

PENDETEKSIAN OBJEK HURUF LONTARA UNTUK LITERASI KE TEKS LATIN

SKRIPSI

Diajukan Sebagai Salah Satu Syarat Untuk Mendapatkan Gelar Sarjana Komputer
(S.Kom) Program Studi Informatika



MOHAMMAD RAMADHAN ZAINUDDIN
NIM 105841113520

**PROGRAM STUDI INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS MUHAMMADIYAH MAKASSAR
2025**



FAKULTAS TEKNIK

PENGESAHAN

Skripsi atas nama Mohammad Ramadhan Zainuddin dengan nomor induk Mahasiswa 105 84 11135 20, dinyatakan diterima dan disahkan oleh Panitia Ujian Tugas Akhir/Skripsi sesuai dengan Surat Keputusan Dekan Fakultas Teknik Universitas Muhammadiyah Makassar Nomor : 0004/SK-Y/55202/091004/2025, sebagai salah satu syarat guna memperoleh gelar Sarjana Komputer pada Program Studi Informatika Fakultas Teknik Universitas Muhammadiyah Makassar pada hari Sabtu, 30 Agustus 2025.

Panitia Ujian :

1. Pengawas Umum

a. Rektor Universitas Muhammadiyah Makassar

Dr. Ir. H. Abd. Rakhim Nanda, ST.,MT.,IPU

b. Dekan Fakultas Teknik Universitas Hasanuddin

Prof. Dr. Eng. Muhammad Isran Ramli, S.T., M.T., ASEAN, Eng

Makassar,

17 Rabi'ul Awal 1447 H

10 September 2025 M

2. Pengaji

a. Ketua : Prof. Dr.Ir. Hafsa Nirwana, M.T.

b. Sekertaris : Darniati, S.Kom., M.T.

3. Anggota

: 1. Muhyiddin A M Hayat, S.Kom., M.T

2. Rizki Yusliana Bakti, S.T.,M.T

3. Lukman, S.Kom., M.T

Pembimbing I

Pembimbing II

Fahrim Irhamna Rahman, S.Kom.,M.T

Titin wahyuni, S.Pd., M.T

Dekan



Gedung Menara Iqra Lantai 3

Jl. Sultan Alauddin No. 259 Telp. (0411) 866 972 Fax (0411) 865 588 Makassar 90221

Web: <https://teknik.unismuh.ac.id/>, e-mail: teknik@unismuh.ac.id





HALAMAN PENGESAHAN

Tugas Akhir ini diajukan untuk memenuhi syarat ujian guna memperoleh gelar Sarjana Komputer (S.Kom) Program Studi Informatika Fakultas Teknik Universitas Muhammadiyah Makassar.

Judul Skripsi : **PENDETEKSIAN OBJEK HURUF LONTARA UNTUK LITERASI KE TEKS LATIN**

Nama : MOHAMMAD RAMADHAN ZAINUDDIN

Stambuk 105 84 11135 20

Makassar, 10 September 2025

Pembimbing I

Pembimbing II

Fahrim Irhamna Rahman, S.Kom.,M.T

Titin wahyuni, S.Pd., M.T

Mengetahui,

Ketua Prodi Informatika



*Rizki Yusliana Bakti, S.T., M.T.

NBM: 1307 284



DAFTAR ISI

HALAMAN SAMPUL	i
HALAMAN PENGESAHAN.....	ii
DAFTAR ISI	iii
BAB I PENDAHULUAN	1
A. Latar Belakang	1
B. Rumusan Masalah	4
C. Tujuan Penelitian	4
D. Manfaat Penelitian	4
E. Ruang Lingkup Penelitian	5
F. Sistematika Penulisan	5
BAB II TINJAUAN PUSTAKA.....	6
A. Landasan Teori	6
B. Penelitian Terkait	14
C. Kerangka Pikir	21
BAB III METODE PENELITIAN	22
A. Tempat Dan Waktu Penelitian	22
B. Alat dan Bahan	22
C. Perancangan Sistem	23
D. Teknik Pengujian Sistem	27
E. Teknik Analisis Data	29
BAB IV HASIL PENELITIAN DAN PEMBAHASAN.....	31
A. Pembuatan Model.....	31

1. Pengambilan Dataset.....	31
2. Pelabelan Gambar.....	33
3. Pembagian Dataset.....	35
4. Training.....	38
B. Pengujian Sistem.....	44
BAB V PENUTUP.....	53
A Kesimpulan.....	53
B. Saran.....	53
DAFTAR PUSTAKA.....	55



ABSTRAK

Kemajuan yang cepat dalam teknologi, terutama dalam bidang komputer, telah memainkan peranan yang besar dalam mendukung inovasi sistem pengenalan karakter secara otomatis, termasuk untuk aksara Lontara'. Salah satu tugas utama dalam penelitian ini adalah merancang penggunaan algoritma deteksi objek YOLOv8 dengan cara yang optimal agar dapat mengenali dan mengklasifikasikan karakter aksara Lontara' dengan akurasi yang tinggi dan kinerja yang dapat diandalkan. Oleh karena itu, tujuan utama dari penelitian ini adalah untuk mengevaluasi sejauh mana efektivitas algoritma YOLOv8 dalam menunjang proses pengenalan citra serta klasifikasi otomatis huruf-huruf dalam aksara Lontara'.

Studi ini dilakukan di Universitas Muhammadiyah Makassar dengan menggunakan metode eksperimen yang mengandalkan teknik pembelajaran mesin sebagai dasar utama dalam menciptakan sistem untuk mendeteksi secara visual. Langkah pertama melibatkan pengumpulan dataset yang berisi gambar digital yang menggambarkan berbagai karakter aksara Lontara'. Dataset ini kemudian melalui proses penandaan manual, di mana setiap gambar diberi label berdasarkan huruf yang ditampilkan untuk menyiapkan data pelatihan yang tepat. Selanjutnya, data dibagi menjadi tiga subset secara teratur: 70% digunakan untuk pelatihan model, 20% untuk validasi yang berguna untuk memantau stabilitas model selama proses pembelajaran, dan 20% Sisanya dimanfaatkan sebagai data uji untuk secara objektif mengevaluasi kinerja akhir model.

Hasil evaluasi kuantitatif terhadap performa model menunjukkan kinerja yang sangat optimal dan sesuai dengan harapan. Model deteksi objek yang dirancang menggunakan algoritma YOLOv8 dalam studi ini berhasil mencatatkan performa tinggi, dengan akurasi mencapai 98,2%, presisi sebesar 98,1%, dan recall sempurna sebesar 100%. Pencapaian metrik tersebut mencerminkan tingkat efisiensi dan reliabilitas sistem dalam mengenali serta mengklasifikasikan karakter aksara Lontara' pada berbagai kondisi visual, sekaligus memperkuat kredibilitas model untuk diimplementasikan dalam skenario dunia nyata.

Sebagai saran untuk penelitian selanjutnya, disarankan untuk memperluas variasi data dengan melibatkan lebih banyak partisipan dan sumber gambar untuk meningkatkan keberagaman dataset, sehingga model dapat lebih baik dalam menggeneralisasi terhadap data baru. Selain itu, mencari algoritma yang lebih modern atau menggunakan metode campuran dengan menggabungkan berbagai teknik pembelajaran mendalam bisa jadi cara yang bagus untuk memperbaiki kinerja dan daya tahan sistem dalam menghadapi situasi operasional yang lebih kompleks.

Kata Kunci: Pengenalan Aksara Lontara', YOLOv8, Deep Learning, Literasi.

ABSTRACT

The rapid advancement of technology, particularly in the field of computing, has played a significant role in supporting innovations in automatic character recognition systems, including those designed for the Lontara script. One of the primary objectives of this research is to design an optimal implementation of the YOLOv8 object detection algorithm to accurately and reliably recognize and classify Lontara characters. Therefore, the main goal of this study is to evaluate the effectiveness of the YOLOv8 algorithm in supporting the image recognition and automatic classification process of characters in the Lontara writing system.

This study was conducted at Universitas Muhammadiyah Makassar using an experimental method that relied on machine learning techniques as the foundation for building a visual detection system. The initial phase involved collecting a dataset consisting of digital images representing various Lontara characters. These images underwent a manual annotation process, where each image was labeled based on the character it depicted, to prepare appropriate training data. The dataset was then systematically divided into three subsets: 70% for training the model, 20% for validation to monitor model stability during the learning phase, and the remaining 20% for testing, which served as the basis for objectively evaluating the model's final performance.

Quantitative evaluation results revealed that the model achieved highly satisfactory and optimal performance. The object detection model developed using the YOLOv8 algorithm demonstrated strong results, recording an accuracy of 98.2%, a precision of 98.1%, and a perfect recall score of 100%. These performance metrics indicate the system's efficiency and reliability in recognizing and classifying Lontara characters under varying visual conditions, thereby strengthening the model's credibility for implementation in real-world scenarios.

For future research, it is recommended to increase dataset diversity by involving more participants and image sources to enrich the variability of the training data, thereby enhancing the model's generalization capabilities. Moreover, exploring more advanced algorithms or integrating hybrid approaches through the combination of various deep learning techniques may offer promising strategies to further improve the system's performance and resilience in handling more complex operational environments.

Keywords: Lontara Script Recognition, YOLOv8, Deep Learning, Literacy.

KATA PENGANTAR

Segala puji bagi Allah SWT Rabbul Alamin yang selalu melimpahkan berbagai nikmat-Nya, termasuk kesehatan, akal yang jernih, kelapangan hati, dan yang paling utama adalah nikmat hidayah yang diberikan kepada Penulis. Shalawat serta salam semoga senantiasa tercurah kepada Nabi Muhammad SAW, seorang utusan Tuhan yang telah mengangkat harkat martabat kemanusian, membimbing manusia ke kodratnya yang asali, menjernihkan pikiran dan hati mansuai dari perangkap kejahiliaan.

Atas berkat pertolongan Allah SWT, Penulis dapat menyelesaikan skripsi yang berjudul **“Pendeteksian Objek Akara Lontara Untuk Literasi”** tepat waktu. Dengan penuh rasa syukur, penulis menyampaikan terima kasih yang sebesar-besarnya kepada semua pihak yang telah memberikan bantuan serta dukungan dalam penyusunan Laporan Tugas Akhir ini, khususnya:

1. **Bapak Dr. Ir. H. Abd. Rakhim Nanda, S.T., M.T.,IPU.** selaku Rektor Universitas Muhammadiyah Makassar.
2. **Bapak Ir. Muhammad Syafa'at S.kuba. ST., M.T.** sebagai Dekan Fakultas Teknik Universitas Muhammadiyah Makassar
3. **Bapak Muhyiddin A.M Hayat, S.Kom., M.T.**, selaku Ketua Program Studi Informatika, Fakultas Teknik, Universitas Muhammadiyah Makassar.
4. **Bapak Fachrim Irhamna Rachman, S.Kom., M.T.,** selaku Dosen Pembimbing I, serta **Ibu Titin Wahyuni, S.Pd, M.T.,** selaku Dosen

Pembimbing II, yang telah membimbing, memberikan arahan, serta motivasi selama penyusunan laporan ini.

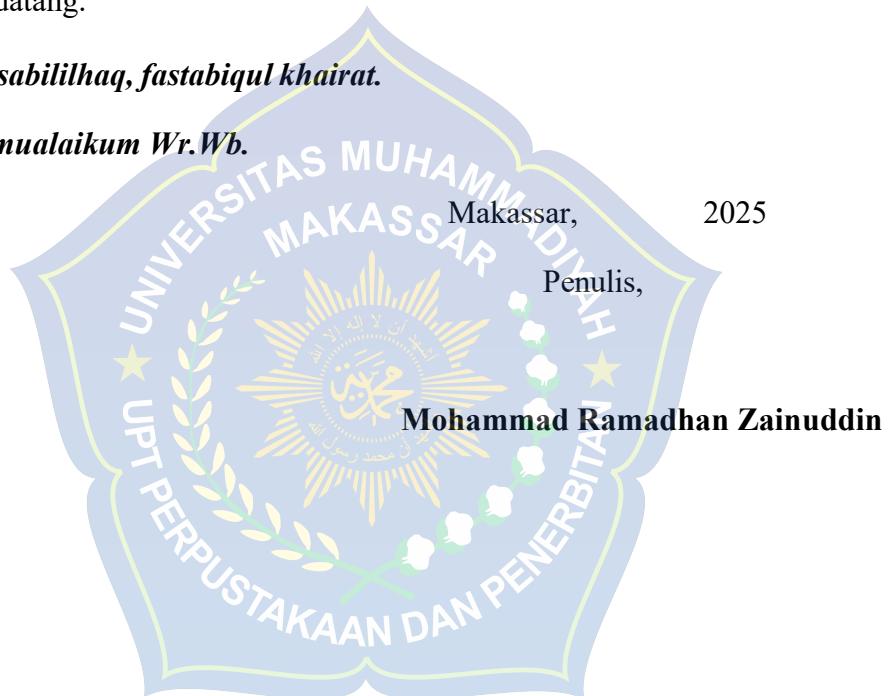
5. **Seluruh Dosen dan Staf Fakultas Teknik Universitas Muhammadiyah Makassar**, yang telah memberikan ilmu dan pengalaman berharga selama masa perkuliahan.
6. **Kedua orang tua Tercinta**, yang senantiasa merawat dengan penuh kasih sayang serta memberikan dukungan tanpa henti, baik secara material maupun nonmaterial. Penulis mengucapkan terima kasih yang sebesar-besarnya atas doa-doa yang tiada putus, yang selalu memberikan ketenangan dan semangat dalam menjalankan penelitian ini, bahkan di saat-saat tersulit sekalipun. Tanpa cinta dan dukungan mereka, penulis mungkin tidak akan mampu melewati setiap tantangan yang ada. Segala usaha yang dilakukan dalam penelitian ini sepenuh hati dipersembahkan untuk diri sendiri dan kedua orang tua sebagai bentuk penghormatan dan ungkapan terima kasih yang mendalam. Mereka adalah pilar kekuatan dalam setiap langkah yang diambil, dan dedikasi ini merupakan wujud kecil dari besarnya peran mereka dalam kehidupan penulis.
7. **Rekan-rekan mahasiswa utamanya angkatan 2020 Fakultas Teknik Universitas Muhammadiyah Makassar**, terima kasih atas dukungan dan kerjasamanya selama menempuh Pendidikan serta penyelesaian penyusunan proposal skripsi ini.
8. **Teman-Teman Grup Neoma SocialSpace**, atas supportnya Penulis mengucapkan terima kasih banyak atas bantuan dan dukungannya baik secara moral maupun materi

9. **Terakhir**, terima kasih untuk diri saya sendiri karena sudah mampu berusaha keras mampu melewati rintangan yang ada karna sudah berjuang sampai sejauh ini

Penulis menyadari bahwa Laporan Tugas Akhir ini masih memiliki beberapa kekurangan. Oleh karena itu, dengan penuh kerendahan hati, penulis sangat mengharapkan kritik dan saran yang membangun guna penyempurnaan laporan ini di masa mendatang.

Billahi fisabililhaq, fastabiqul khairat.

Wassalamualaikum Wr.Wb.



MOTTO

“Allah memang tidak menjanjikan hidupmu mudah, tapi dua kali Allah berjanji bahwa: Fa inna ma’al usri yusro innama’al usri yusro”

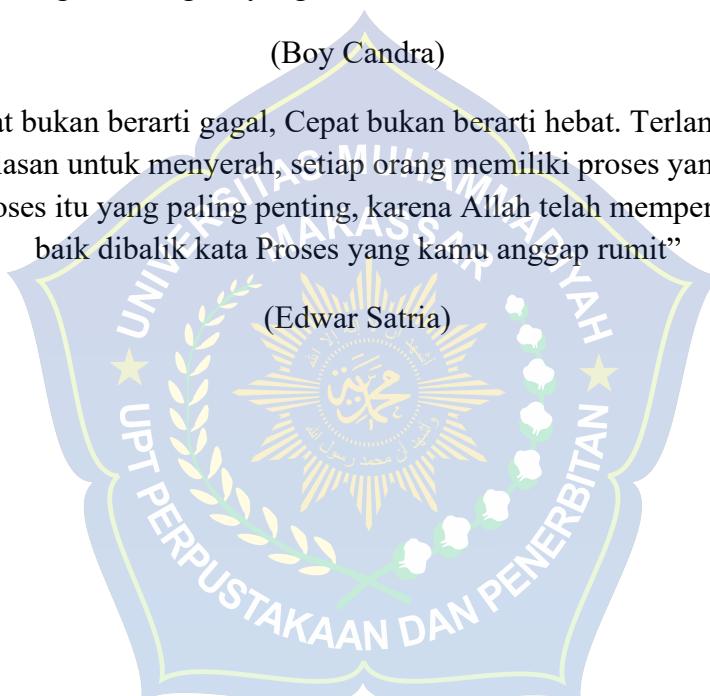
(QS. Al-Insyirah 94:5-6)

“Selalu ada harga dalam sebuah proses. Nikmati saja lelah-lelah itu, lebarkan lagi rasa sabarnya. Semua yang kau investasikan untuk menjadikan dirimu serupa yang kau impikan, mungkin tidak akan selalu berjalan lancar. Tapi, gelombang-gelombang itu yang nanti bisa kamu ceritakan”

(Boy Candra)

“Terlambat bukan berarti gagal, Cepat bukan berarti hebat. Terlambat bukan menjadi alasan untuk menyerah, setiap orang memiliki proses yang berbeda. Percaya proses itu yang paling penting, karena Allah telah mempersiapkan hal baik dibalik kata Proses yang kamu anggap rumit”

(Edwar Satria)



DAFTAR ISTILAH

- Machine Learning** : Machine learning merupakan cabang dari kecerdasan buatan yang memungkinkan sistem komputer untuk meningkatkan kinerjanya secara otomatis melalui pemrosesan dan analisis data, tanpa harus bergantung pada instruksi atau pemrograman yang bersifat eksplisit. Dalam prosesnya, algoritma pembelajaran mesin mengidentifikasi pola-pola tersembunyi dari kumpulan data dan memanfaatkannya untuk menghasilkan prediksi atau mengambil keputusan secara mandiri.
- Flowchart** :Sebuah representasi visual yang menggambarkan tahapan-tahapan atau urutan kegiatan dalam suatu proses, biasanya disusun menggunakan berbagai simbol grafis seperti persegi panjang, panah, dan ikon lainnya untuk menunjukkan hubungan logis antar langkah-langkah tersebut.
- Roboflow** :Sebuah ekosistem terpadu yang dirancang untuk menyederhanakan proses pengembangan dan manajemen model *computer vision*, dengan menyediakan serangkaian fitur komprehensif seperti alat anotasi data, teknik augmentasi citra, serta modul pelatihan model secara efisien dan terstruktur.
- Deep Learning** :Sebuah bagian dari pembelajaran mesin yang memanfaatkan jaringan syaraf buatan dengan beberapa lapisan (jaringan syaraf dalam) untuk menggambarkan dan mengolah data yang rumit, seperti foto dan suara.
- YOLO** :Sebuah algoritma cerdas yang dibuat untuk melaksanakan deteksi objek dengan cepat dan tepat dalam waktu nyata, mampu mengenali dan mengklasifikasikan berbagai objek dalam satu tahap pemrosesan yang terintegrasi.
- Object Detection** : Sebuah teknologi dalam bidang komputasi visual yang berfungsi untuk mendekripsi serta mengenali keberadaan objek-objek tertentu di dalam citra digital maupun rekaman video, dengan tujuan menginterpretasikan informasi visual secara otomatis.
- Preprocessing** : Tahapan awal dalam proses machine learning yang berfokus pada pengolahan dan pembersihan data mentah guna

memastikan bahwa informasi yang digunakan memenuhi standar kualitas serta memiliki format yang konsisten. Langkah ini melibatkan berbagai teknik seperti penanganan data yang tidak lengkap, normalisasi skala nilai, serta transformasi atribut agar siap untuk diolah oleh algoritma pembelajaran mesin.

Computer Vision : Merupakan salah satu disiplin dalam kecerdasan buatan yang mengembangkan kemampuan komputer untuk memahami konten visual seperti gambar dan video secara otomatis. Teknologi ini memungkinkan sistem untuk menafsirkan data visual, mengenali elemen-elemen penting seperti objek, wajah, dan teks (melalui Optical Character Recognition), serta melacak pergerakan. Aplikasinya meluas di berbagai sektor, termasuk sistem pengawasan keamanan, kendaraan otonom, dan diagnosa medis berbasis citra.

Validasi

: Merupakan proses pengujian yang dilakukan untuk menilai sejauh mana sebuah model mampu memberikan hasil prediksi yang tepat terhadap data yang bersifat baru atau tidak dikenal—yakni data yang tidak digunakan selama fase pelatihan. Evaluasi ini bertujuan untuk memverifikasi kemampuan generalisasi model dalam menghadapi beragam kondisi nyata di luar data latih yang telah dipelajari sebelumnya.

DAFTAR GAMBAR

Gambar 1 Arsitektur YOLO	10
Gambar 2 Flowchart	12
Gambar 3 Kerangka Berfikir	16
Gambar 4 Perancangan Sistem	18
Gambar 5 Pengambilan Dataset	24
Gambar 6 upload Dataset	25
Gambar 7 Pembuatan Kelas Dataset	26
Gambar 8 Proses Pelabelan	27
Gambar 9 Pengolahan Gambar (Resize)	28
Gambar 10 Split Dataset Sebelum Augmentasi	28
Gambar 11 Split Dataset Setelah Augmentasi	29
Gambar 12 Split Dataset kedua Sebelum Augmentasi	28
Gambar 13 Split Dataset kedua Setelah Augmentasi	29
Gambar 14 Split Dataset ketiga Sebelum Augmentasi	28
Gambar 15 Split Dataset ketiga Setelah Augmentasi	29
Gambar 16 Proses Pembuatan API	29
Gambar 17 Hasil Training	32
Gambar 18 Model Dataset	32
Gambar 19 Hasil Validasi	33
Gambar 20 Testing Model	34

BAB I

PENDAHULUAN

A. Latar Belakang

Lontara Bugis ialah aksara daerah yang digunakan secara luas dalam kehidupan sosial masyarakat Sulawesi Selatan. Sebagian besar orang di Sulawesi Selatan menggunakan Bahasa Bugis Lontara beberapa abad yang lalu, jauh sebelum berdirinya kerajaan-kerajaan besar seperti Bone, Luwu, dan Gowa. Bahasa Bugis Lontara memiliki ejaan yang unik dibandingkan dengan bahasa daerah lainnya. Mayoritas orang Bugis menggunakan bahasa Bugis, terutama mereka yang tinggal di daerah di Sulawesi Selatan yang didiami asli suku Bugis. Namun, paling mengejutkan adanya fakta bahwa sebagian besar orang asli Bugis tidak tahu sama sekali bahasa ibunya tersebut terutama yang tinggal di luar daerah, dan kalangan remaja yang tidak mempelajari bahasa Bugis secara lansung dari orang tua mereka.

Penelitian ini mengangkat masalah di atas untuk memastikan bahwa bahasa lontara Bugis, yang merupakan bagian dari keberagaman dan kekayaan warisan budaya Indonesia, tidak terkikis oleh zaman atau terlupakan oleh generasi muda, dan betapa pentingnya untuk melindunginya dari ancaman kepunahan. Minimnya materi edukasi bagi pengguna aksara Bugis Lontara juga menjadi permasalahan terkait menurunnya penggunaan aksara tersebut. Bagi individu yang berinteraksi dengan aksara Bugis Lontara, berarti satu-satunya cara untuk memahami aksara lontarak bugis dengan baik di masa sekarang adalah menggunakan bantuan

teknologi. Kecerdasan Buatan merupakan cabang ilmu yang membahas tentang bagaimana kemampuan suatu sistem agar data eksternal secara benar dapat diartikan, dipelajari, serta menggunakan pembelajarannya melalui tugas yang diadaptasi secara fleksibel sehingga tujuannya dapat tercapai. Menurut Hidayat dan Rahmatullah (2023:12) kecerdasan buatan adalah kemampuan sistem komputer untuk memproses informasi, belajar dari pengalaman, dan menyesuaikan perilaku guna mencapai tujuan tertentu.

Salah satu pendekatan terbaru dalam pendeketnsian objek adalah algoritma YOLO. Algoritma ini dianggap mampu memenuhi kebutuhan deteksi dengan akurasi tinggi serta kecepatan yang sangat cepat, memungkinkan operasi real-time secara langsung. YOLO merupakan salah satu metode *deep learning* untuk deteksi objek yang terkenal karena kecepatan dan akurasinya. Sebagaimana dijelaskan oleh Suryawan dan Nugroho (2022:45), YOLO memproses gambar secara menyeluruh dalam satu kali inferensi sehingga mampu mendekripsi objek secara real-time dengan tingkat akurasi tinggi. Pendekatan ini tidak menggunakan tahap *region proposal* seperti metode R-CNN, melainkan langsung memprediksi bounding box dan kelas objek pada satu langkah proses.

YOLO merupakan metode deteksi objek yang mengandalkan satu jaringan saraf tunggal (single neural network) untuk bekerja secara terpadu. Dengan pendekatan ini, sistem dapat sekaligus memprediksi posisi kotak pembatas (bounding box) dan kemungkinan kelas objek di seluruh gambar hanya dalam sekali pemrosesan. Keunggulannya terletak pada kecepatannya: versi standar YOLO mampu memproses hingga 45 frame per detik (FPS), sementara versi ringannya,

Fast YOLO, bahkan bisa mencapai 155 FPS. Angka ini membuat YOLO menjadi salah satu algoritma deteksi objek real-time tercepat yang ada saat ini. Sebagai perbandingan, metode lain seperti Fast R-CNN dan Faster R-CNN hanya mencapai kecepatan sekitar 0,5 FPS dan 7 FPS. Rahasia kecepatannya terletak pada teknik *single shot detection*, di mana CNN dijalankan hanya satu kali untuk seluruh proses deteksi. Berbeda dengan R-CNN dan turunannya yang harus mengulang proses CNN berkali-kali untuk setiap wilayah kandidat (region proposal), YOLO menyelesaikan semuanya dalam satu tembakan.

Studi yang dilakukan oleh Pratama dkk, (2023:6) membandingkan YOLO dan SSD dalam deteksi objek, dan *menemukan bahwa* YOLO menunjukkan akurasi sebesar 99,5% dengan precision 99,7%, sedangkan SSD hanya mencapai akurasi 98,89%. Penelitian lain oleh Wibowo dan Sari (2024:4) pada implementasi YOLOv8 untuk deteksi botol dan kaleng daur ulang menunjukkan mAP, precision, dan recall masing-masing berada pada kisaran 99,5–99,7%, membuktikan performa luar biasa dalam kondisi real-time.

Hingga saat ini, belum terdapat publikasi dalam jurnal ilmiah berbahasa Indonesia yang secara khusus menerapkan algoritma YOLO pada pendekripsi aksara Lontara. Namun, penelitian oleh Arhinza dkk. (2023) menunjukkan pendekatan alternatif yang efektif menggunakan kombinasi Histogram of Oriented Gradients (HOG) dan K-Nearest Neighbors (K-NN), dengan performa akurasi mencapai 85,25 %. Temuan ini membuka peluang eksplorasi lebih lanjut, seperti penerapan YOLO yang terkenal dalam deteksi objek real-time, untuk meningkatkan akurasi dan kecepatan pendekripsi aksara Lontara.

Dalam penelitian ini, fokus utama adalah penerapan algoritma YOLO untuk mendeteksi aksara Lontara sebagai bagian dari upaya literasi. Oleh karena itu, penelitian ini diberi judul “Pendeteksian Aksara Lontara untuk Literasi”.

B. Rumusan Masalah

Adapun permasalahan yang hendak dijawab dalam penelitian ini dirumuskan sebagai berikut: “Bagaimana Model Pendekstnsian Aksara Lontara Untuk Kegiatan Literasi dengan menggunakan Algoritma Yolo?”

C. Target Penelitian

Merancang sistem pendekstnsian aksara Lontara dengan memanfaatkan algoritma YOLO, sehingga mampu mengenali dan mengklasifikasikan setiap karakter secara cepat serta tepat.

D. Manfaat Penelitian

Terdapat beberapa manfaat yang diharapkan pada kajian ini diantaranya :

- a. Bagi penulis, penelitian ini akan meningkatkan kemampuan penulis dalam bidang pemrograman dan pengolahan bahasa alami. Penulis juga memperoleh wawasan yang lebih mendalam mengenai model yang diterapkan dalam proses pendekstnsian aksara Lontara dengan memanfaatkan algoritma YOLO.
- b. Bagi Universitas, Hasil penelitian ini dapat meningkatkan reputasi Universitas sebagai Lembaga yang mendukung inovasi dan pelestarian Bahasa Bugis dan juga dapat berperan sebagai acuan bagi penelitian berikutnya yang berfokus pada bidang teknologi informasi dan bahasa.

- c. Bagi pembaca, Pengaplikasian ini akan membantu dalam pelestarian Bahasa Bugis dengan menyediakan alat yang memudahkan pembelajaran dan penggunaan bahasa.

D. Ruang Lingkup Penelitian

Agar permasalahan pokok tidak terlalu luas, maka penulis membatasi cakupan permasalahan pada model pendekripsi aksara lontara untuk literasi dengan menggunakan algoritma Yolo.

F. Sistematika Penulisan

Laporan akhir ini terbagi menjadi beberapa sistematika bab yang runut diantaranya:

BAB I PENDAHULUAN Bagian yang memberikan gambaran mengenai fenomena permasalahan, rumusan masalah, ruang lingkup, target, dan kelebihan, serta gaya penulisan.

BAB II TINJAUAN PUSTAKA Teori-teori pendukung proses penulisan yang digunakan untuk menyelesaikan proposal dijelaskan pada bagian ini.

BAB III METODE PENELITIAN Bagian memuat materi dan teknik penelitian yang diterapkan dalam pengembangan sistem.

BAB II

TINJAUAN PUSTAKA

A. Landasan Teori

1. Model AI (*Artificial Intelligence*)

Kecanggihan teknologi komputer dirancang untuk melakukan pekerjaan-pekerjaan yang biasanya dilakukan kecerdasan manusia diantaranya biasa digunakan pada pemrosesan bahasa alami atau NLP dan pengenalan objek *picture*, serta suara ataupun ucapan, bahkan juga pada pengambilan keputusan adalah adalah suatu bagian dari sistem yang dikembangkan oleh AI atau *Artificial Intelligence*. AI bisa dimanfaatkan pada setiap tingkatan dalam pembuatan aplikasi. AI dimanfaatkan hampir pada semua aplikasi yang saat ini banyak digunakan, diantaranya untuk memahami apa yang diinginkan oleh pengguna, diantaranya dimanfaatkan dalam mengirim pemberitahuan atau mengisyaratkan suatu isi konten (Manandhar, 2023).

2. *You Only Look Once* (YOLO)

YOLO (*You Only Look Once*) merupakan teknik pembelajaran mendalam yang memproses masukan visual menggunakan teknik *Convolutional Neural Network* (CNN) untuk pengenalan objek secara real-time. Untuk mendeteksi objek, kerangka YOLO pertama-tama membagi gambar masukan ke dalam kotak pembatas menggunakan kotak-kotak SxS. Berdasarkan tingkat kepercayaan dalam mengidentifikasi objek tertentu, sistem kemudian memperkirakan probabilitas kelas (Husnan, dkk, 2023:222).

Sedangkan Maleh, dkk (2023:20) mengemukakan bahwa model terintegrasi yang digunakan oleh YOLO, teknik deteksi yang memungkinkan kotak pembatas dan probabilitas kelas diprediksi secara bersamaan oleh jaringan saraf tunggal dari gambar penuh dalam satu lintasan.

3. *Roboflow*

Alat berbasis web untuk manajemen kumpulan data disebut *Roboflow*. Ini memfasilitasi metode yang lebih baik untuk pengumpulan data, prapemrosesan, dan pelatihan model dengan bertindak sebagai kerangka kerja bagi para programer komputer. *Roboflow* memungkinkan pengguna memanipulasi kumpulan data, termasuk memberi label atau memberi anotasi pada objek untuk dideteksi dengan kotak pembatas, dan membagikannya dengan orang lain. Selain itu, *Roboflow* memungkinkan aktivitas persiapan seperti augmentasi data dan konversi gambar skala abu-abu. Peneliti yang ingin mengumpulkan koleksi foto kendaraan berkualitas tinggi untuk tujuan pendekstrian objek menghadapi kendala karena kumpulan data diperlukan selama tahap akuisisi data untuk menguji pendekstrian objek dengan benar (Hayati, dkk, 2023:91).

4. *Pytorch*

Menurut Hendri, dkk (2021:3) Salah satu paket yang terkenal untuk perhitungan deep learning dalam bahasa pemrograman Python adalah *Pytorch*. Ini menekankan pada kemampuan beradaptasi dengan mengaktifkan sintaksis Python untuk digunakan untuk mengekspresikan model pembelajaran mendalam. Pengadopsi awal dalam komunitas riset telah menerapkan strategi ini dan kemudahan penggunaannya. Perpustakaan telah berkembang menjadi

salah satu sumber daya terpenting untuk aplikasi pembelajaran mendalam sejak pertama kali tersedia. Tensor, array multidimensi yang mirip dengan yang ada di perpustakaan, NumPy (*Numerical Python*) adalah struktur data mendasar yang disediakan oleh PyTorch. Berdasarkan hal ini, sejumlah fitur telah dibuat untuk mempermudah pembuatan dan pelaksanaan proyek serta merencanakan dan melatih eksperimen menggunakan arsitektur jaringan saraf baru. Jika teknologi dan perangkat lunak yang digunakan modern, tensor dapat mempercepat komputasi matematis. PyTorch memiliki perpustakaan tambahan untuk fungsi pembelajaran mendalam secara umum, proses pekerja yang menggunakan banyak sumber daya di satu komputer untuk pemuatan data yang efektif, dan alat *Convolutional Neural Network* (CNN), yang menggunakan neuron kortikal untuk analisis gambar, adalah teknik yang sering digunakan. Karena Torchvision mendukung beberapa desain CNN, seperti *AlexNet*, *VGG*, *ResNet*, *SqueezeNet*, *DenseNet*, *Inception v3*, *GoogLeNet*, *ShuffleNet v2*, *MobileNet v2*, *ResNext*, *Wide ResNet*, dan *MNASNet*, ini sangat berguna untuk pemodelan menggunakan prinsip CNN untuk pelatihan terdistribusi. PyTorch meningkatkan kemampuan komputasi pembelajaran mendalam pada objek digital, seperti foto (2D) dan film (3D) saat digunakan dengan paket torchvision.

5. Aksara Lontara

Naskah kuno Bugis dapat dibagi menjadi dua kategori, yakni yang dibuat dengan aksara Lontarak dan yang ditulis dengan aksara Serang. Fakta bahwa peristiwa-peristiwa tersebut pertama kali didokumentasikan dalam lembaran-lembaran yang terbuat dari daun lontar, itulah asal kata “tulisan lontar”. Dalam

bahasa Makasar (berasal dari bahasa Jawa atau Melayu), “daun lontar” setara dengan “raung” yang berarti daun, dan “talak” yang berarti lontar. Gabungan kedua kata ini menghasilkan “rauttalak” atau “rontalak”. Setelah mengalami metatesis, istilah “rontalak” menjadi “lontarak”. Karena bahasa Arab mempengaruhi tulisan Bugis akibat masuknya Islam ke Sulawesi Selatan pada awal abad ke-17, maka aksara tersebut kini dikenal dengan aksara Arab (Pelras dalam Hamka,2019:4). Selanjutnya menurut Manyambeang dalam Hamka (2019:4) kata “lontarak” bisa diartikan sebagai “aksara, naskah, bacaan, atau catatan”. Lontara dapat memiliki berbagai arti tergantung pada konteks kalimatnya.

Menurut Kem dalam Hamka (2019:5) Aksara lontara bersumber dan aksara Sanskrit yang dikenal sebagai Dewanagari. Sedangkan Manyambeang dalam Hamka (2019:8) Aksara Bugis Makassar diduga diadaptasi dari aksara Dewanagari karya Daeng Pamatte, seorang tokoh terkenal di pelabuhan kerajaan Gowa. Basang dalam Hamka (2019:8) mendukung hal tersebut mengatakan bahwa baik aksara Bugis Lontara maupun aksara Dewanagari sama-sama menggunakan huruf vokal bantu /i, e, o, dan u/ serta mempunyai suku kata huruf. Mereka ditulis dari kiri ke kanan. Selanjutnya Menurut Mattulada dalam Hamka (2019:8-9) mengemukakan bahwa ada kepercayaan dikalangan orang Makassar-Bugis tentang pembentukan aksara lontarak yang kemudian dikenal sebagai aksara lontarak. Dunia digambarkan dalam mitologi Makassar-Bugis sebagai "bolasuji" (Bugis) atau "sulapak appak" (Makassar), yang berarti "berlian dengan empat sisi". Karena huruf S berarti kesatuan, maka

alam merupakan satu kesatuan yang utuh. "Sulapa eppana tau", yang diterjemahkan menjadi "empat sisi tubuh manusia", adalah mikrokosmos yang dilambangkan dengan huruf "s". Kaki berada di bawah, sedangkan kepala, tangan kiri, dan tangan kanan berada di atas. Simbol S terdapat pada daerah kepala disebut juga *sawwang* (SW) yang artinya mulut. "Sadda" (sd) yang artinya suara, merupakan representasi ekspresi mulut. Bunyi-bunyi tersebut digabungkan menjadi kata "ada" (ad), yang berarti perkataan, ketetapan, atau perintah.

Masyarakat Bugis menggunakan sistem penulisan yang disebut aksara Lontara: Tabel di bawah ini menunjukkan 23 karakter berbeda yang membentuk aksara Lontara:

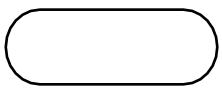
KA	GA	NGA	NGKA
PA	BA	MA	MPA
TA	DA	NA	NRA
CA	JA	NYA	NCA
YA	RA	LA	WA
SA	A	HA	

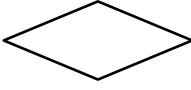
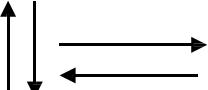
Gambar1. Aksara Lontara Bugis

6. Flowchart

Menurut Rosaly (2019), flowchart merupakan bentuk visual yang merepresentasikan runtutan proses atau langkah-langkah logis dalam suatu sistem, yang bertujuan untuk menggambarkan alur kerja dari algoritma secara sistematis. Analis sistem memanfaatkan flowchart sebagai dokumen acuan untuk mengilustrasikan secara logis struktur sistem yang akan dikembangkan kepada para programmer. Dengan demikian, flowchart berperan dalam mengidentifikasi dan menyelesaikan potensi masalah yang dapat muncul selama proses pengembangan sistem. Secara umum, flowchart disusun dari beragam simbol grafis yang masing-masing memiliki fungsi spesifik dalam merepresentasikan jenis aktivitas atau tahapan tertentu dalam suatu proses. Simbol-simbol tersebut saling terhubung melalui garis pengarah yang berfungsi untuk memperlihatkan urutan atau jalannya proses dari satu tahap ke tahap berikutnya dalam suatu sistem kerja.

Tabel 1 Flowchart

Simbol	Nama	Fungsi
	Terminal	Simbol terminal memiliki peran penting dalam diagram alir karena digunakan untuk menyatakan dimulainya dan berakhirnya suatu rangkaian aktivitas dalam sistem. Simbol ini menjadi titik referensi awal ketika proses mulai dijalankan, serta menjadi indikator bahwa seluruh langkah atau prosedur yang telah dirancang telah mencapai tahap akhir..

	Input/Output	Digunakan untuk menunjukkan aktivitas menerima data (input) atau menampilkan hasil (output), tanpa memperhatikan jenis data yang terlibat.
	Manual Operation	Merepresentasikan suatu tahapan dalam alur kerja yang pelaksanaannya dilakukan secara langsung oleh manusia, tanpa keterlibatan otomatisasi dari sistem komputer.
	Decision	Simbol ini memiliki peran untuk mengarahkan jalannya proses berdasarkan hasil dari suatu kondisi atau pernyataan logis. Dengan kata lain, simbol ini memungkinkan terjadinya percabangan alur, di mana langkah selanjutnya ditentukan oleh evaluasi terhadap syarat tertentu yang harus dipenuhi
	Processing	Menandakan kegiatan pengolahan data yang dilakukan oleh sistem komputer sebagai bagian dari eksekusi program..
	Disk Storage	Menunjukkan bahwa data yang digunakan atau dihasilkan berasal dari atau disimpan di media penyimpanan seperti disk.
	Flow Direction Symbol/Connecting Line	Flow Direction Symbol atau garis penghubung berfungsi sebagai elemen pengait antar simbol dalam sebuah flowchart, yang secara visual memperlihatkan urutan pelaksanaan serta arah perpindahan proses dari satu tahap ke tahap berikutnya dalam sistem yang digambarkan

7. Deteksi Objek

Dan deteksi objek merupakan salah satu pendekatan dalam ranah computer vision yang memiliki tujuan utama untuk mengidentifikasi serta menentukan posisi objek-objek tertentu yang terdapat dalam suatu citra atau tayangan video. Teknik

ini umumnya mengandalkan algoritma berbasis machine learning maupun deep learning guna memperoleh tingkat akurasi dan efisiensi yang tinggi. Manusia secara inheren memiliki kapasitas bawaan untuk mengenali dan memahami keberadaan objek dalam lingkungan visual secara instan dan intuitif, tanpa menuntut keterlibatan kognitif yang kompleks. Di sisi lain, mesin komputasi tidak secara alami memiliki kapasitas perceptual, sehingga diperlukan pendekatan bertingkat yang melibatkan proses-proses algoritmik yang tersusun secara sistematis dan berlapis guna membangun ulang fungsi pengenalan visual secara buatan..

Secara teknis, proses deteksi objek melibatkan dua tahapan utama, yaitu klasifikasi objek yang bertujuan untuk mengidentifikasi jenis atau kategori objek dalam citra dan pelokalan objek melalui penerapan bounding box yang mengelilingi area keberadaan objek tersebut. Dengan demikian, selain mengetahui apa objek yang ditampilkan, sistem juga dapat menentukan dengan tepat letaknya dalam bingkai visual. Meskipun konsep ini memiliki keterkaitan dengan klasifikasi gambar, perbedaan fundamental terletak pada ruang lingkupnya: klasifikasi gambar terbatas pada pemberian label tunggal terhadap keseluruhan citra, sedangkan deteksi objek mencakup kemampuan untuk mengenali berbagai objek sekaligus serta memetakan posisi masing-masing objek secara spesifik.

Dalam implementasinya, sistem deteksi objek beroperasi dengan memanfaatkan citra sebagai data input, yang selanjutnya dievaluasi melalui sejumlah proses komputasi bertingkat guna mengidentifikasi serta menentukan posisi objek-objek yang terkandung di dalamnya. Proses ini tidak hanya mencakup pengenalan masing-masing objek, tetapi juga pengelompokan berdasarkan kategori

atau kelas tertentu sesuai dengan karakteristik visual yang terdeteksi. Entitas yang kerap menjadi pusat perhatian dalam proses deteksi ini mencakup unsur-unsur seperti individu, sarana transportasi, serta struktur bangunan. SKemajusat dalam teknologi kecerdasan buatan dan pengolahan citra telah mendorong peningkatan signifikan dalam kemampuan sistem deteksi objek, yang kini telah diadopsi secara luas di berbagai sektor aplikasi (Imantiyar & Fudholi, 2021).

B. Penelitian Terkait

Penelitian yang dilakukan Jeremia Zhophie dan Hendri Himawan Triharmint (2022) dengan judul “Implementasi Algoritma You Only Look Once (YOLO) menggunakan Web Camera untuk Mendeteksi Objek Statis dan Dinamis”. Tujuan utama dari penelitian ini adalah untuk mengevaluasi sejauh mana algoritma You Only Look Once (YOLO) menunjukkan konsistensi dan akurasi kinerjanya melalui dua skema pengujian yang dirancang secara kontras dalam hal operasionalisasinya. Eksperimen awal diimplementasikan dalam konteks operasional real-time, di mana sistem memperoleh data visual secara langsung melalui integrasi dengan perangkat kamera webcam yang berfungsi sebagai sumber input dinamis. Aliran data visual tersebut kemudian diproses secara simultan, memungkinkan sistem untuk melakukan deteksi objek secara langsung sekaligus memberikan respons terhadap perubahan visual secara instan sesuai dengan dinamika lingkungan sekitar. Sementara itu, eksperimen kedua dilaksanakan dalam konteks non-real-time, yang hanya menggunakan data visual statis yang telah dikumpulkan sebelumnya. Dalam pendekatan ini, sistem tidak berinteraksi secara langsung dengan sumber citra waktunya, melainkan mengandalkan koleksi data visual yang telah dikompilasi

sebelumnya. Kumpulan data yang digunakan dalam pendekatan ini terdiri atas dua jenis format media digital yang berbeda karakteristiknya, yaitu citra statis dengan ekstensi JPG dan rekaman video bergerak dalam format MP4. Kedua tipe data ini diperlakukan sebagai unit analisis terpisah, di mana setiap format menjalani tahapan pemrosesan yang disesuaikan dengan sifat visual masing-masing. Proses deteksi objek dilakukan secara independen pada setiap format, sehingga memungkinkan sistem untuk mengevaluasi kinerja algoritma berdasarkan konteks media yang berbeda—baik pada citra yang bersifat tetap maupun pada rangkaian bingkai visual yang terus berubah dalam video.. Tahapan analisis dimulai dengan proses ekstraksi frame dari setiap gambar dan video, yang kemudian diikuti dengan implementasi algoritma YOLO untuk mengidentifikasi objek berdasarkan fitur visual tertentu. Penilaian kinerja sistem dilakukan dengan metode kuantitatif menggunakan matriks confusion, yang menghasilkan temuan bahwa algoritma ini mampu mencapai akurasi sebesar 69,57%, dengan tingkat kesalahan klasifikasi sebesar 26,09%, nilai presisi mencapai 94,12%, dan tingkat spesifisitas sebesar 50%.

Persamaan penelitian diatas dengan penelitian yang dilakukan oleh peneliti adalah sama-sama menggunakan algoritma Yolo sebagai deteksi objek. Adapun perbedaan penelitian ini dengan penelitian yang dilakukan oleh peneliti adalah peneliti menggunakan algoritma Yolo untuk mendeteksi aksara lontara dengan pengujian format citra JPEG. Sementara penelitian diatas adalah implementasi algoritma YOLO dengan pengujian menggunakan Webcam.

Penelitian yang dilakukan oleh I Made Dwijaya Male, dkk (2023) dengan judul “Implementasi Algoritma You Only Look Once (YOLO) Untuk Object

Detection Sarang Orang Utan". Algoritma You Only Look Once (YOLO) diadaptasi sebagai pendekatan strategis dalam sistem deteksi visual untuk mengidentifikasi keberadaan sarang orangutan di kawasan hutan tropis Kalimantan. Keunggulan utama algoritma ini terletak pada kemampuannya dalam melakukan prediksi koordinat bounding box serta estimasi probabilitas kelas objek secara simultan hanya dalam satu tahapan pemrosesan terhadap keseluruhan citra, sehingga menjadikannya sangat efisien untuk analisis visual berskala besar. Memanfaatkan 1.970 citra pelatihan model dilakukan dengan sebagai data pelatihan 414 citra tambahan yang telah dianotasi secara manual untuk menandai lokasi sarang orangutan di hutan Kalimantan. Evaluasi kinerja algoritma menunjukkan hasil yang sangat baik, dengan nilai precision mencapai 0,973, yang menunjukkan bahwa hampir seluruh prediksi positif sesuai dengan objek yang sebenarnya. Recall sebesar 0,949 hal terebut menegaskan sebagian besar sarang orang utan mampu diidentifikasi oleh sistem. Pengukuran performa berdasarkan mean Average Precision (mAP) juga memberikan hasil yang kuat, dengan skor mAP pada ambang Intersection over Union (IoU) 0,5 sebesar 0,969, serta mAP pada rentang IoU 0,5 hingga 0,95 mencapai 0,630, yang mencerminkan ketepatan deteksi pada berbagai tingkat sensitivitas spasial.

Selama proses pelatihan, model menempuh sebanyak 217 epoch dalam waktu total 58 jam, dan hasil akhir menunjukkan performa deteksi objek yang sangat tinggi. Secara agregat, model mampu mengidentifikasi jumlah sarang orangutan dengan akurasi rata-rata mendekati sempurna, yakni sebesar 99,9%.

Persamaan penelitian tersebut diatas dengan penelitian ini adalah sama-sama mengimplementasikan algoritma Yolo sebagai media deteksi objek. Adapun perbedaan penelitian diatas dengan penelitian ini adalah peneliti menjadikan aksara lontara sebagai objek deteksi semenatarena penelitian diatas menjadikan sarang orang utan sebagai objek deteksinya

Penelitian yang dilakukan oleh Nurhaliza Juliyani Hayat,dkk (2023) dengan judul “Object Tracking Menggunakan Algoritma You Only Look Once (Yolo)v8 Untuk Menghitung Kendaraan”. Penelitian ini dikembangkan untuk menguji model pelacakan objek (object tracking) menggunakan algoritma You Only Look Once (YOLO)v8 untuk mendeteksi objek berupa jenis serta menghitung jumlah kendaraan. Metodologi yang diterapkan mengikuti siklus proyek AI (AI Project Cycle) dengan tahapan problem scoping, data acquisition, data exploration, modeling, dan evaluasi menggunakan confusion matrix. Hasil evaluasi menunjukkan tingkat akurasi sebesar 89%, precision 89%, recall 90%, serta F1-Score sebesar 89% berdasarkan perbandingan bobot precision dan recall. Hasil pendekstian membuktikan bahwa algoritma YOLOv8 terbukti akurat dalam mendeteksi objek untuk pelacakan serta perhitungan jumlah kendaraan.

Persamaan penelitian diatas dengan penelitian ini adalah sama-sama mengimplementasikan algoritma Yolov8 untuk deteksi objek. Adapun perbedaan penelitian diatas dengan penelitian ini adalah penelitian diatas menggunakan algortima Yolov8 untuk mendeteksi jenis dan menghitung jumlah kendaraan sedangkan dalam penelitian ini peneliti mendeteksi aksara lontara untuk literasi.

Penelitian yang dilakukan oleh Aisyah Nurul Hidayah (2020) dengan judul “Deteksi Huruf Pada Tulisan Tangan Latin Menggunakan Metode You Only Look Once (Yolo)”. Studi ini menggunakan algoritma You Only Look Once (YOLO) untuk pendekripsi karakter aksara tulisan tangan. Tujuan utama dari penerapan metode ini adalah untuk mengakselerasi proses pengenalan karakter secara otomatis, sehingga dapat meningkatkan efisiensi dalam sistem identifikasi teks tulisan tangan secara keseluruhan. Pengujian dilakukan melalui pengumpulan sampel gambar dari empat gaya tulisan tangan yang bervariasi guna memastikan generalisasi model terhadap perbedaan bentuk karakter. Proses deteksi dijalankan menggunakan perangkat keras GPU NVIDIA GeForce RTX 1050. Hasil evaluasi eksperimental menunjukkan bahwa sistem yang dikembangkan memiliki kemampuan untuk mengenali huruf-huruf dalam satuan kata dengan efisiensi waktu yang tinggi, yaitu rata-rata waktu pemrosesan sebesar 0,0776 detik untuk setiap citra yang dianalisis. Hal ini mencerminkan performa sistem yang responsif dalam menangani proses deteksi karakter secara real-time. Di samping itu, sistem menerapkan nilai ambang (threshold) sebesar 0,3 pada parameter confidence score untuk memfilter hasil prediksi. Hasil Nilai tersebut merepresentasikan batas minimum tingkat keyakinan yang harus dicapai oleh model agar suatu objek dikenali sebagai huruf yang valid. Dengan kata lain, hanya prediksi dengan tingkat probabilitas di atas 30% yang dianggap cukup meyakinkan untuk diterima sebagai hasil deteksi yang sah oleh sistem.

Persamaan penelitian diatas dengan penelitian ini adalah sama-sama mengimplementasikan algoritma YOLO untuk pendekripsi aksara. Adapun

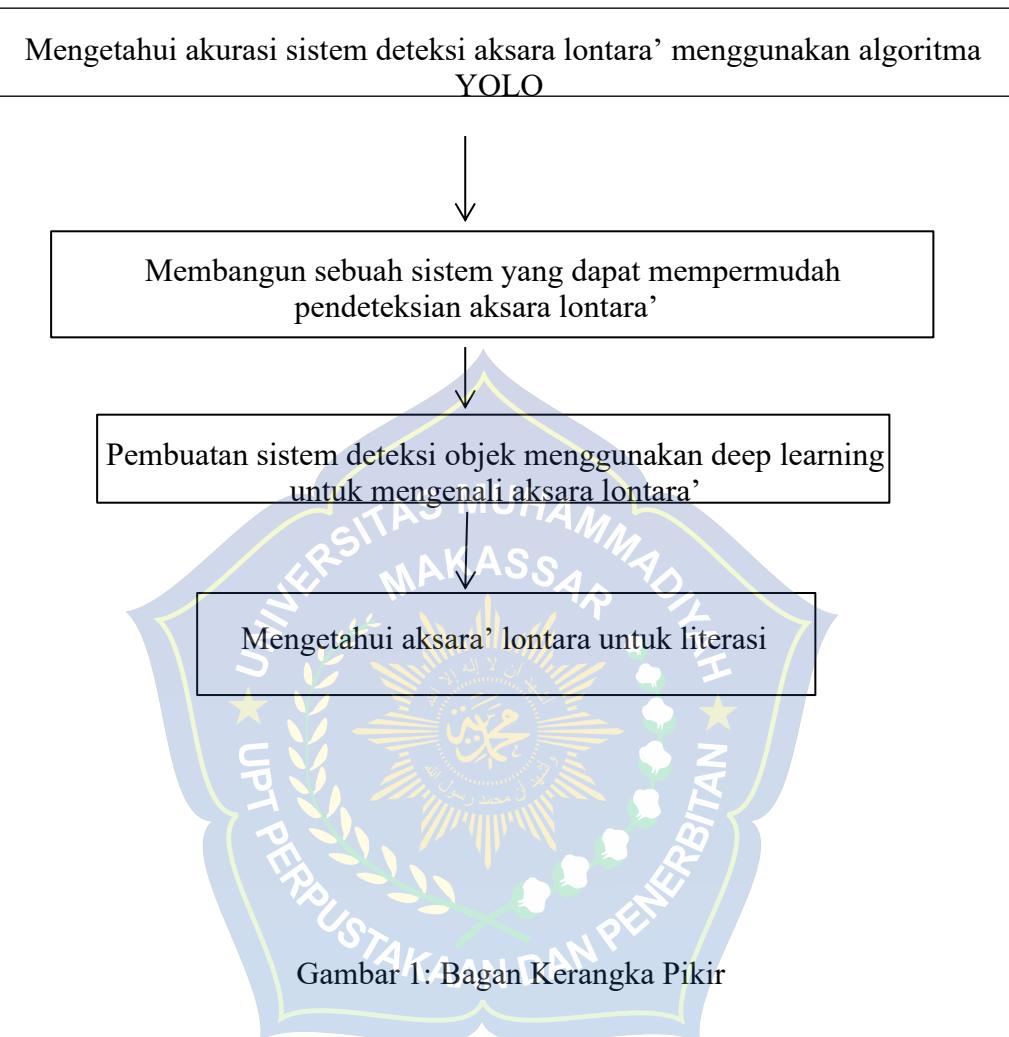
perbedaan dengan penelitian ini dengan penelitian diatas adalah penelitian diatas menggunakan pengujian dengan algoritma Yolo untuk mendeteksi tulisan tangan sementara penelitian ini menggunakan algoritma Yolo untuk mendeteksi aksara lontara.

Penelitian Muhammad Arif Faizin (2023) dengan judul “Deteksi Aksara Jawa Menggunakan YOLO untuk Transliterasi Berbasis LSTM pada Manuskrip Jawa Kuno.” Penelitian ini merancang dan membentuk sebuah kumpulan data khusus yang terdiri atas manuskrip aksara Jawa, diambil dari naskah kuno Serat Sewaka. Dataset berupa 60 citra manuskrip yang telah dianotasi karakter menggunakan cara manual, kemudian selanjutnya diklasifikasikan dan diberi nama Handwritten Javanese Character on Sewaka Manuscripts Detection (HJCS_DETC), sebagai representasi awal data terstruktur dalam pengenalan aksara Jawa tulisan tangan. Dalam pelatihan sistem deteksi, algoritma YOLOv5 dimanfaatkan untuk mengolah dataset HJCS_DETC_SPLIT, dengan tujuan menghasilkan bounding box yang menunjukkan posisi masing-masing karakter dalam citra manuskrip. Informasi lokasi tersebut menjadi fondasi penting dalam proses segmentasi aksara sebagai langkah awal transliterasi. Tahapan transliterasi mencakup serangkaian proses lanjutan, termasuk identifikasi garis tengah (midline detection) untuk menentukan referensi vertikal karakter, pengurutan huruf berdasarkan distribusi spasial, pengenalan pola suku kata, pemisahan bentuk pasangan atau ligatur, serta integrasi hasil deteksi dengan data transliterasi sebagai acuan untuk interpretasi fonetik atau ejaan Latin. Untuk menangani urutan karakter secara temporal dan menjaga keterkaitan antarhuruf dalam konteks aksara, diterapkan arsitektur Long

Short-Term Memory (LSTM). Model ini berfungsi dalam menyempurnakan proses transformasi dari bentuk visual aksara Jawa menjadi representasi fonetik yang lebih akurat dan konsisten secara berkelanjutan. Hasil evaluasi terhadap performa model deteksi objek menunjukkan bahwa pada skenario kesembilan—yang memanfaatkan dataset hasil augmentasi serta skema pembagian data pelatihan dan validasi dalam rasio 5:1 model mampu mencapai F1-score tertinggi adalah 83,2%, kemudian mean Average Precision (mAP) 81,4%. Dataset berupa 67 kelas dataset aksara Jawa tradisional. Di sisi lain, pada tahapan transliterasi, digunakan model Bidirectional LSTM (BiLSTM) yang dioptimalkan dengan algoritma Adam serta pengaturan learning rate sebesar 0,001. Setelah proses Pengujian, model menghasilkan transliterasi Character Error Rate (CER) dengan nilai 16,7% dan Word Error Rate (WER) sebesar 23,3%, hal ini membuktikan performa dianggap cukup andal dalam mengkonversi karakter aksara manuskrip menjadi bentuk fonetik aksara latin secara otomatis. Capaian tersebut mengindikasikan bahwa sistem memiliki tingkat kesalahan yang masih dalam batas toleransi, sehingga cukup kompeten untuk diterapkan pada pemrosesan aksara dari naskah manuskrip kuno secara otomatis dan efisien.

Persamaan penelitian diatas dengan penelitian ini adalah sama-sama menggunakan algoritma Yolo sebagai pendekripsi aksara. Adapun perbedaan penelitian diatas dengan penelitian ini adalah penelitian diatas mendekripsi aksara Jawa menggunakan algoritma Yolov5, sedangkan penelitian ini mendekripsi aksara lontara Bugis menggunakan algoritma Yolov8.

C. Kerangka Pikir



Gambar 1: Bagan Kerangka Pikir

BAB III

METODE PENELITIAN

A. Lokasi dan Priode Penelitian

1. Tempat Penelitian

Penelitian ini dilakukan dengan cara memotret aksara Lontara menggunakan kamera pada telepon genggam.

2. Waktu Penelitian

Kegiatan penelitian direncanakan berlangsung pada bulan Januari hingga Februari 2025.

B. Alat dan Bahan

Terdapat dua kategori utama alat dan bahan yang digunakan dalam penelitian ini, yaitu:

1. Perangkat Keras (Hardware)

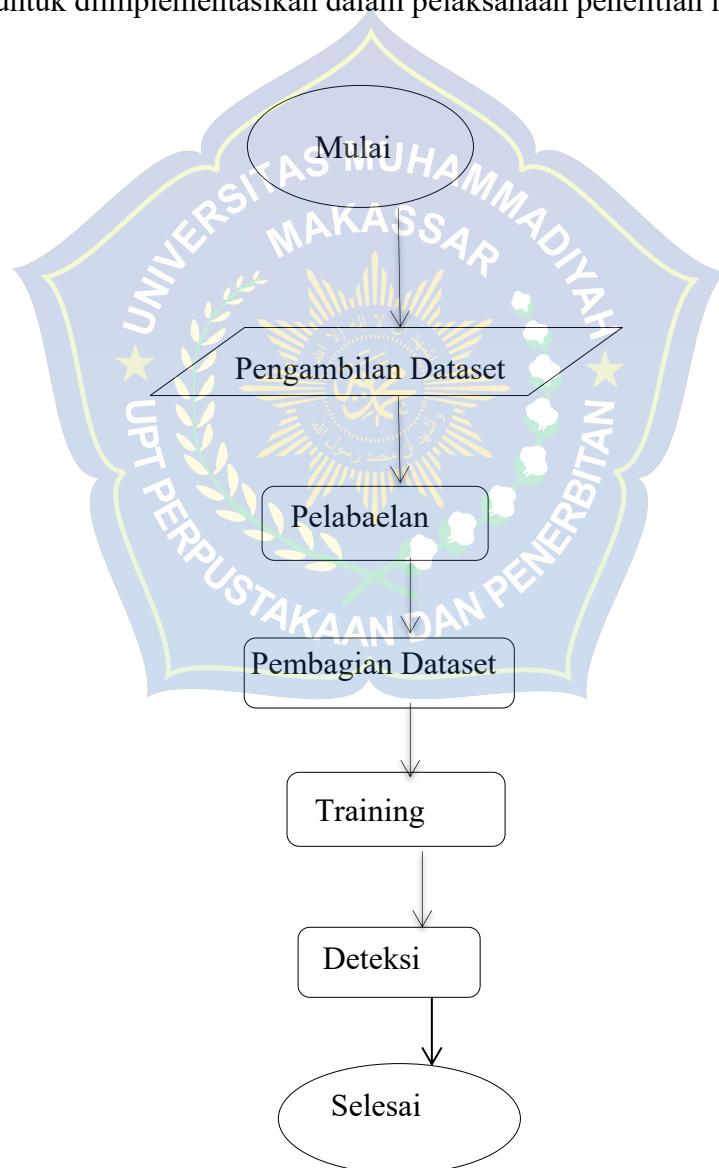
- a. Laptop merek Lenovo
- b. Ponsel Realme C11

2. Perangkat Lunak (Software)

- a. Python
- b. Visual Studio Code
- c. Roboflow
- d. Google Drive
- e. Google Colab

C. Perancangan Sistem

Diagram alur (flowchart) dimanfaatkan sebagai representasi visual dari tahapan-tahapan yang harus ditempuh dalam menyelesaikan proses pada program penelitian ini. Setiap tahap dalam proses tersebut digambarkan secara sistematis menggunakan simbol-simbol tertentu yang dihubungkan oleh panah sebagai penunjuk alur. Ilustrasi di bawah ini menampilkan desain sistem yang telah dirumuskan untuk diimplementasikan dalam pelaksanaan penelitian ini.



1. Pengambilan Dataset

Pada tahap awal ini, jenis data yang dimanfaatkan dalam penelitian berupa gambar digital dari aksara Lontara. Tahapan pengumpulan data dilakukan melalui eksplorasi berbagai platform digital, mencakup laman situs web, kumpulan repositori gambar, serta basis data terbuka yang tersedia secara online. Tujuannya adalah untuk menghimpun koleksi citra aksara yang mampu merepresentasikan bentuk visual huruf secara nyata, dengan variasi bentuk dan gaya yang cukup luas, sehingga mampu mencerminkan keragaman karakter aksara yang akan dianalisis.. Pendekatan ini bertujuan untuk memastikan bahwa dataset yang diperoleh memiliki tingkat relevansi tinggi terhadap objek yang dikaji serta memenuhi standar kualitas visual yang memadai untuk keperluan pelatihan sistem.. Dataset yang telah dikumpulkan tersebut selanjutnya dimanfaatkan sebagai bahan pelatihan bagi sistem deteksi yang tengah dikembangkan.

2. Pengolahan Gambar

Dalam tahap pra-pemrosesan (preprocessing) citra, dilakukan penyesuaian ukuran gambar melalui proses resize guna menyeragamkan dimensi setiap citra yang digunakan. Proses ini memegang peranan penting dalam menjamin keseragaman dimensi pada seluruh citra yang digunakan, sehingga setiap gambar memiliki ukuran yang seragam. Keseragaman tersebut diperlukan untuk menyesuaikan data dengan format input yang telah ditetapkan oleh arsitektur model yang akan dilatih, serta menghindari ketidaksesuaian bentuk data yang dapat mengganggu proses pembelajaran model secara optimal.Selain

memastikan keseragaman ukuran gambar, proses resize juga memberikan dampak signifikan terhadap efisiensi pemanfaatan memori dan kinerja sistem secara keseluruhan. Dengan menyesuaikan dimensi citra ke ukuran yang lebih optimal, kebutuhan ruang penyimpanan data menjadi lebih ringan, sehingga mampu mengurangi beban kerja memori saat data diproses. Dampak dari proses ini juga terlihat secara langsung dalam peningkatan performa sistem, khususnya dalam hal kecepatan pemrosesan dan efisiensi komputasi..

3. Pelabelan Dataset

Pada tahap ini, data yang telah diproses diberi label dengan memanfaatkan aplikasi Roboflow. Penggunaan Roboflow bertujuan untuk menandai atau mengidentifikasi setiap objek pada gambar, sehingga sistem deteksi objek yang dikembangkan dapat mengenali masing-masing elemen secara akurat.

4. Proses Membagi Dataset

Dataset yang telah diberi label kemudian dipisahkan menjadi tiga kelompok, yaitu data pelatihan (training), data validasi, dan data pengujian (testing). Pada tahap ini, model dilatih menggunakan data pelatihan yang tersedia. Sementara itu, data validasi digunakan untuk menilai kinerja model dengan cara menguji dan mengidentifikasi kemampuan model tersebut. Proses ini bertujuan agar model dapat dievaluasi tingkat akurasinya terhadap data baru yang belum pernah dikenali sebelumnya.

5. Training

Pada langkah training, dataset dipisahkan dari sebelumnya menunjukkan data pelatihan (training) dan data pengujian (testing) mulai dimuat kemudian diolah untuk keperluan evaluasi performa model. Informasi pelabelan (labeling) yang telah disimpan sebelumnya diakses melalui Application Programming Interface (API) yang terintegrasi dengan lingkungan kerja Google Colab. Setelah data berhasil dimuat, proses evaluasi terhadap performa model dilakukan dengan menerapkan algoritma YOLOv8, di mana parameter utama yang digunakan untuk mengukur keberhasilan model adalah tingkat akurasi. Jika hasil evaluasi menunjukkan bahwa akurasi yang diperoleh masih berada di bawah standar yang diharapkan, maka akan dilakukan proses pelatihan ulang (retraining). Tahapan ini bertujuan untuk menyempurnakan kemampuan model dalam mengenali pola dan karakteristik data, sehingga dapat meningkatkan ketepatan prediksi pada proses selanjutnya.

6. Deteksi

Pada tahap ini, seluruh citra yang telah dikumpulkan sebelumnya mulai dimasukkan ke dalam sistem untuk menjalani proses pemrosesan lanjutan. Setelah gambar berhasil diunggah, sistem secara otomatis mengaktifkan proses pendekripsi objek dengan memanfaatkan algoritma YOLOv8. Algoritma YOLO dibuat untuk mengenali berbagai jenis objek yang ada di dalam gambar, kemudian akan diberi tanda berupa kotak pembatas dan label sesuai dengan kategori objek yang sudah ditemukan.. Melalui proses ini, model mampu melakukan pemisahan secara akurat antara objek utama dan latar belakang.

Evaluasi performa model ditampilkan dalam bentuk nilai akurasi, yang merepresentasikan tingkat keberhasilan model dalam mengenali objek secara tepat, sekaligus menentukan area deteksi yang relevan melalui konsep Region of Interest (ROI) atau Wilayah Minat.

D. Teknik Pengujian Sistem

Pemanfaatan Algoritma YOLO v8 berfungsi sebagai cara pengujian nyata di dalam studi ini.. Untuk Mendeteksi aksara lontara' untuk literasi dengan menggunakan metode YOLOv8, adalah merupakan suatu proses yang sangat penting untuk memastikan keakuratan sistem yang akan dikembangkan. Pengujian sistem ini dengan menggunakan beberapa tahapan dengan menggunakan teknik yang telah direncanakan dengan mennguji tahapan dari kinerja sistem, meliputi Keakuratan, presisi, dan kemampuan sistem mengingat kembali (recall) saat mendeteksi aksara Lontara'. Berikut ini beberapa langkah pengujian empiris yang umumnya dilakukan:

1. Persiapan Data

Proses awal penelitian dimulai dengan menghimpun dataset yang berisi beragam bentuk dan variasi aksara Lontara. Setelah terkumpul, data tersebut kemudian diorganisasikan ke dalam tiga kelompok utama, yakni data pelatihan, data validasi, dan data pengujian. Setiap citra dalam dataset dilengkapi dengan penanda atau label yang secara spesifik mengidentifikasi jenis aksara Lontara yang terdapat di dalamnya, sehingga memudahkan sistem dalam mengenali dan mempelajari karakter masing-masing aksara secara tepat.

2. Pembuatan Model

Tujuan dari machine learning ini adalah model deteksi yang disebut YOLOv8, yang kemudian akan diunduh dan dipilih berdasarkan kebutuhan penelitian. Disesuaikan dengan perubahan dalam struktur model, contohnya dengan mengubah indikator atau menambah lapisan baru, sebagai upaya meningkatkan performa model dalam mengidentifikasi objek aksara Lontara sesuai kebutuhan penelitian.

3. Pelatihan Model

Pada tahap ini, model untuk mendeteksi objek dilatih menggunakan bagian dari data pelatihan dari kumpulan data. Selama tahap ini, model akan melakukan beberapa putaran untuk secara bertahap memahami pola dan karakteristik yang terkait dengan objek aksara lontara..

4. Validasi Model

Setelah model selesai dibangun, tahap berikutnya adalah melakukan evaluasi untuk menilai tingkat efektivitasnya dengan memanfaatkan data validasi dari keseluruhan dataset. Evaluasi ini bertujuan memastikan bahwa kemampuan model tidak hanya terbatas pada pembelajaran dari data pelatihan, tetapi juga mampu mengaplikasikan pengetahuan tersebut pada data baru yang belum pernah ditemuinya. Pada proses validasi, kinerja model diukur menggunakan tingkat akurasi dalam mendeteksi objek, serta dilengkapi dengan perhitungan metrik lain seperti presisi, recall, dan F1-score guna memberikan gambaran yang lebih menyeluruh terhadap performa deteksi.

BAB IV

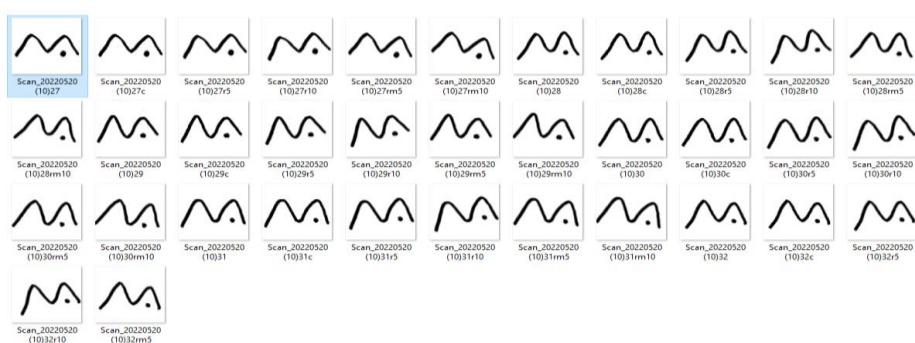
HASIL PENELITIAN DAN PEMBAHASAN

Bab IV ini menyajikan secara rinci hasil penelitian serta analisis terkait akurasi deteksi objek aksara Lontara dengan literasi Latin menggunakan algoritma YOLOv8. Tujuan sentral dari penelitian ini terletak pada proses pengenalan dan pemetaan objek-objek yang merepresentasikan aksara Lontara. Secara keseluruhan, dataset yang digunakan terdiri dari 805 gambar yang telah dikumpulkan, dengan sumber utama berasal dari internet.

A. Pembuatan Model

1. Pengambilan Data Set

Sekumpulan data ini diperoleh dengan mengunduh gambar dari web, lalu menyimpannya di memori ponsel. Selanjutnya, gambar-gambar ini dipindahkan ke sebuah direktori di komputer. Gambar-gambar tersebut meliputi karakter dari aksara lontara Bugis. Adapun jumlah gambar yang dikumpulkan dalam data ini berjumlah 805 gambar aksara lontara. Contoh-contoh gambar di bawah ini adalah bagian dari kumpulan data yang diambil dari internet.



Gambar 3. Pengambilan Dataset

Tabel 1. Jumlah Data

Nama Kelas	Jumlah Data
A	35
Ba	35
Ca	35
Da	35
Ga	35
Ha	35
Ja	35
Ka	35
La	35
Ma	35
Mpa	35
Na	35
Nca	35
Nga	35
Nka	35
Nra	35
Nya	35
Pa	35
Ra	35
Sa	35
Ta	35
Wa	35
Ya	35
Total	805

Setelah semua informasi terkumpul, data tersebut akan dibagi menjadi tiga bagian, yaitu 70% untuk pelatihan, 15% untuk validasi, dan 15% untuk pengujian. Pembagian data ini dilakukan agar proses pengujian dapat memberikan hasil akurasi yang lebih baik.

2. Pelabelan Gambar

Tahap berikutnya setelah mengumpulkan data adalah memberikan label pada gambar-gambar. Pada fase ini, foto-foto tersebut diberi tanda agar sistem mampu mengenali objek yang terdapat di dalamnya. Proses penandaan ini dilakukan dengan bantuan aplikasi Roboflow, yang bertujuan untuk mengidentifikasi label pada gambar-gambar aksara lontara, sesuai dengan kategori dan pengelompokan yang telah ditentukan sebelumnya.

a. Upload dataset

Gambar yang sudah dikumpulkan akan dimasukkan ke dalam Roboflow, dan langkah berikutnya adalah melakukan proses pemberian label.

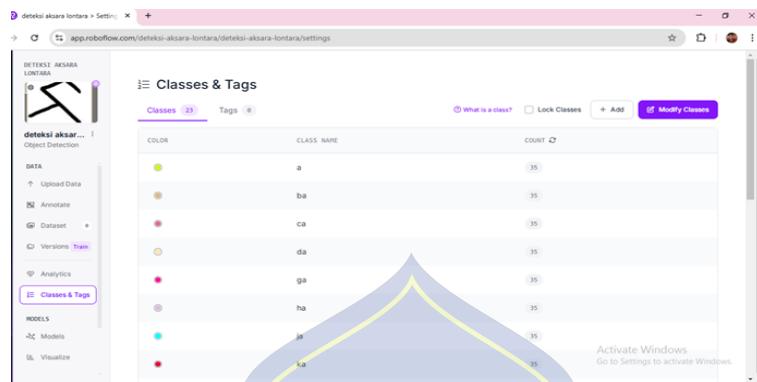


Gambar 4. Upload Data

b. Pembuatan Kelas Dataset

Tujuan dari pembuatan kelas dataset ini adalah untuk mempermudah pemilihan dataset gambar dengan memanfaatkan kotak batas.. Kelas ini akan

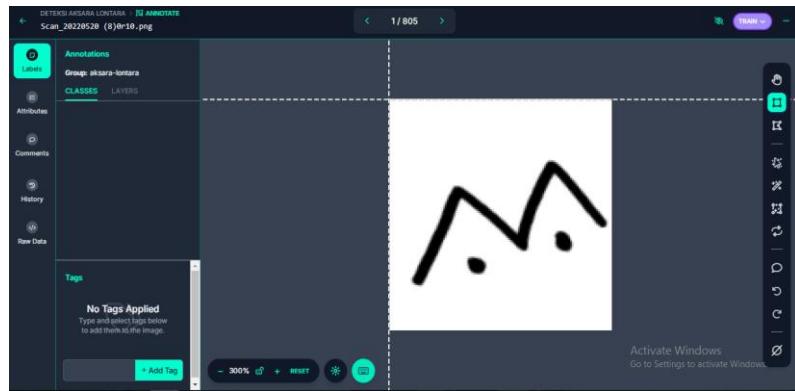
secara otomatis membuat 23 kategori yang sudah ditetapkan sebelumnya, dan dataset akan diorganisir berdasarkan kriteria tertentu.



Gambar 5. Pembuatan kelas dataset

c. Proses Pelabelan

Menggunakan Roboflow untuk memberi label pada gambar, yaitu dengan menempatkan bingkai di sekeliling objek dalam gambar atau kotak pembatas. Setelah kotak pemisah selesai dibuat, kategori pelajaran yang telah ditentukan sebelumnya akan muncul secara otomatis.

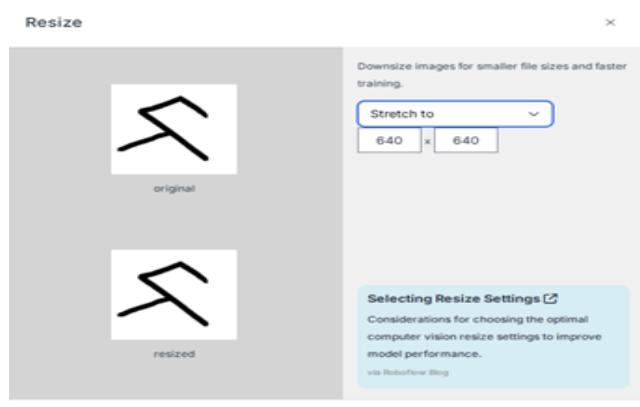


Gambar 6. Proses Pelabaelan

3. Pembagian Dataset

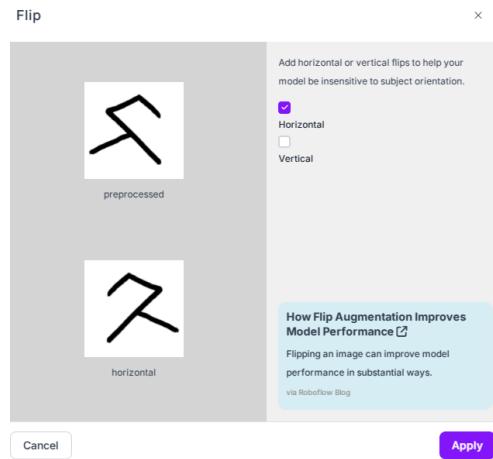
a. Pengolahan Gambar

Manajemen citra dengan Roboflow menawarkan beragam penyesuaian, termasuk modifikasi warna, bentuk, dan ukuran, serta penerapan teknik Preprocessing dan Augmentation. Ukuran citra di augmentasi 640 x 640 piksel. Tujuan dari langkah ini adalah untuk meningkatkan efektivitas kinerja GPU saat melakukan pengujian model, yang berpengaruh pada percepatan keseluruhan proses. Pendekatan yang diterapkan dalam fase ini meliputi pengubah ukuran, dengan ukuran gambar yang dituju adalah 640 x 640 piksel. Langkah ini dirancang untuk memperlancar kinerja GPU ketika melakukan pengujian model, sehingga meningkatkan efektivitas secara keseluruhan.



Gambar 7. Pengolahan Gambar (Rezise)

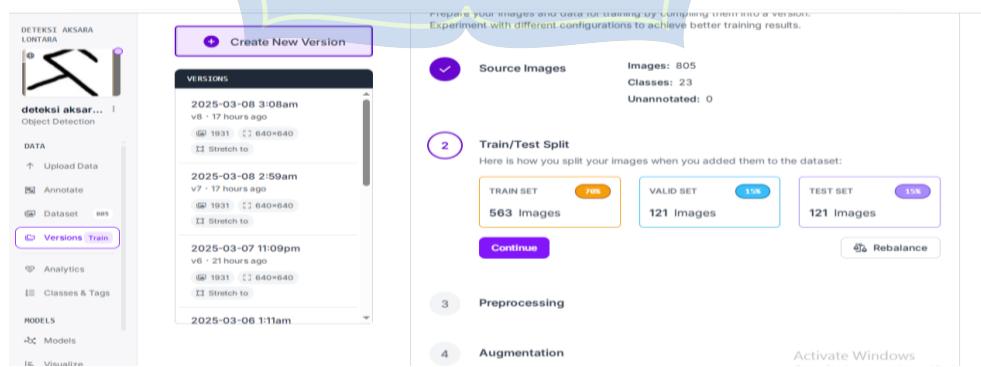
Tahapan peningkatan gambar merupakan bagian penting dari proses pengolahan data visual sebelum analisis, yang dilakukan dengan mengubah berbagai elemen di dalam sebuah gambar digital. Proses ini melibatkan pengeditan bentuk geometris, perubahan lokasi objek, dan penyesuaian fitur visual lain seperti ukuran, arah, atau warna dari gambar asli. Tujuan dari peningkatan ini adalah untuk menghasilkan variasi data yang lebih banyak dan beragam, sehingga model pembelajaran mesin dapat memahami ciri-ciri visual dari berbagai sudut pandang dan kondisi pengambilan gambar. Sehingga, perkembangan ini berpotensi secara signifikan memperbaiki kemampuan model dalam mengenali data yang sebelumnya tidak dikenal.. Dalam konteks peningkatan khususnya pada objek dengan batasan kotak, peneliti umumnya menerapkan metode transformasi seperti membalik atau memutar gambar, baik secara vertikal maupun horizontal. Teknik ini bertujuan untuk mencerminkan berbagai posisi dan orientation objek di dunia nyata, tanpa mengubah arti dari gambar itu sendiri, serta tetap menjaga ketepatan posisi kotak batas pada objek yang dimaksud.



Gambar 8. Pengolahan Gambar (Flip)

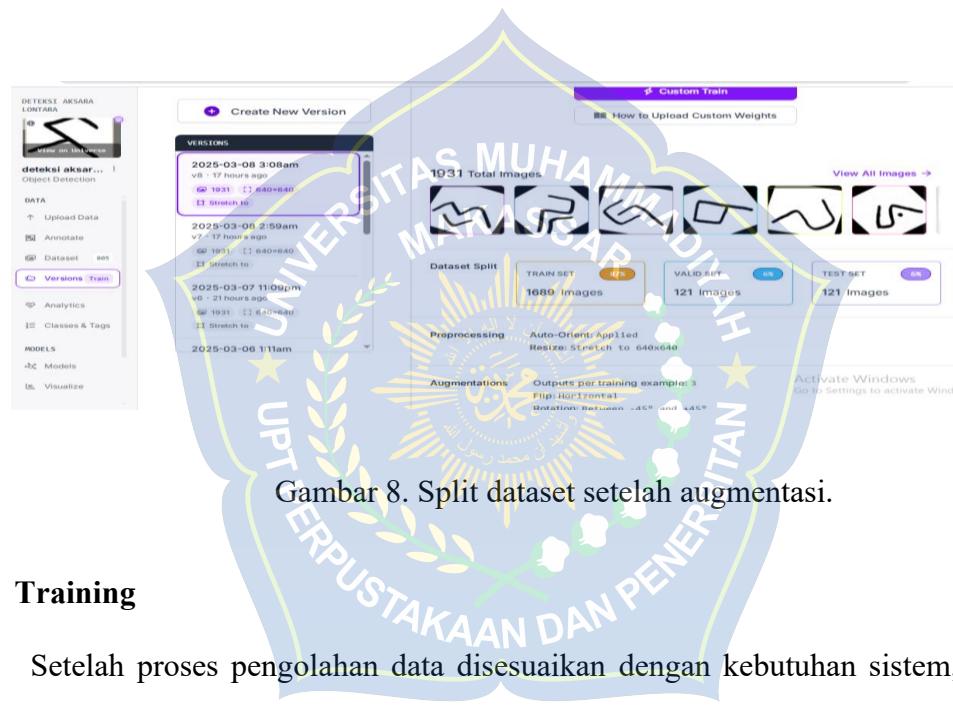
b. Pembagian Data

Pengujian augmentasi ini dilakukan dengan mengupload dataset melalui beberapa langkah, di mana alat Roboflow berfungsi untuk mengelompokkan dataset dan selanjutnya membaginya menjadi data untuk pelatihan, validasi, dan pengujian.. Sebelum tahap dilakukan augmentasi, dataset yang dikumpulkan terdiri dari 805 gambar dan dibagi menjadi 70% untuk data train (563 gambar), 15 % untuk data validasi (121 gambar), dan 15 % untuk data test (20 gambar).



Gambar 7. Split dataset sebelum augmentasi

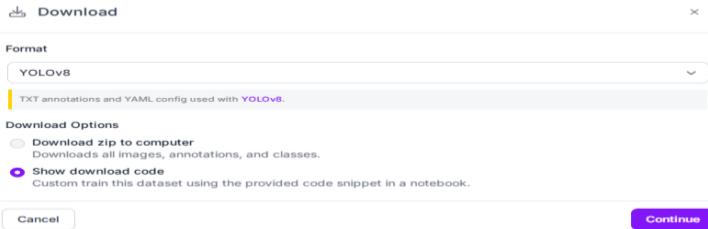
Setelah melewati proses augmentasi yang memanfaatkan bounding box dari data sebelumnya, dataset kemudian dimodifikasi lebih lanjut guna meningkatkan keragaman dalam pengujian. Pada tahap augmentasi ini, kumpulan gambar yang tersedia akan dibagi ke dalam tiga bagian, yakni 87% dialokasikan sebagai data pelatihan (sebanyak 1.689 gambar), 6% sebagai data validasi (121 gambar), dan 6% sisanya sebagai data pengujian (121 gambar).



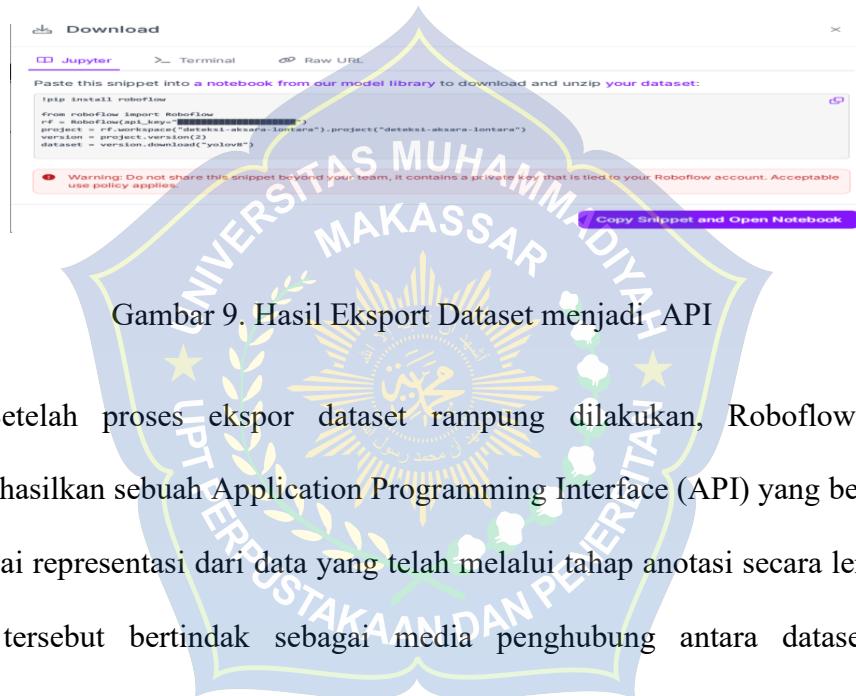
1. Training

Setelah proses pengolahan data disesuaikan dengan kebutuhan sistem, data tersebut kemudian diekspor untuk dimanfaatkan dalam tahap pelatihan model pada platform Google Colab. Hasil ekspor berupa dataset ini turut menyertakan API yang diperlukan sebagai antarmuka integrasi, sehingga dapat digunakan secara optimal dalam melatih model YOLOv8 di lingkungan Google Colab.

a. Eksport Dataset Roboflow



Gambar 9. Proses Pembuatan API



Gambar 9. Hasil Eksport Dataset menjadi API

Setelah proses ekspor dataset rampung dilakukan, Roboflow akan menghasilkan sebuah Application Programming Interface (API) yang berperan sebagai representasi dari data yang telah melalui tahap anotasi secara lengkap. API tersebut bertindak sebagai media penghubung antara dataset dan lingkungan pengembangan, memungkinkan proses pemanggilan data berlangsung secara otomatis dalam tahap pelatihan model YOLOv8 yang diimplementasikan di platform Google Colab.

b. Traing Google Colab

Proses pelatihan model dilakukan dengan memanfaatkan API yang diperoleh melalui platform Roboflow. API tersebut diakses sebagai sarana untuk menjalankan pelatihan model secara langsung di lingkungan Google Colab.

```
!pip install roboflow

from roboflow import Roboflow
rf = Roboflow(api_key="mcvtQQsHY0YtMXeM8q6x")
project = rf.workspace("deteksi-aksara-lontara").project("deteksi-aksara-lontara")
version = project.version(8)
dataset = version.download("yolov8")
```

Langkah pertama dalam program ini adalah membuat folder baru yang disebut "datasets" di dalam direktori utama pengguna (HOME). Folder ini berfungsi

```
%cd {HOME}

!yolo task=detect mode=train model=yolov8s.pt
data=/content/datasets/datafix-8/data.yaml epochs=50 imgsz=640
plots=True
```

sebagai tempat untuk menyimpan dataset yang akan diambil dari Roboflow.

Berikutnya, pustaka roboflow diinstal agar pengelolaan data menjadi lebih mudah.

Dengan menggunakan API key yang sudah disediakan, program membuat objek Roboflow yang digunakan untuk mengakses proyek bernama datafix dan mengunduh versi ke-8 dari dataset tersebut melalui metode download("yolov8").

Kumpulan data yang diperoleh sudah secara otomatis disertai dengan penandaan, sehingga bisa langsung dipakai untuk melatih model deteksi objek yang menggunakan YOLOv8. Langkah ini dirancang untuk mendukung pengembang dalam tahap awal menciptakan model untuk mendeteksi objek, khususnya di platform seperti Google Colab. Setelah akses API dari Roboflow berhasil, API tersebut digunakan sebagai jalan untuk mengintegrasikan proses pelatihan model

Pelaksanaan program dimulai dengan mengarahkan sistem ke direktori utama

pengguna (HOME). Setelah itu, model deteksi objek dijalankan menggunakan perintah YOLO, yang dikonfigurasi untuk melakukan tugas pendekripsi objek dalam mode pelatihan. Model awal yang digunakan adalah “yolov8s.pt”, sebuah model pralatih. Dataset yang dimanfaatkan, yaitu “datafix-8”, telah dipersiapkan lengkap dengan anotasi untuk keperluan pelatihan. Proses pelatihan dijalankan sebanyak 50 epoch dengan resolusi gambar sebesar 640 piksel. Di samping itu, pengaturan plots = True diaktifkan untuk membuat grafik yang menampilkan perkembangan hasil selama proses pelatihan.

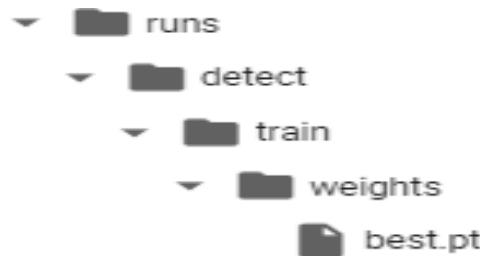


Gambar 10. Hasil Training

Setelah menyelesaikan proses pelatihan model pada platform Google Colab, data yang tersedia dibagi secara proporsional dan seimbang ke dalam tiga kelompok utama, yaitu 70% untuk proses pelatihan, 15% untuk proses validasi, dan 15% sisanya untuk tahap pengujian. Pembagian ini bertujuan untuk memastikan bahwa model memperoleh pelatihan yang memadai sekaligus dapat dievaluasi secara objektif melalui data yang tidak digunakan dalam pelatihan langsung. Berdasarkan

hasil pengujian, model menunjukkan performa yang sangat tinggi dengan capaian akurasi mencapai 99,5%, yang menandakan kemampuan model dalam mengenali pola dan objek secara tepat.

Setelah pelatihan rampung, sistem secara otomatis membuat file model terunggul yang diberi nama best. pt. File ini menyimpan parameter-parameter hasil pelatihan yang telah dioptimalkan, dan berfungsi sebagai representasi akhir dari model yang telah dilatih secara menyeluruh. File tersebut kemudian digunakan dalam tahap inferensi untuk menguji performa model terhadap gambar-gambar baru yang sebelumnya tidak pernah dikenali oleh model. Pengujian ini bertujuan utama untuk mengevaluasi seberapa baik model mampu menemukan dan mengategorikan objek dalam data yang sepenuhnya baru, sehingga dapat menilai seberapa jauh model dapat diterapkan pada data, sehingga dapat mengukur sejauh mana generalisasi model terhadap data di luar distribusi pelatihan awal.



Gambar 11. Model Dataset

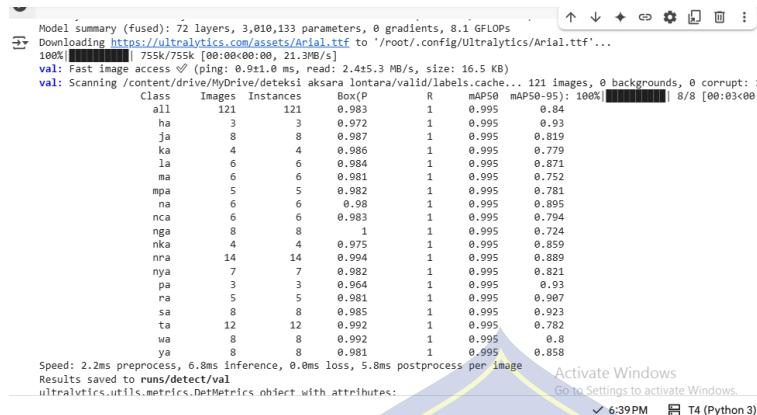
c. Validating

Setelah model selesai dilatih, langkah penting selanjutnya adalah mengevaluasi kinerja menggunakan data uji yang belum pernah dilihat oleh sistem sebelumnya. Tujuan evaluasi ini adalah untuk menilai seberapa baik model dapat merespons data baru yang berbeda dari data latihan, sehingga diketahui seberapa baik model dapat menggeneralisasi variasi data yang mungkin muncul di dunia nyata.

Pengujian ini adalah bagian penting dari proses pengembangan model, karena di tahap ini bisa dipastikan apakah model dapat tetap konsisten dan akurat ketika menghadapi situasi yang tidak sama dengan data pelatihan. Memanfaatkan informasi yang sepenuhnya baru memberi peluang untuk mengevaluasi sejauh mana pengetahuan dan pola yang dipelajari oleh model dapat diterapkan dalam situasi yang belum pernah dipelajari sebelumnya.

Informasi yang didapat dari evaluasi ini sangat penting secara strategis, karena menunjukkan seberapa baik model dapat beradaptasi dengan data yang berbeda dari data pelatihan. Jika kinerja model masih di bawah standar yang diharapkan, perlu dilakukan perbaikan, seperti meningkatkan kualitas dan representasi data, menyesuaikan parameter pelatihan, atau bahkan merancang

ulang struktur model untuk meningkatkan daya tahan dan akurasinya di lapangan.



Class	Images	Instances	Box(P)	R	mAP@0.5	mAP@0.95
all	121	121	0.983	1	0.995	0.84
ha	3	3	0.972	1	0.995	0.93
ja	8	8	0.987	1	0.995	0.819
ka	4	4	0.986	1	0.995	0.779
la	6	6	0.984	1	0.995	0.871
ma	6	6	0.981	1	0.995	0.752
mpa	5	5	0.982	1	0.995	0.781
na	6	6	0.98	1	0.995	0.895
nca	6	6	0.983	1	0.995	0.794
nga	8	8	1	1	0.995	0.724
nka	4	4	0.975	1	0.995	0.859
nra	14	14	0.994	1	0.995	0.889
nya	7	7	0.982	1	0.995	0.821
pa	3	3	0.964	1	0.995	0.93
ra	5	5	0.981	1	0.995	0.987
sa	8	8	0.985	1	0.995	0.923
ta	12	12	0.992	1	0.995	0.782
wa	8	8	0.992	1	0.995	0.8
ya	8	8	0.981	1	0.995	0.858

Gambar 12. Hasil Validasi

Hasil evaluasi model YOLOv8 pada dataset validasi yang terdiri dari 121 gambar menunjukkan performa deteksi yang sangat baik. Model berhasil mencapai nilai *precision* sebesar 0.984, yang berarti 98,4% dari deteksi yang dilakukan adalah benar (minim false positive). Selain itu, nilai *recall* mencapai 1.000, menandakan bahwa model berhasil mendekripsi seluruh objek yang ada pada gambar validasi tanpa terlewatkannya (tanpa false negative).

Tingkat akurasi deteksi secara keseluruhan juga ditunjukkan melalui nilai mAP (mean Average Precision). Pada metrik mAP@0.5 yang mengukur ketepatan deteksi pada ambang kesamaan $\text{IoU} \geq 50\%$ model berhasil memperoleh nilai 0.995, sangat dekat dengan sempurna. Sedangkan mAP@0.5:0.95, yang lebih ketat karena mengukur akurasi pada berbagai tingkat ambang IoU (dari 0.5 hingga 0.95), menunjukkan nilai 0.840. Ini membuktikan bahwa model tidak hanya akurat pada deteksi dasar, tetapi juga cukup andal dalam memahami batas objek secara presisi tinggi.

Dari sisi efisiensi, kecepatan pemrosesan model juga impresif. Rata-rata waktu inferensi yang dibutuhkan untuk memproses satu gambar adalah 7,6 milidetik, menandakan bahwa model sangat cocok untuk digunakan dalam aplikasi deteksi real-time atau waktu nyata. Ini mencerminkan kinerja yang cepat dan responsif, tanpa mengorbankan akurasi.

Secara keseluruhan, kombinasi antara akurasi tinggi dan kecepatan deteksi yang optimal menunjukkan bahwa model YOLOv8 yang telah dilatih sangat andal dan siap digunakan dalam skenario aplikasi nyata, seperti pengenalan aksara, pemantauan visual, atau sistem otomatis berbasis penglihatan komputer.

B. Pengujian Sistem

Salah satu tahap penting dalam proses pengujian sistem adalah penerapan model yang telah dilatih sebelumnya (`best.pt`) ke dalam sistem deteksi objek yang bekerja pada data statis, yaitu gambar. Model `best.pt`, yang diperoleh dari hasil pelatihan terbaik, dimuat menggunakan pustaka YOLOv8 dan diterapkan untuk mendeteksi objek dalam sejumlah gambar uji yang disediakan secara manual, bukan melalui sumber video atau kamera langsung.

Dalam tahap ini, gambar-gambar uji diproses satu per satu. Model akan melakukan inferensi terhadap setiap gambar, mendeteksi keberadaan objek, dan mengidentifikasi kelas dari setiap objek tersebut. Hasil deteksi divisualisasikan dalam bentuk bounding box (kotak pembatas) lengkap dengan label kategori serta nilai probabilitas atau tingkat kepercayaan dari prediksi yang dihasilkan oleh model.

Langkah ini dirancang untuk mengevaluasi keakuratan dan ketangguhan sistem dalam mengidentifikasi objek pada gambar statis yang tidak termasuk dalam data pelatihan. Dengan demikian, pengujian ini memberikan gambaran nyata mengenai kemampuan generalisasi model terhadap data baru.

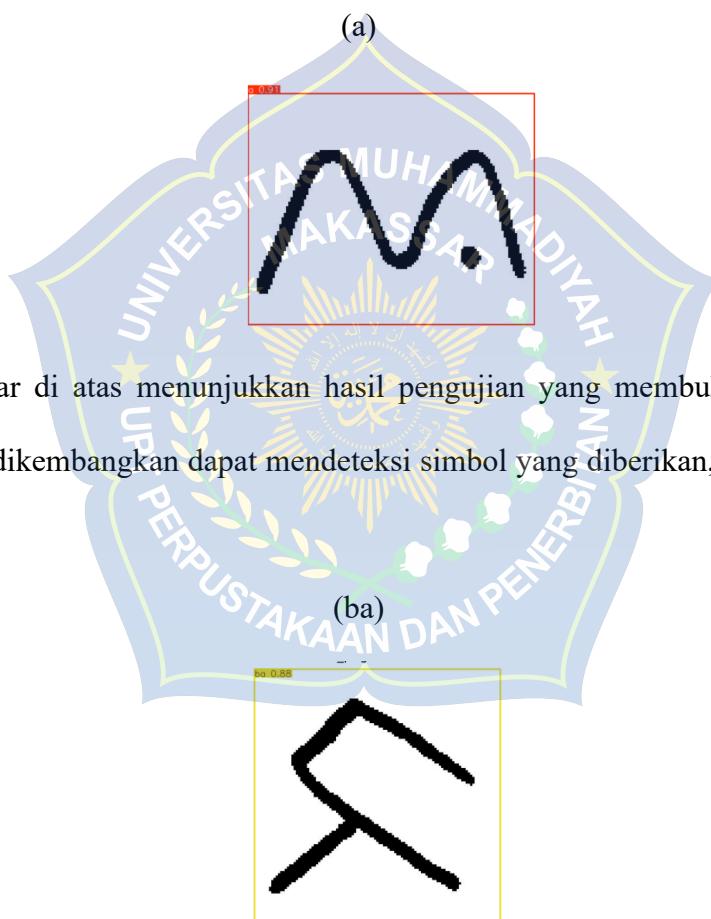
Model YOLOv8 yang telah dilatih digunakan untuk melakukan deteksi objek pada kumpulan gambar uji. Setelah model `best.pt` berhasil dimuat ke dalam sistem, program memulai proses inferensi terhadap gambar-gambar yang tersimpan dalam folder pengujian. Setiap gambar dianalisis untuk mengidentifikasi kemungkinan keberadaan objek sesuai dengan label yang telah didefinisikan dalam data pelatihan.

Ketika objek berhasil dikenali, sistem menampilkan hasil deteksi dalam bentuk kotak pembatas yang mengelilingi objek, disertai label kelas serta nilai keyakinan (confidence score). Deteksi ini tidak dilakukan secara real-time dari aliran video,

melainkan secara berurutan pada setiap gambar statis. Jika tingkat kepercayaan suatu deteksi melebihi ambang batas 50%, maka hasilnya akan ditampilkan secara otomatis dan disimpan dalam folder output.

Dengan pendekatan ini, sistem diuji berdasarkan kemampuannya mendeteksi objek secara akurat pada gambar individual, tanpa keterlibatan kamera atau input video secara langsung. Hasil dari pengujian ini menjadi dasar untuk menilai performa aktual model pada data uji yang baru.

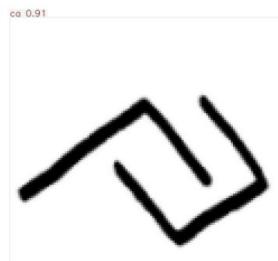
Hasil pengujian pada gambar:



Gambar di atas menunjukkan hasil pengujian yang membuktikan bahwa sistem yang dikembangkan dapat mendeteksi simbol yang diberikan, seperti huruf "a".

Gambar tersebut menunjukkan hasil pengujian yang membuktikan bahwa sistem yang dirancang dapat mendeteksi simbol yang diberikan, seperti huruf "ba".

(ca)



Gambar tersebut merupakan hasil uji coba yang menunjukkan bahwa sistem yang dikembangkan dapat mengenali simbol yang diberikan, seperti huruf “ca”.

Gambar 14. Hasil Pengujian.

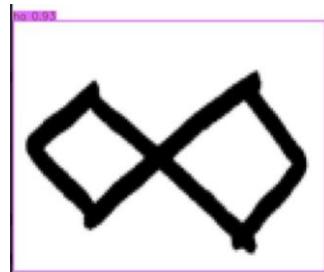
Gambar tersebut menunjukkan bagaimana deteksi objek dilakukan secara langsung menggunakan model YOLO yang telah dilatih khusus untuk mengenali huruf dari ‘a’ sampai ‘ya’. Dalam uji coba ini, kamera webcam digunakan untuk merekam video secara langsung, dan setiap gambar yang dihasilkan akan dianalisis oleh model YOLO untuk mengenali dan mengklasifikasikan huruf-huruf yang terlihat dalam gambar.

Hasil observasi menunjukkan bahwa model YOLO bekerja dengan sangat baik, dengan akurasi tinggi dalam mengenali setiap huruf. Setiap huruf yang berhasil diidentifikasi akan diberi tanda kotak merah di sekelilingnya, lengkap dengan label huruf dan nilai kepercayaan atau probabilitas terkait hasil deteksi itu. Sebagai contoh, jika model mendekripsi huruf ‘a’ dengan tingkat kepercayaan 92%, maka label “a 92%” akan muncul di atas kotak yang mengelilingi huruf tersebut.

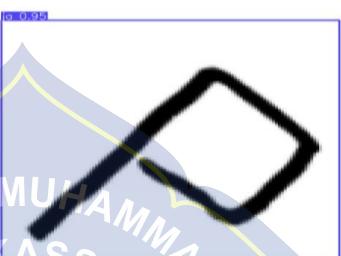
Tabel 2 . Hasil Pengujian

Data pengujian	Kelas dataset	Hasil Deteksi	Tingkat Kepastian
	A		92%
			88%
			91%
			90%

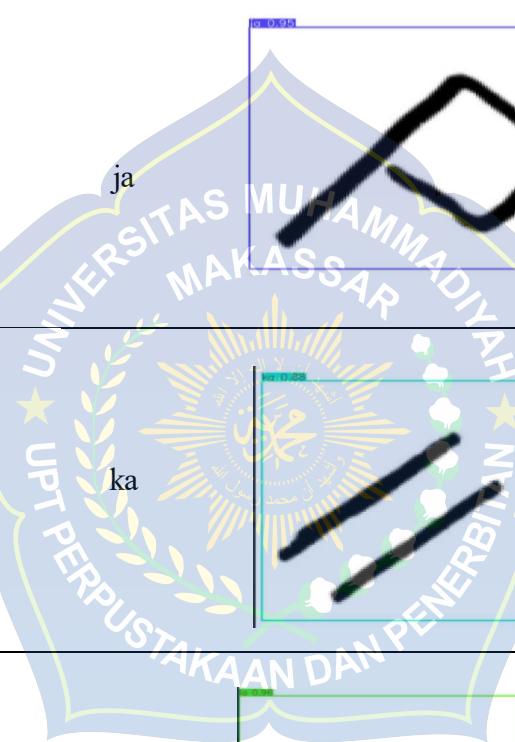
ha



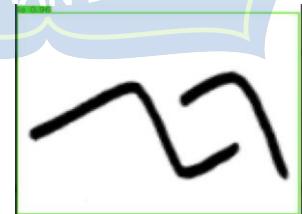
ja

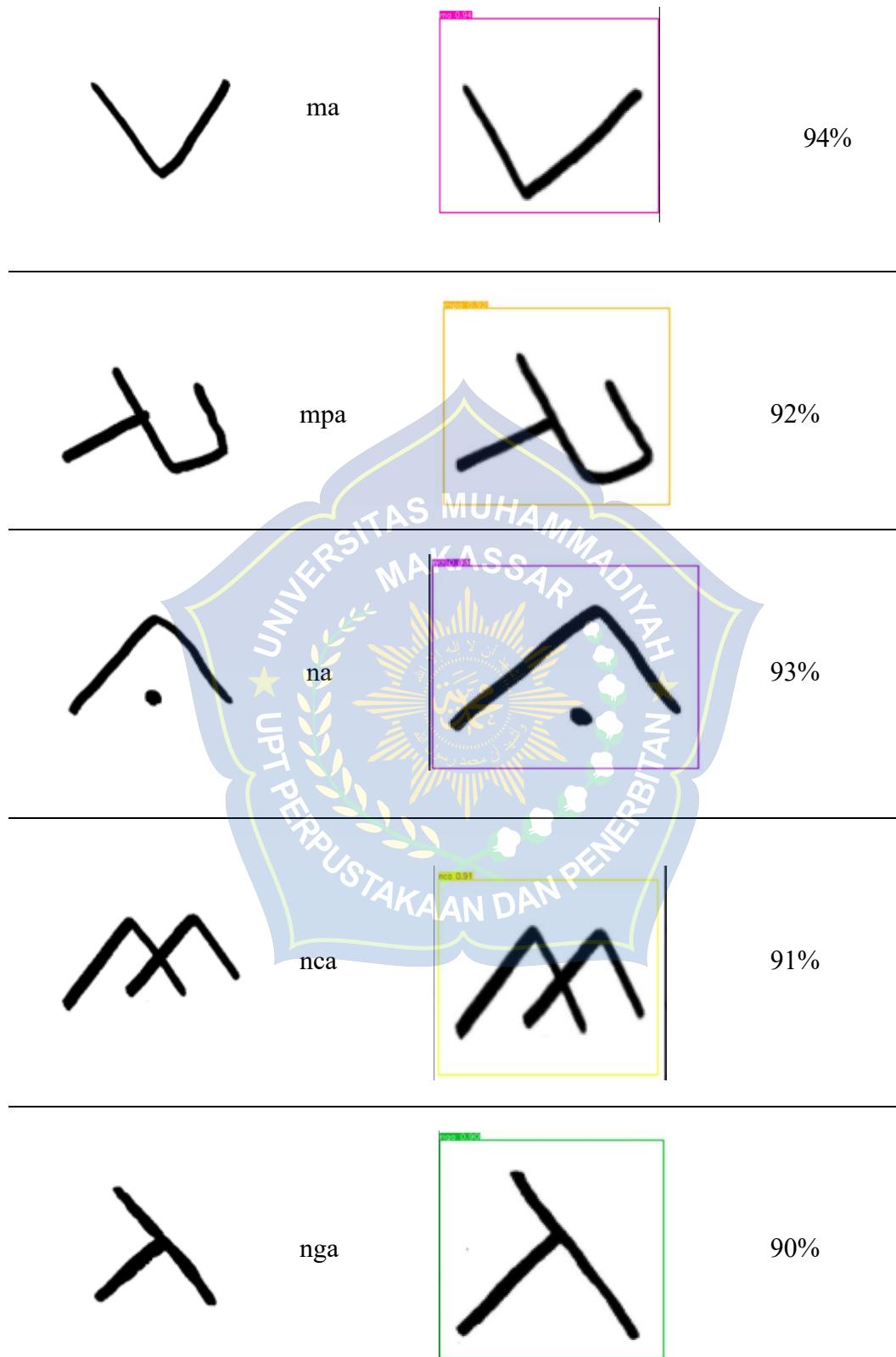


ka

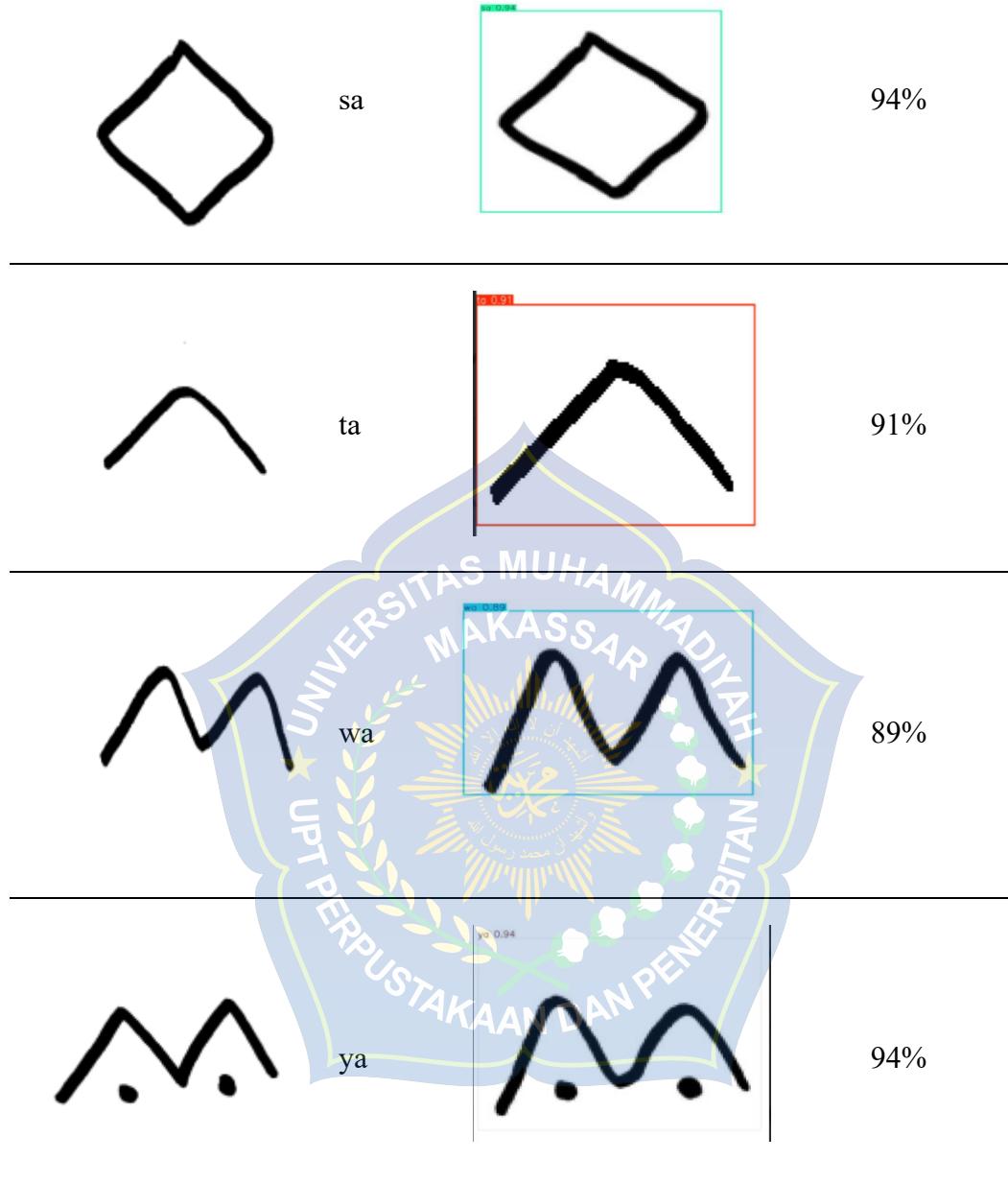


la



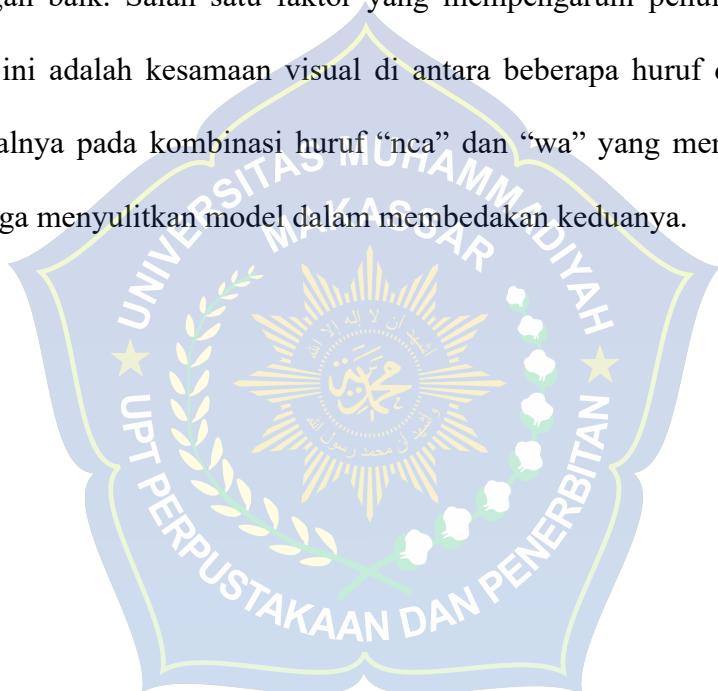


	ngka		95%
	nra		96%
	nya		85%
	pa		92%
	ra		91%



Dalam fase pengujian, digunakan sekumpulan data yang mencakup 23 kategori gambar karakter Lontara, dengan masing-masing kategori berisi 32 gambar. Dalam pengujian ini, model deteksi objek YOLOv8 digunakan sebagai alat utama untuk melakukan identifikasi, dan hasil evaluasi menunjukkan bahwa tingkat

kepercayaan tertinggi yang dicapai oleh model dalam mengenali objek berada pada angka 96%. Sementara itu, tingkat keyakinan terendah tercatat berada di kisaran 85%, mencerminkan adanya variasi dalam kemampuan model dalam mengidentifikasi objek secara konsisten pada seluruh kategori data yang diuji.. Persentase kepercayaan yang lebih rendah tersebut menunjukkan bahwa model masih mengalami tantangan dalam mengenali atau mengklasifikasikan karakter tertentu dengan baik. Salah satu faktor yang mempengaruhi penurunan tingkat kepercayaan ini adalah kesamaan visual di antara beberapa huruf dalam aksara Lontara, misalnya pada kombinasi huruf “nca” dan “wa” yang memiliki bentuk mirip, sehingga menyulitkan model dalam membedakan keduanya.



BAB V

PENUTUP

A. Kesimpulan

Kesimpulan ditetapkan dari penelitian ini oleh peneliti berdasarkan hasil penelitian yang telah dilaksanakan sebelumnya adalah:

1. Penerapan metode YOLO dalam pendekalian objek huruf Lontara terbukti berhasil dan mampu mengenali karakter huruf tersebut dengan baik.
2. Dataset yang digunakan berjumlah 805 citra huruf Lontara yang terbagi ke dalam 23 kelas. Data ini dipecah dengan tiga kategori, yakni dataset training(70%), dataset valida (15%), dan dataset testing(15%). Model YOLO yang diterapkan dalam sistem deteksi objek mencapai akurasi tinggi sebesar 99,5%, dengan presisi 98,4% dan recall 100%. Tingkat kepercayaan deteksi pada setiap kelas berada pada kisaran 86% hingga 94%. Hasil ini membuktikan bahwa metode deteksi objek berbasis YOLO mampu memberikan performa yang akurat dan andal pada tahap pengujian.

B. Saran

Berdasarkan temuan penelitian ini, terdapat sejumlah saran yang dapat dijadikan acuan untuk pengembangan studi di masa mendatang:

1. Menerapkan algoritma terkini atau mengombinasikan beberapa metode algoritmik guna meningkatkan efektivitas proses deteksi dan klasifikasi.
2. Mengembangkan pendekatan deteksi dan klasifikasi yang lebih efisien dengan meneliti teknik-teknik yang mampu meningkatkan akurasi,

khususnya pada situasi dengan pencahayaan minim maupun jarak objek yang lebih jauh.



DAFTAR PUSTAKA

- Arhinza, R. S., Sari, A. P., & Akbar, F. A, 2023. Klasifikasi citra aksara Lontara menggunakan K-NN dan ekstraksi fitur HOG. *Generation Journal*, 7(2), 149–156. Universitas Nusantara PGRI Kediri.
- Faizin, Muhammad Arif (2023) *Deteksi Aksara Jawa Menggunakan YOLO untuk Transliterasi Berbasis LSTM pada Manusrip Jawa Kuno*. Other thesis, Institut Teknologi Sepuluh Nopember.
- Hamka, Alif, 2019. Aksara Lontara Kebudayaan Asli Sulawesi Selatan. Makalah Antropologi Budaya Universitas Hasanuddin Makassar.
- Hendri, dkk, 2021. Penerapan Machine Learning Untuk Mengategorikan Sampah Plastik Rumah Tangga. *Jurnal Teknik Informatika STMIK TIME Medan*, 10 (1), 1-5.
- Hayati, Juliyani Nurhaliza, dkk, 2023. Object Tracking Menggunakan Algoritma You Only Look Once (Yolo) V8 Untuk Menghitung Kendaraan. *Jurnal Teknik Informatika, Sekolah Tinggi Teknologi Wastukancana*, 12 (2), 91-99.
- Hidayah, Aisyah Nurul, 2020. Deteksi Huruf Pada Tulisan Tangan Latin Menggunakan Metode You Only Look Once (Yolo). Tesis Fakultas Teknologi Elektro dan Informatika Cerdas Institut Teknologi Sepuluh Nopember Surabaya.
- Hidayat, R., & Rahmatullah, A, 2023. Peran Artificial Intelligence dalam pembelajaran pendidikan agama Islam. *Syntax Literate: Jurnal Ilmiah Indonesia*, 8(4), 10–18.
- Imantiyar, R., & Fudholi, D. H. (2021). Kajian Pengaruh Dataset dan Bias Dataset terhadap Performa Akurasi Deteksi Objek. *Jurnal Universitas Islam Indonesia* 14 (2), 258–268.
- KBBI (Kamus Besar Bahasa Indonesia), 2024. Kamus versi online/daring (Dalam Jaringan). di akses pada 02 Oktober 2024. <https://kbbi.web.id/didik>.
- Kilawati, Andi & Yanti, Rosmalah, 2021. Aktualisasi Baca Tulis Aksara Lontara Melalui Aplikasi Bugis Keyboard Berbasis Android. *Jurnal Universitas Cokroaminoto Palopo, Indonesia*, 1 (1), 50-55.

Manandhar, Roshan 2023. Kecerdasan Buatan: Apa Itu & Bagaimana Pengaruhnya terhadap Pengembangan Aplikasi? <https://ebpearls.com.au/artificial-intelligence-for-app-development/>, diakses Minggu, 10 November 2024, Pukul 13.53 Wita.

Maleh, Dwijaya Made I, dkk,2023. Implementasi Algoritma You Only Look Once (YOLO) Untuk Object Detection Sarang Orang Utan. Jurnal Jurusan Teknik Informatika, Fakultas Teknik, Universitas Palangkaraya, 10 (1), 19-27.

Nurmalasari, Alifah, 2021. Perancangan Aplikasi Pembelajaran Aksara Lontara Suku Bugis. Skripsi Program Sarjana Teknik Informatika Fakultas Teknologi Informasi Universitas YASRI Makassar.

Pratama, A., Lestari, N., & Gunawan, B, 2023. Analisa komparatif algoritma YOLO dan SSD untuk deteksi objek. *Prosiding Seminar Nasional Inovasi dan Aplikasi Teknologi di Industri*, 4(1), 1–8.

Rosaly, R. 2019. *Definisi Flowchart, Fungsinya, serta Simbol-simbol Flowchart yang Paling Sering Digunakan*.

Suryawan, I., & Nugroho, A, 2022. Penerapan metode YOLO untuk deteksi objek pada sistem pengawasan cerdas. *Jurnal Nasional Teknik Informatika dan Aplikasinya*, 10(2), 40–48.

Suharto, Aslam Fikri Ahmad, 2020. Pembangan Media Pembelajaran Bahasa Lontara Bugis Berbasis Augmented Reality Menggunakan Unity 3D. Skripsi Program Studi Ilmu Komputer Departemen Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Hasanuddin Makassar.

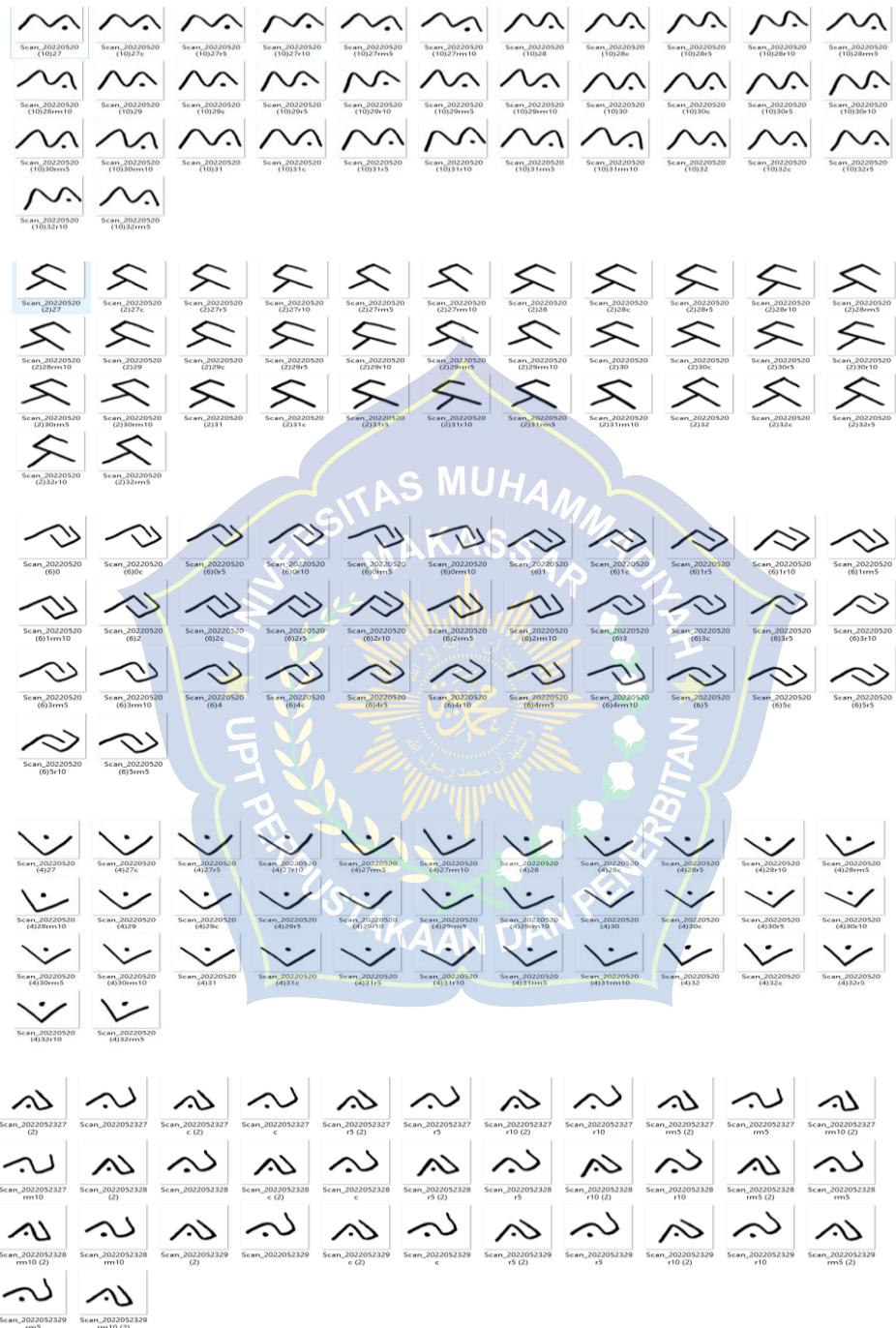
Wibowo, R., & Sari, D, 2024. Implementasi YOLOv8 untuk deteksi objek daur ulang secara real-time. *Indonesian Journal of Computer Science*, 3(1), 1–6.

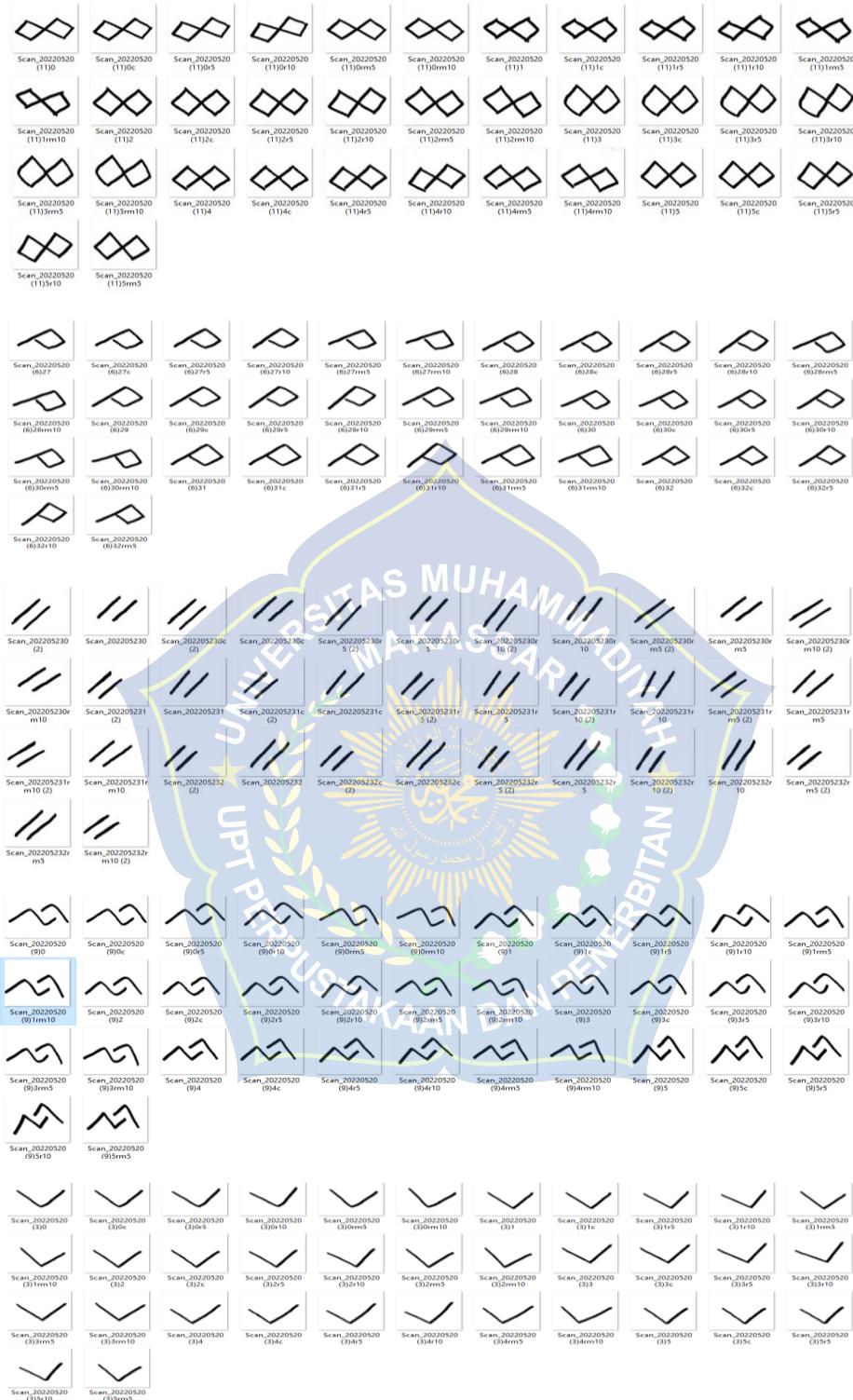
Wulandari, Ayu Dyah,2018. Adroid dan Perkembangannya. Makalah Kurikulum dan Teknologi Pendidikan Fakultas Ilmu Pendidikan Universitas Negeri Semarang.

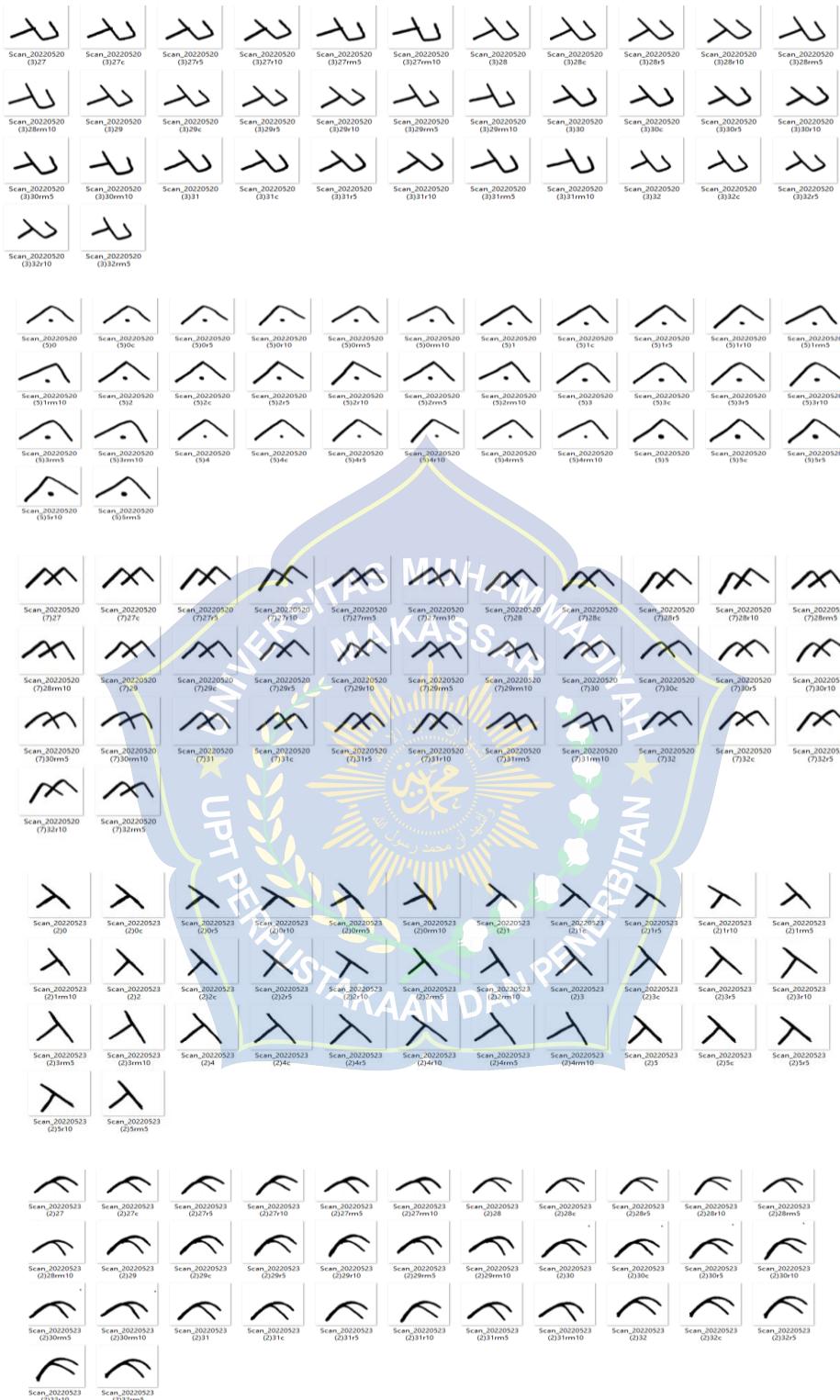
Zophie, Jeremia & Triharminto, Hendri Himawan, 2022. Implemetasi Algoritma You Only Look Once (YOLO) menggunakan Web Camera untuk Mendeteksi Objek Statis dan Dinamis. *Jurnal Departemen Elektronika, Akademi Angkatan Udara*, Yogyakarta. 1,(1), 98-109.

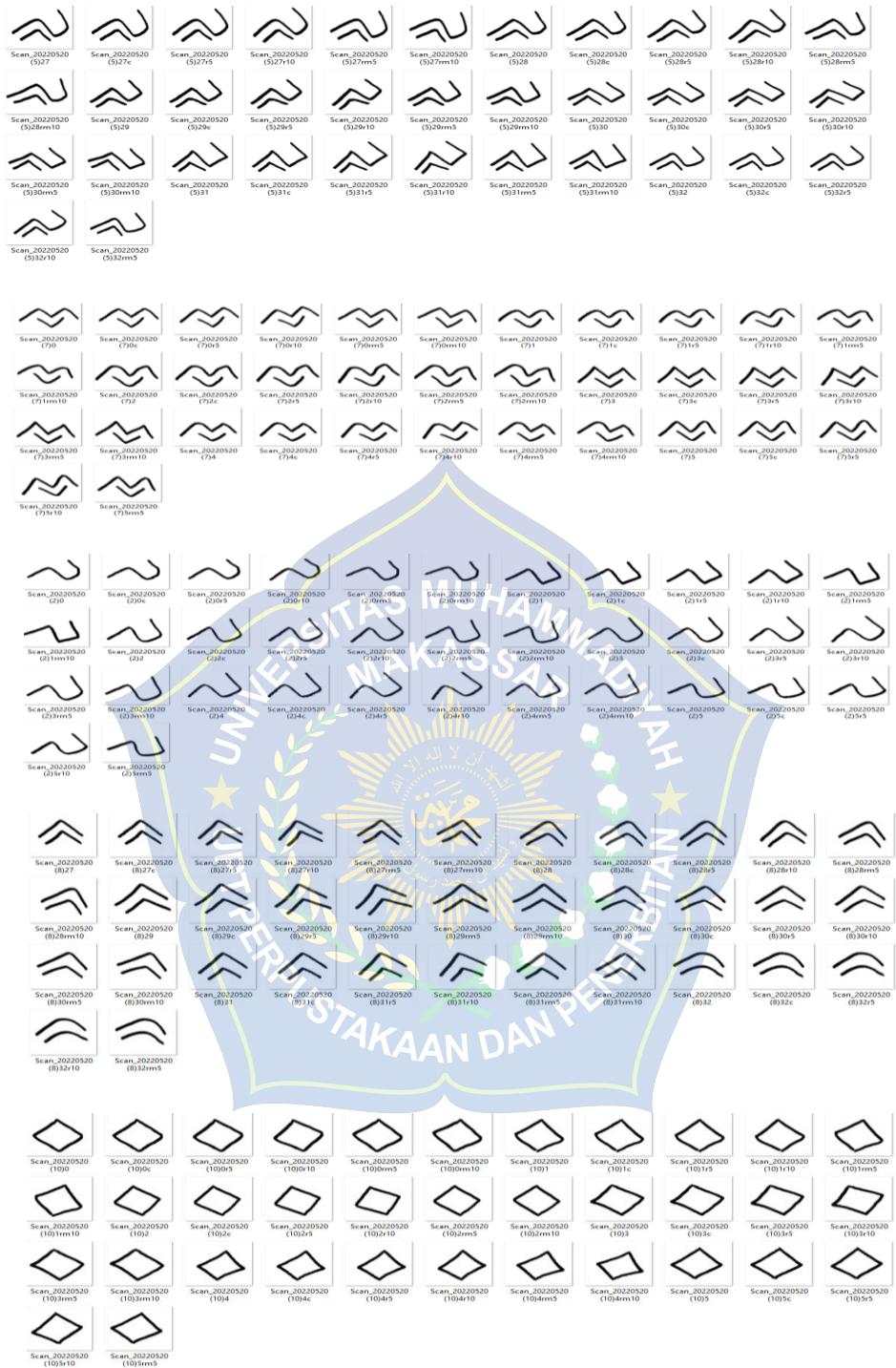
LAMPIRAN:

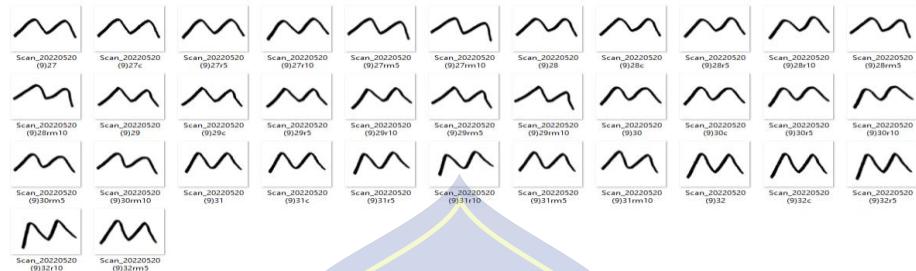
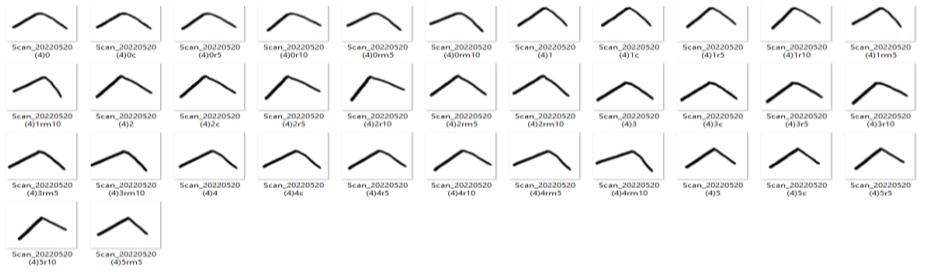
GAMBAR AKSARA LONTARA











Uji Model Train

```
!pip install ultralytics
```

```
!pip install roboflow
```

```
from roboflow import Roboflow
rf = Roboflow(api_key="mcvtQQsHY0YtMXeM8q6x")
project = rf.workspace("deteksi-aksara-lontara").project("deteksi-aksara-lontara")
version = project.version(8)
dataset = version.download("yolov8")
```

```
from ultralytics import YOLO
# Cek versi
import ultralytics
ultralytics.checks()

!ls /content

from ultralytics import YOLO
# Inisialisasi model YOLOv8 (versi ringan)
model = YOLO('yolov8n.pt') # Bisa ganti ke yolov8s.pt,
yolov8m.pt, dsb

# Latih model
model.train(
    data="/content/deteksi-aksara-lontara-8/data.yaml", #
pastikan ini path yang benar
    epochs=50,
    imgsz=640,
    batch=16 # atur sesuai kemampuan GPU (bisa juga 8)
)
```

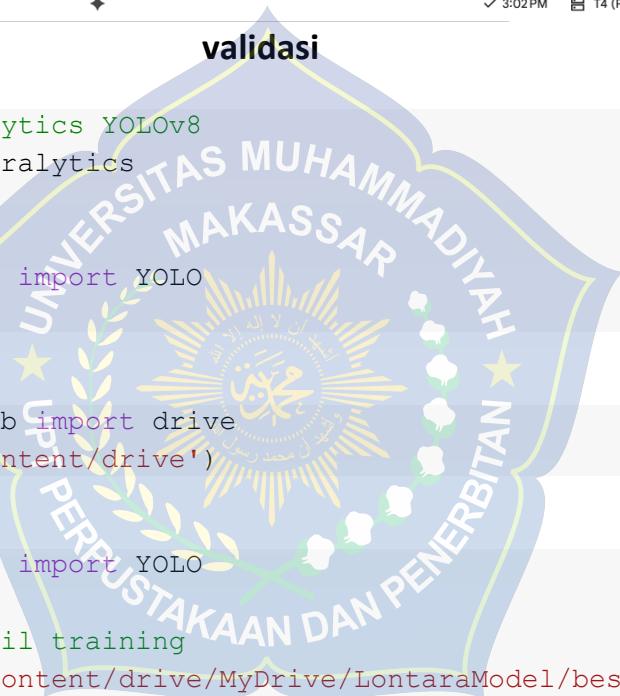
```

Copilot 100% Disk 100% 4/4 [00:02:00]
Validating runs/detect/train2/weights/best.pt...
Ultralytics 8.3.169 Python-3.11.13 torch-2.6.0+cu124 CUDA:0 (Tesla T4, 15095MiB)
Model summary (fused): 72 layers, 3,010,133 parameters, 0 gradients, 8.1 GFLOPs
  Class   Images Instances Box(P) R   mAP50   mAP50-95: 100% [██████████] 4/4 [00:02:00]
    all    121      121    0.983   1   0.995   0.839
    ha      3        3    0.971   1   0.995   0.93
    ja      8        8    0.987   1   0.995   0.816
    ka      4        4    0.985   1   0.995   0.779
    la      6        6    0.984   1   0.995   0.871
    ma      6        6    0.981   1   0.995   0.731
    mpa     5        5    0.982   1   0.995   0.781
    na      6        6    0.98   1   0.995   0.895
    nca     6        6    0.983   1   0.995   0.794
    nga     8        8    0.98   1   0.995   0.724
    nka     4        4    0.975   1   0.995   0.859
    nra     14       14   0.993   1   0.995   0.889
    nya     7        7    0.982   1   0.995   0.821
    pa      3        3    0.963   1   0.995   0.93
    ra      5        5    0.981   1   0.995   0.987
    sa      8        8    0.985   1   0.995   0.923
    ta      12       12   0.992   1   0.995   0.787
    wa      8        8    0.992   1   0.995   0.801
    ya      8        8    0.981   1   0.995   0.858
Speed: 0.3ms preprocess, 3.0ms inference, 0.0ms loss, 7.2ms postprocess per image
Results saved to runs/detect/train2
ultralytics.utils.metrics.DetMetrics object with attributes:

```

Uji

validasi



```

# Install Ultralytics YOLOv8
!pip install ultralytics

# Import
from ultralytics import YOLO

# Load model hasil training
model = YOLO("/content/drive/MyDrive/LontaraModel/best.pt")

# Validasi model pada data validasi
metrics =
model.val(data="/content/drive/MyDrive/deteksi_aksara_lontara/
data.yaml", split="val")

!ls -R "/content/drive/MyDrive/deteksi_aksara_lontara"

!cat "/content/drive/MyDrive/deteksi_aksara_lontara/data.yaml"

!ls -R runs/detect/

```

```
import os
from IPython.display import Image, display

val_dir = "runs/detect/val"
files = [
    "BoxPR_curve.png", "BoxP_curve.png", "BoxR_curve.png",
    "BoxF1_curve.png",
    "confusion_matrix.png", "confusion_matrix_normalized.png",
    "val_batch0_labels.jpg", "val_batch0_pred.jpg",
    "val_batch1_labels.jpg", "val_batch1_pred.jpg",
    "val_batch2_labels.jpg", "val_batch2_pred.jpg"
]

for fname in files:
    fpath = os.path.join(val_dir, fname)
    if os.path.exists(fpath):
        print(f"Menampilkan: {fname}")
        display(Image(filename=fpath))
    else:
        print(f"Tidak ditemukan: {fname}")
```

```
# 1. Install dulu (WAJIB setelah setiap restart runtime)
!pip install ultralytics opencv-python

# 2. Mount Google Drive
from google.colab import drive
drive.mount('/content/drive')

# 3. Import library
from ultralytics import YOLO
import cv2
import os
import numpy as np
```

```

from IPython.display import Image, display
import matplotlib.pyplot as plt

# 4. Muat model YOLO
model = YOLO('/content/drive/MyDrive/LontaraModel/best.pt')

# 5. Path gambar uji
source_path =
'/content/drive/MyDrive/deteksi_aksara_lontara/test/images'

# 6. Buat folder output
output_dir =
'/content/drive/MyDrive/deteksi_aksara_lontara/output_custom'
os.makedirs(output_dir, exist_ok=True)

# 7. Prediksi
results = model.predict(
    source=source_path,
    conf=0.25, # Bisa disesuaikan jika terlalu
banyak/noise
    iou=0.6,
    save=False,
    stream=True
)

# 8. Buat warna yang konsisten untuk setiap kelas (maks 20
kelas)
def get_distinct_colors(n):
    cmap = plt.get_cmap("tab20")
    return [tuple(int(255 * c) for c in cmap(i % 20)[:3]) for
i in range(n)]

num_classes = len(model.names)
colors = {cls_id: get_distinct_colors(num_classes)[cls_id] for
cls_id in range(num_classes)}

# 9. Proses dan simpan hasil
for r in results:
    im = r.orig_img.copy()

    for box in r.boxes:
        x1, y1, x2, y2 = map(int, box.xyxy[0])
        cls = int(box.cls[0])
        conf = float(box.conf[0])
        color = colors[cls]

```

```

label = f" {model.names[cls]} {conf:.2f}"

# Gambar bounding box
cv2.rectangle(im, (x1, y1), (x2, y2), color, 2)

# Ukuran label & latar belakang label (agar label
terlihat di gambar terang/gelap)
font_scale = 0.6
font_thickness = 1
(w, h), _ = cv2.getTextSize(label,
cv2.FONT_HERSHEY_SIMPLEX, font_scale, font_thickness)
label_y = y1 - 10 if y1 - 10 > 10 else y1 + 20

# Latar belakang label
cv2.rectangle(im, (x1, label_y - h), (x1 + w,
label_y), color, -1)
# Tulis teks label
cv2.putText(im, label, (x1, label_y - 2),
cv2.FONT_HERSHEY_SIMPLEX, font_scale, (255, 255, 255),
font_thickness)

# Simpan hasil
img_name = os.path.basename(r.path)
save_path = os.path.join(output_dir, img_name)
cv2.imwrite(save_path, im)

# 10. Tampilkan hasil
for img_name in sorted(os.listdir(output_dir)):
    if img_name.endswith('.jpg') or img_name.endswith('.png'):
        display(Image(filename=os.path.join(output_dir,
img_name)))

```

Kesimpulan

Model

- Telah berhasil mendeteksi seluruh kelas huruf Lontara dengan sangat **tinggi dan stabil**.
- mAP@0.5 = **0.995 (99.5%)**, menunjukkan kualitas model sangat baik.
- Grafik PR ini menunjukkan **hampir semua kelas memiliki PR curve yang tinggi dan mendekati ideal**.



**MAJELIS PENDIDIKAN TINGGI PIMPINAN PUSAT MUHAMMADIYAH
UNIVERSITAS MUHAMMADIYAH MAKASSAR
UPT PERPUSTAKAAN DAN PENERBITAN**

Alamat Kantor: Jl. Sultan Alauddin No.259 Makassar 90221 Tlp. (0411) 866972, 881593, Fax. (0411) 865588

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

SURAT KETERANGAN BEBAS PLAGIAT

**UPT Perpustakaan dan Penerbitan Universitas Muhammadiyah Makassar,
Menerangkan bahwa mahasiswa yang tersebut namanya di bawah ini:**

Nama : Mohammad Ramadhan

Nim : 105841113520

Program Studi : Teknik Informatika

Dengan nilai:

No	Bab	Nilai	Ambang Batas
1	Bab 1	9%	10 %
2	Bab 2	11%	25 %
3	Bab 3	6%	10 %
4	Bab 4	6%	10 %
5	Bab 5	5%	5 %

Dinyatakan telah lulus cek plagiat yang diadakan oleh UPT- Perpustakaan dan Penerbitan
Universitas Muhammadiyah Makassar Menggunakan Aplikasi Turnitin.

Demikian surat keterangan ini diberikan kepada yang bersangkutan untuk dipergunakan
seperlunya.

Makassar, 23 Agustus 2025

Mengetahui,

Kepala UPT- Perpustakaan dan Penerbitan,


Nursiman Sadiqun, M.I.P

NBM- 964 591

Jl. Sultan Alauddin no 259 makassar 90222
Telepon (0411)866972, 881 593,fax (0411)865 588
Website: www.library.unismuh.ac.id
E-mail : perpustakaan@unismuh.ac.id

Bab I Mohammad Ramadhan
Zainuddin 105841113520



Submission date: 20-Aug-2025 06:41AM (UTC+0700)

Submission ID: 2732093042

File name: SKRIPSI_RAMA_BAB_I.docx (72.29K)

Word count: 940

Character count: 6221

9% SIMILARITY INDEX 6% INTERNET SOURCES 2% PUBLICATIONS 6% STUDENT PAPERS

PRIMARY SOURCES

- | | | |
|---|---|----|
| 1 | digilibadmin.unismuh.ac.id
Internet Source | 2% |
| 2 | I Made Dwijaya Maleh, Rony Teguh, Abertun Sagit Sahay, Simon Okta, Muhammad Porkab Pratama. "Implementasi Algoritma You Only Look Once (YOLO) Untuk Object Detection Sarang Orang Utan Di Taman Nasional Sebangau", Jurnal Informatika, 2023
Publication | 2% |
| 3 | Submitted to Universitas Muhammadiyah Makassar
Student Paper | 1% |
| 4 | Submitted to Universitas Muhammadiyah Purwokerto
Student Paper | 1% |
| 5 | docobook.com
Internet Source | 1% |
| 6 | github.com
Internet Source | 1% |
| 7 | docplayer.info
Internet Source | 1% |

Exclude quotes Off
Exclude bibliography Off

Exclude matches Off

Bab II Mohammad Ramadhan

Zainuddin 105841113520

by Tahap Tutup



Submission date: 20-Aug-2025 06:42AM (UTC+0700)

Submission ID: 2732093255

File name: SKRIPSI_RAMA_BAB_II_tutup.docx (307.36K)

Word count: 2949

Character count: 20210

SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS
<hr/>			
11 %	10 %	5 %	4 %
PRIMARY SOURCES			
1 ojs.unikom.ac.id Internet Source			2 %
2 repository.its.ac.id Internet Source			1 %
3 www.scribd.com Internet Source			1 %
4 digilib.uinsby.ac.id Internet Source			1 %
5 eprints.pktj.ac.id Internet Source			1 %
6 digilib.uin-suka.ac.id Internet Source			1 %
7 Submitted to Institut Bisnis dan Teknologi Indonesia (INSTIKI) Student Paper			1 %
8 www.cnblogs.com Internet Source			1 %
9 Submitted to Universitas Negeri Semarang - iTh Student Paper			<1 %
10 Yozika Arvio, Dine Tiara Kusuma, Iriansyah BM Sangadji. "PENDEKATAN ALGORITMA YOLO V5 UNTUK MENDETEKSI CACAT PRODUK MASKER", Dinamika Rekayasa, 2024 Publication			<1 %



11	digilib.iain-jember.ac.id Internet Source	<1 %
12	Dadang Iskandar Mulyana, Sahroni. "Optimasi Penerapan Algoritma Yolo dan Data Augmentasi dalam Klasifikasi Pakaian Tradisional Kebaya", Jurnal JTIK (Jurnal Teknologi Informasi dan Komunikasi), 2024 Publication	<1 %
13	docplayer.info Internet Source	<1 %
14	id.123dok.com Internet Source	<1 %
15	repository.iainkudus.ac.id Internet Source	<1 %
16	repository.uinsu.ac.id Internet Source	<1 %
17	www.coursehero.com Internet Source	<1 %
18	I Made Dwijaya Maleh, Rony Teguh, Abertun Sagit Sahay, Simon Okta, Muhammad Porkab Pratama. "Implementasi Algoritma You Only Look Once (YOLO) Untuk Object Detection Sarang Orang Utan Di Taman Nasional Sebangau", Jurnal Informatika, 2023 Publication	<1 %
19	Muhammad Farrel Golfantara. "PENGGUNAAN ALGORITMA YOLO V8 UNTUK IDENTIFIKASI REMPAH-REMPAH", Jurnal Informatika dan Teknik Elektro Terapan, 2024 Publication	<1 %

Bab III Mohammad Ramadhan
Zainuddin 105841113520



Submission date: 20-Aug-2025 06:43AM (UTC+0700)

Submission ID: 2732093626

File name: SKRIPSI_RAMA_BAB_III.docx (56.9K)

Word count: 1370

Character count: 9264



PRIMARY SOURCES

- | | | |
|---|---|-----|
| 1 | Submitted to Universitas Dian Nuswantoro
Student Paper | 2% |
| 2 | Rizky Sulaiman, Setio Basuki. "IMPLEMENTASI ALGORITMA CONDITIONAL RANDOM FIELDS UNTUK PART OF SPEECH TAGGING BAHASA MADURA", JSil (Jurnal Sistem Informasi), 2025
Publication | 1 % |
| 3 | digilib.uinkhas.ac.id
Internet Source | 1 % |
| 4 | Submitted to Universitas Sebelas Maret
Student Paper | 1 % |
| 5 | Dadang Iskandar Mulyana, Rifdah Alifah.
"Optimasi Deteksi Tumor Otak Menggunakan Adaptive Multiscale Retinex dan YOLOV10 Pada Citra Digital", Jurnal Indonesia : Manajemen Informatika dan Komunikasi, 2024
Publication | 1 % |

Exclude quotes Off
Exclude bibliography Off

Exclude matches Off

Bab IV Mohammad Ramadhan Zainuddin 10584113520

by Tahap Tutup



Submission date: 20-Aug-2025 06:44AM (UTC+0700)

Submission ID: 2732094048

File name: SKRIPSI_RAMA_BAB_IV.docx (7.25M)

Word count: 2394

Character count: 15178

6%	3%	2%	2%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

- | | | |
|----------|--|---------------|
| 1 | kc.umn.ac.id | 2% |
| | Internet Source | |
| 2 | Muhammad Khatama Insani, Dwi Budi Santoso. "Perbandingan Kinerja Model Pre-Trained CNN (VGG16, RESNET, dan INCEPTIONV3) untuk Aplikasi Pengenalan Wajah pada Sistem Absensi Karyawan", Jurnal Indonesia : Manajemen Informatika dan Komunikasi, 2024 | 1% |
| | Publication | |
| 3 | Submitted to Cranfield University | 1% |
| | Student Paper | |
| 4 | Submitted to Universitas Pancasila | 1% |
| | Student Paper | |
| 5 | Andriani Marshanda Putri, Widya Khafa Nofa, Dewi Anggraini Puspa Hapsari. "PENERAPAN METODE BERT UNTUK ANALISIS SENTIMEN ULASAN PENGGUNA APLIKASI SEGARI DI GOOGLE PLAY STORE", Jurnal Ilmiah Teknik, 2025 | 1% |
| | Publication | |
| 6 | docplayer.info | <1% |
| | Internet Source | |



Bab V Mohammad Ramadhan
Zainuddin 105841113520

Submission date: 20-Aug-2025 08:32AM (UTC+0700)

Submission ID: 2732138182

File name: SKRIPSI_RAMA_BAB_V.docx (30.36K)

Word count: 182

Character count: 1163

Bab V Mohammad Ramadhan Zainuddin 105841113520

ORIGINALITY REPORT



PRIMARY SOURCES

1	e-jurnalmitrapendidikan.com	5%
	Internet Source	

Exclude quotes Off
Exclude bibliography Off

