

Implementasi Algoritma *Hybrid* CNN-ViT (LeViT-128) dalam Klasifikasi Motif Batik Indonesia

Kevin Novebrianto ¹, Hadi Zakaria ^{2*}

^{1,2*} Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Pamulang, Kota Tangerang Selatan, Provinsi Banten, Indonesia.

Email: kevin novebrianto212@gmail.com ¹, dosen00274@unpam.ac.id ²

Histori Artikel:

Dikirim 2 September 2026; Diterima dalam bentuk revisi 11 September 2026; Diterima 20 September 2026; Diterbitkan 30 September 2026. Semua hak dilindungi oleh Lembaga Penelitian dan Pengabdian Masyarakat (LPPM) STM IK Indonesia Banda Aceh.

Abstrak

Batik merupakan warisan budaya Indonesia yang memiliki keragaman motif dengan tingkat kemiripan visual yang tinggi antarkelas (*high inter-class similarity*). Proses identifikasi secara visual dan manual sangat rentan terhadap subjektivitas dan kurang efisien. Untuk mengatasi permasalahan klasifikasi tersebut, penelitian ini mengimplementasikan pendekatan *deep learning* menggunakan arsitektur hybrid *Convolutional Neural Network* (CNN) dan *Vision Transformer* (ViT), secara spesifik model LeViT. Penggabungan ini bertujuan menyinergikan kemampuan CNN dalam mengekstraksi fitur lokal—seperti guratan dan titik—dengan mekanisme *self-attention* pada ViT untuk menangkap konteks spasial global secara efisien. Metodologi penelitian mengacu pada kerangka kerja *Cross-Industry Standard Process for Data Mining* (CRISP-DM). Model dilatih menggunakan dataset 5.292 citra yang mencakup 39 kelas motif batik. Untuk mengatasi ketimpangan distribusi data (*data imbalance*), diterapkan teknik penyesuaian bobot kelas dan augmentasi citra. Berdasarkan evaluasi akhir dengan pendekatan *One-vs-Rest* pada data uji, model mencapai akurasi keseluruhan 90,05% dan *Weighted F1-Score* 89,84%, membuktikan efektivitas dan keandalan algoritma hybrid CNN-ViT dalam mengklasifikasikan motif batik dengan kompleksitas visual tinggi.

Kata Kunci: Klasifikasi Batik; *Deep Learning*; Hybrid CNN-ViT; LeViT; CRISP-DM.

Abstract

Batik is an Indonesian cultural heritage featuring a wide variety of motifs with high inter-class visual similarity. The visual and manual identification process is highly subjective and inefficient. To address this classification problem, this study implements a deep learning approach using a hybrid *Convolutional Neural Network* (CNN) and *Vision Transformer* (ViT) architecture, specifically the LeViT model. This integration aims to synergize CNN's ability to extract local features—such as lines and dots—with ViT's *self-attention* mechanism to capture global spatial context efficiently. The research methodology follows the *Cross-Industry Standard Process for Data Mining* (CRISP-DM) framework. The model was trained on a dataset of 5,292 images covering 39 batik motif classes. To address data imbalance, class weight adjustments and image augmentation techniques were applied. Based on the final evaluation using the *One-vs-Rest* approach on the test set, the model achieved an overall accuracy of 90.05% and a *Weighted F1-Score* of 89.84%, demonstrating the effectiveness and reliability of the hybrid CNN-ViT algorithm in classifying batik motifs with high visual complexity.

Keyword: Batik Classification; Deep Learning; Hybrid CNN-ViT; LeViT; CRISP-DM.

1. Pendahuluan

Batik merupakan warisan budaya Indonesia dengan keragaman motif yang kaya dan bernilai filosofis tinggi, namun identifikasi motifnya masih banyak dilakukan secara manual. Pengamatan visual manual bergantung pada pengalaman pengamat sehingga bersifat subjektif, memakan waktu, dan rentan terhadap kesalahan. Tantangan ini semakin besar karena banyak motif batik memiliki tingkat kemiripan visual yang tinggi antarkelas (*high inter-class similarity*), sehingga klasifikasi batik masuk ke ranah *Fine-Grained Visual Classification* (FGVC) (Priyambodo *et al.*, 2026). Oleh karena itu, otomatisasi identifikasi melalui pengolahan citra digital dan kecerdasan buatan menjadi keharusan untuk meminimalkan subjektivitas, membedakan variasi visual yang kompleks, dan meningkatkan efisiensi pengenalan motif.

Meskipun CNN telah dimanfaatkan pada berbagai aplikasi klasifikasi citra di Indonesia (Anggraini & Zakaria, 2023), pendekatan tersebut masih terbatas pada domain dengan variasi visual yang relatif terkendali dan belum menyentuh tantangan *fine-grained classification* pada motif batik nusantara. Berbagai studi terdahulu telah menyoroti urgensi dan efektivitas algoritma *deep learning* dalam klasifikasi batik. Aditama *et al.* (2025) menggunakan CNN berbasis arsitektur VGG16 untuk mendeteksi empat motif Batik Solo dan mencapai akurasi 85,71%; namun model kesulitan mengenali motif dengan tekstur halus dan repetitif—seperti motif Truntum—karena pada konteks tersebut CNN lebih mudah menangkap pola berskala besar dibandingkan detail mikro. Hal ini diperkuat oleh temuan Najar *et al.* (2024) yang mengimplementasikan CNN (DenseNet121) dengan akurasi 88%, di mana intervensi pra-pemrosesan berbasis deteksi tepi justru menghilangkan tekstur esensial yang diandalkan oleh CNN. Untuk mengatasi keterbatasan struktural tersebut, eksplorasi terhadap arsitektur alternatif mulai dieksplorasi. Chandra *et al.* (2025) mengevaluasi perbandingan kinerja CNN dan *Vision Transformer* (ViT) pada klasifikasi motif batik, dan menemukan bahwa CNN memiliki kelemahan mendasar dalam menangkap hubungan spasial global (*global spatial relationships*), sementara ViT terbukti unggul berkat mekanisme *self-attention* yang efektif merepresentasikan struktur spasial yang rumit. Temuan ini diperkuat oleh Nofiantoro dan Saputri (2026) yang menguji CNN dan ViT pada 60 kelas Batik Nitik; hasilnya menunjukkan bahwa CNN mencapai batas efisiensi arsitekturalnya (*ceiling performance*) di angka 63,44%, sedangkan ViT mampu mencapai akurasi 97,40% dengan syarat ketersediaan data yang masif (*data-hungry*).

Penelitian-penelitian tersebut telah membuktikan keandalan CNN dalam ekstraksi fitur lokal serta keunggulan ViT dalam memetakan konteks spasial global. Namun, analisis terhadap literatur tersebut menyingkap adanya kesenjangan: kajian yang secara spesifik mengintegrasikan keunggulan ekstraksi fitur lokal CNN dan kekuatan mekanisme *self-attention* dari keluarga *transformer* ke dalam model *hybrid*—seperti LeViT—untuk klasifikasi motif batik nusantara masih terbatas. Penggabungan ini perlu dieksplorasi guna menciptakan model yang tidak hanya *robust* terhadap variasi tekstur halus, tetapi juga efisien dalam kondisi keterbatasan data. Selain itu, mayoritas penelitian sebelumnya belum mengintegrasikan kerangka kerja metodologi *data mining* yang terstruktur untuk menangani ketimpangan distribusi kelas (*data imbalance*).

Berpijak pada permasalahan dan kesenjangan tersebut, penelitian ini bertujuan mengimplementasikan algoritma *hybrid* CNN-ViT dengan arsitektur LeViT—yang menurut Graham *et al.* (2021) unggul dalam efisiensi komputasi dan representasi spasial berkat struktur piramida serta mekanisme *attention bias*—untuk mengklasifikasikan 39 kelas motif batik nusantara. Penelitian ini dirancang untuk mengatasi keterbatasan hubungan spasial global pada CNN sekaligus menekan kerentanan *data-hungry* pada ViT murni melalui teknik penyesuaian bobot kelas (*class weights*) dan augmentasi citra, dengan seluruh tahapan pengembangan model dijalankan menggunakan metodologi CRISP-DM (Martínez-Plumed *et al.*, 2021).

2. Metode Penelitian

2.1 Pengumpulan dan Persiapan Data

Dataset yang digunakan dalam penelitian ini merupakan hasil kurasi dengan proporsi pengumpulan awal sekitar 70% bersumber dari platform HuggingFace melalui dataset Batik-Indonesia yang disusun oleh Alfaridzi (2024), serta 30% dari repositori publik Kaggle yang menggabungkan dataset milik Hermansyah (2020), HendryHB (2024), dan Sa'diyah (2025). Setelah pengumpulan awal, dilakukan pencegahan duplikasi citra untuk memastikan keunikan data dan mencegah kebocoran data (*data leakage*) antar-*split*, yaitu latih, validasi, dan uji. Eliminasi duplikat dilakukan melalui dua tahap komputasi: deteksi duplikat identik menggunakan pemindaian *cryptographic* MD5 *Hash*, dilanjutkan dengan eliminasi duplikat visual—yakni citra yang tampak sama namun memiliki perbedaan minor pada tingkat kompresi, resolusi, atau warna—menggunakan metode *Perceptual Hashing* (pHash). Pada tahap pHash, apabila sistem mendeteksi duplikat visual, algoritma diprogram untuk mengurutkan *file* berdasarkan ukuran *byte*, mempertahankan satu citra dengan ukuran terbesar sebagai representasi kualitas terbaik, dan menghapus *file* sisanya (Chen *et al.*, 2021).

Hasil deduplikasi menyisakan 5.292 citra dalam 39 kelas motif batik nusantara. Pemilihan 39 kelas didasarkan pada ketersediaan data, representasi geografis 15 daerah dari Aceh hingga Papua, serta variasi visual yang kompleks untuk menguji ketangguhan model pada *fine-grained classification*. Variasinya mencakup motif geometris klasik seperti Kawung dan Parang (Sunarko *et al.*, 2026), motif pesisir seperti Tujuh Rupa (Kusumaningroem *et al.*, 2026), hingga ragam hias figuratif dan kontemporer seperti Tribusono (Rosydi & Ciptandi, 2023). Kualitas label divalidasi secara bertingkat melalui verifikasi silang dengan literatur motif batik klasik, khususnya *Batik Filosofi, Motif dan Kegunaan* (Kusrianto, 2024), *Ensiklopedia The Heritage of Batik* (Supriono, 2024), serta *Motif Batik Klasik Legendaris dan Turunannya* (Kusrianto, 2021). Verifikasi ini penting untuk kelas dengan kemiripan visual tinggi (*high inter-class similarity*), seperti Solo-Yogyakarta_Sidoluhur, Solo-Yogyakarta_Sidomukti, dan Solo-Yogyakarta_Sidomulyo yang struktur geometrisnya hampir identik; citra yang tidak dapat diverifikasi secara jelas dikeluarkan dari dataset. Citra yang dipertahankan beresolusi 224×224 hingga 1.200×1.200 piksel, berformat .jpg dan .png dengan ruang warna RGB. Validasi integritas berkas dilakukan menggunakan *Python Imaging Library* (PIL) untuk memastikan tidak ada citra yang rusak atau terpotong, sehingga seluruh data layak digunakan dalam pelatihan (Kranthi & Ramabhotla, 2026). Tabel 1 menyajikan distribusi lengkap seluruh kelas motif batik beserta jumlah sampelnya.

Tabel 1. Distribusi Kelas dan Jumlah Sampel Dataset Motif Batik

No.	Nama Kelas (Motif Batik)	Jumlah
1	Aceh_Pintu Aceh	47
2	Bali_Barong	55
3	Bali_Poleng	88
4	Betawi_Ondel Ondel	42
5	Cirebon_Liong	150
6	Cirebon_Megamendung	332
7	Cirebon_Singo Barong	150
8	Dayak_Benang Bintik	353
9	Garut_Bali_Merak Ngibing	57
10	Kulon Progo_Geblek Renteng	92
11	Lampung_Gajah	44
12	Lombok_Lumbung	41
13	Madura_Mataketeran	47
14	Magetan_Pring	46
15	Maluku_Pala	129
16	Papua_Asmat	45

17	Papua_Cendrawasih	189
18	Papua_Tifa	42
19	Pekalongan_Buketan	150
20	Pekalongan_Jlamprang	150
21	Pekalongan_Tujuh Rupa	149
22	Pontianak_Corak Insang	124
23	Rembang_Lasem	133
24	Solo-Yogyakarta_Kawung	312
25	Solo-Yogyakarta_Parang	424
26	Solo-Yogyakarta_Sekar Jagad	214
27	Solo-Yogyakarta_Sidoluhur	200
28	Solo-Yogyakarta_Sidomukti	196
29	Solo-Yogyakarta_Sidomulyo	150
30	Solo-Yogyakarta_Tambal	110
31	Solo-Yogyakarta_Wahyu Tumurun	150
32	Solo-Yogyakarta_Wirasat	150
33	Solo_Bokor_Kencono	150
34	Solo_Srikaton	150
35	Solo_Tribusono	150
36	Solo_Truntum	150
37	Sulawesi_Selatan_Lontara	43
38	Sumatera Barat_Rumah Minang	44
39	Sumatera_Utara_Boraspatri	44
Total Keseluruhan		5.292

Berdasarkan Tabel 1, teridentifikasi tiga kelas mayoritas—Solo-Yogyakarta_Parang (424 citra), Dayak_Benang Bintik (353 citra), dan Cirebon_Megamendung (332 citra)—serta tiga kelas minoritas—Lombok_Lambung (41 citra), Betawi_Ondel Ondel (42 citra), dan Papua_Tifa (42 citra)—dengan rasio ketimpangan 1:10. Kondisi ini berpotensi memicu bias prediksi terhadap kelas mayoritas. Untuk memitigasinya, diterapkan bobot kelas dinamis (*class weights*) menggunakan fungsi *compute_class_weight* dari pustaka *scikit-learn* dengan parameter ‘balanced’. Bobot dihitung menggunakan Persamaan (1), di mana w_j adalah bobot untuk kelas ke- j , N adalah total sampel data latih, K adalah jumlah kelas (39), dan n_j adalah jumlah sampel pada kelas ke- j . Bobot yang dihasilkan diintegrasikan ke dalam fungsi kerugian (*loss function*) pada saat pelatihan.

$$w_j = \frac{N}{K \times n_j} \quad (1)$$

Gambar 1 menyajikan contoh citra dari dataset motif batik yang digunakan. Penulis perlu menambahkan narasi penjelasan yang mendeskripsikan isi gambar secara singkat.



Gambar 1. Sampel Gambar pada Dataset Motif Batik

Seluruh citra melalui tahap prapemrosesan dan augmentasi. Data latih diberi augmentasi *Random Resized Crop* (skala 75%–100%), *Random Horizontal Flip* (probabilitas 0,5), dan *Color Jitter* (kecerahan, kontras, saturasi 10%; hue 5%). Data validasi dan uji hanya melalui *resize* ke 256 piksel dan *Center Crop* menjadi 224×224 piksel tanpa augmentasi. Normalisasi dilakukan secara statis menggunakan konstanta standar ImageNet, yaitu rata-rata [0,485; 0,456; 0,406] dan standar deviasi [0,229; 0,224; 0,225], bukan dihitung ulang dari sebaran data latih. Pendekatan ini merupakan praktik standar *transfer learning* untuk menyelaraskan distribusi ruang warna dataset batik dengan *pre-trained weights* LeViT yang diinisialisasi dari ImageNet, sehingga ekstraksi fitur awal tidak mengalami pergeseran distribusi (Sharifi & Alabdullah, 2025). Pembagian data menggunakan *stratified splitting* dengan rasio 70:15:15, menghasilkan 3.704 citra latih, 794 validasi, dan 794 uji. Pemuatan data menggunakan *DataLoader* PyTorch dengan *batch size* 32 dan 2 pekerja paralel.

2.2 Perancangan dan Pelatihan Model Hybrid CNN-ViT

Penelitian ini mengimplementasikan arsitektur *hybrid* CNN-ViT dengan model LeViT-128 (*LeViT: a Vision Transformer in ConvNet's Clothing*) yang diperkenalkan oleh Graham *et al.* (2021). Arsitektur ini menggabungkan prinsip desain CNN ke dalam kerangka *Vision Transformer*, sehingga unggul dalam menangkap fitur lokal (guratan, titik, tekstur) sekaligus memetakan hubungan spasial global melalui mekanisme *self-attention*. LeViT menggunakan struktur piramida bertingkat, *attention bias* yang bersifat *translation-invariant* sebagai pengganti *positional embedding*, serta *Global Average Pooling* (GAP) pada tahap akhir. Secara struktural, LeViT-128 terdiri dari *Stem* CNN berupa 4 lapisan konvolusi 3×3 dengan *stride* 2 yang mereduksi citra 224×224 piksel menjadi 14×14×256, dilanjutkan dengan 3 tahap *Transformer* multi-resolusi: *Stage* 1 berisi 4 blok, 4 kepala *attention*, 128 saluran, resolusi 14×14; *Stage* 2 berisi 4 blok, 6 kepala, 256 saluran, resolusi 7×7; dan *Stage* 3 berisi 4 blok, 8 kepala, 384 saluran, resolusi 7×7. Total parameter yang dapat dilatih sebanyak 8,47 juta dengan kompleksitas 406 juta FLOPs. Implementasi menggunakan pustaka *timm* dengan bobot pra-latih ImageNet (*transfer learning*) dan lapisan klasifikasi akhir disesuaikan untuk 39 kelas. Untuk mencegah *overfitting*, diterapkan *drop rate* dan *drop path rate* sebesar 0,1. Mekanisme *attention bias* pada LeViT menghitung nilai *attention* antara dua piksel pada posisi (x, y) dan (x', y') untuk satu *head* h sebagaimana ditunjukkan pada Persamaan (2), di mana Q dan K adalah vektor *query* dan *key*, sementara B adalah *attention bias* yang bergantung pada *offset* relatif antarpiksel. Pendekatan ini memungkinkan setiap *head* mempelajari pola ketergantungan spasial yang berbeda tanpa vektor posisi tambahan.

$$A_{(x,y),(x',y')}^h = Q_{(x,y)} \cdot K_{(x',y')} + B_{(x-x',|y-y'|)}^h \quad (2)$$

Proses pelatihan menggunakan *optimizer* AdamW dengan laju pembelajaran awal 5×10^{-4} dan *weight decay* 0,01, serta jadwal *Cosine Annealing LR* hingga 1×10^{-6} . Fungsi kerugian menggunakan *Cross Entropy Loss* dengan integrasi bobot kelas dan *label smoothing* 0,1. Pencarian *epoch* optimal dilakukan melalui *hyperparameter sweeping* pada rentang 20 hingga 40 *epoch* dengan interval 5 *epoch*. Model dengan konfigurasi *epoch* yang menghasilkan kombinasi *Validation Loss* terendah dan akurasi tertinggi ditetapkan sebagai model final untuk dievaluasi pada data uji.

2.3 Evaluasi dan Analisis Performa

Evaluasi performa model dilakukan melalui dua pendekatan: evaluasi kualitas klasifikasi menggunakan metrik standar *supervised learning*, dan analisis efisiensi komputasi. Untuk evaluasi klasifikasi, digunakan metrik *accuracy*, *precision*, *recall*, dan *F1-Score* yang dihitung dengan pendekatan *One-vs-Rest* (OvR) untuk klasifikasi multikelas 39 kelas, di mana setiap kelas diperlakukan sebagai positif dan kelas lainnya sebagai negatif. Karena distribusi data yang timpang, perhitungan metrik agregasi menggunakan teknik *weighted average* berdasarkan jumlah sampel aktual (*support*) setiap kelas. Rumus yang digunakan disajikan pada Persamaan (3) hingga (9).

$$Accuracy = \frac{\sum (Prediksi Benar per Kelas)}{Total Data Uji} \quad (3)$$

$$Precision_i = \frac{TP_i}{TP_i + FP_i} \quad (4)$$

$$Recall_i = \frac{TP_i}{TP_i + FN_i} \quad (5)$$

$$F1 - Score_i = 2 \times \frac{Precision_i \times Recall_i}{Precision_i + Recall_i} \quad (6)$$

$$Weighted Precision = \frac{\sum_{i=1}^{39} (Precision_i \times Support_i)}{\sum_{i=1}^{39} Support_i} \quad (7)$$

$$Weighted Recall = \frac{\sum_{i=1}^{39} (Recall_i \times Support_i)}{\sum_{i=1}^{39} Support_i} \quad (8)$$

$$Weighted F1 - Score = \frac{\sum_{i=1}^{39} (F1-Score_i \times Support_i)}{\sum_{i=1}^{39} Support_i} \quad (9)$$

Selain metrik klasifikasi, dilakukan analisis efisiensi komputasi melalui tiga metrik. *Parameter Size* (MB) digunakan untuk menilai kebutuhan penyimpanan, dihitung menggunakan Persamaan (10) dengan asumsi Float32 membutuhkan 4 bytes per parameter. *Latency* (milidetik per citra) digunakan untuk mengukur kecepatan respons, yang dipengaruhi oleh ukuran parameter, FLOPs, dan *bandwidth* memori. *Throughput* (inferensi per detik atau FPS) digunakan untuk mengevaluasi kapasitas pemrosesan, dihitung menggunakan Persamaan (11). Pengukuran dilakukan pada GPU NVIDIA Tesla T4 yang merepresentasikan lingkungan komputasi umum untuk *deep learning*. Untuk transparansi proses klasifikasi, dilakukan analisis aliran data dari citra mentah hingga keputusan akhir, meliputi prapemrosesan, ekstraksi fitur CNN, mekanisme *self-attention*, *Global Average Pooling*, perhitungan *logits*, hingga normalisasi *softmax* sebagaimana ditunjukkan pada Persamaan (12).

$$Parameter Size = Jumlah Parameter \times Ukuran per Parameter \quad (10)$$

$$Throughput = \frac{1}{Latency} \times 1000 \quad (11)$$

$$P(y = c | x) = \frac{e^{z^c}}{\sum_{j=1}^{39} e^{z_j}} \quad (12)$$

3. Hasil dan Pembahasan

3.1 Hasil

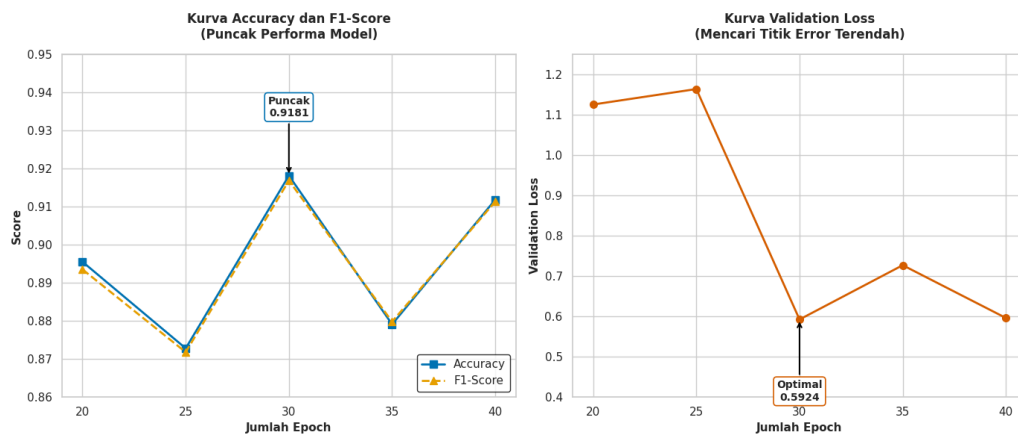
3.1.1 Hasil Pelatihan Model

Pelatihan LeViT-128 menggunakan konfigurasi *hyperparameter* sebagaimana diuraikan pada bagian metode. Dari pencarian *epoch* optimal pada rentang 20–40 *epoch*, konfigurasi 30 *epoch* menghasilkan performa terbaik pada data validasi. Tabel 2 menyajikan perbandingan seluruh konfigurasi yang diuji.

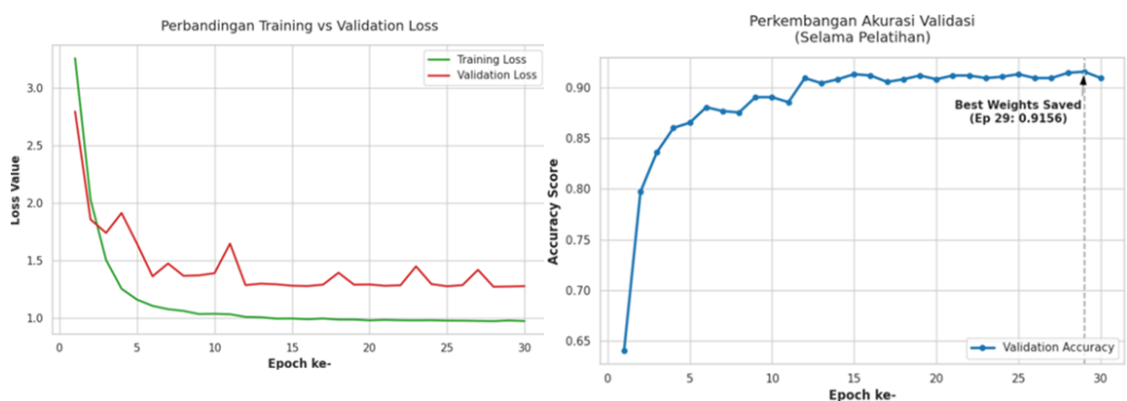
Tabel 2. Hasil Pencarian Durasi Siklus (*Epoch Sweeping*) pada Data Validasi

Konfigurasi <i>Epoch</i>	<i>Validation Loss</i>	Akurasi	Presisi	F1-Score
20 <i>Epochs</i>	11,249	0,8955	0,8999	0,8950
25 <i>Epochs</i>	11,632	0,8728	0,8781	0,8730
30 <i>Epochs</i>	0,5924	0,9181	0,9217	0,9168
35 <i>Epochs</i>	0,7268	0,8791	0,8875	0,8820
40 <i>Epochs</i>	0,5961	0,9118	0,9148	0,9100

Gambar 2 menyajikan perbandingan visual akurasi, F1-Score, dan *Validation Loss* pada seluruh konfigurasi yang diuji.

Gambar 2. Grafik Perbandingan Akurasi, F1-Score, dan *Validation Loss* pada 20–40 Epoch

Berdasarkan Tabel 2 dan Gambar 2, konfigurasi 30 *epoch* menjadi titik keseimbangan terbaik dengan *Validation Loss* terendah (0,5924) serta akurasi (0,9181) dan F1-Score (0,9168) tertinggi. Perpanjangan siklus ke 35 *epoch* memicu indikasi *overfitting*, ditandai dengan naiknya *loss* ke 0,7268; meski sempat membaik pada 40 *epoch* (akurasi 0,9118), performa tersebut tidak melampaui puncak 30 *epoch*. Oleh karena itu, 30 *epoch* ditetapkan sebagai konfigurasi final yang paling stabil. Gambar 3 menyajikan kurva *training loss* dan *validation accuracy* selama proses pelatihan 30 *epoch*.

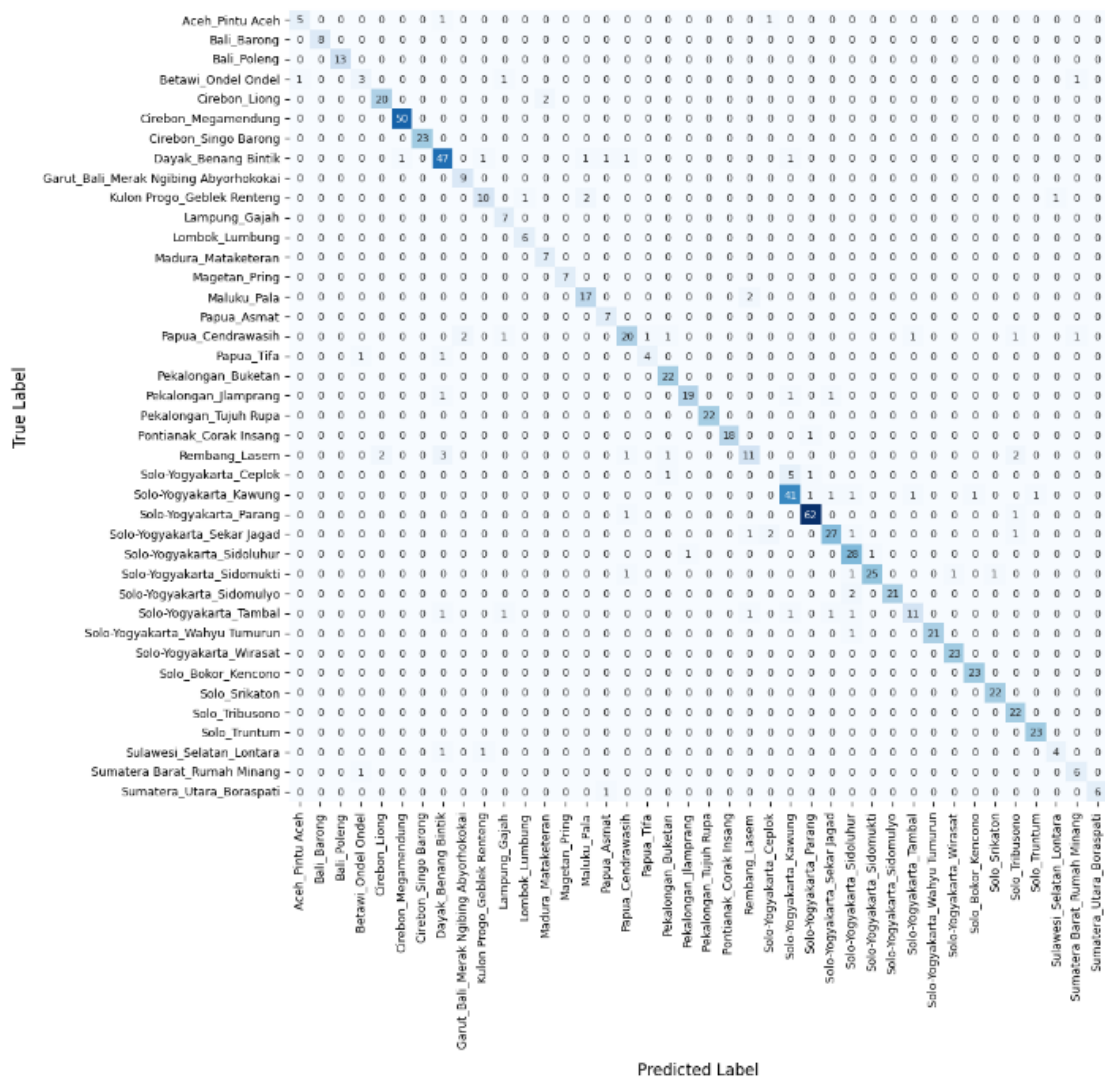
Gambar 3. Kurva *Training Loss* dan *Validation Accuracy* pada Pelatihan 30 Epoch

Pada fase awal (*epoch* 1–5), performa meningkat secara eksponensial: *Validation Accuracy* melesat dari 64,11% ke 86,52%, diiringi penurunan tajam *Validation Loss* ke 1,646. Fase konvergensi (*epoch* 6–20) ditandai dengan akurasi yang stabil di atas 88% meski terdapat fluktuasi *loss* minor. Memasuki fase stabilisasi akhir (*epoch* 21–30), model mencapai keseimbangan: *Training Loss* melandai hingga 0,974,

sementara *Validation Accuracy* mencapai puncak absolut pada epoch ke-29 sebesar 91,56% dengan *loss* 1,274—titik penyimpanan bobot terbaik (*best weights saved*). Jarak kurva latih dan validasi yang terjaga mengonfirmasi bahwa model mampu menggeneralisasi data dengan baik dan terhindar dari *overfitting* yang signifikan.

3.1.2 Evaluasi Akhir pada Data Uji

Model terbaik dari konfigurasi 30 *epoch* dievaluasi menggunakan 794 citra uji yang belum pernah dilihat model selama pelatihan. Gambar 4 menyajikan visualisasi *confusion matrix* prediksi model pada 39 kelas motif batik.



Gambar 4. *Confusion Matrix* pada Data Uji

Secara global, model mencatatkan *Test Loss* 0,6613. Mengingat distribusi dataset yang timpang (*imbalanced*), perhitungan metrik akhir menggunakan pendekatan *weighted average* berdasarkan *support* aktual tiap kelas, sebagaimana dirangkum pada Tabel 3.

Tabel 3. Ringkasan Metrik Evaluasi Akhir pada Data Uji

Metrik	Nilai	Keterangan
<i>Overall Accuracy</i>	90,05%	715 prediksi benar dari 794 total
<i>Weighted Precision</i>	90,68%	Rata-rata presisi berbobot
<i>Weighted Recall</i>	90,05%	Rata-rata <i>recall</i> berbobot
<i>Weighted F1-Score</i>	89,84%	Rata-rata harmonis berbobot
<i>Test Loss</i>	0,6613	Nilai kerugian pada data uji

Capaian *Weighted F1-Score* sebesar 89,84% memvalidasi efektivitas teknik penyesuaian bobot kelas (*class weights*). Model terbukti tidak bias: kelas minoritas seperti Lombok_Lambung dengan hanya 6 sampel uji tetap mendapat bobot yang memadai dan mampu mengimbangi performa kelas mayoritas Solo-Yogyakarta_Parang dengan 64 sampel uji. Meski capaian agregat sangat baik, *confusion matrix* pada Gambar 4 mengungkap kelemahan pada kelas dengan kemiripan visual antarkelas (*high inter-class similarity*) yang ekstrem. Tingkat salah klasifikasi tertinggi terjadi pada motif Rembang_Lasem: dari 20 sampel uji, hanya 11 yang diprediksi tepat (*True Positive*), sementara sisanya tersebar ke Dayak_Benang Bintik (3 citra), Cirebon_Liong (2 citra), dan Solo_Tribusono (2 citra) akibat tumpang tindih palet warna dan kepadatan ornamen pesisiran—motif Lasem merupakan hasil asimilasi budaya pesisiran. Kelemahan serupa tampak pada keluarga motif keraton berkerangka wajik geometris: 2 citra Solo-Yogyakarta_Sidomulyo salah dikenali sebagai Sidoluhur dan 1 citra Sidomukti meleset ke Sidomulyo. Hal ini mengindikasikan bahwa meski mekanisme *self-attention* LeViT unggul dalam memetakan konteks spasial global, resolusi ekstraksi *patch* pada lapisan awal CNN sesekali masih luput menangkap detail mikro—seperti *isen-isen* titik dan garis—yang menjadi pembeda utama antarkelas tersebut. Sebaliknya, model menunjukkan presisi sempurna pada kelas dengan kontras warna dan struktur asimetris yang tegas, seperti Cirebon_Megamendung (50/50 benar) dan Pekalongan_Tujuh Rupa (22/22 benar), tanpa satu pun *False Positive* maupun *False Negative*.

3.2 Pembahasan

Arsitektur *hybrid* CNN-ViT (LeViT-128) terbukti mampu mengatasi tantangan *fine-grained classification* pada motif batik nusantara, dengan akurasi 90,05% dan *Weighted F1-Score* 89,84%. Keunggulan ini bersumber dari sinergi antara ekstraksi fitur lokal oleh CNN—yang menangkap detail guratan canting dan tekstur kain—dengan pemetaan hubungan spasial global oleh mekanisme *self-attention* ViT, yang sangat diperlukan untuk motif geometris berulang seperti Parang dan Kawung. Capaian ini melampaui CNN murni berbasis VGG16 yang dilaporkan Aditama *et al.* (2025) dengan akurasi 85,71% pada 4 motif Solo, sekaligus mengatasi kerentanan *data-hungry* pada ViT murni sebagaimana ditemukan Nofiantoro & Saputri (2026). Teknik penyesuaian bobot kelas terbukti efektif dalam menangani ketimpangan distribusi data dengan rasio 1:10, sehingga *F1-Score* kelas minoritas tetap kompetitif. Dari sisi efisiensi komputasi, model memiliki ukuran 32,66 MB dan *latency* 12,18 ms per citra pada GPU NVIDIA Tesla T4, menjadikannya lebih siap diterapkan pada sistem identifikasi berbasis kecerdasan buatan dibandingkan arsitektur yang lebih besar. Meskipun dataset dalam penelitian ini masih didominasi citra kain datar dan optimasi *hyperparameter* belum dilakukan secara ekstensif, hasil yang dicapai membuktikan bahwa LeViT merupakan solusi yang efektif dan efisien untuk klasifikasi motif batik nusantara dengan kompleksitas visual tinggi.

4. Kesimpulan

Penelitian ini berhasil mengimplementasikan arsitektur *hybrid* CNN-ViT (LeViT-128) untuk klasifikasi 39 motif batik nusantara. Model mencapai akurasi keseluruhan sebesar 90,05% dan *Weighted F1-Score* sebesar 89,84% pada data uji, membuktikan bahwa penggabungan ekstraksi fitur lokal oleh CNN dan pemetaan hubungan spasial global oleh mekanisme *self-attention* ViT efektif dalam mengatasi tantangan klasifikasi halus pada motif batik yang memiliki tingkat kemiripan visual tinggi antarkelas.

Teknik penyesuaian bobot kelas dan augmentasi citra yang diterapkan berhasil menekan kerentanan ViT murni terhadap keterbatasan data serta mengatasi ketimpangan distribusi kelas, sehingga model tetap menunjukkan performa kompetitif pada kelas minoritas. Dari sisi efisiensi, model memiliki ukuran yang ringan (32,66 MB) dan *latency* yang rendah (12,18 ms per citra), menjadikannya siap diterapkan pada sistem identifikasi berbasis kecerdasan buatan. Penelitian selanjutnya dapat difokuskan pada perluasan dataset dengan variasi kondisi nyata seperti lipatan kain dan pencahayaan tidak seragam, penerapan teknik kompresi model untuk perangkat *edge computing*, serta eksplorasi arsitektur *hybrid* lainnya guna meningkatkan akurasi tanpa mengorbankan efisiensi komputasi.

5. Daftar Pustaka

- Aditama, D. F., Sejati, Rr. H. P., & Sanjaya, F. I. (2025). Klasifikasi motif batik Solo menggunakan *convolutional neural network* dengan *transfer learning* VGG16. *TIN: Terapan Informatika Nusantara*, 6(7), 1214–1224. <https://doi.org/10.47065/tin.v6i7.8940>
- Anggraini, A., & Zakaria, H. (2023). Penerapan metode *deep learning* pada aplikasi pembelajaran menggunakan sistem isyarat bahasa Indonesia menggunakan *convolutional neural network*: Studi kasus SLB-BC Mahardika Depok. *JURIHUM: Jurnal Inovasi dan Humaniora*, 1(4), 452–464.
- Brownlee, J. (2021). *Ensemble learning algorithms with Python: Make better predictions with bagging, boosting, and stacking*. Machine Learning Mastery.
- Chandra, A. Y., Setyaningsih, P. W., & Pratama, I. (2025). Comparative analysis of CNN architectures' performance with vision transformer for batik type classification. *Proceedings of the 2025 2nd International Conference on Information System and Information Technology (ICISIT)*, 1–6. <https://doi.org/10.1109/ICISIT66233.2025.11402957>
- Graham, B., El-Nouby, A., Touvron, H., Stock, P., Joulin, A., Jégou, H., & Douze, M. (2021). LeViT: A vision transformer in ConvNet's clothing for faster inference. *Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 12239–12249. <https://doi.org/10.1109/ICCV48922.2021.01204>
- HendryHB. (2024). *Batik Nusantara (Batik Indonesia) dataset* [Data set]. Kaggle. <https://doi.org/10.34740/KAGGLE/DSV/7641980>
- Najar, A. M., Abu, M., Ratianingsih, R., & Jaya, A. I. (2024). Revealing the relationship of batik motifs using convolutional neural network. *SISTEMASI*, 13(5), 2082. <https://doi.org/10.32520/stmsi.v13i5.4480>
- Nofiantoro, R., & Saputri, T. A. (2026). Analisis pengaruh augmentasi data pada arsitektur CNN dan *vision transformer* untuk klasifikasi motif batik. *JATI (Jurnal Mahasiswa Teknik Informatika)*, 10(4), 6677–6685. <https://doi.org/10.36040/jati.v10i4.18777>