

**KLASIFIKASI TINGKAT ANCAMAN KRIMINALITAS BERSENJATA
MENGGUNAKAN METODE YOLO**

LAPORAN TUGAS AKHIR TEKNIK KOMPUTER

MUHAMMAD ABDUL HADI

1611512012



DOSEN PEMBIMBING :

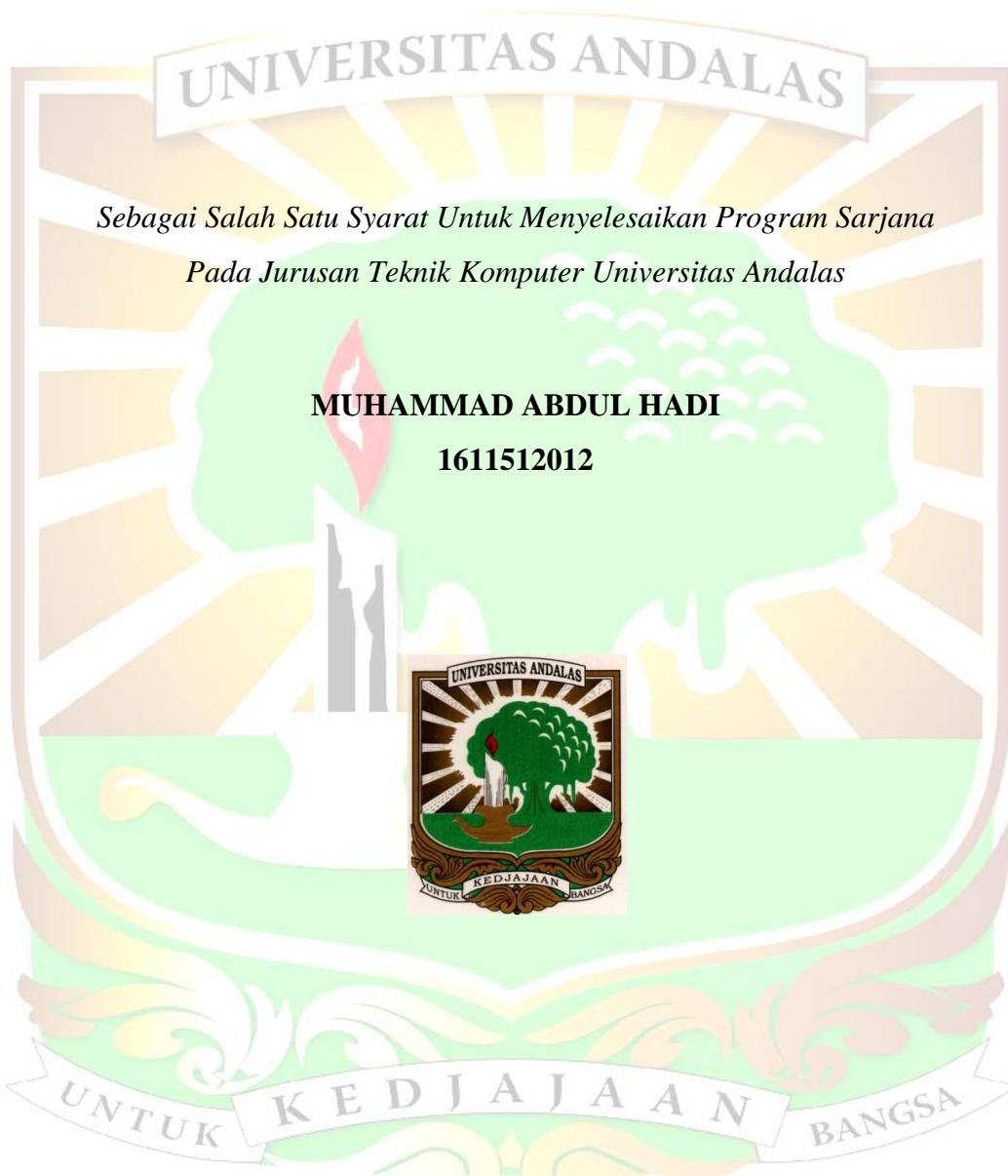
- 1. DR. ENG. RIAN FERDIAN, M.T.**
- 2. LATHIFAH ARIEF, M.T.**

UNTUK KEDAJAAN BANGSA

**JURUSAN TEKNIK KOMPUTER
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS ANDALAS
PADANG
2021**

**KLASIFIKASI TINGKAT ANCAMAN KRIMINALITAS BERSENJATA
MENGGUNAKAN METODE YOLO**

LAPORAN TUGAS AKHIR



**JURUSAN TEKNIK KOMPUTER
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS ANDALAS
PADANG
2021**

HALAMAN PENGESAHAN

Nama : MUHAMMAD ABDUL HADI
No.BP : 1611512012
Judul Tugas Akhir : **KLASIFIKASI TINGKAT ANCAMAN KRIMINALITAS
BERSENJATA MENGGUNAKAN METODE YOLO**

Tugas Akhir ini disetujui oleh Dosen Pembimbing dan disahkan oleh Ketua Jurusan Teknik Komputer, Fakultas Teknologi Informasi Universitas Andalas.

Demikianlah lembaran pengesahan ini dibuat untuk diketahui bersama.

Padang, 23 Maret 2021

Pembimbing 1



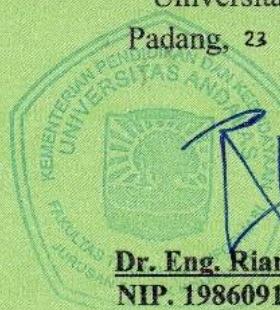
Dr. Eng. Rian Ferdian, M.T.
NIP. 198609162014041001

Pembimbing 2



Lathifah Arief, M.T.
NIP. 198109122014042001

Mengetahui :
Ketua Jurusan Teknik Komputer
Fakultas Teknologi Informasi
Universitas Andalas
Padang, 23 Maret 2021



Dr. Eng. Rian Ferdian, M.T
NIP. 198609162014041001

LEMBARAN PERSETUJUAN TUGAS AKHIR

Dengan ini dinyatakan bahwa :

Nama : MUHAMMAD ABDUL HADI

No.BP : 1611512012

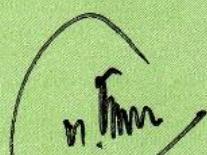
Judul Tugas Akhir : **KLASIFIKASI TINGKAT ANCAMAN KRIMINALITAS
BERSENJATA MENGGUNAKAN METODE YOLO**

Telah diujikan dan telah disetujui Tugas Akhirnya sebagai salah satu syarat untuk memperoleh gelar Sarjana Teknik (S.T) melalui ujian sidang Tugas Akhir yang diadakan pada tanggal 24 Februari 2021 berdasarkan ketentuan yang berlaku.

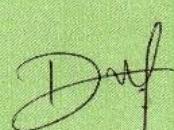
Padang, 17 Maret 2021

Penguji 1

Penguji 2



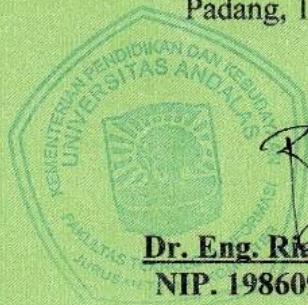
Dodon Vendri, M.Kom.
NIP. 196603091986031001



Desta Yolanda, M.T.
NIDK. 8835411019

Mengetahui :

Ketua Jurusan Teknik Komputer
Fakultas Teknologi Informasi
Universitas Andalas
Padang, 17 Maret 2021



Dr. Eng. Rian Ferdian, M.T.
NIP. 198609162014041001

LEMBARAN PERSETUJUAN TUGAS AKHIR

Tim Pengaji Tugas Akhir Mahasiswa Jurusan Teknik Komputer, menyatakan bahwa :

Nama : MUHAMMAD ABDUL HADI

No.BP : 1611512012

Judul Tugas Akhir : **KLASIFIKASI TINGKAT ANCAMAN KRIMINALITAS
BERSENJATA MENGGUNAKAN METODE YOLO**

Telah diujikan dan telah disetujui Seminar Hasil Tugas Akhirnya pada seminar hasil untuk dilanjutkan pada tahap berikutnya.

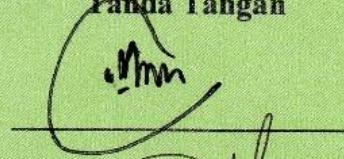
Demikianlah lembaran pengesahan ini dibuat untuk diketahui bersama.

Padang, 23 Maret 2021

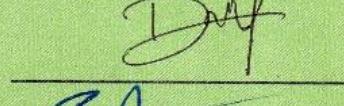
Nama

Tanda Tangan

Pengaji I : Dodon Yendri, M.Kom.



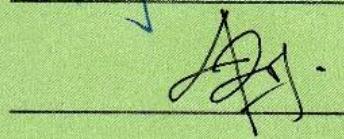
Pengaji II : Desta Yolanda, M.T.



Pembimbing I : Dr. Eng. Rian Ferdian, M.T.



Pembimbing II : Lathifah Arief, M.T.



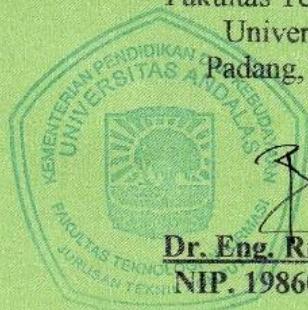
Mengetahui :

Ketua Jurusan Teknik Komputer

Fakultas Teknologi Informasi

Universitas Andalas

Padang, 23 Maret 2021



Dr. Eng. Rian Ferdian M.T.
NIP. 198609162014041001

PERNYATAAN

Dengan ini menyatakan bahwa :

1. Tugas Akhir dengan judul "**Klasifikasi Tingkat Ancaman Kriminalitas Bersenjata Menggunakan Metode YOLO**" adalah asli dan belum pernah diajukan untuk mendapatkan gelar akademik (Sarjana, Magister, dan Doktor), baik di Universitas Andalas maupun di perguruan tinggi lainnya.
2. Tugas Akhir ini murni gagasan dan rancangan saya sendiri, tanpa bantuan tidak sah dari pihak lain, kecuali bantuan dan arahan dari tim pembimbing.
3. Tugas Akhir ini tidak terdapat hasil karya atau pendapat yang ditulis atau dipublikasikan oleh orang lain, kecuali dikutip secara tertulis dengan jelas dan dicantumkan sebagai acuan dalam tulisan saya dengan disebutkan nama pengarang dan dicantumkan dalam daftar pustaka.
4. Pernyataan ini saya buat dengan sesungguhnya, dan apabila dikemudian hari terdapat penyimpangan dan ketidakbenaran dalam pernyataan ini, maka saya bersedia menerima sanksi akademik serta sanksi lainnya sesuai dengan norma dan ketentuan lain yang berlaku.

Demikianlah surat ini dibuat, untuk dapat dipergunakan sebagaimana mestinya.

Padang, 5 Maret 2021

Yang membuat pernyataan,



Muhammad Abdul Hadi
No BP. 1611512012

Kata Pengantar

Puji syukur penulis limpahkan kehadiran Allah SWT yang telah memberikan begitu banyak rahmat dan karunia-Nya sehingga penulis mampu menyelesaikan penelitian Tugas Akhir ini. Shalawat dan salam semoga senantiasa tercurahkan kepada Nabi Muhammad SAW, junjungan hati umat islam hingga kiamat kelak, dan semoga syafaatnya turut menyertai kita di akhirat kelak, aamiin ya rabbal aalamiin.

Dengan selesainya penulisan tugas akhir ini, penulis ingin mengucapkan terima kasih yang sebesar besarnya kepada segenap pihak yang telah memberikan begitu banyak dukungan dan saran dalam menyelesaikan tugas akhir ini. Terkhusus, penulis ingin menyampaikan rasa terima kasih yang sebesar besarnya kepada :

1. Ayah, dan Umi selaku orang tua penulis yang selalu memberikan dukungan dari segala aspek agar penulis dapat terus maju menyelesaikan tantangan-tantangan kehidupan, termasuk dalam penyelesaian tugas akhir ini.
2. Kakak yang selalu pengertian, dan adik yang selalu memperhatikan. Terima kasih atas segala dukungan dan semangatnya.
3. Bapak Dr. Eng. Rian Ferdian, M.T. dan Ibuk Lathifah Arief, M.T. selaku pembimbing I dan pembimbing II dari penulis. Terima kasih atas segala bimbingan, arahan, saran, dan waktu yang diberikan sehingga penulis bisa menyelesaikan tugas akhir ini. Juga atas segala ilmu yang telah diberikan selama masa kuliah penulis.
4. Bapak Dodon Yendri, M.Kom. dan Ibuk Desta Yolanda, M.T. selaku penguji I dan penguji II dalam penelitian tugas akhir ini. Terima kasih atas segala bimbingan, arahan dari Bapak dan Ibuk yang telah memperluas pemahaman dalam penelitian tugas akhir ini.
5. Ibuk Nefy Puteri Novani, M.T. selaku Dosen Pembimbing Akademik yang telah memberikan begitu banyak dukungan selama perjalanan kuliah penulis.
6. Bapak dan Ibuk Dosen Fakultas Teknologi Informasi khususnya Dosen Jurusan Teknik Komputer yang telah memberikan dukungan. Terima kasih yang sebesar-besarnya juga penulis ucapkan atas segala ilmu dan pengetahuan yang diberikan.

7. Karyawan/i di Jurusan Teknik Komputer Fakultas Teknologi Informasi.
8. Teman-teman, kakak tingkat, dan adik-adik dari jurusan Teknik Komputer yang telah memberikan begitu banyak inspirasi dan dukungan.
9. Teman-teman asisten Laboratorium Robotic and Embedded System (RESLAB).



KLASIFIKASI TINGKAT ANCAMAN KRIMINALITAS BERSENJATA MENGGUNAKAN METODE YOLO

Muhammad Abdul Hadi¹, Dr. Eng. Rian Ferdian, M.T², Lathifah Arief, M.T.³

¹Mahasiswa Teknik Komputer Fakultas Teknologi Informasi Universitas Andalas

²Dosen Teknik Komputer Fakultas Teknologi Informasi Universitas Andalas

³Dosen Teknik Komputer Fakultas Teknologi Informasi Universitas Andalas

ABSTRAK

Penelitian ini bertujuan untuk mengenali ancaman kekerasan bersenjata melalui parameter objek senjata yang terdeteksi pada kamera. Untuk metode pengenalan objek senjata digunakan metode YOLO (You Only Look Once) yang diterapkan pada Raspberry Pi 4. Sistem bekerja dengan mendeteksi adanya objek senjata pada kamera dan mengklasifikasikannya diantara 2 kelas senjata : Pistol dan Pisau. Selain melakukan klasifikasi, juga dihitung jumlah objek senjata yang terdeteksi. Saat terdeteksi adanya objek senjata, sistem akan mengirim notifikasi berupa kemungkinan tingkat ancaman ke aplikasi android agar dapat dilakukan penanganan yang cepat oleh operator atau user yang mengawasi. Dari hasil penelitian didapatkan tingkat mAP YOLOv4 tiny sebesar 85.12 % yang diuji dalam ruangan. Dalam penerapannya secara realtime pada Raspberry Pi 4, didapatkan total fps sebesar 1.53. Dari penelitian, didapatkan bahwa penerapan metode YOLO dalam deteksi objek senjata secara realtime dapat diterapkan pada Raspberry Pi 4, dengan nilai fps yang terbatas.

THREAT LEVEL CLASSIFICATION IN ARMED CRIME USING YOLO

Muhammad Abdul Hadi¹, Dr. Eng. Rian Ferdian, M.T², Lathifah Arief, M.T.³

¹Student, Computer Engineering, Faculty of Information and Technology, Andalas University

²Lecturer of Computer Engineering Department, Faculty of Information and Technology, Andalas University

³Lecturer of Computer Engineering Department, Faculty of Information and Technology, Andalas University

ABSTRACT

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.

DAFTAR ISI

LEMBAR PENGESAHAN	i
LEMBAR PERSETUJUAN TUGAS AKHIR.....	ii
LEMBAR PENGESAHAN TUGAS AKHIR	iii
HALAMAN PERNYATAAN.....	iv
KATA PENGANTAR.....	v
ABSTRAK	vi
ABSTRACT	vii
DAFTAR ISI.....	viii
DAFTAR TABEL	xii
DAFTAR GAMBAR.....	xiii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah	3
1.3 Batasan Masalah.....	3
1.4 Tujuan Penelitian.....	4
1.5 Manfaat Penelitian.....	4
1.6 Jenis dan Metodologi Penelitian.....	5
1.7 Sistematika Penulisan.....	8
BAB II LANDASAN TEORI	9
2.1 Kecerdasan Buatan (<i>Artificial Intelligence</i>)	9
2.1.1 Machine Learning	11
2.1.2 Deep Learning	12
2.2 Computer Vision	14
2.3 YOLO (You Only Look Once).....	15
2.3.1 Arsitektur YOLO	16
2.3.2 Training YOLO.....	17
2.3.4 Keunggulan dari YOLO.....	17
2.3.5 Keterbatasan dari YOLO	18

2.3.6 YOLOv2.....	18
2.3.7 YOLOv3.....	19
2.3.9 YOLOv4.....	21
2.4 Pengujian Kualitas Model dengan <i>Medium Average Precisions</i> (mAP)....	22
2.5 Dataset	25
2.6 Raspberry Pi	26
2.7 Webcam.....	27
2.8 Buzzer.....	28
BAB III RANCANGAN SISTEM	29
3.1 Analisa Kebutuhan Sistem	29
3.1.1 Kebutuhan Fungsional Sistem	29
3.1.1 Kebutuhan Non-Fungsional Sistem	29
3.1.2 Kebutuhan Perangkat Keras.....	30
3.1.3 Kebutuhan Perangkat Lunak.....	31
3.1.4 Kebutuhan Data.....	31
3.2 Rancangan Umum Sistem	32
3.3 Rancangan Proses	33
3.3.1 Rancangan Perangkat Keras.....	35
3.3.2 Rancangan Perangkat Lunak.....	37
3.3.2.1 Modul Pelatihan (<i>Training</i>) dan Pengujian Model	37
3.3.2.2 Implementasi Sistem pada Raspberry Pi 4.....	42
3.3.2.3 Aplikasi android untuk menerima notifikasi dan data	45
3.4 Rencana Pengujian	49
3.4.1 Rencana Pengujian Model YOLO	50
3.4.1.1 Skenario Pengujian	50
3.5.2 Rencana Pengujian Fungsional Sistem	51
3.5 Analisis Kebutuhan Penelitian	52
BAB IV IMPLEMENTASI DAN PENGUJIAN	53
4.1 Implementasi	53
4.1.1 Implementasi Perangkat Keras.....	53
4.1.2 Implementasi Perangkat Lunak.....	54
4.1.2.1 Modul Pelatihan dan Pengujian Model YOLO	54

4.1.2.2 Modul implementasi model pendekripsi objek senjata YOLO pada Raspberry Pi 4	57
4.1.2.3 Implementasi Aplikasi Android	58
4.1.2.4 Implementasi Database Firebase	63
4.2 Pengujian dan Analisa	66
4.2.1 Pengujian dan Analisa Kualitas model YOLO	66
4.2.2 Pengujian dan Analisa Fungsional Sistem.....	71
4.2.2.1 Kinerja Sistem dalam melakukan deteksi objek secara <i>realtime</i> ...	72
4.2.2.1.1 Pengujian Jarak Efektif Deteksi	72
4.2.2.1.2 Pengujian kemampuan deteksi dalam beberapa variasi pencahayaan lingkungan	77
4.2.2.1.3 Pengujian kemampuan sistem dalam mendeteksi jumlah senjata	79
4.2.2.1.4 Pengujian kinerja sistem dalam mendeteksi objek selain senjata	82
4.2.2.2 Performa Sistem dalam pendekripsi secara <i>realtime</i>	84
4.2.2.3 Kualitas Jaringan ke Database	86
BAB V PENUTUP	90
5.1 Kesimpulan.....	90
5.2 Saran	90
DAFTAR PUSTAKA	91

Daftar Tabel

Tabel 2.1 Confusion Matrix dari Pengukuran metrik Presisi (<i>Precisions</i>)	22
Tabel 2.2 Confusion Matrix dari Pengukuran metrik Sensitivitas (<i>Recall</i>)	23
Tabel 3.1 Susunan dan Kategori Dataset	37
Tabel 3.2 Pengujian performa deteksi dengan metode <i>confusion matrix</i>	50
Tabel 3.3 Skenario pengujian tingkat performa dan akurasi model YOLO	50
Tabel 3.4 Rencana Pengujian Fungsional Sistem	51
Tabel 3.5 Kebutuhan Penelitian	52
Tabel 4.1 Tabel Confusion Matrix hasil pengujian.....	68
Tabel 4.2 Hasil pengujian jarak efektif deteksi.....	74
Tabel 4.3 Pengujian kinerja deteksi objek terhadap beberapa variasi pencahayaan lingkungan.....	77
Tabel 4.4 Pengujian deteksi pada beberapa objek senjata	79
Tabel 4.5 Pengujian sistem pada objek selain senjata.....	82
Tabel 4.6 Hasil pengujian sistem secara keseluruhan	88

Daftar Gambar

Gambar 1.1 Rancangan Penelitian	6
Gambar 2.1 Pembagian dari kecerdasan buatan atau AI.....	10
Gambar 2.2 Komponen utama dari <i>machine learning</i>	11
Gambar 2.3 Perbedaan <i>Machine Learning</i> dan <i>Deep Learning</i>	13
Gambar 2. 5 Arsitektur YOLO.....	16
Gambar 2. 6 kalkulasi koordinat Bounding Box pada YOLO	17
Gambar 2. 7 Arsitektur Darknet pada YOLOv3	20
Gambar 2.8 Arsitektur YOLOv4.....	21
Gambar 2. 9 Kurva metrik precisions-recall	23
Gambar 2.10 Sketsa proses penentuan bounding boxes dengan metrik IoU	24
Gambar 2.11 Raspberry Pi 4	27
Gambar 2.12 Modul Buzzer.....	28
Gambar 3. 1 Bagan rancangan umum sistem.....	32
Gambar 3.2 Alur Proses secara keseluruhan.....	34
Gambar 3.3 Perancangan Perangkat Keras	36
Gambar 3.4 Alur <i>flowchart</i> program <i>Training</i>	39
Gambar 3.5 Alur <i>flowchart</i> program pengujian model.....	41
Gambar 3.6 <i>flowchart</i> modul implementasi sistem pada raspberry Pi.....	44
Gambar 3.7 Alur proses aplikasi android.....	46
Gambar 3.8 Rancangan Antarmuka homepage.....	47
Gambar 3.9 Rancangan antarmuka video streaming.....	47
Gambar 4.1 (a) Perangkat Keras yang digunakan untuk penelitian, dan (b) tempat pemasangan perangkat keras pada lingkungan pengujian	53
Gambar 4.2 <i>Training</i> model YOLOv4 Tiny pada Google Collabs.....	55
Gambar 4.3 Proses pelatihan model YOLOv4 tiny untuk mengenali objek senjata pisau dan pistol.....	56
Gambar 4.4 Plot hasil pelatihan model YOLO dan hasil perhitungan mAP selama pelatihan	57
Gambar 4.5 Implementasi sistem pendekripsi senjata YOLO pada Raspberry Pi 4	58
Gambar 4.6 Tampilan halaman login aplikasi	59
Gambar 4.7 Tampilan Homepage dari aplikasi.....	60
Gambar 4.8 Tampilan halaman camera	61
Gambar 4.9 Tampilan menu recorded file, (a) gambar riyawat hasil deteksi, dan (b) pengelompokan gambar berdasarkan tanggal dan waktu deteksi.....	62
Gambar 4.10 <i>Push notification</i> dari aplikasi android.....	63
Gambar 4.11 Tabel-tabel yang digunakan pada database	64
Gambar 4.12 Implementasi tabel kamera.....	64
Gambar 4.13 Struktur tabel <i>recorded</i> pada database firebase.....	65
Gambar 4.14 Struktur tabel user pada database Firebase	65

Gambar 4.15 struktur penyimpanan file gambar pada Firebase Storage	66
Gambar 4.16 pengujian performa model pada data-data gambar dari lingkungan pengujian.....	67
Gambar 4.17 Hasil pengukuran mAP dari model YOLO dengan data gambar dari lingkungan pengujian	71
Gambar 4.18 Beberapa sampel gambar uji yang digunakan untuk melakukan pengujian jarak deteksi efektif	73
Gambar 4.19 Grafik hasil perhitungan fps rata-rata selama 1600 detik	84
Gambar 4.20 (a) Grafik utilisasi perangkat Raspberry Pi 4 Model B+ ketika menjalankan program deteksi objek senjata, (b) temperatur CPU Raspberry Pi 4 dalam menjalankan program deteksi objek senjata.....	85
Gambar 4.21 (a) plot latency untuk konektivitas jaringan LAN, (b) plot latency untuk jaringan WLAN, dan (c) plot latency untuk jaringan 4G GSM.....	86



BAB I

PENDAHULUAN

1.1 Latar Belakang

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 serangan 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 dari 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 pendekripsi 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 pendekripsi secara *realtime*[7]. Dari beberapa penelitian yang ada diatas, implementasi sistem untuk memberikan notifikasi dan respon terhadap pendekripsi 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 *protoype* 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.

1.2 Rumusan Masalah

Rumusan masalah pada penelitian ini adalah :

1. Bagaimana merancang sistem pendekripsi objek senjata dan notifikasi peringatan dini dengan menggunakan kamera dan Raspberry Pi.
2. Bagaimana melatih model YOLO untuk mendekripsi objek senjata dan mengklasifikasikan level ancaman berdasarkan jenis senjata dan jumlah senjata yang terdeteksi.
3. Bagaimana mengimplementasikan sistem deteksi dan peringatan dini ancaman penyerangan bersenjata pada kamera Raspberry Pi secara *realtime* pada *live* kamera.

1.3 Batasan Masalah

1. Model YOLO dilatih untuk mengenali 2 kategori senjata, senjata pistol dan senjata pisau.
2. Sistem diterapkan pada Raspberry Pi 4 dengan sensor kamera 5 MP, sehingga tangkapan frame untuk pengujian terbatas dalam hal jarak deteksi dan kejelasan gambar.
3. Kategori ancaman yang diterapkan yaitu normal, sedang, dan kritis berdasarkan parameter yang terbatas, yaitu jenis senjata dan jumlah senjata.

Tingkat ancaman diukur berdasarkan kemungkinan dampak yang ditimbulkan dari senjata.

4. Sistem melakukan deteksi dan klasifikasi objek senjata secara visual pada frame gambar, sehingga sistem tidak mampu membedakan kategori senjata asli atau bukan, karena model dilatih untuk mengenali pola bentuk senjata.
5. Aspek realtime dari sistem adalah *soft realtime* dengan batasan waktu yang ditetapkan untuk toleransi batas degradasi sistem.

1.4 Tujuan Penelitian

Adapun tujuan yang ingin dicapai penulis dalam penelitian ini adalah :

1. merancang sistem pendekripsi objek senjata dan notifikasi peringatan dini menggunakan kamera dan Raspberry Pi
2. melatih model YOLO untuk mendekripsi objek senjata dan mengklasifikasikan level ancaman berdasarkan jenis senjata dan jumlah senjata yang terdeteksi.
3. Dapat mengimplementasikan sistem deteksi dini penyerangan bersenjata pada kamera IP Raspberry Pi.
4. Mendapatkan hasil deteksi objek senjata yang akurat
5. Sistem dapat memberikan respon cepat dan *realtime* dalam pengiriman data.

1.5 Manfaat Penelitian

Manfaat yang ingin dicapai dalam penelitian ini adalah :

1. Memberikan peringatan dini kepada petugas dan aparat keamanan dalam pencegahan dan penanganan tindakan kriminal yang berpotensi menimbulkan korban jiwa.
2. Mencegah dan mengurangi kasus tindakan kekerasan bersenjata dan memberikan rasa aman kepada masyarakat.
3. Mengetahui prinsip kerja deteksi objek dan algoritma deteksi objek.
4. Mengurangi kesalahan operator dalam pengawasan CCTV.
5. Mencoba mengurangi dan mencegah terjadinya kekerasan bersenjata di lingkungan masyarakat.

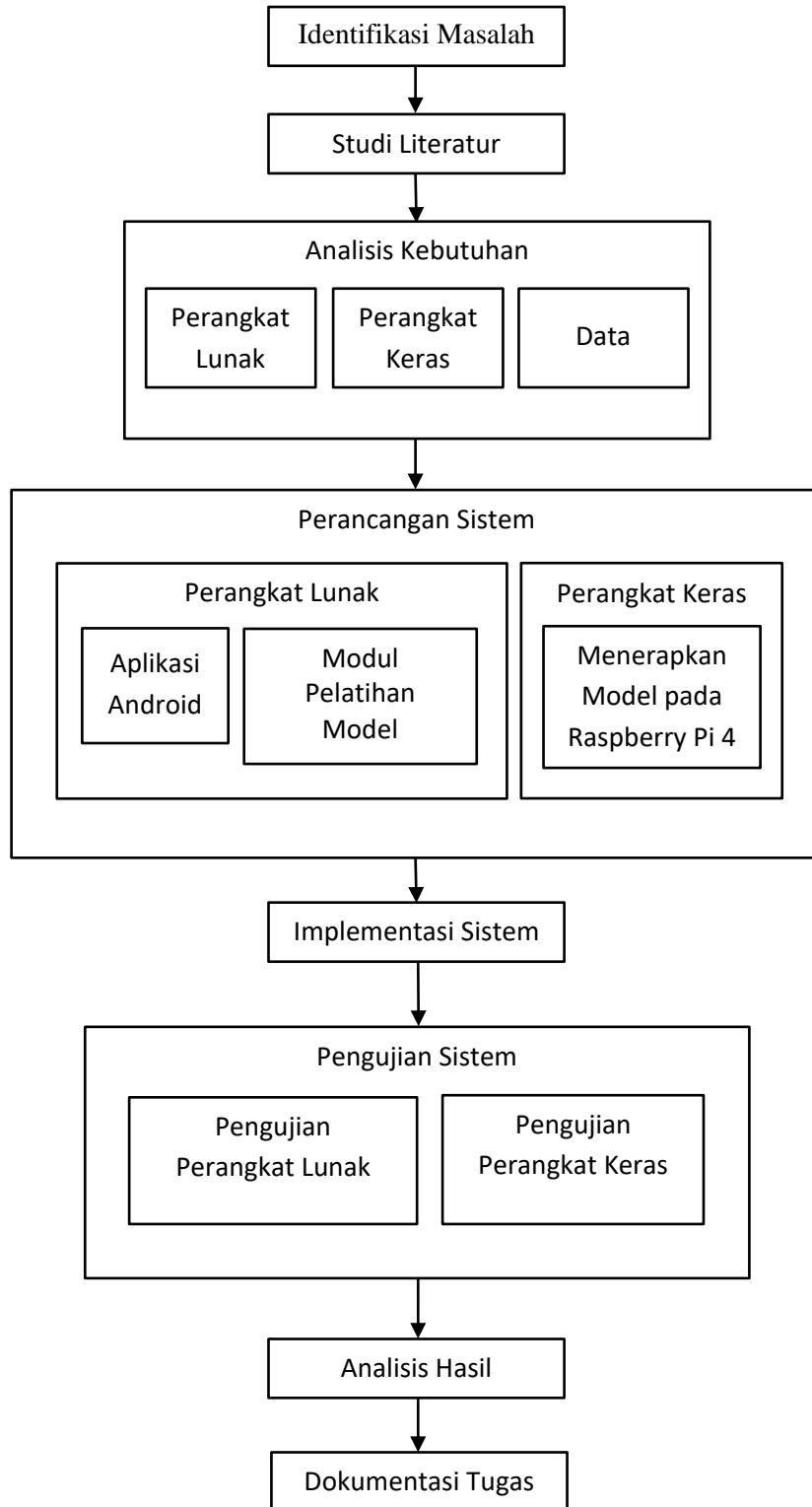
1.6 Jenis dan Metodologi Penelitian

Penelitian yang dilakukan adalah penelitian eksperimental. Dalam penelitian eksperimental, dilihat hubungan sebab akibat. Penelitian ini bertujuan untuk mempelajari parameter masukan dan melakukan klasifikasi keluaran berdasarkan parameter masing-masing yang sudah ditetapkan. Tahapan-tahapan yang dilakukan pada penelitian ini yaitu :

1. Melakukan pelatihan model YOLO untuk mengenali objek senjata dan melakukan pengujian terhadap model YOLO yang telah dilatih.
2. Setelah mendapatkan hasil pelatihan model YOLO untuk mengenali objek senjata, model diimplementasikan pada Raspberry Pi 4 untuk menguji penerapan model pada lingkungan nyata. Beberapa hal yang diterapkan yaitu mengenali objek senjata pada Raspberry Pi, mengkategorikan level ancaman, dan mengirimkan notifikasi kepada aplikasi android jika terdeteksi adanya senjata
3. Sistem monitoring pada aplikasi android

Untuk lebih jelasnya, skema tahapan-tahapan pada penelitian dapat dilihat pada gambar 1.1.





Gambar 1.1 Rancangan Penelitian

1. Identifikasi Masalah

Melakukan identifikasi permasalahan yang diangkat menjadi penelitian tugas akhir. Proses penelitian dilakukan dengan mengenali adanya objek senjata berpa pistol dan pisau melalui kamera. Selanjutnya akan dikirim notifikasi ke aplikasi android jika terdeteksi adanya objek senjata.

2. Studi Literatur

mencari serta mengumpulkan artikel dan jurnal dari penelitian sebelumnya yang berkaitan dengan penelitian ini. Selain itu juga mempelajari teori-teori pendukung yang berkaitan dengan pembuatan tugas akhir ini.

3. Perancangan Sistem

Terdapat dua jenis perancangan dalam sistem ini, yaitu perancangan perangkat keras dan perancangan perangkat lunak.

4. Implementasi Sistem

Rancangan penelitian diimplementasikan dalam bentuk sistem utuh yang terdiri dari perangkat keras dan perangkat lunak. Dalam rancangan penelitian ini, digunakan perangkat Raspberry Pi 4 untuk implementasi sistem.

5. Pengujian Sistem

Tahapan ini bertujuan untuk mengukur tingkat kerberhasilan sistem. Pengujian yang dilakukan adalah menguji performa deteksi model YOLO dan tingkat akurasi model YOLO. Selanjutnya menguji performa deteksi secara realtime pada Raspberry Pi 4 dan kecepatan deteksi objek senjata.

6. Analisis Hasil

Melakukan analisis terhadap hasil pengujian yang didapatkan. Setelah itu dilakukan analisis sistem berdasarkan rumusan masalah yang dirancang.

7. Dokumentasi Tugas Akhir

Membuat dokumentasi dan pelaporan hasil penelitian Tugas Akhir.

1.7 Sistematika Penulisan

Secara garis besar, tugas akhir ini dibagi menjadi beberapa bab. Adapun bab-bab tersebut adalah:

BAB I PENDAHULUAN : Bab ini berisi tentang latar belakang, perumusan masalah, batasan masalah, dan tujuan penelitian.

BAB II LANDASAN TEORI : Bab ini menguraikan teori dasar yang mendukung penelitian tugas akhir ini.

BAB III PERANCANGAN SISTEM : Bab ini berisi tentang analisa dan desain sistem secara terstruktur.

BAB IV IMPLEMENTASI DAN PENGUJIAN : Bab ini berisi analisis terhadap hasil kerja dari perancangan dan keluaran alat.

BAB V PENUTUP : Bab ini berisi simpulan dari hasil penelitian dan saran yang disampaikan penulis berdasarkan hasil analisis dan pembahasan dari penelitian.



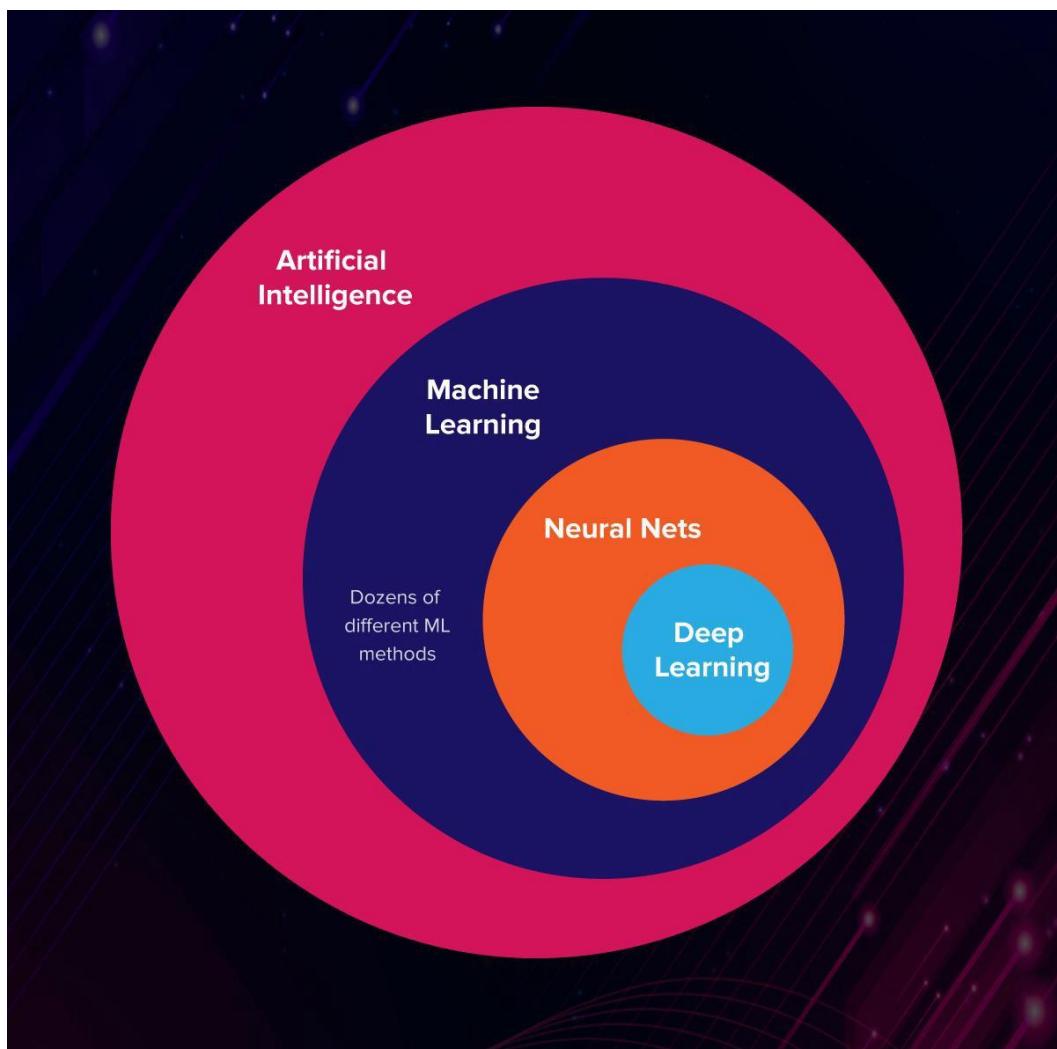
BAB II

LANDASAN TEORI

2.1 Kecerdasan Buatan (*Artificial Intelligence*)

Kecerdasan buatan adalah ilmu yang mempelajari cara untuk membangun program yang cerdas dan mesin yang dapat menyelesaikan masalah, dengan pendekatan prerogatif manusia atau pola pikir manusia. John McCarthy, yang dikenal dengan salah satu bapak penemu kecerdasan buatan atau AI, mendefenisikan AI sebagai ilmu dan teknik rekayasa untuk membangun mesin cerdas. Hal ini dicapai dengan mensimulasikan kecerdasan manusia. Istilah AI sendiri pertama kali digunakan pada tahun 1956 pada konferensi *computer science* di Dartmouth. AI dideskripsikan sebagai model dari otak manusia untuk menghasilkan komputer yang lebih maju dan cerdas. Dalam perkembangannya, AI telah banyak berkembang dan diterapkan pada berbagai sistem cerdas seperti Natural language processing, deteksi objek, machine learning, dan beberapa hal yang menyangkut kreatifitas dan keahlian manusia[8].

Dalam penerapan dan implementasi kecerdasan buatan, ada beberapa istilah yang umum digunakan dalam memberikan gambaran serta tingkat kompleksitas dari pelatihan dan parameter yang diterapkan. Salah satu subset atau bagian dari AI yaitu *machine learning* dan *deep learning*. Jika digambarkan secara visual, tingkatan dan kompleksitas dari AI dapat dilihat pada gambar 2.1

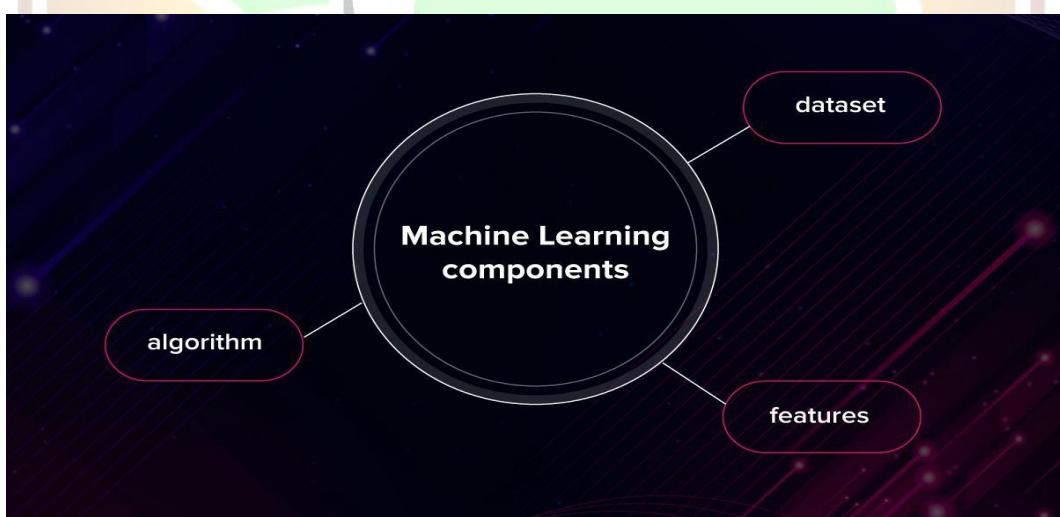


Gambar 2.1 Pembagian dari kecerdasan buatan atau AI[8]

Dari gambar 2.1 dapat dilihat istilah Artificial Intelligence (AI) sendiri merujuk pada konsep kecerdasan buatan secara umum dan luas, yang mencakupi berbagai aspek sistem cerdas dan *machine learning*. Di tingkat yang lebih rinci, istilah *machine learning* yang juga merupakan bagian dari AI mendefenisikan mesin yang mampu belajar dan berkembang dari data-data yang berikan tanpa benar-benar diprogram secara langsung. Dalam penerapannya, *machine learning* melakukan pelatihan dari data-data inputan dengan sistem jaringan saraf tiruan yang dikenal dengan Neural Network untuk menentukan nilai bobot dan nilai berat dari data. Jika penerapan layer dari Neural Network banyak dan kompleks, maka perlu optimasi dan data yang lebih besar untuk melakukan pelatihan, yang dikenal dengan metode *Deep Learning*[8].

2.1.1 Machine Learning

Machine learning merupakan subset atau bagian dari AI yang fokus pada pembelajaran komputer dan mesin untuk dapat melakukan suatu tugas spesifik tanpa benar-benar diprogram sebelumnya untuk melakukan itu. Kunci utama dari *machine learning* yaitu Mesin atau model yang menghasilkan algoritma, dimana mesin belajar dari data dan bisa membuat prediksi untuk data inputan setelahnya. Untuk dapat mengedukasi mesin, diperlukan 3 komponen dasar : dataset, fitur, dan algoritma machine learning. Dataset merupakan kumpulan dari data-data sampel, dapat berupa angka, gambar, teks dan jenis data lainnya. Perlu dataset yang baik agar dapat melatih model *machine learning* yang akurat. Fitur merupakan bagian data penting yang menjadi kunci utama dari pembelajaran. Fitur mendemonstrasikan fokus utama dalam melakukan penyelesaian masalah. Algoritma *machine learning* merupakan langkah-langkah yang dicapai dalam melakukan pelatihan dan edukasi, tergantung dari algoritma yang dipakai, hasil dan performanya dapat berbeda[9].



Gambar 2.2 Komponen utama dari *machine learning*[9]

Jika dikategorikan, terdapat 3 kategori *machine learning* dalam hal menangani data dan metode menghasilkan output :

1. *Supervised Learning*

Model *machine learning* mengambil fitur ini sebagai input dan berdasarkan parameter internal, fitur menghasilkan prediksi untuk harga output. Jenis

pembelajaran yang diawasi ini disebut regresi. Ini ditandai dengan output merupakan nilai kontinu. Secara sederhananya supervised learning menerapkan data input yang telah dilabeli dan mengharapkan output yang sudah diketahui dalam pelatihan. Dengan pengawasan dan penyesuaian yang tepat, maka akan didapatkan hasil yang akurat dari pelatihan.

2. *Unsupervised Learning*

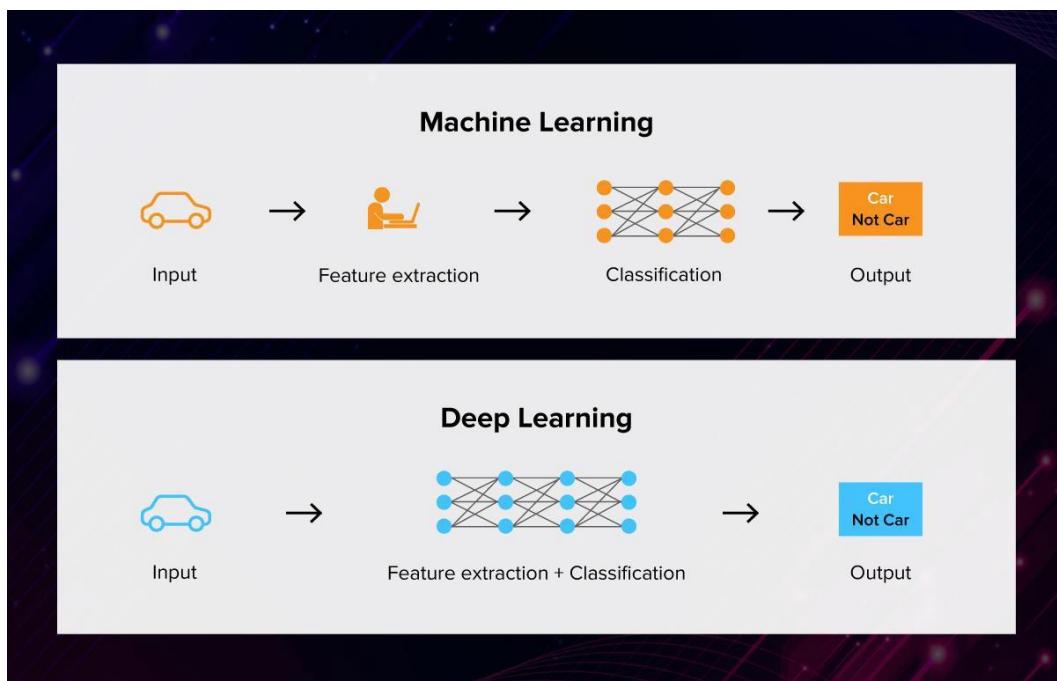
Algoritma *unsupervised learning* mencoba membangun model matematika untuk menemukan pola dalam data. Data yang diberikan kepada model yang tidak diawasi dalam fase pelatihan hanya berisi fitur input dan tidak ada label output yang diinginkan. Algoritma kemudian dibiarkan sendiri untuk mengeksplorasi dan menemukan struktur penting dalam data dan mengelompokkan input ke dalam kelompok (kategori).

3. *Reinforcement Learning*

Dicapai berdasarkan imbalan positif atau negatif untuk tindakan agen di lingkungan. Mengingat kondisi agen dengan lingkungannya, agen hanya mengambil tindakan yang akan memaksimalkan *reward* atau akan mengeksplorasi kemungkinan baru jika *reward* yang didapatkan mengecil. Hasil dari tindakan ini kemudian dimasukkan kembali ke agen, dan berdasarkan itu akan mengubah parameter. Langkah ini diulang berkali-kali untuk meningkatkan perilaku agen untuk keputusan di masa mendatang.

2.1.2 Deep Learning

Deep learning merupakan subset atau bagian dari *machine learning* yang terinspirasi dari struktur otak manusia. *Deep learning* menggunakan jaringan saraf tiruan yang kompleks dan terdiri dari banyak lapisan neuron. Perbedaan dasar dari *machine learning* dan *deep learning* dapat dilihat pada ilustrasi gambar 2.3



Gambar 2.3 Perbedaan *Machine Learning* dan *Deep Learning*[9]

Pada jaringan saraf tiruan (*neural network*), informasi diteruskan dari satu *layer* ke *layer* berikutnya melalui koneksi antar neuron yang disebut *weight*. *Weight* memiliki nilai yang diubah ubah seiring banyaknya data dan pelatihan yang dilakukan pada jaringan saraf tersebut. Dalam rangka melatih model *neural network* tersebut, diperlukan data latih dalam jumlah yang besar karena pada *deep learning* ada banyak parameter yang mendapatkan solusi yang akurat. Secara sederhana, *deep learning* dapat dilihat sebagai *machine learning* yang memiliki lebih banyak *layers* dan memiliki kompleksitas algoritma yang lebih besar. Beberapa aplikasi dari *deep learning* yaitu *speech recognition*, *natural language processing*, deteksi objek, dan banyak lainnya. Dalam penerapan jaringan saraf yang tediri dari beberapa *layers*, dibutuhkan algoritma dan optimasi agar didapatkan output yang sesuai dan akurasi yang tinggi. Ada banyak algoritma yang digunakan dalam melatih model *machine learning*. Pada kategori pendekripsi objek seperti senjata, algoritma yang umum digunakan yaitu *Convolutional Neural Network* (CNN) yang terus dioptimasi dan menghasilkan berbagai algoritma lainnya seperti Regional CNN (R-CNN), Faster R-CNN, dan Mask R-CNN. Algoritma lainnya yang sering digunakan dalam melatih model *deep learning* yaitu You Only Look Once (YOLO), dan telah banyak diterapkan pada sistem deteksi objek secara *realtime*.

karena kecepatan dan tingkat akurasi yang tinggi. Algoritma algoritma diatas memiliki penerapan yang unggul di bidangnya masing-masing, dan tidak selalu ada yang lebih baik dari yang lainnya. Namun, seiring dengan banyaknya penilitian dan percobaan yang dilakukan untuk melatih model CNN dan kebutuhan akan kecepatan dan akurasi, menghasilkan banyak optimasi dari algoritma-algoritma sebelumnya. Pada penelitian ini, karena kebutuhan akan tingkat akurasi dan kecepatan deteksi untuk penerapan secara *realtime*, maka digunakan algoritma YOLO untuk pelatihan model[9].

2.2 Computer Vision

Computer Vision (CV) merupakan subset dari AI dalam bidang yang lebih spesifik, yaitu mengolah informasi dari gambar digital, video, dan input visual lainnya . Hal ini dilakukan dengan penerapan model *deep learning* untuk mengidentifikasi dan mengklasifikasikan objek, sehingga mampu memperoleh keputusan dari apa yang dilihat oleh mesin[10]. Awal diperkenalkannya CV bermula pada tahun 1950, dengan menggunakan *neural network* pertama untuk mendeteksi sudut dari suatu objek pada gambar dan mengkategorikannya kepada bentuk lingkaran dan persegi. Pada tahun 1970-an, penggunaan CV secara komersial digunakan pada penerjemah tulisan tangan dengan pengenalan karakter secara optikal. Penelitian ini digunakan untuk menerjemahkan tulisan tangan menjadi karakter bradley. Ketika memasuki era internet pada tahun 1990-an, terdapat banyak persebaran data gambar untuk dianalisis[8]. Dengan semakin banyaknya kumpulan data yang tersedia, muncul kemampuan untuk mengidentifikasi objek dan wajah manusia pada foto dan video. Pada saat ini, dengan berbagai faktor dan kemajuan teknologi yang semakin berkembang, memungkinkan banyaknya eksplorasi dan penerapan di bidang *Computer Vision*. Kemajuan di berbagai bidang teknologi dan komputasi membuat penerapan CV menjadi lebih mudah dan lebih cepat. Munculnya algoritma optimasi baru dan kemampuan komputasi yang semakin kompleks telah meningkatkan akurasi dari klasifikasi objek dari 50% hingga 99% dalam kurun waktu beberapa tahun. Dan saat ini, kemampuan sistem dalam klasifikasi objek bisa lebih baik daripada manusia dalam menganalisis input visual[11].

Ada banyak penelitian yang dilakukan pada bidang CV, dan banyak implementasi dari CV yang diterapkan pada berbagai sektor bisnis, hiburan, transportasi, kesehatan, dan dalam penunjang aktivitas. Kunci utama dari pesatnya perkembangan ini yaitu aliran data yang semakin mudah dan tersebar luas. Akses informasi yang mudah melalui internet dan semakin pesatnya kemunculan sensor visual pada smartphone, CCTV, televisi, dan sensor optikal lainnya membuat pertumbuhan CV melaju cepat dengan standar penerapan yang lebih tinggi dan kebutuhan yang lebih kompleks. Dalam penelitian ini, CV berperan dalam pengambilan data input dan pengolahan data gambar untuk mendeteksi objek dan mengklasifikasikannya ke kelas yang ada[12].

2.3 YOLO (You Only Look Once)

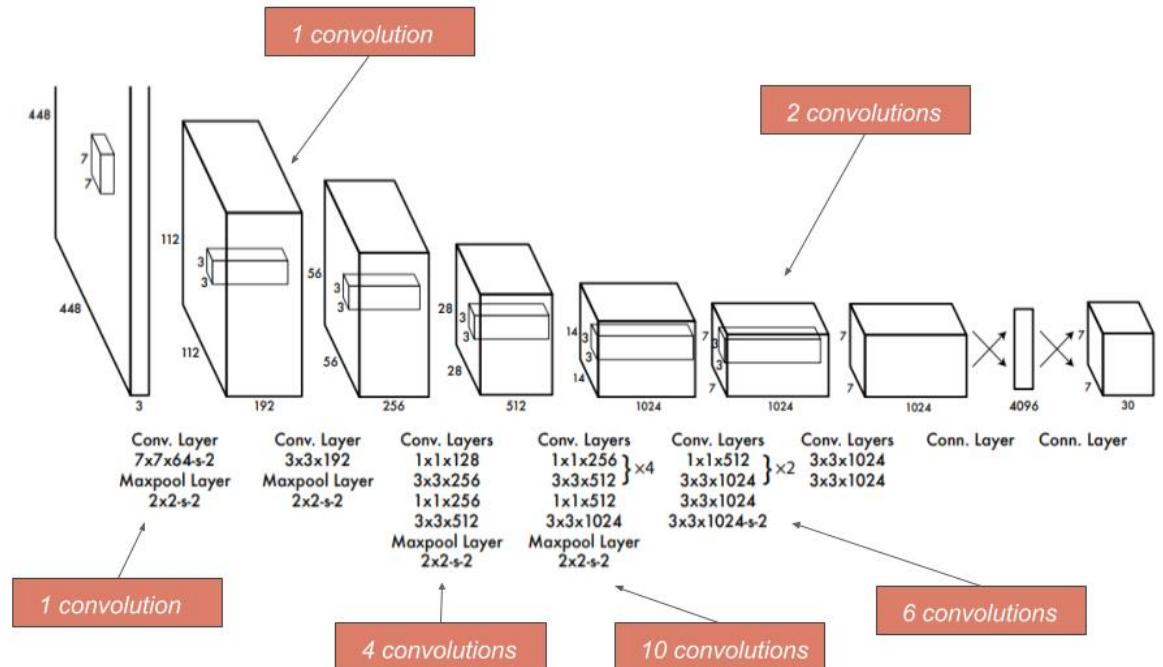
You Only Look Once (YOLO) merupakan salah satu arsitektur dalam algoritma *machine learning* dengan spesifikasi untuk mendeteksi objek pada frame. YOLO pertama kali diperkenalkan pada tahun 2015 oleh Joseph Redmon, Santosh Divvala, dan Ross Girshick dalam jurnal *You Only Look Once : Unified, Real-Time Object Detection*[13]. Dalam jurnal yang dipublikasikan tersebut, YOLO direpresentasikan sebagai pendekatan baru dalam pendekripsi 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.

Metode pendekripsi objek konvensional bergantung pada metode klasifikasi, dimana klasifikasi berjalan pada *sliding window*, atau pada *region proposal network* untuk menghasilkan kemungkinan letak *bounding box*, algoritma yang mengadopsi metode ini yaitu R-CNN, Fast R-CNN, Faster R-CNN, dan Mask R-CNN. Metode pendekripsi ini berjalan dengan baik dan dengan terus berkembangnya optimasi dari algoritma tersebut semakin meningkatkan akurasi

deteksi dan kecepatan prosesnya. Namun, karena kompleksitas arsitektur dan pipeline dari algoritma tersebut membuat kecepatan deteksinya lambat, terutama dalam penerapan deteksi objek secara realtime. Faster R-CNN mampu mencapai tingkat rata rata presisi sebesar 73.2 mAP (*mean Average Precision*), namun dalam sisi kecepatan hanya dapat mengakomodasi 7 *frame per second* (fps). Hal ini membuatnya sulit diterapkan pada deteksi objek secara realtime. Dalam hal kecepatan deteksi, YOLO mampu mengungguli kecepatan algoritma lain sebesar 45 fps dengan mempertahankan akurasi sebesar 63.4 mAP.

2.3.1 Arsitektur YOLO

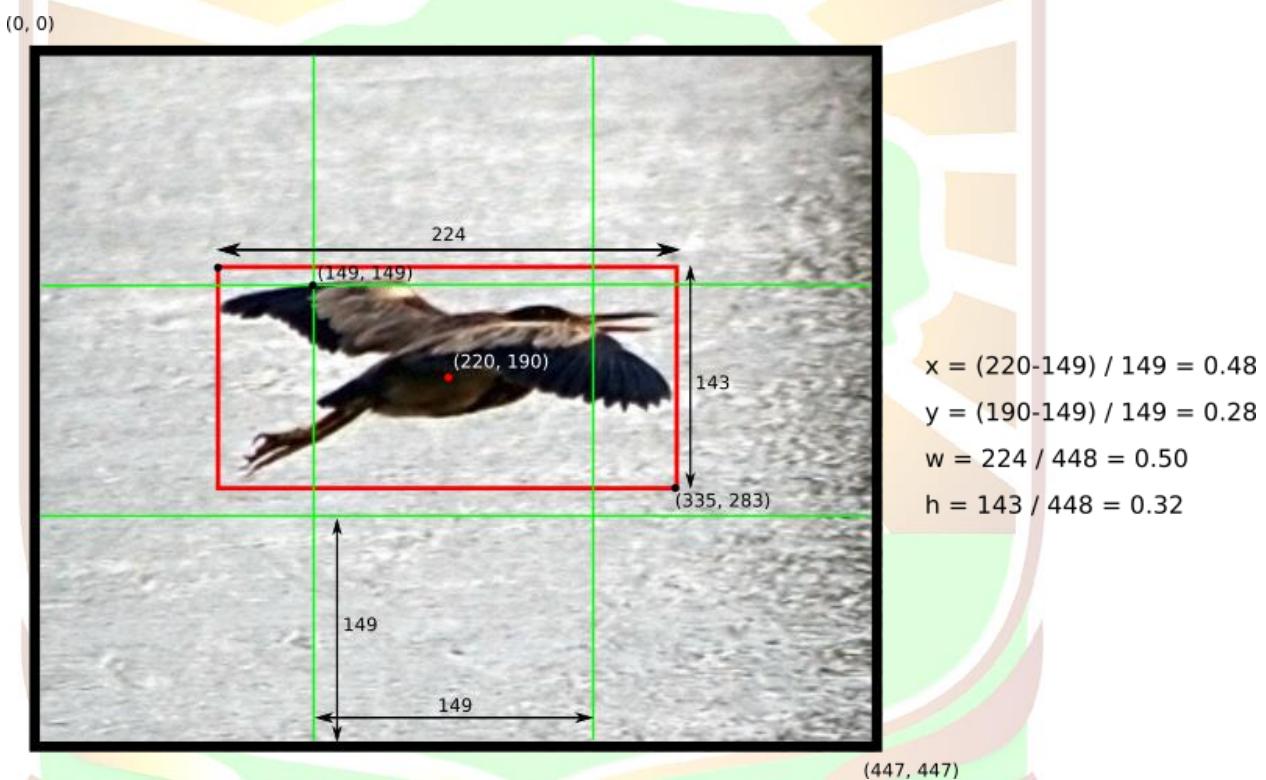
Arsitektur YOLO secara garis besar dipengaruhi oleh backbone GoogleLeNet, dengan jaringan saraf yang terdiri dari 24 *convolution layers* untuk melakukan ekstraksi fitur dan diikuti oleh 2 FCN (*Fully Connected Layer*) untuk melakukan prediksi koordinat *bounding box* dan klasifikasi kelas objek.



Gambar 2. 4 Arsitektur YOLO[13]

2.3.2 Training YOLO

Pelatihan model pendekripsi objek menggunakan YOLO terdiri dari 2 tahapan. Tahapan pertama yaitu melakukan klasifikasi pada gambar input dengan ukuran resolusi di set ke 224x224 menggunakan ImageNet. Tahapan ini menggunakan 20 *convolutional layers* pertama dan diikuti oleh *average pooling* dan FCN. Tahapan kedua yaitu melatih jaringan saraf untuk mendekripsi objek dengan menambahkan 4 *Convolutional layers* dan 2 FCN. Tahapan ini menggunakan resolusi input sebesar 448x448.



Gambar 2. 5 kalkulasi koordinat Bounding Box pada YOLO[13]

2.3.4 Keunggulan dari YOLO

Setelah melihat arsitektur dan cara kerja YOLO dalam melakukan deteksi objek pada gambar, maka dapat dilihat keunggulan yang ditawarkan dari YOLO itu sendiri, diantaranya :

1. Dengan melakukan prediksi koordinat bounding box langsung dari gambar inputan, maka YOLO melakukan deteksi objek dengan paradigma deteksi objek sebagai suatu permasalahan regresi, tidak seperti metode pengklasifikasian pada algoritma R-CNN.
2. Dengan arsitektur sederhana dari YOLO, memungkinkan kecepatan proses dan deteksi yang cepat. Hal ini memungkinkan prediksi secara realtime.

2.3.5 Keterbatasan dari YOLO

Selain melihat keunggulan dari YOLO, perlu juga diketahui batasan dari pendekslsian objek menggunakan algoritma YOLO, beberapa kelemahan dari YOLO yaitu :

1. Performa YOLO rendah terhadap objek dengan ukuran kecil, karena setiap sel grid memprediksi hanya sebanyak $B=2$ bounding Box untuk setiap kelas yang sama. Hal ini membuat batasan spasial terhadap jumlah objek yang dapat diprediksi pada setiap sel grid.
2. YOLO gagal dalam melkaukan deteksi objek untuk bentuk yang baru atau tidak biasa, karena YOLO tidak melakukan generalisasi terhadap apa yang di luar bounding box pada dataset inputan.
3. Dalam kalkulasi *loss function* dilakukan prediksi akar kuadrat dari tinggi dan lebar bounding box. Hal ini membuat error yang lebih besar dalam hal kalkulasi *localization loss*, karena akan meremedi permasalahan yang ada.

2.3.6 YOLOv2

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 pendekslsian 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[14]. Perubahan mendasar yang dilakukan dari YOLOv1 ke YOLOv2 yaitu :

1. *Batch Normalization*, yang mengurangi perubahan nilai pada hidden layer yang meningkatkan stabilitas dari jaringan saraf. Hal ini dibuktikan dengan peningkatan mAP sebesar 22%.
2. Resolusi dalam melakukan klasifikasi yang lebih tinggi, dari 224*224 menjadi 448*448. Untuk meningkatkan kemampuan mendeteksi objek yang lebih kecil.
3. *Anchor Boxes*. Pada YOLOv2, klasifikasi dan prediksi dilakukan dalam satu framework. Anchor boxes bertanggung jawab dalam melakukan prediksi bounding box. Anchor box ini ditandai untuk setiap dataset dengan menggunakan *k-means clustering*.
4. Fitur *Fine Grained*. Salah satu permasalahan utama pada YOLOv1 yaitu kemampuan dalam mendeteksi objek yang lebih kecil. Hal ini dapat diatasi pada YOLOv2 dengan meningkatkan pembagian gambar input menjadi 13*13 sel grid.
5. Multi *scale Training*. Pada YOLOv1, perbedaan ukuran input menjadi keterbatasan dalam melakukan deteksi objek yang sama pada ukuran gambar yang lebih besar. Hal ini diatasi dengan melakukan pelatihan terhadap ukuran gambar secara acak mulai dari 320*320 hingga 608*608 pada YOLOv2. Hal ini memungkinkan jaringan saraf memprediksi objek dengan ukuran dimensi input yang berbeda dengan akurasi yang tinggi.
6. Darknet-19 : YOLOv2 menggunakan arsitektur Darknet 19 dengan 19 *convolutional layers* dan 5 max pooling layers, diikuti softmax layer untuk klasifikasi objek di akhir. Darknet merupakan framework jaringan saraf yang ditulis dalam bahasa C dan CUDA. Hal ini membuatnya sangat cepat dalam melakukan deteksi objek yang esensial dalam melakukan prediksi secara real time.

2.3.7 YOLOv3

Dari versi YOLOv2, penulis awal YOLO mencoba melakukan peningkatan incremental yang kemudian menghasilkan YOLOv3[15]. Peningkatan incremental yang dimaksud yaitu :

1. Prediksi Bounding Box : Dalam YOLOv3, digunakan logistic regression untuk melakukan prediksi tingkat keberadaan objek.
2. *Class Prediction* : Pada YOLOv3, digunakan logistic regression dalam melakukan prediksi kelas dibandingkan versi sebelumnya yang menggunakan softmax. Dengan metode ini YOLOv3 mampu melakukan klasifikasi untuk multi-label.
3. FPN (*Feature Pyramid Networks*) : YOLOv3 melakukan prediksi dengan metode yang mirip dengan FPN, dimana 3 prediksi dilakukan pada setiap lokasi di gambar dan fitur-fitur diekstrak dari setiap hasil prediksi. Metode ini meningkatkan kemampuan deteksi pada setiap ukuran dimensi.
4. Darknet-53 : pada YOLOv2, digunakan Darknet-19 sebagai *feature extractor*. Pada YOLOv3 digunakan Darknet-53[16] untuk melakukan *feature extractor* dengan 53 *convolutional layers*. Selain itu, Darknet-53 banyak terdiri atas filter 3x3 dan 1x1 seperti yang dapat dilihat pada gambar dibawah.

Type	Filters	Size	Output
Convolutional	32	3×3	256×256
Convolutional	64	$3 \times 3 / 2$	128×128
1x	32	1×1	
Convolutional	64	3×3	
Residual			128×128
Convolutional	128	$3 \times 3 / 2$	64×64
2x	64	1×1	
Convolutional	128	3×3	
Residual			64×64
Convolutional	256	$3 \times 3 / 2$	32×32
8x	128	1×1	
Convolutional	256	3×3	
Residual			32×32
Convolutional	512	$3 \times 3 / 2$	16×16
8x	256	1×1	
Convolutional	512	3×3	
Residual			16×16
Convolutional	1024	$3 \times 3 / 2$	8×8
4x	512	1×1	
Convolutional	1024	3×3	
Residual			8×8
Avgpool		Global	
Connected		1000	
Softmax			

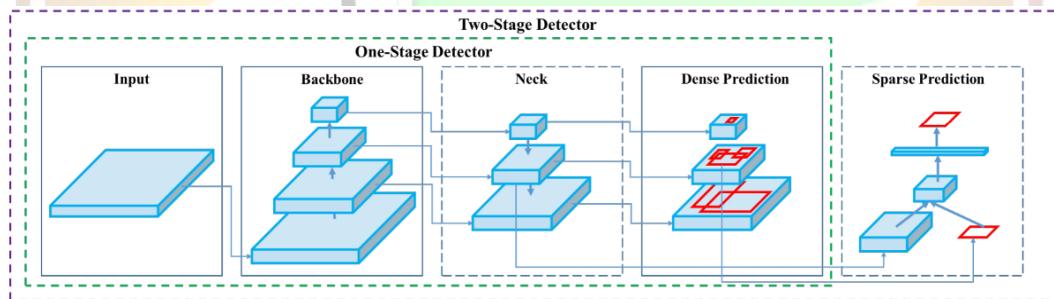
Gambar 2. 6 Aristektur Darknet pada YOLOv3[15]

2.3.9 YOLOv4

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 precision) ketika dites menggunakan dataset MS COCO. Dengan 65 FPS menggunakan Tesla v100. Perubahan utama dari YOLOv4 yaitu 2 keunggulan utama :

1. *Bag of Freebie*, peningkatan spesifik dalam proses training yang tidak memiliki dampak terhadap waktu inferensi, hal ini meningkatkan akurasi.
2. *Bag of Specials*, peningkatan jaringan saraf yang berdampak pada peningkatan waktu inferensi dengan performa yang bagus. Peningkatan ini meningkatkan kemampuan proses dari post processing non-max suppression.

Arsitektur yang digunakan pada YOLOv4 yaitu :



Gambar 2.7 Arsitektur YOLOv4[17]

Ada 4 blok induk dari arsitektur YOLOv4 :

1. Backbone (Dense Block & DenseNet, CSP, CSPDarknet53)
2. Neck (FPN, SPP)
3. Head (Dense Prediction)

Pada YOLOv4, backbone yang digunakan bukan Darknet53, namun CSPDarknet53 (Cross Stage Partial Connections Darknet53). Dalam penelitian ini, digunakan penerapan model YOLOv4 untuk melakukan *training* dan deteksi objek senjata. Untuk implementasinya, digunakan model dari YOLOv4 yang sudah dilatih oleh penulis jurnal YOLOv4 untuk kemudian dilakukan pelatihan dengan model yang

sudah ada tersebut, sehingga proses pelatihan lebih cepat dan implementasinya menjadi jauh lebih sederhana. Model YOLOv4 dari penulis sekaligus pencetus YOLOv4 dapat dilihat pada akun github penulis pada alamat url :

<https://github.com/AlexeyAB/Darknet> [18].

2.4 Pengujian Kualitas Model dengan *Medium Average Precisions* (mAP)

Banyak algoritma deteksi objek seperti Faster R-CNN, MobileNet SSD, dan YOLO, menggunakan mAP untuk mengevaluasi model yang dipublis. mAP merupakan salah satu metrik untuk menentukan kualitas dari model Machine learning[19]. 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 :

$$\text{Precisions} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}}$$

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

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

Tabel 2.1 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 :

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

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

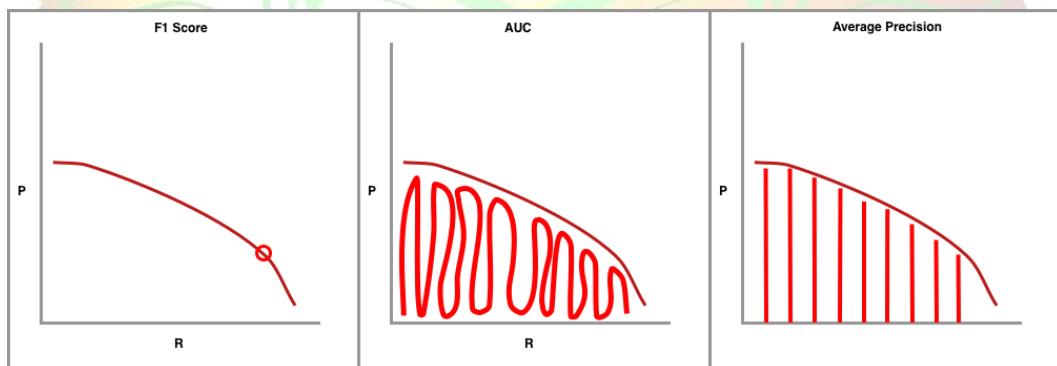
		Predicted	
		Negative	Positive
Actual	Negative	True Negative	False Positive
	Positive	False Negative	True Positive

Tabel 2.2 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 :

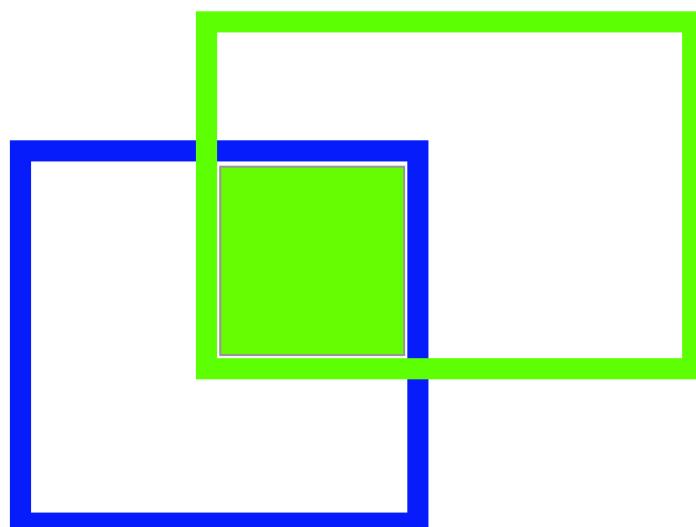
$$F1 = 2x \frac{Precision \times Recall}{Precisions + Recall}$$

Dari 3 metrik diatas, untuk menghitung Average Precisions (AP) digunakan kurva dari precision-recall (kurva F1) untuk menghitung area di bawah kurva tersebut[20].



Gambar 2. 8 Kurva metrik precisions-recall[20]

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 2.9 Sketsa proses penentuan bounding boxes dengan metrik IoU[20]

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

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

Dimana 1/11 didapatkan dari pembagian recall menjadi 11 bagian dari 0-1. $P_{interp(r)}$ merupakan perhitungan nilai interpolasi dari kurva precision-recall, dengan persamaan $P_{interp(r)} = \max p(r_i)$. Perhitungan AP dilakukan untuk masing-masing kelas objek[21]. Setelah didapatkan AP untuk masing-masing kelas objek, dilakukan perhitungan AP rata-rata atau maP dengan persamaan :

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

Dimana AP_i merupakan skor AP (Average Precisions) untuk masing-masing masing-masing kelas objek (N)[21].

2.5 Dataset

Untuk melakukan pelatihan model *machine learning*, diperlukan adanya data-data inputan yang akurat. Semakin banyak data input yang digunakan untuk pelatihan model, maka semakin tinggi probabilitas model untuk mengenali objek, serta semakin tinggi akurasi dalam melakukan pendektsian serta menentukan area objek pada gambar. Untuk menghasilkan model yang akurat, diperlukan kualitas dataset yang tinggi. Oleh karena itu perlu dilakukan preparasi dataset atau suatu prosedur untuk memilih dataset yang akan diinputkan sebagai data latih. Pada penelitian ini, dataset yang digunakan sebagai data latih yaitu gambar yang telah ditandai area objeknya dan dilakukan pelabelan.

Dalam penggunaan dataset, mengetahui objek apa yang akan diprediksi akan membantu dalam memilih kualitas data yang akan dijadikan dataset. Selain itu, beberapa pertimbangan juga diperlukan dalam memilih dataset[22] :

1. Sumber dataset dan mekanisme pengumpulan data
2. Kategori *machine learning* yang digunakan
3. Format dataset

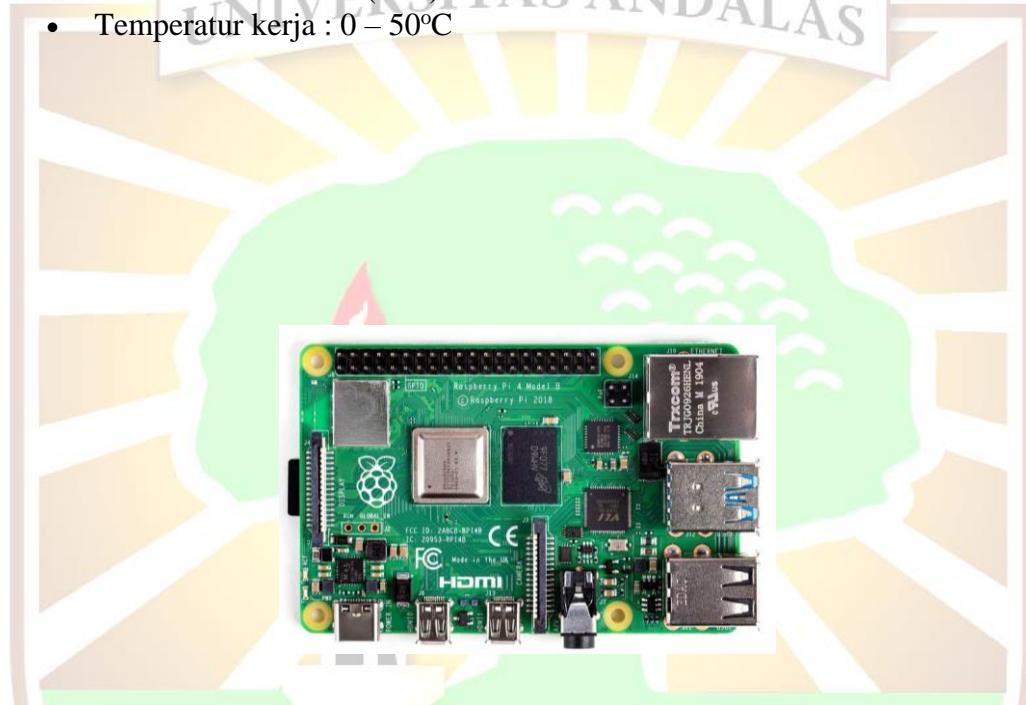
2.6 Raspberry Pi

Raspberry Pi merupakan seri *Single Board Computer* (SBC) yang dikembangkan di Inggris oleh Raspberry Pi Foundation. Di awal pengembangannya, Raspberry Pi ditujukan untuk mempromosikan pengenalan ilmu komputer di sekolah dan negara-negara berkembang. Namun proyek tersebut berkembang dengan pesat dan menjadi populer untuk beragam kebutuhan seperti robotic dan automasi. 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 Bluetoooh 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[23].

Dalam penelitian ini, Raspberry Pi yang digunakan adalah Raspberry Pi 4 Model B+ dengan spesifikasi sebagai berikut :

- Broadcom BCM2711, Quad core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz
- 4GB LPDDR4-3200 SDRAM
- 2.4 GHz and 5.0 GHz IEEE 802.11ac wireless, Bluetooth 5.0, BLE
- Gigabit Ethernet
- 2 USB 3.0 ports; 2 USB 2.0 ports.
- Raspberry Pi standard 40 pin GPIO header
- 2 × micro-HDMI ports

- 2-lane MIPI DSI display port
- 2-lane MIPI CSI camera port
- 4-pole stereo audio and composite video port
- H.265 (4kp60 decode), H264 (1080p60 decode, 1080p30 encode)
- OpenGL ES 3.0 graphics
- Micro-SD card slot
- 5V DC via USB-C connector (minimum 3A*)
- 5V DC via GPIO header (minimum 3A*)
- Power over Ethernet (PoE)
- Temperatur kerja : 0 – 50°C



Gambar 2.10 Raspberry Pi 4[19]

2.7 Webcam

Kamera merupakan alat yang menangkap data gambar. Dalam penelitian dan penerapan di bidang *Computer Vision*, Kamera berfungsi sebagai inputan citra yang diproses dan meneliti fitur yang terdapat di dalam tangkapan gambar. Ada banyak jenis kamera yang digunakan untuk penangkapan gambar, seperti kamera analog, kamera digital, dan webcam.

Webcam memungkinkan pengguna komputer merekam frame gambar dan video secara realtime untuk diproses oleh komputer. Nama webcam sendiri mengacu pada perangkat kamera yang dihubungkan ke komputer atau laptop[24].

2.8 Buzzer

Buzzer merupakan komponen elektronika yang berfungsi untuk mengubah getaran listrik menjadi getaran suara. *Buzzer* terdiri dari kumparan yang terpasang pada diafragma dan kemudian kumparan tersebut dialiri arus sehingga menjadi elektromagnet, kumparan tadi akan tertarik ke dalam atau keluar, tergantung dari arah arus dan polaritas magnetnya, karena kumparan dipasang pada diafragma maka setiap gerakan kumparan akan menggerakkan diafragma secara bolak-balik sehingga membuat udara bergetar yang akan menghasilkan suara. *Buzzer* biasa digunakan sebagai indikator pada suatu proses. Modul *Buzzer* memiliki tegangan kerja DC antara 4V – 9V.

Beberapa Aplikasi dari *buzzer* diantaranya sebagai sirkuit alarm, dimana user diberikan peringatan mengenai sesuatu. Selanjutnya yaitu pada alat komunikasi, lalu pada perangkat elektronik automobile, dan pada proyek-proyek sistem tertanam untuk memberikan output berupa notifikasi alarm[25].

Gambar 2.11 Modul Buzzer

BAB III

RANCANGAN SISTEM

3.1 Analisa Kebutuhan Sistem

Dalam analisis kebutuhan sistem, terdapat 4 aspek kebutuhan yang menjadi kesatuan inti dari penelitian ini. Kebutuhan tersebut yaitu kebutuhan fungsional, kebutuhan non-fungsional, kebutuhan perangkat keras dan kebutuhan perangkat lunak. Kebutuhan fungsional berisi tentang fungsi-fungsi dan struktur yang secara langsung terlibat dalam berjalannya suatu sistem. Kebutuhan non-fungsional memuat fungsi-fungsi pendukung untuk membuat sistem berfungsi dengan baik sesuai tujuan perancangannya. Kebutuhan perangkat keras berisi komponen-komponen dan peralatan fisik yang dibutuhkan sebagai *platform* berjalannya suatu sistem. Kebutuhan perangkat lunak berisi tentang perangkat lunak dan aplikasi yang dibutuhkan agar komponen pada perangkat keras dapat berfungsi sesuai tujuan yang hendak dicapai.

3.1.1 Kebutuhan Fungsional Sistem

Kebutuhan fungsional menggambarkan interaksi antara sistem dengan pengguna. Adapun kebutuhan fungsional yang akan diperlukan pada penelitian ini yaitu :

1. Sistem mampu melakukan pendekripsi 2 kategori senjata, senjata api dan senjata tajam.
2. Sistem mampu melakukan klasifikasi kelas objek dari hasil deteksi senjata
3. Sistem mampu mendekripsi beberapa jumlah senjata dalam satu frame
4. Sistem mampu memberikan notifikasi berupa data jenis dan jumlah senjata ke aplikasi Android
5. Aplikasi Android mampu menerima data notifikasi yang dikirim dari Raspberry Pi

3.1.1 Kebutuhan Non-Fungsional Sistem

Kebutuhan non-fungsional dalam penelitian ini yaitu kinerja atau performa sistem. Parameter kinerja yang diambil adalah akurasi. Kebutuhan Kedua yaitu Tingkat keakuratan deteksi objek senjata yang ditentukan oleh kualitas dataset yang

diberikan, sehingga menjadi faktor penting dalam mendukung keutuhan jalannya sistem. Kebutuhan non-fungsional ketiga yaitu respon waktu yang mendekati *realtime*. Aspek *realtime* dari sistem yaitu *soft realtime*, dimana sistem memiliki batas toleransi waktu dalam melakukan pemrosesan. Jika waktu proses dari sistem melebihi batas toleransi, maka akan terjadi degradasi dari sistem yang memperlambat aliran proses berikutnya. Dalam hal ini, sistem tetap berjalan dan melakukan fungsinya, namun degradasi sistem akan menimbulkan keterlambatan aliran output sistem yang membuat delay output semakin tinggi. Pada sistem penelitian ini, kegagalan mencapai batas toleransi waktu tidak akan membuat sistem gagal beroperasi sepenuhnya dalam melakukan deteksi objek senjata, namun delay output akan semakin besar dan membuat waktu deteksi tidak sesuai dengan waktu sebenarnya. Karena keterbatasan komputasi dari Raspberry Pi dalam menjalankan model YOLO, maka dalam penelitian ini dilakukan pengujian batasan waktu yang dapat ditoleransi sistem sebelum output yang diterima benar-benar tidak relevan lagi.

3.1.2 Kebutuhan Perangkat Keras

Kebutuhan perangkat keras meliputi peralatan yang dibutuhkan sistem untuk menjalankan fungsinya. Perangkat keras yang digunakan pada penelitian ini adalah sebagai berikut :

1. Raspberry Pi sebagai media untuk implementasi model YOLO yang sudah dilatih. Raspberry Pi yang digunakan dalam penelitian ini yaitu Raspberry Pi 4 dengan Spesifikasi :
 - a. Processor ARM Cortex-A72 (ARM8) 64-Bit SoC 1.5GHz
 - b. RAM 4 GB LPDDR4
 - c. 2.4 GHz and 5.0 GHz IEEE 802.11ac wireless, Bluetooth 5.0, BLE
 - d. Gigabit Ethernet
2. Kamera sebagai sensor input untuk menangkap data frame gambar. frame menjadi inputan model YOLO untuk mendeteksi ada tidaknya objek senjata. Kamera yang digunakan untuk penelitian ini adalah Kamera Webcam dengan resolusi 5MP

3. buzzer digunakan untuk output notifikasi dan peringatan berupa Alarm. Modul buzzer yang digunakan yaitu modul buzzer Aktif 5V yang mampu menghasilkan nada frekuensi hingga 2500Hz

3.1.3 Kebutuhan Perangkat Lunak

Perangkat Lunak yang digunakan dalam perancangan sistem yaitu modul-modul dan aplikasi yang diperlukan agar sistem dapat berjalan seutuhnya. Modul-modul dan aplikasi yang digunakan meliputi :

1. Modul pelatihan model *YOLO* untuk mendeteksi objek senjata
2. Modul implementasi sistem pendekripsi objek senjata pada Raspberry Pi 4
3. Aplikasi android untuk menerima notifikasi dan data dari Raspberry Pi.

3.1.4 Kebutuhan Data

Data yang dibutuhkan dalam perancangan sistem ini yaitu kumpulan dataset gambar senjata dari 2 kategori senjata, yaitu senjata api laras pendek (*handgun*) dan senjata tajam. Dari kumpulan dataset yang ada, dibutuhkan 3 kategori data :

1. Data Latih (*Training Set*)

Untuk kebutuhan data latih, digunakan 2 kelas objek, senjata tajam dan senjata api laras pendek (*handgun*). Data yang digunakan untuk data latih yaitu sebesar 80 % dari kumpulan dataset. Banyaknya jumlah data latih awal yaitu 500 gambar yang telah dianotasi untuk 2 kelas objek. Data anotasi yang digunakan yaitu format .txt *Darknet* untuk mendukung pelatihan dengan *YOLO*.

2. Data Validasi (*Validation Set*)

Data validasi menggunakan 10% dari total kumpulan dataset yang ada, dengan format dan tipe data yang sama dengan data latih, yaitu gambar yang telah dianotasi untuk 2 kelas objek senjata. Data validasi berfungsi untuk melakukan kalkulasi error dan mencegah *overfitting* pada model *YOLO*

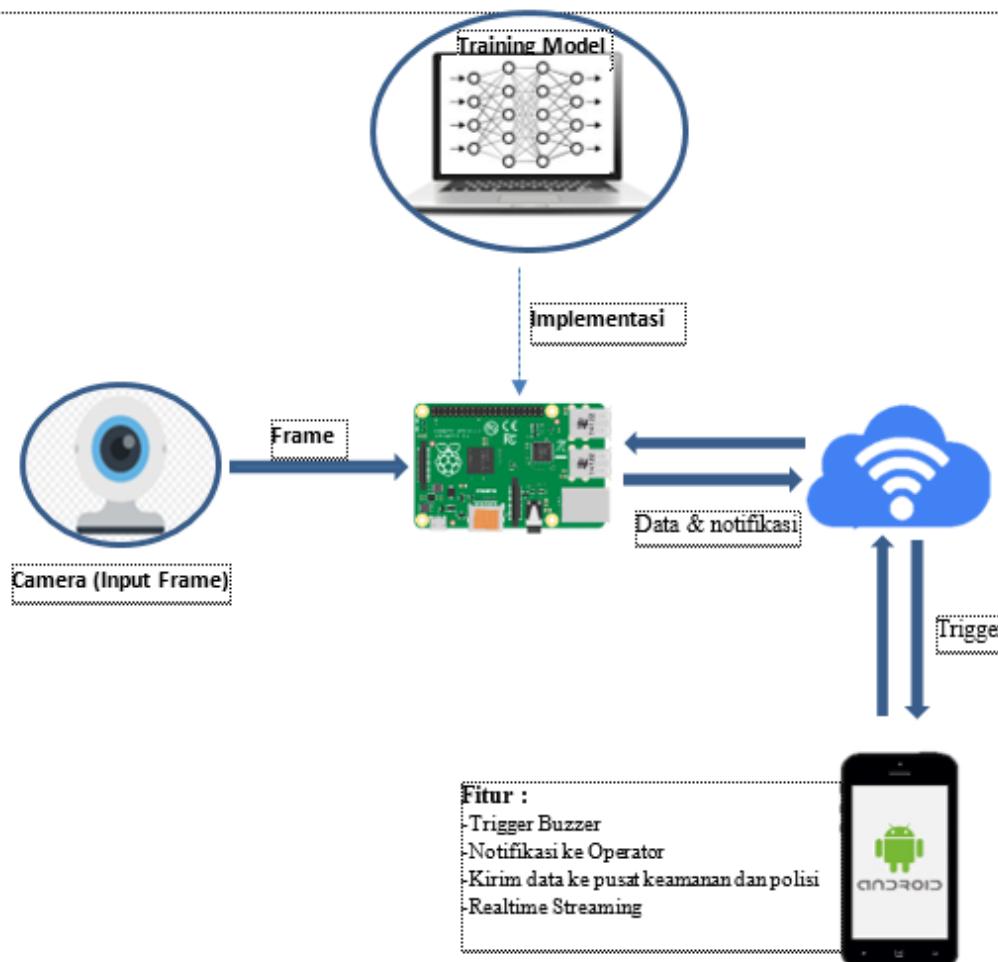
3. Data Uji (*Testing Set*)

Untuk data uji digunakan sebanyak 10% dari total kumpulan dataset. Jenis data yang digunakan yaitu kumpulan gambar yang mengandung 2 kelas objek senjata, dan kumpulan gambar yang tidak mengandung objek senjata.

Data ini digunakan untuk menguji keakuratan dan ketepatan model dalam melakukan deteksi objek senjata.

3.2 Rancangan Umum Sistem

Dalam perancangan sistem, diperlukan adanya integrasi antara data, perangkat lunak, dan perangkat keras yang digunakan. Bagan kerja dari sistem dapat dilihat pada gambar 3.1 berikut ini :



Gambar 3. 1 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 pendekslsian senjata diterapkan pada Rapberry 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.

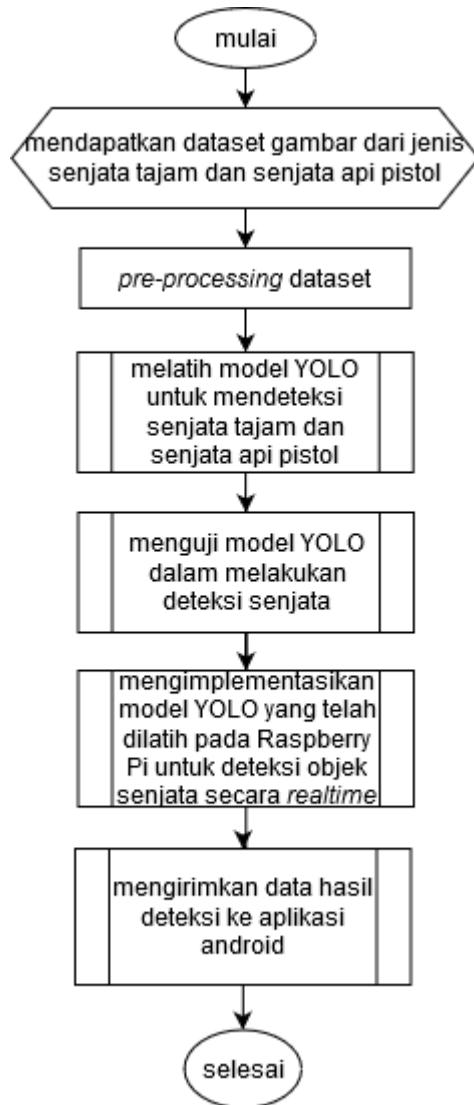
3.3 Rancangan Proses

Secara garis besar, terdapat 3 proses yang menjadi kunci utama untuk keutuhan jalannya sistem pada penelitian ini. Proses-proses tersebut adalah :

1. Pelatihan model YOLO untuk mengenali objek senjata dan pengujian model YOLO dengan menghitung tingkat akurasi dan performa model.
2. Implementasi model YOLO yang telah dilatih pada Raspberry Pi 4.
3. Aplikasi android untuk menerima notifikasi dan data yang dikirim dari Raspberry Pi 4

Untuk lebih jelasnya, dapat dilihat pada Gambar 3.2 mengenai alur proses secara keseluruhan.





Gambar 3.2 Alur Proses secara keseluruhan

Tahapan awal dari penelitian ini adalah mendapatkan dataset untuk melatih model YOLO dalam melakukan deteksi senjata. Model YOLO dilatih untuk mengenali 2 kategori senjata, senjata tajam dan senjata api pistol. Untuk melakukan pelatihan model YOLO, diperlukan dataset berupa gambar dari 2 kategori senjata diatas yang telah dilabeli atau dianotasi untuk dapat dijadikan input pelatihan model YOLO. Dalam penelitian ini, dataset gambar diperoleh dari *Open Image Dataset V6* (OIDv6) milik Google yang memuat dataset gambar dari berbagai kategori objek yang telah dilabeli. Dari OIDv6 diambil 600 gambar dari jenis senjata tajam dan 600 gambar dari jenis senjata pistol. Setelah itu dilakukan *preprocessing* dataset dengan mengubah format pelabelan gambar dari format standar OIDv6 menjadi

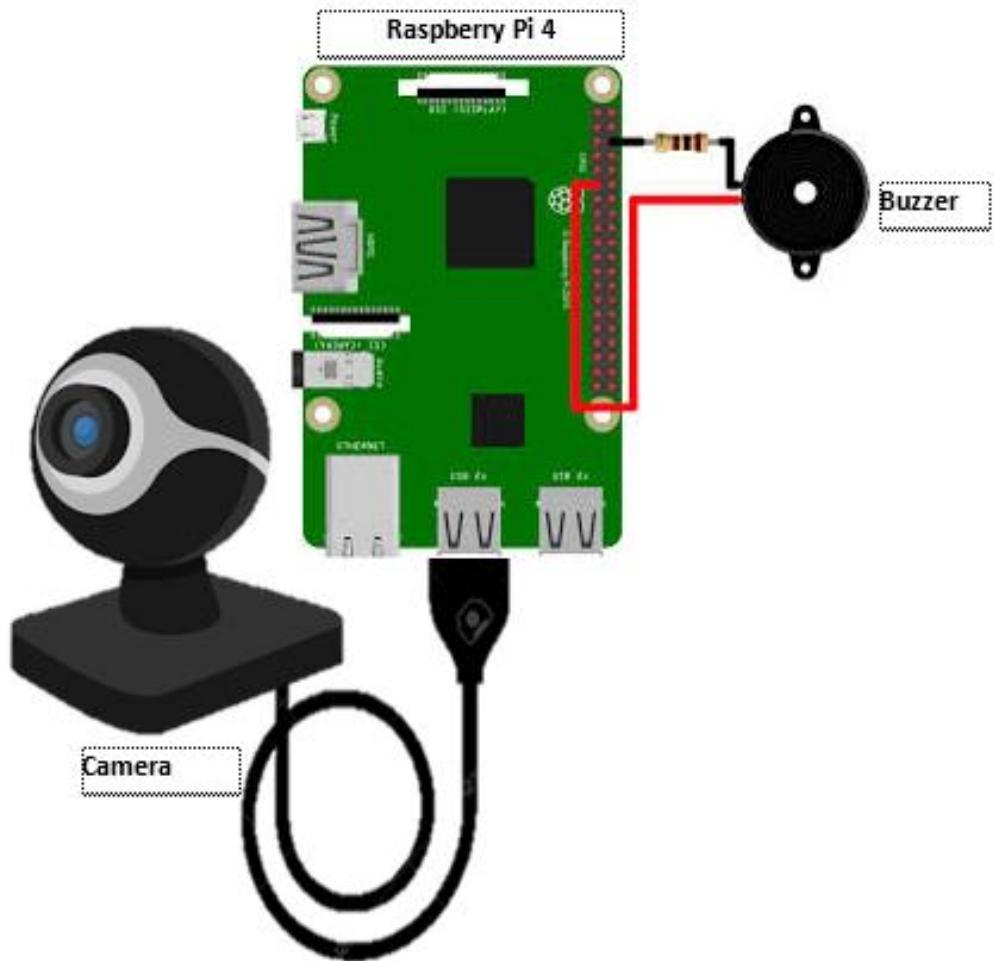
format *.xml* yang dapat dibaca oleh model YOLO. Setelah dataset disiapkan, selanjutnya dilakukan pelatihan model YOLO. Tahapan proses dari pelatihan model dapat dilihat pada bagian perancangan perangkat lunak.

Setelah model YOLO dilatih untuk mendeteksi objek senjata, selanjutnya dilakukan implementasi model pada Raspberry Pi untuk pengujian secara *realtime* melalui input *live* kamera webcam. Ketika sistem mendeteksi adanya objek senjata pada frame inputan kamera, maka dikalkulasikan tingkat ancaman yang mungkin ditimbulkan dari senjata yang terdeteksi dan dikirim notifikasi peringatan ke aplikasi android. Klasifikasi ancaman yang digunakan yaitu normal, sedang, dan kritis. Parameter yang diambil dalam melakukan klasifikasi ancaman yaitu berdasarkan kemungkinan dampak yang ditimbulkan dari objek senjata.

3.3.1 Rancangan Perangkat Keras

Penerapan dari sistem pendekripsi senjata dilakukan pada mini komputer Raspberry Pi 4, dengan menggunakan kamera sebagai input gambar untuk mendeteksi adanya senjata secara *realtime*. Rancangan perangkat keras yang digunakan pada sistem dapat dilihat pada gambar 3.3 dibawah.





Gambar 3.3 Perancangan Perangkat Keras

Berdasarkan gambar 3.4, komponen yang digunakan yaitu :

1. Kamera berfungsi sebagai sensor input untuk menangkap frame yang akan diolah pada Raspberry Pi 4.
2. Raspberry Pi 4 berfungsi sebagai pengolah data dan otak dari sistem, menerima frame input untuk mendeteksi adanya senjata, lalu mengirimkan data ke aplikasi android.
3. Buzzer berfungsi untuk sistem alarm pada sistem, dengan tujuan memberikan peringatan kepada tersangka yang mencoba melakukan tindak kekerasan bersenjata dan memberikan rasa awas terhadap lingkungan sekitar lokasi kejadian.

3.3.2 Rancangan Perangkat Lunak

Dalam perancangan perangkat lunak (*Software*), terdapat 3 modul utama yang digunakan. Modul pertama yaitu modul pelatihan untuk melatih dan menguji model agar dapat mendeteksi beberapa kategori senjata. Selanjutnya yaitu modul implementasi sistem pada raspberry pi untuk menerima inputan langsung dari kamera dan menggunakan data hasil latihan untuk mendeteksi ada atau tidaknya senjata pada frame. Modul selanjutnya yaitu aplikasi android untuk menerima data dan mengirimkan trigger ke Raspberry Pi 4.

3.3.2.1 Modul Pelatihan (*Training*) dan Pengujian Model

Pada program *training*, dataset yang berupa kumpulan gambar senjata api dan senjata tajam akan diproses untuk melatih model dalam mengenali objek senjata. Input yang digunakan yaitu dataset *training* yang telah di anotasi dalam format .txt YOLO Darknet. Susunan Dataset untuk inputan program *training* dapat dilihat pada Tabel 3.1.

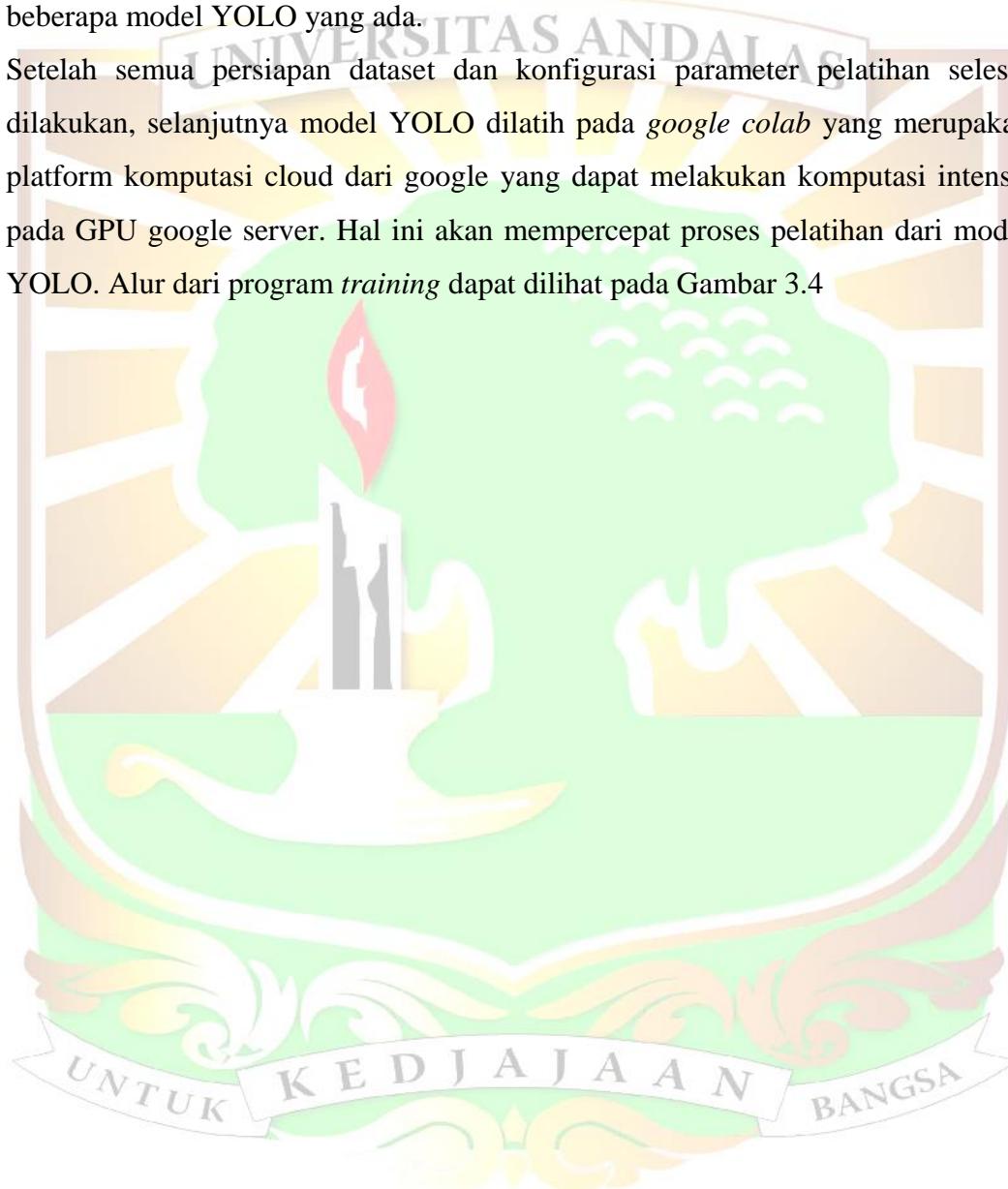
Tabel 3.1 Susunan dan Kategori Dataset

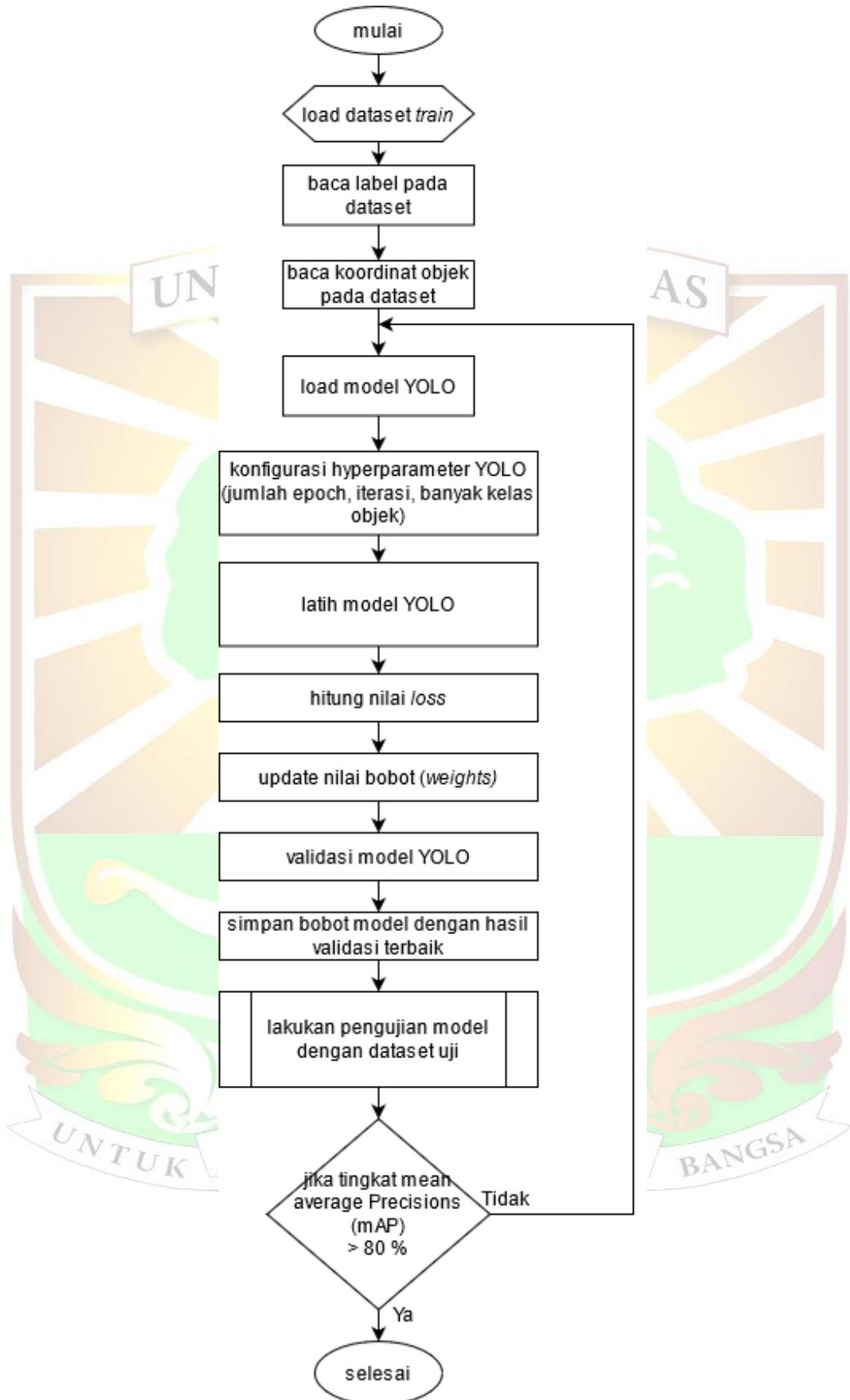
Kategori	Isi
Dataset	Berisi kumpulan gambar senjata api dan senjata tajam yang telah dianotasi. Format file yang digunakan yaitu .JPG. Kumpulan data gambar pada dataset dibagi menjadi 3 bagian : <i>training</i> validasi, dan uji.
Anotasi	Merupakan file dengan tipe .xml yang dihasilkan dari tool anotasi. pada file tersebut terdapat parameter lokasi atau koordinat objek yang ditandai pada gambar.

Program *training* menggunakan bahasa pemrograman python dengan framework Darknet YOLO, dimana YOLO yang digunakan yaitu YOLOv4 tiny. YOLOv4 tiny

dipilih karena kecepatan deteksinya dan akurasi yang mencukupi. Hal ini karena disebabkan oleh keterbatasan komputasi pada Raspberry Pi 4 yang merupakan platform penerapan sistem untuk pengujian. Untuk itu diterapkan YOLOv4 tiny agar dapat diterapkan pada Raspberry Pi 4 setelah sebelumnya mencoba melakukan pengkajian teori mengenai perbandingan kinerja dan kecepatan deteksi dari beberapa model YOLO yang ada.

Setelah semua persiapan dataset dan konfigurasi parameter pelatihan selesai dilakukan, selanjutnya model YOLO dilatih pada *google colab* yang merupakan platform komputasi cloud dari google yang dapat melakukan komputasi intensif pada GPU google server. Hal ini akan mempercepat proses pelatihan dari model YOLO. Alur dari program *training* dapat dilihat pada Gambar 3.4

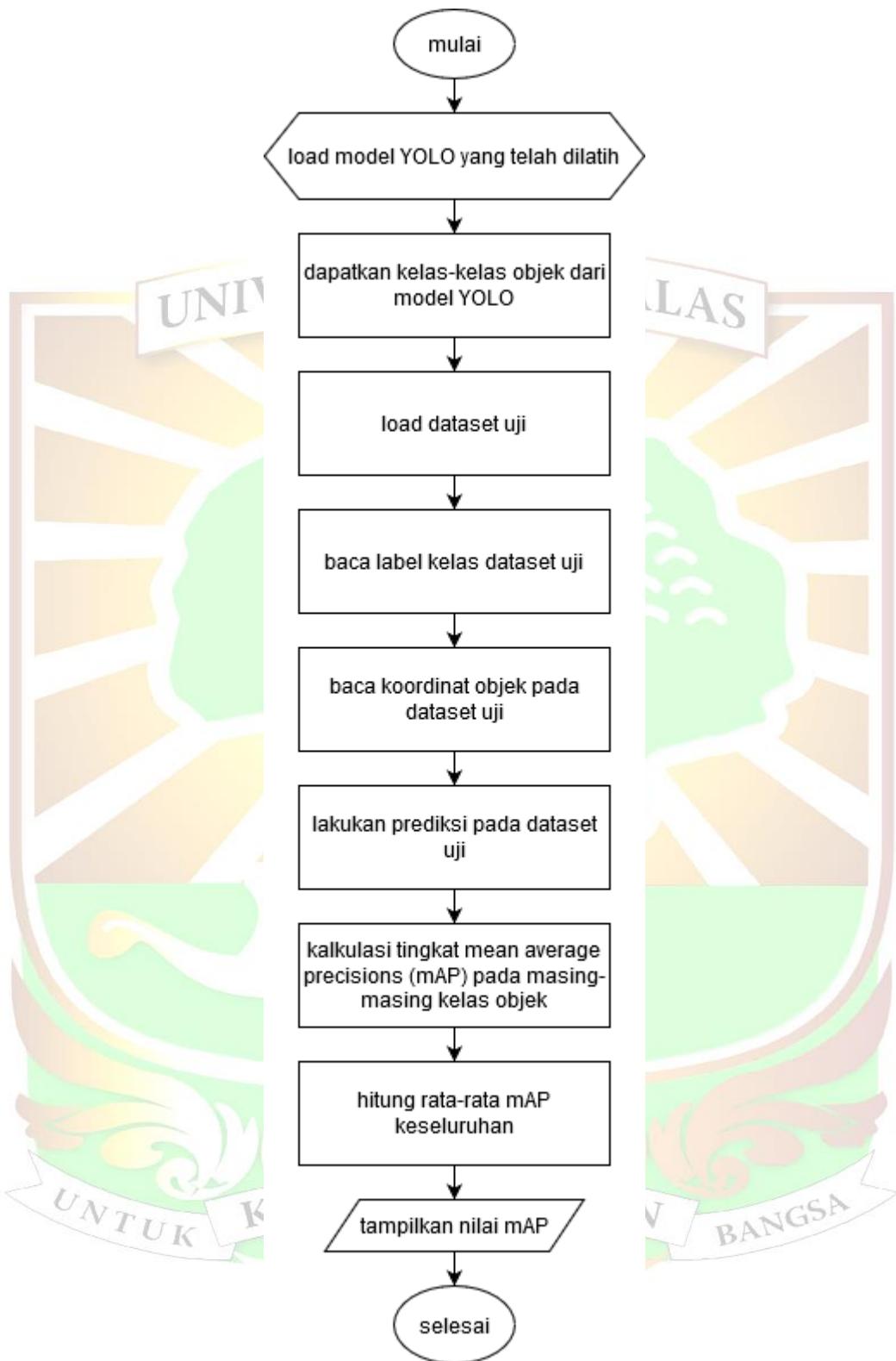




Gambar 3.4 Alur *flowchart* program *Training*

Setelah model YOLOv4 tiny dilatih untuk mengenali objek senjata tajam dan senjata api pistol, selanjutnya dilakukan pengujian tingkat akurasi dan performa dari model yang dilatih. Bagian ini menjelaskan tentang program pengujian kualitas model *YOLO*. Pengujian dilakukan dengan menggunakan dataset gambar uji dan melakukan penilaian tingkat akurasi dan ketepatan deteksi objek. Jika tingkat pengujian menghasilkan kualitas model yang rendah, maka dilakukan kembali *training* dengan penambahan dataset, hal ini dilakukan agar fitur yang dideteksi dapat dikenali oleh model sehingga mampu meningkatkan akurasi deteksi. Alur dari program pengujian dapat dilihat pada Gambar 3.5





Gambar 3.5 Alur *flowchart* program pengujian model

3.3.2.2 Implementasi Sistem pada Raspberry Pi 4

Pada implementasi sistem, digunakan perangkat Raspberry Pi 4 model B+ dengan RAM 4GB, yang mana digunakan untuk mempermudah penerapan sistem dan pengujian sistem. Setelah model YOLOv4 tiny dilatih untuk mengenali objek senjata, selanjutnya model YOLO akan diterapkan pada Raspberry Pi 4 dengan inputan dari live webcam yang kemudian dilakukan deteksi objek senjata menggunakan YOLO pada setiap frame inputan. Ketika terdeteksi adanya senjata berupa pisau atau pistol, selanjutnya dilakukan kalkulasi kategori kemungkinan ancaman yang ditimbulkan dengan parameter dampaknya, yang mana dijabarkan sebagai berikut :

1. Status ancaman tingkat 0 (Normal)

Pada tingkat ancaman 0, sistem tidak mendeteksi adanya objek senjata atau dalam kata lain situasi normal dan terkendali. Pada aplikasi android, status ancaman tingkat 0 ditampilkan sebagai kondisi normal.

2. Status ancaman tingkat 1 (Sedang)

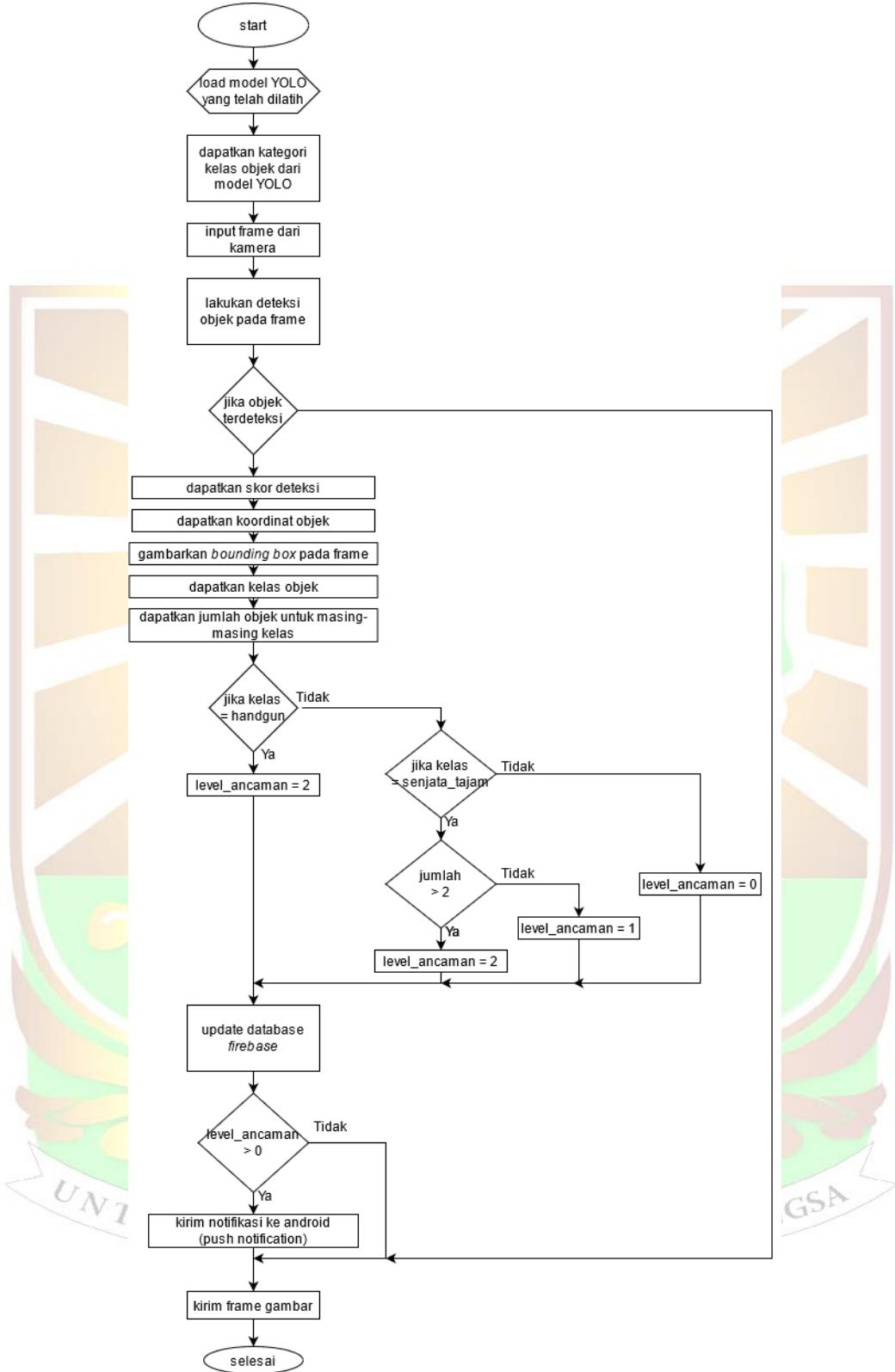
Pada tingkat ancaman 1, sistem mendeteksi adanya objek senjata pisau namun jumlahnya kecil dari 3. Disini sistem akan menampilkan status ancaman sedang pada aplikasi android. Hal ini mengacu pada parameter kemungkinan dampak yang ditimbulkan dari objek senjata, dimana 2 senjata pisau artinya adanya kemungkinan konflik bersenjata antara 2 orang atau 1 orang yang memegang 2 senjata pisau. Dalam hal ini, kemungkinan dampak yang ditimbulkan yaitu adanya korban antara kedua belah pihak yang terlibat, yang cukup meresahkan namun dampaknya dapat ditangani masyarakat secara bersama-sama dan aparat keamanan dapat mengambil tindakan minimum untuk meredam konflik.

3. Status ancaman tingkat 2 (Kritis)

Pada tingkat ancaman 2, sistem mendeteksi adanya 1 atau lebih senjata pistol, atau senjata pisau lebih dari 2, atau terdapat senjata pistol dan pisau yang terdeteksi bersamaan. Dalam hal ini, dilihat dari kemungkinan dampak yang ditimbulkan,

maka sistem akan menampilkan status ancaman kritis karena dampaknya yang luas dan berpotensi menimbulkan banyak korban. Jika dilihat dari kemungkinan dampak yang ditimbulkan, adanya 3 senjata pisau artinya adanya kemungkinan kasus penyerangan senjata berkelompok, dimana terdapat 3 orang yang memegang senjata tajam. Hal ini berpotensi menimbulkan korban jiwa yang luas dan adanya kemungkinan konflik penyerangan bersenjata secara berkelompok seperti tawuran, perampokan yang terorganisir, hingga terorisme. Hal ini berpotensi menimbulkan ketakutan dan ancaman serius pada masyarakat sekitar. Untuk itu penanganan yang cepat dan terukur dari aparat keamanan perlu dilakukan untuk mencegah timbulnya korban yang besar dan konflik yang meluas. Dalam kasus lain, seperti terdeteksi adanya satu senjata pistol atau lebih, maka situasi ancaman juga akan dianggap kritis atau tingkat 2 karena kemungkinan dampak yang ditimbulkan dari senjata api dan pistol sangatlah tinggi. Ancaman penyerangan dengan senjata api memerlukan tindakan penanganan yang hati-hati dari aparat keamanan. Hal ini melihat kemampuan dari senjata api itu sendiri, dimana 1 pucuk senjata pistol dengan peluru penuh akan memiliki kemungkinan ancaman dapat membunuh setidaknya 1 orang untuk 1 butir peluru, sehingga penanganan untuk penyerangan dengan senjata api memerlukan tindakan yang terukur dari aparat keamanan.

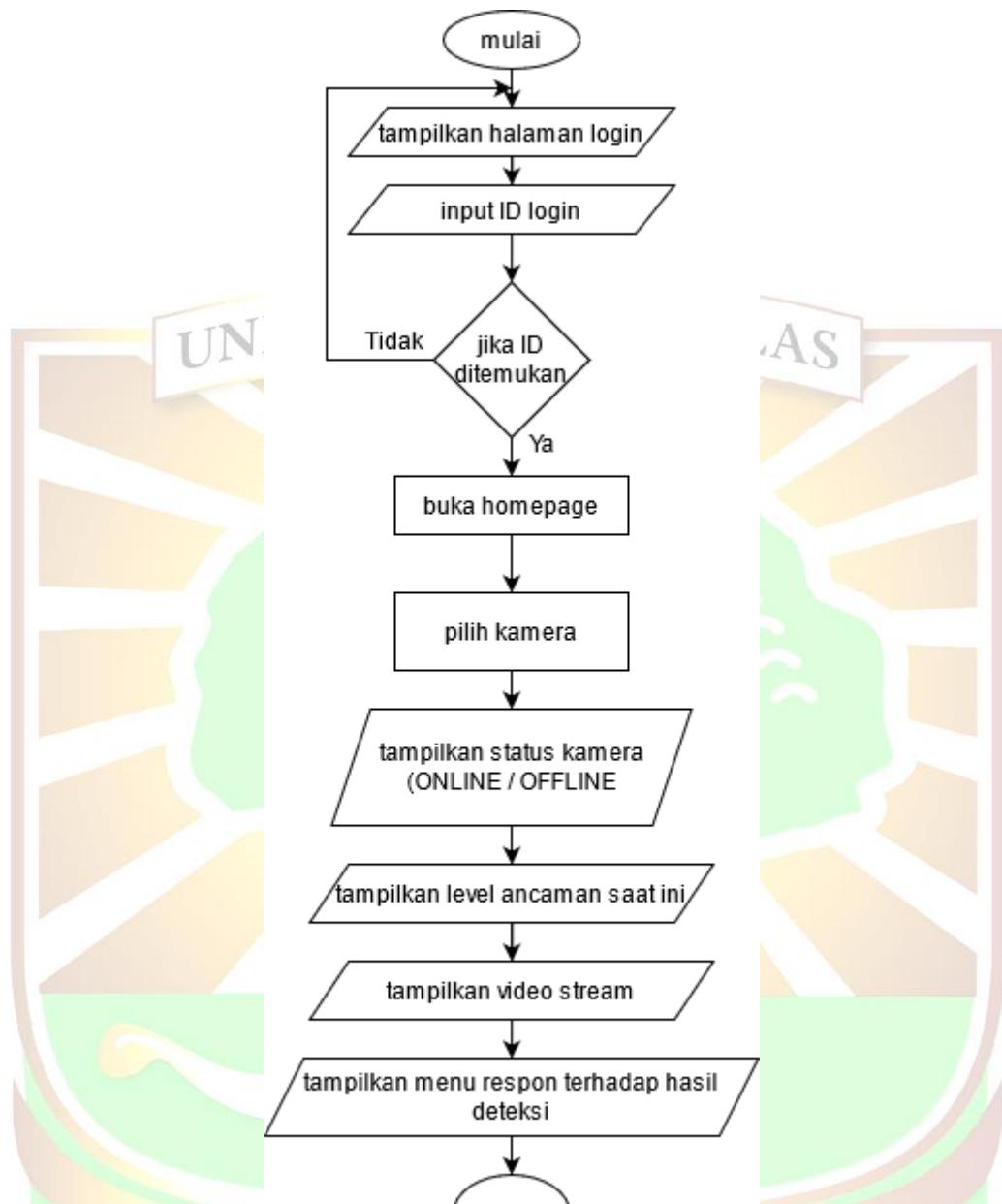
Setelah dilakukan deteksi pada frame gambar dari live webcam dan dilakukan perhitungan status ancaman dari kemungkinan dampak yang ditimbulkan, selanjutnya sistem pada Raspberry Pi 4 akan mengirim data-data hasil deteksi berupa tingkat, jenis senjata yang terdeteksi, dan jumlah senjata yang terdeteksi ke database firebase melalui jaringan internet. Selanjutnya, data-data dari database firebase diambil oleh aplikasi android untuk kemudian ditampilkan hasil dari deteksi sistem pada aplikasi android. Jika terdeteksi adanya senjata, maka akan dikirim notifikasi berupa tingkat ancaman dan jenis senjata yang terdeteksi melalui *push notification* yang akan diterima aplikasi android. Untuk lebih lengkapnya, dapat dilihat alur proses implementasi sistem pada gambar 3.6 dibawah ini.



Gambar 3.6 flowchart modul implementasi sistem pada raspberry Pi

3.3.2.3 Aplikasi android untuk menerima notifikasi dan data

Aplikasi android berfungsi sebagai sistem monitoring dari deteksi objek yang dilakukan pada Raspberry Pi. Saat terdeteksi adanya senjata oleh sistem, maka program pada Raspberry Pi akan mengirimkan update data pada database firebase. Kemudian dari firebase dikirimkan data notifikasi kemungkinan level ancaman menggunakan fitur push notification pada android. Aplikasi android akan mengambil data dari firebase ketika terkoneksi internet dan akan menerima notifikasi saat terdeteksi adanya objek senjata. Setelah diterima notifikasi, user akan diberikan informasi mengenai senjata yang terdeteksi, frame gambar yang memuat senjata, dan informasi lokasi dari kamera. Selanjutnya user akan merespon notifikasi dari android untuk menilai apakah ancaman penyerangan bersenjata tersebut nyata atau hanya kesalahan deteksi. User dapat merespon dalam 2 tindakan, false detection atau threat (ancaman). Jika user memilih ancaman tersebut nyata, maka akan ditampilkan menu untuk menghidupkan alarm pada kamera, menu untuk memanggil pihak berwajib, agar kemudian dapat dilakukan tindakan penanganan oleh berwenang. Perancangan Aplikasi android dilakukan menggunakan framework Kodular. Untuk alur proses dari aplikasi android dapat dilihat pada Gambar 3.8.

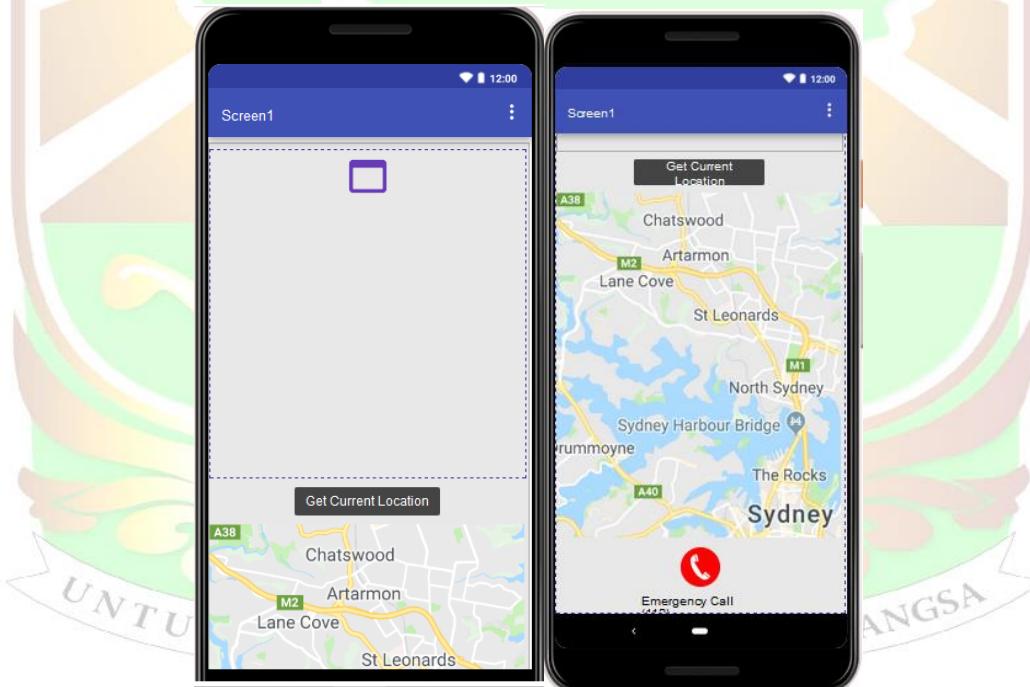


Gambar 3.7 Alur proses aplikasi android

Rancangan Antarmuka dari aplikasi android adalah sebagai berikut :



Gambar 3.8 Rancangan Antarmuka homepage



Gambar 3.9 Rancangan antarmuka video streaming

3.3.2.4 Rancangan Database Firebase

Dalam perancangan aplikasi android, diperlukan database untuk menyimpan data pada sistem dan menghubungkan antara aplikasi android dengan sistem

pendeksi senjata pada Raspberry Pi. Untuk itu diperlukan adanya database yang dapat diakses secara online dan diupdate secara realtime. Untuk kemudahan integrasi aplikasi dan karena kemudahan akses layanan, maka database Firebase dari google digunakan pada rancangan aplikasi. Selain integrasi dengan framework kodular yang mudah, database Firebase juga terdesia secara gratis dan menyediakan beberapa ruang untuk penyimpanan file. Firebase juga menawarkan kemudahan integrasi dengan bahasa pemrograman python, yang memudahkan melakukan update database dan pengambilan data pada sistem Raspberry Pi. Struktur database firebase yang digunakan yaitu :

1. **Tabel Camera** : menyimpan data informasi mengenai kamera dan status kamera

Daftar Tabel :

Id_cam : menyimpan id camera yang unik untuk setiap kamera
(PRIMARY KEY, int)

Name : menyimpan nama kamera untuk identifikasi

Posisi : menyimpan informasi letak kamera

active_caseID : menyimpan data ID kasus saat ini (string)

buzz_trigger : menyimpan status buzzer (ON / OFF, string)

coordinate : menyimpan titik koordinat dari kamera (string)

link_lokasi : menyimpan link alamat koordinat dalam bentuk alamat google maps.

detect : informasi status pendeksi senjata (0 / 1, int)

get_data_firebase : status trigger untuk perangkat raspberry Pi, untuk mengupdate ID kasus (string)

gun_detect : menyimpan jumlah senjata pistol yang terdeteksi (int)

knife_detect : menyimpan jumlah senjata pisau yang terdeteksi (int)

jml_senjata : menyimpan jumlah total senjata yang terdeteksi (int)

lvl_threat : menyimpan status level ancaman berdasarkan jumlah senjata yang terdeteksi

2. **Tabel User** : menyimpan data ID user dan data perangkat yang digunakan untuk login, dengan daftar tabel :

Id_user : menyimpan ID user berupa ID pegawai atau ID personal (PRIMARY KEY, string)

Imei : menyimpan informasi perangkat IMEI untuk pengenalan perangkat yang login (string)

Name : menyimpan nama dari pemilik ID (string)

3. **Tabel Recoded** : menyimpan data riwayat hasil deteksi senjata, yang disimpan dalam data alamat URL gambar hasil deteksi, dengan daftar tabel :

Id_cam : identifikasi id kamera (FOREIGN KEY, int)

caseID : menyimpan data ID kasus dalam format datetime (string)

urls : menyimpan alamat UTL untuk gambar hasil deteksi, yang disimpan pada firebase storage (string)

3.4 Rencana Pengujian

Rancangan pengujian berisi tindakan dan proses yang dilakukan untuk mendapatkan pembuktian dari tujuan sistem yang dibangun. Selain itu juga untuk melakukan pengujian tingkat kesuksesan dan keakuratan sistem secara keseluruhan.

3.4.1 Rencana Pengujian Model YOLO

Pengujian model YOLO dilakukan untuk menguji tingkat kinerja atau performa dari pelatihan model dan menguji tingkat akurasi. Metode pengujian model dilakukan dengan menjalankan model *.weights* YOLO pada dataset uji. Pengujian tingkat akurasi dilakukan melalui perhitungan mAP (*mean Average Precisions*). Untuk pengujian performa model, digunakan metode *ground truth* melalui observasi hasil deteksi model pada gambar uji untuk mengukur tingkat kesalahan deteksi.

Tabel 3.2 Pengujian performa deteksi dengan metode *confusion matrix*

		Actual Class	
		gun	knife
Predicted Class	gun	<i>True Positive</i>	<i>False Positive</i>
	knife	<i>False Negative</i>	<i>True Negative</i>

3.4.1.1 Skenario Pengujian

Skenario pengujian model meliputi pengujian untuk mengukur tingkat kinerja dan akurasi model. Tabel 3.3 Menjelaskan tentang skenario pengujian untuk mengukur tingkat performa dan akurasi model.

Tabel 3.3 Skenario pengujian tingkat performa dan akurasi model YOLO

No	Fungsi Sistem	Rencana Pengujian	Indikator Keberhasilan
1	Performa klasifikasi	Melakukan pengukuran tingkat performa deteksi menggunakan <i>confusion matrix</i> . Tingkat performa model dinilai dari	Persentase kebenaran hasil prediksi

		persentase kebenaran hasil prediksi (tingkat presisi).	
2	Akurasi	Melakukan perhitungan rata-rata mAP (<i>mean Average Precisions</i>).	Persentase hasil mAP YOLO

3.5.2 Rencana Pengujian Fungsional Sistem

Rencana pengujian fungsional sistem berhubungan dengan rumusan masalah yang telah dibuat. Rencana pengujian ini bertujuan untuk mengetahui keberhasilan dari sistem yang akan dibuat. Tabel 3.4 menjelaskan mengenai rancangan pengujian fungsional sistem.

Tabel 3.4 Rencana Pengujian Fungsional Sistem

No	Parameter Pengujian	Rencana Pengujian	Indikator Keberhasilan
1	Kinerja Sistem dalam melakukan deteksi objek	Pengukuran skor deteksi dan jarak efektif deteksi terhadap objek pada kamera webcam dengan parameter resolusi tetap sebesar 5MP.	Mendapatkan jarak efektif deteksi terhadap objek dan skor deteksi pada masing-masing jarak efektif

2	Performa sistem dalam pendeksiian secara <i>realtime</i>	Mengukur tingkat <i>Frame per Second</i> (fps) ketika melakukan deteksi objek pada Raspberry Pi 4.	Mendapatkan hasil pengukuran fps yang presisi
3	Pengiriman Data melalui jaringan internet ke aplikasi android	Mengukur tingkat latency saat pengiriman data	Mendapatkan rata-rata latency pengiriman data
4	Pengujian sistem secara keseluruhan	Menguji fungsional sistem secara keseluruhan	Mendapatkan hasil deteksi yang akurat

3.5 Analisis Kebutuhan Penelitian

Analisa kebutuhan sistem terdiri atas 2 bagian, kebutuhan perangkat keras dan kebutuhan perangkat lunak. Tabel 3.3 merinci kebutuhan perangkat keras, perangkat lunak, dan data yang dibutuhkan dalam penelitian ini.

Tabel 3.5 Kebutuhan Penelitian

Perangkat Keras	Perangkat Lunak	Data
Personal Komputer (PC)	Python IDE	Data Latih
Raspberry Pi 3	Android Studio	Data Uji
Kamera		
Buzzer		

BAB IV

IMPLEMENTASI DAN PENGUJIAN

4.1 Implementasi

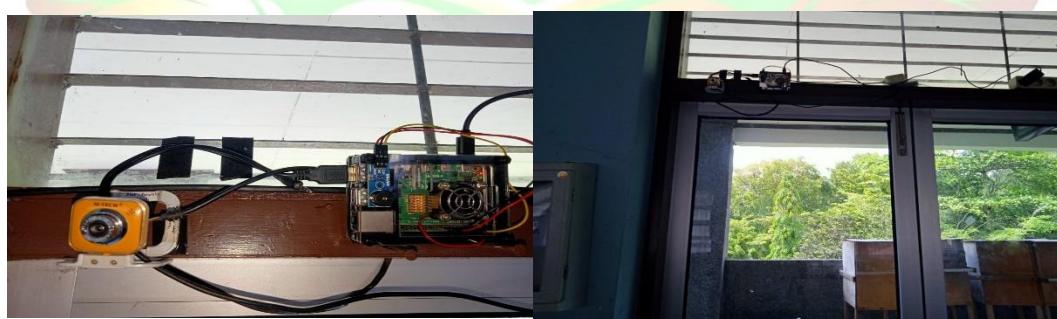
Implementasi sistem dilakukan sesuai dengan rancangan pada BAB III . Dalam penerapannya, terdapat 2 komponen utama, implementasi perangkat keras dan implementasi perangkat lunak.

4.1.1 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 4.1.



(a)

(b)

Gambar 4.1 (a) Perangkat Keras yang digunakan untuk penelitian, dan (b) tempat pemasangan perangkat keras pada lingkungan pengujian

4.1.2 Implementasi Perangkat Lunak

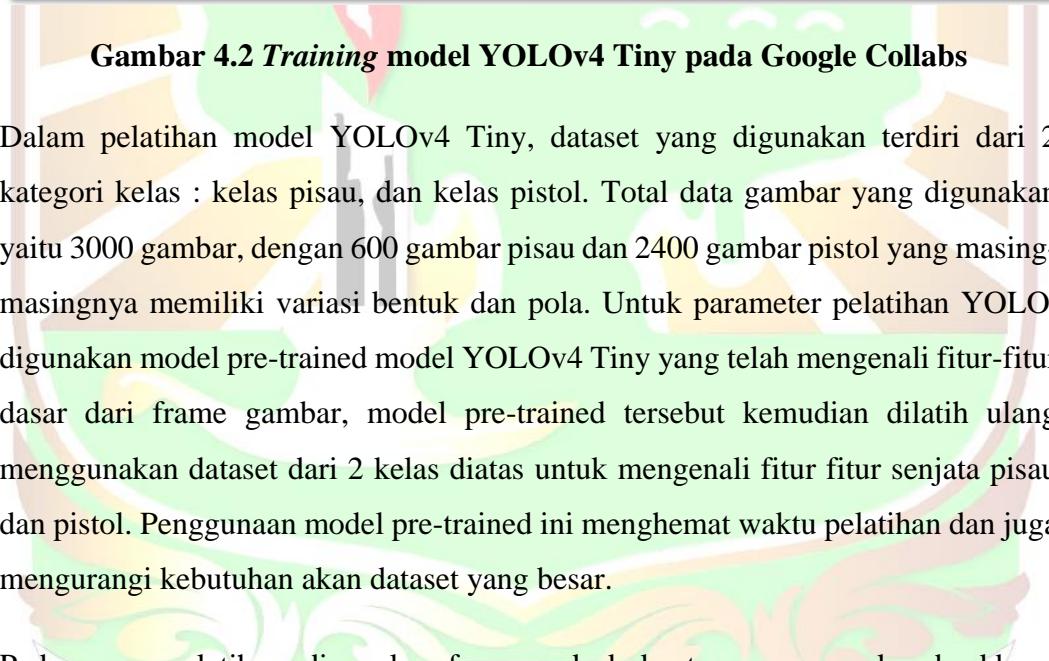
Terdapat 3 Modul perangkat lunak yang menjadi komponen utama dalam keutuhan jalannya sistem. 3 Modul tersebut yaitu :

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.

4.1.2.1 Modul Pelatihan dan Pengujian Model YOLO

Dalam melakukan pelatihan model YOLO, digunakan produk penelitian Google Colab yang menyediakan virtual komputasi melalui cloud untuk pelatihan intensif model-model *machine learning* dan *deep learning*. Untuk itu, digunakan komputasi pada google colab untuk melatih model deteksi senjata pada YOLO. Hasil pelatihan model YOLO pada google colab dapat dilihat pada Gambar 4.2.





```
+ Code + Text
total_bbox = 450276, rewritten_bbox = 0.230747 %
(next mAP calculation at 6145 iterations)

Tensor Cores are used.
Last accuracy mAP@0.5 = 73.62 %, best = 74.73 %
6000: 0.169863, 0.174411 avg loss, 0.000026 rate, 0.509325 seconds, 384000 images, 0.018722 hours left

calculation mAP (mean average precision)...
Detection layer: 30 - type = 28
Detection layer: 37 - type = 28
580
detections_count = 2858, unique_truth_count = 689
class_id = 0, name = pistol, ap = 80.31%           (TP = 460, FP = 125)
class_id = 1, name = knife, ap = 67.36%            (TP = 46, FP = 33)

for conf_thresh = 0.25, precision = 0.76, recall = 0.73, F1-score = 0.75
for conf_thresh = 0.25, TP = 506, FP = 158, FN = 183, average IoU = 57.86 %

IoU threshold = 50 %, used Area-Under-Curve for each unique Recall
mean average precision (mAP@0.50) = 0.738360, or 73.84 %
Total Detection Time: 3 Seconds

Set -points flag:
`-points 101` for MS COCO
`-points 11` for PascalVOC 2007 (uncomment `difficult` in voc.data)
`-points 0` (AUC) for ImageNet, PascalVOC 2010-2012, your custom dataset

mean_average_precision (mAP@0.5) = 0.738360
Saving weights to /content/drive/MyDrive/yolov4_dataset/backup/yolov4-tiny-custom_6000.weights
Saving weights to /content/drive/MyDrive/yolov4_dataset/backup/yolov4-tiny-custom_last.weights
Saving weights to /content/drive/MyDrive/yolov4_dataset/backup/yolov4-tiny-custom_final.weights
If you want to train from the beginning, then use flag in the end of training command: -clear
```

Gambar 4.2 Training model YOLOv4 Tiny pada Google Collabs

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.

Pada proses pelatihan, digunakan framework darknet yang merupakan backbone dari YOLO yang mengoptimasi proses pelatihan dan deteksi objek. Untuk mendapatkan pelatihan model yang optimal, dilakukan penyesuaian parameter sesuai jumlah kelas dan banyaknya perulangan pelatihan yang dilalui. Untuk 2 kelas objek, diterapkan 6000 kali iterasi. Setelah semua dataset dijalankan untuk pelatihan dalam satu sesi iterasi, dilakukan kalkulasi akurasi melalui perhitungan *mean Average Precisions* (mAP) dan dihasilkan model YOLO yang disimpan dengan ekstensi file *.weights*. Dari file *.weights* ini, model menyimpan nilai bobot

hasil pelatihan untuk mengenali fitur objek senjata. Proses pelatihan model YOLO untuk mengenali objek senjata pistol dan pisau dapat dilihat pada gambar Gambar 4.3 dibawah.

```
(next mAP calculation at 6000 iterations)
Last accuracy mAP@0.5 = 84.94 %, best = 85.12 %
5993: 0.238893, 0.169092 avg loss, 0.000026 rate, 0.637527 seconds, 383552 images, 0.020174 hours left
Loaded: 0.001273 seconds

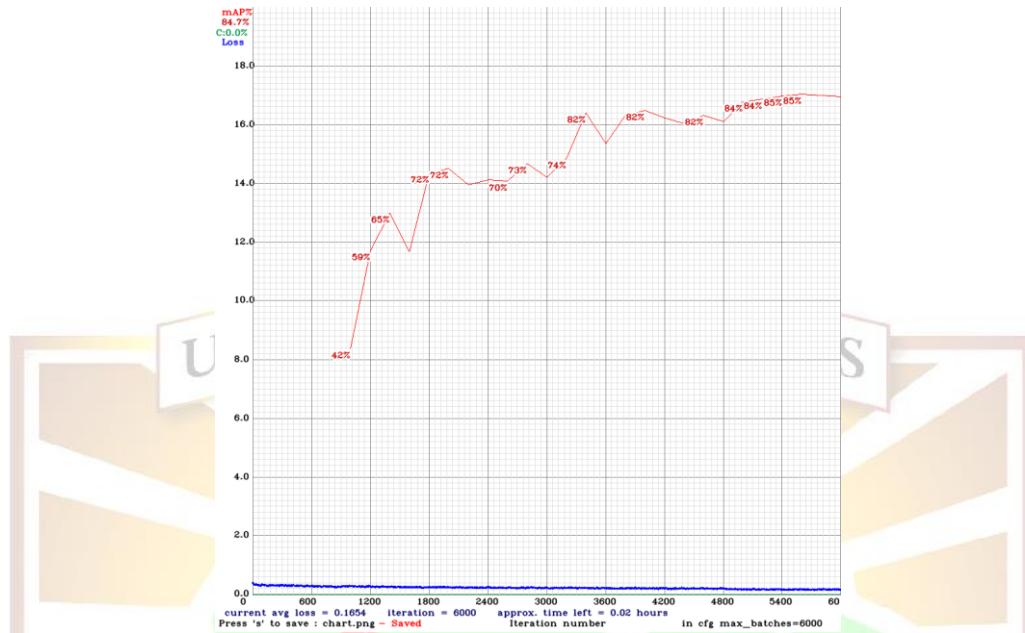
(next mAP calculation at 6000 iterations)
Last accuracy mAP@0.5 = 84.94 %, best = 85.12 %
5994: 0.118421, 0.164025 avg loss, 0.000026 rate, 0.650709 seconds, 383616 images, 0.019984 hours left
Loaded: 0.000040 seconds

(next mAP calculation at 6000 iterations)
Last accuracy mAP@0.5 = 84.94 %, best = 85.12 %
5995: 0.163198, 0.163942 avg loss, 0.000026 rate, 0.667351 seconds, 383680 images, 0.019795 hours left
Loaded: 0.000047 seconds

(next mAP calculation at 6000 iterations)
Last accuracy mAP@0.5 = 84.94 %, best = 85.12 %
5996: 0.161330, 0.163681 avg loss, 0.000026 rate, 0.630000 seconds, 383744 images, 0.019607 hours left
Loaded: 0.000044 seconds
```

Gambar 4.3 Proses pelatihan model YOLOv4 tiny untuk mengenali objek senjata pisau dan pistol

Karena menggunakan *pre-trained* weight dari yolo tiny, atau weight bawaan yang sudah dilatih untuk mengenali fitur dasar dari gambar, maka proses pelatihan menjadi jauh lebih sederhana dan lebih cepat. Selain update nilai loss yang rendah selama pelatihan, dataset yang digunakan juga tidak harus banyak karena model sudah mengenali fitur-fitur dasar, yang dilatih hanya jaringan saraf untuk membedakan objek pada gambar dan mengenali fitur objek. Karena nilai *learning rate* yang kecil, maka nilai update weight untuk setiap iterasi sangat kecil. Pada setiap iterasi pelatihan, tidak selalu iterasi yang besar akan menghasilkan model yang optimal, untuk itu pada setiap 200 iterasi dilakukan perhitungan nilai akurasi dalam metrik mAP. Nilai mAP adalah nilai akurasi keseluruhan dari proses pelatihan model yang diuji dengan data validasi. Iterasi dengan nilai mAP yang tinggi adalah model yang optimal. Untuk itu, dalam penerapan model dipilih bobot atau *weights* dengan nilai mAP tertinggi, yang dari pelatihan didapatkan nilai mAP tertinggi sebesar 85% pada iterasi ke 5800. Plot rangkuman dari proses pelatihan model YOLO dapat dilihat pada gambar 4.4 dibawah ini.



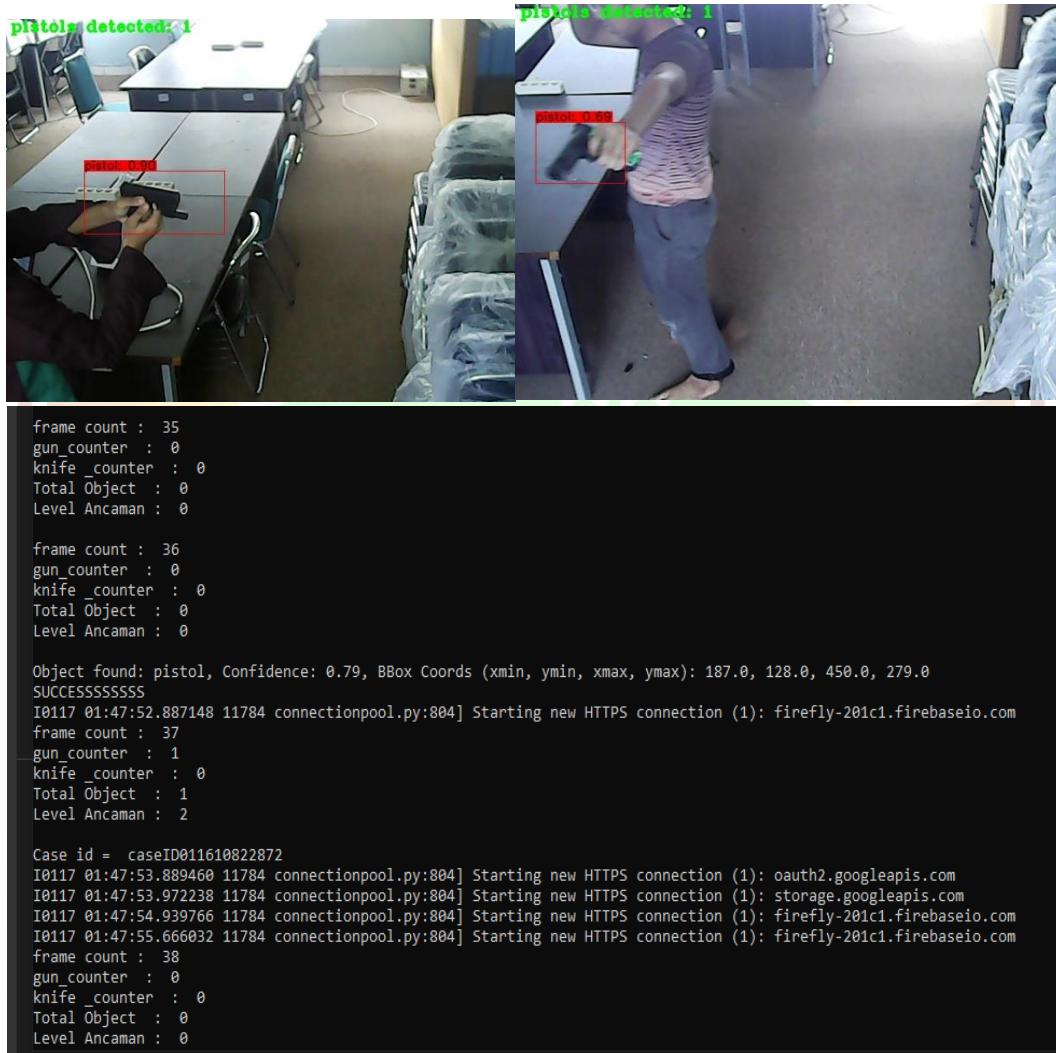
Gambar 4.4 Plot hasil pelatihan model YOLO dan hasil perhitungan mAP selama pelatihan

4.1.2.2 Modul implementasi model pendekripsi objek senjata YOLO 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. Secara alur proses, implementasi perangkat lunak yang dilakukan mengajau pada rancangan proses implementasi sistem pada BAB III di bagian sub bab Implementasi Sistem pada Raspberry Pi. Penerapan Sistem dilakukan pada Sistem Operasi Raspberry OS dengan menggunakan bahasa pemrograman Python 3.6. Dalam membangun model deteksi YOLO, digunakan adaptasi YOLO dari tutorial theAIGuy[26] yang kemudian disesuaikan untuk melatih model pengenalan objek senjata pistol dan pisau. Secara garis besar alur program terdiri dari :

1. Mengambil data frame gambar dari webcam
2. Mengolah gambar dan mendekripsi ada tidaknya senjata dengan model YOLO yang telah dilatih
3. Mengirimkan notifikasi berupa push notification ke aplikasi android jika terdeteksi adanya senjata

- Mengupdate database firebase mengenai informasi pendekripsi dan tingkat ancaman.



Gambar 4.5 Implementasi sistem pendekripsi senjata YOLO pada Raspberry Pi 4

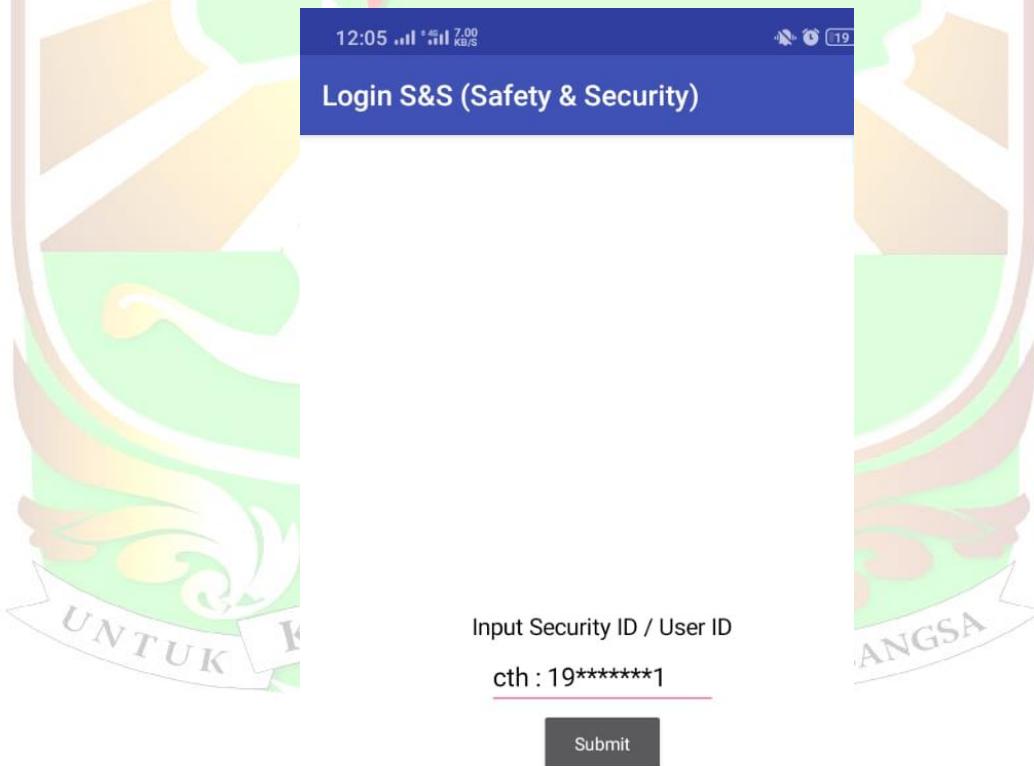
4.1.2.3 Implementasi Aplikasi Android

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.

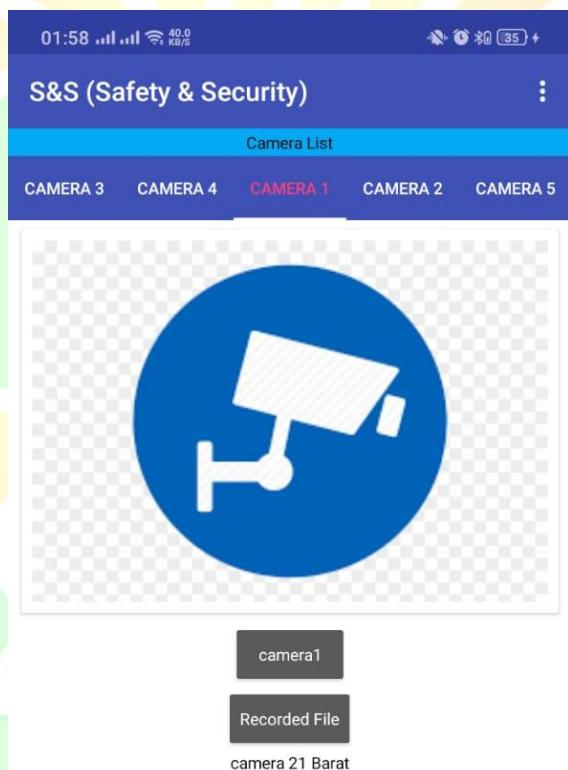
Dalam penerapannya, terdapat lima fungsional utama dari aplikasi android yang dirancang, lima fungsi utama tersebut yaitu :

1. **Halaman Login** : Di awal instalasi program, user yang dalam hal ini operator CCTV ataupun pengawas yang mempunyai wewenang akan melakukan login dengan ID pegawai atau user ID yang diberikan sebelum penerapan sistem. Login dilakukan cukup sekali untuk setiap smartphone android, jika ID ditemukan pada database, maka akan disimpan alamat IMEI dari smartphone agar pengguna dapat dikenali dan tidak harus melakukan login setiap akan menggunakan aplikasi. Jika berhasil login maka user akan ditampilkan menu homepage dari aplikasi. Tampilan halaman login dapat dilihat pada gambar 4.5 dibawah ini.



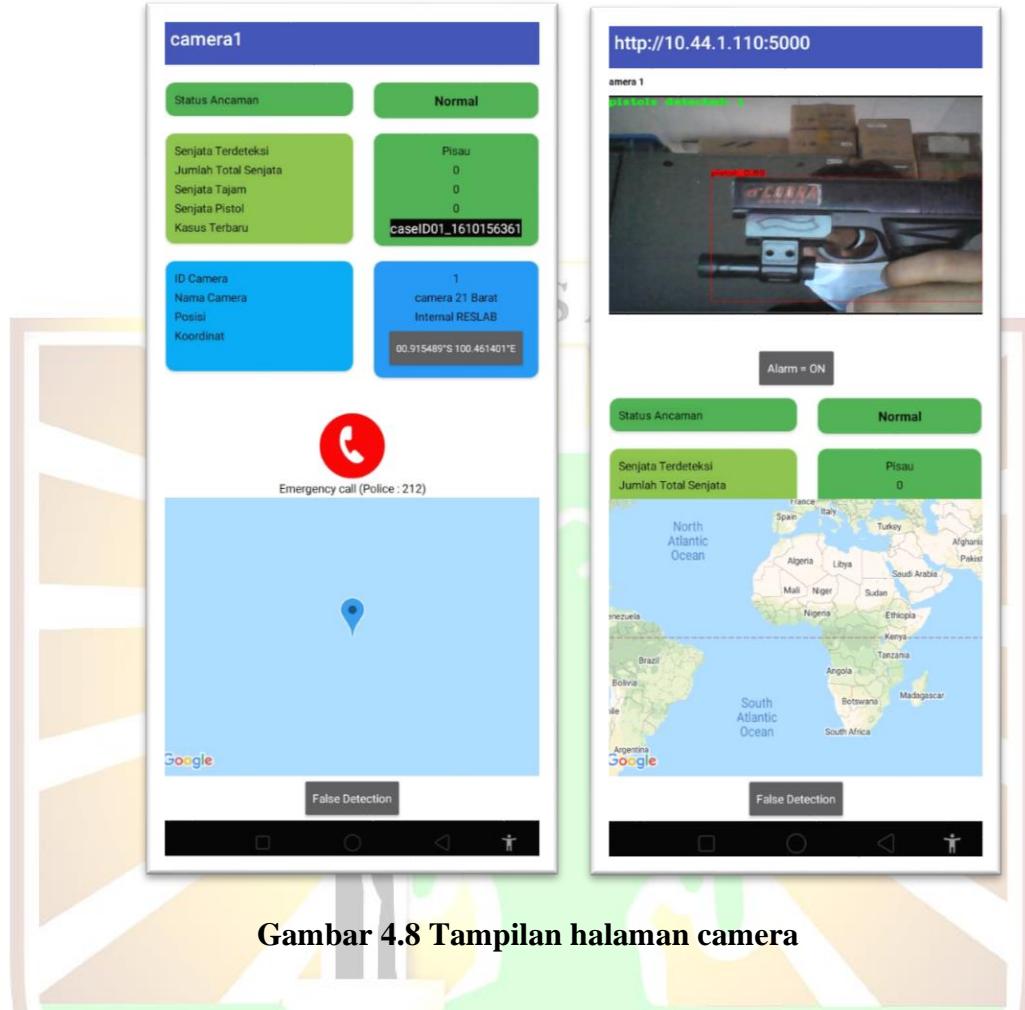
Gambar 4.6 Tampilan halaman login aplikasi

2. **Halaman Utama (Homepage)** : Pada homepage, terdapat list kamera aktif yang dapat diawasi dalam ruang lingkup penerapan sistem. Dari menu kamera, dapat digunakan 2 menu : Camera, dan Recorded File. Menu camera berfungsi untuk menampilkan halaman status terkini kamera dan hasil tangkapan kamera secara realtime. Menu Recorded File berfungsi melihat riwayat pendekripsi senjata dari kamera, yang mana berupa gambar hasil deteksi YOLO. Menu homepage dari aplikasi dapat dilihat pada gambar 4.6 dibawah.



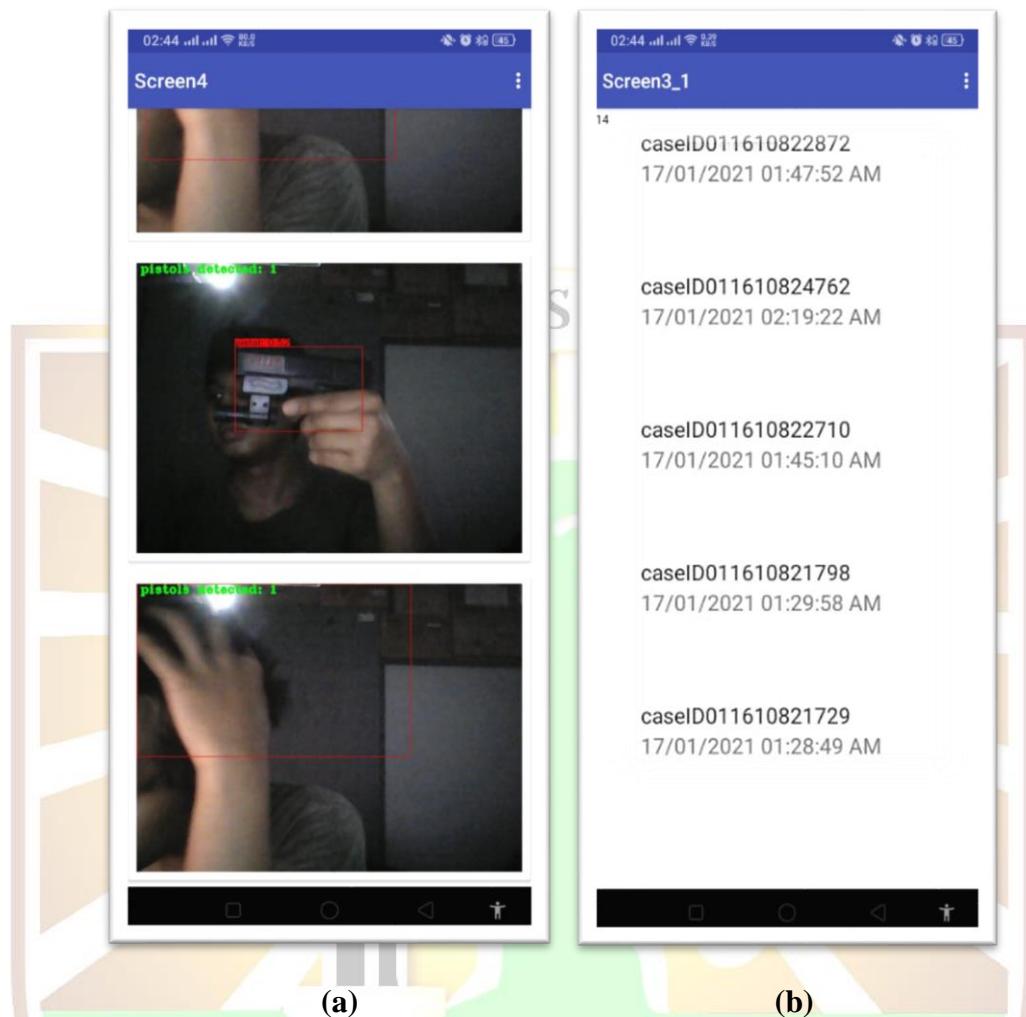
Gambar 4.7 Tampilan Homepage dari aplikasi

3. **Halaman Camera** : Pada halaman Camera, ditampilkan live streaming dari kamera dan status terkini dari kamera, dengan detail status informasi : Status ancaman, Jumlah senjata yang terdeteksi dan tipe senjata yang terdeteksi, ID Kasus saat ini, letak koordinat kamera, dan informasi detail posisi kamera. Selain itu juga terdapat menu untuk pemanggilan jika dirasa perlunya respon yang cepat dari aparat keamanan. Tampilan dari Halaman Camera dapat dilihat pada gambar dibawah.



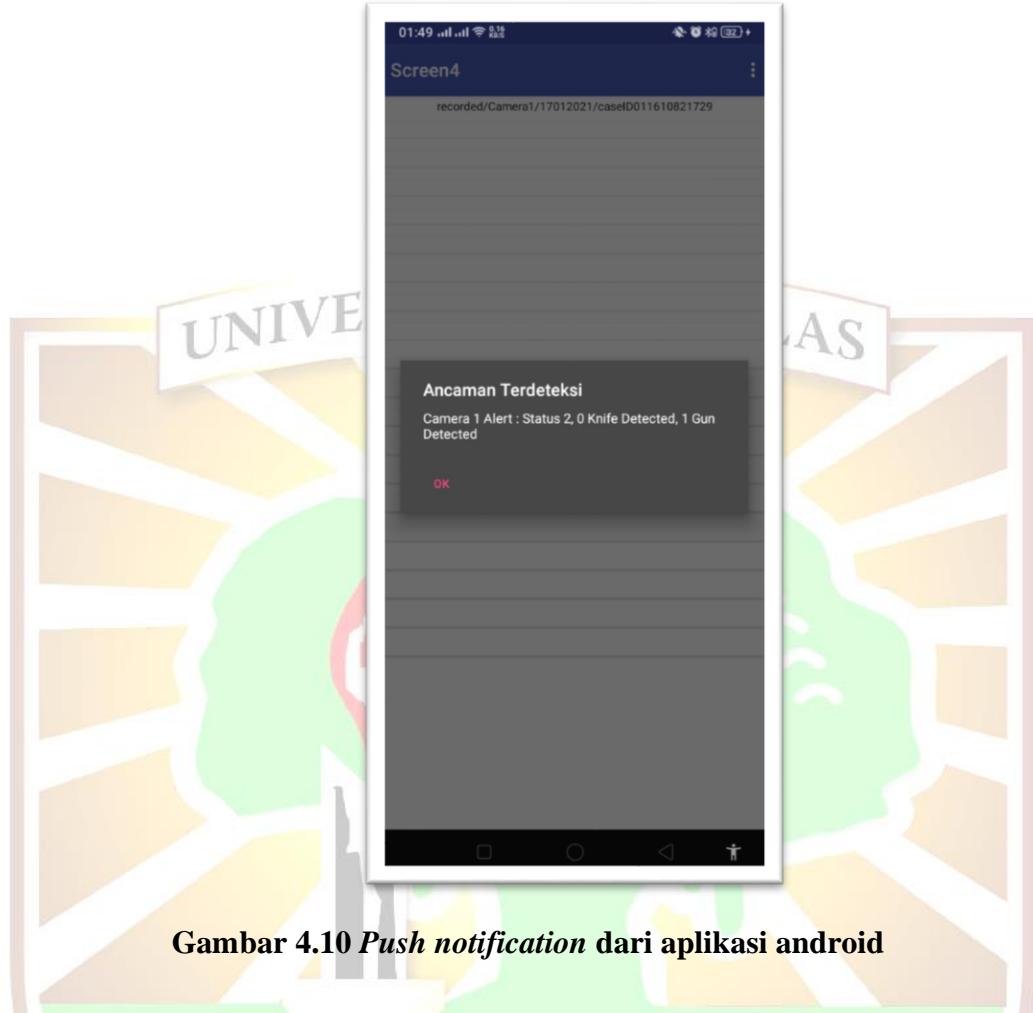
Gambar 4.8 Tampilan halaman camera

4. **Halaman Recorded File** : Pada halaman recorded File, ditampilkan gambar hasil deteksi sistem yang dikelompokkan berdasarkan tanggal, dan waktu kejadian. Tampilan halaman recorded file dapat dilihat pada gambar dibawah.



Gambar 4.9 Tampilan menu recorded file, (a) gambar riyawat hasil deteksi, dan (b) pengelompokan gambar berdasarkan tanggal dan waktu deteksi

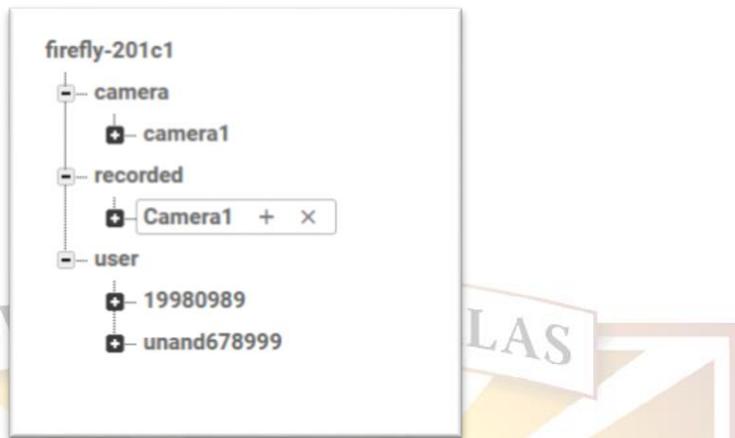
5. ***Push notification*** : *Push notification* berfungsi untuk memberikan notifikasi peringatan kepada aplikasi android berupa *pop up* menu jika layar dalam keadaan terbuka atau *dropdown notification*. Melalui *Push notification*, dikirim informasi status ancaman dan senjata yang terdeteksi seperti dibawah ini.



Gambar 4.10 *Push notification* dari aplikasi android

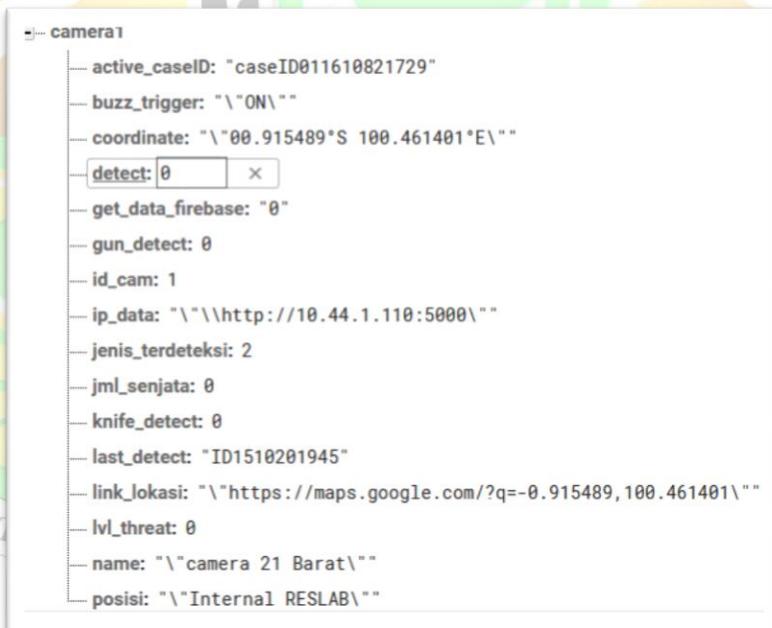
4.1.2.4 Implementasi Database Firebase

Dalam perancangan aplikasi android, digunakan framework **kodular.io** yang merupakan aplikasi app inventor untuk membuat aplikasi android dari modul-modul blok yang ada, sehingga memudahkan dalam pembuatan aplikasi sederhana tanpa harus merancang struktur detail dari aplikasi. Untuk dapat menghubungkan antara aplikasi android dan sistem pendekripsi objek senjata pada Raspberry Pi, digunakan database Firebase sesuai rancangan pada BAB III. Implementasi database pada firebase yaitu :



Gambar 4.11 Tabel-tabel yang digunakan pada database

Pada perancangan database, digunakan 3 tabel data yang terdiri dari : tabel kamera, tabel recorded, dan tabel user. Tabel kamera memuat informasi mengenai kamera dan letak kamera, informasi hasil deteksi, dan trigger untuk menghidupkan atau mematikan buzzer. Struktur dari tabel kamera dapat dilihat pada Gambar 4.12 berikut ini.



Gambar 4.12 Implementasi tabel kamera

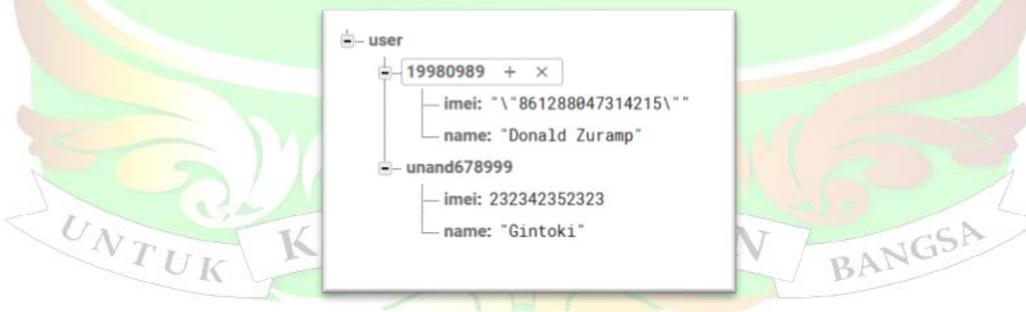
Pada tabel *recorded*, disimpan data-data alamat link url dari gambar-gambar hasil deteksi yang memuat objek senjata. Gambar-gambar hasil deteksi tersebut akan diupload ke firebase storage dan url gambar pada storage disimpan pada database

untuk dapat menampilkan gambar pada aplikasi android dan memudahkan pengelompokan gambar berdasarkan waktu deteksi. Struktur data pada tabel recorded dapat dilihat pada gambar 4.13.



Gambar 4.13 Struktur tabel *recorded* pada database firebase

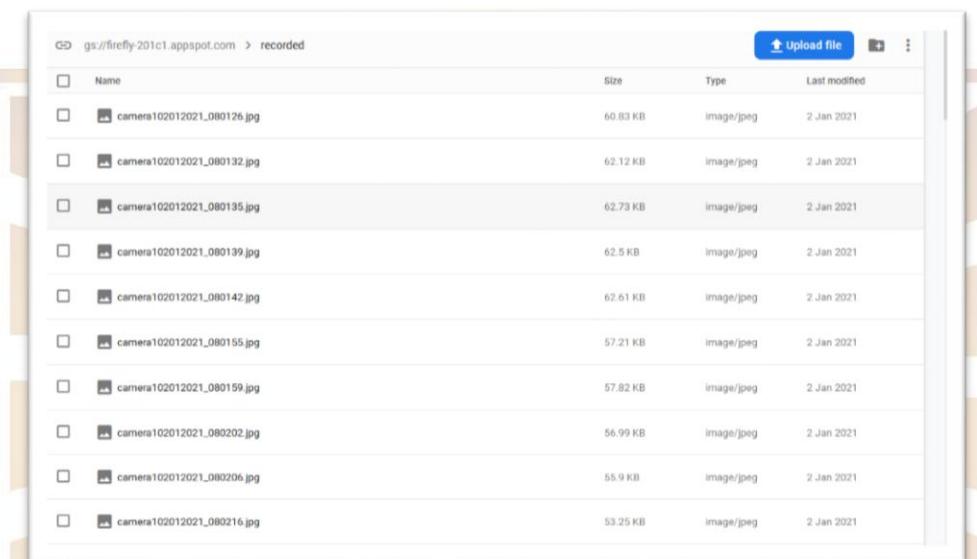
Selanjutnya, tabel user menyimpan data-data user yang berwenang untuk menggunakan aplikasi. Saat pertama menginstall aplikasi, akan dilakukan login berupa ID user yang disesuaikan dengan lembaga terkait. ID user dapat berupa nomor pegawai keamanan atau dalam kasus individu berupa ID user yang diberikan saat pemasangan sistem. Untuk lebih jelasnya, dapat dilihat gambar 4.14 mengenai struktur tabel user pada database firebase.



Gambar 4.14 Struktur tabel user pada database Firebase

Selain penggunaan database pada firebase, juga digunakan penyimpanan pada firebase (firebase storage) untuk menyimpan gambar-gambar hasil deteksi. Gambar diupload saat sistem mendeteksi adanya senjata pada suatu frame gambar dari deteksi realtime pada kamera. Gambar diupload pada firebase storage dalam satu

folder yang sama, untuk itu perlu disimpan data-data link url dari setiap gambar pada database firebase untuk bisa mengelompokkan gambar-gambar hasil deteksi tersebut. Struktur penyimpanan file gambar pada firebase storage dapat dilihat pada gambar 4.15 dibawah ini.



Name	Size	Type	Last modified
camera102012021_080126.jpg	60.83 KB	image/jpeg	2 Jan 2021
camera102012021_080132.jpg	62.12 KB	image/jpeg	2 Jan 2021
camera102012021_080135.jpg	62.73 KB	image/jpeg	2 Jan 2021
camera102012021_080139.jpg	62.5 KB	image/jpeg	2 Jan 2021
camera102012021_080142.jpg	62.61 KB	image/jpeg	2 Jan 2021
camera102012021_080155.jpg	57.21 KB	image/jpeg	2 Jan 2021
camera102012021_080159.jpg	57.82 KB	image/jpeg	2 Jan 2021
camera102012021_080202.jpg	56.99 KB	image/jpeg	2 Jan 2021
camera102012021_080206.jpg	55.9 KB	image/jpeg	2 Jan 2021
camera102012021_080216.jpg	53.25 KB	image/jpeg	2 Jan 2021

Gambar 4.15 struktur penyimpanan file gambar pada Firebase Storage

4.2 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.

4.2.1 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 4.16 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 = 18

Kelas : Knife (Pisau)

True Positive = 104

False Positive = 14

Tabel 4.1 Tabel Confusion Matrix hasil pengujian

		Actual Class	
		Terdeteksi	Normal
Predicted Class	Terdeteksi	True Positive = 246	False Positive = 32
	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 presisi dan recall. Lalu dibandingkan hasil keduanya untuk menghasilkan F1 metrik. Ketiga metrik pengukuran ini memiliki karakteristik masing-masing yang menentukan tingkat penerimaan model dan performa model sebelum di *deploy* sepenuhnya.

Presisi

Presisi mengukur tingkat akurasi dari prediksi model, atau persentase tingkat prediksi yang benar. Dalam kasus ini, tingkat presisi model diukur dari persentase klasifikasi yang benar dari semua hasil prediksi gambar senjata. Untuk mengukur presisi digunakan rumus :

$$\text{Precisions} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}}$$

$$Precisions = \frac{246}{246 + 32} = 0.88$$

Dari perhitungan diatas, didapat tingkat presisi dari model YOLO untuk mendeteksi senjata pistol dan pisau sebesar 88 %.

Recall (Sensitifitas)

Recall mengukur berapa banyak yang benar-benar positif dari hasil prediksi model dibandingkan dengan semua yang dilabeli positif. Dalam kasus ini, tingkat recall diukur dari tingkat hasil klasifikasi senjata yang benar dari seluruh gambar senjata yang ada. Pengukuran tingkat recall dilakukan dengan rumus :

$$Precisions = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

$$Precisions = \frac{246}{246 + 56} = 0.81$$

Dari perhitungan recall, didapat tingkat sensitifitas dari model sebesar 81 %.

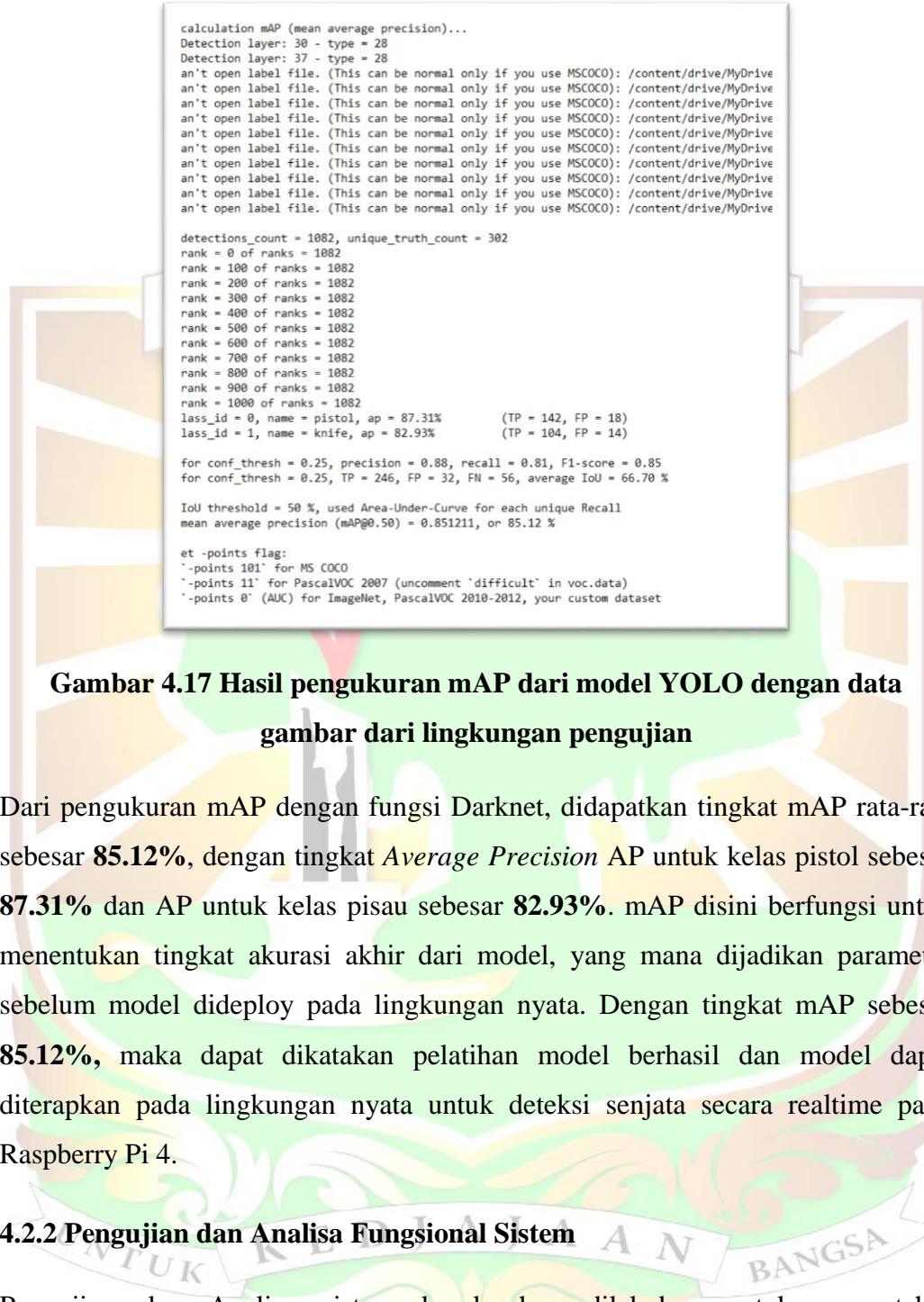
Perhitungan metrik dan recall digunakan untuk mengukur tingkat performa dari model machine learning dan deep learning. Perhitungan recall dan presisi saling bertolak belakang karena perbedaan parameter yang digunakan dalam melakukan kalkulasi performa model. Dalam penerapan model pada lingkungan nyata, tingkat pengujian akurasi atau presisi yang tinggi tidak menjamin model dapat berjalan baik di lingkungan nyata, karena perhitungan tingkat presisi dan akurasi sendiri belum cukup untuk dijadikan pedoman penerapan. Dengan tingginya tingkat presisi model, maka model memiliki tingkat kebenaran deteksi yang sangat tinggi, dimana hampir semua hasil deteksinya benar. Namun, di sisi lain model akan melewatkannya banyak kasus dimana terdapat senjata yang harusnya terdeteksi, namun tidak dianggap sebagai senjata (False Negative). Hal ini menjadikan metrik pengukuran performa model perlu keseimbangan antara tingkat presisi dan recall. Untuk itu terdapat metrik pengukuran tingkat performa model yang mencari titik seimbang antara kedua metrik diatas (presisi dan recall). Metrik ini disebut F1 Score,dengan anotasi :

$$F1 = 2 * \frac{Precision * recall}{Precision + recall}$$

$$F1 = 2 * \frac{0.88 * 0.81}{0.88 + 0.81} = 0.84$$

Hasil F1 score dari pengujian yaitu sebesar 84 %. Dalam penerapan model, perlu dicari tingkat metrik pengukuran performa yang menjadi prioritas agar tujuan akhir dari model dapat dicapai. Jika model melakukan klasifikasi terhadap objek yang dimana kegagalan mendeteksi akan berakibat fatal, maka metrik recall perlu ditingkatkan nilainya. Di sisi lain, jika model berfokus pada tingkat persentase kebenaran deteksi, maka metrik presisi yang perlu ditingkatkan. Dalam penelitian ini, fokus utama yang digunakan yaitu harmoni atau keseimbangan tingkat presisi dan recall untuk mendukung penerapan model yang optimal pada kamera CCTV. Karena sistem deteksi objek senjata berperan sebagai support sistem bagi operator CCTV, maka harmoni atau titik keseimbangan dari kedua metrik dirasa perlu dalam penerapan di lingkungan penerapan sistem.

Dalam pengujian kualitas model YOLO, terdapat metrik lain selain pengukuran tingkat presisi dan recall, yaitu *medium Average Precisions* (mAP). Saat melakukan deteksi pada frame gambar, selain menentukan ada tidaknya objek pada gambar, juga dilakukan lokalisasi atau proses menemukan letak objek pada gambar berupa koordinat objek dan besar bounding dari objek. Untuk itu dari pengukuran tingkat presisi dan recall, juga terdapat parameter tingkat kebenaran letak bounding box yang diukur dengan tingkat *Intersection over Union* (IoU). Untuk menentukan tingkat akurasi dari klasifikasi dan lokalisasi model, maka digunakan metrik mAP. Untuk mengukur tingkat mAP, perlu dilakukan pengukuran presisi dan recall untuk setiap objek yang ada pada gambar, dengan tingkat prediksi bounding box diatas batas threshold IoU. Untuk mempermudah pengukuran mAP dari 452 gambar uji, digunakan fungsi pengukuran mAP dari Darknet. Hasil dari pengukuran tingkat mAP pada model YOLO adalah :



```
calculation mAP (mean average precision)...
Detection layer: 30 - type = 28
Detection layer: 37 - type = 28
an't open label file. (This can be normal only if you use MSCOCO): /content/drive/MyDrive
an't open label file. (This can be normal only if you use MSCOCO): /content/drive/MyDrive
an't open label file. (This can be normal only if you use MSCOCO): /content/drive/MyDrive
an't open label file. (This can be normal only if you use MSCOCO): /content/drive/MyDrive
an't open label file. (This can be normal only if you use MSCOCO): /content/drive/MyDrive
an't open label file. (This can be normal only if you use MSCOCO): /content/drive/MyDrive
an't open label file. (This can be normal only if you use MSCOCO): /content/drive/MyDrive
an't open label file. (This can be normal only if you use MSCOCO): /content/drive/MyDrive
an't open label file. (This can be normal only if you use MSCOCO): /content/drive/MyDrive
detections_count = 1082, unique_truth_count = 302
rank = 0 of ranks = 1082
rank = 100 of ranks = 1082
rank = 200 of ranks = 1082
rank = 300 of ranks = 1082
rank = 400 of ranks = 1082
rank = 500 of ranks = 1082
rank = 600 of ranks = 1082
rank = 700 of ranks = 1082
rank = 800 of ranks = 1082
rank = 900 of ranks = 1082
rank = 1000 of ranks = 1082
lass_id = 0, name = pistol, ap = 87.31%           (TP = 142, FP = 18)
lass_id = 1, name = knife, ap = 82.93%            (TP = 104, FP = 14)

for conf_thresh = 0.25, precision = 0.88, recall = 0.81, F1-score = 0.85
for conf_thresh = 0.25, TP = 246, FP = 32, FN = 56, average IoU = 66.70 %

IoU threshold = 50 %, used Area-Under-Curve for each unique Recall
mean average precision (mAP@0.50) = 0.851211, or 85.12 %

et -points flag:
`-points 101` for MS COCO
`-points 11` for PascalVOC 2007 (uncomment `difficult` in voc.data)
`-points 0` (AUC) for ImageNet, PascalVOC 2010-2012, your custom dataset
```

Gambar 4.17 Hasil pengukuran mAP dari model YOLO dengan data gambar dari lingkungan pengujian

Dari pengukuran mAP dengan fungsi Darknet, didapatkan tingkat mAP rata-rata sebesar **85.12%**, dengan tingkat *Average Precision AP* untuk kelas pistol sebesar **87.31%** dan AP untuk kelas pisau sebesar **82.93%**. mAP disini berfungsi untuk menentukan tingkat akurasi akhir dari model, yang mana dijadikan parameter sebelum model dideploy pada lingkungan nyata. Dengan tingkat mAP sebesar **85.12%**, maka dapat dikatakan pelatihan model berhasil dan model dapat diterapkan pada lingkungan nyata untuk deteksi senjata secara realtime pada Raspberry Pi 4.

4.2.2 Pengujian dan Analisa Fungsional Sistem

Pengujian dan Analisa sistem keseluruhan dilakukan untuk mengetahui kemampuan sistem secara keseluruhan serta melakukan analisa pada data yang diperoleh dari hasil pengujian. Beberapa parameter pengujian yang digunakan untuk menguji kualitas dan performa sistem yaitu : Kinerja sistem dalam melakukan deteksi objek, performa sistem dalam pendekripsi secara *realtime*, dan pengujian kualitas jaringan untuk pengiriman data.

4.2.2.1 Kinerja Sistem dalam melakukan deteksi objek secara *realtime*

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 :

1. Pengujian jarak efektif deteksi : Menguji kemampuan sistem dalam melakukan deteksi pada beberapa variasi jarak dan posisi senjata. Tujuan dari pengujian ini yaitu untuk mendapatkan jarak efektif deteksi pada pendekripsi dengan perangkat Raspberry Pi 4. Dalam pengambilan data untuk pengujian, digunakan variasi jarak setiap 50 cm.
2. Pengujian kemampuan deteksi dalam beberapa variasi pencahayaan lingkungan.
3. Pengujian kemampuan sistem dalam mendeteksi lebih dari satu senjata : mencoba variasi beberapa senjata dalam satu frame untuk menghitung tingkat kemampuan sistem dalam mendeteksi beberapa senjata.
4. Pengujian sistem pada objek selain senjata.

4.2.2.1.1 Pengujian Jarak Efektif Deteksi

Untuk pengujian jarak efektif deteksi, dilakukan deteksi objek senjata dari beberapa variasi jarak setiap 50cm, dengan jarak yang digunakan antara 50cm hingga 1000 cm. Untuk setiap jarak, digunakan variasi 20 posisi memegang senjata untuk senjata pistol dan 10 variasi posisi untuk senjata pisau. Dari masing-masing jarak, diukur kemampuan deteksi objek senjata untuk setiap posisi. Beberapa sampel gambar yang digunakan untuk pengujian jarak efektif deteksi dapat dilihat pada gambar 4.18 dibawah.



Gambar 4.18 Beberapa sampel gambar uji yang digunakan untuk melakukan pengujian jarak deteksi efektif

Dari pengujian jarak yang dilakukan dengan berbagai variasi posisi senjata dan sudut kamera, maka didapatkan hasil pengujian jarak efektif deteksi sebagai berikut :

Tabel 4.2 Hasil pengujian jarak efektif deteksi

Kelas Senjata	Jarak	Hasil Deteksi	Skor rata-rata Deteksi
Pistol	50 cm	15 dari 20 Gambar	0.76
	100 cm	19 dari 20 Gambar	0.82
	150 cm	17 dari 20 Gambar	0.68
	200 cm	18 dari 20 Gambar	0.61
	250 cm	18 dari 20 Gambar	0.72
	300 cm	19 dari 20 Gambar	0.68
	350 cm	20 dari 20 Gambar	0.60
	400 cm	20 dari 20 Gambar	0.58
	450 cm	19 dari 20 Gambar	0.59
	500 cm	17 dari 20 Gambar	0.58

	550 cm	15 dari 20 Gambar	0.55
	600 cm	14 dari 20 Gambar	0.52
	650 cm	12 dari 20 Gambar	0.56
	700 cm	5 dari 20 Gambar	0.52
	750 cm	4 dari 20 Gambar	0.51
	800 cm	3 dari 20 Gambar	0.54
	850 cm	1 dari 20 Gambar	0.61
	900 cm	1 dari 20 Gambar	0.55
	950 cm	4 dari 20 Gambar	0.51
	1000 cm	0 dari 20 Gambar	-
Pisau	50 cm	7 dari 10 Gambar	0.80
	100 cm	8 dari 10 Gambar	0.68

	150 cm	7 dari 10 Gambar	0.78
	200 cm	5 dari 10 Gambar	0.58
	250 cm	1 dari 10 Gambar	0.65
	300 cm	0 dari 10 Gambar	-
	350 cm	0 dari 10 Gambar	-
	400 cm	0 dari 10 Gambar	-
	450 cm	0 dari 10 Gambar	-
	500 cm	0 dari 10 Gambar	-

Dari Tabel 4.2 diatas, dapat dilihat pola deteksi objek senjata berdasarkan variasi jarak dan posisi senjata. Dari hasil yang didapatkan, untuk senjata pistol didapatkan jarak efektif deteksi sejauh 650 cm atau 6,5 m. Sistem mampu mendeteksi adanya senjata pistol dari beberapa variasi posisi dan mendeteksi lebih dari setengahnya pada jarak antara 50 cm – 650 cm, untuk itu didapatkan jarak efektif sejauh 650 cm.

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.

4.2.2.1.2 Pengujian kemampuan deteksi dalam beberapa variasi pencahayaan lingkungan

Untuk pengujian kinerja sistem dalam melakukan deteksi objek senjata, dilakukan pengujian efek pencahayaan lingkungan terhadap kemampuan deteksi sistem yang diterapkan pada Raspberry Pi 4. Hasil pengujian dapat dilihat pada tabel 4.4. dibawah.

Tabel 4.3 Pengujian kinerja deteksi objek terhadap beberapa variasi pencahayaan lingkungan

No	Pencahayaan	Hasil Deteksi	Keterangan Deteksi
1	Terang (Intensitas Cahaya : 352 lx)		5 dari 6 objek senjata terdeteksi

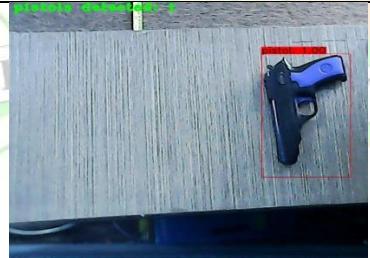
2	Cukup Terang (Intensitas Cahaya : 192 lx)		6 dari 6 objek senjata terdeteksi
3	Redup (Intensitas Cahaya : 116 lx)		5 dari 6 objek senjata terdeteksi
4	Cukup Gelap (Intensitas Cahaya : 50 lx)		3 dari 7 objek senjata terdeteksi
5	Gelap (Intensitas Cahaya : 15 lx)		0 dari 7 objek senjata terdeteksi

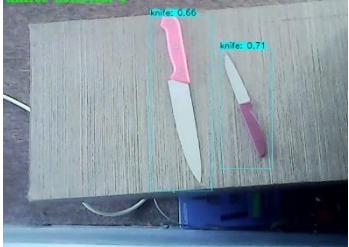
Dari tabel 4.3, dapat dilihat pengaruh cahaya terhadap kemampuan deteksi objek pada sistem Raspberry Pi. Pendekripsi dilakukan menggunakan kamera webcam 5.0 MP dengan resolusi input 640x480 Pixel. Dari hasil yang didapatkan, efek pencahayaan ruangan memiliki pengaruh eksternal dalam kinerja deteksi. Untuk pembagian kualitas pencahayaan, digunakan sensor ldr untuk mengukur intensitas cahaya ruangan pada lingkungan uji. Dari hasil pada tabel 4.3, dapat dilihat efektifitas kemampuan deteksi terhadap kondisi pencahayaan optimal pada kondisi terang dengan pencahayaan yang merata pada objek.

4.2.2.1.3 Pengujian kemampuan sistem dalam mendekripsi jumlah senjata

Pengujian dilakukan untuk menguji kinerja sistem dalam menentukan jumlah senjata dari 2 kelas senjata yang ada, senjata pistol dan pisau. Hasil pengujian dapat dilihat pada tabel 4.4 dibawah.

Tabel 4.4 Pengujian deteksi pada beberapa objek senjata

No	Jumlah Senjata	Hasil Deteksi	Keterangan Deteksi
1	1 Pisau		1 dari 1 objek senjata terdeteksi
2	1 Pistol		1 dari 1 objek senjata terdeteksi

3	2 Pisau		2 dari 2 objek senjata terdeteksi
4	2 Pistol		2 dari 2 objek senjata terdeteksi
5	1 Pisau 1 Pistol		1 dari 2 objek senjata terdeteksi
6	3 Pisau		3 dari 3 objek senjata terdeteksi
7	3 Pistol		3 dari 3 objek senjata terdeteksi

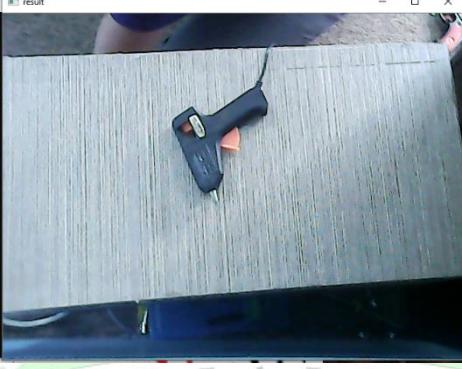
8	4 Pisau		4 dari 4 objek senjata terdeteksi
9	4 Pistol		4 dari 4 objek senjata terdeteksi
10	1 Pistol 3 Pisau		1 dari 4 objek senjata terdeteksi
11	2 Pistol 3 Pisau		2 dari 5 objek senjata terdeteksi
12	4 Pistol 2 Pisau		5 dari 6 objek senjata terdeteksi

Dari hasil pengujian kinerja sistem dalam melakukan deteksi jumlah senjata, dapat dikatakan kualitas deteksi jumlah cukup akurat, walaupun dipengaruhi oleh banyak faktor eksternal lainnya seperti pencahayaan dan posisi senjata. Dalam melakukan pengujian dengan beberapa model senjata, didapatkan kesimpulan bahwa sistem mampu mendeteksi jumlah senjata dengan cukup baik. Selain itu, juga di didapatkan hasil kualitas kemampuan deteksi jumlah senjata pistol lebih tinggi daripada senjata pisau.

4.2.2.1.4 Pengujian kinerja sistem dalam mendeteksi objek selain senjata

Dalam penerapan sistem pada Raspberry Pi, perlu diperhatikan kemampuan sistem dalam membedakan objek senjata dan bukan senjata, untuk itu dilakukan beberapa pengujian dengan melakukan deteksi terhadap objek-objek yang bukan senjata. Hasil pengujian.

Tabel 4.5 Pengujian sistem pada objek selain senjata

No	Jumlah Objek	Hasil Deteksi	Keterangan Deteksi
1	1		0 Kesalahan Deteksi

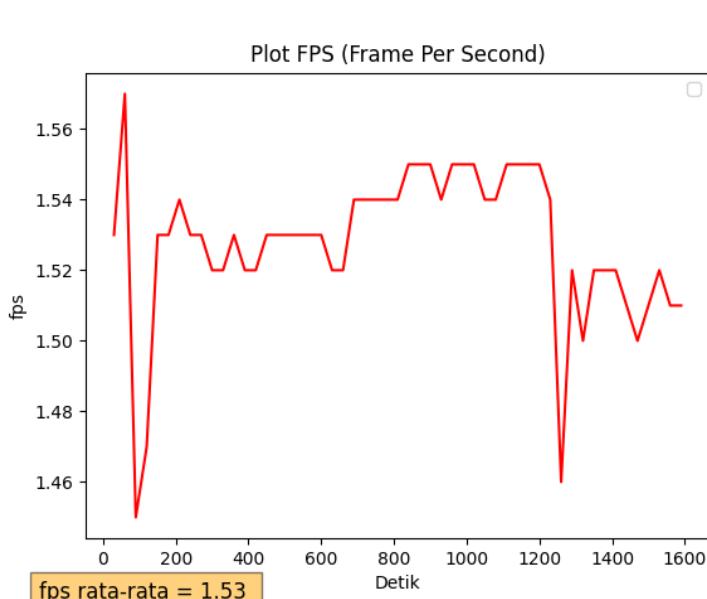
2	9 Objek		0 Kesalahan Deteksi
3	9 Objek		1 Kesalahan Deteksi
4	6 Objek		2 Kesalahan Deteksi
5	11 Objek		0 Kesalahan Deteksi

6	14 Objek		0 Kesalahan Deteksi
---	----------	--	---------------------

Dari tabel 4.5 diatas, dapat disimpulkan bahwa kualitas deteksi objek senjata pada sistem berjalan dengan baik dan sistem mampu membedakan objek senjata dan bukan senjata dengan tepat.

4.2.2.2 Performa Sistem dalam pendekslan secara realtime

Pengujian dilakukan untuk melihat tingkat performa pendekslan 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.

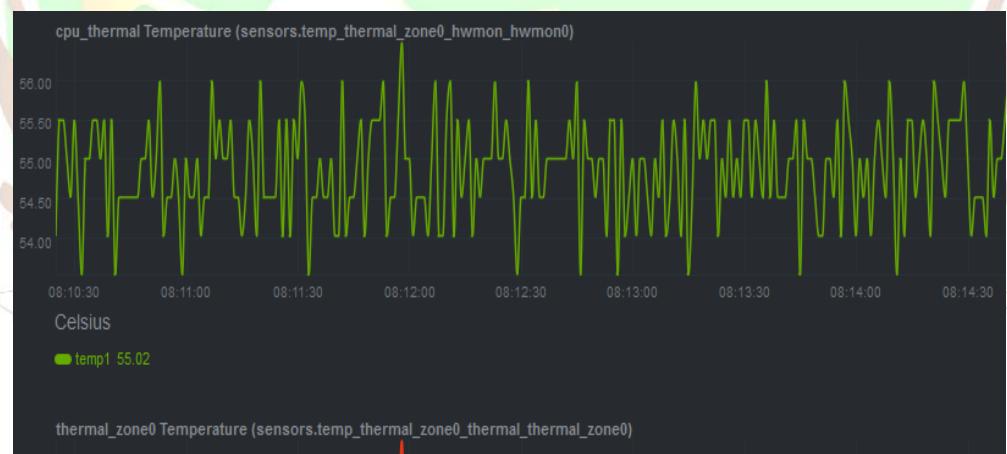


Gambar 4.19 Grafik hasil perhitungan fps rata-rata selama 1600 detik

Dari hasil perhitungan fps pada Raspberry Pi 4, didapatkan fps rata-rata sebesar 1.53 frame per second. Nilai fps yang didapatkan tergolong kecil untuk implementasi secara realtime, namun masih mencukupi untuk implementasi realtime dengan terdapat sedikit jeda antar frame yang ditampilkan, yang mana menjadi batasan dalam melakukan deteksi secara realtime. Selain mengukur tingkat fps, juga dilihat performa perangkat keras Raspberry Pi dalam melakukan deteksi objek senjata. Gambar 4.20 menjelaskan plot utilisasi hardware dari Raspberry Pi selama melakukan deteksi objek dengan model YOLOv4 Tiny.



(a)

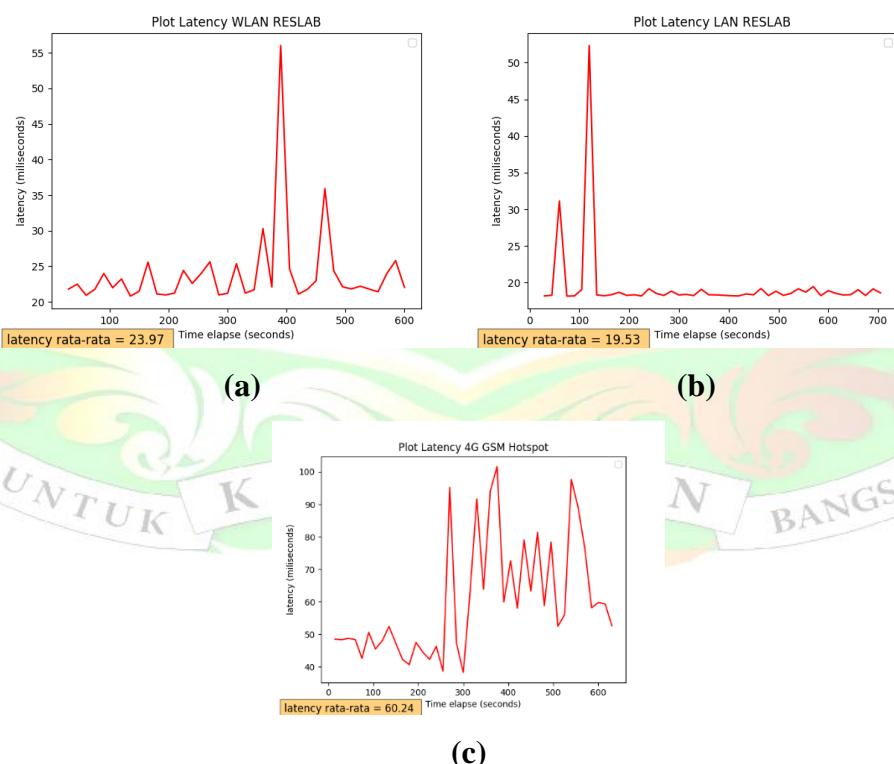


(b)

Gambar 4.20 (a) Grafik utilisasi perangkat Raspberry Pi 4 Model B+ ketika menjalankan program deteksi objek senjata, (b) temperatur CPU Raspberry Pi 4 dalam menjalankan program deteksi objek senjata.

4.2.2.3 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 konektifitas 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 4.21 berikut ini.



Gambar 4.21 (a) plot latency untuk konektivitas jaringan LAN, (b) plot latency untuk jaringan WLAN, dan (c) 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. Dari hasil yang didapatkan, dapat disimpulkan bahwa untuk mendukung pengiriman data secara *realtime*, maka kualitas pengiriman data ke database juga mempengaruhi. Selain kualitas jaringan antara sistem Raspberry Pi 4 dengan database, juga dipengaruhi oleh koneksi jaringan dari database ke perangkat *mobile* android. Namun, karena perangkat mobile bersifat *portable* dan bergerak, maka pengujian kualitas jaringan tidak dapat dilakukan karena kondisi jaringan yang selalu berubah ubah. Dalam hal ini, kualitas jaringan yang digunakan untuk konektifitas Raspberry Pi ke database memiliki kualitas terbaik pada jaringan LAN, dengan waktu latency yang rendah dan kondisi jaringan yang stabil. Sebaliknya, untuk jaringan 4G GSM didapatkan hasil pengiriman data dengan waktu latency yang besar sehingga menimbulkan delay pengiriman data yang besar. Selain itu, pada jaringan 4G GSM juga terdapat turbulensi yang besar sehingga tidak stabil dibandingkan melalui konektivitas LAN dan WLAN. 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.

4.2.2.4 Pengujian Fungsional Sistem secara Keseluruhan

Pada pengujian keseluruhan fungsional sistem, dilakukan pengujian efektifitas sistem saat diterapkan secara struktural, mulai dari kemampuan sistem pendekripsi objek senjata hingga tampilan pada aplikasi android. Untuk itu, dalam pengujian ini digunakan beberapa skenario kemungkinan penyerangan bersenjata yang umum ditemui dan dilihat kemampuan sistem dalam melakukan deteksi hingga respon dari user terhadap hasil deteksi. Data-data hasil pengujian dari beberapa percobaan dapat dilihat pada tabel 4.6 dibawah.

Tabel 4.6 Hasil pengujian sistem secara keseluruhan

Percobaan	Skenario Pengujian	Hasil Deteksi	Keterangan Deteksi	Respon sistem	Respon user
1	Situasi normal, tidak ada objek senjata	0 objek senjata terdeteksi	-	Mengupdate tampilan hasil deteksi pada aplikasi	-
2	Situasi normal, tidak ada objek senjata	1 objek pisau terdeteksi	1 kesalahan deteksi	Mengupdate tampilan hasil deteksi pada aplikasi Mengirim notifikasi ke aplikasi	Menekan menu false detection untuk mereset parameter sistem
3	1 ancaman senjata pisau	1 objek pisau terdeteksi	Hasil deteksi sesuai dengan keadaan sebenarnya	Mengupdate tampilan hasil deteksi pada aplikasi Mengirim notifikasi ke aplikasi Buzzer aktif	Melakukan pemanggilan aparat keamanan melalui menu call pada aplikasi Menghidupkan buzzer (alarm)
4	2 ancaman senjata pisau	2 objek pisau terdeteksi	Hasil deteksi sesuai dengan keadaan sebenarnya	Mengupdate tampilan hasil deteksi pada aplikasi Mengirim notifikasi ke aplikasi Buzzer aktif	Melakukan pemanggilan aparat keamanan melalui menu call pada aplikasi Menghidupkan buzzer (alarm)
5	1 ancaman senjata pistol	1 objek senjata pisau dan 1 objek	Hasil deteksi sesuai dengan	Mengupdate tampilan hasil	Melakukan pemanggilan aparat keamanan melalui

	dan 1 ancaman senjata pisau	senjata pistol terdeteksi	keadaan sebenarnya	deteksi pada aplikasi Mengirim notifikasi ke aplikasi Buzzer aktif	menu call pada aplikasi Menghidupkan buzzer (alarm)
6	1 ancaman senjata pisau	2 objek senjata pisau terdeteksi	1 kesalahan deteksi	Mengupdate tampilan hasil deteksi pada aplikasi Mengirim notifikasi ke aplikasi	Melakukan pemanggilan aparat keamanan melalui menu call pada aplikasi
7	2 ancaman senjata pistol	2 objek senjata pistol terdeteksi	Hasil deteksi sesuai dengan keadaan sebenarnya	Mengupdate tampilan hasil deteksi pada aplikasi Mengirim notifikasi ke aplikasi	Melakukan pemanggilan aparat keamanan melalui menu call pada aplikasi
8	2 ancaman senjata pisau dan 1 senjata pistol	2 objek senjata pisau terdeteksi	1 objek senjata pistol tidak terdeteksi	Mengupdate tampilan hasil deteksi pada aplikasi Mengirim notifikasi ke aplikasi	Melakukan pemanggilan aparat keamanan melalui menu call pada aplikasi

Tabel 4.6 diatas merupakan rangkuman dari pengujian fungsional sistem yang dilakukan, dengan data-data pengujian yang terdapat pada **lampiran 3** tugas akhir, dimana memuat data-data gambar hasil deteksi pada beberapa skenario pengujian yang berbeda. Dari hasil pengujian sistem secara keseluruhan yang mencoba mensimulasikan kejadian berdasarkan situasi nyata, didapatkan hasil deteksi sistem yang cukup akurat dan mampu diterapkan untuk membantu pengawasan aparat keamanan agar mengurangi adanya human error dalam pengawasan kamera cctv dan mendapatkan pelaporan yang cepat saat adanya kemungkinan penyerangan bersenjata.

BAB V

PENUTUP

5.1 Kesimpulan

Dari hasil implementasi 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 **85.12%**, dengan tingkat AP (*Average Precision*) untuk kelas pistol sebesar **87.31%** dan AP untuk kelas pisau sebesar **82.93%**.
3. Pada implementasi model di perangkat Raspberry Pi 4, didapatkan performa yang mencukupi untuk pendekslsian 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.

5.2 Saran

Mengingat masih terdapat keterbatasan dalam melakukan penelitian serta cakupan penelitian yang diambil. maka diusulkan beberapa pengembangan dalam peneltian, antara lain :

1. Memperbanyak data latih yang digunakan serta variasi model senjata pada data latih untuk pelatihan model YOLO
2. Perbandingan beberapa tipe YOLO terhadap kualitas akurasi dan performa deteksi realtime.
3. Pengembangan lanjutan untuk melihat performa YOLO dalam deteksi objek dibandingkan dengan algoritma deteksi objek lainnya.

Daftar Pustaka

- [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=NjZjMDExNGVkJc1MTdhMzMwNjM4NzFm&xzmn=aHR0cHM6Ly93d3cuYnBzLmdvLmlkL3B1YmxpY2F0aW9uLzIwMTkyMTIyNjZjMDExNGVkJc1MTdhMzMwNjM4NzFmL3N0YXRpc3Rpay1rcmltaW5hbC0yMDE5Lmh0bWw%3D&twoadfnoarfeauf=MjAyMC0wOS0wNiAwMD01OTozOQ%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] Oyedeffi, 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 VideosUsing Deep Learning*. NEUCOM (NeuroComputing) 18416, Vol 275, 66-72. DOI : https://arxiv.org/abs /1702.05147.
- [8] Varanasi Sushant, 2020. Artificial Intelligence: A game-changer for the Healthcare industry. Diakses pada 6 Agustus 2020 : <https://medium.com/@sallyrobotics.blog/artificial-intelligence-a-game-changer-for-the-healthcare-industry-1867093eed72>.

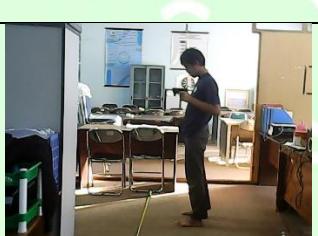
- [9] Serokel. April 2020. *Artificial Intelligence vs. Machine Learning vs. Deep Learning*: What's the Difference. Diakses pada 2 September 2020 : <https://medium.com/@sallyrobotics.blog/artificial-intelligence-a-game-changer-for-the-healthcare-industry-1867093eed72>.
- [10] Nicholson, Chris. 2020. *Artificial Intelligence (AI) vs. Machine Learning vs. Deep Learning*. Diakses pada 6 Agustus 2020 : <https://wiki.pathmind.com/ai-vs-machine-learning-vs-deep-learning>.
- [11] IBM. 2020. *Computer Vision : Use machine learning and neural networks to teach computers to see*. Diakses pada 6 September 2020 : <https://www.ibm.com/topics/computer-vision>.
- [12] sas. 2020. *Computer Vision : What it is and why it matters*. Diakses pada 4 Juli 2020 : https://www.sas.com/en_us/insights/analytics/computer-vision.html.
- [13] Redmon, Joseph. Divvala Santosh, Ross Girshick, Ali Farhadi. 2016. *You Only Look Once : Unified Real-Time Object Detection*. arXiv. DOI : arXiv:1506.02640.
- [14] Redmon, Joseph, Ali Farhadi. *YOLO9000 : Better, Faster, Stronger*. arXiv. DOI : arXiv. 1612.08242
- [15] Redmon, Joseph, Ali Farhadi. *YOLOv3 : An Incremental Improvement*. arXiv. DOI : arXiv.1804.02767.
- [16] J.Redmon. Darknet. 2016. "Open source neural networks inc". Diakses pada tanggal 29 September 2020 : <http://pjreddie.com/darknet/>.
- [17] Balsys, Rokas. 2020. "Introduction to YOLOv4 object detection". Diakses pada tanggal 18 Desember 2020 : <https://pylessons.com/YOLOv4-TF2-introduction/>.
- [18] Bochkovskiy, Alexey, Chien Yao Wang, Hong Yuan Mark Liao. *YOLOv4 : Optimal Speed and Accuracy of Object Detection*. Computer Vision and Pattern Recognition (cs.CV). DOI : arXiv:2004.10934.
- [19] Shung, Koo Ping. 2018. *Accuracy, Precision, Recall or F1 ?* . Diakses pada tanggal 18 Desember 2020 : <https://towardsdatascience.com/accuracy-precision-recall-or-f1-331fb37c5cb9>.

- [20] 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/>.
- [21] Yohanandan, Shivy. 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>.
- [22] AltexSoft. Juni 2017. *Preparing Your Dataset for Machine Learning: 8 Basic Techniques That Make Your Data Better*. AltexSoft. Diakses pada tanggal 6 September 2020 : <https://www.altexsoft.com/blog/datascience/preparing-your-dataset-for-machine-learning-8-basic-techniques-that-make-your-data-better/>.
- [23] 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>.
- [24] Fell, Julia. 2017. *How to Choose a Machine Vision Camera*. BNP Media : Quality Magazine, December 2020 Editon. Diakses pada tanggal 7 September 2020 : <https://www.qualitymag.com/articles/93861-how-to-choose-a-machine-vision-camera>.
- [25] Components101. 2017. *Active Passive Buzzer*. Components101.com. Diakses pada tanggal 7 September 2020. <https://components101.com/buzzer-pinout-working-datasheet>.
- [26] The AI Guy, 2020. *YOLOv4 in the Cloud:Build and Train Custom Object Detector (Free GPU)*. <https://www.youtube.com/watch?v=mmj3nxGT2YQ>. Diakses pada tanggal 15 Desember 2020.

Hasil pengujian jarak efektif deteksi pistol

No	Jarak	Data Uji (Resolusi 640x480)	Gambar Hasil (Resolusi 416x416)	Status deteksi	Skor deteksi
1	50 cm			TP	0.77
2	100 cm			TP	0.35
3	150 cm			FN	-
4	200 cm			FN	-
5	250 cm			FN	-

Lampiran 1 : Data pengujian Jarak Efektif deteksi

6	300 cm			total objek detected: 1	TP	0.50
7	350 cm			total objek detected: 1	TP	0.93
8	400 cm				FN	-
9	450 cm				FN	-
10	500 cm				FN	-
11	550 cm				FN	-

Lampiran 1 : Data pengujian Jarak Efektif deteksi

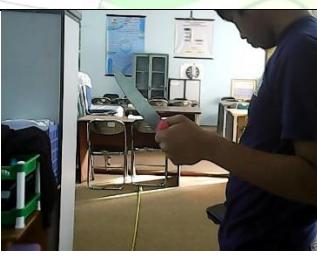
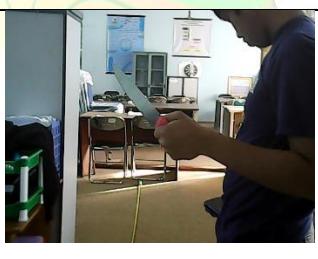
12	600 cm				FN	-
13	650 cm				TP	0.34
14	700 cm				FN	-
15	750 cm				FN	-
16	800 cm				FN	-
17	850 cm				FN	-

Lampiran 1 : Data pengujian Jarak Efektif deteksi

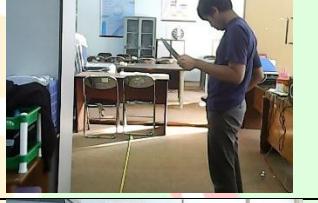
18	900 cm			TP	0.57
19	950 cm			FN	-
20	1000 cm			FN	-

*TP : True Positive, FP : False Positive, FN : False Negative

Hasil pengujian jarak efektif deteksi pisau

No	Jarak	Data Uji (Resolusi 640x480)	Gambar Hasil (Resolusi 416x416)	Status deteksi	Skor deteksi
1	50 cm			FN	-
2	100 cm			FN	-

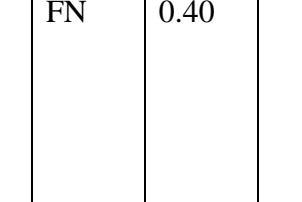
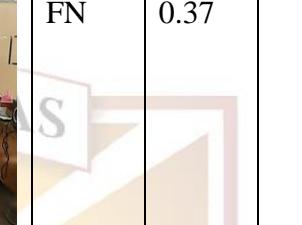
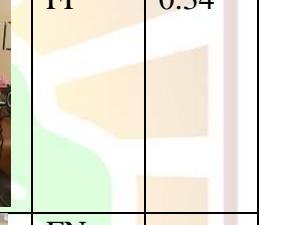
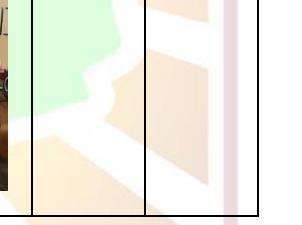
Lampiran 1 : Data pengujian Jarak Efektif deteksi

3	150 cm				FN	-
4	200 cm				FN	-
5	250 cm				FN	-
6	300 cm				FN	-
7	350 cm				FN	-
8	400 cm				FN	-
9	450 cm				FN	-

Lampiran 1 : Data pengujian Jarak Efektif deteksi

10	500 cm					FP	0.50
11	550 cm					FN	-
12	600 cm					FN	-
13	650 cm					FN	-
14	700 cm					FN	-
15	750 cm					FN	-
16	800 cm					FN	-

Lampiran 1 : Data pengujian Jarak Efektif deteksi

17	850 cm				FN	0.40
18	900 cm				FN	0.37
19	950 cm				FP	0.34
20	100-0 cm				FN	-

*TP : True Positive, FP : False Positive, FN : False Negative



1. Skenario 1 : situasi normal

Pada skenario ini, dilakukan percobaan dengan menggunakan inputan video dari webcam pada kondisi dimana tidak ada objek senjata pada ruangan, kemudian diuji kemampuan sistem dalam melakukan deteksi. Hasil dari percobaan skenario ini yaitu didapatkan 2 kondisi, dimana saat sistem tidak mendeteksi adanya objek senjata dan saat sistem mengira terdapat adanya objek senjata:

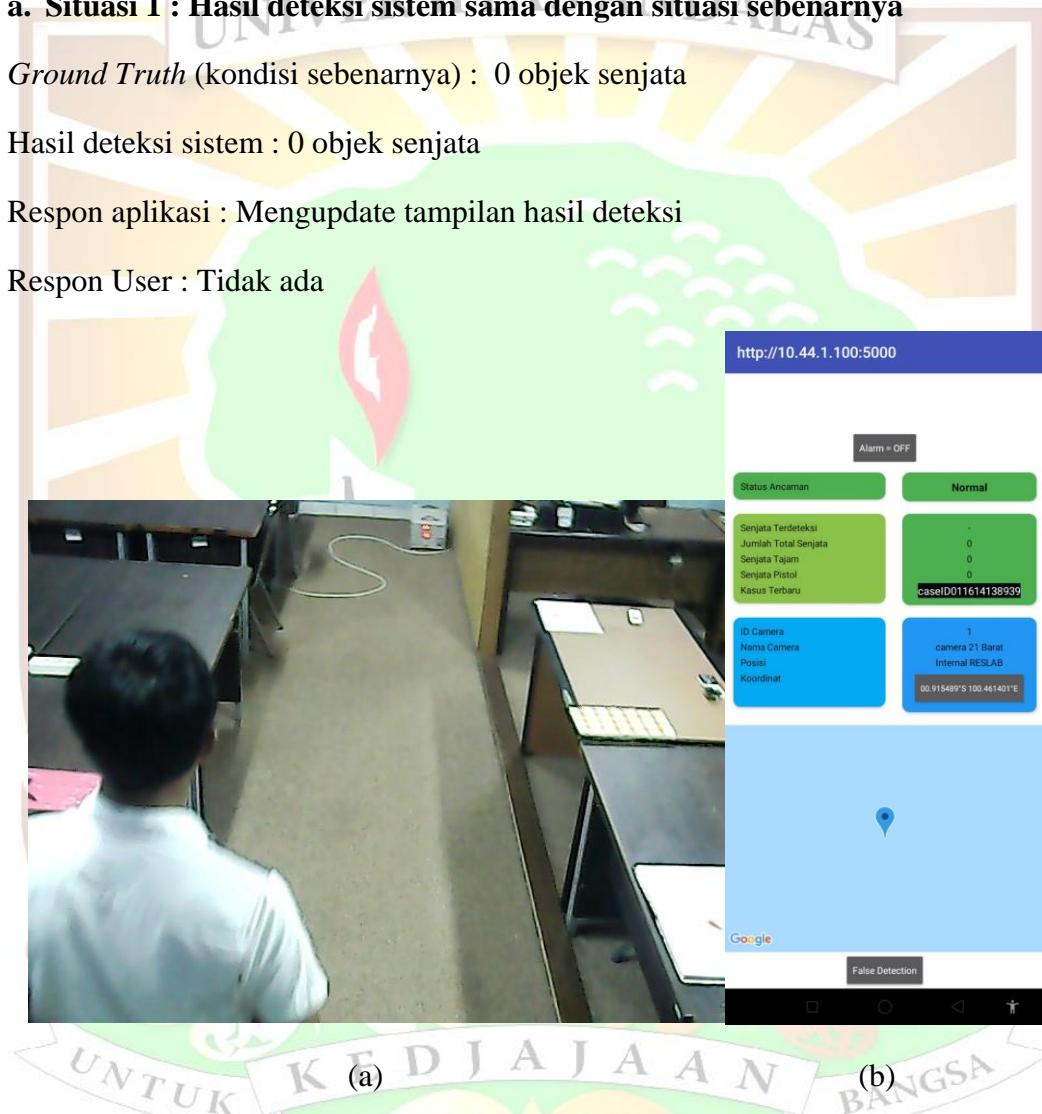
a. Situasi 1 : Hasil deteksi sistem sama dengan situasi sebenarnya

Ground Truth (kondisi sebenarnya) : 0 objek senjata

Hasil deteksi sistem : 0 objek senjata

Respon aplikasi : Mengupdate tampilan hasil deteksi

Respon User : Tidak ada



Gambar 1. (a) Hasil deteksi sistem pada percobaan pertama dalam situasi normal, dan (b) tampilan aplikasi android dari hasil percobaan pertama

b. Situasi 2 : Hasil deteksi sistem sama tidak sesuai dengan situasi sebenarnya

Ground Truth (kondisi sebenarnya) : 0 objek senjata

Hasil deteksi sistem : 1 objek senjata pisau

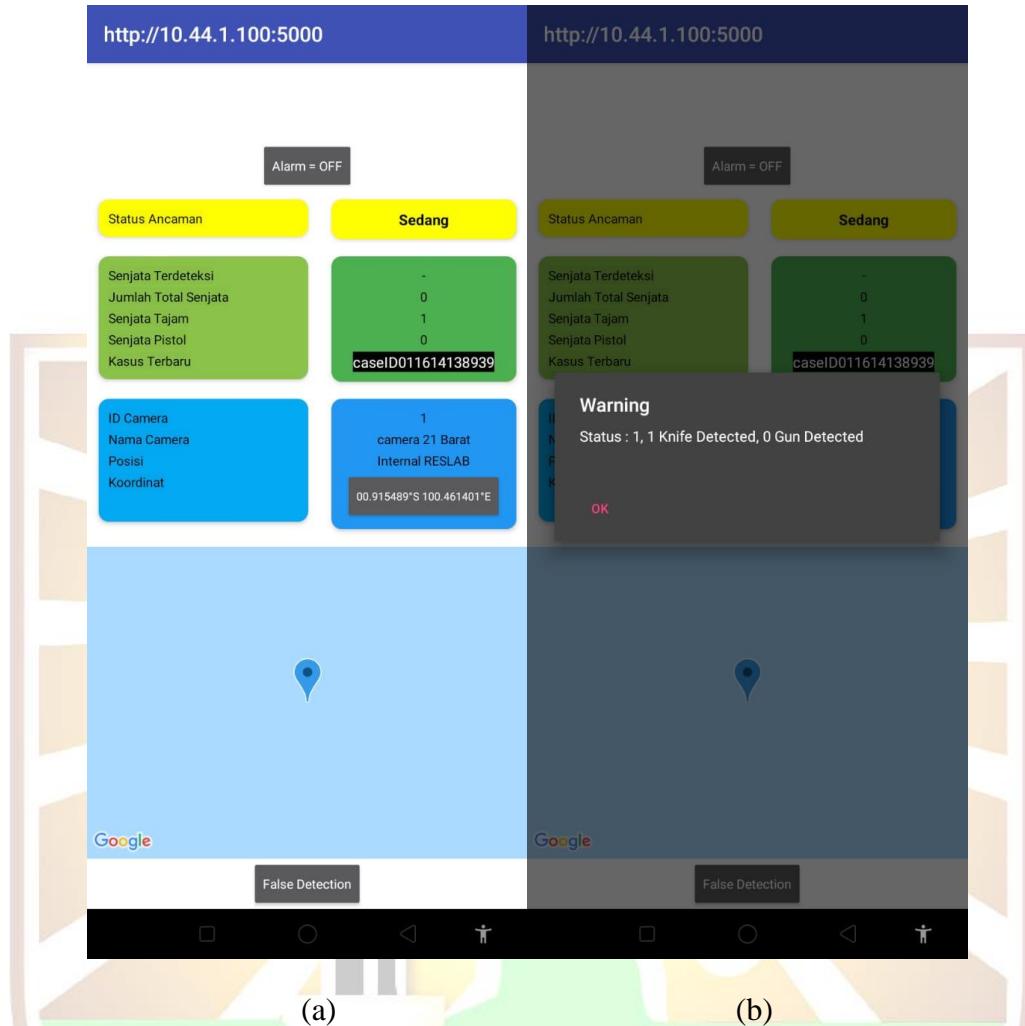
Respon aplikasi : menerima notifikasi dan mengupdate tampilan hasil deteksi

Respon User : melaporkan kasus *false detection* atau kesalahan deteksi



Gambar 2. Hasil deteksi sistem pada percobaan 2 dalam situasi normal

Lampiran 2 : Data pengujian sistem secara keseluruhan



Gambar 3 (a) tampilan aplikasi android pada percobaan kedua dalam situasi normal, (b) notifikasi yang dikirim ke aplikasi android

2. Skenario 2 : 1 Senjata Pisau

Dalam skenario ini, seorang oknum membawa senjata tajam ke dalam ruangan tanpa adanya kejadian penyerangan bersenjata, atau dalam kata lain hanya membawa dan menunjukkan senjata tajam. Untuk itu dilakukan deteksi dan didapatkan hasil :

Ground Truth (kondisi sebenarnya) : 1 objek senjata pisau

Hasil deteksi sistem : 1 objek senjata pisau

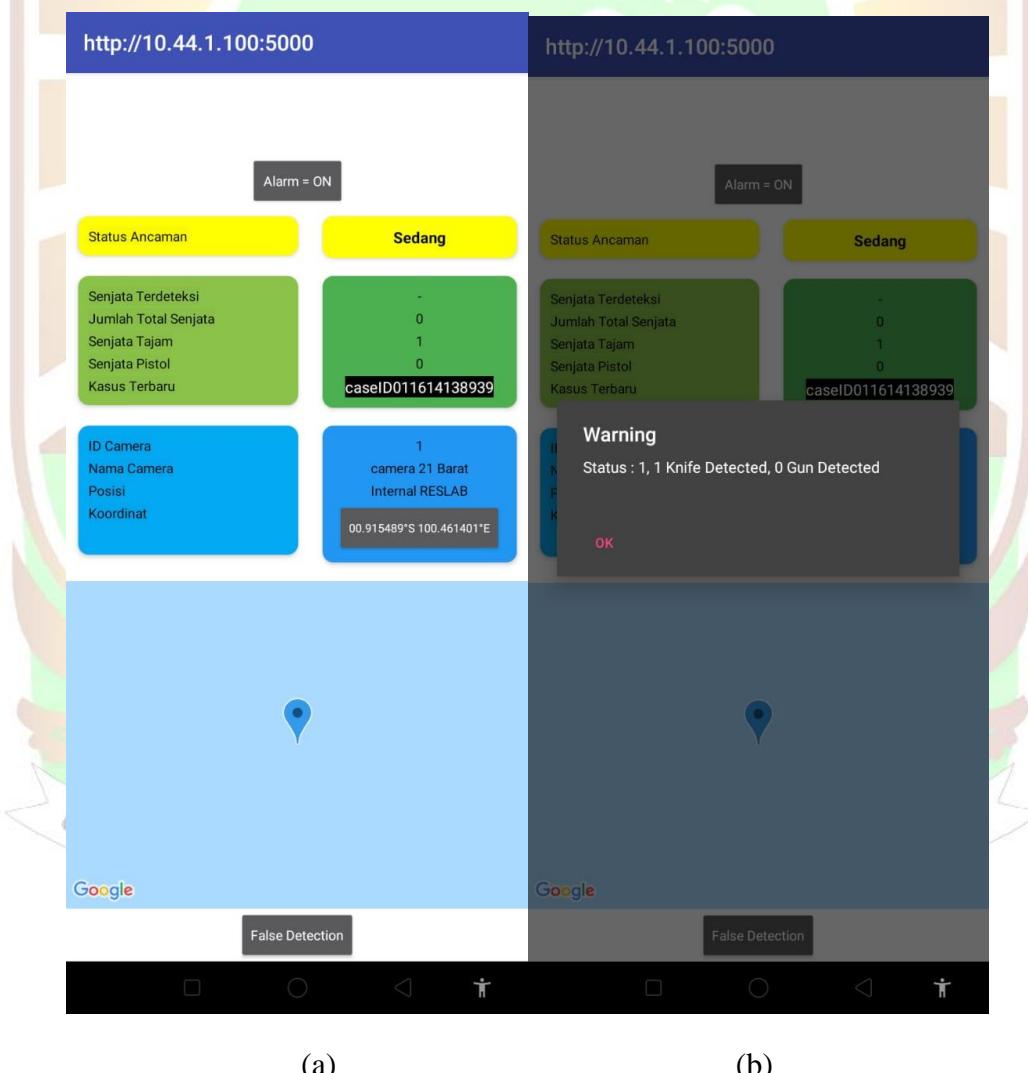
Respon aplikasi : menerima notifikasi dan mengupdate tampilan hasil deteksi

Respon User : mengaktifkan alarm untuk memberikan peringatan kepada pelaku agar mengurungkan niatnya.

Lampiran 2 : Data pengujian sistem secara keseluruhan



Gambar 4. Hasil deteksi sistem pada skenario 2



(a)

(b)

Gambar 5 (a) tampilan aplikasi android pada percobaan skenario 2, (b) notifikasi yang dikirim ke aplikasi android

3. Skenario 3 : Konflik senjata tajam

Dalam skenario ini, terdapat 2 senjata tajam berupa pisau dan 2 orang yang saling menyerang, selanjutnya dilakukan deteksi sistem dan didapatkan hasil :

Ground Truth (kondisi sebenarnya) : 2 objek senjata pisau

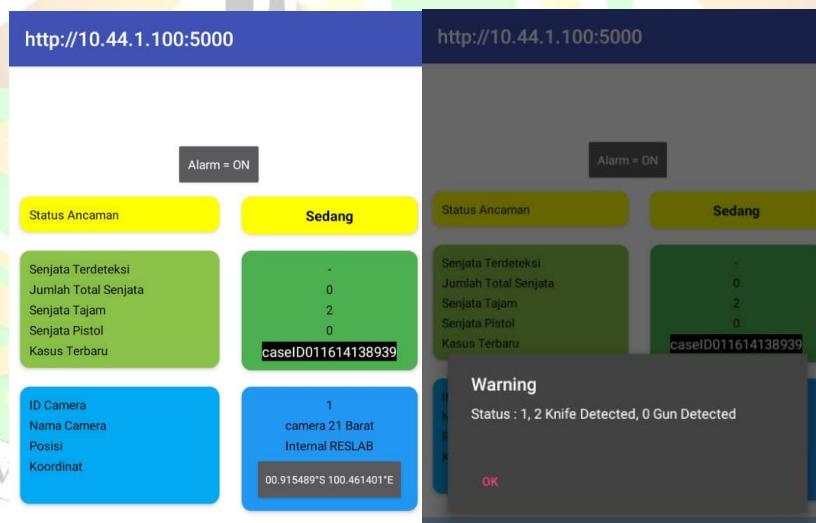
Hasil deteksi sistem : 2 objek senjata pisau

Respon aplikasi : menerima notifikasi dan mengupdate tampilan hasil deteksi

Respon User : mengaktifkan alarm dan melaporkan ke pihak keamanan lokal (satpam)



Gambar 6. Hasil deteksi pada percobaan skenario 3



(a) Application Dashboard:

Alarm = ON	
Status Ancaman	Sedang
Senjata Terdeteksi	-
Jumlah Total Senjata	0
Senjata Tajam	2
Senjata Pistol	0
Kasus Terbaru	caseID011614138939

ID Camera
Nama Camera
Posisi
Koordinat

(b) Warning Message:

Warning
Status : 1, 2 Knife Detected, 0 Gun Detected

OK

00.915489°S 100.461401°E

Gambar 7 (a) tampilan aplikasi android pada percobaan skenario 3, (b) notifikasi yang dikirim ke aplikasi android

4. Skenario 4 : Penyerangan bersenjata Pistol dan Senjata tajam

Pada skenario ini, dicoba lakukan kasus penyerangan bersenjata dimana terdapat 2 pihak yang bertikai. Pada percobaan ini pihak pertama menggunakan senjata pisau dan pihak kedua menggunakan senjata pistol. Hasil deteksi sistem dari percobaan ini yaitu :

Ground Truth (kondisi sebenarnya) : 1 objek senjata pistol dan 1 objek senjata pisau

Hasil deteksi sistem : 1 senjata pistol dan 1 senjata pisau

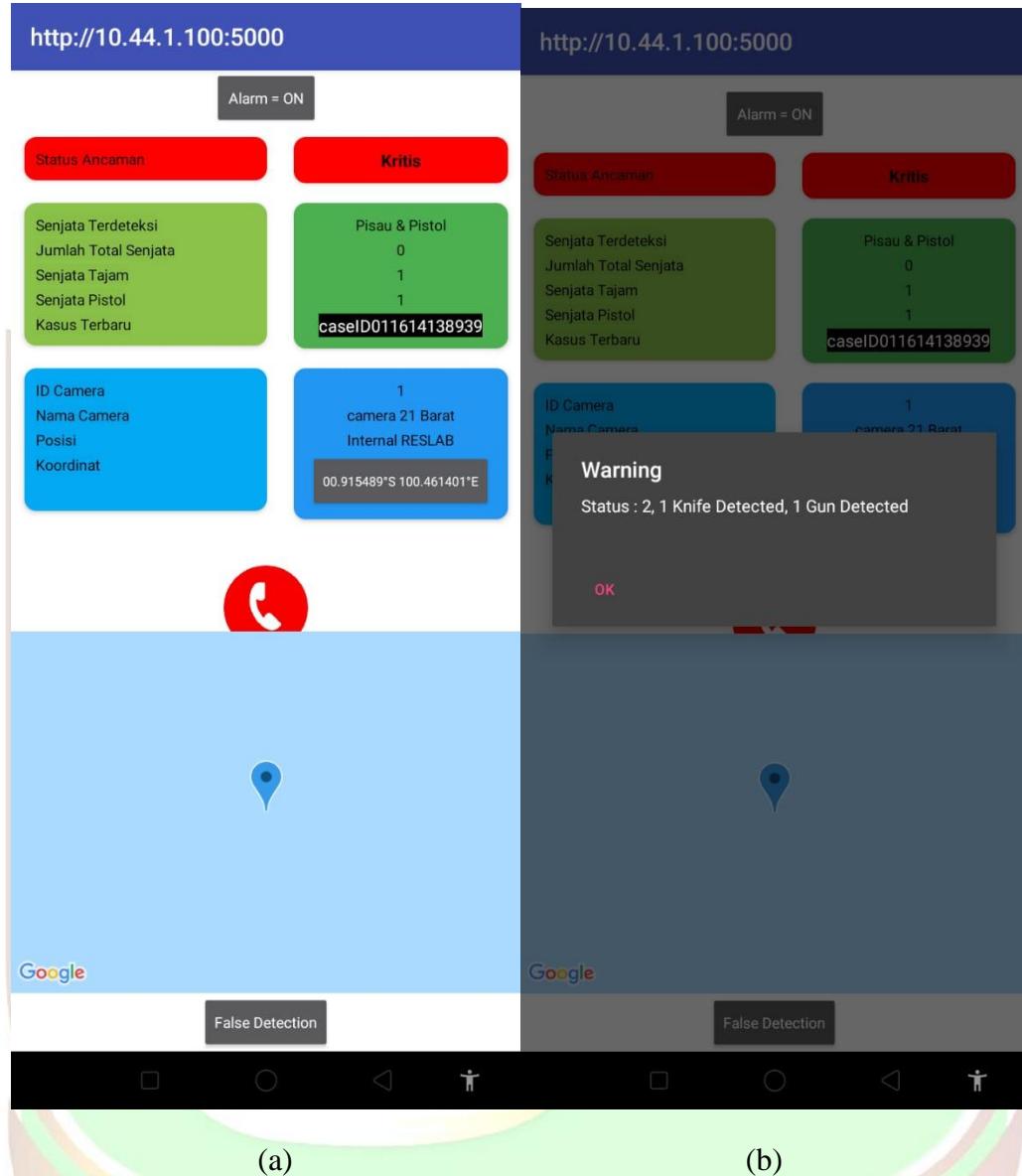
Respon aplikasi : menerima notifikasi dan mengupdate tampilan hasil deteksi

Respon User : mengaktifkan alarm dan melaporkan ke pihak kepolisian melalui telepon



Gambar 8. Hasil deteksi sistem pada percobaan skenario 4

Lampiran 2 : Data pengujian sistem secara keseluruhan



Gambar 9 (a) tampilan aplikasi android pada percobaan skenario 4 (b) notifikasi yang dikirim ke aplikasi android

5. Skenario 5 : Perampukan dan penodongan menggunakan senjata tajam

Dalam skenario ini, dicoba lakukan pengujian pada kondisi ancaman penodongan senjata berupa senjata tajam, dengan senjata yang digunakan yaitu 1 senjata pisau. Hasil yang didapatkan dari deteksi sistem yaitu :

Ground Truth (kondisi sebenarnya) : 1 objek senjata pisau

Hasil deteksi sistem : 2 senjata pisau (1 kesalahan deteksi)

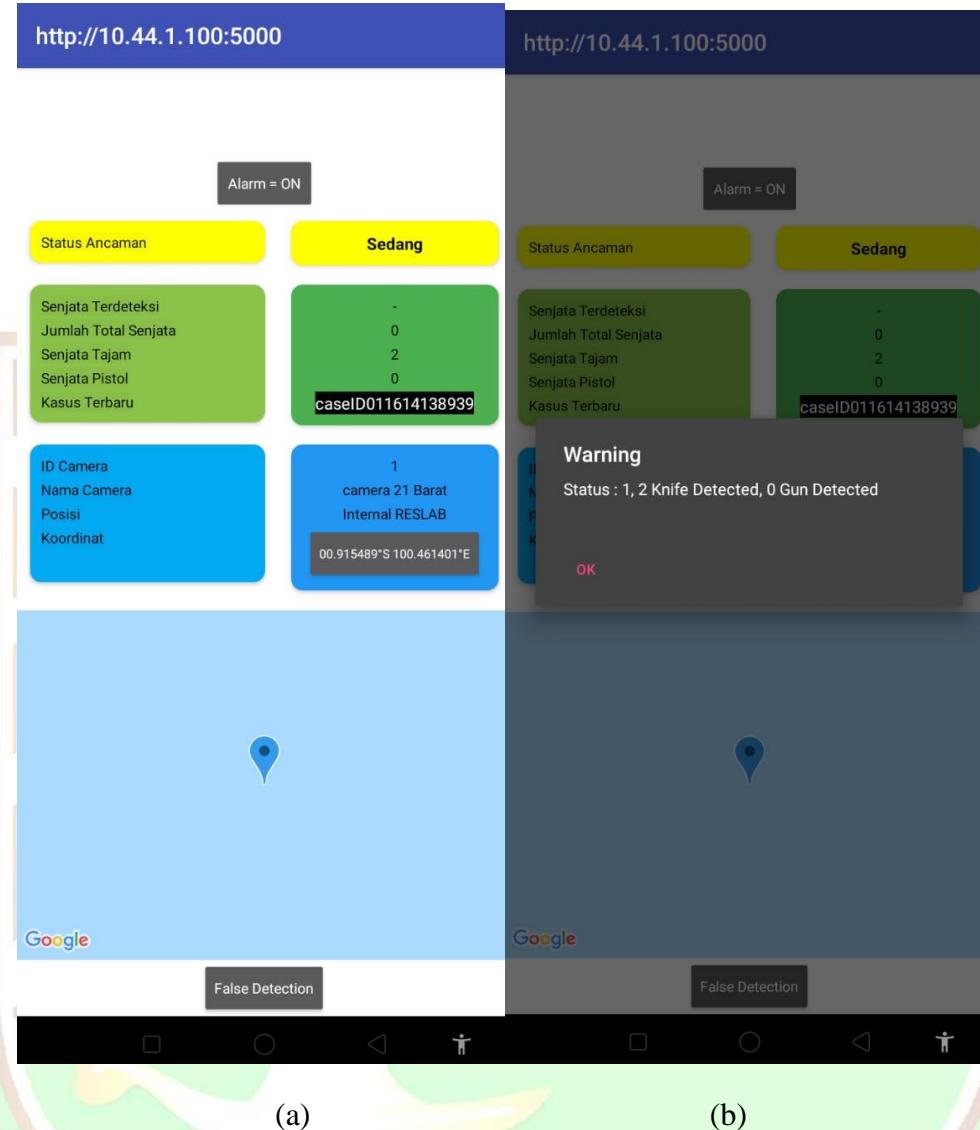
Respon aplikasi : menerima notifikasi dan mengupdate tampilan hasil deteksi

Respon User : mengaktifkan alarm dan melaporkan ke pihak kepolisian melalui telepon



Gambar 10. Hasil deteksi pada percobaan skenario 5

Lampiran 2 : Data pengujian sistem secara keseluruhan



Gambar 11 (a) tampilan aplikasi android pada percobaan skenario 5, (b) notifikasi yang dikirim ke aplikasi android



6. Skenario 6 : Konflik bersenjata

Pada skenario ini, terdapat 2 orang yang saling menodongkan senjata pistol. Saat dilakukan deteksi, didapatkan hasil :

Ground Truth (kondisi sebenarnya) : 2 objek senjata pistol

Hasil deteksi sistem : 2 senjata pistol

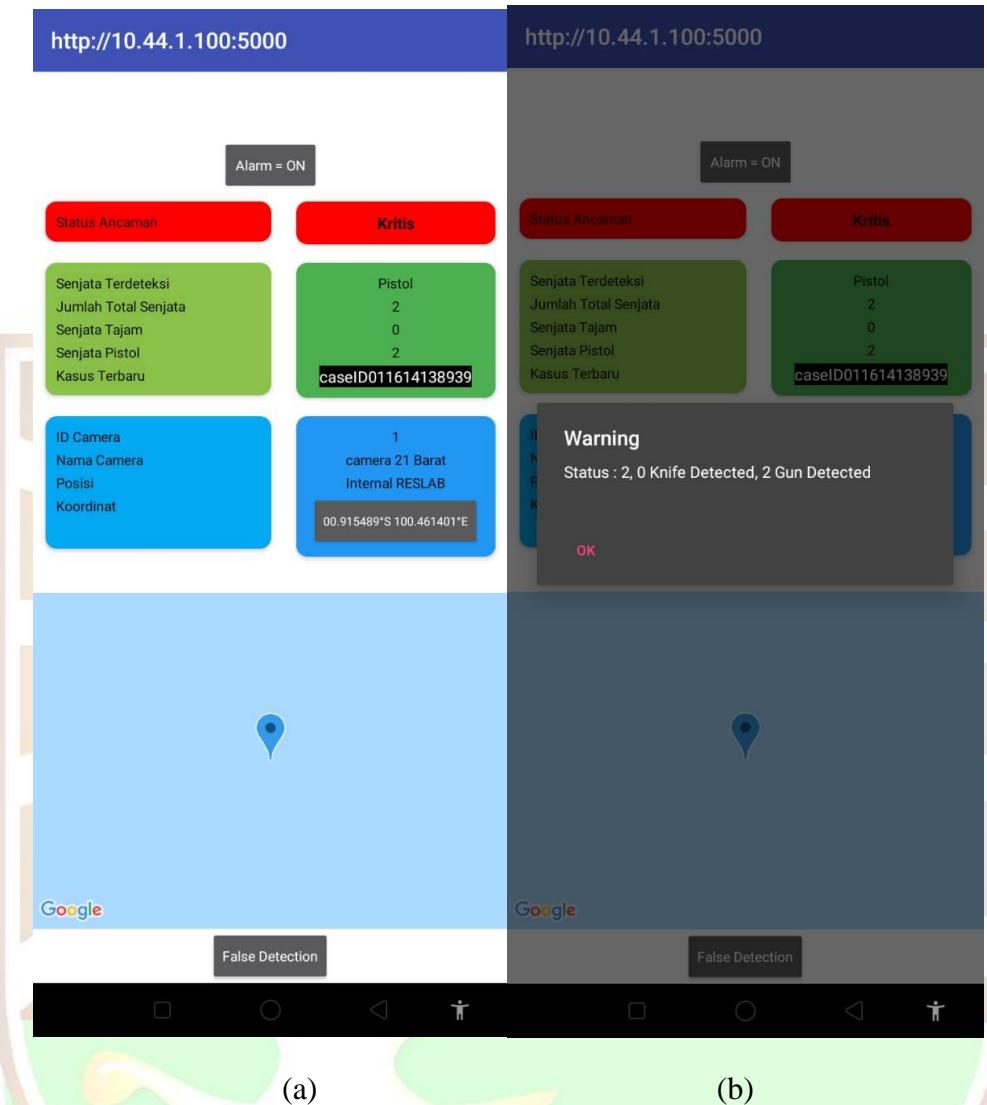
Respon aplikasi : menerima notifikasi dan mengupdate tampilan hasil deteksi

Respon User : melaporkan ke pihak kepolisian melalui telepon dan menjelaskan tingkat ancaman yang tinggi karena pelaku menggunakan pistol.



Gambar 12. Hasil deteksi pada percobaan skenario 6

Lampiran 2 : Data pengujian sistem secara keseluruhan



Gambar 11 (a) tampilan aplikasi android pada percobaan skenario 6, (b) notifikasi yang dikirim ke aplikasi android

7. Skenario 7 : Perampokan dengan senjata tajam dan senjata api

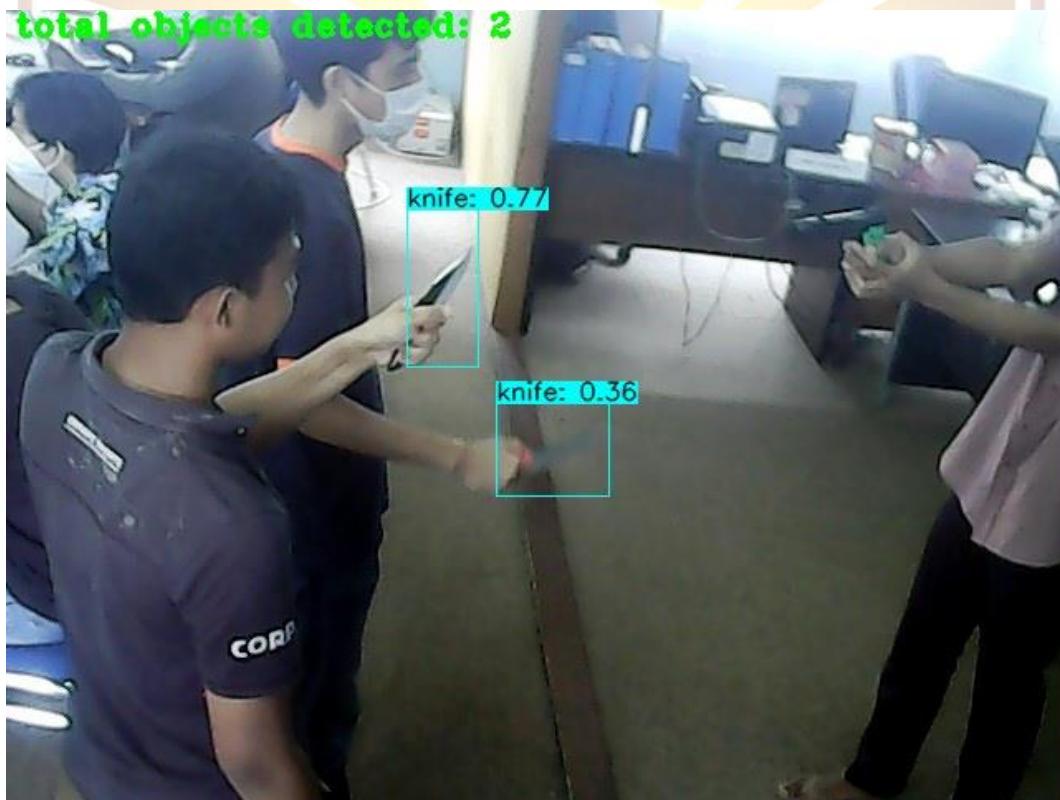
Dalam skenario ini, terdapat 2 senjata pisau dan 1 senjata pistol. Dari hasil deteksi sistem didapatkan hasil sebagai berikut :

Ground Truth (kondisi sebenarnya) : 2 objek senjata pisau dan 1 senjata pistol

Hasil deteksi sistem : 2 senjata pisau (1 objek pistol tidak terdeteksi)

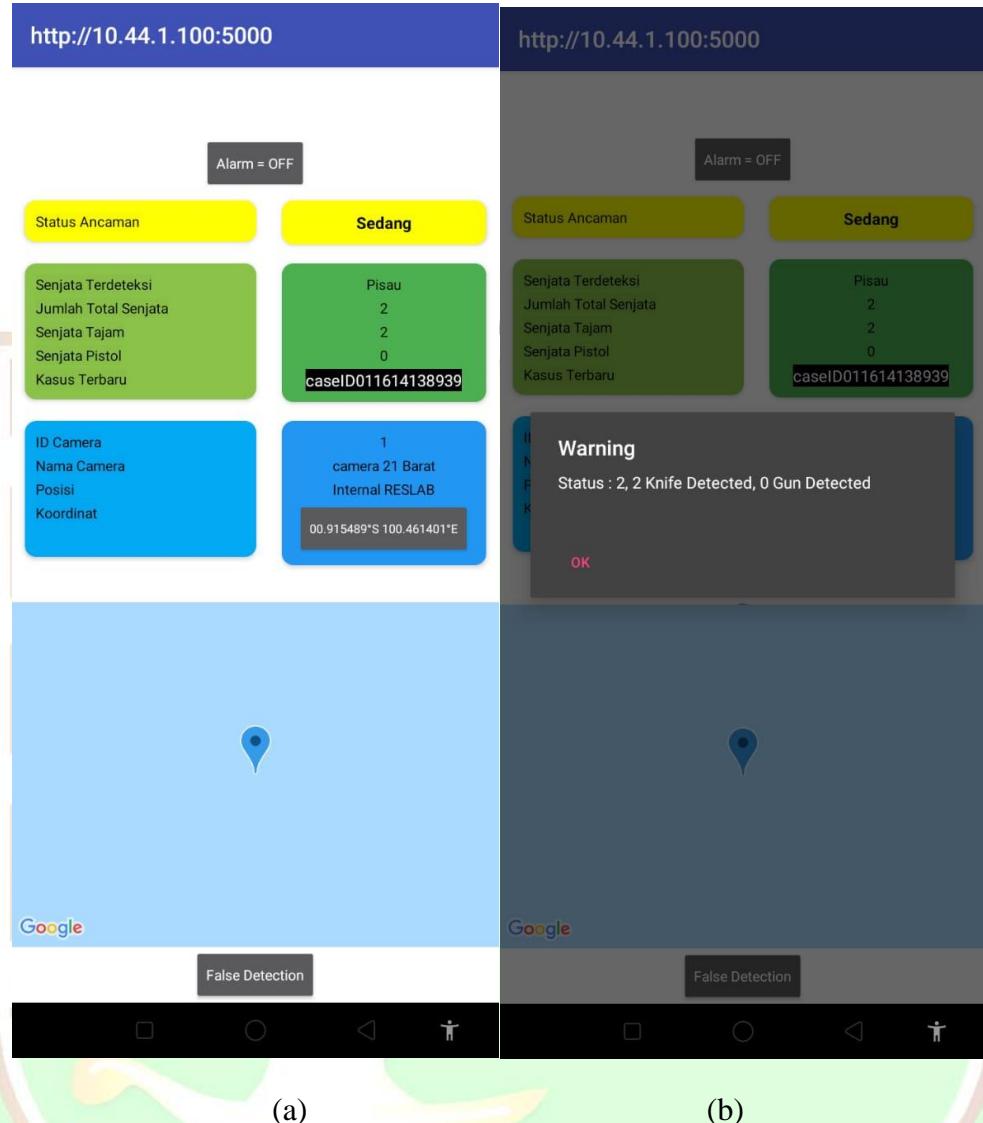
Respon aplikasi : menerima notifikasi dan mengupdate tampilan hasil deteksi

Respon User : melaporkan ke pihak kepolisian melalui telepon.



Gambar 12 Hasil deteksi pada percobaan skenario 7

Lampiran 2 : Data pengujian sistem secara keseluruhan



Gambar 13 (a) tampilan aplikasi android pada percobaan skenario 7, (b) notifikasi yang dikirim ke aplikasi android

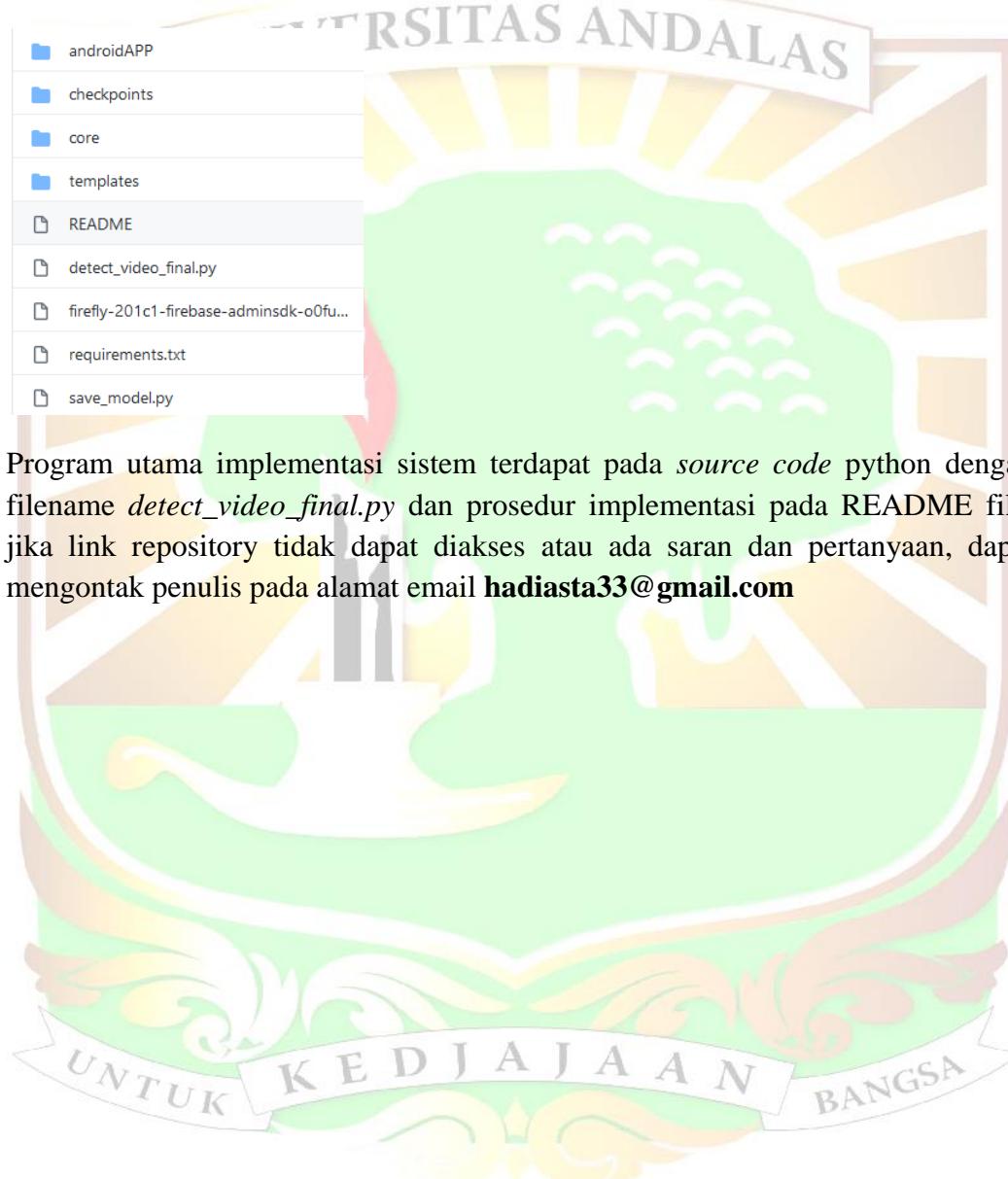


Source Code

Untuk source code dari Tugas Akhir ini dapat diakses pada repository github dengan alamat url :

https://github.com/Ghost51Recon/TugasAkhir_weapon_detection

Struktur file dari repository github dapat dilihat sebagai berikut :



Program utama implementasi sistem terdapat pada *source code* python dengan filename *detect_video_final.py* dan prosedur implementasi pada README file. jika link repository tidak dapat diakses atau ada saran dan pertanyaan, dapat mengontak penulis pada alamat email **hadiasta33@gmail.com**

SURAT PERNYATAAN

Saya yang bertanda tangan dibawah ini :

Nama : Muhammad Abdul Hadi
NIM : 1611512012
Jurusan : S1 Teknik Komputer

Menyatakan bahwa telah melakukan pemeriksaan similarity dengan Turnitin dengan similarity index < 25%.. Hal ini dapat dilihat pada lampiran yang saya sertakan bersama surat ini.
Demikian surat pernyataan ini saya buat agar dapat digunakan sebaik baik nya.

Padang, 19 Maret 2021

Yang bertanda tangan,



Muhammad Abdul Hadi

1611512012

TA

by Muhammad Abdul Hadi

Submission date: 10-Mar-2021 12:05AM (UTC-0800)

Submission ID: 1529161579

File name: Muhammad_Abdul_Hadi-2nd_TA_Draft_after_semhas_01032021_final.pdf (6.65M)

Word count: 17684

Character count: 114718



PRIMARY SOURCES

1	Submitted to Universitas Andalas Student Paper	2%
2	repo.unand.ac.id Internet Source	1%
3	scholar.unand.ac.id Internet Source	1%
4	repository.its.ac.id Internet Source	1%
5	www.slideshare.net Internet Source	1%
6	github.com Internet Source	<1%
7	belajarmikrokontroler-2018.blogspot.com Internet Source	<1%
8	123dok.com Internet Source	<1%
9	Submitted to Universitas Brawijaya Student Paper	<1%

10	repository.usd.ac.id Internet Source	<1 %
11	repository.uin-suska.ac.id Internet Source	<1 %
12	Submitted to Universitas Negeri Jakarta Student Paper	<1 %
13	es.scribd.com Internet Source	<1 %
14	www.mdpi.com Internet Source	<1 %
15	eprints.umm.ac.id Internet Source	<1 %
16	medium.com Internet Source	<1 %
17	Submitted to Universitas Muhammadiyah Surakarta Student Paper	<1 %
18	Submitted to University of Wolverhampton Student Paper	<1 %
19	www.spiedigitallibrary.org Internet Source	<1 %
20	Submitted to University of Teesside Student Paper	<1 %
	repository.radenintan.ac.id	

21	Internet Source	<1 %
22	Lathifah Arief, Alif Ziden Tantowi, Nefy Puteri Novani, Tri A. Sundara. "Implementation of YOLO and Smoke Sensor for Automating Public Service Announcement of Cigarette's Hazard in Public Facilities", 2020 International Conference on Information Technology Systems and Innovation (ICITSI), 2020 Publication	<1 %
23	docplayer.info Internet Source	<1 %
24	dspace.uii.ac.id Internet Source	<1 %
25	text-id.123dok.com Internet Source	<1 %
26	webthesis.biblio.polito.it Internet Source	<1 %
27	Submitted to Universitas Putera Batam Student Paper	<1 %
28	addi.ehu.es Internet Source	<1 %
29	en.wikipedia.org Internet Source	<1 %

30	id.123dok.com	<1 %
Internet Source		
31	repositori.umsu.ac.id	<1 %
Internet Source		
32	repository.unika.ac.id	<1 %
Internet Source		
33	sleeplab.com.br	<1 %
Internet Source		
34	cloudit-eg.com	<1 %
Internet Source		
35	eprints.undip.ac.id	<1 %
Internet Source		
36	Submitted to US Navy Naval War College	<1 %
Student Paper		
37	Submitted to North East Surrey College of Technology, Surrey	<1 %
Student Paper		
38	Submitted to University of West London	<1 %
Student Paper		
39	id.scribd.com	<1 %
Internet Source		
40	repository.unhas.ac.id	<1 %
Internet Source		

41	www.ncbi.nlm.nih.gov	<1 %
42	Submitted to Fakultas Ekonomi Universitas Indonesia	<1 %
	Student Paper	
43	Submitted to UIN Sunan Kalijaga Yogyakarta	<1 %
	Student Paper	
44	eresearch.stikom-bali.ac.id	<1 %
	Internet Source	
45	mafiadoc.com	<1 %
	Internet Source	
46	Submitted to Sriwijaya University	<1 %
	Student Paper	
47	Submitted to Universitas Muria Kudus	<1 %
	Student Paper	
48	anzdoc.com	<1 %
	Internet Source	
49	Submitted to University of Hong Kong	<1 %
	Student Paper	
50	Submitted to University of Missouri, Kansas City	<1 %
	Student Paper	
51	repository.dinamika.ac.id	<1 %
	Internet Source	

52	sistemasi.ftik.unisi.ac.id	<1 %	
53	Internet Source	www.pps.unud.ac.id	<1 %
54	doku.pub	<1 %	
55	Internet Source	dokumen.pub	<1 %
56	id.sciencewal.com	<1 %	
57	Internet Source	idoc.pub	<1 %
58	repo.darmajaya.ac.id	<1 %	
59	Internet Source	repozitorij.etfos.hr	<1 %
60	contohmakalah4.blogspot.com	<1 %	
61	Internet Source	digilib.uinsby.ac.id	<1 %
62	jitce.fti.unand.ac.id	<1 %	
63	Internet Source	jom.unpak.ac.id	<1 %

		<1 %
64	library.binus.ac.id Internet Source	<1 %
65	repository.uksw.edu Internet Source	<1 %
66	repozitorij.unin.hr Internet Source	<1 %
67	tbevidence.org Internet Source	<1 %
68	www.altexsoft.com Internet Source	<1 %
69	www.scribd.com Internet Source	<1 %
70	Submitted to Australian Institute of Business Student Paper	<1 %
71	Muhammad Hidayatullah, Jauharul Fat, Titi Andriani. "Prototype Sistem Telemetri Pemantauan Kualitas Air Pada Kolam Ikan Air Tawar Berbasis Mikrokontroler", POSITRON, 2018 Publication	<1 %
72	Muhammad Tahir Bhatti, Muhammad Gufran Khan, Masood Aslam, Muhammad Junaid Fiaz.	<1 %

"Weapon Detection in Real-Time CCTV Videos using Deep Learning", IEEE Access, 2021

Publication

-
- 73 Rahmat Novrianda Dasmen, Rasmila .. **<1 %**
"Implementasi Raspberry Pi 3 pada Sistem Pengontrol Lampu berbasis Raspbian Jessie", Jurnal Edukasi dan Penelitian Informatika (JEPIN), 2019
Publication
-
- 74 Windra Swastika, Albert Wahyudi Nur, Oesman Hendra Kelana. "Monitoring Ruangan Untuk Deteksi Manusia Berbasis CNN Dengan Fitur Push Notification", Teknika, 2019 **<1 %**
Publication
-
- 75 dspace.cc.tut.fi **<1 %**
Internet Source
-
- 76 moam.info **<1 %**
Internet Source
-
- 77 pt.scribd.com **<1 %**
Internet Source
-
- 78 tel.archives-ouvertes.fr **<1 %**
Internet Source
-
- 79 widuri.raharja.info **<1 %**
Internet Source
-
- 80 www.eetree.cn **<1 %**
Internet Source

81

www.electronicscomp.com

Internet Source

<1 %

82

Basworo Ardi Pramono, Aria Hendrawan, April Firman Daru. "RASPBERRY PI DENGAN MODUL KAMERA DAN MOTION SENSOR SEBAGAI SOLUSI CCTV LAB FTIK UNIV. SEMARANG", Jurnal Pengembangan Rekayasa dan Teknologi, 2019

Publication

<1 %

83

docobook.com

Internet Source

<1 %

Exclude quotes

Off

Exclude matches

Off

Exclude bibliography

Off

DAFTAR RIWAYAT HIDUP

(CURICULUM VITAE)



DATA PRIBADI

NAMA : MUHAMMAD ABDUL HADI
NAMA PANGGILAN : Hadi
ALAMAT : AMAN, JORONG KOTOHILALANG,
KENAGARIAN LAMBAH, KECAMATAN
AMPEK ANGKEK, KABUPATEN AGAM,
PROVINSI SUMATERA BARAT.
KODE POST : 25161
NOMOR HANDPHONE : +6282288152033
EMAIL : hadiasta33@gmail.com
JENIS KELAMIN : LAKI-LAKI
TEMPAT TANGGAL LAHIR : BUKITTINGGI / 02 SEPTEMBER 1998
WARGA NEGARA : INDONESIA
AGAMA : ISLAM
GOLONGAN DARAH : B

RIWAYAT PENDIDIKAN

Periode			Sekolah / Institusi / Universitas
2009	-	2014	SDN 29 KOTOHILALANG
2012	-	2014	MTsN IV ANGKAT CANDUNG
2014	-	2016	MAN 1 BUKITTINGGI
2016	-	2021	JURUSAN TEKNIK KOMPUTER, FAKULTAS TEKNOLOGI INFORMASI, UNIVERSITAS ANDALAS

PENGALAMAN BERORGANISASI DAN PEKERJAAN

- Anggota Himpunan Mahasiswa Sistem Komputer 2017-2020
- Koordinator Sekretariat UKMF FSI Al-Fatih 2017-2018
- Koordinator dan Asisten Laboratorium Robotic and Embedded System Laboratory (RESLab) Periode kepengurusan 2019-2020

PENGALAMAN SEMINAR, PELATIHAN , DAN PRESTASI

- Open Recruitment UKM Neotelemetri UNAND tahun 2017
- Open Recruitment UKM FKI Rabbani UNAND tahun 2017
- Panitia dan Koordinator Konsumsi Musyawarah Besar FKI Rabbani tahun 2018
- Latihan Keterampilan Manajemen Mahasiswa Tingkat Dasar (LKMMTD) Fakultas Teknologi Informasi UNAND tahun 2018
- Asisten Magang Laboratorium Robotic and Embedded System Laboratory (RESLab) tahun 2018-2019
- Finalis Lomba Simulasi Jaringan Nasional (CCNA&CCNP) Universitas Sriwijaya tahun 2018
- Peserta Program Student Mobility UNAND ke Universitas Ibaraki, Jepang tahun 2019

KEMAMPUAN BAHASA DAN KOMPUTER

Bahasa :

- Bahasa Indonesia
- Bahasa Inggris

Komputer

- Microsoft Windows & Applications (Excel, Word, Power Point)
- Analisis Perangkat Jaringan
- Python Programming
- Video Editing
- Embedded System Design



Muhammad Abdul Hadi S.T.