PENINGKATAN AKURASI KLASIFIKASI DIGIT MELALUI MODIFIKASI CNN DENGAN *BATCH NORMALIZATION*, *DROPOUT*, DAN *DATA AUGMENTATION*

Aldrey Diriyah[1*], Khalida Zia Qinthara[2], Ajeng Windi Setianingsih[3], Yemima Perangin-Angin[4], I Gde Eka Dirgayussa[5]
[1,2,3,4,5]Teknik Biomedis, Institut Teknologi Sumatera
aldrey.122430054@student.itera.ac.id[1], khalida.123430070@student.itera.ac.id[2]
ajeng.123430093@student.itera.ac.id[3] , yemima.123430083@student.itera.ac.id[4],
i.dirgayussa@bm.itera.ac.id[5]

Submitted November 26, 2025; Revised April 2, 2026; Accepted April 5, 2026

Abstrak

Model *Simple Convolutional Neural Network* (CNN) rentan terhadap *overfitting* dan memiliki kemampuan generalisasi yang terbatas pada klasifikasi citra digit berskala menengah. Meskipun *transfer learning* dan model hibrida dapat meningkatkan kinerja model, pendekatan tersebut umumnya memerlukan kompleksitas dan beban komputasi yang tinggi. Penelitian ini mengusulkan modifikasi arsitektur CNN melalui integrasi pendalaman blok konvolusi, *batch normalization*, *dropout* (0,5), serta terknik *data augmentation* spasial seperti rotasi, translasi, *zoom* untuk meningkatkan performa. Evaluasi dilakukan secara komparatif menggunakan 10.160 citra digit tulisan tangan (0–9) berbasis MNIST dengan pembagian data 80% pelatihan dan 20% validasi. Hasil eksperimen menunjukkan bahwa model *Simple CNN* gagal mengekstraksi fitur diskriminatif (*validation accuracy* 10,00%; *loss* 28,19%), sedangkan model yang diusulkan mencapai *validation accuracy* 97,69%, *loss* 4,80%, dan *F1-score* 97,69%. Hasil temuan ini menunjukkan bahwa optimalisasi arsitektur CNN secara terintegrasi mampu meningkatkan generalisasi model secara signifikan dan tetap menjaga efisiensi komputasi.
Kata Kunci : Augmentasi Data, *Batch Normalization*, CNN, *Dropout*, Klasifikasi Digit

Abstract

Simple Convolutional Neural Network (CNN) model is prone to overfitting and exhibits limited generalization capability in medium-scale digit image classification tasks. Although transfer learning and hybrid models can improve model performance, these approaches generally require high complexity and computational cost. This study proposes a modification of the CNN architecture through the integration of deeper convolutional blocks, batch normalization, dropout (0.5), as well as spatial data augmentation techniques such as rotation, translation, and zoom to enhance performance. The evaluation was conducted comparatively using 10,160 handwritten digit images (0–9) based on the MNIST dataset, with a data split of 80% for training and 20% for validation. The experimental results show that the Simple CNN model fails to extract discriminative features (validation accuracy of 10.00%; loss of 28.19%), whereas the proposed model achieves a validation accuracy of 97.69%, loss of 4.80%, and an F1-score of 97.69%. These findings indicate that integrated optimization of CNN architecture can significantly improve model generalization while maintaining computational efficiency.
Keywords : *batch normalization, CNN, data augmentation, digit classification, dropout*

1. PENDAHULUAN

Seiring dengan perkembangan teknologi *computer vision* dan *deep learning,* klasifikasi citra digit telah menjadi topik yang sangat penting karena relevansinya dalam berbagai aplikasi seperti pengenalan tulisan tangan, otomatisasi dokumen, dan sistem interaksi manusia-mesin [1]. Di antara berbagai metode yang ada, *Convolutional Neural Network* (CNN) telah terbukti efektif dalam mengekstraksi fitur spasial secara hierarkis dan mencapai performa tinggi pada berbagai tugas klasifikasi citra [2], [3]. Meskipun demikian, kinerja CNN sangat dipengaruhi

oleh karakteristik data. Pada kondisi dengan data yang terbatas atau tingginya variasi antar kelas, arsitektur CNN sederhana cenderung mengalami *overfitting* sehingga berdampak pada menurunnya kemampuan generalisasi terhadap data baru [4], [5].

Berbagai upaya telah dilakukan untuk mengatasi keterbatasan tersebut melalui optimasi arsitektur maupun strategi pelatihan. Model CNN standar menunjukkan performa yang baik pada kondisi terkontrol, namun kinerjanya seringkali tidak stabil ketika dihadapkan pada distribusi data yang lebih kompleks [6]. Teknik seperti *batch normalization* dan *dropout* telah digunakan untuk menstabilkan proses pelatihan serta mengurangi *overfitting* dan terbukti mampu meningkatkan konvergensi serta ketahanan model [7], [8].

Meskipun demikian, sebagian besar penelitian masih mengevaluasi teknik-teknik tersebut secara terpisah, sehingga peningkatan performa yang dihasilkan belum optimal ketika diterapkan pada skenario klasifikasi yang lebih menantang. Demikian pula, *batch normalization* diketahui mampu mempercepat konvergensi, namun penggunaannya secara tunggal seringkali belum cukup untuk menghasilkan performa klasifikasi terbaik tanpa dikombinasikan dengan strategi regularisasi lainnya [4].

Pendekatan lain yang banyak diadopsi adalah dengan melakukan peningkatan kedalaman arsitektur CNN. Meskipun strategi ini dapat memperkaya representasi fitur dan meningkatkan akurasi, pendekatan ini juga menyebabkan meningkatnya kompleksitas model serta kebutuhan sumber daya komputasi yang tinggi [9]. Beberapa arsitektur yang lebih kompleks seperti DIGITNET juga menunjukkan peningkatan performa, namun cenderung berfokus pada penskalaan arsitektur tanpa secara sistematis mengintegrasikan berbagai teknik optimasi agar model menjadi lebih ringan [10]. Selain itu, pengembangan model tidak hanya terbatas pada peningkatan kompleksitas arsitektur, tetapi juga mencakup eksplorasi pendekatan lain.

Di sisi lain, pendekatan hibrida seperti kombinasi CNN dengan arsitektur rekuren atau pemanfaatan *Generative Adversarial Networks* (GAN) untuk augmentasi data telah menunjukkan peningkatan performa yang signifikan [11]. Namun, metode-metode ini secara inheren meningkatkan kompleksitas sistem dan beban komputasi, sehingga kurang efisien untuk diimplemetasikan. Selain itu, teknik regularisasi seperti *dropout* dan *early stopping* terbukti efektif dalam mengurangi kesalahan klasifikasi, tetapi umumnya masih dikaji secara parsial dan belum diintegrasikan secara komprehensif dalam satu kerangka yang terstruktur [12], [13].

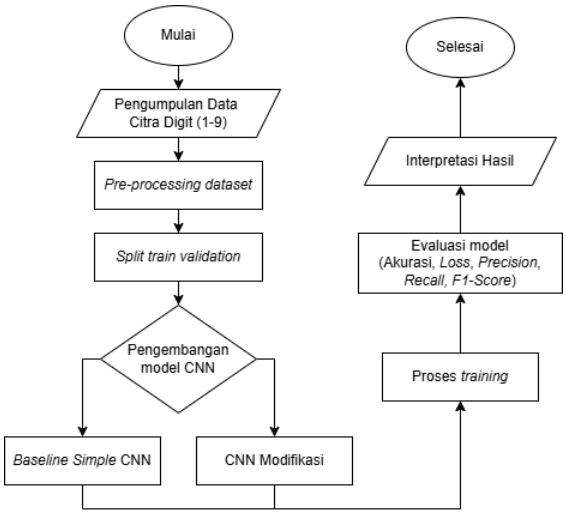
Untuk mengoptimalkan performa tanpa membebani komputasi, teknik modifikasi arsitektur seperti *data augmentation*, *batch normalization*, dan *dropout* menjadi solusi yang sangat relevan [14]. *Data augmentation* memungkinkan perluasan keragaman citra *training* secara artifisial, sementara *batch normalization* dan *dropout* secara efektif menstabilkan proses pelatihan dan mencegah model menghafal data (*overfitting*) [4]. Integrasi ketiga teknik ini berpotensi menghasilkan keseimbangan antara peningkatan performa dan efisiensi komputasi.

Oleh karena itu, penelitian ini difokuskan untuk mengukur efektivitas integrasi ketiga metode tersebut pada klasifikasi citra digit (0-9) menggunakan dataset berskala menengah. Arsitektur *baseline Simple CNN* dikembangkan melalui penambahan kedalaman jaringan yang dipadukan dengan teknik-teknik optimasi tersebut. Evaluasi komparatif yang mencakup matriks evaluasi dan *confusion matrix* diharapkan dapat membuktikan bahwa modifikasi arsitektur dasar mampu menghasilkan

peningkatan performa klasifikasi yang tinggi tanpa harus bergantung pada metode *transfer learning* yang lebih kompleks dan memerlukan sumber daya besar [8].

2. METODE PENELITIAN

Penelitian ini dilakukan dengan mengikuti tahapan yang sistematis untuk memastikan pencapaian tujuan optimasi model. Secara garis besar, alur kerja penelitian ini terbagi menjadi empat fase utama, yaitu pengumpulan dan *pre-processing* data, pengembangan model komparasi, *training*, serta evaluasi performa menyeluruh. Tahapan-tahapan tersebut digambarkan secara detail pada diagram alir di bawah ini pada Gambar 1.



Gambar 1. Diagram Alir Penelitian

Penelitian ini dilaksanakan melalui beberapa tahapan terstruktur yang selaras dengan diagram alir penelitian, diawali dengan tahap pengumpulan data citra digit bernilai nol hingga sembilan. Data yang digunakan mencakup variasi bentuk penulisan yang beragam guna merepresentasikan kondisi nyata dalam klasifikasi citra. Contoh dataset citra digit yang digunakan yakni pada Gambar 2.

Gambar 2. Contoh Dataset Citra Digit yang Digunakan

Tahap selanjutnya adalah *pre-processing* dataset, di mana seluruh citra diubah formatnya menjadi *grayscale* dan disesuaikan ukurannya menjadi resolusi 28x28 piksel. Pada tahap ini pula, intensitas piksel citra dinormalisasi ke dalam rentang nilai nol hingga satu guna memastikan stabilitas dan mempercepat konvergensi model saat *training* [5]. Setelah *pre-processing* selesai, dataset dibagi menjadi dua bagian, yaitu *data training* dan *data validation* dengan rasio pembagian sebesar 80:20. Proporsi ini dipilih sebagai standar yang optimal alokasi 80% data dirancang agar model memiliki sampel masukan yang memadai untuk mempelajari fitur visual secara mendalam, sementara 20% sisanya dinilai sangat representatif untuk mengevaluasi kemampuan generalisasi model serta memantau indikasi *overfitting* tanpa harus banyak mengurangi kapasitas *training* [15].

Tahap berikutnya adalah pengembangan model *Convolutional Neural Network* (CNN) yang dilakukan melalui dua pendekatan paralel untuk tujuan komparasi. Pendekatan pertama adalah perancangan model *baseline Simple* CNN yang hanya terdiri dari satu lapisan konvolusi, lapisan *max-pooling*, lapisan *flatten*, dan satu lapisan *fully connected* sebagai pemetaan akhir klasifikasi. Arsitektur model *Simple* CNN yang digunakan pada penelitian ini ditampilkan pada Gambar 3 berikut.

p-ISSN: 2527 - 9661
e-ISSN: 2549 - 2837

Model: "sequential_2"

Layer (type)	Output Shape	Param #
conv2d_4 (Conv2D)	(None, 26, 26, 32)	320
max_pooling2d_4 (MaxPooling2D)	(None, 13, 13, 32)	0
conv2d_5 (Conv2D)	(None, 11, 11, 64)	18,496
max_pooling2d_5 (MaxPooling2D)	(None, 5, 5, 64)	0
flatten_2 (Flatten)	(None, 1600)	0
dense_4 (Dense)	(None, 128)	204,928
dense_5 (Dense)	(None, 1)	129

Total params: 223,873 (874.50 KB)
Trainable params: 223,873 (874.50 KB)
Non-trainable params: 0 (0.00 B)

Gambar 3. Arsitektur Model *Simple* CNN Yang Digunakan

Arsitektur *Simple* CNN dirancang dengan dua lapisan konvolusi yang diikuti oleh lapisan *max-pooling* untuk melakukan reduksi dimensi spasial. Peningkatan jumlah filter dari bertujuan untuk memperkaya representasi fitur yang diekstraksi dari citra input. Selanjutnya, hasil ekstraksi fitur diubah menjadi vektor satu dimensi melalui proses *flatten* untuk mempersiapkan data sebelum masuk ke lapisan *fully connected*. Lapisan *dense* digunakan untuk memetakan fitur ke dalam ruang representasi yang lebih baik. Selain itu, konfigurasi jumlah parameter yang lebih besar pada lapisan *fully connected* dirancang untuk mendukung proses pemetaan fitur secara optimal pada tahap klasifikasi [9].

Pendekatan kedua adalah perancangan CNN Modifikasi guna meningkatkan performa model dasar tersebut. Pada model modifikasi ini, diterapkan teknik *data augmentation* berupa rotasi, translasi, dan *zoom* acak untuk memperluas variasi visual. Arsitektur jaringan juga diperdalam menjadi tiga blok konvolusi dengan jumlah filter berturut-turut 32, 64, dan 128. Selain itu, *batch normalization* ditambahkan untuk menstabilkan distribusi aktivasi internal, serta *dropout* pada lapisan *fully connected* difungsikan sebagai regularisasi untuk meminimalisasi risiko *overfitting* [16].

Arsitektur model CNN Modifikasi yang digunakan pada penelitian ini ditunjukkan pada Gambar 4.

Model: "cnn_modifikasi_digits"

Layer (type)	Output Shape	Param #
input_layer_1 (InputLayer)	(None, 28, 28, 1)	0
conv2d (Conv2D)	(None, 28, 28, 32)	320
batch_normalization (BatchNormalization)	(None, 28, 28, 32)	128
max_pooling2d (MaxPooling2D)	(None, 14, 14, 32)	0
conv2d_1 (Conv2D)	(None, 14, 14, 64)	18,496
batch_normalization_1 (BatchNormalization)	(None, 14, 14, 64)	256
max_pooling2d_1 (MaxPooling2D)	(None, 7, 7, 64)	0
conv2d_2 (Conv2D)	(None, 7, 7, 128)	73,856
batch_normalization_2 (BatchNormalization)	(None, 7, 7, 128)	512
max_pooling2d_2 (MaxPooling2D)	(None, 3, 3, 128)	0
flatten (Flatten)	(None, 1152)	0
dropout (Dropout)	(None, 1152)	0
dense (Dense)	(None, 128)	147,584
dropout_1 (Dropout)	(None, 128)	0
dense_1 (Dense)	(None, 10)	1,290

Total params: 242,442 (947.04 KB)
Trainable params: 241,994 (945.29 KB)
Non-trainable params: 448 (1.75 KB)

Gambar 4. Arsitektur model modifikasi CNN yang digunakan

Setelah arsitektur kedua model terbangun, tahap selanjutnya adalah proses *training* yang dijalankan menggunakan *optimizer* Adam dan fungsi *loss sparse categorical crossentropy*. *Training* dilakukan selama 15 *epoch* dengan menerapkan mekanisme *early stopping* berdasarkan pantauan nilai *validation loss*, sehingga model akan berhenti berlatih secara otomatis apabila tidak terjadi lagi peningkatan performa [7].

Tahap terakhir adalah evaluasi model dan interpretasi hasil. Pada tahap evaluasi, keandalan klasifikasi model diukur secara komprehensif tidak hanya menggunakan metrik akurasi, *loss*, dan *confusion matrix* , tetapi juga dianalisis lebih spesifik menggunakan perhitungan *precision*, *recall*, dan *F1-score*. Seluruh hasil pengukuran tersebut kemudian diinterpretasikan guna memvalidasi signifikansi peningkatan performa yang dihasilkan dari modifikasi arsitektur CNN yang diusulkan.

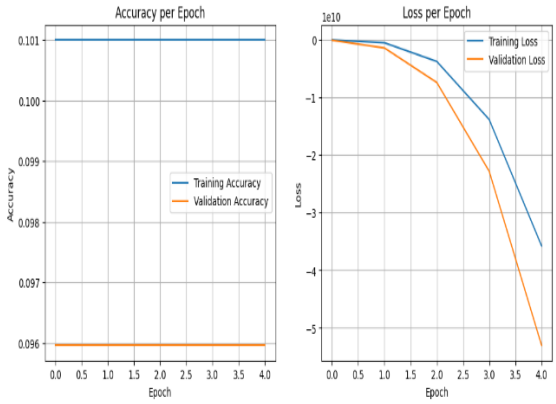
p-ISSN: 2527 - 9661
e-ISSN: 2549 - 2837

3. HASIL DAN PEMBAHASAN

Berdasarkan tahapan penelitian yang telah dirancang, analisis pertama difokuskan pada pengumpulan dan *pre-processing* data. Dataset yang digunakan merupakan dataset citra digit tulisan tangan (0-9) berbasis MNIST yang diperoleh melalui repositori publik Kaggle dan pustaka terintegrasi Scikit-learn, dengan total keseluruhan mencapai 10.160 citra. Dataset ini kemudian dialokasikan secara proporsional menjadi 80% data training (8.128 citra) dan 20% data validation (2.032 citra).

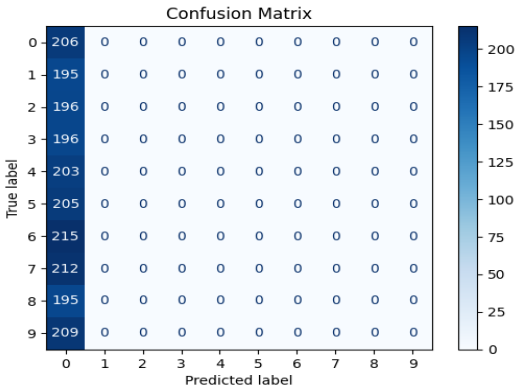
Seluruh citra berhasil dikonversi menjadi format *grayscale* (28x28 piksel) dan dinormalisasi intensitasnya (0-1) untuk menyeragamkan *input*. Pada model CNN Modifikasi, penerapan teknik data *augmentation* (rotasi, translasi, dan *zoom* acak) secara efektif mensimulasikan variasi gaya tulisan tangan tanpa menambah jumlah data mentah secara fisik.

Setelah *pre-processing*, tahap selanjutnya adalah pengujian arsitektur melalui proses *training* selama 15 *epoch*. Hasil *training* menunjukkan perbedaan performa yang sangat kontras. Pada model *Simple* CNN, arsitektur yang terlalu dangkal membuat proses ekstraksi fitur tidak berjalan optimal. Hal ini terlihat jelas pada grafik performa di bawah ini (Gambar 5).



Gambar 5. Grafik Akurasi dan *Loss* Model *Simple* CNN Selama Proses *Training*

Sebagaimana ditunjukkan pada Gambar 5, kurva akurasi cenderung stagnan berupa garis lurus di angka yang sangat rendah (mendekati 0.1 pada sumbu y), menandakan model gagal mempelajari pola visual digit. Hal ini sejalan dengan kurva *loss* yang bertahan di tingkat yang sangat tinggi, menunjukkan tingginya tingkat kesalahan selama *training*. Kegagalan ekstraksi fitur ini dibuktikan lebih jelas melalui visualisasi *confusion matrix* pada Gambar 6.

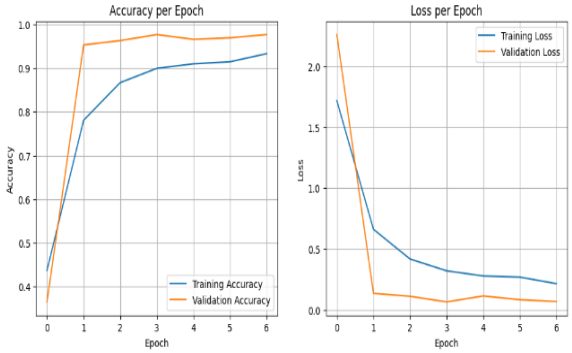


Gambar 6. Hasil *Confusion Matrix* Klasifikasi Digit Model *Simple* CNN

Berdasarkan *confusion matrix* pada Gambar 6, terlihat jelas bahwa model *Simple CNN* mengalami kegagalan klasifikasi yang fatal. Model memprediksi seluruh citra masukan pada *validation set* sebagai kelas '0', terlepas dari apapun label aslinya (seluruh prediksi menumpuk pada kolom pertama). Kegagalan model dalam membedakan antar kelas inilah yang secara visual mengonfirmasi mengapa kurva performa sebelumnya sangat rendah [5] . Model sama sekali tidak mengekstraksi fitur pembeda, melainkan hanya menebak satu angka secara buta, yang mana pada dataset dengan 10 kelas (0-9), probabilitas tebakan acak semacam ini secara logis hanya akan benar sekitar sepersepuluh bagian saja.

Sebaliknya, untuk mengatasi kelemahan fatal tersebut, arsitektur *CNN Modifikasi* menunjukkan konvergensi proses belajar yang sangat baik dan stabil, sebagaimana diperlihatkan pada Gambar 7.

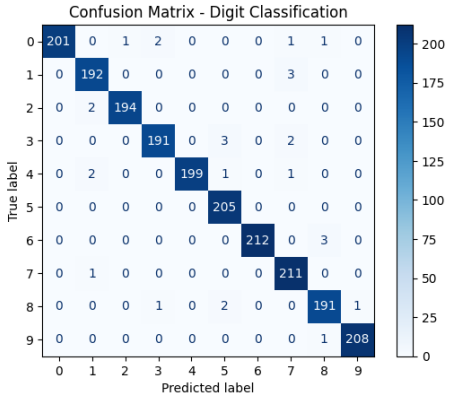
p-ISSN: 2527 - 9661
e-ISSN: 2549 - 2837



Gambar 7. Grafik Hasil Akurasi Dan *Loss* Model CNN Modifikasi Selama Proses *Training*

Merujuk pada Gambar 7, kurva akurasi terlihat menanjak tajam pada *epoch-epoch* awal dan stabil di angka yang sangat tinggi (mendekati 1.0). Sementara itu, kurva *loss* menurun drastis secara konsisten. Peningkatan kedalaman jaringan yang dipadukan dengan *batch normalization* terbukti mempercepat konvergensi pembelajaran, sedangkan penggunaan *dropout* (0,5) sukses mencegah kurva *validation loss* melonjak kembali (*overfitting*).

Untuk melihat lebih detail sebaran distribusi prediksi klasifikasi pada model *CNN* Modifikasi, evaluasi divisualisasikan menggunakan *confusion matrix* seperti pada Gambar 8.



Gambar 8. Hasil *Confusion Matrix* model Modifikasi CNN

Berdasarkan *confusion matrix* pada Gambar 8, terlihat pemusatan warna biru gelap pada garis diagonal utama. Ini merepresentasikan bahwa model mampu membedakan karakteristik setiap digit dengan tepat. Kesalahan prediksi (*misclassification)* yang terjadi (angka di luar garis diagonal) sangatlah minim dan tidak terpusat pada bias kelas tertentu.

Keberhasilan klasifikasi secara visual pada grafik dan matriks tersebut kemudian dibuktikan secara kuantitatif melalui perhitungan persentase akhir serta penambahan metrik *precision, recall,* dan *F1-score,* yang dirangkum komprehensif pada Tabel 1.

Tabel 1. Hasil Perbandingan Evaluasi Kedua Model

Matriks Evaluasi	*Simple* CNN	CNN Modifikasi
Validation Accuracy	10,00%	97,69 %
Loss	28,19%	4,80%
Precision	1,00%	97,70%
Recall	10,00%	97,69%
F1-Score	2,00%	97,69%

Berdasarkan rincian hasil perbandingan evaluasi kedua model pada Tabel 1, perbedaan kapasitas pembelajaran antara kedua arsitektur terekam secara jelas dan terukur. Model *Simple* CNN terhenti pada *validation accuracy* 10,00% dengan nilai *loss* yang sangat tinggi mencapai 28,19%. Kegagalan ini semakin terbukti melalui metrik pendukungnya, di mana nilai rata-rata *precision* hanya menyentuh 1,00% dan *F1-score* sebesar 2,00%. Ketimpangan angka metrik ini secara matematis memvalidasi analisis *confusion matrix* sebelumnya. Model sama sekali tidak mampu mengenali fitur pembeda antar digit dan hanya menebak kelas '0' secara konstan, sehingga menghasilkan banyak *false positive* yang menghancurkan nilai *precision*.

Sebaliknya, model CNN Modifikasi mencetak lonjakan performa yang amat signifikan. Tidak hanya *validation*

accuracy yang berhasil menyentuh 97,69% dan *loss* yang berhasil ditekan hingga 4,80%, tetapi metrik *precision, recall*, dan *F1-score* juga menunjukkan angka yang sangat konsisten dan seimbang (berada di kisaran 97,69% hingga 97,70%). Keseimbangan nilai rata-rata (*macro average*) pada ketiga metrik tersebut menjadi indikator yang membuktikan bahwa model modifikasi ini tidak memiliki bias terhadap kelas digit tertentu. Model benar-benar berhasil mengekstraksi representasi visual yang tepat dan memiliki kemampuan generalisasi yang sangat baik dalam mengklasifikasikan seluruh variasi angka.

Jika dibandingkan dengan *state-of-the-art* pada penelitian-penelitian terdahulu, capaian *validation accuracy* sebesar 97,69% pada model *CNN Modifikasi* ini memperlihatkan kontribusi yang kuat. Hasil ini sejalan dengan tren optimasi yang dilaporkan oleh [12] dan [13], di mana penerapan regularisasi *dropout* mampu menstabilkan performa klasifikasi digit di atas akurasi 95%.

Namun, berbeda dengan pendekatan penelitian Muthyas yang bergantung pada pemrosesan hibrida (CNN-BiLSTM) atau teknik augmentasi berbasis *Generative Adversarial Networks* (GAN) yang membutuhkan komputasi besar, penelitian ini menawarkan rute komputasi yang lebih ringan [11]. Capaian ini sekaligus mempertegas temuan terbaru bahwa optimasi arsitektur dasar secara mandiri melalui kombinasi *batch normalization* dan augmentasi data sudah sangat memadai untuk menekan *overfitting* dan menghasilkan performa tinggi pada dataset berskala menengah, tanpa harus bergantung pada metode *transfer learning*.

4. SIMPULAN

Berdasarkan hasil pengujian dan analisis yang telah dilakukan, dapat disimpulkan bahwa modifikasi arsitektur secara komprehensif terbukti efektif dalam mengatasi keterbatasan ekstraksi fitur pada model CNN dasar. Model *Simple CNN* sebelumnya mengalami kegagalan klasifikasi yang signifikan, ditandai dengan capaian *validation accuracy* yang terhenti pada angka 10,00% dan *validation loss* yang sangat tinggi (28,19%). Namun, melalui integrasi pendalaman blok konvolusi, *batch normalization*, *dropout*, dan *data augmentation* spasial, performa model *CNN Modifikasi* melonjak tajam dengan pencapaian *validation accuracy* sebesar 97,69% dan *loss* yang berhasil ditekan hingga 4,80%.

Keandalan dan konsistensi model ini divalidasi lebih lanjut melalui capaian *precision*, *recall*, dan *F1-score* yang stabil di angka 97,69% hingga 97,70%, serta hasil *confusion matrix* yang menunjukkan minimnya kesalahan prediksi antar kelas. Capaian ini menjawab rumusan masalah penelitian, yakni membuktikan bahwa kombinasi teknik optimasi dasar mampu meningkatkan stabilitas pelatihan, mencegah *overfitting*, dan memperkuat generalisasi model secara efisien pada dataset berskala menengah. Performa klasifikasi yang tinggi dapat dicapai tanpa keharusan mengandalkan pendekatan *transfer learning* atau model hibrida kompleks yang membutuhkan beban komputasi besar.

DAFTAR PUSTAKA

[1] H. Hsia, T. Chang, C. Y. Chiang, and H. Chan, “Mask R-CNN with New Data Augmentation Features for Smart Detection of Retail Products,” *Applied Sciences (Switzerland)*, vol. 12, no. 6, 2022, doi: 10.3390/app12062902.

[2] D. Bhatt *et al.*, “CNN Variants for Computer Vision: History, Architecture, Application, Challenges and Future Scope,” 2021. doi: 10.3390/electronics10202470.

[3] S. Basha, S. Dubey, V. Pulabaigari, and S. Mukherjee, "Impact of Fully Connected Layers on Performance of Convolutional Neural Networks for Image Classification," *Neurocomputing*, vol. 378, 2020, doi: 10.1016/j.neucom.2019.10.008.

[4] G. Zhang and W. Abdulla, "Optimizing Hyperspectral Imaging Classification Performance with CNN and Batch Normalization," *Applied Spectroscopy Practica*, vol. 1, no. 2, 2023, doi: 10.1177/27551857231204622.

[5] A. Zafar *et al.*, "Convolutional Neural Networks: A Comprehensive Evaluation and Benchmarking of Pooling Layer Variants," *Symmetry (Basel).*, vol. 16, no. 11, Nov. 2024, doi: 10.3390/sym16111516.

[6] A. Biswas and Md. S. Islam, "An Efficient CNN Model for Automated Digital Handwritten Digit Classification," *Journal of Information Systems Engineering and Business Intelligence*, vol. 7, no. 1, p. 42, Apr. 2021, doi: 10.20473/jisebi.7.1.42-55.

[7] Norhikmah, A. Lutfhi, and Rumini, "The Effect of Layer Batch Normalization and Dropout of CNN Model Performance on Facial Expression Classification," *International Journal on Informatics Visualization*, vol. 6, no. 2–2, 2022, doi: 10.30630/joiv.6.2-2.921.

[8] A. Zafar *et al.*, "A Comparison of Pooling Methods for Convolutional Neural Networks," *Applied Sciences (Switzerland)*, vol. 12, no. 17, Sep. 2022, doi: 10.3390/app12178643.

[9] K. Muthureka, U. Srinivasulu Reddy, and B. Janet, "An Improved Customized CNN Model for Adaptive Recognition of Cerebral Palsy People's Handwritten Digits in Assessment," *Int. J. Multimed. Inf. Retr.*, vol. 12, no. 2, 2023, doi: 10.1007/s13735-023-00291-8.

[10] H. Kusetogullari, A. Yavariabdi, J. Hall, and N. Lavesson, "DIGITNET: A Deep Handwritten Digit Detection and Recognition Method Using a New Historical Handwritten Digit Dataset," *Big Data Research*, vol. 23, 2021, doi: 10.1016/j.bdr.2020.100182.

[11] Muhtyas Yugi, F. S. Utomo, and A. S. Barkah, "Improving Handwritten Digit Recognition Using CycleGAN-Augmented Data with CNN–BILSTM Hybrid Model," *JITK (Jurnal Ilmu Pengetahuan dan Teknologi Komputer)*, vol. 11, no. 2, pp. 334–341, Nov. 2025, doi: 10.33480/jitk.v11i2.6982.

[12] S. Patre and M. Swati Tiwari, "Practical CNN Enhancement For Handwritten Digit Recognition: Integrating Dropout, Data Augmentation, And Early Stopping," *International Journal of Novel Trends and Innovation*, vol. 3, no. 7, p. 36, Jul. 2025, [Online]. Available: www.ijnti.org

[13] M. Maha Seetha and J. Ruby MCA, "Hand Writing Recognition Using Deep Learning Algorithm," *International Journal of Creative Research Thoughts (IJCRT)*, vol. 10, Dec. 2022, [Online]. Available: www.ijcrt.org

[14] R. Rosalina and A. Y. Husodo, "Development of a Convolutional Neural Network Method for Classifying Ripeness Levels of Servo Variety Tomatoes," *Jurnal Teknik Informatika (Jutif)*, vol. 6, no. 2, pp. 501–520, Apr. 2025, doi: 10.52436/1.jutif.2025.6.2.4168.

[15] S. Lasniari, S. Sanjaya, F. Yanto, and M. Affandes, "Pengaruh Hyperparameter Convolutional Neural Network Arsitektur ResNet-50 Pada Klasifikasi Citra,"

Universitas Islam Negeri Sultan Syarif Kasim Riau Jl. H.R Soebrantas No. 155 KM, vol. 5, no. 3, p. 28293, Jun. 2022.

[16] S. E. Prasetyo, H. Haeruddin, and E. Elvis, "Evaluasi Efektivitas Teknik Regularisasi Dalam Mengurangi Overfitting Pada Model CNN," *EXPERT: Jurnal Manajemen Sistem Informasi dan Teknologi*, vol. 15, no. 2, p. 123, Dec. 2025, doi: 10.36448/expert.v15i2.4676.