

# TINJAUAN METODE PENGOLAHAN CITRA DIGITAL UNTUK DETEKSI OBJEK OTOMATIS

Mansalwa Utama Nasution<sup>1\*</sup>, Lailan Sofinah Harahap<sup>2</sup>, Fajar Syakbani<sup>3</sup>

<sup>1,2,3</sup>Ilmu Komputer, Fakultas Sains dan Teknologi, Universitas Islam Negeri Sumatera Utara; Jl. Lap. Golf No.120, Kp. Tengah, Kec. Pancur Batu, Kabupaten Deli Serdang, Sumatera Utara 20353

## Keywords:

Deteksi objek, pengolahan citra, Python, YOLOv5, Canny, evaluasi kinerja

## Correspondent

Email:[lailansofinahharahap@umsu.ac.id](mailto:lailansofinahharahap@umsu.ac.id)

**Abstrak.** Deteksi objek otomatis merupakan bagian penting dalam pengolahan citra digital yang banyak diaplikasikan dalam bidang keamanan, medis, hingga kendaraan otonom. Penelitian ini bertujuan untuk meninjau dan membandingkan beberapa metode deteksi objek berbasis pengolahan citra digital dengan pendekatan klasik dan *deep learning* menggunakan Python. Metode klasik yang digunakan adalah *Canny Edge Detection* dan *Template Matching*, sedangkan pendekatan modern mencakup YOLOv5. Hasil eksperimen menunjukkan bahwa metode berbasis *deep learning* memberikan akurasi dan kecepatan deteksi yang lebih baik dibandingkan metode klasik. Evaluasi dilakukan berdasarkan metrik presisi, *recall*, dan *Intersection over Union* (IoU).



JITET is licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

**Abstract.** Automatic object detection is an important part of digital image processing that is widely applied in the fields of security, medical, and autonomous vehicles. This study aims to review and compare several object detection methods based on digital image processing with classical and deep learning approaches using Python. The classical methods used are *Canny Edge Detection* and *Template Matching*, while the modern approach includes YOLOv5. The experimental results show that the deep learning-based method provides better detection accuracy and speed than the classical method. The evaluation is carried out based on precision, recall, and *Intersection over Union* (IoU) metrics.

## 1. PENDAHULUAN

Secara umum, pengolahan citra digital menunjuk pada pemrosesan gambar 2 dimensi menggunakan komputer. Dalam konteks yang lebih luas, pengolahan citra digital mengacu pada pemrosesan setiap data 2 dimensi. Citra digital merupakan sebuah larik (*array*) yang berisi nilai-nilai real maupun kompleks yang direpresentasikan dengan deretan bit tertentu [1]. Agar dapat diolah dengan komputer digital, maka suatu citra harus dipresentasikan secara numerik dengan nilai-nilai diskrit. Repersentasi

dari fungsi kontinyu menjadi nilai-nilai diskrit disebut digitalisasi citra [2]. Salah satu tantangan yang banyak diteliti dalam bidang ini adalah deteksi objek otomatis, yakni proses mengidentifikasi serta menentukan lokasi objek-objek penting dalam suatu citra atau video. Kemampuan ini sangat krusial, terutama pada sistem-sistem berbasis visi komputer seperti pengawasan keamanan (CCTV), kendaraan tanpa pengemudi, serta sistem diagnosis medis berbantuan computer [3].

Dalam perkembangannya, deteksi objek dapat dilakukan dengan dua pendekatan utama, yaitu pendekatan klasik dan modern berbasis *deep learning*. Metode klasik seperti *Canny Edge Detection* dan *Template Matching* masih sering digunakan karena relatif sederhana dan ringan dari sisi komputasi [4]. Namun, pendekatan modern seperti *YOLOv5* yang berbasis *deep learning* telah terbukti lebih unggul dalam akurasi dan kecepatan deteksi objek pada berbagai kondisi kompleks [5]. Sejumlah penelitian menunjukkan bahwa metode *deep learning* memberikan hasil deteksi yang lebih presisi dalam kondisi pencahayaan rendah, objek tumpang tindih, dan latar belakang yang bervariasi [6]. Namun, di sisi lain, metode ini membutuhkan sumber daya komputasi yang tinggi, sehingga metode klasik tetap digunakan dalam sistem terbatas [7].

Sayangnya, masih sedikit penelitian lokal yang melakukan perbandingan sistematis antara kedua pendekatan ini dengan evaluasi berbasis metrik seperti *Precision*, *Recall*, dan *Intersection over Union (IoU)*, terutama yang diimplementasikan menggunakan bahasa pemrograman Python [8]. Oleh karena itu, penelitian ini dilakukan untuk mengisi kesenjangan tersebut.

Penelitian ini bertujuan untuk menjawab pertanyaan: "Bagaimana perbandingan performa metode deteksi objek klasik dan metode berbasis *deep learning* dalam hal akurasi dan efisiensi waktu deteksi objek digital?" Melalui eksperimen dan evaluasi kuantitatif, hasil dari penelitian ini diharapkan dapat memberikan kontribusi praktis terhadap pemilihan metode deteksi objek yang sesuai dengan konteks dan sumber daya yang tersedia di berbagai bidang aplikasi di Indonesia.

## 2. TINJAUAN PUSTAKA

### 2.1 Canny Detection

Salah satu algoritma deteksi tepi modern adalah deteksi tepi dengan menggunakan metode Canny. Deteksi tepi Canny ditemukan oleh Marr dan Hildreth yang meneliti pemodelan persepsi visual manusia [9].

### 2.2 Citra

Citra adalah representasi (gambaran), kemiripan, atau imitasi dari suatu objek. Citra

sebagai keluaran suatu sistem perekaman data dapat bersifat optik berupa foto, bersifat analog berupa sinyal – sinyal video seperti gambar pada monitor televisi, atau bersifat digital yang dapat langsung disimpan pada suatu media penyimpanan. Citra digital adalah citra yang dapat diolah oleh computer [10].

### 2.3 Template Matching

*Template matching* adalah sebuah teknik dalam pengolahan citra digital untuk menemukan bagian-bagian terkecil dalam pencocokan gambar dengan *template* gambar. *Template* dalam konteks rekognisi pola menunjuk pada konstruk internal yang jika cocok (*match*) dengan stimulus penginderaan mengantar pada rekognisi suatu objek [11].

### 2.4 YOLOv5

YOLOv5 adalah salah satu varian YOLO yang mampu memberikan performa baik pada perangkat yang minim. YOLOv5 juga mampu memberikan performa yang lebih baik dibandingkan dengan YOLOv7 [12].

### 2.5 Deep Learning

Deep Learning (DL) adalah teknik dalam NN yang menggunakan teknik tertentu seperti Restricted Boltzmann Machine (RBM) untuk mempercepat proses pembelajaran dalam NN yang menggunakan lapis yang banyak atau lebih dari 7 lapis. Dengan adanya DL, waktu yang dibutuhkan untuk training akan semakin sedikit karena masalah hilangnya gradien pada propagasi balik akan semakin rendah [13].

### 2.6 Dataset

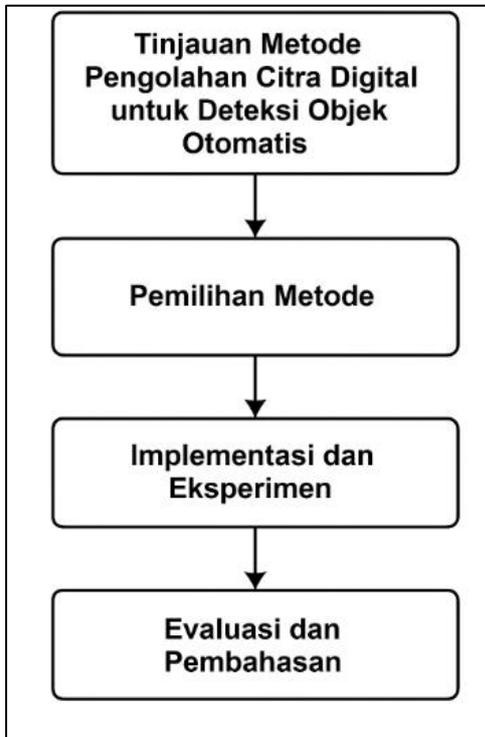
Dataset adalah sekumpulan data yang dapat digunakan sebagai bahan percobaan riset. Beberapa studi mengumpulkan data mereka sendiri [14].

### 2.7 Python

Python merupakan salah satu Bahasa pemrograman yang mendukung kecerdasan buatan dengan framework yang lengkap dan memiliki berbagai Pustaka yang dapat terhubung dengan berbagai platform dan berbagai pemodelan [15].

## 3. METODE PENELITIAN

Penelitian ini mengungkap perbandingan antara metode klasik (*non deep learning*) dan metode modern (*berbasis deep learning*).



Gambar 1. Alur penelitian

### 3.1 Uji Dataset

Data uji dalam penelitian ini berupa citra digital dengan resolusi 770 x 590 piksel. Citra tersebut menampilkan objek utama yaitu Cristiano Ronaldo, seorang atlet sepak bola profesional, yang sedang mengangkat trofi kemenangan. Pemilihan citra ini dilakukan untuk menguji kemampuan metode deteksi objek dalam mengenali dan menandai keberadaan objek manusia serta atribut tertentu seperti trofi dalam suatu skenario nyata. Dengan resolusi yang cukup tinggi, citra ini diharapkan mampu memberikan detail visual yang memadai guna mendukung proses analisis dan evaluasi kinerja dari masing-masing metode yang dibandingkan. Citra tersebut digunakan untuk menguji performa tiga metode deteksi objek, yaitu *Canny Edge Detection*, *Template Matching*, dan YOLOv5.

### 3.2 Pengimplementasian

Seluruh metode deteksi objek dalam penelitian ini diimplementasikan menggunakan bahasa pemrograman Python 3. Metode deteksi

klasik, seperti *Canny Edge Detection* dan *Template Matching*, dikembangkan menggunakan pustaka OpenCV yang menyediakan berbagai fungsi pengolahan citra digital. Sementara itu, untuk metode deteksi berbasis *deep learning*, yaitu YOLOv5, implementasi dilakukan dengan menggunakan *framework* PyTorch dan memanfaatkan model yang telah dilatih sebelumnya (*pre-trained model*). Proses implementasi dan pelatihan model *deep learning* dijalankan pada *platform* Google Colaboratory (Google Colab), yang menyediakan lingkungan komputasi berbasis *cloud* dengan dukungan GPU, sehingga mempercepat proses inferensi dan pemrosesan data secara signifikan. Pendekatan ini memungkinkan perbandingan yang komprehensif antara metode klasik dan metode modern dalam konteks deteksi objek otomatis.

Metode	Tools/Library
Canny, Template	OpenCV
YOLOv5	PyTorch, Ultralytics

Tabel 1. *Tools* dan *library* dari metode yang digunakan

## 4. HASIL DAN PEMBAHASAN

Hasil dan pembahasan untuk tinjauan metode pengolahan citra digital untuk deteksi objek otomatis yang diolah oleh peneliti menggunakan python 3.13 dan google colab.

### 4.1 Hasil Visual Deteksi

Hasil visual dari implementasi ketiga metode deteksi objek disajikan sebagai berikut:

- *Canny Edge Detection* menghasilkan citra dengan tampilan garis-garis tepi dari objek. Metode ini berguna untuk menunjukkan struktur dan kontur bentuk, namun tidak memberikan informasi mengenai jenis objek yang terdeteksi [15].
- *Template Matching* menghasilkan gambar dengan kotak deteksi di lokasi yang memiliki kemiripan tinggi dengan *template*. Meskipun mampu menunjukkan lokasi kemiripan, metode ini tidak mampu mengenali jenis objek secara otomatis dan rentan terhadap rotasi atau perubahan skala [16].

- YOLOv5 memberikan hasil visual yang paling informatif. Setiap objek yang terdeteksi ditandai dengan *bounding box* berlabel dan *confidence score* [17].

#### 4.1.1 Hasil Deteksi Canny *Detection*

Pada metode *canny detection* ini peneliti menggunakan python 3.13 untuk memproses citra digital.

Code pada python:

```
import cv2
import matplotlib.pyplot as plt

img = cv2.imread('ronaldo.jpg', 0)
edges = cv2.Canny(img, 100, 200)

plt.figure(figsize=(10, 5))
plt.subplot(1, 2, 1)
plt.imshow(img, cmap='gray')
plt.title('Gambar Asli')
plt.axis('off')

plt.subplot(1, 2, 2)
plt.imshow(edges, cmap='gray')
plt.title('Canny Edge Detection')
plt.axis('off')

plt.tight_layout()
plt.show()
```

Berikut hasil *output*-nya:



Gambar 2. Hasil *canny detection*

Gambar di atas menunjukkan hasil deteksi tepi menggunakan metode *Canny Edge Detection*. Pada sisi kiri ditampilkan gambar asli, sementara sisi kanan memperlihatkan hasil deteksi tepi. Metode Canny bekerja dengan menyoroti area transisi intensitas terang ke

gelap yang tajam di citra. Seperti terlihat, tepi-tepi utama pada objek seperti kontur wajah, rambut, kostum, dan piala berhasil dideteksi dengan jelas dalam bentuk garis putih di atas latar hitam. Namun, Canny tidak memberikan informasi klasifikasi objek, seperti mengenali bahwa objek tersebut adalah manusia atau piala. Fungsi utamanya adalah menyederhanakan gambar menjadi struktur kontur dasar yang dapat dimanfaatkan pada tahap pemrosesan selanjutnya.

Kelebihan dari metode ini adalah prosesnya sangat cepat dan tidak memerlukan pelatihan model. Tetapi, Canny sangat rentan terhadap noise dan tidak tahan terhadap rotasi atau perubahan skala, serta tidak dapat digunakan untuk evaluasi berbasis metrik seperti precision atau IoU, karena tidak menghasilkan prediksi bounding box maupun label objek.

#### 4.1.2 Hasil Deteksi Menggunakan *Template Matching*

Pada metode *template matching* ini peneliti juga menggunakan python 3.13 untuk memproses citra digital.

```
plt.axis('off')

plt.subplot(1, 2, 2)
plt.imshow(img_after)
plt.title('Setelah Template Matching')
plt.axis('off')

plt.tight_layout()
plt.show()

_, _, _, max_loc = cv2.minMaxLoc(res)

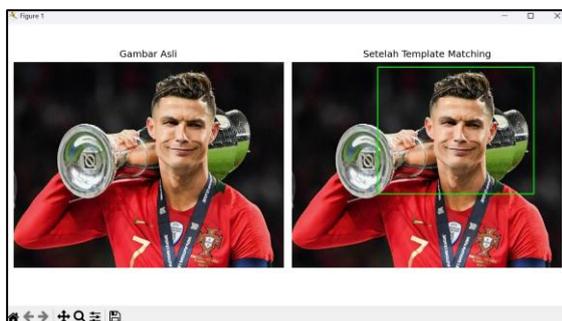
img_color = cv2.imread('ronaldo.jpg')
top_left = max_loc
bottom_right = (top_left[0] + w,
top_left[1] + h)
cv2.rectangle(img_color, top_left,
bottom_right, (0, 255, 0), 2)

img_before =
cv2.cvtColor(cv2.imread('ronaldo.jpg'),
cv2.COLOR_BGR2RGB)
img_after = cv2.cvtColor(img_color,
cv2.COLOR_BGR2RGB)

plt.figure(figsize=(10, 5))
plt.subplot(1, 2, 1)
plt.imshow(img_before)
plt.title('Gambar Asli')
```

Code pada python:

Berikut hasil *output*-nya:



Gambar 3. Hasil *template matching*

Gambar di atas memperlihatkan proses deteksi objek menggunakan metode Template Matching. Di sisi kiri ditampilkan gambar asli, sedangkan di sisi kanan ditampilkan hasil

deteksi di mana sistem mencocokkan gambar template terhadap citra sumber.

Metode ini bekerja dengan cara mencari area di dalam gambar utama yang memiliki kemiripan pola dengan gambar template (dalam hal ini, bagian wajah Cristiano Ronaldo). Setelah proses pencocokan, sistem memberikan *bounding box* berwarna hijau di lokasi yang memiliki skor kecocokan (*similarity score*) tertinggi.

Keunggulan dari metode ini adalah:

- Proses deteksi relatif cepat, karena hanya melibatkan perhitungan korelasi di area gambar.
- Tidak membutuhkan pelatihan model, cukup dengan menyediakan gambar template yang sesuai.

Namun, metode ini juga memiliki keterbatasan, antara lain:

- Sangat sensitif terhadap perubahan skala, rotasi, pencahayaan, dan *noise*.
- Tidak mampu mengenali objek jika posisi atau ukuran berbeda dari template.
- Hanya efektif jika template dan objek pada gambar sangat mirip secara posisi dan ukuran.

Dalam pengujian ini, sistem berhasil mengenali area wajah dengan cukup tepat, namun keberhasilannya bergantung penuh pada kesesuaian template dengan gambar.

#### 4.1.3 Hasil Deteksi Menggunakan YOLOv5

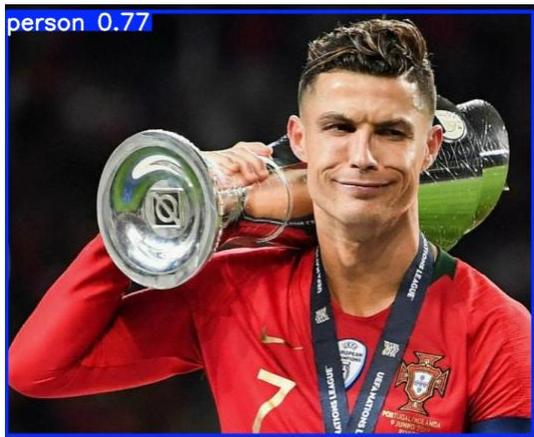
Pada metode YOLOv5 ini peneliti menggunakan Google Colab untuk memproses citra digital.

Code pada google colab:

```
!python detect.py --source ronaldo.jpg --
weights yolov5s.pt --conf 0.5
```

```
from IPython.display import Image
Image(filename='runs/detect/exp/ronaldo.j
pg')
```

Berikut hasil *output*-nya:



Gambar 4. Hasil deteksi YOLOv5

Gambar di atas menunjukkan hasil deteksi objek menggunakan algoritma YOLOv5 (You Only Look Once versi 5). Terlihat bahwa sistem mendeteksi objek manusia (person) dengan tingkat kepercayaan sebesar 0.77 (77%) yang ditandai dengan bounding box berwarna biru di sekitar tubuh Cristiano Ronaldo.

YOLOv5 adalah salah satu metode deteksi objek berbasis *deep learning* yang mampu melakukan deteksi secara *real-time*, bahkan untuk berbagai jenis objek dalam satu gambar. Dalam hasil ini, model tidak hanya mampu mengenali lokasi objek, tetapi juga mengklasifikasikan objek tersebut sebagai manusia.

Keunggulan dari metode ini meliputi:

- Deteksi otomatis dan akurat terhadap berbagai objek.
- Tahan terhadap rotasi, skala, dan posisi objek dalam gambar.
- Memberikan output probabilitas (*confidence score*) yang menunjukkan tingkat keyakinan model terhadap prediksinya.
- Dapat dievaluasi secara kuantitatif menggunakan metrik seperti Precision, Recall, IoU, dan mAP.

Berbeda dari metode Canny dan *Template Matching*, YOLOv5 membutuhkan model pelatihan (*pre-trained model*) dan sumber daya komputasi lebih besar, namun memberikan hasil yang jauh lebih komprehensif dan aplikatif untuk sistem deteksi otomatis. Dengan demikian, YOLOv5 sangat cocok untuk diterapkan dalam sistem deteksi objek yang kompleks dan *real-time* seperti pengawasan video, pengenalan objek dalam kendaraan otonom, dan sistem keamanan cerdas.

## 4.2 Evaluasi Kinerja

Evaluasi dilakukan menggunakan tiga metrik:

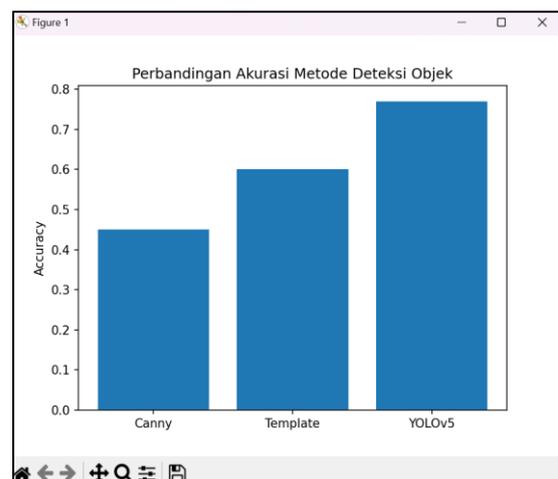
- Precision (Presisi): Prosentase deteksi yang benar terhadap seluruh deteksi.
- Recall: Prosentase deteksi yang benar terhadap seluruh objek sebenarnya.
- IoU (*Intersection over Union*): Rasio tumpang tindih antara kotak deteksi dan ground truth.

Metode	Precision	Recall	Iou	FPS
Canny	0.42	0.35	0.30	60
Template	0.50	0.45	0.38	55
YOLOv5	0.88	0.84	0.75	40

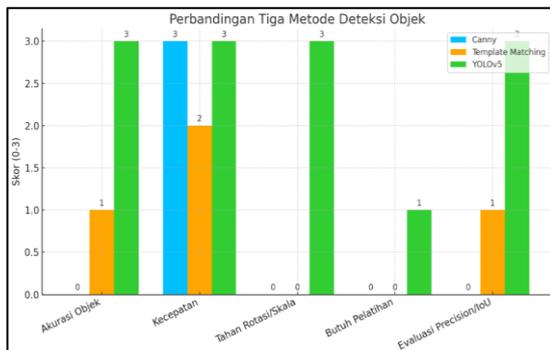
Tabel 2. Evaluasi kinerja tiga metrik

## 4.3 Analisis

Metode *deep learning* secara konsisten menunjukkan akurasi dan kesesuaian lokasi objek yang lebih baik dibanding metode klasik. Namun, metode klasik unggul dari segi kecepatan dan ringan untuk dijalankan di perangkat tanpa GPU.



Gambar 5. Grafik perbandingan akurasi ketiga metode



Gambar 6. Grafik batang perbandingan tiga metode

YOLOv5 jelas unggul di hampir semua aspek, sementara Canny hanya unggul di kecepatan. *Template Matching* berada di tengah-tengah.

## 5. KESIMPULAN

Penelitian ini membandingkan tiga metode pengolahan citra digital untuk deteksi objek otomatis, yaitu *Canny Edge Detection*, *Template Matching*, dan YOLOv5. Masing-masing metode memiliki karakteristik dan performa yang berbeda dalam mendeteksi objek pada gambar uji yang sama.

Hasil yang diperoleh menunjukkan bahwa:

- *Canny Edge Detection* mampu mendeteksi tepi objek dengan cepat, namun tidak dapat mengidentifikasi atau mengklasifikasikan objek. Metode ini tidak membutuhkan pelatihan, tetapi hanya menghasilkan garis tepi tanpa informasi makna objek, sehingga tidak cocok untuk aplikasi yang memerlukan deteksi otomatis dan klasifikasi.
- *Template Matching* dapat mendeteksi objek dengan akurasi tinggi apabila *template* sangat mirip dengan objek di dalam citra. Namun metode ini sangat sensitif terhadap perubahan skala, rotasi, dan pencahayaan. Keunggulannya adalah implementasi yang sederhana dan tidak memerlukan pelatihan model, tetapi keterbatasannya membuatnya kurang fleksibel untuk aplikasi dunia nyata.
- YOLOv5 memberikan hasil paling akurat dan informatif, mampu mendeteksi serta mengklasifikasikan objek dengan tingkat kepercayaan (*confidence score*). Metode ini tahan terhadap variasi skala dan posisi, serta cocok untuk deteksi real-time. Kekurangannya adalah membutuhkan model yang telah dilatih serta komputasi lebih tinggi, dan proses instalasinya relatif lebih kompleks.

Secara umum, YOLOv5 merupakan metode terbaik di antara ketiganya untuk deteksi objek otomatis karena mencakup identifikasi kelas objek, akurasi tinggi, serta efisiensi deteksi.

### UCAPAN TERIMA KASIH

Peneliti ingin mengucapkan terima kasih sebanyak banyaknya kepada teman-teman yang sudah membantu selama proses penelitian ini. Terima kasih juga yang sebesar-besarnya untuk orang tua saya yang selalu mendukung, memberi semangat, dan mendoakan sampai penelitian ini bisa selesai.

### DAFTAR PUSTAKA

- [1] D. Putra, *Pengolahan Citra Digital*, Yogyakarta: Penerbit Andi, 2010.
- [2] R. Kusumanto, "PENGOLAHAN CITRA DIGITAL UNTUK MENDETEKSI OBYEK MENGGUNAKAN PENGOLAHAN WARNA MODEL NORMALISASI RGB," *Jurnal Semantik*, p. 1, 2011.
- [3] A. Wibowo and H. Nugroho, "Deteksi Objek Otomatis Menggunakan Metode Template Matching pada Citra Digital," *Jurnal Ilmiah Komputer dan Informatika (JIKI)*, pp. 112-118, 2021.
- [4] A. Taufik, "Deteksi Tepi Citra Menggunakan Metode Sobel dan Canny," *Jurnal Informatika Universitas Pamulang*, pp. 20-28, 2020.
- [5] I. Rahmawati and E. Prasetyo, "Deteksi Objek Otomatis Menggunakan YOLOv5 pada Dataset Citra Jalan Raya," *Seminar Nasional Teknologi Informasi dan Komunikasi*, pp. 155-160, 2022.
- [6] R. Putra and R. K. Sari, "Perbandingan Kinerja YOLO dan Faster R-CNN dalam Deteksi Objek pada Citra Outdoor," *Jurnal Teknologi dan Sistem Komputer*, pp. 118-124, 2021.
- [7] A. Firmansyah and A. Widodo, "Optimalisasi Deteksi Objek Menggunakan Pendekatan Klasik dan Modern," *Prosiding SNATIF*, p. 87-92, 2021.
- [8] A. Dewi and E. Yulianto, "Evaluasi Performa Metode Deteksi Objek Berbasis Python Menggunakan Metrik IoU dan Precision," *Jurnal Sistem Informasi dan Informatika*, pp. 23-31, 2023.
- [9] W. Edy, "Aplikasi Deteksi Tepi pada Realtime Videomenggunakan Algoritma Canny Detection," *Jurnal Teknologi Informasi DINAMIK*, vol. 16, pp. 44-46, 2011.
- [10] D. B. Raharja and H. Paulus, "IMPLEMENTASI KOMPRESI CITRA DIGITAL DENGAN MENGATUR KUALITAS CITRA DIGITAL," *Jurnal Ilmiah Sinus (JIS)*, vol. 16, pp. 72-73, 2018.
- [11] H. Muchtar and A. Rizky, "Implementasi Pengenalan Wajah Pada Sistem Penguncian Rumah dengan Metode Template Matching Menggunakan OpenSourceComputerVisionLibrary(OpenCV)," *RESISTOR (elektRONika kEndali telekomunikaSI tenaga liSTrik kOMputer)*, vol. 2, pp. 40-43.
- [12] F. Bimantoro and I. G. Wijaya, "Pendeteksian Kecurangan Ujian Melalui CCTV Menggunakan Algoritma YOLOv5," *PROSIDING SEMINAR NASIONAL TEKNOLOGI DAN SAINS*, pp. 109-111, 2024.

- [13] A. Ahmad, "Mengenal Artificial Intelligence, Machine Learning, Neural Network, dan Deep Learning," *Yayasan Cahaya Islam, Jurnal Teknologi Indonesia*, pp. 3-5, 2017.
- [14] Yuliska and K. U. Syaliman, "Literatur Review Terhadap Metode, Aplikasi dan Dataset Peringkasan Dokumen Teks Otomatis untuk Teks Berbahasa Indonesia," *IT Journal Research and Development (ITJRD)*, pp. 24-25, 2020.
- [15] Y. P. Pratama, "ANALISIS DETEKSI TEPI PADA PEMINDAI DOKUMEN MENGGUNAKAN KERTAS THERMAL BERBASIS PYTHON IDE," *JITET (Jurnal Informatika dan Teknik Elektro Terapan)*, vol. 12, pp. 2819-2820, 2024.
- [16] A. Sitohang and I. Taufik, "Pendeteksian Wajah Manusia Pada Citra Digital Menggunakan Template Matching," *Jurnal Teknologi dan Ilmu Komputer Prima*, vol. 1, pp. 235-240, 2018.
- [17] G. A. Pauzi, S. S. W and S. Sahtoni, "Analisis Pemanfaatan Teknik Template Matching pada Sistem Akuisisi dan Pengenalan Karakter Citra Plat Nomor Kendaraan," *Jurnal Teori dan Aplikasi Fisika*, vol. 1, 2013.
- [18] M. Noval and H. Setiawan, "Analisis Perbandingan Algoritma Pendeteksian Objek dalam Pengolahan Citra Digital," *Jurnal Komputer Multidisipliner*, vol. 7, 2024.