



## KLASIFIKASI MOTIF BATIK YOGYAKARTA MENGGUNAKAN METODE GLCM DAN CNN

Ananda Rizki Dani<sup>1</sup>, Irma Handayani<sup>2</sup>

<sup>1,2</sup>Informatika, Universitas Teknologi Yogyakarta  
Yogyakarta, D.I Yogyakarta, Indonesia 55164  
anadadanipark@gmail.com, irma.handayani@staff.uty.ac.id

### Abstract

Yogyakarta batik motifs represent Indonesia's cultural heritage, but automatic classification remains challenging. This study develops a Yogyakarta batik motif classification system using a combination of the Gray Level Co-occurrence Matrix (GLCM) for texture feature extraction and Convolutional Neural Network (CNN) based on MobileNetV2 for image classification. GLCM was chosen for its ability to extract detailed texture features, while MobileNetV2 was used for its efficiency in visual pattern recognition with minimal computational resources. The dataset consists of 3,223 images from five batik motifs: Batik Ceplok, Batik Kawung, Batik Truntum, Batik Parang, and Batik Ciptoning, sourced from the Batik Keraton Museum Yogyakarta and Kaggle. The model achieved 99% accuracy, demonstrating the effectiveness of the approach in recognizing complex batik patterns. The results suggest that this system can be implemented into a mobile application with a client-server architecture for automatic motif detection. Despite promising results, the study is limited by dataset size and the complexity of specific motifs. Future research should expand the dataset and explore data augmentation techniques to improve classification accuracy for more complex motifs.

**Keywords:** Batik, Classification, CNN, GLCM, MobileNetV2

### Abstrak

Motif batik Yogyakarta memiliki keanekaragaman yang mencerminkan budaya Indonesia, namun klasifikasinya secara otomatis masih menghadapi tantangan. Penelitian ini bertujuan untuk mengembangkan sistem klasifikasi motif batik Yogyakarta menggunakan kombinasi metode *Gray Level Co-occurrence Matrix* (GLCM) untuk ekstraksi fitur tekstur dan *Convolutional Neural Network* (CNN) berbasis *MobileNetV2* untuk klasifikasi citra. GLCM dipilih karena kemampuannya dalam mengekstraksi informasi tekstur yang mendetail dari gambar, sementara CNN berbasis *MobileNetV2* digunakan karena efisiensinya dalam pengenalan pola visual dengan sumber daya komputasi yang rendah. *Dataset* yang digunakan berjumlah 3.223 gambar, terdiri dari lima kategori motif batik: Batik Ceplok, Batik Kawung, Batik Truntum, Batik Parang, dan Batik Ciptoning, yang diambil dari berbagai sumber, seperti Museum Batik Keraton Yogyakarta dan *Kaggle*. Hasil pengujian menunjukkan akurasi tinggi sebesar 99%, menandakan efektivitas pendekatan ini dalam mengidentifikasi pola motif batik yang kompleks. Implikasi penelitian ini adalah pengembangan sistem klasifikasi yang dapat diintegrasikan dalam aplikasi *mobile* berbasis *client-server* untuk mendeteksi otomatis motif batik. Meskipun hasil yang dicapai sangat baik, penelitian ini terbatas pada ukuran *dataset* dan kompleksitas motif batik tertentu. Penelitian selanjutnya disarankan untuk memperluas *dataset* dan mengeksplorasi augmentasi data untuk meningkatkan akurasi pada motif batik dengan kompleksitas lebih tinggi.

**Kata kunci:** Batik, CNN, GLCM, Klasifikasi, *MobileNetV2*

### 1. PENDAHULUAN

Sebagai negara dengan keberagaman budaya yang kaya, Indonesia mampu membanggakan warisan budayanya yang unik dan beragam. Salah satu aset budaya yang sangat dihargai adalah seni batik, khususnya motif batik khas Yogyakarta. Batik adalah salah satu bentuk warisan budaya yang berasal dari Indonesia dan telah mendapatkan pengakuan dari *UNESCO* sebagai Warisan Budaya

Takbenda Dunia[1]. Seni batik dapat diibaratkan sebagai lukisan dua dimensi dengan kain sebagai media utama. Karya ini memiliki makna dan nilai estetika yang mendalam, mencerminkan identitas budaya yang kuat. Setiap motif batik mengandung simbolisme yang terinspirasi oleh nilai-nilai animisme dan dinamisme, serta diwariskan secara turun-temurun sebagai bagian dari tradisi dan kearifan lokal[1].

Penelitian sebelumnya dalam klasifikasi batik sebagian besar terfokus pada pengenalan motif berdasarkan kota asalnya, seperti batik Solo dan Pekalongan. Beberapa penelitian juga mengkhususkan pada klasifikasi motif berdasarkan satu jenis motif atau motif batik secara umum dan kurang memfokuskan pada ciri khas motif batik khas Yogyakarta[1]. Selain itu, penggunaan CNN dan metode ekstraksi fitur tekstur GLCM telah terbukti meningkatkan akurasi klasifikasi[2], [3].

Penelitian ini mengidentifikasi celah dalam riset sebelumnya yang kurang mengoptimalkan penggunaan kombinasi GLCM dan CNN untuk klasifikasi motif batik Yogyakarta secara lebih spesifik. Oleh karena itu penelitian ini mengembangkan model yang menggabungkan metode GLCM dan CNN untuk mengklasifikasikan lima motif batik khas Yogyakarta, yaitu Batik Ceplok, Batik Kawung, Batik Truntum, Batik Parang, dan Batik Ciptoning yang dapat dibedakan berdasarkan polanya[2]. Diharapkan dengan metode ini dapat menghasilkan akurasi yang lebih tinggi dibandingkan penelitian sebelumnya.

Penelitian ini menggabungkan dua teknik yang telah terbukti efektif, yaitu *Gray Level Co-occurrence Matrix* (GLCM) dan *Convolutional Neural Network* (CNN). GLCM digunakan untuk mengekstraksi fitur tekstur dari gambar batik, yang memberikan informasi penting mengenai pola tekstur yang ada pada setiap motif batik. GLCM memiliki keunggulan dalam hal efisiensi waktu komputasi dan akurasi, serta terbukti lebih cepat dan lebih efisien dibandingkan dengan metode ekstraksi tekstur lainnya, seperti *Local Binary Patterns* (LBP) atau *Histogram of Oriented Gradients* (HOG)[4].

Sementara itu, CNN adalah algoritma yang sangat efektif dalam klasifikasi gambar, khususnya untuk data citra yang kompleks seperti motif batik. CNN dapat mengekstrak fitur visual yang lebih kompleks melalui lapisan konvolusi dan dapat mempelajari pola yang mendalam dari gambar *input*. Dalam penelitian ini, CNN didukung oleh *transfer learning* menggunakan arsitektur *MobileNetV2* yang memungkinkan model memproses dan mengenali pola gambar dengan lebih akurat[5], [6]. Dengan menggabungkan GLCM dan CNN pada penelitian ini bertujuan untuk memberikan hasil klasifikasi yang lebih optimal dalam mengenali motif batik Yogyakarta, yang memiliki keanekaragaman pola tekstur yang kompleks.

Pengembangan teknologi klasifikasi motif batik dengan memanfaatkan kombinasi GLCM dan CNN, diharapkan mampu meningkatkan akurasi klasifikasi motif batik Yogyakarta dibandingkan dengan penelitian-penelitian sebelumnya. Diharapkan hasil penelitian ini dapat memberikan kontribusi signifikan dalam pengembangan teknologi untuk pelestarian budaya batik Yogyakarta, serta menawarkan solusi klasifikasi motif batik yang lebih efisien dan akurat.

## 2. METODE PENELITIAN

### 2.1 Tinjauan Pustaka

Penelitian terkait penggunaan CNN untuk klasifikasi motif batik telah banyak dilakukan oleh peneliti di berbagai wilayah. Salah satu penelitian yang relevan dilakukan oleh Ramadhan *et al.* yang mengimplementasikan CNN untuk klasifikasi motif batik dengan menggunakan arsitektur *VGG-16*. Penelitian ini menggunakan *dataset* citra batik yang terdiri dari berbagai motif bersumber dari *Kaggle*, dan menghasilkan akurasi sekitar 85% pada data pengujian. Meskipun demikian, penelitian ini hanya terbatas pada beberapa motif batik dan belum memperhatikan penggunaan metode ekstraksi fitur tekstur secara mendalam seperti GLCM untuk meningkatkan akurasi klasifikasi[1].

Penelitian lain yang dilakukan oleh Saputra *et al.* fokus pada klasifikasi motif batik pesisir, seperti batik gentongan, lasem, dan pekalongan menggunakan CNN. Dengan menggunakan *dataset* citra batik dari situs *Kaggle* dengan nama *dataset Indonesian Batik Motifs*, penelitian ini berhasil mencapai akurasi 83% dalam klasifikasi motif. Namun, penelitian ini hanya terbatas pada motif batik pesisir dan belum mengidentifikasi motif batik khas Yogyakarta secara spesifik, yang memiliki kompleksitas pola dan tekstur yang berbeda[7].

Selanjutnya, penelitian oleh Am & Sela menggabungkan teknik ekstraksi fitur *Gray Level Co-occurrence Matrix* (GLCM) dengan jaringan saraf tiruan (JST) untuk klasifikasi motif batik Pekalongan. Meskipun GLCM terbukti efektif dalam menangkap informasi tekstur, hasil penelitian menunjukkan akurasi 55,5% pada data uji, yang masih dapat ditingkatkan dengan pengolahan citra yang lebih kuat[8].

Penelitian yang dilakukan oleh Ma'ruf *et al.* memanfaatkan metode ekstraksi fitur GLCM untuk menganalisis tekstur pada motif batik Besurek. Hasil penelitian menunjukkan bahwa penggunaan GLCM berhasil mencapai akurasi hingga 93%, yang menandakan efektivitas metode ini dalam mengklasifikasikan motif batik dengan baik. Meskipun demikian, penelitian ini hanya berfokus pada satu jenis batik dari daerah tertentu dan belum mengintegrasikan penggunaan teknologi lain seperti CNN untuk meningkatkan akurasi klasifikasi secara keseluruhan[3].

Penggunaan CNN untuk klasifikasi motif batik khas Yogyakarta telah dilakukan oleh Prayoga *et al.* yang menggunakan CNN dengan arsitektur *InceptionResNetV2* dan memperoleh akurasi 87,83%. Meskipun demikian, penelitian ini belum memanfaatkan ekstraksi fitur GLCM untuk mendeteksi tekstur halus yang ada pada motif batik Yogyakarta [2].

Dari penelitian-penelitian terdahulu, telah dilakukan berbagai perancangan atau implementasi sistem untuk mengklasifikasi motif batik menggunakan beberapa metode yang berbeda. Hasilnya, terdapat variasi dalam akurasi yang

dicapai oleh setiap penelitian, serta efisiensi dalam penerapannya.

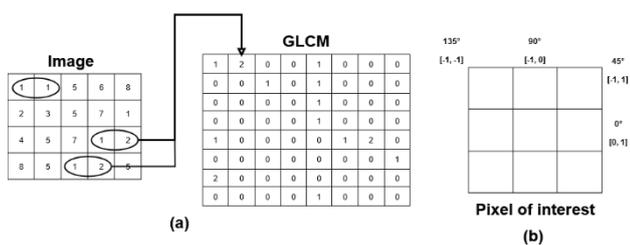
Penelitian ini memiliki keterkaitan dengan penelitian-penelitian sebelumnya. Perbedaan utama penelitian ini terletak pada algoritma yang digunakan dan jenis motif batik. Sebagian penelitian sebelumnya hanya fokus pada identifikasi motif batik tanpa membedakan jenisnya, sedangkan penelitian ini menggabungkan identifikasi motif dan jenis batik Yogyakarta, yang membuatnya menjadi penelitian yang berbeda dan inovatif dari penelitian sebelumnya. Lebih lanjut, dalam implementasinya, penelitian ini menggunakan metode GLCM untuk ekstraksi fitur tekstur dari gambar batik, serta CNN untuk pemrosesan gambar secara lebih mendalam.

## 2.2 Landasan Teori

### 2.2.1 Gray Level Co-occurrence Matrix (GLCM)

GLCM digunakan untuk menganalisis pola tekstur dalam gambar dengan menghitung fitur seperti kontras, homogenitas, energi, dan entropi, yang memberikan informasi tentang struktur lokal gambar. Metode ini memiliki aplikasi luas, termasuk dalam pengenalan pola, klasifikasi citra, dan analisis medis, seperti mendeteksi pola anomali pada gambar medis. Dengan demikian, GLCM menjadi alat yang penting untuk memahami dan memanfaatkan hubungan spasial antar-piksel dalam analisis citra, yang dapat digunakan untuk membedakan motif batik Yogyakarta[9].

GLCM cocok digunakan dalam klasifikasi motif batik karena fitur-fitur teksturnya mampu menangkap perbedaan pola yang sering kali hanya terlihat pada tingkat mikroskopis. Dalam motif batik yang rumit, GLCM membantu mendeteksi pola, arah, atau keteraturan, yang kemudian meningkatkan akurasi model *deep learning* seperti *MobileNetV2* dalam mengenali motif dengan lebih baik, meskipun jumlah data pelatihan terbatas.

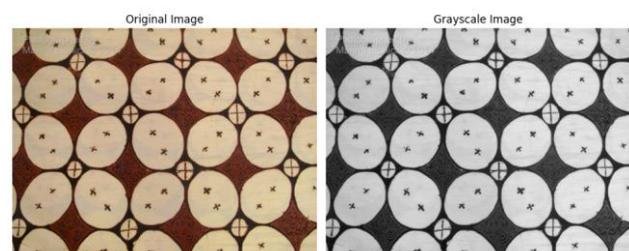


Gambar 1. Ilustrasi cara kerja GLCM.

Gambar 1 mengilustrasikan cara kerja GLCM, metode yang digunakan dalam analisis citra untuk menghitung hubungan spasial antara intensitas piksel dalam gambar. Pada panel (a), gambar *input* berupa matriks intensitas piksel ditunjukkan, di mana setiap elemen mewakili nilai intensitas piksel (contoh: 1, 2, 5, dll.). GLCM menghitung seberapa sering pasangan nilai piksel tertentu muncul berdekatan dalam arah tertentu. Dalam contoh ini, hubungan dihitung dalam arah horizontal ( $0^\circ$ ), sehingga matriks GLCM

terbentuk dengan elemen-elemen yang merepresentasikan jumlah kemunculan pasangan nilai piksel, seperti (1,1), (1,2), dan seterusnya. Sebagai contoh, pasangan (1,2) muncul dua kali di gambar, sehingga nilai pada matriks GLCM di posisi (1,1) bernilai 2.

Panel (b) menjelaskan arah yang dapat digunakan untuk menghitung hubungan antar-piksel, yaitu  $0^\circ$  (horizontal),  $45^\circ$  (diagonal ke kanan atas),  $90^\circ$  (vertikal), dan  $135^\circ$  (diagonal ke kiri bawah), dengan jarak antar piksel sebesar 1 piksel [10]. Setiap arah merepresentasikan posisi relatif piksel yang berpasangan, seperti piksel di sebelah kanan, di bawah, atau pada diagonal tertentu. Proses ini menghasilkan matriks GLCM untuk masing-masing arah, yang kemudian dapat digunakan untuk ekstraksi fitur tekstur.



Gambar 2. Perubahan citra sebelum dan sesudah dilakukan ekstraksi fitur GLCM.

Sebagai perbandingan gambar 2 merupakan perbedaan citra sebelum dan sesudah diekstraksi dengan fitur GLCM. Pada bagian kiri gambar 2 menampilkan motif batik dalam bentuk dan warna aslinya. Warna pada batik sering kali memiliki makna budaya dan estetika, namun dalam proses analisis tekstur berbasis GLCM, informasi warna diabaikan. Hal ini karena GLCM fokus pada hubungan spasial dan intensitas piksel, sehingga data warna tidak diperlukan untuk analisis.

Bagian kanan Gambar 2 menunjukkan motif batik yang telah dikonversi menjadi *grayscale* atau skala abu-abu. Setiap piksel dalam citra *grayscale* memiliki intensitas antara 0 (hitam) hingga 255 (putih). Langkah ini sangat penting karena GLCM hanya bekerja pada citra *grayscale*, dengan menganalisis hubungan antar piksel berdasarkan intensitas. Dengan mengubah gambar menjadi *grayscale*, analisis berfokus pada tekstur dan pola tanpa gangguan variasi warna.

Proses transformasi dari gambar berwarna ke *grayscale* memungkinkan analisis yang lebih sederhana namun akurat terhadap pola tekstur. Pada motif batik, keteraturan dan variasi pola menjadi informasi penting dalam pengenalan motif.

### 2.2.2 Convolutional Neural Network (CNN)

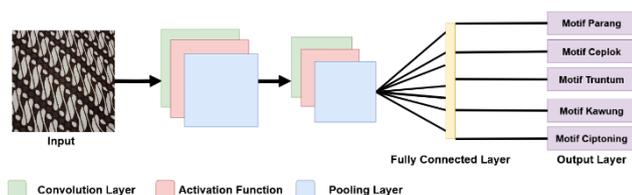
CNN merupakan jenis jaringan saraf tiruan yang dirancang khusus untuk memproses data berbentuk *grid* seperti gambar. CNN sangat efektif dalam mengekstraksi dan mengenali pola dari data citra karena memiliki kemampuan

untuk memahami fitur sederhana seperti garis atau tepi hingga fitur kompleks seperti bentuk atau pola tertentu. Dalam penelitian ini, CNN digunakan untuk melakukan klasifikasi motif batik, dengan kemampuan untuk mengenali tekstur dan pola khas dari motif batik yang berbeda[11].

Jaringan saraf bekerja menyerupai cara kerja otak manusia, di mana model dapat dilatih secara bertahap untuk meningkatkan pemahaman dan mencapai akurasi yang optimal. Dengan menganalisis setiap piksel pada gambar dan mencocokkannya dengan data yang tersedia, CNN sangat efektif dalam memproses dokumen atau teks yang rusak. Teknologi ini juga ideal untuk menangani permasalahan spesifik, seperti analisis data pasar saham. Hingga saat ini, CNN terbukti lebih efisien dibandingkan dengan berbagai metode lain dalam menyelesaikan beragam tugas klasifikasi dan prediksi.[1].

Arsitektur CNN terdiri dari beberapa lapisan utama, yaitu *convolution layer*, *pooling layer*, dan *fully connected layer*. *Convolution layer* bertugas mengekstraksi fitur dari citra *input* melalui kernel atau filter yang bergerak melintasi seluruh citra, menghasilkan peta fitur. Setelah itu, *pooling layer* digunakan untuk mengurangi dimensi peta fitur ini, dengan mempertahankan informasi penting sekaligus mengurangi kompleksitas komputasi. Seluruh fitur yang telah diekstraksi di lapisan sebelumnya kemudian dihubungkan menjadi satu vektor fitur dalam *fully connected layer*. Lapisan ini memadukan semua informasi dari peta fitur dan meneruskannya ke *output layer*, di mana setiap neuron pada *output layer* mewakili satu kelas motif batik, seperti Parang, Ceplok, Truntum, Kawung, dan Ciptoning. Model menghasilkan probabilitas untuk setiap kelas, dan kelas dengan probabilitas tertinggi dipilih sebagai hasil prediksi.

Untuk meningkatkan kemampuan model dalam belajar pola yang lebih kompleks, *activation function ReLU (Rectified Linear Unit)* diterapkan setelah *Convolution layer*. *ReLU* berfungsi untuk memperkenalkan non-linearitas pada jaringan, memungkinkan model untuk mempelajari hubungan yang lebih rumit antara fitur yang diekstraksi dari citra. *ReLU* bekerja dengan cara mengubah nilai negatif menjadi nol, sementara nilai positif tetap tidak berubah. Dengan penerapan *ReLU*, jaringan saraf menjadi lebih cepat dalam proses pelatihan, meningkatkan efisiensi dan akurasi model dalam memprediksi klasifikasi motif batik.



Gambar 3. Ilustrasi arsitektur CNN pada penelitian ini.

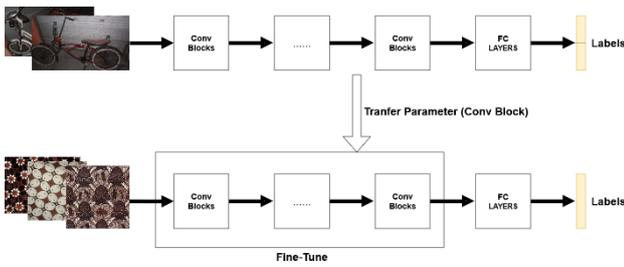
Ilustrasi pada gambar 3 memperlihatkan proses yang dilakukan oleh CNN dalam klasifikasi motif batik. Citra *input* berupa gambar motif batik diproses melalui *convolution layer*, yang berfungsi mendeteksi fitur dasar seperti garis, pola tepi, atau bentuk sederhana. Fungsi aktivasi *ReLU* diterapkan pada setiap *convolution layer* untuk mempercepat proses pelatihan dan meningkatkan kemampuan jaringan dalam menangkap pola yang lebih kompleks. Peta fitur yang dihasilkan kemudian melalui *pooling layer* untuk mereduksi dimensi sambil mempertahankan fitur-fitur penting. Setelah melewati semua layer ini, peta fitur digabungkan menjadi vektor pada *fully connected layer* dan diteruskan ke *output layer*, di mana hasil klasifikasi berupa jenis motif batik ditentukan.

### 2.2.5 Transfer Learning

*Transfer learning* adalah teknik dalam pembelajaran mesin yang memungkinkan pemanfaatan model yang telah dilatih sebelumnya pada suatu *dataset* untuk diterapkan pada *dataset* baru dengan beberapa modifikasi. Teknik ini sangat berguna ketika pengumpulan data berlabel sulit dilakukan, seperti dalam klasifikasi motif batik, karena model dapat memperoleh pengetahuan dari *dataset* yang lebih besar dan umum. Dalam proses *transfer learning*, model yang telah dilatih pada *dataset* besar, seperti *ImageNet*, digunakan sebagai titik awal, yang memungkinkan model belajar lebih cepat dan mencapai akurasi yang lebih baik meskipun hanya menggunakan *dataset* pelatihan yang terbatas[12].

Pada penelitian ini, menggunakan model *MobileNetV2* sebagai dasar arsitektur CNN. *MobileNetV2* adalah model yang sangat efisien, menggunakan teknik seperti *depthwise separable convolution* untuk mengurangi jumlah parameter tanpa mengorbankan akurasi. Dengan proses *transfer learning*, model yang telah dilatih sebelumnya pada *dataset* umum seperti *ImageNet* digunakan untuk memulai pelatihan pada *dataset* motif batik. Model yang sudah dilatih ini membawa pengetahuan dasar tentang pola umum dalam gambar, dan hanya perlu disesuaikan dengan *dataset* spesifik melalui proses *fine-tuning*.

*Fine-tuning* dalam *transfer learning* adalah langkah selanjutnya di mana lapisan-lapisan tertentu dari model yang sudah dilatih diperbarui untuk lebih menyesuaikan diri dengan data baru, dalam hal ini motif batik Yogyakarta. Pada proses *fine-tuning*, lapisan-lapisan awal yang berfungsi untuk mengekstraksi fitur dasar seperti garis, tepi, dan bentuk biasanya tetap tidak berubah, karena fitur-fitur tersebut bersifat umum dan dapat diterapkan pada berbagai *dataset*. Hanya lapisan-lapisan lebih dalam yang perlu diperbarui untuk mengenali pola yang lebih spesifik terkait motif batik.



Gambar 4. Ilustrasi cara kerja *Transfer Learning*.

Proses *transfer learning* ini bekerja dengan baik karena model sudah memiliki pengetahuan tentang pola dasar dari *dataset* yang lebih besar dan dapat dengan cepat disesuaikan dengan pola baru yang ada pada motif batik. Seperti yang terlihat pada gambar 4, bahwa model awal dilatih menggunakan *dataset* pertama (misalnya gambar sepeda). Model ini terdiri dari beberapa blok *convolutional* (*Conv Blocks*) untuk mengekstrak fitur-fitur penting dari gambar dan *fully connected layers* (*FC Layers*) untuk melakukan klasifikasi. Setelah model berhasil dilatih, parameter yang dipelajari oleh bagian *Conv Blocks* ditransfer ke model baru. Pada bagian bawah gambar, model baru diadaptasi untuk *dataset* kedua (gambar motif batik) dengan menggunakan *Conv Blocks* dari model awal. Proses ini diikuti dengan *fine-tuning*, di mana lapisan *Conv Blocks* bisa dipertahankan (dibekukan) atau disesuaikan, sementara *FC Layers* dirancang ulang agar sesuai dengan label dari *dataset* baru. *Transfer learning* memungkinkan model untuk belajar lebih cepat dan efisien, terutama ketika *dataset* baru berukuran kecil, karena model sudah memiliki kemampuan awal dalam mengenali fitur umum dari data sebelumnya.

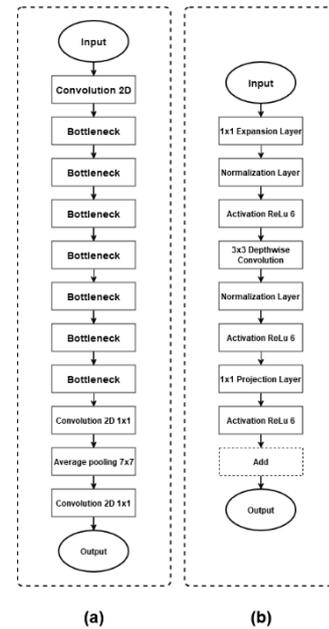
Dengan memanfaatkan teknik *transfer learning* dan *fine-tuning*, model dapat memberikan prediksi yang sangat akurat pada motif batik meskipun data pelatihan terbatas, berkat pengetahuan awal yang diperoleh dari model yang telah dilatih pada *dataset* yang lebih besar dan beragam. Teknik ini sangat efektif dalam meningkatkan kinerja model pada tugas spesifik, seperti pengenalan pola pada motif batik Yogyakarta[13].

### 2.2.6 *MobileNetV2*

*MobileNetV2* adalah model *pre-trained* yang dirancang untuk tugas *computer vision* pada perangkat dengan keterbatasan komputasi, seperti perangkat seluler dan sistem *embedded*. Dirilis pada 2018, *MobileNetV2* mengoptimalkan efisiensi dan akurasi dengan memperkenalkan beberapa perbaikan pada arsitektur versi sebelumnya[14]. Salah satu keunggulan utama dari *MobileNetV2* adalah penggunaan *Depthwise Separable Convolution* dan *Inverted Residuals with Linear Bottlenecks*, yang memungkinkan model ini mengurangi jumlah parameter dan operasi tanpa mengorbankan performa[15], sehingga sangat efektif dalam klasifikasi motif batik. Selain efisiensinya, *MobileNetV2* juga memberikan hasil yang akurat dalam mendeteksi pola-pola

yang kompleks pada citra batik, menjadikannya pilihan tepat dalam penelitian ini[5].

Seperti yang ditunjukkan pada gambar 5 di bawah ini, bagian (a), *MobileNetV2* dimulai dengan lapisan konvolusi 2D yang bertugas untuk mengekstrak fitur dasar dari gambar *input*. Selanjutnya, serangkaian blok *bottleneck* digunakan untuk menangkap fitur lebih lanjut dengan efisiensi tinggi. Blok *bottleneck* ini terdiri dari tiga tahap: pertama, *expansion* untuk memperbesar dimensi *channel*, kemudian *depthwise convolution* untuk melakukan konvolusi secara terpisah pada setiap *channel*, dan akhirnya *projection* yang mengembalikan dimensi *channel* ke ukuran awal. Proses ini memungkinkan *MobileNetV2* untuk mengurangi jumlah parameter secara signifikan, sekaligus mempertahankan kualitas fitur yang diekstrak.



Gambar 5. Ilustrasi arsitektur *MobileNetV2*.

Pada gambar 5 bagian (b), detail dari blok *bottleneck* juga digambarkan lebih rinci. Setiap blok dimulai dengan *1x1 Expansion Layer* untuk memperbesar dimensi *channel*, diikuti oleh normalisasi dan aktivasi *ReLU6*, yang sangat berguna untuk menjaga stabilitas numerik pada perangkat dengan keterbatasan daya komputasi. Setelah itu, dilakukan *3x3 Depthwise Convolution*, yang memberikan efisiensi besar dalam mengurangi jumlah parameter. Tahap ini diakhiri dengan *1x1 Projection Layer*, yang mengembalikan dimensi *channel* ke ukuran awal. Koneksi residual ditambahkan jika dimensi *input* dan *output* sama, menjaga aliran informasi antar lapisan. Struktur *Inverted Residuals* membalikkan pendekatan residual tradisional dengan memperbesar dimensi *channel* terlebih dahulu sebelum mengembalikannya ke dimensi awal. Keberadaan aktivasi linear di akhir *bottleneck* membantu mempertahankan informasi yang penting. Keunggulan efisiensi ini menjadikan *MobileNetV2* pilihan tepat untuk pengenalan motif batik pada perangkat dengan sumber daya terbatas,

sehingga dapat memberikan performa tinggi dengan ukuran model yang ringan dan pemrosesan yang cepat.

### 2.3 Pengumpulan Data

*Dataset* yang digunakan dalam penelitian ini berupa citra motif batik yang berasal dari berbagai sumber. Citra diperoleh dengan cara memotret langsung menggunakan kamera *handphone* dari beberapa lokasi, yaitu Museum Batik Keraton Yogyakarta, Museum Sonobudoyo, dan *Langitosa Studio*. Selain itu, sebagian data juga diunduh melalui platform *Kaggle* untuk melengkapi jumlah *dataset*. Secara keseluruhan, total citra yang digunakan dalam penelitian ini mencapai 3.223 gambar, mencakup lima kategori motif batik, yaitu Batik Ceplok, Batik Kawung, Batik Truntum, Batik Parang, dan Batik Ciptoning.

Proses pengumpulan data dilakukan secara bertahap untuk memastikan kualitas citra yang optimal. Untuk citra yang diambil langsung, pemotretan dilakukan dengan berbagai sudut dan pencahayaan guna menangkap detail pola batik secara jelas. Setiap gambar dipotret berulang kali untuk meminimalkan *noise* dan memastikan keseragaman kualitas citra. Sementara itu, gambar yang diunduh dari *Kaggle* dipilih secara selektif dengan mempertimbangkan kesesuaian motif dan resolusi gambar.

Batik Ceplok merupakan motif batik khas Kabupaten Bantul, Daerah Istimewa Yogyakarta. Motif ini terinspirasi dari komponen tanaman pepaya, yang dalam bahasa Jawa disebut "kates." Unsur utama pada motif ini adalah bentuk biji dan bunga pepaya, yang menjadi ciri khas desainnya.

Pola Batik Kawung terdiri dari bulatan-bulatan yang menyerupai buah kawung, seperti kelapa atau aren (kolangkaling), yang disusun secara geometris dan simetris[16]. Pola ini sering kali diartikan sebagai bunga lotus dengan empat kelopak yang berkembang sempurna. Dalam konteks budaya, bunga lotus melambangkan kesucian dan umur panjang.

Batik Truntum memiliki makna filosofis tentang cinta yang tumbuh dan berkembang kembali. Motif ini melambangkan cinta sejati yang tulus, abadi, dan semakin subur seiring berjalannya waktu. Oleh karena itu, motif Truntum kerap dikenakan oleh orang tua pengantin pada hari pernikahan sebagai simbol restu dan cinta tanpa syarat[17].

Batik Parang merupakan salah satu motif batik tertua di Indonesia. Kata "parang" berasal dari istilah Jawa "*pèrèng*," yang berarti lereng. Polanya berupa rangkaian huruf "S" yang tersusun saling berkaitan dan membentuk pola diagonal miring menyerupai lereng gunung. Desain ini mencerminkan dinamika hidup dan kekuatan yang berkelanjutan[18].

Batik Ciptoning mengandung elemen simbolis seperti gurdo, parang, dan wayang, yang melambangkan kebijaksanaan. Motif ini menggambarkan harapan agar pemakainya terlihat berwibawa, santun, dan bijak. Batik

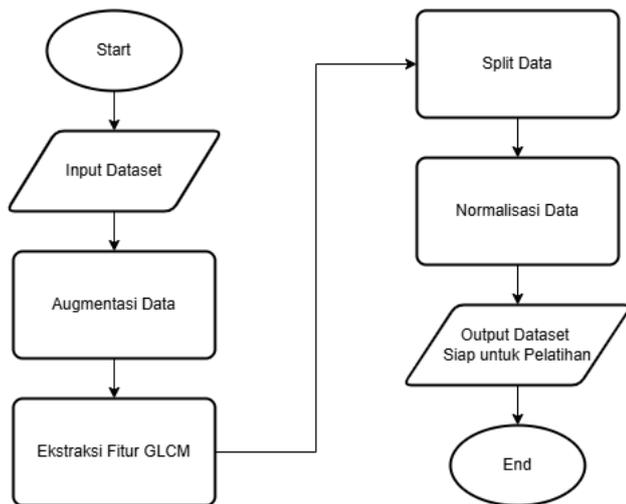
Ciptoning sering digunakan pada acara resmi karena kesan anggun dan penuh makna yang terkandung di dalamnya[17]

Tabel 1. Gambar Motif Batik

Nama Motif Batik	Motif Batik
Batik Ceplok	
Batik Kawung	
Batik Truntum	
Batik Parang	
Batik Ciptoning	

### 2.4 Preprocessing

Tahapan *preprocessing* data dalam penelitian ini dirancang untuk memastikan *dataset* batik siap digunakan secara optimal oleh model CNN.



Gambar 6. Flowchart proses Preprocessing.

Seperti yang terlihat pada *flowchart* di gambar 6 langkah pertama yang dilakukan adalah *augmentasi data*, yang bertujuan untuk memperkaya variasi *dataset* citra batik. Teknik *augmentasi* yang digunakan mencakup rotasi, pergeseran posisi, *shear*, dan *zoom*. Proses ini menghasilkan citra baru dengan variasi yang beragam tanpa mengubah label aslinya. *Augmentasi* membantu model menjadi lebih tangguh dalam mengenali pola dari berbagai variasi motif batik, sehingga meningkatkan kemampuan generalisasi model terhadap data baru[19][20].

Setelah proses *augmentasi* selesai, langkah berikutnya adalah ekstraksi fitur tekstur menggunakan metode GLCM. Metode ini digunakan untuk mengambil informasi tekstur yang relevan dari setiap citra batik. Empat parameter utama yang dihitung melalui GLCM adalah kontras, homogenitas, energi, dan entropi. Kontras mengukur perbedaan intensitas antara piksel dengan tetangganya, yang mencerminkan tingkat ketajaman tekstur. Homogenitas menggambarkan keseragaman pola tekstur dengan melihat kedekatan distribusi nilai intensitas piksel terhadap diagonal matriks GLCM. Energi mencerminkan konsistensi pola dalam tekstur, sementara entropi menunjukkan tingkat kerandaman atau ketidakpastian dalam pola tekstur[21][4].

Data hasil ekstraksi fitur GLCM ini kemudian digabungkan dengan informasi visual dari citra asli. Selanjutnya, *dataset* dibagi menjadi *train set* dan *validation set* dengan rasio 80:20, dilakukan secara acak untuk menghindari bias akibat urutan data. Setiap piksel pada citra juga dinormalisasi dengan membagi nilai intensitasnya dengan 255, sehingga semua nilai berada dalam rentang 0 hingga 1. Proses normalisasi ini bertujuan untuk meningkatkan efisiensi komputasi dan kestabilan selama pelatihan model.

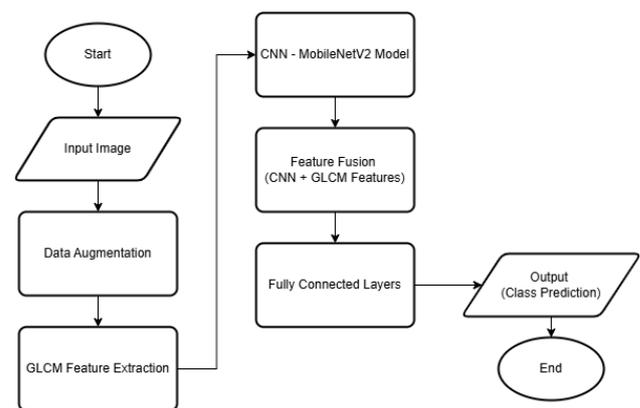
Dengan melalui serangkaian tahapan ini, *dataset* batik dipersiapkan dengan baik untuk digunakan pada tahap pelatihan model CNN. Informasi tekstur yang diperoleh melalui ekstraksi fitur GLCM digabungkan dengan pola visual citra, memungkinkan model untuk mengenali pola dan karakteristik unik dari setiap motif batik secara akurat.

## 2.5 Pembuatan Model

Dalam penelitian ini, arsitektur *MobileNetV2* yang didasarkan pada CNN diterapkan untuk klasifikasi gambar batik. Model CNN terdiri dari beberapa komponen utama, antara lain lapisan konvolusi, *pooling*, dan *fully connected layer*. Lapisan konvolusi berfungsi untuk mengekstrak fitur-fitur penting dari gambar, seperti kontur, pola, dan tekstur, dengan memanfaatkan filter atau *kernel* yang diterapkan pada data masukan.

Penggunaan lapisan-lapisan ini memungkinkan model untuk mengenali karakteristik unik dari berbagai motif batik secara efektif. Setiap filter akan menghasilkan *feature map* yang merepresentasikan bagian penting dari gambar. Lapisan *pooling*, seperti *MaxPooling*, bertujuan mengurangi dimensi fitur tanpa menghilangkan informasi penting, sehingga mempercepat proses komputasi dan mengurangi risiko *overfitting*[6].

Dalam penelitian ini, alur kerja sistem pemrosesan data dirancang untuk menghasilkan *dataset* yang siap digunakan dalam pelatihan model kecerdasan buatan untuk klasifikasi motif batik. Proses dimulai dengan menerima gambar *input*, yang merupakan citra batik, sebagai data utama yang akan diproses lebih lanjut.



Gambar 7. Flowchart System.

Seperti yang diilustrasikan pada *flowchart* di gambar 7, citra tersebut kemudian melalui tahap *augmentasi* data bertujuan untuk meningkatkan keragaman *dataset* dengan menciptakan variasi pada gambar tanpa mengubah labelnya. Hal ini memperkuat kemampuan model dalam mengenali pola dari berbagai variasi motif batik selama pelatihan.

Setelah proses *augmentasi*, gambar yang telah dimodifikasi diproses untuk ekstraksi fitur tekstur menggunakan metode GLCM (*Gray-Level Co-occurrence Matrix*). GLCM mengukur hubungan spasial antara piksel dalam citra dan menghasilkan empat fitur tekstur utama, yaitu kontras, homogenitas, energi, dan entropi. Fitur-fitur tekstur ini memberikan informasi penting yang menggambarkan karakteristik pola tekstur pada motif batik. Informasi tekstur ini sangat berguna untuk memperkaya representasi data gambar dalam proses pelatihan model, meningkatkan

kemampuan model dalam mengklasifikasikan motif batik berdasarkan pola visual dan tekstur yang ada pada gambar.

Citra yang telah mengalami augmentasi dan ekstraksi fitur tekstur kemudian dimasukkan ke dalam model CNN berbasis *MobileNetV2*. Arsitektur *MobileNetV2* dipilih karena efisiensinya dalam mengekstraksi fitur visual dari gambar dengan performa yang optimal, terutama untuk *dataset* yang besar dengan keterbatasan sumber daya. *MobileNetV2* akan mengenali pola-pola, bentuk, dan struktur visual pada motif batik. Fitur yang dihasilkan dari *MobileNetV2* kemudian digabungkan dengan fitur tekstur yang diperoleh dari GLCM dalam proses yang disebut *Feature Fusion*. Penggabungan kedua fitur ini bertujuan untuk memanfaatkan kelebihan dari masing-masing metode, yaitu fitur visual yang ditangkap oleh CNN dan informasi tekstur dari GLCM, sehingga menghasilkan representasi data yang lebih kaya dan mendalam.

Hasil penggabungan fitur visual dan tekstur kemudian diproses melalui *fully connected layers*, yang bertugas untuk menyusun dan merestrukturisasi fitur-fitur gabungan menjadi format yang siap untuk pelatihan model. Proses ini membantu meningkatkan akurasi model dengan mengoptimalkan representasi data. Hasil akhir sistem berupa *Output*, yang memberikan prediksi kelas berdasarkan data masukan. Sistem ini dirancang untuk menyelesaikan tugas klasifikasi dengan memanfaatkan informasi yang kaya dari fitur visual dan tekstur. Sistem ini kemudian siap untuk melatih model kecerdasan buatan dengan menggunakan data yang telah dioptimalkan, yang diharapkan dapat menghasilkan model klasifikasi motif batik dengan tingkat akurasi yang tinggi.

## 2.6 Evaluasi

Pada bagian *Classification Report*, model dievaluasi berdasarkan empat metrik utama. Pertama, *accuracy* digunakan untuk mengukur rasio antara jumlah prediksi yang benar dengan keseluruhan prediksi yang dihasilkan. Selanjutnya, *precision* mengevaluasi ketepatan dari setiap prediksi positif yang dihasilkan oleh model. *Recall* berfungsi untuk menilai sejauh mana model mampu mendeteksi kasus positif yang seharusnya dikenali. Terakhir, *F1-Score* menyajikan nilai keseimbangan antara *precision* dan *recall*.

Evaluasi model dengan menggunakan metrik seperti *accuracy*, *precision*, *recall*, dan *F1-score* memungkinkan kita untuk menilai kemampuan model dalam membuat prediksi yang benar dan memahami bagaimana model menangani data dengan berbagai karakteristik. Dalam konteks klasifikasi motif batik, di mana data bisa sangat bervariasi dan tidak selalu seimbang antar kelas, penggunaan metrik yang lebih detail ini penting untuk memastikan bahwa model tidak hanya memberikan hasil yang akurat secara keseluruhan, tetapi juga dapat menangani setiap kelas dengan baik.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$$

*Accuracy* mengukur seberapa banyak prediksi yang benar dilakukan oleh model dibandingkan dengan total prediksi yang dihasilkan. Metrik ini memberikan gambaran umum tentang kinerja model secara keseluruhan, namun memiliki keterbatasan, terutama pada *dataset* yang tidak seimbang. Misalnya, jika satu kelas jauh lebih dominan daripada kelas lainnya, model mungkin bisa mencapai akurasi tinggi meskipun gagal mengenali kelas minoritas dengan baik. Rumus *accuracy* dihitung dengan membagi jumlah prediksi yang benar (*True Positive + True Negative*) dengan total jumlah data (*True Positive + True Negative + False Positive + False Negative*). Oleh karena itu, meskipun *accuracy* memberikan indikasi umum kinerja model, ia tidak cukup untuk menilai performa model secara menyeluruh dalam pengenalan pola yang lebih kompleks seperti motif batik.

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

*Precision* mengukur ketepatan model dalam memprediksi kelas positif. Dalam konteks klasifikasi motif batik, *precision* membantu menilai seberapa sering model memprediksi motif tertentu dengan benar, tanpa memprediksi motif lain sebagai motif yang dimaksud. *Precision* dihitung dengan membagi jumlah *True Positive* dengan total prediksi positif (*True Positive + False Positive*). *Precision* yang tinggi menunjukkan bahwa model jarang melakukan kesalahan positif (*false positive*). *Precision* sangat penting jika kesalahan positif (misalnya, salah mengklasifikasikan motif) harus dihindari, karena kesalahan tersebut dapat merugikan dalam aplikasi praktis seperti pengenalan motif batik.

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

*Recall* berfungsi untuk mengukur seberapa baik model dalam mendeteksi seluruh kasus positif, yang penting terutama dalam situasi di mana kehilangan prediksi positif (*false negative*) dapat berbahaya. *Recall* dihitung dengan membagi *True Positive* dengan total data positif sebenarnya (*True Positive + False Negative*). Sebuah model dengan *recall* tinggi mampu mengidentifikasi sebagian besar contoh positif yang ada dalam data, yang berarti lebih sedikit motif yang terlewatkan. Nilai *recall* yang tinggi berarti model berhasil menemukan sebagian besar kasus positif, yang sangat penting dalam situasi di mana mengabaikan kasus positif berbahaya, seperti deteksi penipuan atau pengenalan motif yang langka.

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

*F1-Score* menggabungkan *precision* dan *recall* menjadi satu angka yang memberikan gambaran keseimbangan antara keduanya. *F1-score* sangat berguna dalam situasi di mana data tidak seimbang, atau ketika kita perlu menyeimbangkan akurasi prediksi positif dan cakupan

positif. *F1-Score* dihitung menggunakan rumus  $2 \times ((Precision \times Recall) / (Precision + Recall))$ . Metrik ini sangat penting dalam pengklasifikasian motif batik, di mana baik presisi maupun *recall* harus diperhatikan agar model dapat mengenali dan mengklasifikasikan motif dengan akurat tanpa mengabaikan kelas minoritas.

## 2.7 Eksperimen

Pada tahap eksperimen dalam penelitian ini, desain pengujian dirancang untuk mengevaluasi performa beberapa model *transfer learning* yang digunakan dalam klasifikasi motif batik, yaitu *MobileNetV2*, *ResNet50*, dan *InceptionResNetV2*. Setiap model diuji pada *dataset* yang sama, dengan pembagian data *training* dan *validation* menggunakan rasio 80:20. Rasio ini dipilih untuk memberikan *dataset* yang cukup besar bagi model untuk belajar, sekaligus menyediakan data validasi yang representatif untuk mengukur kemampuan generalisasi model.

Proses pengujian melibatkan pelatihan model selama 100 *epoch*, dengan *optimizer Adam* yang digunakan untuk mempercepat proses pelatihan dan menyesuaikan bobot jaringan secara efisien. Fungsi aktivasi *ReLU* diterapkan pada *fully connected layers*, yang dirancang untuk mengolah fitur-fitur dari *convolutional layers* secara lebih mendalam sebelum menghasilkan prediksi akhir. Setiap model memanfaatkan fitur tekstur dari GLCM yang dikombinasikan dengan fitur CNN untuk meningkatkan akurasi klasifikasi motif batik.

*Optimizer Adam (Adaptive Moment Estimation)* dipilih dalam penelitian ini karena kemampuannya untuk menyesuaikan *learning rate* secara adaptif selama proses pelatihan. *Adam* menggabungkan keunggulan dari momentum dan *RMSProp (Root Mean Square Propagation)*, yang memanfaatkan estimasi rata-rata momentum gradien untuk mempercepat konvergensi dan estimasi kedua momen gradien untuk mengontrol kecepatan pembaruan parameter. Pendekatan ini menghasilkan optimasi yang lebih stabil, terutama pada *dataset* dengan fitur kompleks seperti motif batik, di mana gradien seringkali tidak homogen atau tidak stabil.

*Adam* telah terbukti sangat efektif dalam melatih jaringan yang dalam (*deep networks*) dan *dataset* yang besar. *Optimizer* ini menawarkan keunggulan dibandingkan metode tradisional seperti *SGD (Stochastic Gradient Descent)*, yang membutuhkan pengaturan *learning rate* secara manual untuk mencapai hasil optimal. Kecepatan konvergensi *Adam* membuatnya menjadi pilihan utama dalam banyak penelitian klasifikasi gambar. Secara *default*, *Adam* menggunakan *learning rate* 0.001, yang telah terbukti optimal dalam berbagai tugas klasifikasi gambar termasuk pengenalan pola dan pengolahan citra[6], [11].

Jumlah *epoch* dalam pelatihan model adalah faktor penting yang memengaruhi performa model. Dalam penelitian ini,

100 *epoch* dipilih berdasarkan eksperimen awal yang menunjukkan bahwa model mencapai konvergensi optimal pada jumlah *epoch* ini. Selama pelatihan dengan jumlah *epoch* di bawah 100, terlihat bahwa akurasi model pada data validasi terus meningkat secara signifikan. Namun, jika jumlah *epoch* melebihi 100, model mulai menunjukkan tanda-tanda *overfitting*, yaitu ketika akurasi data latih tetap meningkat tetapi performa pada data validasi *stagnan* atau bahkan menurun.

Eksperimen menunjukkan bahwa pada *epoch* ke-100, akurasi model sudah mendekati nilai maksimum, dengan *train loss* yang stabil. Setelah titik ini, peningkatan akurasi menjadi sangat kecil, sementara nilai *loss* pada data validasi cenderung tidak berubah atau sedikit meningkat. Berdasarkan grafik *training curve*, 100 *epoch* memberikan keseimbangan terbaik antara akurasi yang tinggi dan kemampuan generalisasi model.

Keputusan ini juga didukung oleh studi-studi sebelumnya menunjukkan bahwa model *transfer learning* seperti *MobileNetV2* dan *InceptionResNetV2* biasanya mencapai performa optimal dalam rentang 50 hingga 150 *epoch*, tergantung pada ukuran *dataset* dan kompleksitas model[12], [13].

*MobileNetV2*, *ResNet50*, dan *InceptionResNetV2* adalah arsitektur CNN yang dirancang untuk menangani berbagai tugas klasifikasi gambar, namun dengan perbedaan mendasar dalam desain dan cara mereka menangani kompleksitas data. Berikut adalah penjelasan perbedaan utama antara ketiganya:

*MobileNetV2* dirancang untuk efisiensi tinggi, terutama pada perangkat dengan keterbatasan sumber daya seperti *smartphone* dan perangkat *embedded*. Model ini menggunakan teknik *depthwise separable convolutions* yang memisahkan operasi konvolusi pada ruang dan saluran warna, mengurangi jumlah perhitungan yang diperlukan tanpa kehilangan terlalu banyak akurasi. Fitur inovatif dari *MobileNetV2* adalah blok *inverted residual* dan *linear bottleneck* yang menjaga informasi tetap utuh dalam jaringan meski ukurannya kecil. *MobileNetV2* sangat ideal untuk aplikasi yang memerlukan pengolahan cepat, terutama pada perangkat dengan keterbatasan memori dan kecepatan[15].

*ResNet50*, bagian dari keluarga *Residual Networks*, memiliki pendekatan yang unik dalam menangani tantangan pada jaringan neural yang dalam. Dengan menggunakan *residual blocks*, *ResNet50* memungkinkan jaringan untuk menambah lapisan secara mendalam tanpa menghadapi masalah *vanishing gradient*, yang sering terjadi pada jaringan berlapis banyak. Blok *residual* ini memungkinkan *input* asli untuk diteruskan langsung ke *output* lapisan berikutnya setelah beberapa operasi non-linear. Dengan 50 lapisan, *ResNet50* menawarkan performa yang baik pada berbagai aplikasi tanpa beban memori berlebih, meski dalam hal efisiensi belum setinggi *MobileNetV2*[22].

*InceptionResNetV2* adalah gabungan dari arsitektur *Inception* dan *ResNet*, menggabungkan kekuatan *multi-scale convolutions* dari *Inception* dengan keunggulan *residual connections* dari *ResNet*. Dengan berbagai ukuran konvolusi dalam satu blok, model ini menangkap fitur dalam berbagai skala pada satu tahap konvolusi, memungkinkan identifikasi pola lebih mendetail. Meskipun ukurannya besar dan membutuhkan daya komputasi tinggi, *InceptionResNetV2* mampu memberikan akurasi yang sangat tinggi, terutama pada data yang kompleks. Model ini sering dipilih untuk aplikasi klasifikasi yang menuntut akurasi optimal tanpa batasan sumber daya yang ketat[23].

Pengujian model mencakup beberapa langkah utama. Pertama, melakukan analisis akurasi dan *loss* pada data *training* dan *validation*. Selama pelatihan, akurasi dan *loss* pada data *training* serta *validation* dipantau secara berkala. Hal ini bertujuan untuk memastikan bahwa model tidak mengalami *overfitting* atau *underfitting*. Perbandingan antara akurasi dan *loss* pada data *training* dan *validation* membantu untuk menilai kemampuan model dalam generalisasi, yaitu sejauh mana model dapat mengklasifikasikan motif batik yang berbeda dengan akurat.

Selanjutnya, evaluasi performa pada *dataset* validasi dilakukan menggunakan metrik evaluasi utama seperti *accuracy*, *precision*, *recall*, dan *F1-score*. Metrik-metrik ini memberikan gambaran tentang kemampuan model dalam mengenali pola dan tekstur unik dari masing-masing motif batik, yang penting untuk mengetahui seberapa baik model dalam mengklasifikasikan motif yang belum pernah dilihat sebelumnya, atau kelas yang memiliki performa lebih rendah.

Terakhir, analisis perbandingan antar model dilakukan dengan membandingkan hasil evaluasi setiap model berdasarkan rata-rata metrik untuk semua kelas. Selain itu, visualisasi *confusion matrix* digunakan untuk mengidentifikasi pola kesalahan klasifikasi, seperti motif yang sering salah diprediksi atau kelas yang memiliki performa lebih rendah.

### 3. HASIL DAN PEMBAHASAN

#### 3.1 Hasil Training Model

Pada tabel 2 di bawah ini, bertujuan untuk menyajikan perkembangan akurasi dan *loss* selama pelatihan model *MobileNetV2* pada *dataset* batik. Tabel ini berfungsi sebagai *tools* evaluasi yang menunjukkan seberapa baik model dapat belajar dan beradaptasi dengan data pelatihan serta seberapa baik kemampuan model untuk menggeneralisasi pada data yang belum pernah dilihat (*validasi*). Dengan melihat data pada berbagai *epoch* (20, 40, 60, 80, dan 100), kita dapat menganalisis sejauh mana model mengalami peningkatan kinerja dan apakah ada potensi masalah seperti *overfitting* atau *underfitting*.

**Tabel 2.** Pelatihan menggunakan *MobileNetV2*

No	Epoch	Accuracy		Loss	
		Train	Validation	Train	Validation
1	20	0.9590	0.9715	0.1238	0.0757
2	40	0.9912	0.9876	0.0281	0.0279
3	60	0.9932	0.9913	0.0184	0.0257
4	80	0.9948	0.9895	0.0108	0.0206
5	100	0.9950	0.9885	0.0097	0.0262

Secara lebih detail, akurasi pelatihan yang meningkat dari 0.9590 pada *epoch-20* menjadi 0.9950 pada *epoch* ke-100 menunjukkan bahwa model terus belajar dan meningkatkan kemampuannya dalam mengenali pola-pola batik yang ada dalam *dataset* pelatihan. Akurasi validasi yang juga meningkat dari 0.9715 menjadi 0.9885 menunjukkan bahwa model mampu menggeneralisasi dengan baik, artinya model tidak hanya menghafal pola dari data pelatihan, tetapi juga dapat mengenali pola serupa pada data yang tidak digunakan dalam pelatihan.

Di sisi lain, *loss* yang menurun selama pelatihan, terutama dari 0.1238 pada *epoch-20* menjadi 0.0097 pada *epoch* ke-100 untuk data pelatihan, dan dari 0.0757 menjadi 0.0262 untuk data validasi, memberikan indikasi bahwa model semakin efisien dalam meminimalkan kesalahan prediksi. *Loss* yang lebih rendah menunjukkan bahwa model semakin dekat dengan *output* yang benar dalam klasifikasi motif batik, meskipun ada fluktuasi pada nilai *loss* pada data validasi pada *epoch* ke-100 yang perlu dicermati.

Secara keseluruhan, Tabel 2 memberikan gambaran yang sangat berharga tentang progres pelatihan model *MobileNetV2* dan kinerjanya pada *dataset* batik. Dengan data ini, kita bisa menilai stabilitas pelatihan, mengidentifikasi potensi masalah yang terjadi seperti *overfitting* atau *underfitting*, serta memperkuat pemahaman tentang seberapa baik model ini bekerja dalam mengklasifikasikan motif batik secara akurat.

**Tabel 3.** Pelatihan menggunakan *ResNet50*

No	Epoch	accuracy		Loss	
		Train	Validation	Train	Validation
1	20	0.4083	0.4483	1.2962	1.2523
2	40	0.6910	0.7403	0.7466	0.6564
3	60	0.8386	0.8591	0.4264	0.3864
4	80	0.8826	0.8933	0.3055	0.2981
5	100	0.9105	0.9041	0.2388	0.2523

Tabel 3 menunjukkan hasil pelatihan menggunakan model *ResNet50* dalam klasifikasi motif batik dengan 100 *epoch*. Pada *epoch* pertama (20), akurasi pelatihan masih rendah, yaitu 0.4083, sedangkan akurasi validasi sedikit lebih

rendah di 0.4483. Namun, seiring berjalannya *epoch*, akurasi pelatihan meningkat secara signifikan, mencapai 0.9105 pada *epoch* 100, sedangkan akurasi validasi mencapai 0.9041. Penurunan *loss* juga terlihat dengan stabilitas yang cukup baik, meskipun lebih tinggi dibandingkan dengan *MobileNetV2*. *Loss train* dimulai di angka 1.2962 dan turun menjadi 0.2388, sedangkan *loss validation* mulai dari 1.2523 dan turun menjadi 0.2523. Hasil ini menunjukkan bahwa meskipun performa *ResNet50* cukup baik, *MobileNetV2* lebih unggul dalam hal efisiensi dan kecepatan konvergensi pada *dataset* ini.

Dari segi performa, *ResNet50* menunjukkan peningkatan akurasi yang cukup stabil, dengan perbedaan akurasi pelatihan dan validasi yang tidak terlalu jauh. Hal ini menunjukkan kemampuan generalisasi yang cukup baik, meskipun lebih rendah dibandingkan *MobileNetV2*. *MobileNetV2* lebih cepat mencapai akurasi tinggi, bahkan dengan *dataset* yang terbatas, berkat arsitektur yang ringan dan efisien dalam komputasi. Di sisi lain, *ResNet50* memiliki keunggulan dalam hal kemampuan model untuk menangani kompleksitas data lebih dalam, meskipun membutuhkan lebih banyak sumber daya komputasi dan memori.

Dalam perbandingan antara *MobileNetV2* dan *ResNet50*, *MobileNetV2* lebih unggul dalam hal efisiensi komputasi dan kecepatan konvergensi, yang menjadikannya lebih ideal untuk penggunaan pada perangkat dengan keterbatasan memori dan daya komputasi. *ResNet50*, meskipun lebih lambat dalam mencapai konvergensi optimal, memiliki keunggulan dalam kemampuan generalisasi, yang membuktikan bahwa model ini mampu mengenali pola motif batik dengan baik meskipun akurasi validasinya sedikit lebih rendah. Oleh karena itu, pemilihan antara kedua model ini harus mempertimbangkan kebutuhan aplikasi, seperti efisiensi waktu pelatihan dan sumber daya yang tersedia.

Secara keseluruhan, meskipun *ResNet50* memberikan performa yang baik, *MobileNetV2* memberikan hasil yang lebih efisien dalam waktu pelatihan dengan akurasi yang lebih tinggi. Oleh karena itu, untuk aplikasi pengenalan motif batik yang membutuhkan efisiensi dalam komputasi dan akurasi yang tinggi, *MobileNetV2* adalah pilihan yang lebih baik. Namun, untuk aplikasi yang lebih menuntut kemampuan model untuk menangani data yang lebih kompleks dengan kedalaman jaringan yang lebih besar, *ResNet50* tetap menjadi pilihan yang solid.

**Tabel 4.** Pelatihan menggunakan *InceptionResNetV2*

No	Epoch	accuracy		Loss	
		Train	Validation	Train	Validation
1	20	0.9497	0.9631	0.1525	0.1031
2	40	0.9870	0.9845	0.0325	0.0381
3	60	0.9913	0.9842	0.0172	0.0342

No	Epoch	accuracy		Loss	
		Train	Validation	Train	Validation
4	80	0.9925	0.9863	0.0174	0.0303
5	100	0.9934	0.9891	0.0156	0.025

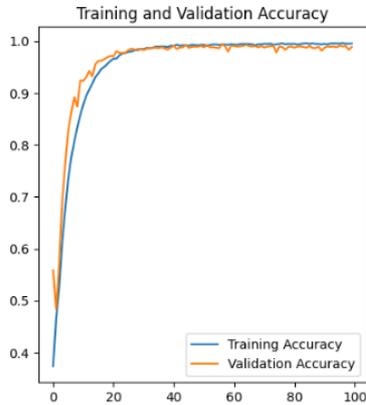
Tabel 4 menunjukkan hasil pelatihan model *InceptionResNetV2* pada *dataset* motif batik dengan 100 *epoch* pelatihan dan validasi. Pada *epoch* ke-20, akurasi pelatihan model adalah 0.9497 dan akurasi validasi 0.9631. Seiring bertambahnya *epoch*, akurasi pelatihan terus meningkat, mencapai 0.9934 pada *epoch* ke-100, sementara akurasi validasi mencapai 0.9891. Hal ini menunjukkan bahwa model *InceptionResNetV2* dapat mengenali pola motif batik dengan sangat baik. Selain itu, nilai *loss* juga menurun dari 0.1525 pada *epoch* pertama hingga 0.0156 pada *epoch* ke-100, yang menunjukkan bahwa model semakin baik dalam meminimalkan kesalahan prediksi.

Dibandingkan dengan *ResNet50*, model *InceptionResNetV2* menunjukkan performa yang lebih baik dalam hal akurasi. Sebagai contoh, pada *epoch* ke-100, *InceptionResNetV2* mencapai akurasi validasi 0.9891, sedangkan *ResNet50* memiliki akurasi validasi di kisaran 0.90-0.94. Hal ini disebabkan oleh perbedaan dalam arsitektur keduanya. *InceptionResNetV2* menggabungkan kekuatan *Inception* yang menggunakan *multi-scale convolutions* dengan *ResNet* yang menggunakan *residual blocks*. Pendekatan ini memungkinkan model untuk menangkap fitur dari berbagai skala dalam satu tahap konvolusi, meningkatkan kemampuan model dalam mengenali pola batik yang lebih kompleks.

Sementara *InceptionResNetV2* menunjukkan akurasi yang sangat tinggi, model ini memerlukan lebih banyak daya komputasi karena ukurannya yang lebih besar dan kompleks. Di sisi lain, *MobileNetV2* yang dirancang untuk efisiensi tinggi, mampu mencapai hasil akurasi yang serupa namun dengan penggunaan sumber daya yang lebih rendah. Pada *epoch* ke-100, *MobileNetV2* mencapai akurasi validasi sekitar 0.98-0.99, yang hampir setara dengan *InceptionResNetV2*. Meskipun demikian, *MobileNetV2* memiliki keunggulan dalam hal kecepatan pelatihan dan penggunaan memori yang lebih efisien, menjadikannya pilihan yang lebih baik untuk aplikasi yang membutuhkan pemrosesan cepat dan penggunaan daya komputasi yang rendah.

Dalam keseluruhan perbandingan, *MobileNetV2* tampil lebih unggul dalam keseimbangan akurasi dan efisiensi, sementara *InceptionResNetV2* menunjukkan performa tinggi dengan daya komputasi yang lebih besar. *ResNet50*, meski berada di peringkat lebih rendah pada akurasi, menawarkan stabilitas yang baik dan dapat dipertimbangkan dalam kondisi *dataset* yang membutuhkan lapisan lebih dalam.

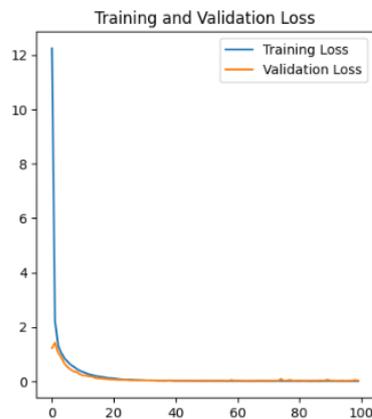
Selain itu, pengujian menggunakan metode GLCM juga dilakukan untuk mengevaluasi fitur tekstur pada citra batik. Metode ini membantu dalam menganalisis pola dan karakteristik tekstur yang ada dalam gambar, sehingga meningkatkan keandalan model dalam mengklasifikasikan berbagai jenis batik.



Gambar 8. Accuracy Train dan Validation pada MobileNetV.

Gambar 8 menunjukkan peningkatan akurasi pada data pelatihan (*training accuracy*) dan data validasi (*validation accuracy*) selama proses pelatihan model *MobileNetV2*. Pada awal pelatihan (*epoch* 0-20), akurasi training dan validasi dimulai dari nilai yang relatif rendah, sekitar 0.4 hingga 0.5. Hal ini wajar karena model berada dalam tahap awal pembelajaran, sehingga belum sepenuhnya memahami pola dalam data.

Seiring bertambahnya *epoch*, tepatnya pada periode 20 hingga 60 *epoch*, akurasi model mengalami peningkatan yang signifikan. Pada tahap ini, model menunjukkan kemampuan belajar yang baik tanpa mengalami *overfitting*, karena akurasi validasi tetap stabil dan hampir sejajar dengan akurasi pelatihan. Pada akhir pelatihan (*epoch* 60-100), akurasi *training* dan validasi mendekati nilai sempurna, yaitu 1.0. Hasil ini menunjukkan bahwa model telah belajar dengan optimal dan mampu mengenali pola dengan baik pada data validasi.



Gambar 9. Loss Train dan Validation pada MobileNetV2

Gambar 9 menggambarkan penurunan *loss* untuk data pelatihan (*training loss*) dan validasi (*validation loss*)

selama proses pelatihan. Pada awal pelatihan (*epoch* 0-20), nilai *loss* cukup tinggi, yang menunjukkan bahwa prediksi model masih memiliki banyak kesalahan. Namun, pada periode pertengahan pelatihan (*epoch* 20-60), *loss training* dan validasi mengalami penurunan yang signifikan. Hal ini mengindikasikan bahwa model secara efektif mempelajari pola data pelatihan dan mampu menerapkannya dengan baik pada data validasi.

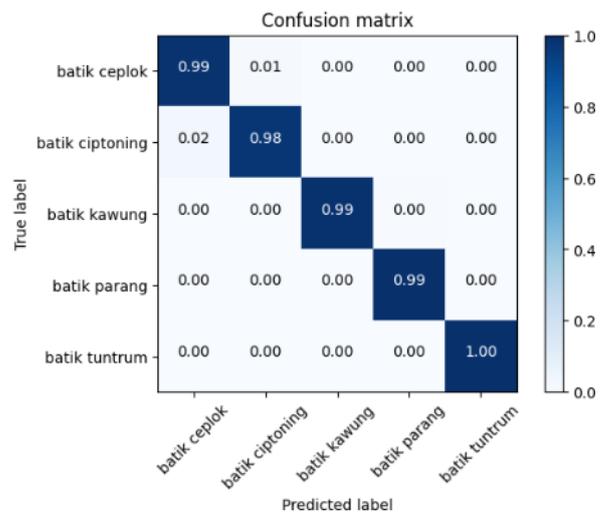
Pada akhir pelatihan (*epoch* 60-100), nilai *loss training* dan validasi mendekati nol, menunjukkan bahwa model telah berhasil mengurangi kesalahan prediksi secara drastis. Perbedaan antara *loss training* dan validasi tetap kecil, yang menjadi indikasi bahwa model memiliki kemampuan generalisasi yang baik tanpa *overfitting* terhadap data pelatihan.

### 3.2 Evaluasi Model

Berikut adalah hasil evaluasi dari tiga model *transfer learning* yaitu *MobileNetV2*, *ResNet50*, dan *InceptionResNetV2* dalam klasifikasi motif batik. Ketiga model ini digunakan untuk melihat performa dan efektivitasnya dalam mengenali pola motif batik dengan tepat, berdasarkan hasil analisis *Classification Report* dan *Confusion Matrix* untuk masing-masing model.

	precision	recall	f1-score	support
batik ceplok	0.97	0.99	0.98	694
batik ciptoning	0.99	0.98	0.98	724
batik kawung	1.00	0.99	0.99	615
batik parang	0.99	0.99	0.99	598
batik tuntrum	1.00	1.00	1.00	592
accuracy			0.99	3223
macro avg	0.99	0.99	0.99	3223
weighted avg	0.99	0.99	0.99	3223

Gambar 10. Classification Report MobileNetV2



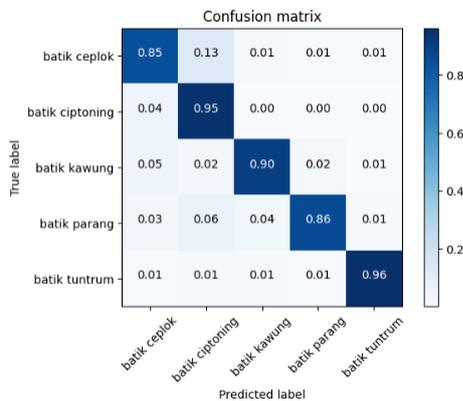
Gambar 11. Confusion Matrix MobileNetV2

Pada gambar 10, hasil evaluasi model *MobileNetV2* menunjukkan performa tinggi dengan hampir semua kelas mencapai *precision*, *recall*, dan *f1-score* mendekati 0.99.

Model ini berhasil mencapai akurasi keseluruhan sebesar 99%, yang mengindikasikan bahwa model mampu mengenali pola batik secara akurat. *Confusion Matrix MobileNetV2* di gambar 11, juga memperlihatkan bahwa sebagian besar pola batik terklasifikasi dengan benar, walaupun terdapat sedikit kesalahan klasifikasi pada motif Ciptoning yang terkadang diprediksi sebagai motif Ceplok. Meskipun demikian, performa *MobileNetV2* sangat baik, terutama dalam mempertahankan akurasi tinggi pada model yang efisien dalam hal kecepatan dan pemakaian memori.

	precision	recall	f1-score	support
batik ceplok	0.88	0.85	0.86	694
batik ciptoning	0.82	0.95	0.88	724
batik kawung	0.94	0.90	0.92	615
batik parang	0.96	0.86	0.90	598
batik tuntrum	0.97	0.96	0.96	592
accuracy			0.90	3223
macro avg	0.91	0.90	0.91	3223
weighted avg	0.91	0.90	0.90	3223

Gambar 12. Classification Report ResNet50



Gambar 13. Confusion Matrix ResNet50

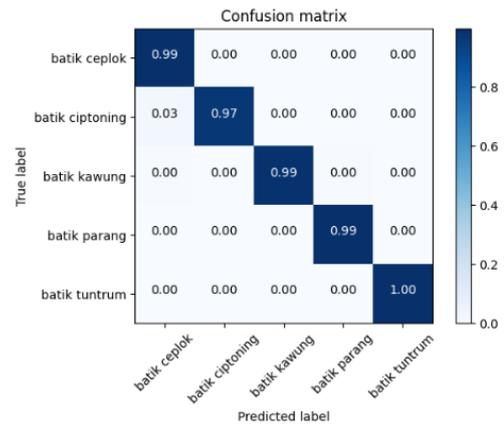
Seperti yang ada di gambar 12, evaluasi model *ResNet50* juga memberikan hasil yang cukup baik, meskipun sedikit di bawah *MobileNetV2* dalam beberapa aspek. *ResNet50* mencapai *accuracy* sebesar 90%, dengan *precision* dan *recall* rata-rata untuk masing-masing kelas 0.90. Beberapa motif, seperti Ciptoning, mengalami sedikit penurunan dalam *recall* dan *precision* dibandingkan dengan *MobileNetV2*, yang menunjukkan bahwa model ini lebih rentan terhadap kesalahan klasifikasi dalam mendeteksi beberapa pola motif.

Namun, terlihat pada gambar 13, terdapat beberapa kesalahan klasifikasi. Misalnya, motif batik Ceplok salah diklasifikasikan sebagai motif Ciptoning sekitar 13% dari kasus, dan motif batik Parang terkadang diklasifikasikan sebagai batik Kawung. Kesalahan ini menunjukkan bahwa pola tertentu memiliki kemiripan, sehingga model sesekali salah mengelompokkannya. Meski demikian, *ResNet50* mampu menangani klasifikasi yang kompleks dengan menggunakan *residual connections* yang mencegah

degradasi *accuracy* pada model yang dalam, sehingga tetap cocok untuk *dataset* yang kompleks seperti motif batik ini.

	precision	recall	f1-score	support
batik ceplok	0.97	0.99	0.98	694
batik ciptoning	1.00	0.97	0.98	724
batik kawung	0.99	0.99	0.99	615
batik parang	0.99	0.99	0.99	598
batik tuntrum	1.00	1.00	1.00	592
accuracy			0.99	3223
macro avg	0.99	0.99	0.99	3223
weighted avg	0.99	0.99	0.99	3223

Gambar 14. Classification Report InceptionResNet2



Gambar 15. Confusion Matrix InceptionResNet2

Sementara itu pada gambar 14, hasil model *InceptionResNetV2* menampilkan performa yang mirip dengan *MobileNetV2*, dengan akurasi sebesar 99% serta *precision*, *recall*, dan *f1-score* yang konsisten pada setiap kelas motif batik. Model ini menunjukkan kekuatan dalam mendeteksi pola-pola yang kompleks berkat kombinasi dari *multi-scale convolutions* dalam arsitektur *Inception* dan *residual connections* dari *ResNet*. Dalam gambar 15, yang menampilkan *Confusion Matrix*, model ini menunjukkan sedikit sekali kesalahan prediksi, dengan motif Ciptoning terkadang diprediksi sebagai Ceplok, tetapi secara keseluruhan model ini menunjukkan kemampuan yang unggul untuk mengklasifikasikan pola batik dengan akurasi tinggi.

Secara keseluruhan, ketiga model menunjukkan hasil yang sangat baik dengan akurasi tinggi, namun *MobileNetV2* dan *InceptionResNetV2* memiliki kinerja yang lebih unggul dalam klasifikasi motif batik dibandingkan *ResNet50*. *MobileNetV2* juga lebih unggul dalam hal efisiensi, cocok untuk aplikasi berbasis perangkat *mobile*, sementara *InceptionResNetV2* menawarkan akurasi yang tinggi dengan kapasitas model yang besar, sehingga cocok untuk kebutuhan yang menuntut ketelitian dan sumber daya yang cukup besar.

#### 4. KESIMPULAN

Hasilnya penelitian ini berhasil mengembangkan model klasifikasi motif batik Yogyakarta menggunakan kombinasi

metode *Gray Level Co-occurrence Matrix* (GLCM) untuk ekstraksi fitur tekstur dan *Convolutional Neural Network* (CNN) berbasis *MobileNetV2* untuk klasifikasi citra. *Dataset* yang digunakan terdiri dari 3.223 gambar yang mencakup lima kategori motif batik khas Yogyakarta, yaitu Batik Ceplok, Batik Kawung, Batik Truntum, Batik Parang, dan Batik Ciptoning. Hasil pengujian menunjukkan bahwa model mencapai akurasi tinggi sebesar 99%, menegaskan efektivitas pendekatan ini dalam mengenali pola motif batik yang kompleks.

Dengan menggabungkan GLCM dan CNN berbasis *MobileNetV2*, penelitian ini memberikan hasil klasifikasi yang lebih optimal dalam mengenali motif batik Yogyakarta yang memiliki keanekaragaman pola tekstur yang kompleks. GLCM berperan penting dalam mengekstraksi informasi tekstur yang mendetail, sementara *MobileNetV2*, sebagai arsitektur CNN yang efisien dan ringan, memungkinkan pengenalan pola visual dengan presisi tinggi tanpa memerlukan sumber daya komputasi yang besar. Pendekatan ini tidak hanya meningkatkan akurasi klasifikasi dibandingkan penelitian-penelitian sebelumnya, tetapi juga menawarkan solusi yang efisien dan akurat untuk sistem klasifikasi motif batik. Selain itu, model ini direncanakan akan diimplementasikan ke dalam aplikasi *mobile* dengan arsitektur *client-server*, sehingga mendukung penggunaan praktis dalam mendeteksi motif batik secara otomatis di berbagai perangkat *mobile*.

Meskipun hasil yang dicapai sangat baik, penelitian ini terbatas pada ukuran *dataset* dan kompleksitas motif batik tertentu. Untuk penelitian selanjutnya, perlu dilakukan perluasan *dataset* dengan menambahkan lebih banyak variasi motif dan meningkatkan keragamannya. Selain itu, eksplorasi metode augmentasi data yang lebih beragam dan pengembangan arsitektur model yang lebih canggih diperlukan untuk meningkatkan akurasi pada motif batik dengan kompleksitas yang lebih tinggi.

#### DAFTAR PUSTAKA

- [1] D. Nur Ramadhan, R. Aldi Erwanto, and R. Tan Enwan, "Klasifikasi Batik Menggunakan Algoritma CNN (Convolutional Neural Network)," Kediri, 2024.
- [2] A. Prayoga, Maimunah, P. Sukmasetya, Muhammad Resa Arif Yudianto, and Rofi Abul Hasani, "Arsitektur Convolutional Neural Network untuk Model Klasifikasi Citra Batik Yogyakarta," *Journal of Applied Computer Science and Technology*, vol. 4, no. 2, pp. 82–89, Nov. 2023, doi: 10.52158/jacost.v4i2.486.
- [3] M. Taufik Ma'ruf, ); Erwin, D. Putra, Y. Reswan, and U. Juhardi, "Klasifikasi Motif Kain Batik Besurek Menggunakan Ekstraksi Ciri Grey Level Co-Occurrence Matrix (GLCM)," *JURNAL KOMITEK*, vol. 3, no. 2, pp. 229–236, 2023, doi: 10.53697/jkomitek.v3i2.
- [4] R. Widodo *et al.*, "Pemanfaatan Ciri Gray Level Co-Occurrence Matrix (GLCM) Citra Buah Jeruk Keprok (*Citrus reticulata* Blanco) untuk Klasifikasi Mutu," 2018. [Online]. Available: <http://j-ptiik.ub.ac.id>
- [5] K. Dong, C. Zhou, Y. Ruan, and Y. Li, "MobileNetV2 Model for Image Classification," in *Proceedings - 2020 2nd International Conference on Information Technology and Computer Application, ITCA 2020*, Institute of Electrical and Electronics Engineers Inc., Dec. 2020, pp. 476–480. doi: 10.1109/ITCA52113.2020.00106.
- [6] Y. Gulzar, "Fruit Image Classification Model Based on MobileNetV2 with Deep Transfer Learning Technique," *Sustainability (Switzerland)*, vol. 15, no. 3, Feb. 2023, doi: 10.3390/su15031906.
- [7] Bagus Untung Saputra, Gunawan, and Wresti Andriani, "PENGENALAN MOTIF BATIK PESISIR PULAU JAWA MENGGUNAKAN CONVOLUTIONAL NEURAL NETWORK," *NUANSA INFORMATIKA*, vol. 17, no. 2, pp. 119–125, Jul. 2023, doi: 10.25134/ilkom.v17i2.32.
- [8] Fathul Am and E. I. Sela, "Klasifikasi Batik Pekalongan Berdasarkan Citra dengan Metode GLCM dan JST Backpropagation," *Jurnal Indonesia : Manajemen Informatika dan Komunikasi*, vol. 5, no. 1, pp. 614–621, Jan. 2024, doi: 10.35870/jimik.v5i1.532.
- [9] A. A. Kasim, R. Wardoyo, and A. Harjoko, "Feature extraction methods for batik pattern recognition: A review," in *AIP Conference Proceedings*, American Institute of Physics Inc., Jul. 2016. doi: 10.1063/1.4958503.
- [10] R. Widodo *et al.*, "Pemanfaatan Ciri Gray Level Co-Occurrence Matrix (GLCM) Citra Buah Jeruk Keprok (*Citrus reticulata* Blanco) untuk Klasifikasi Mutu," 2018. [Online]. Available: <http://j-ptiik.ub.ac.id>
- [11] N. D. Girsang, "Literature Study of Convolutional Neural Network Algorithm for Batik Classification," vol. 1, no. 1, 2021, doi: 10.47709/briliance.v1i1.1069.
- [12] R. Moyazzoma, M. A. A. Hossain, M. H. Anuz, and A. Sattar, "Transfer Learning Approach for Plant Leaf Disease Detection Using CNN with Pre-Trained Feature Extraction Method Mobilnetv2," in *International Conference on Robotics, Electrical and Signal Processing Techniques*, 2021, pp. 526–529. doi: 10.1109/ICREST51555.2021.9331214.
- [13] M. Iman, H. R. Arabnia, and K. Rasheed, "A Review of Deep Transfer Learning and Recent

- Advancements,” Apr. 01, 2023, *MDPI*. doi: 10.3390/technologies11020040.
- [14] Y. Zhou, S. Chen, Y. Wang, and W. Huan, “Review of research on lightweight convolutional neural networks,” 2020.
- [15] P. K. Das, V. A. Diya, S. Meher, R. Panda, and A. Abraham, “A Systematic Review on Recent Advancements in Deep and Machine Learning Based Detection and Classification of Acute Lymphoblastic Leukemia,” 2022, *Institute of Electrical and Electronics Engineers Inc.* doi: 10.1109/ACCESS.2022.3196037.
- [16] A. Dewi, I. Christanti, F. Y. Sari, E. Pramita, and P. Matematika, “ETNOMATEMATIKA PADA BATIK KAWUNG YOGYAKARTA DALAM TRANSFORMASI GEOMETRI,” 2020.
- [17] Medina Diyah Kusumawati and Endang Ruswanti Hartowiyono, “Philosophy, Design Batik Yogyakarta, and Batik Surakarta Made in Indonesia,” *International Journal of Latest Trends in Engineering and Technology*.
- [18] N. Regine, H. Laswandi, and S. Meliana, “DEVELOPMENT OF BATIK PATTERNS AT THE NATIONAL LIBRARY OF INDONESIA WITH THE STYLIZATION METHOD,” *International Journal of Application on Social Science and Humanities*, vol. 1, no. 1, pp. 1782–1790, 2023, doi: 10.24912/ijassh.v1i2.27925.
- [19] D. Tsalsabila Rhamadiyahanti, “Analisa Performa Convolutional Neural Network dalam Klasifikasi Citra Apel dengan Data Augmentasi,” *Media Online*, vol. 5, no. 1, pp. 154–162, 2024, doi: 10.30865/klik.v5i1.2023.
- [20] M. Toyib, T. Decky, and K. Pratama, “Penerapan Algoritma CNN Untuk Mendeteksi Tulisan Tangan Angka Romawi dengan Augmentasi Data,” *Kebumian dan Angkasa*, vol. 2, no. 3, pp. 108–120, 2024, doi: 10.62383/algoritma.v2i3.69.
- [21] F. Roberti de Siqueira, W. Robson Schwartz, and H. Pedrini, “Multi-scale gray level co-occurrence matrices for texture description,” *Neurocomputing*, vol. 120, pp. 336–345, Nov. 2013, doi: 10.1016/j.neucom.2012.09.042.
- [22] M. B. Hossain, S. M. H. S. Iqbal, M. M. Islam, M. N. Akhtar, and I. H. Sarker, “Transfer learning with fine-tuned deep CNN ResNet50 model for classifying COVID-19 from chest X-ray images,” *Inform Med Unlocked*, vol. 30, Jan. 2022, doi: 10.1016/j.imu.2022.100916.
- [23] T. Garg, M. Garg, O. P. Mahela, and A. R. Garg, “Convolutional Neural Networks with Transfer Learning for Recognition of COVID-19: A Comparative Study of Different Approaches,” *AI (Switzerland)*, vol. 1, no. 4, pp. 586–606, Dec. 2020, doi: 10.3390/ai1040034.