

Sistem Cerdas Identifikasi Aksara Jawa Menggunakan *Convolutional Neural Network* (CNN) Berbasis *Canvas* Python (Studi Kasus: SMPN 2 Nglepok)

^{1*}Nimas Ayu Anggun Kharisma, ²Abdi Pandu Kusuma, ³Sabitul Kirom

¹Teknik Informatika, Universitas Islam Balitar

^{2,3}Sistem Komputer, Universitas Islam Balitar

E-mail: ¹nimasayu498@gmail.com, ²pans.uin1blitar@gmail.com, ³sabitulkirom@gmail.com

Penulis Korespondens : Nimas Ayu Anggun Kharisma

Abstrak— Aksara Jawa merupakan sistem tulisan tradisional yang kaya nilai budaya, namun semakin jarang digunakan seiring perkembangan jaman. Kondisi ini mendorong perlunya upaya pelestarian melalui pendekatan teknologi interaktif. Penelitian ini bertujuan mengimplementasikan algoritma deep learning pada sistem canvas digital yang memungkinkan pengguna berlatih menulis aksara Jawa secara realtime dengan fitur pengenalan aksara langsung. Penelitian ini menggunakan metode penelitian terapan dengan algoritma Convolutional Neural Network (CNN) yang diintegrasikan pada media canvas digital berbasis tkinter Python. Dataset terdiri dari 24.388 citra dari 244 kelas aksara Jawa beserta sandhangan setelah augmentasi, dengan rasio pembagian 70:15:15 untuk data latih, validasi, dan uji. Arsitektur CNN terdiri dari 4 blok konvolusi dengan optimizer AdamW, dilanjutkan fine-tuning 15 epoch untuk meningkatkan performa. Proses penelitian meliputi identifikasi masalah, studi literatur, pengumpulan data, pra-pemrosesan, perancangan arsitektur, pelatihan model, dan evaluasi. Hasil evaluasi menunjukkan performa model yang baik dengan akurasi 94.83%, precision 0.95, recall 0.95, dan f1-score bernilai 0.95. Pengujian pada canvas digital menunjukkan sebagian besar aksara dapat dikenali dengan tingkat keyakinan yang tinggi. Sistem ini diharapkan menjadi media pembelajaran interaktif yang berkontribusi pada pelestarian dan pembelajaran aksara Jawa di era digital.

Kata Kunci— aksara Jawa , CNN, deep learning, identifikasi karakter Jawa, identifikasi tulisan tangan

Abstract— *Javanese script is a traditional writing system with significant cultural value, but its use has gradually declined over time. This study aims to support its preservation by implementing a deep learning algorithm in a digital canvas system that enables users to practice writing Javanese characters in real time with instant recognition. The research applies a Convolutional Neural Network (CNN) model integrated into a Python tkinter-based digital canvas. The dataset consists of 24,388 images from 244 classes of Javanese characters and sandhangan after augmentation, divided into training, validation, and testing sets with a ratio of 70:15:15. The CNN architecture comprises four convolutional blocks optimized using AdamW, followed by 15 epochs of fine-tuning to improve performance. The research stages include problem identification, literature review, data collection, preprocessing, model development, training, and evaluation. The model achieved an accuracy of 94.83%, with precision, recall, and F1-score of 0.95. Testing on the digital canvas showed that most handwritten characters were recognized with high confidence. The proposed system can serve as an interactive learning medium for supporting the preservation and learning of Javanese script in the digital era.*

Keywords— CNN, deep learning, Javanese script,, Javanese character recognition, handwritten recognition

This is an open access article under the CC BY-SA License



I. PENDAHULUAN

Aksara Jawa merupakan sistem tulisan tradisional yang menjadi bagian dari identitas budaya masyarakat Jawa. Aksara ini berkembang dari aksara Kawi atau Jawa Kuno dan dikenal dengan beberapa sebutan, seperti Hanacaraka, Carakan, dan Dentawyanjana. Secara historis, aksara Jawa digunakan pada berbagai media tulis, seperti kertas, kayu, batu, dan kain [1]. Namun, seiring perkembangan zaman, penggunaan aksara Jawa semakin berkurang dan umumnya hanya digunakan oleh kalangan sastrawan maupun pemerhati budaya [2].

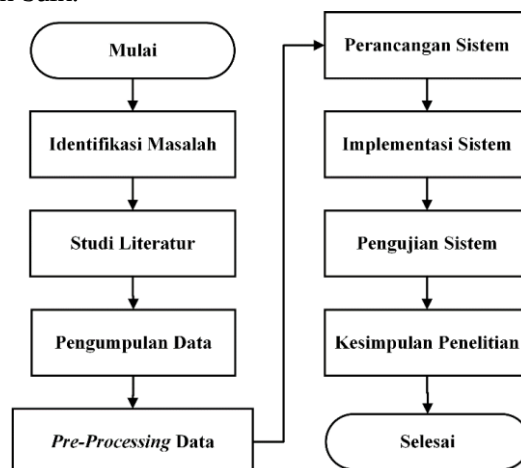
Menurunnya penggunaan aksara Jawa dipengaruhi oleh berbagai faktor. Kompleksitas bentuk dan aturan penulisannya sering menimbulkan kesulitan bagi pelajar maupun pengajar. Kesulitan tersebut mencakup penggunaan pasangan, penempatan sandhangan, serta perbedaan aksara yang memiliki bentuk serupa, seperti “da”, “pa”, dan “na” [3]. Selain itu, keterbatasan waktu pembelajaran di sekolah menyebabkan siswa memiliki sedikit kesempatan untuk berlatih menulis aksara Jawa secara berkelanjutan [4]. Perkembangan teknologi digital yang serba instan juga membuat pembelajaran aksara Jawa kurang diminati oleh generasi muda [4].

Siswa cenderung menganggap aksara Jawa sebagai materi kuno dan kurang relevan dengan kehidupan digital mereka. Persepsi tersebut diperkuat oleh metode pembelajaran yang masih konvensional dan minimnya kesempatan praktik di luar sekolah [3]. Akibatnya, keterampilan menulis aksara Jawa mudah memudar karena tidak digunakan dalam kehidupan sehari-hari, sementara waktu latihan yang terbatas belum mampu membentuk kemampuan yang bertahan lama [4].

Berdasarkan permasalahan tersebut, diperlukan media pembelajaran yang lebih interaktif untuk mendukung latihan menulis aksara Jawa. Penelitian ini memanfaatkan algoritma *Convolutional Neural Network* (CNN), yang dikenal efektif dalam mengenali dan mengekstraksi fitur dari data citra [5]. Berbagai penelitian menunjukkan bahwa CNN mampu memberikan tingkat akurasi tinggi pada pengenalan tulisan tangan, termasuk aksara Jawa, dengan akurasi lebih dari 90% bahkan mencapai 99,83% pada kondisi tertentu [6], [7]. Oleh karena itu, penelitian ini mengimplementasikan CNN pada media *canvas* berbasis Python untuk mengenali tulisan aksara Jawa secara langsung, sehingga dapat menjadi sarana pembelajaran yang lebih praktis dan interaktif.

II. METODE

Pada penelitian ini, metode yang digunakan untuk identifikasi aksara Jawa menggunakan *Convolutional Neural Network* (CNN) yang diintegrasikan pada media *input canvas* Python. Alura tahapan penelitian seperti pada gambar 1 agar teratur dan terarah, hal ini bertujuan agar penelitian berjalan dengan baik.



Gambar 1. Alur Penelitian

2.1. Identifikasi Masalah

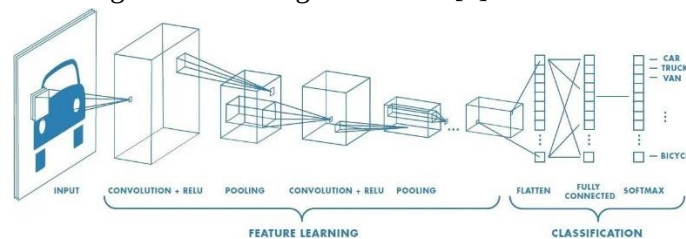
Pada tahap pertama penelitian ini adalah mengidentifikasi masalah terkait tentang kepenulisan aksara Jawa yang menjadi fokus utama terutama pada bidang pendidikan, khususnya pada sekolah menengah pertama (SMP). Dilakukan observasi pada lokasi penelitian serta wawancara khususnya dengan guru Bahasa Jawa untuk mengetahui permasalahan terkait pembelajaran menulis aksara Jawa.

2.2. Studi Literatur

Setelah mengetahui permasalahan yang ada, langkah selanjutnya adalah studi literatur yaitu mengkaji artikel atau jurnal ilmiah terdahulu yang berkaitan dengan pemecahan masalah. Dikumpulkan berbagai artikel yang berkaitan dengan identifikasi aksara Jawa menggunakan algoritma CNN, serta mengkaji teori-teori lain yang mendukung penelitian.

2.2.1) *Convolutional Neural Network (CNN)*

Pada perkembangan teknologi *machine learning*, salah satu cabangnya yang merupakan *deep learning* memiliki algoritma yang sering digunakan bernama *Convolutional Neural Network (CNN)*. CNN merupakan jenis dari *artificial neural network* yang pada umumnya digunakan untuk menganalisis gambar visual. Di CNN, tidak semua *neuron* pada sebuah lapisan saling berhubungan dengan neuron di lapisan lain. Hal tersebut bertujuan untuk mencegah *overfitting* [8]. Struktur CNN terdiri dari beberapa lapisan kunci, termasuk *Convolutional Layer*, *Activation Function*, *Pooling Layer*, *Fully Connected Layer*, dan *Output Layer*. Semua bagian itu saling bekerja bersama untuk mengeloah dan menganalisis data[9].



Gambar 2. *Convolutional Neural Network*

2.2.2) *Handwritten Recognition*

Handwriting recognition atau analisis tulisan tangan adalah salah satu bentuk dari pengenalan pola yang fokus pada identifikasi tulisan tangan manusia. Analisis tulisan tangan ini umumnya dibagi menjadi dua jenis: online, yaitu ketika penulisan direkam secara langsung, dan offline, yaitu ketika tulisan dianalisis dari gambar atau foto. Kedua jenis ini memiliki tantangan masing-masing. Selain dari kebutuhan penelitian akademis, kebutuhan teknologi analisis serta identifikasi tulisan tangan yang secara real-time juga semakin meningkat. Misalnya untuk penggunaan pena digital, verifikasi tanda tangan, dan pengelolaan dokumen di berbagai bidang usaha. Dari penelitian yang telah dilakukan oleh banyak peneliti, menunjukkan bahwa adanya perbedaan bahasa dan budaya dapat memengaruhi cara sistem mengenali tulisan tangan [10].

2.3. Pengumpulan Data

Pengumpulan data dilakukan melalui pengambilan data primer berupa tulisan tangan siswa dan guru di SMPN 2 Nglegok sebagai lokasi penelitian. Dataset mencakup tujuh kategori aksara Jawa, yaitu aksara dasar, pasangan, sandhangan, angka, murda, swara, dan rekan, sehingga dapat merepresentasikan berbagai variasi karakter aksara Jawa yang digunakan dalam penelitian.

2.4. *Pre-Processing Data*

Pada tahap ini dilakukan pembersihan data, penyeragaman ukuran citra, serta augmentasi berupa rotasi, shear, shift, zoom, dan penambahan noise ringan. Pelabelan dilakukan secara

manual untuk menjaga kualitas data, sedangkan dataset publik telah memiliki label yang siap digunakan.

2.5. Perancangan Sistem

Pada bagian ini, sistem yang dirancang merupakan arsitektur model CNN yang akan digunakan sebagai algoritma identifikasi aksara Jawa. Model yang digunakan adalah CNN kostumisasi dengan *empat blok konvolusi dengan Batch Normalization, Dropout, Global Average Pooling, dan Fully Connected Classifier dengan optimizer AdamW*. Model CNN ini juga dilengkapi dengan proses tambahan yaitu *fine-tuning* agar performa model menjadi lebih kuat.

2.6. Implementasi Sistem

Bagian ini merupakan proses *training* yang dilakukan oleh model CNN yang sudah dirancang. Terdapat dua fase yaitu, fase *training*, di mana data yang sudah disiapkan akan dilatih melalui 50 *epoch*, serta proses pelatihan ulang atau *fine-tuning* dari *checkpoint* model latih dengan 15 *epoch*. Kemudian pada implementasi program CNN dengan media *canvas* Python, dilakukan pada *library Tkinter* di Visual Studio Code dengan program berbentuk prototipe sederhana.

2.7. Pengujian Sistem

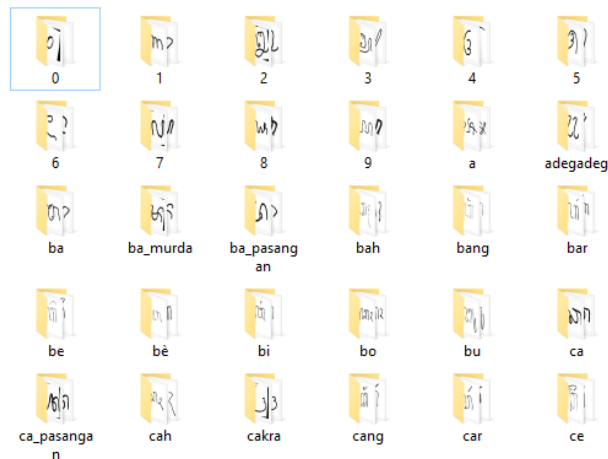
Pada bagian ini, model yang sudah disusun akan diuji menggunakan metrik evaluasi dengan nilai akurasi, presisi, *recall*, dan *f1-score* yang kemudian diperlihatkan pada grafik *confusion matrix* untuk mengukur kemampuan model dalam mengenali data aksara Jawa.

III. HASIL DAN PEMBAHASAN

Berdasarkan hasil eksperimen dan pencocokan pada jenis data, augmentasi, dan model *Convolutional Neural Network* (CNN), hal-hal yang dapat dijabarkan adalah sebagai berikut.

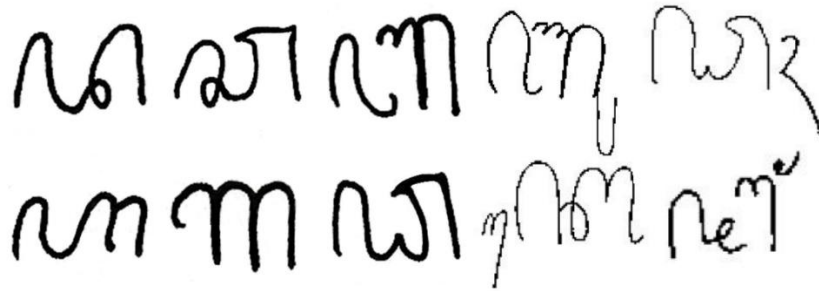
3.1. Hasil Pengumpulan dan Pre-processing Data

Data penelitian diperoleh melalui pengisian formulir penulisan aksara Jawa secara manual dan pemanfaatan dataset publik. Dataset diklasifikasikan ke dalam tujuh kategori utama, yaitu aksara dasar (20 huruf), aksara pasangan (20 huruf), sandhangan (20 karakter), aksara angka (10 karakter), aksara murda (8 huruf), aksara swara (5 huruf), dan aksara rekan (5 huruf). Pengelompokan ini dilakukan untuk memastikan representasi berbagai jenis aksara Jawa dalam proses pelatihan model. Selain data yang dikumpulkan secara mandiri, penelitian juga memanfaatkan dataset publik dari Kaggle.com sebagai sumber tambahan untuk memperkaya variasi data dan meningkatkan kemampuan model dalam mengenali beragam bentuk tulisan aksara Jawa.



Gambar 3. Sampel Folder Data

Data diedit dengan menyamakan ukuran, membersihkan garis atau titik yang tidak perlu. Dataset yang digunakan berupa kumpulan gambar aksara Jawa tulisan tangan dengan ukuran awal 200 x 200 piksel. Dari proses *editing* data, didapatkan total 1764 data dari data primer yang telah dikumpulkan. Sementara data sekunder dari Kaggle.com diambil sekitar 9,360 data.



Gambar 4. Sampel Data Primer dan Sekunder

Setelah data dibersihkan dan dilabeli berdasarkan jenis masing-masing, data kemudian melalui proses augmentasi, agar data lebih bervariasi dengan augmentasi yaitu rotasi, shear (miring), kondisi shift, zoom, dan ditambahkan juga noise ringan. Kemudian data menjadi data train, test, dan validation dengan rasio perbandingan 70:15:15, 70% data latih (8784), 15% data uji (1952), dan 15% data validasi (1952). Proses-proses tersebut dilakukan pada Google Colab yang terintegrasi dengan Google Drive, sehingga proses menjadi lebih fleksibel dan cepat.

3.2. Hasil Perancangan Sistem

Pada perancangan sistem, dibuat model CNN dengan empat blok konvolusi dengan *Batch Normalization*, *Dropout*, *Global Average Pooling*, dan *Fully Connected Classifier* dengan *optimizer AdamW* untuk klasifikasi 244 kelas aksara Jawa dari dataset gambar yang ada di Google Drive. Hal ini bertujuan agar CNN cocok dengan dataset yang sudah ada dan berjalan lebih cepat saat proses *training* di Google Colab.

Layer (type:depth-idx)	Output Shape	Param #
CustomCNN	[1, 244]	--
Sequential: 1-1	[1, 256, 4, 4]	--
ConvBlock: 2-1	[1, 32, 32, 32]	--
Sequential: 3-1	[1, 32, 32, 32]	10,208
ConvBlock: 2-2	[1, 64, 16, 16]	--
Sequential: 3-2	[1, 64, 16, 16]	55,552
ConvBlock: 2-3	[1, 128, 8, 8]	--
Sequential: 3-3	[1, 128, 8, 8]	221,696
ConvBlock: 2-4	[1, 256, 4, 4]	--
Sequential: 3-4	[1, 256, 4, 4]	885,760
AdaptiveAvgPool2d: 1-2	[1, 256, 1, 1]	--
Sequential: 1-3	[1, 244]	--
Flatten: 2-5	[1, 256]	--
Linear: 2-6	[1, 512]	131,584
BatchNorm1d: 2-7	[1, 512]	1,024
ReLU: 2-8	[1, 512]	--
Dropout: 2-9	[1, 512]	--
Linear: 2-10	[1, 244]	125,172
Total params: 1,430,996		
Trainable params: 1,430,996		
Non-trainable params: 0		
Total mult-adds (Units.MEGABYTES): 211.42		
Input size (MB): 0.05		
Forward/backward pass size (MB): 7.87		
Params size (MB): 5.72		
Estimated Total Size (MB): 13.65		

Gambar 5. Model CNN

Model CNN menerima citra berukuran 64×64 piksel dengan 3 kanal RGB dan diproses dalam batch berukuran 64. Arsitektur terdiri atas empat *ConvBlock* dengan pola *Conv2D–Batch Normalization–ReLU–Conv2D–Batch Normalization–ReLU–MaxPooling–Dropout*. Jumlah filter meningkat secara bertahap dari 32, 64, 128, hingga 256 untuk mengekstraksi fitur yang semakin kompleks, sementara dimensi spasial citra berkurang dari 64×64 menjadi 4×4. Setelah

proses ekstraksi fitur, digunakan *Global Average Pooling (AdaptiveAvgPool2d)* untuk merangkum setiap *feature map* menjadi satu nilai sehingga lebih efisien dan membantu mengurangi *overfitting*. Pada tahap klasifikasi, fitur diproses melalui *layer fully connected* 256→512 yang dilengkapi Batch Normalization, ReLU, dan Dropout 0,5, kemudian diteruskan ke *output layer* 512→244 untuk menghasilkan prediksi 244 kelas aksara Jawa. Pada proses pelatihan model menggunakan *optimizer AdamW* dengan *learning rate* 1×10^{-3} (0,001) yang membantu mempercepat konvergensi sekaligus mengurangi *overfitting* melalui mekanisme *weight decay*. Model CNN ini dilengkapi dengan *fine-tuning* setelah proses *training* sehingga performa model lebih bagus.

3.3. Hasil Implementasi Sistem

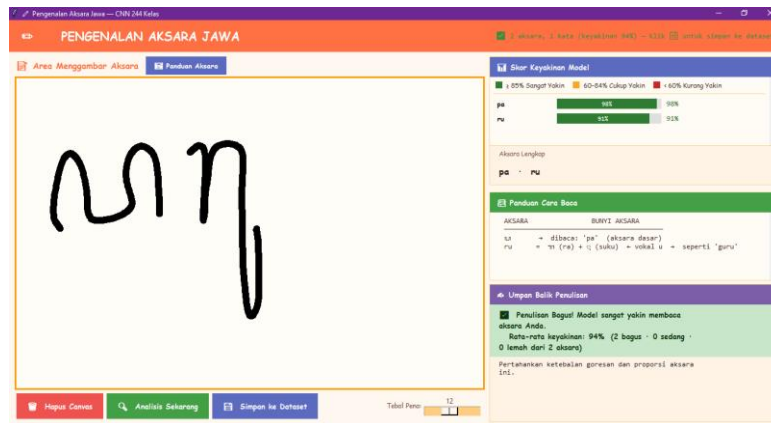
Proses pelatihan dilakukan selama 50 *epoch* dengan *scheduler CosineAnnealingLR* untuk *learning rate* turun mengikuti kurva *cosinus* dan menghindari *local minimum*. Digunakan juga *loss function* dengan pilihan *CrossEntropyLoss* ditambah dengan *label smoothing* 0.1 untuk mencegah model terlalu *overconfident*. Nilai akurasi validasi pada proses *training* mencapai 91.65% pada sebelum proses *fine-tuning*. Kemudian dilakukan pelatihan ulang lagi dengan *fine-tuning* 15 *epoch* yang kemudian meningkatkan nilai akurasi validasi menjadi 94.4%.

Ep	TrLoss	TrAcc	VLoss	VLAcc					
1	4.7572	2.4%	4.7390	1.7% + best	26	1.2283	86.5%	1.0563	91.8%
2	4.1148	5.7%	3.4993	12.6% + best	27	1.2149	87.4%	1.0395	92.4% + best
3	3.5906	13.5%	2.8276	35.0% + best	28	1.2012	87.7%	1.0311	93.0% + best
4	3.1402	22.3%	2.4847	44.7% + best	29	1.1894	88.3%	1.0330	92.1%
5	2.7990	32.3%	2.1085	59.6% + best	30	1.1858	88.4%	1.0257	93.0% + best
6	2.5155	40.8%	1.8371	65.9% + best	31	1.1720	88.8%	1.0223	92.7%
7	2.2787	49.0%	1.6500	72.3% + best	32	1.1632	89.2%	1.0134	93.7% + best
8	2.0769	56.0%	1.5283	76.7% + best	33	1.1550	89.4%	1.0111	93.9% + best
9	1.9358	60.8%	1.4601	80.2% + best	34	1.1425	89.9%	1.0047	93.8%
10	1.8214	65.6%	1.3842	81.5% + best	35	1.1389	90.3%	1.0014	93.5%
11	1.7329	68.4%	5.1554	15.5%	36	1.1343	90.2%	1.0011	93.5%
12	1.6499	70.9%	1.2772	85.8% + best	37	1.1254	90.4%	0.9950	93.8%
13	1.5795	74.3%	1.2488	86.5% + best	38	1.1172	90.8%	0.9916	93.9% + best
14	1.5443	74.9%	1.2343	87.0% + best	39	1.1170	90.5%	0.9923	94.4% + best
15	1.4856	77.4%	1.2080	86.4%	40	1.1084	91.7%	0.9893	94.0%
16	1.4556	78.3%	1.1742	88.3% + best	41	1.1098	91.2%	0.9832	94.5% + best
17	1.4056	80.5%	1.1496	88.5% + best	42	1.1067	91.6%	0.9805	94.4%
18	1.3775	80.9%	1.1435	89.0% + best	43	1.1077	91.3%	0.9781	94.3%
19	1.3506	82.5%	1.1267	88.1%	44	1.1023	91.4%	0.9749	94.2%
20	1.3224	83.4%	1.1027	90.0% + best	45	1.1019	91.4%	0.9762	94.5%
21	1.3118	83.7%	1.1046	90.6% + best	46	1.0942	91.9%	0.9752	94.4%
22	1.2955	84.4%	1.0824	90.8% + best	47	1.0969	91.6%	0.9758	94.8% + best
23	1.2622	85.5%	1.0632	92.2% + best	48	1.0910	92.0%	0.9731	94.4%
24	1.2530	85.8%	1.0702	90.8%	49	1.0941	91.7%	0.9780	94.5%
25	1.2389	86.9%	1.0546	92.0%	50	1.0970	91.5%	0.9761	94.6%

Gambar 6. Log Epoch Training

Gambar 6 merupakan *log training* 50 *epoch* yang menunjukkan perkembangan nilai *loss* dan *accuracy* pada data *training* dan validasi di setiap *epoch*. Tanda **best** menandakan model mencapai nilai *validation accuracy* tertinggi dan bobotnya disimpan sebagai *checkpoint*. Pada awal pelatihan, performa model masih rendah dengan *training accuracy* 2,4% dan *validation accuracy* 1,7%. Akurasi kemudian meningkat secara bertahap, meskipun pada *epoch* ke-11 terjadi penurunan performa yang ditandai lonjakan *validation loss* menjadi 5,1554 dan turunnya *validation accuracy* menjadi 15,5%. Setelah itu, model kembali menunjukkan peningkatan yang stabil. Menjelang akhir pelatihan, *training accuracy* mencapai sekitar 90% dan *validation accuracy* mendekati 94%. Performa terbaik diperoleh pada *epoch* ke-47 dengan *validation accuracy* 94,8% dan *validation loss* 0,9758.

Kemudian, model CNN yang telah dilatih diimplementasikan ke dalam sebuah program berbasis Python sebagai prototipe awal sistem identifikasi aksara Jawa secara digital. Program ini dibangun menggunakan *library Tkinter* sebagai antarmuka grafis berbasis *canvas*, di mana pengguna dapat menulis aksara Jawa secara langsung menggunakan *mouse* atau perangkat *input* lainnya.



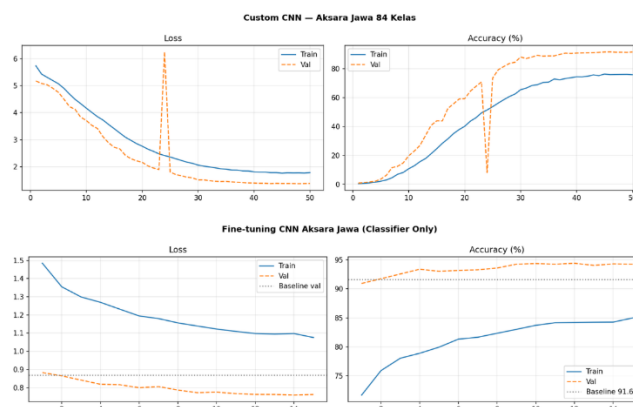
Gambar 7. Prototipe Program Canvas

Gambar 7 menunjukkan hasil implementasi prototipe sistem identifikasi aksara Jawa yang telah dikembangkan. Berdasarkan hasil pengujian, sistem berhasil mengenali aksara yang ditulis pada canvas, seperti “pa” dan “ru”, serta menampilkan hasil identifikasi, bentuk Unicode, dan panduan pembacaan suku kata. Sistem juga menampilkan nilai keyakinan untuk menunjukkan tingkat kepercayaan model terhadap hasil prediksi. Hasil tersebut menunjukkan bahwa model CNN yang telah diintegrasikan mampu melakukan identifikasi aksara Jawa dengan baik.

3.4. Hasil Pengujian

Dari proses *training* 50 *epoch* dan *fine-tuning* 15 *epoch* menghasilkan nilai akurasi, presisi, *recall*, dan *f1-score* yang baik. Nilai akurasi pada proses *training* mencapai 91.65%, dan pada *fine-tuning* akurasi meningkat menjadi 94,3%.

Perbandingan Kurva Akurasi dan Loss Training dan Fine-Tuning



Gambar 8. Grafik Akurasi dan Loss

Grafik pada Gambar 8 menunjukkan bahwa nilai *loss* menurun dan akurasi meningkat selama proses pelatihan. Setelah dilakukan *fine-tuning* selama 15 *epoch*, akurasi validasi meningkat dari sekitar 91,6% menjadi 94,3% disertai penurunan *validation loss*, yang menunjukkan peningkatan performa dan stabilitas model dalam mengenali aksara Jawa. Kemudian, hasil metrik evaluasi model ditunjukkan pada tabel berikut.

Tabel 1. Metrik Evaluasi

Metrik	Nilai
Training	
Accuracy	92.62%

Macro Average Precision	0.93
Macro Average Recall	0.93
Macro Average F1-Score	0.92
Fine-Tuning	
Accuracy	94.83%
Macro Average Precision	0.95
Macro Average Recall	0.95
Macro Average F1-Score	0.95

Berdasarkan evaluasi model, diperoleh accuracy sebesar 92% dengan *test accuracy* mencapai 92.62% dan pada *fine-tuning* berhasil bertambah menjadi 94.83% dengan *macro average precision* dan *recall* sebesar 0.95. Nilai ini diperoleh dari total 1952 data uji yang terdiri dari 244 kelas, yang di setiap kelas memiliki setidaknya 8 sampel (*support*). Dan nilai support menunjukkan jumlah data uji pada masing-masing kelas.

Tabel 2. Tabel *Classification Report*

Kelas	Precisio n	Reca ll	F1- Scor e	Suppo rt	Kelas	Precisio n	Reca ll	F1- Scor e	Suppo rt
0	1.00	0.88	0.93	8	nga_pasangan	0.67	0.50	0.57	8
1	0.50	0.12	0.20	8	ngah	1.00	1.00	1.00	8
2	1.00	1.00	1.00	8	ngang	0.89	1.00	0.94	8
3	1.00	1.00	1.00	8	ngar	0.89	1.00	0.94	8
4	1.00	0.88	0.93	8	nge	1.00	1.00	1.00	8
5	1.00	1.00	1.00	8	ngè	1.00	0.88	0.93	8
6	0.67	0.25	0.36	8	ngi	1.00	0.88	0.93	8
....									
....									
la	0.36	0.50	0.42	8	yo	1.00	1.00	1.00	8
la_pasangan	1.00	0.88	0.93	8	yu	1.00	1.00	1.00	8
lah	1.00	1.00	1.00	8	za	1.00	0.88	0.93	8

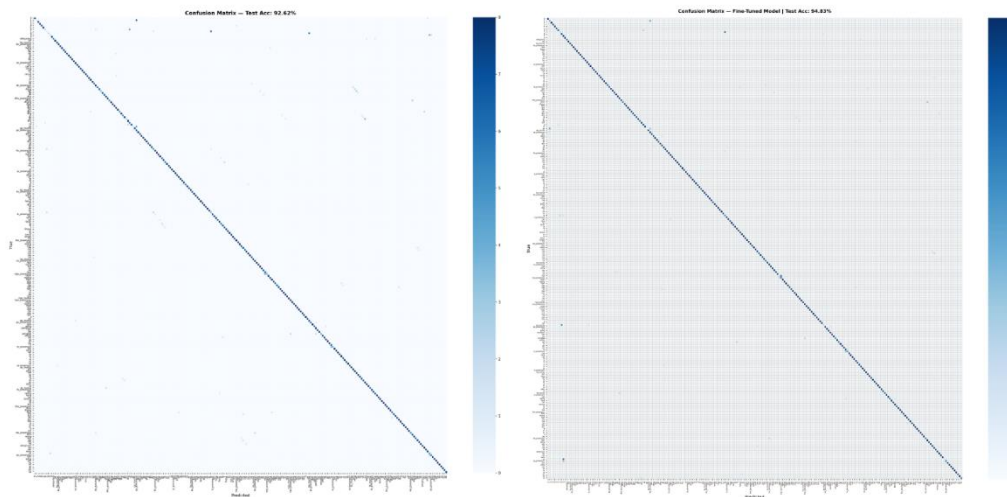
Classification report digunakan untuk mengevaluasi performa model berdasarkan metrik presisi, *recall*, dan *f1-score*. Presisi menunjukkan ketepatan prediksi model terhadap suatu kelas, *recall* menunjukkan kemampuan model menemukan seluruh data yang benar pada kelas tersebut, sedangkan *f1-score* merupakan rata-rata harmonis dari kedua metrik tersebut. Berdasarkan hasil pengujian, model memperoleh *Test Accuracy* sebesar 92,62%, yang menunjukkan kemampuan yang baik dalam mengenali berbagai jenis aksara Jawa.

Sebagian besar kelas memperoleh nilai presisi, *recall*, dan *f1-score* yang tinggi, bahkan mendekati atau mencapai 1,00. Hasil ini menunjukkan bahwa model mampu membedakan karakter aksara Jawa dengan baik pada sebagian besar kelas, baik aksara dasar, pasangan, murda, maupun sandhangan. Namun, beberapa kelas masih menunjukkan performa yang lebih rendah, seperti “1”, “7”, “ga”, “la”, “ya”, dan “ga_pasangan”. Kelas-kelas tersebut memiliki nilai presisi,

recall, atau *f1-score* yang relatif rendah dibandingkan kelas lainnya, menandakan masih terjadinya kesalahan klasifikasi.

Perbedaan performa antar kelas diduga disebabkan oleh kemiripan bentuk visual antaraksara, terutama pada aksara pasangan yang memiliki pola goresan serupa. Meskipun demikian, nilai *macro average* dan *weighted average* yang sama-sama mencapai sekitar 0,93 menunjukkan bahwa secara keseluruhan model telah memiliki performa yang sangat baik dalam mengenali berbagai variasi aksara Jawa. Selanjutnya, hasil klasifikasi tersebut divisualisasikan melalui *confusion matrix*.

Perbandingan Confusion Matrix dan Loss Training dan Fine-Tuning



Gambar 9. Hasil *Confusion Matrix*

Confusion matrix menunjukkan bahwa sebagian besar aksara berhasil diklasifikasikan dengan benar, yang terlihat dari dominasi prediksi pada diagonal utama. Setelah fine-tuning, performa model semakin baik, meskipun beberapa kelas seperti “ga”, “nga”, “la”, “ya”, “1”, “7”, dan “ga_pasangan” masih sering tertukar karena kemiripan bentuk visual. Pengujian pada pengguna juga menunjukkan bahwa sistem tetap mampu mengenali sebagian besar aksara dengan baik meskipun kualitas tulisan tidak selalu konsisten.

IV. KESIMPULAN

Berdasarkan hasil rancangan dan implementasi, sistem identifikasi aksara Jawa berbasis *Convolutional Neural Network* (CNN) dengan media input berupa *canvas* digital menggunakan Python. Model dikembangkan melalui tahapan pra-pemrosesan data, augmentasi, pelatihan, dan implementasi sistem. Arsitektur CNN yang terdiri dari empat blok konvolusi dengan *optimizer AdamW* mampu mengenali pola tulisan aksara Jawa dengan baik. Hasil evaluasi menunjukkan akurasi sebesar 92,62% pada proses pelatihan dan meningkat menjadi 94,83% setelah *fine-tuning* selama 15 epoch. Nilai presisi, *recall*, dan *f1-score* rata-rata mencapai 0,95, yang menunjukkan performa model yang baik dan stabil. Selain itu, pengujian pada program *canvas* menunjukkan bahwa sebagian besar aksara yang ditulis pengguna dapat dikenali dengan tingkat keyakinan yang tinggi, sehingga integrasi model dengan sistem yang dikembangkan telah berjalan dengan baik.

UCAPAN TERIMAKASIH

Penulis mengucapkan terima kasih kepada Prodi Teknik Informatika, Fakultas Teknik dan Informatika, Universitas Islam Balitar yang telah memberi dukungan financial terhadap penelitian ini.

DAFTAR PUSTAKA

- [1] A. Jonathan and I. Wasito, "Perancangan Aplikasi Pengenalan Aksara Jawa Digital Menggunakan Convolutional Neural Network dan Computer Vision," *Decode: Jurnal Pendidikan Teknologi Informasi*, vol. 3, no. 2, pp. 364–377, Aug. 2023, doi: 10.51454/decode.v3i2.209.
- [2] A. A. Pratama, D. Muriyatmoko, and T. Harmini, "SEMINAR NASIONAL CORISINDO Implementasi Algoritma Convolutional Neural Network untuk Koreksi Tulisan pada Aplikasi Pembelajaran Aksara Jawa."
- [3] S. H. Smpn and K. Probolinggo, "ALFABETA: Jurnal Bahasa, Sastra, dan Pembelajarannya Metode Mnemonic dalam Pembelajaran Menulis Aksara Jawa (Kajian Psikolinguistik Lanjut)." [Online]. Available: <http://ejurnal.budiutomomalang.ac.id/index.php/alfabeta>
- [4] Abdussalam, E. Rahmawan Pramudya, A. Susanto, K. Widyatmoko, and R. A. Rico, "APLIKASI MENGENAL AKSARA JAWA DENGAN ALGORITMA SHUFFLE RANDOM BERBASIS ANDROID UNTUK Mendukung BELAJAR MANDIRI," vol. 22, no. 2, p. 2021, 2021.
- [5] K. Azmi, S. Defit, and U. Putra Indonesia YPTK Padang Jl Raya Lubuk Begalung-Padang-Sumatera Barat, "Implementasi Convolutional Neural Network (CNN) Untuk Klasifikasi Batik Tanah Liat Sumatera Barat," vol. 16, no. 1, p. 2023.
- [6] F. Syah, P. W. Ciptadi, A. E. Frinayanti, D. F. Anggraini, E. Amalia, and J. Informatika, "Implementasi CNN untuk Penerjemahan Bahasa melalui Pengenalan Citra Tulisan Tangan," *Jurnal Dinamika Informatika*, vol. 13, no. 2, 2024.
- [7] L. Abdiansah, S. Sumarno, A. Eviyanti, and N. L. Azizah, "Penerapan Algoritma Convolutional Neural Networks untuk Pengenalan Tulisan Tangan Aksara Jawa," *MALCOM: Indonesian Journal of Machine Learning and Computer Science*, vol. 5, no. 2, pp. 496–504, Mar. 2025, doi: 10.57152/malcom.v5i2.1814.
- [8] I. Wijaya, C. Lubis, and K. Kunci, "Jurnal Ilmu Komputer dan Sistem Informasi PENGIMPLEMENTASIAN OCR MENGGUNAKAN CNN UNTUK EKSTRAKSI TEKS PADA GAMBAR."
- [9] F. Nuraeni and A. Pambudi, "IMPLEMENTASI ALGORITMA CONVOLUTIONAL NEURAL NETWORK (CNN) UNTUK PENGENALAN BACAAN TAJWID BERDASARKAN GAMBAR TULISAN DALAM AL-QUR'AN," 2024.
- [10] V. K. Agrawal, S. R. Bogireddy, L. N. Patil, M. Sonekar, Y. M. Patil, and V. S. Panwar, "Optimizing Handwritten Character Recognition Systems: Neural Networks vs. Statistical Methods," in *2024 International Conference on Intelligent Systems and Advanced Applications, ICISAA 2024*, Institute of Electrical and Electronics Engineers Inc., 2024. doi: 10.1109/ICISAA62385.2024.10828993.