang sesuai dengan parameter yang telah ditentukan sebelumnya.

Terakhir, hasil deteksi tersebut ditampilkan di monitor, sehingga pengguna dapat melihat objek-objek yang telah dikenali dan posisinya dalam gambar atau video yang sedang dianalisis. Dengan demikian, sistem ini memungkinkan pengguna untuk mendeteksi objek secara otomatis dan menampilkan hasilnya secara visual.

C. Flowchart Sistem



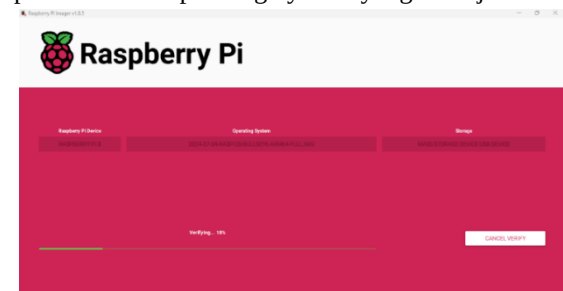
Gambar 3. Flowchart Sistem

Flowchart di atas menunjukkan cara kerja sistem dalam mendeteksi objek. Memulai dengan start pada program yang menjadi parameter untuk inisialisasi sistem untuk memastikan semua komponen siap digunakan dalam mendeteksi objek yang dituju, lalu kamera akan mengambil video untuk mendeteksi menggunakan kamera raspicam. Selanjutnya mulai mendeteksi objek apakah objek tersebut berbentuk kotak atau tidak. Selanjutnya analisis objek, analisis ini dilakukan oleh algoritma yang telah dimasukkan kedalam raspberry pi untuk mengenali apakah objek tersebut kotak atau tidak, hasil deteksi kemudian ditampilkan di layar atau monitor apakah objek yang dideteksi merupakan bentuk kotak atau tidak dan selanjutnya lanjut atau tidak merupakan sebuah keputusan jika ingin berhenti maka sistem akan berhenti mendeteksi namun jika ingin terus mendeteksi maka sistem akan kembali mengambil gambar untuk dideteksi.

D. Perancangan Hardware

1. Install Operating System

Operating system merupakan perangkat utama yang mengelola perangkat keras dan perangkat lunak lain dalam didalam perangkat tersebut. Dengan beberapa kegunaan seperti manajemen perangkat keras raspberry pi, platform aplikasi untuk menyediakan lingkungan untuk aplikasi dapat dijalankan termasuk perangkat lunak untuk pengembangan, multimedia dan lainnya, dan juga sebagai antarmuka pengguna untuk memberikan antarmuka pengguna grafis(GUI) atau antarmuka baris perintah(CLI). Berikut adalah dokumentasi proses install operating system yang ditunjukkan di

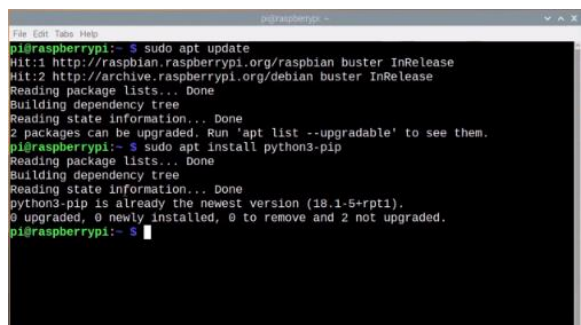


Gambar 4. Install OS

Gambar 4 proses instalasi OS raspberry pi

2. Install Python

Python merupakan bahasa pemrograman tingkat tinggi yang digunakan di raspberry pi dengan beberapa kegunaan pustaka dan modul integrasi untuk mendukung berbagai pustaka dan modul yang penting untuk deteksi objek, seperti opencv untuk pemrosesan gambar, selanjutnya kemudahan pengembang untuk memudahkan pengembangan untuk menguji kode dengan cepat, dan juga kompatibilitas dengan raspberry pi. Berikut foto gambar proses instalasi python yang ditunjukkan oleh Gambar 5

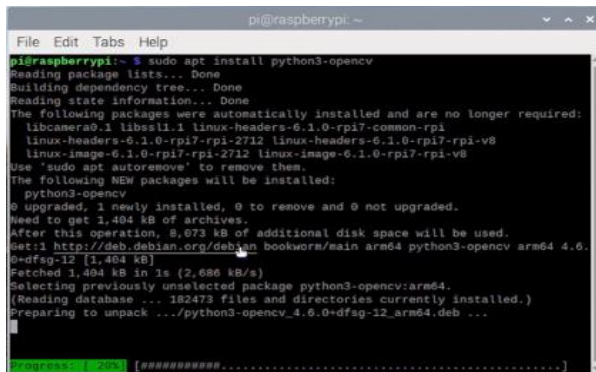


Gambar 5. Install Python

Dengan menggunakan perintah "sudo apt install python3-pip". Pada Perintah sudo apt install python3-pip digunakan untuk menginstall pip, yaitu package manager untuk Python, pada sistem operasi berbasis Debian seperti Ubuntu dan Raspberry Pi OS.

3. Install Opencv

Opencv merupakan pustaka perangkat lunak yang sangat berguna untuk aplikasi visi komputer dan pemrosesan gambar pada raspberry pi dengan beberapa kegunaan yaitu, pengolahan gambar dan video untuk memproses gambar dan video secara real, deteksi dan pelacak objek, pengenalan pola, dan integrasi dengan python yang memungkinkan integrasi yang mudah dengan skrip python. Berikut proses instalasi opencv di raspberry pi yang di tunjukkan Gambar 6.



Gambar 6. Proses Instalasi OpenCV

4. Install Yolo

Yolo merupakan model deteksi objek berbasis deep learning yang efektif dan cepat untuk deteksi objek secara real-time yaitu untuk mendeteksi objek dalam waktu nyata dengan kecepatan tinggi, akurasi tinggi dengan kemampuan untuk men deteksi berbagai jenis objek dalam gambar dan video, serta kemudahan pengguna yang mudah digunakan dengan dukungan untuk pelatihan dan inferensi menggunakan pytorch, dan juga komunitas yang aktif.

III. HASIL DAN PEMBAHASAN

A. Hasil Pengujian Deteksi Kamera Raspberry-Pi Dengan Berbagai Jarak (cm)

Pengujian ini bertujuan untuk mengukur jarak deteksi dan optimasi deteksi pada kamera TurtleBot dengan menggunakan metode YOLOv5. Pengujian dilakukan dengan variasi jarak pengukuran mulai dari 20 cm, 30 cm, 40 cm, 50 cm, 60 cm, hingga 70 cm.

Tabel 1. Pengujian Deteksi Kamera Raspberry-Pi Dengan Berbagai Jarak

No	Objek	Jarak (cm)	Hasil	
			Terdeteksi	Tingkat kepercayaan
1		20	Ya	0.86
		30	Ya	0.81
		40	Ya	0.70
		50	Ya	0.82

	Balok hijau	60	Ya	0.85
		70	Ya	0.88
2.	 Balok biru	20	Ya	0.78
		30	Ya	0.79
		40	Ya	0.69
		50	Ya	0.50
		60	Ya	0.49
		70	Ya	0.48
3.	 Balok kayu	20	Ya	0.87
		30	Ya	0.81
		40	Ya	0.78
		50	Ya	0.81
		60	Ya	0.82
		70	Ya	0.80
4.	 kartus	20	Ya	0.79
		30	Ya	0.79
		40	Ya	0.78
		50	Ya	0.77
		60	Ya	0.80
		70	Ya	0.74
5.	 Handphone	20	Ya	0.77
		30	Ya	0.80
		40	Ya	0.79
		50	Ya	0.80
		60	Ya	0.79
		70	Ya	0.75
6.	 Botol	20	Ya	0.78
		30	Ya	0.81
		40	Ya	0.79
		50	Ya	0.82
		60	Ya	0.85
		70	Ya	0.86

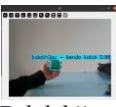
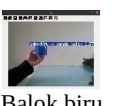
Tabel tersebut menunjukkan hasil uji deteksi objek pada berbagai jarak, yang meliputi informasi tentang keberhasilan deteksi serta tingkat kepercayaan (confidence score) dari sistem. Setiap objek diuji dari jarak 20 hingga 70 meter untuk mengevaluasi konsistensi dan akurasi deteksi pada berbagai rentang. Pada objek **balok hijau**, sistem berhasil mendeteksi objek ini pada setiap jarak dengan tingkat kepercayaan yang cukup tinggi dan relatif stabil, berkisar antara 0.70 hingga 0.88. Menariknya, tingkat kepercayaan cenderung meningkat pada jarak yang lebih jauh, mencapai puncaknya di 70 meter dengan skor 0.88. Hal ini menunjukkan bahwa sistem memiliki performa deteksi yang baik untuk objek ini, bahkan pada jarak yang lebih jauh. Sementara itu, **balok biru** terdeteksi dengan baik pada awalnya, namun tingkat kepercayaan mengalami penurunan seiring dengan bertambahnya jarak. Pada jarak 20 meter, tingkat kepercayaan tercatat 0.78, tetapi menurun hingga hanya 0.48 pada 70 meter. Penurunan signifikan ini mungkin menunjukkan bahwa sistem mengalami kesulitan dalam mempertahankan keakuratan deteksi pada balok biru, terutama pada jarak yang lebih jauh, mungkin karena faktor warna atau karakteristik visual lainnya yang mempengaruhi deteksi. Pada **balok kayu**, sistem menunjukkan tingkat kepercayaan yang konsisten dan tinggi di semua jarak, berkisar antara 0.78 hingga 0.87. Dengan hasil yang stabil ini, tampak bahwa sistem mampu mendeteksi

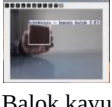
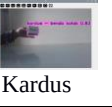
objek dengan karakteristik kayu ini tanpa penurunan kepercayaan yang berarti meskipun jarak berubah. Objek **kardus** juga berhasil terdeteksi di seluruh rentang jarak yang diuji, dengan tingkat kepercayaan yang cukup stabil antara 0.74 hingga 0.80. Meskipun terdapat sedikit penurunan pada jarak yang lebih jauh, tingkat kepercayaan tetap dalam batas yang layak, menunjukkan performa deteksi yang cukup baik untuk objek ini. Pada pengujian **handphone**, deteksi menunjukkan hasil yang cukup stabil dengan tingkat kepercayaan berkisar antara 0.75 hingga 0.80 di semua jarak. Hal ini menunjukkan bahwa sistem memiliki performa yang baik dalam mengenali objek handphone, tanpa adanya perubahan signifikan pada tingkat kepercayaan saat jarak bertambah. Objek terakhir, yaitu **botol**, menunjukkan hasil deteksi yang sangat baik dengan tingkat kepercayaan yang cenderung meningkat seiring dengan bertambahnya jarak. Pada 20 meter, tingkat kepercayaan tercatat 0.78, yang kemudian meningkat hingga 0.86 pada 70 meter. Hal ini mengindikasikan bahwa sistem sangat mampu mendeteksi objek ini bahkan pada jarak jauh, dan faktor jarak justru tampaknya mendukung performa deteksi untuk objek botol. Secara keseluruhan, hasil uji menunjukkan bahwa sistem berhasil mendeteksi semua objek di berbagai jarak yang diuji. Namun, setiap objek tampak memiliki pola tingkat kepercayaan yang berbeda. Objek seperti balok hijau dan botol menunjukkan peningkatan kepercayaan pada jarak lebih jauh, sementara balok biru mengalami penurunan. Perbedaan ini mungkin disebabkan oleh karakteristik visual objek, seperti warna atau bahan, yang mempengaruhi kemampuan sistem dalam mempertahankan keakuratan deteksi pada berbagai jarak.

B. Hasil Pengujian Deteksi Kamera Raspberry-Pi Dengan Berbagai Kondisi Pencahayaan

Pengujian ini bertujuan untuk mengukur pendeteksian dan optimasi deteksi pada kamera turtlebot dengan kondisi pencahayaan yang berbeda mulai dari gelap, redup, dan terang. Menurut IESNA (Illuminating Engineering Society of North America) membagi kategori pencahayaan sebagai berikut yaitu: gelap (0-50 lux), redup (50-200 lux) terang (200-500).

Tabel 2. Pengujian Deteksi Objek Kamera Raspberry-Pi Dengan Berbagai Kondisi Pencahayaan

No	Objek	Cahaya (lumen)	Hasil	
			Terdeteksi	Tingkat kepercayaan
1	 Balok hijau	0-50	Ya	0.80
		50-200	Ya	0.79
		200-500	Ya	0.70
2.	 Balok biru	0-50	Ya	0.65
		50-200	Ya	0.66
		200-500	Ya	0.56

3.	 Balok kayu	0-50	Ya	0.79
		50-200	Ya	0.81
		200-500	Ya	0.78
4.	 Kardus	0-50	Ya	0.79
		50-200	Ya	0.81
		200-500	Ya	0.75
5.	 Handphone	0-50	Ya	0.77
		50-200	Ya	0.80
		200-500	Ya	0.60
6.	 Botol	0-50	Ya	0.69
		50-200	Ya	0.80
		200-500	Ya	0.60

Tabel ini menyajikan hasil uji deteksi objek dengan variasi intensitas cahaya (dalam satuan lumen), yang bertujuan untuk mengetahui bagaimana sistem deteksi objek beradaptasi dengan perubahan tingkat pencahayaan. Setiap objek diuji dalam tiga rentang pencahayaan: 0-50 lumen (cahaya rendah), 50-200 lumen (cahaya sedang), dan 200-500 lumen (cahaya tinggi). Selain menunjukkan apakah objek berhasil terdeteksi, tabel ini juga menyertakan tingkat kepercayaan (confidence score) dari sistem deteksi pada setiap kondisi cahaya. Pada objek **balok hijau**, sistem berhasil mendeteksi objek ini di semua rentang pencahayaan dengan tingkat kepercayaan yang bervariasi antara 0.70 hingga 0.80. Terlihat bahwa dalam kondisi pencahayaan rendah (0-50 lumen), tingkat kepercayaan mencapai 0.80, tetapi mengalami sedikit penurunan menjadi 0.70 pada pencahayaan tinggi (200-500 lumen). Penurunan ini mengindikasikan bahwa sistem deteksi mungkin lebih efektif pada pencahayaan rendah hingga sedang untuk objek balok hijau, sementara pencahayaan yang lebih terang justru sedikit mengurangi keakuratannya. **Balok biru** juga terdeteksi pada semua tingkat pencahayaan, namun dengan tingkat kepercayaan yang lebih rendah dibandingkan balok hijau. Pada pencahayaan rendah, tingkat kepercayaan berada di 0.65 dan tidak menunjukkan peningkatan signifikan dalam kondisi pencahayaan sedang (50-200 lumen). Pada pencahayaan tinggi, tingkat kepercayaan menurun menjadi 0.56. Ini menunjukkan bahwa objek balok biru mungkin memiliki karakteristik visual yang sulit dideteksi secara akurat, terutama pada pencahayaan tinggi, yang menyebabkan penurunan tingkat kepercayaan. Pada objek **balok kayu**, sistem menunjukkan hasil yang cukup stabil di semua kondisi cahaya. Tingkat kepercayaan tertinggi tercatat pada pencahayaan sedang (0.81) dan hanya sedikit menurun pada pencahayaan tinggi menjadi 0.78. Stabilitas ini menunjukkan bahwa sistem dapat mendeteksi objek balok kayu secara konsisten tanpa banyak terpengaruh oleh perubahan pencahayaan. **Kardus** juga menunjukkan pola yang mirip dengan balok kayu. Tingkat kepercayaan relatif stabil di semua tingkat pencahayaan, berada di kisaran 0.75 hingga 0.81. Sistem deteksi tampaknya bekerja

dengan baik pada objek kardus, dengan hanya sedikit penurunan tingkat kepercayaan di pencahayaan tinggi. Hal ini menandakan bahwa objek kardus dapat dikenali secara konsisten di berbagai tingkat pencahayaan. Pada objek **handphone**, tingkat kepercayaan cukup baik pada pencahayaan rendah dan sedang, mencapai 0.77 dan 0.80. Namun, pada pencahayaan tinggi, tingkat kepercayaan mengalami penurunan signifikan hingga 0.60. Ini mengindikasikan bahwa deteksi handphone menjadi kurang akurat dalam kondisi pencahayaan yang lebih terang, yang mungkin disebabkan oleh pantulan cahaya atau fitur visual lainnya yang kurang optimal pada cahaya tinggi. Objek terakhir, yaitu **botol**, memiliki tingkat kepercayaan yang meningkat dari 0.69 pada pencahayaan rendah ke 0.80 pada pencahayaan sedang, tetapi kemudian menurun tajam ke 0.60 pada pencahayaan tinggi. Hasil ini menunjukkan bahwa botol cenderung lebih mudah dikenali dalam pencahayaan sedang, sedangkan pencahayaan tinggi mungkin memperkenalkan hambatan pada sistem deteksi, seperti pantulan atau perubahan kontras yang mempersulit identifikasi bentuk objek. Secara keseluruhan, sistem deteksi mampu mengenali semua objek pada berbagai tingkat pencahayaan. Namun, setiap objek menunjukkan respons yang berbeda terhadap perubahan intensitas cahaya. Beberapa objek, seperti balok kayu dan kardus, menunjukkan stabilitas yang baik di berbagai kondisi cahaya, sementara objek lain, seperti balok biru dan handphone, mengalami penurunan kepercayaan pada cahaya tinggi. Perbedaan ini mungkin disebabkan oleh karakteristik visual dan reflektifitas masing-masing objek yang mempengaruhi kemampuan deteksi dalam kondisi pencahayaan yang beragam.

IV. KESIMPULAN

- Implementasi Pengenalan Objek Berbentuk Kotak: Penelitian ini berhasil mengimplementasikan deteksi objek berbentuk kotak pada TurtleBot dengan modul kamera Raspberry Pi. Tahapan deteksi meliputi akuisisi gambar, pemrosesan citra, dan penerapan algoritma YOLO. Pengujian dilakukan pada tiga kondisi pencahayaan: gelap (0-50 lumen), redup (50-200 lumen), dan terang (200-500 lumen), dengan jarak pengamatan 20-70 cm. Hasil menunjukkan deteksi optimal pada pencahayaan terang karena kamera dapat menangkap detail objek dengan jelas, memungkinkan algoritma YOLO bekerja lebih cepat dan akurat. Pada pencahayaan redup, keakuratan menurun akibat gambar yang kurang tajam, yang mempersulit identifikasi bentuk kotak. Dalam kondisi gelap, deteksi hampir tidak berhasil karena minimnya cahaya yang menghambat proses pengambilan detail objek. Hal ini menegaskan bahwa pencahayaan cukup sangat penting untuk hasil deteksi yang efektif.
- Peningkatan Akurasi Deteksi Objek Berbentuk Kotak melalui Modul Kamera: Penelitian ini menunjukkan bahwa optimalisasi modul kamera Raspberry Pi pada TurtleBot dapat meningkatkan akurasi deteksi objek kotak, terutama pada pencahayaan terang. Pada jarak 20-50 cm dengan pencahayaan terang, tingkat kepercayaan deteksi melebihi 0,80. Namun, pada jarak 60-70 cm, akurasi menurun dengan tingkat kepercayaan serendah 0,71. Pencahayaan yang cukup mendukung kamera dalam menghasilkan deteksi yang optimal, sementara pada kondisi redup atau gelap, akurasi berkurang signifikan dan deteksi sering gagal. Untuk mengatasi hal ini, penyesuaian parameter kamera seperti eksposur dan sensitivitas cahaya dapat dilakukan untuk meningkatkan kinerja pada berbagai kondisi pencahayaan.

DAFTAR PUSTAKA

- [1] M. I. Moha, C. Poekoel, M. Eldad, I. Najoran, and R. F. Robot, "Implementasi Kamera 360 Derajat Untuk Mendeteksi Objek Pada Robot Sepak Bola Beroda," *Jurnal Teknik Informatika*, vol. 14, no. 3, 2019.
- [2] D. A. Prabowo, D. Abdullah, and A. Manik, "DETEKSI DAN PERHITUNGAN OBJEK BERDASARKAN WARNA MENGGUNAKAN COLOR OBJECT TRACKING," 2018.
- [3] Muliady, Y. Louise, dan Y. Susanthi, "Mapping dan Navigasi untuk Robot Pengantar Makanan di Restoran Berbasis ROS," *Techné Jurnal Ilmiah Elektroteknika*, vol. 22, no. 1, hal. 111-128, Apr. 2023.
- [4] J. Mahestra, R. Aprilino, and P. Santoso, "PENERAPAN SISTEM NAVIGASI MOBILE ROBOT PADA TURTLEBOT DENGAN DEPTH SENSOR CAMERA," *Seminar Nasional Fortei Regional*, vol. 7, 2020
- [5] M. Abrar Masril and D. Putra Caniago, "Optimasi Teknologi Computer Vision pada Robot Industri Sebagai Pemindah Objek Berdasarkan Warna," vol. 11, no. 1, pp. 46-57, 2023.
- [6] A. Rahman, "Penerapan SLAM Gmapping dengan Robot Operating System Menggunakan Laser Scanner pada Turtlebot," *J. Rekayasa Elektrika*, vol. 16, no. 2, hal. 103-109, Agustus 2020.
- [7] Rasyid, Rizky Abdillah. "Penghindaran Rintangan untuk Robot TurtleBot3 dengan Menggunakan Algoritma Greedy dan Sensor Laser Scan." Makalah dalam IF2211 Strategi Algoritma, Program Studi Teknik Informatika, Sekolah Teknik Elektro dan Informatika, Institut Teknologi Bandung, 2022.
- [8] M. Rizqi Efrian et al., "IMAGE RECOGNITION BERBASIS CONVOLUTIONAL NEURAL NETWORK (CNN) UNTUK MENDETEKSI PENYAKIT KULIT PADA MANUSIA," *Jurnal POLEKTRO: Jurnal Power Elektronik*, vol. 11, no. 1, p. 2022.
- [9] A. Dewi Parwati and A. Akbar Riadi, "Implementation Of Raspberry Pi As a Wireless Camera Design," vol. 2, no. 2, pp.77 – 81, 2022, Available: <http://journal.stiestekom.ac.id/index.php/TEKNIK>
- [10] J. Zophie, H. Himawan Triharminto, D. Elekrnika, and A. Angkatan Udara, "Implementasi Algoritma You Only Look Once (YOLO) menggunakan Web Camera untuk Mendeteksi Objek Statis dan Dinamis Implementation of You Only Look Once (YOLO) Algorithm using Web Camera for Static dan Dinamic Object Detection," vol. 1, no. 1, 2022.
- [11] G. Xie and W. Lu, "Image Edge Detection Based On Opencv," *International Journal of Electronics and Electrical Engineering*, vol. 1, no. 2, pp. 104-106, 2013, doi: 10.12720/ijeee.1.2.104-106.

- [12] A. Paszke et al., "Automatic differentiation in PyTorch."
- [13] A. Khosravi, F. Albardi, H. M. D. Kabir, M. M. I. Bhuiyan, P. M. Kebria, and S. Nahavandi, "A Comprehensive Study on Torchvision Pre-trained Models for Fine-grained Inter-species Classification," Oct. 2021,
- [14] C. Darujati, T. Susim, and I. Artikel, "PENGOLAHAN CITRA UNTUK PENGENALAN WAJAH (FACE RECOGNITION) MENGGUNAKAN OPENCV," Jurnal Syntax Admiration, vol. 2, no. 3, 2021.