

Pengembangan Sistem Otomatis Verifikasi Tanda Tangan Menggunakan *Deep Learning*

Muhammad Riza Randhikata¹

¹Teknik Informatika, Fakultas Teknik dan Ilmu Komputer, Universitas Nusantara PGRI Kediri
E-mail: *¹muhammadrizarandhikata@gmail.com

Abstrak – Verifikasi tanda tangan merupakan proses penting dalam memastikan keaslian identitas pada dokumen resmi, transaksi keuangan, dan proses administrasi. Namun, metode verifikasi manual masih memiliki kelemahan karena bergantung pada subjektivitas manusia sehingga rentan terhadap kesalahan. Penelitian ini bertujuan untuk mengembangkan sistem verifikasi tanda tangan otomatis menggunakan metode Convolutional Neural Network (CNN). Dataset yang digunakan terdiri dari 150 citra tanda tangan dari lima pemilik yang berbeda, dengan penerapan *preprocessing* berupa konversi grayscale, resize, normalisasi, serta augmentasi data. Model CNN dirancang dengan beberapa lapisan konvolusi, max pooling, dan fully connected layer sebelum diintegrasikan ke dalam aplikasi berbasis web menggunakan Streamlit. Evaluasi performa dilakukan menggunakan data validasi dengan parameter akurasi, loss, dan confusion matrix. Model yang dikembangkan mampu mencapai akurasi validasi sebesar 94% dan menunjukkan tingkat generalisasi yang baik dalam mengenali identitas pemilik tanda tangan. Sistem yang diimplementasikan memberikan hasil verifikasi secara otomatis dan interaktif sehingga dapat meningkatkan efisiensi serta keandalan proses autentikasi dokumen di berbagai bidang.

Kata Kunci — Autentikasi Dokumen, CNN, Deep Learning, Streamlit, Verifikasi Tanda Tangan

1. PENDAHULUAN

Tanda tangan merupakan metode autentikasi yang umum digunakan dalam berbagai aktivitas penting seperti penandatanganan dokumen resmi, transaksi keuangan, serta proses administrasi untuk memastikan validitas identitas seseorang [1] [2]. Dalam bidang hukum, khususnya pada ranah perdata dan kenotariatan, tanda tangan berperan sebagai bukti persetujuan, pengesahan, serta bentuk tanggung jawab pihak penandatanganan [3] [4]. Meskipun perannya sangat krusial, proses verifikasi tanda tangan masih banyak dilakukan secara manual [5] dan bergantung pada subjektivitas manusia. Metode ini rentan terhadap kesalahan dalam mengenali keaslian tanda tangan sehingga diperlukan solusi verifikasi yang lebih objektif, efisien, dan andal dalam mendukung keamanan autentikasi dokumen [6].

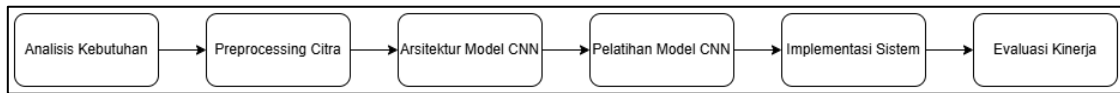
Perkembangan teknologi kecerdasan buatan, khususnya *Deep Learning* dengan pendekatan *Convolutional Neural Network* (CNN), mampu memberikan solusi yang lebih akurat dalam mengenali pola visual kompleks pada citra tanda tangan [7]. CNN dapat mengekstraksi fitur mulai dari pola garis [8] dan kontur hingga karakteristik unik yang merepresentasikan identitas penulis secara otomatis tanpa proses ekstraksi fitur manual. Berbagai penelitian terdahulu telah mengembangkan sistem verifikasi tanda tangan berbasis *machine learning* maupun *deep learning*. Penelitian oleh Kurniawan dkk. [9] menggunakan metode *K-Nearest Neighbor* (KNN) dengan fitur bentuk dan memperoleh akurasi sebesar 98,25%, namun masih terbatas pada klasifikasi bentuk tanda tangan tanpa verifikasi identitas pemilik secara spesifik. Kemudian, penelitian oleh Kemendikbud dkk. [10] mengombinasikan CNN dengan algoritma *XGBoost* untuk melakukan verifikasi tanda tangan dan pengenalan tulisan tangan pada dokumen cek bank. Hasilnya menunjukkan bahwa model hybrid *CNN-XGBoost* dapat meningkatkan akurasi verifikasi dibandingkan penggunaan CNN tunggal.

Meskipun hasil penelitian sebelumnya menunjukkan performa yang baik, sebagian besar masih berfokus pada identifikasi pola tanda tangan tanpa mempertimbangkan verifikasi berbasis identitas pemilik secara langsung serta belum banyak diimplementasikan pada platform aplikasi interaktif yang dapat dimanfaatkan secara umum.

Berdasarkan permasalahan tersebut, penelitian ini mengembangkan sistem verifikasi tanda tangan otomatis berbasis CNN yang mampu mengenali identitas pemilik tanda tangan secara akurat. Sistem diintegrasikan dalam aplikasi berbasis web menggunakan *Streamlit* agar pengguna dapat melakukan verifikasi secara langsung dengan hanya mengunggah citra tanda tangan dan memilih identitas pemiliknya. Penelitian ini diharapkan memberikan kontribusi dalam peningkatan keamanan autentikasi dokumen dan mendukung transformasi digital pada berbagai layanan yang membutuhkan validasi identitas secara terpercaya dan efisien.

2. METODE PENELITIAN

Metode penelitian ini disusun melalui beberapa tahapan yang terstruktur untuk memastikan proses verifikasi tanda tangan berbasis CNN dapat berjalan secara optimal. Tahapan utama meliputi analisis kebutuhan, *preprocessing* citra, perancangan arsitektur CNN, pelatihan model, implementasi ke dalam sistem web, serta evaluasi performa model. Alur lengkap penelitian ini dapat dilihat pada Gambar 1.



Gambar 1. Alur Penelitian Verifikasi Tanda Tangan Berbasis CNN

2.1 Analisis Kebutuhan

Tahap awal penelitian adalah melakukan analisis kebutuhan terhadap dataset, perangkat keras, perangkat lunak, dan metode yang digunakan. Dataset berupa citra tanda tangan dari lima pengguna dengan jumlah sampel yang cukup untuk proses pelatihan dan validasi model.

Perangkat lunak yang digunakan antara lain *Python* sebagai bahasa pemrograman, *TensorFlow* sebagai library utama untuk membangun model CNN, dan *Streamlit* sebagai antarmuka sistem berbasis *web*. Hasil analisis kebutuhan ini menjadi dasar dalam pemilihan metode pengolahan citra serta desain arsitektur model yang sesuai.

2.2 Preprocessing Citra

Agar citra tanda tangan yang digunakan dalam proses pelatihan memiliki format yang seragam dan siap diolah oleh model, dilakukan *preprocessing* yang mencakup beberapa tahap pengolahan citra. Tahap pertama adalah mengubah ukuran citra menjadi 128×128 piksel dan mengonversinya ke dalam *grayscale* sehingga informasi visual yang diproses lebih sederhana namun tetap mewakili bentuk tanda tangan secara utuh. Selanjutnya dilakukan normalisasi nilai piksel menjadi rentang 0 hingga 1 untuk meningkatkan stabilitas dan kecepatan proses pembelajaran pada model CNN. Proses normalisasi ini dinyatakan dalam Persamaan (1).

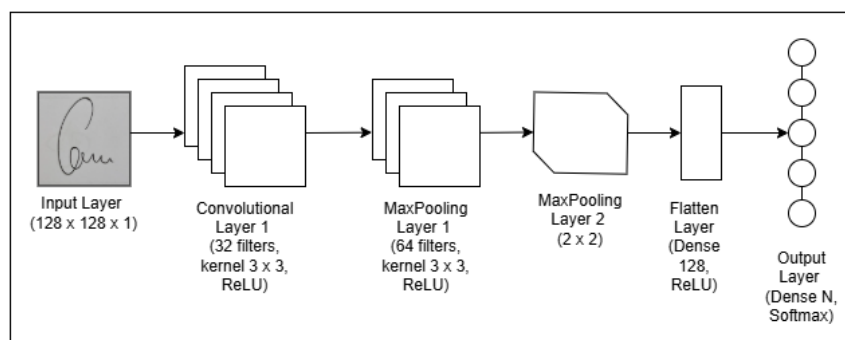
$$x_{norm} = \frac{x}{255} \dots \dots \dots (1)$$

Di mana X merupakan nilai intensitas piksel asli dan X_{norm} adalah nilai piksel hasil normalisasi. Selain itu, digunakan teknik augmentasi seperti rotasi dan pergeseran kecil untuk meningkatkan variasi data sehingga model mampu beradaptasi dan mengenali pola tanda tangan dalam berbagai kondisi input.

2.3 Arsitektur Model CNN

Arsitektur model CNN yang digunakan dalam penelitian ini dirancang untuk mempelajari pola visual unik dari setiap tanda tangan. Model menerima input berupa citra *grayscale* berukuran 128×128 piksel [11], kemudian diproses melalui beberapa lapisan konvolusi dengan fungsi aktivasi *ReLU* untuk mengekstraksi fitur garis dan tekstur. Setelah itu, lapisan *MaxPooling* diterapkan untuk mereduksi dimensi [12] dan mengurangi kompleksitas komputasi [13]. Lapisan konvolusi dengan jumlah filter yang lebih besar digunakan untuk menangkap fitur tingkat lanjut dari tanda tangan [14].

Hasil ekstraksi fitur tersebut kemudian diratakan melalui *Flatten Layer* dan dihubungkan ke *Fully Connected Layer* dengan aktivasi *ReLU* [15]. *Dropout* dengan nilai 0.5 diterapkan untuk mengurangi risiko overfitting. Pada bagian output, digunakan *Softmax* untuk mengklasifikasikan citra tanda tangan ke dalam kelas pemilik yang sesuai. Hyperparameter utama yang digunakan meliputi *optimizer Adam* dengan *learning rate* 0.0001 [16] dan fungsi *loss categorical cross-entropy*. Struktur lengkap jaringan CNN ini ditunjukkan pada Gambar 2.



Gambar 2. Arsitektur Model CNN

2.4 Pelatihan Model CNN

Model dilatih menggunakan data *training* yang telah melalui proses augmentasi, sedangkan data validasi digunakan untuk memantau kinerja model selama pelatihan. Proses pelatihan dilakukan selama beberapa *epoch* hingga model mencapai nilai akurasi yang optimal dan *loss* yang stabil. *Callback* seperti *EarlyStopping* dan *ModelCheckpoint* diterapkan untuk mencegah overfitting berlebihan serta menyimpan model terbaik berdasarkan nilai validasi.

2.5 Implementasi Sistem

Setelah model CNN berhasil dilatih, model diintegrasikan ke dalam aplikasi berbasis *web* menggunakan *Streamlit*. Implementasi ini bertujuan untuk memudahkan pengguna dalam melakukan proses verifikasi tanda tangan secara langsung. Pada aplikasi ini, pengguna dapat mengunggah gambar tanda tangan, kemudian sistem akan memproses citra tersebut dan menampilkan hasil prediksi mengenai identitas pemilik tanda tangan secara otomatis dan interaktif.

2.6 Evaluasi Kinerja

Tahap evaluasi dilakukan untuk mengukur performa model dalam mengenali tanda tangan pada data validasi yang belum pernah dilihat sebelumnya. Parameter evaluasi yang digunakan antara lain nilai akurasi dan *loss* yang diperoleh selama proses pelatihan, serta *confusion matrix* yang digunakan untuk menganalisis tingkat keberhasilan model dalam mengklasifikasikan setiap kelas tanda tangan. Hasil evaluasi ini menjadi dasar untuk menilai kemampuan generalisasi model dan mengidentifikasi potensi perbaikan yang diperlukan. Selain itu, untuk mengukur performa klasifikasi secara lebih detail pada setiap kelas tanda tangan, digunakan metrik evaluasi berupa *precision*, *recall*, *accuracy*, dan *F1-Score*. Perhitungan setiap metrik tersebut dijelaskan pada Persamaan (2) hingga (5).

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

$$Precision = \frac{TP}{TP+FP} \dots\dots\dots(3)$$

$$Recall = \frac{TP}{TP+FN} \dots\dots\dots(4)$$

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

Di mana TP (*True Positive*), FP (*False Positive*), FN (*False Negative*), dan TN (*True Negative*) merupakan nilai-nilai yang diperoleh melalui *confusion matrix*.

3. HASIL DAN PEMBAHASAN

3.1 Dataset dan Input Gambar

Dataset yang digunakan dalam penelitian ini merupakan kumpulan citra tanda tangan yang diperoleh dari lima pemilik tanda tangan yang berbeda. Setiap pemilik menyediakan sebanyak 30 citra, sehingga total dataset yang digunakan berjumlah 150 citra dengan format PNG atau JPG. Seluruh citra memiliki variasi dalam pencahayaan, ukuran, dan gaya tulisan, sehingga diperlukan tahap praproses agar citra memiliki kualitas yang lebih seragam sebelum digunakan oleh model.

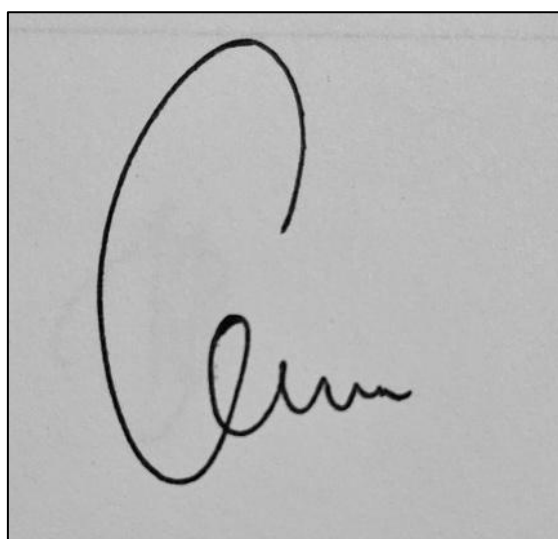
Dataset dibagi menjadi dua bagian utama, yaitu data latih dan data validasi dengan jumlah 20 citra untuk data latih dan 10 citra untuk data validasi pada setiap identitas. Dengan demikian, rasio pembagian dataset menjadi 66.7% untuk data latih dan 33.3% untuk data validasi. Pembagian data ini dilakukan untuk memastikan model dapat dilatih dengan baik sekaligus diuji pada data yang tidak digunakan saat pelatihan. Data latih digunakan dalam proses pembelajaran model CNN untuk mengenali karakteristik tanda tangan dari masing-masing pemilik, sementara data uji digunakan pada tahap evaluasi untuk mengukur kemampuan model dalam melakukan verifikasi pada citra tanda tangan baru yang belum pernah dilihat sebelumnya. Pembagian yang seimbang pada setiap kelas diharapkan dapat meminimalkan bias model terhadap salah satu kelas tertentu. Deskripsi pembagian dataset berdasarkan jumlah sampel pada setiap identitas pemilik ditunjukkan pada Tabel 1.

Tabel 1. Deskripsi Dataset Tanda Tangan

Identitas Pemilik	Total Citra	Train	Validation	Format Citra
Gama	30	20	10	PNG/JPG
Eka	30	20	10	PNG/JPG
Rama	30	20	10	PNG/JPG
Rafli	30	20	10	PNG/JPG
Rizki	30	20	10	PNG/JPG
Total Keseluruhan	150	100	50	

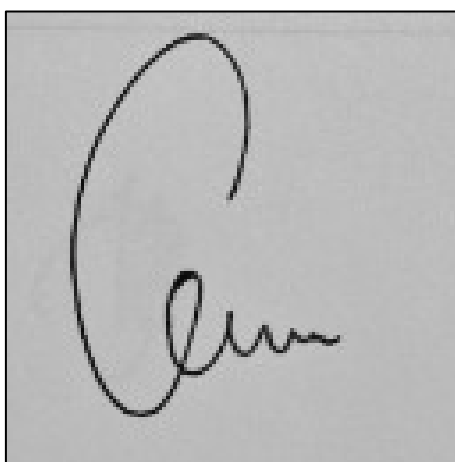
3.2 Praproses Data (*Preprocessing*)

Tahap praproses dilakukan untuk menyeragamkan citra tanda tangan dan meningkatkan kualitas fitur visual yang akan diekstraksi oleh model CNN. Pada penelitian ini terdapat tiga tahap utama yang diterapkan secara berurutan sebelum citra digunakan dalam pelatihan model. Pertama, citra dikonversi ke *grayscale* untuk menghilangkan informasi warna yang tidak diperlukan sehingga jaringan dapat berfokus pada pola garis, lengkungan, dan struktur utama tanda tangan yang menjadi ciri khas identitas penulis. Contoh hasil konversi citra tanda tangan ke dalam *grayscale* dapat dilihat pada Gambar 3.



Gambar 3. Citra tanda tangan hasil konversi *grayscale*

Selanjutnya, citra diubah ukurannya menjadi 128×128 piksel seperti ditunjukkan pada Gambar 4. Ukuran ini dipilih karena mampu menjaga detail tanda tangan sekaligus efisien dalam komputasi sehingga proses pelatihan model menjadi lebih cepat dan stabil.



Gambar 4. Citra tanda tangan hasil resize 128×128 piksel

Tahap terakhir dari proses praproses adalah normalisasi piksel, yaitu mengubah nilai intensitas piksel dari rentang 0–255 menjadi rentang 0–1. Tahap ini bertujuan untuk menjaga stabilitas pelatihan serta mempercepat

proses konvergensi model sehingga risiko error akibat perbedaan skala nilai piksel dapat diminimalkan. Contoh perhitungan normalisasi pada beberapa nilai piksel ditunjukkan pada Tabel 2, di mana nilai normalisasi dihitung menggunakan persamaan 1.

Berdasarkan persamaan 1 tersebut, Contoh hasil normalisasi intensitas piksel pada beberapa nilai ditunjukkan pada Tabel 2.

Tabel 2. Tabel Contoh Normalisasi Nilai Piksel Citra

Nilai Piksel Asli (0-255)	Rumus Normalisasi	Hasil Normalisasi (0-1)
0	$0 \div 255$	0.000
10	$10 \div 255$	0.039
128	$128 \div 255$	0.502
200	$200 \div 255$	0.784
255	$255 \div 255$	1.000

Nilai piksel yang ditampilkan pada tabel tersebut hanya merupakan contoh perhitungan pada beberapa titik piksel tertentu untuk menggambarkan proses normalisasi. Pada implementasinya, normalisasi diterapkan pada seluruh piksel yang terdapat dalam citra tanda tangan sehingga menghasilkan citra yang siap digunakan sebagai input model CNN.

Melalui tahapan praproses ini, kualitas citra tanda tangan menjadi lebih seragam dari segi dimensi dan intensitas visual, sehingga model CNN mampu mempelajari pola tanda tangan secara lebih optimal serta meningkatkan akurasi dan kemampuan generalisasi pada data baru.

3.3 Pelatihan Model (*Training*)

Pada tahap ini dilakukan proses pelatihan model *Convolutional Neural Network* (CNN) untuk mempelajari pola visual unik pada citra tanda tangan. Pelatihan dilakukan menggunakan data latih yang sebelumnya telah melalui proses praproses. Proses pelatihan dilakukan menggunakan 100 data latih, sedangkan 50 data digunakan sebagai data validasi untuk mengukur performa model pada data baru.

Pada proses pelatihan, CNN melakukan ekstraksi fitur secara otomatis melalui beberapa lapisan konvolusi yang mendeteksi pola garis, lengkungan, ketebalan stroke, kontur, dan karakteristik visual lain yang terdapat pada tanda tangan. Setelah fitur berhasil diekstraksi, proses dilanjutkan pada tahap klasifikasi untuk memetakan fitur yang telah diperoleh ke dalam identitas pemilik tanda tangan yang sesuai.

Selama proses *training*, parameter model diperbarui menggunakan algoritma optimasi *Adam*. Optimasi dilakukan secara iteratif berdasarkan nilai *loss* dan akurasi pada setiap *epoch*, sehingga performa model meningkat secara bertahap. Output dari proses pelatihan ini adalah model CNN terlatih yang dapat mengenali pola tanda tangan dan melakukan verifikasi identitas secara otomatis. Parameter pelatihan yang digunakan dalam model CNN ditunjukkan pada Tabel 3 berikut.

Tabel 3. Parameter Pelatihan Model CNN

Parameter	Nilai
Epoch	30
Batch Size	16
Optimizer	Adam
Learning Rate	0.001
Loss Function	Categorical Crossentropy
Validation Split	20%
Ukuran Input Citra	128 x 128
Jumlah Kelas	5

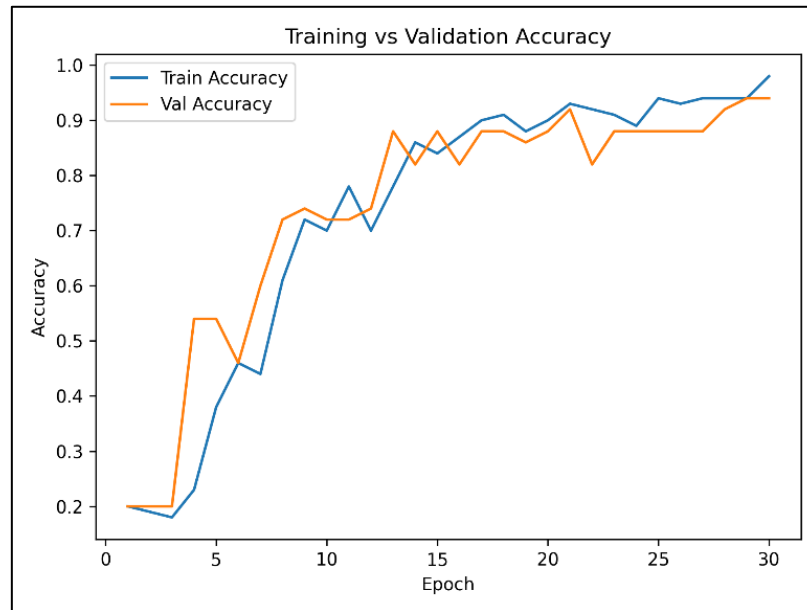
3.4 Implementasi Sistem

Implementasi model CNN dilakukan menggunakan *Streamlit* sebagai antarmuka aplikasi berbasis *web* yang dapat digunakan oleh pengguna secara langsung. Aplikasi dirancang agar mudah dioperasikan dengan alur penggunaan yang jelas, mulai dari pemilihan identitas pemilik tanda tangan, unggah citra tanda tangan, hingga proses verifikasi secara otomatis oleh sistem.

3.5 Evaluasi Kinerja

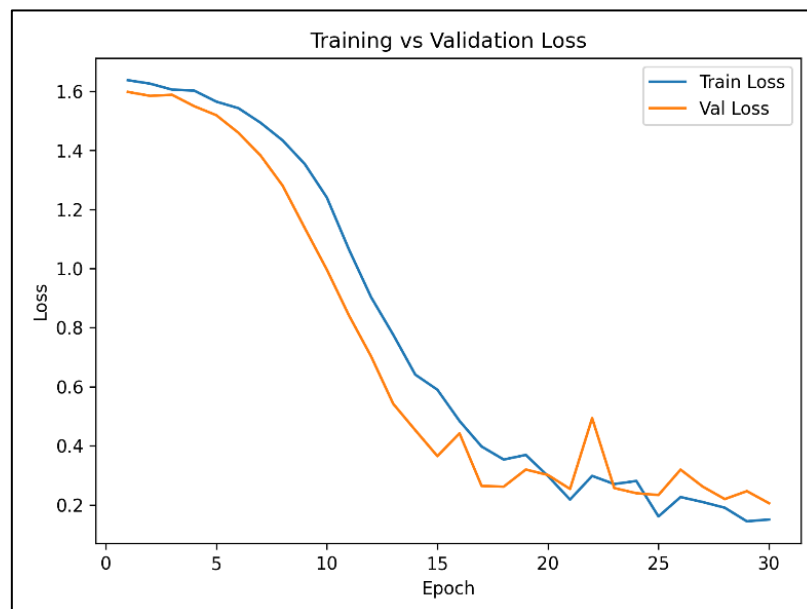
Evaluasi dilakukan untuk mengukur kemampuan model dalam mengenali identitas tanda tangan pada data validasi. Grafik akurasi menunjukkan bahwa nilai akurasi pelatihan dan validasi meningkat secara stabil selama proses pelatihan. Pada *epoch* akhir, model mencapai akurasi pelatihan sekitar 97% dan akurasi validasi sekitar 94%. Nilai akurasi validasi yang tinggi dan konsisten menunjukkan bahwa model mampu melakukan

klasifikasi tanda tangan dengan baik pada data yang belum pernah dilihat sebelumnya. Grafik perkembangan akurasi pelatihan dan validasi selama proses *training* ditunjukkan pada Gambar 5.



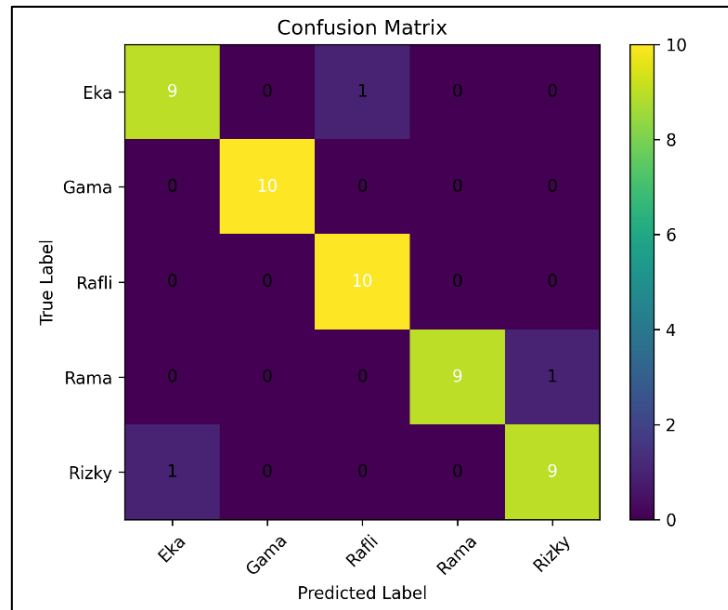
Gambar 5. *Training vs Validation Accuracy*

Selain akurasi, grafik nilai *loss* juga menunjukkan tren penurunan yang signifikan pada kedua data pelatihan dan validasi. *Loss* terus menurun hingga mencapai nilai yang rendah di akhir *epoch*, dengan jarak yang sangat kecil antara *training loss* dan *validation loss*. Kondisi ini mengindikasikan bahwa model telah belajar secara optimal tanpa mengalami gejala *overfitting*. Tren penurunan nilai *loss* selama pelatihan dan validasi ditampilkan pada Gambar 6.



Gambar 6. *Training vs Validation Loss*

Pengukuran performa klasifikasi pada setiap kelas, digunakan *confusion matrix* yang ditunjukkan pada Gambar di bawah ini. Hasil evaluasi menunjukkan bahwa sebagian besar kelas dapat dikenali secara tepat oleh model. Kelas Gama, Rafli, dan Rizky berhasil terklasifikasi dengan akurasi sempurna. Hanya terdapat sedikit kesalahan prediksi pada kelas Eka dan Rama, masing-masing sebanyak satu sampel yang salah terklasifikasi ke kelas lain. Secara umum, tingkat kesalahan yang rendah tersebut masih dapat diterima dan menunjukkan bahwa model memiliki performa yang sangat baik dalam melakukan verifikasi identitas tanda tangan. Evaluasi klasifikasi pada setiap kelas pemilik tanda tangan digambarkan melalui *confusion matrix* pada Gambar 7.



Gambar 7. *Confusion Matrix*

Selain melihat distribusi prediksi melalui *confusion matrix*, diperlukan pula analisis metrik evaluasi yang lebih detail untuk setiap kelas tanda tangan. Perhitungan metrik seperti *precision*, *recall*, dan *F1-Score* digunakan untuk mengetahui kemampuan model dalam membedakan pola tanda tangan pada masing-masing pemilik secara lebih akurat dan menyeluruh. Nilai metrik evaluasi untuk setiap kelas tanda tangan ditunjukkan pada Tabel 4 berikut.

Tabel 4. Metrik Evaluasi Model CNN pada Setiap Kelas

Kelas	Precision	Recall	F1-Score	Support
Eka	0.90	0.90	0.90	10
Gama	1.00	1.00	1.00	10
Rafli	0.91	1.00	0.95	10
Rama	1.00	0.90	0.95	10
Rizki	0.90	0.90	0.90	10
Rata-rata	0.94	0.94	0.94	10

Berdasarkan hasil evaluasi pada Tabel 4, model CNN menunjukkan performa yang sangat baik dengan nilai rata-rata *precision*, *recall*, dan *F1-Score* sebesar 0.94. Hal ini membuktikan bahwa model mampu mengidentifikasi pola tanda tangan dari setiap pemilik dengan tingkat akurasi yang tinggi dan konsisten. Kesalahan klasifikasi terjadi pada kelas Eka, Rama, dan Rizki, yaitu masing-masing satu sampel yang salah terprediksi. Kondisi ini mengindikasikan adanya kemiripan pola tertentu pada ciri visual tanda tangan ketiga kelas tersebut. Meskipun demikian, tingkat kesalahan yang rendah dan nilai *F1-Score* yang tinggi menunjukkan bahwa model memiliki kemampuan generalisasi yang cukup kuat dalam melakukan verifikasi identitas tanda tangan secara otomatis.

4. SIMPULAN

Penelitian ini berhasil mengembangkan sistem verifikasi tanda tangan berbasis *Convolutional Neural Network* (CNN) yang mampu mengenali pola visual tanda tangan dari berbagai pemilik secara akurat. Model CNN yang diterapkan menunjukkan performa tinggi dengan akurasi validasi sekitar 94% dan nilai *loss* yang rendah serta stabil pada akhir pelatihan. *Confusion matrix* menunjukkan bahwa sebagian besar tanda tangan dapat diklasifikasikan dengan benar, sehingga model memiliki tingkat generalisasi yang baik.

Implementasi model ke dalam antarmuka berbasis *web* menggunakan *Streamlit* memberikan kemudahan bagi pengguna untuk melakukan verifikasi tanda tangan secara langsung. Sistem mampu menampilkan hasil prediksi identitas pemilik tanda tangan beserta tingkat keyakinannya secara otomatis. Dengan hasil tersebut, dapat disimpulkan bahwa CNN efektif digunakan dalam membangun sistem verifikasi tanda tangan otomatis dan berpotensi untuk diterapkan pada proses autentikasi dokumen di berbagai bidang.

5. SARAN

Dalam upaya meningkatkan performa dan ketahanan sistem verifikasi tanda tangan pada berbagai kondisi nyata, penelitian selanjutnya dapat memperluas cakupan dataset dengan variasi bentuk tulisan, ketebalan tinta, gaya penulisan, serta kondisi pencahayaan yang berbeda. Pengembangan arsitektur model atau penyesuaian parameter pelatihan juga dapat dilakukan untuk memperoleh hasil yang lebih optimal pada skala data yang lebih besar. Selain itu, penggunaan metrik evaluasi tambahan seperti *precision*, *recall*, dan *F1-score* dapat memberikan analisis performa yang lebih detail pada setiap kelas tanda tangan. Dengan pengembangan berkelanjutan, sistem verifikasi tanda tangan ini diharapkan mampu memberikan tingkat akurasi yang lebih tinggi serta kinerja yang lebih andal ketika diimplementasikan pada lingkungan operasional yang sesungguhnya.

DAFTAR PUSTAKA

- [1] F. Irianti and S. Rahman, “Journal of Lex Philosophy (JLP),” vol. 5, 2024.
- [2] T. N. Cahyadi, “Aspek Hukum Pemanfaatan Digital Signature Dalam Meningkatkan Efisiensi, Akses Dan Kualitas Fintech Syariah,” *J. Rechts Vinding Media Pemb. Huk. Nas.*, vol. 9, no. 2, p. 219, 2020.
- [3] P. D. I. Pengadilan, “Umbas, Sita Arini. 2017. ‘Kedudukan Akta Di Bawah Tangan Yang Telah Dilegalisasi Notaris Dalam Pembuktian Di Pengadilan.’ Jurnal Lex Crimen Vol:6, No.1,” vol. VI, no. 1, pp. 79–87, 2017.
- [4] R. F. Mayana and T. Santika, “Legalitas tanda tangan elektronik: posibilitas dan tantangan notary digitalization di Indonesia,” *ACTA DIURNAL J. Ilmu Huk. Kenotariatan*, vol. 4, no. 2, pp. 244–262, 2021.
- [5] R. M. A. S. Ikhsan and R. Kusuma, “Kedudukan hukum barcode pada tanda tangan notaris dalam sistem peraturan perundang-undangan di Indonesia,” *Nusant. J. Ilmu Pengetah. Sos.*, vol. 12, no. 6, pp. 2495–2504, 2025.
- [6] P. Khobragade and U. Gawande, “A comprehensive review on CNN-based applications for medical imaging classification and segmentation,” *AIP Conf. Proc.*, vol. 3188, no. 1, Dec. 2024, doi: 10.1063/5.0244772.
- [7] M. Kurniawan, N. Saidatin, H. Nugroho, I. Teknologi, and A. Tama, “Implementasi Shape Feature dan K-Nearest Neighbor untuk Klasifikasi Tanda Tangan,” pp. 155–162.
- [8] I. Maulana, N. Khairunisa, and R. Mufidah, “Deteksi bentuk wajah menggunakan convolutional neural network (CNN),” *JATI (Jurnal Mhs. Tek. Inform.)*, vol. 7, no. 6, pp. 3348–3355, 2023.
- [9] T. Kemendikbud *et al.*, “Identifikasi Tanda Tangan Dengan Menggunakan Metode Convolution Neural Network (CNN),” vol. i, no. 204, pp. 138–147, 2022.
- [10] U. I. Batam, F. I. Komputer, P. S. Informasi, J. G. Mada, and T. Indah, “PENGEMBANGAN VERIFIKASI INFORMASI CEK BANK DENGAN MENGGUNAKAN CNN-XGBOOST IMAGE CLASSIFICATION UNTUK VERIFIKASI TANDA TANGAN DAN PENGENALAN TULISAN TANGAN,” vol. 5, no. 2, pp. 162–172, 2024.
- [11] I. Wulandari, H. Yasin, and T. Widiari, “Klasifikasi citra digital bumbu dan rempah dengan algoritma convolutional neural network (cnn),” *J. Gaussian*, vol. 9, no. 3, pp. 273–282, 2020.
- [12] A. Peryanto, A. Yudhana, and R. Umar, “Rancang bangun klasifikasi citra dengan teknologi deep learning berbasis metode convolutional neural network,” *Format J. Ilm. Tek. Inf.*, vol. 8, no. 2, p. 138, 2020.
- [13] Y. K. Bintang and H. Imaduddin, “Pengembangan model deep learning untuk deteksi retinopati diabetik menggunakan metode transfer learning,” *JUPI (Jurnal Ilm. Penelit. dan Pembelajaran Inform.)*, vol. 9, no. 3, pp. 1442–1455, 2024.
- [14] E. J. A. Sinaga, G. Oktavia, H. Syahputra, and F. Ramadhani, “Identifikasi Tanda Tangan Dengan Menggunakan Metode Convolution Neural Network (CNN),” *J-INTECH*, vol. 12, no. 1, pp. 138–147, 2024.
- [15] R. A. Pangestu, B. Rahmat, and F. T. Anggraeny, “Implementasi algoritma CNN untuk klasifikasi citra lahan dan perhitungan luas,” *J. Inform. dan Sist. Inf.*, vol. 1, no. 1, pp. 166–174, 2020.

- [16] N. Rochmawati, H. B. Hidayati, Y. Yamasari, H. P. A. Tjahyaningtijas, W. Yustanti, and A. Prihanto, “Analisa *Learning Rate* dan Batch Size pada Klasifikasi Covid Menggunakan Deep Learning dengan Optimizer *Adam*,” *JIEET (Journal Inf. Eng. Educ. Technol.*, vol. 5, no. 2, pp. 44–48, 2021.