



# Implementasi *Convolutional Neural Network* (CNN) untuk Klasifikasi Citra Batik Nusantara

Zufar Faiil Haq<sup>1</sup>, Mufti Ari Bianto<sup>2\*</sup>, Afifah Agustin<sup>3</sup>, Moch. Ryan Nurfebrianto<sup>4</sup>

<sup>1-4</sup> Teknik Komputer, Universitas Muhammadiyah Lamongan, Indonesia  
[failhaq83@gmail.com](mailto:failhaq83@gmail.com)<sup>1</sup>, [muftiari10@gmail.com](mailto:muftiari10@gmail.com)<sup>2\*</sup>, [afifahagustin344@gmail.com](mailto:afifahagustin344@gmail.com)<sup>3</sup>,  
[ryanfebrianto140@gmail.com](mailto:ryanfebrianto140@gmail.com)<sup>4</sup>

Alamat: Jl. Plalangan Plosowahyu, Lamongan Jawa Timur, Indonesia 62218

Korespodensi email: [muftiari10@gmail.com](mailto:muftiari10@gmail.com)

**Abstract.** *Batik is a cultural heritage of the nation, with each batik having a unique and diverse pattern motif. The batik culture is very strong in Indonesia, so batik can be found in all regions of the archipelago. Each batik has its own characteristics and traits to distinguish itself in each area. However, many people find it difficult to differentiate the types of batik motif patterns, one of which is the Nusantara Megamendung batik. Therefore, this research aims to introduce the classification process of Nusantara batik motif patterns using one of the Deep Learning methods, namely Convolutional Neural Network (CNN), to differentiate the types of batik motif patterns in each region. The dataset is taken from the numeric representations of Red, Green, and Blue (RGB) values of each pixel, which are used as model learning features to study color patterns and textures. From the results of the experiments conducted, the batik image classification using the CNN method has a high level of accuracy. The batik classification model achieved an accuracy of 85%, demonstrating a fairly good ability to identify batik images, one of which is the Mega Mendung batik. The Mega Mendung and Keraton classes showed perfect performance, with precision, recall, and F1-score close to 1.00. However, the Bali class was the main weak point, with a recall of only 60%, indicating that 40% of Bali Batik samples were misclassified, primarily as Keraton.*

**Keywords:** *Convolutional Neural Networks, VGG-16, Batik Mega Mendung, Image Classification, Deep Learning*

**Abstrak.** Batik merupakan warisan budaya bangsa, Setiap batik memiliki pola motif yang unik dan beragam. Budaya batik sangat kental di Indonesia sehingga batik dapat di temukan di semua wilayah Nusantara, Setiap batik memiliki ciri khas dan karakteristik masing-masing untuk membedakannya di setiap daerah. Namun banyak masyarakat yang sulit untuk membedakan jenis-jenis motif pola batik, salah satunya adalah batik Nusantara Megamendung, Sehingga penelitian ini bertujuan untuk proses pengenalan klasifikasi pola motif batik Nusantara dengan menggunakan salah satu metode Deep Learning yaitu Convolutional Neural Network (CNN) untuk membedakan jenis motif pola batik di setiap daerah. Data set diambil dari nilai representasi numerik Red, Green, Blue (RGB) setiap pixel yang di gunakan sebagai fitur pembelajaran model untuk mempelajari pola warna dan tekstur, dari hasil percobaan yang di lakukan klasifikasi citra batik menggunakan metode CNN memiliki tingkat hasil akurasi yang tinggi model klasifikasi batik mencapai akurasi 85%, menunjukkan kemampuan yang cukup baik dalam mengidentifikasi gambar batik, salah satunya adalah batik Mega Mendung. Kelas Megamendung dan Keraton memiliki performa sempurna dengan precision, recall, dan F1-score mendekati 1.00. Namun, kelas Bali menjadi titik lemah utama dengan recall hanya 60%, mengindikasikan 40% sampel Batik Bali salah diklasifikasikan, terutama sebagai Keraton.

**Kata kunci:** Jaringan Saraf Konvolusional, VGG-16, Batik Mega Mendung, klasifikasi Citra, Pembelajaran Mendalam

## 1. PENDAHULUAN

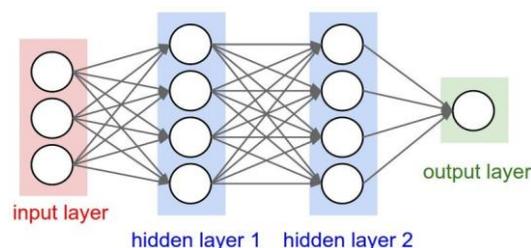
Batik adalah budaya yang sangat kental di Indonesia, sehingga dapat ditemukan di seluruh daerah Nusantara dengan keunikan pola motif batik masing-masing. Bahkan UNESCO telah mengakui keberadaan batik sebagai warisan budaya dunia pada tanggal 2 Oktober 2009 dengan memasukkan Batik Indonesia kedalam daftar Representatif sebagai Budaya Tak-Benda Warisan Manusia (Representative List of the Intangible Cultural

Heritage of Humanity) dan pada tanggal tersebut diperingati sebagai Hari Batik Nasional. Indonesia memiliki sekitar 5.849 motif batik dari sabang sampai merauke, salah satunya Batik Mega Mendung, Batik Bali, Batik Keraton, Batik Pekalongan.[1]

Karena batik memiliki motif yang sangat beragam, hampir setiap tema batik di setiap daerah memiliki motif yang agak mirip tetapi tidak persis sama., metode pengenalan pola merupakan salah satu cara untuk mengidentifikasi motif batik. Namun, pengetahuan tentang mengidentifikasi berbagai jenis motif batik mungkin hanya dimiliki oleh orang-orang tertentu yang memiliki keahlian di bidang terkait seperti pembuatan batik.[2]

Deep Learning adalah salah satu cabang ilmu dari Machine Learning yang memanfaatkan jaringan syaraf tiruan untuk penerapan masalah dengan kumpulan data besar, yang dapat menghasilkan hasil yang lebih tepat karena meniru fungsi otak manusia, CNN merupakan salah satu teknik Pembelajaran Mendalam yang dapat menghasilkan hasil yang luar biasa dalam hal pengenalan objek visual. CNN merupakan jenis jaringan saraf yang sering diterapkan pada data gambar dan memiliki kemampuan untuk mengidentifikasi dan mendeteksi objek dalam gambar.[2] CNN mempunyai kemampuan untuk mengekstraksi fitur-fitur yang relevan dari data visual dan dapat menggunakannya untuk melakukan proses klasifikasi.[3]

CNN juga menggunakan beberapa lapisan dan operasi untuk melakukan banyak hal, seperti melakukan proses mengklasifikasikan gambar, mengidentifikasi objek dan segmentasi gambar.[4] Salah satu teknik Pembelajaran Mendalam yang dapat digunakan untuk memasukkan gambar dan menerapkan bobot dan bias pada berbagai objek dan elemen gambar guna membedakan satu gambar dari gambar lainnya adalah Jaringan Syaraf Tiruan Konvolusional (CNN). CNN merupakan varian dari Multi Layer Perception (MLP) yang terinspirasi dari jaringan otak manusia.[5] Ada tiga jenis layer Utama CNN (convolutional neural network), yaitu Input Layer, Output Layer, dan beberapa Hidden Layer. Lapisan ini merupakan lapisan Convolution, fungsi dari lapisan ini adalah untuk ekstraksi fitur, lapisan penyatuan untuk mengurangi dimensi data, dan lapisan yang terhubung sepenuhnya untuk melakukan proses klasifikasi dan pengambilan keputusan.[6]

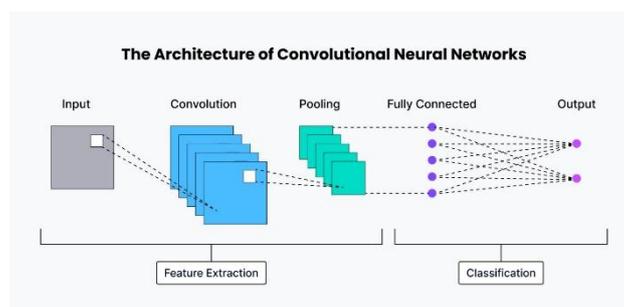


**Gambar 1.** *Arsitektur Multi Layer Perception*

Penjelasan gambar 1.

Sebuah arsitektur desain MLP ini terdiri dari *input layer*, *hidden layer*, dan *output layer*. *Ji neuron* (lingkaran putih) ditemukan di setiap lapisan. Untuk menghasilkan output, MLP mengambil input data satu dimensi dan menyebarkannya ke seluruh jaringan. Kualitas mode ditentukan oleh setiap hubungan antara parameter bobot satu dimensi. Operasi non-linier yang dikenal sebagai fungsi aktivasi digunakan untuk mengubah hasil perhitungan setelah operasi linier selesai dengan nilai bobot saat ini di setiap data input pada lapisan.[5]

Pendekatan CNN terdiri dari dua fase: klasifikasi citra *feedforward* pada tahap pertama dan *learning stage* yang menggunakan *backpropagation method* pada tahap kedua. Dengan menggunakan bobot dan bias yang diperbarui dari proses *backpropagation*, proses *Feedforward* merupakan langkah awal yang menambahkan beberapa lapisan untuk klasifikasi citra. Prosedur pengujian juga akan memanfaatkan langkah ini lagi. Hasil dari proses *Feedforward* terkandung dalam langkah kedua, yang dikenal sebagai *backpropagation*, yang menelusuri lapisan keluaran kembali ke lapisan pertama. Mendapatkan bobot dan bias dengan nilai yang baru merupakan tanda pelacakan data. Sebelum proses klasifikasi dimulai, akan dilakukan *preprocessing* dengan cara *wrap* dan *cropping* dataset agar foto-foto terkonsentrasi pada objek yang akan diklasifikasi. Berdasarkan penelitian “Sistem Pemberian Saran Resep Kuliner Indonesia Menggunakan Metoda *Case Based Reasoning* Dengan Algoritma Similaritas *Czekanowski* Berbobot”, menemukan bahwa jika bahan masakan digunakan dalam sebagian besar resep tetapi tidak semuanya, nilai bobotnya akan semakin tinggi. Di sini, bahan masakan dapat dianggap sebagai nilai, dan orang yang membuat resep dapat dianggap sebagai pembangkit nilai. Lapisan CNN mengambil item dari foto input menggunakan filter. Bobot dalam filter digunakan untuk mengidentifikasi tepi, lengkungan, dan warna objek. Untuk mencocokkan informasi spasial dalam data gambar, konvolusi akan menciptakan transformasi *linear* dari gambar *input*. Urutan bidang *receptive* dibuat dengan menerapkan filter berulang kali. Berikut adalah rincian beberapa lapisan CNN.[5]



**Gambar 2.** Layer Convolutional Neural Network CNN

Penjelasan gambar 2.

Gambar tersebut menunjukkan arsitektur dasar Convolutional Neural Network (CNN) yang terdiri dari lima lapisan utama. Pertama, Input Layer menerima data gambar seperti gambar batik. Kemudian, Convolution Layer menggunakan filter untuk mengekstrak fitur-fitur lokal seperti tepi dan tekstur. Pooling Layer berfungsi untuk mengurangi dimensi data sambil mempertahankan fitur penting. Selanjutnya, Fully Connected Layer menghubungkan semua fitur yang telah diekstrak untuk proses klasifikasi. Terakhir, Output Layer menghasilkan prediksi akhir, misalnya mengidentifikasi jenis motif batik tertentu. Secara keseluruhan, CNN bekerja melalui dua tahap utama: ekstraksi fitur (Convolution + Pooling) dan klasifikasi (Fully Connected + Output), menjadikannya sangat efektif untuk tugas pengenalan gambar seperti klasifikasi batik Nusantara.

*Convolutional Layer* merupakan lapisan yang digunakan untuk melakukan operasi konvolusi pada *output layer*. Sebagai elemen pembangun utama CNN, lapisan ini terdiri dari sejumlah filter yang dipelajari secara acak untuk menjalankan operasi konvolusional, untuk mempelajari representasi fitur lapisan input melalui ekstraksi fitur. Teknik konvolusi diterapkan pada data gambar untuk mengekstraksi fitur dari gambar input., bergantung pada informasi *spasial* yang disertakan dalam data, konvolusi akan menghasilkan transformasi linear dari data input. Agar kernel dapat dilatih menggunakan input CNN, bobot pada lapisan akan menyediakan persyaratan untuk kernel konvolusi.[5]

*Pooling Layer* adalah lapisan yang berfungsi untuk meminimalkan sumber daya komputasi yang diperlukan untuk memproses data, lapisan penggabungan mengurangi ukuran *spasial* fitur konvolusi. Dengan demikian, dimensi *feature map* berkurang, yang mempercepat proses komputasi karena lebih sedikit parameter yang digunakan.[5] Banyak metode dalam Pooling Layer ini yang bisa dipakai diantaranya maxpooling yaitu proses pengambilan nilai terbesar dari seluruh nilai matriks keluaran hasil fungsi aktivasi, selain itu ada juga Average Pooling yaitu proses pengambilan nilai rata-rata.[7]

*Fully Connected Layer* merupakan lapisan yang mengubah dimensi data untuk memungkinkan klasifikasi *linear* data. Untuk memastikan fitur mana yang paling terkait dengan kelas tertentu, *layer* ini menerima informasi dari prosedur sebelumnya. Fungsi *layer* ini adalah menyatukan semua *node* menjadi satu dimensi.[5] Fully Connected Layer digunakan dalam pendekatan MLP yang bertujuan untuk memproses data agar dapat melakukan proses klasifikasi.[8]

## 2. TINJAUAN PUSTAKA

Implementasi deep learning untuk klasifikasi gambar menggunakan convolutional neural network (CNN) pada batik sasambo : Kesimpulan yang dapat ditarik dari diskusi adalah: Klasifikasi batik sasambo dengan motif kangkung, bunga aruna, dan peresean menggunakan CNN dan Deep Learning diperkirakan telah menghasilkan hasil yang cukup baik. Untuk data pengujian, hasil klasifikasi CNN menunjukkan tingkat akurasi sebesar 80%.[2]

Klasifikasi Image Tumbuhan Obat Sirih dan Binahong Menggunakan Metode Convolutional Neural Network (CNN) : Pendekatan Convolutional Neural Network digunakan dalam penelitian ini untuk mengkategorikan tanaman obat binahong dan sirih. 900 foto berukuran 200 x 150 piksel digunakan dalam penelitian ini, dengan menggunakan rasio pembagian data 7:2:1 untuk pelatihan, validasi, dan pengujian. Gambar dapat diklasifikasikan secara efektif menggunakan pendekatan Convolutional Neural Network (CNN). Tingkat akurasi data pengujian sebesar 95,6% menjadi buktinya. Keakuratan dan kecepatan prosedur klasifikasi gambar untuk tanaman obat binahong dan sirih menjadi dasar perbandingan. Tiga skenario periode 15, 20, dan 25 epoch digunakan dalam penyelidikan ini. Keakuratan proses kategorisasi gambar adalah 95,67%, menurut data pada epoch 25.[9]

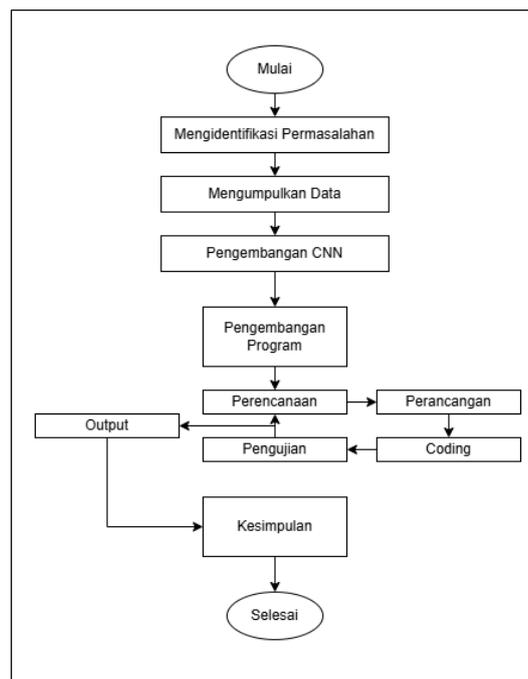
Klasifikasi penyakit mata menggunakan convolutional neural network (CNN) : Metode CNN dengan arsitektur model AlexNet digunakan untuk mengklasifikasikan penyakit mata. Berdasarkan temuan, empat kelas normal, katarak, glaukoma, dan penyakit retina digunakan sebagai pemutakhiran. Langkah pra-pemrosesan melibatkan perubahan ukuran gambar menjadi 224x224 piksel. Menurut desain yang disarankan, ekstraksi fitur merupakan tahap berikutnya, yang melibatkan tiga lapisan: lapisan konvolusional, lapisan penyatuan, dan lapisan yang terhubung sepenuhnya. Hasil akurasi studi klasifikasi penyakit mata yang menggunakan pendekatan CNN adalah 98,37% pada tahap implementasi CNN, yang menggunakan 150 epoch.[10]

Klasifikasi Penyakit Tanaman Tomat Menggunakan Metode Convolutional Neural Network (CNN) : Hasil penyakit tanaman tersebut diklasifikasikan menggunakan Convolutional Neural Network (CNN) dengan model VGG-16, akurasi klasifikasi dapat mendekati 98% untuk pelatihan dan 82% untuk validasi. Para peneliti telah berhasil mengintegrasikan klasifikasi penyakit tanaman tomat ke dalam sistem berbasis web berdasarkan temuan pengujian aplikasi dan kegunaan. Halaman web ini dimaksudkan untuk membantu dalam klasifikasi penyakit tanaman tomat.[11]

Pembangunan Aplikasi Penerjemah Bahasa Isyarat Dengan Metode CNN Berbasis Android : Penelitian ini menggunakan pendekatan Convolutional Neural Network (CNN) berhasil membuat model untuk memprediksi gambar dari bahasa isyarat. PreTrained Model digunakan dalam proses pengembangan untuk mencapai hasil akurasi yang tinggi. Dengan 200 epoch dan tingkat akurasi 97%, model pra-terlatih tipe Efficient-Net Lite1 digunakan. Tiga angka epoch yang berbeda 100, 150, dan 200 digunakan untuk menguji pengembangan. Di antara hasil pengujian Epoch adalah Epoch 100, yang memiliki tingkat akurasi 95% dan kehilangan pelatihan 0,95; Epoch 150, yang memiliki tingkat akurasi 93% dan kehilangan pelatihan 0,93; dan Epoch 200, yang memiliki tingkat akurasi 97% dan kehilangan pelatihan 0,89. Epoch 200 mencapai akurasi maksimum, dan model dengan akurasi tertinggi diubah menjadi model Tensorflow Lite untuk digunakan dalam aplikasi berbasis Android. Setelah penerapan, akurasi prediksi model pada bahasa isyarat alfabet L turun hingga 73%. Ada beberapa label lain yang salah diprediksi, yang menyebabkan prediksi bahasa isyarat tidak akurat. Meskipun tidak semua bahasa isyarat dapat diantisipasi dengan andal, aplikasi tersebut masih dapat digunakan untuk melakukan proses penerjemahan.[5]

### 3. METODE PENELITIAN

Tahapan metode penelitian yang dilakukan untuk mengklasifikasi gambar batik adalah sebagai berikut



Gambar 3. Tahapan Metode Penelitian

Penjelasan gambar 3.

Tahapan "Mengidentifikasi Permasalahan", di mana masalah yang akan diselesaikan didefinisikan dengan jelas. Tahapan "Mengumpulkan Data" untuk memperoleh informasi atau dataset yang diperlukan dalam pengembangan model. Tahap "Pengembangan CNN" merupakan inti dari proses ini, di mana arsitektur dan parameter model CNN dirancang. Tahap "Pengembangan Program" untuk mengimplementasikan model tersebut ke dalam kode. Tahap selanjutnya adalah "Output", yang mungkin merujuk pada hasil atau produk dari model yang dikembangkan. Proses kemudian dilanjutkan dengan "Perencanaan" dan "Perancangan" untuk menyusun strategi dan desain teknis lebih detail. Tahap "Pengujian" dilakukan untuk mengevaluasi kinerja model, diikuti dengan "Coding" yang mungkin merupakan penyempurnaan kode berdasarkan hasil pengujian. Terakhir, "Kesimpulan" dibuat untuk menganalisis hasil secara keseluruhan sebelum proyek atau penelitian dinyatakan "Selesai".

### **Pengembangan Model CNN**

Pada tahap ini, Peneliti akan membangun model Convolutional Neural Network (CNN) untuk melakukan klasifikasi gambar batik nusantara. Model dikembangkan menggunakan algoritma CNN, dengan python sebagai bahasa pemrograman *TensorFlow/keras* sebagai library untuk membangun arsitektur model dan *jupyter notebook* (Dalam Lingkungan Anaconda) sebagai platform pengembangan dan eksekusi kode.

#### Data Set

Peneliti mengumpulkan 200 data gambar untuk metode kategorisasi gambar batik dalam penelitian ini dengan mencari data di Google Image dan situs web Kaggle. Setelah itu, dataset akan diupload ke Gdrive, yang berfungsi sebagai perangkat penyimpanan Jupyter Notebook..



**Gambar 4.** *Sample Batik*

### Preprocessing Dataset.

Data preprocessing merupakan tahapan dimana data akan dilakukan pengisian data yang kosong, menghilangkan duplikasi data, memeriksa inkonsistensi data, pembersihan data serta memperbaiki kesalahan pada data [5]. Datasets yang digunakan dalam studi ini terdiri dari berbagai resolusi gambar dan akan menjadi ukuran yang sangat besar untuk melakukan proses *training* data. Maka dari itu, pada *preprocessing* data ini, akan dilakukan *resize* dimensi ukuran gambar ke resolusi yang lebih rendah yaitu 128 x 128 piksel dengan tiga saluran warna (RGB). Tahap ini berupaya untuk menjamin bahwa semua data memiliki dimensi yang konsisten, meminimalkan beban memori, dan mempercepat proses pelatihan. Untuk mempercepat konvergensi saat pelatihan menggunakan algoritma pengoptimalan seperti Adam, gambar dapat dinormalisasi selain diubah ukurannya sehingga nilai piksel berada di antara 0 dan 1.

### Membuat Model CNN.

Pada tahap ini, peneliti akan menggunakan teknik Convolutional Neural Network (CNN) untuk membuat model klasifikasi gambar batik Nusantara. Peneliti menggunakan arsitektur VGG-16 yang telah terbukti berhasil dalam tugas klasifikasi gambar untuk membangun sistem ini. Pendekatan transfer learning yang menggunakan bobot yang telah dilatih sebelumnya dari dataset ImageNet dan memodifikasinya untuk klasifikasi motif batik.

### Training Model.

Proses training datasets merupakan tahapan dalam proses pengembangan model, dimana pada proses ini datasets dilatih agar dapat memperoleh akurasi yang tinggi dari klasifikasi yang dilakukan. Pada tahap ini datasets akan dilakukan proses training dengan menggunakan metode CNN[5]. Pengujian training datasets dilakukan dengan menggunakan 40 Epoch. Pre-trained Model juga akan digunakan pada proses training, dengan tujuan didapatkannya akurasi tinggi ketika proses training selesai.

### Pengujian Model.

Pada tahap ini, peneliti akan menguji model yang dikembangkan dengan menulis skrip yang akan memungkinkan mereka menentukan apakah prediksi tersebut akurat atau tidak. Selain itu, data pengujian berbeda dari data pelatihan dan validasi.[12]

### **Pengembangan Sistem**

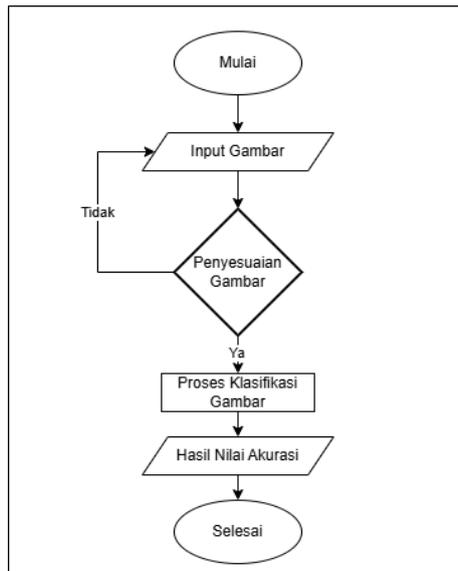
Tahap implementasi metode extreme programming merupakan tahap pembuatan Klasifikasi Citra Batik Nusantara Berbasis Web dengan mengimplementasikan model CNN

yang telah selesai dibuat sebelumnya. Tahapan dalam metode pengembangan Extreme Programming, yaitu:

Planning.

Tahap ini dimulai konteks sistem yang akan dibuat, mendefinisikan keluaran (output), fitur yang ada pada aplikasi, Fungsi dari aplikasi yang dibuat, serta alur pengembangan aplikasi.

Desain perancangan sistem yang dibuat



**Gambar 5.** Desain Perancangan Sistem

Penjelasan gambar 5.

“Mulai” persiapan sistem untuk klasifikasi gambar, “Input Gambar” gambar dimasukkan ke dalam sistem untuk diproses, “Penyesuaian Gambar” jika gambar tidak sesuai akan Kembali ke tahap input gambar dan jika gambar sudah sesuai maka akan melakukan proses klasifikasi oleh sistem, “Proses Klasifikasi” sistem melakukan proses ekstraksi fitur gambar, Prediksi kelas, Perhitungan probabilitas klasifikasi, “Hasil Nilai Akurasi” sistem akan menampilkan label hasil, Tingkat akurasi.

Coding.

Pendekatan Convolutional Neural Network (CNN), yang telah dibuat sebelumnya, digunakan untuk membuat arsitektur sistem. Pendekatan Extreme Programming, yang lebih sering digunakan untuk mengembangkan aplikasi dasar, digunakan untuk menulis skrip pengkodean dalam sistem yang memanfaatkan HTML.

Testing

Tahap ini memfokuskan pada pengujian fitur-fitur yang ada pada aplikasi sehingga tidak ada kesalahan (error).

```

Model: "sequential"
-----
Layer (type)                Output Shape                Param #
-----
xception (Functional)       (None, 7, 7, 2048)         20861480
conv2d_4 (Conv2D)           (None, 7, 7, 32)           589856
max_pooling2d (MaxPooling2D) (None, 3, 3, 32)           0
dropout (Dropout)           (None, 3, 3, 32)           0
flatten (Flatten)           (None, 288)                 0
dense (Dense)                (None, 4)                   1156
-----
Total params: 21,452,492
Trainable params: 591,012
Non-trainable params: 20,861,480
    
```

**Gambar 6. Model Summary**

Penjelasan gambar 6.

Tahap pengujian ini menggunakan arsitektur model (model summary) dari sebuah metode convolutional neural network (CNN) untuk klasifikasi gambar batik. Struktur model menggunakan Xception (pre-trained model) sebagai base

dengan ditambahkan layer tambahan yaitu : Conv2D dengan 32 filter, MaxPooling2D untuk reduksi dimensi, Dropout untuk mencegah overfitting, Flatten dan Dense (output layer dengan 4 neuron sesuai kelas batik). Total parameter: 21,452,492 (591,012 parameter yang dilatih ulang, sisa 20,861,480 parameter dari Xception dibekukan (*non-trainable*), dan menghasilkan bentuk bentuk tensor di setiap layer (None 7, 7, 2048) dari Xception.[12]

#### 4. HASIL DAN PEMBAHASAN

##### Hasil Training Model

Proses selanjutnya adalah melakukan *training* pada model. Sebelum melakukan *training* pada model, ditentukan terlebih dahulu *epochs* yang ingin digunakan. Penentuan *epochs* bisa bervariasi tergantung jumlah data dan arsitektur yang digunakan.[13]

```

Epoch 1/40
3/3 [=====] - 29s 8s/step - loss: 2.3874 - accuracy: 0.3313 - val_loss: 1.7331 - val_accuracy: 0.5000
Epoch 2/40
3/3 [=====] - 24s 7s/step - loss: 1.6693 - accuracy: 0.5875 - val_loss: 0.6407 - val_accuracy: 0.7500
Epoch 3/40
3/3 [=====] - 25s 7s/step - loss: 0.7773 - accuracy: 0.6375 - val_loss: 0.8464 - val_accuracy: 0.5500
Epoch 4/40
3/3 [=====] - 25s 7s/step - loss: 0.8200 - accuracy: 0.6875 - val_loss: 0.9247 - val_accuracy: 0.6000
Epoch 5/40
3/3 [=====] - 24s 10s/step - loss: 0.7798 - accuracy: 0.6875 - val_loss: 0.8335 - val_accuracy: 0.6000
Epoch 6/40
3/3 [=====] - 25s 7s/step - loss: 0.6752 - accuracy: 0.7188 - val_loss: 0.7720 - val_accuracy: 0.6000
Epoch 7/40
3/3 [=====] - 25s 7s/step - loss: 0.5831 - accuracy: 0.7812 - val_loss: 0.7263 - val_accuracy: 0.6500
Epoch 8/40
3/3 [=====] - 24s 10s/step - loss: 0.5120 - accuracy: 0.8000 - val_loss: 0.7920 - val_accuracy: 0.6500
Epoch 9/40
3/3 [=====] - 25s 7s/step - loss: 0.4481 - accuracy: 0.8125 - val_loss: 0.8637 - val_accuracy: 0.7000
Epoch 10/40
3/3 [=====] - 24s 10s/step - loss: 0.4019 - accuracy: 0.8938 - val_loss: 0.6811 - val_accuracy: 0.8000
Epoch 11/40
3/3 [=====] - 25s 7s/step - loss: 0.4060 - accuracy: 0.8625 - val_loss: 0.5934 - val_accuracy: 0.7500
Epoch 12/40
3/3 [=====] - 25s 10s/step - loss: 0.3324 - accuracy: 0.8687 - val_loss: 0.6055 - val_accuracy: 0.7500
Epoch 13/40
3/3 [=====] - 25s 7s/step - loss: 0.2341 - accuracy: 0.9250 - val_loss: 0.7200 - val_accuracy: 0.7500
Epoch 14/40
3/3 [=====] - 23s 7s/step - loss: 0.2529 - accuracy: 0.9002 - val_loss: 0.6802 - val_accuracy: 0.7500
Epoch 15/40
3/3 [=====] - 25s 7s/step - loss: 0.1862 - accuracy: 0.9312 - val_loss: 0.5898 - val_accuracy: 0.8000
Epoch 16/40
3/3 [=====] - 25s 10s/step - loss: 0.1931 - accuracy: 0.9438 - val_loss: 0.6329 - val_accuracy: 0.8000
Epoch 17/40
3/3 [=====] - 25s 7s/step - loss: 0.1823 - accuracy: 0.9438 - val_loss: 0.6080 - val_accuracy: 0.7500
Epoch 18/40
3/3 [=====] - 26s 10s/step - loss: 0.1575 - accuracy: 0.9438 - val_loss: 0.7861 - val_accuracy: 0.7500
Epoch 19/40
3/3 [=====] - 29s 8s/step - loss: 0.1600 - accuracy: 0.9500 - val_loss: 0.8226 - val_accuracy: 0.8000
Epoch 20/40
3/3 [=====] - 25s 7s/step - loss: 0.1193 - accuracy: 0.9563 - val_loss: 0.8295 - val_accuracy: 0.8000
    
```

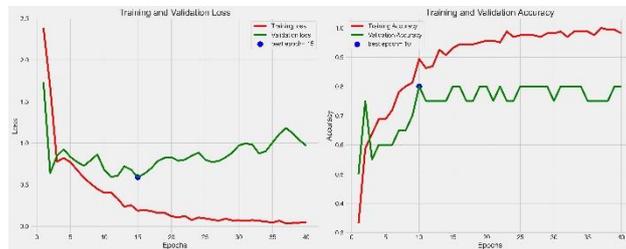
**Gambar 7. Hasil Model Training**

```

Epoch 23/40
3/3 [=====] - 266 18s/step - loss: 0.1068 - accuracy: 0.9563 - val_loss: 0.7874 - val_accuracy: 0.7500
Epoch 22/40
3/3 [=====] - 255 18s/step - loss: 0.1226 - accuracy: 0.9500 - val_loss: 0.8014 - val_accuracy: 0.8000
Epoch 23/40
3/3 [=====] - 255 7s/step - loss: 0.0735 - accuracy: 0.9875 - val_loss: 0.8474 - val_accuracy: 0.7500
Epoch 24/40
3/3 [=====] - 255 7s/step - loss: 0.1047 - accuracy: 0.9688 - val_loss: 0.8858 - val_accuracy: 0.7500
Epoch 25/40
3/3 [=====] - 255 18s/step - loss: 0.0934 - accuracy: 0.9750 - val_loss: 0.8006 - val_accuracy: 0.8000
Epoch 26/40
3/3 [=====] - 255 18s/step - loss: 0.0792 - accuracy: 0.9750 - val_loss: 0.7726 - val_accuracy: 0.8000
Epoch 27/40
3/3 [=====] - 255 18s/step - loss: 0.0668 - accuracy: 0.9750 - val_loss: 0.7839 - val_accuracy: 0.8000
Epoch 28/40
3/3 [=====] - 255 18s/step - loss: 0.0895 - accuracy: 0.9688 - val_loss: 0.8265 - val_accuracy: 0.8000
Epoch 29/40
3/3 [=====] - 255 7s/step - loss: 0.0701 - accuracy: 0.9812 - val_loss: 0.8837 - val_accuracy: 0.8000
Epoch 30/40
3/3 [=====] - 255 18s/step - loss: 0.0717 - accuracy: 0.9812 - val_loss: 0.9723 - val_accuracy: 0.7500
Epoch 31/40
3/3 [=====] - 266 7s/step - loss: 0.0646 - accuracy: 0.9875 - val_loss: 0.9561 - val_accuracy: 0.8000
Epoch 32/40
3/3 [=====] - 266 8s/step - loss: 0.0729 - accuracy: 0.9688 - val_loss: 0.9811 - val_accuracy: 0.8000
Epoch 33/40
3/3 [=====] - 266 7s/step - loss: 0.0643 - accuracy: 0.9875 - val_loss: 0.8747 - val_accuracy: 0.8000
Epoch 34/40
3/3 [=====] - 255 18s/step - loss: 0.0541 - accuracy: 0.9875 - val_loss: 0.9033 - val_accuracy: 0.8000
Epoch 35/40
3/3 [=====] - 255 7s/step - loss: 0.0440 - accuracy: 0.9875 - val_loss: 1.0076 - val_accuracy: 0.7500
Epoch 36/40
3/3 [=====] - 266 7s/step - loss: 0.0654 - accuracy: 0.9750 - val_loss: 1.1091 - val_accuracy: 0.7500
Epoch 37/40
3/3 [=====] - 255 18s/step - loss: 0.0325 - accuracy: 1.0000 - val_loss: 1.1806 - val_accuracy: 0.7500
Epoch 38/40
3/3 [=====] - 255 18s/step - loss: 0.0402 - accuracy: 0.9937 - val_loss: 1.1181 - val_accuracy: 0.7500
Epoch 39/40
3/3 [=====] - 266 7s/step - loss: 0.0398 - accuracy: 0.9937 - val_loss: 1.0363 - val_accuracy: 0.8000
Epoch 40/40
3/3 [=====] - 255 7s/step - loss: 0.0481 - accuracy: 0.9812 - val_loss: 0.9677 - val_accuracy: 0.8000
    
```

**Gambar 8. Hasil Model Training**

Hasil akurasi *training* dan validasi dapat dilihat pada Gambar berikut



**Gambar 9. Analisis Loss dan Akurasi Training Validasi**

```

1/1 [=====] - 2s 2s/step - loss: 0.6259 - accuracy: 0.8500
Accuracy on the test set is 85.00%
    
```

**Gambar 10. Hasil Akurasi Nilai**

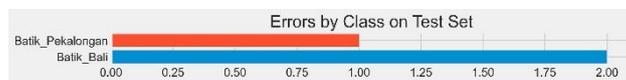
Penjelasan gambar 9 dan 10.

Analisis Loss dan Akurasi Training-Validasi, Grafik menunjukkan dua matriks kunci selama 40 epoch. **Nilai Loss** training terus menurun secara konsisten, Loss validasi mencapai titik terendah di epoch 15, Setelah epoch 15, loss validasi mulai naik sementara loss training terus turun. **Nilai Akurasi** training terus membaik, Akurasi validasi mencapai puncak di epoch 10 (lebih awal dibanding loss minimum), Jarak antara akurasi training dan validasi melebar setelah epoch 10.

Hasil dari matriks nilai loss sebesar 0,6259 dan nilai akurasi sebesar 0,8500. Jadi model sudah menunjukkan potensi yang bagus, Tetapi masih membutuhkan proses penyesuaian untuk mencegah *Overfitting*.

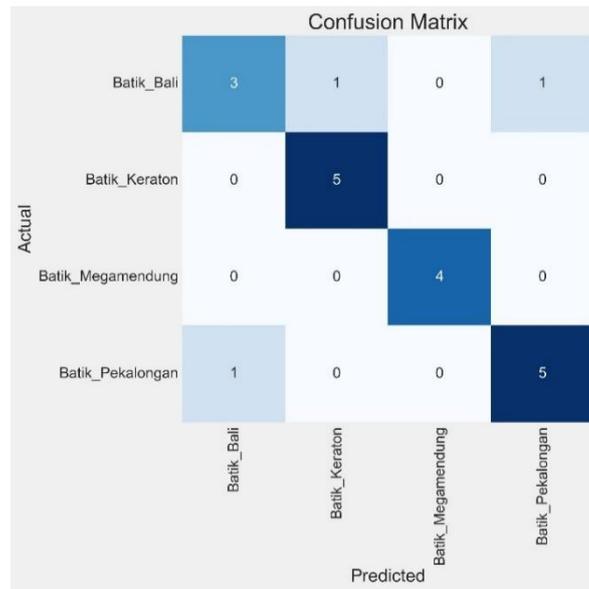
**Pengujian Model**

Confusion Matriks merupakan sebuah tabel yang di gunakan untuk proses evaluasi model klasifikasi dengan cara membandingkan model prediksi terhadap nilai aktual.[12]



**Gambar 11. Nilai Error**

Pada tahap ini dilakukan pengujian untuk menguji tingkat akurasi arsitektur VGG-16 dan label prediksi pada sistem yang telah dibuat.



**Gambar 12.** Confusion Matrix

Penjelasan gambar 12.

Confusion matrix tersebut mengevaluasi performa model klasifikasi dalam mengidentifikasi empat jenis batik, yaitu Batik\_Bali, Batik\_Keraton, Batik\_Megamendung, dan Batik\_Pekalongan. Matrix ini membandingkan prediksi model dengan label sebenarnya, di mana angka pada diagonal utama menunjukkan prediksi yang benar, sedangkan angka di luar diagonal menunjukkan kesalahan klasifikasi. Dari data tersebut, terlihat bahwa model memiliki akurasi tinggi untuk Batik\_Keraton, Batik\_Megamendung, dan Batik\_Pekalongan dengan prediksi yang sempurna, kecuali satu kesalahan pada Batik\_Pekalongan yang terdeteksi sebagai Batik\_Bali. Namun, model mengalami kesulitan dalam mengklasifikasikan Batik\_Bali, dengan dua kesalahan prediksi. Satu sampel Batik\_Bali salah diklasifikasikan sebagai Batik\_Keraton, dan satu lagi sebagai Batik\_Pekalongan. Hal ini menunjukkan bahwa model mungkin kurang mampu membedakan ciri khas Batik\_Bali dari kedua jenis batik lainnya. Kesalahan ini dapat disebabkan oleh kesamaan pola atau warna antara ketiga jenis batik tersebut, atau kurangnya data latih untuk Batik\_Bali. Secara keseluruhan, confusion matrix ini memberikan gambaran yang jelas tentang kelebihan dan kelemahan model. Untuk meningkatkan performa, disarankan untuk menambah jumlah data latih khususnya untuk Batik\_Bali, serta melakukan optimasi pada fitur ekstraksi seperti tekstur atau pola warna. Dengan demikian, model dapat belajar lebih efektif untuk membedakan Batik\_Bali dari jenis batik lainnya, sehingga mengurangi kesalahan klasifikasi.

Classification Report adalah laporan evaluasi yang digunakan untuk mengukur performa model klasifikasi dalam machine learning.[14]Matriks yang dapat mengukur kinerja model dapat dihitung menggunakan data Matriks Kebingungan; ini dikenal sebagai Laporan Klasifikasi. Matriks ini digunakan: Precision: Menentukan tingkat akurasi antara hasil prediksi model dan data yang diberikan. Recall: Menjelaskan seberapa baik model menemukan kembali informasi. Presisi dan ingatan rata-rata tertimbang dibandingkan menggunakan F1-Score. Support: untuk penilaian.[15]

```

Classification Report:
-----
              precision    recall  f1-score   support

   Batik_Bali         0.75      0.60      0.67         5
   Batik_Keraton       0.83      1.00      0.91         5
   Batik_Megamendung   1.00      1.00      1.00         4
   Batik_Pekalongan    0.83      0.83      0.83         6

 accuracy                   0.85         20
 macro avg                  0.85         20
 weighted avg               0.85         20

```

**Gambar 13.** Klasifikasi Report

Penjelasan gambar 13.

Grafik pertama menunjukkan distribusi error pada data testing untuk setiap kelas batik. Dari grafik terlihat bahwa kelas Batik Bali memiliki error yang lebih tinggi dibanding kelas lainnya, dengan nilai error mendekati 1.5. Sementara itu, kelas Batik Keraton dan Megamendung menunjukkan error yang sangat rendah, mendekati 0. Ini mengindikasikan bahwa model mengalami kesulitan khusus dalam mengklasifikasikan Batik Bali dengan benar, sementara untuk kelas lainnya performanya cukup baik. klasifikasi memberikan metrik performa yang lebih rinci untuk setiap kelas. Model mencapai akurasi keseluruhan sebesar 85%, dengan performa terbaik pada kelas Batik Megamendung yang mendapatkan precision, recall, dan F1-score sempurna (1.00). Kelas Batik Keraton juga menunjukkan performa sangat baik dengan recall 1.00 dan F1-score 0.91. Namun, kelas Batik Bali menjadi titik lemah dengan recall terendah (0.60) dan F1-score 0.67, menunjukkan bahwa 40% sampel Batik Bali tidak terdeteksi dengan benar.

Model melakukan prediksi sempurna untuk kelas Keraton (5 benar dari 5) dan Megamendung (4 benar dari 4). Namun untuk kelas Bali, dari 5 sampel hanya 3 yang benar, dengan 2 salah diklasifikasikan sebagai Keraton. Kelas Pekalongan juga menunjukkan 1 kesalahan dari 6 sampel, dimana 1 sampel diklasifikasikan sebagai Bali. Dari proses klasifikasi model telah menunjukkan performa yang baik dengan menghasilkan nilai keakuratan sekitar 85%.

## Implementasi

Telah dilakukan pengembangan website dengan menggunakan Visual Studio Code untuk website klasifikasi citra batik nusantara. Berikut merupakan beberapa tampilan website yang telah dibuat dan dikembangkan oleh peneliti.

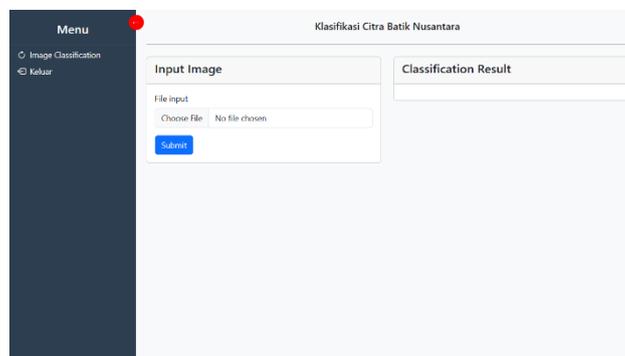
- Tampilan home dari proses Klasifikasi Batik Nusantara



**Gambar 14.** Tampilan Home

Pada Gambar 14 menunjukkan tampilan awal atau beranda dari sistem klasifikasi batik nusantara dirancang sebagai antarmuka utama yang informatif dan interaktif. Untuk memberikan gambaran menyeluruh kepada pengguna mengenai kekayaan motif batik dari berbagai daerah di Nusantara, serta memberikan navigasi menuju proses klasifikasi yang lebih mendalam.

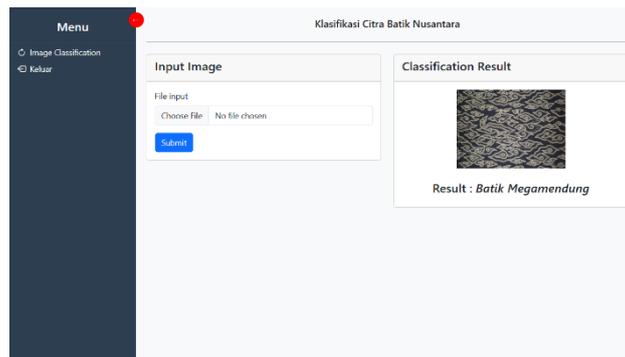
- Tampilan halaman proses klasifikasi Batik Nusantara



**Gambar 15.** Proses Klasifikasi Batik

Pada Gambar 15 menunjukkan tampilan proses klasifikasi citra batik yang memanfaatkan teknologi pembelajaran mendalam (deep learning), khususnya Convolutional Neural Network (CNN), untuk mengenali dan mengklasifikasikan motif batik berdasarkan gambar yang diunggah oleh pengguna.

- Tampilan halaman hasil klasifikasi Batik Nusantara



**Gambar 16.** Hasil Klasifikasi Batik Nusantara

Pada Gambar 16 menunjukkan tampilan lanjutan dari proses klasifikasi citra batik, yang berfungsi untuk menampilkan hasil identifikasi dari gambar batik yang diunggah oleh pengguna. Pada contoh tampilan ini, sistem berhasil mengenali gambar sebagai Batik Megamendung, yang merupakan motif khas dari Cirebon.

## 5. KESIMPULAN

Model klasifikasi batik mencapai akurasi 85%, menunjukkan kemampuan yang cukup baik dalam mengidentifikasi empat jenis batik. Kelas Megamendung dan Keraton memiliki performa sempurna dengan precision, recall, dan F1-score mendekati 1.00. Namun, kelas Bali menjadi titik lemah utama dengan recall hanya 60%, mengindikasikan 40% sampel Batik Bali salah diklasifikasikan, terutama sebagai Keraton.

Analisis confusion matrix mengungkapkan pola kesalahan spesifik dimana model sering salah membedakan Batik Bali dengan Keraton (2 dari 5 sampel salah) dan Batik Pekalongan dengan Bali (1 dari 6 sampel salah). Kemiripan visual motif dan kemungkinan ketidakseimbangan data diduga menjadi penyebab utama kesalahan klasifikasi ini.

## DAFTAR PUSTAKA

- [1] D. Q. A'yun, M. Y. Tanzil, and F. R. Julia, "Pengaplikasian elemen desain pada motif batik mega mendung," *Semin. Nas. Envisi Ind. Kreat.*, pp. 118–127, 2020.
- [2] M. Malika and E. Widodo, "Implementasi Deep Learning Untuk Klasifikasi Gambar Menggunakan Convolutional Neural Network (Cnn) Pada Batik Sasambo," *Pattimura Proceeding Conf. Sci. Technol.*, pp. 335–340, 2022, doi: 10.30598/pattimurasci.2021.knmxx.335-340.
- [3] M. H. V. Sinaga, M. Albirra, and M. F. Sidiq, "Klasifikasi Gambar Pemandangan dengan Kecerdasan Buatan Berbasis CNN," *J. JTIK (Jurnal Teknol. Inf. dan Komunikasi)*, vol. 8, no. 2, pp. 412–417, 2024, doi: 10.35870/jtik.v8i2.1424.
- [4] I. Gilang Perwati, N. Suarna, and T. Suprapti, "Analisis Klasifikasi Gambar Bunga Lily

- Menggunakan Metode Convolutional Neural Network (Cnn) Dalam Pengolahan Citra,” *JATI (Jurnal Mhs. Tek. Inform.*, vol. 8, no. 3, pp. 2908–2915, 2024, doi: 10.36040/jati.v8i3.9193.
- [5] R. H. Alfikri, M. S. Utomo, H. Februariyanti, and E. Nurwahyudi, “Pembangunan Aplikasi Penerjemah Bahasa Isyarat Dengan Metode Cnn Berbasis Android,” *J. Teknoinfo*, vol. 16, no. 2, p. 183, 2022, doi: 10.33365/jti.v16i2.1752.
- [6] A. Zalvadila, “Klasifikasi Penyakit Tanaman Bawang Merah Menggunakan Metode SVM dan CNN,” *J. Inform. J. Pengemb. IT*, vol. 8, no. 3, pp. 255–260, 2023, doi: 10.30591/jpit.v8i3.5341.
- [7] H. Herdianto and D. Nasution, “Implementasi Metode Cnn Untuk Klasifikasi Objek,” *METHOMIKA J. Manaj. Inform. dan Komputerisasi Akunt.*, vol. 7, no. 1, pp. 54–60, 2023, doi: 10.46880/jmika.vol7no1.pp54-60.
- [8] U. S. Rahmadhani and N. L. Marpaung, “Klasifikasi Jamur Berdasarkan Genus Dengan Menggunakan Metode CNN,” *J. Inform. J. Pengemb. IT*, vol. 8, no. 2, pp. 169–173, 2023, doi: 10.30591/jpit.v8i2.5229.
- [9] Rizky Prabowo, Azzah Roudhoh, and Afifah, “Klasifikasi Image Tumbuhan Obat Sirih dan Binahong Menggunakan Metode Convolutional Neural Network (CNN),” *J. Komputasi*, vol. 10, no. 2, pp. 48–58, 2022.
- [10] Verdy and Ery Hartati, “Klasifikasi Penyakit Mata Menggunakan Convolutional Neural Network Model Resnet-50,” *J. Rekayasa Sist. Inf. dan Teknol.*, vol. 1, no. 3, pp. 199–206, 2024, doi: 10.59407/jrsit.v1i3.529.
- [11] R. Soekarta, N. Nurdjan, and A. Syah, “Klasifikasi Penyakit Tanaman Tomat Menggunakan Metode Convolutional Neural Network (CNN),” *Insect (Informatics Secur. J. Tek. Inform.*, vol. 8, no. 2, pp. 143–151, 2023, doi: 10.33506/insect.v8i2.2356.
- [12] R. Soekarta *et al.*, “IMPLEMENTASI DEEP LEARNING UNTUK DETEKSI JENIS OBAT proses kehidupan dan merupakan senyawa tubuh ( Prabowo , 2021 ). Saat ini berbagai jenis dibedakan beberapa bentuk sediaan obat . Dalam lingkungan farmasi yang untuk automatisasi dalam pengelolaan obat s,” vol. 7, no. 4, pp. 455–464, 2023.
- [13] M. Khoiruddin, A. Junaidi, and W. A. Saputra, “Klasifikasi Penyakit Daun Padi Menggunakan Convolutional Neural Network,” *J. Dinda Data Sci. Inf. Technol. Data Anal.*, vol. 2, no. 1, pp. 37–45, 2022, doi: 10.20895/dinda.v2i1.341.
- [14] Anissa Ollivia Cahya Pratiwi, “Klasifikasi Jenis Anggur Berdasarkan Bentuk Daun Menggunakan Convolutional Neural Network Dan K-Nearest Neighbor,” *J. Ilm. Tek. Inform. dan Komun.*, vol. 3, no. 2, pp. 201–224, 2023, doi: 10.55606/juitik.v3i2.535.
- [15] M. I. Syahputra and A. T. Wibowo, “Klasifikasi Genus Tanaman Anggrek berdasarkan Citra Kuntum Bunga Menggunakan Metode Convolutional Neural Network (CNN),” *e-Proceeding Eng.*, vol. 7, no. 2, pp. 8015–8023, 2020, [Online]. Available: <https://www.programmersought.com/article/3724355693/>