

# **Penggunaan YOLOV3 dan OpenCV untuk Deteksi Senjata Tajam dalam Analisis Forensik Digital**

**Keisha Nuur Al Deena<sup>\*</sup>, Aulia Najmina Nugrahani, Grace Philbertha Maharani, Nur Hakim Kadarisman, Marha Karamina Ghassani, Angelina Hermanto**

Program Studi Statistika, Departemen Pendidikan Matematika, Fakultas Matematika dan Ilmu Pengetahuan, Universitas Negeri Yogyakarta, Yogyakarta

<sup>\*</sup>Corresponding author [keishanuur.2022@student.uny.ac.id](mailto:keishanuur.2022@student.uny.ac.id)

## **Abstrak**

Keamanan publik merupakan salah satu aspek penting dalam kehidupan bermasyarakat. Pada era modern ini, ancaman terhadap keamanan publik semakin beragam dan kompleks. Penggunaan senjata tajam dalam aksi kriminal, baik di tempat umum maupun di area tertutup adalah salah satu ancaman yang paling sering terjadi. Ancaman tersebut dapat dicegah dengan menggunakan kamera pengawas yang berperan sebagai alat dokumentasi dan pemantauan aktivitas yang terjadi pada suatu area ketika terjadi sesuatu yang tidak diinginkan. Kamera pengawas dapat menjadi barang bukti yang sah, ketika terjadi hal kriminal yang terekam. Penelitian ini bertujuan mengembangkan model *object detection* berbasis pengolahan citra untuk mengenali senjata tajam dan api (pistol) dari rekaman kamera pengawas. Dataset yang digunakan berasal dari rekaman kamera pengawas yang diunggah di YouTube. Proses pengembangan sistem melibatkan penggunaan OpenCV dan YOLOv3. Pengujian dilakukan pada berbagai kondisi gambar, termasuk rotasi, perubahan kecerahan, dan ketajaman gambar. Hasil pengujian menunjukkan tingkat akurasi berkisar antara 50% hingga 85%, dengan beberapa video tidak terdeteksi. Faktor seperti jarak, pencahayaan, dan kualitas gambar mempengaruhi performa deteksi. Untuk meningkatkan akurasi, direkomendasikan untuk melakukan optimasi algoritma dan augmentasi data, meningkatkan kualitas dataset, serta menggunakan video resolusi tinggi. Penelitian ini berkontribusi pada pengembangan sistem keamanan otomatis dan forensik digital untuk pencegahan dan penanganan kejahatan bersenjata.

Kata kunci: Senjata, Kamera Pengawas, *Object Detection*, OpenCV, YOLOv3, Forensik Digital

## 1. PENDAHULUAN

Keamanan publik merupakan salah satu aspek penting dalam kehidupan bermasyarakat. Pada era modern ini, ancaman terhadap keamanan publik semakin beragam dan kompleks. Penggunaan senjata tajam dalam aksi kriminal, baik di tempat umum maupun di area tertutup adalah salah satu ancaman yang paling sering terjadi. Berdasarkan data yang didapat dari aplikasi EMP Pusiknas Bareskrim Polri, sebanyak 3.655 kasus kekerasan dan kejahatan terkait senjata tajam maupun api ditindak Polri sejak Januari sampai 14 November 2023. Tantangan yang harus diatasi adalah menangkap dan mencegah penggunaan senjata tajam secara cepat dan efisien, mengingat dampaknya yang fatal terhadap korban dan dampak psikologis yang ditimbulkan pada masyarakat. Efektivitas penegakan hukum dalam menangani kasus-kasus ini sangat tergantung pada kecepatan respon dan kemampuan teknologi yang digunakan dalam mendeteksi dan mencegah tindakan kriminal.

Kejahatan tersebut dapat dicegah dengan pemantauan menggunakan kamera pengawas yang berperan sebagai alat dokumentasi dan pemantauan aktivitas yang terjadi pada suatu area ketika terjadi sesuatu yang tidak diinginkan. Kamera pengawas dapat menjadi barang bukti yang sah, ketika terjadi hal kriminal yang terekam. Penggunaan kamera pengawas telah menjadi standar dalam sistem keamanan di berbagai tempat, berkat kemajuan teknologi. Namun, kamera pengawas konvensional hanya dapat melakukan perekaman dan pemantauan visual, tidak dapat mengidentifikasi senjata tajam secara otomatis dan memberikan peringatan dini. Oleh karena itu, perlu dilakukan pengolahan citra hasil kamera pengawas dengan model *object detection* sehingga model dapat mengenali suatu tindak kriminalitas bersenjata khususnya mendeteksi senjata api sehingga dapat membantu pihak keamanan atau penegak hukum dalam menerima laporan ketika terjadinya suatu tindak kriminalitas bersenjata.

Penelitian ini bertujuan untuk mengembangkan model *object detection* berbasis pengolahan citra yang mampu mengenali dan mendeteksi senjata tajam dan senjata api (pistol) dari rekaman kamera pengawas. Implementasi model tersebut dalam sistem pemantauan kamera pengawas diharapkan dapat memberikan peringatan dini secara otomatis ketika terdeteksi adanya senjata tajam atau senjata api, sehingga memudahkan pihak keamanan atau penegak hukum dalam menangani tindak kriminalitas bersenjata. Penelitian ini juga berkontribusi dalam konteks forensik digital dengan mengidentifikasi bukti atau motif kejahatan dalam citra atau data digital. Penggunaan teknologi deteksi kejahatan dalam forensik digital dapat membantu mengungkap dan menganalisis bukti visual yang mungkin terlewatkan oleh pengamatan manusia, serta menyediakan alat yang kuat bagi penyidik untuk memahami konteks dan kronologi suatu kejadian kriminal. Dengan menggabungkan kemajuan dalam pengolahan citra dan forensik digital, penelitian ini dapat meningkatkan kemampuan penegak hukum dalam mengidentifikasi dan menindak pelaku kejahatan secara lebih efektif.

Penelitian ini mengembangkan aplikasi pendeteksian video dari kamera pengawas yang terdapat senjata tajam dan senjata api (pistol) akan disimpan dalam dokumen berbentuk foto yang dapat dijadikan barang bukti forensik digital. Ketika model *object detection* mendeteksi senjata dalam rekaman kamera pengawas, sistem secara otomatis akan mengambil cuplikan gambar dari momen tersebut dan menyimpannya sebagai dokumen gambar. Gambar-gambar ini kemudian dapat digunakan sebagai bukti visual dalam investigasi kriminal, membantu penegak hukum dalam mengidentifikasi pelaku dan menyusun kronologi kejadian. Dokumentasi ini juga memudahkan proses hukum, karena menyediakan bukti yang jelas dan dapat diandalkan dari kejadian yang terekam. Dengan demikian, integrasi teknologi deteksi otomatis dengan sistem kamera pengawas tidak hanya meningkatkan pencegahan dan respons terhadap kejahatan bersenjata tetapi juga memperkuat kapasitas forensik digital dalam mengungkap dan menindak pelaku kejahatan.

## **2. METODOLOGI**

### **2.1. Pengumpulan Dataset**

Dataset pada model deteksi objek merupakan komponen yang penting terdiri dari kumpulan citra yang akan diproses pada tahap pelatihan model. Pemilihan dataset yang tepat sangat penting karena harus sesuai dengan permasalahan yang dihadapi agar model yang dihasilkan dapat berfungsi secara optimal. Dalam konteks penelitian ini, tujuannya adalah untuk membangun model deteksi senjata, khususnya pistol, yang dapat membantu dalam pengawasan dan keamanan. Penelitian ini menggunakan data yang diambil dari video di YouTube yang menampilkan hasil rekaman CCTV terkait kejadian kejahatan yang melibatkan penggunaan pistol. Proses pengumpulan dataset dimulai dengan pemilihan video yang relevan, memastikan video memiliki kualitas yang cukup baik agar citra yang diekstrak jelas dan dapat digunakan untuk pelatihan model.

### **2.2. Pengembangan Sistem**

#### **2.2.1. Perangkat lunak yang digunakan**

##### **a. Open CV**

*OpenCV* (*Open Computer Vision*) merupakan sebuah *library* yang digunakan untuk pemrosesan citra. Fungsinya adalah mengubah citra dari format RGB menjadi format biner serta menjadi platform yang digunakan untuk mengolah citra menggunakan beragam bahasa pemrograman. Pada *library* ini akan di gunakan beberapa perintah seperti *cv2.dnn.readNet*, *cv2.dnn.blobFromImage*, *cv2.rectangle*, *cv2.dnn.NMSBoxes*, *cv2.imshow*, *cv2.putText*, *cv2.destroyAllWindows*, *cv2.imwrite*.

b. *Numpy*

*Numpy (Numerical Python)* adalah *library* yang fokus pada komputasi ilmiah. *Numpy* menyediakan fungsi yang siap pakai untuk memudahkan perhitungan saintifik seperti operasi pada matriks, aljabar linear, statistik, dan sebagainya. *Library* ini menyediakan array objek yang kuat, N-dimensional array (*ndarray*), yang digunakan untuk menyimpan dan memanipulasi data dalam bentuk vektor dan matriks. Salah satu kekuatan utama *NumPy* adalah kemampuannya untuk melakukan operasi matematis dan logis pada array dengan sangat efisien

c. *YOLOv3*

*YOLOv3* adalah Sistem pendeteksian yang dilakukan dengan menggunakan *repurpose classifer* atau *localizer* untuk melakukan deteksi. Sebuah model diterapkan pada sebuah citra di beberapa lokasi dan skala. *YOLOv3* dirancang untuk menjadi detektor multi-skala daripada pengklasifikasi gambar. Sistem ini membagi gambar input menjadi grid, dan setiap grid cell memprediksi bounding boxes serta probabilitas kelas untuk objek yang mungkin ada dalam cell tersebut. *YOLOv3* menggunakan jaringan saraf konvolusional (*Convolutional Neural Network*) yang dioptimalkan untuk menangani variasi skala objek dengan lebih baik, melalui penggunaan tiga skala berbeda untuk membuat prediksi, sehingga mampu mendeteksi objek kecil maupun besar secara lebih akurat.

d. *Non-Maximum Suppression (NMS)*

*Non-Maximum Suppression (NMS)* adalah teknik yang digunakan dalam deteksi objek untuk mengurangi jumlah prediksi berlebih (*redundant*) dan memastikan bahwa deteksi objek yang berdekatan tidak menyebabkan banyak kotak pembatas (*bounding boxes*) yang menumpuk di sekitar objek yang sama. Proses *NMS* membantu dalam memilih kotak pembatas terbaik di antara semua prediksi yang tumpang tindih berdasarkan skor kepercayaan (*confidence score*).

e. Tahap Deteksi

Berikut adalah langkah-langkah deteksi senjata api dan penyimpanan dokumen foto menggunakan *Jupyter* dengan menggunakan bahasa *Python*.

1. Memuat model *Yolov3* menggunakan bobot yang telah dilatih sebelumnya (*yolov3\_training\_2000.weights*) dan file konfigurasi (*yolov3\_testing.cfg*).

2. Mengambil input menggunakan *cv2.VideoCapture* dari *OpenCV*.
3. Membaca dan memproses frame menggunakan *cv2.dnn.blobFromImage* untuk membuat blob yang bisa dimasukkan ke dalam model Yolov3.
4. Deteksi objek menggunakan yolov3 dengan memproses input untuk menghasilkan deteksi (*net.forward*). Hasilnya berisi koordinat bounding box dan skor kepercayaan untuk setiap objek yang terdeteksi.
5. Menerapkan *Non-Maximum Suppression (NMS)* untuk menyaring bounding box yang tumpang tindih berdasarkan skor kepercayaannya (*cv2.dnn.NMSBoxes*).
6. Langkah terakhir menampilkan hasil dengan bounding box dan label menggunakan *cv2.imshow*.

## 2.3. Pengujian dan Evaluasi

### 2.3.1. Pengujian

Pengujian ini dilakukan untuk mengevaluasi kinerja model dalam berbagai kondisi dan menentukan keefektifan serta keandalannya dalam mendeteksi senjata, khususnya pistol, pada rekaman video CCTV. Pengujian dilakukan pada beberapa aspek perubahan gambar yaitu grayscale, rotasi 90°, rotasi 180°, tingkat kecerahan, blur (*gaussian 3px*) dan jarak objek dari sebuah kamera yang akan diuji dengan memisalkan nilai akurasi sebesar 0.25 yang berarti 25% . Uji pendeteksian pada citra grayscale mencapai akurasi yang cukup baik dengan akurasi mencapai angka 60% sampai dengan lebih dari 90% pada pendeteksian banyak objek. Nilai akurasi ini memberikan gambaran tentang seberapa sering model dapat membuat prediksi yang benar dari keseluruhan data yang diuji. Cara mendapatkan nilai akurasi adalah dengan rumus:

$$Akurasi = \frac{Jumlah\ Data\ Benar}{Jumlah\ Seluruh\ Data} \times 100\%$$

Dengan cara ini, nilai akurasi memberikan ukuran yang jelas tentang keandalan model dalam mendeteksi senjata pada berbagai kondisi gambar yang telah diuji. Pengujian yang menyeluruh ini membantu dalam memahami batasan dan kekuatan model, serta memberikan dasar untuk perbaikan lebih lanjut dalam pengembangan model deteksi senjata.

### 2.3.2. Evaluasi

Evaluasi yang digunakan untuk mengevaluasi kinerja sistem deteksi pistol ada beberapa faktor yang dapat mempengaruhi sebuah

objek tidak dapat di deteksi seperti jarak objek terhadap kamera yang terlalu jauh, pencahayaan yang kurang, posisi objek deteksi yang kurang memungkinkan untuk dideteksi, dan objek yang tidak terlalu jelas. Pada aspek deteksi video yang diburamkan, berbagai tingkat blur akan diterapkan pada video untuk mengevaluasi kinerja model dalam mendeteksi objek yang tidak sepenuhnya jelas, mensimulasikan kondisi di mana rekaman mungkin tidak tajam seperti pada kamera pengawas dengan resolusi rendah atau fokus yang buruk. Variasi tingkat kecerahan juga akan diuji dengan menyesuaikan video dengan berbagai tingkat kecerahan, baik lebih terang maupun lebih gelap, untuk memastikan bahwa deteksi tidak terganggu oleh perubahan pencahayaan lingkungan.

### **3. HASIL DAN PEMBAHASAN**

#### **3.1. Penerapan Deteksi Kejahatan**

##### **3.1.1. Perancangan Sistem Perangkat Lunak**

Pada penerapan forensik digital dengan fokus utama pada identifikasi potensi ancaman senjata api menggunakan teknologi pendeteksi objek dirancang untuk menyimpan hasil deteksi sebagai langkah lanjutan yang penting, dengan memanfaatkan sistem pendeteksian senjata tajam yang telah dibuat. Untuk setiap bagian yang mengandung deteksi senjata, program menggambar kotak pembatas di sekitar objek dan menambahkan label "weapon". Bagian tersebut kemudian disimpan sebagai file gambar dengan format nama "detected\_frame\_X.jpg", di mana X adalah nomor frame. Selanjutnya, frame yang sedang diproses ditampilkan pada jendela tampilan, memungkinkan pengguna untuk melihat hasil deteksi secara *realtime*. Sistem ini akan menggunakan algoritma YOLOv3 (You Only Look Once ver. 3) yang dikenal efektif dalam proses deteksi objek.

Penggunaan YOLOv3 pada penelitian ini diterapkan dalam penggunaan skrip berbasis open CV dan pustaka Numpy, serta memuat model YOLOv3 yang telah dilatih sebelumnya. Model YOLOv3 pada penelitian ini diharuskan dilengkapi dengan file bobot ("yolov3\_training\_2000.weights") dan konfigurasi ("yolov3\_testing.cfg"), yang dioptimalkan khusus untuk mendeteksi senjata. Dalam penelitian ini, daftar kelas hanya berisi satu kelas, yaitu "weapon", dan warna acak dihasilkan untuk penanda deteksi visual.

##### **3.1.2. Hasil Implementasi Deteksi Kejahatan**

Deteksi kejahatan dilakukan dengan menggunakan program Python. Hasil pendeteksian kejahatan ditampilkan dalam bentuk kotak deteksi yang mengelilingi pistol atau individu yang membawa pistol. Pada ujung kiri atas kotak tersebut terdapat tulisan "Weapon" dan

angka akurasi. Warna dari kotak deteksi tersebut dapat berubah-ubah dan tidak bisa diatur. Hasil dari implementasi ini dimuat dalam **Gambar 1**.



**Gambar 1.** Hasil implementasi deteksi kejahatan

### **3.2. Penerapan Forensik Digital**

#### **3.2.1. Perancangan Sistem Perangkat Lunak**

Pada penerapan forensik digital dengan fokus utama pada identifikasi potensi ancaman senjata api menggunakan teknologi pendeteksi objek dirancang untuk menyimpan hasil deteksi sebagai langkah lanjutan yang penting dengan memanfaatkan sistem pendeteksian senjata yang telah dibuat. Untuk setiap input yang mengandung senjata api, program akan mendeteksi input tersebut sebagai “weapon”. Hasil deteksi tersebut kemudian disimpan sebagai dokumen foto dengan format nama "detected\_frame\_X.jpg", di mana X adalah nomor bagian. Selanjutnya, bagian yang sedang diproses ditampilkan pada jendela tampilan, memungkinkan pengguna untuk melihat hasil deteksi secara *realtime*.

Perancangan sistem ini melibatkan beberapa komponen utama, termasuk akuisisi video, deteksi objek, dan penyimpanan hasil deteksi. Pada tahap awal, video yang masuk akan diproses frame-by-frame. Setiap frame akan dianalisis menggunakan algoritma pendeteksi objek yang telah dilatih untuk mengenali senjata api. Ketika senjata api terdeteksi dalam sebuah frame, algoritma akan menandai frame tersebut dan menyimpannya dalam format gambar. Selain itu, frame yang sedang diproses juga ditampilkan pada jendela tampilan untuk memungkinkan pengguna melihat hasil deteksi secara real-time. Fitur ini sangat berguna dalam situasi di mana respons cepat diperlukan, seperti dalam operasi keamanan.

Seluruh proses ini dirancang agar efisien dan akurat, memastikan bahwa setiap potensi ancaman dapat diidentifikasi dan ditangani

dengan cepat. Penggunaan teknologi pendeteksi objek yang canggih, dikombinasikan dengan kemampuan penyimpanan hasil deteksi, membuat sistem ini menjadi alat yang efektif dalam forensik digital untuk pencegahan dan penanganan kejahatan.

### 3.2.2. Hasil Implementasi Deteksi dengan Forensik Digital

Implementasi deteksi kejahatan dengan forensik digital dilakukan dengan menggunakan program Python. Hasil akhir dari implementasi deteksi kejahatan dengan forensik digital ini kemudian ditampilkan dalam bentuk output yang mencakup penyimpanan dan pengolahan video atau foto yang terdeteksi. Visualisasi hasil dari implementasi ini dimuat dalam **Gambar 2**. Output penyimpanan video/foto yang terdeteksi, yang memberikan gambaran jelas mengenai bukti-bukti yang telah berhasil diidentifikasi dan dianalisis oleh sistem forensik digital yang telah dikembangkan. Hasil dari implementasi ini dimuat dalam **Gambar 3**. Hasil video/foto terdeteksi yang tersimpan

```
Saved detected_frame_2_9.jpg  
Saved detected_frame_2_10.jpg  
Saved detected_frame_2_11.jpg  
Saved detected_frame_2_12.jpg  
Saved detected_frame_2_13.jpg  
Saved detected_frame_2_14.jpg  
Saved detected_frame_2_15.jpg  
Saved detected_frame_2_16.jpg  
Saved detected_frame_2_17.jpg  
Saved detected_frame_2_18.jpg  
Saved detected_frame_2_19.jpg  
Saved detected_frame_2_20.jpg  
Saved detected_frame_2_21.jpg  
Saved detected_frame_2_22.jpg  
Saved detected_frame_2_23.jpg  
Saved detected_frame_2_24.jpg  
Saved detected_frame_2_25.jpg  
Saved detected_frame_2_26.jpg  
Saved detected_frame_2_27.jpg  
Saved detected_frame_2_28.jpg  
Saved detected_frame_2_29.jpg
```



**Gambar 2.** Output penyimpanan video/foto yang terdeteksi

**Gambar 3.** Hasil video/foto terdeteksi yang tersimpan

## 3.3. Pengujian dan Evaluasi

















### 3.3.1. Pengujian









Pengujian dilakukan pada 12 video diluar dari dataset (train, valid, test) yang diambil secara acak dari platform YouTube. Pengujian dilakukan pada beberapa aspek perubahan gambar yaitu rotasi 90°, rotasi 180°, tingkat kecerahan, ketajaman gambar, dan jarak objek dari sebuah kamera yang akan diuji. Hasil pengujian 12 dataset tersebut



dipaparkan dalam **Tabel 1**. Hasil pengujian dataset **Error! Reference source not found.** sebagai berikut.

**Tabel 1.** Hasil pengujian dataset

No.	Gambar Kejadian	Gambar Deteksi Objek	Akurasi
1.			Terdeteksi ( <i>Confidence:</i> 53%)
2.			Terdeteksi ( <i>Confidence:</i> 82%)
3.			Terdeteksi ( <i>Confidence:</i> 83%)
4.			Terdeteksi ( <i>Confidence:</i> 72%)
5.			Tidak Terdeteksi
6.			Tidak Terdeteksi
7.			Terdeteksi ( <i>Confidence:</i> 52%)
8.			Terdeteksi ( <i>Confidence:</i> 75%)

9.			Tidak Terdeteksi
10.			Terdeteksi ( <i>Confidence:</i> 64%)
11.			Terdeteksi ( <i>Confidence:</i> 68%)
12.			Tidak Terdeteksi

Dari hasil pengujian 12 video, diperoleh 8 video terdeteksi dan 4 video tidak terdeteksi sehingga hasil akurasi adalah:

$$\frac{8}{12} \times 100\% = 66.7\%$$

dengan rata-rata akurasi setiap videonya adalah 68,625%.

### 3.3.2. Evaluasi

Berdasarkan hasil deteksi, diperoleh tingkat akurasi deteksi sebesar 66.7% menunjukkan bahwa metode yang digunakan mampu mendeteksi objek senjata pada sebagian besar video yang diuji. Akan tetapi, masih terdapat 33.3% video yang tidak terdeteksi dengan benar, yang mengindikasikan adanya ruang untuk perbaikan dalam sistem deteksi ini. Faktor-faktor seperti jarak yang terlalu jauh, pencahayaan yang kurang, posisi objek yang kurang memungkinkan, dan objek yang tidak terlalu jelas berkontribusi terhadap kegagalan ini. Jarak yang terlalu jauh membuat detail objek sulit terlihat, sehingga diperlukan peningkatan resolusi dan algoritma penskalaan untuk memperbesar objek kecil. Pencahayaan yang kurang menyebabkan gambar menjadi gelap atau memiliki kontras rendah, sehingga penerangan tambahan dan algoritma peningkatan gambar diperlukan. Evaluasi ini penting untuk meningkatkan deteksi dan teknologi terkait untuk mencapai deteksi yang lebih andal dan efektif dalam berbagai kondisi.

Berikut beberapa langkah perbaikan yang dapat dilakukan dalam upaya meningkatkan akurasi deteksi objek senjata:

- 1) **Peningkatan Kualitas Dataset:** Menggunakan dataset yang lebih besar dan lebih beragam untuk melatih model, mencakup berbagai kondisi pencahayaan, sudut pengambilan gambar, dan variasi objek.
- 2) **Optimasi Algoritma:** Menyesuaikan parameter model dan menggunakan teknik optimasi yang lebih canggih untuk meningkatkan kemampuan deteksi.
- 3) **Praktik Augmentasi Data:** Mengaplikasikan teknik augmentasi data untuk memperkaya variasi data latih, seperti rotasi, perubahan kecerahan, dan penambahan noise.
- 4) **Peningkatan Resolusi Video:** Menggunakan video dengan resolusi lebih tinggi untuk mendapatkan detail yang lebih jelas, yang dapat membantu dalam proses deteksi.

Dengan menerapkan langkah-langkah perbaikan tersebut, diharapkan akurasi deteksi objek senjata pada video CCTV tindakan kriminal dapat meningkat, sehingga sistem menjadi lebih andal dan efektif dalam mendukung penegakan hukum dan keamanan.

### 3.3.3. Perbandingan dengan Penelitian Sebelumnya

Penelitian sebelumnya yang telah dilakukan oleh Imam Maulana, Nining Rahaningsih, dan Tati Suprpti dengan judul “ANALISIS PENGGUNAAN MODEL YOLOV8 (YOU ONLY LOOK ONCE) TERHADAP DETEKSI CITRA SENJATA BERBAHAYA“ dilakukan secara eksperimental menggunakan algoritma YOLOv8 dengan transfer learning. Langkah-langkah yang diambil berupa pengumpulan dan anotasi dataset citra, pra-pemrosesan dan augmentasi citra, pelatihan model menggunakan dataset yang telah dibagi menjadi data pelatihan, validasi, dan pengujian, serta evaluasi hasil pelatihan berdasarkan matrix *precision*, *recall*, mAP, dan F1-Score. Hasil penelitian menunjukkan model YOLOv8 mencapai *precision* sebesar 84%, *recall* 77%, mAP 84%, dan F1-Score 88% dengan waktu pelatihan selama 4 jam 6 menit. Pengujian deteksi pada gambar dengan kondisi yang berbeda menunjukkan hasil yang cukup baik namun perlu peningkatan lebih lanjut untuk hasil yang lebih maksimal. Evaluasi dilakukan menggunakan metrik *precision*, *recall*, mAP, dan F1-Score. Model diuji pada berbagai kondisi gambar seperti grayscale, gambar berotasi, tingkat cahaya berbeda, dan gambar blur untuk mengukur performanya.

Sedangkan dalam penelitian sistem deteksi senjata api ini lebih berfokus pada implementasi deteksi kejahatan melalui perancangan sistem perangkat lunak dan penerapan forensik digital. Metodologi yang digunakan termasuk perancangan sistem perangkat lunak untuk

deteksi kejahatan dan pengujian serta evaluasi hasil implementasi dengan menggunakan model YOLOv3. Hasil implementasi deteksi kejahatan dalam penelitian ini melibatkan perancangan sistem perangkat lunak yang dapat mendeteksi dan mengidentifikasi objek menggunakan forensik digital. Pengujian dilakukan untuk memastikan akurasi dan efisiensi sistem yang telah dirancang. Pengujian melibatkan evaluasi sistem perangkat lunak yang dirancang dengan fokus pada efektivitas deteksi dan akurasi hasil. Hasil yang diperoleh yaitu tingkat akurasi deteksi sebesar 66.7% menunjukkan bahwa metode yang digunakan cukup efektif dalam mendeteksi objek senjata pada sebagian besar video, namun masih diperlukan evaluasi lebih lanjut karena 33.3% video tidak terdeteksi dengan benar. Evaluasi dilakukan untuk memastikan sistem dapat berfungsi sesuai dengan tujuan yang diinginkan yaitu mendeteksi senjata berbahaya berupa pistol sebagai upaya membantu dalam pengawasan dan keamanan.

#### 4. Kesimpulan

Penelitian ini berfokus untuk mengembangkan model *object detection* berbasis pengolahan citra yang mampu mengenali dan mendeteksi senjata tajam dan senjata api (pistol) dari rekaman kamera pengawas menggunakan YOLOv3 dan menyimpan hasil deteksi yang diperoleh. Pengujian dilakukan dengan menggunakan data pelatihan yang melalui proses augmentasi berupa rotasi 90°, rotasi 180°, perubahan tingkat kecerahan dan ketajaman gambar, serta jarak objek dari kamera. Hasil pengujian menunjukkan bahwa terdapat tingkat akurasi yang berkisar antara 50% hingga 85%. Dari 12 video yang diuji, terdapat 4 video yang tidak dapat terdeteksi. Hal ini mengindikasikan bahwa performa sistem belum sempurna dan masih memiliki keterbatasan dalam mengatasi faktor-faktor seperti jarak, pencahayaan, dan kualitas gambar. Untuk pengembangan sistem lebih lanjut, disarankan untuk mengoptimalkan algoritma, meningkatkan kualitas dataset dan resolusi video, serta melakukan praktik augmentasi data.

#### Daftar Pustaka

- Dw Agung Md Krisna Pranata, I., Nahak, S., & Made Minggu Widyantara, I. (2019). Peranan Closed Circuit Television (CCTV) Sebagai Alat Bukti Dalam Persidangan Perkara Pidana. *Jurnal Analogi Hukum*, 1(2). <https://doi.org/10.22225/ah.1.2.1749.163-168>
- Maulana, I., Rahaningsih, N., & Suprpti, T. (2023). ANALISIS PENGGUNAAN MODEL YOLOV8 (YOU ONLY LOOK ONCE) TERHADAP DETEKSI CITRA SENJATA BERBAHAYA. *Jurnal Mahasiswa Teknik Informatika*, 7(6).

- Narejo, S., Pandey, B., Esenarro Vargas, D., Rodriguez, C., & Anjum, M. R. (2021). Weapon Detection Using YOLO V3 for Smart Surveillance System. *Mathematical Problems in Engineering*, 2021. <https://doi.org/10.1155/2021/9975700>
- Indaryanto, F., Nugroho, A., & Suni, A. F. (2021). Aplikasi Penghitung Jarak dan Jumlah Orang Berbasis YOLO Sebagai Protokol Kesehatan Covid-19. *Edu Komputika*, 8(1). <http://journal.unnes.ac.id/sju/index.php/edukom>
- Area, C. N. (2021, 5 13). [https://youtu.be/LbcCFCsEe2Y?si=5Oec2z-y\\_i5u9xMc](https://youtu.be/LbcCFCsEe2Y?si=5Oec2z-y_i5u9xMc)
- DC, F. 5. (2020, 6 15). <https://youtu.be/K3RiJxu-yKU?si=hM9MdQ7yyPehKtZ5>
- edition, i. (2022, 5 14). <https://youtu.be/mmergG4syvk?si=JANcOjR4SM4bG64C>
- Edition, I. (2022, Juni 2). <https://youtu.be/fYp3rxg0OWI?si=CDZAKdVKRC98O15B>
- Edition, I. (2022, 4 21). [https://youtu.be/p8hTG\\_owbqQ?si=YIWouibd19M3Pv2H](https://youtu.be/p8hTG_owbqQ?si=YIWouibd19M3Pv2H)
- Edition, I. (2022, 4 21). [https://youtu.be/p8hTG\\_owbqQ?si=3MjGNd6jImzsuEXf](https://youtu.be/p8hTG_owbqQ?si=3MjGNd6jImzsuEXf)
- Edition, I. (2023, 1 10). <https://youtu.be/bxKYHntHjAg?si=LNzWP9KjXgh3h7e4>
- Globe, T. b. (2023, 5 27). <https://youtu.be/PEyj5mGDqls?si=rnJ8ry48iFjDJgPV>
- philadelphia, F. 2. (2022, 7 29). [https://youtu.be/AAHrzz\\_5Ags?si=P80zdfyJ2RYKDexJ](https://youtu.be/AAHrzz_5Ags?si=P80zdfyJ2RYKDexJ)
- Philadelphia, F. 2. (2022, 7 21). <https://youtu.be/ZIR7FWXpV4E?si=onUMfidkvUxffQxM>

## Lampiran

```
import cv2
import numpy as np
# Memuat Yolo
# Mengunduh file Yolo(yolov3_training_2000.weights) dari link
ini:
https://drive.google.com/file/d/10uJEsUpQI3EmD98iwrwzbd4e19Ps-
LHZ/view?usp=sharing
net = cv2.dnn.readNet("yolov3_training_2000.weights",
"yolov3_testing.cfg")
classes = ["Weapon"]
output_layer_names = net.getUnconnectedOutLayersNames()
colors = np.random.uniform(0, 255, size=(len(classes), 3))
# Memuat foto
def value():
    val = input("Enter file name or press enter to start webcam
: \n")
    if val == "":
        val = 0
    return val
# Mengambil video
cap = cv2.VideoCapture(value())
while True:
    _, img = cap.read()
    if not _:
        print("Error: Failed to read a frame from the video
source.")
        break
    height, width, channels = img.shape
    blob = cv2.dnn.blobFromImage(img, 0.00392, (416, 416), (0,
0, 0), True, crop=False)
    net.setInput(blob)
    outs = net.forward(output_layer_names)
    class_ids = []
    confidences = []
    boxes = []
    for out in outs:
        for detection in out:
            scores = detection[5:]
            class_id = np.argmax(scores)
            confidence = scores[class_id]
            if confidence > 0.5:
                # Mendeteksi objek
                center_x = int(detection[0] * width)
                center_y = int(detection[1] * height)
```

```

        w = int(detection[2] * width)
        h = int(detection[3] * height)
        # Membuat kotak
        x = int(center_x - w / 2)
        y = int(center_y - h / 2)
        boxes.append([x, y, w, h])
        confidences.append(float(confidence))
        class_ids.append(class_id)
    indexes = cv2.dnn.NMSBoxes(boxes, confidences, 0.5, 0.4)
    print(indexes)
    if indexes == 0: print("weapon detected in frame")
    font = cv2.FONT_HERSHEY_PLAIN
    for i in range(len(boxes)):
        if i in indexes:
            x, y, w, h = boxes[i]
            label = str(classes[class_ids[i]])
            color = colors[class_ids[i]]
            cv2.rectangle(img, (x, y), (x + w, y + h), color, 2)
            cv2.putText(img, label, (x, y + 30), font, 3, color,
3)
            cv2.imshow("Image", img)
            key = cv2.waitKey(1)
            if key == 27:
                break
    cap.release()
    cv2.destroyAllWindows()

# Menyimpan dokumen foto
import cv2
import numpy as np
# Memuat Yolo
net = cv2.dnn.readNet("yolov3_training_2000.weights",
"yolov3_testing.cfg")
classes = ["Weapon"]
output_layer_names = net.getUnconnectedOutLayersNames()
colors = np.random.uniform(0, 255, size=(len(classes), 3))
def value():
    val = input("Enter file name or press enter to start
webcam: \n")
    if val == "":
        val = 0
    return val
# Mengambil video
cap = cv2.VideoCapture(value())
frame_number = 0

```

```

while True:
    ret, img = cap.read()
    if not ret:
        print("Error: Failed to read a frame from the video
source.")
        break
    height, width, channels = img.shape
    # Mendeteksi objek
    blob = cv2.dnn.blobFromImage(img, 0.00392, (416, 416), (0,
0, 0), True, crop=False)
    net.setInput(blob)
    outs = net.forward(output_layer_names)
    # Menampilkan informasi pada layar
    class_ids = []
    confidences = []
    boxes = []
    for out in outs:
        for detection in out:
            scores = detection[5:]
            class_id = np.argmax(scores)
            confidence = scores[class_id]
            if confidence > 0.5:
                # Objek terdeteksi
                center_x = int(detection[0] * width)
                center_y = int(detection[1] * height)
                w = int(detection[2] * width)
                h = int(detection[3] * height)
                # Membuat kotak
                x = int(center_x - w / 2)
                y = int(center_y - h / 2)
                boxes.append([x, y, w, h])
                confidences.append(float(confidence))
                class_ids.append(class_id)
    indexes = cv2.dnn.NMSBoxes(boxes, confidences, 0.5, 0.4)
    font = cv2.FONT_HERSHEY_PLAIN
    weapon_detected = False
    for i in range(len(boxes)):
        if i in indexes:
            x, y, w, h = boxes[i]
            label = str(classes[class_ids[i]])
            color = colors[class_ids[i]]
            cv2.rectangle(img, (x, y), (x + w, y + h), color,
2)
            cv2.putText(img, label, (x, y + 30), font, 3,
color, 3)

```



```
        weapon_detected = True

# Mrnyimpan dokumen jika senjata terdeteksi
if weapon_detected:
    frame_name = f"detected_frame_{frame_number}.jpg"
    cv2.imwrite(frame_name, img)
    print(f"Saved {frame_name}")
# Menampilkan dokumen yang tersimpan
cv2.imshow("Image", img)
frame_number += 1
key = cv2.waitKey(1)
if key == 27:
    break
cap.release()
cv2.destroyAllWindows()
```