



Klasifikasi Tingkat Ancaman Kriminalitas Bersenjata Menggunakan Metode *You Only Look Once (YOLO)*

Muhammad Abdul Hadi¹, Dr. Eng. Rian Ferdian², M.T., Lathifah Arief, M.T.³

¹ Mahasiswa Jurusan Teknik Komputer, Fakultas Teknologi Informasi Universitas Andalas, Padang, Sumatera Barat 25163 Indonesia

^{2,3} Dosen Jurusan Teknik Komputer, Fakultas Teknologi Informasi Universitas Andalas, Padang, Sumatera Barat 25163 Indonesia

INFORMASI ARTIKEL

Sejarah Artikel:

Diterima Redaksi: 22 April 2021

Revisi Akhir: 29 April 2021

Diterbitkan Online: 30 April 2021

KATA KUNCI

YOLO object detection, weapon threat classification, monitoring.

KORESPONDENSI

Phone: +6282288152033

E-mail: hadiasta33@gmail.com

A B S T R A C T

The research aim to recognize potential weapon threats through object detection on camera. This research utilize YOLO (You Only Look Once) method in object detection which implemented on Raspberry Pi 4. The process was by detecting object from the camera and classify the object class in 2 available classes : Gun and Knife. Meanwhile, in the classifying process, it also count the object in every classes. When the system detect object in the process, it will send notification in terms of threat level through android application so that the user or operator can mitigate the threat immediately. From the research, we achieve the mAP of 85.12% in which YOLOv4 tiny is used and the testing is done inside a room environment. In its application in detecting weapon in Raspberry Pi 4, the result is around 1.53 fps (frame per second), in which is accommodate to be implemented on, but with a very limited fps.

PENDAHULUAN

Penyerangan bersenjata dan pembunuhan merupakan tindakan kriminal berat, dan merupakan salah satu kejahatan yang paling tinggi tingkatannya dalam klasifikasi kejahatan internasional. Di Indonesia sendiri, kasus penyerangan bersenjata berupa pembegalan, penusukan, dan serangkaian aksi terorisme telah mewarnai tahun 2019 dan menimbulkan keresahan dalam masyarakat. Data publikasi statistik kriminal dari Badan Pusat Statistik (BPS) tahun 2019 menunjukkan bahwa tingkat kejahatan disertai kekerasan dan senjata yaitu 8.423 kejadian selama tahun 2018[1]. Kasus kejadian ini cenderung fluktuatif dari tahun tahun sebelumnya, namun jumlah kejadian tetaplah tinggi dan menimbulkan keresahan dalam masyarakat, terutama tindakan kekerasan yang disertai senjata dan pembunuhan. Sering kita melihat adanya berita dan kasus pencurian serta kekerasan yang disertai ancaman bersenjata. Di lingkungan kampus UNAND contohnya, dimana pada Bulan Juli Tahun 2019 lalu, seorang mahasiswi diancam dengan senjata kapak untuk menyerahkan motor dan barang berharga lainnya. Selain itu, pada bulan januari 2020 kampus UNAND juga dihebohkan dengan adanya kasus pembegalan dan pencurian sepeda motor yang kemudian dikejar oleh sejumlah mahasiswa. Dalam melakukan aksinya, tidak jarang pelaku melakukan ancaman dengan

menodongkan senjata untuk membuat korban diam dan patuh. Kasus-kasus penyerangan bersenjata yang menelan korban jiwa seperti pembegalan, perampokan, penusukan, tawuran, penembakan, dan terorisme merupakan tindakan kriminal yang harus cepat disadari dan memerlukan respon yang cepat dari aparat keamanan dan kepolisian. Jika terlambat menyadarinya, maka akan dapat menelan korban jiwa.

Untuk menangani kasus-kasus ancaman dan penyerangan bersenjata tersebut, Berbagai upaya dilakukan oleh pemerintah dan kepolisian dalam mengurangi kasus kejadian dan mencegah terjadinya tindakan kriminal serupa. Dalam penanganan tindakan kriminal, terdapat klasifikasi dan tingkat prioritas yang menjadi pedoman dalam melakukan respon terhadap tindakan kriminal. Tingkatan ini didasari oleh jenis kejahatan dengan melihat dampak yang ditimbulkan dari suatu tindakan kriminal, bagaimana kejahatan tersebut dilakukan, lalu target dari kejadian [1]. Tingkatan ini menjadi pedoman bagi aparat kepolisian dalam melakukan penanganan kasus kriminal dan mengetahui respon yang perlu diberikan. Salah satu parameter utama untuk melakukan analisa situasi yaitu mendapatkan kondisi terkini di lapangan, yang biasanya menggunakan akses kamera CCTV di sekitar lokasi kejadian.

Meningkatnya sistem pengawasan CCTV mampu menekan kasus kejahatan dan mempermudah pencarian pelaku kriminal[2]. Namun di sisi lain, pengawasan dengan CCTV bergantung penuh pada operator yang bertugas memantau dan mengawasi. Terkadang, seorang operator harus mengawasi banyak kamera CCTV, yang mengurangi efektivitas karena keterbatasan ruang lingkup pengawasan yang dapat dilakukan oleh manusia[3]. Untuk itu, Penerapan sistem cerdas dalam sistem keamanan (*video surveillance*) menjadi solusi untuk meningkatkan kualitas pengawasan CCTV.

Berbagai pendekatan seperti kecerdasan buatan dan *deep learning* diterapkan untuk meningkatkan kemampuan deteksi dan klasifikasi objek menggunakan kamera[4]. Michael Grega[5] dalam penelitiannya menggunakan pendekatan *deep learning* untuk mendeteksi adanya objek senjata api dan pisau pada kamera. Hasilnya didapatkan tingkat akurasi deteksi sebesar 91%, namun hal ini juga bergantung pada jarak dari kamera dan jenis kamera yang digunakan dalam melakukan deteksi. Penelitian lain mengimplementasikan algoritma pendeteksian senjata pada perangkat SBC ODROID-XU4 untuk diterapkan pada drone dan memberikan notifikasi berupa alarm ketika terdeteksi adanya senjata[6]. Penelitian sejenis lainnya mencoba menerapkan optimasi pada algoritma deteksi senjata untuk mempercepat respon waktu pada pendeteksian secara *realtime*[7]. Dari beberapa penelitian yang ada diatas, implementasi sistem untuk memberikan notifikasi dan respon terhadap pendeteksian senjata belum sepenuhnya dilakukan, dan lebih kepada penelitian konseptual.

Dari permasalahan diatas, diusulkan penelitian berupa implementasi sistem cerdas untuk mengklasifikasikan tingkat ancaman kriminalitas bersenjata dengan parameter kategori senjata dan jumlah senjata yang terdeteksi pada Kamera. Jika terdeteksi objek senjata oleh kamera, maka akan dikirim notifikasi dan peringatan berupa kemungkinan tingkat ancaman yang ditimbulkan kepada operator CCTV melalui notifikasi aplikasi android. Hal ini dapat menjadi pedoman bagi pihak berwenang untuk kemudian dilakukan penanganan dan atau pencegahan tindakan kriminal bersenjata dengan cara yang tepat. Sistem diterapkan pada perangkat Raspberry Pi berupa *prototype* alat yang diterapkan pada lingkungan pengujian terkontrol di dalam ruangan. Pemilihan perangkat Raspberry sebagai platform implementasi sistem dikarenakan *form factor* dari perangkat Raspberry Pi yang dikemas dalam ukuran yang mudah di terapkan dan mudah diintegrasikan pada berbagai lingkungan. Selain itu perangkat Raspberry Pi mampu mengimbangi kemampuan komputasi yang setara dengan perangkat PC (Personal Computer) standar dan lebih hemat daya untuk penggunaan dalam jangka waktu yang lama. Untuk mendukung implementasi klasifikasi tingkat ancaman senjata pada Raspberry Pi, digunakan metode YOLO karena kecepatan dalam deteksi objek. Selain YOLO, terdapat beberapa algoritma klasifikasi objek seperti *K-Nearest Neighbour* (KNN), *C4.5 Algorithm*, Naive Bayes, *Convolutional Neural Network* (CNN), Random Forest, dan algoritma lainnya yang intensif dalam komputasi sehingga mampu menghasilkan akurasi yang tinggi[8]. Namun, disamping itu tidak sanggup untuk diterapkan pada PC standar ataupun raspberry Pi, sehingga sulit dalam melakukan pengujian secara *realtime* dari inputan kamera dengan komputasi yang

terbatas. Untuk itu digunakan metode YOLO untuk merancang sistem deteksi objek senjata pada penelitian ini.

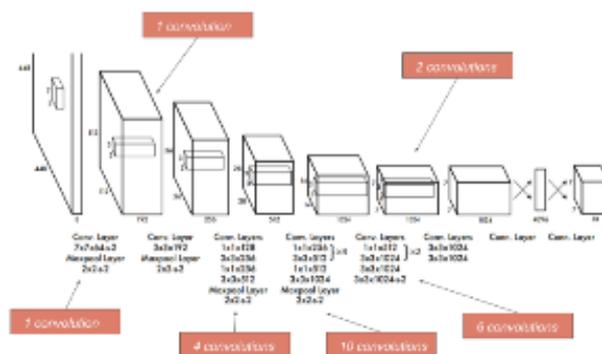
LANDASAN TEORI

You Only Look Once (YOLO) Object Detection

YOLO merupakan salah satu metode dalam algoritma *machine learning* dengan spesifikasi untuk mendeteksi objek pada frame. *YOLO* pertama kali diperkenalkan pada tahun 2015 oleh Joseph Redmon, Santosh Divvala, dan Ross Girshick dalam jurnal *You Only Look Once : Unified, Real-Time Object Detection*[9]. Dalam jurnal yang dipublikasikan tersebut, *YOLO* direpresentasikan sebagai pendekatan baru dalam pendeteksian objek. Dibandingkan pendekatan yang menggunakan pipeline untuk melakukan klasifikasi dan menemukan lokasi objek dalam beberapa tahapan, *YOLO* mengoptimasi proses deteksi dengan melakukan setiap prosesnya dalam satu *neural network*, hal ini dicapai dengan mengubah pendekatan deteksi objek menjadi satu masalah regresi, langsung dari pixel pixel pada gambar untuk menentukan koordinat *bounding box* dan klasifikasi kelas objek. *YOLO* dirancang untuk melatih gambar secara utuh dan mengoptimasi deteksi objek, dengan membagi gambar menjadi beberapa segmen NxN.

Arsitektur YOLO

Arsitektur *YOLO* secara garis besar dipengaruhi oleh backbone GoogleLeNet, dengan jaringan saraf yang terdiri dari 24 *convolution layers* untuk melakukan ekstraksi fitur dan diikuti oleh 2 FCN (*Fully Connected Layer*) untuk melakukan prediksi koordinat *bounding box* dan klasifikasi kelas objek.



Gambar 1. Arsitektur YOLO

Dalam upaya untuk meningkatkan keakuratan deteksi dan kecepatan proses pada *YOLO*, penulis awal *YOLO* mencoba untuk menurunkan batasan dalam melakukan deteksi dan klasifikasi kelas. Hal ini yang menghasilkan *YOLO9000*, suatu sistem pendeteksian objek yang dapat melakukan deteksi terhadap 9000 kategori objek. Lebih lanjut, karena pada *YOLO* original (*YOLOv1*) terdapat keterbatasan besar dari segi *localization loss* dan kemampuan untuk memprediksi ulang, maka diperkenalkan suatu peningkatan kerja dari *YOLO* yang menghasilkan *YOLO9000* atau selanjutnya dikenal dengan *YOLOv2*. Dari versi *YOLOv2*, penulis awal *YOLO* mencoba melakukan peningkatan incremental yang kemudian menghasilkan *YOLOv3*. Pada April 2020, Alexey Bochkovskiy mengumumkan metode peningkatan dari *YOLOv3* yang dikenal

dengan YOLOv4, dengan peningkatan kecepatan optimal dan akurasi deteksi. YOLOv4 memiliki akurasi yang tinggi, dimana mencapai 43.5% AP (average precisions). Dalam penelitian ini, digunakan YOLOv4 tiny dalam pelatihan model untuk mendeteksi objek senjata pisau dan pistol.

Pengujian Kualitas Model dengan Medium Average Precisions (mAP)

mAP merupakan salah satu metrik untuk menentukan kualitas dari model Machine learning[10]. Dalam machine learning, terdapat beberapa parameter metrik yang perlu diperhatikan :

Presisi (Precision) : pengukuran tingkat akurasi persentase prediksi yang benar, dengan parameter dari semua yang diprediksi benar, berapa yang benar-benar positif (actual positive). Hal ini dapat dinotasikan dalam persamaan :

$$Precisions = \frac{True\ Positive}{True\ Positive + False\ Positive} \tag{1}$$

$$Precisions = \frac{True\ Positive}{Total\ predicted\ positive} \tag{2}$$

Presisi berguna untuk dijadikan parameter pengukuran kualitas model untuk kasus false positif yang tinggi.

		Predicted	
		Negative	Positive
Actual	Negative	True Negative	False Positive
	Positive	False Negative	True Positive

Gambar 2. Confusion Matrix dari Pengukuran metrik Presisi (Precisions)

Recall : Recall mengukur berapa banyak yang benar-benar positif dari hasil prediksi model dibandingkan dengan semua yang dilabeli positif. Hal ini dapat dinotasikan dengan :

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative} \tag{3}$$

$$Recall = \frac{True\ Positive}{Total\ Actual\ Positive} \tag{4}$$

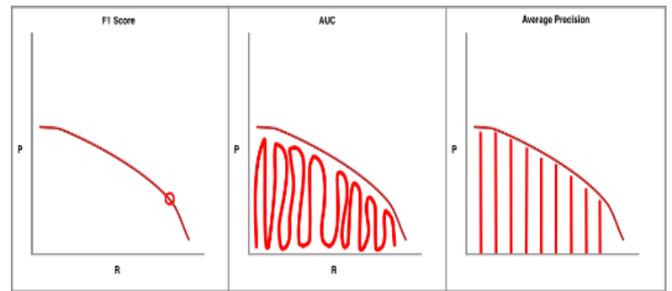
		Predicted	
		Negative	Positive
Actual	Negative	True Negative	False Positive
	Positive	False Negative	True Positive

Gambar 3. Confusion Matrix dari Pengukuran metrik Recall

F1 Score : Selain metrik presisi dan recall, terdapat fungsi yang mencari keseimbangan antara metrik recall dan presisi. Hal ini dicapai dengan rumus :

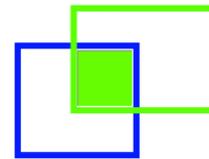
$$F1 = 2x \frac{Precision \times Recall}{Precisions + Recall} \tag{5}$$

Dari 3 metrik diatas, untuk menghitung Average Precisions (AP) digunakan kurva dari *precision-recall* (kurva F1) untuk menghitung area di bawah kurva tersebut[11].



Gambar 4. Kurva metrik *precisions-recall*[11]

Dalam deteksi objek, prediksi dilakukan dalam representasi bounding box dan kelas label. Dalam penerapannya, penentuan bounding box melalui koordinat X, Y, h, dan w akan menghasilkan sedikit ketidakcocokan dengan objek aslinya (*ground truth*). Untuk mengatasi hal tersebut, dikenalkan metode IOU (*Intersection Over Union*) yang menyediakan metrik untuk menyesuaikan batasan bounding box dan menentukan bounding box yang pas dari beberapa bounding boxes yang saling tumpang tindih. IoU biasanya diatur pada threshold 60 sampai 75 persen. Jadi, IoU digunakan untuk validasi deteksi apakah hasil prediksi benar atau tidak.



Gambar 5. Sketsa proses penentuan bounding boxes dengan metrik IoU[11]

Setelah didapatkan hasil perhitungan metrik-metrik diatas, selanjutnya dilakukan Perhitungan AP atau tingkat akurasi prediksi rata-rata atau area dibawah kurva *precision-recall*. Hal ini didapatkan dengan persamaan :

$$AP = \frac{1}{11} \sum_{r \in \{0,0.1,0.2,0.3,0.4,0.4,0.5,0.6,0.7,0.8,0.9,1\}} P_{interp(r)} \tag{6}$$

Dimana 1/11 didapatkan dari pembagian recall menjadi 11 bagian dari 0-1. $P_{interp(r)}$ merupakan perhitunga nilai interpolasi dari kurva *precision-recall*, dengan persamaan $P_{interp(r)} = \max p(r_i)$. Perhitungan AP dilakukan untuk masing-masing kelas objek[12]. Setelah didapatkan AP untuk masing-masing kelas objek, dilakukan perhitungan AP rata-rata atau mAP dengan persamaan :

$$mAP = \frac{1}{N} \sum_{i=1}^N AP_i \tag{7}$$

Dimana AP_i merupakan skor AP (*Average Precisions*) untuk masing-masing masing-masing kelas objek (N)[12].

Raspberry Pi

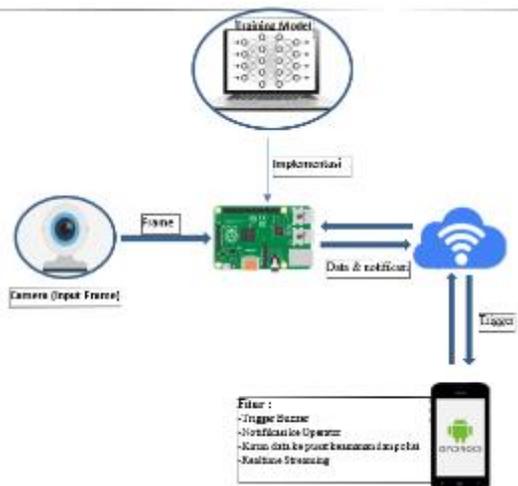
Raspberry Pi merupakan seri *Single Board Computer* (SBC) yang dikembangkan di Inggris oleh *Raspberry Pi Foundation*. Raspberry Pi merupakan salah satu seri SBC yang menjadi *top selling* dan telah hadir dalam beberapa seri dan generasi. Pada dasarnya, semua model dari Raspberry Pi menggunakan CPU ARM dan seri terbaru juga mencakup GPU yang juga dari manufaktur ARM. Generasi pertama dari Raspberry Pi yaitu Raspberry Pi Model B yang diluncurkan pada tahun 2012, dengan 700 Mhz *processor* dan 256MB RAM, seri ini disusul oleh Raspberry Pi Model A. Pada tahun 2014, pengembang Raspberry Pi meluncurkan seri Model B+ dengan beberapa peningkatan dari pendahulunya. Generasi kedua dari Raspberry Pi yaitu Raspberry Pi 2 yang diluncurkan pada tahun 2015, dengan 900 Mhz *Processor Quad-Core ARM Cortex-A7* dan 1GB RAM. Selanjutnya menyusul dikembangkannya seri Raspberry Pi Zero yang memiliki ukuran lebih kecil. Generasi selanjutnya yaitu Raspberry Pi 3 Model B dan Raspberry Pi 3 Model B+ yang diluncurkan pada tahun 2016 dimana dilengkapi dengan tambahan *on-board Wi-Fi dan Bluetooth 802.11n*, selain itu juga terdapat fitur untuk Boot dari USB. Pada tahun 2019, Raspberry Pi foundation meluncurkan Raspberry Pi 4 Model B dengan 1.2 GHz 64-Bit quad-core processor, *on-board Wi-Fi 802.11ac, Bluetooth 5, USB 3.0, dan HDMI yang mensupport resolusi 4K*. Dalam penelitian ini, digunakan *board Raspberry Pi 4* untuk penerapan deteksi objek[13].



Gambar 6. Raspberry Pi 4

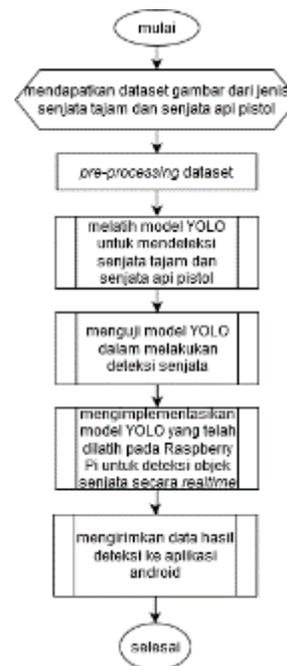
METODOLOGI PENELITIAN

Rancangan Umum Sistem



Gambar 7. Bagan rancangan umum sistem

Dalam perancangan sistem, pertama dilakukan pelatihan model YOLO untuk mendeteksi adanya objek senjata dari 2 klasifikasi jenis senjata : senjata pisau dan senjata pistol. Setelah model dilatih, implementasi pendeteksian senjata diterapkan pada Raspberry Pi 4 untuk *realtime detection* menggunakan kamera. Jika terdeteksi adanya objek senjata, maka akan dikirim notifikasi dan data gambar serta lokasi dari kamera ke aplikasi android, untuk kemudian diambil tindakan oleh operator pengawas. Dengan adanya notifikasi *realtime* jika terdeteksi adanya senjata, maka diharapkan dapat mengurangi kesalahan operator dalam melakukan pengawasan dan mengurangi tindak kejahatan bersenjata dan kekerasan. Dengan adanya sistem notifikasi *early warning* tindak kejahatan bersenjata, maka respon dan tindakan yang dilakukan aparat diharapkan akan lebih cepat. Untuk lebih jelasnya, dapat dilihat alur proses dari sistem pada gambar dibawah.



Gambar 8. Alur proses secara keseluruhan

HASIL DAN PEMBAHASAN

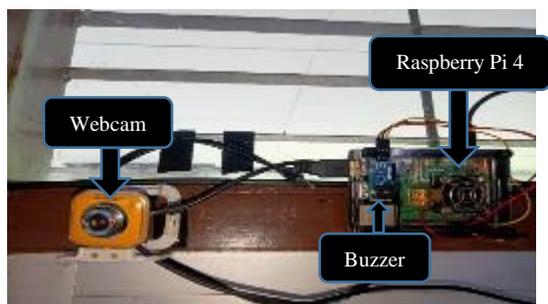
Implementasi

Implementasi Perangkat Keras

Untuk implementasi perangkat keras, komponen yang digunakan yaitu :

1. Kamera, berfungsi untuk menangkap tampilan frame video secara *realtime* dan mengirimkan datanya ke Raspberry Pi untuk diolah dan dilakukan deteksi objek senjata menggunakan YOLOv4 tipe *tiny*. Kamera yang digunakan yaitu webcam dengan resolusi 5.0 *MegaPixel* (MP).
2. Raspberry Pi 4 Model B 4 GB RAM, berfungsi sebagai otak sistem untuk melakukan pemrosesan frame dari tangkapan kamera dan melakukan deteksi objek dengan metode YOLOv4 tipe *tiny*.
3. Buzzer, berfungsi untuk memberikan peringatan berupa alarm jika terdeteksi adanya objek senjata.

Komponen-komponen diatas dipasang pada titik yang menyerupai pemasangan CCTV untuk yang diterapkan pada area pengujian. Gambaran implementasi perangkat keras penelitian ini dapat dilihat pada gambar berikut.



Gambar 9. Perangkat Keras yang digunakan untuk penelitian

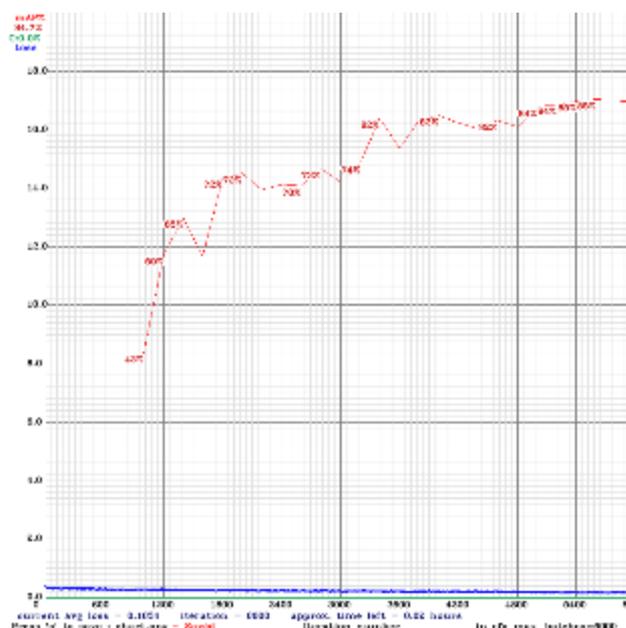
Implementasi Perangkat Lunak

Terdapat 3 Modul perangkat lunak yang menjadi komponen utama dalam keutuhan jalannya sistem. 3 Modul tersebut yaitu :

1. Modul Pelatihan dan Pengujian Model YOLO, modul program ini berfungsi untuk melakukan pelatihan pada model YOLO untuk mendeteksi objek senjata pistol dan senjata pisau. Setelah dilakukan pelatihan model, selanjutnya model diuji untuk mengukur tingkat akurasi deteksi. Program pelatihan dan pengujian model YOLO diterapkan menggunakan bahasa pemrograman Python dengan tipe YOLOv4 Tiny.
2. Modul implementasi model deteksi objek senjata YOLO pada Raspberry Pi 4. Setelah model YOLO untuk deteksi objek senjata dilatih, selanjutnya model diterapkan pada Raspberry Pi 4 untuk melakukan deteksi objek senjata secara *realtime* dan mengirim data hasil deteksi ke database.
3. Modul Aplikasi Android, untuk menerima data dan notifikasi peringatan ketika terdeteksi adanya objek senjata oleh sistem.

Hasil pelatihan modul YOLO dan pengujian Kualitas Model YOLO

Dalam pelatihan model YOLOv4 Tiny, dataset yang digunakan terdiri dari 2 kategori kelas : kelas pisau, dan kelas pistol. Total data gambar yang digunakan yaitu 3000 gambar, dengan 600 gambar pisau dan 2400 gambar pistol yang masing-masingnya memiliki variasi bentuk dan pola. Untuk parameter pelatihan YOLO, digunakan model pre-trained model YOLOv4 Tiny yang telah mengenali fitur-fitur dasar dari frame gambar, model pre-trained tersebut kemudian dilatih ulang menggunakan dataset dari 2 kelas diatas untuk mengenali fitur fitur senjata pisau dan pistol. Penggunaan model pre-trained ini menghemat waktu pelatihan dan juga mengurangi kebutuhan akan dataset yang besar.



Gambar 10. Hasil pelatihan dan pengujian kualitas model YOLOv4 tiny untuk deteksi objek senjata.

Implementasi model Deteksi objek senjata YOLOv4 Tiny pada Raspberry Pi 4

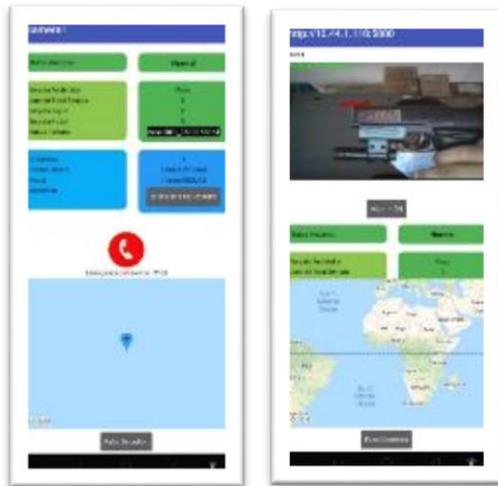
Setelah model YOLO dilatih untuk mengenali senjata dan diuji tingkat akurasinya, selanjutnya model diterapkan pada perangkat Raspberry Pi untuk pengujian secara *realtime* melalui input *live* webcam.



Gambar 11. Implementasi Sistem pendeteksi objek senjata YOLOv4 pada Raspberry Pi 4.

Implementasi Aplikasi Android

Aplikasi android berfungsi untuk menerima notifikasi jika terdeteksi adanya senjata oleh sistem pada perangkat Raspberry Pi 4. Saat terdeteksi adanya senjata, sistem akan memperbarui database firebase yang kemudian dibaca oleh aplikasi android. Jika terdapat perubahan data dan terdeteksi adanya senjata, maka sistem notifikasi akan muncul pada smartphone android berupa *push notification*, yang dapat diterima walaupun tidak membuka aplikasi dan smartphone dalam keadaan terkunci. Namun, untuk dapat menerima notifikasi secara *realtime* diperlukan adanya koneksi internet yang stabil antara user dan server database firebase.



Gambar 12. Tampilan halaman camera

Pengujian dan Analisa

Pengujian dan analisa dilakukan untuk mengkaji dan menganalisa hasil dari implementasi perangkat lunak dan perangkat keras. Dalam penelitian ini, pengujian yang dilakukan terdiri dari 2 bagian : pengujian kualitas model YOLO, dan pengujian fungsional sistem.

Pengujian dan Analisa Kualitas model YOLO

Dalam melakukan pengujian terhadap kualitas model YOLO, terdapat 2 parameter pengujian, yaitu menguji tingkat performa model dan tingkat akurasi atau keakuratan model. Untuk menguji tingkat performa model, digunakan data-data gambar dari lingkungan pengujian yang kemudian dilakukan prediksi dengan model YOLO untuk melihat persentase kebenaran dari hasil prediksi model. Data-data ini kemudian disajikan dalam metrik *confusion matrix* untuk melihat persentase kebenaran hasil prediksi. Hasil dari pengujian dengan data-data gambar senjata dapat dilihat pada gambar dibawah ini.



Gambar 13. pengujian performa model pada data-data gambar dari lingkungan pengujian

Dengan melakukan hal yang sama seperti pengujian diatas, yaitu dengan melakukan prediksi terhadap gambar-gambar dari lingkungan pengujian sebanyak 452 gambar dari variasi senjata pistol, senjata pisau, dan gambar yang tidak mengandung keduanya, maka didapatkan hasil pengujian sebagai berikut :

Kelas : Pistol

True Positive = 142

False Positive = 25

Kelas : Knife (Pisau)

True Positive = 104

False Positive = 14

Tabel 1. Tabel Confusion Matrix hasil pengujian

		Actual Class	
		Terdeteksi	Normal
Predicted Class	Terdeteksi	True Positive = 246	False Positive = 39
	Normal	False Negative = 56	True Negative = 111

Dari hasil *confusion matrix* diatas, selanjutnya diukur kualitas performa model. Untuk mengukur tingkat performa dari model, digunakan metrik *mean Average Precisions (mAP)*, yang didapatkan dari perhitungan *precisions* dan *recall* dari hasil *confusion matrix* diatas. Untuk perhitungan mAP, digunakan fungsi pada darknet yang mana didapatkan hasil :

```

detections count = 1701, unique truth count = 312
class = 0 of masks = 1701
class = 100 of masks = 1483
class = 200 of masks = 1701
class = 300 of masks = 1701
class = 400 of masks = 1483
class = 500 of masks = 1701
class = 600 of masks = 1701
class = 700 of masks = 1483
class = 800 of masks = 1701
class = 900 of masks = 1701
class = 1000 of masks = 1483
class = 1100 of masks = 1701
class = 1200 of masks = 1701
class = 1300 of masks = 1483
class = 1400 of masks = 1701
class = 1500 of masks = 1701
class = 1600 of masks = 1483
class = 1700 of masks = 1701
class_0 = 0, name = pistol, pr = 84.25%   fpr = 142, pp = 25
class_1 = 1, name = knife, pr = 82.49%   fpr = 39, pp = 14

for class in classes:
    class_0 = 0, name = pistol, pr = 84.25%   fpr = 142, pp = 25
    class_1 = 1, name = knife, pr = 82.49%   fpr = 39, pp = 14

mAP Average Precision 1000000000 = 0.8437

```

Gambar 14. Hasil pengukuran mAP model YOLOv4

Dari pengukuran mAP dengan fungsi Darknet, didapatkan tingkat mAP rata-rata sebesar **84.37%**, dengan tingkat *Average Precision AP* untuk kelas pistol sebesar **86.25%** dan AP untuk kelas pisau sebesar **82.49%**. mAP disini berfungsi untuk menentukan tingkat akurasi akhir dari model, yang mana dijadikan parameter sebelum model dideploy pada lingkungan nyata. Dengan tingkat mAP sebesar **84.37 %**, maka dapat dikatakan pelatihan model berhasil dan model dapat diterapkan pada lingkungan nyata untuk deteksi senjata secara realtime pada Raspberry Pi 4.

Pengujian Performa Penerapan Model pada Raspberry Pi 4

Dalam implementasi sistem, model YOLO yang telah dilatih untuk mengenali senjata pistol dan pisau diterapkan pada perangkat Raspberry Pi 4 dengan inputan secara realtime dari webcam dengan resolusi 5.0 MP. Perangkat Raspberry Pi 4

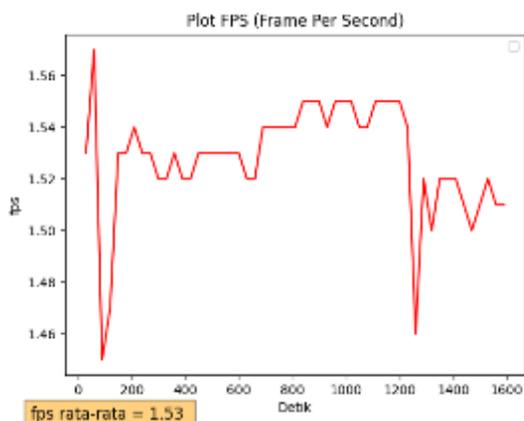
digunakan Untuk memudahkan penerapan model dalam melakukan pengujian. Dalam melakukan deteksi objek dan klasifikasi senjata dan jumlahnya, maka perlu dilakukan pengujian kinerja atau kualitas sistem saat melakukan deteksi. Untuk pengujian kinerja deteksi objek pada perangkat Raspberry Pi 4, digunakan rancangan pengujian sebagai berikut :

Dari hasil yang didapatkan, untuk senjata pistol didapatkan jarak efektif deteksi sejauh **650 cm atau 6,5 m**. Untuk senjata Pisau, didapatkan jarak efektif deteksi sejauh **200 cm atau 2 m**. Dari hasil deteksi dengan beberapa variasi posisi pada jarak yang sama, sistem mampu mendeteksi adanya senjata pisau dengan benar pada jarak 200 m kurang, di luar itu terdapat banyak kasus kesalahan deteksi dan tidak terdeteksinya senjata pisau. Kemampuan sistem dalam mendeteksi adanya senjata pisau tidak sebaik performa deteksi senjata pistol, hal ini karena besarnya variasi dari model senjata pisau pada data latih, yang memiliki banyak variasi bentuk dan jenis yang berbeda. Hal ini menyebabkan kemampuan sistem dalam mendeteksi objek pisau tidak terlalu baik, selain juga dipengaruhi oleh faktor eksternal lainnya seperti tipe YOLO yang digunakan, kualitas pelatihan model, resolusi kamera yang digunakan, dan efek pencahayaan lingkungan disekitar kamera.

Untuk pengujian kemampuan deteksi dalam beberapa kondisi pencahayaan, diambil 5 sampel pengujian dengan kondisi pencahayaan Terang, Cukup Terang, Redup, Cukup Gelap, dan Gelap. Dari hasil pengujian, didapatkan tingkat kemampuan deteksi sistem yang optimal pada kondisi pencahayaan yang terang merata.

Pengujian Performa fungsional sistem dalam pendeteksian secara realtime

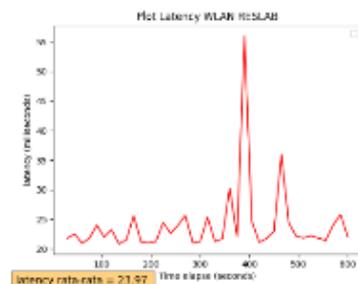
Pengujian dilakukan untuk melihat tingkat performa pendeteksian objek pada perangkat Raspberry Pi 4 secara realtime. Untuk itu dilakukan perhitungan tingkat *fps* (frame per second) saat dilakukan deteksi dan performa perangkat keras Raspberry Pi 4 saat melakukan deteksi. Jenis Raspberry Pi yang digunakan yaitu Raspberry Pi 4 model B+ dengan 4GB RAM. Hasil pengujian performa sistem dapat dilihat pada Gambar 4.19.



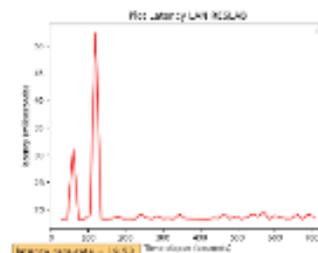
Gambar 15. Grafik hasil perhitungan fps rata-rata selama 1600 detik

Pengujian Kualitas Jaringan ke Database

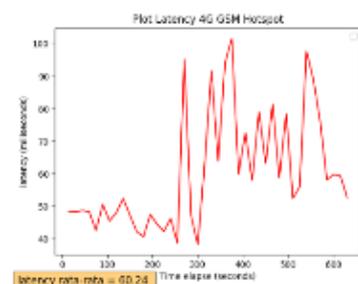
Pada implementasi sistem, ketika terdeteksi adanya objek senjata pada sistem Raspberry Pi 4, maka data objek senjata yang terdeteksi dan jumlahnya akan dikirim ke database firebase melalui jaringan internet. Untuk itu, dilakukan pengujian kualitas jaringan ke database dari beberapa metode konektivitas pada Raspberry Pi. Pengujian dilakukan dengan mengukur waktu latency (lama waktu pengiriman data) antara perangkat Raspberry Pi 4 dengan *server database* firebase dalam selang waktu 600 detik jalannya program. Untuk mengukur kualitas jaringan yang baik, digunakan perbandingan dari beberapa konektivitas jaringan yang tersedia. Beberapa variasi konektivitas yang digunakan yaitu : koneksi *Local Area Network* (LAN), koneksi melalui jaringan Wi-Fi atau *Wireless LAN* melalui Router, dan koneksi Wi-Fi melalui jaringan 4G GSM Hotspot. Untuk mendapatkan perbandingan yang setara, maka untuk lingkungan pengujian jaringan dilakukan pada laboratorium RESLAB. Hasil pengujian kualitas jaringan pada perangkat Raspberry Pi 4 dapat dilihat pada gambar berikut ini.



Gambar 16. Plot latency untuk konektivitas jaringan LAN



Gambar 17. Plot latency untuk jaringan WLAN



Gambar 18. Plot latency untuk jaringan 4G GSM.

Dalam pengujian kualitas jaringan, digunakan akses jaringan Wi-Fi kampus dalam lingkungan yang sama yaitu di laboratorium RESLAB. Dari hasil grafik latency pada gambar 4.21, didapatkan waktu latency rata-rata atau lama pengiriman data ke *database* sebesar 23.97 milisecond pada jaringan WLAN, 19.53 milisecond

pada jaringan LAN, dan 60.24 milisecond pada jaringan 4G GSM. Secara umum, dari hasil pengujian diatas, dapat disimpulkan bahwa sistem mampu melakukan pengiriman dan penerimaan data dari database dengan baik saat terkoneksi ke jaringan yang stabil, dengan lama waktu pengiriman yang dipengaruhi oleh waktu latency jaringan dan kualitas koneksi jaringan.

KESIMPULAN

Dari hasil implemmentasi sistem dan pengujian sistem, maka didapatkan kesimpulan :

1. Dalam pelatihan model YOLO untuk mendeteksi adanya objek senjata pistol dan pisau, YOLOv4 dengan tipe tiny memungkinkan untuk diimplementasikan secara realtime pada perangkat Raspberry Pi 4, dengan nilai fps yang terbatas.
2. Dengan menggunakan model YOLOv4 Tiny, didapatkan akurasi yang dihitung dalam metrik mAP (mean Average Precision) sebesar **84.37%**, dengan tingkat AP (*Average Precision*) untuk kelas pistol sebesar **86.25%** dan AP untuk kelas pisau sebesar **82.49%**.
3. Dari pengujian jarak efektif deteksi, didapatkan jarak efektif untuk objek senjata pistol sejauh **6,5 meter**. Untuk objek senjata pisau didapatkan jarak efektif deteksi sejauh **2 meter**.
4. Dalam pengujian kualitas deteksi saat berada pada kondisi pencahayaan lingkungan yang berbeda, didapatkan hasil optimal pada pencahayaan yang terang dan merata.
5. Pada implementasi model di perangkat Raspberry Pi 4, didapatkan performa yang mencukupi untuk pendeteksian secara realtime melalui input webcam, yang dilihat dari besarnya nilai FPS (frame per second). FPS rata-rata yang didapatkan pada implementasi sistem yaitu sebesar 1.53 fps.
6. Untuk kualitas pengiriman data, didapatkan latency (lama waktu respon ke database) jaringan LAN lebih stabil daripada latency jaringan WLAN, dengan waktu latency rata-rata untuk jaringan LAN sebesar 19.53 milisecond, sedangkan latency untuk jaringan WLAN adalah sebesar 23.97, dan untuk jaringan 4G GSM sebesar 60.24 milisecond
7. Sistem mampu mengirimkan data hasil deteksi dan mengambil data dari database firebase dengan baik saat berada pada jaringan yang stabil, dengan delay pengiriman yang bergantung pada latency jaringan dan kualitas koneksi.

REFERENSI

- [1] Statistik, B. P. 2020. *Statistik Kriminal 2019*. Badan Pusat Statistik. [1-40] Diakses pada tanggal 30 Maret 2020 <https://www.bps.go.id/publication/download.html?nrbvfeve=NjZjMDExNGVhYjc1MTdhMzMwNjM4NzFm&xzmn=aHR0cHM6Ly93d3cuYnBzLmdvLmlkL3B1YmtpY2F0aW9uLzIwMTkvMTIvMTIvNjZjMDExNGVhYjc1MTdhMzMwNjM4NzFmL3N0YXRpc3Rpay1rcmltaW5hbC0yMDE5Lmh0bWw%3D&twoadfnorfeauf=MjAyMC0wOS0wNiAwMD0lOT0zOQ%3D%3D>
- [2] Jung, Yeondae, Andrew P. Wheeler. 2019. *The effect of public surveillance cameras on crime clearance rates*. University of Texas, Dallas. DOI: 10.31219/osf.io/eh5bg.
- [3] Cohen, N., Gattuso, J., MacLennan-Brown, K., Branch, G.B.H.O.S.D. 2009. *CCTV Operational Requirements Manual*. Criminal Justice System Race Unit. The Home Office: London.UK.
- [4] Oyedjeji, A., Salami, A., Folorunsho, O., & Abolade, O. 2020. March 30. *Analysis and Prediction of Student Academic Performance Using Machine Learning*. JITCE (Journal of Information Technology and Computer Engineering).4(01),10-15. <https://doi.org/https://doi.org/10.25077/jitce.4.01.10-15.2020>.
- [5] Grega, M., & Andrzej Metiolanski, P. G. 2016. *Automated Detection of Firearms and Knives in a CCTV Image*. Sensors (Basel). 2016 Jan 1;16(1):47. DOI: 10.3390/s16010047.
- [6] S. S. Kunapuli, P. C. B.H. and U. Singh, "Designing and Implementing a Real Time Weapons Detection System on ODROID-XU4," 2018 IEEE International WIE Conference on Electrical and Computer Engineering (WIECON-ECE), Chonburi, Thailand, 2018, pp. 137-140, DOI: 10.1109/WIECON-ECE.2018.8783187.
- [7] Olmos, Roberto, Siham Tabik, Francisco Herrera. 2017. *Automatic Handgun Detection Alarm in Videos Using Deep Learning*. NEUCOM (NeuroComputing) 18416, Vol 275, 66-72. DOI : <https://arxiv.org/abs/1702.05147>.
- [8] Nofriani, N. (2020, September 30). Machine Learning Application for Classification Prediction of Household's Welfare Status. *JITCE (Journal of Information Technology and Computer Engineering)*, 4(02), 72-82. <https://doi.org/https://doi.org/10.25077/jitce.4.02.72-82.202>
- [9] Redmon, Joseph. Divvala Santosh, Ross Girshick, Ali Farhadi. 2016. *You Only Look Once : Unified Real-Time Object Detection*. arXiv. DOI : arXiv:1506.02640.
- [10] Shung, Koo Ping. 2018. *Accuracy, Precision, Recall or F1 ?* . Diakses pada tanggal 18 Desember 2020 : <https://towardsdatascience.com/accuracy-precision-recall-or-f1-331fb37c5cb9>.
- [11] Solawetz, Jacob. 2020. *What is Mean Average Precision (mAP) in Object Detection*. Roboflow. Diakses pada tanggal 18 Desember 2020 : <https://blog.roboflow.com/mean-average-precision/>.
- [12] Yohanandan, Shivya. 2020. *mAP (mean Average Precision) might confuse you!* . Toward Data Science. Diakses pada tanggal 18 Desember 2020 : <https://towardsdatascience.com/map-mean-average-precision-might-confuse-you-5956f1bfa9e2>.
- [13] Raspberry Pi Foundation. n.d. *Raspberry Pi 4 Tech Specs*. Diakses pada tanggal 6 September 2020 : <https://www.raspberrypi.org/products/raspberry-pi-4-modelb/specifications/?resellerType=home>.