

# ANALISIS OBJEK MANUSIA MENGGUNAKAN ALGORITMA YOLOv5

**Muh.Yusuf<sup>1\*</sup>, Muhammad Basri<sup>2</sup>, Untung Suwardoyo<sup>3</sup>**

<sup>1\*,3</sup> Program Studi Teknik Informatika, Universitas Muhammadiyah Parepare, Indonesia

<sup>2</sup> Program Studi Teknik Elektro, Universitas Muhammadiyah Parepare, Indonesia

\*Email : [muhammadyusufsapnur@gmail.com](mailto:muhammadyusufsapnur@gmail.com)

**Abstrak:** Menghitung jumlah individu yang keluar dan masuk di area dengan tingkat keramaian tinggi merupakan tantangan yang memerlukan akurasi tinggi. Untuk mengatasi masalah ini, diperlukan sistem *computer vision* yang mampu menganalisis objek manusia secara *real-time*. Penelitian ini bertujuan untuk mengembangkan sistem berbasis algoritma YOLOv5 yang dapat mendeteksi, melacak, dan mencatat pergerakan orang yang masuk dan keluar, serta diintegrasikan dengan MySQL untuk mendukung pengelolaan data dan pelaporan. Metode penelitian menggunakan *System Development Methodology* (SDM), yang mencakup tahapan Perancangan Sistem, Detail Sistem, serta pengujian dan evaluasi sistem. Penelitian ini dibuat menggunakan bahasa pemrograman *Python* serta didukung oleh *Roboflow*, *Google Colaboratory* dan model YOLOv5. Hasil pembuatan aplikasi menunjukkan bahwa sistem deteksi objek berbasis YOLOv5 bekerja efektif untuk menghitung jumlah orang secara *real-time* dan mengintegrasikan ke MySQL. Sistem ini mencapai akurasi rata-rata 64,52%, meskipun masih terdapat kelemahan berupa kesalahan deteksi (*false negatives*) akibat pencahayaan yang kurang optimal dan jarak kamera yang tidak ideal.

**Kata kunci :** *Computer Vision*, Analisis Objek Manusia, YOLOv5, Penghitungan *Real-time*.

## 1. PENDAHULUAN

Perkembangan teknologi *computer vision* dalam beberapa tahun terakhir telah memberikan dampak signifikan pada berbagai bidang, termasuk pengawasan dan analisis pergerakan objek di lingkungan yang dinamis. Menurut (Yasin & Suryanegara., 2022) Teknologi ini memanfaatkan algoritma dan model pembelajaran mesin untuk mengenali dan memproses informasi visual secara otomatis, yang memungkinkan analisis gambar atau video secara cepat dan akurat tanpa pengamatan manual. Menurut (Sutisna et al., 2023) Penggunaan *computer vision* dalam pengelolaan keramaian sangat penting, karena teknologi ini mampu memberikan pemantauan *real-time* pada area yang penuh sesak, seperti di stasiun, bandara, atau pusat perbelanjaan.

Teknologi ini juga dapat digunakan untuk meningkatkan pengelolaan keamanan publik dengan mendeteksi keramaian yang berpotensi menyebabkan masalah keamanan. Dengan demikian, pengelola dapat segera mengambil langkah-langkah pencegahan, seperti mengalihkan jalur pergerakan atau meningkatkan pengawasan, sebagaimana yang dinyatakan oleh (Le et al., 2019). Pengelolaan keramaian yang lebih baik memungkinkan respons yang lebih cepat terhadap masalah potensial dan memperbaiki pemantauan dalam situasi dengan keramaian tinggi, menurut (Wang & Liu, 2020).

Analisis objek manusia merupakan komponen penting dalam teknologi *computer vision*, yang melibatkan deteksi, pelacakan, dan pengenalan objek manusia dalam video atau gambar. Menurut (Li et al., 2020) Dengan analisis objek manusia, sistem dapat memberikan informasi lebih mendalam mengenai pergerakan manusia, seperti pola pergerakan, kecepatan, dan distribusi objek di area tertentu. Sebagaimana dinyatakan oleh (Aksoy & Gokberk., 2021) Analisis ini memungkinkan pemantauan yang lebih efektif dan lebih terstruktur dalam situasi keramaian yang dinamis.

Selain itu, analisis objek manusia juga sangat berguna dalam menghitung jumlah orang yang berada dalam suatu area pada waktu tertentu. Data ini dapat membantu mengidentifikasi potensi masalah dalam keramaian, seperti kemacetan atau risiko kecelakaan. Menurut (Gao et al., 2021) Hal ini juga dapat merencanakan aliran manusia dan meningkatkan efisiensi pergerakan di area keramaian. Dalam penelitian (Winata et al, 2020) analisis objek manusia terbukti dapat membantu meningkatkan deteksi interaksi dalam keramaian.

Algoritma YOLOv5 (*You Only Look Once version 5*) merupakan salah satu algoritma deteksi objek terbaru yang sangat efisien dalam hal kecepatan dan akurasi. YOLOv5 bekerja dengan prinsip mendeteksi objek dalam satu proses pencarian gambar, sehingga dapat mendeteksi objek secara langsung tanpa perlu pencarian berulang. Keuntungan utama YOLOv5 adalah kecepatannya dalam memproses gambar dan efisiensinya dalam situasi nyata yang dinamis, sebagaimana dinyatakan oleh (Redmon et al., 2016). Dalam konteks keramaian, algoritma ini dapat menangani kepadatan tinggi dan mendeteksi objek manusia dalam waktu singkat, berdasarkan pernyataan (Bojarski et al., 2020).

YOLOv5 diterapkan untuk mendeteksi objek manusia secara efisien dalam situasi keramaian. Kemampuannya dalam mengidentifikasi individu dengan kecepatan tinggi sangat berguna untuk pemantauan langsung, terutama dalam kondisi keramaian yang dinamis (Khairunnas et al., 2021). Selain itu, YOLOv5 juga telah terbukti efektif dalam meningkatkan akurasi deteksi objek di lingkungan dengan kepadatan tinggi, seperti yang ditunjukkan oleh (Setiawan et al., 2020), yang menggunakan YOLOv5 untuk memantau kerumunan di berbagai tempat umum. Keefektifan YOLOv5 dalam situasi ini menjadikannya pilihan yang tepat untuk pengelolaan keramaian di Indonesia.

Penghitungan *real-time* sangat penting dalam pengelolaan keramaian di area publik yang padat. Teknologi *computer vision* memungkinkan penghitungan objek, seperti manusia, secara langsung dan tanpa penundaan, sehingga pengelola dapat segera mengidentifikasi area yang terlalu padat atau membutuhkan perhatian khusus. Misalnya, penelitian oleh (Mokayed et al., 2020), mengembangkan sistem deteksi dan penghitungan manusia secara *real-time* menggunakan teknik *deep learning*, yang menunjukkan efektivitas dalam manajemen keramaian. Selain itu, penelitian oleh (Gündüz et al., 2020), mengusulkan metode berbasis YOLOv5 untuk deteksi kerumunan

secara *real-time* dari video, yang menunjukkan kinerja yang baik dalam mendeteksi dan menghitung orang dalam waktu nyata.

Penelitian terdahulu menunjukkan penerapan algoritma YOLOv5 dalam berbagai bidang. (Agarwal & Gupta., 2023) menerapkan YOLOv5 untuk deteksi manusia dalam video *streaming* dengan hasil akurasi tinggi meskipun dalam kondisi lingkungan yang kompleks. Penelitian (Ashar., 2022) di Indonesia mengimplementasikan YOLOv5 untuk deteksi penggunaan masker di kantor. (Sari et al, 2023) mengembangkan aplikasi pendeteksi warna berbasis *Python* dan MySQL untuk analisis data efisien.

Berdasarkan pembahasan di atas, dapat disimpulkan bahwa teknologi *computer vision* dan algoritma YOLOv5 berperan penting dalam meningkatkan efisiensi dan akurasi dalam deteksi objek manusia, khususnya dalam pengelolaan keramaian. Penerapan kedua teknologi ini memungkinkan pemantauan *real-time* yang lebih cepat dan akurat, serta memberikan solusi yang efektif dalam mengidentifikasi dan menganalisis pergerakan manusia di area yang ramai. Dengan demikian, penelitian ini memberikan kontribusi signifikan terhadap pengembangan teknologi deteksi objek, yang dapat diterapkan untuk meningkatkan pengelolaan keramaian dan memastikan keamanan serta kenyamanan di tempat-tempat umum.

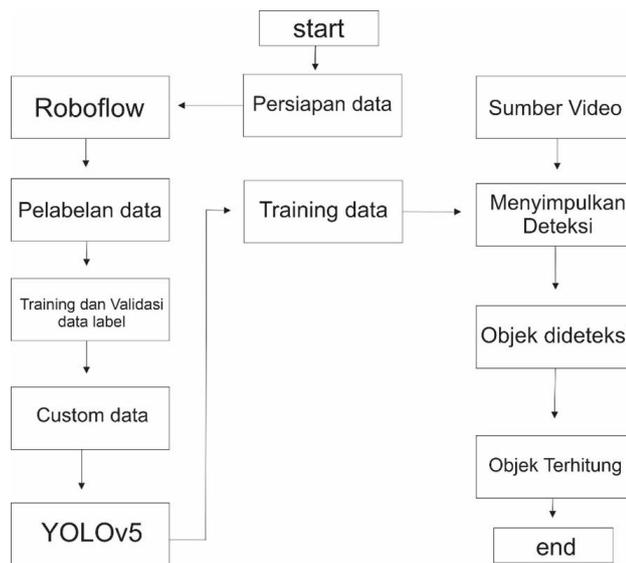
## **2. METODOLOGI PENELITIAN**

Penelitian ini menggunakan metode *System Development Methodology* (SDM), yang melibatkan tahapan Perancangan Sistem, Detail Aplikasi, serta pengujian dan evaluasi sistem. Data dikumpulkan melalui analisis kebutuhan sistem dan studi literatur, dengan pengembangan dilakukan menggunakan bahasa pemrograman *Python*, serta didukung oleh *Roboflow*, *Google Colaboratory* dan model YOLOv5. Pengujian sistem dilakukan selama tiga bulan di Kota Parepare pada tahun 2024. Sistem yang dikembangkan diuji menggunakan video *real-time* yang diambil dari lokasi lapangan untuk mendeteksi pergerakan objek manusia, dan hasil deteksi tersebut disimpan dalam database MySQL untuk dianalisis lebih lanjut.

### **2.1 Perancangan Sistem**

Perancangan sistem implementasi algoritma YOLOv5 pada perhitungan keluar masuk manusia ini meliputi perancangan perangkat keras dan perancangan perangkat lunak. Berikut blok diagram dan sistem yang akan di buat:

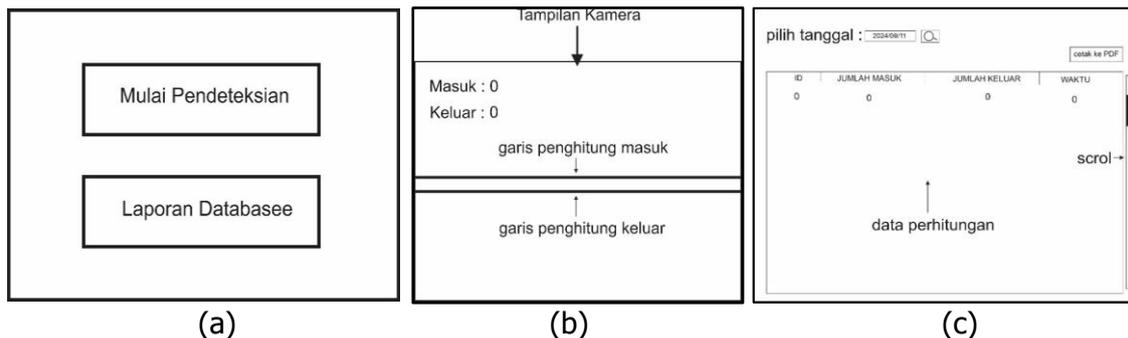
- a. Block Diagram Sistem



**Gambar 1.** Block Diagram Sistem

Gambar 1. Langkah pertama adalah menyiapkan data berupa foto. Data kemudian diolah oleh *Roboflow*, yang memungkinkan bagi kita untuk membagikan data sekaligus pelabelan dan training. Setelah pelabelan dan *training*, data berikutnya yang kita perlukan untuk membuat custom data dan menerapkan algoritma YOLOv5 pada data itu. Kemudian, data yang benar-benar harus dilatih kembali secara menyeluruh agar model bertindak optimal. Ketika sumber video siap, deteksi dapat dilakukan. Karena objek sebelumnya telah melalui pelabelan dan validasi, ketika dideteksi oleh video, objek akan dihitung ketika ia menyentuh garis yang telah kita tetapkan sebelumnya dalam algoritma YOLOv5.

b. Desain Interface



**Gambar 2.** (a) Tampilan Awal, (b) Tampilan Pendeteksian, (c) Tampilan Laporan Database

Gambar 2. (a), Pengguna diberikan dua pilihan menu, yaitu Mulai Pendeteksian, untuk membuka tampilan pendeteksian secara *realtime*, dan Laporan Database, untuk membuka tampilan data yang telah tersimpan dalam *database*. Pada gambar (b), tampilan kamera *realtime* dilengkapi dengan dua perhitungan, yaitu jumlah keluar dan

masuk, dengan garis yang berfungsi untuk mendeteksi serta menghitung objek yang melintas keluar dan masuk. Pada gambar (c) di atas, ditampilkan data jumlah objek keluar dan masuk, dilengkapi dengan fitur pencarian berdasarkan tanggal, serta menampilkan data yang terhitung secara harian, bulanan, dan tahunan. Selain itu, tersedia menu untuk mencetak laporan dalam format PDF.

## 2. 2 Teknik Pengumpulan Data

Pengumpulan data dilakukan dengan mengumpulkan informasi melalui pencarian buku dan jurnal untuk mendapatkan informasi yang berhubungan dengan pelatihan *dataset* dan pelatihan model YOLOv5. pengumpulan data secara langsung dilakukan dengan observasi Lokasi penelitian.

## 2. 3 Teknik Pengujian

Pengujian aplikasi dilakukan menggunakan 2 teknik pengujian yaitu *white box* dan *black box*.

- a. *Whitebox Testing* adalah metode pengujian struktural di mana perangkat lunak transparan bagi penguji. Pengujian ini dirancang berdasarkan struktur kode untuk menyusun *testcase* yang memprediksi cara kerja perangkat lunak, mencakup logika, kondisi, dan perulangan.(Arfah & Suwardoyo, 2022).
- b. *Blackbox Testing* bertujuan untuk menguji seluruh fungsi dalam aplikasi tanpa melihat struktur internalnya. Metode ini mengevaluasi keluaran berdasarkan data atau kondisi masukan yang diberikan, tanpa memperhatikan bagaimana proses di dalam perangkat lunak menghasilkan keluaran tersebut. (Basri & Gushari, 2021).

## 2. 4 Alat dan Bahan Penelitian

Alat dan bahan yang digunakan selama penelitian adalah sebagai berikut:

- a. Laptop Asus Vivobook

**Tabel 1.** Spesifikasi Laptop

No.	Nama	Spesifikasi
1	Processor	AMD Ryzen 5 5600HS with Radeon Graphics (16 CPUs), ~3.2GHz
2	RAM	RAM 24 GB
3	SSD	512 GB
4	LCD Monitor	LCD Monitor : 14"

- b. Software yang digunakan dalam pembuatan aplikasi ini adalah:

**Tabel 2.** *Software* yang digunakan

No.	Nama	Keterangan
1	Sistem Operasi	Windows 11

2	Model	YOLOv5
3	Bahasa pemrograman	Python
4	Tools	Roboflow, Google Collaborate, Visual Studio Code

### 3. HASIL DAN PEMBAHASAN

#### 3.1. Hasil

**Tabel 3.** Perhitungan Tingkat Keberhasilan

Tanggal/waktu	Kategori	Jumlah manual	Jumlah sistem	Tingkat keberhasilan (%)
05-11-2024 jam 10	Jumlah masuk	40	27	$\frac{27}{40} \times 100\% = 67,5\%$
05-11-2024 jam 10	Jumlah keluar	35	27	$\frac{27}{35} \times 100\% = 77,14\%$
05-11-2024 jam 11	Jumlah masuk	29	26	$\frac{26}{29} \times 100\% = 89,66\%$
05-11-2024 jam 11	Jumlah keluar	61	52	$\frac{52}{61} \times 100\% = 82,25\%$
05-11-2024 jam 13	Jumlah masuk	2	0	$\frac{0}{2} \times 100\% = 0,0\%$
05-11-2024 jam 13	Jumlah keluar	13	8	$\frac{8}{13} \times 100\% = 61,54\%$
06-11-2024 jam 11	Jumlah masuk	46	33	$\frac{33}{46} \times 100\% = 71,74\%$
06-11-2024 jam 11	Jumlah keluar	60	38	$\frac{38}{60} \times 100\% = 63,33\%$

$$\text{Tingkat keberhasilan} = \frac{\text{Total tingkat keberhasilan semua percobaan}}{\text{jumlah percobaan}}$$

Total Tingkat keberhasilan semua percobaan:

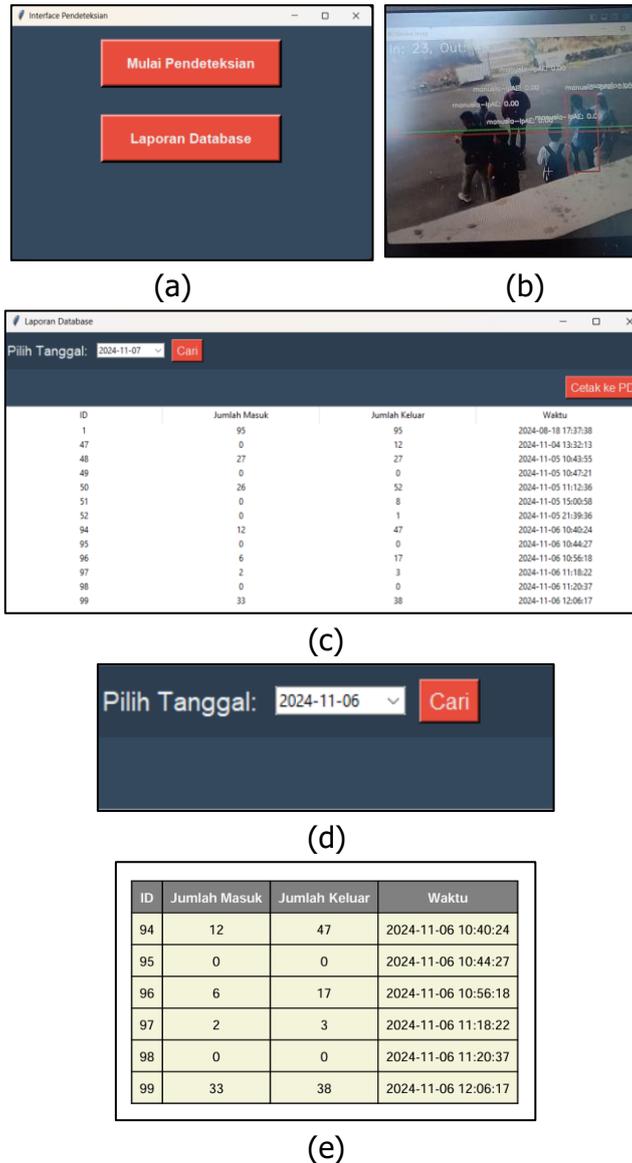
$$67,5\% + 77,14\% + 89,66\% + 82,25\% + 0,0\% + 61,54\% + 71,74\% + 63,33\% = 516,16\%$$

Jumlah percobaan: Terdapat 8 total percobaan.

$$\text{Tingkat keberhasilan} = \frac{516,16\%}{8} = \mathbf{64,52\%}$$

Hasil pengujian menunjukkan bahwa sistem deteksi berbasis algoritma YOLOv5 memiliki akurasi 64,52%. Meskipun masih terdapat potensi kesalahan *false positives* dan *false negatives*, sistem tetap berfungsi optimal dalam mendeteksi serta menghitung manusia yang keluar dan masuk. Faktor seperti pencahayaan yang tidak stabil dan sudut pandang kamera mempengaruhi akurasi deteksi. Dengan peningkatan ambang batas *confidence score* ke 70-80% serta optimalisasi lingkungan, sistem berpotensi meningkatkan keandalan dan akurasi dalam perhitungan.

### 3.2. Detail Sistem



**Gambar 3.** (a)Tampilan Utama (b) Tampilan dendeteksi dan perhitungan (c) Tampilan laporan *database* (d) Tampilan pencarian data sesuai tanggal (e) Hasil cetak ke format pdf.

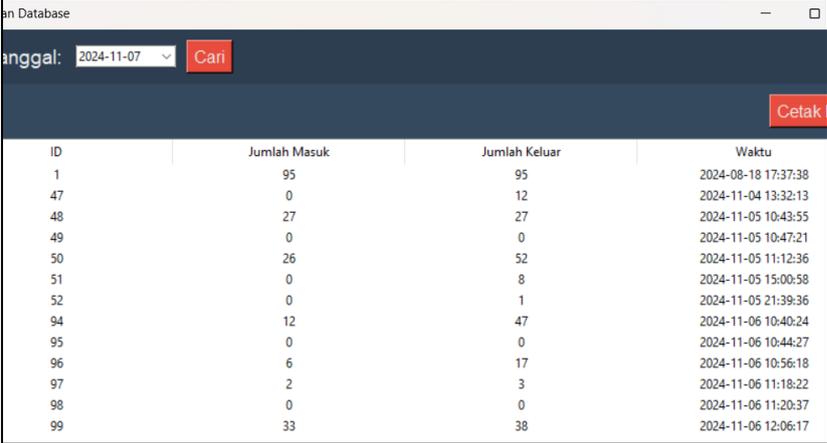
Gambar 3. (a), Tampilan ini merupakan tampilan awal aplikasi saat pertama kali dibuka, dengan dua pilihan menu utama, yaitu Mulai Pendeteksian dan Laporan *Database*, yang masing-masing memiliki fungsi tersendiri. Gambar 3 (b), halaman ini menampilkan proses pendeteksi dan perhitungan secara *realtime*, dengan dua garis: garis *in* (masuk) berwarna hijau dan garis *out* (keluar) berwarna merah. Di sudut kiri layar juga terdapat informasi jumlah objek yang masuk dan keluar. Gambar 3 (c), halaman ini merupakan tampilan hasil perhitungan objek. Pada bagian bawah halaman terdapat menu pencarian untuk mencari data berdasarkan tanggal yang diinginkan, serta opsi

untuk mencetak data ke format PDF. Gambar 3 (d), halaman ini menampilkan tampilan khusus saat pengguna ingin mencari data berdasarkan tanggal tertentu. Sistem akan menampilkan data yang sesuai dengan tanggal yang dimasukkan. Gambar 3 (e), merupakan tampilan hasil data yang telah dicetak ke dalam format PDF.

### 3.3. Pengujian Sistem

#### a. Pengujian *BlackBox Testing*

**Tabel 4.** Pengujian *Black Box*

No	Tes Faktor	Hasil	Keterangan
1	User memilih menu Mulai Pendeteksian	✓	Berhasil menampilkan halaman Pendeteksian
<b>Screenshot</b>			
			
No	Tes Faktor	Hasil	Keterangan
2	User Menekan menu Laporan Database	✓	Berhasil menampilkan halaman Laporan Database
<b>Screenshot</b>			
			
No	Tes Faktor	Hasil	Keterangan
3	User memilih Tanggal dan cari	✓	Berhasil menampilkan data berdasarkan tanggal yang di cari
<b>Screenshot</b>			

ID	Jumlah Masuk	Jumlah Keluar	Waktu
1	95	95	2024-08-18 17:37:38
47	0	12	2024-11-04 13:32:13
48	27	27	2024-11-05 10:43:55
49	0	0	2024-11-05 10:47:21
50	26	52	2024-11-05 11:12:36
51	0	8	2024-11-05 15:00:58
52	0	1	2024-11-05 21:39:36
84	12	47	2024-11-06 10:40:24
95	0	0	2024-11-06 10:44:27
96	6	17	2024-11-06 10:56:18
97	2	3	2024-11-06 11:18:22
98	0	0	2024-11-06 11:20:37
99	33	38	2024-11-06 12:06:17
100	11	8	2024-11-15 18:04:14
101	0	0	2024-11-15 18:04:50

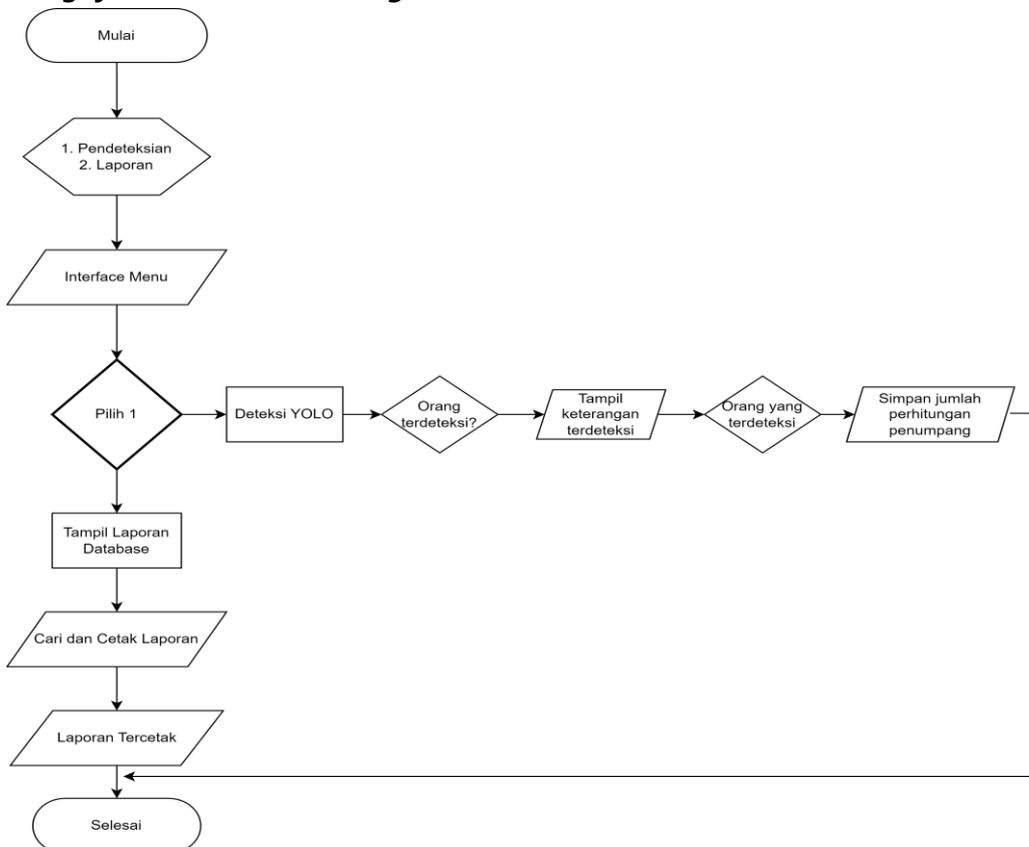
  

No	Tes Faktor	Hasil	Keterangan
4	User menekan menu cetak ke pdf	✓	Berhasil menampilkan hasil data perhitungan dalam bentuk file pdf

**Screenshot**

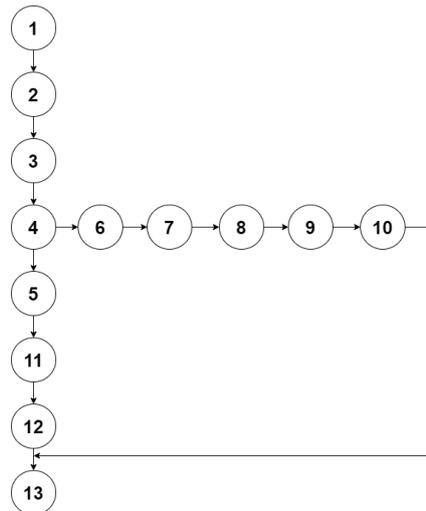
ID	Jumlah Masuk	Jumlah Keluar	Waktu
94	12	47	2024-11-06 10:40:24
95	0	0	2024-11-06 10:44:27
96	6	17	2024-11-06 10:56:18
97	2	3	2024-11-06 11:18:22
98	0	0	2024-11-06 11:20:37
99	33	38	2024-11-06 12:06:17

b. Pengujian *WhiteBox Testing*



**Gambar 4.** *Flowchart* Aplikasi

*Flowchart* ini menjelaskan pengujian *white box testing* pada sistem pendeteksian menggunakan algoritma YOLOv5 dan pengelolaan laporan *database*. Proses dimulai dengan dua pilihan utama, yaitu Pendeteksian dan Laporan. Jika pendeteksian dipilih, sistem akan mendeteksi manusia menggunakan YOLOv5, menampilkan keterangan, dan menyimpan jumlah perhitungan ke *database*. Jika laporan dipilih, sistem menampilkan data, memungkinkan pencarian, dan mencetak laporan. *Flowchart* ini memastikan setiap langkah berjalan sesuai logika yang ditentukan.



**Gambar 5.** *Flowgraph* Aplikasi

*Flowgraf* ini menunjukkan alur logika pengujian sistem dari awal hingga selesai, dengan dua fungsi utama, pendeteksian manusia menggunakan YOLOv5 dan pengelolaan laporan *database*. Pengujian *white box* memastikan setiap langkah dalam sistem berfungsi sesuai dengan logika dan skenario yang dirancang.

**Tabel 5.** Grafik Matriks Aplikasi

	1	2	3	4	5	6	7	8	9	10	11	12	13	E - 1
1		1												$1 - 1 = 0$
2			1											$1 - 1 = 0$
3				1										$1 - 1 = 0$
4					1	1								$2 - 1 = 1$
5											1			$1 - 1 = 0$
6							1							$1 - 1 = 0$
7								1						$1 - 1 = 0$
8									1					$1 - 1 = 0$
9										1				$1 - 1 = 0$
10													1	$1 - 1 = 0$
11												1		$1 - 1 = 0$
12													1	$1 - 1 = 0$
13														0
SUM (E + 1)														$1 + 1 = 2$

Tabel 2. Diatas menunjukkan Nilai *Cyclomatic Complexity*, yang menunjukkan bahwa sistem memiliki 2 jalur *independen*. Hal ini berarti pengujian harus mencakup 2 jalur *eksekusi* berbeda untuk memastikan bahwa semua logika pada *flowgraph* diuji secara menyeluruh.

#### 4. KESIMPULAN

Penelitian ini berhasil membuat sebuah sistem yang mampu mendeteksi dan menghitung orang yang keluar dan masuk dengan akurasi sebesar 64,52%. Berdasarkan hasil pengujian, sistem menunjukkan performa yang cukup baik dalam menjalankan fungsinya. Namun, masih terdapat kelemahan berupa kesalahan deteksi (*false negatives*) yang disebabkan oleh pencahayaan yang kurang optimal dan jarak kamera yang tidak ideal. Meskipun demikian, penelitian ini berhasil membuktikan bahwa algoritma YOLOv5 dapat diimplementasikan secara efektif untuk mendukung otomatisasi perhitungan jumlah orang. Hasil penelitian ini memberikan kontribusi pada pengembangan teknologi deteksi objek dalam berbagai aplikasi operasional.

#### REFERENSI

- Agarwal, A. , & G. R. (2023). *Real-time human detection using YOLOv5 in video streaming environments. Journal of Artificial Intelligence and Applications*, 10(3),(123-135.).
- Aksoy, S. , & G. B. (2021). *Human tracking and activity recognition in dynamic crowds using computer vision techniques. . Sensors and Actuators A: Physical*, 9(7),(1157-1168.).
- Arfah, A. A., & Artikel, I. (2022). Aplikasi Absensi Karyawan Dan *Finger Print* Berbasis Android (Vol. 2). <https://jurnal.umpar.ac.id/index.php/sylog>
- Ashar, M. H. (2022). Implementasi Algoritma YOLOv5 dalam Mendeteksi Penggunaan Masker pada Kantor Biro Umum Gubernur Sulawesi Barat. *Jurnal KLIK*, 9(2),(295–302).
- Basri, M., & Gushari, M. F. (2021). Penerapan *Steganografi* Gambar Berwarna Pada Delapan *Image Cover* Menggunakan Metode Lsb. 1(3). <https://doi.org/10.31850/jsilog.v1i3>
- Bojarski, M. , Y. P. , C. A. , C. K. , F. B. , J. L. , & M. U. (2020). *Explaining How a Deep Neural Network Trained with End-to-End Learning Steers a Car. . ArXiv Preprint ArXiv*; 1704.07911.
- Gao, Y. , Z. Y. , & W. X. (2021). . *Real-time crowd counting and density estimation using deep learning. . IEEE Transactions on Circuits and Systems for Video Technology*, , 31(5),(1742-1753.).

- Gündüz, M. Ş. , & I. G. (2020). . *A new YOLO-based method for real-time crowd detection from video and performance analysis of YOLO models. . Journal of Real-Time Image Processing* , 20(5),(1–12.).
- Khairunnas, K. , Y. E. M. , & Z. A. (2021). Pembuatan Modul Deteksi Objek Manusia Menggunakan Metode YOLO untuk Mobile Robot. *Jurnal Teknologi Informasi Dan Ilmu Komputer (JTIK)*, , 8(4),(679–688.).
- Le, T. , N. M. , & K. H. (2019). *Crowd density estimation and anomaly detection using computer vision for public safety management. . Applied Sciences*, 13(1),(248.).
- Li, J. , Z. T. , & X. Y. (2020). *Human object detection and tracking for intelligent surveillance systems. . Journal of Computer Vision*, 15(4),(292–304).
- Mokayed, R. , A. W. , & K. S. (2020). *Real-time crowd counting and human detection system using deep learning techniques. Journal of Computational Vision and Imaging Systems*, , 12(4),(310-320.).
- Redmon, J. , D. S. , G. R. , & F. A. (2016). *You Only Look Once: Unified, Real-Time Object Detection. . Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, , 779–788.
- Sari, D. P. , et al. (2023). Rancang Bangun Aplikasi Web Pendeteksi Warna pada Citra Digital Menggunakan Python. . *Jurnal Teknologi Informasi* , 14(1),(45-55.).
- Setiawan, A. , P. B. W. , & Y. M. (2020). . Implementasi Algoritma YOLOv5 untuk Deteksi Kerumunan di Ruang Publik. . *Jurnal Ilmu Komputer Dan Teknologi Informasi* , 5(2),(110–118.).
- Sutisna, T. , R. A. R. , S. H. E. , & P. V. H. C. (2023). Penggunaan Computer Vision untuk Menghitung Jumlah Kendaraan Menuju Tempat Keramaian. . *Jurnal Innovative*, 9(1),(45-56.).
- Wang, J. , & L. Y. (2020). *An analysis of artificial intelligence techniques in surveillance applications. . Applied Sciences* , 13(8),(4956.).
- Yasin, M. F. N. , & S. M. (2022). Kombinasi Deteksi Objek, Pengenalan Wajah dan Perilaku Anomali menggunakan State Machine untuk Kamera Pengawas. *Jurnal Teknologi Informasi Dan Ilmu Komputer (JTIK)*, 9(1),(45-56.).