

No. Reg: 241142010110051

LAPORAN PENELITIAN



TRANSLITERASI "HARAH JAWOE "-LATIN MENGGUNAKAN METODE CONVOLUTIONAL NEURAL NETWORK DENGAN ARSITEKTUR YOLOV8

Ketua Peneliti

Khairan AR, M.Kom

NIDN: 2004078602

NIPN: 200407860210070

Anggota:

1. Hendri Ahmadian, M.I.M
2. Aesha Durratul Nasihah

Klaster	Penelitian Dasar Interdisipliner
Bidang Ilmu Kajian	Sains dan Teknologi
Sumber Dana	DIPA UIN Ar-Raniry Banda Aceh Tahun 2024

**PUSAT PENELITIAN DAN PENERBITAN
LEMBAGA PENELITIAN DAN PENGABDIAN KEPADA MASYARAKAT
UNIVERSITAS ISLAM NEGERI AR-RANIRY BANDA ACEH
SEPTEMBER 2024**

**LEMBARAN IDENTITAS DAN PENGESAHAN LAPORAN PENELITIAN
PUSAT PENELITIAN DAN PENERBITAN LP2M UIN AR-RANIRY BANDA ACEH
TAHUN 2024**

1. a. Judul : Transliterasi "Harah Jawoe"-Latin Menggunakan Metode Convolutional Neural Network dengan Arsitektur YOLOv8
- b. Klaster : Penelitian Dasar Interdisipliner
- c. No. Registrasi : 241142010110051
- d. Bidang Ilmu yang diteliti : Sains dan Teknologi

2. Peneliti/Ketua Pelaksana
 - a. Nama Lengkap : Khairan AR, M.Kom
 - b. Jenis Kelamin : Laki-Laki
 - c. NIP *(Kosongkan bagi Non PNS)* : 198607042014031001
 - d. NIDN : 2004078602
 - e. NIPN (ID Peneliti) : 200407860210070
 - f. Pangkat/Gol. : Penata Tk. 1/III d
 - g. Jabatan Fungsional : Lektor
 - h. Fakultas/Prodi : Sains dan Teknologi/Teknologi Informasi

 - i. Anggota Peneliti 1
 - Nama Lengkap : Hendri Ahmadian, M.I.M
 - Jenis Kelamin : Laki-Laki
 - Fakultas/Prodi : Sains dan Teknologi/Teknologi Informasi

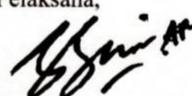
 - j. Anggota Peneliti 2 *(Jika Ada)*
 - Nama Lengkap : Aesha Durratul Nasihah
 - Jenis Kelamin : Perempuan
 - Fakultas/Prodi : Sains dan Teknologi/Teknologi Informasi

3. Lokasi Kegiatan : Banda Aceh
4. Jangka Waktu Pelaksanaan : 6 (enam) Bulan
5. Tahun Pelaksanaan : 2024
6. Jumlah Anggaran Biaya : Rp. 30.000.000
7. Sumber Dana : DIPA UIN Ar-Raniry B. Aceh Tahun 2023
8. *Output dan Outcome* : a. Laporan Penelitian; b. Publikasi Ilmiah; c. HKI

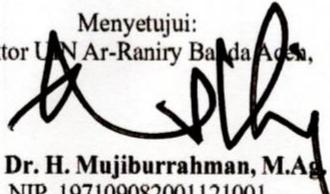
Mengetahui,
Kepala Pusat Penelitian dan Penerbitan
LP2M UIN Ar-Raniry Banda Aceh,


Dr. Anton Widyanto, M. Ag.
NIP. 197610092002121002

Banda Aceh, 5 September 2024
Pelaksana,


Khairan AR, M.Kom
NIDN. 200407860210070

Menyetujui:
Rektor UIN Ar-Raniry Banda Aceh,


Prof. Dr. H. Mujiburrahman, M.Ag.
NIP. 197109082001121001

PERNYATAAN

Saya yang bertanda tangan di bawah Ini:

Nama : **Khairan AR**
NIDN : 2004078602
Jenis Kelamin : Laki-Laki
Tempat/ Tgl. Lahir : Ambon/04 Juli 1986
Alamat : Desa Lampeudaya
Fakultas/Prodi : Sains dan Teknologi/Teknologi Informasi

Dengan ini menyatakan dengan sesungguhnya bahwa penelitian yang berjudul: **“Transliterasi “Harah Jawoe”-Latin Menggunakan Metode Convolutional Neural Network dengan Arsitektur YOLOv8”** adalah benar karya asli saya yang dihasilkan melalui kegiatan yang memenuhi kaidah dan metode ilmiah secara sistematis sesuai otonomi keilmuan dan budaya akademik serta diperoleh dari pelaksanaan penelitian pada klaster Penelitian Dasar Interdisipliner yang dibiayai sepenuhnya dari DIPA UIN Ar-Raniry Banda Aceh Tahun Anggaran 2024. Apabila terdapat kesalahan dan kekeliruan di dalamnya, sepenuhnya menjadi tanggung jawab saya.

Demikian surat pernyataan ini saya buat dengan sesungguhnya.

Banda Aceh, 05 September 2024

Saya yang membuat pernyataan,

Ketua Peneliti,



Khairan AR

NIDN. 200407860

TRANSLITERASI “HARAH JAWOE” -LATIN MENGGUNAKAN METODE CONVOLUTIONAL NEURAL NETWORK DENGAN ARSITEKTUR YOLOV8

Ketua Peneliti:

Khairan AR

Anggota Peneliti:

Hendri Ahmadian; Aesha Durratul Nasihah

Abstrak

Penelitian ini bertujuan untuk mengkaji efektivitas arsitektur YOLOv8 dalam mendeteksi gambar kata huruf Arab Melayu Aceh (harah jawoe). Berdasarkan pengujian, model YOLOv8 small mencapai nilai Mean Average Precision (mAP50-95) tertinggi sebesar 74.4% pada epoch ke-30, namun menunjukkan penurunan performa pada epoch selanjutnya sebelum sedikit meningkat kembali pada epoch ke-100. Model medium dan normal menunjukkan pola fluktuasi performa yang serupa, sementara model big memiliki tingkat konsistensi terbaik, dengan nilai mAP terbaik yang diperoleh adalah sebesar 72.4% pada epoch ke-30 dan ke-100. Temuan ini sejalan dengan penelitian sebelumnya yang menekankan pentingnya tuning jumlah epoch dan parameter pelatihan untuk meningkatkan akurasi deteksi. Penelitian ini juga mengindikasikan bahwa variasi dalam teknik augmentasi data dan manipulasi parameter pelatihan diperlukan untuk mencapai kinerja optimal pada epoch yang lebih tinggi. Berdasarkan pada hasil penelitian dapat disimpulkan bahwa arsitektur YOLOv8 dapat diterapkan secara efektif dalam pendeteksian gambar kata harah jawoe, dengan model big menunjukkan performa terbaik dan memiliki stabilitas yang konsisten untuk setiap epoch. Saran untuk penelitian lanjutan, agar dapat mengeksplorasi lebih lanjut teknik augmentasi data dan parameter pelatihan serta membandingkan dengan arsitektur deep learning lainnya. Hasil penelitian ini memberikan kontribusi signifikan dalam bidang pendeteksian citra dan diharapkan menjadi dasar untuk penelitian lebih mendalam dan komprehensif di masa yang akan datang.

Kata Kunci: YOLOv8, harah jawoe, deteksi gambar, deep learning.

KATA PENGANTAR



Syukur Alhamdulillah kepada Allah SWT, karena dengan rahmat dan hidayah-Nya penulis telah dapat menyelesaikan laporan penelitian dengan judul **“Transliterasi “Harah Jawoe”-Latin Menggunakan Metode Convolutional Neural Network dengan Arsitektur YOLOv8”**. Salawat beriring salam penulis persembahkan kepada Nabi Muhammad SAW, beserta keluarga dan para sahabatnya.

Dalam proses penelitian dan penulisan laporan ini tentu banyak pihak yang ikut memberikan motivasi, bimbingan dan arahan. Oleh karena itu penulis tidak lupa menyampaikan ucapan terima kasih kepada yang terhormat:

1. Rektor Universitas Islam Negeri Ar-Raniry Banda Aceh;
2. Ketua LP2M UIN Ar-Raniry Banda Aceh;
3. Sekretaris LP2M UIN Ar-Raniry Banda Aceh;
4. Kepala Pusat Penelitian dan Penerbitan UIN Ar-Raniry Banda Aceh;
5. Dekan dan Wakil Dekan Fakultas Sains dan Teknologi;
6. Ketua dan Sekretaris Program Studi Teknologi Informasi;
7. Reviewer Penelitian LP2M Tahun 2024.

Akhirnya hanya Allah SWT yang dapat membalas amalan mereka, semoga menjadikannya sebagai amal yang baik.

Harapan penulis, semoga hasil penelitian ini bermanfaat dan menjadi salah satu amalan penulis yang diperhitungkan sebagai ilmu yang bermanfaat di dunia dan akhirat. *Amin ya Rabbal ‘Alamin.*

Banda Aceh, 5 September 2024

Ketua Peneliti,

Khairan AR

DAFTAR ISI

HALAMAN Sampul

Halaman Pengesahan

Halaman Pernyataan

ABSTRAK.....	i
KATA PENGANTAR	ii
DAFTAR ISI.....	iii
DAFTAR TABEL	v
DAFTAR GAMBAR	vi
DAFTAR LAMPIRAN.....	vii

BAB I : PENDAHULUAN

A. Latar Belakang.....	1
B. Rumusan Masalah	4
C. Tujuan Penelitian.....	5
D. Batasan Masalah.....	5
E. Manfaat Penelitian.....	6

BAB II : LANDASAN TEORI

A. Penelitian Terdahulu	7
B. Teori-Teori yang Relevan	11
C. Metode Evaluasi.....	20
D. Kerangka Berfikir Penelitian	24

BAB III : METODE PENELITIAN

A. Jenis Penelitian	25
B. Tahapan Penelitian	25
C. Metode Pengumpulan Data	26
D. Metode Simulasi.....	27
E. Metode Analisis Data	32

F. Alat Bantu Penelitian.....	33
G. Rencana Pembahasan.....	33
H. Waktu dan Lokasi Penelitian	34

BAB IV : HASIL PENELITIAN DAN PEMBAHASAN

A. Anotasi Dataset	36
B. Temuan Hasil Penelitian.....	48
C. Evaluasi Model.....	55
D. <i>Recall, Precision, F1-Score dan Accuracy</i>	69
E. Pembahasan Hasil Penelitian	82

BAB V : PENUTUP

A. Kesimpulan.....	87
B. Saran-saran.....	88

DAFTAR PUSTAKA.....	89
----------------------------	-----------

LAMPIRAN-LAMPIRAN.....	94
-------------------------------	-----------

BIODATA PENELITI	123
-------------------------------	------------

DAFTAR TABEL

Tabel 2.1 Dampak Arsitektur terhadap Tingkat Akurasi.....	9
Tabel 3.1 Jadwal Pelaksanaan Penelitian	35
Tabel 4.1 Jumlah Dataset Awal	37
Tabel 4.2 Jumlah Dataset Setelah Augmentasi.....	45
Tabel 4.3 Pembagian Dataset Latih.....	46
Tabel 4.4 Hasil Akurasi Model <i>Small</i>	49
Tabel 4.5 Hasil Akurasi Model <i>Medium</i>	50
Tabel 4.6 Hasil Akurasi Model Normal	51
Tabel 4.7 Hasil Akurasi Model <i>Big</i>	52
Tabel 4.8 Nilai TP Model <i>Small</i> 30 Epoch	70
Tabel 4.9 Nilai FN Model <i>Small</i> 30 Epoch.....	71
Tabel 4.10 Nilai FP Model <i>Small</i> 30 Epoch.....	72
Tabel 4.11 Nilai TN Model <i>Small</i> 30 Epoch.....	72
Tabel 4.12 <i>Accuracy, Precision, Recall</i> dan <i>F1-Score Small</i>	76
Tabel 4.8 Nilai TP Model <i>Big</i> 30 Epoch	76
Tabel 4.9 Nilai FN Model <i>Big</i> 30 Epoch.....	77
Tabel 4.10 Nilai FP Model <i>Big</i> 30 Epoch	78
Tabel 4.11 Nilai TN Model <i>Big</i> 30 Epoch	79
Tabel 4.12 <i>Accuracy, Precision, Recall</i> dan <i>F1-Score Big</i>	82

DAFTAR GAMBAR

Gambar 2.1 Arsitektur CNN.....	14
Gambar 2.2 Proses Konvolusi.....	15
Gambar 2.3 Ilustrasi proses Stride	16
Gambar 2.4 Proses Penambahan Padding.....	16
Gambar 2.5 Ilustrasi Proses Pooling Layer	18
Gambar 2.6 Arsitektur YOLO.....	20
Gambar 2.7 Confusion Matrix	22
Gambar 2.8 Kerangka Berpikir Penelitian	24
Gambar 3.1 Tahapan Penelitian	26
Gambar 3.2 Alur Implementasi Metode	28
Gambar 3.3 Arsitektur dari YOLOV8.....	31
Gambar 4.1 Dataset "Harah jawoe " awal	36
Gambar 4.2 Proses Crop Dataset.....	38
Gambar 4.3 <i>Dataset setelah Crop</i>	39
Gambar 4.4 Contoh Rotasi Gambar	39
Gambar 4.5 Source Code Rotasi Gambar	40
Gambar 4.6 Contoh Perubahan Warna Gambar	41
Gambar 4.7 Source Code Mengubah Warna Background	42
Gambar 4.8 <i>Contoh Perubahan Warna pada Tulisan</i>	43
Gambar 4.9 Source Code Perubahan Warna Tulisan.....	43
Gambar 4.10 Contoh Pencerahan Gambar	44
Gambar 4.11 Source Code Kecerahan Gambar	44
Gambar 4.12 <i>Struktur Label pada YOLO</i>	46
Gambar 4.13 Contoh Class Id pada Model YOLO	48
Gambar 4.14 Grafik Visualisasi Performa Keseluruhan Model ...	53
Gambar 4.15 Grafik Confusion Matrix 30 Epoch Small.....	56
Gambar 4.16 Grafik Confusion Matrix 50 Epoch Small.....	57
Gambar 4.17 Grafik Confusion Matrix 70 Epoch Small.....	59
Gambar 4.18 Grafik Confusion Matrix 90 Epoch Small.....	60
Gambar 4.19 Grafik Confusion Matrix 100 Epoch Small.....	62
Gambar 4.20 Grafik Confusion Matrix 30 Epoch Big.....	63
Gambar 4.21 Grafik Confusion Matrix 50 Epoch Big.....	65
Gambar 4.22 Grafik Confusion Matrix 70 Epoch Big.....	66
Gambar 4.23 Grafik Confusion Matrix 90 Epoch Big.....	68
Gambar 4.24 Grafik Confusion Matrix 100 Epoch Big.....	69

DAFTAR LAMPIRAN

Lampiran 1 Hasil Deteksi.....	94
Lampiran 2 Grafik Training dan Testing.....	122
Lampiran 3 Biodata Peneliti	123

BAB I

PENDAHULUAN

A. Latar Belakang

Transliterasi adalah proses mengubah tata bahasa suatu kata atau kalimat ke dalam tata bahasa lain, agar lebih dipahami (Yusra dkk., 2020). Dalam proses transliterasi, aksara atau kata-kata dari bahasa asal diubah ke dalam bentuk lain yang sesuai dengan pengucapan atau bunyi kata tersebut. Merujuk pada sejumlah ulasan, proses transliterasi membutuhkan waktu yang cukup lama, terkecuali dilakukan oleh seorang filologi (Widiarti dkk., 2018). Ditambah, fakta dilapangan menunjukkan beragamnya bahasa dan aksara yang terdapat dalam naskah (Arifin Nugraha dkk., 2023), tentu saja hal ini menjadi hambatan bagi masyarakat pada umumnya untuk memahami konteks kata dari naskah tersebut (Nurrahma Kuswati, 2021).

Berdasarkan beberapa catatan, naskah yang ditulis menggunakan bahasa Arab melayu atau aksara jawi merupakan naskah kuno yang sering menjadi objek penelitian transliterasi ((Abdul Ghani dkk., 2009); (Wan dkk., 2013); (Nurwakit dkk., 2020)). Aksara jawi menggabungkan karakter aksara Arab dengan beberapa tambahan aksara yang tidak ada dalam huruf Arab, seperti huruf ز، ن، ك، ف، غ، چ untuk melengkapi huruf-huruf dalam bahasa Arab (Ramala, 2020). Aksara jawi tidak menggunakan tanda harakat seperti fattach, kasrah, tanwin, dan dhammah dalam penulisannya.

Aksara Jawi telah digunakan sebagai bahasa komunikasi sejak zaman dahulu. Meskipun ditulis dalam aksara Arab, namun dalam pembacaannya mengikuti tata bahasa Indonesia (Afrianto dkk., 2018a). Banyak naskah kuno

yang berisi cerita-cerita sejarah, budaya dan pengetahuan, ditulis dengan menggunakan aksara Jawi. Aksara Jawi tidak hanya mengikuti tata bahasa Indonesia, tetapi juga mengikuti tata bahasa lokal setempat misalnya bahasa Aceh, terutama dalam pengucapan bahasanya. Pengucapan bahasa tersebut dikenal sebagai “Harah Jawoe” atau aksara Arab Jawoe.

Berdasarkan kajian awal diketahui bahwa, naskah kuno Aceh cukup sulit untuk dibaca karena minimnya sumber daya manusia (SDM) yang mampu melakukan kajian mendalam tentang isi naskah-naskah kuno tersebut (Hermansyah, 2015a). Sejauh ini, hanya sedikit penelitian yang benar-benar mengeksplorasi konten dari naskah-naskah ini, sehingga hanya mereka yang memiliki pengetahuan filologi yang memadai yang mampu membacanya (Mandasari, 2022). Ditambah, keberadaan tiga bahasa berbeda yang tertulis pada naskah “Harah Jawoe”, yaitu Bahasa Aceh, Bahasa Arab, dan Bahasa Melayu yang membuat naskah ini cukup sulit untuk dibaca dan dipahami oleh masyarakat pada umumnya.

Harah Jawoe dulunya digunakan oleh masyarakat Aceh untuk menulis hikayat-hikayat, yang memiliki pesan historis yang penting bagi masyarakat. Salah satu contohnya adalah hikayat yang diperkenalkan oleh Juynboll (Hermansyah, 2020b), dan hikayat Aceh yang tertulis dalam buku *Atjehsche Taal* karangan K.F.H. Van Langen (Van Langen, 1889). Buku yang disebutkan terakhir, merupakan buku rujukan utama yang akan diambil kata-kata yang tertulis didalamnya, untuk dijadikan sebagai dataset latihan, validasi dan testing dalam proses transliterasi penelitian ini.

Merujuk pada penjelasan mengenai transliterasi sebelumnya, bahwa transliterasi merupakan cara yang dapat dilakukan, agar naskah kuno yang ditulis dalam aksara Jawi seperti “Harah Jawoe” dapat dipahami oleh masyarakat

umumnya. Salah satu pendekatan transliterasi yang dapat dilakukan adalah dengan pendekatan transliterasi berbasis komputasi, karena data inputan aksara Jawi terbukti mampu dikenali dengan baik ((Yanto dkk., 2020); (Akil & Chaidir, 2021)). Akan tetapi, sebelum proses transliterasi berbasis komputasi dilakukan, diperlukan penelitian lebih mendalam mengenai seberapa besar tingkat akurasi model komputasi terhadap data masukan untuk mentransliterasikan “Harah Jawoe” ke dalam huruf latin tersebut, sehingga model yang dihasilkan benar-benar memiliki kemampuan yang baik dalam menerjemahkan “Harah Jawoe”.

Terdapat beberapa penelitian terdahulu mengenai transliterasi aksara seperti aksara Jawa ke latin, aksara Sunda ke latin, aksara Jawi ke latin dengan menggunakan beberapa metode, seperti algoritma *Freeman Chain Code* dan *Support Vector Machine* (Afrianto dkk., 2018b), *Local Binary Pattern* (Atika Sari & Shinta Sari, 2022), dan *Convolutional Neural Network* (Rahmawati dkk., 2021). Dari beberapa penelitian tersebut menunjukkan bahwa penelitian yang dilakukan dengan menggunakan metode *Convolutional Neural Network* memiliki hasil akurasi tertinggi dibandingkan dengan penelitian yang menggunakan *Support Vector Machine* dan *Local Binary Pattern*, yaitu sebesar 92%.

Didasarkan pada paparan hasil penelitian di atas, diketahui bahwa proses transliterasi berbasis komputasi sangat mungkin untuk dilakukan. Namun, yang perlu diperhatikan adalah, dua dari tiga penelitian tersebut hanya menitikberatkan pada pembacaan karakter huruf saja, belum dalam bentuk gabungan huruf (kata). Sehingga hal ini, menjadi tantangan tersendiri bagi penelitian lanjutan, karena umumnya tulisan aksara jawi memiliki beragam bentuk kata baik dibagian awal, tengah, dan akhir kalimat (Afrianto dkk., 2018c). Selain itu, proses pembuatan anotasi dari setiap kata naskah kuno yang tertulis dalam buku *Atjehsche Taal*, juga perlu menjadi perhatian serius, karena sangat menentukan hasil akurasi transliterasi.

Menariknya adalah, dalam penelitian yang menggunakan metode *Convolutional Neural Network* terdapat berbagai jenis arsitektur yang dapat digunakan dalam membaca masukan gambar dalam bentuk anotasi (Sanjong Tamang, 2023), seperti *You Only Look Once* atau YOLO (Sani & Rahmadinni, 2022a), *MobileNet* (Ellyadi, 2022), *Baseline CNN* (Sabri, 2022), dan VGG19 (Marcella & Devella, 2022). Sehingga dapat digunakan dalam melakukan anotasi gambar masukan, berupa gabungan huruf “Harah Jawoe”. Menurut empat kajian tersebut, arsitektur YOLO berhasil membaca masukan gambar dengan tingkat akurasi tertinggi.

Arsitektur YOLO terdiri dari beberapa varian yang terus mengalami perubahan, mulai dari YOLOv1 sampai YOLOv8. YOLOv8 dipilih untuk digunakan dalam penelitian ini, dikarenakan YOLOv8 merupakan varian arsitektur paling terbaru dari arsitektur YOLO. Sehingga dalam penelitian ini, akan menggunakan metode *Convolutional Neural Network* dengan arsitektur YOLOv8 dalam melakukan transliterasi aksara “Harah Jawoe” ke latin.

B. Rumusan Masalah

Naskah kuno Aceh, merupakan salah satu naskah kuno yang ditulis dalam format aksara Arab Melayu (Jawi), yang dibaurkan dengan dengan bahasa lokal setempat, yang lebih dikenal dengan sebutan “Harah Jawoe”. Menurut beberapa catatan, diketahui bahwa naskah kuno Aceh cukup sulit untuk dibaca dan dipahami oleh masyarakat secara luas. Hal ini disebabkan karena terbatasnya sumber daya manusia (SDM) yang memahami bacaan “Harah Jawoe” tersebut. Disamping itu, “Harah Jawoe” juga memiliki tiga kandungan bahasa yang berbeda, yakni Bahasa Arab, Bahasa Aceh dan Bahasa Melayu, yang menjadikan tingkat kesulitan pembacaan hurufnya semakin sulit. Transliterasi berbasis komputasi merupakan salah satu solusi dalam mengatasi persoalan tersebut.

Namun penting untuk menjadi perhatian, sebelum melakukan proses transliterasi naskah kuno “Harah Jawoe”, perlu dilakukan pengambilan anotasi yang baik terhadap dataset gambar kata “Harah Jawoe”, yang sumber utamanya dari buku *Atjehsche Taal*. Selanjutnya, juga menjadi tantangan tersendiri, bagaimana tingkat akurasi dari dataset gambar kata “Harah Jawoe” yang telah dianotasikan tersebut. Menurut beberapa kajian, arsitektur YOLO merupakan arsitektur yang memiliki kemampuan pembacaan anotasi yang baik, sehingga dimungkinkan untuk meningkatkan akurasi terhadap model pendeteksian kata “Harah Jawoe” yang dibangun dalam penelitian ini. Oleh sebab itu, perlu kajian lebih lanjut, untuk menjawab pertanyaan penelitian berikut:

1. Bagaimana proses anotasi “Harah Jawoe” ke dalam bahasa latin?
2. Bagaimana tingkat akurasi transliterasi “Harah Jawoe” ke dalam bahasa latin menggunakan arsitektur YOLOv8?

C. Tujuan Penelitian

Adapun tujuan dari penelitian ini yaitu sebagai berikut:

1. Untuk mengetahui proses anotasi “Harah Jawoe” ke dalam bahasa latin.
2. Untuk mengetahui tingkat akurasi transliterasi “Harah Jawoe” ke dalam bahasa latin menggunakan arsitektur YOLOv8.

D. Batasan Masalah

Batasan-batasan masalah yang diambil oleh peneliti dalam penelitian ini adalah:

1. Data yang digunakan sebagai dataset dalam penelitian ini adalah potongan gambar kata “Harah Jawoe” yang diambil dari buku yang

berjudul *Atjehsche Taal* yang ditulis oleh K.F.H. Van Langen Tahun 1889.

2. Jumlah dataset yang digunakan yaitu 3000 gambar kata “Harah Jawoe” yang terdiri dari hasil potongan gambar kata dari buku yang berjudul *Atjehsche Taal*.

E. Manfaat Penelitian

Manfaat dari penelitian yang dilakukan ini adalah:

1. Agar diperoleh dataset dan model yang memiliki anotasi dan tingkat akurasi yang baik, terkait naskah kuno “Harah Jawoe”.
2. Agar hasil dari penelitian ini bisa digunakan sebagai acuan untuk penelitian selanjutnya mengenai transliterasi aksara kuno lainnya, khususnya “Harah Jawoe” yang menggunakan arsitektur YOLOv8.

BAB II

LANDASAN TEORI

A. Penelitian Terdahulu

Sebagian besar penelitian yang terkait dengan pendeteksian gambar, menitikberatkan pada tingkat akurasi, waktu pemrosesan dan proses *tune* dengan meningkatkan jumlah *epoch*, yang bertujuan untuk meningkatkan keterbacaan arsitektur yang digunakan terhadap dataset ((Khairunnas dkk., 2021a); (Tamang dkk., 2023)). Jumlah dataset yang digunakan, ukuran gambar dan alat anotasi gambar memainkan peranan penting dalam meningkatkan akurasi pendeteksian dataset oleh arsitektur yang digunakan dalam berbagai kajian pendeteksian gambar ((Rukundo, 2021); (Orlando dkk., 2022); (Rebinth dkk., 2019)).

Convolutional Neural Network (CNN) merupakan salah satu bagian dari metode *deep learning* yang biasa digunakan dalam teknologi *computer vision* ((Adi Nugroho dkk., 2020)). Berdasarkan kajian yang telah dilakukan, CNN memiliki tingkat akurasi yang baik dalam melakukan pendeteksian obyek gambar, dengan tingkat akurasi sebesar 91% setelah *epoch* terakhir diproses, dengan jumlah dataset *training* dan *testing* sebanyak 13.440 dan 3.360 (Akil & Chaidir, 2021). Hasil tersebut berselisih 4,5% dari hasil akurasi CNN penelitian yang dilakukan oleh Rachmawanto dan Andono (2022) yang memperoleh tingkat akurasi sebesar 86,5%, dan nilai ini didapatkan setelah dilakukan peningkatan *epoch* sebanyak 200 kali, dengan jumlah dataset 1000 data (Rachmawanto & Andono, 2022).

Kajian penelitian CNN yang lain menunjukkan bahwa, tingkat akurasi juga dipengaruhi oleh banyaknya data training yang digunakan. Hal ini sejalan dengan hasil penelitian yang dilakukan oleh Rahmawati (2021) yang mendapatkan akurasi pendeteksian aksara sunda sampai dengan 92%

(Rahmawati dkk., 2021). Tetapi, yang menarik adalah metode CNN juga sangat dimungkinkan untuk digunakan pada dataset dan jumlah epoch yang tidak terlalu banyak. Seperti kajian yang dilakukan oleh Wulan Anggraini (2020) yang hanya mengoprasionalkan 250 citra latih dan 50 citra uji, dengan hasil akurasi masing-masing yakni sebesar 92% dan 87% (Anggraini, 2020). Kajian lain yang dilakukan oleh Sukma Hanindria dan Hendry (2022) juga membuktikan bahwa, dengan jumlah epoch 50 kali, dapat diperoleh tingkat akurasi cukup baik yakni sebesar 85% (Sukma Hanindria & Hendry, 2022).

Sebagai bagian dari metode CNN, arsitektur YOLO sangat mungkin untuk diaplikasi dalam pendeteksian citra (Khairunnas dkk., 2021b). Beberapa kajian membuktikan bahwa, penggunaan arsitektur YOLO berkontribusi sangat baik dalam mendeteksi citra maupun video. Seperti kajian yang dilakukan oleh Muhammad Nur dan Arnisa Stefanie (2023) yang berhasil mendeteksi citra fundus mata dengan tingkat akurasi pembacaan dataset oleh arsitektur yolo sebesar 92% (Nur dkk., 2023). Begitu juga dengan kajian yang dilakukan oleh Bernard Alaric Sudjono dkk (2023) yang mendapatkan tingkat akurasi sebesar 73% dalam mendeteksi real time video (Sudjono dkk., 2023a).

Sementara itu, arsitektur Xception yang berfokus pada 475 kelas kata-kata Jawoe juga bekerja cukup baik. Hasil analisis menunjukkan bahwa model ini mencapai tingkat akurasi sebesar 67,60% setelah melalui 90 epoch. Hasil tersebut menunjukkan bahwa arsitektur Xception membutuhkan variasi data yang lebih luas untuk mencapai tingkat keakuratan yang lebih tinggi (Aesha Durratul Nasihah, 2023). Bagian yang menarik adalah, akurasi pendeteksian citra menggunakan arsitektur YOLO sangat mungkin ditingkatkan, dengan melakukan manipulasi pada *background* citra pada dataset. Hal ini didukung oleh kajian yang dilakukan oleh Abdullah Sani dan Suci Rahmadinni (2022), yang melatari *background* citra dengan tiga warna yakni: biru, putih dan kuning,

dengan hasil akurasi berturut 97% untuk *background* putih, 98% untuk *background* biru dan 98% untuk *background* kuning (Sani & Rahmadinni, 2022b).

Tabel 2.1 Dampak Arsitektur CNN dan YOLO terhadap Tingkat Akurasi

Peneliti	Arsitektur	Kajian	Jumlah Dataset	Akurasi
Pulung Nurtatio Andono & Eko Hari Rachmawanto (2022)	CNN	Penggunaan arsitektur ditujukan untuk deteksi karakter tulisan tangan Hiragana. Proses segmentasi dilakukan dengan metode thresholding, kemudian dilakukan penghilangan noise, menukar ukuran citra gambar, dan melakukan proses normalisasi	1000 citra gambar tulisan tangan, dengan pembagian data latih dan uji sebanyak 2 kali yakni 70:30 dan 60:40	Proses latih pertama diperoleh akurasi 86,5%, dan proses data latih kedua diperoleh 83%
Sukma Hanindria & Hendry, 2022	CNN	Arsitektur ditujukan untuk mengklasifikasikan Aksara Jawa. Proses preprocessing dilakukan dengan mengubah ukuran citra, dengan ukuran gambar 150*150 pixels. Aktivasi yang digunakan ReLu	2580 citra aksara Jawa, terdiri dari 20 kelas data. Setiap kelas data memiliki 108 citra aksara jawa	Tingkat akurasi yang didapatkan adalah 85%
Rahmawati dkk., 2021	CNN	Arsitektur digunakan untuk mendeteksi Aksara Sunda.	847 data aksara Sunda	Menghasilkan 96.71% dengan data training dan 92.02%

Peneliti	Arsitektur	Kajian	Jumlah Dataset	Akurasi
		Aktivasi yang digunakan ADAM, dengan nilai learning rate 0.1. Epoch dilakukan empat kali, yakni 20, 50, 100 dan 500		dengan data testing
Anggraini, 2020)	CNN	Arsitektur digunakan untuk deteksi wajah berhijab dengan pengambilan berbagai ekspresi wajah. Epoch dilakukan lima kali, dengan jumlah untuk setiap tahapan yakni: 20, 30, 50, 100 dan 120	300 citra berhijab dengan beragam ekspresi wajah	nilai akurasi sebesar 92% pada data training dan 87% pada data testing
Muhammad Nur & Arnisa Stefanie, 2023	YOLO	Arsitektur digunakan untuk proses klasifikasi Penyakit Mata Berdasarkan Citra Fundus. Epoch dilakukan sebanyak 100 kali, dengan sebelumnya mengubah ukuran citra menjadi 224*224	Terdiri dari 4 kelas yaitu mata normal, katarak, glaukoma, dan <i>diabetic retinopathy</i>	92% untuk akurasi data, 91 untuk nilai <i>precision</i> , 92% untuk nilai <i>recall</i> dan 91% untuk nilai <i>F1-score</i>
Bernard Alaric Sudjono, dkk (2023),	YOLO	Arsitektur digunakan untuk Object Detection secara real time pada citra bergerak ayam jenis broiler pada kandang <i>closed</i>	68 citra data latih, yang diambil langsung secara mandiri pada kandang <i>closed house</i> .	menghasilkan nilai akurasi sebesar 73% pada video bergerak ayam broiler yang berada pada

Peneliti	Arsitektur	Kajian	Jumlah Dataset	Akurasi
		<i>house</i> . Epoch dilakukan sebanyak 100 kali.		kadang <i>Closed house</i>
Aesha Durratul Nasihah	Xception	Menggunakan 445 <i>class</i> kata <i>Harah Jawoe</i> dengan 90 epoch	300 Dataset, 2500 data latih dan 500 data uji	Nilai akurasi 67,60% pada epoch ke 90
Abdullah Sani & Suci Rahmadinni (2022)	YOLO	Arsitektur digunakan untuk mendeteksi gestur tangan. Dilakukan beragam manipulasi gestur tangan, baik dengan mengubah <i>backhground</i> citra, memperhatikan intensitas cahaya, serta uji coba jarak yang berbeda dalam menangkap gestur tangan.	1500 dataset, terdiri 150 data dari setiap <i>gesture</i> dengan menggunakan 3 layar <i>background</i>	Nilai keberhasilan deteksi pada <i>background</i> putih yaitu sebesar 97%, <i>background</i> biru sebesar 98% dan <i>background</i> kuning sebesar 98%.

B. Teori-Teori yang Relevan

B1. Bahasa Arab-Melayu

Bahasa Arab Melayu atau aksara Jawi adalah salah bentuk Bahasa yang digunakan oleh bangsa Melayu dengan memadukan seluruh karakter aksara Arab dengan menambahkan aksara lain yang tidak termasuk dalam huruf Arab seperti huruf چ، غ، ف، ك، ن، و untuk melengkapi huruf dalam bahasa Arab (Ramala, 2020). Huruf tambahan ini dibuat dengan menambahkan tanda baca tambahan (diakritik) pada huruf untuk mengubah nilai dan membedakan bunyinya. Dalam pembacaannya menggunakan tata bahasa Indonesia. Bahasa Arab-Melayu ini

telah berfungsi sebagai bahasa perantara dalam urusan-urusan pemerintahan yang digunakan sejak zaman dahulu. Bahasa Arab-Melayu juga menggunakan tata bahasa Aceh dalam pengucapannya. Ini dikenal dengan istilah *Harah Jawoe* atau “Harah Jawoe” (Tengku & Sakti, 2011).

Harah Jawoe atau “Harah Jawoe” merupakan bahasa yang menggunakan tiga bahasa yaitu bahasa Aceh, bahasa Arab dan Bahasa Melayu. “Harah Jawoe” banyak ditemukan pada naskah dan hikayat-hikayat yang ditulis para cendikiawan Aceh. Hikayat Aceh merupakan salah satu jenis karya sastra Aceh yang umumnya berbentuk puisi, syair dan prosa yang berisikan berbagai bidang ilmu dan corak meliputi dogeng, nasehat, cerita Sejarah, kisah dan lain sebagainya (Hermansyah, 2020c). Hikayat dan naskah Aceh ini telah tersimpan diberbagai museum, Lembaga swasta dan koleksi personal (Mandasari, 2022b).

B2. Deep Learning

Deep Learning merupakan bagian bidang yang berkaitan dengan *Machine Learning* yang terdiri dari algoritma pemodelan abstraksi tingkat tinggi pada data dengan menggunakan sekumpulan fungsi transformasi non-linear yang ditata berlapis-lapis dan mendalam (Lorentius dkk., 2019). *Deep learning* menggunakan metadata yang digunakan sebagai inputan dan untuk mengolahnya digunakan sejumlah lapisan yang tersembunyi dari transformasi non-linear yang dimiliki inputan agar dapat menghitung hasil nilai dari *output*.

Pada *deep learning*, mesin akan mempelajari berbagai macam model dan melakukan klasifikasi terhadap berbagai macam bentuk data untuk menghasilkan informasi dan hasil prediksi yang memiliki nilai akurasi yang tinggi. *Deep Learning* juga mampu melakukan ekstraksi fitur citra secara otomatis (Hidayatullah, 2021a). Disamping itu, perlu menjadi perhatian bahwa hingga saat

ini belum ada panduan yang benar-benar baku mengenai berapa jumlah *hidden layer* terbaik untuk sebuah arsitektur *deep learning* (Hidayatullah, 2021b).

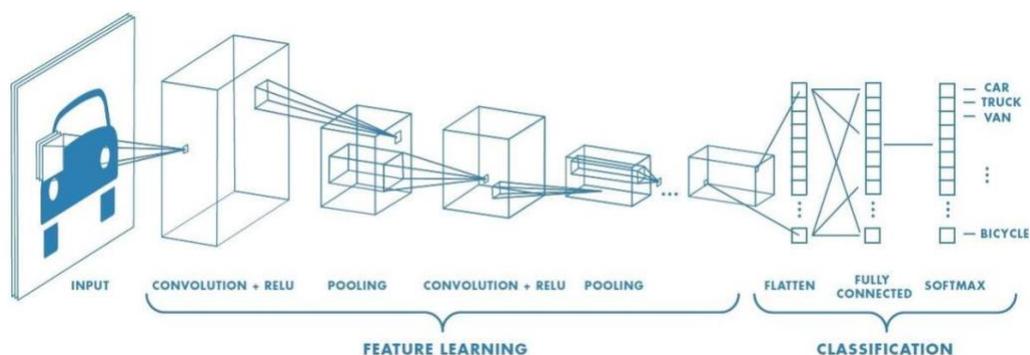
B3. Object Detection

Object detection adalah suatu teknik pada *computer* yang digunakan untuk menemukan contoh objek dalam gambar atau video (Nufus dkk., 2021). Mesin akan melakukan *scanning* atau pemindaian terhadap gambar untuk menentukan objek mana yang ada di dalam gambar dan mana yang bukan, berdasarkan nilai yang telah dilatih sebelumnya. Dengan *object detection*, mesin juga dapat mengetahui posisi dari objek-objek yang ada pada gambar yang diinputkan.

Berdasarkan pada buku Priyanto Hidayatullah (2021), *object detection* merupakan gabungan dari *classification* dan *localization*. Tujuan utama sebuah *object detection* adalah, untuk memperkirakan posisi suatu objek dengan kotak pembatas (*bounding box*) objek yang telah di anotasi sebelumnya, selanjutnya melakukan klasifikasi terhadap setiap objek yang ada pada setiap *bounding box* (Hidayatullah, 2021c).

B4. Convolutional Neural Network

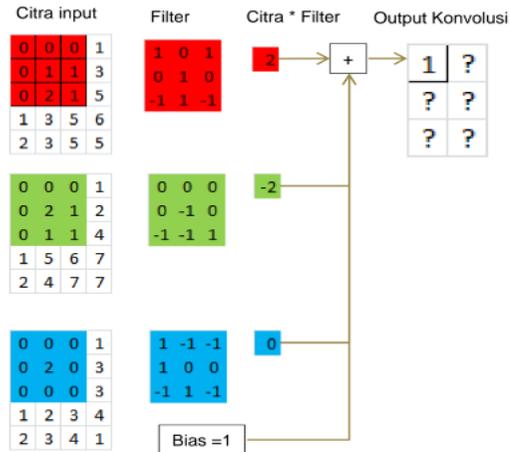
Convolutional Neural Network (CNN) adalah salah satu metode pembelajaran dari *deep learning* yang digunakan untuk mendeteksi dan mengenali data dari pemrosesan gambar (Lina, 2019). CNN terdiri dari lapisan-lapisan *neuron* yang disusun untuk membentuk dengan *filter* atau piksel yang panjang dan tinggi. CNN menggunakan proses *convolution* untuk mengekstrak dan mendapatkan informasi dari gambar. Informasi dari gambar akan dihubungkan ke dalam *neuron*.



Gambar 2.1 Arsitektur CNN (Anhar & Putra, 2023)

Sebuah arsitektur CNN terdiri dari beberapa lapisan yang memiliki tugas dan fungsinya masing-masing. Secara umum, arsitektur CNN terdiri dari dua bagian, yaitu *Feature Learning* dan *Classification*. *Feature learning* merupakan proses *encoding*, dimana citra gambar yang akan dikodekan menjadi fitur-fitur yang terdiri dari angka-angka yang mendefinisikan citra tersebut. *Feature Learning* terdiri dari dua bagian yaitu *Convolution layer* dan *Pooling layer*.

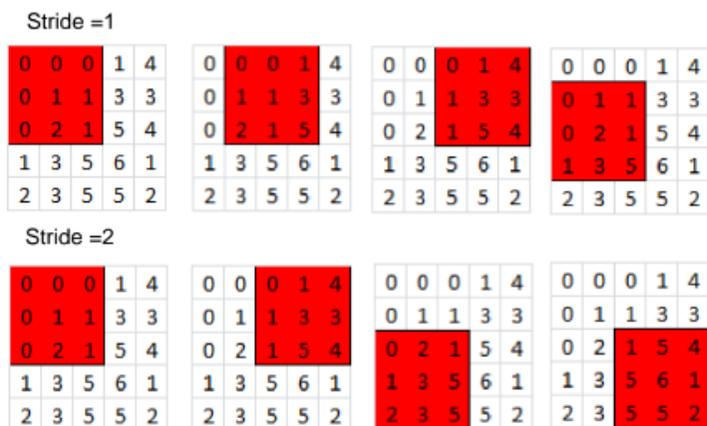
Convolution Layer adalah lapisan pertama yang menerima inputan citra gambar dan memperoleh informasi mengenai posisi dan nilai hasil *convolution* pada lapisan. *Convolution* merupakan teknik perkalian citra dengan menggunakan *filter*. Fungsi *filter* adalah untuk mengekstraks fitur yang digunakan untuk mengetahui keberadaan fitur tersebut. Pada dasarnya, *filter* adalah *array* yang terdiri dari dua dimensi yang memiliki ukuran yang lebih kecil dari gambar inputan. Proses *Convolution* dilakukan dengan cara menjumlahkan hasil perkalian antara citra gambar dengan *filter* pada setiap pergeseran yang dilakukan pada bagian gambar, Perkalian ini akan menghasilkan sebuah hasil *output* yang disebut dengan *feature map* atau *activation map*. Misalnya, pada lapisan pertama dari lapisan ekstraksi ciri yaitu layer *convolution* dengan ukuran 3x3x3. Dimana panjangnya 3 px (piksel), tingginya 3 px dan jumlah/tebal 3 buah.



Gambar 2.2 Proses Konvolusi (Hidayatullah, 2021d)

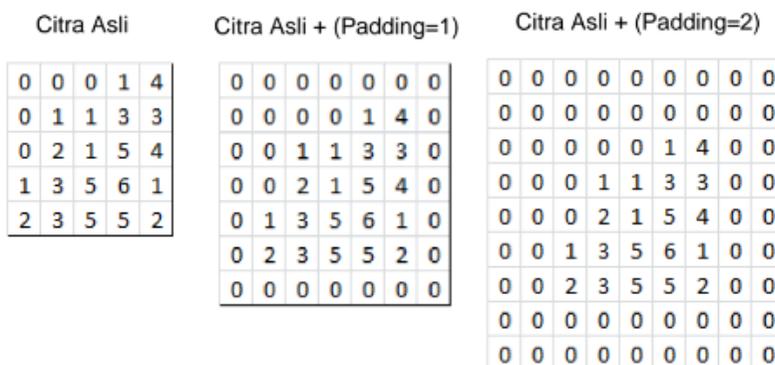
Pada gambar diatas, citra inputan yang memiliki tiga kanal dengan ukuran 3x3 piksel difilter dengan menggunakan *output* satu kanal. Dimana hasil perkalian antara citra *input* pada kanal pertama menghasilkan nilai 2, pada kanal kedua menghasilkan nilai -2 dan pada kanal ketiga menghasilkan nilai 0.

Stride adalah bagian dari lapisan *Convolution* yang merupakan parameter yang digunakan untuk menentukan banyaknya jumlah pergeseran yang dilakukan oleh *filter*. Jika nilai yang dimiliki oleh *stride* adalah 1 maka, *filter* akan bergeser secara *horizontal* dan secara *vertical* sebanyak satu kali untuk setiap piksel. Semakin besar nilai yang dimiliki dari *stride* maka informasi yang didapatkan juga semakin kecil dari sebuah citra inputan.



Gambar 2.3 Ilustrasi proses Stride (Hidayatullah, 2021e)

Padding atau dikenal dengan *zero padding* adalah bagian dari lapisan *Convolution* yang merupakan parameter yang digunakan untuk menentukan jumlah piksel yang akan ditambahkan pada setiap sisi pada inputan gambar. Biasanya, *padding* diterapkan dengan menambahkan nilai 0, yang digunakan untuk dapat mengubah dimensi *output* pada *convolution layer*.



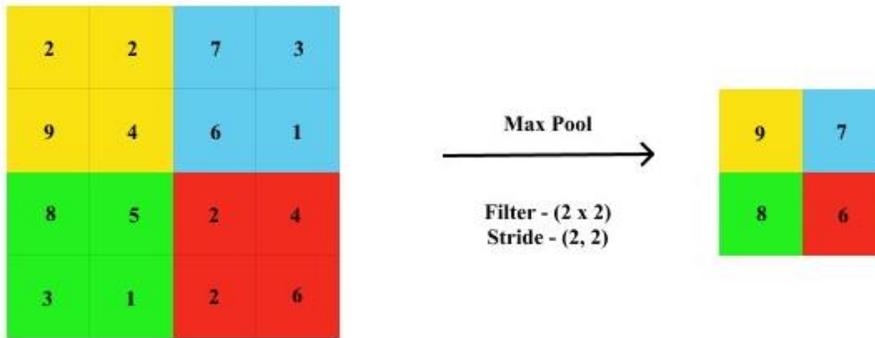
Gambar 2.4 Proses Penambahan Padding (Hidayatullah, 2021f)

Menurut (Sena, 2017), Tujuan penggunaan *Padding* pada *Convolutional Neural Network* yaitu:

1. Dimensi dari hasil output pada *layer Convolution* selalu lebih kecil dari dimensi inputan, kecuali pada penggunaan filter berukuran 1x1 dengan ukura *stride* 1. Sehingga hasil output ini akan digunakan Kembali sebagai inputan dari proses *layer convolution* selanjutnya. Hal ini menyebabkan semakin banyak informasi yang terbuang. Dengan menggunakan dan mengatur ukuran *padding*, dimensi dari hasil *output* akan tetap sama dengan dimensi inputan atau bisa jadi tidak berkurang secara drastis.
2. Dapat meningkatkan kinerja dari model karena filter akan berfokus pada informasi yang sebenarnya yaitu berada diantara *zero padding*,

ReLU (*Rectified Linear Unit*) berfungsi sebagai proses atau langkah-langkah untuk menghilangkan *vanishing gradient* menggunakan aktivasi elemen $f(x) = \max(0, X)$ (Rachmawanto & Andono, 2022a). Dimana hasil *output* dari proses *convolution* yang memiliki nilai negatif akan diubah atau diaktivasi kedalam nilai positif berupa nilai 0.

Lapisan kedua adalah *Pooling Layer*. Lapisan *Pooling* berada setelah lapisan pertama yaitu lapisan *convolution*. *Pooling* terdiri dari sebuah filter yang memiliki ukuran dan nilai *stride* tertentu yang melakukan permindahan dengan mengeser pada seluruh area *feature map* yang bertujuan untuk mengurangi dimensi dari *feature map* sehingga dapat mempercepat komputasi dengan menambahkan lebih sedikit parameter yang harus diupdate semakin sedikit dan mengatasi *overfitting*. *Pooling* yang paling umum digunakan adalah *Max Pooling* dan *Average Pooling*.



Gambar 2.5 Ilustrasi Proses Pooling Layer (Hidayatullah, 2021g)

Pada Gambar 2.5 diatas, dapat dijabarkan bahwa pada proses *layer pooling* menggunakan *Max pooling* pada hasil *feature map* yaitu dengan melihat nilai maksimum yang dimiliki pada setiap bloknnya. Pada blok pertama yang berwarna kuning, nilai maksimum yang dimiliki adalah 9. Pada blok kedua yang berwarna biru, nilai maksimum yang dimilikinya adalah 7. Pada blok ketiga yang berwarna hijau, memiliki nilai maksimum adalah 8. Dan pada blok terakhir yang berwarna merah, memiliki nilai maksimum yaitu 6. Sehingga diperoleh *feature map* yang memiliki array berdimensi 2 x 2.

Pada bagian kedua yaitu bagian *Classification*. Pada *classification* berfungsi untuk membuat klasifikasi pada setiap *neuron* yang telah melewati proses ekstrasi pada tahap sebelumnya. Pada bagian *Fully Connected Layer*, *output feature map* yang dihasilkan pada bagian pertama yaitu bagian ekstraksi fitur masih berupa *array multidimensi*, sehingga perlu untuk dilakukan "*flutten*" atau *reshape feature map* menjadi sebuah vektor yang dapat digunakan sebagai inputan dari lapisan *fully connected*. Hal ini dilakukan supaya nilai tersebut dapat digunakan sebagai inputan pada *layer fully connected layer*. Pada lapisan ini, semua aktivitas *neuron* dari lapisan sebelumnya saling terhubung dengan *neuron* pada lapisan berikutnya. Dengan demikian, setiap aktivitas pada lapisan sebelumnya harus di ubah menjadi

data dari suatu dimensi yang terhubung pada semua *neuron* di lapisan *fully connected*. *Layer fully connected* bertujuan untuk melakukan transformasi di dimensi data agar data yang diperoleh pada *feature maps* dapat dilakukan pengklasifikasian secara linear.

Menurut Pulung, Eko (2022) Softmax memiliki fungsi sebagai metode klasifikasi yang menggunakan beberapa jumlah kelas yang lebih dari satu. *Softmax* merupakan sebuah fungsi dan dapat digunakan untuk mengubah dimensi vektor berupa sebuah nilai yang berbentuk real kedalam nilai yang berbentuk vektor. Langkah ini digunakan untuk mencari nilai dari probabilitas terhadap target yang akan digunakan sebagai kelas target inputan (Rachmawanto & Andono, 2022b).

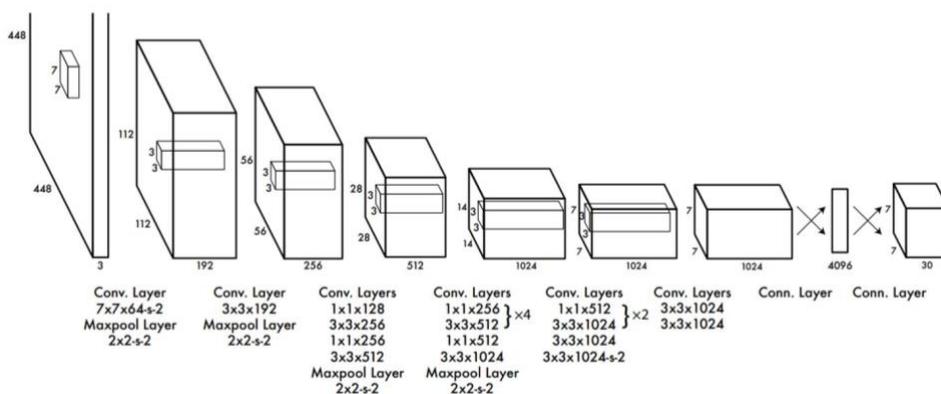
B5. You Only Look Once (YOLO)

YOLO adalah arsitektur CNN yang diperkenalkan oleh Redmon, dkk. Yang mampu mendeteksi objek secara *real-time*. YOLO memiliki kecepatan pendeteksian objek yang lebih cepat dibandingkan dengan metode deep learning lainnya. Metode ini memiliki performa hingga 45 FPS dalam mendeteksi objek secara *real-time*. YOLO menggunakan *Convolutional Neural Network* dalam mendeteksi objek. Arsitektur YOLO menerapkan *single neural network* pada seluruh citra (Sani & Rahmadinni, 2022b).

Fitur yang berada pada arsitektur YOLO akan melalui 24 konvolusi, 4 max pooling dan melewati 2 *fully connected* untuk mendapatkan dan menghasilkan grid yang mengandung nilai untuk dapat diklasifikasi dan diregresi. Pada YOLO, fungsi *activation* yang digunakan pada lapisan terakhir yaitu *linear activation function*, sedangkan pada layer lainnya menggunakan *leaky ReLU*. Setelah mendapatkan *grid-grid* yang dihasilkan dari konvolusi, *classification* akan dilakukan diawali dengan melakukan perkalian antara *confidence score* tiap *bounding box* dengan semua *class score* pada *grid*-nya sehingga mendapatkan $S \times S$

x 2 tensor. Kemudian mengubah nilai *class* dari setiap *grid* yang lebih rendah dari nilai *threshold* yang telah ditentukan menjadi 0.

Pada YOLO, terdapat *Non-Maximum Suppression* (NMS) yang digunakan untuk menghilangkan beberapa *bounding box* atau kotak pembatas yang mengidentifikasi posisi dan tipe dari objek pada gambar dengan prediksi objek yang sama, dan menyisakan satu *bounding box* (Daniels, 2020). NMS akan dilakukan untuk mencari *redundant boxes* (2 atau lebih *bounding box* yang mengandung 1 objek yang sama) dan kemudian *score* untuk *class* tersebut akan diubah menjadi 0. Dan nilai *class* tertinggi akan menjadi *class* pada *bounding box* tersebut (Shianto dkk., 2019).



Gambar 2.6 Arsitektur YOLO (Nur dkk., 2023)

C. Metode Evaluasi

Pada penelitian ini, peneliti menggunakan metode *Evaluation Measurement* atau pengukuran evaluasi. Metode evaluasi yang digunakan yaitu *recall* dan *precision* dan *accuracy*.

C1. Recall dan Precision

Melakukan pengukuran evaluasi dengan menggunakan nilai *recall* (R) dan *precision* (P) bertujuan untuk mengetahui tingkat kesesuaian prediksi yang dilakukan dengan melihat ketepatan model dalam melakukan pencarian dan pengenalan informasi yang diminta oleh pengguna model. Prediksi ini dilakukan terhadap kelompok yang bernilai *positive*.

Nilai *recall* adalah nilai yang digunakan untuk menunjukkan tingkat perolehan hasil yang dikembalikan dari sebuah *system* yang diperoleh dengan membandingkan jumlah item relevan yang dikembalikan dari *system* dengan menggunakan total jumlah item yang ada. Pada nilai *recall*, semakin besar nilai yang dihasilkan maka nilai tersebut tidak dapat menunjukkan bahwa suatu *system* itu baik atau tidak. Secara umum, nilai tertinggi dari nilai *recall* adalah 1. Nilai *precision* bertujuan untuk menunjukkan tingkat ketepatan sebuah *system* untuk mengembalikan informasi relevan kepada pengguna.

Dalam evaluasi pengukuran model algoritma, digunakan acuan pada *confusion matrix* yang memberikan informasi berupa perbandingan dari hasil klasifikasi yang dilakukan oleh model dengan hasil klasifikasi sebenarnya (Kurniawan dkk., 2022b) Dimana dalam dengan menggunakan *confusion matrix* dapat mentukan hasil *accuracy*, *recall*, *f1 score* dan *precision*. *Confusion matrix* terdiri dari dua buah kolom yaitu kolom, yaitu kolom aktual (kondisi sebenarnya) yang bernilai *positive* (1) dan kolom aktual yang bernilai *negative* (0) dan terdiri dari dua baris, yaitu baris prediksi yang bernilai *positive* (1) dan baris prediksi yang bernilai *negative* (0).

	Positif 1	Negatif 0
Positif 1	TP	FP
Negatif 0	FN	TN

Gambar 2.7 *Confusion Matrix* (Hidayatullah, 2021h)

Pada Gambar 2.7, TP atau *True Positive* menggambarkan jumlah data yang di prediksi bernilai benar atau *positive* yang berada pada kelompok yang bernilai *positive*. FP atau *False Positive* menggambarkan jumlah data yang di prediksi bernilai salah atau *negative* yang berada pada kelompok yang bernilai *positive*. FN atau *False Negative* menggambarkan jumlah data yang di prediksi bernilai salah yang berada pada kelompok yang bernilai *negative*. TN atau *True Negative* menggambarkan jumlah data yang di prediksi bernilai benar yang berada pada kelompok *negative*.

Berdasarkan pada acuan *confusion matrix* diatas, untuk menentukan nilai *recall* (R) dan *precision* (P) pada kelompok *positive* dapat menggunakan formula sebagai berikut (Hidayatullah, 2021i):

$$R = \frac{TP}{TP+FN} \dots\dots\dots (1)$$

$$P = \frac{TP}{TP+FP} \dots\dots\dots (2)$$

Dari formula diatas, dapat disimpulkan bahwa nilai *recall* (R) adalah nilai yang diperoleh dari jumlah data prediksi yang bernilai benar pada kelompok

positive (TP) dibagi dengan jumlah nilai yang diperoleh dari data prediksi TP ditambah FN atau jumlah data yang berada pada kelompok *positive* yang sebenarnya. Sedangkan nilai *precision* (P) adalah nilai yang diperoleh dari jumlah data prediksi yang bernilai benar pada kelompok *positive* (TP) dibagi dengan jumlah nilai yang diperoleh dari data TP ditambah FP atau jumlah data sebenarnya yang berada pada kelompok *positive*.

C2. F1-Score

F1-Score merupakan *Harmonic Mean* atau perbandingan rata-rata antara *precision* dengan *recall*. Nilai F1 score berada antara nilai 0 dan 1. Nilai 1 adalah nilai terbaik, dan nilai 0 adalah nilai terburuk. Nilai F1 score yang baik menjelaskan bahwa model memiliki nilai *precision* dan *recall* yang baik. Persamaan pengukuran nilai F1 Score adalah sebagai berikut (Harun & Brilian Kharisma, 2023):

$$F1\ Score = 2 \times \frac{Precision * Recall}{Precision + Recall} \dots \dots \dots (3)$$

C3. Accuracy

Accuracy adalah pengukuran untuk melihat seberapa sering algoritma klasifikasi berhasil dalam membuat prediksi yang memiliki nilai benar. Dimana untuk menghasilkan nilai *accuracy* dengan melihat rasio jumlah total prediksi yang bernilai benar dibagi dengan keseluruhan total dari prediksi (Kurniawan dkk., 2022). Persamaan pengukuran nilai akurasi adalah sebagai berikut:

$$Accuracy = \frac{TP+TN}{TP+FP+FN+TN} \dots \dots \dots (4)$$

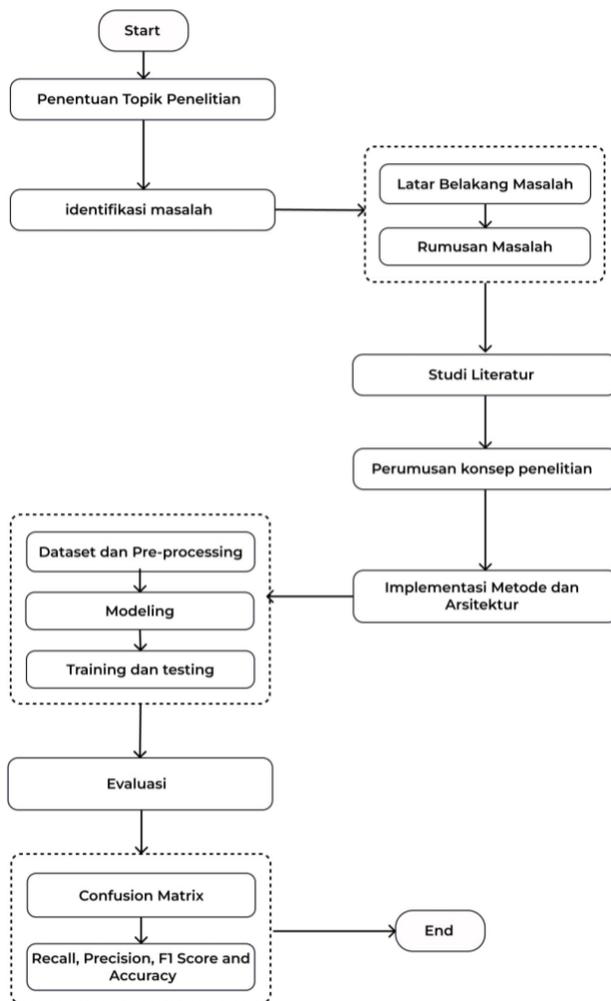
Keterangan:

- Accuracy* : Total akurat model dalam mengenali dataset
- TP : Jumlah data benar diprediksi oleh model

- TN : jumlah data *negative* yang benar diprediksi oleh model
 FP : jumlah data *negative* yang diprediksi sebagai data *positive*
 FN : jumlah data *positive* yang diprediksi sebagai data *negative*

D. Kerangka Berpikir Penelitian

Gambar 2.8 berikut, merupakan alur kerangka penelitian yang akan dilaksanakan dalam penelitian ini.



Gambar 2.8 Kerangka Berpikir Penelitian

BAB III

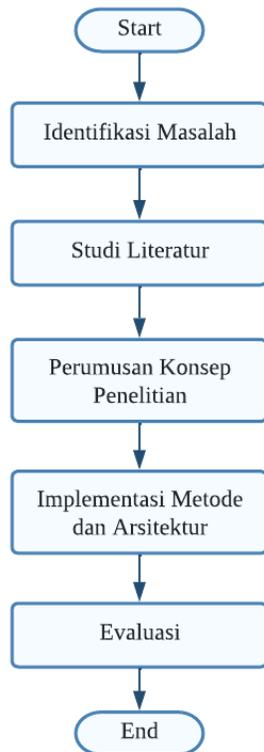
METODE PENELITIAN

A. Jenis Penelitian

Penelitian ini menggunakan pendekatan metode penelitian kuantitatif yaitu penelitian yang menganalisis data angka untuk menemukan jawaban atas pertanyaan penelitian yang berkaitan dengan fenomena yang akan diteliti. Penelitian kuantitatif memiliki jawaban yang pasti. Dalam penelitian ini, masalah yang diangkat adalah transliterasi “Harah Jawoe” ke dalam bahasa latin menggunakan arsitektur YOLOv8.

B. Tahapan Penelitian

Langkah-langkah yang akan di lakukan dalam menyelesaikan penelitian ini dapat dilihat pada gambar tahapan-tahapan penelitian berikut ini:



Gambar 3.1 Tahapan Penelitian

C. Metode Pengumpulan Data

Peneliti menggunakan metode pengumpulan data yang terdiri dari identifikasi masalah dan studi literatur dalam penelitian ini. Untuk mengidentifikasi masalah, peneliti melakukan observasi. Sementara untuk studi literatur, peneliti melibatkan perbandingan pada penelitian-penelitian sebelumnya yang dapat diadopsi dalam penelitian ini.

C1. Identifikasi Masalah

Tahapan ini dilakukan untuk menemukan topik penelitian yang sesuai dengan bidang ilmu yang dipelajari. Peneliti mengambil topik penelitian yang

berkaitan dengan penggunaan *deep learning* dengan metode *Convolutional Neural Network*. Peneliti menemukan masalah mengenai kesulitan dalam memahami “Harah Jawoe” dikarenakan kurangnya penelitian yang membahas mengenai isi dari naskah kuno diikuti dengan menggunakan tiga bahasa berbeda didalamnya. Dalam menyelesaikan permasalahan ini, peneliti menggunakan metode CNN dengan arsitektur YOLOv8 diikuti dengan penggunaan *framework Tensorflow* untuk proses transliterasi.

C2. Studi Literatur

Pada tahapan ini, peneliti menggunakan metode studi literatur untuk mengidentifikasi berbagai tindakan yang akan dilakukan terkait dengan pengumpulan data dan memberikan gambaran mengenai masalah yang ada berkaitan dengan transliterasi “Harah Jawoe” yang bersumber dari jurnal atau artikel, serta buku dan bahan pendukung lainnya. Sebelum memulai penelitian ini, peneliti melakukan kajian melalui penelitian-penelitian sebelumnya yang berkenaan dengan topik penelitian yang dipilih. Referensi yang diperoleh dari tahapan ini mencakup penelitian yang berhubungan dengan penggunaan metode CNN dan arsitektur YOLOv8 untuk proses transliterasi.

D. Metode Simulasi

Metode yang diterapkan pada model akan digambarkan pada tahapan ini. Metode simulasi terdiri dari beberapa tahapan yaitu perumusan konsep penelitian dan implementasi metode dan arsitektur.

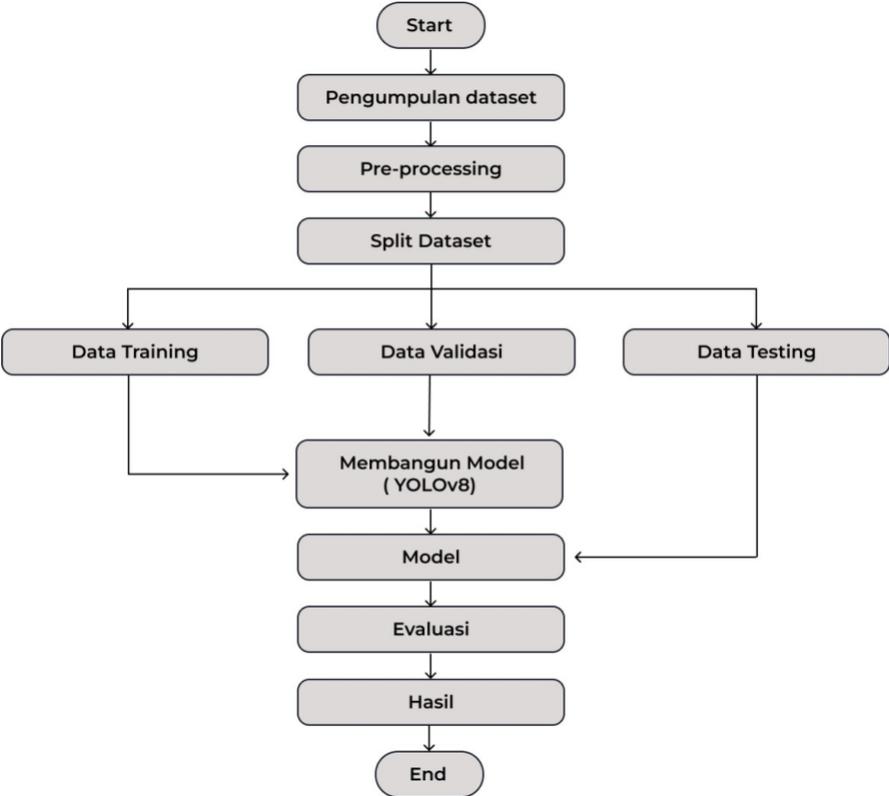
D1. Perumusan Konsep Penelitian

Peneliti merumuskan konsep penelitian berdasarkan hasil identifikasi masalah dan studi literatur yang telah dilakukan. Penelitian ini menggunakan metode CNN dengan arsitektur YOLOv8 untuk melakukan transliterasi “*Harah Jawoe*” ke latin. Penelitian ini memiliki proses pengujian model yang terdiri dari

tahapan *training* terhadap dataset training dan dataset validasi, membangun model untuk mampu melakukan transliterasi “*Harah Jawoe*” ke latin dan bertujuan untuk mendapatkan hasil akurasi yang tepat.

D2. mplementasi Metode dan Arsitektur

Tahapan ini merupakan tahapan yang digunakan untuk mentransformasikan konsep penelitian untuk di implementasikan pada model yang dibangun. Implementasi terdiri dari alur yang dikembangkan pada penelitian. Peneliti menggambarkan alur proses transliterasi “*Harah Jawoe*” ke latin seperti pada Gambar 3.2 dibawah ini.



Gambar 3.2 Alur Implementasi Metode

Adapun rincian mengenai alur implementasi metode dan arsitektu adalah sebagai berikut:

1. Pengumpulan Dataset

Tahapan ini merupakan tahapan awal yang dilakukan dalam penelitian dengan melakukan pengumpulan dataset yang akan digunakan. Dimana dataset yang digunakan dalam penelitian ini gambar “Harah Jawoe” yang peneliti tulis dengan berdasarkan pada buku karangan K.F.H. Van Langen (1889) yang berjudul *Atjehsle Taal*. Dataset yang digunakan berjumlah 3000 gambar dengan format JPG. Dataset ini nantinya akan digunakan untuk model dalam mengenali dan mengklasifikasikan data berdasarkan label yang dimilikinya serta digunakan sebagai data dalam mengevaluasi kinerja model dalam mengukur seberapa baik dalam model dalam pelatihan.

2. *Pre-Processing*

Tahapan selanjutnya adalah melakukan *pre-processing*. *Pre-processing* merupakan tahapan yang dilakukan dalam mempersiapkan data yang masih mentah agar dapat digunakan dalam melakukan *training* atau pelatihan pada model. Tahapan ini dilakukan agar dataset yang digunakan dapat memberikan hasil yang optimal dengan memiliki kualitas dataset yang baik. tahapan *pre-processing* mencakup beberapa aktivitas yaitu anotasi terhadap dataset atau pemberian label pada dataset dan augmentasi terhadap dataset.

3. *Split* Dataset

Kemudian akan dilakukannya *split* dataset yaitu untuk membagi dataset menjadi tiga bagian, yaitu dataset *training*, dataset validasi dan dataset *testing*. Dataset *training* adalah dataset yang digunakan untuk menghasilkan model *system* dari YOLOv8. Dataset validasi digunakan untuk memastikan kebenaran

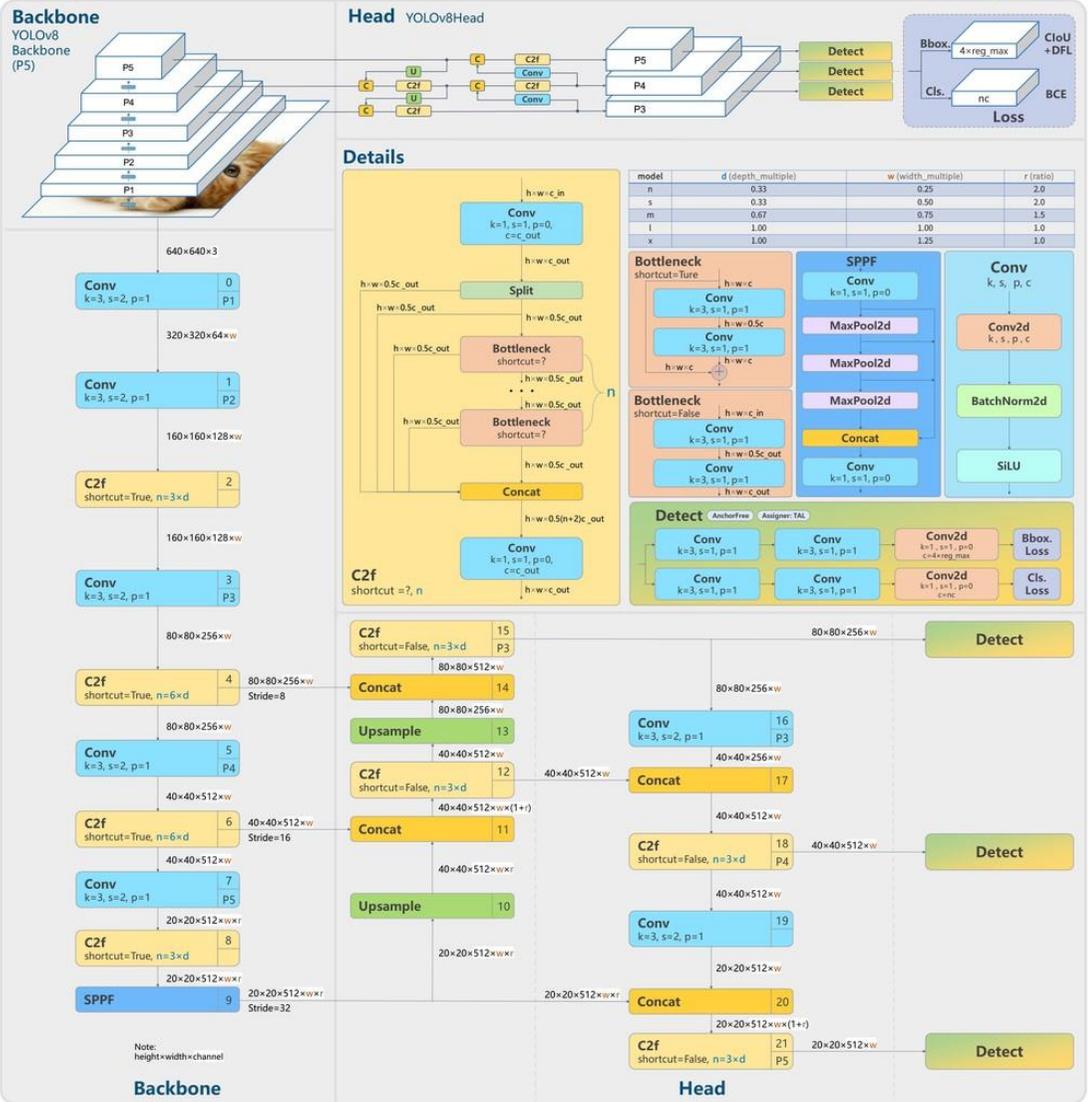
dari data yang telah dilatih. Selain itu juga digunakan untuk pengoptimalan data saat proses *training* model berlangsung. Sedangkan dataset *testing* adalah dataset yang digunakan untuk menghasilkan nilai pada saat evaluasi dilakukan.

4. Membangun Model

Pada tahapan ini, akan membangun model yang akan digunakan sebagai media pembelajaran dan pelatihan untuk dapat mengolah dataset dengan menggunakan metode *Convolutional Neural Network* untuk dapat melakukan prediksi. Model yang digunakan pada tahapan ini menggunakan arsitektur YOLOv8.

YOLOv8 merupakan arsitektur yang terdiri dari beberapa *layer* yang dibagi menjadi 3 bagian utama yaitu *Backbone*, *Neck* dan *Head*. *Backbone* berfungsi untuk meningkatkan nilai akurasi saat sebelum melakukan pendeteksian. Bagian *Head* merupakan bagian utama yang akan melakukan proses pendeteksian yang diawali dengan memposisikan *image* atau video pada area input dan kemudian diarahkan kedalam bagian *Backbone*. Bagian *Neck* merupakan bagian yang akan menggabungkan *feature maps* dari *stage* yang lain setelah bagian *Backbone* dilewati (Barokah Asmarahman Takarob, 2022).

YOLOv8



Gambar 3.3 Arsitektur dari YOLOV8 (RangeKing, 2023)

Hasil output deteksi dengan YOLO menghasilkan sebuah *bounding box* yang didalamnya terdapat identitas mengenai nama kelas atau label kelas dan nilai *confidence* atau nilai akurasi. Pada *bounding box* terdiri dari *variable* x, y, w dan h serta nilai *confidencinya*. Untuk *variable* x dan y akan mendefinisikan koordinat

titik pusat dari *bounding box*, sedangkan *variable* w dan h akan digunakan sebagai parameter piksel dari gambar. Nilai *confidence* dalam *bounding box* digunakan sebagai nilai kelas gambar yang terdeteksi antara *predicted box* dengan *ground truth*.

Sesudah membangun model tersebut, maka akan dilakukan proses *training* dengan menggunakan dataset untuk melatih model dalam memahami dan mengenali objek. Setelah proses *training* selesai akan dilakukan proses *testing* untuk menguji seberapa tepat kedua model ini dalam mengenali objek yang diberikan.

5. Model

Pada tahapan ini, model yang telah dibangun akan dilakukan proses pengujian dengan menggunakan dataset testing. Hal ini dilakukan untuk melihat seberapa baik model dalam melakukan prediksi terhadap “Harah Jawoe” berdasarkan pada pembelajaran dan pelatihan yang telah dilakukan pada saat membangun model. Tahapan ini juga akan dilihat tingkat performa yang dihasilkan oleh model terhadap dataset yang telah di inputkan untuk melihat nilai akurat model dalam mengenal dataset yang diujikan.

E. Metode Analisis Data

Tahapan terakhir dari penelitian ini adalah analisis data. Analisis data ini dilakukan dengan menilai *system* yang digunakan dalam penelitian ini.

E1. Evaluasi

Evaluasi adalah metode yang digunakan untuk mengukur kinerja *system* pada penelitian ini. Peneliti menggunakan *recall*, *precision*, *f1 score* dan *accuracy* yang mengacu pada model *confusion matrix*. Untuk melihat tingkat nilai *recall* akan diukur dengan menggunakan persamaan (2.1). *Precision* menggunakan

persamaan (2.2). untuk melihat nilai *f1 score* menggunakan persamaan (2.3) dan untuk melihat nilai akurasi secara keseluruhan akan menggunakan persamaan (2.4). Dimana evaluasi ini akan membandingkan hasil dari confusion matrix yang terdiri dari jumlah data *true positive*, *true false*, *false positive*, dan *false negative*.

E2. Alat Bantu Penelitian

Pada penelitian ini, menggunakan alat bantu berupa software dan hardware. Hardware yang digunakan yaitu satu unit laptop HP 14s-fq2 dengan spesifikasi sebagai berikut:

1. Processor AMD Ryzen 7 5825U with Radeon Graphics 2.00 GHz
2. RAM 16,0 GB (15,3 GB usable)
3. Storage 512 GB M.2 NVMe PCIe 3.0 SSD.

Untuk software yang digunakan pada penelitian ini yaitu Sistem Operasi Microsoft Windows 11 *Home Single Language Version 22H2* dan menggunakan tools lainnya, seperti bahasa pemrograman *Python*, *Tensorflow*, dan *Google Colaboration*.

F. Rencana Pembahasan

1. Preprocessing merupakan tahapan awal untuk memastikan arsitektur YOLOv8 dapat menerima data masukan (input) yang benar-benar sesuai dengan kebutuhan yang diharapkan. Dalam tahapan ini akan dilakukan:
 - a. *Resizing* Gambar: Dalam tahap ini, akan dilakukan perubahan citra gambar menjadi 1080 x 1080 piksel, supaya gambar-gambar masukan memiliki kesamaan ukuran, dan konsistensi yang baik.

- b. Normalisasi: Dalam tahap ini, dilakukan normalisasi citra gambar "Harah Jawoe" agar memiliki jumlah piksel dalam rentang yang tidak terlalu tinggi (0 hingga 1).
 - c. Augmentasi Data: Pada tahap ini, dilakukan rotasi, pergeseran, dan perubahan bentuk citra gambar "Harah Jawoe", sehingga dataset memiliki beragam variasi.
 - d. Pemrosesan Label: Pada tahap ini, dilakukan proses anotasi (bounding box) yang bersesuaian dengan perubahan ukuran gambar yang telah dilakukan pada tahapan "Resizing Gambar".
 - e. Encoding One-Hot: Pada tahap ini, dilakukan one-hot encoding, di mana setiap kelas objek mewakili vektor biner yang menunjukkan kelas objek yang ada dalam gambar.
 - f. Data Augmentation Tambahan (Opsional): Pada tahap ini, dilakukan augmentasi data tambahan, apabila dalam augmentasi awal, diperoleh hasil yang tidak terlalu baik (cutmix, mixup, atau augmentasi lainnya) yang bertujuan untuk meningkatkan keragaman data pelatihan.
2. Kemudian, setelah dilakukan proses *preprocessing*, dilakukan proses pelatihan dataset. Proses pelatihan akan dibagi ke dalam 5 kali *epoch*, dengan pembagian setiap *epoch* 30, 50, 70, 90 dan 100.
 3. Tahapan terakhir, adalah melakukan pengujian terhadap arsitektur yang telah memenuhi syarat pemenuhan dalam data latih sebelumnya.

G. Waktu dan Lokasi Penelitian

Penelitian dilakukan bertempat pada laboratorium Komputer Prodi. Teknologi Informasi Fakultas Sains dan Teknologi dengan menggunakan

komputer yang memiliki spesifikasi yang cukup tinggi. Dimulai sejak bulan Maret 2024, yang dilaksanakan dalam kurun waktu kurang lebih 6 bulan.

Tabel 3.1 Jadwal Pelaksanaan Penelitian

Kegiatan	Bulan ke 1	Bulan ke 2	Bulan ke 3	Bulan ke 4	Bulan ke 5	Bulan ke 6
Persiapan Awal						
Studi Literatur						
Identifikasi Literatur yang bersesuaian						
<i>Preprocessing</i> Dataset						
<i>Cleaning</i> Dataset						
Proses Latih dan Uji Dataset						
Penulisan publikasi						
Penulisan Laporan dan Draft Artikel						
Submit draft artikel						
Pengurusan HKI						
Pelaksanaan Seminar Laporan Akhir						

BAB IV

HASIL DAN PEMBAHASAN

A. Anotasi Dataset

A1. Pengumpulan Dataset

Dalam penelitian ini, dataset yang digunakan berbentuk gambar "Harah Jawoe" yang peneliti tulis dengan berdasarkan pada buku karangan K.F.H. Van Langen (1889) yang berjudul *Atjehsle Taal*. Proses penulisan "Harah Jawoe" ini menggunakan tools yang tersedia pada website <https://www.lexilogos.com/keyboard/jawi.html> dan beberapa font yang tersedia di Microsoft Word yaitu *Arabic Tradisional*, *Arial*, *Adobe Arabic*, *Tradisional Arabic* dan *Courier New*. Untuk pengambilan gambar datasetnya menggunakan *Snipping Tools*. Dataset ini akan digunakan sebagai inputan yang akan diproses dalam model.



Gambar 4.1 Dataset "Harah Jawoe" awal

Dataset yang digunakan pada awalnya berjumlah 132 gambar dengan format JPG dengan ukuran 1080x1080 piksel. Dataset ini terdiri dari 20 kata yang masing-masing katanya terdiri dari 12 dan 18 gambar. Perbedaan jumlah gambar yang digunakan ini diakibatkan dengan adanya variasi bentuk penulisan yang lebih banyak pada beberapa kata yang menyebabkan jumlah gambar yang dimiliki kata menjadi bervariasi. Untuk jumlah dataset dan kata yang digunakan dapat dilihat pada Tabel 4.1

Tabel 4.1 Jumlah Dataset Awal

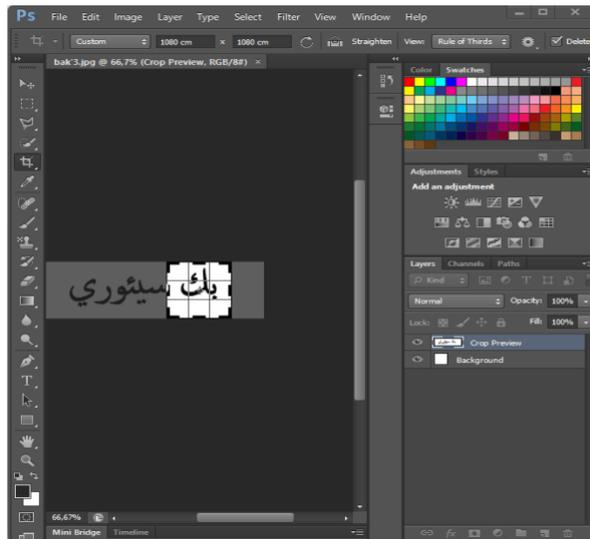
Nama Kata	Jumlah gambar
Adat, Bak, Bijaksana, Dipat, Gadöh, Gaséh, Hana, Hoka, Jak, Jikeumeung, Jimarit, Jipajöh, Jipeugah, Pikir, Sibak, Singoh, Tamöng, Tapinah, Ubak	12
Nibak	18
Total	132

A2. Pre-Processing

Pada tahap pengumpulan dataset, jumlah awal dari dataset yaitu 132 gambar sehingga diperlukan untuk dilakukan peningkatan variasi dari dataset yang ada. Aktivitas yang bisa dilakukan untuk meningkatkan variasi dataset yaitu dengan melakukan augmentasi data. Augmentasi data merupakan teknik-teknik yang digunakan dalam *machine learning* dan pengolahan data untuk meningkatkan ukuran dan kualitas data dengan membuat data yang bervariasi.

Augmentasi ini dilakukan untuk menghasilkan dataset akhir yang jumlah 3000 gambar.

Teknik pertama yang dilakukan dalam augmentasi data adalah melakukan crop terhadap data awal yang masih berbentuk kalimat untuk menyisakan dataset dalam bentuk kata dengan menggunakan bantuan tools yaitu *Adobe Photoshop*



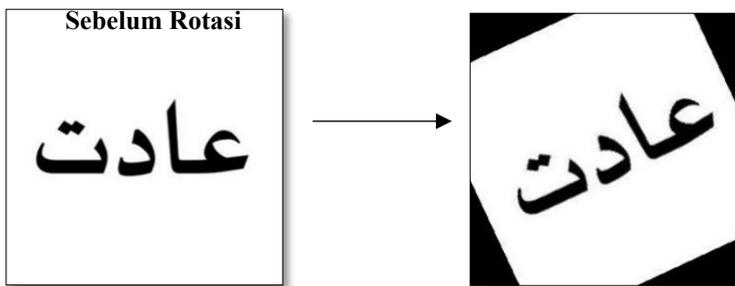
Gambar 4.2 Proses *Crop* Dataset

Dataset yang telah *dicrop* akan disimpan dalam satu folder dengan nama latin dari kata sebagai nama dari file gambar.

جماريت jimarit-1.jpg	جماريت jimarit-2.jpg	جماريت jimarit-3.jpg	جماريت jimarit-4.jpg	جماريت jimarit-5.jpg	جماريت jimarit-6.jpg	جماريت jimarit-7.jpg	جماريت jimarit-8.jpg	جماريت jimarit-9.jpg	جماريت jimarit-10.jpg
جماريت jimarit-11.jpg	جماريت jimarit-12.jpg	فيكر pikir-1.jpg	فيكر pikir-2.jpg	فيكر pikir-3.jpg	فيكر pikir-4.jpg	فيكر pikir-5.jpg	فيكر pikir-6.jpg	فيكر pikir-7.jpg	فيكر pikir-8.jpg
فيكر pikir-9.jpg	فيكر pikir-10.jpg	فيكر pikir-11.jpg	فيكر pikir-12.jpg	سيك si bak-1.jpg	سيك si bak-2.jpg	سيك si bak-3.jpg	سيك si bak-4.jpg	سيك si bak-5.jpg	سيك si bak-6.jpg
سيك si bak-7.jpg	سيك si bak-8.jpg	سيك si bak-9.jpg	سيك si bak-10.jpg	سيك si bak-11.jpg	سيك si bak-12.jpg	سيغوه singoh-1.jpg	سيغوه singoh-2.jpg	سيغوه singoh-3.jpg	سيغوه singoh-4.jpg
سيغوه	سيغوه	سيغوه	سيغوه	سيغوه	سيغوه	سيغوه	سيغوه	تموغ	تموغ

Gambar 4.3 Dataset setelah Crop

Kemudian akan dilakukan rotasi yaitu memutar gambar dengan berbagai sudut untuk menciptakan variasi orientasi objek dengan menggunakan bantuan library yang dimiliki oleh python yaitu *Python Image Library*. Contoh gambar yang telah dilakukan proses rotasi yaitu



Gambar 4.4 Contoh Rotasi Gambar

```

from PIL import Image
import os
import cv2
from google.colab import drive

# Mount Google Drive
drive.mount('/content/gdrive')

# Direktori sumber gambar
source_directory = '/content/gdrive/MyDrive/cerah_shear'

# Direktori tujuan untuk gambar yang sudah dirotasi
output_directory = '/content/gdrive/MyDrive/rotasi2'

if not os.path.exists(output_directory):
    os.makedirs(output_directory, exist_ok=True)

# Loop melalui semua gambar dalam direktori sumber
for filename in os.listdir(source_directory):
    if filename.endswith(".jpg"): # Sesuaikan ekstensi gambar
        # Baca gambar dengan OpenCV
        img_path = os.path.join(source_directory, filename)
        img = cv2.imread(img_path)

        # Konversi gambar ke format PIL
        pil_img = Image.fromarray(cv2.cvtColor(img, cv2.COLOR_BGR2RGB))

        # Rotasi gambar sebesar 15 derajat
        rotated_img = pil_img.rotate(15)

        # Simpan gambar yang sudah dirotasi ke direktori tujuan
        output_path = os.path.join(output_directory, filename)
        rotated_img.save(output_path)

print("Rotated images are saved in:", output_directory)

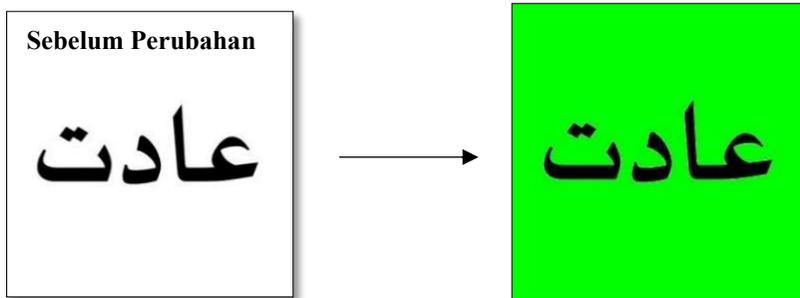
```

Gambar 4.5 *Source Code* Rotasi Gambar

Pada *Source code* diatas dijelaskan bahwa tahap awal dimulai dengan melakukan mount pada *google drive* sebagai media penyimpanan data. Step selanjutnya yaitu melakukan deklarasi terhadap sumber Lokasi data awal yang akan dirotasikan dan juga Lokasi data akan disimpan. Apabila Lokasi penyimpanan tidak ditemukan maka fungsi akan melakukan pembuatan folder

Lokasi data disimpan. Selanjutnya akan dilakukan proses augmentasi data dengan menentukan ukuran derajat rotasi yang akan di lakukan.

Selain melakukan rotasi, pada penelitian ini juga melakukan teknik augmentasi lainnya seperti mengubah warna *background* gambar, mencerahkan gambar dan meredupkan gambar. Pada gambar 4.6 merupakan contoh *background* gambar yang diubah menjadi warna hijau



Gambar 4.6 Contoh Perubahan Warna Gambar

Source code yang digunakan untuk melakukan perubahan warna *background* pada gambar memiliki Langkah-langkah yang sama seperti yang ditunjukkan pada gambar 4.5. tetapi memiliki tambahan *coding* yang ditunjukkan pada Gambar 4.7

```

# warna latar belakang baru (misalnya, merah)
background_color = (0, 255, 0) # RGB: Merah

# Loop melalui semua gambar dalam direktori sumber
for filename in os.listdir(source_directory):
    if filename.endswith(".jpg"): # Sesuaikan ekstensi gambar
        # Baca gambar dengan OpenCV
        img_path = os.path.join(source_directory, filename)
        img = cv2.imread(img_path)

        # Ubah warna latar belakang menjadi warna yang baru
        # Misalnya, ubah warna yang hampir putih menjadi merah
        hsv = cv2.cvtColor(img, cv2.COLOR_BGR2HSV)
        lower_white = np.array([0, 0, 200])
        upper_white = np.array([255, 255, 255])
        mask = cv2.inRange(hsv, lower_white, upper_white)
        img[mask > 0] = background_color

        # Konversi gambar kembali ke format PIL
        pil_img = Image.fromarray(cv2.cvtColor(img, cv2.COLOR_BGR2RGB))

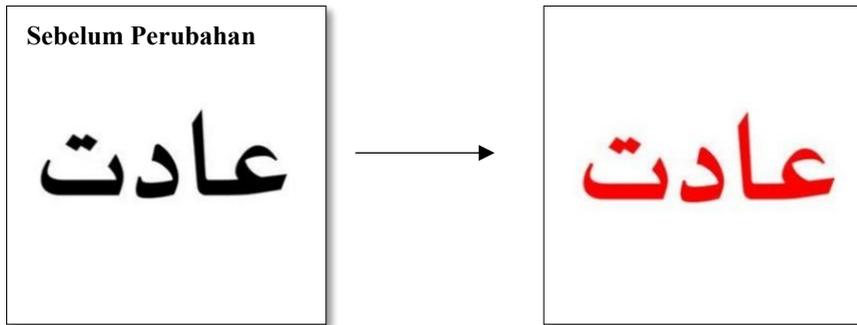
        # Simpan gambar yang sudah diubah warnanya ke direktori tujuan
        output_path = os.path.join(output_directory, filename)
        pil_img.save(output_path)

```

Gambar 4.7 *Source Code Mengubah Warna Background Gambar*

Pada source code diatas, dapat dilihat bahwa tahapan pertama yang dilakukan adalah membuat latar belakang baru yang Dimana disini digunakan warna hijau dengan menggunakan kode nilai RGB. Tahap selanjutnya gambar akan diubah kedala, mode warna HSV (hue-saturation-value) yang kemudian akan dikembalikan kedalam bentuk mode nilai RGB.

Variasi lainnya yang dilakukan yaitu mengubah warna tulisan kata yang ada pada gambar, dalam proses ini gambar akan diubah kedalam beberapa warna seperti merah dan hijau.



Gambar 4.8 Contoh Perubahan Warna pada Tulisan

```
# Ubah warna tulisan (misalnya, ubah warna tulisan menjadi merah)
# Anda dapat menyesuaikan nilai BGR sesuai keinginan Anda
img[:, :, 2] = 255 # Komponen Merah menjadi maksimum, komponen lain tetap

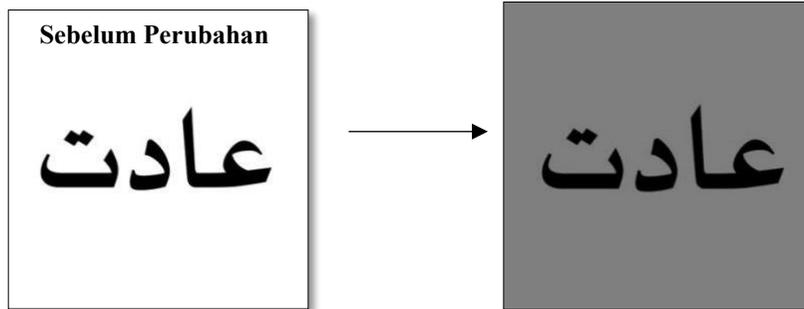
# Konversi gambar kembali ke format PIL
pil_img = Image.fromarray(cv2.cvtColor(img, cv2.COLOR_BGR2RGB))

# Simpan gambar yang sudah diubah warnanya ke direktori tujuan
output_path = os.path.join(output_directory, filename)
pil_img.save(output_path)
```

Gambar 4.9 Source Code Perubahan Warna Tulisan

Pada *source code* terdapat perintah untuk menentukan warna yang akan digunakan, Dimana dalam penentuan warna ini digunakan code nilai BRG (*Blue-Green-Red*) yang diatur sehingga nilai komponen merah menjadi maksimum yang menyatakan bahwa gambar awal akan diubah kedalam warna merah. Untuk warna lainnya tinggal diatur pada bagian nilai BRG nya.

Teknik untuk meredupkan gambar atau pencahayaan juga digunakan dalam penelitian ini



Gambar 4.10 Contoh Pencerahan Gambar

Step awal dalam proses augmentasi memiliki *source code* yang sama sehingga tahapan-tahapan yang dijelaskan mencakup hal yang berbeda. Kecerahan yang digunakan sebesar 0.3 yang menunjukkan semakin redup nilai factor kecerahan yang digunakan maka gambar yang dihasilkan akan semakin redup.

```
# Ubah kecerahan gambar (redup)
brightness_factor = 0.3 # Sesuaikan nilai kecerahan sesuai keinginan Anda
enhancer = ImageEnhance.Brightness(pil_img)
dimmed_img = enhancer.enhance(brightness_factor)
```

Gambar 4.11 *Source Code* Kecerahan Gambar

Penggunaan *enchacer* yang dibuat menggunakan kelas `ImageEnhance.Brithness` yang merupakan bagian dari *Python Image Library* digunakan untuk dapat memodifikasi kecerahan pada gambar. tahapan selanjutnya digunakan untuk mengurangi kecerahan yang dimiliki gambar berdasarkan dengan nilai factor kecerahan atau *brightness factor* yang telah ditentukan sebelumnya.

Variasi lain dilakukan dengan melakukan penggabungan beberapa teknik yang telah dilakukan seperti rotasi dengan *background* dan sebagainya sehingga memberikan jumlah data yang bervariasi dibandingkan dengan jumlah

data awal. Jumlah akhir yang dimiliki oleh dataset setelah melakukan proses augmentasi data dapat dilihat pada tabel 4.2

Tabel 4.2 Jumlah Dataset Setelah Augmentasi

Nama Kata	Jumlah gambar
Adat, Bak, Bijaksana, Dipat, Gadöh, Gaséh, Hana, Hoka, Jak, Jikeumeung, Jimarit, Jipajöh, Jipeugah, Pikir, Sibak, Singoh, Tamöng, Tapinah, Ubak	147
Nibak	207
Total	3000

Dataset setelah augmentasi yang berjumlah 3000 gambar kemudian ditingkatkan menjadi 9000 gambar. Penambahan ini dilakukan untuk menguji beberapa bentuk model YOLO, mulai dari model *small*, *medium*, hingga *big*. Dataset akhir yang terdiri dari 3000 gambar ini kemudian diubah menjadi beberapa ukuran sesuai dengan model, yaitu 80px, 540px, dan 1080px. Data yang telah diubah ukurannya ini akan digabungkan menjadi satu dataset penelitian. Tujuannya adalah agar saat proses pelatihan dilakukan, model dapat menyesuaikan diri dengan berbagai ukuran data yang tersedia.

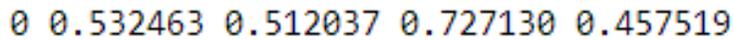
A3. Split Dataset dan Anotasi Data

Kemudian akan dilakukannya *split* dataset yaitu untuk membagi dataset menjadi tiga bagian, yaitu dataset *training*, dataset validasi dan dataset *testing*. Dalam melakukan pembagian dataset ini akan menggunakan rasio pembagian yaitu 80:10:10. Dengan 80% digunakan untuk dataset training, 10% untuk dataset validasi dan 10% untuk dataset testing sehingga jumlah dataset *training* harus lebih besar dibandingkan jumlah dataset pada dataset testing. Hal ini dikarenakan untuk menghasilkan model transliterasi yang memiliki akurasi yang tinggi yang disebabkan oleh model akan lebih banyak mengenal dan memahami pola pada gambar yang di masukan. Untuk melihat jumlah pembagian dataset dapat dilihat pada tabel 4.3.

Tabel 4.3 Pembagian Dataset Latih

Label Dataset	Jumlah Dataset yang digunakan
Dataset <i>Training</i>	7800
Dataset Validasi	600
Dataset <i>Testing</i>	600
Total Dataset	9000

Dataset yang telah di *split* tadi akan dilakukan pemberian label dengan menggunakan tools LabelImg yang akan disimpan dalam format .txt. File berformat .txt pada YOLO digunakan untuk menyimpan informasi mengenai Lokasi dan kelas dari objek dalam sebuah gambar. struktur dari label .txt pada YOLO dapat dilihat pada gambar 4.12



0 0.532463 0.512037 0.727130 0.457519

Gambar 4.12 Struktur Label pada YOLO

Pada gambar diatas, nilai 0 menyatakan nama kelas objek yang digunakan. kelas-kelas ini telah didefinisikan dalam file konfigurasi atau file dengan format .yaml yang berada dalam file terpisah yang mengidentifikasi *indeks* kelas tertentu. Sebagai contoh, dalam penelitian ini nama kelas dimulai dengan *indeks* 0 yang mewakili kelas “Adat”, indeks 1 mewakili kelas “Bak” dan seterusnya.

Nilai selanjutnya akan menyatakan *bounding box* atau kotak pembatas yang menentukan koordinat atau jarak yang menyatakan label dari gambar. nilai 0.532463 pada gambar 4.12 menyatakan titik Tengah koordinat *x_center* dari *bounding box* yang menyatakan lebar gambar. Nilai 0.512037 menyatakan titik Tengah *y_center* dari *bounding box* yang menyatakan tinggi gambar. Nilai 0.727130 menyatakan lebar *bounding box* dan nilai 0.457519 menyatakan tinggi *bounding box*.

Dalam model YOLO, nama label suatu gambar diberikan dengan label numerik yang dikenal dengan *class id*. Untuk penggunaan *class id* pada YOLO digunakan untuk mendeklarasikan urutan indeks dari masing-masing kelas yang digunakan yang Dimana *class id* ini akan digunakan sebagai rujukan pada proses *labeling* dataset dalam menentukan label kelas dari sebuah gambar. *class id* harus sesuai dengan label kelas yang akan digunakan dalam *data training*. Pada penelitian ini menggunakan 20 kata sehingga *class id* yang dimiliki berjumlah 19 indeks. Nilai *class id* ini akan disimpan didalam file terpisah yang disimpan

dengan format .yaml. Contoh class id yang digunakan dalam penelitian ini dapat dilihat pada gambar 4.13.

```
# Class Names
names:
  0: Adat
  1: Bak
  2: Bijaksana
  3: Dipat
  4: Gadöh
  5: Gaséh
  6: Hana
  7: Hoka
  8: Jak
  9: Jikeumeung
  10: Jimarit
  11: Jipajöh
  12: Jipeugah
  13: Nibak
  14: Pikir
  15: Sibak
  16: Singoh
  17: Tamöng
  18: Tapinah
  19: Ubak
```

Gambar 4.13 Contoh *Class Id* pada Model YOLO

B. Temuan Hasil Penelitian

Bagian ini membahas hasil pengujian untuk data gambar kata *Harah Jawoe* berdasarkan ukuran gambar 50px, 540px dan 1080px. Kemudian keseluruhan gambar tersebut dicampurkan menjadi satu bagian dataset. Artinya, dari keseluruhan dataset yang berjumlah 3000 gambar kata *Harah Jawoe* dikalikan 3, sehingga jumlah dataset kata *Harah Jawoe* berjumlah 9000 dataset.

B1. Pengujian Akurasi Kata *Harah Jawoe* untuk Model *Small*

Tabel 4.4 merupakan sebaran nilai hasil akurasi deteksi kata *Harah jawoe*, untuk Model *Small* (80px, 540px dan 1080px).

Tabel 4.4 Hasil Akurasi mAP50-95 Model *Small*

Model	Epoch	Nilai mAP50-95	%
<i>Small</i>	30	0.744	74.4
	50	0.719	71.9
	70	0.717	71.7
	90	0.704	70.4
	100	0.707	70.7

Berdasarkan hasil pengujian model *small* yang dilakukan terhadap keseluruhan gambar *Harah Jawoe* dengan berbagai ukuran, yaitu 80px, 540px, dan 1080px, diperoleh nilai akurasi yang seperti terlihat dalam Tabel 4.4. Pengujian ini dilakukan pada beberapa *epoch* yang menunjukkan nilai *Mean Average Precision* (mAP) yang bervariasi. Tabel tersebut memuat nilai mAP pada *epoch* 30, 50, 70, 90, dan 100 dengan nilai masing-masing sebesar 0.744 (74.4%), 0.719 (71.9%), 0.717 (71.7%), 0.704 (70.4%), dan 0.707 (70.7%).

Pada *epoch* ke-30, model *small* menghasilkan nilai mAP tertinggi yaitu sebesar 0.744 (74.4%). Nilai ini menunjukkan bahwa pada *epoch* tersebut, model mampu mendeteksi gambar *Harah Jawoe* dengan tingkat akurasi yang paling optimal dibandingkan dengan *epoch* lainnya. Penurunan nilai mAP yang terjadi setelah *epoch* ke-30, di mana pada *epoch* ke-50 nilai mAP turun menjadi 0.719 (71.9%), kemudian sedikit menurun lagi pada *epoch* ke-70 dengan nilai 0.717 (71.7%), dan terus turun hingga mencapai nilai 0.704 (70.4%) pada *epoch* ke-90, sebelum akhirnya sedikit meningkat pada *epoch* ke-100 dengan nilai mAP sebesar

0.707 (70.7%), menunjukkan adanya penurunan dan fluktuasi dalam kinerja model.

B2. Pengujian Akurasi Kata *Harah Jawoe* untuk Model *Medium*

Tabel 4.5 merupakan sebaran nilai hasil akurasi deteksi kata *Harah jawoe*, untuk Model *Medium* (80px, 540px dan 1080px).

Tabel 4.5 Hasil Akurasi mAP50-95 Model *Medium*

Model	Epoch	Nilai mAP50-95	%
<i>Medium</i>	30	0.722	72.2
	50	0.713	71.3
	70	0.72	72
	90	0.714	71.4
	100	0.711	71.1

Berdasarkan hasil pengujian model *medium* yang dilakukan terhadap keseluruhan gambar *Harah Jawoe* dengan berbagai ukuran, yaitu 80px, 540px, dan 1080px, diperoleh nilai akurasi yang tercantum dalam Tabel 4.5. Pengujian ini dilakukan pada beberapa *epoch* yang menunjukkan nilai *Mean Average Precision* (mAP) yang bervariasi. Tabel tersebut memuat nilai mAP pada *epoch* 30, 50, 70, 90, dan 100 dengan nilai masing-masing sebesar 0.722 (72.2%), 0.713 (71.3%), 0.72 (72%), 0.714 (71.4%), dan 0.711 (71.1%).

Pada *epoch* ke-30, model *medium* menghasilkan nilai mAP tertinggi yaitu sebesar 0.722 (72.2%). Nilai ini menunjukkan bahwa pada *epoch* tersebut, model mampu mendeteksi gambar *Harah Jawoe* dengan tingkat akurasi yang paling optimal dibandingkan dengan *epoch* lainnya. Nilai mAP yang mengalami penurunan setelah *epoch* ke-30, di mana pada *epoch* ke-50 nilai mAP turun menjadi

0.713 (71.3%), kemudian sedikit meningkat pada *epoch* ke-70 dengan nilai 0.72 (72%), dan akhirnya turun kembali pada *epoch* ke-90 menjadi 0.714 (71.4%), sebelum sedikit menurun lagi pada *epoch* ke-100 dengan nilai mAP sebesar 0.711 (71.1%), menunjukkan adanya fluktuasi dalam kinerja model.

B3. Pengujian Akurasi Kata *Harah Jawoe* untuk Model Normal

Tabel 4.6 merupakan sebaran nilai hasil akurasi deteksi kata *Harah jawoe*, untuk Model Normal (80px, 540px dan 1080px).

Tabel 4.6 Hasil Akurasi mAP50-95 Model Normal

Model	Epoch	Nilai mAP50-95	%
Normal	30	0.723	72.3
	50	0.722	72.2
	70	0.718	71.8
	90	0.704	70.4
	100	0.715	71.5

Berdasarkan hasil pengujian model normal yang dilakukan terhadap keseluruhan gambar *Harah Jawoe* dengan berbagai ukuran, yaitu 80px, 540px, dan 1080px, diperoleh nilai akurasi yang tercantum dalam Tabel 4.6. Pengujian ini dilakukan pada beberapa *epoch* yang menunjukkan nilai *Mean Average Precision* (mAP) yang bervariasi. Tabel tersebut memuat nilai mAP pada *epoch* 30, 50, 70, 90, dan 100 dengan nilai masing-masing sebesar 0.723 (72.3%), 0.722 (72.2%), 0.718 (71.8%), 0.704 (70.4%), dan 0.715 (71.5%).

Pada *epoch* ke-30, model normal menghasilkan nilai mAP tertinggi yaitu sebesar 0.723 (72.3%). Nilai ini menunjukkan bahwa pada *epoch* tersebut, model mampu mendeteksi gambar *Harah Jawoe* dengan tingkat akurasi yang paling optimal dibandingkan dengan *epoch* lainnya. Nilai mAP yang mengalami

penurunan setelah *epoch* ke-30, di mana pada *epoch* ke-50 nilai mAP turun menjadi 0.722 (72.2%), kemudian terus menurun pada *epoch* ke-70 dengan nilai 0.718 (71.8%), dan akhirnya turun lebih jauh pada *epoch* ke-90 menjadi 0.704 (70.4%), sebelum sedikit meningkat kembali pada *epoch* ke-100 dengan nilai mAP sebesar 0.715 (71.5%), menunjukkan adanya penurunan dan fluktuasi dalam kinerja model.

B4. Pengujian Akurasi Kata *Harah Jawoe* untuk Model *Big*

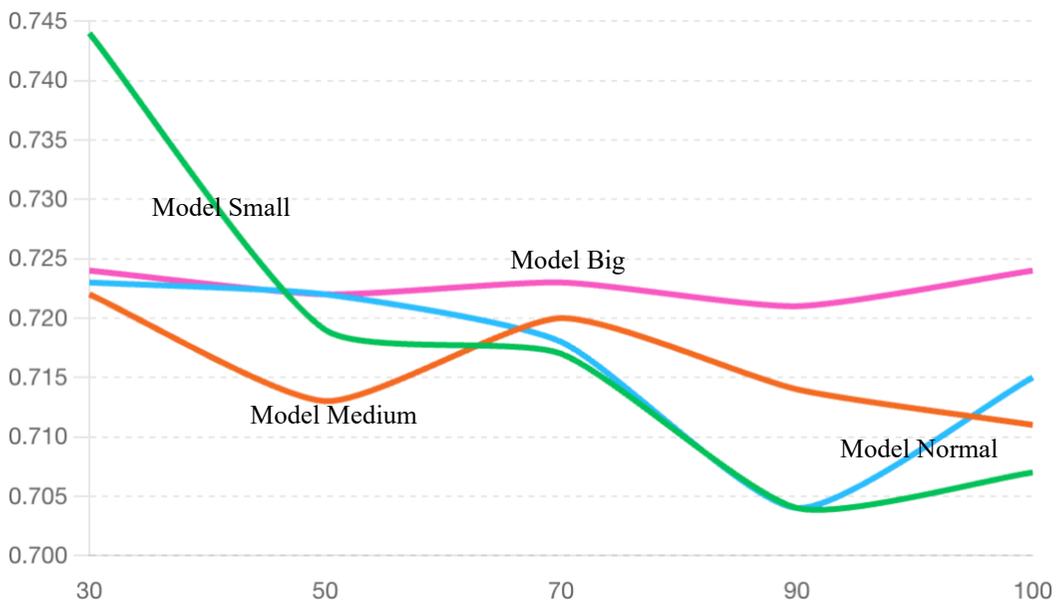
Tabel 4.7 merupakan sebaran nilai hasil akurasi deteksi kata *Harah jawoe*, untuk Model *Big* (80px, 540px dan 1080px).

Tabel 4.7 Hasil Akurasi mAP50-95 Model *Big*

Model	Epoch	Nilai mAP50-95	%
Big	30	0.724	72.4
	50	0.722	72.2
	70	0.723	72.3
	90	0.721	72.1
	100	0.724	72.4

Berdasarkan hasil pengujian model *big* yang dilakukan terhadap keseluruhan gambar *Harah Jawoe* dengan berbagai ukuran, yaitu 80px, 540px, dan 1080px, diperoleh nilai akurasi yang tercantum dalam Tabel 4.4. Pengujian ini dilakukan pada beberapa *epoch* yang menunjukkan nilai *Mean Average Precision* (mAP) yang bervariasi. Tabel tersebut memuat nilai mAP pada *epoch* 30, 50, 70, 90, dan 100 dengan nilai masing-masing sebesar 0.724 (72.4%), 0.722 (72.2%), 0.723 (72.3%), 0.721 (72.1%), dan 0.724 (72.4%).

Pada *epoch* ke-30 dan ke-100, model big menghasilkan nilai mAP tertinggi yaitu sebesar 0.724 (72.4%). Nilai ini menunjukkan bahwa pada *epoch* tersebut, model mampu mendeteksi gambar *Harah Jawoe* dengan tingkat akurasi yang paling optimal. Fluktuasi nilai mAP setelah *epoch* ke-30 menunjukkan penurunan yang tidak signifikan, di mana pada *epoch* ke-50 nilai mAP turun sedikit menjadi 0.722 (72.2%), kemudian meningkat kembali pada *epoch* ke-70 dengan nilai 0.723 (72.3%), dan turun sedikit pada *epoch* ke-90 menjadi 0.721 (72.1%) sebelum mencapai puncaknya lagi pada *epoch* ke-100 dengan nilai 0.724 (72.4%). Berikut grafik visualisasi performa dari masing-masing model pengujian akurasi kata *Harah Jawoe*.



Gambar 4.14 Grafik Visualisasi Performa Keseluruhan Model, Berdasarkan Dataset yang Dicampur

Grafik di atas menunjukkan perbandingan akurasi mAP50-95 antara berbagai model (*Small*, *Medium*, *Normal*, dan *Big*) pada berbagai *epoch* (30, 50, 70,

90, dan 100) dalam mendeteksi gambar kata *Harah Jawoe*. Grafik ini memberikan gambaran mengenai bagaimana setiap performa model pada setiap *epoch*, yakni:

1. Model *Small* menunjukkan nilai mAP tertinggi pada *epoch* ke-30 dengan nilai 0.744 (74.4%). Setelah itu, performanya menurun hingga *epoch* ke-90 dan sedikit meningkat pada *epoch* ke-100.
2. Model *Medium* memiliki nilai mAP tertinggi pada *epoch* ke-30 dengan nilai 0.722 (72.2%). Performanya relatif stabil namun sedikit menurun setelah *epoch* ke-30.
3. Model *Normal* menunjukkan nilai mAP tertinggi pada *epoch* ke-30 dengan nilai 0.723 (72.3%). Performanya menurun pada *epoch* ke-50 dan seterusnya, dengan sedikit peningkatan pada *epoch* ke-100.
4. Model *Big* menunjukkan konsistensi yang baik, dengan nilai mAP tertinggi pada *epoch* ke-30 dan ke-100 dengan nilai yang sama sebesar 0.724 (72.4%). Performanya relatif stabil di antara *epoch-epoch* tersebut.

Berdasarkan hasil analisis dari keempat model, model yang menunjukkan performa terbaik adalah Model *Big*, dengan nilai mAP tertinggi sebesar 0.724 (72.4%) pada *epoch* ke-30 dan ke-100. Model *Big* juga menunjukkan konsistensi yang baik dalam performanya dibandingkan dengan model lainnya.

Selain itu, penting untuk dicatat bahwa pada *epoch* ke-30, semua model menunjukkan performa yang baik dalam proses pengujian kata *Harah Jawoe*. Nilai mAP tertinggi dicapai oleh model *Small*, *Normal*, dan *Big* pada *epoch* ke-30, mengindikasikan bahwa *epoch* tersebut merupakan titik optimal dalam pelatihan untuk mencapai akurasi deteksi terbaik.

Penurunan performa setelah *epoch* ke-30 pada beberapa model menunjukkan adanya kebutuhan untuk menyesuaikan parameter pelatihan misalnya atau teknik augmentasi data untuk mempertahankan atau

meningkatkan kinerja model pada *epoch* yang lebih tinggi. Secara keseluruhan, analisis ini menunjukkan bahwa Model *Big* memiliki performa dan konsistensi terbaik dalam mendeteksi gambar *Harah Jawoe* pada berbagai ukuran gambar dan *epoch*.

B5. Hasil Testing

Berdasarkan pada sebaran data gambar grafik 4.14, diketahui bahwa model yang bekerja dengan tingkat performa terbaik dalam mendeteksi obyek kata *Harah Jawoe* adalah model *small*. Tetapi secara keseluruhan model *big* adalah model yang memiliki performa paling konsisten jika dibandingkan dengan model yang lain dalam mendeteksi obyek kata *Harah Jawoe*. Pada bagian ini dijelaskan hasil evaluasi terhadap dua model tersebut berdasarkan hasil *confusion matrix*.

C. Evaluasi Model

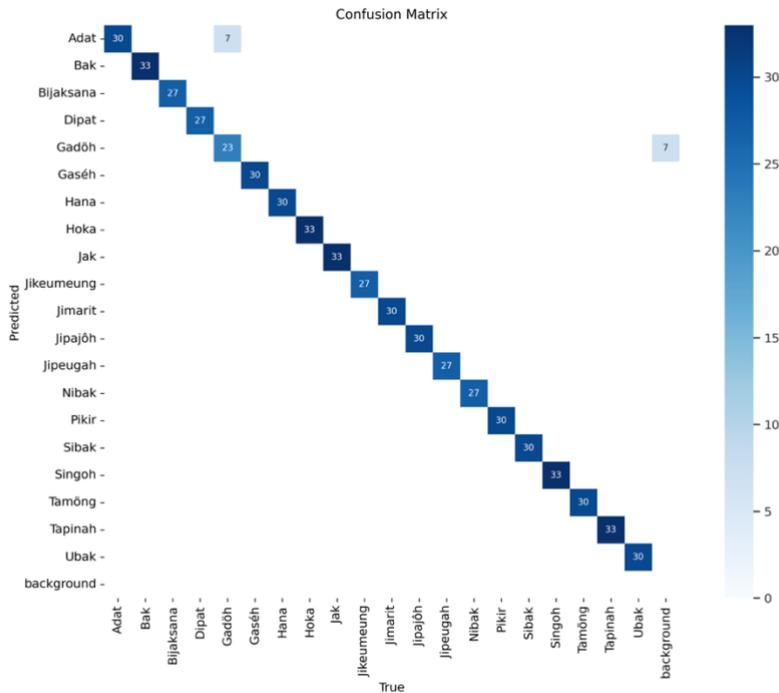
C1. Confusion Matrix 30 Epoch Small

Confusion matrix yang dihasilkan dari model *small* pada *epoch* ke-30 menunjukkan bahwa model ini memiliki kemampuan yang baik dalam melakukan prediksi terhadap data keseluruhan yang mencakup berbagai ukuran gambar. Namun, analisis lebih mendalam terhadap *confusion matrix* tersebut mengungkapkan adanya kesalahan prediksi yang cukup signifikan.

Model *small* pada *epoch* ke-30 berhasil melakukan prediksi yang akurat pada sebagian besar data, namun terdapat beberapa kesalahan prediksi. Secara spesifik, model memprediksi 7 data sebagai 'Adat', padahal sebenarnya data tersebut berlabel 'Gadöh'. Kesalahan ini menunjukkan bahwa model mengalami kebingungan dalam membedakan antara label 'Adat' dan 'Gadöh' pada beberapa instance.

Selain itu, model juga mendeteksi 7 data *background* sebagai 'Gadöh'. *Background* dalam konteks ini mengacu pada semua piksel dalam gambar yang

bukan bagian dari objek yang diberi label, yang seharusnya tidak terdeteksi sebagai objek tertentu. Berikut gambar visualisasi *confusion matrix* berdasarkan uraian penjelasan di atas.



Gambar 4.15 Grafik Visualisasi *Confusion Matrix* 30 Epoch Small

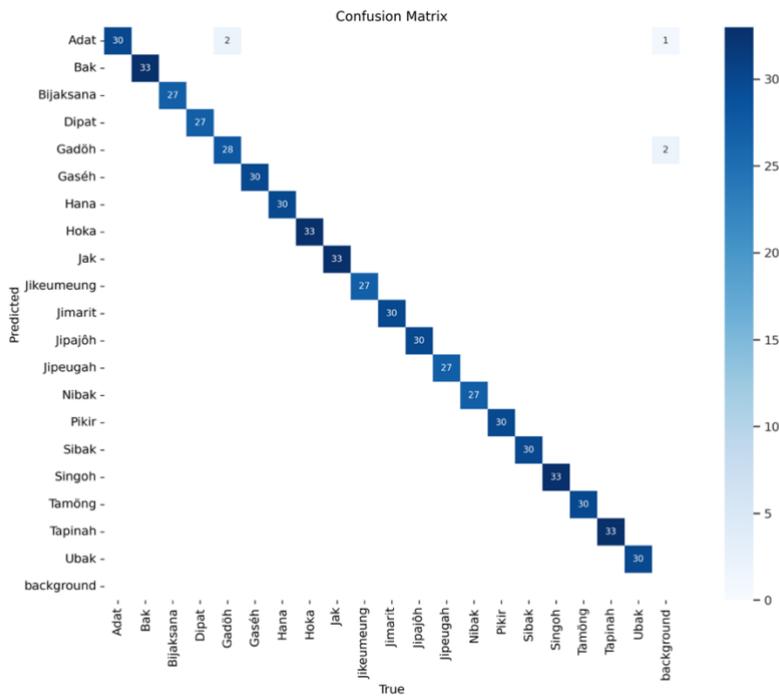
C2. *Confusion Matrix* 50 Epoch Model Small

Confusion matrix yang dihasilkan dari model *small* pada *epoch ke-50* menunjukkan bahwa model ini memiliki kemampuan yang baik dalam melakukan prediksi terhadap data keseluruhan yang mencakup berbagai ukuran gambar. Meskipun performa prediksi secara keseluruhan cukup baik, terdapat beberapa kesalahan prediksi yang perlu dianalisis lebih mendalam untuk meningkatkan akurasi model di masa mendatang.

Pada *epoch ke-50*, model *small* berhasil memprediksi dengan benar sejumlah besar data untuk berbagai label. Data yang diprediksi dengan benar oleh

model antara lain: 30 data untuk label 'Adat', 33 data untuk 'Bak', 27 data untuk 'Bijaksana', 27 data untuk 'Dipat', 28 data untuk 'Gadöh', 30 data untuk 'Gaséh', 30 data untuk 'Hana', 33 data untuk 'Hoka', 33 data untuk 'Jak', 27 data untuk 'Jikeumeung', 30 data untuk 'Jimarit', 30 data untuk 'Jipajöh', 27 data untuk 'Jipeugah', 27 data untuk 'Nibak', 30 data untuk 'Pikir', 30 data untuk 'Sibak', 33 data untuk 'Singoh', 30 data untuk 'Tamöng', 33 data untuk 'Tapinah', dan 30 data untuk 'Ubak'.

Namun, terdapat kesalahan prediksi yang signifikan yang perlu diperhatikan. Model memprediksi 2 data sebagai 'Adat' padahal sebenarnya data tersebut berlabel 'Gadöh'. Selain itu, model juga mendeteksi 1 data *background* sebagai 'Adat' dan 2 data *background* sebagai 'Gadöh'. Kesalahan ini menunjukkan bahwa model masih mengalami kesulitan dalam membedakan antara label 'Adat' dan 'Gadöh', serta dalam mengenali data *background* yang seharusnya tidak terdeteksi sebagai label tertentu. Berikut gambar visualisasi *confusion matrix* berdasarkan uraian penjelasan di atas.



Gambar 4.16 Grafik Visualisasi *Confusion Matrix* 50 Epoch *Small*

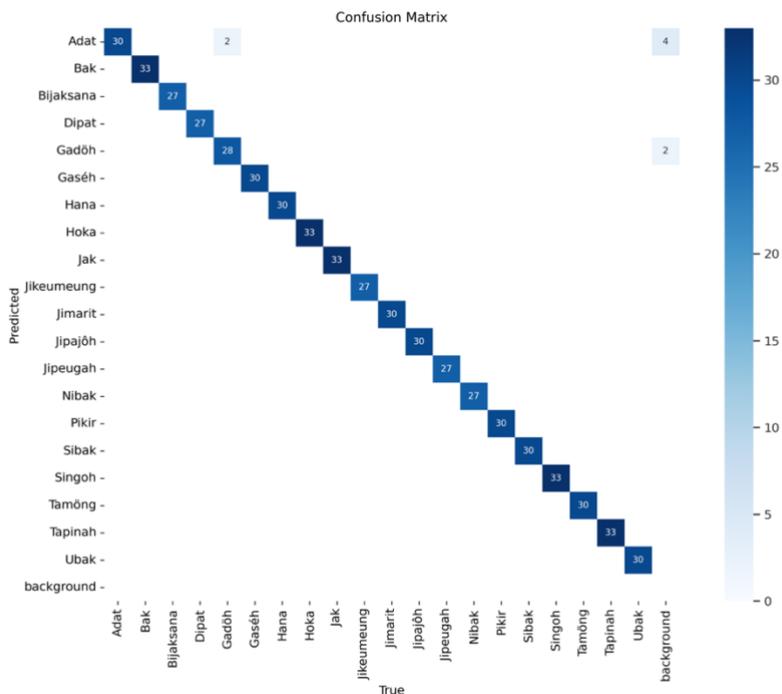
C3. *Confusion Matrix* 70 Epoch Model *Small*

Confusion matrix yang dihasilkan dari model *small* pada *epoch* ke-70 menunjukkan bahwa model ini memiliki kemampuan yang cukup baik dalam melakukan prediksi terhadap data keseluruhan yang mencakup berbagai ukuran gambar. Namun, meskipun performa prediksi secara keseluruhan cukup baik, terdapat beberapa kesalahan prediksi yang perlu dianalisis lebih mendalam untuk meningkatkan akurasi model di masa mendatang.

Pada *epoch* ke-70, model *small* berhasil memprediksi dengan benar sejumlah besar data untuk berbagai label. Data yang diprediksi dengan benar oleh model antara lain: 30 data untuk label 'Adat', 33 data untuk 'Bak', 27 data untuk 'Bijaksana', 27 data untuk 'Dipat', 28 data untuk 'Gadöh', 30 data untuk 'Gaséh', 30 data untuk 'Hana', 33 data untuk 'Hoka', 33 data untuk 'Jak', 27 data untuk

'Jikeumeung', 30 data untuk 'Jimarit', 30 data untuk 'Jipajöh', 27 data untuk 'Jipeugah', 27 data untuk 'Nibak', 30 data untuk 'Pikir', 30 data untuk 'Sibak', 33 data untuk 'Singoh', 30 data untuk 'Tamöng', 33 data untuk 'Tapinah', dan 30 data untuk 'Ubak'.

Namun, terdapat kesalahan prediksi yang signifikan yang perlu diperhatikan. Model memprediksi 2 data sebagai 'Adat' padahal sebenarnya data tersebut berlabel 'Gadöh'. Selain itu, model juga mendeteksi 4 data *background* sebagai 'Adat' dan 2 data *background* sebagai 'Gadöh'. Kesalahan ini menunjukkan bahwa model masih mengalami kesulitan dalam membedakan antara label 'Adat' dan 'Gadöh', serta dalam mengenali data *background* yang seharusnya tidak terdeteksi sebagai label tertentu. Berikut gambar visualisasi *confusion matrix* berdasarkan uraian penjelasan di atas.



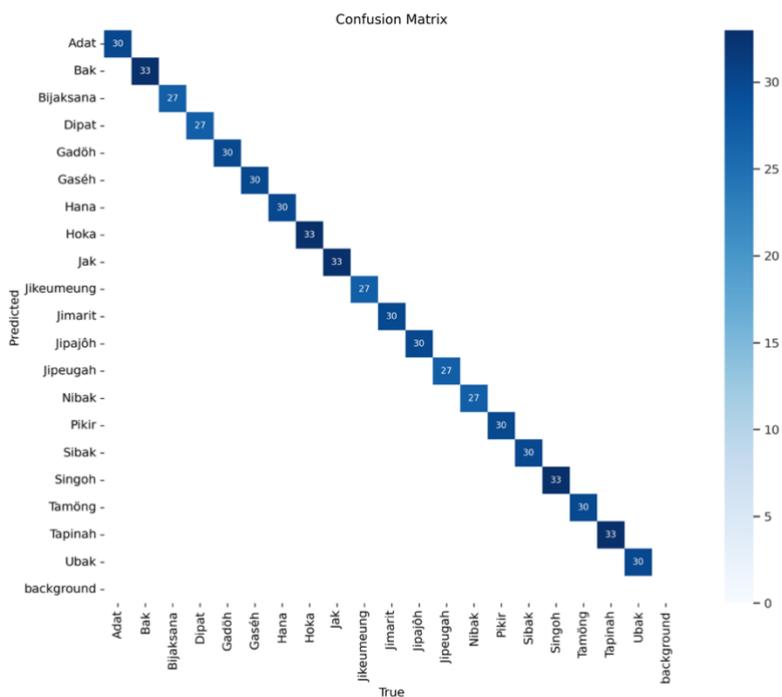
Gambar 4.17 Grafik Visualisasi *Confusion Matriks* 70 Epoch Small

C4. *Confusion Matrix* 90 Epoch Model *Small*

Confusion matrix yang dihasilkan dari model *small* pada *epoch* ke-90 menunjukkan bahwa model ini memiliki kemampuan yang cukup baik dalam melakukan prediksi terhadap data keseluruhan yang mencakup berbagai ukuran gambar. Meskipun performa prediksi secara keseluruhan cukup baik, terdapat beberapa kesalahan prediksi yang perlu dianalisis lebih mendalam untuk meningkatkan akurasi model di masa mendatang.

Pada *epoch* ke-90, model *small* berhasil memprediksi dengan benar sejumlah besar data untuk berbagai label. Data yang diprediksi dengan benar oleh model antara lain: 30 data untuk label 'Adat', 33 data untuk 'Bak', 27 data untuk 'Bijaksana', 27 data untuk 'Dipat', 28 data untuk 'Gadöh', 30 data untuk 'Gaséh', 30 data untuk 'Hana', 33 data untuk 'Hoka', 33 data untuk 'Jak', 27 data untuk 'Jikeumeung', 30 data untuk 'Jimarit', 30 data untuk 'Jipajöh', 27 data untuk 'Jipeugah', 27 data untuk 'Nibak', 30 data untuk 'Pikir', 30 data untuk 'Sibak', 33 data untuk 'Singoh', 30 data untuk 'Tamöng', 33 data untuk 'Tapinah', dan 30 data untuk 'Ubak'.

Namun, terdapat kesalahan prediksi yang signifikan yang perlu diperhatikan. Model memprediksi 2 data sebagai 'Adat' padahal sebenarnya data tersebut berlabel 'Gadöh'. Selain itu, model juga mendeteksi 2 data *background* sebagai 'Gadöh'. Kesalahan ini menunjukkan bahwa model masih mengalami kesulitan dalam membedakan antara label 'Adat' dan 'Gadöh', serta dalam mengenali data *background* yang seharusnya tidak terdeteksi sebagai label tertentu. Berikut gambar visualisasi *confusion matrix* berdasarkan uraian penjelasan di atas.



Gambar 4.18 Grafik Visualisasi *Confusion Matrix* 90 Epoch *Small*

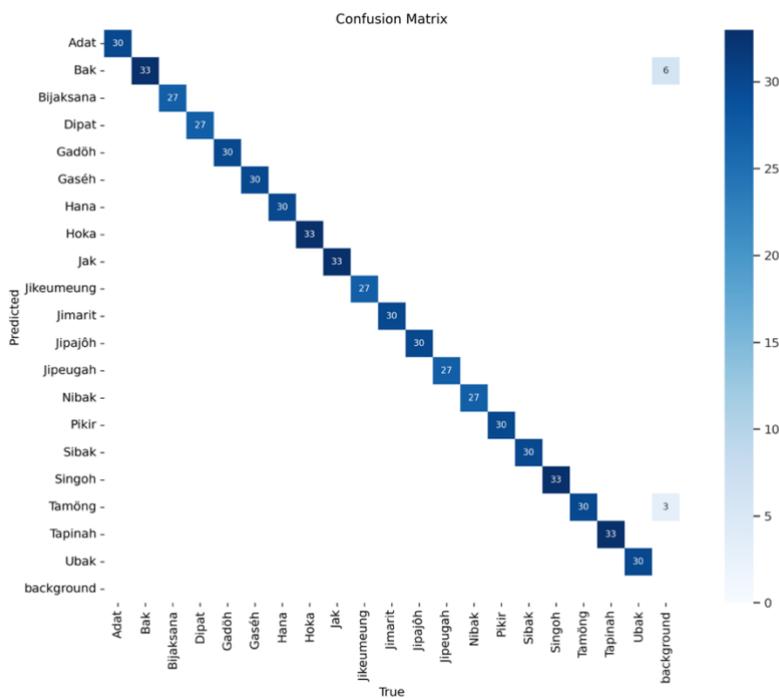
C5. *Confusion Matrix* 100 Epoch Model *Small*

Confusion matrix yang dihasilkan dari model *small* pada *epoch* ke-100 menunjukkan bahwa model ini memiliki kemampuan yang baik dalam melakukan prediksi terhadap data keseluruhan yang mencakup berbagai ukuran gambar. Meskipun performa prediksi secara keseluruhan cukup baik, terdapat beberapa kesalahan prediksi yang perlu dianalisis lebih mendalam untuk meningkatkan akurasi model di masa mendatang.

Pada *epoch* ke-100, model *small* berhasil memprediksi dengan benar sejumlah besar data untuk berbagai label. Data yang diprediksi dengan benar oleh model antara lain: 30 data untuk label 'Adat', 33 data untuk 'Bak', 27 data untuk 'Bijaksana', 27 data untuk 'Dipat', 30 data untuk 'Gadöh', 30 data untuk 'Gaséh', 30 data untuk 'Hana', 33 data untuk 'Hoka', 33 data untuk 'Jak', 27 data untuk

'Jikeumeung', 30 data untuk 'Jimarit', 30 data untuk 'Jipajôh', 27 data untuk 'Jipeugah', 27 data untuk 'Nibak', 30 data untuk 'Pikir', 30 data untuk 'Sibak', 33 data untuk 'Singoh', 30 data untuk 'Tamöng', 33 data untuk 'Tapinah', dan 30 data untuk 'Ubak'.

Namun, terdapat kesalahan prediksi yang signifikan yang perlu diperhatikan. Model memprediksi 6 data *background* sebagai 'Bak' dan 3 data *background* sebagai 'Tamöng'. Kesalahan ini menunjukkan bahwa model masih mengalami kesulitan dalam mengenali data *background* yang seharusnya tidak terdeteksi sebagai label tertentu. Berikut gambar visualisasi *confusion matrix* berdasarkan uraian penjelasan di atas.



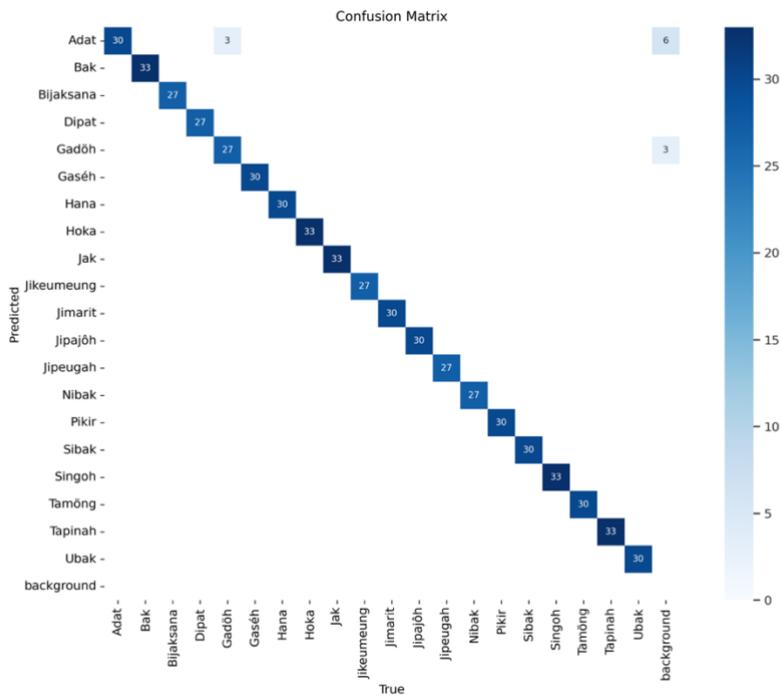
Gambar 4.19 Grafik Visualisasi *Confusion Matrix* 100 Epoch Small

C6. *Confusion Matrix* 30 Epoch Model Big

Confusion matrix yang dihasilkan dari model *big* pada *epoch* ke-30 menunjukkan bahwa model ini memiliki kemampuan yang sangat baik dalam melakukan prediksi terhadap data keseluruhan yang mencakup berbagai ukuran gambar. Meskipun performa prediksi secara keseluruhan cukup baik, terdapat beberapa kesalahan prediksi yang perlu dianalisis lebih mendalam untuk meningkatkan akurasi model di masa mendatang.

Pada *epoch* ke-30, model *big* berhasil memprediksi dengan benar sejumlah besar data untuk berbagai label. Data yang diprediksi dengan benar oleh model antara lain: 30 data untuk label 'Adat', 33 data untuk 'Bak', 27 data untuk 'Bijaksana', 27 data untuk 'Dipat', 27 data untuk 'Gadöh', 30 data untuk 'Gaséh', 30 data untuk 'Hana', 33 data untuk 'Hoka', 33 data untuk 'Jak', 27 data untuk 'Jikeumeung', 30 data untuk 'Jimarit', 30 data untuk 'Jipajöh', 27 data untuk 'Jipeugah', 27 data untuk 'Nibak', 30 data untuk 'Pikir', 30 data untuk 'Sibak', 33 data untuk 'Singoh', 30 data untuk 'Tamöng', 33 data untuk 'Tapinah', dan 30 data untuk 'Ubak'.

Namun, terdapat kesalahan prediksi yang signifikan yang perlu diperhatikan. Model memprediksi 3 data sebagai 'Adat' padahal sebenarnya data tersebut berlabel 'Gadöh'. Selain itu, model juga mendeteksi 3 data *background* sebagai 'Gadöh' dan 6 data *background* sebagai 'Adat'. Kesalahan ini menunjukkan bahwa model masih mengalami kesulitan dalam membedakan antara label 'Adat' dan 'Gadöh', serta dalam mengenali data *background* yang seharusnya tidak terdeteksi sebagai label tertentu. Berikut gambar visualisasi *confusion matrix* berdasarkan uraian penjelasan di atas.



Gambar 4.20 Grafik Visualisasi *Confusion Matrix* 30 Epoch Big

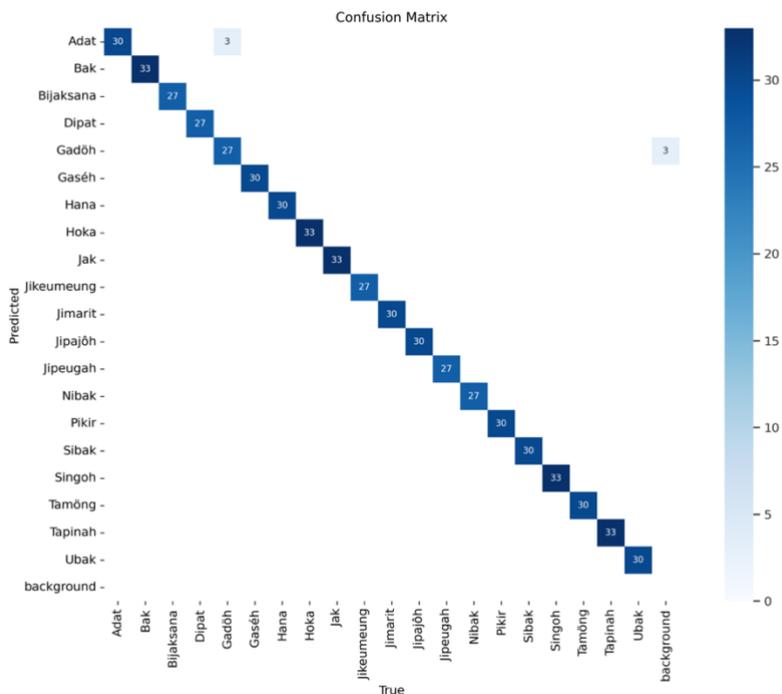
C7. *Confusion Matrix* 50 Epoch Model Big

Confusion matrix yang dihasilkan dari model *big* pada *epoch* ke-50 menunjukkan bahwa model ini memiliki kemampuan yang baik dalam melakukan prediksi terhadap data keseluruhan yang mencakup berbagai ukuran gambar. Meskipun performa prediksi secara keseluruhan cukup baik, terdapat beberapa kesalahan prediksi yang perlu dianalisis lebih mendalam untuk meningkatkan akurasi model di masa mendatang.

Pada *epoch* ke-50, model *big* berhasil memprediksi dengan benar sejumlah besar data untuk berbagai label. Data yang diprediksi dengan benar oleh model antara lain: 30 data untuk label 'Adat', 33 data untuk 'Bak', 27 data untuk 'Bijaksana', 27 data untuk 'Dipat', 27 data untuk 'Gadöh', 30 data untuk 'Gaséh', 30 data untuk 'Hana', 33 data untuk 'Hoka', 33 data untuk 'Jak', 27 data untuk

'Jikeumeung', 30 data untuk 'Jimarit', 30 data untuk 'Jipajöh', 27 data untuk 'Jipeugah', 27 data untuk 'Nibak', 30 data untuk 'Pikir', 30 data untuk 'Sibak', 33 data untuk 'Singoh', 30 data untuk 'Tamöng', 33 data untuk 'Tapinah', dan 30 data untuk 'Ubak'.

Namun, terdapat kesalahan prediksi yang signifikan yang perlu diperhatikan. Model memprediksi 3 data sebagai 'Adat' padahal sebenarnya data tersebut berlabel 'Gadöh'. Selain itu, model juga mendeteksi 3 data *background* sebagai 'Gadöh'. Kesalahan ini menunjukkan bahwa model masih mengalami kesulitan dalam membedakan antara label 'Adat' dan 'Gadöh', serta dalam mengenali data *background* yang seharusnya tidak terdeteksi sebagai label tertentu. Berikut gambar visualisasi *confusion matrix* berdasarkan uraian penjelasan di atas.



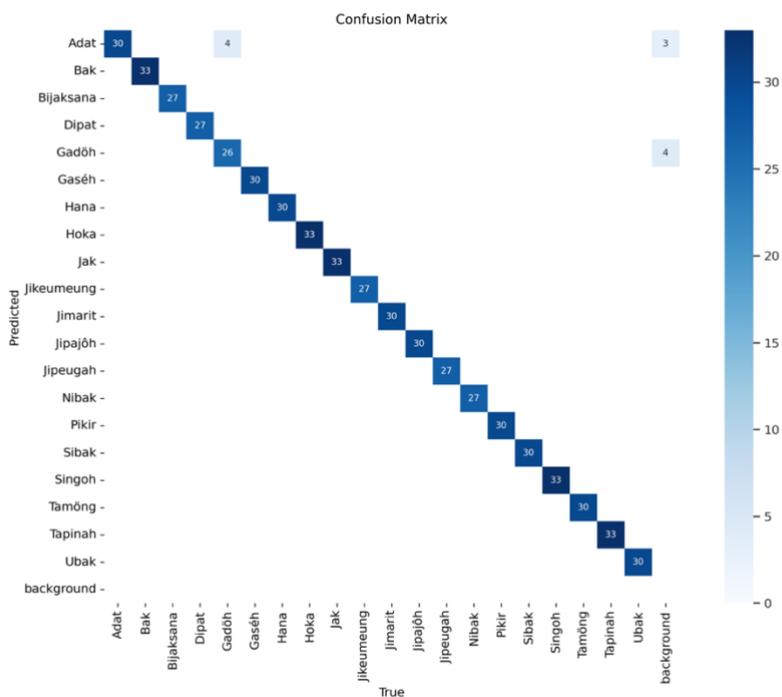
Gambar 4.21 Grafik Visualisasi *Confusion Matrix* 50 Epoch Big

C8. *Confusion Matrix* 70 Epoch Model *Big*

Confusion matrix yang dihasilkan dari model *big* pada *epoch* ke-70 menunjukkan bahwa model ini memiliki kemampuan yang baik dalam melakukan prediksi terhadap data keseluruhan yang mencakup berbagai ukuran gambar. Meskipun performa prediksi secara keseluruhan cukup baik, terdapat beberapa kesalahan prediksi yang perlu dianalisis lebih mendalam untuk meningkatkan akurasi model di masa mendatang.

Pada *epoch* ke-70, model *big* berhasil memprediksi dengan benar sejumlah besar data untuk berbagai label. Data yang diprediksi dengan benar oleh model antara lain: 30 data untuk label 'Adat', 33 data untuk 'Bak', 27 data untuk 'Bijaksana', 27 data untuk 'Dipat', 26 data untuk 'Gadöh', 30 data untuk 'Gaséh', 30 data untuk 'Hana', 33 data untuk 'Hoka', 33 data untuk 'Jak', 27 data untuk 'Jikeumeung', 30 data untuk 'Jimarit', 30 data untuk 'Jipajöh', 27 data untuk 'Jipeugah', 27 data untuk 'Nibak', 30 data untuk 'Pikir', 30 data untuk 'Sibak', 33 data untuk 'Singoh', 30 data untuk 'Tamöng', 33 data untuk 'Tapinah', dan 30 data untuk 'Ubak'.

Namun, terdapat kesalahan prediksi yang signifikan yang perlu diperhatikan. Model memprediksi 4 data sebagai 'Adat' padahal sebenarnya data tersebut berlabel 'Gadöh'. Selain itu, model juga mendeteksi 4 data *background* sebagai 'Gadöh' dan 3 data *background* sebagai 'Adat'. Kesalahan ini menunjukkan bahwa model masih mengalami kesulitan dalam membedakan antara label 'Adat' dan 'Gadöh', serta dalam mengenali data *background* yang seharusnya tidak terdeteksi sebagai label tertentu. Berikut gambar visualisasi *confusion matrix* berdasarkan uraian penjelasan di atas.



Gambar 4.22 Grafik Visualisasi *Confusion Matrix* 70 Epoch Big

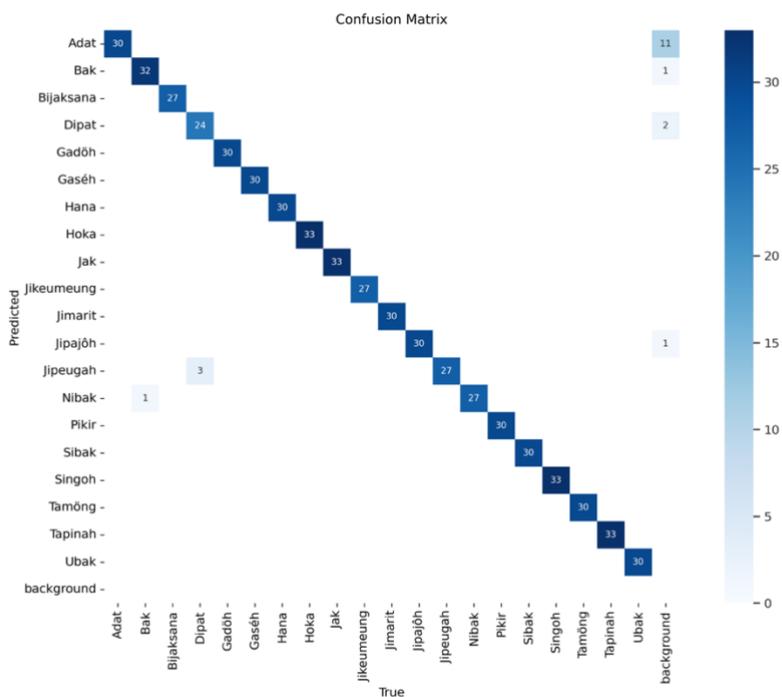
C9. *Confusion Matrix* 90 Epoch Model Big

Confusion matrix yang dihasilkan dari model *big* pada *epoch* ke-90 menunjukkan bahwa model ini memiliki kemampuan yang baik dalam melakukan prediksi terhadap data keseluruhan yang mencakup berbagai ukuran gambar. Meskipun performa prediksi secara keseluruhan cukup baik, terdapat beberapa kesalahan prediksi yang perlu dianalisis lebih mendalam untuk meningkatkan akurasi model di masa mendatang.

Pada *epoch* ke-90, model *big* berhasil memprediksi dengan benar sejumlah besar data untuk berbagai label. Data yang diprediksi dengan benar oleh model antara lain: 30 data untuk label 'Adat', 32 data untuk 'Bak', 27 data untuk 'Bijaksana', 24 data untuk 'Dipat', 30 data untuk 'Gadöh', 30 data untuk 'Gaséh', 30

data untuk 'Hana', 33 data untuk 'Hoka', 33 data untuk 'Jak', 27 data untuk 'Jikeumeung', 30 data untuk 'Jimarit', 30 data untuk 'Jipajôh', 27 data untuk 'Jipeugah', 27 data untuk 'Nibak', 30 data untuk 'Pikir', 30 data untuk 'Sibak', 33 data untuk 'Singoh', 30 data untuk 'Tamöng', 33 data untuk 'Tapinah', dan 30 data untuk 'Ubak'.

Namun, terdapat kesalahan prediksi yang signifikan yang perlu diperhatikan. Model memprediksi 1 data sebagai 'Nibak' padahal sebenarnya data tersebut berlabel 'Bak'. Selain itu, model memprediksi 3 data sebagai 'Jipeugah' padahal sebenarnya data tersebut berlabel 'Dipat'. Model juga mendeteksi 1 data *background* sebagai 'Bak', 2 data sebagai 'Dipat', 1 data sebagai 'Jipeugah', dan 11 data *background* sebagai 'Adat'. Kesalahan ini menunjukkan bahwa model masih mengalami kesulitan dalam membedakan antara beberapa label tertentu dan dalam mengenali data *background* yang seharusnya tidak terdeteksi sebagai label tertentu. Berikut gambar visualisasi *confusion matrix* berdasarkan uraian penjelasan di atas.



Gambar 4.23 Grafik Visualisasi *Confusion Matrix* 90 Epoch Big

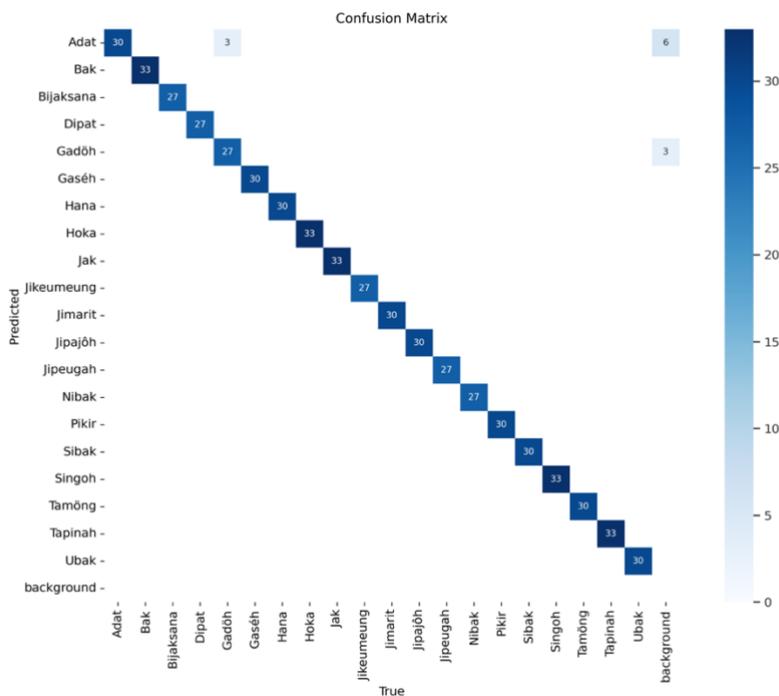
C10. *Confusion Matrix* 100 Epoch Model Big

Confusion matrix yang dihasilkan dari model *big* pada *epoch* ke-100 menunjukkan bahwa model ini memiliki kemampuan yang baik dalam melakukan prediksi terhadap data keseluruhan yang mencakup berbagai ukuran gambar. Meskipun performa prediksi secara keseluruhan cukup baik, terdapat beberapa kesalahan prediksi yang perlu dianalisis lebih mendalam untuk meningkatkan akurasi model di masa mendatang.

Pada *epoch* ke-100, model *big* berhasil memprediksi dengan benar sejumlah besar data untuk berbagai label. Data yang diprediksi dengan benar oleh model antara lain: 30 data untuk label 'Adat', 33 data untuk 'Bak', 27 data untuk 'Bijaksana', 27 data untuk 'Dipat', 27 data untuk 'Gadöh', 30 data untuk 'Gaséh', 30 data untuk 'Hana', 33 data untuk 'Hoka', 33 data untuk 'Jak', 27 data untuk

'Jikeumeung', 30 data untuk 'Jimarit', 30 data untuk 'Jipajôh', 27 data untuk 'Jipeugah', 27 data untuk 'Nibak', 30 data untuk 'Pikir', 30 data untuk 'Sibak', 33 data untuk 'Singoh', 30 data untuk 'Tamông', 33 data untuk 'Tapinah', dan 30 data untuk 'Ubak'.

Namun, terdapat kesalahan prediksi yang signifikan yang perlu diperhatikan. Model memprediksi 3 data sebagai 'Adat' padahal sebenarnya data tersebut berlabel 'Gadôh'. Kesalahan ini menunjukkan bahwa model masih mengalami kesulitan dalam membedakan antara label 'Adat' dan 'Gadôh'. Berikut gambar visualisasi *confusion matrix* berdasarkan uraian penjelasan di atas.



Gambar 4.24 Grafik Visualisasi *Confusion Matrix* 100 Epoch Big

D. Recall, Precision, F1-Score dan Accuracy

D1. Model *Small* 30 Epoch

Berdasarkan hasil *confusion matrix* yang dihasilkan dari model *small* pada *epoch* ke-30, dapat dilihat bahwa model ini memiliki performa yang sangat baik dalam melakukan prediksi terhadap data keseluruhan yang mencakup berbagai ukuran gambar. Untuk mengevaluasi kinerja model, digunakan beberapa metrik evaluasi yaitu *accuracy*, *precision*, *recall*, dan *f1-score*. Tabel 4.8 merupakan nilai *true positif* dari model *small* untuk 30 *epoch*.

Tabel 4.8 Nilai TP Setiap Label pada Model *Small* 30 Epoch

Label	Nilai TP	Label	Nilai TP
Adat	30	Jimarit	30
Bak	33	Jipajôh	30
Bijaksana	27	Jipeugah	27
Dipat	27	Nibak	27
Gadöh	23	Pikir	30
Gaséh	30	Sibak	30
Hana	30	Singoh	33
Hoka	33	Tamông	30
Jak	33	Tapinah	33
Jikeumeung	27	Ubak	30

Confusion matrix menunjukkan bahwa nilai *True Positive* (TP) untuk setiap label adalah sebagai berikut: 'Adat' sebanyak 30, 'Bak' sebanyak 33, 'Bijaksana' sebanyak 27, 'Dipat' sebanyak 27, 'Gadöh' sebanyak 23, 'Gaséh' sebanyak 30, 'Hana' sebanyak 30, 'Hoka' sebanyak 33, 'Jak' sebanyak 33, 'Jikeumeung' sebanyak 27, 'Jimarit' sebanyak 30, 'Jipajôh' sebanyak 30, 'Jipeugah'

sebanyak 27, 'Nibak' sebanyak 27, 'Pikir' sebanyak 30, 'Sibak' sebanyak 30, 'Singoh' sebanyak 33, 'Tamöng' sebanyak 30, 'Tapinah' sebanyak 33, dan 'Ubak' sebanyak 30. Total nilai TP keseluruhan adalah 593.

$$\begin{aligned} \text{Overall TP} &= 30 + 33 + 27 + 27 + 23 + 30 + 30 + 33 + 33 + 27 + 30 + 30 + 27 + \\ &27 + 30 + 30 + 33 + 30 + 33 + 30 \\ &= 593 \end{aligned}$$

Nilai *False Negative* (FN) untuk setiap label menunjukkan bahwa hanya label 'Adat' yang memiliki FN sebanyak 7, sementara label lainnya memiliki nilai FN sebesar 0, sesuai yang tertera pada table 4.8. Total nilai FN keseluruhan adalah 7.

Tabel 4.9 Nilai FN Setiap Label pada Model *Small 30 Epoch*

Label	Nilai FN	Label	Nilai FN
Adat	7	Jimarit	0
Bak	0	Jipajöh	0
Bijaksana	0	Jipeugah	0
Dipat	0	Nibak	0
Gadöh	0	Pikir	0
Gaséh	0	Sibak	0
Hana	0	Singoh	0
Hoka	0	Tamöng	0
Jak	0	Tapinah	0
Jikeumeung	0	Ubak	0

Nilai *False Positive* (FP) untuk setiap label menunjukkan bahwa hanya label 'Gadöh' yang memiliki FP sebanyak 7, sementara label lainnya memiliki nilai FP sebesar 0. Total nilai FP keseluruhan adalah 7.

Tabel 4.10 Nilai FP Setiap Label pada Model *Small 30 Epoch*

Label	Nilai FP	Label	Nilai FP
Adat	0	Jimarit	0
Bak	0	Jipajöh	0
Bijaksana	0	Jipeugah	0
Dipat	0	Nibak	0
Gadöh	7	Pikir	0
Gaséh	0	Sibak	0
Hana	0	Singoh	0
Hoka	0	Tamông	0
Jak	0	Tapinah	0
Jikeumeung	0	Ubak	0

Berdasarkan pada nilai TP, FP dan FN yang telah diperoleh sebelumnya, maka dihitung nilai TN dengan cara mengurangi nilai total data dengan jumlah nilai TP, FP dan FN dengan persamaan sebagai berikut:

$$\text{True Negative (TN)} = \text{Total} - (\text{TP} + \text{FP} + \text{FN}) \dots\dots\dots (1)$$

Sehingga dari hasil perhitungan persamaan (1) di atas, diperoleh hasil TN seperti yang tertera pada tabel 4.11 berikut:

Tabel 4.11 Nilai TN Setiap Label pada Model *Small 30 Epoch*

Label	Nilai TN
Adat	563
Bak	567
Bijaksana	573
Dipat	573
Gadöh	570
Gaséh	570
Hana	570
Hoka	567
Jak	567
Jikeumeung	573

Label	Nilai TN
Jimarit	570
Jipajöh	570
Jipeugah	573
Nibak	573
Pikir	570
Sibak	570
Singoh	567
Tamông	570
Tapinah	567
Ubak	570

Seperti sebaran nilai yang tertera pada Tabel 4.11 di atas, maka nilai TN untuk setiap label adalah sebagai berikut: 'Adat' sebanyak 563, 'Bak' sebanyak 567, 'Bijaksana' sebanyak 573, 'Dipat' sebanyak 573, 'Gadöh' sebanyak 570, 'Gaséh' sebanyak 570, 'Hana' sebanyak 570, 'Hoka' sebanyak 567, 'Jak' sebanyak 567, 'Jikeumeung' sebanyak 573, 'Jimarit' sebanyak 570, 'Jipajöh' sebanyak 570, 'Jipeugah' sebanyak 573, 'Nibak' sebanyak 573, 'Pikir' sebanyak 570, 'Sibak' sebanyak 570, 'Singoh' sebanyak 567, 'Tamông' sebanyak 570, 'Tapinah' sebanyak 567, dan 'Ubak' sebanyak 570. Total nilai TN keseluruhan adalah 11.399.

$$\begin{aligned}
 \text{Overall TN} &= 563 + 567 + 573 + 573 + 570 + 570 + 570 + 567 + 567 + 573 + 570 \\
 &\quad + 570 + 573 + 573 + 570 + 570 + 567 + 570 + 567 + 570 \\
 &= 11.399
 \end{aligned}$$

Berdasarkan sebaran nilai TP, FP, FN dan TN yang telah diperoleh, maka selanjutnya dilakukan perhitungan terkait dengan nilai *Accuracy*, *Recall*, *Precision*

dan *Recall*. Nilai *Accuracy* merupakan nilai yang menyatakan seberapa baik model menghasilkan prediksi yang benar terhadap data kata *Harah Jawoe* yang diujikan. Berikut persamaan untuk menentukan nilai *Accuracy*.

$$Accuracy = \frac{TP+TN}{TP+FP+FN+TN} \times 100\% \dots\dots\dots (2)$$

Sehingga, nilai *Accuracy* yang diperoleh adalah:

$$Accuracy = \frac{593 + 11.399}{593 + 11.399 + 7 + 7} \times 100\%$$

$$Accuracy = \frac{11.416}{11.430} \times 100\%$$

$$Accuracy = 0.9988 \times 100\%$$

$$Accuracy = 99.88\%$$

Selain itu, berdasarkan sebaran nilai TP, FP, FN dan TN juga dilakukan perhitungan untuk mendapatkan nilai *Recall*. Perhitungan ini dimaksudkan untuk melihat sebaik apa model dapat mendeteksi *instance*/obyek positif, atau lebih sederhananya, seberapa banyak obyek dapat di deteksi oleh model. Berikut persamaan untuk menentukan nilai *Recall*.

$$Recall = \frac{TP}{TP+FN} \times 100\% \dots\dots\dots (3)$$

Sehingga, nilai *Recall* yang diperoleh adalah:

$$Recall = \frac{593}{593 + 7} \times 100\%$$

$$Recall = \frac{593}{600} \times 100\%$$

$$Recall = 0.9883 \times 100\%$$

$$Recall = 98.83\%$$

Berdasarkan pada sebaran nilai TP, FP, FN dan TN juga dilakukan perhitungan untuk mendapatkan nilai *Precision*. Perhitungan ini dimaksudkan untuk melihat seberapa baik model dapat memprediksi *instance/obyek* yang positif sebagai *instance* yang benar-benar positif. Berikut persamaan untuk menentukan nilai *Precision*.

$$Precision = \frac{TP}{TP+FP} \times 100\% \dots\dots\dots (4)$$

Sehingga, nilai *Precision* yang diperoleh adalah:

$$Precision = \frac{593}{593 + 7} \times 100\%$$

$$Precision = \frac{593}{600} \times 100\%$$

$$Precision = 0.9883 \times 100\%$$

$$Precision = 98.83\%$$

Terakhir, berdasarkan hasil perhitungan *Recall* dan *Precision* sebelumnya, dilakukan perhitungan untuk menentukan nilai *F1 Score* dengan persamaan sebagai berikut:

$$F1\ Score = 2 \times \frac{Precision \times Recall}{Precision+Recall} \times 100\% \dots\dots\dots (5)$$

Sehingga, nilai *F1 Score* yang diperoleh adalah:

$$F1\ Score = 2 \times \frac{0.9883 \times 0.9883}{0.9883 + 0.9883} \times 100\%$$

$$F1\ Score = 2 \times \frac{1.9539}{1.9766} \times 100\%$$

$$F1\ Score = 0.9883 \times 100\%$$

$$F1\ Score = 98.83\%$$

Setelah keseluruhan perhitungan dilakukan, maka diperoleh sebaran nilai seperti tertera pada Tabel 4.12 berikut ini.

Tabel 4.12 *Accuracy, Precision, Recall dan F1 Score* pada Model *Small* 30 Epoch

Metrik Evaluasi	%
<i>Accuracy</i>	99.88
<i>Precision</i>	98.83
<i>Recall</i>	98.83
<i>F1-Score</i>	98.83

Jika dilihat dari sebaran nilai metrik evaluasi yang ditunjukkan oleh Tabel 4.12. Dapat disimpulkan bahwa model *small* untuk 30 epoch menunjukkan performa yang sangat baik dengan nilai *accuracy* sebesar 99.88%, *precision* sebesar 98.83%, *recall* sebesar 98.83%, dan *f1-score* sebesar 98.83%.

D2. Model *Big* 30 Epoch

Apabila dilihat dari *confusion matrix* yang dihasilkan selama proses pelatihan model *big* untuk jumlah *epoch* sebanyak 30, dapat dievaluasi kinerja model dengan menggunakan metrik seperti *recall*, *precision*, *accuracy*, dan *f1-score*. Berdasarkan *Confusion matrix* dapat dilihat pula sebaran nilai *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN).

Tabel 4.13 Nilai TP Setiap Label pada Model *Big* 30 Epoch

Label	Nilai TP
Adat	30
Bak	33
Bijaksana	27
Dipat	27
Gadöh	27
Gaséh	30
Hana	30
Hoka	33
Jak	33
Jikeumeung	27

Label	Nilai TP
Jimarit	30
Jipajöh	30
Jipeugah	27
Nibak	27
Pikir	30
Sibak	30
Singoh	33
Tamông	30
Tapinah	33
Ubak	30

Hasil evaluasi menunjukkan bahwa nilai TP untuk masing-masing label adalah sebagai berikut: 'Adat' sebanyak 30, 'Bak' sebanyak 33, 'Bijaksana' sebanyak 27, 'Dipat' sebanyak 27, 'Gadöh' sebanyak 27, 'Gaséh' sebanyak 30, 'Hana' sebanyak 30, 'Hoka' sebanyak 33, 'Jak' sebanyak 33, 'Jikeumeung' sebanyak 27, 'Jimarit' sebanyak 30, 'Jipajöh' sebanyak 30, 'Jipeugah' sebanyak 27, 'Nibak' sebanyak 27, 'Pikir' sebanyak 30, 'Sibak' sebanyak 30, 'Singoh' sebanyak 33, 'Tamông' sebanyak 30, 'Tapinah' sebanyak 33, dan 'Ubak' sebanyak 30. Secara keseluruhan, jumlah TP adalah 597.

$$\begin{aligned}
 \text{Overall TP} &= 30 + 33 + 27 + 27 + 27 + 30 + 30 + 33 + 33 + 27 + 30 + 30 + 27 + 27 + \\
 &30 + 30 + 33 + 30 + 33 + 30 \\
 &= 597
 \end{aligned}$$

Jika dilihat dari nilai FN, diketahui bahwa hanya label 'Adat' yang memiliki FN sebanyak 3, sementara label lainnya memiliki nilai FN sebesar 0. Sehingga Total nilai FN keseluruhan adalah 3, seperti tertera pada tabel 4.14.

Tabel 4.14 Nilai FN Setiap Label pada Model *Big 30 Epoch*

Label	Nilai FN	Label	Nilai FN
Adat	3	Jimarit	0
Bak	0	Jipajöh	0
Bijaksana	0	Jipeugah	0
Dipat	0	Nibak	0
Gadöh	0	Pikir	0
Gaséh	0	Sibak	0
Hana	0	Singoh	0
Hoka	0	Tamông	0
Jak	0	Tapinah	0
Jikeumeung	0	Ubak	0

Oleh sebab itu maka, disimpulkan bahwa nilai *Overall* FN adalah 3. Sementara itu, terlihat bahwa nilai FP menunjukkan bahwa hanya label 'Gadöh' yang memiliki FP sebanyak 3, sementara label lainnya memiliki nilai FP sebesar 0. Sehingga, total nilai FP keseluruhan adalah 3.

Tabel 4.15 Nilai FP Setiap Label pada Model *Big 30 Epoch*

Label	Nilai FP	Label	Nilai FP
Adat	0	Jimarit	0
Bak	0	Jipajöh	0
Bijaksana	0	Jipeugah	0
Dipat	0	Nibak	0
Gadöh	3	Pikir	0
Gaséh	0	Sibak	0
Hana	0	Singoh	0

Hoka	0
Jak	0
Jikeumeung	0

Tamông	0
Tapinah	0
Ubak	0

Berdasarkan pada nilai TP, FP dan FN yang telah diperoleh sebelumnya, maka dihitung nilai TN dengan cara mengurangi nilai total data dengan jumlah nilai TP, FP dan FN dengan persamaan sebagai berikut:

$$\text{True Negative (TN)} = \text{Total} - (\text{TP} + \text{FP} + \text{FN}) \dots\dots\dots (6)$$

Sehingga dari hasil perhitungan persamaan (1) di atas, diperoleh hasil TN seperti yang tertera pada tabel 4.16 berikut:

Tabel 4.16 Nilai TN Setiap Label pada Model *Big 30 Epoch*

Label	Nilai TN
Adat	567
Bak	567
Bijaksana	573
Dipat	573
Gadöh	570
Gaséh	570
Hana	570
Hoka	567
Jak	567
Jikeumeung	573

Label	Nilai TN
Jimarit	570
Jipajöh	570
Jipeugah	573
Nibak	573
Pikir	570
Sibak	570
Singoh	567
Tamông	570
Tapinah	567
Ubak	570

Berdasarkan pada tabel 4.16 di atas, diketahui bahwa nilai TN untuk masing-masing label adalah sebagai berikut: 'Adat' sebanyak 567, 'Bak' sebanyak 567, 'Bijaksana' sebanyak 573, 'Dipat' sebanyak 573, 'Gadöh' sebanyak 570, 'Gaséh'

sebanyak 570, 'Hana' sebanyak 570, 'Hoka' sebanyak 567, 'Jak' sebanyak 567, 'Jikeumeung' sebanyak 573, 'Jimarit' sebanyak 570, 'Jipajôh' sebanyak 570, 'Jipeugah' sebanyak 573, 'Nibak' sebanyak 573, 'Pikir' sebanyak 570, 'Sibak' sebanyak 570, 'Singoh' sebanyak 567, 'Tamöng' sebanyak 570, 'Tapinah' sebanyak 567, dan 'Ubak' sebanyak 570. Total nilai TN keseluruhan adalah 11.403.

$$\begin{aligned} \text{Overall TN} &= 567 + 567 + 573 + 573 + 570 + 570 + 570 + 567 + 567 + 573 + 570 + \\ &570 + 573 + 573 + 570 + 570 + 567 + 570 + 567 + 570 \\ &= 11.403 \end{aligned}$$

Berdasarkan sebaran nilai TP, FP, FN dan TN yang telah diperoleh, maka selanjutnya dilakukan perhitungan terkait dengan nilai *Accuracy*, *Recall*, *Precision* dan *Recall*. Nilai *Accuracy* merupakan nilai yang menyatakan seberapa baik model menghasilkan prediksi yang benar terhadap data kata *Harah Jawoe* yang diujikan. Berikut persamaan untuk menentukan nilai *Accuracy*.

$$Accuracy = \frac{TP+TN}{TP+FP+FN+TN} \times 100\% \dots\dots\dots (7)$$

Sehingga, nilai *Accuracy* yang diperoleh adalah:

$$Accuracy = \frac{597 + 11.403}{597 + 11.403 + 3 + 3} \times 100\%$$

$$Accuracy = \frac{12.000}{12.006} \times 100\%$$

$$Accuracy = 0.9995 \times 100\%$$

$$Accuracy = 99.95\%$$

Selain itu, perhitungan nilai *Recall* juga dilakukan berdasarkan sebaran nilai *True Positive* (TP), *False Positive* (FP), *False Negative* (FN), dan *True Negative* (TN). Tujuan dari perhitungan ini adalah untuk mengevaluasi kemampuan model dalam mendeteksi *instance* atau objek positif. Dengan kata lain, nilai *Recall* menunjukkan seberapa banyak objek yang dapat dideteksi oleh model. Persamaan untuk menentukan nilai *Recall* adalah sebagai berikut.

$$Recall = \frac{TP}{TP+FN} \times 100\% \dots\dots\dots (8)$$

Dengan demikian, nilai *Recall* yang diperoleh adalah sebagai berikut:

$$Recall = \frac{597}{597 + 3} \times 100\%$$

$$Recall = \frac{597}{597 + 3} \times 100\%$$

$$Recall = \frac{597}{600} \times 100\%$$

$$Recall = 0.995 \times 100\%$$

$$Recall = 99.5\%$$

Perhitungan nilai *Precision* juga dilakukan berdasarkan sebaran nilai *True Positive* (TP), *False Positive* (FP), *False Negative* (FN), dan *True Negative* (TN). Perhitungan ini bertujuan untuk mengevaluasi seberapa akurat model dalam memprediksi *instance* atau objek positif sebagai *instance* yang benar-benar positif. Berikut adalah persamaan untuk menentukan nilai *Precision*.

$$Precision = \frac{TP}{TP+FP} \times 100\% \dots\dots\dots (9)$$

Sehingga dihasilkan nilai *precision* adalah sebagai berikut:

$$Precision = \frac{597}{597 + 3} \times 100\%$$

$$Precision = \frac{597}{600} \times 100\%$$

$$Precision = 0.995 \times 100\%$$

$$Precision = 99.5\%$$

Terakhir, berdasarkan hasil perhitungan *Recall* dan *Precision* yang telah dilakukan sebelumnya, nilai *F1 Score* dapat ditentukan dengan menggunakan persamaan sebagai berikut:

$$F1\ Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \times 100\% \dots\dots\dots (10)$$

Dengan demikian, nilai *F1 Score* yang diperoleh adalah sebagai berikut:

$$F1\ Score = 2 \times \frac{0.995 \times 0.995}{0.995 + 0.995} \times 100\%$$

$$F1\ Score = 2 \times \frac{1.990025}{1.99} \times 100\%$$

$$F1\ Score = 0.995 \times 100\%$$

$$F1\ Score = 99.5\%$$

Setelah keseluruhan perhitungan dilakukan, sebaran nilai yang diperoleh dapat dilihat pada Tabel 4.17 berikut ini.

Tabel 4.17 Accuracy, Precision, Recall dan F1 Score pada Model Big 30 Epoch

Metriks Evaluasi	%
Accuracy	99.95
Precision	99.5

<i>Recall</i>	99.5
<i>F1-Score</i>	99.5

Berdasarkan perhitungan metrik evaluasi pada Tabel 4.17, terlihat bahwa model *big* untuk 30 *epoch* memperlihatkan performa yang sangat baik dengan nilai akurasi sebesar 99.95%, *precision* sebesar 99.5%, *recall* sebesar 99.5%, dan *F1-score* sebesar 99.5%.

E. Pembahasan Hasil Penelitian

Hasil penelitian ini menunjukkan bahwa penggunaan arsitektur YOLOv8 dalam mendeteksi gambar kata huruf Arab Melayu Aceh (*harah jawoe*) memberikan hasil yang signifikan dalam hal akurasi. Berdasarkan pengujian yang dilakukan, model *small* mencapai nilai *Mean Average Precision* (mAP50-95) tertinggi sebesar 0.744 (74.4%) pada *epoch* ke-30. Namun, performa model mengalami penurunan setelah *epoch* ke-30, dengan nilai mAP menurun menjadi 0.704 (70.4%) pada *epoch* ke-90 sebelum sedikit meningkat kembali pada *epoch* ke-100 dengan nilai 0.707 (70.7%). Fluktuasi kinerja model ini menunjukkan adanya kebutuhan untuk penyesuaian lebih lanjut pada parameter pelatihan agar dapat mempertahankan atau meningkatkan kinerja pada *epoch* yang lebih tinggi.

Model *medium* menunjukkan hasil yang serupa, dengan nilai mAP tertinggi sebesar 0.722 (72.2%) pada *epoch* ke-30. Setelah itu, kinerja model sedikit menurun dan fluktuatif, mencapai nilai mAP sebesar 0.711 (71.1%) pada *epoch* ke-100. Model normal juga memperlihatkan pola yang sama, dengan nilai mAP tertinggi sebesar 0.723 (72.3%) pada *epoch* ke-30, yang kemudian menurun menjadi 0.704 (70.4%) pada *epoch* ke-90 sebelum meningkat sedikit pada *epoch* ke-100 dengan nilai mAP sebesar 0.715 (71.5%).

Model *big* menunjukkan konsistensi yang lebih baik dibandingkan model lainnya, dengan nilai mAP tertinggi sebesar 0.724 (72.4%) pada *epoch* ke-30 dan ke-100. Kinerja model ini relatif stabil di antara *epoch-epoch* tersebut, menunjukkan bahwa model *big* memiliki kemampuan yang lebih baik dalam mendeteksi gambar *Harah Jawoe* dibandingkan dengan model *small*, *medium*, dan *normal*.

Hasil penelitian ini sejalan dengan berbagai penelitian terdahulu yang menunjukkan efektivitas arsitektur YOLO dan CNN dalam pendeteksian citra. Penelitian Khairunnas dkk (2021a) dan Tamang dkk (2023) menitikberatkan pada peningkatan akurasi melalui *tuning* jumlah *epoch*, yang bertujuan untuk meningkatkan keterbacaan arsitektur terhadap dataset. Jumlah dataset, ukuran gambar, dan alat anotasi juga berperan penting dalam meningkatkan akurasi pendeteksian, sebagaimana disampaikan oleh Rukundo (2021), Orlando dkk (2022), dan Rebinth dkk (2019).

Convolutional Neural Network (CNN), sebagai salah satu metode *deep learning* yang umum digunakan dalam teknologi *computer vision*, telah terbukti efektif. Penelitian Adi Nugroho dkk (2020) menunjukkan bahwa CNN mencapai akurasi sebesar 91% setelah *epoch* terakhir diproses, dengan jumlah dataset *training* dan *testing* sebanyak 13.440 dan 3.360. Penelitian Rachmawanto dan Andono (2022) juga menunjukkan akurasi sebesar 86.5% setelah peningkatan *epoch* sebanyak 200 kali dengan jumlah dataset 1000 data. Penelitian lain oleh Rahmawati dkk. (2021) menunjukkan bahwa akurasi pendeteksian aksara Sunda mencapai 92%, sementara penelitian Anggraini (2020) menunjukkan bahwa CNN dapat mencapai akurasi sebesar 92% dengan hanya menggunakan 250 citra latih dan 50 citra uji. Kajian oleh Sukma Hanindria dan Hendry (2022) juga membuktikan bahwa jumlah *epoch* yang lebih sedikit, yakni 50 kali, masih dapat menghasilkan tingkat akurasi yang cukup baik sebesar 85%.

Sebagai bagian dari metode CNN, arsitektur YOLO sangat mungkin diaplikasikan dalam pendeteksian citra. Penelitian Muhammad Nur dan Arnisa Stefanie (2023) menunjukkan bahwa YOLO dapat mencapai akurasi 92% dalam mendeteksi citra fundus mata, sementara Bernard Alaric Sudjono dkk (2023) mencapai akurasi 73% dalam deteksi video *real-time*. Hal ini menunjukkan bahwa penggunaan arsitektur YOLO sangat efektif dalam berbagai aplikasi pendeteksian citra. Penelitian Aesha Durratul Nasihah (2023) menunjukkan bahwa arsitektur Xception yang digunakan untuk mendeteksi kata harah jawoe mencapai akurasi 67.60% setelah melalui 90 *epoch*, menunjukkan kebutuhan variasi data yang lebih luas untuk mencapai tingkat akuratan yang lebih tinggi.

Penelitian Abdullah Sani dan Suci Rahmadinni (2022) menunjukkan bahwa manipulasi *background* citra pada dataset dapat meningkatkan akurasi pendeteksian citra menggunakan arsitektur YOLO. Dengan melatari *background* citra dengan tiga warna yaitu biru, putih, dan kuning, penelitian ini berhasil mencapai akurasi sebesar 97% untuk *background* putih, 98% untuk *background* biru, dan 98% untuk *background* kuning.

Hasil penelitian ini mendukung temuan-temuan sebelumnya mengenai efektivitas arsitektur YOLO dan CNN dalam pendeteksian citra. Model *big* menunjukkan performa terbaik dan konsistensi yang baik dalam mendeteksi gambar *Harah Jawoe* pada berbagai ukuran gambar dan *epoch*. Meskipun model *small* menunjukkan nilai mAP tertinggi pada *epoch* ke-30, penurunan kinerja setelah *epoch* tersebut menunjukkan perlunya penyesuaian lebih lanjut terhadap parameter pelatihan dan teknik augmentasi data.

Analisis *confusion matrix* menunjukkan bahwa model *small* pada *epoch* ke-30 mampu melakukan prediksi yang cukup baik, meskipun terdapat beberapa kesalahan dalam membedakan antara label 'Adat' dan 'Gadöh' serta data

background. Model big pada *epoch* ke-30 menunjukkan performa yang lebih baik, dengan sedikit kesalahan prediksi dan kemampuan yang baik dalam membedakan label-label tersebut.

Secara keseluruhan, penelitian ini menunjukkan bahwa arsitektur YOLOv8 dapat diaplikasikan secara efektif dalam mendeteksi gambar kata *Harah Jawoe* dengan akurasi yang tinggi, terutama pada *epoch* ke-30. Namun, untuk mencapai kinerja yang lebih konsisten pada *epoch* yang lebih tinggi, diperlukan penyesuaian lebih lanjut terhadap parameter pelatihan dan teknik augmentasi data. Saran untuk penelitian selanjutnya adalah melakukan eksplorasi lebih lanjut terhadap teknik augmentasi data dan parameter pelatihan untuk meningkatkan akurasi deteksi pada *epoch* yang lebih tinggi. Selain itu, studi lebih mendalam mengenai penggunaan berbagai arsitektur *deep learning* lainnya juga dapat dilakukan untuk membandingkan kinerja dan efektivitasnya dalam mendeteksi gambar kata *harah jawoe*.

Hasil deteksi oleh model yang diuji dapat dilihat pada bagian lampiran. Lampiran ini berisi sebagian gambar hasil deteksi dengan model YOLOv8 pada berbagai *epoch*, termasuk *epoch* ke-30, ke-50, ke-70, ke-90, dan ke-100. Gambar-gambar ini dapat menunjukkan model dalam melakukan prediksi terhadap data dan memperlihatkan bagaimana performa model berubah seiring dengan bertambahnya jumlah *epoch* dan memahami fluktuasi nilai *Mean Average Precision* (mAP) yang telah dijelaskan.

BAB 5

KESIMPULAN DAN SARAN

A. Kesimpulan

Berdasarkan hasil temuan penelitian diketahui bahwa arsitektur YOLOv8 memiliki potensi besar dalam mendeteksi gambar kata huruf Arab Melayu Aceh (harah jawoe) dengan akurasi yang tinggi. Model YOLOv8 *small* mencapai nilai *Mean Average Precision* (mAP50-95) tertinggi sebesar 74.4% pada epoch ke-30, namun mengalami penurunan performa pada epoch selanjutnya sebelum sedikit meningkat kembali pada epoch ke-100. Pola yang sama juga terlihat pada model medium dan normal. Model big menunjukkan performa yang lebih stabil dengan nilai mAP tertinggi sebesar 72.4% pada epoch ke-30 dan ke-100, mengindikasikan bahwa model ini lebih baik dalam mendeteksi gambar harah jawoe dibandingkan model lainnya.

Hasil ini sejalan dengan penelitian sebelumnya yang menekankan pentingnya tuning jumlah epoch dan parameter pelatihan untuk meningkatkan akurasi deteksi. Kajian literatur menunjukkan bahwa penggunaan arsitektur YOLO dan CNN secara umum efektif dalam pendeteksian citra, dengan berbagai studi menunjukkan akurasi yang bervariasi tergantung pada jumlah dataset, ukuran gambar, dan teknik augmentasi data yang digunakan.

Kesimpulan utama dari penelitian ini adalah bahwa arsitektur YOLOv8 dapat diterapkan secara efektif dalam pendeteksian gambar kata harah jawoe, dengan model *big* menunjukkan performa dan stabilitas terbaik di antara model-model yang diuji. Meskipun demikian, fluktuasi performa pada epoch yang lebih

tinggi menunjukkan perlunya penyesuaian lebih lanjut terhadap parameter pelatihan untuk mencapai kinerja yang lebih konsisten.

B. Saran

Untuk penelitian selanjutnya, disarankan untuk melakukan eksplorasi lebih lanjut terhadap teknik augmentasi data dan parameter pelatihan untuk meningkatkan akurasi deteksi pada epoch yang lebih tinggi. Teknik augmentasi data seperti variasi background dan manipulasi citra dapat diuji lebih lanjut untuk melihat dampaknya terhadap akurasi model. Selain itu, studi lebih mendalam mengenai penggunaan berbagai arsitektur *deep learning* lainnya dapat dilakukan untuk membandingkan kinerja dan efektivitasnya dalam mendeteksi gambar kata harah jawoe. Eksperimen tambahan dengan jumlah dataset yang lebih besar dan variasi gambar yang lebih luas juga perlu dilakukan untuk mendapatkan hasil yang lebih komprehensif.

Penelitian ini juga menunjukkan bahwa model *big* memiliki potensi untuk diaplikasikan dalam pendeteksian citra dengan akurasi yang tinggi, namun diperlukan pengujian lebih lanjut untuk memastikan kestabilan performanya. Peneliti juga disarankan untuk melakukan tuning terhadap parameter pelatihan seperti *learning rate*, *batch size*, dan jumlah epoch untuk mencapai kinerja optimal. Secara keseluruhan, hasil penelitian ini memberikan kontribusi signifikan dalam bidang pendeteksian citra menggunakan arsitektur YOLOv8, dan diharapkan dapat menjadi dasar untuk penelitian lanjutan yang lebih mendalam dan komprehensif dalam bidang ini.

DAFTAR PUSTAKA

- Abdul Ghani, R., Shanudin Zakaria, M., & Omar, K. (2009). Jawi-Malay Transliteration. *International Conference On Electrical Engineering And Informatics*, 154–157.
- Adi Nugroho, P., Fenriana, I., & Arijanto, R. (2020). Implementasi Deep Learning Menggunakan Convolutional Neural Network (Cnn) Pada Ekspresi Manusia. *Jurnal Algor*, 2(1). [Https://jurnal.buddhidharma.ac.id/index.php/algor/index](https://jurnal.buddhidharma.ac.id/index.php/algor/index)
- Afrianto, I., Riyanda, R., & Atin, S. (2018). Implementasi Algoritma Freeman Chain Code Dan Support Vector Machine (Svm) Pada Identifikasi Aksara Arab Melayu. In *Seminar Nasional Aplikasi Teknologi Informasi (Snati)*.
- Akil, I., & Chaidir, I. (2021). Inti Nusa Mandiri Deteksi Karakter Huruf Arab Dengan Menggunakan Convolutional Neural Network. *Inti Nusa Mandiri*, 15(2), 183–188. [Https://doi.org/10.33480/inti.v15i2.xxxx](https://doi.org/10.33480/inti.v15i2.xxxx)
- Anggraini, W. (2020). *Deep Learning Untuk Deteksi Wajah Yang Berhijab Menggunakan Algoritma Convolutional Neural Network (Cnn) Dengan Tensorflow*. [Https://repository.ar-raniry.ac.id/id/eprint/13513/](https://repository.ar-raniry.ac.id/id/eprint/13513/)
- Anhar, A., & Putra, R. A. (2023). Perancangan Dan Implementasi Self-Checkout System Pada Toko Ritel Menggunakan Convolutional Neural Network (Cnn). *Elkomika: Jurnal Teknik Energi Elektrik, Teknik Telekomunikasi, & Teknik Elektronika*, 11(2), 466. [Https://doi.org/10.26760/elkomika.v11i2.466](https://doi.org/10.26760/elkomika.v11i2.466)
- Arifin Nugraha, Z. (2023). Manuscripts And Philological Studies In Indonesia. *Culturalistics: Journal Of Cultural, Literary, And Linguistic Studies*, 7(1), 52–57. [Http://ejournal.undip.ac.id/index.php/culturalistics52](http://ejournal.undip.ac.id/index.php/culturalistics52)
- Atika Sari, C., & Shinta Sari, W. (2022). Pengaruh Linear Binary Pattern (Lbp) Dalam Pengenalan Citra Aksara Jawa Berbasis Optical Character Recognition (Ocr). *2 St Proceeding Stekom, 2022*.
- Barokah Asmarahman Takarob. (2022). *Deteksidan Identifikasi Rambu-Rambu Lalu Lintasberbasis Algoritma You Only Look Once (Yolo)* [Tugas Akhir, Universitas Muhammadiyah]. [Http://repository.umj.ac.id/11987/](http://repository.umj.ac.id/11987/)

- Daniels, S. (2020). *Pengenalan Bahasa Isyarat Pada Data Video Menggunakan Metode Cnn Dengan Arsitektur Yolo*.
- Ellyadi, M. (2022a). *Deteksi Tajwid Nun Mati Pada Ayat Al-Quran Dengan Metode Convolutional Neural Network Menggunakan Model Training Ssd Mobilenet*. <https://Repository.Ar-Raniry.Ac.Id/Id/Eprint/29791/>
- Ellyadi, M. (2022b). *Deteksi Tajwid Nun Mati Pada Ayat Al-Quran Dengan Metode Convolutional Neural Network Menggunakan Model Training Ssd Mobilenet*.
- Harun, A., & Brilian Kharisma, O. (2023). *Jurnal Media Informatika Budidarma Implementasi Deep Learning Menggunakan Metode You Only Look Once Untuk Mendeteksi Rokok*. <https://doi.org/10.30865/Mib.V7i1.5409>
- Hermansyah. (2015). *Manuskrip Aceh: Revitalisasi Kearifan Masyarakat Aceh Diera Global*. *Wacana Etnik*, 5(1), 25–36.
- Hermansyah. (2020). *Perspektif Nilai Sejarah Naskah Hikayat Aceh*. *Indonesian Journal Of Islamic History And Culture*, 1(2), 138–146.
- Hidayatullah, P. (2021). *Buku Sakti Deep Learning* (S. Vision, Ed.; 2nd Ed., Vol. 2). Stunning Vision Ai Academy.
- Khairunnas, Mulyanto Yuniarno, E., & Zaini, A. (2021). *Pembuatan Modul Deteksi Objek Manusia Menggunakan Metode Yolo Untuk Mobile Robot*. *Jurnal Teknik Its*, 10(1), 50–55.
- Kurniawan, A. A., Mustikasari, M., & Korespondensi, P. (2022a). *Evaluasi Kinerja Mllib Apache Spark Pada Klasifikasi Berita Palsu Dalam Bahasa Indonesia*. 9(3). <https://doi.org/10.25126/Jtiik.202293538>
- Kurniawan, A. A., Mustikasari, M., & Korespondensi, P. (2022b). *Evaluasi Kinerja Mllib Apache Spark Pada Klasifikasi Berita Palsu Dalam Bahasa Indonesia*. 9(3). <https://doi.org/10.25126/Jtiik.202293538>
- Lina, Q. (2019, January). *Apa Itu Convolutional Neural Network?* Medium. <https://medium.com/@16611110/apa-itu-convolutional-neural-network-836f70b193a4>
- Lorentius, C. A., Adipranata, R., & Tjondrowiguno, A. (2019). *Pengenalan Aksara Jawa Dengan Menggunakan Metode Convolutional Neural Network*. *Jurnal Infra*, 7(1), 221–227. <https://publication.petra.ac.id/index.php/teknik-informatika/article/view/8075/7285>

- Mandasari, E. (2022). The Role Of The Manuscript Collector In Banda Aceh. *Indonesian Journal Of Islamic History And Culture*, 3(1), 43–79.
- Marcella, D., & Devella, S. (2022). *Klasifikasi Penyakit Mata Menggunakan Convolutional Neural Network Dengan Arsitektur Vgg-19*. 3(1), 60–70.
- Nasihah, A. D. (2023). *Transliterasi Aksara Jawoe-Latin Menggunakan Metode Convolutional Neural Network Dengan Arsitektur Xception* (Doctoral dissertation, Universitas Islam Negeri Ar-Raniry).
- Nufus, N., Ariffin, D. M., Satyawan, A. S., Nugraha, R. A. S., Asyasyakuur, M. I., Marlina, N. N. A., Parangin, C. H., & Ema, E. (2021). Sistem Pendeteksi Pejalan Kaki Di Lingkungan Terbatas Berbasis Ssd Mobilenet V2 Dengan Menggunakan Gambar 360° Ternormalisasi. *Prosiding Seminar Nasional Sains Teknologi Dan Inovasi Indonesia (Senastindo)*, 3, 123–134. <https://doi.org/10.54706/Senastindo.V3.2021.123>
- Nur, M., Muhlashin, I., Stefanie, A., Universitas, S., Karawang, J. H., Ronggo, W., & Karawang, I. (2023). Klasifikasi Penyakit Mata Berdasarkan Citra Fundus Menggunakan Yolo V8. In *Jurnal Mahasiswa Teknik Informatika* (Vol. 7, Issue 2).
- Nurrahma Kuswati, S. (2021). Kegiatan Digitalisasi Naskah Kuno Sebagai Upayadiseminasi Informas. *Libria*, 13(1), 107–129.
- Nurwakit, A., Utami, E., & Al Fatta, H. (2020). Transliterasi Huruf Arab Pegon-Latin Menggunakan Eigenspace. *Saintekbu*, 12(2), 58–67.
- Orlando, N., Gyacskov, I., Gillies, D. J., Guo, F., Romagnoli, C., D'souza, D., Cool, D. W., Hoover, D. A., & Fenster, A. (2022). Effect Of Dataset Size, Image Quality, And Image Type On Deep Learning-Based Automatic Prostate Segmentation In 3d Ultrasound. *Physics In Medicine And Biology*, 67(7). <https://doi.org/10.1088/1361-6560/AC5a93>
- Rachmawanto, E. H., & Andono, P. N. (2022a). Deteksi Karakter Hiragana Menggunakan Metode Convolutional Neural Network. *Jurnal Nasional Pendidikan Teknik Informatika (Janapati)*, 11(3), 183–191. <https://doi.org/10.23887/Janapati.V11i3.50144>
- Rachmawanto, E. H., & Andono, P. N. (2022b). Deteksi Karakter Hiragana Menggunakan Metode Convolutional Neural Network. *Jurnal Nasional*

Pendidikan Teknik Informatika (Janapati), 11(3), 183–191.
<https://doi.org/10.23887/Janapati.V11i3.50144>

Rahmawati, S. N., Hidayat, E. W., & Mubarak, H. (2021). Implementasi Deep Learning Pada Pengenalan Aksara Sunda Menggunakan Metode Convolutional Neural Network. *Insert: Information System And Emerging Technology Journal*, 2(1).

Ramala, D. E. (2020). Aksara Jawi: Warisan Budaya Dan Bahasa Masyarakat Alam Melayu Dalam Tinjauan Sosiolingustik. In *Jurnal Islamika* (Vol. 3, Issue 2).

Rangeking. (2023, January 10). *Brief Summary Of Yolov8 Model Structure #189*. Github. <https://github.com/Ultralytics/Ultralytics/issues/189>

Rebinth, A., Kumar, M., & Mohan Kumar, S. (2019). Importance Of Manual Image Annotation Tools And Free Datasets For Medical Research. In *Jour Of Adv Research In Dynamical & Control Systems* (Vol. 11). <https://www.researchgate.net/publication/340949752>

Rukundo, O. (2021). Effects Of Image Size On Deep Learning. *Arxiv - Cornell University*.
https://www.bing.com/search?q=1articleeffects+of+image+size+on+deep+learning+olivier+rukundo&cvid=08afc22558c64f749eeeb38cd1fa8cf4&gs_lcrp=egzjahjvbwuybggaeuyotihcae6wcyqnbzcxnwowajsoagcwaga&form=anab01&pc=hcts

Sabri, A. (2022). *Analisis Perbandingan Model Arsitektur Cnn dalam Pendeteksian Coronavirusdiseasemenggunakan Citra X-Rayparu-Paru*. <https://repository.ar-raniry.ac.id/id/eprint/27444/>

Sani, A., & Rahmadinni, S. (2022a). Deteksi Gestur Tangan Berbasis Pengolahan Citra. *Jurnal Rekayasa Elektrika*, 18(2).
<https://doi.org/10.17529/jre.v18i2.25147>

Sani, A., & Rahmadinni, S. (2022b). Deteksi Gestur Tangan Berbasis Pengolahan Citra. *Jurnal Rekayasa Elektrika*, 18(2).
<https://doi.org/10.17529/jre.v18i2.25147>

Sena, S. (2017, November). *Pengenalan Deep Learning Part 7: Convolutional Neural Network (Cnn)*. Medium.
<https://medium.com/@samuelsena/pengenalan-deep-learning-part-7-convolutional-neural-network-cnn-b003b477dc94>

- Shianto, K. A., Gunadi, K., & Setyati, E. (2019). Deteksi Jenis Mobil Menggunakan Metode Yolo Dan Faster R-Cnn. *Infra*, 7(1). <https://Publication.Petra.Ac.Id/Index.Php/Teknik-Informatika/Article/View/8065>
- Sudjono, B. A., Munadi, M., & Setiawan, J. D. (2023a). Aplikasi Model Object Detection Pada Ayam Broiler Berbasis Deep Learning Pada Kandang Closed House. In *Jurnal Teknik Mesin S-1* (Vol. 11, Issue 2).
- Sudjono, B. A., Munadi, M., & Setiawan, J. D. (2023b). Aplikasi Model Object Detection Pada Ayam Broiler Berbasis Deep Learning Pada Kandang Closed House. In *Jurnal Teknik Mesin S-1* (Vol. 11, Issue 2).
- Sukma Hanindria, I., & Hendry. (2022). Pengklasifikasian Aksara Jawa Metode Convolutional Neural Network. *Jurnal Teknik Informatika Dan Ssitem Informasi*, 9(3), 2727-2737. <https://Jurnal.Mdp.Ac.Id/Index.Php/Jatisi/Article/View/2177>
- Tamang, S., Sen, B., Pradhan, A., Sharma, K., & Singh, V. K. (2023). International Journal Of Intelligent Systems And Applications In Engineering Enhancing Covid-19 Safety: Exploring Yolov8 Object Detection For Accurate Face Mask Classification. *Original Research Paper International Journal Of Intelligent Systems And Applications In Engineering Ijisaee*, 2023(2), 892-897. www.ijisaee.org
- Tengku, O. :, & Sakti, A. (2011). Perkembangan Dan Pelestarian Manuskrip Arab Melayu Di Aceh. *Jurnal Sejarah Citra Lekha*, 16(2), 19-30.
- Van Langen, K. F. H. (1889). *Atjehsche Taal* (M. Nijhoff, Ed.). Koninklijk Instituut.
- Wan, C., Bahri, S., Ahmad, C. W., Nasrudin, M. F., & Murah, M. Z. (2013). *Machine Transliteration Design For Old Malay Manuscript*. <https://www.researchgate.net/publication/236970180>
- Widiarti, A. R., Pulungan, R., Harjoko, A., Marsono, & Hartati, S. (2018). A Proposed Model For Javanese Manuscript Images Transliteration. *Journal Of Physics: Conference Series*, 1098(1). <https://doi.org/10.1088/1742-6596/1098/1/012014>
- Yanto, B., Herawan Hayadi, B., & Pasir Pengaraian, U. (2020). Indentifikasi Pola Aksara Arab Melayu Dengan Jaringan Syaraf Tiruan Convolutional Neural

Network (Cnn). *Jsai : Journal Scientific And Applied Informatics*, 3(3), 106–114.
<https://doi.org/10.36085>

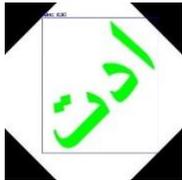
Yusra, Fikry, M., Yani, S. S., Irsyad, M., & Sanjaya, S. (2020). Aplikasi Android Untuk Alih Aksara Latin Ke Arab Melayu Dengan Pendekatan Berbasis Aturan. *Corelit*, 6(2), 117–121.
<https://doi.org/10.24014/corelit.v6i2.11645>

DAFTAR LAMPIRAN

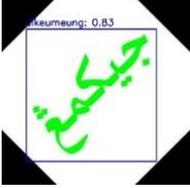
LAMPIRAN-LAMPIRAN

A. Lampiran 1. Hasil Pengujian Model *Small*

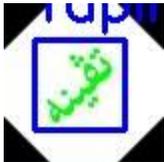
30 Epoch Model <i>Small</i>			
Kelas	Citra	Output	Status Identifikasi
Jipeugah		Jipeugah	Benar
Jipajöh		Jipajöh	Benar
Adat		Adat	Benar

30 Epoch Model <i>Small</i>			
Kelas	Citra	Output	Status Identifikasi
Adat		Adat	Benar
Adat		Adat	Benar

50 Epoch Model <i>Small</i>			
Kelas	Citra	Output	Status Identifikasi
Hana		Hana	Benar
Hana		Hana	Benar

50 Epoch Model <i>Small</i>			
Kelas	Citra	Output	Status Identifikasi
Hoka		Hoka	Benar
Jikeumeung		Jikeumeung	Benar
Jimarit		Jimarit	Benar

70 Epoch Model <i>Small</i>			
Kelas	Citra	Output	Status Identifikasi
Ubak		Ubak	Benar

70 Epoch Model <i>Small</i>			
Kelas	Citra	Output	Status Identifikasi
Ubak		Ubak	Benar
Ubak		Ubak	Benar
Ubak		Ubak	Benar
Tapinah		Tapinah	Benar

90 Epoch Model <i>Small</i>			
Kelas	Citra	Output	Status Identifikasi
Bijaksana		Bijaksana	Benar
Gaséh		Gaséh	Benar
Hana		Hana	Benar
Hana		Hana	Benar
Hana		Hana	Benar

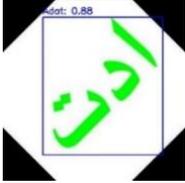
90 Epoch Model <i>Small</i>			
Kelas	Citra	Output	Status Identifikasi
Hoka		Hoka	Benar
Jikeumeung		Jikeumeung	Benar
Jikeumeung		Jikeumeung	Benar

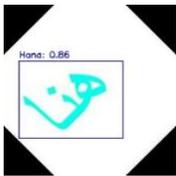
100 Epoch Model <i>Small</i>			
Kelas	Citra	Output	Status Identifikasi
Jipeugah		Jipeugah	Benar

100 Epoch Model <i>Small</i>			
Kelas	Citra	Output	Status Identifikasi
Jipeugah		Jipeugah	Benar
Jipeugah		Jipeugah	Benar
Nibak		Nibak	Benar
Nibak		Nibak	Benar
Pikir		Pikir	Benar

100 Epoch Model <i>Small</i>			
Kelas	Citra	Output	Status Identifikasi
Pikir		Pikir	Benar

B. Lampiran 2. Hasil Pengujian Model *Medium*

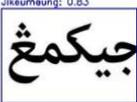
30 Epoch Model <i>Medium</i>			
Kelas	Citra	Output	Status Identifikasi
Adat		Adat	Benar
Adat		Adat	Benar
Adat		Adat	Benar

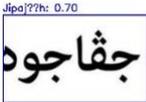
30 Epoch Model <i>Medium</i>			
Kelas	Citra	Output	Status Identifikasi
Gaséh		Gaséh	Benar
Hana		Hana	Benar
Hana		Hana	Benar
Jipeugah		Jipeugah	Benar

50 Epoch Model <i>Medium</i>			
Kelas	Citra	Output	Status Identifikasi
Hoka		Hoka	Benar
Hoka		Hoka	Benar
Gaséh		Gaséh	Benar
Hoka		Hoka	Benar
Jikeumeung		Jikeumeung	Benar

50 Epoch Model <i>Medium</i>			
Kelas	Citra	Output	Status Identifikasi
Jikeumeung		Jikeumeung	Benar
Jimarit		Jimarit	Benar

70 Epoch Model <i>Medium</i>			
Kelas	Citra	Output	Status Identifikasi
Hana		Hana	Benar
Nibak		Nibak	Benar

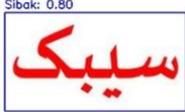
70 Epoch Model <i>Medium</i>			
Kelas	Citra	Output	Status Identifikasi
Nibak		Nibak	Benar
Nibak		Nibak	Benar
Jikeumeung		Jikeumeung	Benar
Jimarit		Jimarit	Benar
Jimarit		Jimarit	Benar

70 Epoch Model <i>Medium</i>			
Kelas	Citra	Output	Status Identifikasi
Pikir		Pikir	Benar
Jipajòh		Jipajòh	Benar
Jipajòh		Jipajòh	Benar
Singoh		Singoh	Benar

90 Epoch Model <i>Medium</i>			
Kelas	Citra	Output	Status Identifikasi
Jipeugah		Jipeugah	Benar
Tamöng		Tamöng	Benar
Jikeumeung		Jikeumeung	Benar
Dipat		Dipat	Benar
Bak		Bak	Benar

100 Epoch Model *Medium*

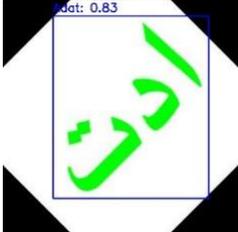
Kelas	Citra	Output	Status Identifikasi
Bak		Bak	Benar
Gadöh		Gadöh	Benar
Jipeugah		Jipeugah	Benar
Sibak		Sibak	Benar
Tamöng		Tamöng	Benar

100 Epoch Model <i>Medium</i>			
Kelas	Citra	Output	Status Identifikasi
Sibak		Sibak	Benar
Sibak		Sibak	Benar
Singoh		Singoh	Benar
Singoh		Singoh	Benar

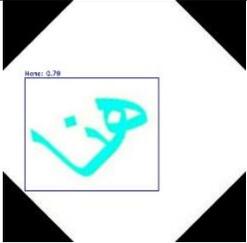
C. Lampiran 3. Hasil Pengujian Model *Big*

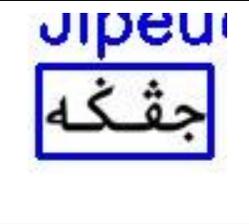
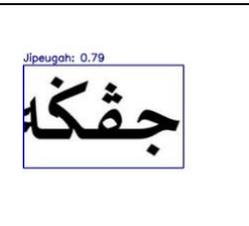
Kelas	Citra	Output	Status Identifikasi
30 Epoch Model Big			
Dipat		Dipat	Benar
Bijak		Bijak	Benar
Gaséh		Gaséh	Benar
Dipat		Dipat	Benar
Gadöh		Gadöh	Benar

Kelas	Citra	Output	Status Identifikasi
30 Epoch Model Big			
Bak		Bak	Benar

50 Epoch Model Big			
Kelas	Citra	Output	Status Identifikasi
Adat		Adat	Benar
Adat		Adat	Benar
Adat		Adat	Benar

50 Epoch Model <i>Big</i>			
Kelas	Citra	Output	Status Identifikasi
Adat		Adat	Benar

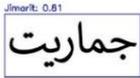
70 Epoch Model <i>Big</i>			
Kelas	Citra	Output	Status Identifikasi
Hana		Hana	Benar
Hana		Hana	Benar
Jimarit		Jimarit	Benar

70 Epoch Model <i>Big</i>			
Kelas	Citra	Output	Status Identifikasi
Jimarit		Jimarit	Benar
Jipeugah		Jipeugah	Benar
Jipeugah		Jipeugah	Benar
Jipeugah		Jipeugah	Benar

90 Epoch Model <i>Big</i>			
Kelas	Citra	Output	Status Identifikasi
Jipajôh		Jipajôh	Benar
Jipeugah		Jipeugah	Benar
Pikir		Pikir	Benar
Tamöng		Tamöng	Benar
Singoh		Singoh	Benar

90 Epoch Model <i>Big</i>			
Kelas	Citra	Output	Status Identifikasi
Sibak		Sibak	Benar

100 Epoch Model <i>Big</i>			
Kelas	Citra	Output	Status Identifikasi
Hoka		Hoka	Benar
Hana		Hana	Benar
Hana		Hana	Benar

100 Epoch Model <i>Big</i>			
Kelas	Citra	Output	Status Identifikasi
Jikeumeung		Jikeumeung	Benar
Jimarit		Jimarit	Benar

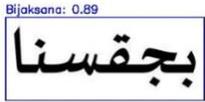
D. Lampiran 4. Hasil Pengujian Model Normal

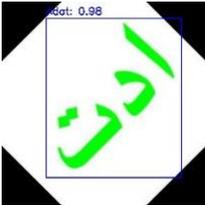
30 Epoch Model Normal			
Kelas	Citra	Output	Status Identifikasi
Hana		Hana	Benar
Jipeugah		Jipeugah	Benar
Hoka		Hoka	Benar
Bak		Bak	Benar
Nibak		Nibak	Benar

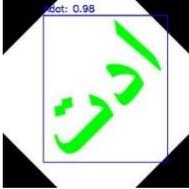
30 Epoch Model Normal			
Kelas	Citra	Output	Status Identifikasi
Nibak		Nibak	Benar

50 Epoch Model Normal			
Kelas	Citra	Output	Status Identifikasi
Gadöh		Gadöh	Benar
Hoka		Hoka	Benar
Bak		Bak	Benar

50 Epoch Model Normal

Kelas	Citra	Output	Status Identifikasi
Bijaksana		Bijaksana	Benar
Bijaksana		Bijaksana	Benar
Hana		Hana	Benar
Hana		Hana	Benar

70 Epoch Model Normal			
Kelas	Citra	Output	Status Identifikasi
Singoh		Singoh	Benar
Jipajoh		Jipajoh	Benar
Adat		Adat	Benar
Adat		Adat	Benar
Adat		Adat	Benar

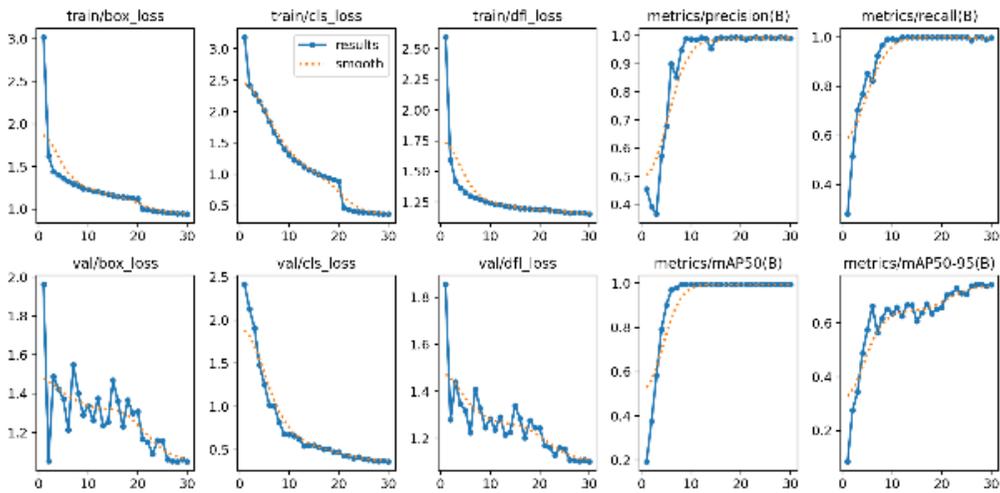
90 Epoch Model Normal			
Kelas	Citra	Output	Status Identifikasi
Jipajôh		Jipajôh	Benar
Adat		Adat	Benar
Tamöng		Tamöng	Benar
Tamöng		Tamöng	Benar
Singoh		Singoh	Benar

90 Epoch Model Normal			
Kelas	Citra	Output	Status Identifikasi
Adat		Adat	Benar

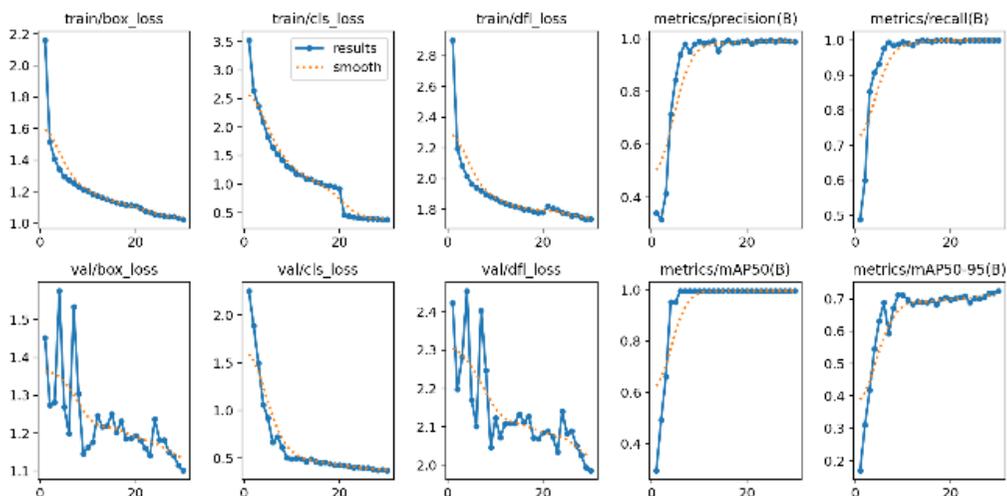
100 Epoch Model Normal			
Kelas	Citra	Output	Status Identifikasi
Nibak		Nibak	Benar
Nibak		Nibak	Benar
Nibak		Nibak	Benar

100 Epoch Model Normal			
Kelas	Citra	Output	Status Identifikasi
Pikir		Pikir	Benar

E. Lampiran 5. Grafik Hasil Pengujian



Grafik Hasil Pengujian Model *Small* untuk 30 Epoch



Grafik Hasil Pengujian Model *Big* untuk 30 Epoch

F. Lampiran 6. Biodata Peneliti



BIODATA PENELITI PUSAT PENELITIAN DAN PENERBITAN LP2M UNIVERSITAS ISLAM NEGERI AR-RANIRY BANDA ACEH

A. Identitas Diri

1.	Nama Lengkap <i>(dengan gelar)</i>	Khairan AR
2.	Jenis Kelamin L/P	Laki-Laki
3.	Jabatan Fungsional	Lektor
4.	NIP	198607042014031001
5.	NIDN	2004078602
6.	NIPN <i>(ID Peneliti)</i>	200407860210070
7.	Tempat dan Tanggal Lahir	Ambon, 04 Juli 1986
8.	E-mail	khairan.ar@ar-raniry.ac.id
9.	Nomor Telepon/HP	082174335947
10.	Alamat Kantor	Jl. Syekh Abdur Rauf Kopelma Darussalam
11.	Nomor Telepon/Faks	-
12.	Bidang Ilmu	Sistem Informasi dan Teknologi Informasi
13.	Program Studi	Teknologi Informasi

14.	Fakultas	Sains dan Teknologi
-----	----------	---------------------

B. Riwayat Pendidikan

No.	Uraian	S1	S2	S3
1.	Nama Perguruan Tinggi	UPI “YPTK”	UPI “YPTK”	
2.	Kota dan Negara PT	Padang, Indonesia	Padang, Indonesia	
3.	Bidang Ilmu/ Program Studi	Sistem Informasi	Sistem Informasi	
4.	Tahun Lulus	2008	2011	

C. Pengalaman Penelitian dalam 3 Tahun Terakhir

No.	Tahun	Judul Penelitian	Sumber Dana
1.	2017	Pengukuran Indeks Kepuasan Mahasiswa Dengan Pendekatan <i>Partial Least Square</i> (Pls) Pada Uin Ar-Raniry Banda Aceh	DIPA UIN Ar-Raniry
2.	2018	Perancangan Fitur <i>Auto Complete</i> Berbasis Algoritma <i>Knuth-Morris-DIPA UIN Ar-Raniry99 Pratt</i> Untuk Meningkatkan Efisiensi Proses Pencarian “Defenisi Istilah” Standar Operasional Prosedur (Sop) Pada Lembaga Penjaminan Mutu Uin Ar-Raniry Dan Universitas Serambi Mekkah Banda Aceh	DIPA UIN Ar-Raniry
3.	2020	Faktor-Faktor yang Mempengaruhi Tindakan <i>Knowledge Sharing</i> Antara Dosen di Perguruan Tinggi Keagamaan Islam (PTKI) Studi Kasus: UIN Ar-Raniry Banda Aceh dan STAIN Tengku Dirundeng Meulaboh	DIPA UIN Ar-Raniry
4.	2022	Evaluasi Layanan Kenaikan Jabatan Fungsional Dosen Di Perguruan Tinggi Keagamaan Islam Negeri Kementerian Agama (studi Kasus Uin Ar-raniry Banda Aceh Dan Stain Tengku Dirundeng Meulaboh)	DIPA UIN Ar-Raniry

D. Pengalaman Pengabdian Kepada Masyarakat dalam 3 Tahun Terakhir

No.	Tahun	Judul Pengabdian	Sumber Dana
1.	2018	Pendampingan mahasiswa pada acara bakti sosial di gampong buengcala kec. kuta baro aceh besar	Mandiri
2.	2019	Sosialisasi Penggunaan Teknologi	Mandiri

		Informasi Mahasiswa Saintek	
3.	2020	Penyuluhan Sekolah Tanggap Bencana	Fak. Sains dan Teknologi
4.	2020	Pengabdian Masyarakat Berbasis Penelitian, Penulisan Buku, dan Bakti Sosial di Masa Pandemi Covid-19	Mandiri

E. Publikasi Artikel Ilmiah dalam Jurnal dalam 5 Tahun Terakhir

No.	Judul Artikel Ilmiah	Nama Jurnal	Volume/Nomor/Tahun/Url
1.	Pengaruh Penggunaan Metode Videocd Untuk Meningkatkan Hasil Belajar Bahasa Indonesia Pada Mis Langugob Banda Aceh	Cyberspace	Volume 2, Nomor 1, Maret 2018, 92-96
2.	Pengembangan Media Pembelajaran Bahasa Jepang Dengan Metode Computer Based Learning Pada Sma Negeri 2 Banda Aceh	Proceeding	PROCEEDINGS ICTE2018 & ICITED 2018
3.	Penerapan Algoritma Cyberspace Volume 3, Nomor 1, Maret 2019, 26-40100 Knuth-Morris-Pratt Pada Fitur Pencarian Definisi Istilah Standar Operasional Prosedur (Sop) Pada Lembaga Penjaminan Mutu Uin Ar-Raniry	Cyberspace	Volume 3, Nomor 1, Maret 2019, 26-40
4.	Analisis Tingkat Pemahaman Guru Terhadap Penggunaan Teknologi Informasi Pada Smk Kabupaten Aceh Besar	Cyberspace	Volume 3, Nomor 2, Oktober 2019, 106-114

F. Karya Buku dalam 5 Tahun Terakhir

No.	Judul Buku	Tahun	Tebal Halaman	Penerbit
1.	Mahir Pemrograman Visual dengan Java	2017		Unimal Press

G. Perolehan HKI dalam 10 Tahun Terakhir

No.	Judul/Tema HKI	Tahun	Jenis	Nomor P/ID
1.	Perancangan Fitur Auto Complete Berbasis Algoritma KnuthMorris-Pratt Untuk Meningkatkan Efisiensi Proses Pencarian “Defenisi Istilah” Standar Operasional Prosedur (Sop) Pada Lembaga Penjaminan Mutu Uin Ar-Raniry Dan Universitas Serambi Mekkah Banda Aceh	2018	Laporan Penelitian	EC00201852942

Demikian biodata ini saya buat dengan sebenarnya.

Banda Aceh, 22 Juli 2024

Ketua/Anggota Peneliti,



Khairan AR

NIDN. 2004078602



BIODATA PENELITI
PUSAT PENELITIAN DAN PENERBITAN LP2M
UNIVERSITAS ISLAM NEGERI AR-RANIRY BANDA ACEH

H. Identitas Diri

1.	Nama Lengkap <i>(dengan gelar)</i>	Hendri Ahmadian, S.Si, M.IM
2.	Jenis Kelamin L/P	Laki-Laki
3.	Jabatan Fungsional	Lektor / III/d
4.	NIP	19830102014031002
5.	NIDN	2004018303
6.	NIPN <i>(ID Peneliti)</i>	200401830310161
7.	Tempat dan Tanggal Lahir	Aceh Selatan, 4 Januari 1983
8.	E-mail	hendri@ar-raniry.ac.id
9.	Nomor Telepon/HP	082370742500
10.	Alamat Kantor	Jl. Syech Abdurrauf Kopelma Darussalam
11.	Nomor Telepon/Faks	
12.	Bidang Ilmu	Teknologi Informasi
13.	Program Studi	Teknologi Informasi
14.	Fakultas	Sains dan Teknologi UIN Ar-Raniry

I. Riwayat Pendidikan

No.	Uraian	S1	S2	S3
1.	Nama Perguruan Tinggi	Universitas Syiah Kuala	Chung Hua University	
2.	Kota dan Negara PT	Banda Aceh Indonesia	Hsinchu Taiwan	
3.	Bidang Ilmu/ Program Studi	Matematika/Matematika	Information Management/Information Management	
4.	Tahun Lulus	2006	2011	

J. Pengalaman Penelitian dalam 3 Tahun Terakhir

No.	Tahun	Judul Penelitian	Sumber Dana
1.	2021	Analisa Topik Pendidikan Dalam Al-Quran dengan Pendekatan Text Mining	Mandiri
2.	2021	Analisis Clustering Penduduk Miskin Di Provinsi Aceh Menggunakan Algoritma K-Means Dan X-Means	Mandiri

K. Pengalaman Pengabdian Kepada Masyarakat dalam 3 Tahun Terakhir

No.	Tahun	Judul Pengabdian	Sumber Dana
1.	2021	Pengayaan Kemampuan TI Untuk Para Pendidik Dalam Rangka Menghadapi Pembelajaran Daring	Mandiri
2.	2021	Menulis Buku Rampai “Selaksa Harapan di masa Pandemi”	Mandiri

L. Publikasi Artikel Ilmiah dalam Jurnal dalam 5 Tahun Terakhir

No.	Judul Artikel Ilmiah	Nama Jurnal	Volume/Nomor/Tahun/Url
3.	Implementasi Sistem Informasi Pendukung Akreditasi Berbasis Web pada Prodi Teknologi Informasi Uin Ar-Raniry	Cyberspace: Jurnal Pendidikan Teknologi Informasi	Volume 3, Issue 2 2019
4.	Penerapan Algoritma Knuth-Morris-Pratt Pada Fitur Pencarian Definisi Istilah Standar Operasional Prosedur (Sop) Pada Lembaga Penjaminan Mutu Uin Ar-Raniry	Cyberspace: Jurnal Pendidikan Teknologi Informasi	Volume 3, Issue 1 2019
3.	Sistem Informasi E-Laboratory Berbasis Web Pada Laboratorium Multifungsi UIN Ar-Raniry Banda Aceh	Jurnal Infomedia: Teknik Informatika, Multimedia & Jaringan	Volume 5 Issue 2 2020
5	OVERVIEW OF PIRATED SOFTWARE ON CAMPUS: EDUCATORS' PERSPECTIVE	Cyberspace: Jurnal Pendidikan Teknologi Informasi	Volume 4, Issue 1 2020
6	Analisa Topik Pendidikan Dalam Al-Quran dengan Pendekatan Text Mining	Jurnal Serambi Engineering	Volume 6 Issue 1 2021
7	Perancangan Sistem Informasi Berbasis Web pada Kecamatan Ulee Kareng untuk Pemetaan Sekolah Berdasarkan Tingkat Pendidikan	Jurnal Komputer, Informasi Teknologi, dan Elektro	Volume 6 Issue 3 2021

M. Karya Buku dalam 5 Tahun Terakhir

No.	Judul Buku	Tahun	Tebal Halaman	Penerbit
1.	Selaksa Harapan di masa Pandemi	2021	-	Bandar Publishing; ISBN: 978-623-6114-54-4

N. Perolehan HKI dalam 10 Tahun Terakhir

No.	Judul/Tema HKI	Tahun	Jenis	Nomor P/ID
1.				
2.				
dst.				

Demikian biodata ini saya buat dengan sebenarnya.

Banda Aceh, 22 Juli 2024

Ketua Peneliti,



Hendri Ahmadian, S.Si, M.I.M

NIDN. 2004018303



BIODATA PENELITI
PUSAT PENELITIAN DAN PENERBITAN LP2M
UNIVERSITAS ISLAM NEGERI AR-RANIRY BANDA ACEH

O. Identitas Diri

1.	Nama Lengkap <i>(dengan gelar)</i>	Aesha Durratul Nasihah, S.Kom.
2.	Jenis Kelamin L/P	Perempuan
3.	Jabatan Fungsional	Mahasiswa
4.	NIP	-
5.	NIDN	-
6.	NIPN <i>(ID Peneliti)</i>	-
7.	Tempat dan Tanggal Lahir	Aceh Besar, 28 Agustus 2002
8.	E-mail	Aeshanasihah28@gmail.com
9.	Nomor Telepon/HP	082361091020
10.	Alamat Kantor	-
11.	Nomor Telepon/Faks	-
12.	Bidang Ilmu	Teknologi Informasi
13.	Program Studi	Teknologi Informasi
14.	Fakultas	Sains dan Teknologi

P. Riwayat Pendidikan

No.	Uraian	S1	S2	S3
1.	Nama Perguruan Tinggi	UIN Ar-Raniry		
2.	Kota dan Negara PT	Banda Aceh, Indonesia		
3.	Bidang Ilmu/ Program Studi	Teknologi Informasi		
4.	Tahun Lulus	2024		

Q. Pengalaman Penelitian dalam 3 Tahun Terakhir

No.	Tahun	Judul Penelitian	Sumber Dana
1.			
2.			
3.			
dst.			

R. Pengalaman Pengabdian Kepada Masyarakat dalam 3 Tahun Terakhir

No.	Tahun	Judul Pengabdian	Sumber Dana
1.			
2.			
3.			
dst.			

S. Publikasi Artikel Ilmiah dalam Jurnal dalam 5 Tahun Terakhir

No.	Judul Artikel Ilmiah	Nama Jurnal	Volume/Nomor/Tahun/Url
5.			
6.			
dst.			

T. Karya Buku dalam 5 Tahun Terakhir

No.	Judul Buku	Tahun	Tebal Halaman	Penerbit
2.				
3.				
dst.				

U. Perolehan HKI dalam 10 Tahun Terakhir

No.	Judul/Tema HKI	Tahun	Jenis	Nomor P/ID
1.	Desain Website Program Studi Teknologi Informasi Menggunakan Laravel	2023	Karya Ilmiah	000523676 tahun 2023
2.				
dst.				

Demikian biodata ini saya buat dengan sebenarnya.

Banda Aceh, 22 Juli 2024
Anggota Peneliti,

Aesha Durratul Nasihah