

**PENGENALAN OBJEK UNTUK PEMBELAJARAN ANAK-ANAK
MENGUNAKAN 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 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. Hasil dari penelitian ini kurang terlihat perbandingan mAP tiap model sehingga dilakukan pencarian parameter terbaik yang dapat meningkatkan mAP model. Dari hasil implementasi *hyperparameter* dan pencarian parameter terbaik, ditemukan parameter terbaik menghasilkan mAP sebesar 88% dan mengalami perkembangan dari pelatihan sebelumnya. Selanjutnya dengan bantuan Streamlit, akan dibuat *website* untuk diujicobakan dan disebarluaskan 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. The results of initial study showed that the mAP of each model was lack of comparison sources, so it is essential to carry out the searching for the best parameters that can improve the mAP of the model. From the results of the hyperparameter implementation and the search for the best parameters, it was found that the best parameters produced a mAP of 88% and experienced progress from previous training. 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.....	1
HALAMAN PERSETUJUAN PEMBIMBING	3
ABSTRAK	4
ABSTRACT	5
DAFTAR ISI.....	6
DAFTAR TABEL	9
DAFTAR GAMBAR.....	10
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. Annotated Image	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
3.1. Gambaran Umum Penelitian	24
3.2. Data	25
3.3. Pre-Processing	25
3.3.1. Labeling.....	26
3.3.2. Resize Image	26
3.3.3. Augmentasi	27
3.3.4. Data Split	28
3.4. Modeling	29
3.4.1. Mendapatkan Dependency Yolo Versi 8.....	29
3.4.2. Mendapatkan Dataset dari Roboflow	29
3.4.3. Train.....	29
3.5. Evaluate.....	30
3.6. Skenario Pengujian	30
3.7. Pengujian Kelayakan Kuesioner dan Kegunaan Hasil Akhir	31
3.7.1. Waktu dan Tempat Pengujian.....	31
3.7.2. Uji Kelayakan Kuesioner	31
BAB VI.....	35
HASIL PENELITIAN DAN PEMBAHASAN	35

4.1.	Pengumpulan Data.....	35
4.2.	<i>Pre-processing Data</i>	35
4.2.1.	<i>Annotate Image</i>	35
4.2.2.	<i>Rescale / Resize Image</i>	37
4.2.3.	<i>Augmentasi Image</i>	37
4.2.4.	<i>Data Split</i>	40
4.3.	Modeling	41
4.3.1.	Menginstall YOLO.....	41
4.3.2.	Mengimpor Dataset.....	41
4.3.3.	Pelatihan Model.....	43
4.4.	Analisis Hasil Pengujian	46
4.4.1.	Hasil Pelatihan Model.....	46
4.4.2.	Pengujian <i>Hyperparameter</i>	49
4.4.3.	Analisis Uji <i>Hyperparameter</i>	51
4.5.	Implementasi Aplikasi Deteksi Objek.....	54
4.5.1.	<i>Interface Aplikasi</i>	54
4.5.2.	Implementasi <i>Code</i> untuk Deteksi	57
4.6.	Pengujian Kegunaan Aplikasi.....	61
4.6.1.	Uji Validitas	61
4.6.2.	Uji Reliabilitas	63
4.6.3.	Uji <i>Usability</i>	64
BAB V	67
PENUTUP	67
5.1.	Kesimpulan	67
5.2.	Saran	67
DAFTAR PUSTAKA	69

DAFTAR TABEL

Tabel 2. 1: Tabel review literature	7
Tabel 3. 1: Tabel skenario Pengujian.....	30
Tabel 3. 2: Rincian Pernyataan Kuesioner.....	32
Tabel 3. 3: Ketentuan skala likert	32
Tabel 3. 4: Tingkat reliabilitas Cronbach's Alpha.....	33
Tabel 3. 5: Kategori Kegunaan	34
Tabel 4. 1: Hasil penelitian model berdasarkan skenario pengujian.....	46
Tabel 4. 2: Tabel perbandingan precision, recall, dan validasi tiap model.....	47
Tabel 4. 3: Rincian hyperparameter	50
Tabel 4. 4: Hasil kuesioner pengujian.....	61
Tabel 4. 5: Hasil perhitungan rHitung	62
Tabel 4. 6: Hasil uji validitas	63
Tabel 4. 7: Rincian responden.....	64
Tabel 4. 8: Tabel hasil kuesioner	65
Tabel 4. 9: Tabel hasil perhitungan usability	66

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 Modelling	24
Gambar 3. 1: Flowchart pre-processing	24
Gambar 3. 2: Flowchart penelitian	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	26
Gambar 3. 8: Contoh augmentasi yang diterapkan pada citra	28
Gambar 3. 9: Data split dari Roboflow	28
Gambar 3. 10: Contoh confusion matrix	30
Gambar 4. 1: Proses upload data ke Roboflow	35
Gambar 4. 2: Proses annotate di Roboflow	36
Gambar 4. 3: Data hasil labeling atau bounding box	36
Gambar 4. 4: Proses resize pada dataset	37
Gambar 4. 5: Proses augmentasi dataset	38
Gambar 4. 6: Proses exposure dataset	38
Gambar 4. 7: Proses rotation dataset	39
Gambar 4. 8: Proses flip dataset	40
Gambar 4. 9: Data split oleh Roboflow	40

Gambar 4. 10: Mengimpor YOLO ke dalam proyek	41
Gambar 4. 11: Code untuk menginstall library	41
Gambar 4. 12: Code untuk mengimpor dataset dari Roboflow	42
Gambar 4. 13: Keterangan jika berhasil mengunduh dan mengekstrak dataset ...	42
Gambar 4. 14: Hasil ekstraksi dataset	42
Gambar 4. 15: Isi dari file data.yaml.....	43
Gambar 4. 16: Code untuk pembuatan model dengan custom dataset	43
Gambar 4. 17: Detail yang digunakan pada pelatihan	45
Gambar 4. 18: Output dari rincian tiap epoch.....	45
Gambar 4. 19: Grafik pelatihan model.....	47
Gambar 4. 20: Code untuk mengombinasikan tiap parameter	50
Gambar 4. 21: Code untuk melakukan training dan mencari best parameter	51
Gambar 4. 23: Kombinasi parameter dengan hasil mAP tertinggi	52
Gambar 4. 22: Hasil validasi terhadap model	52
Gambar 4. 25: Kumpulan grafik evaluasi	53
Gambar 4. 24: Confusion matrix.....	53
Gambar 4. 26: Tampilan halaman pertama ketika website diakses	55
Gambar 4. 27: Tampilan halaman jika memilih menu gambar.....	55
Gambar 4. 28: Tampilan jika memilih menu video	56
Gambar 4. 29: Tampilan jika memilih menu youtube	56
Gambar 4. 30: Tampilan ketika memilih menu Real-Time	57
Gambar 4. 31: Code untuk mendeteksi gambar yang diupload	57
Gambar 4. 32: Function untuk menampilkan gambar hasil deteksi.....	58
Gambar 4. 33: Function untuk mencari video dari link youtube	58

Gambar 4. 34: Function untuk memproses video yang diupload.....	59
Gambar 4. 35: Function untuk mengambil foto atau gambar secara langsung.....	60
Gambar 4. 36: Function untuk realtime	60
Gambar 4. 37: Code untuk mendeteksi realtime.....	61
Gambar 4. 38: Chart rincian responden	64

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 alternatif 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 untuk mengenali objek sekitar secara?

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 dan 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

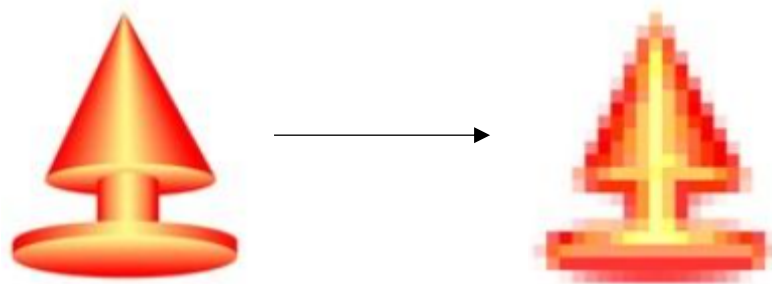
yang tepat dapat membantu perkembangan ini berjalan optimal (Nur dkk., 2020). Aspek kognitif menjadi hal utama dalam perkembangan anak karena berkaitan dengan bagaimana cara mereka berpikir dan mulai berfungsi. Karena itu, tujuan utama pendidikan anak usia dini dalam hal perkembangan kognitif adalah 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



Gambar 2. 1: Ilustrasi proses digitalisasi

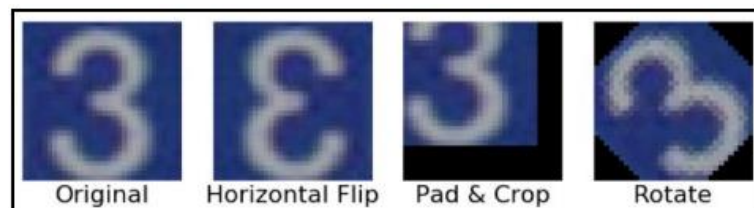
Sumber: <https://slideplayer.info/slide/11809302/>

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).

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)

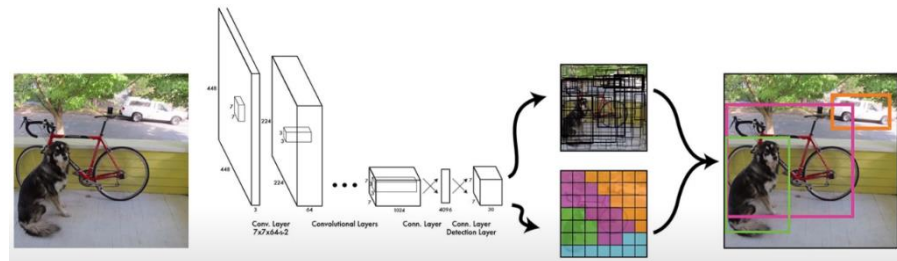
You Only Look Once (YOLO) adalah sebuah arsitektur yang menerapkan pendekatan baru untuk mendeteksi objek. YOLO pertama kali dikenalkan pada tahun 2015 (Redmon dkk., 2015). Algoritma ini biasanya digunakan untuk mendeteksi objek secara *realtime*. YOLO menggunakan jaringan saraf tiruan untuk mempelajari 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 klusterisasi 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. 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.

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

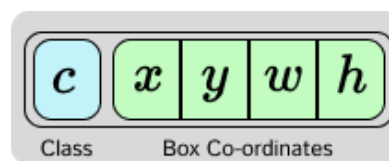


Gambar 2. 3: Ilustrasi proses deteksi YOLO

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).

Ultralytics merilis YOLO v8 yang dikembangkan dari versi-

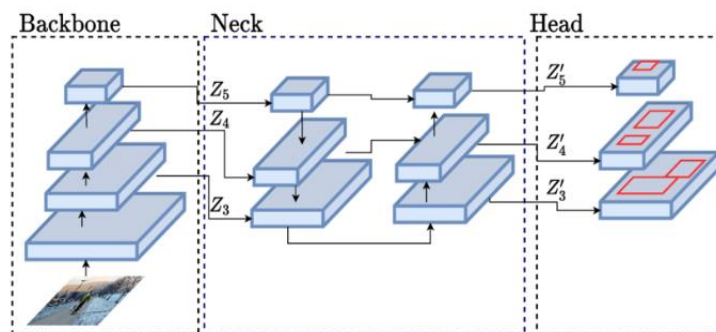


Gambar 2. 4: Ilustrasi nilai yang merepresentasikan bounding box

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).

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,



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

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. Annotated Image

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 r tabel (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.

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

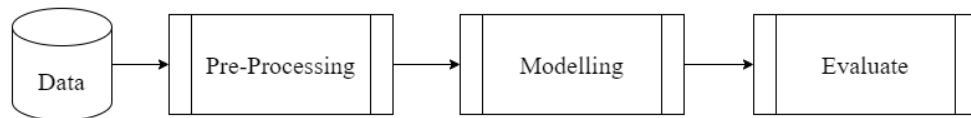
(Sufandi dkk., 2022).

BAB III: METODE PENELITIAN

3.1. Gambaran Umum Penelitian

Secara garis besar, alur dari penelitian yang akan dilakukan menggunakan bahasa pemrograman *python* ini dapat dilihat dari *flowchart* 3.2.

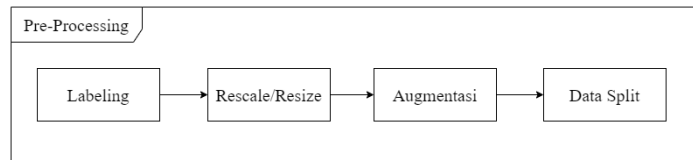
Dari *flowchart* tersebut, terdapat empat tahap utama dalam penelitian ini. Pada



Gambar 3. 3: Flowchart penelitian

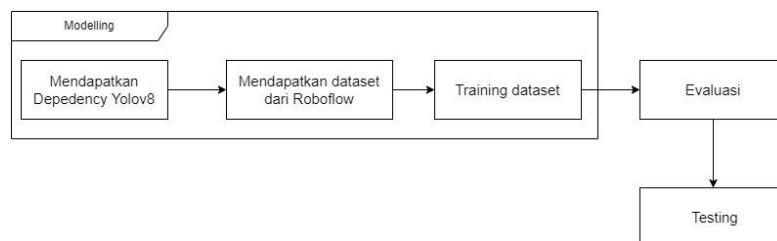
pelatihan ini setelah data dikumpulkan, selanjutnya akan dilakukan *pre-processing* sebelum dibuat pemodelan atau pelatihan. Dan setelah model dibuat, akan dilakukan evaluasi terhadap model tersebut. Dalam tahap *pre-processing* dan modeling dapat di lihat secara rinci pada *flowchart* 3.1 dan 3.3.

Untuk proses *pre-processing* akan menggunakan alat bantu Roboflow. Hal ini



Gambar 3. 2: Flowchart pre-processing

dikarenakan Roboflow dapat membantu peneliti dalam mempersiapkan data mulai dari pelabelan hingga pada *pre-processing* dan augmentasi dataset sehingga dapat mempermudah sekaligus mempercepat proses *pre-processing* .



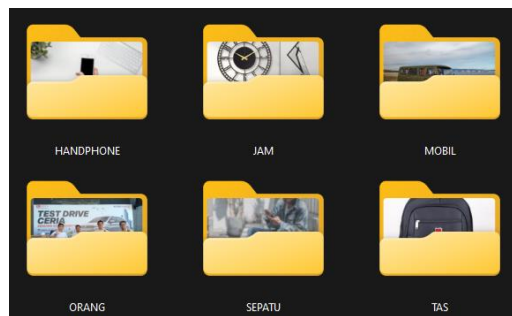
Gambar 3. 1: Flowchart Modelling

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.

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.



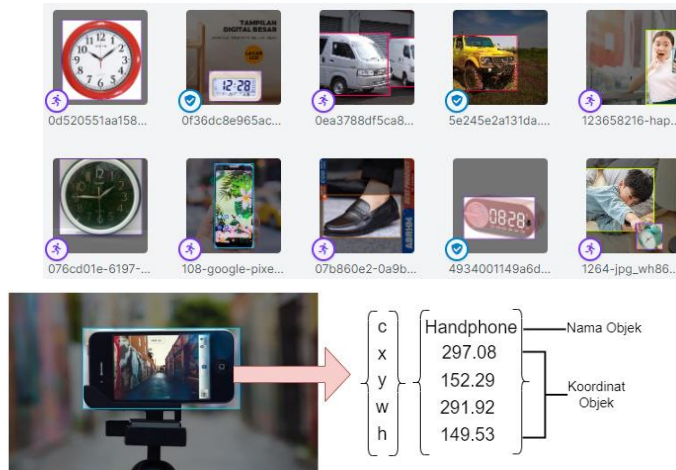
Gambar 3. 4: Gambar kumpulan dataset sesuai kelas

3.3. Pre-Processing

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

3.3.1. 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



Gambar 3. 6: Ilustrasi nilai bounding box

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.

3.3.2. Resize Image

Pada tahap ini, gambar akan melalui tahan *resize* dengan ukuran dua ukuran yang berbeda, yaitu 416 x 416, dan 608 x 608.



Gambar 3. 7: Contoh hasil rescale

Tahap ini dilakukan agar gambar mempunyai ukuran yang seragam sebelum masuk ke dalam tahap *training*. 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. Karena ukuran sebelumnya sering digunakan dalam pelatihan untuk mendeteksi objek, peneliti mencoba membandingkan dengan ukuran yang lebih besar yaitu 608 x 608 seperti yang diterapkan Gajalakshmi dan teman-temannya (Gajalakshmi dkk., 2020) untuk melihat seberapa besar peningkatan model dalam mendeteksi objek.

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:

a) Grayscale

b) Exposure

Tingkat *exposure* yang digunakan dalam penelitian ini adalah 15%. Untuk tingkat yang lebih tinggi akan membuat gambar menjadi lebih terang dan/atau lebih gelap sehingga dapat sulit dalam mengenali objek.

c) Rotation

Tingkat rotasi yang digunakan dalam penelitian ini adalah 10° .

d) *Flip*: Horizontal

Flip gambar yang diterapkan dalam penelitian ini adalah *flip* secara horizontal.

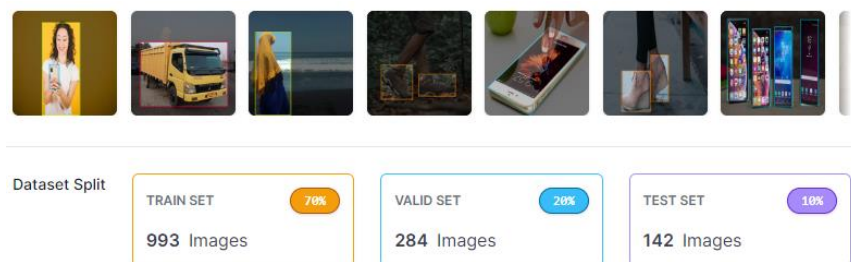
Proses augmentasi ini bertujuan untuk menambah variasi dan jumlah data. Sehingga dapat mengoptimalkan dalam proses *training* dan menghindari *overfitting*.



Gambar 3. 8: Contoh augmentasi yang diterapkan pada citra

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*, *testing*, dan *validation* dengan dengan pembagian 70:20:10. Dikarenakan menggunakan Roboflow, maka ketika dataset



Gambar 3. 9: Data split dari Roboflow

terbentuk, Roboflow akan secara otomatis membagi dataset menjadi tiga bagian, seperti yang terlihat di gambar 3.8.

3.4. Modeling

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

3.4.1. 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 mengloning *repository* yang sudah disiapkan pengembang dan menginstall persyaratannya (Jocher & Sergiuwaxmann, 2023).

3.4.2. Mendapatkan Dataset dari Roboflow

Dataset yang sebelumnya sudah dibuat 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).

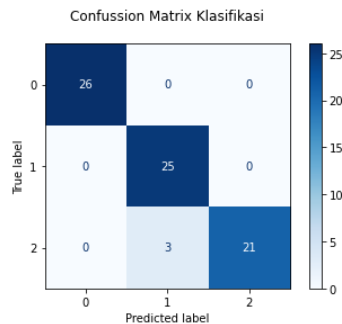
3.4.3. Train

Pada tahap ini, dataset yang didapatkan dari Roboflow akan dilatih dengan *depedency* 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).

3.5. 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 confusion matrix

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

3.6. Skenario Pengujian

Dalam penelitian akan dilakukan evaluasi terhadap skenario pengujian dengan memvariasikan variabel *image size* dan augmentasi. Variasi yang dilakukan ini demi melihat seberapa besar mAP yang dihasilkan jika ukuran gambar dinaikkan menjadi 608x608 baik menggunakan augmentasi maupun tidak pada datasetnya. Secara ringkas terkait skenario pengujian dapat dilihat dari tabel 3.1.

Tabel 3. 1: Tabel skenario Pengujian

Bantuan <i>Labeling</i>	<i>Pre-Processing</i>		Jumlah <i>Epoch</i>
	<i>Rescale/Resize</i>	Augmentasi	
Roboflow	416 x 416	<i>Yes</i>	

	608 x 608	<i>No</i>	200
	416 x 416		
	608 x 608		

3.7. Pengujian Kelayakan Kuesioner dan Kegunaan Hasil Akhir

3.7.1. Waktu dan Tempat Pengujian

Pengujian kelayakan hasil akhir ini akan dilakukan melalui kuesioner. Kuesioner ini nantinya akan dibagi menjadi dua bagian. Pertama, kuesioner yang ditujukan kepada orang tua yang memiliki anak-anak atau guru-guru di kelompok bermain (PAUD) yang menjadi titik fokus pada penelitian ini. Dan yang kedua, kuesioner akan disebarakan untuk diuji oleh orang-orang selain anak-anak. Kuesioner akan dilakukan dengan menyebarkan *link* hasil akhir dan *link* kuesioner ke masyarakat umum secara *online* selama seminggu.

3.7.2. 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 (Budiyanta, 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

3.7.1.1. 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).

3.7.1.2. 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).

3.7.2. 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

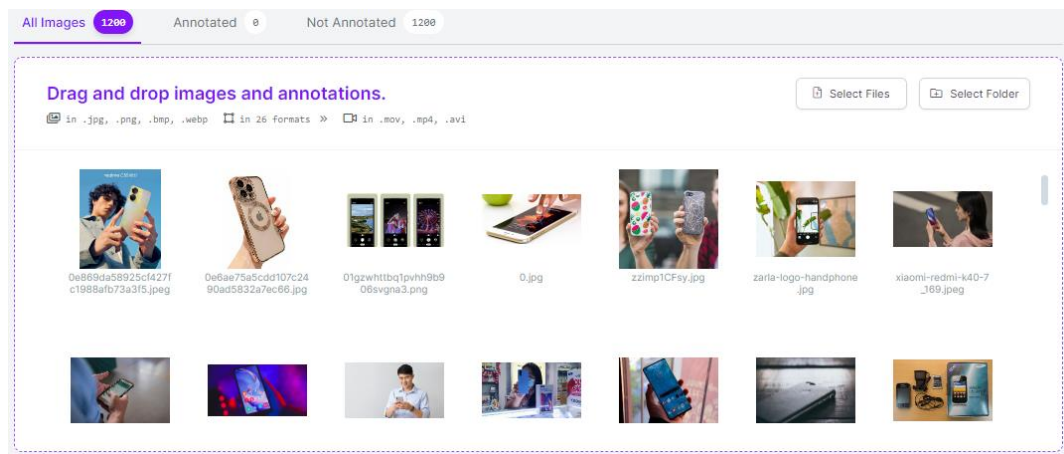
(Kusuma dkk., 2016).

BAB VI

HASIL PENELITIAN DAN PEMBAHASAN

4.1. Pengumpulan Data

Data yang dikumpulkan adalah data-data *random* yang diambil dari internet. Data yang dikumpulkan ini kemudian dipilah sesuai kelas-kelas yang nantinya akan digunakan pada penelitian seperti pada gambar 3.3. Setelah terkumpul, data akan dimasukkan ke dalam proyek 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

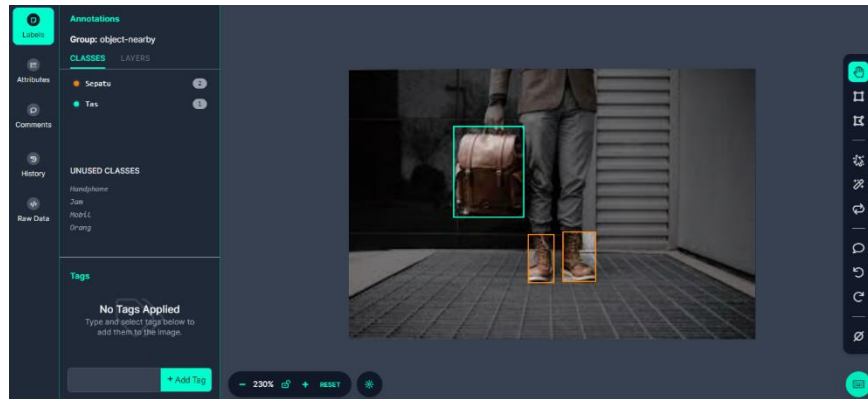
Tahap *pre-processing* pada data akan melalui beberapa tahap seperti yang telah disebutkan pada bab 3.

4.2.1. *Annotate Image*

Pada tahap ini, gambar yang sebelumnya sudah diunggah ke *workshop* Roboflow akan dilakukan *annotate* atau melabelkan 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.

Seperti yang dijelaskan pada bagian 3.3.1 mengenai *labeling*,



Gambar 4. 2: Proses annotate di Roboflow

label atau *bounding box* yang dibuat ini akan menyimpan informasi-informasi yang dibutuhkan algoritma untuk proses pemodelan.

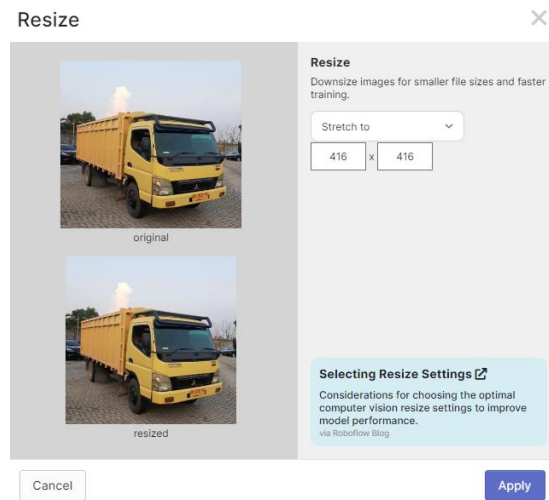


Gambar 4. 3: Data hasil labeling atau bounding box

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 dijadikan dataset. Untuk dapat melakukan *resize* ditahap ini, terlebih dahulu mesti membuat sebuah versi dari dataset. Kemudian gambar akan diubah ukurannya sesuai dengan skenario pengujian pada tabel 3.1.



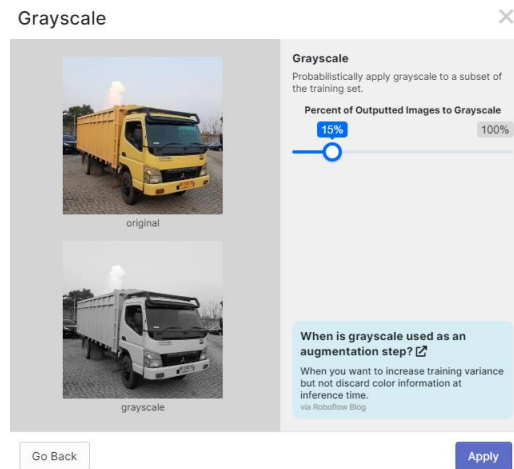
Gambar 4. 4: Proses resize pada dataset

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).

4.2.3. *Augmentasi Image*

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

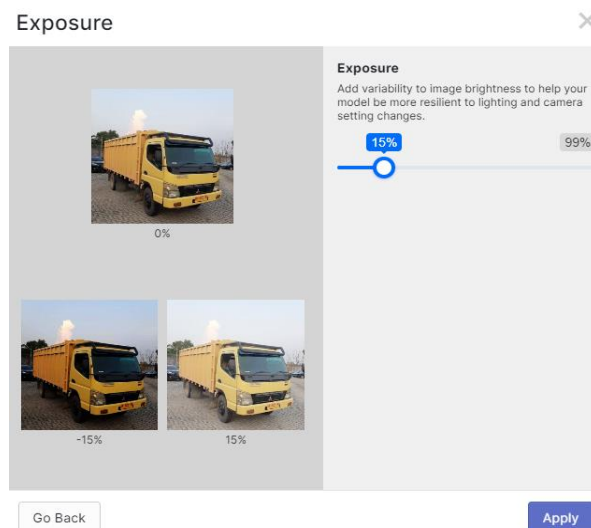
4.2.3.1. *Grayscale*



Gambar 4. 5: Proses augmentasi dataset

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.

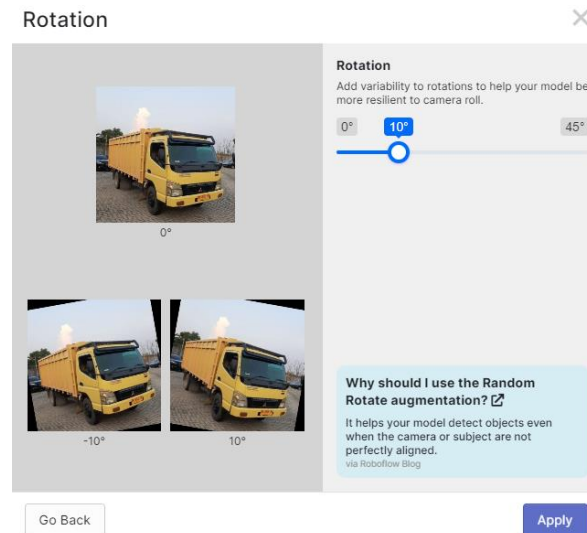
4.2.3.2. *Exposure*



Gambar 4. 6: Proses exposure dataset

Seperti yang dijelaskan pada bab 3 di bagian 3.3.3, untuk *exposure* akan menggunakan tingkatan sebesar 15%.

4.2.3.3. *Rotation*

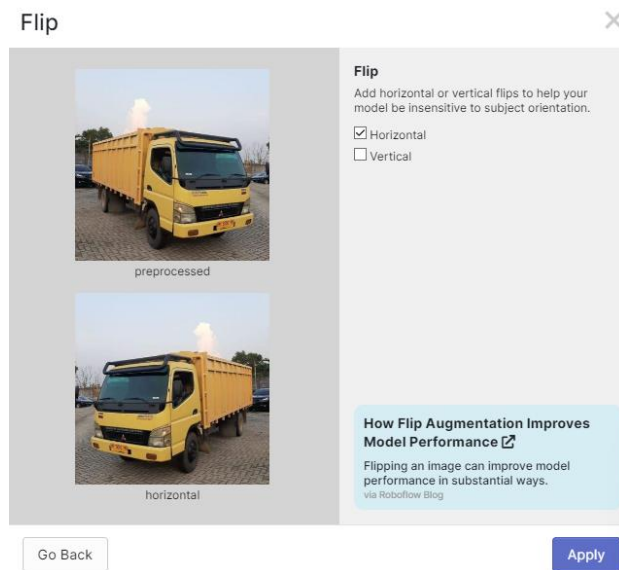


Gambar 4. 7: Proses rotation dataset

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*

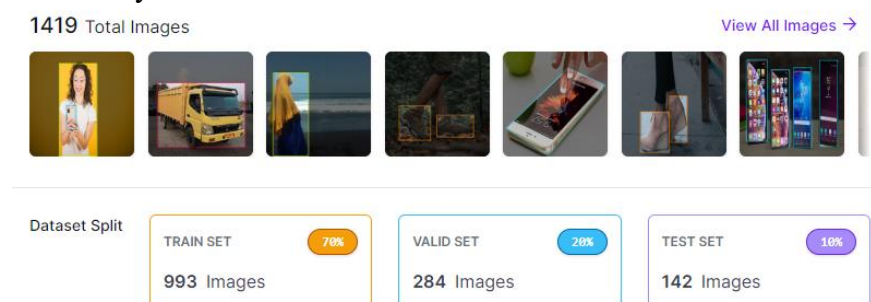
Proses ini nantinya akan membalikan gambar secara horizontal. Proses ini digunakan agar dataset memiliki variasi terhadap data dari beberapa sudut pandang.



Gambar 4. 8: Proses flip dataset

4.2.4. *Data Split*

Pada tahap *data split*, setelah tahapan *pre-processing* 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: Data split oleh Roboflow


Ditahap ini dataset dengan versinya telah siap digunakan ke dalam proyek yang akan dibuat.

4.3. 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).

4.3.1. Menginstall YOLO

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 4.10.



```
pip install ultralytics
```

Gambar 4. 11: Code untuk menginstall library

Setelah mendapatkan *dependency* dari Ultralytics, selanjutnya akan mengimpor YOLO ke dalam proyek yang dibuat dengan menjalankan *code* seperti pada gambar 4.11. Dengan menjalankan



```
from ultralytics import YOLO
```

Gambar 4. 10: Mengimpor YOLO ke dalam proyek

kedua *code* tersebut, YOLO telah siap digunakan ke dalam proyek untuk pembuatan model.

4.3.2. Mengimpor Dataset

Dataset yang sebelumnya sudah dibentuk di Roboflow akan diimpor ke dalam proyek untuk membantu pembuatan model dengan *custom* dataset. Untuk mengimpor dataset, Roboflow sendiri telah

```

!pip install roboflow
from roboflow import Roboflow

rf = Roboflow(api_key="A88agLIxIm30CSF5vNOe")
project = rf.workspace("gabriel-advent-hobto").project(
    "detect-tdz36")
version = project.version(1)
dataset = version.download("yolov8")

```

Gambar 4. 12: Code untuk mengimpor dataset dari Roboflow

memberikan kemudahan dengan *code* yang sudah disiapkan seperti yang dapat dilihat dari gambar 4.12. *Code* yang diberikan ini nantinya akan mengunduh dalam bentuk zip dan mengekstrak

```

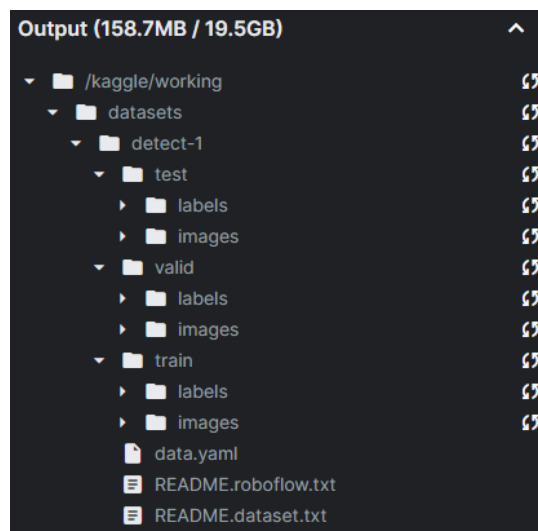
loading Roboflow workspace...
loading Roboflow project...
Dependency ultralytics==8.0.196 is required but found version=8.2.5, to fix: `pip install ultralytics==8.0.196`
Downloading Dataset Version Zip in detect-1 to yolov8:: 100%|██████████| 143885/143885 [00:03<00:00, 41629.65 it/s]
Extracting Dataset Version Zip to detect-1 in yolov8:: 100%|██████████| 6822/6822 [00:00<00:00, 7276.67it/s]

```

Gambar 4. 13: Keterangan jika berhasil mengunduh dan mengekstrak dataset

dataset versi pertama 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 4.13.

Setelah berhasil diekstrak, dapat dilihat pada gambar 4.14



Gambar 4. 14: Hasil ekstraksi dataset

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 digunakan dalam model nantinya, dan beberapa informasi yang relevan. Secara lebih jelas terkait isi dari *file*

```
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. 15: Isi dari file data.yaml

data.yaml, dapat dilihat dari gambar 4.15.

4.3.3. Pelatihan Model

Terdapat dua cara yang dapat digunakan untuk melakukan *training*, yaitu dengan melalui *python environment* dan melalui *Command Line Interface* atau CLI (Jocher & Sergiuwaxmann, 2023). Pada tahap pembuatan model ini akan menggunakan cara CLI sehingga untuk *code* yang digunakan dapat dilihat pada gambar 4.16.

```
data_yaml='/kaggle/working/datasets/detect-1/data.yaml'

!yolo task=detect mode=train model=yolov8m.pt data=data_yaml epochs=200
  imgsz=416 plots=True
```

Gambar 4. 16: Code untuk pembuatan model dengan custom dataset

Pada gambar tersebut, di awal akan menginisialisasi variabel `data_yaml` dengan *path* yang mengarah pada *file* data.yaml. Setelah

itu, program akan menjalankan perintah untuk *training* pada *task detect*. Selanjutnya terdapat beberapa parameter tambahan, seperti pada *code* ini akan menggunakan model yolov8m.pt. Penggunaan parameter model yang sudah dilatih sebelumnya 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). Jenis model yang digunakan ini adalah model yolov8 versi medium (yolov8m.pt). Hal ini dilakukan dengan mengingat bahwa ukuran gambar yang digunakan adalah ukuran medium sehingga dapat membantu meringankan beban komputasi (Zhao & Li, 2020).

Parameter selanjutnya yang diatur adalah *epoch*. Nilai *default* dari *epoch* adalah 100 *epoch*. Berdasarkan skenario pengujian pada tabel 3.1, maka nilai *epoch* akan diatur dengan nilai 200 *epoch*. Selanjutnya untuk parameter *imgsz* (*image size*) akan diatur sesuai dengan ukuran dari dataset yang sebelumnya sudah dilakukan *resize*. Nilai *imgsz* ini akan mempengaruhi bagaimana model YOLO akan memproses gambar dan mendeteksi objek di dalamnya (Jocher & Sergiuwaxmann, 2023). Pada parameter terakhir yang diatur adalah terkait *plots*. Parameter ini akan menyimpan grafik-grafik hasil penelitian sehingga dapat membantu analisis di akhir nantinya. Terlepas dari parameter yang diatur, sebenarnya ada banyak parameter yang dijalankan atau diimplementasi di dalam pelatihan. Namun *value* dari parameter

yang lain akan diatur sesuai dengan nilai *default*. Untuk lebih rinci dapat dilihat pada gambar 4.17.

```

Ultralytics YOLOv8.0.20 Python-3.10.13 torch-2.1.2 CUDA:0 (Tesla T4, 15102MiB)
yolo/engine/trainer: task=detect, mode=train, model=yolov8m.yaml, data=/kaggle/working/datasets/detect-2/data.yaml,
epochs=200, patience=50, batch=16, imgsz=416, save=True, cache=False, device=, workers=8, project=None, name=None,
exist_ok=False, pretrained=False, optimizer=SGD, verbose=True, seed=0, deterministic=True, single_cls=False, image
weights=False, rect=False, cos_lr=False, close_mosaic=10, resume=False, overlap_mask=True, mask_ratio=4, dropout=Fa
lse, val=True, save_json=False, save_hybrid=False, conf=0.001, iou=0.7, max_det=300, half=False, dnn=False, plots=T
rue, source=ultralytics/assets/, show=False, save_txt=False, save_conf=False, save_crop=False, hide_labels=False, h
ide_conf=False, vid_stride=1, line_thickness=3, visualize=False, augment=False, agnostic_nms=False, classes=None, r
etina_masks=False, boxes=True, format=torchscript, keras=False, optimize=False, int8=False, dynamic=False, simplify
=False, opset=17, workspace=4, nms=False, lrf=0.01, lrf=0.01, momentum=0.937, weight_decay=0.001, warmup_epochs=3.
0, warmup_momentum=0.8, warmup_bias_lr=0.1, box=7.5, cls=0.5, dfl=1.5, fl_gamma=0.0, label_smoothing=0.0, nbs=64, h
sv_h=0.015, hsv_s=0.7, hsv_v=0.4, degrees=0.0, translate=0.1, scale=0.9, shear=0.0, perspective=0.0, flipud=0.0, fl
ip_lr=0.5, mosaic=1.0, mixup=0.1, copy_paste=0.1, cfg=None, v5loader=False, save_dir=runs/detect/train2

```

Gambar 4. 17: Detail yang digunakan pada pelatihan

Pada tiap *epoch* nantinya akan ditampilkan nilai-nilai yang dapat digunakan untuk menganalisis performa dari model yang dibuat. Seperti yang terlihat pada gambar 4.18, terdapat nilai-nilai

Epoch	GPU_mem	box_loss	cls_loss	dfl_loss	Instances	Size
1/200	3.42G	1.319	1.705	1.411	11	416: 1
	Class	Images	Instances	Box(P	R	mAP50 m
	all	284	584	0.753	0.799	0.823 0.566
	Handphone	284	150	0.923	0.867	0.934 0.761
	Jam	284	40	0.575	0.825	0.825 0.578
	Mobil	284	75	0.898	0.867	0.87 0.65
	Orang	284	124	0.705	0.847	0.817 0.466
	Sepatu	284	134	0.711	0.644	0.709 0.411
	Tas	284	61	0.704	0.742	0.781 0.531
Epoch	GPU_mem	box_loss	cls_loss	dfl_loss	Instances	Size
2/200	4.26G	1.227	1.122	1.29	7	416: 1
	Class	Images	Instances	Box(P	R	mAP50 m
	all	284	584	0.782	0.772	0.811 0.538
	Handphone	284	150	0.899	0.888	0.933 0.751
	Jam	284	40	0.732	0.75	0.834 0.635
	Mobil	284	75	0.932	0.813	0.855 0.58
	Orang	284	124	0.739	0.753	0.774 0.425
	Sepatu	284	134	0.754	0.593	0.689 0.397
	Tas	284	61	0.635	0.836	0.783 0.436
Epoch	GPU_mem	box_loss	cls_loss	dfl_loss	Instances	Size
3/200	4.27G	1.303	1.254	1.355	15	416: 1
	Class	Images	Instances	Box(P	R	mAP50 m
	all	284	584	0.6	0.683	0.66 0.419
	Handphone	284	150	0.441	0.947	0.808 0.639
	Jam	284	40	0.631	0.641	0.658 0.458
	Mobil	284	75	0.759	0.733	0.76 0.5
	Orang	284	124	0.586	0.589	0.604 0.29
	Sepatu	284	134	0.658	0.53	0.582 0.311
	Tas	284	61	0.526	0.656	0.55 0.319

Gambar 4. 18: Output dari rincian tiap epoch

seperti nilai `box_loss`, `cls_loss`, `dfl_loss`, `mAP50`, dan nilai-nilai yang lain seperti yang ditampilkan pada gambar 4.18. Nilai-nilai ini juga akan disimpan ke dalam sebuah *file* csv yang nanti akan digunakan untuk menganalisis performa model.

Pelatihan yang dilakukan sebanyak 200 *epoch* ini akan dilakukan dengan *patience* dengan nilai 50 *epoch*. Sehingga ketika dalam 50 *epoch* terakhir tidak mengalami peningkatan pada

performa maka pelatihan akan dihentikan. Hal ini dilakukan agar membantu untuk mencegah *overfitting*.

4.4. Analisis Hasil Pengujian

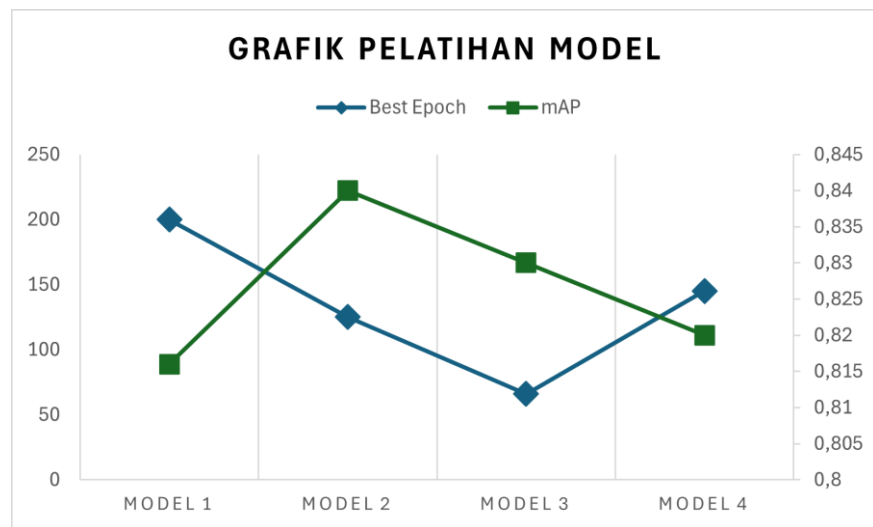
4.4.1. Hasil Pelatihan Model

Berdasarkan skenario pengujian pada tabel 3.1, maka dilakukan pelatihan model sebanyak empat kali. Hasil dari pelatihan ini akan ditampilkan pada tabel 4.1. Pada tabel tersebut dapat dilihat

Tabel 4. 1: Hasil penelitian model berdasarkan skenario pengujian

Rangkuman Hasil					
Model	Epoch	mAP	Epoch Best	mAP	Jenis Dataset
Model 1	200	0,816	-	-	608 & augmentasi
Model 2	175	0,821	125	0,84	416 & augmentasi
Model 3	116	0,812	66	0,83	416 !augmentasi
Model 4	195	0,833	145	0,820	608 !augmentasi

bahwa pada pelatihan model pertama dengan dataset berukuran 608x608 dan memiliki augmentasi melalui pelatihan dengan 200 *epoch* dan mendapatkan mAP sebesar 81%, selanjutnya pelatihan pada model kedua berhenti pada *epoch* ke-175 dengan mAP 82% dan mAP terbaiknya didapatkan pada *epoch* ke-125 dengan mAP sebesar 84%. Berhentinya pelatihan ini karena selama 50 *epoch* terakhir tidak ada peningkatan mAP. Lebih lanjut lagi pada pemodelan ketiga mendapat mAP terbaik pada *epoch* ke-66 dengan mAP sebesar 83% dan pada pemodelan keempat pada *epoch* ke-145 didapatkan mAP terbaiknya dengan nilai 82%. Secara lebih ringkas dapat dilihat pada gambar grafik 4.19.



Gambar 4. 19: Grafik pelatihan model

Secara garis besar hasil yang didapatkan sudah cukup bagus dengan mAP terbesar didapatkan pada model kedua. Namun secara detail dan dilihat dari grafik tersebut, perbedaan mAP tiap model tidak memiliki perbedaan yang signifikan sehingga belum dapat menentukan apakah model tersebut merupakan model dengan hasil yang memuaskan (Karna dkk., 2023; Sholahuddin dkk., 2023). Perubahan yang kurang signifikan ini juga dapat dilihat pada perbandingan *precision* dan *recall* tiap model pada tabel 4.2.

Tabel 4. 2: Tabel perbandingan precision, recall, dan validasi tiap model

Model	Precision	Recall	Validasi
model 1	0,84	0,75	0,821
model 2	0,85	0,77	0,839
model 3	0,83	0,78	0,839
model 4	0,86	0,77	0,846

Pada proses validasi data pun tidak menemukan perbandingan yang signifikan di antara keempat model.

Berdasarkan skenario pengujian yang digunakan untuk melihat seberapa besar mAP dari model yang dihasilkan jika ukuran

gambar dinaikkan menjadi 608x608 baik itu menggunakan augmentasi maupun tanpa augmentasi, terbukti bahwa pada penelitian ini dengan menggunakan ukuran 416x416 baik itu memiliki augmentasi maupun tidak memiliki augmentasi dapat menghasilkan mAP yang lebih besar dari yang dihasilkan oleh ukuran gambar 608x608. Secara lebih lanjut, pada model 416x416 yang menggunakan augmentasi mampu menghasilkan mAP yang lebih baik dibandingkan dengan tanpa menggunakan augmentasi. Hal ini cukup membuktikan bahwa dengan menambahkan augmentasi ke dalam dataset mampu meningkatkan kualitas model sehingga lebih terlatih dalam mengenali objek nantinya.

Namun, dari hasil analisis sebelumnya terkait hasil pelatihan, dapat dilihat bahwa keempat model yang dievaluasi, yaitu model 1, model 2, model 3, dan model 4, kurang menunjukkan perbedaan yang signifikan dalam kinerja modelnya. Hal ini dapat dilihat dari nilai mAP, *precision*, *recall*, dan validasi yang relatif serupa antara keempat model tersebut. Hal ini menyebabkan kesulitan dalam menentukan model yang baik. Oleh karena itu, akan dilakukan pengembangan model dengan melibatkan penyesuaian *hyperparameter* untuk memperbaiki dan mengoptimalkan model sehingga dapat menemukan model yang benar-benar baik dalam pengenalan objek.

4.4.2. Pengujian *Hyperparameter*

Pada pengujian *hyperparameter* ini akan dilakukan pelatihan model seperti yang sudah dilakukan sebelumnya namun dengan melibatkan beberapa parameter yang digunakan dalam pelatihan model YOLO. Pengujian model menggunakan *hyperparameter* ini dilakukan untuk melihat seberapa besar model dapat berkembang dari sebelumnya. Untuk menentukan model yang menghasilkan mAP terbaik, konsep *grid search* akan diterapkan dalam kode untuk membandingkan setiap mAP yang dihasilkan (Sholahuddin dkk., 2023).

Berdasarkan penarikan kesimpulan terkait penggunaan augmentasi dalam dataset, maka pengujian ini hanya akan melibatkan dataset yang memiliki augmentasi dan akan dilakukan variasi pada ukuran datasetnya. Hal ini dilakukan untuk tetap menjaga variasi yang beragam dalam dataset dan variasi pada ukuran akan membantu dalam menemukan ukuran gambar yang sesuai untuk penelitian ini.

Pada pengujian menggunakan *hyperparemeter* ini akan menggunakan beberapa parameter yang akan membantu untuk pengoptimalan model seperti yang dapat dilihat pada tabel 4.3. (Karna dkk., 2023; Sandhya & Kashyap, 2024; Sholahuddin dkk., 2023; Tamang dkk., 2023).

Tabel 4. 3: Rincian hyperparameter

No.	Hyperparameter	Variasi
1.	Model YOLO	yolov8n.pt
		yolov8s.pt
		yolov8m.pt
2.	Ukuran Gambar	240
		256
		416
3.	Batch	16
		32
		64
4.	Momentum	0,937
		0,8
5.	Weight Decay	0,0005
6.	Learning Rate	0,0001
7.	Optimizer	adam
		adamW

Untuk melakukan pengujian di tiap parameter, maka sebelumnya akan mengombinasikan tiap parameter sehingga nantinya akan menjalankan *code* secara otomatis. Untuk mengombinasikan tiap parameter dapat dilihat pada gambar 4.20.

```
import subprocess
from itertools import product

param_grid = {
    'model': ['yolov8n.pt', 'yolov8s.pt', 'yolov8m.pt'],
    'imgsz': [240, 256, 416],
    'batch': [16, 32, 64],
    'momentum': [0.937, 0.8],
    'optimizer' : ['adam', 'adamW']
}

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

Gambar 4. 20: Code untuk mengombinasikan tiap parameter

pada *code* tersebut setelah tiap parameter dimasukan ke dalam variabel `param_list`, selanjutnya variabel tersebut akan dikombinasikan dengan bantuan dari *library* *itertools*. Setelah itu, akan dilakukan perulangan tiap `param_combinations` untuk memasukkan kombinasi-kombinasi tersebut ke dalam *code* untuk *training* menggunakan *hyperparameter*. Seperti yang terlihat pada

gambar 4.21, pada *code* tersebut setiap *command* akan diproses pada variabel *result*. Nantinya *code* ini akan menerapkan konsep *grid search* untuk mencari parameter terbaik dengan membandingkan mAP yang dihasilkan dari tiap parameternya. Tiap mAP yang dihasilkan akan dibandingkan dengan nilai dari variabel *best_map*. Jika nilainya lebih besar maka nilai mAP dan kombinasi parameternya akan disimpan. Hal inilah yang kemudian akan membantu dalam proses pencarian mAP tertinggi dengan parameter terbaiknya.

```
best_map = 0
best_params = {}

# Loop over the grid of hyperparameters
for params in param_combinations:
    model, imgsz, optimizer, batch, momentum = params
    command = f"yolo detect train \
data='/kaggle/working/datasets/detect-2/data.yaml' \
model='{model}' \
epochs=200 \
imgsz='{imgsz}' \
optimizer='{optimizer}' \
lr=0.0001 \
batch='{batch}' \
plots=True \
weight_decay=0.0005 \
momentum='{momentum}' \
save=True"

    # Run the command and get the validation loss
    results = subprocess.run(command, shell=True, capture_output=True, text=True)
    if results['val/mAP_0.5'] > best_map:
        best_map = results['val/mAP_0.5']
        best_params = params

print("Best mAP: {:.4f} with Parameters: {}".format(best_map, best_params))
```

Gambar 4. 21: Code untuk melakukan training dan mencari best parameter

4.4.3. Analisis Uji *Hyperparameter*

Uji coba dengan menggunakan *hyperparameter* ini membuat model mengalami kenaikan yang cukup signifikan jika dibandingkan dengan sebelumnya dengan melihat perbandingan mAP yang naik sekisar 3% – 4%. Pada pencarian parameter terbaik dengan menerapkan konsep seperti *grid search*, berhasil

```
!yolo detect train \
  data='/kaggle/working/datasets/detect-2/data.yaml' \
  model=yolov8m.pt \
  epochs=200 \
  imgsz=240 \
  optimizer='Adam' \
  lr0=0.0001 \
  batch=64 \
  plots=True \
  weight_decay=0.0005 \
  momentum=0.937 \
  save=True
```

Gambar 4. 22: Kombinasi parameter dengan hasil mAP tertinggi

mendapatkan mAP sebesar 0.88971 atau 88% dengan kombinasi parameter seperti pada gambar 4.22. Model yang dioptimalkan menggunakan *hyperparameter* ini ketika dilakukan validasi pun mendapatkan hasil yang bagus seperti yang terlihat pada gambar 4.23. Pada hasil validasi ini dapat dilihat bahwa selain objek sepatu,

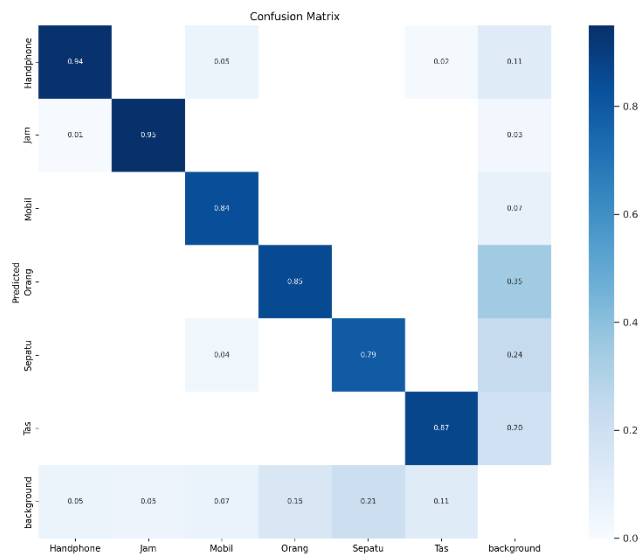
```
segment dataset, not a detect-segment mixed dataset.
```

Class	Images	Instances	Box(P	R	mAP50	m
all	284	584	0.88	0.827	0.879	0.648
Handphone	284	150	0.912	0.913	0.95	0.783
Jam	284	40	0.921	0.925	0.915	0.726
Mobil	284	75	0.934	0.827	0.912	0.717
Orang	284	124	0.825	0.762	0.824	0.486
Sepatu	284	134	0.841	0.701	0.789	0.488
Tas	284	61	0.849	0.836	0.885	0.687

```
Speed: 0.2ms pre-process, 8.8ms inference, 0.0ms loss, 6.2ms post-process per image
Saving runs/detect/val/predictions.json...
```

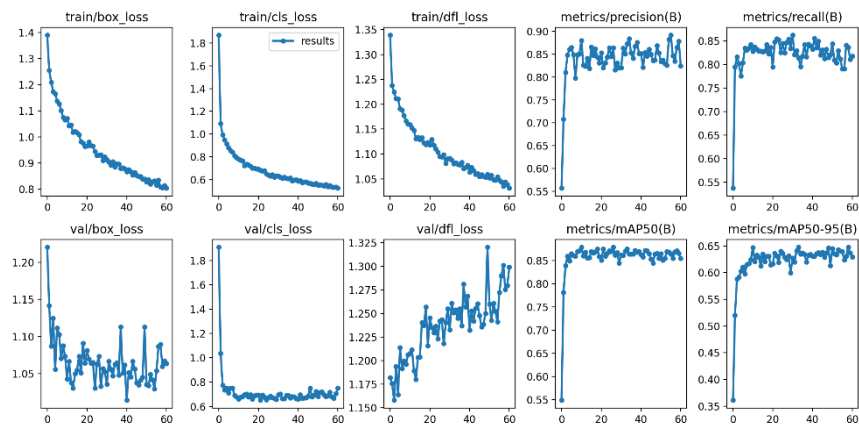
Gambar 4. 23: Hasil validasi terhadap model

model telah dapat memprediksi tiap objek dengan baik yang ditunjukkan dengan hasil di atas 85%. Hal ini pun diperkuat dengan *confusion matrix* yang dapat dilihat dari gambar 4.24. Dari *confusion matrix* tersebut dapat dipastikan bahwa model memiliki *precision* dan *recall* yang tinggi untuk sebagian besar kelas objek, seperti handphone, jam, mobil, orang, dan tas. Namun dibalik itu, model ini memiliki *precision* dan *recall* yang rendah untuk kelas sepatu yang sering dideteksi sebagai *background*.



Gambar 4. 25: Confusion matrix

Secara lebih lanjut, dapat dilihat pada gambar 4.25 yang menampilkan garis besar dari beberapa grafik yang menentukan perkembangan model. Pada gambar 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.



Gambar 4. 24: Kumpulan grafik evaluasi

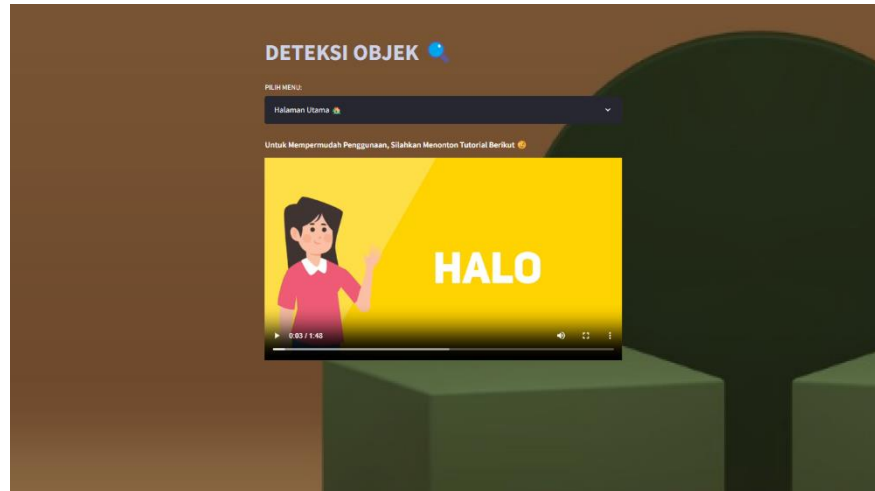
Dengan ini, uji yang dilakukan menggunakan *hyperparameter* mampu memperbaiki model yang sebelumnya. Karena itu, pada penelitian ini akan menggunakan model yang menghasilkan mAP terbaik, yaitu model *hyperparameter* yang menghasilkan mAP sebesar 88%.

4.5. Implementasi Aplikasi Deteksi Objek

4.5.1. *Interface* Aplikasi

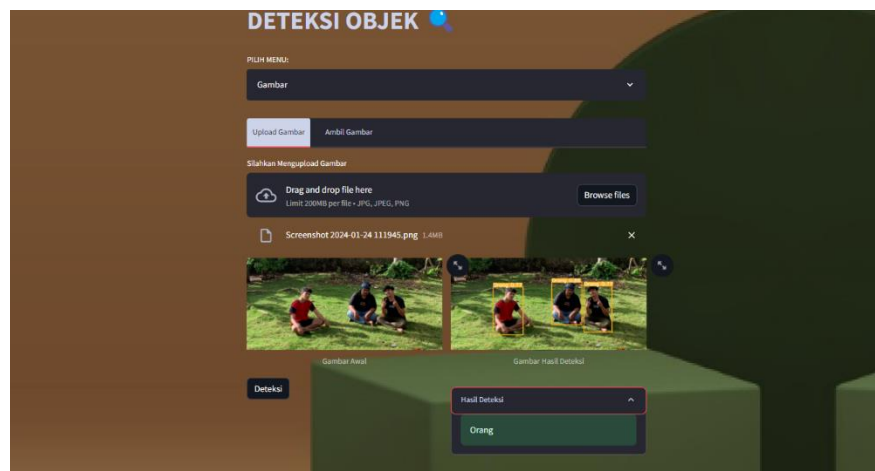
Implementasi model ke dalam *interface* akan dibantu dengan *framework* Streamlit. Alasan mengapa menggunakan *framework* ini adalah karena pihak Streamlit telah menyediakan *cloud* khusus bagi para anggota komunitas untuk *deploy* hasil kerjanya (Sholahuddin dkk., 2023). Pada *website* ini akan terdapat 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 mempermudah proses deteksi dengan model yang sudah dibuat sebelumnya.

Berikut adalah *interface* yang dibuat di Streamlit. Ketika pertama kali diakses maka akan menampilkan halaman seperti pada gambar 4.26. Pada halaman ini akan menampilkan *dropdown* dari



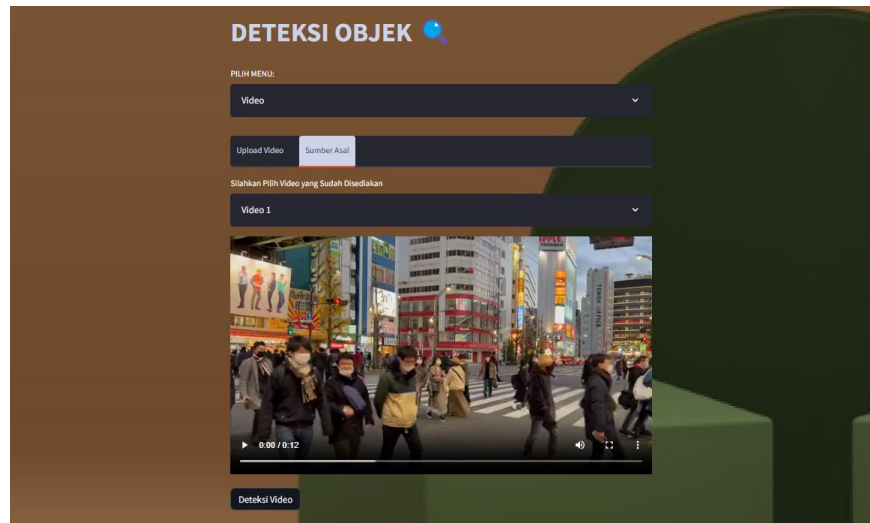
Gambar 4. 26: Tampilan halaman pertama ketika website diakses

menu-menu yang disediakan juga menampilkan video tutorial penggunaan *website*. Berikutnya, tampilan pada gambar 4.27 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 fitur yang sama. Hal ini pun berlaku pada tampilan



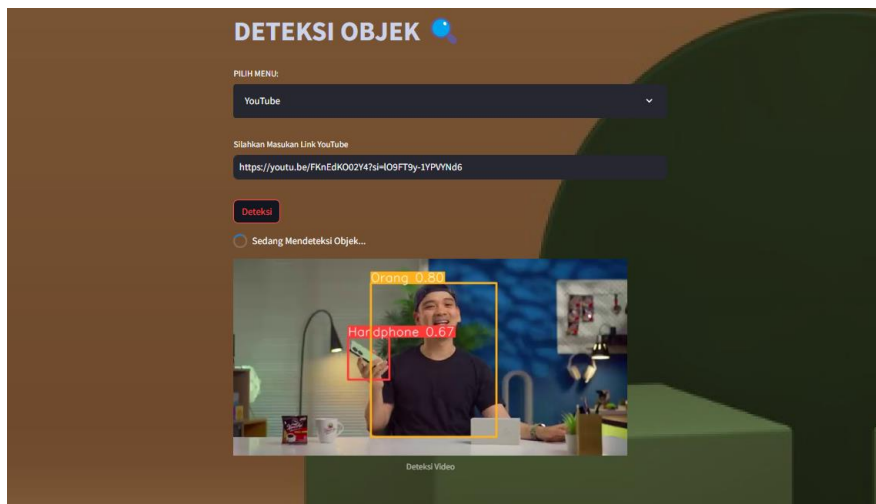
Gambar 4. 27: Tampilan halaman jika memilih menu gambar

jika memilih menu video seperti yang terlihat pada gambar 4.28. pada menu ini telah disediakan lima video yang dapat langsung



Gambar 4. 28: Tampilan jika memilih menu video

digunakan untuk deteksi sehingga user tidak perlu mengunggah videonya sendiri. Selanjutnya pada gambat 4.29 dapat dilihat tampilan jika memilih menu youtube. Pada menu ini, user hanya



Gambar 4. 29: Tampilan jika memilih menu youtube

perlu memasukan link YouTube yang sudah disalin dan menekan tombol 'deteksi'. Terakhir adalah tampilan jika user memilih menu Real-Time. Seperti yang terlihat pada gambar 4.30, menu ini akan

mendeteksi secara realtime menggunakan kamera baik ponsel maupun komputer.



Gambar 4. 30: Tampilan ketika memilih menu Real-Time

4.5.2. Implementasi *Code* untuk Deteksi

Terdapat beberapa *function* yang diimplementasi untuk menunjang proses deteksi di tiap menu. Pada gambar 4.31 adalah

```
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. 31: Code untuk mendeteksi gambar yang diupload

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

```
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. 32: Function untuk menampilkan gambar hasil deteksi

tersebut. Berikut pada gambar 4.32 adalah *function* yang digunakan untuk melakukan deteksi dan menampilkan hasil deteksi ke *frame* yang sudah disiapkan sebelumnya. Perlu diperhatikan bahwa nilai *confidence* yang digunakan pada tiap deteksi akan menggunakan *confidence* sebesar 0.5. Pada gambar 4.33 merupakan *function* untuk mengisi *link* YouTube yang disalin. Ketika tombol deteksi maka program akan *capture* tiap *frame* video dengan bantuan *library*

```
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. 33: Function untuk mencari video dari link youtube

open-cv dan mengirimnya ke *function* showDetectFrame. Selanjutnya terdapat *function* untuk memproses video yang diunggah user. Pada *function* yang dapat dilihat dari gambar 4.34

```

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. 34: Function untuk memproses video yang diupload

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.

Function selanjutnya yang digunakan adalah *function* `take_picture` yang dapat dilihat pada gambar 4.35. *Function* ini akan mengambil foto atau gambar secara langsung menggunakan bantuan `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

```
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. 35: Function untuk mengambil foto atau gambar secara langsung melakukan deteksi dan menampilkan hasil. *Function* terakhir yang digunakan adalah *function* untuk membantu proses *realtime*. 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

```
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. 36: Function untuk realtime adalah konfigurasi `rtc`. Pada perubahan infrastruktur terbaru terkait `webrtc_streamer`, ditetapkan harus menggunakan bantuan Twilio agar dapat menjalankan *live* tersebut. Selain itu, terdapat juga class

VideoTransformer seperti yang terlihat pada gambar 4.37 yang digunakan untuk mendeteksi secara *realtime* tiap *frame* dari hasil

```
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. 37: Code untuk mendeteksi realtime

live tersebut.

Penerapan-penerapan *code* ini mampu menjalankan fitur-fitur yang sudah dibuat dengan Streamlit. *Code* ini akan disimpan di sebuah *repository* GitHub dan dihubungkan pada Cloud Community Share agar dapat diakses secara *online* baik dari ponsel maupun komputer.

4.6. Pengujian Kegunaan Aplikasi

4.6.1. Uji Validitas

Uji validitas akan dilakukan sebelum kuesioner akan disebarkan secara resmi. Hal ini dilakukan untuk mengetahui kelayakan sebuah instrumen yang akan diuji. Pada pengujian ini, kuesioner dan *website* akan disebarkan secara tertutup dengan mengirimkan *link* melalui media sosial WhatsApp kepada kenalan penulis. Adapun hasil dari kuesioner pengujian dari 25 responden dapat dilihat pada tabel 4.4.

Tabel 4. 4: Hasil kuesioner pengujian

R	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q11	T
R1	4	4	4	4	3	4	4	3	4	3	3	40
R2	4	4	3	4	3	3	3	3	4	3	4	38

R3	4	4	4	3	4	4	4	4	4	4	4	43
R4	4	4	3	4	4	4	4	4	4	3	4	42
R5	3	3	3	3	4	3	3	3	3	3	3	34
R6	3	3	3	3	3	3	3	3	3	3	3	33
R7	4	3	4	4	4	4	4	4	4	4	4	43
R8	4	4	3	3	2	4	4	4	3	3	3	37
R9	4	4	4	4	4	4	4	4	4	4	4	44
R10	4	4	4	3	4	3	4	3	4	3	4	40
R11	4	4	4	3	4	4	4	4	4	3	4	42
R12	4	4	3	4	4	4	4	4	4	4	4	43
R13	3	3	3	3	3	3	3	3	3	3	3	33
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

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*

<i>rHitung</i>											
0,83	0,64	0,61	0,56	0,49	0,66	0,77	0,63	0,84	0,52	0,80	

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 20 maka *rTabel* akan bernilai 0,423. 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	0,83	0,64	0,61	0,56	0,49	0,66	0,77	0,63	0,84	0,52	0,80
rT	0,42	0,42	0,42	0,42	0,42	0,42	0,42	0,42	0,42	0,42	0,42
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 siap untuk diuji secara umum.

4.6.2. 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 .

$$r = \left(\frac{11}{11 - 1} \right) \left(1 - \frac{2.797}{13.187} \right)$$

$$r = 0.867$$

Dari perhitungan tersebut diketahui bahwa nilai r adalah 0.867. Selanjutnya jika dikaitkan dengan tabel 3.4 tentang tingkatan atau interval reliabilitas, maka diketahui bahwa instrumen tersebut memiliki reliabilitas sangat tinggi dengan nilai $r = 0.867$.

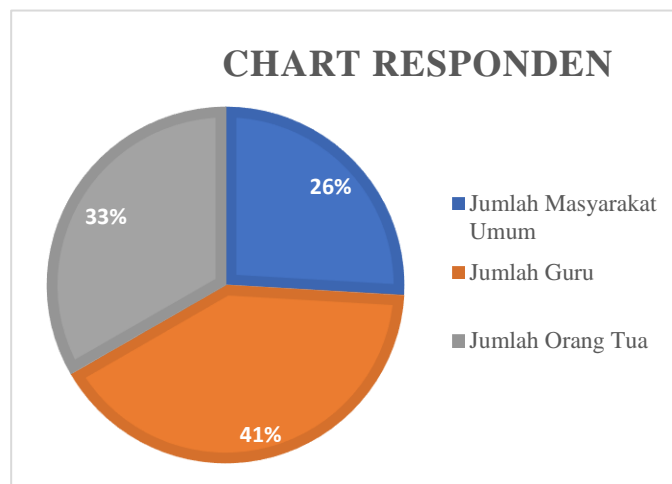
4.6.3. Uji Usability

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. 38: 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	4	4	4	4	4	4	4	4	4	44
R2	OT	4	3	4	4	3	3	3	4	4	3	3	38
R3	G	4	4	4	4	4	4	4	4	4	4	4	44
R4	G	4	4	4	4	4	4	4	4	4	4	4	44
R5	G	4	4	4	4	4	4	4	4	4	4	4	44
R6	OT	4	4	4	4	4	4	4	4	4	4	4	44
R7	OT	4	4	4	4	4	4	4	4	4	4	4	44
R8	OT	4	4	4	4	4	4	4	4	4	4	4	44
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	4	4	4	4	4	4	4	44
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	4	4	4	4	4	4	4	4	4	44
R15	OT	3	4	4	3	4	3	3	4	3	3	4	38
R16	OT	4	4	4	4	4	4	4	4	4	4	4	44
R17	OT	4	4	4	4	4	4	4	4	4	4	4	44
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.

$$usability = \frac{44}{44} \times 100$$

$$usability = 100$$

Dari perhitungan tersebut dapat dilihat bahwa *usability* atau kegunaan dari responden pertama mendapat nilai 100. 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	44	100	SL	R15	38	86,4	SL
R2	38	86,4	SL	R16	44	100	SL
R3	44	100	SL	R17	44	100	SL
R4	44	100	SL	R18	41	93,2	SL
R5	44	100	SL	R19	42	95,5	SL
R6	44	100	SL	R20	41	93,2	SL
R7	44	100	SL	R21	41	93,2	SL
R8	44	100	SL	R22	41	93,2	SL
R9	39	88,6	SL	R23	39	88,6	SL
R10	43	97,7	SL	R24	39	88,6	SL
R11	44	100	SL	R25	41	93,2	SL
R12	39	88,6	SL	R26	41	93,2	SL
R13	40	90,9	SL	R27	40	90,9	SL
R14	44	100	SL	Rata – Rata		94,9	SL

Dari hasil tabel di atas dapat dilihat bahwa dari 27 responden kuesioner tersebut memberikan rata-rata nilai 97,9% dengan total keseluruhan sebesar 1127 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 keempat macam pengujian yang dilakukan berdasarkan tabel skenario, model yang dihasilkan memiliki perbedaan yang kurang signifikan dengan rata-rata mAP yang dihasilkan sebesar 0,831 atau 83%. Dari hasil ini dilakukan pengembangan dengan melibatkan *hyperparameter* dan mendapatkan mAP sebesar 0,88971 atau 88%.

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. Selanjutnya, menerapkan *hyperparameter optimization*

dalam pencarian parameter terbaik dengan menggunakan *random search*, dan *bayesian search*, atau *optimization* yang lainnya untuk mendapatkan konfigurasi parameter yang dapat memberikan hasil yang baik.

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>
- 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>

Gajalakshmi, P., Satyanarayana, J. V., Venkat Reddy, G., & Dhavale, S. (2020). Detection of Strategic Targets of Interest in Satellite Images using YOLO. *2020 4th International Conference on Computer, Communication and Signal Processing (ICCCSP)*, 1–5. <https://doi.org/10.1109/ICCCSP49186.2020.9315197>

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>
- 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.
- 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 Muslih, 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>
- Tamang, S., Sen, B., Pradhan, A., Sharma, K., & Singh, V. K. (2023). Enhancing COVID-19 Safety: Exploring YOLOv8 Object Detection for Accurate Face Mask Classification. *International Journal of Intelligent Systems and Applications in Engineering IJISAE*, 11(2), 892–897. www.ijisae.org
- 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>