

Klasifikasi Jenis Tumbuhan Batang Daun dan Buah

Rotua Eka Wati Br. Sitorus¹, David Batee², Grace Helena Hutagaol³, Mila Rachmika⁴, David Christian⁵, Frans Mikael Sinaga^{*6}

^{1,2,3,4,5,6}Universitas Mikroskil, Jln. M. H. Thamrin No. 112, 124, 140, (061) 4573767

^{1,2,3,4,5,6}Fakultas Informatika, Teknik Informatika, Universitas Mikroskil, Medan

e-mail: ¹211111870@students.mikroskil.ac.id, ²211111930@students.mikroskil.ac.id,

³211110948@students.mikroskil.ac.id, ⁴211111750@students.mikroskil.ac.id,

⁵211112742@students.mikroskil.ac.id, ⁶frans.sinaga@mikroskil.ac.id

Dikirim: 04-02-2025 | Diterima: 10-04-2025 | Diterbitkan: 30-04-2025

Abstrak

Indonesia memiliki keanekaragaman tumbuhan yang sangat luas, namun identifikasi dan klasifikasi bagian tumbuhan seperti daun, batang, dan buah secara manual sering kali memakan waktu dan rentan terhadap kesalahan. Penelitian ini mengembangkan sistem klasifikasi berbasis Artificial Neural Network (ANN) untuk otomatisasi identifikasi. Dengan menggunakan 6.707 citra tumbuhan, model ANN dilatih dengan fitur visual seperti bentuk, tekstur, dan warna. Hasilnya menunjukkan potensi besar ANN dalam meningkatkan efisiensi dan akurasi identifikasi tumbuhan. Sistem ini diimplementasikan dalam aplikasi web, yang mendukung penelitian botani, pertanian, dan pemantauan lingkungan.

Kata kunci: *Artificial Neural Network*, klasifikasi tumbuhan, identifikasi visual, *preprocessing*, aplikasi berbasis web

Abstract

Indonesia has a vast diversity of plants, but the manual identification and classification of plant parts such as leaves, stems, and fruits are often time-consuming and prone to errors. This research develops an Artificial Neural Network (ANN)-based classification system to automate identification. Using a dataset of 6,707 plant images, the ANN model is trained with visual features such as shape, texture, and color. The results demonstrate the great potential of ANN in improving the efficiency and accuracy of plant identification. This system is implemented in a web application, supporting botanical research, agriculture, and environmental monitoring.

Keywords: *Artificial Neural Network, plant classification, visual identification, preprocessing, web-based application*

1. PENDAHULUAN

Indonesia memiliki keanekaragaman tumbuhan yang sangat luas, mulai dari tanaman pangan hingga tanaman hias. Salah satu tantangan dalam bidang botani dan pertanian adalah bagaimana mengidentifikasi serta mengklasifikasikan bagian-bagian tumbuhan seperti daun, batang, dan buah secara otomatis. Klasifikasi ini sangat penting untuk mendukung penelitian, pemantauan lingkungan, serta pengelolaan sumber daya alam. Namun, proses identifikasi secara manual sering kali membutuhkan waktu yang lama, keahlian khusus, dan tidak selalu menghasilkan akurasi yang tinggi.

Algoritma Neural Network telah terbukti efektif dalam klasifikasi citra berdasarkan fitur-fitur tertentu. Dengan memanfaatkan kekuatan pemrosesan paralel dan kemampuan pembelajaran mesin, Neural Network mampu mengenali pola-pola kompleks dalam citra, termasuk fitur-fitur yang unik untuk membedakan jenis tumbuhan mangga[1]. ANN merupakan metode pembelajaran mesin yang dapat mengenali pola dalam data, termasuk gambar, melalui proses pelatihan. Dalam konteks klasifikasi tumbuhan, ANN dapat digunakan untuk membedakan bagian-bagian tumbuhan berdasarkan karakteristik visualnya seperti bentuk, tekstur, dan warna. Dengan adanya pelatihan pada dataset gambar daun, batang, dan buah, ANN dapat mempelajari pola-pola spesifik yang kemudian diterapkan dalam mengidentifikasi citra baru secara akurat. Pendekatan ini menjanjikan peningkatan efisiensi dan akurasi dibandingkan metode konvensional, dengan kemampuan adaptasi dan pembelajaran yang lebih baik dalam menangani variasi data yang kompleks.

Penggunaan ANN dalam klasifikasi bagian tumbuhan ini diharapkan dapat meningkatkan efisiensi serta mengurangi tingkat kesalahan manusia dalam proses pengenalan visual[2]. Selain itu, penerapan teknologi ini juga berpotensi untuk dimanfaatkan dalam berbagai sektor, seperti penelitian ilmiah, pertanian, dan pemantauan ekosistem. Dengan mengembangkan sistem klasifikasi berbasis ANN, penelitian ini berkontribusi dalam mempercepat inovasi dalam bidang botani dan ilmu pengetahuan alam.

Penelitian ini bertujuan untuk mengembangkan sistem klasifikasi otomatis bagian-bagian tumbuhan menggunakan ANN agar dapat meningkatkan akurasi dan efisiensi dalam identifikasi daun[3], batang, dan buah. Kontribusi utama dari penelitian ini adalah pengembangan model ANN yang dapat mengklasifikasikan bagian tumbuhan dengan lebih akurat, penyediaan dataset citra tumbuhan untuk pelatihan dan pengujian, serta optimalisasi parameter ANN[4] guna meningkatkan performa klasifikasi. Selain itu, penelitian ini juga bertujuan untuk menyajikan metode klasifikasi berbasis pembelajaran mesin yang dapat diterapkan dalam berbagai bidang, terutama botani dan pertanian serta menunjukkan keunggulan ANN dalam klasifikasi bagian tumbuhan dibandingkan dengan metode lain, dengan menekankan pada kemampuan pembelajaran mendalam dan adaptasi model terhadap variasi data yang luas.

Penelitian sebelumnya telah mengeksplorasi pengguna kecerdasan buatan dalam klasifikasi citra tumbuhan, seperti penggunaan Convolutional Neural Network (CNN) untuk klasifikasi daun tanaman. Namun klasifikasi bagian tumbuhan seperti daun, batang, dan buah menggunakan ANN masih kurang dieksplorasi. Penelitian ini memperluas cakupan dengan fokus pada klasifikasi bagian tumbuhan yang spesifik, dengan memanfaatkan ANN untuk menangani tantangan dalam identifikasi daun, batang, dan buah. Pendekatan ini diharapkan memberikan kontribusi signifikan dalam pengembangan teknologi klasifikasi otomatis di bidang botani dan pertanian dengan penekanan pada kemampuan adaptasi dan pembelajaran mendalam ANN dalam menangani variasi data yang kompleks.

2. TINJAUAN PUSTAKA

2.1 Klasifikasi Tumbuhan Menggunakan Deep Learning

Klasifikasi tumbuhan adalah proses pengelompokan tumbuhan berdasarkan karakteristik morfologi dan genetiknya. Dengan kemajuan teknologi, metode deep learning, khususnya Artificial Neural Network (ANN), telah banyak digunakan untuk meningkatkan akurasi dan efisiensi dalam klasifikasi tumbuhan. ANN mampu mengenali pola-pola spesifik pada gambar tumbuhan dengan mendeteksi fitur seperti bentuk, tekstur, dan warna. [5] mengembangkan algoritma deep learning dengan arsitektur ANN untuk mengenali jenis daun berdasarkan citra yang diberikan, menunjukkan efektivitas pendekatan ini dalam klasifikasi tumbuhan.

2.2 Pengenalan Pola pada Citra Daun

Pengenalan pola adalah cabang kecerdasan buatan yang bertujuan untuk mengidentifikasi pola dalam data. Dalam konteks klasifikasi tumbuhan, pengenalan pola pada citra daun menjadi fokus

utama.[6] mengimplementasikan deep learning pada identifikasi jenis tumbuhan berdasarkan citra daun menggunakan Artificial Neural Network (ANN), menunjukkan bahwa metode ini efektif dalam mengenali pola-pola pada citra daun untuk klasifikasi tumbuhan.

2.3 Implementasi Deep Learning dalam Klasifikasi

Implementasi deep learning dalam klasifikasi buah telah menunjukkan hasil yang menjanjikan.[7] menggunakan algoritma Artificial Neural Network (ANN) untuk mengklasifikasikan berbagai jenis buah berdasarkan citra, menunjukkan bahwa metode ini efektif dalam klasifikasi buah.

2.4 Transfer Learning untuk Identifikasi Daun Tanaman Obat

Transfer learning adalah teknik dalam deep learning di mana model yang telah dilatih pada satu tugas digunakan kembali pada tugas lain yang serupa.[8] menerapkan transfer learning menggunakan arsitektur MobileNetV2 untuk identifikasi daun tanaman obat, menunjukkan bahwa pendekatan ini dapat meningkatkan akurasi dan efisiensi dalam klasifikasi tumbuhan.

3. METODE PENELITIAN

Sistem berbasis web ini dirancang untuk mengklasifikasikan gambar ke dalam tiga kategori: Batang, Buah, dan Daun, menggunakan model Artificial Neural Network (ANN). Sistem terdiri dari beberapa komponen utama, yaitu frontend sebagai antarmuka pengguna, backend yang mengelola pemrosesan data dan model ANN, serta dataset gambar yang digunakan untuk pelatihan model.

Dataset yang digunakan berjumlah 6.707 gambar, yang telah melalui berbagai tahap preprocessing, termasuk rescaling, augmentasi citra, dan pembagian data untuk pelatihan dan validasi. Model ANN yang diterapkan dilatih menggunakan TensorFlow dan Keras dengan beberapa lapisan tersembunyi dan fungsi aktivasi softmax untuk menghasilkan probabilitas klasifikasi.

Alur kerja sistem dimulai dari pengguna mengunggah gambar, backend melakukan preprocessing dan prediksi menggunakan model ANN, lalu hasil klasifikasi ditampilkan kepada pengguna dalam bentuk teks dan probabilitas. Dengan pendekatan ini, sistem diharapkan dapat membantu mengotomatisasi klasifikasi bagian tumbuhan dan dengan tingkat akurasi yang tinggi serta kemudahan penggunaan melalui antarmuka web yang responsif dan interaktif.

3.1 Desain Sistem

Desain sistem berbasis web ini dirancang untuk mengklasifikasikan gambar tumbuhan menjadi tiga kategori: Batang, Buah, dan Daun. Web ini akan memungkinkan pengguna mengunggah gambar bagian tumbuhan, yang kemudian akan diproses oleh model Artificial Neural Network (ANN) yang telah dilatih untuk melakukan klasifikasi berdasarkan kategori yang telah ditentukan. Berikut adalah penjelasan lebih detail tentang desain sistem berbasis web ini:

1. Arsitektur Sistem

Sistem berbasis web ini terdiri dari beberapa komponen utama:

- a. Frontend (User Interface): Bagian antarmuka pengguna yang memungkinkan interaksi dengan pengguna.
- b. Backend (Server dan API): Bagian yang menangani pemrosesan data dan logika bisnis, termasuk pengolahan gambar dan klasifikasi menggunakan model ANN.
- c. Model ANN: Model machine learning yang dilatih untuk mengklasifikasikan gambar tumbuhan berdasarkan kategori yang ditentukan.

- d. Database (Opsional): Untuk menyimpan data gambar yang diunggah oleh pengguna (misalnya, untuk analisis lebih lanjut atau log aktivitas).
2. Frontend (User Interface)

Frontend akan dibangun menggunakan HTML, CSS, dan JavaScript untuk memberikan antarmuka yang mudah digunakan oleh pengguna. Komponen utama frontend meliputi:

 - a. Form Upload Gambar:
 - i. Pengguna dapat mengunggah gambar bagian tumbuhan dalam format gambar yang umum (JPG, PNG, etc.).
 - ii. Gambar yang diunggah akan ditampilkan di antarmuka untuk memastikan gambar yang diunggah sesuai dengan input yang diinginkan.
 - b. Tampilan Hasil Klasifikasi:

Setelah gambar diproses, hasil klasifikasi akan ditampilkan kepada pengguna dalam bentuk teks yang menjelaskan hasil prediksi (misalnya, "Gambar ini termasuk dalam kategori Daun") dan Menampilkan probabilitas masing-masing kelas (Batang, Buah, Daun) yang diperoleh dari model ANN.
 - c. Desain Responsif
Antarmuka dioptimalkan untuk digunakan di berbagai perangkat, seperti desktop, tablet, dan smartphone, dengan menggunakan CSS Grid dan Flexbox.
 - d. Interaksi Pengguna
Pengguna dapat mengunggah gambar melalui tombol upload gambar. Setelah gambar diproses, hasil klasifikasi ditampilkan di halaman yang sama dengan tombol Klasifikasi gambar lainnya untuk mengunggah gambar baru.
3. Backend (Server dan API)

Backend sistem ini akan dibangun menggunakan Node.js sebagai framework web untuk menangani routing dan komunikasi antara frontend dan model ANN. Bagian-bagian utama dari backend adalah sebagai berikut:

 - a. Server dan Routing:
 - i. Express akan digunakan untuk membuat server yang menangani request dari pengguna.
 - ii. Akan ada rute (route) seperti /upload untuk menerima gambar yang diunggah oleh pengguna dan classify untuk memproses gambar dan mengembalikan hasil klasifikasi.
 - b. API untuk Pemrosesan Gambar:

Ketika pengguna mengunggah gambar, backend akan menerima gambar dan memrosesnya.
 - c. Berikut Penerapan Model ANN:
 - i) Backend akan memuat model ANN yang telah dilatih di Google Colab atau Jupyter Notebook. Model ini dapat disimpan dalam format H5 atau SavedModel untuk dapat dipakai di server.
 - ii) Gambar yang diunggah akan diproses, diubah ukurannya menjadi 150x150 piksel, dan dinormalisasi sebelum dimasukkan ke dalam model ANN.
 - iii) Hasil prediksi dari model akan dikirim kembali ke frontend dalam bentuk JSON, yang kemudian ditampilkan di antarmuka pengguna
 - d. Libraries dan Framework:

TensorFlow.js dapat digunakan untuk menjalankan model ANN langsung di dalam aplikasi web, atau jika model sudah dilatih menggunakan TensorFlow atau Keras, backend dapat menggunakan mongodb

4. Model ANN

Model Artificial Neural Network (ANN) akan digunakan untuk melakukan klasifikasi gambar. Model ini sudah dilatih dengan dataset gambar yang terbagi menjadi tiga kategori: Batang, Buah, dan Daun.

a. Pelatihan Model:

- i. Model dilatih dengan Keras menggunakan TensorFlow di backend untuk memproses gambar bagian tumbuhan.
- ii. Model memiliki beberapa lapisan tersembunyi (hidden layers) dan softmax activation function di lapisan output untuk mengklasifikasikan gambar ke dalam salah satu dari tiga kelas.

b. Memprediksi Gambar Baru:

- i. Setelah model dilatih dan disimpan, gambar yang diunggah oleh pengguna akan diproses dan dimasukkan ke dalam model untuk menghasilkan klasifikasi.
- ii. Probabilitas setiap kelas (Batang, Buah, Daun) akan dihitung, dan kelas dengan probabilitas tertinggi akan dipilih sebagai hasil klasifikasi.

5. Proses Klasifikasi

Proses klasifikasi dimulai ketika pengguna mengunggah gambar ke server melalui formulir di frontend. Berikut adalah alur kerjanya:

a. Pengunggahan Gambar:

Pengguna memilih gambar dari perangkat mereka dan mengklik tombol upload.

b. Proses Gambar di Backend:

- i. Backend menerima gambar yang diunggah dan mengubah ukurannya menjadi 150 x 150 piksel (sesuai dengan ukuran input yang diperlukan oleh model).
- ii. Gambar kemudian dinormalisasi untuk mencocokkan rentang nilai yang digunakan selama pelatihan model.

c. Prediksi dengan Model ANN:

- i. Gambar yang telah diproses dimasukkan ke dalam model ANN yang telah dilatih.
- ii. Model menghasilkan probabilitas untuk masing-masing kategori (Batang, Buah, Daun).

d. Kembali ke Frontend:

- i. Hasil klasifikasi yang berisi label kelas (misalnya, "Batang") dan probabilitasnya (misalnya, 80% untuk Batang, 10% untuk Buah, dan 10% untuk Daun) dikirimkan ke frontend.
- ii. Hasil ini ditampilkan di antarmuka pengguna dengan cara yang mudah dimengerti.

3.2 Dataset

Dataset ini digunakan untuk klasifikasi bagian-bagian tumbuhan (Batang, Daun, Buah) dengan menggunakan citra gambar. Dataset ini diambil berdasarkan hasil penelitian team kami yang langsung mengambil data langsung atau realnya, dan totalnya 6707 gambar yang dibagi ke dalam tiga kategori

kelas utama: Batang, Daun, dan Buah. Setiap gambar dalam dataset ini menggambarkan salah satu bagian tanaman tersebut dan diberi label sesuai dengan kategori yang relevan.

1. Kelas 1: Batang – Gambar-gambar yang menggambarkan bagian batang dari tanaman.
2. Kelas 2: Daun – Gambar-gambar yang menggambarkan bagian daun dari tanaman.
3. Kelas 3: Buah – Gambar-gambar yang menggambarkan bagian buah dari tanaman

```
📁 Batang Directory: 2465 images
Sample images: ['frame_0000.jpg', 'frame_0001.jpg', 'frame_0002.jpg', 'frame_0003.jpg', 'frame_0004.jpg']

Buah Directory: 2421 images
Sample images: ['frame_0000.jpg', 'frame_0001.jpg', 'frame_0002.jpg', 'frame_0003.jpg', 'frame_0004.jpg']

Daun Directory: 1821 images
Sample images: ['frame_0000.jpg', 'frame_0001.jpg', 'frame_0002.jpg', 'frame_0003.jpg', 'frame_0004.jpg']

Total number of images in all directories: 6707
```

Gambar 1 Dataset

3.3 Preprocessing

Preprocessing data adalah tahap penting dalam membangun model Artificial Neural Network (ANN), karena memastikan bahwa data yang digunakan dalam pelatihan sudah dalam bentuk yang sesuai agar model dapat belajar dengan efektif. Untuk dataset gambar yang digunakan dalam proyek klasifikasi bagian tumbuhan ini, ada beberapa langkah preprocessing yang dilakukan:

1. Rescaling (Skalasi Gambar)
 - a. Semua gambar di-rescale agar berada dalam rentang nilai [0, 1].
 - b. Rescaling dilakukan dengan membagi setiap nilai piksel gambar dengan 255 (nilai maksimum piksel pada gambar 8-bit). Hal ini berguna untuk mempercepat konvergensi model karena nilai piksel yang lebih kecil dapat membantu model dalam mengupdate bobot dengan lebih efisien.
2. Augmentasi Citra (Image Augmentation)

Augmentasi citra digunakan untuk memperkaya dataset dengan menciptakan variasi dari gambar yang sudah ada. Teknik ini membantu model untuk tidak terlalu bergantung pada data pelatihan tertentu, yang dapat mengurangi risiko overfitting. Berikut adalah teknik augmentasi yang diterapkan:

 - a. Rotation Range: Memutar gambar dengan rentang sudut tertentu untuk memberikan variasi orientasi gambar.
 - b. Horizontal Flip: Membalik gambar secara horizontal, yang membantu model untuk mengenali objek terlepas dari orientasi kiri atau kanan.
 - c. Shear Range: Memberikan distorsi citra untuk menguji model dengan gambar yang sedikit terdistorsi.
 - d. Zooming: Mengubah ukuran gambar secara acak untuk memberi model pengalaman dengan gambar yang lebih besar atau lebih kecil.
 - e. Vertical Flip: Membalik gambar secara vertikal (lebih banyak digunakan untuk gambar tumbuhan).

- f. Fill Mode: Digunakan untuk mengisi area kosong saat melakukan rotasi atau distorsi citra.

```
✓ 0s ▶ # augmentation and generate image data
train_datagen = ImageDataGenerator(
    rescale = 1./255,
    rotation_range = 20,
    horizontal_flip = True,
    shear_range = 0.2,
    fill_mode = 'nearest',
)
test_datagen = ImageDataGenerator(
    rescale = 1./255,
    rotation_range = 20,
    horizontal_flip = True,
    vertical_flip = True,
    shear_range = 0.2,
    fill_mode = 'nearest'
)
```

Gambar 2 Augmentasi Citra

3. Pembagian Data (Train and Validation Split)

Pembagian data dilakukan dengan tujuan untuk menguji dan melatih model secara terpisah. Ini memastikan bahwa model tidak melihat data yang sama baik untuk pelatihan maupun evaluasi, yang dapat mengakibatkan overfitting.

- Train-Validation Split: Data pelatihan dibagi menjadi 80% untuk pelatihan dan 20% untuk validasi. Pembagian ini memungkinkan model untuk mempelajari pola dari data pelatihan dan diuji dengan data validasi yang belum pernah dilihat sebelumnya.
- Pembagian dilakukan menggunakan fungsi `train_test_split` dari `sklearn`, di mana 40% gambar dari setiap kategori (Batang, Daun, Buah) digunakan untuk validasi dan 60% digunakan untuk pelatihan.

```
▶ # train dan test split
# memecah setiap direktori menjadi data train dan data validasi(validation 40% of dataset)
train_roc_dir, val_roc_dir = train_test_split(os.listdir(roc_dir), test_size = 0.40)
train_pap_dir, val_pap_dir = train_test_split(os.listdir(pap_dir), test_size = 0.40)
train_sci_dir, val_sci_dir = train_test_split(os.listdir(sci_dir), test_size = 0.40)
```

Gambar 3 Train and Validation Split

4. Penggunaan Image Generator

- `ImageDataGenerator` digunakan untuk memudahkan proses augmentasi dan rescaling gambar. Objek ini memungkinkan aliran gambar secara langsung ke dalam model selama pelatihan tanpa harus memuat seluruh dataset ke dalam memori.
- Data pelatihan dan validasi digenerasi menggunakan metode `flow from directory`, di mana gambar dibaca secara otomatis dari direktori yang telah diatur sebelumnya. Fungsi ini juga

memungkinkan untuk menetapkan ukuran target gambar (misalnya, 150x150 piksel) dan ukuran batch.

```
▶ train_generator = train_datagen.flow_from_directory(  
    train_dir,  
    target_size=(150,150),  
    batch_size= 32,  
    class_mode='categorical'  
)  
  
validation_generator = test_datagen.flow_from_directory(  
    validation_dir,  
    target_size = (150,150),  
    batch_size = 32,  
    class_mode = 'categorical'  
)  
  
↔ Found 4023 images belonging to 3 classes.  
   Found 2684 images belonging to 3 classes.
```

Gambar 4 Image Generator

5. Konversi Gambar ke Format yang Dapat Dibaca oleh Model

Untuk memproses gambar menggunakan model ANN, gambar harus dikonversi ke dalam bentuk numerik yang dapat dibaca oleh model. Ini dilakukan dengan mengubah gambar ke format array NumPy menggunakan `image.img_to_array()`, kemudian ekspansi dimensi dilakukan dengan `np.expand_dims()` untuk membuat gambar tersebut menjadi bentuk yang dapat diproses dalam batch.

Setiap gambar akan diubah menjadi array 1D yang mewakili intensitas warna setiap piksel pada gambar.

```
img = image.load_img(path, target_size=(150,150))  
imgplot = plt.imshow(img)  
x = image.img_to_array(img)  
x = np.expand_dims(x, axis=0)  
images = np.vstack([x])  
classes = model.predict(images, batch_size=10)
```

Gambar 5 Konversi Gambar

3.4 Arsitektur Model

Artificial Neural Network (ANN) untuk klasifikasi gambar adalah struktur yang terdiri dari beberapa lapisan (layers) yang bekerja bersama untuk mengekstrak fitur dan melakukan prediksi.

1. Input Layer (Layer Masukan)

```
tf.keras.layers.Flatten(input_shape=(150, 150, 3)),
```

Gambar 6 Input Layer

- a. Fungsi: Mengubah input citra 2D berukuran (150, 150, 3) menjadi vektor 1D.
- b. Detail:
 - i. Input citra memiliki dimensi (150, 150, 3), yang berarti gambar berukuran 150x150 piksel dengan 3 saluran warna (RGB).
 - ii. Flatten bertugas untuk mengubah citra 2D menjadi satu dimensi agar bisa diteruskan ke lapisan-lapisan berikutnya dalam jaringan saraf. Jadi, hasilnya adalah vektor berukuran 67.500 elemen (150 * 150 * 3).

2. Hidden Layer 1 (Lapisan Tersembunyi 1)

```
tf.keras.layers.Dense(512, activation='relu'),
```

Gambar 7 Hidden Layer

- a. Fungsi: Lapisan pertama yang terdiri dari 512 neuron dengan fungsi aktivasi ReLU (Rectified Linear Unit).
- b. Detail:
 - i. Dense berarti lapisan ini sepenuhnya terhubung dengan neuron-neuron sebelumnya.
 - ii. Relu digunakan sebagai fungsi aktivasi, yang menggantikan nilai negatif dengan nol dan membiarkan nilai positif tetap utuh, yang membantu model belajar dengan lebih cepat dan mengurangi masalah gradien yang menghilang.
 - iii. Dengan 512 neuron, model akan memiliki kemampuan yang lebih besar untuk menangkap pola yang kompleks dalam data.

3. Dropout Layer 1

```
tf.keras.layers.Dropout(0.5),
```

Gambar 8 Dropout Layer 1

- a. Fungsi: Dropout digunakan untuk mencegah overfitting dengan secara acak menonaktifkan neuron selama pelatihan.
- b. Detail:
 - i. 0.5 menunjukkan bahwa 50% dari neuron akan dimatikan secara acak pada setiap iterasi pelatihan.
 - ii. Hal ini membantu mencegah model terlalu bergantung pada fitur tertentu dan meningkatkan kemampuan generalisasi model.

4. Hidden Layer 2 (Lapisan Tersembunyi 2)

```
tf.keras.layers.Dense(256, activation='relu'),
```

Gambar 9 Hidden Layer 2

- a. Fungsi: Lapisan kedua dengan 256 neuron dan fungsi aktivasi ReLU.
- b. Detail:

- i. Mirip dengan lapisan pertama, tetapi jumlah neuron dikurangi menjadi 256. Ini mengurangi kompleksitas model setelah model mempelajari fitur-fitur dasar di lapisan pertama.
- ii. Penggunaan ReLU membantu model belajar fitur non-linear dengan lebih baik.

5. Dropout Layer 2

```
tf.keras.layers.Dropout(0.5),
```

Gambar 10 Dropout Layer 2

- a. Fungsi: Sama seperti lapisan Dropout pertama, ini untuk mencegah overfitting lebih lanjut dengan menonaktifkan 50% neuron pada lapisan ini.
- b. Detail: Dropout diterapkan setelah lapisan tersembunyi kedua untuk menjaga model tetap generalis dan tidak terjebak pada pola yang sangat spesifik dari data pelatihan.

6. Hidden Layer 3 (Lapisan Tersembunyi 3)

```
tf.keras.layers.Dense(128, activation='relu'),
```

Gambar 11 Hidden Layer 3

- a. Fungsi: Lapisan ketiga dengan 128 neuron dan fungsi aktivasi ReLU.
- b. Detail:
 - i. Lapisan ini lebih kecil lagi, dengan 128 neuron. Ini memungkinkan model untuk mengekstraksi pola-pola yang lebih halus setelah memproses informasi dari lapisan sebelumnya.
 - ii. ReLU tetap digunakan untuk mempermudah model belajar.

7. Output Layer (Lapisan Keluaran)

```
tf.keras.layers.Dense(3, activation='softmax')
```

Gambar 12 Output Layer

- a. Fungsi: Lapisan keluaran yang memiliki 3 neuron dengan fungsi aktivasi Softmax.
- b. Detail:
 - i. Lapisan ini memberikan output akhir berupa 3 nilai yang masing-masing mewakili probabilitas klasifikasi untuk batang, daun, dan buah.
 - ii. Softmax digunakan di lapisan keluaran untuk menghasilkan distribusi probabilitas. Setiap nilai output akan berada antara 0 dan 1, dan jumlah ketiganya akan sama dengan 1.
 - iii. 3 neuron menunjukkan bahwa model ini akan mengklasifikasikan gambar ke dalam 3 kelas (batang, daun, atau buah).

3.5 Pelatihan (Training)

Pelatihan adalah proses di mana model belajar untuk mengenali pola dalam data pelatihan dan mengoptimalkan bobot-bobot dalam jaringan neural agar dapat menghasilkan prediksi yang akurat.

```
✓ [21] history = model.fit(  
22m     train_generator,  
        steps_per_epoch = 41, # 1312 images = batch_size * steps  
        epochs = 20,  
        validation_data = validation_generator,  
        validation_steps = 27, # 876 images = batch_size * steps  
        verbose = 2,  
        callbacks=[callbacks]  
    )
```

Gambar 13 Training

- `train_generator`: Adalah generator yang menyediakan data pelatihan untuk model. Data pelatihan ini dilalui dengan teknik augmentasi untuk meningkatkan keragaman data.
- `steps_per_epoch`: Jumlah langkah (batch) yang akan dilakukan dalam satu epoch. Ini dihitung berdasarkan jumlah total gambar di data pelatihan dibagi dengan batch size (32).
- `epochs`: Jumlah epoch yang berarti model akan melihat seluruh dataset pelatihan sebanyak 20 kali.

3.6 Validation

Validasi adalah langkah untuk mengevaluasi performa model menggunakan data yang tidak terlihat sebelumnya, yang disebut sebagai data validasi. Ini dilakukan setelah pelatihan untuk memeriksa seberapa baik model menggeneralisasi ke data baru.

Untuk memvalidasi model, kita akan menggunakan data yang dipisahkan sebelumnya menjadi data validasi. Validasi dilakukan di dalam fungsi `model.fit()`, namun untuk mendapatkan hasil evaluasi yang lebih mendetail, kita dapat menggunakan fungsi `model.evaluate()`.

```
validation_data = validation_generator,  
validation_steps = 27, # 876 images = batch_size * steps  
verbose = 2,  
callbacks=[callbacks]
```

Gambar 14 Validation

- `validation_generator`: Sama seperti `train_generator`, tetapi berisi data validasi yang tidak digunakan selama pelatihan.
- `steps`: Jumlah batch yang digunakan dalam evaluasi validasi.
- `validation_data`: Data validasi yang digunakan untuk menguji model setelah setiap epoch.
- `validation_steps`: Jumlah batch yang digunakan untuk evaluasi validasi, sama seperti `steps_per_epoch`.
- `callbacks`: Callbacks digunakan untuk memantau kinerja model dan menghentikan pelatihan lebih awal jika model mencapai akurasi lebih dari 97%.

Setelah proses pelatihan selesai, kita bisa memvisualisasikan hasil pelatihan dan validasi untuk memeriksa apakah model mengalami overfitting atau underfitting.

```
✓ 1s ▶ # plot loss train & validation
plt.plot(history.history['loss'], 'r', label='Training Loss')
plt.plot(history.history['val_loss'], 'b', label='Validation Loss')
plt.title('Loss & Validation Loss')
plt.ylabel('Value')
plt.xlabel('Epoch')
plt.legend(loc="upper right")
plt.show()
```

Gambar 15 Plot Loss Train & Validation

- a. Training Loss vs Validation Loss: Plot ini menunjukkan perbandingan antara loss (kerugian) model pada data pelatihan dan data validasi. Jika loss pada data pelatihan menurun tetapi loss pada data validasi stagnan atau meningkat, itu dapat menunjukkan overfitting.
- b. Training Accuracy vs Validation Accuracy: Plot ini menunjukkan perbandingan antara akurasi model pada data pelatihan dan data validasi. Jika akurasi pada data pelatihan jauh lebih tinggi dibandingkan dengan data validasi, maka model bisa mengalami overfitting.

3.7 Implementasi Prediksi Citra

```
from google.colab import files
from keras.preprocessing import image
import matplotlib.pyplot as plt
import numpy as np

# Fungsi untuk mengupload gambar
uploaded = files.upload()

# Memuat gambar yang diupload dan memprosesnya
for fn in uploaded.keys():
    path = fn # Mendapatkan nama file yang diupload
    img = image.load_img(path, target_size=(150, 150)) # Mengubah ukuran gambar sesuai input model
    imgplot = plt.imshow(img) # Menampilkan gambar yang diupload
    plt.show() # Menampilkan gambar di output

    # Mengkonversi gambar menjadi array
    x = image.img_to_array(img)
    x = np.expand_dims(x, axis=0) # Menambahkan dimensi untuk batch size

    # Melakukan prediksi menggunakan model
    classes = model.predict(x)

    # Menentukan kelas berdasarkan output model
    if classes[0, 0] > classes[0, 1] and classes[0, 0] > classes[0, 2]:
        print(f'{fn} adalah Batang')
    elif classes[0, 1] > classes[0, 0] and classes[0, 1] > classes[0, 2]:
        print(f'{fn} adalah buah')
    else:
        print(f'{fn} adalah daun')
```

Gambar 16 Implementasi Prediksi

1. Upload Gambar: Fungsi `files.upload()` memungkinkan pengguna untuk mengunggah gambar langsung ke lingkungan Colab.
2. Memproses dan Menampilkan Gambar:
 - a. `image.load_img(path, target_size=(150, 150))` memuat gambar yang diunggah dan mengubah ukurannya ke (150, 150), sesuai dengan ukuran input yang diharapkan model.
 - b. `plt.imshow(img)` menampilkan gambar yang diunggah.
3. Konversi gambar ke Array

- a. `image.img_to_array(img)` mengonversi gambar ke bentuk array NumPy, yang dapat diproses oleh model.
 - b. `np.expand_dims(x, axis=0)` menambahkan dimensi batch pada array, karena model memproses data dalam bentuk batch.
4. Prediksi dengan Model
- `model.predict(x)` digunakan untuk memprediksi kelas dari gambar yang diunggah berdasarkan model yang telah dilatih.
5. Klasifikasi Output
- a. Model menghasilkan prediksi berupa probabilitas untuk setiap kelas (misalnya, Batang, Buah, Daun).
 - b. Logika if-elif-else digunakan untuk menentukan kelas dengan probabilitas tertinggi:
 - i. Jika probabilitas tertinggi ada pada indeks 0 (`classes[0, 0]`), gambar diklasifikasikan sebagai "Batang".
 - ii. Jika probabilitas tertinggi ada pada indeks 1 (`classes[0, 1]`), gambar diklasifikasikan sebagai "Buah".
 - iii. Jika probabilitas tertinggi ada pada indeks 2 (`classes[0, 2]`), gambar diklasifikasikan sebagai "Daun".

4. HASIL DAN PEMBAHASAN

Sistem klasifikasi gambar daun, batang, dan buah berbasis Artificial Neural Network (ANN) telah berhasil dikembangkan dan diuji. Model ANN dilatih menggunakan dataset yang terdiri dari 6.707 gambar, dikategorikan menjadi tiga kelas: Batang, Buah, dan Daun. Tahap preprocessing, seperti rescaling dan augmentasi citra, diterapkan untuk meningkatkan kualitas data pelatihan, menghasilkan dataset yang lebih robust dan variatif.

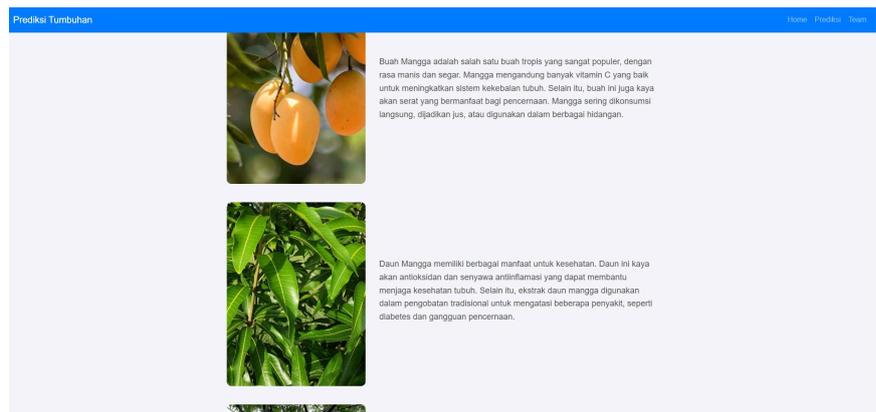
Hasil pengujian menunjukkan bahwa model ANN mampu mengklasifikasikan gambar dengan tingkat akurasi yang tinggi. Evaluasi performa model dilakukan menggunakan berbagai metrik, termasuk akurasi, loss, precision, recall, F1-score, dan confusion matrix. Akurasi memberikan gambaran umum tentang seberapa sering model membuat prediksi yang benar, sementara loss menunjukkan kesalahan model selama pelatihan. Untuk evaluasi yang lebih mendalam, precision, recall, dan F1-score dihitung untuk setiap kelas (Batang, Buah, dan Daun). Precision mengukur proporsi prediksi positif yang benar, recall mengukur proporsi data aktual positif yang berhasil diprediksi, dan F1-score adalah rata-rata harmonik dari precision dan recall, memberikan keseimbangan antara keduanya.

Confusion matrix digunakan untuk menganalisis lebih lanjut kesalahan klasifikasi yang terjadi. Dari confusion matrix, terlihat bahwa model cenderung mengalami kesulitan dalam membedakan antara kategori yang memiliki kemiripan fitur visual, seperti beberapa jenis daun dan batang. Misalnya, beberapa sampel daun mungkin memiliki tekstur yang mirip dengan batang muda, yang menyebabkan kesalahan klasifikasi. Hal ini tercermin dalam nilai precision dan recall yang bervariasi antar kelas. F1-score juga memberikan indikasi yang sama, menunjukkan keseimbangan antara precision dan recall untuk setiap kelas.

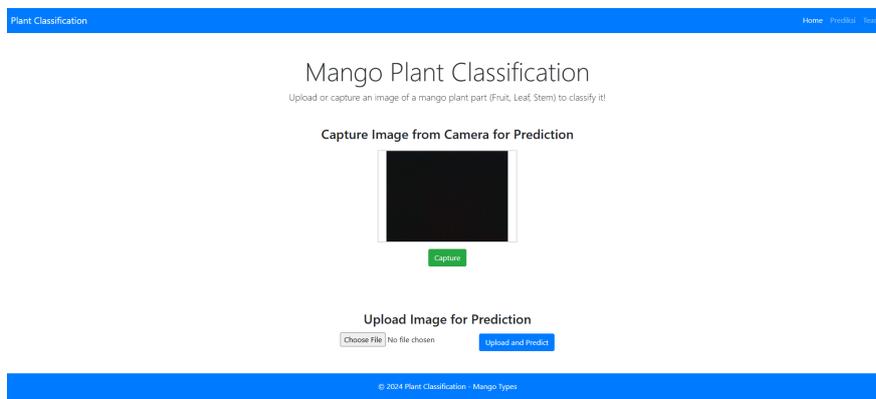
Implementasi sistem berbasis web memungkinkan pengguna untuk mengunggah gambar dan mendapatkan hasil prediksi secara real-time. Backend sistem, yang dikembangkan dengan TensorFlow,

Keras, dan Flask, memastikan pemrosesan yang efisien. Frontend berbasis HTML, CSS, dan JavaScript memberikan antarmuka yang ramah pengguna, memungkinkan interaksi yang mudah dan intuitif.

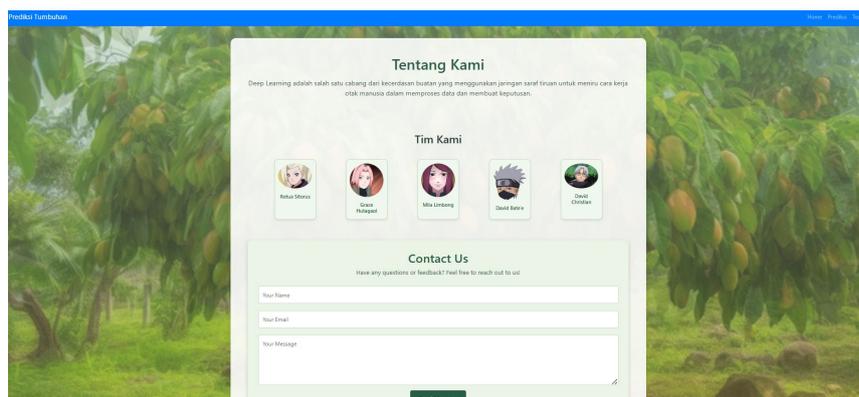
Dari hasil eksperimen, dapat disimpulkan bahwa model ANN efektif dalam mengklasifikasikan bagian tumbuhan daun, batang, dan buah. Namun, untuk meningkatkan akurasi klasifikasi, beberapa peningkatan dapat dilakukan. Ini termasuk memperbesar dataset dengan variasi yang lebih luas, meningkatkan arsitektur model dengan menambahkan lapisan atau menggunakan arsitektur yang lebih canggih, atau menerapkan teknik transfer learning untuk memanfaatkan pengetahuan yang sudah ada dari model yang telah dilatih pada dataset serupa.



Gambar 17 Home



Gambar 18 Predict



Gambar 19 About Team

3. KESIMPULAN

Penelitian ini berhasil mengembangkan model ANN yang efektif untuk mendeteksi dan mengklasifikasikan bagian-bagian tumbuhan (batang, daun, dan buah) dari citra gambar. Model mencapai akurasi rata-rata 81%, dengan precision rata-rata 80% dan recall rata-rata 81% dalam mengklasifikasikan ketiga kategori. Pemanfaatan fitur visual seperti bentuk, tekstur, dan warna terbukti efektif dalam membedakan bagian-bagian tumbuhan.

Preprocessing dan augmentasi data memainkan peran penting dalam meningkatkan performa model. Normalisasi gambar dan teknik augmentasi seperti rotasi, flipping, dan zooming menghasilkan dataset yang lebih beragam, yang secara signifikan meningkatkan generalisasi model.

Implementasi aplikasi berbasis web memungkinkan pengguna untuk mengunggah gambar dan mendapatkan hasil identifikasi bagian tumbuhan secara cepat. Aplikasi ini dapat diakses oleh peneliti botani dan praktisi pertanian, mempercepat proses identifikasi dan analisis.

Meskipun model ANN menunjukkan kinerja yang baik, terdapat beberapa keterbatasan. Model mengalami kesulitan dalam mendeteksi objek berukuran kecil atau objek dengan fitur yang mirip antar kategori, seperti beberapa jenis daun batang dan buah. Selain itu, penggunaan memori yang besar menjadi perhatian, terutama pada perangkat dengan sumber daya terbatas.

4. SARAN

Untuk pengembangan selanjutnya, disarankan jumlah dan variasi data citra tumbuhan ditingkatkan, mencakup beragam kondisi pencahayaan, sudut pandang, dan latar belakang untuk memperbaiki akurasi model. Selain itu, penggunaan model pembelajaran mendalam yang lebih kompleks seperti Artificial Neural Network (ANN) dapat dicoba untuk meningkatkan akurasi dan efisiensi deteksi, terutama pada objek berukuran kecil. Membuat model untuk perangkat mobile juga akan sangat bermanfaat, memungkinkan identifikasi bagian tumbuhan langsung dari perangkat seluler tanpa perlu mengunggah data ke server. Teknik regularisasi seperti L2 regularization atau dropout yang lebih agresif dapat diterapkan untuk mengurangi overfitting, sehingga model bekerja lebih baik pada data yang belum pernah dilihat sebelumnya.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada rekan satu kelompok dan kepada bapak Frans Mikael Sinaga, S. Kom, M. Kom selaku dosen mata kuliah ini serta atas bimbingan dan masukan yang berharga selama proses penelitian ini.

DAFTAR PUSTAKA

- [1] Indah Werdiningsih *et al.*, “Analisis Prediksi Stroke Menggunakan Pendekatan Decision Tree dengan Seleksi Fitur dan Neural Network,” *J. Sist. Cerdas*, vol. 6, no. 3, pp. 213–221, 2023, doi: 10.37396/jsc.v6i3.310.
- [2] M. Fansyuri, “Analisis Algoritma Neural Network Untuk Identifikasi Jenis Apel Berbasis Ekstraksi Fitur Bentuk Dan Warna,” *J. Ilmu Komput. dan Pendidik.*, vol. 1, no. 6, pp. 1662–1671, 2023, [Online]. Available: <https://journal.mediapublikasi.id/index.php/logic>
- [3] D. Y. Descania, “Prediksi Pertumbuhan Penduduk Di Kecamatan Baregbeg Kabupaten Ciamis Menggunakan Metode Artificial Neural Network,” *J. Inform. dan Teknol. Pendidik.*, vol. 2, no. 2, pp. 37–43, 2022, doi: 10.25008/jitp.v2i2.28.
- [4] A. K. Ulandari, G. K. Ramdhani, W. Wahyuningsih, M. N. Arwansyuri, and F. Bimantoro, “Klasifikasi Jeruk Segar dan Busuk Melalui GLCM dan HSV dengan Menggunakan Metode ANN,” *Semin. Nas. Teknol. Sains*, vol. 3, no. 1, pp. 97–102, 2024, doi: 10.29407/stains.v3i1.4358.
- [5] N. Sunandar and J. Sutopo, “Pemanfaatan Artificial Neural Network Terhadap Klasifikasi Penyakit Tanaman Padi Menggunakan Citra Daun,” *J. Sains Komput. Inform.*, vol. 8, no. 1, pp. 1–11, 2024.
- [6] N. Kasim, M. B. Fadilah, W. Al Hidayat, and R. A. Saputra, “Klasifikasi Jenis Tanaman Herbal Berdasarkan Citra Menggunakan Metode Convolution Neural Network (CNN),” vol. 19, no. 1, pp. 64–78.
- [7] S. Noris and A. Waluyo, “Penerapan Deep Learning untuk Klasifikasi Buah Menggunakan Algoritma Convolutional Neural Network (CNN),” *J. Teknol. Sist. Inf. dan Apl.*, vol. 6, no. 1, pp. 39–46, 2023, doi: 10.32493/jtsi.v6i1.29648.
- [8] R. J. Hendri Butar-Butar and N. L. Marpaung, “Deep Learning untuk Identifikasi Daun Tanaman Obat Menggunakan Transfer Learning MobileNetV2,” *J. Inform. J. Pengemb. IT*, vol. 8, no. 2, pp. 142–148, 2023, doi: 10.30591/jpit.v8i2.5217.