

Sistem Deteksi Objek Senjata Tajam Secara *Real-Time* Menggunakan YOLOv8n *Deep Learning*

Nadziah Fitriani¹, Arif Fadllullah^{1*}

¹Jurusan Teknik Komputer, Universitas Borneo Tarakan, Tarakan, Kalimantan Utara, Indonesia

e-mail: nadziahfitriani@gmail.com, arif.fadl@borneo.ac.id

Diterima
04-04-2026

Direvisi
13-04-2026

Disetujui
18-04-2026

Abstract: This research is motivated by the increasing potential for crimes involving the use of sharp weapons and the limitations of manual surveillance systems in detecting threats quickly and accurately. This condition indicates the need for an automated system based on deep learning capable of real-time object detection. This study aims to develop a sharp weapon detection system using the YOLOv8n algorithm with a focus on three types of objects: knives, sickles, and machetes. The Dataset used consists of 1,200 images (600 primary images and 600 secondary images) which are processed through annotation and augmentation stages (saturation, brightness, and resizing) to produce 2,640 images. The Dataset is then divided into 2,160 training data, 240 validation data, and 240 test data. The model is trained for 100 epochs using the training data. The training results show a precision value of 0.865 and a recall of 0.815. Evaluation on test data yielded a precision of 0.837, a recall of 0.819, and an F1-score of 0.827, with the best performance in the crescent class with an F1-score of 0.924. Real-time testing at a distance of 1–4 meters showed the system was able to detect objects responsively with an average F1-score of 0.790.

Keywords: Sharp Weapon Detection, YOLOv8n, Deep Learning, Object Detection, Real-Time

Abstrak: Penelitian ini dilatarbelakangi oleh meningkatnya potensi tindak kriminal yang melibatkan penggunaan senjata tajam serta keterbatasan sistem pengawasan manual dalam mendeteksi ancaman secara cepat dan akurat. Kondisi ini menunjukkan perlunya sistem otomatis berbasis *deep learning* yang mampu melakukan deteksi objek secara *real-time*. Penelitian ini bertujuan untuk mengembangkan sistem deteksi senjata tajam menggunakan algoritma YOLOv8n dengan fokus pada tiga jenis objek, yaitu pisau, sabit, dan parang. *Dataset* yang digunakan terdiri dari 1.200 citra (600 citra primer dan 600 citra sekunder) yang diproses melalui tahap anotasi dan augmentasi (saturasi, kecerahan, dan *resize*) sehingga menghasilkan 2.640 citra. *Dataset* kemudian dibagi menjadi 2.160 data latih, 240 data validasi, dan 240 data uji. Model dilatih selama 100 *epoch* menggunakan data latih. Hasil pelatihan menunjukkan nilai *precision* sebesar 0,865 dan *recall* sebesar 0,815. Evaluasi pada data uji menghasilkan *precision* 0,837, *recall* 0,819, dan *F1-score* 0,827, dengan performa terbaik pada kelas sabit dengan *F1-score* sebesar 0,924. Pengujian *real-time* pada jarak 1–4 meter menunjukkan sistem mampu mendeteksi objek secara responsif dengan rata-rata *F1-score* 0,790.

Kata kunci: Sharp Weapon Detection, YOLOv8n, Deep Learning, Object Detection, Real-Time

I. PENDAHULUAN

Tindak kriminal merupakan permasalahan sosial global yang seringkali melibatkan penggunaan senjata berbahaya, khususnya senjata tajam. Salah satu bentuk kriminalitas yang meresahkan adalah tindak kejahatan begal (Maulana dkk., 2024), yaitu tindakan mengambil barang milik orang lain secara paksa disertai ancaman atau kekerasan (Nugraha, 2023). Ancaman hukum pelaku terhadap tindak kriminal diatur dalam Kitab Undang-Undang Hukum Pidana (KUHP), dengan tingkat hukuman yang bervariasi tergantung jenis dan tingkat kejahatan yang dilakukan. Tingginya angka kriminalitas tersebut menunjukkan perlunya sistem pengawasan yang lebih efektif dalam mendeteksi potensi ancaman sejak dini (Kitab Undang-Undang Hukum Pidana, 2021).

Salah satu upaya preventif yang umum dilakukan adalah penggunaan CCTV sebagai alat pemantauan. Meskipun CCTV mampu merekam kejadian sebagai bukti, sistem ini memiliki keterbatasan karena bersifat pasif, tidak mampu memberikan peringatan secara *real-time*, serta kualitas citra yang tidak selalu optimal (Pradana dkk., 2022). Kasus pembegalan di Kota Baubau (Jamil Azali, 2024) menunjukkan bahwa meskipun kejadian terekam dengan jelas, sistem CCTV belum mampu mencegah kejadian maupun memberikan respon cepat terhadap ancaman. Hal ini

menunjukkan adanya kesenjangan antara kemampuan sistem pengawasan konvensional dengan kebutuhan deteksi dini yang responsif (Kojongian dkk, 2024).

Oleh sebab itu, perlunya penerapan teknologi yang lebih canggih untuk mengidentifikasi tindak kejahatan dan memberikan peringatan dini secara otomatis ketika terdeteksi penggunaan senjata tajam dan menginformasikan kepada pihak yang berwenang (Aziz dkk., 2024). Dalam era perkembangan teknologi yang pesat ini, *Computer Vision* dan *Object Detection* sudah menjadi bidang yang sangat penting (Dompeipen dkk, 2020). *Computer Vision* berfokus pada pengembangan sistem yang mampu memahami dan menafsirkan informasi visual seperti manusia. *Object Detection*, sebagai bagian dari *Computer Vision*, teknologi ini memungkinkan komputer untuk mendeteksi, mengidentifikasi dan mengenali objek yang terdapat dalam gambar atau video secara otomatis (Sutisna dkk., 2024).

Salah satu algoritma yang banyak digunakan adalah YOLO (*You Only Look Once*) yang dikenal memiliki keseimbangan antara kecepatan dan akurasi deteksi (Liunanda dkk., 2020). Meskipun sudah ada beberapa penelitian yang membahas deteksi senjata tajam menggunakan YOLO, masih terdapat beberapa tantangan yang perlu diatasi. Seperti penelitian yang dilakukan oleh (Sholahuddin dkk., 2023) berhasil mendeteksi pistol dengan akurasi tinggi, namun hanya fokus pada satu jenis senjata. Penelitian lain yang dilakukan oleh (Maulana dkk., 2024) menunjukkan bahwa akurasi deteksi senjata tajam menggunakan YOLOv8 sangat dipengaruhi oleh kualitas gambar dan jarak objek dari kamera. Selain itu penelitian yang dilakukan oleh (Pradana dkk., 2022) menekankan pentingnya penggunaan *Dataset* yang bervariasi dan teknik *transfer learning* untuk meningkatkan akurasi deteksi. Hal ini menunjukkan bahwa faktor kondisi lingkungan nyata, seperti pencahayaan, resolusi, dan jarak objek, memiliki pengaruh langsung terhadap kinerja sistem deteksi.

Berdasarkan analisis tersebut, dapat diidentifikasi hubungan sebab-akibat bahwa pada variasi kondisi nyata, khususnya jarak objek terhadap kamera dan kualitas citra, berpengaruh terhadap penurunan atau peningkatan akurasi deteksi objek senjata tajam. Namun, penelitian yang secara spesifik mengkaji deteksi multi-objek senjata tajam dengan mempertimbangkan variasi kondisi tersebut masih terbatas. Oleh karena itu diperlukan sistem pengembangan yang tidak hanya mampu mendeteksi objek secara *real-time*, tetapi juga mampu mempertahankan kinerja pada berbagai kondisi operasional.

Dengan demikian, Hipotesis yang ada dalam penelitian ini diharapkan bahwa sistem yang dibangun mampu mendeteksi senjata tajam dengan akurasi yang telah ditentukan, memberikan peringatan dini terhadap potensi ancaman. Berdasarkan hal tersebut, penelitian ini bertujuan untuk merancang dan membangun sistem deteksi otomatis objek senjata tajam berbasis *deep learning* menggunakan algoritma YOLOv8n. Sistem yang dikembangkan diharapkan mampu mendeteksi lebih dari satu jenis senjata tajam secara *real-time* dan dapat diimplementasikan pada lingkungan publik sebagai sistem peringatan dini guna meningkatkan keamanan

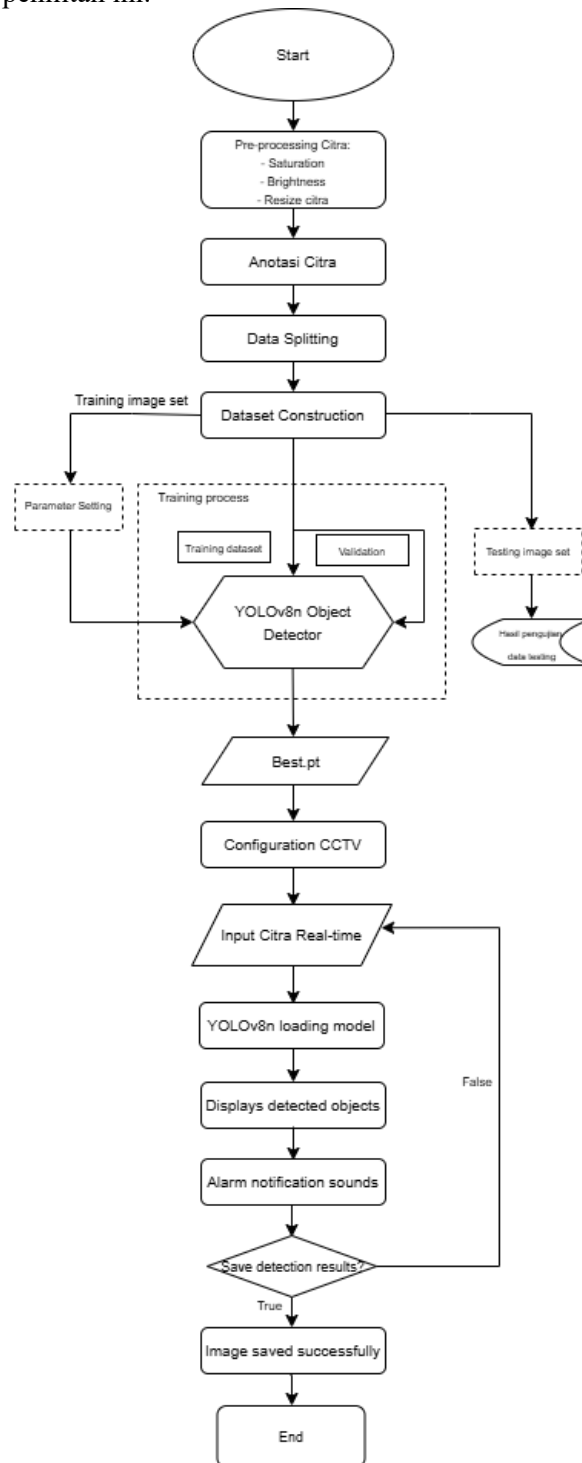
II. METODE PENELITIAN

Penelitian ini menggunakan pendekatan eksperimental untuk membangun sistem deteksi senjata tajam berbasis *deep learning* dengan algoritma YOLOv8n. Penelitian eksperimental dipilih karena tujuan utama penelitian adalah mengembangkan sebuah sistem baru dan menyesuaikan sistem sesuai dengan kebutuhan yang spesifik yaitu mendeteksi secara otomatis penggunaan senjata tajam berdasarkan algoritma YOLOv8n secara *real-time*. Dengan menggunakan jenis penelitian eksperimental, peneliti dapat menguji berbagai parameter dan konfigurasi model YOLOv8n untuk menemukan kombinasi yang menghasilkan kinerja terbaik dalam mendeteksi senjata tajam. Hal ini meliputi akurasi deteksi, kecepatan pemrosesan, dan kemampuan generalisasi model.

1. Alur Perancangan Sistem

Alur perancangan sistem atau biasa disebut Flowchart merupakan suatu metode yang digunakan untuk merepresentasikan alur logika dalam proses penyelesaian masalah. Dengan demikian, Flowchart menjadi gambaran urutan langkah penyelesaian masalah yang disajikan

melalui simbol-simbol visual yang mudah dipahami (Tuasamu dkk., 2023). Gambar 1 merupakan Flowchart sistem pada penelitian ini.



Gambar 1. Flowchart Sistem

Flowchart sistem pada penelitian ini menggambarkan alur kerja perancangan dan implementasi sistem deteksi otomatis penggunaan senjata tajam berbasis *deep learning* menggunakan algoritma YOLOv8n. Proses diawali dengan pengumpulan *Dataset* citra senjata tajam yang terdiri dari tiga kelas objek, yaitu pisau, sabit, dan parang. *Dataset* yang telah

dikumpulkan kemudian melalui tahap *preprocessing* citra yang meliputi penyesuaian *resize* (ukuran citra), *saturation* (kehangatan), dan *brightness* (kecerahan).

Selanjutnya dilakukan proses anotasi citra dengan pemberian *bounding box* pada setiap objek senjata tajam sesuai dengan kelasnya. *Dataset* yang telah dianotasi kemudian dibagi ke dalam tiga bagian (data *splitting*), yaitu data latih (*training*), validasi data (*validation*), dan data uji (*testing*). Data latih dan validasi data digunakan dalam proses pelatihan model YOLOv8n dengan memanfaatkan teknik *transfer learning* dari model pelatihan YOLOv8n. Pada tahap ini dilakukan pengaturan parameter pelatihan serta evaluasi kinerja model menggunakan metrik *precision*, *recall*, dan *F1-Measure* untuk mendapatkan model terbaik. Selanjutnya dilakukan pengujian menggunakan data *testing*. Tujuan dilakukannya pengujian ini adalah untuk mengetahui kemampuan generalisasi model YOLOv8n terhadap data yang belum pernah dilihat sebelumnya pada saat proses pelatihan (Budi, 2021). Selain itu, hasil dari proses pengujian data pengujian ini secara otomatis akan tersimpan ke dalam *path*.

Setelah model terbaik diperoleh, tahapan dilanjutkan ke implementasi sistem. Proses dimulai dari konfigurasi CCTV sebagai sumber input video, kemudian sistem menerima input citra secara *real-time*. Model YOLOv8n yang telah dilatih sebelumnya dimuat menggunakan file *best.pt* untuk melakukan proses deteksi objek pada setiap *frame* yang masuk. Hasil deteksi ditampilkan pada layar dalam bentuk pembatas kotak dan label kelas objek yang teridentifikasi. Apabila sistem mendeteksi adanya objek senjata tajam, maka notifikasi alarm akan berbunyi sebagai bentuk peringatan dini. Selain itu, sistem juga menyimpan hasil deteksi sebagai dokumentasi, yang ditandai dengan notifikasi bahwa gambar berhasil disimpan. Seluruh rangkaian proses ini berakhir pada tahap *End*, yang menunjukkan bahwa sistem telah berjalan sesuai dengan alur yang dirancang dalam penelitian.

2. Teknik Pengumpulan Data

Penelitian ini dilakukan di Warung Mama Mia yang berlokasi di Kelurahan Lingkas Ujung. Pada penelitian ini dilakukan pengumpulan data dibagi menjadi dua bagian yaitu, data *training* dan data *testing*.

A. Data Training

Teknik pengumpulan data *training* dalam penelitian sebanyak 1.200 citra ini dilakukan melalui dua sumber utama, yaitu data primer dan data sekunder. Data primer diperoleh secara langsung melalui proses pengambilan citra menggunakan kamera CCTV tipe EZVIZ H7C yang dipasang di lokasi penelitian sebanyak 600 citra. Pengambilan data dilakukan dengan mempertimbangkan variasi kondisi nyata, seperti perbedaan jarak objek, sudut pengambilan gambar, serta kondisi pencahayaan. Hal ini bertujuan untuk memperoleh data yang merepresentasikan situasi lingkungan sebenarnya sehingga model yang dikembangkan memiliki kemampuan generalisasi yang lebih baik. Sementara itu, data sekunder diperoleh dari <https://id.pinterest.com/> sebanyak 600 citra. Penggunaan data sekunder ini bertujuan untuk memperkaya variasi *Dataset* serta mengurangi risiko bias data yang dapat mempengaruhi kinerja model (Nazar, 2024).

Seluruh data yang telah dikumpulkan kemudian melalui tahap konstruksi *Dataset*, yang meliputi proses akuisisi data, anotasi objek menggunakan pembatas kotak, serta augmentasi data seperti penyesuaian kecerahan, saturasi, dan *resize* akan diproses melalui platform *Roboflow*. Tahapan ini bertujuan untuk meningkatkan kualitas dan variasi *Dataset* sehingga model menjadi lebih kokoh terhadap perubahan kondisi lingkungan serta mampu mendukung proses pelatihan secara optimal dalam meningkatkan akurasi deteksi objek senjata tajam secara *real-time* (Hayati dkk., 2023).

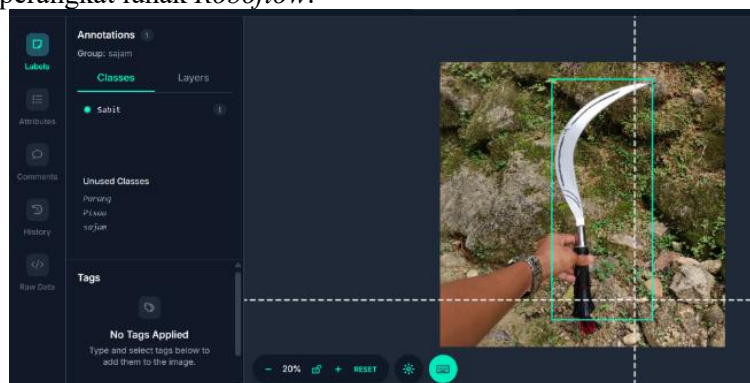
B. Data Testing

Untuk mengevaluasi kinerja model, peneliti menggunakan *Dataset* uji yang terdiri dari 90 *frame* yang dikumpulkan secara *primer* dalam jarak 1-4 meter. *Dataset* ini akan mencakup tiga kelas objek yang menjadi target deteksi yaitu pisau, parang, dan sabit. Hasil deteksi model pada *Dataset* uji akan dievaluasi menggunakan metrik *precision*, *recall*, dan *F1-Score*. Metrik-metrik ini akan memberikan gambaran yang lebih jelas mengenai kemampuan model dalam

mengklasifikasikan objek dengan benar, menemukan semua objek yang ada, serta keseimbangan antara keduanya.

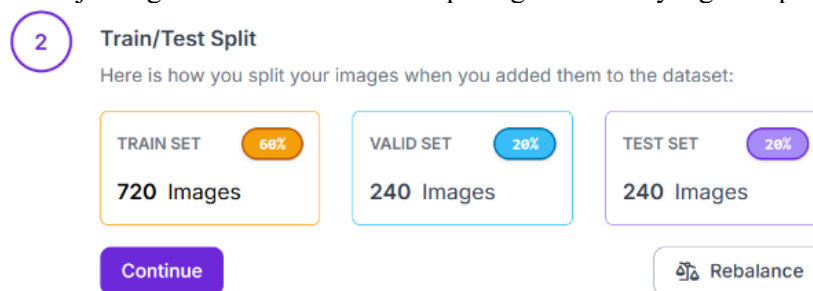
3. Kontruksi *Dataset*

Proses konstruksi *Dataset* dilakukan terhadap 1.200 citra yang diperoleh dari data primer dan sekunder dengan fokus pada tiga objek senjata tajam, yaitu parang, pisau, dan sabit. Akuisisi data merupakan tahapan pengumpulan informasi visual dari lingkungan nyata yang dikonversi ke dalam bentuk digital untuk diolah oleh sistem komputer (Mandela dkk., 2025). *Dataset* yang telah dikumpulkan akan diproses melalui platform *Roboflow*. Selanjutnya anotasi data merupakan aktifitas pelabelan pada setiap gambar dengan membuat *bounding box* dan memberi nama kelas (pisau, parang, dan sabit), serta modifikasi gambar untuk meningkatkan akurasi deteksi oleh YOLOv8n. Tujuan anotasi data adalah untuk memperkaya informasi yang terkandung dalam setiap gambar dengan memberikan label atau keterangan tambahan (Maulana dkk., 2024). Proses anotasi citra dilakukan terhadap total 1.200 gambar yang telah dikumpulkan dengan memanfaatkan perangkat lunak *Roboflow*.



Gambar 2. Proses Anotasi Data

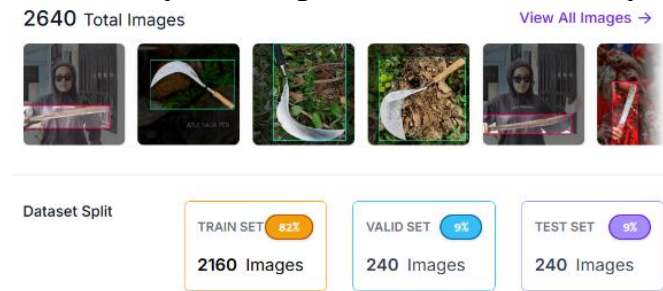
Selanjutnya proses Augmentasi data yang merupakan metode untuk memperkaya *Dataset* pelatihan dengan cara memanipulasi data yang tersedia secara artifisial (Mandela dkk., 2025). Augmentasi yang digunakan diantaranya *resize* (mengubah ukuran data), *saturation* (kehangatan), dan *brightness* (kecerahan). *Dataset* yang telah di di augmentasi, akan dibagi ke dalam tiga bagian (data *splitting*), yaitu data latih, data validasi, dan data uji. Pembagian *Dataset* dilakukan melalui platform *Roboflow* dengan proporsi 60% data pelatihan, 20% validasi, dan 20% pengujian. Rasio ini dipilih untuk memberikan ruang yang optimal bagi YOLOv8n dalam mempelajari fitur senjata tajam, sekaligus memungkinkan pemantauan *overfitting* atau *underfitting* selama proses latih. Hasilnya, performa akhir model dapat dievaluasi secara independen dan objektif guna memastikan kemampuan generalisasi yang baik pada data baru.



Gambar 3. Data Splitting

Setelah melalui proses augmentasi, jumlah total citra bertambah menjadi 2.640 gambar. Proporsi *Dataset* kemudian disesuaikan kembali, dengan alokasi sebesar 82% atau 2160 gambar digunakan untuk pelatihan, sementara validasi data dan pengujian data masing-masing terdiri dari

9% atau 240 gambar. Proses pembagian proporsi *Dataset* setelah augmentasi dapat dilihat pada Gambar 4. *Dataset* tersebut diimpor ke Google Colab untuk melakukan pelatihan data



Gambar 4. Jumlah Dataset Setelah Augmentasi

Perubahan proporsi *Dataset* terjadi karena proses augmentasi yang dikhususkan pada data latih, sehingga jumlahnya meningkat lebih signifikan dibandingkan validasi data dan uji. Selain itu, platform *Roboflow* melakukan pembagian ulang secara otomatis untuk menjaga distribusi data tetap seimbang, sehingga proporsi akhir menjadi sekitar 82% data latih dan masing-masing 9% pada data validasi dan data uji.

4. Konfigurasi dan Pelatihan YOLOv8n

Pada tahap ini, konfigurasi dilakukan pada Google Colab meliputi pengecekan Google Colab, instalasi pustaka yang diperlukan, impor *Dataset*, konfigurasi *hyperparameter* YOLOv8 dan pelatihan. Tahap awal persiapan data meliputi konfigurasi jalur *Dataset* pada Google Drive sebagai direktori utama serta pengaturan file YAML. Konfigurasi ini mencakup penentuan jalur *Dataset*, jumlah kelas (nc), dan nama kelas yang dideteksi. Pada penelitian ini, nilai nc ditetapkan menjadi 3 untuk mendeteksi kelas pisau, parang, dan sabit. Konfigurasi file YAML dapat dilihat pada Gambar 5.

```

Notebook data.yaml X
1 names:
2 - Parang
3 - Pisau
4 - Sabit
5 nc: 3
6 roboflow:
7   license: CC BY 4.0
8   project: detection-im5j9
9   url: https://universe.roboflow.com/ziahbansir/detection-im5j9/dataset/3
10  version: 3
11  workspace: ziahbansir
12 test: /content/drive/MyDrive/detection-object2/test/images
13 train: /content/drive/MyDrive/detection-object2/train/images
14 val: /content/drive/MyDrive/detection-object2/valid/images
15

```

Gambar 5. Konfigurasi file YAML

Setelah konfigurasi *Dataset* melalui file YAML selesai, berikutnya adalah menyiapkan lingkungan perangkat lunak dengan mengekstrak atau memanggil *Dataset* dari *Roboflow* dengan memasukkan API Key sebagaimana ditunjukkan pada Gambar 6.

```

!pip install roboflow

from roboflow import Roboflow
rf = Roboflow(api_key="odZClcbuesUue6u1p0Fn")
project = rf.workspace("ziahbansir").project("detection-im5j9")
version = project.version(3)
dataset = version.download("yolov8")

Collecting roboflow
  Downloading roboflow-1.2.11-py3-none-any.whl.metadata (9.7 kB)
Requirement already satisfied: certifi in /usr/local/lib/python3.12/dist-packages (from roboflow) (2025.11.12)
Collecting idna==3.7 (from roboflow)

```

Gambar 6. Mengekstrak Dataset dari *Roboflow*

Setelah mengekstrak *Dataset* dari *Roboflow*, berikutnya menginstal pustaka yang diperlukan. Pustaka utama dalam penelitian ini adalah *Ultralytics*, yang merupakan modul *Python* resmi dari pengembang model *YOLOv8*. Instalasi pustaka dilakukan menggunakan perintah *pip install ultralytics*. Dengan diinstalnya modul ini, seluruh lingkungan kerja telah siap untuk melanjutkan ke tahap eksekusi pelatihan. Proses instalasi ditampilkan pada Gambar 7.

```

pip install ultralytics

Collecting ultralytics
  Downloading ultralytics-8.3.218-py3-none-any.whl.metadata (37 kB)
Requirement already satisfied: numpy>=1.23.0 in /usr/local/lib/python3.12/dist-packages (from ultralytics)
Requirement already satisfied: matplotlib>=3.3.0 in /usr/local/lib/python3.12/dist-packages (from ultralyti
Requirement already satisfied: opencv-python>=4.6.0 in /usr/local/lib/python3.12/dist-packages (from ultral
Requirement already satisfied: pillow>=7.1.2 in /usr/local/lib/python3.12/dist-packages (from ultralytics)
Requirement already satisfied: pyyaml>=5.3.1 in /usr/local/lib/python3.12/dist-packages (from ultralytics)
Requirement already satisfied: requests>=2.23.0 in /usr/local/lib/python3.12/dist-packages (from ultralytic

```

Gambar 7. Instalasi Modul *Ultralytics*

Pada penelitian ini digunakan *YOLOv8n (nano)* karena memiliki keseimbangan antara akurasi dan kecepatan untuk deteksi *real-time*. Arsitekturnya memungkinkan pendeteksian objek senjata tajam (parang, pisau, sabit), termasuk objek kecil atau tipis, dengan presisi tinggi tanpa memerlukan sumber daya komputasi yang besar. Oleh karena itu, *YOLOv8n (nano)* dipilih sebagai model yang efektif untuk mendukung sistem implementasi dalam penelitian ini.

Sebelumnya, peneliti juga telah melakukan eksperimen menggunakan varian model *YOLOv8* lainnya, yaitu *YOLOv8s (small)* dan *YOLOv8m (medium)*. Meskipun kedua model tersebut menunjukkan hasil akurasi dan kinerja prediksi yang cukup tinggi pada tahap pelatihan dan evaluasi, implementasi pada pengujian *real-time* menunjukkan adanya kendala pada aspek kecepatan pemrosesan. Sistem mengalami *delay* yang signifikan, dengan waktu respon mencapai 10–30 detik, sehingga tidak memenuhi kebutuhan deteksi secara *real-time*. Selain itu, beban komputasi yang lebih tinggi juga berdampak pada penurunan kualitas visual hasil deteksi, di mana tampilan citra menjadi kurang optimal atau terdistorsi. Oleh karena itu, model *YOLOv8n* dipilih karena mampu memberikan keseimbangan yang lebih baik antara akurasi dan kecepatan, sehingga lebih sesuai untuk implementasi sistem deteksi secara *real-time*.

Setelah model *YOLOv8n* dipilih, proses pelatihan dilakukan menggunakan seluruh *Dataset* selama 100 *epoch* dengan ukuran *batch* 16. Pengaturan ini dipilih untuk menyeimbangkan efisiensi komputasi dan kinerja model, di mana 100 *epoch* dianggap cukup karena model telah mencapai kondisi konvergen. Pemilihan jumlah *epoch* dan *batch* ini juga bertujuan untuk mencegah *overfitting* maupun *underfitting*, yang bergantung pada kompleksitas *Dataset* dan kapasitas model.

5. Teknik Analisis Data

Untuk mengevaluasi kebenaran hasil prediksi dari sistem yang diusulkan, tiga metrik perhitungan akan digunakan, yaitu *precision*, *recall*, dan *F-Measure*. *Precision* merupakan metrik evaluasi yang digunakan untuk mengukur tingkat presisi model dalam memprediksi kelas positif (Clara dkk., 2021). Nilai *precision* menunjukkan proporsi prediksi positif yang benar (*true positive*) terhadap seluruh prediksi positif yang dihasilkan model, sehingga metrik ini penting untuk meminimalkan kesalahan berupa *false positive*. *Recall* merupakan metrik yang digunakan untuk mengukur kemampuan model dalam mengidentifikasi seluruh data positif yang sebenarnya. Nilai *recall* menunjukkan proporsi data positif yang berhasil terdeteksi (*true positive*) terhadap seluruh data positif yang ada, sehingga memainkan penting dalam meminimalkan *false negative*.

Sementara itu, *F-Measure (F1-score)* merupakan rata-rata harmonik dari *precision* dan *recall* yang digunakan untuk memberikan evaluasi kinerja model secara menyeluruh. Metrik ini memberikan gambaran yang lebih seimbang terhadap kinerja klasifikasi dibandingkan penggunaan *presisi* atau *recall* secara terpisah.

$$Precision = \frac{TP}{TP+FP} \quad (1)$$

$$Recall = \frac{TP}{TP+FN} \quad (2)$$

$$F - measure = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (3)$$

Evaluasi kinerja sistem dilakukan menggunakan tiga metrik utama, yaitu *precision*, *recall* dan *F1-score* yang diterapkan pada data uji dan uji data *real-time*. Perhitungan metrik ini didasarkan pada nilai *True Positive* (TP) yang merupakan jumlah prediksi yang salah dalam kelas positif, *False Positive* (FP) merupakan jumlah prediksi yang salah dalam kelas negatif, dan *False Negative* (FN) merupakan jumlah prediksi yang benar dalam kelas positif (Ainun dkk., 2023). Analisis terhadap metrik tersebut digunakan untuk menilai efektivitas sistem dalam mendeteksi objek secara akurat.

III. HASIL DAN PEMBAHASAN

1. Pelatihan *Dataset*

Penelitian ini melakukan proses pelatihan dengan menggunakan seluruh *Dataset* selama 100 *epoch* dengan ukuran *batch* 16 untuk menyesuaikan parameter internalnya sehingga mampu mengenali pola dan fitur objek dengan lebih akurat. Pada penelitian ini, model YOLOv8n berhasil dilatih secara penuh dan menyelesaikan pelatihan pada *epoch* 100. Detail *script* pelatihan ditunjukkan pada Gambar 8.

```
import os
from ultralytics import YOLO

!yolo task=detect mode=train epochs=100 data=/content/drive/MyDrive/detection-object2/data.yaml model=yolov8n.yaml imgsz=640 batch=16
```

Gambar 8. *Script* Pelatihan Data

Setelah proses pelatihan selesai, model secara otomatis melanjutkan ke tahap validasi, yaitu proses evaluasi terhadap performa model yang telah dilatih. Tahapan ini bertujuan untuk memilih model terbaik berdasarkan hasil pelatihan yang menunjukkan tingkat akurasi dan kinerja paling optimal. Validasi data dilakukan sebagai langkah penting untuk memastikan bahwa hasil pelatihan model tidak mengalami bias atau kesalahan interpretasi akibat ketidaksesuaian data. Melalui proses ini, performa model yang telah dilatih dapat dievaluasi kembali sebelum digunakan sebagai model akhir. Gambar 9. menunjukkan hasil validasi yang dilakukan setelah seluruh proses pelatihan data selesai.

```
100 epochs completed in 1.221 hours.
Optimizer stripped from /content/runs/detect/train/weights/last.pt, 6.2MB
Optimizer stripped from /content/runs/detect/train/weights/best.pt, 6.2MB

Validating /content/runs/detect/train/weights/best.pt...
Ultralytics 8.3.235 Python-3.12.12 torch-2.9.0+cu126 CUDA:0 (Tesla T4, 15095MiB)
YOLOv8n summary (fused): 72 layers, 3,006,233 parameters, 0 gradients, 8.1 GFLOPs

```

Class	Images	Instances	Box(P)	R	mAP50	mAP50-95)
all	240	248	0.865	0.815	0.895	0.649
Parang	80	83	0.808	0.761	0.839	0.593
Pisau	72	74	0.81	0.784	0.901	0.629
Sabit	88	91	0.976	0.901	0.945	0.726

```
Speed: 0.3ms preprocess, 2.4ms inference, 0.0ms loss, 4.0ms postprocess per image
Results saved to /content/runs/detect/train
Learn more at https://docs.ultralytics.com/modes/train
```

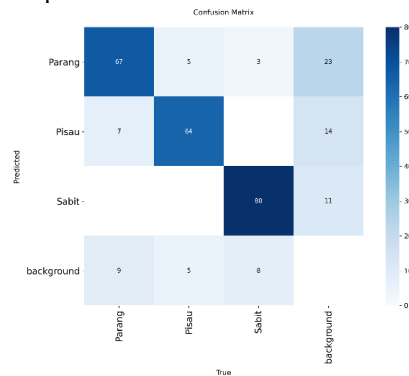
Gambar 9. Hasil Validasi Data

Hasil validasi model YOLOv8n menunjukkan kinerja yang cukup baik dalam mendeteksi objek senjata tajam pada validasi data. Nilai *precision* sebesar 0,865 menunjukkan bahwa sebagian besar prediksi yang dihasilkan model adalah benar, sedangkan nilai *recall* sebesar 0,815 mengindikasikan bahwa model mampu mendeteksi sebagian besar objek yang ada meskipun masih terdapat beberapa objek yang terlewat. Nilai mAP50 sebesar 0,895 menunjukkan tingkat akurasi yang tinggi pada ambang *Intersection over Union* (IoU) $\geq 0,5$, yaitu ukuran kesesuaian antar *bounding box* prediksi dan *ground truth*, di mana semakin tinggi nilai IoU maka semakin akurat posisi deteksi. Sementara itu, nilai mAP50-95 sebesar 0,649 yang dihitung pada rentang IoU 0,5 hingga 0,95 menunjukkan bahwa performa model menurun pada evaluasi yang lebih ketat, sehingga masih terdapat ketidaktepatan dalam penentuan lokasi *bounding box*. Berdasarkan hasil per kelas, model menunjukkan performa terbaik pada kelas sabit dengan nilai *mAP50*

sebesar 0,945, diikuti oleh pisau sebesar 0,901 dan parang sebesar 0,839, yang mengindikasikan bahwa karakteristik objek visual mempengaruhi kemampuan deteksi model. Secara keseluruhan, hasil ini menunjukkan bahwa model telah mampu mendeteksi objek dengan baik, namun masih memerlukan peningkatan dalam aspek presisi lokasi untuk mencapai kinerja yang lebih optimal.

2. Evaluasi Hasil Model YOLOv8n

Evaluasi model YOLOv8n dilakukan untuk memastikan bahwa sistem yang dikembangkan memiliki kinerja yang andal dan dapat diterapkan pada kondisi nyata. *Confusion Matrix* merupakan metrik evaluasi yang digunakan untuk menganalisis kinerja model dalam mengklasifikasikan objek. Matriks ini menampilkan perbandingan antara kelas sebenarnya dan kelas yang diprediksi oleh model (Sholahuddin dkk., 2023). *Confusion Matrix* yang menunjukkan hasil evaluasi performa model YOLOv8 dalam mendeteksi dan mengklasifikasikan objek Parang, Pisau, dan Sabit. Visualisasi *Confusion Matrix* dari model yang dihasilkan dapat dilihat pada Gambar 10.



Gambar 10. Hasil Confusion Matrix Data Validasi

Tabel 1. Metrik hasil evaluasi data Validasi

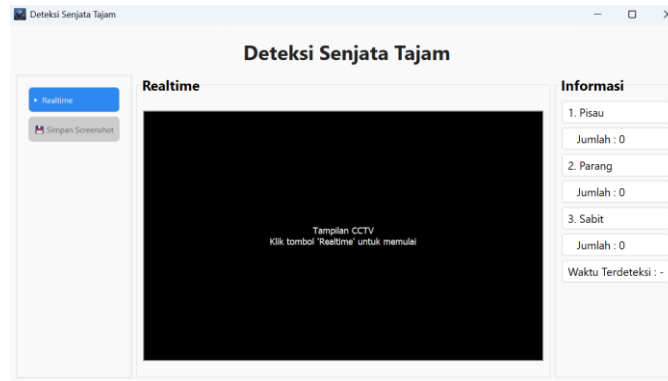
Kelas	TP	FP	FN	Precision	Recall	F-Measure
Parang	67	31	16	0.684	0.807	0.736
Pisau	64	21	10	0.753	0.865	0.865
Sabit	80	11	11	0.879	0.879	0.878

Berdasarkan hasil *Confusion Matrix* Data Validasi tersebut, Tabel 1. memperlihatkan metrik evaluasi *Confusion Matrix* data validasi model YOLOv8n dalam mendeteksi dan mengklasifikasikan objek Parang, Pisau, dan Sabit. Berdasarkan hasil tersebut, kelas sabit menunjukkan performa terbaik dengan nilai *precision*, *recall*, dan *F-measure* yang tinggi serta tingkat deteksi kesalahan yang rendah. Sementara itu, kelas pisau dan parang menunjukkan kinerja yang cukup baik, namun masih memerlukan peningkatan, khususnya pada nilai *recall*. Rendahnya *precision* dan tingginya nilai *false positive* pada beberapa kelas mengindikasikan masih adanya kesalahan dalam proses deteksi objek. Secara keseluruhan, model memberikan kinerja optimal pada kelas sabit dengan nilai rata-rata *F-measure* sebesar 0,878.

3. Pengujian Data YOLOv8n secara Real-Time

Setelah model YOLOv8n selesai melalui tahap pelatihan, langkah selanjutnya adalah melakukan tahap pengujian yang mencakup proses integrasi sistem ke dalam aplikasi GUI serta pengujian performa model YOLOv8n secara *real-time*.

A. Integrasi Grapichal User Interface (GUI)



Gambar 11. Design Interface System

Gambar 11. merupakan desain dari sistem GUI yang digunakan. Dengan rancangan ini, sistem memiliki karakteristik yang *user-friendly*, mendukung peningkatan keamanan melalui deteksi *realtime*, serta memudahkan proses dokumentasi dan pelaporan kejadian secara lebih sistematis.

B. Data Uji *Real-Time*

Untuk menampilkan model kinerja dalam mendeteksi senjata tajam secara *real-time*, digunakan *Dataset* pengujian yang terdiri dari 90 *frame* pada setiap variasi jarak (1-4 meter). Pengambilan data dilakukan secara langsung di warung “Mama Mia” yang berlokasi di Kelurahan Lingkas Ujung, Kota Tarakan.



Gambar 12. Hasil Deteksi Real-Time

Gambar 12. menampilkan hasil deteksi senjata tajam terhadap variasi jarak deteksi. Jika senjata tajam berhasil terdeteksi maka pemberitahuan alarm berbunyi, begitu pula sebaliknya. Berikut merupakan tabel akurasi notifikasi alarm pada setiap meter nya.

Tabel 2. Akurasi Alarm

No	Objek	1 meter	2 meter	3 meter	4 meter
1.	Parang	100%	100%	76,6%	76,6%
2.	Pisau	100%	96,6%	86,6%	0%
3.	Sabit	100%	100%	56,6%	33,3%
4.	Total	100%	98,8%	73,3%	34,4%

Pada tahap pengujian sistem, mekanisme notifikasi alarm juga dijelaskan sebagai bagian dari evaluasi kinerja deteksi. Sistem dirancang untuk mengaktifkan bunyi alarm secara otomatis ketika objek senjata tajam berhasil terdeteksi oleh model dengan tingkat kepercayaan tertentu baik objek yang terdeteksi benar atau salah prediksi, alarm akan tetap berbunyi karena terdeteksi nya senjata tajam. Sebaliknya, apabila pada suatu frame tidak teridentifikasi objek senjata tajam, maka alarm tidak akan berbunyi. Fitur ini berfungsi sebagai bentuk peringatan dini (*early warning system*) kepada masyarakat, sehingga keberadaan objek yang berpotensi membahayakan dapat segera diketahui dan ditindaklanjuti secara cepat.

4. Analisis Hasil Data Uji *Real-Time*

Tabel 3. Hasil evaluasi performa klasifikasi terhadap data uji real-time

Nama kelas	<i>Precision</i>	<i>Recall</i>	<i>F-Measure</i>
Parang	$75 / (75+29)$ = 0,721	$75 / (75+16)$ = 0,824	$2 * (0,721*0,824) /$ $(0,721+0,824) = 0,769$
Pisau	$76 / (76+8)$ = 0,905	$76 / (76+35)$ = 0,685	$2 * (0,905*0,685) /$ $(0,905+0,685) = 0,780$
Sabit	$84 / (84+13)$ = 0,865	$84 / (84+23)$ = 0,785	$2 * (0,865*0,785) /$ $(0,865+0,785) = 0,823$

Tabel 4. Rekapitulasi perhitungan performa terhadap data uji real-time

Nama Kelas	TP	FP	FN	<i>Precision</i>	<i>Recall</i>	<i>F-Measure</i>
Parang	75	29	16	0,721	0,824	0,769
Pisau	76	8	35	0,905	0,685	0,780
Sabit	84	13	23	0,865	0,785	0,823
Total	235	50	74	0,824	0,760	0,790

Secara umum, hasil pengujian *real-time* menunjukkan bahwa model mampu mendeteksi tiga kelas senjata tajam serta mengaktifkan notifikasi alarm sebagai sistem peringatan dini ketika objek teridentifikasi. Performa deteksi bervariasi antar kelas, di mana parang memiliki *recall* tertinggi sehingga paling responsif terdeteksi, pisau menunjukkan keseimbangan antara *presisi* dan *recall*, sedangkan sabit memiliki *precision* tinggi namun *recall* relatif rendah. Secara keseluruhan, model memiliki rata-rata *precision* sebesar 0,824, *recall* sebesar 0,760, dan *F-measure* sebesar 0,790 yang menunjukkan kemampuan deteksi yang cukup baik secara *real-time*. Variasi ini mengindikasikan bahwa karakteristik objek visual, seperti ukuran dan bentuk, mempengaruhi kemampuan deteksi model.

Selain itu, penurunan performa pada jarak tertentu disebabkan oleh berkurangnya resolusi dan detail objek yang ditangkap kamera seiring bertambahnya jarak, sehingga fitur visual menjadi kurang jelas untuk dikenal oleh model. Faktor lain seperti pencahayaan, *noise*, serta keterbatasan resolusi kamera juga turut mempengaruhi kualitas citra yang dihasilkan, yang pada akhirnya mempengaruhi akurasi deteksi. Meskipun sistem telah menunjukkan kinerja yang cukup baik untuk implementasi awal, optimalisasi lebih lanjut masih diperlukan guna meningkatkan konsistensi deteksi, terutama pada objek dengan bentuk dan ukuran yang lebih kompleks, serta untuk mendukung deteksi multi-objek dalam cakupan area yang lebih luas.

Berdasarkan hasil pengujian yang dilakukan, hipotesis penelitian yang menyatakan bahwa sistem deteksi senjata tajam berbasis YOLO mampu mengenali tiga jenis objek pisau, parang, dan sabit dengan akurasi yang baik pada kondisi *real-time* dinyatakan dapat mendeteksi senjata tajam seperti pisau, parang, dan sabit dengan baik dan dapat memberikan peringatan dini berupa notifikasi pemberitahuan alarm. Penelitian ini memberikan kontribusi dalam pengembangan sistem deteksi senjata tajam berbasis YOLOv8n yang tidak hanya menitikberatkan pada akurasi, tetapi juga pada performa *real-time* dalam kondisi nyata. Berbeda dengan penelitian sebelumnya yang umumnya fokus pada uji data statistik atau satu jenis objek, penelitian ini mengkaji deteksi multi-objek (pisau, parang, dan sabit) dengan mempertimbangkan variasi jarak dan kualitas citra. Hasil penelitian menunjukkan adanya hubungan antara jarak objek dan kualitas citra terhadap penurunan kinerja deteksi, yang mengindikasikan pentingnya faktor lingkungan dalam implementasi sistem. Selain itu, penelitian ini juga menekankan bahwa pemilihan model harus mempertimbangkan efisiensi komputasi, di mana YOLOv8n terbukti lebih optimal untuk deteksi *real-time* dibandingkan model yang lebih kompleks. Oleh karena itu, penelitian ini menawarkan pendekatan yang lebih komprehensif dengan mengintegrasikan aspek akurasi, kecepatan, dan kondisi operasional nyata.

IV. KESIMPULAN

Penelitian ini berhasil merancang dan mengimplementasikan sistem deteksi senjata tajam berbasis YOLOv8n menggunakan 2.640 citra dan pelatihan selama 100 epoch, serta

diintegrasikan dengan antarmuka GUI berbasis PyQt5 untuk pemantauan *real-time*. Berdasarkan hasil pengujian, model mampu mendeteksi objek senjata tajam dengan rata-rata *precision* sebesar 0,824, *recall* sebesar 0,760, dan *F1-score* sebesar 0,790 pada skenario *real-time*. Performa model menunjukkan variasi antar kelas, di mana parang memiliki *recall* tertinggi, pisau menunjukkan *presisi* yang baik, sedangkan sabit masih menjadi kelas yang paling menantang untuk dideteksi secara konsisten. Sistem yang dikembangkan tidak hanya mampu mendeteksi keberadaan senjata tajam, tetapi juga dilengkapi dengan mekanisme peringatan dini berupa notifikasi alarm yang aktif secara otomatis ketika objek terdeteksi dengan tingkat kepercayaan tertentu, sehingga dapat mendukung respon cepat terhadap ancaman potensi. Berdasarkan hasil penelitian, menunjukkan bahwa model YOLOv8n memiliki potensi yang baik untuk implementasi deteksi senjata tajam secara *real-time*, di mana hipotesis penelitian dinyatakan mampu dan dapat digunakan mendeteksi tiga jenis senjata tajam. Namun optimalisasi lebih lanjut masih diperlukan untuk meningkatkan konsistensi dan akurasi deteksi pada setiap kelas, serta mendukung deteksi multi-objek dalam cakupan area yang lebih luas pada kondisi lingkungan yang lebih kompleks.

Pengembangan selanjutnya disarankan untuk menggunakan kamera dengan resolusi lebih tinggi guna meningkatkan kualitas citra, serta menambahkan variasi *Dataset* seperti kondisi oklusi, sudut pandang, latar belakang, dan pencahayaan. Selain itu, sistem perlu dikembangkan untuk mendukung deteksi multi-objek dalam satu frame agar lebih adaptif pada kondisi nyata.

REFERENSI

- Ainun, N., Dira, K., Khairunnisa, A., Aras, S., & Data, A. P. (2023). *Deteksi Penggunaan Safety Helmet Menggunakan YOLOv5*. 07, 74–77.
- Aziz, A. N., Khoiriyah, H., Abdillah, F., & Wiryawan, I. G. (2024). *Prototipe Sederhana Sistem Deteksi Kriminal Berbasis Internet Of Things Menggunakan Teknologi YOLOv5 Simple Prototype of Internet of Things Based Crime Detection System Using YOLOv5 Technology*. 13(148), 139–147. <https://doi.org/10.34010/komputika.v13i1.12217>
- Budi, D. A. (2021). Perancangan Sistem Login pada Aplikasi Berbasis GUI Menggunakan QtDesigner Python. *Jurnal SIMADA (Sistem Informasi Dan Manajemen Basis Data)*, 4(2), 92–100. <https://doi.org/10.30873/simada.v4i2.2961>
- Clara, S., Laksmi Prianto, D., Al Habsi, R., Friscila Lumbantobing, E., & Chamidah, N. (2021). Implementasi Seleksi Fitur Pada Algoritma Klasifikasi Machine Learning Untuk Prediksi Penghasilan Pada Adult Income Dataset. *Seminar Nasional Mahasiswa Ilmu Komputer Dan Aplikasinya (SENAMIKA) Jakarta-Indonesia*, 2(1), 741–747.
- Dompeipen, T. A., & Sompie, S. R. U. . (2020). Penerapan computer vision untuk pendeteksian dan penghitung jumlah manusia. *Jurnal Teknik Informatika*, 15(4), 1–12.
- Hayati, N. J., Singasatia, D., & Muttaqin, M. R. (2023). Object Tracking Menggunakan Algoritma You Only Look Once (YOLO)v8 untuk Menghitung Kendaraan. *Komputa : Jurnal Ilmiah Komputer Dan Informatika*, 12(2), 91–99. <https://doi.org/10.34010/komputa.v12i2.10654>
- Jamil Azali, F. J. (2024). *Aksi Begal Bersenjata Tajam Todong Empat Mahasiswa di Baubau Terekam CCTV*. Tvonenews.Com. <https://www.tvonenews.com/daerah/sulawesi/234944-aksi-begal-bersenjata-tajam-todong-empat-mahasiswa-di-baubau-terekam-cctv%0A>
- Kitab Undang Undang Hukum Pidana. (2021). Kitab Undang-Undang Hukum Pidana. *Indonesia*, 5(8), 1–143.
- Kojongian, R., & Anggriyani, R. (2024). *Analisis Kriminologis terhadap Fenomena Kejahatan Begal dengan Senjata Tajam di Kota Kendari*. VIII(1), 342–348.
- Liunanda, C. N., Rostianingsih, S., & Purbowo, A. N. (2020). Implementasi Algoritma YOLO pada Aplikasi Pendeteksi Senjata Tajam di Android. *Jurnal Infra, Vol 8, No.*, 1–7.
- Mandela, N., Langi, R., Sampul, H., Studi, P., Komputer, T., Teknik, F., & Tarakan, U. B. (2025). *APD MULTI OBJEK PADA AREA KONTRUKSI BERBASIS COMPUTER VISION DENGAN DEEP-LEARNING APD MULTI OBJEK PADA AREA KONTRUKSI BERBASIS COMPUTER VISION DENGAN DEEP-LEARNING*.
- Maulana, I., Rahaningsih, N., & Suprapti, T. (2024). Analisis Penggunaan Model Yolov8 (You Only Look Once) Terhadap Deteksi Citra Senjata Berbahaya. *JATI (Jurnal Mahasiswa Teknik Informatika)*, 7(6), 3621–3627. <https://doi.org/10.36040/jati.v7i6.8271>

- Nazar, R. (2024). Implementasi Pemrograman Python Menggunakan Google Colab. *Jurnal Informatika Dan Komputer (JIK)*, 15(1), 50–56.
- Nugraha. (2023). *Kebijakan Kriminal Terhadap Kejahatan Begal Di Wilayah Hukum Polsek Medan Barat*. 71. <https://repositori.uma.ac.id/handle/123456789/21979>
- Pradana, I. C., Mulyanto, E., & Rachmadi, R. F. (2022). Deteksi Senjata Genggam Menggunakan Faster R-CNN Inception V2. *Jurnal Teknik ITS*, 11(2). <https://doi.org/10.12962/j23373539.v11i2.86587>
- Sholahuddin, M. R., Atqiya, F., Wulan, S. R., Harika, M., Fitriani, S., & Sofyan, Y. (2023). Implementasi Sistem Identifikasi Senjata Real Time Menggunakan YOLOv7 dan Notifikasi Chat Telegram. *Journal of Information System Research (JOSH)*, 4(2), 598–606. <https://doi.org/10.47065/josh.v4i2.2774>
- Sutisna, T., Raharja, A. R., Solihin, S., Hariyadi, E., & Putra, V. H. C. (2024). Penggunaan computer vision untuk menghitung jumlah kendaraan dengan menggunakan metode SSD (Single Shoot Detector). *Innovative: Journal of Social Science Research*, 4(2), 6060–6067.
- Tuasamu, Z., Lewaru, N. A. I. M., Idris, M. R., Syafaat, A. B. N., Faradilla, F., Fadlan, M., Nadiva, P., & Efendi, R. (2023). Analisis Sistem Informasi Akuntansi Siklus Pendapatan Menggunakan DFD Dan Flowchart Pada Bisnis Porobico. *Jurnal Bisnis Dan Manajemen (JURBISMAN)*, 1(2), 495–510.