

**Pengenalan Objek untuk Pembelajaran Anak-Anak
Menggunakan Algoritma YOLO**

SKRIPSI

Diajukan untuk memenuhi salah satu syarat
memperoleh Gelar Sarjana Komputer
Program Studi Informatika



Disusun oleh:

Gabriel Advent Batan

NIM: 205314096

**FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS SANATA DHARMA
YOGYAKARTA**

2024

HALAMAN PERSETUJUAN PEMBIMBING

SKRIPSI

**Pengenalan Objek untuk Pembelajaran Anak-Anak
Menggunakan Algoritma YOLO**

Disusun oleh:

Gabriel Advent Batan

NIM: 205314096

Dosen Pembimbing,

(Drs. Hari Suparwito, S.J., M.App.IT)

(tanggal persetujuan)

ABSTRAK

Pentingnya periode usia dini sebagai waktu peka anak terhadap rangsangan menjadi dasar bagi pendekatan inovatif dalam pembelajaran. Perkembangan teknologi telah memungkinkan pembelajaran interaktif, namun pemanfaatannya pada usia dini masih kurang optimal. Penelitian ini mengaitkan konsep pengenalan objek dalam *computer vision*, khususnya melalui algoritma *You Only Look Once* (YOLO), dengan konteks pembelajaran anak usia dini. YOLO, sebagai pendekatan integratif deteksi dan klasifikasi objek, telah menjadi populer dalam pengenalan objek *real-time*. Penelitian ini mencoba mengoptimalkan teknologi *computer vision*, khususnya dengan implementasi YOLO, dalam pemahaman lingkungan sekitar untuk pembelajaran anak usia dini. Penelitian ini melibatkan beberapa parameter-parameter yang digunakan dalam YOLO. Dari 32 percobaan berdasarkan kombinasi parameter, diketahui bahwa percobaan ketujuh mampu menghasilkan mAP yang lebih tinggi dari percobaan-percobaan yang lain. Percobaan ketujuh ini mampu menghasilkan mAP sebesar 0,87916 atau 87%. Selanjutnya dengan bantuan Streamlit, akan dibuat *website* untuk diujicobakan dan disebarkan bersamaan dengan kuesioner yang sudah diuji dengan uji validitas dan reliabilitas. Dari 27 responden kuesioner ini akan dihitung berdasarkan aspek *usability* dan mendapatkan hasil sangat layak. Hal ini mampu membuktikan bahwa pengimplementasian YOLO ke dalam *website* dapat membantu tumbuh kembang anak.

Kata kunci: YOLO, *object detection*, *computer vision*, usia dini, pendidikan

ABSTRACT

The importance of the early years as a time when children are sensitive to stimuli is the basis for innovative approaches to learning. Technological developments have enabled interactive learning, but its utilization in early childhood is still less than optimal. This research links the concept of object recognition in computer vision, particularly through the You Only Look Once (YOLO) algorithm, to the context of early childhood learning. YOLO, as an integrative approach of object detection and classification, has become popular in real-time object recognition. This research tries to optimize computer vision technology, especially with the implementation of YOLO, in understanding the surrounding environment for early childhood learning. This research involves several parameters used in YOLO. Of the 32 experiments based on the combination of parameters, it is known that the seventh experiment was able to produce a higher mAP than the other experiments. This seventh experiment was able to produce a mAP of 0.87916 or 87%. Furthermore, with the help of Streamlit, a website will be created to be tested. Later, it will be distributed along with a questionnaire that has been tested with validity and reliability tests. From 27 respondents, the answers of the questionnaire will be calculated based on usability aspects and get very feasible results. This can prove that implementing YOLO into the website can help children's growth and development.

Keywords: YOLO, object detection, computer vision, early childhood, education.

DAFTAR ISI

HALAMAN JUDUL	Kesalahan! Bookmark tidak ditentukan.
HALAMAN PERSETUJUAN PEMBIMBING	ii
ABSTRAK	iii
ABSTRACT	iv
DAFTAR ISI.....	v
DAFTAR GAMBAR.....	ix
BAB I:.....	1
PENDAHULUAN.....	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah.....	3
1.3. Batasan Masalah	3
1.4. Tujuan Penelitian	3
1.5. Manfaat Penelitian	4
1.6. Sistematika Penulisan	4
BAB II:	6
TINJAUAN PUSTAKA	6
2.1. Review Literature.....	6
2.2. Landasan Teori	12
2.2.1. Pendidikan Anak Usia Dini	12
2.2.2. Image Processing.....	13
2.2.3. Augmentasi	15
2.2.4. You Only Look Once (YOLO)	15
2.2.5. Roboflow	19
2.2.6. Labeling Gambar	19
2.2.7. Confusion Matrix	19

2.2.8. Mean Average Precision	20
2.2.9. Skala Likert	20
2.2.10. Uji Validitas	21
2.2.11. Uji Reliabilitas	22
2.2.12. Uji Kegunaan (Usability)	23
BAB III	24
METODE PENELITIAN	24
3.	24
3.1. Gambaran Umum Penelitian	24
3.2. Data	25
3.3. Pre-Processing	25
3.3.1. Labeling Gambar	26
3.3.2. Resize Image	26
3.3.3. Augmentasi	27
3.3.4. Data Split	27
3.4. Mengimport Dataset Ke Proyek	28
3.5. Modeling	28
3.5.1. Mendapatkan Dependency Yolo Versi 8	28
3.5.2. Pelatihan Model	29
3.6. Evaluate	29
3.7. Skenario Pengujian	29
3.8. Pengujian Kelayakan Kuesioner dan Kegunaan Hasil Akhir	30
3.8.1. Waktu dan Tempat Pengujian	30
3.8.2. Uji Kelayakan Kuesioner	31
BAB IV	34
HASIL PENELITIAN DAN PEMBAHASAN	34

4.1.	Pengumpulan Data.....	34
4.2.	<i>Pre-processing</i> Data.....	34
4.2.1.	Labeling Gambar	35
4.2.2.	<i>Rescale / Resize Image</i>	36
4.2.3.	<i>Augmentasi Image</i>	36
4.2.4.	<i>Data Split</i>	39
4.3.	Impor Dataset	40
4.4.	Modeling	41
4.4.1.	Mendapatkan Dependency YOLO	42
4.4.2.	Pelatihan Model.....	42
4.5.	Analisis Hasil Pengujian	45
4.6.	Implementasi Aplikasi Deteksi Objek.....	50
4.6.1.	<i>Interface Aplikasi</i>	50
4.6.2.	Implementasi <i>Code</i> untuk Deteksi	53
4.7.	Pengujian Kegunaan Aplikasi.....	58
4.7.1.	Uji Validitas	58
4.7.2.	Uji Reliabilitas	60
4.7.3.	Uji <i>Usability</i>	61
BAB V.....		64
PENUTUP.....		64
5.1.	Kesimpulan	64
5.2.	Saran	64
DAFTAR PUSTAKA		66

DAFTAR TABEL

Tabel 2. 1: Tabel review literature	7
Tabel 3. 1: Tabel Skenario Pengujian	30
Tabel 3. 2: Rincian Pernyataan Kuesioner	31
Tabel 3. 3: Ketentuan skala likert	32
Tabel 3. 4: Tingkat reliabilitas Cronbach's Alpha.....	33
Tabel 3. 5: Kategori Kegunaan	33
Tabel 4. 1: Daftar kombinasi parameter.....	43
Tabel 4. 2: Hasil pelatihan model	45
Tabel 4. 3: Kombinasi yang menghasilkan mAP terbaik.....	46
Tabel 4. 4: Hasil kuesioner pengujian.....	58
Tabel 4. 5: Hasil perhitungan rHitung	59
Tabel 4. 6: Hasil uji validitas	60
Tabel 4. 7: Rincian responden.....	61
Tabel 4. 8: Tabel hasil kuesioner	62
Tabel 4. 9: Tabel hasil perhitungan usability	63

DAFTAR GAMBAR

Gambar 2. 1: Ilustrasi proses digitalisasi	14
Gambar 2. 2: Ilustrasi proses augmentasi	15
Gambar 2. 3: Ilustrasi proses deteksi YOLO	17
Gambar 2. 4: Ilustrasi nilai yang merepresentasikan bounding box	17
Gambar 2. 5: Arsitektur YOLO v8	18
Gambar 3. 3: Flowchart penelitian	24
Gambar 3. 2: Flowchart pre-processing	24
Gambar 3. 1: Flowchart Modelling	24
Gambar 3. 4: Gambar kumpulan dataset sesuai kelas	25
Gambar 3. 5: Contoh gambar yang telah dilakukan labeling	26
Gambar 3. 6: Ilustrasi nilai bounding box	26
Gambar 3. 7: Contoh hasil rescale	27
Gambar 3. 8: Contoh augmentasi yang diterapkan pada citra	27
Gambar 3. 9: Data split dari Roboflow	28
Gambar 3. 10: Contoh confusion matrix	29
Gambar 4. 1: Proses upload data ke Roboflow	34
Gambar 4. 2: Proses annotate di Roboflow	35
Gambar 4. 3: Data hasil labeling atau bounding box	35
Gambar 4. 4: Proses resize pada dataset	36
Gambar 4. 5: Proses augmentasi dataset	37
Gambar 4. 6: Proses exposure dataset	37
Gambar 4. 7: Proses rotation dataset	38
Gambar 4. 8: Proses flip dataset	39

Gambar 4. 9: Proses pembagian dataset.....	39
Gambar 4. 10: Code untuk mengimpor dataset dari Roboflow	40
Gambar 4. 11: Keterangan jika berhasil mengunduh dan mengekstrak dataset ...	40
Gambar 4. 12: Hasil ekstrasi dataset.....	41
Gambar 4. 13: Isi dari file data.yaml.....	41
Gambar 4. 14: Mengimpor YOLO ke dalam proyek	42
Gambar 4. 15: Code untuk menginstall library	42
Gambar 4. 16: Code untuk mengombinasikan parameter.....	42
Gambar 4. 17: Code untuk melakukan pelatihan model pada YOLO	44
Gambar 4. 18: Grafik precision dan recall	46
Gambar 4. 19: Perbandingan optimizer	46
Gambar 4. 20: Kumpulan grafik hasil pelatihan model	47
Gambar 4. 21: Confusion matrix percobaan ketujuh	48
Gambar 4. 23: Gambar yang kurang tepat diprediksi	49
Gambar 4. 22: Gambar yang diprediksi dengan benar.....	49
Gambar 4. 25: Tampilan halaman pertama ketika website diakses	50
Gambar 4. 24: Tampilan halaman jika memilih menu gambar.....	51
Gambar 4. 26: Tampilan jika memilih menu video	51
Gambar 4. 27: Tampilan jika memilih menu youtube	52
Gambar 4. 29: Tampilan ketika memilih menu Real-Time	52
Gambar 4. 28: Code untuk mendeteksi gambar yang diupload	53
Gambar 4. 30: Function untuk menampilkan gambar hasil deteksi.....	53
Gambar 4. 31: Function untuk mencari video dari link youtube	54
Gambar 4. 33: Function untuk mengambil foto atau gambar secara langsung.....	56

Gambar 4. 32: Function untuk memproses video yang diupload.....	56
Gambar 4. 34: Function untuk realtime	57
Gambar 4. 35: Code untuk mendeteksi realtime.....	57
Gambar 4. 36: Chart rincian responden	61

BAB I:

PENDAHULUAN

1.1. Latar Belakang

Pendidikan anak usia dini merupakan periode perkembangan yang penting dalam kehidupan anak (Heri Pratikno dkk., 2023; Musdalifah dkk., 2020; Supriadi dkk., 2021; Yuni Wulandari dkk., 2022). Pendidikan sejak usia dini turut mengambil peran dalam kesuksesan di masa depan anak. Hal ini juga ditekankan dalam undang-undang sistem pendidikan nasional Republik Indonesia no. 20 tahun 2003 pada bab 1 butir ke-14 (Heri Pratikno dkk., 2023). Usia dini merupakan waktu masa anak mulai peka dalam menerima rangsangan sehingga anak dapat mudah menerima hal yang baru dan menarik. Di usia seperti ini juga waktu yang tepat untuk mengajarkan kepada anak mengenai benda-benda atau objek yang ada di sekitarnya sehingga dapat membantu perkembangan kemampuan kognitifnya (Priyono dkk., 2021; Yuni Wulandari dkk., 2022).

Pengenalan objek adalah salah satu bidang di dalam bidang *computer vision*, yang digunakan untuk mengenali objek yang ada di sekitar dan bertujuan untuk mengidentifikasi dan melokalisasi objek-objek tertentu dalam gambar atau video. Pengenalan objek semakin banyak diminati sejak 1960-an dan terus berkembang hingga saat ini. Hal ini dibuktikan dengan penerapan *object detection* di berbagai bidang seperti di bidang medis dan bidang industri (Aini dkk., 2021; Karlina & Indarti, 2019; Rafly Alwanda dkk., 2020; Wu dkk., 2020). Pengenalan objek telah menjadi elemen kunci dalam berbagai aplikasi teknologi yang melibatkan analisis visual dan pemahaman konteks (Karlina &

Indarti, 2019; Rafly Alwanda dkk., 2020). Dalam beberapa dekade terakhir, kemajuan dalam teknologi pengolahan citra dan *computer vision* telah memungkinkan pengenalan objek menjadi lebih efisien dan akurat.

You Only Look Once (YOLO) adalah salah satu arsitektur dalam bidang *computer vision* yang bisa digunakan untuk pengenalan objek. YOLO menggunakan pendekatan yang menggabungkan deteksi dan klasifikasi objek dalam satu tahap (*one-stage-detector*) sehingga memberikan kecepatan dan efisiensi yang signifikan dibandingkan dengan metode tradisional yang memerlukan beberapa tahap pemrosesan (*two-stage-detector*) (Karlina & Indarti, 2019). YOLO telah menjadi salah satu pendekatan yang populer dalam pengenalan objek *real-time* dan banyak penelitian terkait telah memperluas aplikasi teknik ini dalam berbagai konteks seperti yang dilakukan dalam beberapa penelitian terakhir ini (Adarsh & Rathi, 2020; Aini dkk., 2021; Karlina & Indarti, 2019; Kumari dkk., 2021; Lou dkk., 2023; Zhang dkk., 2019).

Dengan memanfaatkan kecepatan dan efisiensi YOLO dalam mengenali objek secara *real-time* (Redmon dkk., 2015), penelitian ini bertujuan untuk menciptakan sebuah metode pembelajaran yang dapat membantu anak-anak dalam memahami atau mengenali lingkungan sekitar dengan mendeteksi objek-objek tersebut. Diharapkan bahwa integrasi teknologi ini akan memberikan kontribusi positif dalam memfasilitasi proses belajar anak-anak pada usia dini, memperluas cakupan pemahaman mereka terhadap objek di sekitar, dan mendukung perkembangan kemampuan kognitif serta pemahaman visual mereka secara menyeluruh.

1.2. Rumusan Masalah

Usia dini merupakan masa-masa emas untuk anak-anak dalam mempelajari objek-objek yang ada di sekitar untuk membantu perkembangan kemampuan kognitif anak-anak. Dalam beberapa hal, teknologi belum dapat dimanfaatkan secara optimal untuk membantu perkembangan kemampuan kognitif anak-anak. Karena itu, bagaimana penerapan teknologi *computer vision* khususnya menggunakan YOLO agar dapat membantu anak-anak untuk mengenali objek sekitar?

1.3. Batasan Masalah

Pada penelitian ini terdapat beberapa batasan, yaitu:

1. Usia anak-anak yang menjadi fokus penelitian ini adalah anak-anak *pre-school* dengan rentang umur 2 sampai 4 tahun.
2. Pada pengenalan objek untuk anak-anak, peneliti hanya berfokus pada objek-objek yang ada di sekitar dan mudah ditemui.
3. Objek pengenalan yang dijadikan *dataset* akan berbentuk gambar yang memuat 6 objek yang berbeda, yaitu: *handphone*, mobil, tas, manusia, jam, dan sepatu.
4. Kuesioner yang dilakukan akan mengacu pada aspek kegunaan sehingga yang menjadi target kuesioner adalah guru kelompok bermain, orang tua, masyarakat umum.

1.4. Tujuan Penelitian

Adapun tujuan penelitian ini adalah:

1. Menerapkan atau membuat sistem pengenalan objek menggunakan arsitektur YOLO yang dapat membantu anak-anak dalam mempelajari

objek-objek yang ada di sekitarnya dengan lebih mudah dan efektif sehingga dapat meningkatkan kualitas anak.

2. Mengukur dan mengevaluasi hasil penelitian dalam pengenalan objek sekitar berdasarkan *confusion matrix* dan mAP untuk melihat kebagusan model.
3. Mengukur dan mengevaluasi hasil penelitian berdasarkan kuesioner untuk melihat aspek kegunaan *website*.

1.5. Manfaat Penelitian

Manfaat dari penelitian ini adalah:

1. Meningkatkan pembelajaran anak pada usia dini. Penelitian ini dapat membantu meningkatkan kualitas pembelajaran anak-anak usia dini dengan memanfaatkan teknologi *computer vision*. Anak-anak akan dapat belajar lebih efektif dengan bantuan pengenalan objek yang lebih baik.
2. Pengembangan lingkungan pendidikan. Hasil penelitian ini dapat membantu dalam pengembangan lingkungan pendidikan yang lebih modern dan adaptif, yang dapat memenuhi tuntutan era digital.
3. Kontribusi pada penelitian selanjutnya. Penelitian ini dapat menjadi dasar untuk penelitian lebih lanjut dalam penggunaan teknologi *computer vision* dalam konteks pendidikan anak-anak usia dini, dan dapat memberikan panduan bagi pengembangan teknologi serupa.

1.6. Sistematika Penulisan

Dalam penulisan penelitian ini terdapat lima bab utama. Pada bab I akan berbicara mengenai latar belakang penulisan, rumusan masalah, batasan masalah, tujuan penelitian, dan manfaat penelitian. Setelah membahas

mengenai latar belakang, maka pada bab II akan memuat teori-teori atau konsep-konsep yang akan dipakai pada penelitian ini. Dan pada bab III akan berisi tentang metodologi yang akan digunakan di dalam penelitian ini. Selanjutnya pada bab IV akan berisi pembahasan dan analisis dari langkah-langkah penelitian yang dijelaskan pada bab III. Dan yang terakhir pada bab V akan berisi kesimpulan terkait penelitian yang sudah dilakukan dan saran terhadap pengembangan selanjutnya.

BAB II:

TINJAUAN PUSTAKA

2.1. Review Literature

Pada penelitian sebelumnya telah berhasil dilakukan pengenalan objek binatang berbasis *Augmented Reality* (AR) oleh I Dewa Gede Wahya Dhiyatmika dan teman-temannya (Dhiyatmika dkk., 2015). Dalam perjalanan seturut perkembangan *deep learning* juga telah banyak dilakukan penelitian untuk mendeteksi objek seperti yang dilakukan oleh Muhammad Fadhlan Supriadi dan teman-teman (Supriadi dkk., 2021) menggunakan *MobileNet*. Penelitian ini berhasil mendapatkan *mean Average Precision* (mAP) sebesar 99,34%. Pada penelitian yang dilakukan Ike Yuni Wulandari dan teman-teman (Yuni Wulandari dkk., 2022) pun mendapatkan hasil yang baik dan efektif. Pada penelitian ini pun disimpulkan bahwa dengan menggunakan *python* objek dapat dikenali dengan baik dan variasi pengambilan data tidak menjadi hambatan.

Selain menggunakan dukungan AR dan Raspberry, algoritma *Convolutional Neural Network* (CNN) dapat digunakan untuk membantu pengenalan objek seperti yang dilakukan oleh Dennis Saputra Ariansyah (Ariansyah, t.t.). Pada penelitian ini Dennis mencoba mengklasifikasi hewan dengan menggunakan CNN dan *transfer learning* dari *GoogleNet*. Penggunaan CNN dan *transfer learning* ini cukup baik dalam klasifikasi hewan yang dibuktikan dengan tingkat akurasi mencapai 98,36%. Ada lagi penelitian untuk pengenalan gestur jari tangan yang dilakukan Muhammad Rifki Pratama dan teman-teman (Heri Pratikno dkk., 2023). Penelitian ini mencoba dua cara

dalam pengenalan gestur jari, yaitu *framework MediaPipe* dan CNN. Dari penelitian ini diketahui bahwa penggunaan CNN kurang optimal dibandingkan *MediaPipe* jika dilakukan secara *real-time*.

Berbicara mengenai pengenalan objek secara *real-time*, YOLO dalam perkembangannya mendapat perhatian yang khusus dalam pengenalan objek dan mendeteksinya. Hal ini dibuktikan dengan beberapa penelitian, seperti yang dilakukan oleh Pranav Adarsh, Pratibha Rathi, dan Manoj Kumar (Adarsh & Rathi, 2020) dan Haitong Lou bersama kawan-kawannya (Lou dkk., 2023) yang menunjukkan bahwa YOLO dapat digunakan untuk mendeteksi objek dalam waktu yang lebih singkat dibandingkan dengan CNN dan mendapatkan hasil yang bagus baik secara *real-time* maupun tidak.

Secara lebih lanjut berikut adalah tabel *review literatur* mengenai penelitian-penelitian yang berkaitan dengan penelitian yang akan dibuat.

Tabel 2. 1: Tabel review literature

No.	Judul, Peneliti, Penerbit dan Tahun Terbit	Hasil Penelitian, Keterbatasan, Peluang
1.	Judul: Pengenalan Gestur Jari Tangan Sebagai Media Pembelajaran Berhitung Bagi Paud Berbasis Visi Komputer dan Deep Learning	Hasil Penelitian: Hasil penelitian ini menunjukkan bahwa dalam konteks pengenalan gestur jari tangan sebagai media pembelajaran berhitung bagi anak paud, arsitektur <i>Convolutional Neural Network</i> (CNN) mencapai akurasi hasil pelatihan 100% pada <i>epoch</i> ke-5, dengan total waktu komputasi 17,113 detik. CNN memerlukan waktu komputasi 12 detik pada setiap langkah (step) dan 3,366 - 3,452 detik pada setiap <i>epoch</i> -nya. Namun, saat dibandingkan dengan <i>MediaPipe</i> , hasil komparasi menunjukkan bahwa <i>MediaPipe</i> memiliki persentase akurasi rata-rata sebesar 89,9% dengan FPS (<i>frame per second</i>) antara 25-30, sedangkan CNN memiliki persentase akurasi rata-rata sebesar 20% dengan FPS antara 12-15. Performa CNN kurang optimal untuk deteksi

		objek secara <i>real-time</i> karena memerlukan proses pelatihan <i>dataset</i> gestur jari tangan yang membebani komputasi, sedangkan <i>MediaPipe</i> dirancang khusus untuk deteksi gestur tangan.
	Peneliti: Muhammad Rifki Pratama, Heri Pratikno, Yosefine Triwidyastuti, dan Musayyanah	Keterbatasan: Kelemahan penelitian ini adalah performa CNN dalam deteksi objek <i>real-time</i> ternyata kurang optimal, dan metode ini lebih sesuai untuk klasifikasi citra. Selain itu, hasil pelatihan CNN mencapai akurasi 100%, tetapi performanya dalam proses pengujian (testing) tidak sebaik <i>MediaPipe</i> , yang memiliki akurasi yang lebih tinggi dan FPS yang lebih tinggi.
	Penerbit dan Tahun Terbit: COMPLETE: Journal of Computer, Electronic, and Telecommunication, 2023	Peluang: Peluang dari penelitian ini adalah pengembangan lebih lanjut dengan membandingkan kinerjanya dengan arsitektur <i>deep learning</i> lainnya, seperti menggunakan <i>pre-trained network</i> model seperti LSTM, Faster-RCNN, dan ResNet5.0. Hal ini dapat memberikan wawasan lebih lanjut tentang metode terbaik untuk deteksi gestur jari tangan dalam konteks pembelajaran berhitung anak paud. Selain itu, pengembangan metode yang lebih efisien untuk CNN dalam deteksi objek <i>real-time</i> mungkin dapat meningkatkan kinerjanya, sehingga dapat menjadi lebih bersaing dengan <i>MediaPipe</i> dalam hal akurasi dan FPS.
2.	Judul: You Only Look Once: Unified, Real-Time Object Detection	Hasil Penelitian: Penelitian ini memperkenalkan YOLO, sebuah model yang menyatukan pendekatan untuk deteksi objek. Model ini dirancang dengan sederhana dan dapat dilatih langsung pada gambar utuh. YOLO juga diakui sebagai pendekatan tercepat dalam literatur untuk deteksi objek secara umum dan mendorong perkembangan dalam deteksi objek <i>real-time</i> . Selain itu, YOLO mampu memberikan hasil yang baik dalam berbagai domain, menjadikannya ideal untuk aplikasi yang mengandalkan deteksi objek yang cepat dan andal.

	<p>Peneliti: Joseph Redmon, Santosh Divvala, Ross Girshick, dan Ali Farhadi</p>	<p>Keterbatasan: Berdasarkan jurnal dan hasil penelitian, tidak ditemukan keterbatasan karena berhubung ini adalah sebuah penemuan yang menjadi titik tolak kemajuan algoritma YOLO</p>
	<p>Penerbit dan Tahun Terbit: Proceedings of the IEEE conference on computer vision and pattern recognition 2015</p>	<p>Peluang: Dikarenakan tidak adanya keterbatasan yang diberikan, kemungkinan peluang yang dapat diambil dari penelitian ini adalah mengenai pengaplikasian YOLO secara <i>real-time</i> dan pengaplikasian ke dalam berbagai hal.</p>
3.	<p>Judul: YOLBO: You Only Look Back Once—A Low Latency Object Tracker Based on YOLO and Optical Flow</p>	<p>Hasil Penelitian: Penelitian ini membangun pada paradigma pelacakan yang sudah berhasil dan dapat diterapkan pada berbagai kasus penggunaan dan tipologi perangkat keras yang berbeda.</p>
	<p>Peneliti: Daniel S. Kaputa, dan Brian P. Landy</p>	<p>Keterbatasan:</p>
	<p>Penerbit dan Tahun Terbit: IEEE Access 2021</p>	<p>Peluang: Peluang yang bisa dikembangkan dari penelitian ini berkaitan dengan penelitian yang menggunakan <i>frame rate</i> dinamis, dan penggunaan YOLBO pada embedded DPGA Soc</p>
4.	<p>Judul: A Real-Time Method to Estimate Speed of Object Based on Object Detection and Optical Flow Calculation</p>	<p>Hasil Penelitian: Dari penelitian ini didapatkan bahwa metode yang digunakan dapat mengestimasi kecepatan dari berbagai jenis objek.</p>
	<p>Peneliti: Kaizhan Liu, Yunming Ye, Xutao Li, dan Yan Li</p>	<p>Keterbatasan:</p>
	<p>Penerbit dan Tahun Terbit: Journal of Physics: Conference Series 2018</p>	<p>Peluang: Peluang yang dapat diambil dari penelitian ini adalah mengenai pengaplikasian di berbagai bidang yang memerlukan pengukuran kecepatan objek, juga dapat dikembangkan untuk mengukur atau menganalisis sebuah pergerakan.</p>
5.	<p>Judul: Deteksi Masker Wajah</p>	<p>Hasil Penelitian:</p>

	Menggunakan Deep Transfer Learning dan Augmentasi Gambar	Hasil penelitian menunjukkan bahwa penggunaan teknik <i>deep transfer learning</i> dan augmentasi gambar dapat meningkatkan kinerja model deteksi masker wajah secara signifikan. Penerapan kedua pendekatan tersebut memberikan kontribusi peningkatan kinerja sebesar 12-13%. Berdasarkan pengujian akhir model <i>deep learning</i> yang dibangun mencapai akurasi 98,3% dan skor F1 98,7% pada <i>dataset</i> validasi. Penelitian selanjutnya dapat diarahkan untuk mendeteksi hal yang lebih kompleks seperti mendeteksi penggunaan masker ganda (masker bedah dan masker kain).
	Peneliti: Raden B. Hadiprakoso, dan Nurul Qomariasih	Keterbatasan:
	Penerbit dan Tahun Terbit: Jurnal Informatika dan Komputer 2022	Peluang: Peluang yang dapat dilanjutkan dari penelitian ini mengenai penerapan <i>deep learning</i> dan augmentasi dalam mendeteksi objek di berbagai bidang.
6.	Judul: Data Augmentasi Untuk Mengatasi Keterbatasan Data Pada Model Penerjemah Bahasa Isyarat Indonesia (BISINDO)	Hasil Penelitian: Dari penelitian ini menghasilkan model penerjemah alfabet BISINDO dengan algoritma CNN yang mencapai akurasi 94.38% padahal sebelumnya tanpa menggunakan <i>pre-processing</i> hanya mencapai akurasi 30%.
	Peneliti: Riestiya Zain Fadillah, Ade Irawan, dan Meredita Susanty	Keterbatasan: Keterbatasan yang terdapat dari penelitian ini berupa ruang lingkup terjemahan yaitu alfabet.
	Penerbit dan Tahun Terbit: Jurnal Informatika 2021	Peluang: Peluang yang dapat dilanjutkan adalah mengenai perluasan cakupan penerjemahan hingga pada penerjemahan kata dan kalimat.
7.	Judul: A Review of YOLO Algorithm Developments	Hasil Penelitian: Hasil penelitian ini mengidentifikasi bahwa YOLO versi-versi memiliki banyak perbedaan, tetapi masih memiliki beberapa fitur yang sama, sehingga tetap memiliki kemiripan. Ini menunjukkan bahwa ada ruang bagi penelitian lebih lanjut dalam pengembangan versi-versi

		YOLO, terutama dalam konteks implementasi skenario.
	Peneliti: Peiyuan Jiang, Daji Ergu, Fangyao Liu, Ying Cai, dan Bo Ma	Keterbatasan: Keterbatasan makalah ini termasuk kurangnya fokus pada implementasi perbandingan, seperti analisis skenario, yang bisa menjadi area penelitian yang lebih mendalam.
	Penerbit dan Tahun Terbit: Procedia Computer Science 2022	Peluang: Peluang yang dapat dilanjutkan dari penelitian ini mengenai implementasi YOLO.
8.	Judul: YOLO v3-Tiny: Object Detection and Recognition using one stage improved model	Hasil Penelitian: Dari penelitian ini ditemukan bahwa YOLO v3-Tiny dapat meningkatkan kecepatan deteksi objek dengan akurasi yang bagus.
	Peneliti: Pranav Adarsh, Pratibha Rathi, dan Manoj Kumar	Keterbatasan: Keterbatasan yang ada di dalam penelitian ini terkait penanganan pencahayaan yang kurang bagus, variasi skala objek dan kompleksitas latar belakang.
	Penerbit dan Tahun Terbit: International Conference on Advanced Computing & Communication Systems (ICACCS) 2020	Peluang: Peluang yang ada di dalam penelitian ini berkaitan dengan meningkatkan akurasi lokalisasi target yang kecil, juga memperluas deteksi objek gambar statis.
9.	Judul: DC-YOLOv8: Small-Size Object Detection Algorithm Based on Camera Sensor	Hasil Penelitian: Hasil penelitian ini adalah pengembangan algoritma deteksi objek berukuran kecil yang berbasis pada sensor kamera yang dikombinasikan dengan kecerdasan buatan (<i>artificial intelligence</i>). Algoritma ini mencapai hasil yang lebih baik daripada algoritma-algoritma sebelumnya dalam hal akurasi dan kecepatan deteksi. Eksperimen dan pengujian dilakukan pada beberapa dataset, termasuk Visdrone, Tinyperson, dan PASCAL VOC2007, dan algoritma ini berhasil membuktikan keunggulannya dalam mendeteksi objek kecil dalam berbagai situasi kompleks.
	Penulis: Haitong Lou, Xuehu	Keterbatasan:

	Duan, Junmei Guo, Haiying Liu, Jason Gu, Lingyun Bi, dan Haonan Chen	Keterbatasan umum dalam penelitian deteksi objek mungkin termasuk performa yang kurang baik dalam situasi pencahayaan yang buruk, kebingungan dalam mengenali objek yang tumpang tindih, atau pengaruh latar belakang yang rumit.
	Penerbit dan Tahun Terbit: Electronics 2023	Peluang: Peluang yang dapat dilanjutkan atau dikembangkan adalah mengenai peningkatan algoritma deteksi objek untuk berbagai ukuran objek. Penelitian tersebut ingin mencapai tingkat akurasi yang lebih baik dalam waktu secepat mungkin.

2.2. Landasan Teori

2.2.1. Pendidikan Anak Usia Dini

Anak usia dini, yaitu anak dengan rentang usia 0 hingga 6 tahun, merupakan sosok sosiokultural yang sedang mengalami perkembangan fundamental. Pada fase ini, anak mampu menerima dan mengolah informasi dengan cepat serta tahan lama. Anak-anak akan memasuki tahap pra-operasional antara usia 2 hingga 7 tahun, di mana mereka mulai menggunakan citra-citra untuk mengenali lingkungan sekitar serta mengembangkan kemampuan berpikir simbolis. (Putri dkk., 2021; Safita & Suryana, 2022; Sunarti dkk., 2023; Zulwati dkk., 2022).

Pendidikan usia dini, selain menekankan nilai agama dan moral, juga memiliki fokus pada pengembangan kognitif anak sebagai dasar pijakannya di masa yang akan datang. Perkembangan kognitif anak usia dini merupakan tahap penting dalam proses belajar dan memahami dunia di sekitarnya. Karena itu, stimulus

yang tepat dapat membantu perkembangan ini berjalan optimal (Nur dkk., 2020). Aspek kognitif menjadi hal utama dalam perkembangan anak karena berkaitan dengan bagaimana cara mereka berpikir dan mulai berfungsi. Karena itu, tujuan utama pendidikan anak usia dini dalam hal perkembangan kognitif adalah agar anak mampu berpikir kritis, menalar, memecahkan masalah, dan menemukan sebab-akibat dari hasil belajarnya secara mandiri (Komang Ayu & Surya Manuaba, 2021; Nur dkk., 2020).

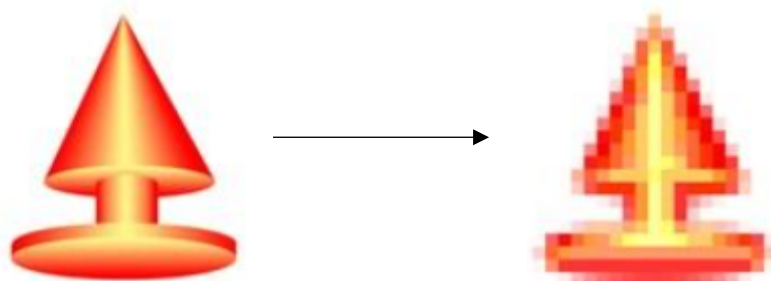
Kelompok bermain atau PAUD dirancang khusus untuk merangsang perkembangan kognitif anak usia yang memasuki usia 2 – 4 tahun dengan melalui berbagai kegiatan bermain yang edukatif dan menyenangkan. Dengan stimulasi dan pendidikan yang tepat, anak usia 2 – 4 tahun akan memiliki fondasi kognitif yang kuat yang akan bermanfaat bagi perkembangan mereka selanjutnya (Putri dkk., 2021).

2.2.2. Image Processing

Citra adalah suatu gambaran atau kemiripan atau suatu imitasi dari suatu objek (Andono dkk., 2017; Hidayatullah, 2017). Pada umumnya citra dibagi menjadi dua jenis, yaitu citra analog dan citra digital (Iryanto & Zaini, 2014). Citra analog tidak dapat direpresentasikan oleh komputer. Agar komputer dapat mengolah citra analog tersebut, citra harus melewati proses sampling dan kuantisasi yang disebut “digitalisasi” di mana proses digitalisasi ini akan mengubah citra analog menjadi citra digital (Andono dkk.,

2017). Proses digitalisasi ini akan memecahkan citra analog menjadi sejumlah baris (m) dan kolom (n). Sehingga citra digital $a[m, n]$ dapat dijelaskan sebagai ruang diskrit 2D yang berasal dari sebuah citra analog $a[x, y]$. Titik pertemuan antara baris dan kolom dalam citra digital disebut *pixel* (Andono dkk., 2017; Iryanto & Zaini, 2014).

Citra pada umumnya menyimpan banyak informasi. Namun, seringkali citra mengalami penurunan kualitas citra sehingga dibutuhkan sebuah pemrosesan citra untuk meningkatkan kualitasnya (Adhinata dkk., 2020; Hidayatullah, 2017; Panggalih dkk., 2022). Pemrosesan citra adalah bidang studi yang fokus pada pengolahan dan analisis data pada citra digital yang bertujuan untuk meningkatkan kualitas citra. Tahapan pengolahan citra digital pada umumnya adalah akuisisi citra, peningkatan kualitas citra, segmentasi citra, ekstraksi fitur citra, dan klasifikasi citra (Adhinata dkk., 2020). Hasil dari pengolahan citra sebagian besar dapat mempengaruhi bagian tingkat tinggi selanjutnya untuk melakukan pengenalan dan pemahaman terhadap data citra (Chen dkk., 2021).

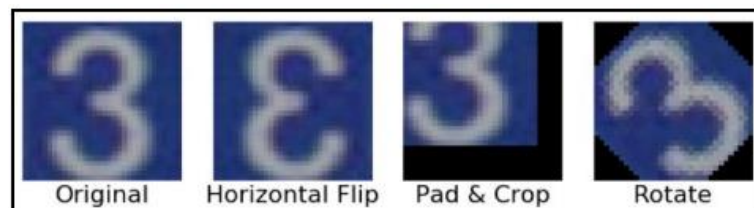


Gambar 2. 1: Ilustrasi proses digitalisasi

Sumber: <https://slideplayer.info/slide/11809302/>

2.2.3. Augmentasi

Augmentasi data adalah teknik untuk memperluas *dataset* pelatihan dengan membuat variasi pada data yang ada dengan memanipulasi transformasi dimensi gambar (Perez dkk., 2018; Sanjaya & Ayub, 2020). Teknik augmentasi data seperti *cropping*, *padding*, dan *flipping horizontal* bertujuan untuk meningkatkan kinerja model pembelajaran mesin dengan memberikan lebih banyak variasi pada data pelatihan, sehingga model dapat belajar fitur yang lebih umum dan dapat digeneralisasi dengan baik pada data baru. Selain itu, dengan augmentasi pun dapat mengurangi *overfitting* seminimal mungkin. (Fadillah dkk., 2021; Perez dkk., 2018).



Gambar 2. 2: Ilustrasi proses augmentasi

Sumber: <https://www.51cto.com/article/626105.html>

2.2.4. You Only Look Once (YOLO)

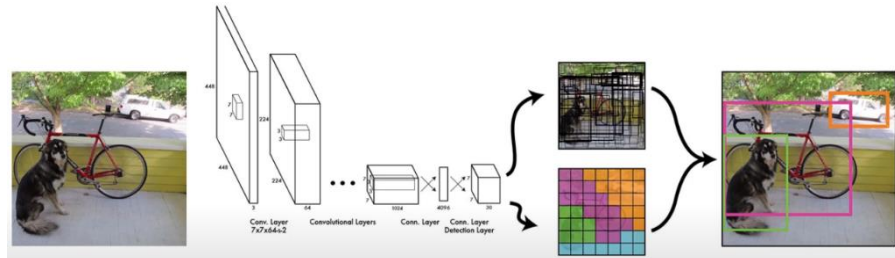
You Only Look Once (YOLO) adalah sebuah arsitektur yang menerapkan pendekatan baru untuk mendeteksi objek. YOLO pertama kali dikenalkan pada tahun 2015 (Redmon dkk., 2015). Algoritma ini biasanya digunakan untuk mendeteksi objek secara *realtime*. YOLO menggunakan jaringan saraf tiruan untuk mempelajari pola-pola sehingga dapat mendeteksi dan mengenali objek sekaligus memprediksi *bounding box* dan kelasnya. YOLO merupakan algoritma *One-stage Detector* hasil dari pengembangan

algoritma *Convolutional Neural Network* (CNN) yang menggabungkan proses ekstraksi fitur dan proses regresi atau klasterisasi dalam satu *network* (Thoriq dkk., 2023). Hal inilah yang membuat YOLO dapat memprediksi kotak pembatas dan probabilitas kelas langsung dari gambar penuh dalam satu evaluasi. Proses prediksi ini dapat dilakukan juga karena arsitektur YOLO terdiri dari 24 *convolutional layer* yang digunakan untuk mengekstraksi fitur inputan dan memiliki dua *conneted layer* untuk memperdiksi probabilitas dan koordinat objek (Jiang dkk., 2022; Redmon dkk., 2015).

YOLO akan membagi inputan menjadi grid $S \times S$ dan setiap sel bertanggung jawab untuk memprediksi sejumlah bounding box dan confidence skor. Confidence skor ini nantinya akan merefleksikan tiap objek yang terdeteksi. Setelah itu, hasil dari tiap deteksi akan melewati proses *threshold* (ambang batas) dan *Non-Maximum Suppression* (NMS). Pada proses *threshold* kotak yang memiliki nilai *confidence* yang rendah akan tidak dianggap dan pada proses NMS, akan menghilangkan *bounding box* yang memiliki *overlap* dengan membandingkan nilai IoU dan nilai *threshold*. Adapun rumus dari IoU seperti pada persamaan 1.

$$IoU = \frac{\text{Area of Intersection}}{\text{Area of Unio}} \quad (1)$$

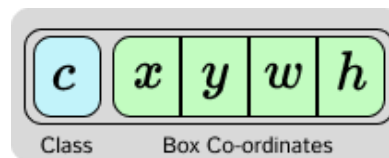
Jika diketahui bahwa nilai IoU lebih besar dari *threshold* maka akan di hilangkan (Maulana, 2021; Redmon dkk., 2015). Secara lebih rinci dari proses YOLO ini dapat dilihat dari gambar 2.3.



Gambar 2. 3: Ilustrasi proses deteksi YOLO

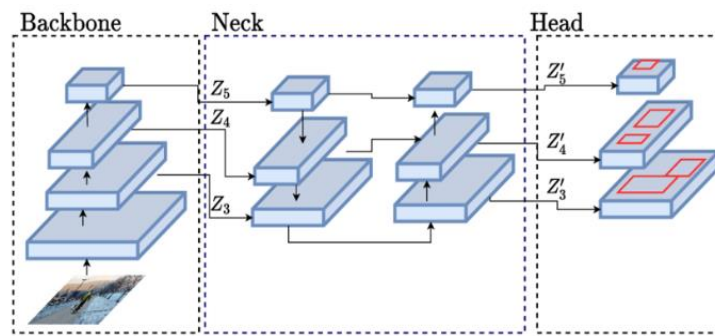
Sumber: <https://www.youtube.com/watch?v=kMDf35Ta-84&list=WL&index=1>

Setiap *bounding box* yang dihasilkan dan/atau dibuat akan menghasilkan lima nilai di dalamnya. Nilai-nilai ini terdiri dari nilai c , nilai x dan y , nilai w dan h , di mana x dan y mewakili pusat dari kotak relatif ke batas sel *grid*, w dan h merupakan lebar dan tinggi dari *bounding box*, dan c adalah jenis objek yang ada. Hal inilah yang membuat YOLO menjadi algoritma yang cepat dan akurat (Kaputa & Landy, 2021; Kumari dkk., 2021; Lou dkk., 2023).



Gambar 2. 4: Ilustrasi nilai yang merepresentasikan bounding box

Ultralytics merilis YOLO v8 yang dikembangkan dari versi-versi sebelumnya. Pada YOLO v8 ini terdapat beberapa perubahan pada arsitektur dan optimasi terhadap akurasi (Jocher & Sergiuwaxmann, 2023). Secara garis besar, arsitektur YOLO v8 terdiri dari tiga bagian utama: *backbone*, *neck*, dan *head* (Lou dkk., 2023).



Gambar 2. 5: Arsitektur YOLO v8

Sumber: https://www.researchgate.net/figure/A-one-stage-object-detection-model-generally-consists-of-a-backbone-for-feature_fig1_372313282

Backbone bertugas sebagai penyaring fitur dasar dari gambar yang masuk. Pada YOLOv8, *backbone* atau tulang punggung menggunakan CSPDarknet53 yang dimodifikasikan. Ditahap ini, data *input* akan melewati lapisan konvolusi yang membantu mengekstraksi fitur-fitur lokal seperti tepi, sudut, dan tekstur, kemudian melewati lapisan *pooling* yang akan mengurangi dimensi spasial dari fitur-fitur tersebut. *Neck* yang berada di antara *backbone* dan *head* akan menyusun kembali fitur-fitur yang telah diekstrak oleh *backbone*. Penyusunan kembali ini akan melibatkan beberapa lapisan konvolusi dan *pooling* tambahan sehingga dapat membantu *neck* untuk menyusun kembali informasi yang relevan. Bagian terakhir adalah *head*. *Head* akan bertugas untuk mengklasifikasi, regresi, dan mendeteksi objek berdasarkan fitur-fitur yang telah disusun oleh *neck*. *Neck* bertujuan untuk mengintegrasikan informasi yang didapatkan sehingga menghasilkan representasi fitur yang lebih kompleks dan kuat (Jocher & Sergiuwaxmann, 2023; Khare dkk., 2023; Lou dkk., 2023).

2.2.5. Roboflow

Roboflow adalah sebuah platform perangkat lunak yang memiliki kemampuan dalam mengorganisasi data, anotasi sampai pada pelatihan dan penerapan model yang dibuat. Roboflow digunakan untuk membuat label *bounding box* pada dataset gambar yang bisa digunakan sebagai dataset pelatihan dan testing (Dwyer & Gallagher, 2023).

2.2.6. Labeling Gambar

Anotasi gambar atau pelabelan gambar merupakan sebuah proses memberikan label atau keterangan pada gambar. Label atau keterangan ini merupakan informasi lanjutan mengenai objek yang ada di dalamnya. Pelabelan ini biasanya berupa *bounding box* yang menunjukkan lokasi dari objek sekaligus mengidentifikasi dan mengategorikan objek dalam gambar. Proses ini sering dilakukan dalam konteks *computer vision* untuk melatih model yang mampu mengenali dan mengklasifikasi objek. Dengan pelabelan ini juga dapat membantu model untuk mengenali pola objek dan membedakan tiap objek (Heri Pratikno dkk., 2023; Maulana, 2021).

2.2.7. Confusion Matrix

Confusion matrix adalah tabel yang digunakan untuk mengevaluasi hasil pemodelan pada klasifikasi *dataset*. *Confusion matrix* nantinya akan menampilkan atau memvisualisasikan gambaran lengkap mengenai hasil klasifikasi yang tepat dan yang salah. Pada umumnya terdapat empat nilai utama dalam *confusion*

matrix, yaitu 1) *true positive*, 2) *true negative*, 3) *false negative*, dan 4) *false negative* (Maurya dkk., 2021).

2.2.8. Mean Average Precision

Mean Average Precision (mAP) adalah sebuah metrik evaluasi yang digunakan dalam bidang pemrosesan citra dan pengenalan objek, terutama dalam tugas deteksi objek. *Mean Average Precision* (mAP) mengukur sejauh mana model dapat mengidentifikasi objek dengan akurasi yang tinggi. *Mean Average Precision* (mAP) mengambil hasil dari *Precision-Recall* (P-R) *Curve* untuk setiap kelas objek dan kemudian menghitung *Average Precision* (AP) untuk setiap kelas berdasarkan *Area Under the Curve* (AUC) dari *P-R Curve* tersebut. Setelah itu, mAP mengambil rata-rata dari AP untuk semua kelas objek yang ada, sehingga semakin tinggi nilai mAP, semakin baik kinerja model deteksi objek tersebut (Maxwell dkk., 2021; Q. Wang dkk., 2018; Zhao & Li, 2020).

2.2.9. Skala Likert

Skala Likert yang dikembangkan oleh Rensis Likert merupakan sebuah skala pengukuran kemampuan yang bertujuan untuk melihat tingkat persetujuan terhadap pertanyaan atau pernyataan yang diberikan (Suasapha, 2020). Kemampuan yang diukur ini akan dijabarkan menjadi beberapa indikator dan digunakan untuk menyusun instrumen. Bentuk kuesioner yang jamak ini adalah pernyataan yang disertai dengan skala pengukuran, di mana skala tersebut merupakan pilihan sikap terkait pernyataan

yang mengikutinya. Karena itu, pilihan sikap yang sering kali dilihat pada kuesioner skala Likert biasanya mengenai persetujuan terhadap pernyataan atau pertanyaan terkait (Ayuka dkk., 2021; Suasapha, 2020).

Adapun jawaban instrumen dari skala Likert ini dikembangkan menjadi empat kategori, yaitu sangat setuju (SS), setuju (S), tidak setuju (TS), dan sangat tidak setuju (STJ) (Ayuka dkk., 2021).

2.2.10. Uji Validitas

Uji validitas adalah pengujian terhadap alat ukur untuk melihat sejauh mana sebuah instrumen dapat mengukur apa yang akan diukur. Uji validitas juga digunakan agar dapat menguji ketepatan dan ketetapan suatu alat ukur yang digunakan sebagai pengukur (Ayuka dkk., 2021; Rosita dkk., 2021). Alat ukur atau kuesioner dinyatakan valid jika setiap butir pertanyaan dapat digunakan sebagai perantara untuk mengungkapkan sesuatu yang akan diukur oleh kuesioner. Valid dan tidaknya sebuah alat ukur dilihat dari nilai r hitung yang lebih besar dari rtabel (Budiyanta, 2018; Rosita dkk., 2021).

Untuk menghitung valid tidaknya sebuah alat ukur dapat dilihat pada persamaan 2.

$$r = \frac{n(\sum xy) - (\sum x)(\sum y)}{\sqrt{[n\sum x^2 - (\sum x)^2][n\sum y^2 - (\sum y)^2]}} \quad (2)$$

Di mana:

n = koefisien korelasi

$\sum x$ = jumlah skor item

$\sum y$ = jumlah skor total

n = jumlah responden

(Budyanta, 2018; Ernawati & Sukardiyono, 2017).

2.2.11. Uji Reliabilitas

Uji reliabilitas digunakan untuk mengukur sejauh mana instrumen atau alat ukur memiliki kekonsistenan untuk mengukur apa yang semestinya diukur. Pengujian ini dilakukan untuk melihat sejauh mana alat ukur dapat dipercaya atau diandalkan. Alat ukur dikatakan reliabel jika menghasilkan hasil yang sama meskipun dilakukan pengukuran berkali-kali (Ernawati & Sukardiyono, 2017; Rosita dkk., 2021). Uji reliabilitas ini dilakukan setelah alat ukur melewati pengujian validitas dan dinyatakan valid. Pengukuran ini akan menggunakan metode Cronbach's Alpha yang menggunakan rumus seperti pada persamaan 3.

$$r = \left(\frac{k}{k-1} \right) \left(\frac{\sum \sigma_b^2}{\sigma_t^2} \right) \quad (3)$$

Di mana:

r = koefisien reliabilitas

k = jumlah butir pertanyaan sah

$\sum \sigma_b^2$ = jumlah varian butir

σ_t^2 = jumlah skor total

(Rosita dkk., 2021).

2.2.12. Uji Kegunaan (Usability)

Aspek *Usability* atau aspek kegunaan adalah aspek yang akan mengukur kemampuan produk sejauh mana produk dapat dipahami dan dipelajari oleh pengguna. *Usability* ini merupakan atribut yang berhubungan dengan upaya yang diperlukan ketika pengguna mulai menggunakan perangkat yang dibuat atau dihasilkan. *Usability* menggambarkan seberapa puas pengguna dalam mengoperasikan perangkat. Ada empat sub aspek yang akan dilihat dari sisi *usability*, yaitu: (1) sub *understandability* yang akan mengukur kemampuan perangkat untuk dapat dipahami dengan mudah; (2) Sub *learnability* yang akan mengukur kemampuan perangkat lunak untuk dapat dipelajari dengan mudah; (3) Sub *operability* yang akan mengukur kemampuan perangkat untuk dapat dioperasikan dengan mudah; dan (4) sub *attractiveness* yang akan mengukur kemampuan perangkat lunak untuk dapat tampil menarik bagi pengguna (Ernawati & Sukardiyono, 2017; Kusuma dkk., 2016; Sufandi dkk., 2022).

Untuk menghitung kegunaan dari sebuah perangkat berdasarkan kuesioner akan melalui rumus pada persamaan 4.

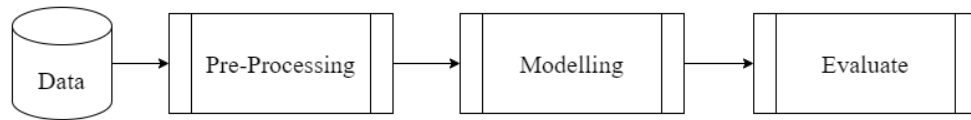
$$usability = \frac{\text{skor yang diobservasi}}{\text{skor yang diharapkan}} \times 100 \quad (4)$$

(Sufandi dkk., 2022).

BAB III

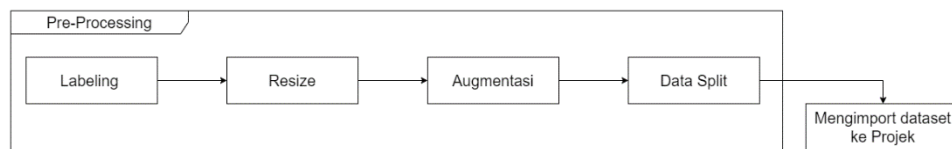
METODE PENELITIAN

3.1. Gambaran Umum Penelitian



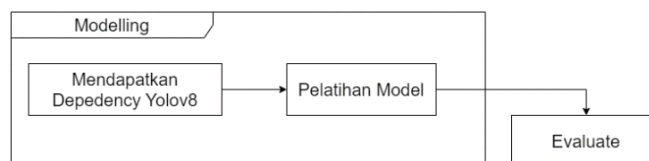
Gambar 3. 1: Flowchart penelitian

Secara garis besar, alur dari penelitian yang akan dilakukan menggunakan bahasa pemrograman *python* ini dapat dilihat dari *flowchart* 3.3. Dari *flowchart* tersebut, terdapat empat tahap utama dalam penelitian ini. Pada penelitian ini setelah data dikumpulkan, selanjutnya akan dilakukan *pre-processing* sebelum dibuat model. Dan setelah model dibuat, akan dilakukan evaluasi terhadap model tersebut. Dalam tahap *pre-procesing* dan modeling dapat di lihat secara rinci pada *flowchart* 3.2 dan 3.1.



Gambar 3. 2: Flowchart pre-processing

Untuk pengolahan dataset dari data mentah akan menggunakan *tools* Roboflow. Hal ini dikarenakan Roboflow dapat membantu peneliti dalam mengolah data mulai dari pelabelan hingga pada *pre-processing* dan augmentasi dataset sehingga dapat mempermudah sekaligus mempercepat proses *pre-processing*.

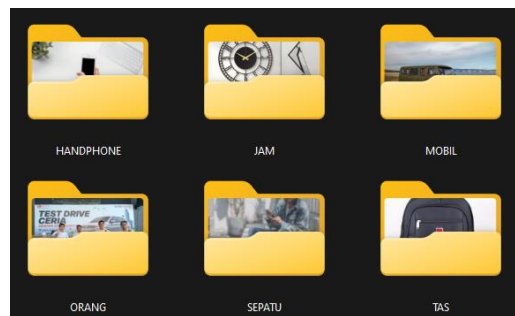


Gambar 3. 3: Flowchart Modelling

Selanjutnya, detail dari keempat tahapan di atas dijelaskan pada poin-poin berikutnya.

3.2. Data

Dataset yang digunakan dalam penelitian ini adalah kumpulan gambar sesuai dengan kelas-kelas yang diambil. Adapun kelas-kelas atau objek yang akan digunakan dalam penelitian ini berupa *handphone*, mobil, orang, tas, ransel, dan jam. Pengambilan kelas-kelas atau objek ini dikarenakan kelas-kelas atau objek ini sering ditemukan di antara atau sekitar kita sehingga mempermudah dalam proses eksplorasi dan dalam pengenalan lingkungan sekitar.



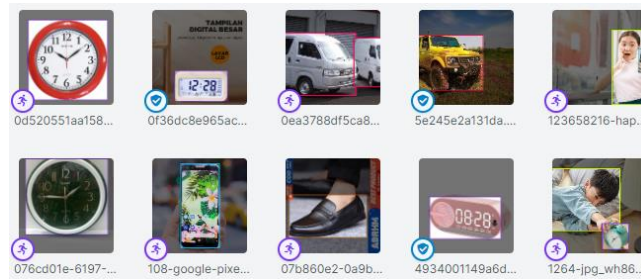
Gambar 3. 4: Gambar kumpulan dataset sesuai kelas

Sumber utama dari pengambilan gambar-gambar ini adalah internet. Gambar nantinya akan diambil secara *random* sesuai dengan kelas-kelasnya. Seperti yang terlihat pada gambar 3.4, data ini nantinya akan digabung menjadi satu *dataset* yang akan diuji. Total data yang digunakan dalam penelitian ini adalah 1200 gambar di mana masing-masing kelas terdiri dari 200 gambar.

3.3. Pre-Processing

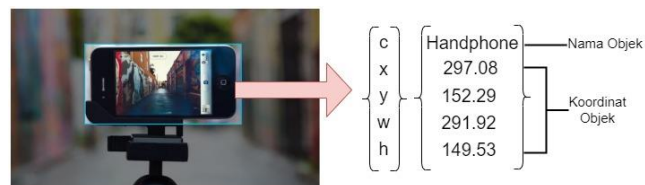
Dalam *pre-processing* gambar yang berguna untuk mengoptimalkan pelatihan model yang nantinya akan digunakan, terdapat beberapa proses.

3.3.1. Labeling Gambar



Gambar 3. 5: Contoh gambar yang telah dilakukan labeling

Labeling adalah tahap awal untuk menentukan objek dari gambar yang mau dilatih. Dalam proses ini, tiap gambar akan dibuat *bounding box* dengan menggunakan *roboflow* yang dapat diakses dari link <https://roboflow.com>. Seperti yang sudah dijelaskan pada poin mengenai YOLO, *bounding box* ini berguna untuk menentukan objek-objek dalam sebuah gambar. Setelah dilakukan *bounding box*, maka akan tersimpan data yang mencatat nilai-nilai yang menunjuk keterangan dan koordinat dari objek tersebut seperti yang terlihat pada gambar 3.6.



Gambar 3. 6: Ilustrasi nilai bounding box

3.3.2. Resize Image

Pada tahap ini, gambar akan di-*resize* dengan ukuran 416 x 416. Tahap ini dilakukan agar gambar mempunyai ukuran yang seragam sebelum masuk ke dalam tahap *training* atau pelatihan. Penentuan pilihan ukuran citra ini didasari pada penelitian yang dilakukan oleh Liquan Zhao (Zhao & Li, 2020) yang mendapatkan performa yang bagus dalam mendeteksi objek. Selain itu, ukuran ini

pun sering digunakan pada pelatihan YOLO di beberapa penelitian sebelumnya.



Gambar 3. 7: Contoh hasil rescale

3.3.3. Augmentasi

Dalam augmentasi gambar akan melewati beberapa proses. Proses-proses ini diambil berdasarkan keputusan dari uji coba dengan beberapa sampel gambar. Adapun proses-proses augmentasi tersebut adalah *grayscale*, *exposure* (15%), *rotation* (10%), dan *flip* (horizontal).



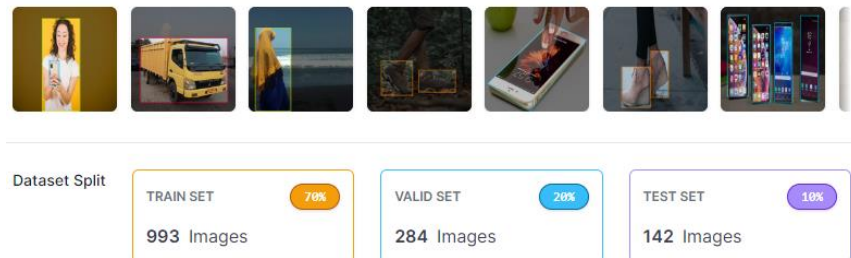
Gambar 3. 8: Contoh augmentasi yang diterapkan pada citra

Proses augmentasi ini bertujuan untuk menambah variasi dan jumlah data. Sehingga dapat mengoptimalkan dalam proses *training* dan menghindari *overfitting*.

3.3.4. Data Split

Pada penelitian ini, *dataset* yang telah melalui proses *resize* dan augmentasi akan masuk ke tahap pembagian dataset. Pada pembagian ini akan dibagi menjadi tiga bagian, yaitu *training*, *validation* dan *testing* dengan pembagian 70:20:10. Dikarenakan

menggunakan Roboflow, maka ketika dataset terbentuk, Roboflow akan secara otomatis membagi dataset menjadi tiga bagian, seperti yang terlihat di gambar 3.9.



Gambar 3. 9: Data split dari Roboflow

3.4. Mengimport Dataset Ke Proyek

Dataset yang telah didapatkan dari internet dan diolah di Roboflow akan diimpor ke dalam proyek. Untuk mengimpornya pun dapat dilakukan dengan dua cara, yaitu dengan menggunakan *code* yang nantinya akan dimasukkan ke dalam proyek yang dibuat, atau dengan cara mendownload langsung datasetnya (Dwyer & Gallagher, 2023).

3.5. Modeling

Dalam proses modeling akan dilakukan beberapa tahap seperti pada gambar 3.3.

3.5.1. Mendapatkan Dependency Yolo Versi 8

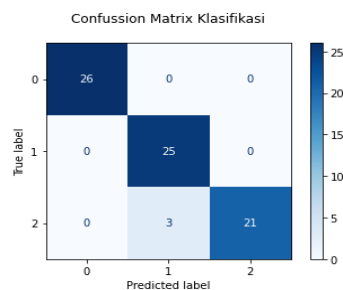
Untuk mendapatkan *dependency* yolo versi 8, pada tahap ini akan dilakukan penginstalan YOLO. Untuk menginstall dapat menggunakan dua cara, yaitu dengan menginstall *library* ultralytics *package* atau dengan menduplikat *repository* yang sudah disiapkan pengembang dan menginstall persyaratannya (Jocher & Sergiuwaxmann, 2023).

3.5.2. Pelatihan Model

Pada tahap ini, dataset yang diimpor dari Roboflow akan dilatih dengan *dependency* dari yolo versi 8 yang sebelumnya sudah didapatkan. Proses ini akan melibatkan beberapa parameter yang dapat mempengaruhi kinerja, kecepatan, dan akurasi model (Jocher & Sergiuwaxmann, 2023).

3.6. Evaluate

Untuk mengetahui hasil pelatihan model pada tahap sebelumnya, maka pada tahap selanjutnya akan dilakukan evaluasi. Evaluasi dilakukan dengan melihat hasil *confusion matrix* yang berguna untuk memantau kesalahan kelas prediksi yang dilakukan model. Selain itu, evaluasi juga dilakukan dengan melihat grafik dari *mean average precision* (mAP) yang dapat menentukan sebaik apa model dapat mengenali objek.



Gambar 3. 10: Contoh confussion matrix

Sumber: <https://haloryan.com/blog/cara-membaca-confusion-matrix>

3.7. Skenario Pengujian

Dalam penelitian ini akan dilakukan beberapa eksperimen dengan melibatkan parameter-parameter yang digunakan di dalam YOLO. Adapun parameter-parameter tersebut adalah *dropout*, *batch*, *learning rate*, dan *optimizer*. Sedangkan untuk datasetnya sendiri akan menggunakan dataset

dengan ukuran gambar 416x416 yang memiliki augmentasi. Penggunaan parameter-parameter ini dilandasi dari beberapa penelitian yang sebelumnya sudah dilakukan dan mendapatkan hasil yang bagus (Gibran dkk., t.t.; Karna dkk., 2023; Li dkk., 2016; Sandhya & Kashyap, 2024; Sholahuddin dkk., 2023).

Secara lebih rinci terkait parameter dan nilai yang digunakan dapat dilihat pada tabel 3.1.

Tabel 3. 1: Tabel Skenario Pengujian

Parameter	Nilai Parameter
Model	Yolov8s.pt
	Yolov8m.pt
Dropout	0.2
	0.5
Batch	32
	64
Learning rate	0.001
	0.0001
Optimizer	Adam
	RMSProp

3.8. Pengujian Kelayakan Kuesioner dan Kegunaan Hasil Akhir

3.8.1. Waktu dan Tempat Pengujian

Pengujian kelayakan hasil akhir ini akan dilakukan melalui kuesioner. Kuesioner ini nantinya akan dibagi menjadi dua bagian. Pertama, kuesioner dibuka untuk dilakukan pengujian validitas dan pengujian reliabilitas terhadap instrumen pengujian. Dan yang kedua, kuesioner akan disebarkan untuk diuji orang tua yang memiliki anak-anak atau guru-guru di kelompok bermain (PAUD)

yang menjadi titik fokus pada penelitian ini. Kuesioner akan dilakukan dengan menyebarkan *link* hasil akhir dan *link* kuesioner ke masyarakat umum secara *online* selama seminggu.

3.8.2. Uji Kelayakan Kuesioner

Uji kelayakan pada hasil penelitian ini berdasarkan aspek kegunaan dari penerapan algoritma You Only Look Once (YOLO) dalam pengenalan lingkungan sekitar bagi anak-anak. Tanggapan dari kuesioner ini akan mewakili empat sub aspek, yaitu pemahaman (*understandability*), pengoperasian (*operability*), daya tarik (*attractiveness*), dan kemampuan belajar (*learnability*). Adapun rincian dari keempat sub aspek tersebut dapat dilihat dari tabel 3.2 (Budyanta, 2018; Ernawati & Sukardiyono, 2017).

Tabel 3. 2: Rincian Pernyataan Kuesioner

No.	Pernyataan	Sub Aspek
1	Sistem kerja website mudah untuk dipahami	<i>Understandability</i>
2	Navigasi atau arahan website ini mudah dipahami	<i>Understandability</i>
3	Website menyediakan informasi yang cukup jelas	<i>Understandability</i>
4	Website berfungsi secara efisien dalam pengenalan objek dilingkungan sekitar	<i>Operability</i>
5	Kesulitan dalam menemukan fitur-fitur yang dibutuhkan	<i>Operability</i>
6	Desain antarmuka dapat dipahami dengan mudah	<i>Attractiveness</i>
7	Penggunaan warna, gambar, dan elemen desain menarik perhatian	<i>Attractiveness</i>
8	Penilaian terhadap desain keseluruhan dari website	<i>Attractiveness</i>
9	Website sangat mudah untuk dipelajari	<i>Learnability</i>
10	Tata cara penggunaan dapat dengan mudah diingat	<i>Learnability</i>

11	Hasil deteksi sesuai dengan objek yang dideteksi	<i>Learnability</i>
----	--	---------------------

Adapun ketentuan yang digunakan adalah ketentuan dari skala likert, sehingga ketentuan penilaian dapat dilihat pada tabel 3.3

Tabel 3. 3: Ketentuan skala likert

Skala	Nilai
1	Sangat Tidak Setuju
2	Tidak Setuju
3	Setuju
4	Sangat Setuju

3.8.2.1. Uji Validitas

Uji validitas akan digunakan untuk mengetahui kelayakan dari butir-butir dalam pertanyaan yang mendefinisikan variabel. Metode uji validitas yang akan digunakan pada penelitian ini adalah metode korelasi produk momen (*corellate bivariate pearson*) dengan 5% untuk rtabel signifikan (Rosita dkk., 2021).

3.8.2.2. Uji Reliabilitas

Berdasarkan poin 2.2.9, maka suatu instrumen dianggap reliabel apabila instrumen tersebut dapat dipercayai sebagai alat ukur dari penelitian. Uji reliabilitas pada penelitian ini akan menggunakan metode Cronbach's Alpha dengan rumus yang sesuai pada persamaan 3. Untuk mengetahui tinggi rendahnya reliabilitas dapat menggunakan kategori seperti pada tabel 3.4

Tabel 3. 4: Tingkat reliabilitas Cronbach's Alpha

Interval Reliabilitas	Kategori
$0,80 < r \leq 1,00$	Reliabilitas sangat tinggi
$0,60 < r \leq 0,80$	Reliabilitas tinggi
$0,40 < r \leq 0,60$	Reliabilitas sedang
$0,20 < r \leq 0,40$	Reliabilitas rendah
$0,00 < r \leq 0,20$	Tidak reliabel

(Ernawati & Sukardiyono, 2017).

3.8.3. Analisis Kegunaan (*Usability*)

Setelah mengetahui bahwa instrumen yang hendak diukur adalah valid dan reliabilitas, maka akan dilakukan pengukuran kegunaan atau *usability*. Pengukuran *usability* akan dilakukan dengan menghitung presentasi jawaban dari responden dengan menggunakan rumus pada persamaan 4. Setelah data diperoleh, maka data akan dikonversi ke dalam kategorinya berdasarkan pada tabel 3.5.

Tabel 3. 5: Kategori Kegunaan

Data (%)	Kategori
< 21	Sangat tidak layak
21 – 40	Tidak layak
41 – 60	Cukup
61 – 80	Layak
81 – 100	Sangat Layak

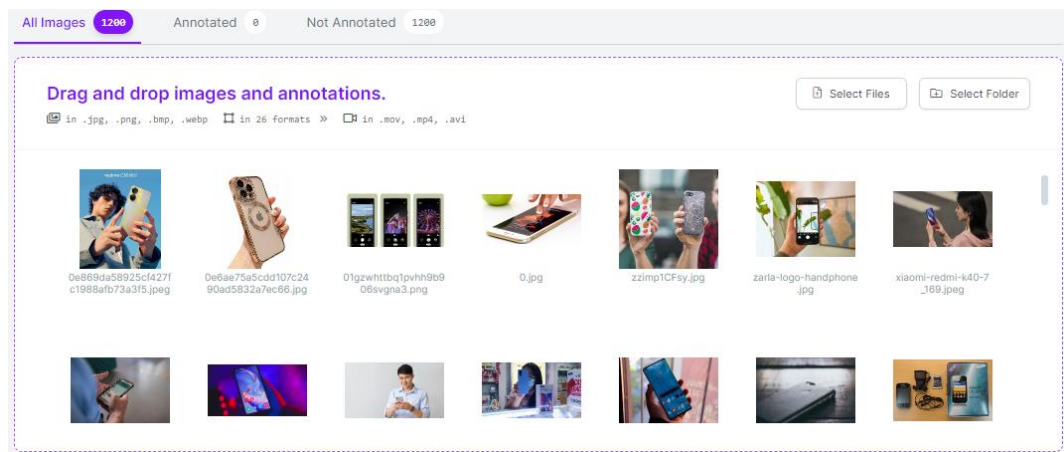
(Kusuma dkk., 2016).

BAB IV

HASIL PENELITIAN DAN PEMBAHASAN

4.1. Pengumpulan Data

Data yang dikumpulkan adalah data yang diambil dari internet berupa gambar dan dipilih secara *random*. Data yang dikumpulkan ini kemudian dikelompokkan sesuai kelas-kelas yang nantinya akan digunakan pada penelitian seperti pada gambar 3.3. Setelah terkumpul, data akan dimasukkan ke dalam *workshop* yang sebelumnya sudah dibuat di Roboflow seperti yang dilihat pada gambar 4.1. Dengan demikian, data akan siap masuk pada tahap *pre-processing*.

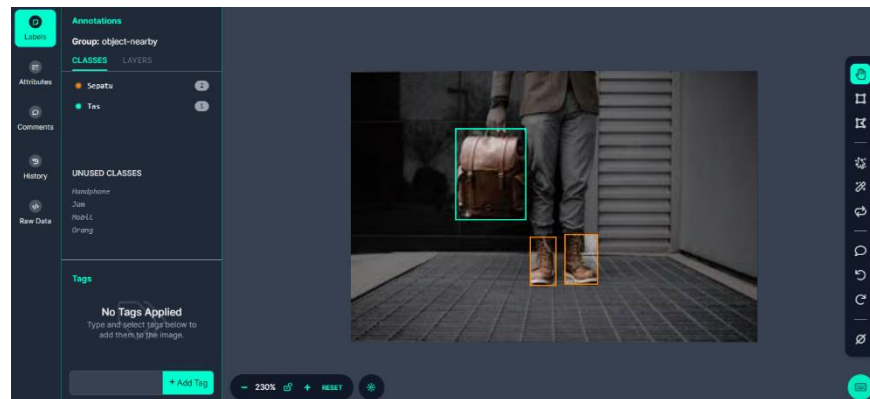


Gambar 4. 1: Proses upload data ke Roboflow

4.2. *Pre-processing* Data

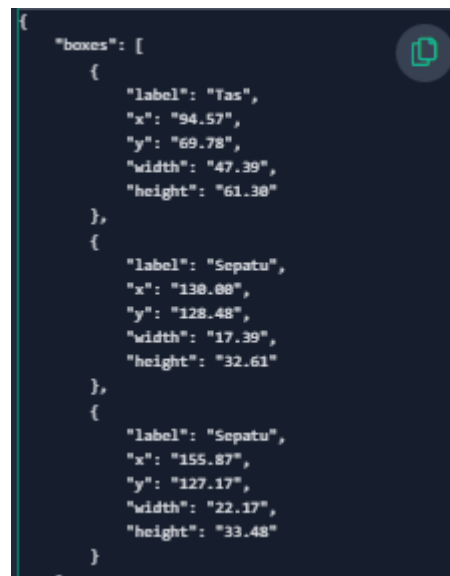
Tahap *pre-processing* pada data akan melalui beberapa tahap seperti yang telah disebutkan pada bab 3. Pengolahan dataset ini akan dikerjakan menggunakan *tools* Roboflow mulai dari pelabelan gambar, *resize* gambar, augmentasi gambar, dan pembagian data.

4.2.1. Labeling Gambar



Gambar 4. 2: Proses annotate di Roboflow

Pada tahap ini, gambar yang sebelumnya sudah diunggah ke *workshop* Roboflow akan dilakukan *annotate* atau pelabelan objek yang terdapat di dalam gambar. Pelabelan pada sebuah gambar tidak terpaku hanya pada satu objek. Jika di dalam gambar terdapat lebih dari satu objek yang jelas, maka objek tersebut akan dilabeli. Hal ini dapat terlihat seperti pada gambar 4.2.



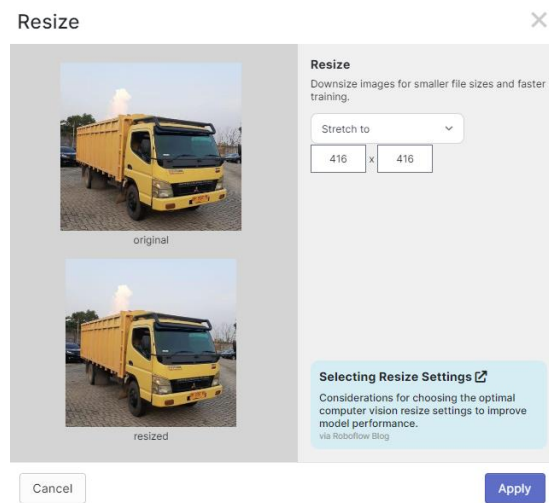
Gambar 4. 3: Data hasil labeling atau bounding box

Seperti yang dijelaskan pada bagian 3.3.1 mengenai *labeling*, label atau *bouding box* yang dibuat ini akan menyimpan informasi-

informasi yang dibutuhkan algoritma untuk proses pemodelan. Adapun data yang tersimpan seperti yang terlihat pada gambar 4.3. Data tersebut memuat mengenai *class* dari objek yang dilabel dan koordinat dari objek tersebut.

4.2.2. *Rescale / Resize Image*

Setelah dilabel, gambar akan masuk pada tahap selanjutnya, yaitu *resize*. *Resize* ini membuat semua gambar pada versi yang dipilih akan memiliki ukuran yang sama. Hal ini dilakukan agar dapat membantu meningkatkan efisiensi dalam pengolahan data dan penggunaan sumber daya komputasi pada saat pemodelan nantinya (Iryanto & Zaini, 2014; Zhao & Li, 2020).



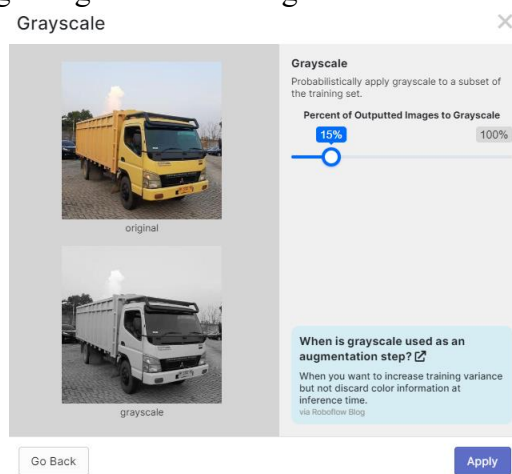
Gambar 4. 4: Proses resize pada dataset

4.2.3. *Augmentasi Image*

Pada tahap ini akan dilakukan proses augmentasi seperti yang sudah dijelaskan pada bab 3.

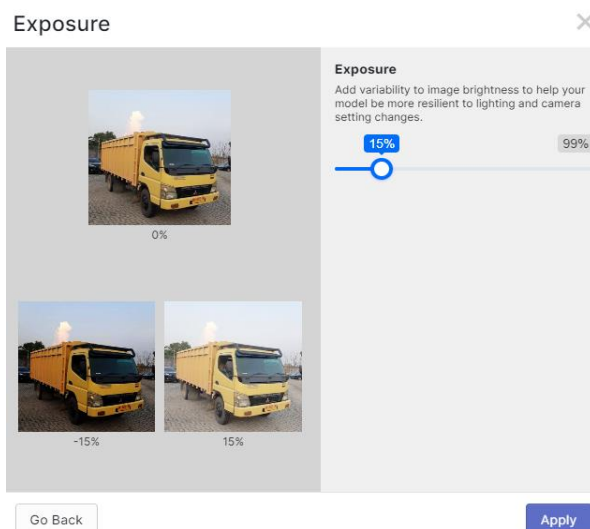
4.2.3.1. *Grayscale*

Pada augmentasi untuk *grayscale*, akan dipilih tingkat *grayscale* sebesar 15%. Tingkatan ini dapat membuat variasi dataset yang signifikan jika dibandingkan dengan tingkatan di bawahnya juga tidak membuat gambar terlalu gelap sehingga dapat menghilangkan detail dari gambar.



Gambar 4. 5: Proses augmentasi dataset

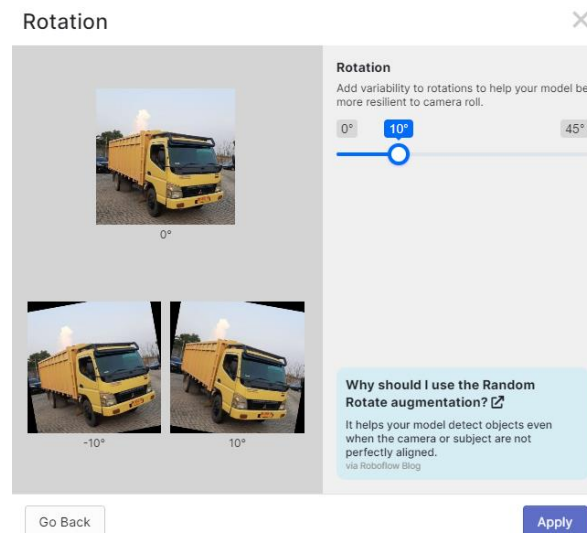
4.2.3.2. *Exposure*



Gambar 4. 6: Proses exposure dataset

Untuk bagian *exposure* dataset akan menggunakan tingkatan sebesar 15% sehingga membuat dataset menjadi lebih gelap dan lebih terang dengan nilai 15%.

4.2.3.3. *Rotation*

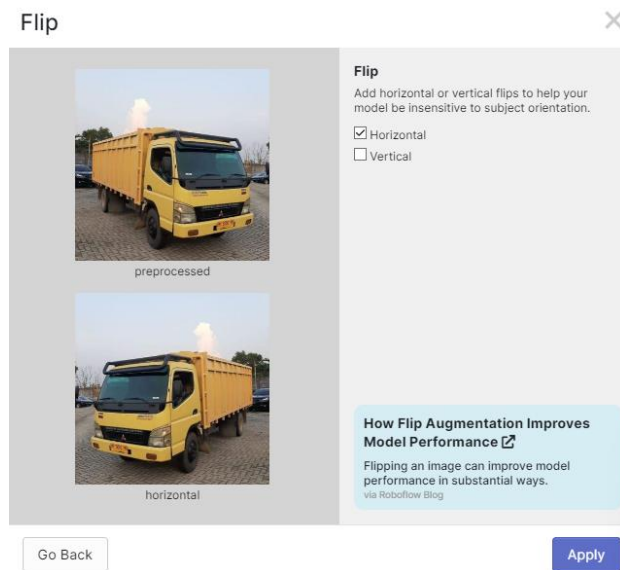


Gambar 4. 7: Proses rotation dataset

Pada tahap ini dataset akan dilakukan proses *rotation* dengan tingkatan sebesar 10%. Hal ini dilakukan agar terdapat beberapa variasi dataset yang memiliki titik yang berbeda dari gambar aslinya. *Rotation* ini akan membuat gambar memutar dengan sejumlah derajat tertentu (Perez dkk., 2018).

4.2.3.4. *Flip*

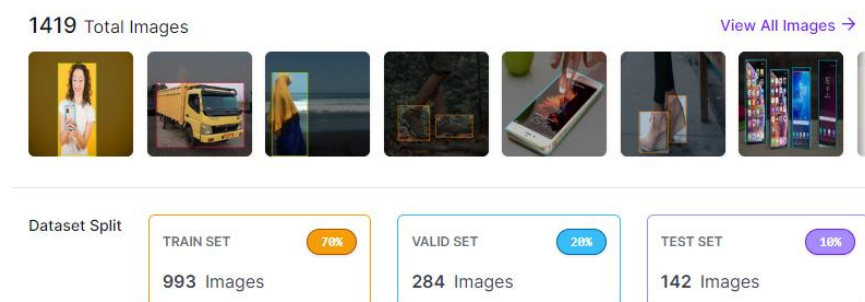
Proses ini nantinya akan membalikan gambar secara horizontal. Proses ini digunakan agar dataset memiliki variasi terhadap data dari beberapa sudut pandang.



Gambar 4. 8: Proses flip dataset

4.2.4. *Data Split*

Pada tahap *data split*, setelah tahapan *resize* dan augmentasi dilakukan, maka secara otomatis Roboflow akan menghitung total gambar kemudian membaginya ke dalam bagiannya masing-masing. Untuk pembagiannya akan menerapkan perbandingan 70:20:10 dengan rincian 70% untuk *train*, 20% untuk *valid*, dan 10% untuk *testing* seperti yang dijelaskan pada bagian sebelumnya.



Gambar 4. 9: Proses pembagian dataset

Ditahap ini dataset telah selesai diolah dan siap digunakan ke dalam proyek yang akan dibuat untuk digunakan dalam pelatihan model.

4.3. Impor Dataset

```
!pip install roboflow
from roboflow import Roboflow

rf = Roboflow(api_key="A88agLIXim30CSF5vN0e")
project = rf.workspace("gabriel-advent-hobto").project(
    "detect-tdz36")
version = project.version(1)
dataset = version.download("yolov8")
```

Gambar 4. 10: Code untuk mengimpor dataset dari Roboflow

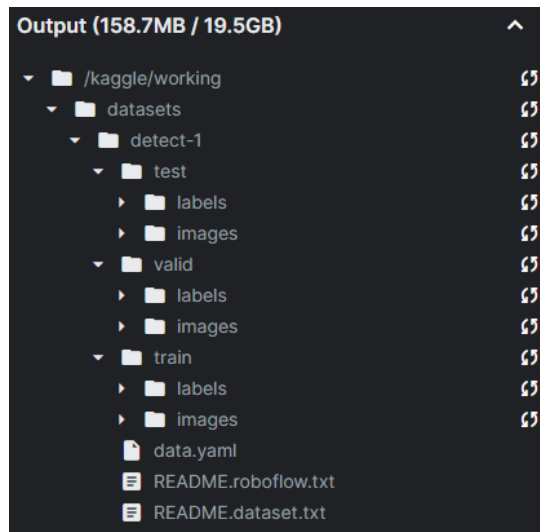
Dataset yang sebelumnya sudah diolah akan diimpor ke dalam proyek untuk membantu pelatihan model dengan *custom* dataset. Untuk mengimpor dataset, Roboflow sendiri telah memberikan kemudahan dengan *code* yang sudah disiapkan seperti yang dapat dilihat dari gambar 4.10. *Code* yang diberikan ini nantinya akan mengunduh dalam bentuk zip dan mengekstrak dataset tersebut agar bisa dipakai. Jika dataset yang diimpor berhasil diunduh dan diekstrak dari bentuk zip-nya maka pada *output* yang dihasilkan akan terlihat seperti pada gambar 4.11.

```
loading Roboflow workspace...
loading Roboflow project...
Dependency ultralytics==8.0.196 is required but found version=8.2.5, to fix: `pip install ultralytics==8.0.196`
Downloading Dataset Version Zip in detect-1 to yolov8:: 100%|██████████| 143885/143885 [00:03<00:00, 41629.65 it/s]

Extracting Dataset Version Zip to detect-1 in yolov8:: 100%|██████████| 6822/6822 [00:00<00:00, 7276.67it/s]
```

Gambar 4. 11: Keterangan jika berhasil mengunduh dan mengekstrak dataset

Setelah berhasil diekstrak, dapat dilihat pada gambar 4.12 bahwa Roboflow telah membagikan dataset ke dalam foldernya masing-masing sesuai pembagian yang ada di bagian 4.2.4 sebelumnya mengenai *data split*. Pembagian dataset ini juga telah dilengkapi dengan label yang memiliki *extension* .txt. File inilah yang berisi data-data objek dari gambar sesuai pelabelan yang telah dilakukan. Selain itu, Roboflow juga telah membuat *file* data.yaml. *File* ini digunakan untuk mengonfigurasi dataset. Sehingga *file* data.yaml ini akan berisi informasi mengenai *path* dataset, kelas-kelas yang



Gambar 4. 12: Hasil ekstrasi dataset

digunakan dalam model nantinya, dan beberapa informasi yang relevan. Secara lebih jelas terkait isi dari *file* data.yaml, dapat dilihat dari gambar 4.13.

```
train: ../train/images
val: ../valid/images
test: ../test/images

nc: 6
names: ['Handphone', 'Jam', 'Mobil', 'Orang', 'Sepatu', 'Tas']

roboflow:
  workspace: gabriel-advent-hobto
  project: detect-tdz36
  version: 1
  license: CC BY 4.0
  url: https://universe.roboflow.com/gabriel-advent-hobto/detect-tdz36/dataset/1
```

Gambar 4. 13: Isi dari file data.yaml

4.4. Modeling

Proses pembuatan model dari YOLO ini nantinya akan dibuat dengan bantuan Kaggle Notebook. Hal ini dikarenakan platform ini menyediakan akses GPU dengan *limit* 30 jam setiap minggunya. Meskipun ada batasan, fitur ini sangat menguntungkan karena mampu menjalankan program yang membutuhkan GPU di dalamnya (A. Y. Wang dkk., 2021).

4.4.1. Mendapatkan Dependency YOLO

```
!pip install ultralytics
```

Gambar 4. 15: Code untuk menginstall library

Proses pembuatan model yang pertama kali dilakukan adalah dengan mendapatkan *dependency* YOLO terbaru yang sudah disiapkan oleh Ultralytics. Untuk mendapatkannya dapat dengan menginstall *library* yang sudah disiapkan seperti pada gambar 4.15. Setelah mendapatkan *dependency* dari Ultralytics, selanjutnya akan mengimpor YOLO ke dalam proyek yang dibuat dengan menjalankan *code* seperti pada gambar 4.14. Dengan menjalankan kedua *code* tersebut, YOLO telah siap digunakan ke dalam proyek untuk pembuatan model.

```
from ultralytics import YOLO
```

Gambar 4. 14: Mengimpor YOLO ke dalam proyek

4.4.2. Pelatihan Model

Pada pelatihan model ini akan dilakukan eksperimen-eksperimen berdasarkan tabel skenario pengujian. Adapun total keseluruhan eksperimen ini adalah 32 eksperimen.

```
import subprocess
from itertools import product

param_grid = {
    'model': ['yolov8s.pt', 'yolov8m.pt'],
    'dropout': [0.2, 0.5],
    'batch': [32, 64],
    'learning_rate': [0.001, 0.0001],
    'optimizer': ['adam', 'RMSProp'],
}

# Generate semua kombinasi
param_combinations = list(product(*param_grid.values()))
```

Gambar 4. 16: Code untuk mengombinasikan parameter

32 eksperimen ini merupakan hasil kombinasi yang dilakukan seperti pada *code* di gambar 4.16. Adapun rincian kombinasi tersebut dapat dilihat pada tabel 4.1.

Tabel 4. 1: Daftar kombinasi parameter

No	Model	Dropout	Batch	Learning Rate	Optimizer
1	yolov8m	0,2	32	0,001	Adam
2	yolov8m	0,2	32	0,001	RMSProp
3	yolov8m	0,2	32	0,0001	Adam
4	yolov8m	0,2	32	0,0001	RMSProp
5	yolov8m	0,2	64	0,001	Adam
6	yolov8m	0,2	64	0,001	RMSProp
7	yolov8m	0,2	64	0,0001	Adam
8	yolov8m	0,2	64	0,0001	RMSProp
9	yolov8m	0,5	64	0,001	Adam
10	yolov8m	0,5	32	0,001	RMSProp
11	yolov8m	0,5	32	0,0001	Adam
12	yolov8m	0,5	32	0,0001	RMSProp
13	yolov8m	0,5	64	0,001	Adam
14	yolov8m	0,5	64	0,001	RMSProp
15	yolov8m	0,5	32	0,0001	Adam
16	yolov8m	0,5	64	0,0001	RMSProp
17	yolov8s	0,2	32	0,001	Adam
18	yolov8s	0,2	32	0,001	RMSProp
19	yolov8s	0,2	32	0,0001	Adam
20	yolov8s	0,2	32	0,0001	RMSProp
21	yolov8s	0,2	64	0,001	Adam
22	yolov8s	0,2	64	0,001	RMSProp
23	yolov8s	0,2	64	0,0001	Adam
24	yolov8s	0,2	64	0,0001	RMSProp
25	yolov8s	0,5	64	0,001	Adam
26	yolov8s	0,5	32	0,001	RMSProp
27	yolov8s	0,5	32	0,0001	Adam
28	yolov8s	0,5	32	0,0001	RMSProp
29	yolov8s	0,5	64	0,001	Adam
30	yolov8s	0,5	64	0,001	RMSProp
31	yolov8s	0,5	32	0,0001	Adam
32	yolov8s	0,5	64	0,0001	RMSProp

Pada pelatihan ini akan menggunakan bantuan model yang telah dilatih oleh Ultralytics. Hal ini bertujuan untuk membantu pembuatan model baru sehingga pada saat pelatihan atau pembuatan model tidak memakan waktu dan daya komputasi yang berlebihan (Jocher & Sergiuwaxmann, 2023). Dan untuk jenis model yang digunakan adalah model dengan ukuran *small* (yolov8s) dan ukuran *medium* (yolov8m) karena ukuran gambar 416x416 termasuk dalam kategori kecil hingga menengah (Banovbi dkk., 2022; Li dkk., 2016). Selanjutnya pelatihan ini akan menggunakan 200 *epoch* dan menerapkan mekanisme *early stopping* dengan mengatur parameter *patience* ke 50. Hal ini memungkinkan pelatihan berhenti lebih awal untuk menghemat waktu komputasi jika model tidak menunjukkan perbaikan dalam jangka waktu tersebut. Secara lebih lanjut pelatihan ini akan dijalankan dengan *code* pada gambar 4.17.

```

commands = []
for params in param_combinations:
    model, dropout, batch, lr, optimizer = params
    command = f"yolo detect train \
data='/kaggle/working/datasets/detect-2/data.yaml' \
model='{model}' \
epochs=200 \
imgsz='416' \
optimizer='{optimizer}' \
lr0='{lr}' \
batch='{batch}' \
dropout='{dropout}' \
plots=True \
save=True"
    commands.append(command)

import subprocess

# Loop setiap perintah dan jalankan menggunakan subprocess
for command in commands:
    print(command)
    subprocess.run(command, shell=True)

```

Gambar 4. 17: Code untuk melakukan pelatihan model pada YOLO

4.5. Analisis Hasil Pengujian

Tabel 4. 2: Hasil pelatihan model

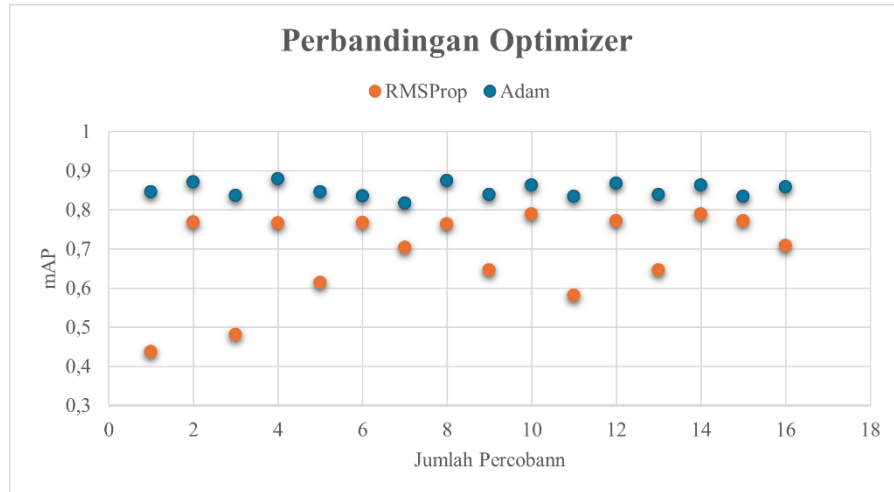
No	Model	Dropout	Batch	Learning Rate	Optimizer	mAP
1	yolov8m	0,2	32	0,001	Adam	0,84499
2	yolov8m	0,2	32	0,001	RMSProp	0,43691
3	yolov8m	0,2	32	0,0001	Adam	0,87112
4	yolov8m	0,2	32	0,0001	RMSProp	0,7682
5	yolov8m	0,2	64	0,001	Adam	0,83652
6	yolov8m	0,2	64	0,001	RMSProp	0,48063
7	yolov8m	0,2	64	0,0001	Adam	0,87916
8	yolov8m	0,2	64	0,0001	RMSProp	0,76636
9	yolov8m	0,5	64	0,001	Adam	0,84499
10	yolov8m	0,5	32	0,001	RMSProp	0,6132
11	yolov8m	0,5	32	0,0001	Adam	0,83462
12	yolov8m	0,5	32	0,0001	RMSProp	0,7672
13	yolov8m	0,5	64	0,001	Adam	0,81623
14	yolov8m	0,5	64	0,001	RMSProp	0,70356
15	yolov8m	0,5	32	0,0001	Adam	0,87481
16	yolov8m	0,5	64	0,0001	RMSProp	0,7635
17	yolov8s	0,2	32	0,001	Adam	0,8386
18	yolov8s	0,2	32	0,001	RMSProp	0,64615
19	yolov8s	0,2	32	0,0001	Adam	0,86293
20	yolov8s	0,2	32	0,0001	RMSProp	0,78881
21	yolov8s	0,2	64	0,001	Adam	0,83396
22	yolov8s	0,2	64	0,001	RMSProp	0,58158
23	yolov8s	0,2	64	0,0001	Adam	0,86755
24	yolov8s	0,2	64	0,0001	RMSProp	0,77183
25	yolov8s	0,5	64	0,001	Adam	0,8386
26	yolov8s	0,5	32	0,001	RMSProp	0,64615
27	yolov8s	0,5	32	0,0001	Adam	0,86293
28	yolov8s	0,5	32	0,0001	RMSProp	0,78881
29	yolov8s	0,5	64	0,001	Adam	0,83396
30	yolov8s	0,5	64	0,001	RMSProp	0,77103
31	yolov8s	0,5	32	0,0001	Adam	0,85842
32	yolov8s	0,5	64	0,0001	RMSProp	0,70814

Dari hasil eksperimen-eksperimen pada tabel 4.2 tersebut, diketahui bahwa percobaan yang mendapatkan hasil tertinggi terdapat pada percobaan ketujuh yang menghasilkan mAP sebesar 0,87916 atau sekitar 87% dan dengan

kombinasi parameter model jadi yang digunakan adalah YOLOv8m, *dropout* sebesar 0,2, *batch size* 64, *learning rate* 0,0001, dan *optimizer* Adam. Rincian ini dapat dilihat pada tabel 4.3.

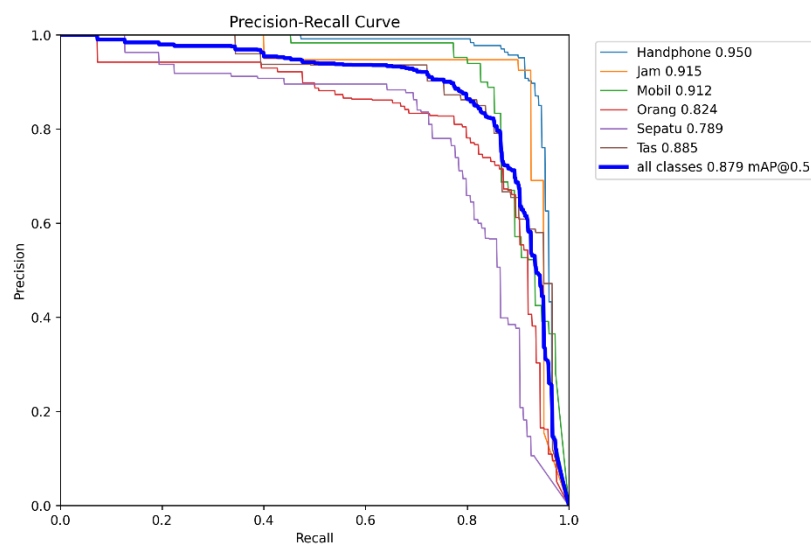
Tabel 4. 3: Kombinasi yang menghasilkan mAP terbaik

Model	Dropout	Batch	Learning Rate	Optimizer	mAP
yolov8m	0,2	64	0,0001	Adam	0,87916



Gambar 4. 19: Perbandingan optimizer

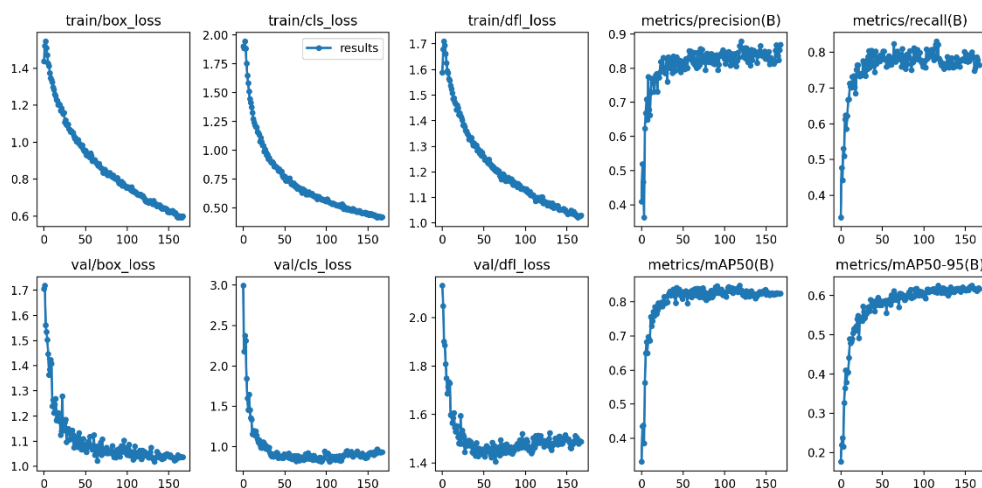
Selain itu, dapat dilihat bahwa *optimizer* Adam yang digunakan juga membantu memberikan hasil yang maksimal dibandingkan dengan *optimizer* RMSProp. Hal ini ditunjukkan dengan rata-rata mAP yang diberikan oleh



Gambar 4. 18: Grafik precision dan recall

optimizer Adam sebesar 0.8506 sedangkan rata-rata mAP yang diberikan oleh *optimizer* RMSProp adalah 0,6876. Hal ini pun dapat dilihat pada gambar grafik perbandingan 4.19.

Selanjutnya pada gambar grafik 4.18, dapat dilihat mAP untuk tiap objek memiliki nilai di 80% kecuali objek sepatu yang memiliki mAP sebesar 78%. Dan dengan nilai rata-rata mAP sebesar 0.879 atau 87% ini mempunyai performa yang baik dalam mendeteksi objek (Jönsson Hyberg & Sjöberg, 2023; Terven & Cordova-Esparza, 2023). Hal ini pun kembali ditekankan pada kumpulan grafik pada gambar 4.20.

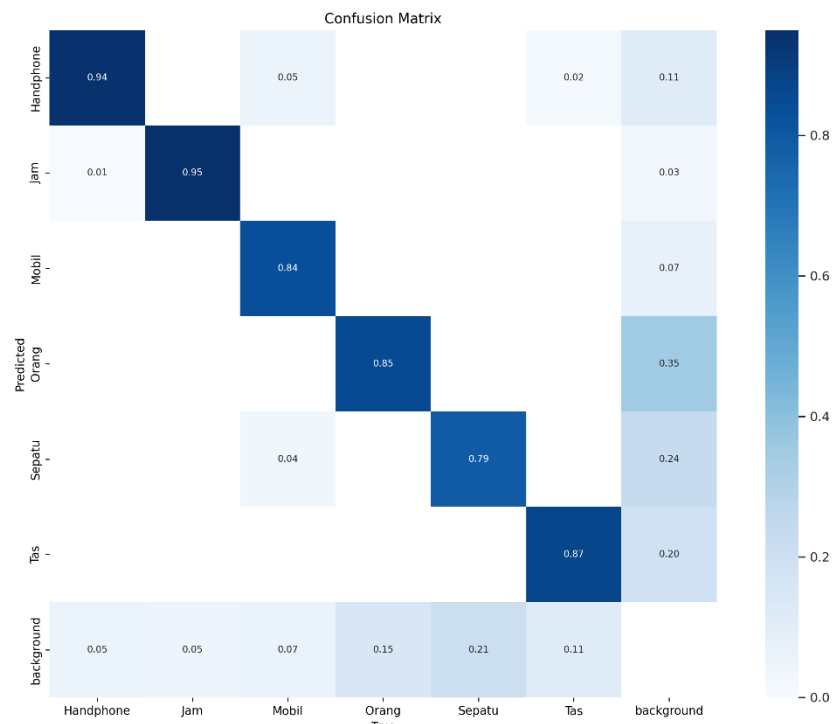


Gambar 4. 20: Kumpulan grafik hasil pelatihan model

Pada grafik tersebut dapat dilihat bahwa model menunjukkan performa yang baik dengan ditunjukkan pada grafik *loss* yang semakin menurun dalam proses pelatihan, grafik *precision*, *recall*, dan mAP yang tinggi. Kombinasi dari penurunan nilai *loss*, nilai *precision*, *recall* dan nilai mAP yang tinggi menunjukkan bahwa model mampu mendeteksi objek dengan akurat dan efisien.

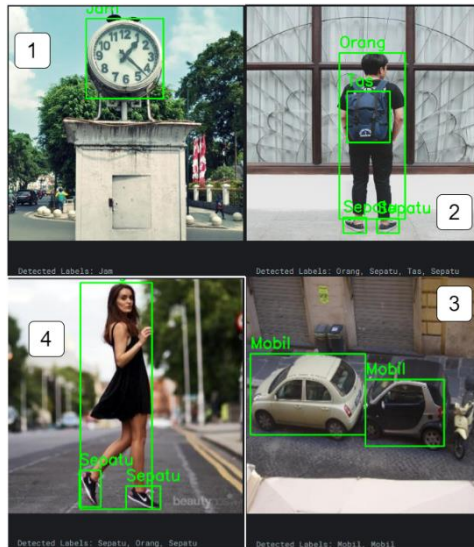
Berikutnya akan ditampilkan hasil prediksi data validasi oleh model pada percobaan ketujuh. Berdasarkan hasil *confusion matrix* pada gambar 4.21,

dapat dipastikan bahwa model dapat memprediksi objek dengan hasil yang baik, seperti pada objek *handphone*, jam, mobil, orang, dan tas. Namun dibalik itu, model ini kurang mampu untuk mendeteksi objek sepatu yang sering dideteksi sebagai *background*. Hal inilah yang cukup menjelaskan kenapa mAP pada objek sepatu di gambar 4.18 lebih kecil dibandingkan dengan objek-objek lainnya.



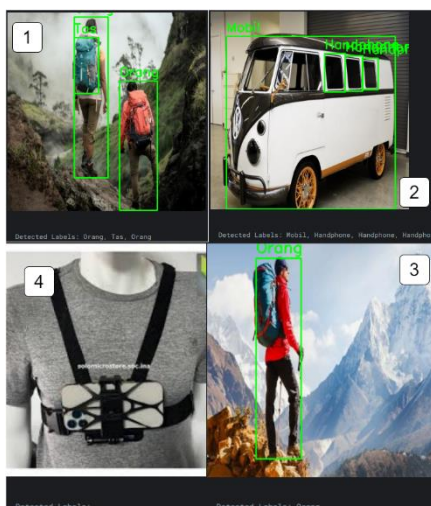
Gambar 4. 21: Confusion matrix percobaan ketujuh

Hasil model yang baik ini pun didukung dengan hasil pengujian pada data *test* yang berjumlah 142 gambar. Dari 142 gambar ini sebagian besar gambar dapat dideteksi dengan benar, namun ada beberapa juga gambar yang bisa dideteksi dengan kurang benar oleh model.



Gambar 4. 23: Gambar yang diprediksi dengan benar

Seperti yang terlihat pada gambar 4.23, model dapat mendeteksi objek pada gambar dengan benar selama gambar yang diprediksi memiliki kualitas yang bagus atau gambar yang jelas. Dengan kualitas yang kurang bagus atau dengan objek yang kurang terlihat jelas mampu membuat model salah atau tidak dapat memprediksi objek seperti yang ada pada gambar 4.21 khususnya pada gambar 4. Selain itu, terlihat bahwa model masih mendeteksi objek yang memiliki bentuk atau fitur yang serupa seperti pada gambar mobil pada nomor



Gambar 4. 22: Gambar yang kurang tepat diprediksi

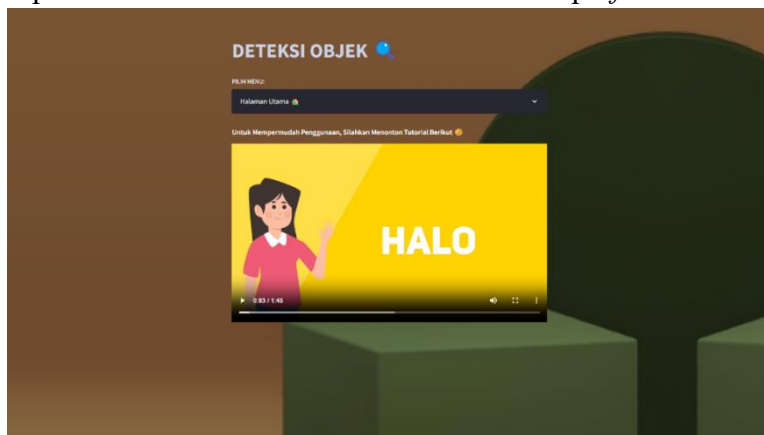
2. Kemudian model pun kurang mampu mendeteksi sepatu yang sering dianggap sebagai *background*.

Berdasarkan hasil pelatihan, validasi dan testing model, maka model pada skenario atau percobaan ketujuh akan digunakan dalam *website* deteksi objek sebagai bentuk implementasi hasil akhir.

4.6. Implementasi Aplikasi Deteksi Objek

4.6.1. *Interface* Aplikasi

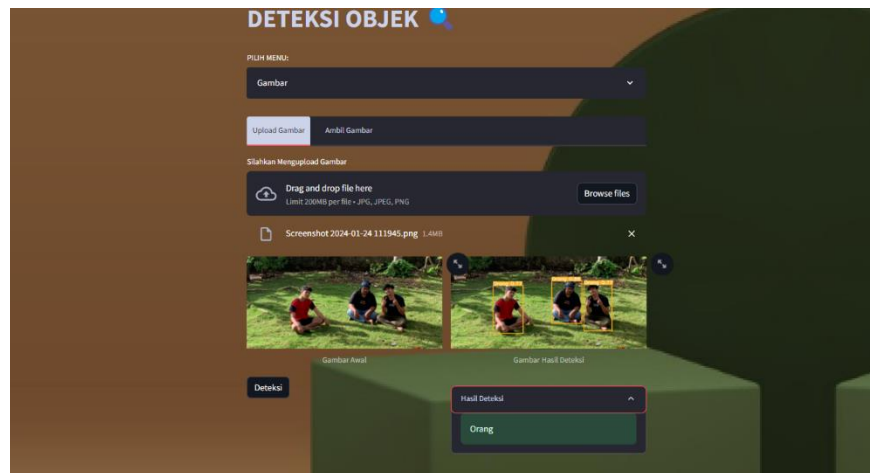
Implementasi model ke dalam *interface* akan dibantu dengan *framework* Streamlit. Alasan mengapa menggunakan *framework* ini adalah karena pihak Streamlit telah menyediakan *cloud* khusus bagi para anggota komunitas untuk *deploy* hasil kerjanya (Sholahuddin dkk., 2023). Pada *website* ini akan terdapat beberapa fitur utama. Fitur-fitur tersebut antara lain mendeteksi gambar dengan mengunggah gambar atau dengan mengambilnya secara langsung, mendeteksi video dengan mengunggah video atau dengan beberapa video yang sudah disiapkan sebelumnya, mendeteksi dengan memasukan *link* YouTube, dan mendeteksi secara *realtime*. Pengimplementasian ini akan melibatkan beberapa *function* untuk



Gambar 4. 24: Tampilan halaman pertama ketika website diakses

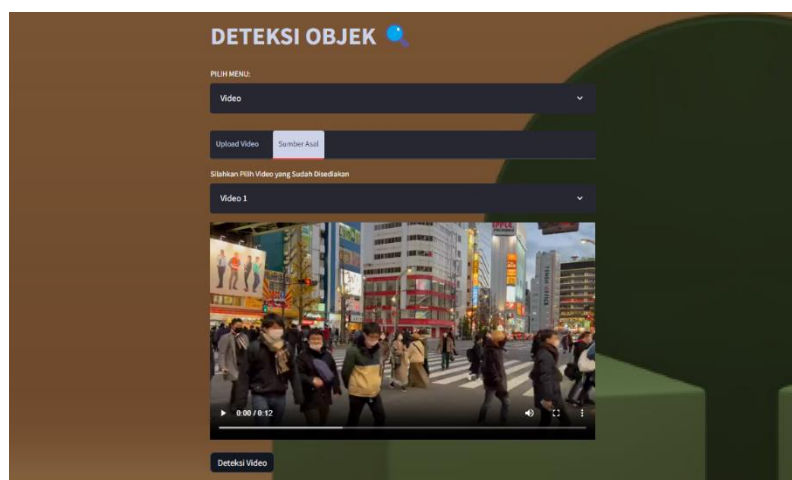
mempermudah proses deteksi dengan model yang sudah dibuat sebelumnya.

Pada gambar 4.25 adalah *interface* yang dibuat di Streamlit. Ketika pertama kali diakses. Pada halaman ini akan menampilkan *dropdown* dari menu-menu yang disediakan juga menampilkan video tutorial penggunaan *website*.



Gambar 4. 25: Tampilan halaman jika memilih menu gambar

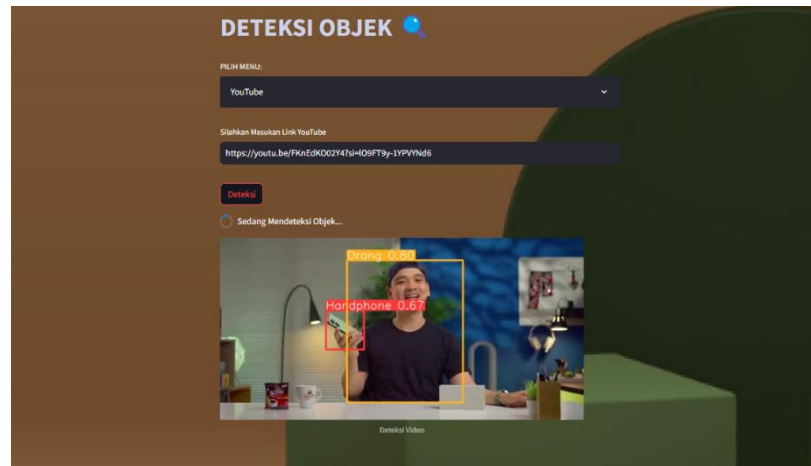
Berikutnya, tampilan pada gambar 4.24 adalah tampilan jika memilih menu gambar dan telah melakukan deteksi dengan mengunggah foto atau gambar. Dapat dilihat bahwa di bawah *dropdown* menu terdapat *tab* untuk memilih ingin menggunakan



Gambar 4. 26: Tampilan jika memilih menu video

fitur yang sama. Hal ini pun berlaku pada tampilan jika memilih menu video seperti yang terlihat pada gambar 4.26.

Pada menu ini telah disediakan lima video yang dapat langsung digunakan untuk deteksi sehingga user tidak perlu mengunggah videonya sendiri.



Gambar 4. 27: Tampilan jika memilih menu youtube

Selanjutnya pada gambat 4.27 dapat dilihat tampilan jika memilih menu youtube. Pada menu ini, user hanya perlu memasukan link YouTube yang sudah disalin dan menekan tombol ‘deteksi’. Terakhir adalah tampilan jika user memilih menu Real-Time. Seperti



Gambar 4. 28: Tampilan ketika memilih menu Real-Time

yang terlihat pada gambar 4.29, menu ini akan mendeteksi secara realtime menggunakan kamera baik ponsel maupun komputer.

4.6.2. Implementasi *Code* untuk Deteksi

```
try:
    if source_img is None:
        default_image_path = str(settings.DEFAULT_IMAGE)
        default_image = PIL.Image.open(default_image_path)
        st.image(default_image_path, caption="Gambar Awal",
                  use_column_width=True)
    else:
        uploaded_image = PIL.Image.open(source_img)
        st.image(source_img, caption="Gambar Awal",
                  use_column_width=True)

    # Tombol Detect Objects di sini
    if st.button('Deteksi'):
        res = model.predict(uploaded_image, conf=confidence)
        boxes = res[0].boxes
        res_plotted = res[0].plot()[ :, :, ::-1]

except Exception as ex:
    st.error("Ada Kesalahan Saat Membaca File")
    st.error(ex)
```

Gambar 4. 29: Code untuk mendeteksi gambar yang diupload

Terdapat beberapa *function* yang diimplementasi untuk menunjang proses deteksi di tiap menu. Pada gambar 4.28 adalah *code* yang digunakan untuk mendeteksi gambar yang diunggah oleh user. Pada gambar tersebut, jika terdapat *value* di variabel `source_img` maka akan muncul tombol 'deteksi' yang akan memproses prediksi jika ditekan dan menampilkan hasil prediksi tersebut.

```
def showDetectFrame(conf, model, st_frame, image,
                    is_display_tracking=None, tracker=None):

    # Predict the objects in the image using the YOLOv8 model
    res = model.predict(image, conf=conf)

    # Plot the detected objects on the video frame
    res_plotted = res[0].plot()
    st_frame.image(res_plotted,
                   caption='Detected Video',
                   channels="BGR",
                   )
```

Gambar 4. 30: Function untuk menampilkan gambar hasil deteksi

Berikut pada gambar 4.30 adalah *function* yang digunakan untuk melakukan deteksi dan menampilkan hasil deteksi ke *frame* yang

```
def play_youtube(conf, model):
    source_youtube = st.text_input("Silahkan Masukan Link YouTube")

    if st.button('Deteksi'):
        try:
            yt = YouTube(source_youtube)
            stream = yt.streams.filter(file_extension="mp4", res=720).first()
            vid_cap = cv2.VideoCapture(stream.url)
            st_frame = st.empty()
            while (vid_cap.isOpened()):
                success, image = vid_cap.read()
                if success:
                    showDetectFrame(conf,
                                    model,
                                    st_frame,
                                    image
                                    )
                else:
                    vid_cap.release()
                    break
            except Exception as e:
                st.error("Ada Kesalahan Saat Memproses Link: " + str(e))
```

Gambar 4. 31: Function untuk mencari video dari link youtube

sudah disiapkan sebelumnya. Perlu diperhatikan bahwa nilai *confidence* yang digunakan pada tiap deteksi akan menggunakan *confidence* sebesar 0.5.

Pada gambar 4.31 merupakan *function* untuk mengisi *link* YouTube yang disalin. Ketika tombol deteksi maka program akan *capture* tiap *frame* video dengan bantuan *library* open-cv dan mengirimnya ke *function* showDetectFrame. Selanjutnya terdapat *function* untuk memproses video yang diunggah user. Pada *function* yang dapat dilihat dari gambar 4.32 ini, setelah video diunggah, video akan disimpan pada penyimpanan sementara. Dari hasil penyimpanan ini, video akan diambil gambar tiap *frame* video kemudian mengirimkan hasil *capture* kepada *function* showDetectFrame. Setelah selesai dideteksi tiap *frame*, video yang disimpan di penyimpanan sementara akan dihapus. Algoritma ini

juga digunakan pada *function* untuk mendeteksi video-video yang sudah disiapkan.

```
def process_uploaded_video(conf, model):
    uploaded_video = st.file_uploader(
        "Upload a video file", type=["mp4", "avi", "mov"])

    if uploaded_video is not None:
        with NamedTemporaryFile(delete=False, suffix='.mp4') as temp_file:
            temp_file.write(uploaded_video.read())
            temp_video_path = temp_file.name

        with open(temp_video_path, 'rb') as video_file:
            video_bytes = video_file.read()
        if video_bytes:
            st.video(video_bytes)

    if st.button('Deteksi'):
        try:
            vid_cap = cv2.VideoCapture(temp_video_path)
            st_frame = st.empty()
            while (vid_cap.isOpened()):
                success, image = vid_cap.read()
                if success:
                    showDetectFrame(conf,
                                    model,
                                    st_frame,
                                    image
                                    )
                else:
                    vid_cap.release()
                    break
        except Exception as e:
            st.error("Error loading video: " + str(e))
```

Gambar 4. 33: Function untuk memproses video yang diupload

Function selanjutnya yang digunakan adalah *function* `take_picture` yang dapat dilihat pada gambar 4.33. *Function* ini akan mengambil foto atau gambar secara langsung menggunakan bantuan

```
def take_picture(conf, model):
    picture = st.camera_input("Silahkan Mengambil Gambar")

    if picture:
        with NamedTemporaryFile(delete=False, suffix=".mp4") as temp_file:
            temp_file.write(picture.read())
            temp_pict_path = temp_file.name

    if st.button("Deteksi Foto"):
        with st.spinner("Sedang Mendeteksi Objek..."):
            try:
                vid_cap = cv2.VideoCapture(temp_pict_path)
                st_frame = st.empty()
                while vid_cap.isOpened():
                    success, image = vid_cap.read()
                    if success:
                        showDetectFrame(
                            conf, model, st_frame, image,
                            caption="Deteksi Gambar"
                        )
                    else:
                        vid_cap.release()
                        break
            except Exception as e:
                st.error("Error loading video: " + str(e))
```

Gambar 4. 32: Function untuk mengambil foto atau gambar secara langsung

st.camera_input()). Sama seperti video yang diupload, gambar ini pun akan disimpan di penyimpanan sementara. Kemudian gambar tersebut akan dikirim ke *function* showDetectShow untuk melakukan deteksi dan menampilkan hasil.

```
def live(conf, model):
    webrtc_ctx = webrtc_streamer(
        key="object-detection",
        mode=WebRtcMode.SENDRECV,
        rtc_configuration={
            "iceServers": turn.get_ice_servers(),
            "iceTransportPolicy": "relay",
        },
        video_transformer_factory=lambda: VideoTransformer(model, conf),
        media_stream_constraints={"video": True, "audio": False},
        async_processing=True,
        video_processor_factory=lambda: VideoProcessorFactory(fps=60),
    )
```

Gambar 4. 34: Function untuk realtime

Function terakhir yang digunakan adalah *function* untuk membantu proses *realtime* seperti pada gambar 4.34. Proses ini akan menggunakan fitur webrtc_streamer dari *framework* Streamlit. Ada beberapa konfigurasi yang diperlukan untuk dapat menjalankan fitur ini dengan baik. Hal utama yang diperlukan adalah konfigurasi rtc. Pada perubahan infrastruktur terbaru terkait webrtc_streamer, ditetapkan harus menggunakan bantuan Twilio agar dapat menjalankan *live* tersebut.

```
class VideoTransformer(VideoTransformerBase):
    def __init__(self, model, conf):
        self.model = model
        self.conf = conf

    def transform(self, frame):
        img = frame.to_ndarray(format="bgr24")
        res = self.model.predict(img, show=False, conf=self.conf)
        res_plotted = res[0].plot()
        return res_plotted
```

Gambar 4. 35: Code untuk mendeteksi realtime

R14	4	4	4	3	4	3	4	3	4	4	4	41
R15	4	4	3	4	4	4	4	4	4	3	4	42
R16	3	3	3	3	4	3	3	3	3	3	3	34
R17	4	4	3	3	2	4	4	4	3	3	3	37
R18	4	4	4	4	4	4	4	4	4	3	4	43
R19	4	4	3	4	3	3	3	3	4	4	4	39
R20	4	4	3	3	4	4	4	4	3	3	3	39
R21	4	3	4	3	4	4	4	4	3	4	3	40
R22	4	4	4	4	4	4	4	4	4	4	4	44
R23	4	4	4	3	4	4	4	4	4	3	3	41
R24	4	3	4	4	4	4	4	4	4	3	4	42
R25	4	4	3	3	4	3	3	4	4	3	4	39
R26	4	4	4	3	3	3	4	3	4	4	4	40
R27	3	4	3	4	4	3	4	3	3	4	4	39
R28	4	4	4	3	4	4	3	4	4	3	4	41
R29	4	3	4	4	4	3	3	4	4	3	4	40
R30	3	4	4	4	4	4	4	3	4	3	4	41

Dari tabel 4.4 tersebut, selanjutnya akan dilakukan perhitungan untuk mencari nilai $rHitung$ tiap pertanyaan sesuai dengan rumus pada persamaan 1. Hasil perhitungan tersebut dapat dilihat pada tabel 4.5.

Tabel 4. 5: Hasil perhitungan $rHitung$

$rHitung$											
0,67	0,48	0,61	0,50	0,49	0,61	0,64	0,57	0,76	0,41	0,69	

Perhitungan nilai $rHitung$ ini menggunakan bantuan Excel dengan menggunakan rumus $=CORREL(\sum Q \sum T)$. Selanjutnya untuk menentukan valid dan tidak validnya sebuah instrumen, nilai $rHitung$ akan dibandingkan dengan nilai $rTabel$ dengan distribusi signifikan sebesar 5%. Dikarenakan responden berjumlah 30 maka $rTabel$ akan bernilai 0,349. Pada perbandingan ini, jika nilai $rHitung$ lebih besar dari $rTabel$ maka instrumen dikatakan valid, dan jika sebaliknya maka instrumen dikatakan tidak valid.

Secara lebih lanjut, hasil dari perbandingan tersebut dapat dilihat dari tabel 4.6.

Tabel 4. 6: Hasil uji validitas

rH	0,67	0,48	0,61	0,50	0,49	0,61	0,64	0,57	0,76	0,41	0,69
rT	0,349	0,349	0,349	0,349	0,349	0,349	0,349	0,349	0,349	0,349	0,349
K	V	V	V	V	V	V	V	V	V	V	V

Keterangan:

rH : *rHitung*

rT : *rTabel*

K : Keterangan

V : Valid

TV : Tidak Valid

Dari tabel tersebut dapat dilihat bahwa tiap instrumen dikatakan valid dan selanjutnya akan dilakukan pengujian reliabilitas.

4.7.2. Uji Reliabilitas

Setelah melakukan uji validitas untuk melihat apakah tiap instrumen valid untuk diujikan, selanjutnya akan dilakukan uji reliabilitas untuk melihat apakah instrumen dapat konsisten dalam mengukur hal yang harus diukur. Untuk menghitung suatu reliabilitas akan menggunakan rumus pada persamaan 2. Dengan mengacu pada tabel 3.4 tentang tingkatan atau interval reliabilitas, maka akan dicari nilai r .

$$r = \left(\frac{11}{11-1} \right) \left(1 - \frac{2.641}{9.752} \right) \quad (5)$$

$$r = 0.802 \quad (6)$$

Dari perhitungan tersebut diketahui bahwa nilai r adalah 0.802. Selanjutnya jika dikaitkan dengan tabel 3.4 tentang tingkatan atau interval reliabilitas, maka diketahui bahwa instrumen tersebut memiliki reliabilitas sangat tinggi dengan nilai $r = 0.802$.

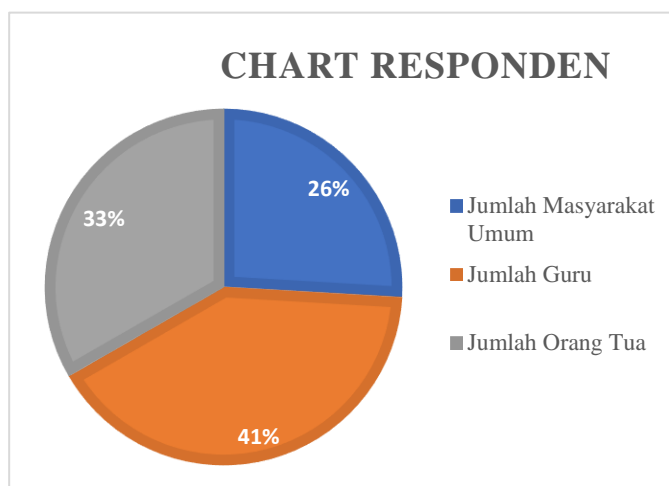
4.7.3. Uji Usability

Setelah melakukan uji validitas dan uji reliabilitas terhadap instrumen atau alat ukur pada kuesioner, maka selanjutnya kuesioner akan disebarakan secara terbuka. Adapun kuesioner akan berfokus pada guru-guru dari beberapa kelompok bermain, orang tua, dan masyarakat umum yang sebelumnya telah dibahas pada poin 3.7.1 terkait waktu dan tempat pengujian.

Dengan rentang waktu tujuh hari, berhasil mengumpulkan responden sebanyak 27 responden dengan masing-masing rincian dapat dilihat dari tabel 4.7 atau dengan melihat diagram pada gambar 4.38.

Tabel 4. 7: Rincian responden

Jumlah Responden Guru	7
Jumlah Responden Orang Tua	11
Jumlah Responden Masyarakat Luar	9
Total	27



Gambar 4. 36: Chart rincian responden

Selanjutnya untuk hasil kuesioner dapat dilihat pada tabel 4.8.

Adapun keterangan dari tabel tersebut adalah:

a) R : Responden

- b) St : Status
 c) Q : *Question*
 d) T : Total
 e) G : Guru
 f) OT : Orang Tua
 g) MU : Masyarakat Umum

Tabel 4. 8: Tabel hasil kuesioner

R	St	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q11	T
R1	G	4	4	3	4	3	4	4	4	4	4	3	41
R2	OT	4	3	4	4	3	3	3	4	4	3	3	38
R3	G	4	4	4	3	4	4	4	4	3	4	4	42
R4	G	4	4	4	3	4	3	4	4	4	4	3	41
R5	G	4	3	4	3	4	4	4	4	4	4	3	41
R6	OT	4	4	4	4	4	4	4	4	4	4	4	44
R7	OT	4	4	4	3	4	4	4	4	4	3	4	42
R8	OT	4	3	4	4	4	4	4	4	4	3	4	42
R9	G	4	3	3	4	4	4	4	3	3	3	4	39
R10	OT	4	4	4	3	4	4	4	4	4	4	4	43
R11	G	4	4	4	4	3	4	3	4	4	4	3	41
R12	G	3	3	4	4	3	4	3	4	4	3	4	39
R13	G	4	4	3	3	4	4	4	4	4	3	3	40
R14	G	4	4	3	4	4	4	3	4	4	4	3	41
R15	OT	3	4	4	3	4	3	3	4	3	3	4	38
R16	OT	4	4	3	4	4	4	4	4	4	4	3	42
R17	OT	4	4	4	4	3	4	4	4	3	4	4	42
R18	OT	4	4	4	4	4	3	3	4	4	3	4	41
R19	MU	4	4	4	4	4	4	4	3	4	3	4	42
R20	MU	4	4	4	4	3	4	4	3	4	4	3	41
R21	MU	4	4	4	3	4	3	4	3	4	4	4	41
R22	MU	4	4	4	3	4	3	4	3	4	4	4	41
R23	G	4	4	3	3	4	3	3	4	4	3	4	39
R24	G	4	4	4	3	3	3	4	3	4	3	4	39
R25	MU	4	4	3	4	4	3	4	4	3	4	4	41
R26	MU	4	4	4	3	4	4	3	4	4	3	4	41
R27	MU	4	3	4	3	4	4	4	4	3	4	3	40

Berikutnya dengan menggunakan rumus pada persamaan 3, maka akan dihitung nilai *usability*. Sebagai contoh maka akan dilakukan perhitungan untuk melihat *usability* dari responden pertama.

$$usability = \frac{41}{44} \times 100 \quad (7)$$

$$usability = 93,2 \quad (8)$$

Dari perhitungan tersebut dapat dilihat bahwa *usability* atau kegunaan dari responden pertama mendapat nilai 93,2. Dari hasil ini, nilai kemudian akan dikonversikan ke tabel kategorinya sesuai pada

tabel 3.5. Secara lebih lengkap, hasil dari perhitungan tersebut dapat dilihat pada tabel 4.9 dengan keterangan:

- a) SP (%) : Skor Persentase
- b) SL : Sangat Layak
- c) L : Layak
- d) C : Cukup
- e) TL : Tidak Layak
- f) STL : Sangat Tidak Layak

Tabel 4. 9: Tabel hasil perhitungan usability

R	SKOR	SP (%)	K	R	SKOR	SP (%)	K
R1	41	93,18	SL	R15	38	86,36	SL
R2	38	86,36	SL	R16	42	95,45	SL
R3	42	95,45	SL	R17	42	95,45	SL
R4	41	93,18	SL	R18	41	93,18	SL
R5	41	93,18	SL	R19	42	95,45	SL
R6	44	100	SL	R20	41	93,18	SL
R7	42	95,45	SL	R21	41	93,18	SL
R8	42	95,45	SL	R22	41	93,18	SL
R9	39	88,63	SL	R23	39	88,63	SL
R10	43	97,72	SL	R24	39	88,63	SL
R11	41	93,18	SL	R25	41	93,18	SL
R12	39	88,63	SL	R26	41	93,18	SL
R13	40	90,90	SL	R27	40	90,90	SL
R14	41	93,18	SL	Rata – Rata		92,76	SL

Dari hasil tabel di atas dapat dilihat bahwa dari 27 responden kuesioner tersebut memberikan rata-rata nilai 92,76% dengan total keseluruhan sebesar 1102 dari 1188 skor harapan. Dengan hasil ini dan hasil konversi nilai dengan intervalnya, menunjukkan bahwa hasil perhitungan *usability* pada *website* deteksi objek memiliki nilai “sangat layak”.

BAB V

PENUTUP

5.1. Kesimpulan

Berdasarkan rangkaian pelatihan, analisis, dan pengujian yang telah dilakukan pada bab sebelumnya, dapat ditarik beberapa kesimpulan, bahwa dari ke-32 skenario atau percobaan yang dilakukan, diketahui bahwa percobaan ketujuh memiliki hasil mAP terbaik daripada percobaan-percobaan yang lainnya dengan nilai mAP sebesar 0,8791 atau sebesar 87%. Model yang dihasilkan ini pun mampu mendeteksi dengan baik yang ditunjukkan dengan hasil validasi model dan pengujian model yang dilakukan.

Pengujian instrumen pengukuran yang dilakukan baik itu pengujian validitas maupun pengujian reliabilitas memiliki hasil yang memuaskan di mana memiliki instrumen yang valid dengan tingkat kekonsistenan yang sangat tinggi. Dari hasil pengujian ini, selanjutnya pada pengujian dari aspek kegunaan atau *usability* yang dilakukan dengan menyebarkan kuesioner pun memiliki nilai “sangat layak” dengan rata-rata nilai sebesar 94,9%.

5.2. Saran

Dari penelitian yang telah dilakukan sebelumnya, terdapat beberapa saran yang dapat dilakukan untuk pengembangan selanjutnya. Adapun saran-saran tersebut adalah untuk melakukan penelitian lebih lanjut untuk mencari sebab kesalahan prediksi pada model, khususnya pada objek sepatu. Selain itu, disarankan juga untuk memperbanyak variasi baik itu pada dataset maupun pada objek yang dideteksi sehingga model dapat mendeteksi objek yang lebih tepat dan lebih luas.

Pada website yang dihasilkan pun disarankan agar membuat website yang lebih interaktif sehingga mampu mengembangkan pola pikir anak dalam segi kognitifnya. Pengujian yang dilakukan pun disarankan untuk melihat aspek-aspek lainnya selain aspek kegunaan atau *usability* seperti aspek kesesuaian konten dan lainnya.

DAFTAR PUSTAKA

- Adarsh, P., & Rathi, P. (2020). YOLO v3-Tiny: Object Detection and Recognition using one stage improved model. *International Conference on Advanced Computing & Communication Systems*, 687–694.
- Adhinata, F. D., Wardhana, A. C., Rakhmadani, D. P., & Jayadi, A. (2020). Peningkatan Kualitas Citra pada Citra Digital Gelap. *Jurnal E-Komtek (Elektro-Komputer-Teknik)*, 4(2), 136–144. <https://doi.org/10.37339/e-komtek.v4i2.373>
- Aini, Q., Lutfiani, N., Kusumah, H., & Zahran, M. S. (2021). Deteksi dan Pengenalan Objek dengan Model Machine Learning: Model YOLO. 6(2), 2502–2714.
- Andono, P. N., Sutojo, T., & Muljono. (2017). *Pengolahan Citra Digital* (A. Pramesta, Ed.; 1 ed.). ANDI.
- Ariansyah, D. S. (t.t.). Klasifikasi Hewan dengan Menggunakan Trasfer Learning Googlenet. *JIFT: Jurnal Informatika*.
- Ayuka, F., Pradana, P., Universitas, M., & Wacana, K. S. (2021). Pengembangan Instrumen Penilaian Sikap Disiplin Menggunakan Skala Likert Dalam Pembelajaran Tematik Kelas IV SD. *Jurnal Pendidikan Dasar*, 5(1), 13–29. <https://ejournal.stitpn.ac.id/index.php/fondatia>
- Banovbi, R., Irsal, P., & Utaminingrum, F. (2022). Sistem Pengenalan Gerak Kepala sebagai Navigasi Kursi Roda Pintar dengan menggunakan Metode YOLOV5 berbasis TX2. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 6(12), 5576–5581. <http://j-ptiik.ub.ac.id>

- Budiyanta, N. E. (2018). Pengembangan Kelayakan Sistem Informasi Manajemen Untuk Workshop Dan Laboratorium. *JURNAL ELEKTRO*, 11(1), 1–14.
- Chen, H., Wang, Y., Guo, T., Xu, C., Deng, Y., Liu, Z., Ma, S., Xu, C., Xu, C., & Gao, W. (2021). Pre-Trained Image Processing Transformer. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 12299–12310. <https://github>.
- Dhiyatmika, I. D. W., Putra, I. K. G. D., & Mandenni, N. M. I. M. (2015). Aplikasi augmented reality magic book pengenalan binatang untuk siswa TK. *Lontar Komputer*, 6(2), 120–127.
- Dwyer, B., & Gallagher, J. (2023, Maret 16). *Getting Started with Roboflow*. Roboflow Blog. <https://blog.roboflow.com/getting-started-with-roboflow/>
- Ernawati, I., & Sukardiyono, T. (2017). Uji Kelayakan Media Pembelajaran Interaktif Pada Mata Pelajaran Administrasi Server. *Elinvo: Electronics, Informatics, and Vocational Educational*, 2(2), 204–210.
- Fadillah, R. Z., Irawan, A., Susanty, M., & Artikel, I. (2021). Data Augmentasi Untuk Mengatasi Keterbatasan Data Pada Model Penerjemah Bahasa Isyarat Indonesia (BISINDO). *JURNAL INFORMATIKA*, 8(2). <http://ejournal.bsi.ac.id/ejurnal/index.php/ji>
- Gibran, H., Purnama, B., Kosala, G., & Pengoptimasian Pengukuran Kepadatan Jalan Raya, G. (t.t.). Optimizing Highway Density Measurement with CCTV Using the Yolov8 Method. *Technomedia Journal (TMJ)*, 9(1), 9. <https://doi.org/10.33050/tmj.v9i1.2216>
- Heri Pratikno, Muhammad Rifki Pratama, Yosefine Triwidyastuti, & Musayyanah. (2023). Pengenalan Gestur Jari Tangan Sebagai Media Pembelajaran

- Berhitung Bagi PAUD Berbasis Visi Komputer Dan Deep Learning. *Journal of Computer Electronic and Telecommunication*, 4(1).
<https://doi.org/10.52435/complete.v4i1.355>
- Hidayatullah, P. (2017). *Pengolahan Citra Digital: Teori dan Aplikasi Nyata*. Informatika Bandung: Penerbit Informatika.
- Iryanto, S. Y., & Zaini, T. M. (2014). *Pengolahan Citra Digital*. Anggota IKAPI.
- Jiang, P., Ergu, D., Liu, F., Cai, Y., & Ma, B. (2022). A Review of Yolo Algorithm Developments. *Procedia Computer Science*, 199, 1066–1073.
- Jocher, G., & Sergiuwaxmann. (2023, Januari 9). *Ultralytics YOLOv8 Docs*. Ultralytics. <https://docs.ultralytics.com>
- Jönsson Hyberg, J., & Sjöberg, A. (2023). *Investigation Regarding The Performance of YOLOv8 in Pedestrian Detection* (TRITA-EECS-EX, Nomor 2023:282).
- Kaputa, D. S., & Landy, B. P. (2021). YOLBO: You only Look Back Once-A Low Latency Object Tracker Based on YOLO and Optical Flow. *IEEE Access*, 9, 82497–82507. <https://doi.org/10.1109/ACCESS.2021.3080136>
- Karlina, O. E., & Indarti, D. (2019). Pengenalan Objek Makaxnan Cepat Saji pada Video dan Real Time Webcam Menggunakan Metode Youu Only Look Once (YOLO). *Jurnal Ilmiah Informatika Komputer*, 24(3), 199–208.
<https://doi.org/10.35760/ik.2019.v24i3.2362>
- Karna, N. B. A., Putra, M. A. P., Rachmawati, S. M., Abisado, M., & Sampedro, G. A. (2023). Toward Accurate Fused Deposition Modeling 3D Printer Fault Detection Using Improved YOLOv8 With Hyperparameter Optimization.

IEEE Access, 11, 74251–74262.

<https://doi.org/10.1109/ACCESS.2023.3293056>

Khare, O. M., Gandhi, S., Rahalkar, A. M., & Mane, S. (2023). *YOLOv8-Based Visual Detection of Road Hazards: Potholes, Sewer Covers, and Manholes*.

<http://arxiv.org/abs/2311.00073>

Komang Ayu, N., & Surya Manuaba, I. B. (2021). Media Pembelajaran Zoolfabeth Menggunakan Multimedia Interaktif untuk Perkembangan Kognitif Anak Usia Dini. *Jurnal Pendidikan Anak Usia Dini Undiksha*, 9(2), 194–201.

<https://ejournal.undiksha.ac.id/index.php/JJPAUD/index>

Kumari, N., Ruf, V., Mukhametov, S., Schmidt, A., Kuhn, J., & Küchemann, S. (2021). Mobile Eye-Tracking Data Analysis Using Object Detection via YOLO v4. *Sensors*, 21(22). <https://doi.org/10.3390/s21227668>

Kusuma, W. A., Noviasari, V., & Marthasari, G. I. (2016). Analisis Usability dalam User Experience pada Sistem KRS-Online UMM menggunakan USE Questionnaire. *JNTETI: Jurnal Nasional Teknik Elektro dan Teknologi Informasi*, 5(4), 294–301.

Li, M., Ma, L., Blaschke, T., Cheng, L., & Tiede, D. (2016). A Systematic Comparison of Different Object-Based Classification Techniques Using High Spatial Resolution Imagery in Agricultural Environments. *International Journal of Applied Earth Observation and Geoinformation*, 49, 87–98. <https://doi.org/10.1016/j.jag.2016.01.011>

Lou, H., Duan, X., Guo, J., Liu, H., Gu, J., Bi, L., & Chen, H. (2023). DC-YOLOv8: Small-Size Object Detection Algorithm Based on Camera Sensor. *Electronics (Switzerland)*, 12(10). <https://doi.org/10.3390/electronics12102323>

- Maulana, F. (2021). *Machine Learning Object Detection Tanaman Obat Secara Real-Time Menggunakan Metode YOLO (You Only Look Once)*.
- Maurya, L. S., Hussain, M. S., & Singh, S. (2021). Developing Classifiers through Machine Learning Algorithms for Student Placement Prediction Based on Academic Performance. *Applied Artificial Intelligence*, 35(6), 403–420. <https://doi.org/10.1080/08839514.2021.1901032>
- Maxwell, A. E., Warner, T. A., & Guillén, L. A. (2021). Accuracy Assessment in Convolutional Neural Network-Based Deep Learning Remote Sensing Studies—part 1: Literature review. *Remote Sensing*, 13(13). <https://doi.org/10.3390/rs13132450>
- Musdalifah, Anas, M., & Sadaruddin. (2020). Peningkatan Kreativitas Anak Melalui Metode Discovery pada Pembelajaran Sains di Taman Kanak-Kanak Aisyiyah Bustanul Athfal Mario. *TEMATIK: Jurnal Pemikiran dan Penelitian Pendidikan Anak Usia Dini*, 6(1), 42–52.
- Nur, L., Hafina, A., & Rusmana, N. (2020). Kemampuan Kognitif Anak Usia Dini Dalam Pembelajaran Akuatik. *Scholaria: Jurnal Pendidikan dan Kebudayaan*, 10(1), 42–50.
- Panggalih, K., Kurniawan, W., & Gata, W. (2022). Implementasi Perbandingan Deteksi Tepi Pada Citra Digital Menggunakan Metode Roberst, Sobel, Prewitt dan Canny. *Infotek: Jurnal Informatika dan Teknologi*, 5(2), 337–347. <https://doi.org/10.29408/jit.v5i2.5923>
- Perez, F., Vasconcelos, C., Avila, S., & Valle, E. (2018). Data augmentation for skin lesion analysis. *Lecture Notes in Computer Science (including subseries*

- Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics*), 11041 LNCS, 303–311. https://doi.org/10.1007/978-3-030-01201-4_33
- Priyono, F. H., Rahmawati, A., & Pudyaningtyas, A. R. (2021). Kemampuan Berpikir Simbolik Pada Anak Usia 5-6 Tahun. *Jurnal Kumara Cendekia*, 9(4), 212–217. <https://jurnal.uns.ac.id/kumara>
- Putri, V. L., Wijayanti, A., & Kusumastuti, N. D. (2021). Pengembangan Media Frueelin Untuk Meningkatkan Perkembangan Kognitif Anak Usia Dini. *Jurnal Golden Age*, 5(02), 155–163. <https://doi.org/10.29408/jga.v5i01.3385>
- Rafly Alwanda, M., Putra, R., Ramadhan, K., & Alamsyah, D. (2020). Implementasi Metode Convolutional Neural Network Menggunakan Arsitektur LeNet-5 untuk Pengenalan Doodle. *Jurnal Algoritme*, 1(1), 45.
- Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2015). You Only Look Once: Unified, Real-Time Object Detection. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 779–788. <http://arxiv.org/abs/1506.02640>
- Rosita, E., Hidayat, W., & Yuliani, W. (2021). Uji Validitas Dan Reliabilitas Kuesioner Perilaku Prososial. *FOKUS (Kajian Bimbingan & Konseling dalam Pendidikan)*, 4(4), 279. <https://doi.org/10.22460/fokus.v4i4.7413>
- Safita, M., & Suryana, D. (2022). Pengenalan Warna Melalui Media Stick Warna Terhadap Kemampuan Kognitif Anak Usia 4-5 Tahun. *Bunayya: Jurnal Pendidikan Anak*, 8(1), 28–43.
- Sandhya, & Kashyap, A. (2024). A Novel Method for Real-Time Object-Based Copy-Move Tampering Localization in Videos Using Fine-Tuned YOLO V8.

- Forensic Science International: Digital Investigation*, 48.
<https://doi.org/10.1016/j.fsidi.2023.301663>
- Sanjaya, J., & Ayub, M. (2020). Augmentasi Data Pengenalan Citra Mobil Menggunakan Pendekatan Random Crop, Rotate, dan Mixup. *Jurnal Teknik Informatika dan Sistem Informasi*, 6(2).
<https://doi.org/10.28932/jutisi.v6i2.2688>
- Sholahuddin, M. R., Harika, M., Awaludin, I., Dewi, Y. C., Dhia Fauzan, F., Sudimulya, B. P., & Widarta, V. P. (2023). Optimizing YOLOv8 for Real-Time CCTV Surveillance: A Trade-off Between Speed and Accuracy. *Jurnal Online Informatika*, 8(2), 261–270. <https://doi.org/10.15575/join.v8i2.1196>
- Suasapha, A. H. (2020). Skala Likert Untuk Penelitian Pariwisata; Beberapa Catatan Untuk Menyusunnya Dengan Baik. *JURNAL KEPARIWISATAAN*, 19(1), 26–37. <https://doi.org/10.52352/jpar.v19i1.407>
- Sufandi, U. U., Priono, M., Aprijani, D. A., Wicaksono, B. A., & Trihapningsari, D. (2022). Uji Usability fungsi Aplikasi Web Sistem Informasi Dengan Use Questionnaire. (Studi Kasus: Aplikasi Web Sistem Informasi Tiras Dan Transaksi Bahan Ajar). *Jurnal Pendidikan Teknologi dan Kejuruan*, 19(1), 24–34.
- Sunarti, A., Yusuf Muslihin, H., & Abdul Muiz Lidinillah, D. (2023). Pengembangan Instrumen Deteksi Dini Perkembangan Kognitif Anak Usia 3 Tahun. *Jurnal PAUD Agapedia*, 7(1), 41–50.
<https://ejournal.upi.edu/index.php/agapedia>
- Supriadi, M. F., Rachmawati, E., & Arifianto, A. (2021). Pembangunan Aplikasi Mobile Pengenalan Objek Untuk Pendidikan Anak Usia Dini. *Jurnal*

- Teknologi Informasi dan Ilmu Komputer*, 8(2), 357–364.
<https://doi.org/10.25126/jtiik.2021824363>
- Terven, J. R., & Cordova-Esparza, D. M. (2023). A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS. *Machine Learning and Knowledge Extraction*, 5(4), 1680–1716.
<https://doi.org/10.3390/make5040083>
- Thoriq, M. Y. A., Permana, K. E., & Siradjuddin, I. A. (2023). Deteksi Wajah Manusia Berbasis One Stage Detector Menggunakan Metode You Only Look Once (YOLO). *JURNAL TEKNOINFO*, 17(1), 66–73.
<https://ejurnal.teknokrat.ac.id/index.php/teknoinfo/index>
- Wang, A. Y., Wang, D., Drozdal, J., Liu, X., Park, S., Oney, S., & Brooks, C. (2021). What Makes a Well-Documented Notebook? A Case Study of Data Scientists' Documentation Practices in Kaggle. *Conference on Human Factors in Computing Systems - Proceedings*.
<https://doi.org/10.1145/3411763.3451617>
- Wang, Q., Bi, S., Sun, M., Wang, Y., Wang, D., & Yang, S. (2018). Deep Learning Approach to Peripheral Leukocyte Recognition. *PLoS ONE*, 14(6).
<https://doi.org/10.1371/journal.pone.0218808>
- Wu, X., Sahoo, D., & Hoi, S. C. H. (2020). Recent Advances in Deep Learning for Object Detection. *Neurocomputing*, 396, 39–64.
<https://doi.org/10.1016/j.neucom.2020.01.085>
- Yuni Wulandari, I., Indroasyoko, N., Mudia Alti, R., Asri, Y. N., & Hidayat, R. (2022). Pengenalan Sistem Deteksi Objek untuk Anak Usia Dini

Menggunakan Pemrograman Python. *remik*, 6(4), 664–673.
<https://doi.org/10.33395/remik.v6i4.11772>

Zhang, S., Wang, T., Wang, C., Wang, Y., Shan, G., & Snoussi, H. (2019). Video Object Detection base on RGB and Optical Flow Analysis. *2019 2nd China Symposium on Cognitive Computing and Hybrid Intelligence (CCHI)*, 280–284. 10.1109/CCHI.2019.8901921

Zhao, L., & Li, S. (2020). Object Detection Algorithm Based on Improved YOLOv3. *Electronics (Switzerland)*, 9(3).
<https://doi.org/10.3390/electronics9030537>

Zulwati, P. R., Fatmawati, F. A., & Agustina, R. (2022). Pengembangan Media Pembelajaran Pop Up Book Untuk Meningkatkan Perkembangan Kognitif Anak Usia 5-6 Tahun Di Tk Aba 42 GBA. *Jurnal Golden Age*, 6(02), 635–647. <https://doi.org/10.29408/goldenage.v6i02.77360>