

PENGEMBANGAN APLIKASI ANDROID UNTUK DETEKSI DAN KLASIFIKASI MOTIF BATIK BERBASIS *CONVOLUTIONAL NEURAL NETWORK (CNN)*

1st Muhammad Haikal Fikri As'ad
Program Studi Informatika,
Universitas Telkom, Kampus Surabaya,
Surabaya, Indonesia
haikalasad@student.telkomuniversity.ac.id

2nd Ardian Yusuf Wicaksono
Program Studi Informatika,
Universitas Telkom, Kampus Surabaya,
Surabaya, Indonesia
ardianyw@telkomuniversity.ac.id

3rd Daud Muhamajir
Program Studi Informatika,
Universitas Telkom, Kampus Surabaya,
Surabaya, Indonesia
daudmuhamajir@telkomuniversity.ac.id

Abstrak — Batik merupakan warisan budaya Indonesia yang kaya akan motif dan nilai filosofis. Namun, masyarakat masih mengalami kesulitan dalam mengenali motif batik akibat kompleksitas visual dan keterbatasan akses informasi. Penelitian ini bertujuan mengembangkan aplikasi Android bernama Batikara yang mampu mendeteksi dan mengklasifikasikan motif batik secara otomatis menggunakan metode *Convolutional Neural Network (CNN)* dengan arsitektur MobileNetV2, serta memberikan informasi edukatif melalui artikel batik. Model klasifikasi MobileNetV2 memperoleh akurasi tertinggi sebesar 92%, diikuti oleh DenseNet121 dengan akurasi 90%, serta ResNet50 dengan akurasi 77%. MobileNetV2 menunjukkan performa yang paling stabil dan efisien untuk implementasi pada perangkat *mobile*, serta memiliki kapabilitas yang unggul dalam mengklasifikasikan motif batik yang memiliki kemiripan visual dan pola yang kompleks. Sebaliknya, ResNet50 cenderung kurang optimal dalam membedakan motif-motif batik yang serupa secara visual. Untuk deteksi objek, model SSD MobileNetV2 mencatat nilai *Average Precision (AP)* tertinggi 0,722 pada IoU 0,50, meskipun performa menurun pada objek kecil. Evaluasi usability melalui System Usability Scale (SUS), black box testing, dan task scenario menghasilkan skor SUS sebesar 83,17, efektivitas 88,9%, dan efisiensi 77,16%. Responden, termasuk pelaku UMKM, menilai fitur pemindaian sangat membantu dan edukatif. Aplikasi Batikara berpotensi menjadi sarana pelestarian budaya sekaligus memberdayakan UMKM batik melalui teknologi digital.

Kata kunci— Aplikasi Android, Batik, CNN, Klasifikasi Citra, MobileNetV2, TensorFlow Lite

I. PENDAHULUAN

Batik merupakan salah satu warisan budaya Indonesia yang telah diakui dunia, bahkan oleh UNESCO sebagai warisan budaya tak benda pada tahun 2009 [1]. Kain batik memiliki keunikan tersendiri dalam motif dan coraknya, dengan setiap daerah di Indonesia memiliki motif batik khas yang melambangkan nilai-nilai dan kearifan lokal masing-masing. Motif batik ini sangat beragam, mulai dari yang sederhana hingga yang kompleks, seringkali dengan makna filosofis yang mendalam.

Namun, seiring dengan perkembangan zaman, pengenalan dan pemahaman terhadap ragam motif batik semakin sulit dilakukan oleh masyarakat umum [2]. Banyak orang kesulitan dalam mengidentifikasi motif batik, terlebih apabila motif tersebut berasal dari daerah yang kurang dikenal. Selain itu, pengaruh budaya global turut menggerus nilai-nilai luhur yang terkandung dalam batik, khususnya di kalangan masyarakat Jawa, yang menjadi penyanga utama batik [3].

Jumlah motif batik di Indonesia sangatlah banyak, dengan lebih dari 5.849 varian dari berbagai daerah menurut Bandung Fe Institute (2015). Setiap motif memiliki bentuk, nama, dan makna yang khas, mencerminkan kekayaan budaya lokal. Namun, keragaman ini justru menjadi tantangan dalam proses identifikasi, terutama bagi masyarakat awam yang minim pengetahuan tentang citra visual batik. Permasalahan ini diperparah oleh keterbatasan media informasi yang dapat diakses secara praktis dan cepat untuk mempelajari batik secara menyeluruh [4]. Akibatnya, upaya pelestarian batik sebagai warisan budaya menjadi terhambat karena kurangnya pemahaman masyarakat terhadap keanekaragaman motif yang ada.

Untuk mengatasi permasalahan tersebut, teknologi dapat memainkan peran yang sangat penting. Salah satu solusi yang relevan adalah pengembangan aplikasi berbasis Android yang dapat mendeteksi dan mengklasifikasikan motif batik secara otomatis menggunakan teknologi *Convolutional Neural Network (CNN)*. Platform Android dipilih karena dominasi penggunaannya di Indonesia. Berdasarkan data dari [5], pada September 2024, 87,54% pengguna *smartphone* di Indonesia menggunakan sistem operasi Android. Dengan menggunakan platform ini, aplikasi dapat menjangkau lebih banyak pengguna dan berpotensi lebih efektif dalam membantu pelestarian budaya.

Penelitian-penelitian sebelumnya telah menunjukkan bahwa CNN dapat digunakan untuk klasifikasi gambar dengan akurasi yang baik, seperti yang ditunjukkan dalam penelitian oleh [6], [7]. Oleh karena itu, penelitian ini

bertujuan untuk mengembangkan aplikasi Android yang dapat mendeteksi dan mengklasifikasikan motif batik menggunakan CNN, dengan harapan dapat mempermudah masyarakat dalam mengenali dan memahami motif batik, serta mendukung pelestarian warisan budaya batik Indonesia.

Teknologi, khususnya dalam bentuk aplikasi berbasis Android, telah terbukti menjadi alat yang efektif dalam mempermudah akses informasi mengenai budaya lokal. Penelitian oleh [8] mengenai aplikasi pengenalan budaya Sumba menunjukkan bahwa aplikasi semacam ini dapat membantu masyarakat mempelajari budaya lokal dengan lebih mudah dan efisien. Hasil penelitian tersebut juga mengungkapkan bahwa pengembangan aplikasi berbasis Android merupakan langkah yang efektif dalam meningkatkan minat dan pengetahuan pengguna, terutama di kalangan generasi muda, terhadap kekayaan budaya local.

Selain penggunaan arsitektur *Convolutional Neural Network* (CNN), pemanfaatan teknologi on-device melalui TensorFlow Lite (TF Lite) memberikan nilai tambah yang signifikan dalam pengembangan aplikasi klasifikasi motif batik. Teknologi ini memungkinkan proses deteksi dan klasifikasi dilakukan langsung di perangkat tanpa perlu terhubung ke internet, sehingga tetap dapat digunakan secara optimal di wilayah dengan keterbatasan jaringan. Hal ini menjadi penting mengingat masih banyak daerah di Indonesia yang memiliki akses internet terbatas..

Dengan pendekatan ini, pengguna dapat mengakses aplikasi kapan saja dan di mana saja, tanpa bergantung pada koneksi eksternal. Selain meningkatkan fleksibilitas, penggunaan model secara on-device juga mengurangi latensi dan mempercepat respons sistem, sehingga hasil klasifikasi dapat ditampilkan secara *real-time*. Penelitian oleh [9] mendukung efektivitas pendekatan ini, di mana aplikasi CENTING yang dikembangkan menggunakan TF Lite terbukti mendukung deteksi mandiri, akses offline, serta edukasi pencegahan, dengan tingkat penerimaan pengguna mencapai 89,5%. Temuan tersebut menunjukkan bahwa teknologi on-device memiliki potensi besar untuk diterapkan pada aplikasi lain, termasuk dalam konteks pelestarian budaya dan edukasi digital berbasis citra seperti motif batik.

Aplikasi berbasis Android ini diharapkan dapat menjadi solusi yang relevan dan strategis dalam pelestarian batik sebagai warisan budaya Indonesia. Tidak hanya mempermudah masyarakat dalam mengenali dan memahami motif batik, aplikasi ini juga dapat berfungsi sebagai alat untuk meningkatkan kesadaran masyarakat terhadap pentingnya melestarikan batik. Hal ini akan mendukung pelestarian budaya batik, serta meningkatkan pemahaman dan apresiasi terhadap identitas budaya Indonesia.

II. KAJIAN TEORI

Penelitian sebelumnya menunjukkan bahwa metode *Convolutional Neural Network* (CNN) memiliki potensi tinggi dalam tugas klasifikasi dan deteksi citra, termasuk pada pengenalan motif batik. Penelitian oleh [6] mengkaji penggunaan arsitektur EfficientNet-B0 untuk mengklasifikasikan 15 kelas motif batik dan memperoleh akurasi yang cukup tinggi, meskipun implementasinya belum mencakup proses deteksi dan tidak diintegrasikan ke dalam

platform Android. Sementara itu, penelitian oleh [2] berhasil mengembangkan aplikasi Android untuk klasifikasi motif batik menggunakan MobileNet dengan akurasi di atas 70%, namun terbatas hanya pada empat kelas dan dengan antarmuka pengguna yang kurang optimal. Penelitian lain oleh [10] serta [11], meskipun fokus objeknya berbeda, menunjukkan bahwa implementasi model CNN ke dalam aplikasi Android memungkinkan dilakukan secara efisien menggunakan TensorFlow. Di sisi lain, penelitian oleh [12] membandingkan performa CNN dan SVM dalam mendeteksi katarak, dengan hasil bahwa CNN menunjukkan akurasi yang lebih tinggi. Sementara itu, pendekatan konvensional seperti *Gray Level Co-Occurrence Matrix* (GLCM) dan *K-Nearest Neighbors* (KNN) dalam penelitian oleh [13] juga digunakan untuk mengenali motif batik, namun dinilai kurang relevan untuk data dengan jumlah kelas yang kompleks.

A. Batik

Batik merupakan seni kriya yang telah berkembang di berbagai wilayah Indonesia sejak masa lampau. Secara etimologis, kata "batik" berasal dari bahasa Jawa, yaitu amba (kain) dan titik (titik), yang menggambarkan proses membentuk pola pada kain melalui titik-titik tertentu [14]. Selain nilai estetika, batik mengandung simbol, filosofi, dan makna budaya yang mencerminkan dinamika masyarakat dari zaman kerajaan hingga era globalisasi [15], [16]. Pada 2 Oktober 2009, batik Indonesia diakui oleh UNESCO sebagai Warisan Budaya Takbenda Dunia [17]. Pengakuan ini menjadi dorongan bagi berbagai pihak untuk terus melestarikan batik secara konkret. Namun demikian, pemahaman masyarakat terhadap batik masih terbatas pada aspek pemakaian, padahal batik memiliki potensi lebih luas dalam aspek sejarah, filosofi, budaya, hingga pariwisata [16], [18].

B. Image Processing

Image processing adalah tahap awal penting dalam sistem *computer vision* untuk mempersiapkan citra sebelum diproses oleh model *deep learning* seperti CNN. Proses ini mencakup augmentasi data untuk memperkaya variasi dataset dan mengurangi *overfitting*, serta *preprocessing* seperti normalisasi dan *resize* citra agar sesuai dengan format input model. Teknik-teknik ini bertujuan meningkatkan kualitas, konsistensi, dan efektivitas data citra, sehingga mendukung pelatihan model yang lebih stabil, efisien, dan akurat [19].

C. Deep Learning dan Machine Learning

Machine Learning merupakan cabang dari *Artificial Intelligence* yang memungkinkan komputer belajar dari data tanpa pemrograman eksplisit. Pendekatan ini mencakup metode seperti *supervised*, *unsupervised*, dan *reinforcement learning*. *Deep Learning*, sebagai bagian dari Machine Learning, menggunakan jaringan saraf tiruan bertingkat (*deep neural networks*) untuk memproses data kompleks, khususnya data tidak terstruktur seperti gambar dan teks [20]. Berkat arsitektur berlapis, model seperti *Convolutional Neural Network* (CNN) mampu mengenali pola yang kompleks dan telah menunjukkan kinerja tinggi dalam tugas pengenalan gambar, klasifikasi objek, pemrosesan bahasa alami, serta segmentasi visual.

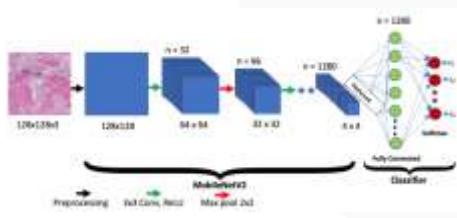
D. Convolutional Neural Network

Convolutional Neural Network (CNN) merupakan jenis jaringan saraf tiruan yang secara khusus dirancang untuk

mengenali dan mengolah data berdimensi grid seperti gambar. CNN memiliki kemampuan untuk mengekstraksi fitur secara otomatis melalui operasi konvolusi, sehingga cocok digunakan dalam tugas-tugas seperti klasifikasi citra. Struktur utama CNN terdiri dari tiga jenis lapisan: lapisan konvolusi, lapisan *pooling*, dan lapisan *fully connected*. Lapisan konvolusi berfungsi mengekstraksi fitur dari gambar input dengan menggunakan filter atau kernel yang menghasilkan *feature map*, di mana pola-pola sederhana seperti tepi atau tekstur dapat dikenali dan dikembangkan menjadi pola yang lebih kompleks pada lapisan-lapisan selanjutnya. Selanjutnya, lapisan *pooling* berperan untuk mereduksi dimensi *feature map*, menjaga fitur penting, sekaligus mengurangi beban komputasi jaringan. Dua teknik pooling yang umum digunakan adalah *Max Pooling*, yang mengambil nilai maksimum dari area tertentu, dan *Average Pooling*, yang menghitung nilai rata-rata. Reduksi ini membantu meningkatkan efisiensi model tanpa menghilangkan karakteristik penting. Terakhir, lapisan *fully connected* bertugas menghubungkan semua neuron dari lapisan sebelumnya untuk menggabungkan semua fitur yang telah diekstraksi. Lapisan ini berperan penting dalam proses klasifikasi akhir, karena menghasilkan *output* berdasarkan seluruh informasi fitur yang diperoleh dari proses sebelumnya [21], [22].

E. MobileNetV2

MobileNetV2 adalah versi lanjutan dari MobileNet yang dikembangkan oleh Google untuk meningkatkan efisiensi model pada perangkat *mobile* dengan keterbatasan daya dan kapasitas komputasi. Arsitektur ini memperkenalkan konsep inverted residuals dan linear *bottlenecks*, di mana lapisan input dan output memiliki dimensi rendah sementara lapisan tengah diekspansi secara temporer, sehingga memungkinkan aliran informasi yang lebih optimal. Dengan tetap menggunakan *depthwise separable convolutions*, MobileNetV2 berhasil mengurangi jumlah parameter tanpa mengorbankan akurasi, menjadikannya pilihan populer untuk aplikasi *real-time* pada perangkat *mobile* [23]. Arsitektur MobileNetV2 dapat dilihat pada gambar 1.

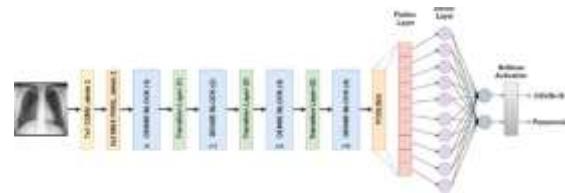


GAMBAR 1
(Architecture MobileNetV2)

Sumber (<https://www.analyticsvidhya.com/blog/2023/12/what-is-mobilenetv2/>)

F. DenseNet121

DenseNet121 merupakan varian dari arsitektur CNN yang mengusung konsep *dense connectivity*, di mana setiap lapisan terhubung langsung dengan seluruh lapisan setelahnya dalam satu blok. Pendekatan ini memungkinkan aliran informasi dan gradien yang lebih efisien, membantu mengatasi masalah *vanishing gradient* dan meningkatkan efisiensi parameter. Dengan total 121 lapisan, DenseNet121 memiliki struktur yang ringan namun tetap mampu memberikan performa tinggi dalam tugas klasifikasi maupun segmentasi citra, menjadikannya pilihan tepat untuk aplikasi *computer vision* yang membutuhkan akurasi tinggi dan efisiensi komputasi [24]. Arsitektur DenseNet121 dapat dilihat pada gambar 2.

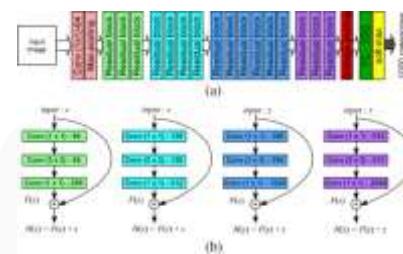


GAMBAR 2
(Architecture DenseNet121)

Sumber (https://www.researchgate.net/figure/DenseNet121-architecture-designed-for-binary-classification_fig11_353494880)

G. ResNet50

ResNet50 (*Residual Neural Network*) adalah arsitektur CNN yang dirancang untuk mengatasi masalah *vanishing gradient* pada jaringan yang sangat dalam dengan memperkenalkan *residual block*, yaitu mekanisme *shortcut connection* antar lapisan. Salah satu varian yang paling banyak digunakan adalah ResNet50, yang terdiri dari 50 lapisan dan menggabungkan konvolusi, *batch normalization*, ReLU, serta *shortcut connection* untuk menjaga kestabilan pelatihan. Arsitektur ini memungkinkan ekstraksi fitur yang kompleks secara efisien dan menjadi pilihan populer dalam berbagai aplikasi *computer vision* karena mampu memberikan performa tinggi dengan efisiensi komputasi yang baik [25]. Arsitektur ResNet50 dapat dilihat pada gambar 3.

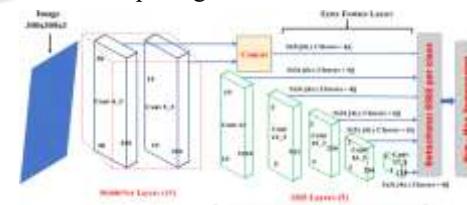


GAMBAR 3
(Architecture ResNet50)

Sumber (https://www.researchgate.net/figure/DenseNet121-architecture-designed-for-binary-classification_fig11_353494880)

H. Single Shot MultiBox Detector (SSD)

Single Shot MultiBox Detector (SSD) adalah algoritma deteksi objek yang cepat dan efisien, sehingga ideal untuk aplikasi *real-time* seperti pemrosesan video. SSD memanfaatkan CNN untuk mendeteksi dan memberi *bounding box* pada berbagai objek dalam satu citra, serta bekerja pada berbagai skala untuk mendeteksi objek kecil hingga besar [26]. Dalam penelitian ini, SSD digunakan untuk mendeteksi area bermotif batik dalam citra, yang kemudian diekstraksi dan diklasifikasikan lebih lanjut untuk mengidentifikasi jenis motif batik secara spesifik. Arsitektur SSD MobileNetV2 pada gambar 4.



GAMBAR 4
(Architecture SSD MobileNetV2)

Sumber (https://www.researchgate.net/figure/Optimized-MobileNet-SSD-backbone-network-through-Conv4-3-layer-proposed-method_fig8_353306246)

I. Tensorflow Lite

TensorFlow Lite adalah *framework open source* dari Google yang dirancang untuk menjalankan model machine learning secara efisien di perangkat dengan sumber daya terbatas, seperti Android. *Framework* ini mengonversi model TensorFlow ke format *.tflite* yang lebih ringan, memungkinkan inferensi lokal tanpa internet, sehingga meningkatkan privasi dan kecepatan pemrosesan. Dengan dukungan fitur seperti *ML Model Binding*, pengembang dapat dengan mudah mengintegrasikan model ke dalam aplikasi untuk deteksi dan klasifikasi secara langsung di perangkat [9], [27].

J. Android dan Android Studio

Android merupakan sistem operasi populer untuk perangkat seluler yang dikembangkan oleh Google, memungkinkan pengembangan aplikasi untuk berbagai keperluan seperti bisnis, layanan, dan penyediaan konten, dengan keunggulan jangkauan pengguna yang luas [28]. Untuk mendukung pengembangan ini, digunakan Android Studio sebagai IDE utama yang berbasis *IntelliJ IDEA*. Android Studio menyediakan fitur lengkap seperti editor kode, sistem *build Gradle*, *emulator*, *Instant Run*, serta integrasi dengan GitHub dan layanan Google Cloud, sehingga mempermudah proses pengembangan dan pengujian aplikasi Android [28], [29].

K. System Usability Scale (SUS)

System Usability Scale (SUS) adalah metode cepat untuk menilai pengalaman pengguna terhadap suatu sistem [30]. SUS terdiri dari 10 pertanyaan sederhana dengan skala Likert 1–5, mencakup aspek efektivitas, efisiensi, dan kepuasan. Keunggulannya: mudah dijawab, fleksibel untuk berbagai jenis antarmuka, dan memiliki skala 0–100 yang mudah dipahami. Hasil SUS dikonversi ke *letter grades*:

TABEL 1
(Tabel SUS)

SUS	Grade	Kriteria
nilai >= 80.3	A	Sangat Bagus
74 <= nilai < 80.3	B	Bagus
68 <= nilai < 74	C	Cukup
51 <= nilai < 68	D	Kurang
nilai < 51	E	Sangat Kurang

L. Blackbox Testing

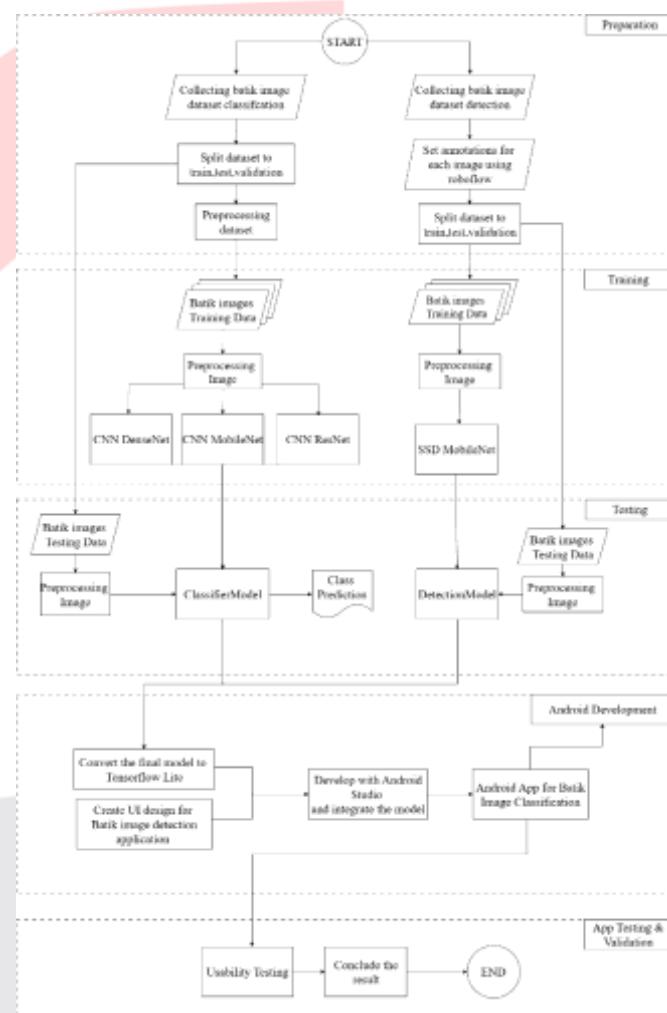
Black Box Testing adalah teknik pengujian perangkat lunak yang memeriksa fungsi eksternal aplikasi tanpa melihat kode atau logika internal. Fokusnya memastikan input menghasilkan output sesuai spesifikasi, mencakup aspek seperti fungsionalitas, basis data, antarmuka, kinerja, hingga inisialisasi dan terminasi sistem.[31].

M. Task Scenario

Task Scenario adalah metode pengujian perangkat lunak yang menguji interaksi pengguna dalam menyelesaikan tugas tertentu [32]. Skenario disusun sesuai alur tugas, misalnya mengunggah gambar atau mengakses informasi batik. Tujuannya memastikan aplikasi mendukung interaksi dengan lancar, fitur berfungsi sesuai kebutuhan, dan masalah alur penggunaan teridentifikasi. Fokusnya pada antarmuka dan interaksi pengguna, berbeda dari *black box testing* yang menilai *input-output* tanpa memperhatikan interaksi pengguna.

III. METODE

Pada bagian ini menjelaskan tahapan penelitian dan rancangan sistem yang mencakup analisis kebutuhan, pengembangan model, serta implementasi aplikasi. Sistem terdiri dari model deteksi motif batik menggunakan SSD dengan backbone MobileNetV2, model klasifikasi dengan perbandingan MobileNetV2, DenseNet121, dan ResNet50, serta aplikasi Android untuk deteksi dan klasifikasi otomatis. Proses meliputi pengumpulan dan preprocessing data, pelatihan dan evaluasi model, hingga konversi ke TensorFlow Lite (TFLite) untuk integrasi optimal di perangkat mobile. Visualisasi alur penelitian dapat dilihat pada GAMBAR 5.



GAMBAR 5
(Alur Perancangan Sistem)

A. Perancangan Model Klasifikasi Batik

Perancangan model klasifikasi motif batik diawali dengan proses pengumpulan dan persiapan dataset yang memuat berbagai jenis motif dari 12 kelas berbeda, seperti Parang, Kawung, Mega Mendung, Truntum, Ceplok, dan lainnya. Dataset diperoleh dari hasil penelitian sebelumnya yang dilakukan oleh [33] , serta dari platform terbuka seperti Kaggle [34]. Setiap kelas memiliki jumlah data citra yang bervariasi, dan rincian distribusi jumlah citra per kelas dapat dilihat pada Tabel 1.

TABEL 2
(Dataset Klasifikasi Batik)

No	Jenis Motif	Gambar Motif	Jumlah
1.	Barong		1.300
2.	Buketan		1.300
3.	Ceplok		1.300
4.	Insang		1.300
5.	Kawung		1.300
6.	Megamendung		1.300
7.	Parang		1.300
8.	Pring		1.300
9.	Sekarjagad		1.300
10.	Semen		1.300
11.	Sidoluhur		1.300
12.	Truntum		1.300

Setelah proses pengumpulan, dilakukan tahapan *preprocessing* yang mencakup seleksi kualitas citra serta augmentasi seperti rotasi, *flipping*, *zooming*, dan penyesuaian *brightness* untuk meningkatkan keragaman data dan mengurangi risiko *overfitting*. Seluruh citra dinormalisasi ke skala piksel [0,1] dan diubah ukurannya menjadi 224×224 piksel agar sesuai dengan standar input arsitektur CNN. Dataset kemudian dibagi menjadi tiga skenario pembagian data pelatihan, validasi, dan pengujian, yaitu 80:10:10; 70:15:15; dan 60:20:20, guna menguji konsistensi performa model terhadap variasi distribusi data.

Model pelatihan menggunakan pendekatan *transfer learning* dengan tiga arsitektur CNN: MobileNetV2,

DenseNet121, dan ResNet50. Pemilihan model didasarkan pada performa dan efisiensi komputasi untuk kebutuhan integrasi ke dalam aplikasi *mobile* berdasarkan pada penelitian-penelitian sebelumnya. Evaluasi dilakukan melalui *confusion matrix* dan empat metrik utama: akurasi, presisi, *recall*, dan *F1-score*. Perhitungan metrik-metrik tersebut dilakukan berdasarkan nilai *True Positive (TP)*, *False Positive (FP)*, *True Negative (TN)*, dan *False Negative (FN)*, dengan rumus sebagai berikut:

Akurasi mengukur seberapa tepat model dalam melakukan klasifikasi secara keseluruhan, dihitung dengan rumus:

$$Akurasi = \frac{TP + TN}{TP + FN + FP + TN} \quad (1)$$

Presisi menunjukkan proporsi prediksi positif yang benar-benar relevan, dihitung dengan:

$$Presisi = \frac{TP}{TP + FP} \quad (2)$$

Recall menunjukkan kemampuan model dalam menemukan seluruh kasus positif yang benar, dengan rumus:

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

F1-score merupakan rata-rata harmonis dari presisi dan *recall*, yang berguna saat dibutuhkan keseimbangan antara keduanya. Rumus *F1-score* adalah:

$$F1 - score = \frac{Presisi \times Recall}{Presisi + Recall} \quad (4)$$

Evaluasi ini bertujuan untuk menilai tingkat ketepatan model dalam mengenali dan mengklasifikasikan motif batik yang kompleks dan beragam. Model terbaik yang diperoleh kemudian dikonversi ke format TensorFlow Lite (.tflite) agar dapat diimplementasikan secara ringan dalam aplikasi Android.

B. Perancangan Model Deteksi Batik

Perancangan model deteksi motif batik dimulai dengan pengumpulan dataset gambar dari berbagai sumber *online*. Gambar mencakup variasi latar belakang, ukuran, dan posisi motif batik, rincian dataset dapat dilihat pada Tabel 2.

No	Jenis Motif	Gambar Motif	Jumlah
1.	Barong		59
2.	Buketan		60
3.	Ceplok		63
4.	Insang		63

No	Jenis Motif	Gambar Motif	Jumlah
5.	Kawung		62
6.	Megamendung		61
7.	Parang		63
8.	Pring		52
9.	Sekarjagad		59
10.	Semen		62
11.	Sidoluhur		61
12.	Truntum		60

Dataset kemudian dianotasi menggunakan Roboflow dengan *bounding box* dan label sesuai jenis motif. Total gambar yang diperoleh setelah augmentasi adalah **1.357** gambar dan dibagi menjadi tiga bagian: 84% pelatihan, 8% validasi, dan 8% pengujian. Model deteksi menggunakan arsitektur SSD MobileNetV2 yang telah pre-trained pada dataset COCO, dilatih menggunakan TensorFlow Object Detection API. Model ini dipilih karena ringan dan optimal untuk perangkat *mobile*.

Setelah pelatihan, model diuji pada data validasi dan pengujian menggunakan metrik standar: *Mean Average Precision* (mAP) mengukur rata-rata presisi pada berbagai nilai *IoU*.

$$AP(IoU) = \frac{1}{n} \sum_{i=1}^n Precision(i) \quad (5)$$

Average Recall (AR) menilai kemampuan model mendeteksi seluruh objek.

$$mAP = \frac{1}{n} \sum_{i=1}^n AP(IoU_i) \quad (6)$$

Localization Loss (LL) menghitung selisih prediksi dan lokasi asli objek:

$$Localization Loss = \sum_{i=1}^n smooth_L1(x_i, \hat{x}_i) \quad (7)$$

Classification Loss (CL) menghitung kesalahan dalam prediksi label kelas:

$$Classification Loss = - \sum_{i=1}^n \log P(y_i|x_i) \quad (8)$$

Regularization Loss (RL) menghindari overfitting dengan penalti bobot model:

$$Regularization Loss = \lambda \sum_{i=1}^n ||\theta_i||^2 \quad (9)$$

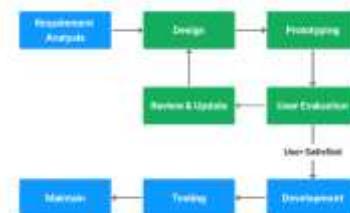
Total Loss adalah gabungan seluruh fungsi loss:

$$Total Loss = LL + CL + RL \quad (10)$$

Model akhir dikonversi ke format TensorFlow Lite (.tflite) agar efisien dan kompatibel untuk integrasi ke dalam aplikasi Android.

C. Perancangan Aplikasi Android

Pengembangan aplikasi Android dilakukan menggunakan metode *Software Development Life Cycle* (SDLC) model Prototype, yang menekankan iterasi cepat berdasarkan umpan balik pengguna. Alur proses SDLC model Prototype ditampilkan pada GAMBAR 6.



GAMBAR 6
(Alur SDLC Prototype)

Proses dimulai dari analisis kebutuhan, dilanjutkan dengan perancangan wireframe dan mockup. Prototipe awal dikembangkan untuk diuji langsung oleh pengguna. Jika belum sesuai, dilakukan revisi hingga disetujui. Setelah itu, aplikasi dikembangkan secara penuh menggunakan Kotlin di Android Studio, mencakup fitur klasifikasi batik, riwayat, artikel, dan pengelolaan profil.

Evaluasi dilakukan melalui *Black Box Testing*, *Task Scenario*, serta survei pengalaman pengguna menggunakan metode *System Usability Scale* (SUS) untuk mengukur tingkat keterpakaian aplikasi. Tahap pemeliharaan (maintain) dilakukan secara berkala untuk perbaikan dan peningkatan fitur berdasarkan umpan balik pengguna.

IV. HASIL DAN PEMBAHASAN

Pada bagian ini menyajikan hasil dari implementasi dan evaluasi aplikasi deteksi dan klasifikasi motif batik. Pembahasan mencakup performa model klasifikasi dan deteksi menggunakan metrik evaluasi seperti akurasi, presisi, dan recall, serta hasil pengujian fungsionalitas aplikasi melalui skenario tugas dan evaluasi pengalaman pengguna menggunakan kuesioner SUS. Temuan dari pengujian ini kemudian dianalisis untuk menilai efektivitas sistem dalam memenuhi kebutuhan pengguna serta potensi pengembangannya di masa depan.

A. Hasil Pengujian Model Klasifikasi Batik

Model klasifikasi diuji menggunakan tiga arsitektur CNN (MobileNetV2, DenseNet121, ResNet50) dan tiga rasio pembagian data. Hasil evaluasi ditampilkan pada Tabel IV.1 berikut:

Model	Rasio Data	Akurasi	Presisi	Recall	F1-Score
MobileNetV2	80:10:10	92%	92%	92%	92%
MobileNetV2	70:15:15	91%	91%	91%	90%
MobileNetV2	60:20:20	87%	88%	87%	87%
DenseNet121	80:10:10	89%	89%	89%	89%
DenseNet121	70:15:15	90%	90%	90%	90%
DenseNet121	60:20:20	87%	88%	87%	86%
ResNet50	80:10:10	77%	80%	77%	75%
ResNet50	70:15:15	75%	75%	75%	75%
ResNet50	60:20:20	74%	76%	74%	73%

Model terbaik adalah MobileNetV2 dengan rasio 80:10:10, dengan akurasi dan *F1-Score* sebesar 92%.

B. Hasil Pengujian Model Deteksi Batik

Model SSD MobileNetV2 diuji menggunakan metrik evaluasi seperti mAP, AR, dan Loss. Ringkasan hasil pengujian disajikan dalam Tabel IV.2 berikut:

Metrik Evaluasi	Nilai
Average Precision (AP) @ [IoU=0.50:0.95, maxDets=100]	0.476
Average Precision (AP) @ [IoU=0.50, maxDets=100]	0.722
Average Precision (AP) @ [IoU=0.75, maxDets=100]	0.539
Average Precision (AP) - Small Objects	0.200
Average Precision (AP) - Medium Objects	0.310
Average Precision (AP) - Large Objects	0.541
Average Recall (AR) @ [IoU=0.50:0.95, maxDets=1]	0.541
Average Recall (AR) @ [IoU=0.50:0.95, maxDets=10]	0.620
Average Recall (AR) @ [IoU=0.50:0.95, maxDets=100]	0.622
Average Recall (AR) - Small Objects	0.200
Average Recall (AR) - Medium Objects	0.490
Average Recall (AR) - Large Objects	0.674
Loss - Localization	0.139024
Loss - Classification	0.501531
Loss - Regularization	0.142259
Total Loss	0.782814

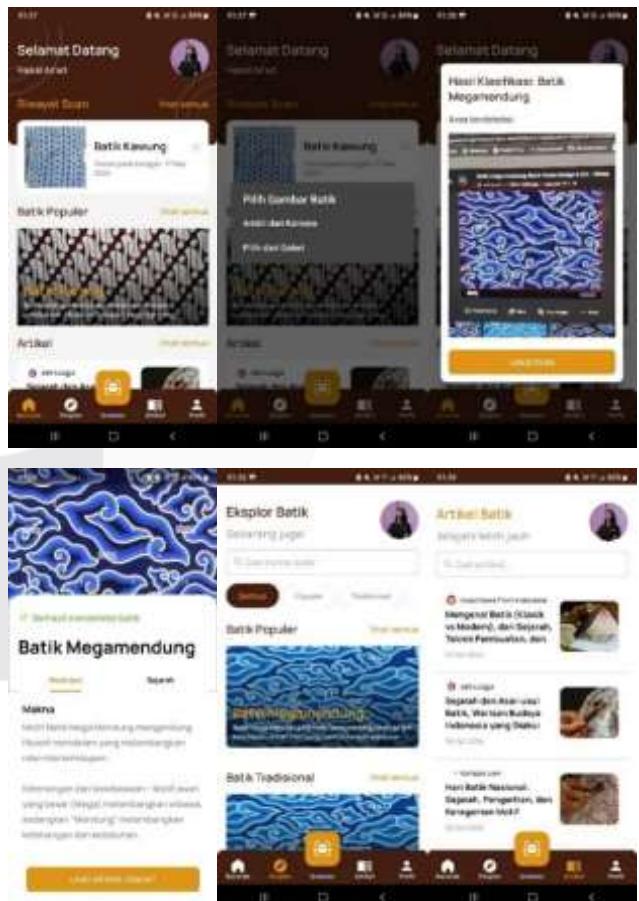
C. Hasil Pengujian Usability dan Fungsionalitas Aplikasi

Hasil pengujian usability dan fungsionalitas menunjukkan bahwa aplikasi telah berjalan dengan baik dan memenuhi standar kenyamanan serta kemudahan penggunaan. Berdasarkan pengisian kuesioner *System Usability Scale* (SUS) oleh 26 responden, diperoleh skor rata-

rata sebesar **83,17** yang termasuk dalam kategori "Sangat Bagus" (A). Pengujian fungsionalitas melalui metode *Black Box Testing* terhadap 22 skenario, mulai dari registrasi hingga fitur favorit dan artikel, menunjukkan bahwa seluruh fitur dapat dijalankan sesuai ekspektasi tanpa kendala. Sementara itu, pada pengujian *Task Scenario* yang melibatkan 7 responden, tercatat bahwa 56 dari 63 tugas berhasil diselesaikan dengan baik, menghasilkan tingkat efektivitas sebesar **88,89%** yang masuk dalam kategori *Sangat Efektif*. Hal ini menunjukkan bahwa sebagian besar pengguna mampu menyelesaikan tugas-tugas yang diberikan dengan lancar dan sesuai alur yang dirancang dalam aplikasi. Selain itu, nilai efisiensi yang diperoleh mencapai **77,16%**, yang mengindikasikan bahwa pengguna tidak hanya mampu menyelesaikan tugas dengan benar, tetapi juga dalam waktu dan usaha yang relatif efisien.

D. Hasil Pengembangan Aplikasi Android

Aplikasi Batikara terdiri dari 23 halaman antarmuka yang dirancang untuk mendukung proses deteksi, klasifikasi, dan edukasi motif batik secara menyeluruh. Lima halaman utama yang mencerminkan fitur inti aplikasi ditampilkan pada Gambar 7, meliputi halaman Home, halaman Proses Deteksi dan Klasifikasi, serta halaman Eksplorasi Batik. Desain antarmuka dikembangkan secara user-friendly, dengan fokus pada kemudahan navigasi dan kenyamanan pengguna dalam menjelajahi fitur-fitur aplikasi.



GAMBAR 7
(Tampilan Aplikasi)

V. KESIMPULAN

Berdasarkan hasil penelitian, implementasi sistem, serta evaluasi performa dan usability, dapat disimpulkan bahwa aplikasi Batikara berhasil dikembangkan sebagai aplikasi Android yang mampu mendeteksi dan mengklasifikasikan motif batik secara otomatis menggunakan model *Convolutional Neural Network* (CNN). Aplikasi ini memudahkan pengguna dalam mengenali motif batik melalui proses scan yang cepat dan informatif. Dari sisi performa, model klasifikasi MobileNetV2 menunjukkan akurasi tertinggi sebesar 92%, disusul DenseNet121 dengan 90%, keduanya lebih unggul dibandingkan ResNet50 yang kurang optimal dalam membedakan motif serupa. Model deteksi SSD MobileNetV2 juga memberikan hasil baik dengan *Average Precision* (AP) tertinggi 0,722 pada IoU 0,50, meskipun kurang efektif untuk objek kecil dan detail. Evaluasi *usability* menunjukkan aplikasi mudah digunakan dengan skor SUS sebesar 83,17, efektivitas 88,9%, dan efisiensi 77,16%, serta mendapat tanggapan positif dari pengguna, terutama UMKM. Masukan pengguna meliputi kebutuhan fitur pencarian berdasarkan daerah, navigasi yang lebih intuitif, dan integrasi *e-commerce*. Oleh karena itu, pengembangan selanjutnya disarankan meliputi integrasi *Feature Pyramid Networks* (FPN) untuk meningkatkan akurasi deteksi objek kecil, perluasan dataset batik lokal agar lebih inklusif, pengembangan fitur *e-commerce* untuk mendukung UMKM, serta mode tampilan ramah anak guna menarik minat generasi muda dalam pelestarian budaya batik secara digital dan interaktif.

REFERENSI

- [1] A. Atika, N. Kholifah, S. Nurrohmah, and R. Purwiningsih, "Eksistensi Motif batik klasik pada generasi Z," *TEKNOBUGA: Jurnal Teknologi Busana dan Boga*, vol. 8, no. 2, pp. 141–144, 2020.
- [2] I. Maulana, H. Sastypratiwi, H. Muhardi, N. Safradi, and H. Sujaini, "Implementasi Convolutional Neural Network (CNN) untuk Klasifikasi Motif Batik pada Aplikasi Computer Vision Berbasis Android," *JEPIN (Jurnal Edukasi dan Penelitian Informatika)*, vol. 9, no. 3, pp. 384–393, 2023.
- [3] A. Suharson, "BATIK DALAM KONSTELASI BUDAYA GLOBAL: MERAJUT KEMBALI NILAI-NILAI ESTETIKA, ETIKA, DAN RELIGIUS," in *Prosiding Seminar Nasional Industri Kerajinan dan Batik*, 2021, pp. 11–13.
- [4] D. Fauziza and Z. Kolina, "Aplikasi Interaktif Pengenalan Batik untuk Pengunjung Pameran Berbasis Android dan Image Processing," *JTET (Jurnal Teknik Elektro Terapan)*, vol. 11, no. 1, pp. 8–18, 2022.
- [5] Statista Research Department, "Indonesia: mobile OS share 2024 | Statista," <https://www.statista.com/statistics/262205/market-share-held-by-mobile-operating-systems-in-indonesia/>. Accessed: Jan. 15, 2025. [Online]. Available: <https://www.statista.com/statistics/262205/market-share-held-by-mobile-operating-systems-in-indonesia/>
- [6] disty anastasya, S. Fahri, and S. Situmorang, "Implementasi Metode Convolutional Neural Network (CNN) Dalam Klasifikasi Motif Batik," *NUANSA INFORMATIKA*, vol. 18, no. 1, pp. 1–5, Jan. 2024, doi: 10.25134/ilkom.v18i1.21.
- [7] A. A. Elngar *et al.*, "Image classification based on CNN: a survey," *Journal of Cybersecurity and Information Management*, vol. 6, no. 1, pp. 18–50, 2021.
- [8] F. Arin and Y. S. Dwanoko, "APLIKASI PENGENALAN BUDAYA SUMBA BERBASIS ANDROID: APLIKASI PENGENALAN BUDAYA SUMBA BERBASIS ANDROID," *RAINSTEK: Jurnal Terapan Sains dan Teknologi*, vol. 3, no. 1, pp. 64–74, Mar. 2021, doi: 10.21067/jtst.v3i1.4508.
- [9] R. B. Abiyyi, E. R. Subhiyakto, and F. T. Sabilillah, "Centing: Aplikasi Cegah Stunting Anak berbasis Android menggunakan TensorFlow Lite," *Edumatic: Jurnal Pendidikan Informatika*, vol. 8, no. 2, pp. 625–634, 2024.
- [10] R. R. M. Akbar, F. Rizal, and W. J. Shudiq, "Implementasi Algoritma Convolutional Neural Network (Cnn) Untuk Deteksi Kesegaran Telur Berbasis Android," *Jusikom: Jurnal Sistem Komputer Musirawas*, vol. 8, no. 1, pp. 1–10, 2023.
- [11] M. R. D. Ulhaq, M. A. Zaidan, and D. Firdaus, "Pengenalan Ekspresi Wajah Secara Real-Time Menggunakan Metode SSD Mobilenet Berbasis Android," *Journal of Technology and Informatics (JoTI)*, vol. 5, no. 1, pp. 48–52, 2023.
- [12] A. R. Prasetyo and B. Aditya, "Analisis Perbandingan Algoritma Support Vector Machine (SVM) dan Convolutional Neural Network (CNN) untuk Sistem Deteksi Katarak," *Jurnal Ilmiah Teknik Mesin, Elektro dan Komputer*, vol. 3, no. 1, pp. 1–10, 2023.
- [13] L. Hakim, S. P. Kristanto, D. Yusuf, and F. N. Afia, "Pengenalan Motif Batik Banyuwangi Berdasarkan Fitur Grey Level Co-Occurrence Matrix," *Jurnal Teknoinfo*, vol. 16, no. 1, pp. 1–7, 2022.
- [14] F. Haswan and H. Nopriandi, "Perancangan Model Sistem Informasi Penjualan Batik Kuansing," *Jurnal Teknik Komputer*, vol. 6, no. 2, pp. 207–211, 2020.
- [15] Y. P. Supriono, *Ensiklopedia The Heritage Of Batik, Identitas Pemersatu Kebanggaan Bangsa*. Penerbit Andi, 2024. Accessed: Mar. 17, 2025. [Online]. Available: https://books.google.co.id/books?hl=id&lr=&id=Q5T3EAAAQBAJ&oi=fnd&pg=PR2&dq=Ensiklopedi+a+The+Heritage+Of+Batik,+Identitas+Pemersatu+Kebanggaan+Bangsa&ots=kwLwDam2MY&sig=9CpMnkgT0HaYJZYgk2I8XYNmM7M&redir_esc=y#v=onepage&q=Ensiklopedia%20The%20Heritage%20Of%20Batik%2C%20Identitas%20Pemersatu%20Kebanggaan%20Bangsa&f=false
- [16] A. Wulandari, *Batik Nusantara: Makna filosofis, cara pembuatan, dan industri batik*. Penerbit Andi, 2022.
- [17] D. Djuniwarti, H. Kurniawan, and S. Karisa, "PENGEMBANGAN MOTIF BATIK SINGOBARONG KERATON CIREBON BERGAYA FLATDESIGN UNTUK GENERASI MILLENIAL," *Bookchapter ISBI Bandung*, 2024.

- [18] M. E. Widiana and K. Karsam, "BUKU REFERENSI Strategi Manajemen Standarisasi Batik Untuk Meningkatkan Pemasaran Yang Berdaya Saing Di Era Revolusi Industri 4.0," 2024.
- [19] T. B. Sasongko, H. Haryoko, and A. Amrullah, "Analisis Efek Augmentasi Dataset dan Fine Tune pada Algoritma Pre-Trained Convolutional Neural Network (CNN)," *Jurnal Teknologi Informasi dan Ilmu Komputer*, vol. 10, no. 4, pp. 763–768, 2024.
- [20] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep learning*, vol. 1, no. 2. MIT press Cambridge, 2016.
- [21] S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network 2017 International Conference on Engineering and Technology (ICET)," *New York. IEEE*, pp. 1–6, 2017.
- [22] M. Krichen, "Convolutional neural networks: A survey," *Computers*, vol. 12, no. 8, p. 151, 2023.
- [23] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4510–4520.
- [24] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700–4708.
- [25] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.
- [26] M. A. Aziz, A. S. Rachman, and I. M. B. Suksmadana, "PENGUJIAN DETEKSI OBJEK MANUSIA MENGGUNAKAN JETSON NANO DENGAN MODEL SSD MOBILENETV2," *Jurnal Informatika Teknologi dan Sains (Jinteks)*, vol. 6, no. 4, pp. 830–840, 2024.
- [27] Y. H. Natbais and A. B. S. Umbu, "Aplikasi deteksi penyakit pada daun tomat berbasis android menggunakan model terlatih Tensorflow Lite," *Teknotan: Jurnal Industri Teknologi Pertanian*, vol. 17, no. 2, pp. 83–90, 2023.
- [28] A. M. Sampurna, A. Setiawan, and H. A. Al Kautsar, "Development of an Android Application to Identify Fish Species Using Kotlin-based Android Studio," *Jurnal Teknik Komputer*, vol. 11, no. 1, pp. 19–31, 2025.
- [29] S. Herlinah and K. H. Musliadi, *Pemrograman Aplikasi Android dengan Android Studio, Photoshop, dan Audition*. Elex Media Komputindo, 2019.
- [30] I. Rachmawati and R. Setyadi, "Evaluasi Usability Pada Sistem Website Absensi Menggunakan Metode SUS," *Journal of Information System Research (JOSH)*, vol. 4, no. 2, pp. 551–561, 2023.
- [31] E. Novalia and A. Voutama, "Black Box Testing dengan Teknik Equivalence Partitions Pada Aplikasi Android M-Magazine Mading Sekolah," *Syntax: Jurnal Informatika*, vol. 11, no. 01, pp. 23–35, 2022.
- [32] H. Firdaus and A. Zakiah, "Implementation of usability testing methods to measure the usability aspect of management information system mobile application (Case study sukamiskin correctional institution)," *IJ Mod. Educ. Comput. Sci*, vol. 5, pp. 58–67, 2021.
- [33] A. Y. Wicaksono, N. Suciati, C. Fatichah, K. Uchimura, and G. Koutaki, "Modified convolutional neural network architecture for batik motif image classification," *IPTEK Journal of Science*, vol. 2, no. 2, 2017.
- [34] N. M. Safira, "Indonesia Batik Dataset," 2024, Accessed: Jul. 17, 2025. [Online]. Available: <https://www.kaggle.com/datasets/nadyams/indonesia-batik-dataset/data>