

ANALISIS PENGARUH AUGMENTASI DATA TERHADAP KINERJA DETEKSI POLA *BASELINE* TULISAN TANGAN MENGGUNAKAN *FRAMEWORK* YOLO

Aurahaqqi Aprianoputri¹, Rendra Gustriansyah¹, Muhammad Haviz Irfani*¹

¹Program Studi Teknik Informatika, Universitas Indo Global Mandiri, Kota Palembang, Indonesia

Penulis korespondensi: Muhammad Haviz Irfani (m.haviz@uigm.ac.id)

Deteksi pola baseline tulisan tangan merupakan tahapan penting dalam analisis tulisan karena dapat menggambarkan kecenderungan arah penulisan. Namun, ketidakseimbangan distribusi kelas (class imbalance) menyebabkan performa model pada kelas minoritas menjadi rendah. Penelitian ini bertujuan menganalisis pengaruh augmentasi data terhadap kinerja deteksi menggunakan YOLOv11. Dataset terdiri atas 150 citra tulisan tangan dengan 1.432 objek baseline yang diklasifikasikan menjadi Ascending Baseline (AB) sebanyak 530 objek, Descending Baseline (DB) sebanyak 611 objek, dan Wavy Baseline (WB) sebanyak 291 objek. Hasil menunjukkan bahwa augmentasi meningkatkan Precision dari 0,383 menjadi 0,517, F1-Score dari 0,505 menjadi 0,596, mAP@50 dari 0,503 menjadi 0,614, mAP@50-95 dari 0,355 menjadi 0,470, serta tingkat deteksi kelas WB berdasarkan normalized confusion matrix dari 4% menjadi 37%. Hasil tersebut menunjukkan bahwa augmentasi data memberikan pengaruh positif terhadap kinerja deteksi pola baseline tulisan tangan menggunakan YOLOv11 serta memberikan kontribusi dalam pengembangan metode deteksi berbasis kecerdasan buatan.

Keywords - *Augmentasi Data, Baseline Tulisan Tangan, Deep Learning, Deteksi Objek, YOLOv11.*

I. PENDAHULUAN

Perkembangan teknologi kecerdasan buatan atau *artificial intelligence* (AI) telah memberikan dampak besar pada berbagai bidang, termasuk sistem pengolahan citra digital. AI mencakup berbagai teknik, seperti *machine learning* dan *computer vision*, yang memungkinkan komputer mempelajari pola dari data serta menganalisis informasi visual secara otomatis. Kemajuan *computer vision* kemudian mendorong hadirnya berbagai metode yang mampu melakukan klasifikasi, pelokalan, dan deteksi objek secara lebih akurat [1], [2]. Salah satu pendekatan yang paling banyak digunakan adalah *Deep Learning*, terutama *Convolutional Neural Network* (CNN), karena kemampuannya mengekstraksi pola visual yang kompleks secara adaptif [3]. Selain itu, algoritma deteksi objek *real-time* seperti *You Only Look Once* (YOLO) menawarkan kecepatan dan efisiensi tinggi

dengan memproses prediksi posisi dan kategori objek dari satu input gambar [4]. CNN dan YOLO berpotensi menjadi metode yang tepat untuk menangani tantangan analisis tulisan tangan, khususnya dalam mendeteksi *baseline* yang bersifat variatif dan tidak konsisten antar penulis.

Tulisan tangan merupakan hasil aktivitas motorik halus yang dikendalikan oleh sistem saraf pusat. Dalam analisis karakteristik tulisan tangan, komponen *baseline* atau garis dasar imajiner tempat huruf-huruf diletakkan merupakan salah satu fitur struktural paling penting. Pola *baseline* dapat memberikan informasi mengenai kondisi psikologis, konsistensi emosional, maupun indikasi gangguan motorik atau kognitif tertentu [5]. Selanjutnya, Analisis karakteristik visual pada data tulisan tangan dapat digunakan untuk mengidentifikasi pola tertentu yang menjadi pembeda antarobjek atau kategori tulisan [6]. Namun, deteksi otomatis terhadap *baseline* tulisan tangan menghadapi tantangan besar karena variasi gaya penulisan individu yang sangat tinggi [7]. Selain itu, ketidakaturan bentuk karakter dan variasi pola tulisan antarindividu meningkatkan kompleksitas deteksi [8]. Oleh karena itu, diperlukan metode yang adaptif dan akurat

Penelitian terkait analisis karakteristik tulisan tangan telah banyak dilakukan dengan memanfaatkan pendekatan *deep learning*, khususnya CNN berbasis *framework* YOLO. Salah satu pendekatan CNN berbasis YOLOv5 digunakan untuk mendeteksi kemiringan tulisan tangan (tegak, miring kiri, miring kanan) dengan hasil *mAP@50* sebesar 0,894, menunjukkan kemampuan model dalam menangani variasi orientasi tulisan [9]. Selain itu, YOLOv11 juga diterapkan untuk mendeteksi zona spasi antarkata dengan nilai *mAP@50* sebesar 0,455, menunjukkan bahwa model mampu melakukan deteksi meskipun masih terdapat tantangan pada kemiripan visual antarkelas spasi [10].

Berdasarkan kedua penelitian tersebut, pendekatan CNN berbasis YOLO telah berhasil diterapkan dalam analisis karakteristik tulisan tangan, khususnya pada kemiringan huruf dan spasi antarkata. Di sisi lain, *baseline* tulisan tangan merupakan salah satu aspek struktur global yang penting dalam analisis karakteristik penulisan, yang merepresentasikan kecenderungan arah tulisan dalam satu baris, dan dapat diklasifikasikan

menjadi naik, turun, dan bergelombang [11]. Oleh karena itu, masih terdapat ruang pengembangan penelitian dalam deteksi *baseline* tulisan tangan berbasis YOLOv11. Penelitian ini diharapkan dapat memberikan kontribusi berupa model deteksi *baseline* yang tangguh terhadap variasi pola penulisan mahasiswa menggunakan kombinasi pemrosesan geometri MATLAB dan pembelajaran mendalam YOLOv11.

II. TINJAUAN PUSTAKA

A. Baseline Tulisan Tangan

Baseline merupakan garis imajiner yang menjadi acuan posisi karakter pada suatu baris tulisan tangan. Meskipun tidak tampak secara fisik, *baseline* berfungsi sebagai referensi utama dalam menggambarkan kecenderungan arah tulisan sehingga banyak dimanfaatkan pada berbagai sistem analisis dokumen dan *handwritten text recognition* [12]. Variasi posisi *baseline* dapat menunjukkan karakteristik tulisan yang berbeda pada setiap individu, sehingga menjadi salah satu fitur penting dalam proses analisis tulisan tangan [11], [13].

Berdasarkan bentuknya, pola *baseline* umumnya dapat dikelompokkan menjadi tiga kategori, yaitu *Ascending Baseline* (AB), *Descending Baseline* (DB), dan *Wavy Baseline* (WB). *Ascending Baseline* menunjukkan kecenderungan garis tulisan yang bergerak naik dari kiri ke kanan, sedangkan *Descending Baseline* menunjukkan arah yang menurun. Sementara itu, *Wavy Baseline* memiliki pola yang berfluktuasi sehingga membentuk gelombang [13], [14]. Variasi tersebut menyebabkan proses deteksi menjadi lebih kompleks karena model harus mampu mengenali perubahan posisi *baseline* yang tidak seragam.

B. Augmentasi Data

Augmentasi data merupakan teknik untuk memperkaya variasi data latih dengan menerapkan berbagai transformasi pada data yang sudah ada tanpa mengubah informasi utama. Pendekatan ini membantu model mempelajari karakteristik data yang lebih beragam sehingga meningkatkan kemampuan generalisasi, terutama pada kondisi jumlah data yang terbatas [15], [16].

Pada penelitian ini, augmentasi diterapkan menggunakan pustaka *Albumentations* melalui kombinasi transformasi geometris dan fotometrik. Transformasi geometris dilakukan menggunakan *Affine* untuk mensimulasikan perubahan posisi dan ukuran tulisan, sedangkan transformasi fotometrik dilakukan menggunakan *Random Brightness and Contrast*, *Gaussian Blur*, dan *Gaussian Noise* untuk menghasilkan variasi kondisi pencahayaan maupun kualitas citra. Seluruh transformasi diterapkan secara acak berdasarkan probabilitas tertentu sehingga menghasilkan kombinasi data pelatihan yang lebih beragam [15], [17].

Berbeda dengan penelitian yang menggunakan augmentasi secara menyeluruh pada seluruh *dataset*, penelitian ini memfokuskan augmentasi pada citra yang mengandung kelas *Wavy Baseline* (WB) sebagai kelas minoritas. Pendekatan tersebut bertujuan mengurangi ketidakseimbangan distribusi data sehingga model

memperoleh representasi yang lebih baik terhadap karakteristik kelas yang sebelumnya kurang terwakili.

C. YOLOv11 untuk Deteksi Baseline

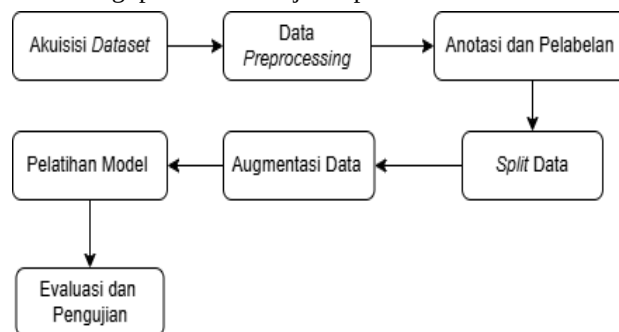
YOLO (*You Only Look Once*) merupakan salah satu metode *object detection* berbasis *Convolutional Neural Network* (CNN) yang mampu melakukan klasifikasi dan lokalisasi objek secara bersamaan dalam satu kali proses inferensi [18]. CNN berperan sebagai pengekstraksi fitur citra pada berbagai tugas *computer vision*. Integrasinya dengan *framework* deteksi objek seperti YOLO meningkatkan efisiensi proses deteksi karena CNN berperan sebagai *backbone* untuk mengekstraksi fitur citra secara mendalam [19], [20].

Pada penelitian ini digunakan YOLOv11 karena memiliki peningkatan kemampuan ekstraksi fitur melalui penyempurnaan arsitektur *backbone*, *neck*, dan *head*. Arsitektur tersebut memungkinkan model mendeteksi objek dengan ukuran maupun bentuk yang beragam secara lebih efisien [21]. Dalam penelitian ini, objek yang dideteksi berupa *bounding box* dari tiga kelas *baseline*, diantaranya *Ascending Baseline* (AB), *Descending Baseline* (DB), dan *Wavy Baseline* (WB).

Evaluasi performa model dilakukan menggunakan metrik *Precision*, *Recall*, *F1-score*, *mean Average Precision* (mAP), serta *normalized confusion matrix*. Kombinasi metrik tersebut memberikan gambaran mengenai kemampuan model dalam mengidentifikasi maupun melokalisasi objek secara akurat [11].

III. METODE PENELITIAN

Penelitian ini dilaksanakan melalui serangkaian tahapan yang meliputi akuisisi data, *preprocessing*, anotasi dan pelabelan, penerapan augmentasi, hingga pelatihan dan evaluasi model *deep learning*. Setiap tahapan dirancang untuk menganalisis pengaruh augmentasi data terhadap kinerja deteksi pola *baseline* tulisan tangan menggunakan YOLOv11. Kerangka metodologi penelitian disajikan pada Gambar 1.



Gambar 1. Kerangka kerja penelitian

A. Persiapan Dataset dan Data Preprocessing

Dataset yang digunakan dalam penelitian ini diperoleh dari tulisan tangan mahasiswa tingkat awal Universitas Indo Global Mandiri Palembang dengan total 150 citra gambar. Pengambilan data dilakukan menggunakan kertas polos tanpa garis bantu untuk menangkap kebiasaan menulis alami dari responden.

Tahap *preprocessing* awal diterapkan untuk memisahkan objek tulisan dari latar belakang citra menggunakan metode *adaptive thresholding*. Citra

berwarna terlebih dahulu dikonversi menjadi citra keabuan (*grayscale*), lalu dibinerisasi sehingga piksel tulisan bernilai satu (*foreground*) dan latar belakang bernilai nol (*background*). Setelah binerisasi, dilakukan segmentasi baris tulisan menggunakan metode proyeksi horizontal dengan rumus penjumlahan piksel biner pada setiap baris citra. Batas baris tulisan ditentukan berdasarkan nilai ambang dari nilai maksimum proyeksi horizontal untuk mengeliminasi komponen *noise* [6].

B. Penentuan Parameter dan Ambang Batas Klasifikasi Baseline

Untuk menentukan label *ground truth* secara objektif, kumpulan titik terbawah piksel tulisan pada setiap baris diekstraksi. Titik-titik ini kemudian dimodelkan menggunakan dua pendekatan regresi di perangkat lunak MATLAB, yaitu regresi *linear* orde satu dan regresi polinomial orde dua. Pendekatan tersebut digunakan untuk memperoleh karakteristik pola *baseline* yang selanjutnya menjadi dasar dalam proses pengelompokan ke dalam tiga kelas. [6]. Selanjutnya, penentuan kelas

Tabel 1. Hasil klasifikasi awal *baseline* menggunakan MATLAB

No.	Contoh Dataset	Aturan Klasifikasi	Kelas
1.		$\Delta < -12$	AB
2.		$\Delta > 12$	DB
3.		$-12 \leq \Delta \leq 12$ dan <i>Wave Energy</i> > 3	WB

Tabel 1. menyajikan contoh hasil perhitungan nilai Δ dan *wave energy* beserta hasil klasifikasi *baseline* pada beberapa sampel citra. Berdasarkan aturan pada Tabel 1, *baseline* tulisan tangan kemudian dikelompokkan ke dalam tiga kelas utama, yaitu *Ascending Baseline* (AB), *Descending Baseline* (DB), dan *Wavy Baseline* (WB).

C. Anotasi dan Pembagian Data (*Split Data*)

Berdasarkan hasil klasifikasi MATLAB, dilakukan anotasi area tulisan menggunakan LabelMe. Anotasi berbentuk *rectangle* dan *polygon* digunakan, terutama untuk menangani baris tulisan dengan kemiringan ekstrem agar tidak terjadi *overlap* antarkelas *bounding box*. Hasil anotasi berformat JSON kemudian dikonversi ke format .txt YOLO *Segmentation* yang memuat *class_id* dan koordinat ternormalisasi (0–1). Total anotasi pada data latih menghasilkan 1.432 objek.

Setelah proses anotasi selesai, sebanyak 150 citra asli dibagi (*split data*) dengan rasio 80:20 menjadi dua kelompok data utama:

1. Data Latih (*Training Set*): 120 citra (80%) dialokasikan sebagai data pembelajaran utama model dan menjadi target penerapan augmentasi data.
2. Data Validasi (*Validation Set*): 30 citra (20%) dialokasikan untuk mengevaluasi parameter terbaik dan memantau performa model selama proses pelatihan.

Pembagian data dilakukan sebelum augmentasi untuk memastikan data validasi tidak terlibat dalam augmentasi data latih sehingga meminimalkan risiko kebocoran data (*data leakage*). Pengujian akhir menggunakan 15 citra tulisan tangan independen di luar 150 citra *dataset* awal

dilakukan berdasarkan *threshold* empiris yang diperoleh melalui pengamatan terhadap karakteristik data. Dua parameter utama yang dihitung adalah:

1. Rasio Kemiringan *Baseline* (Δ).

Parameter ini dihitung dari selisih nilai koordinat vertikal hasil regresi polinomial antara titik awal dan titik akhir *baseline*. *Threshold* kemiringan ditetapkan sebesar $\Delta_{th} = 12$ piksel.

- Jika $\Delta < -12$, maka diklasifikasikan sebagai *Ascending Baseline* (AB).
- Jika $\Delta > 12$, maka diklasifikasikan sebagai *Descending Baseline* (DB).

2. Rasio Gelombang *Baseline* (*Wave Energy*)

Parameter ini merupakan nilai rata-rata selisih absolut antara hasil fungsi regresi polinomial dan regresi *linear*. *Threshold* gelombang ditetapkan sebesar $Wave_{th} = 3$ piksel.

- Jika *wave energy* > 3 , maka baris tulisan diklasifikasikan sebagai *Wavy Baseline* (WB).

sebagai *independent testing set* (*unseen data*) untuk menguji kemampuan generalisasi model terhadap data yang belum digunakan dalam pelatihan maupun validasi.

D. Penerapan Augmentasi Data

Ketidakeimbangan distribusi data, dengan dominasi kelas *Ascending Baseline* (AB) dan *Descending Baseline* (DB) dibandingkan *Wavy Baseline* (WB), mendorong penerapan augmentasi data pada citra yang mengandung objek WB. Augmentasi menggunakan pustaka *Alumentations* dengan kombinasi transformasi fotometrik dan geometris ringan sebanyak tiga kali secara probabilistik.

Tabel 2. Jumlah objek per kelas pada data latih

Class	Sebelum Augmentasi	Setelah Augmentasi
AB	530	1656
DB	611	1531
WB	291	1072
All	1432	4259

Berdasarkan Tabel 2. augmentasi ini sekaligus memicu peningkatan jumlah data pada seluruh kelas karena satu lembar citra masukan umumnya mengandung representasi objek multi-kelas, sehingga saat citra yang memuat kelas WB diaugmentasi, kelas objek lain (AB dan DB) yang berada pada citra yang sama secara otomatis ikut bertambah. Kondisi tersebut menyebabkan proporsi WB tetap lebih rendah, meskipun keterwakilannya meningkat setelah augmentasi.

Meskipun jumlah kelas WB setelah augmentasi masih relatif lebih sedikit dibandingkan kelas AB dan DB, kondisi tersebut dinilai cukup karena augmentasi tidak ditujukan untuk menghasilkan distribusi kelas yang

sepenuhnya seimbang, melainkan untuk meningkatkan representasi kelas minoritas yang sebelumnya kurang terwakili. Proporsi objek WB pada data latih meningkat dari 20,3% menjadi 25,2% setelah augmentasi, sehingga keterwakilan kelas WB dalam proses pembelajaran model diharapkan dapat menjadi lebih baik. Pendekatan ini diharapkan dapat membantu model membentuk batas keputusan (*decision boundary*) yang lebih representatif terhadap karakteristik visual kelas WB, sekaligus meminimalkan ketergantungan model terhadap perubahan visual yang dihasilkan dari proses augmentasi agar model tetap belajar pola *baseline* asli.

E. Konfigurasi Arsitektur CNN dan YOLOv11

Model yang digunakan adalah YOLO11n-seg (*You Only Look Once Segmentation*). Pemilihan model ini didasarkan pada karakteristik data anotasi yang tidak hanya menggunakan *bounding box* berbentuk *rectangle*, tetapi juga *polygon*. Konvolusi pada arsitektur *Convolutional Neural Network* (CNN) ini bertugas mengekstraksi fitur visual dasar seperti tepi, garis, dan tekstur tulisan, dengan fungsi aktivasi *Sigmoid-Weighted Linear Unit* (SiLU) yang dirumuskan pada persamaan berikut:

$$\text{SiLU } \sigma(x) = x \cdot \sigma(x) \quad (1)$$

Di mana:

$\text{SiLU } \sigma(x)$ = fungsi aktivasi *Sigmoid Linear Unit* (SiLU)

x = nilai masukan (*input*) ke fungsi aktivasi

$\sigma(x)$ = fungsi *sigmoid*

Fungsi *sigmoid* tersebut didefinisikan sebagai berikut:

$$\sigma(x) = \frac{1}{1 + e^{-x}} \quad (2)$$

Di mana:

e = bilangan Euler ($\approx 2,71828$)

Model selanjutnya dilatih menggunakan konfigurasi parameter yang telah ditentukan untuk memperoleh performa deteksi yang optimal. Proses pelatihan dijalankan menggunakan *platform* Google Colab berbasis *Graphics Processing Unit* (GPU). Parameter pelatihan diatur secara konsisten dengan resolusi citra masukan sebesar 640×640 piksel, *batch size* bernilai 16, durasi pelatihan sebanyak 80 *epoch*, serta menggunakan *optimizer* AdamW. Untuk memperoleh performa deteksi yang optimal, eksperimen *hyperparameter tuning* dilakukan dengan menguji kombinasi nilai *momentum* (0.75, 0.85), *weight decay* (0.0004 hingga 0.0008), dan *learning rate* (0.1, 0.01, 0.001, 0.0001).

IV. HASIL DAN PEMBAHASAN

A. Hasil Eksperimen Hyperparameter Tuning

Pada pengujian awal menggunakan *dataset* asli, dilakukan pencarian kombinasi *hyperparameter* terbaik. Model YOLO11n-seg menghasilkan evaluasi berbasis *bounding box* (*Box*) dan *segmentation* (*Mask*), penelitian ini menggunakan metrik *Box* sebagai indikator utama dalam menilai performa model karena hanya berfokus pada deteksi objek *baseline*. Ringkasan percobaan kombinasi *hyperparameter* terbaik ditunjukkan pada Tabel 3.

Tabel 3. Hasil percobaan pelatihan berdasarkan *box*

Run	Momentum	Weight Decay	Learning Rate	Precision	Recall	F1-Score	mAP@50	mAP@50-95
1	0.75	0,0006	0,001	0,393	0,633	0,485	0,495	0,349
2	0.85	0,0006	0,001	0,362	0,513	0,424	0,448	0,321
3	0.75	0,0004	0,001	0,376	0,509	0,432	0,474	0,329
4	0.75	0,0005	0,001	0,373	0,701	0,486	0,477	0,339
5	0.75	0,0007	0,001	0,383	0,743	0,505	0,503	0,355
6	0.75	0,0008	0,001	0,310	0,646	0,419	0,492	0,345
7	0.75	0,0007	0,01	0,149	0,701	0,246	0,246	0,135
8	0.75	0,0007	0,1	0,160	0,601	0,252	0,211	0,087
9	0.75	0,0007	0,0001	0,276	0,541	0,365	0,251	0,150

Berdasarkan Tabel 3, pengujian *hyperparameter* dilakukan secara bertahap dengan menguji nilai *momentum*, *weight decay*, dan *learning rate*. Pengujian diawali dengan variasi nilai *momentum*, sementara *learning rate* dan *weight decay* dipertahankan pada nilai awal. Nilai *momentum* dengan performa terbaik kemudian digunakan pada tahap pengujian *weight decay*. Selanjutnya, nilai *learning rate* divariasikan dengan mempertahankan nilai terbaik dari *momentum* dan *weight decay* untuk memperoleh konfigurasi *hyperparameter* dengan performa model paling optimal.

Konfigurasi pada Run ke-5 (*Momentum*: 0.75, *Weight Decay*: 0.0007, *Learning Rate*: 0.001) dipilih sebagai model terbaik karena menghasilkan nilai *mAP@50* tertinggi sebesar 0,503. Namun, jika performa model terbaik ini dibedah per-kelas, terlihat perbedaan performa yang cukup signifikan sebagai dampak dari ketidakseimbangan distribusi kelas (*class imbalance*). Performa model terbaik per kelas disajikan pada Tabel 4.

Tabel 4. Performa model per kelas pada percobaan ke-5 pelatihan awal

Class	Precision	Recall	F1-Score	mAP@50	mAP@50-95
AB	0,476	0,876	0,617	0,678	0,5
DB	0,446	0,949	0,607	0,62	0,435
WB	0,226	0,404	0,290	0,211	0,129
All	0,383	0,743	0,505	0,503	0,355

Berdasarkan Tabel 4, performa model menunjukkan perbedaan pada setiap kelas *baseline*. Kelas *Ascending Baseline* (AB) dan *Descending Baseline* (DB) memperoleh hasil yang relatif baik dengan nilai *Recall* masing-masing sebesar 0,876 dan 0,949 serta *mAP@50* di atas 0,620. Sebaliknya, kelas *Wavy Baseline* (WB) memiliki performa terendah dengan *Precision* sebesar 0,226, *Recall* sebesar 0,404, dan *mAP@50* sebesar 0,211. Hasil tersebut menunjukkan bahwa model lebih mampu mengenali kelas mayoritas dibandingkan kelas minoritas. Rendahnya performa pada kelas WB sejalan dengan kondisi ketidakseimbangan distribusi data latih. Temuan ini menjadi dasar penerapan augmentasi data pada kelas WB untuk meningkatkan representasi data dan mengevaluasi pengaruhnya terhadap kinerja model.

B. Pengaruh Penerapan Augmentasi Data terhadap Validasi Model

Berdasarkan hasil evaluasi pada pelatihan awal, kelas *Wavy Baseline* (WB) menunjukkan performa deteksi yang paling rendah dibandingkan kelas lainnya. Oleh karena itu, model dilatih kembali menggunakan *dataset* yang telah melalui proses augmentasi, khususnya pada citra yang mengandung objek kelas WB. Proses pelatihan ini mempertahankan konfigurasi *hyperparameter* terbaik dari *Run* ke-5, sehingga perubahan kinerja model dapat dievaluasi berdasarkan pengaruh penambahan variasi data melalui augmentasi tanpa dipengaruhi oleh perubahan parameter pelatihan. Performa model pada data pelatihan setelah augmentasi disajikan pada Tabel 5.

Tabel 5. Performa model per kelas setelah penerapan augmentasi data

Class	Precision	Recall	F1-Score	mAP@50	mAP@50-95
AB	0,656	0,74	0,696	0,689	0,521
DB	0,68	0,789	0,730	0,788	0,612
WB	0,378	0,343	0,360	0,365	0,277
All	0,571	0,624	0,596	0,614	0,47

Berdasarkan Tabel 5, penerapan augmentasi menghasilkan peningkatan performa pada seluruh kelas *baseline*. Kelas *Descending Baseline* (DB) memperoleh performa terbaik dengan nilai *F1-Score* sebesar 0,730 dan *mAP@50* sebesar 0,788, diikuti oleh kelas *Ascending Baseline* (AB) dengan *F1-Score* sebesar 0,696 dan *mAP@50* sebesar 0,689. Meskipun kelas WB masih memiliki performa terendah, nilai *Precision*, *Recall*, *F1-Score*, dan *mAP@50* meningkat dibandingkan sebelum augmentasi. Hasil tersebut menunjukkan bahwa augmentasi memberikan kontribusi positif terhadap kemampuan model dalam mengenali kelas minoritas.

Perbandingan performa model secara keseluruhan (*All Class*) sebelum dan sesudah augmentasi dilakukan untuk mengevaluasi pengaruh augmentasi data menggunakan metrik *Precision*, *Recall*, *F1-Score*, *mAP@50*, dan

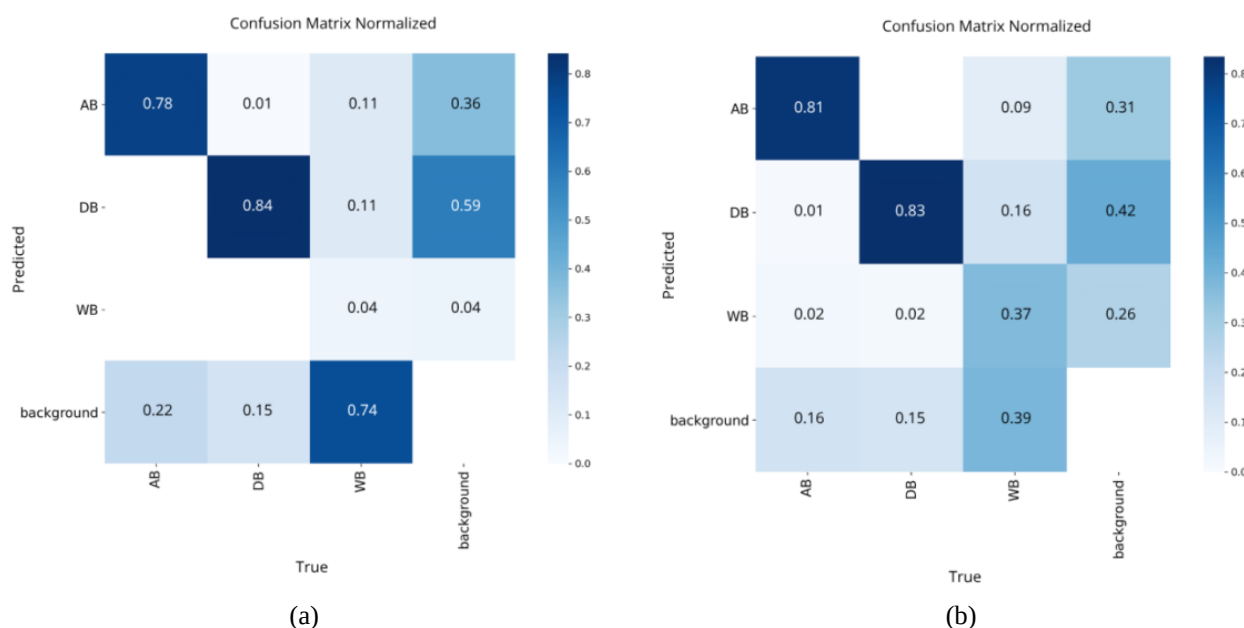
mAP@50-95. Evaluasi ini bertujuan untuk mengetahui perubahan kemampuan model dalam mendeteksi objek *baseline* setelah penerapan augmentasi data. Hasil perbandingan disajikan pada Tabel 6.

Tabel 6. Perbandingan performa makro validasi model

Matrix	Sebelum Augmentasi	Setelah Augmentasi	Perubahan (%)
Precision	0,383	0,517	+13,4%
Recall	0,743	0,624	-11,9%
F1-Score	0,505	0,596	+9,1%
mAP@50	0,503	0,614	+11,1%
mAP@50-95	0,355	0,47	+11,5%

Berdasarkan Tabel 6, penerapan augmentasi data memberikan perubahan terhadap kinerja model secara keseluruhan. Nilai *Precision* meningkat dari 0,383 menjadi 0,517, *F1-Score* dari 0,505 menjadi 0,596, *mAP@50* dari 0,503 menjadi 0,614, dan *mAP@50-95* dari 0,355 menjadi 0,470. Sementara itu, nilai *Recall* mengalami penurunan dari 0,743 menjadi 0,624. Kondisi ini menunjukkan bahwa setelah augmentasi model menjadi lebih selektif dalam menghasilkan prediksi, sehingga mampu mengurangi kesalahan deteksi (*false positive*) dan meningkatkan ketepatan prediksi, meskipun masih terdapat beberapa objek *baseline* yang tidak berhasil terdeteksi.

Secara keseluruhan, peningkatan pada nilai *Precision*, *F1-Score*, dan *mAP* menunjukkan bahwa augmentasi data memberikan pengaruh positif terhadap kinerja deteksi pola *baseline* menggunakan YOLOv11. Peningkatan presisi makro menunjukkan adanya perbaikan ketepatan prediksi model setelah penerapan augmentasi. Penyajian *normalized confusion matrix* bentuk persentase berikut memudahkan analisis proporsi prediksi benar dan kesalahan klasifikasi pada setiap kelas *baseline* sebelum dan setelah augmentasi data.



Gambar 1. Normalized Confusion Matrix: (a) Pelatihan Awal, dan (b) Setelah Augmentasi Data

Berdasarkan *normalized confusion matrix* pada Gambar 2, augmentasi data meningkatkan kemampuan model dalam mengenali pola *baseline*, terutama pada kelas WB yang meningkat dari 4% menjadi 37%. Sementara itu,

tingkat deteksi kelas AB meningkat dari 78% menjadi 81% dan kelas DB relatif stabil dari 84% menjadi 83%. Hasil ini menunjukkan bahwa augmentasi efektif meningkatkan representasi kelas minoritas, meskipun

kesalahan prediksi terhadap *background* masih ditemukan.

C. Hasil Pengujian Akhir

Model terbaik (*best.pt*) hasil augmentasi selanjutnya dievaluasi menggunakan data uji independen (*unseen data*) untuk mengukur kemampuan generalisasi model terhadap citra baru yang tidak terlibat dalam proses pelatihan maupun validasi. Pengujian dilakukan

Tabel 7. Hasil pengujian akhir performa model

Class	Jumlah Objek	Precision	Recall	F1-Score	mAP@50	mAP@50-95
AB	16	0,672	0,896	0,768	0,791	0,525
DB	56	0,532	0,768	0,629	0,647	0,456
WB	9	0,066	0,111	0,083	0,122	0,090
All	81	0,423	0,592	0,493	0,520	0,357

Berdasarkan Tabel 7, hasil pengujian akhir menunjukkan nilai rata-rata *Precision* sebesar 0,423, *Recall* sebesar 0,592, dan *mAP@50* sebesar 0,520. Kelas AB memperoleh performa terbaik dengan nilai *Recall* mencapai 0,896, yang menunjukkan bahwa sebagian besar pola garis naik dapat terdeteksi dengan baik. Sementara itu, kelas DB menunjukkan performa yang relatif stabil dengan nilai *mAP@50* sebesar 0,647. Sebaliknya, performa kelas WB pada data uji independen mengalami penurunan yang cukup besar, dengan nilai *Precision* sebesar 0,066 dan *mAP@50* sebesar 0,122.

Perbedaan performa antarkelas menunjukkan bahwa kemampuan model dalam mengenali pola WB pada data validasi belum sepenuhnya dapat digeneralisasikan pada data uji independen. Penurunan tersebut diduga berkaitan dengan karakteristik pola WB yang memiliki variasi morfologi dan fluktuasi lokal lebih kompleks, sehingga pola pada data uji dapat berbeda dari karakteristik yang dominan dipelajari selama pelatihan dan memengaruhi kemampuan model dalam mengenali kelas tersebut.

Selain itu, augmentasi berbasis *Affine* hanya menghasilkan perubahan posisi, skala, dan orientasi, sehingga belum sepenuhnya merepresentasikan variasi bentuk dan kurvatur nonlinier pada pola WB. Meskipun augmentasi meningkatkan jumlah objek WB, jumlahnya masih lebih rendah dibandingkan kelas AB dan DB. Kondisi tersebut dapat menyebabkan representasi kelas mayoritas lebih dominan dalam pembentukan *decision boundary*, sehingga model lebih sulit membedakan pola WB dengan karakteristik yang berbeda dari data latih. Hal ini dapat menjadi salah satu faktor yang memengaruhi penurunan performa WB pada data uji.

V. KESIMPULAN DAN SARAN

Penelitian ini berhasil mengimplementasikan metode *Convolutional Neural Network* (CNN) berbasis *framework* YOLOv11 untuk mendeteksi dan mengklasifikasikan tiga jenis *baseline* tulisan tangan. Penerapan teknik pengolahan geometri MATLAB (regresi *linear* dan polinomial) membantu meminimalkan subjektivitas penentuan label *ground truth*. Hasil pengujian menunjukkan penerapan augmentasi data mampu meningkatkan presisi makro model pada data validasi (*mAP@50* naik dari 0,503 menjadi 0,614), serta mencatatkan nilai *Precision* 0,423 dan *Recall* 0,592 pada data uji acak. Model mampu mengenali kelas *Ascending Baseline* (AB) dan *Descending Baseline* (DB) secara

menggunakan 15 citra tulisan tangan dengan total 81 objek *baseline* aktual yang terdiri dari tiga kelas, yaitu *Ascending Baseline* (AB), *Descending Baseline* (DB), dan *Wavy Baseline* (WB). Evaluasi ini bertujuan untuk mengetahui performa akhir model dalam melakukan deteksi pada kondisi yang mendekati penggunaan sebenarnya. Hasil evaluasi akhir model pada data uji ini disajikan pada Tabel 7.

konsisten, tetapi masih bias dan kesulitan menggeneralisasi kelas *Wavy Baseline* (WB) akibat tingginya kompleksitas bentuk gelombang tulisan.

Untuk penelitian selanjutnya, disarankan untuk memperluas akuisisi *dataset*, khususnya pada kelas minoritas WB, melalui pengumpulan lebih banyak sampel tulisan tangan dengan variasi karakteristik yang beragam. Selain itu, menguji pendekatan yang lebih agresif seperti arsitektur berbasis *Generative Adversarial Network* (GAN) untuk menghasilkan data sintesis kelas WB yang lebih natural, serta penerapan *Class Weighting* dan *Focal Loss* dapat dipertimbangkan untuk meningkatkan keseimbangan pembelajaran antarkelas.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada dosen pembimbing atas arahan, masukan, dan bimbingan selama proses penelitian serta penyusunan artikel ini.

DAFTAR PUSTAKA

- [1] Armita, F. Marpaung, F. Aulia, N. Suryani, and R. C. Nabila, *Computer Vision dan Pengolahan Citra Digital*, 1st ed. Surabaya: Pustaka Aksara, 2022.
- [2] A. Sanmorino, R. Gustriansyah, S. Puspasari, A. Bamidele, and D. Z. Rodriguez, "Fine-Tuning for a Deep Learning Model in Optimizing Ransomware Attack Detection in the Industry 4.0 and 5.0 Era," in *Computational Intelligence in Industry 4.0 and 5.0 Applications*, 1st ed., Auerbach Publications, 2025, p. 33. doi: 10.1201/9781003581963-6.
- [3] N. Kasim and G. S. Nugraha, "Pengenalan Pola Tulisan Tangan Aksara Arab Menggunakan Metode Convolution Neural Network," *J. Teknol. Informasi, Komputer, dan Apl.*, vol. 3, no. 1, pp. 85–95, 2021, doi: 10.29303/jtika.v3i1.136.
- [4] P. Jiang, D. Ergu, F. Liu, Y. Cai, and B. Ma, "A Review of Yolo Algorithm Developments," *Procedia Comput. Sci.*, vol. 199, pp. 1066–1073, 2021, doi: 10.1016/j.procs.2022.01.135.
- [5] F. Burgio, R. Pezzetta, J. Gooijers, S. Nordio, and D. Mantini, "Neural and motor mechanisms of handwriting: from healthy aging to neurodegenerative disorders.," *Front. Aging Neurosci.*, vol. 18, p. 1758541, 2026, doi: 10.3389/fnagi.2026.1758541.
- [6] G. Gasim, R. Heriansyah, M. H. Irfani, and S. Cahyani, "Buku Ajar PENGENALAN POLA 'PATTERN RECOGNITION': Teori dan Implementasi Menggunakan MATLAB," *Repos. Glob. Aksara Pers*, vol. 2, no. 2, Jul. 2026, [Online]. Available: <https://jurnal.globalaksarapers.com/index.php/repository/article/view/367>
- [7] M. Rosyada, Indriani Zabrina Putri, Saskia Aila Virida, D. A. Adi Saputra, and A. Setiawan, "Pengenalan Angka Tulisan Tangan Menggunakan Metode Machine Learning Random Forest," *JSI (Jurnal Sist. Informasi) Univ. Suryadarma*, vol. 12, no. 2, pp. 74–80, Jul. 2025, doi: 10.35968/jsi.v12i2.1476.

- [8] M. H. Irfani and Gasim, "Segmentasi Teks pada Citra Tulisan Tangan Kalimat Menggunakan Metode Median Filtering dan Otsu," *Teknosains Media Inf. Sains dan Teknol.*, vol. 18, no. 1, pp. 88–97, 2024, doi: 10.24252/teknosains.v18i1.44307.
- [9] A. Nurlita, M. H. Irfani, and Z. R. Mair, "Pendekatan Convolutional Neural Network dalam Mendeteksi Kemiringan Tulisan Tangan Menggunakan Framework YOLO," *J. FASILKOM (Teknologi Inf. dan Ilmu Komputer)*, vol. 15, no. 3, pp. 640–647, 2025, doi: 10.37859/jf.v15i3.10406.
- [10] N. Mufti, R. Heriansyah, and M. H. Irfani, "Analisis Struktur Tulisan Tangan melalui Deteksi Zona Spasi Antarkata menggunakan Convolutional Neural Network Mendalam," *SMARTICS J.*, vol. 12, no. 1, pp. 28–35, Apr. 2026, doi: 10.21067/smartics.v12i1.13937.
- [11] M. H. Irfani, Samsuryadi, Abdiansah, and R. Heriansyah, "Accurate Detection of Handwriting Features Using Bounding Box Optimization," *Int. Inf. Eng. Technol. Assoc.*, vol. 30, no. 9, pp. 2499–2509, 2025, doi: 10.18280/isi.300923.
- [12] P. Tripathi, "Variations in Handwriting Characteristics due to Different Grip Patterns," *Acad. J. Forensic Sci.*, vol. 05, no. 01, pp. 26–42, 2022, [Online]. Available: <https://xournals.com/journal/variations-in-handwriting-characteristics-due-to-different-grip-patterns>
- [13] I. Rabaev and M. Litvak, "Recent Advances in Text Line Segmentation and Baseline Detection in Historical Document Images: A Systematic Review," *Int. J. Doc. Anal. Recognit.*, pp. 1–37, 2025, doi: 10.1007/s10032-025-00526-w.
- [14] D. Gagliu and D. Sendrescu, "Detection of Personality Traits Using Handwriting and Deep Learning," *Appl. Sci.*, vol. 15, no. 4, pp. 1–18, 2025, doi: <https://doi.org/10.3390/app15042154>.
- [15] M. Xu, S. Yoon, A. Fuentes, and D. S. Park, "A Comprehensive Survey of Image Augmentation Techniques for Deep Learning," *Pattern Recognit.*, vol. 137, p. 109347, 2023, doi: 10.1016/j.patcog.2023.109347.
- [16] Z. Wang *et al.*, "A Comprehensive Survey on Data Augmentation," vol. 14, no. 8, pp. 1–20, 2021, doi: 10.48550/arXiv.2405.09591.
- [17] A. Mumuni and F. Mumuni, "Data augmentation: A comprehensive survey of modern approaches," *Array*, vol. 16, p. 100258, 2022, doi: <https://doi.org/10.1016/j.array.2022.100258>.
- [18] E. Nhancements, "YOLOv11: An Overview of The Key Architectural Enhancements," Ithaca, 2024. doi: 10.48550/arXiv.2410.17725.
- [19] Z. Mair *et al.*, "An Enhanced Deep Learning Framework for Diabetic Retinopathy Classification Using Multiple Convolutional Neural Network Architectures," *Int. J. Adv. Comput. Sci. Appl.*, vol. 16, no. 11, pp. 769–776, Nov. 2025, doi: 10.14569/IJACSA.2025.0161176.
- [20] M. Toyib, T. D. K. Pratama, and I. Aqil, "Penerapan Algoritma CNN Untuk Mendeteksi Tulisan Tangan Angka Romawi dengan Augmentasi Data," *J. Mat. Ilmu Pengetah. Alam, Kebumian dan Angkasa*, vol. 2, no. 3, pp. 108–120, 2024, doi: 10.62383/algoritma.v2i3.69.
- [21] A. F. Rasheed and M. Zarkoosh, "YOLOv11 Optimization for Efficient Resource," *J. Supercomput.*, vol. 81, pp. 1–21, 2025, doi: 10.1007/s11227-025-07520-3.