

LAPORAN FINAL PROJECT
Pengenalan Citra Digital (Tanaman Rempah)



TEAM:

[Project Manager dan Fullstack] I Putu Marcel Wiguna

[UI/UX] Chelsy Elisabet Gultom

[Backend] Nanda Nikola

[Frontend] Ronaldito Juan Bantaras Tarigan

INFORMATIKA

FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM

UNIVERSITAS UDAYANA

2023

PENDAHULUAN

Permasalahan

Kurangnya pengetahuan umum tentang tanaman rempah menjadi kendala utama. Banyak orang tidak dapat mengidentifikasi atau membedakan antara berbagai jenis tanaman rempah, termasuk nama, ciri-ciri fisik, dan manfaatnya. Selain itu, kesulitan dalam identifikasi manual menjadi permasalahan serius, terutama bagi mereka yang tidak memiliki latar belakang botani atau kuliner. Risiko penggunaan yang tidak tepat dari tanaman rempah juga menjadi kekhawatiran, mengingat sifat dan manfaat yang berbeda-beda. Hal ini dapat membahayakan kesehatan dan menyebabkan efek samping. Pentingnya konservasi tanaman rempah juga harus ditekankan, karena kurangnya kesadaran dapat mengancam keberlangsungan populasi tanaman rempah. Terakhir, keterbatasan akses informasi yang mudah dimengerti juga menjadi hambatan. Tidak semua orang memiliki akses mudah ke informasi terpercaya dan mudah dimengerti tentang tanaman rempah. Oleh karena itu, penerapan pengenalan citra untuk identifikasi tanaman rempah bertujuan untuk mengatasi isu-isu tersebut dengan memberikan solusi yang mudah digunakan, memberikan informasi yang akurat dan menarik tentang tanaman rempah, serta meningkatkan pemahaman dan apresiasi masyarakat terhadap keanekaragaman tanaman rempah. Dengan aplikasi ini, diharapkan masyarakat dapat dengan mudah mengenali, memahami, dan memanfaatkan tanaman rempah dengan bijak.

Latar Belakang

Tanaman rempah-rempah memiliki peran yang sangat penting dalam industri kuliner dan pengobatan tradisional di berbagai negara di seluruh dunia. Sejak zaman dahulu Indonesia dikenal sebagai penghasil rempah terbaik dan terbesar di dunia. Berdasarkan data Food and Agricultural Organization (FAO) tahun 2022 Indonesia menduduki posisi ke lima sebagai produsen rempah-rempah terbesar di dunia. Namun pada era modern saat ini kurangnya pengetahuan umum tentang tanaman rempah telah menjadi kendala utama dalam memaksimalkan manfaat dari keanekaragaman tanaman ini. Masalah utama yang dihadapi masyarakat adalah kurangnya pengetahuan tentang identifikasi dan perbedaan antara berbagai jenis tanaman rempah. Selain itu, seringkali sulit untuk mengidentifikasi tanaman rempah secara manual, terutama bagi masyarakat awam yang tidak memiliki latar belakang dalam botani atau kuliner. Oleh karena itu, risiko penggunaan tanaman rempah yang tidak tepat meningkat karena rempah yang berbeda mempunyai karakteristik dan manfaat yang berbeda

pula. Penggunaan yang tidak tepat utama dapat mengubah cita rasa masakan, dapat menimbulkan risiko Kesehatan hingga menimbulkan efek samping.

Untuk mengatasi berbagai permasalahan tersebut pengenalan citra digital adalah solusi yang potensial dan relevan. Salah satu metode yang dapat digunakan dalam pengenalan citra adalah metode template matching. Metode template matching akan membandingkan gambar yang diambil oleh pengguna dengan template gambar yang telah disimpan dalam database aplikasi. Informasi yang akurat tentang nama, ciri-ciri fisik, manfaat, dan bahkan saran penggunaan dapat disediakan melalui aplikasi ini berdasarkan kesesuaian dengan template yang sudah ada. Melalui aplikasi ini, diharapkan masyarakat dapat dengan mudah mengenali, memahami, dan memanfaatkan tanaman rempah dengan bijak, yang pada gilirannya akan memberikan manfaat yang lebih besar baik dalam pengobatan tradisional maupun dalam pengolahan makanan.

Rumusan Masalah

1. Bagaimana cara mengelola inputan gambar tanaman rempah menggunakan Convolutional Neural Network guna menentukan hasil berupa nama dan penjelasan singkat mengenai tanaman rempah yang diinput?
2. Bagaimana tingkat akurasi sistem dalam mengenali tanaman rempah menggunakan metode Convolutional Neural Network?

Tujuan

Tujuan dari penelitian ini adalah untuk mengembangkan sistem klasifikasi tanaman rempah menggunakan Convolutional Neural Network

Batasan Masalah

1. Penelitian ini berfokus hanya pada pengenalan beberapa tanaman rempah tertentu sebagai objek utama, seperti jahe, kunyit, lengkuas, dll.
2. Inputan berupa gambar dengan format JPG
3. Sistem hanya dapat mendeteksi satu tanaman rempah dalam satu gambar.

Analisis Kebutuhan Fungsional

1. Aplikasi mampu mengidentifikasi nama dari gambar rempah yang diinputkan
2. „ Aplikasi mampu menerima inputan berupa gambar
3. „ Aplikasi mampu mendeskripsikan kegunaan dari rempah tersebut
4. „ Aplikasi mampu memberikan hasil yang terpercaya dan akurasi yang cukup tinggi

Analisis Kebutuhan Non Fungsional

1. Aplikasi harus bisa diakses melalui perangkat yang tersambung dengan jaringan internet
2. Aplikasi harus memastikan keamanan dan privasi data pengguna, termasuk informasi yang mungkin diunggah atau digunakan selama proses pengenalan
3. Aplikasi harus efisien dalam penggunaan memori maupun daya baterai perangkat
4. Antarmuka harus ramah pengguna, mudah digunakan, dan memiliki desain intuitif untuk beragam kelompok pengguna.
5. Aplikasi harus mendukung berbagai resolusi layar pada perangkat yang berbeda dengan baik.

RANCANGAN UMUM

Model dan Arsitektur

Rancangan dari model yang akan kami buat ini menerapkan konsep klasifikasi citra, dimana kumpulan citra yang sudah kami kumpulkan ini nantinya akan digunakan untuk membuat dan melatih model yang berfungsi sebagai proses dari klasifikasi citra itu sendiri.

Tujuan dari model ini adalah untuk mengklasifikasikan citra tanaman rempah ke dalam 9 kategori yang berbeda. Agar dapat memenuhi kebutuhan tersebut, kami memilih arsitektur MobileNetV2 sebagai basis model CNN. MobileNetV2 merupakan pilihan yang ideal karena dirancang untuk menjadi ringan dan efisien, cocok untuk perangkat dengan sumber daya terbatas.

Convolutional Neural Network (CNN) adalah kelas arsitektur deep learning yang secara khusus dirancang untuk menangani tugas-tugas visi komputer, seperti pengenalan pola dalam citra atau video. CNN telah mencapai tingkat keberhasilan yang luar biasa dalam sejumlah besar aplikasi, termasuk klasifikasi gambar, deteksi objek, dan segmentasi gambar.

CNN, atau Convolutional Neural Network, memiliki struktur dasar yang terdiri dari beberapa jenis lapisan utama. Pertama, terdapat Lapisan Konvolusi, yang memainkan peran kunci dalam mengekstrak fitur-fitur penting dari citra dengan menerapkan operasi konvolusi. Pada lapisan ini, filter atau kernel digunakan untuk mendeteksi pola-pola khusus dalam citra, memungkinkan model untuk memahami karakteristik yang lebih kompleks.

Selanjutnya, terdapat Lapisan Pooling, yang bertujuan mereduksi dimensi spasial citra dengan mengambil nilai maksimum atau rata-rata dalam jendela tertentu. Fungsi ini membantu mengurangi jumlah parameter dan komputasi, serta menciptakan invariansi terhadap translasi kecil dalam citra. Lapisan Aktivasi juga menjadi komponen penting, menggunakan fungsi aktivasi seperti ReLU untuk menambah kapasitas non-linearitas pada model.

Akhirnya, terdapat Lapisan Fully Connected (Dense) yang menghubungkan setiap neuron di lapisan ini ke setiap neuron di lapisan sebelumnya dan sesudahnya. Biasanya terletak di akhir arsitektur, lapisan ini bertanggung jawab untuk melakukan klasifikasi atau regresi berdasarkan representasi-fitur yang telah diekstrak sebelumnya. Dengan kombinasi

lapisan-lapisan ini, CNN menjadi sangat efektif dalam menangani tugas-tugas seperti pengenalan pola dan klasifikasi citra

MobileNetV2 adalah arsitektur jaringan saraf tiruan (neural network) yang dikembangkan oleh Google, dirancang khusus untuk aplikasi dengan sumber daya terbatas, seperti perangkat seluler dan perangkat yang memerlukan efisiensi komputasi tinggi. MobileNetV2 merupakan evolusi dari versi sebelumnya, MobileNet, dan memperkenalkan berbagai inovasi untuk meningkatkan akurasi dan efisiensi.

User Interface (Front End)

Pengenalan Tanaman dan Edukasi Rempah

Silahkan Upload Gambar tanaman rempah yang ingin dikenali!

Choose an image...

Browse files

Untuk WebApp kami menggunakan streamlit sebagai hostingnya dan pada gambar diatas merupakan tampilan dari aplikasi untuk mengupload gambar yang ingin dikenali oleh pengguna.



Hasil prediksi: lengkuas

Pada gambar diatas merupakan hasil prediksi yang dilakukan oleh aplikasi yang dimana mencocokannya dengan hasil pelatihan yang dilakukan pada model.

Rencana Implementasi

1. Tools yang digunakan: Figma, Google Colab, Visual Studio Code, Google Drive
 2. Bahasa Pemrograman yang digunakan: Python, PHP, HTML CSS
 3. Pustaka Yang digunakan (dalam Model): Keras + Tensorflow, Sckit Learn, Open CV, Numpy, Matplotlib, Pandas dan Seaborn
-
- Tahap deployment akan di deploy pada <https://streamlit.io/>.

Rencana Evaluasi

Pada penelitian ini, matrik evaluasi yang digunakan adalah confusion matrix. Confusion matrix merupakan salah satu matrik evaluasi yang digunakan dalam machine learning. Pada dasarnya confusion matrix memberikan informasi perbandingan antara hasil klasifikasi yang dilakukan oleh model dengan hasil klasifikasi sebenarnya. Gambar dibawah ini merupakan confusion matrix dengan 4 kombinasi nilai prediksi dan nilai aktual yang berbeda.

		Nilai Aktual	
		Positive	Negative
Nilai Prediksi	Positive	TP	FP
	Negative	FN	TN

- True Positive (TP), kondisi dimana nilai prediksi positif dengan kondisi asli positif
- False Positive (FP), kondisi dimana nilai prediksi positif, namun kondisi asli negatif
- True Negative (TN), kondisi dimana nilai prediksi negatif dan kondisi asli negatif
- False Negative (FN), kondisi dimana nilai prediksi negatif, namun kondisi asli positif

Evaluasi model dilakukan dengan menggunakan *confusion matrix* untuk mencari tingkat akurasi dari data uji terhadap model yang dibuat. Selain untuk mencari nilai akurasi, confusion matrix juga digunakan untuk mencari nilai *precision*, *recall*, dan *F1 Score*.

- Accuracy: Metrik yang digunakan untuk mengukur seberapa sering model kita memberikan prediksi yang benar

$$\text{accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

- Precision: Metrik yang digunakan untuk mengukur seberapa akurat model kita ketika memprediksi positif

$$\text{precision} = \frac{TP}{TP + FP}$$

- Recall: Metrik yang digunakan untuk mengukur seberapa baik model kita dalam menemukan nilai positif

$$\text{recall} = \frac{TP}{TP + FN}$$

- F1 Score: Metrik gabungan dari nilai Precision dan recall

$$F_1 = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$$

Dalam melakukan evaluasi model, terdapat penentuan epoch. Epoch adalah parameter seberapa banyak algoritma akan melakukan proses dimana seluruh dataset melewati seluruh proses model sebanyak 1 kali. Pada evaluasi biasanya terdapat pengujian terhadap beberapa epoch berbeda untuk menentukan epoch terbaik. Epoch terbaik dapat dilihat dari nilai akurasi, presisi, dan recall dari masing masing epoch. Contoh epoch: 25, 50, 75, 100.

Rencana Deploy

Aplikasi yang telah dirancang akan di deploy pada <https://streamlit.io/>. Streamlit adalah sebuah framework open-source yang digunakan untuk membangun antarmuka pengguna (UI) interaktif untuk aplikasi data science. Framework ini dirancang khusus untuk mempermudah pengembangan aplikasi web dengan menggunakan Python. Streamlit digunakan untuk mempermudah pembuatan desain interaktif aplikasi dan juga sesuai dengan tujuan awal aplikasi yaitu berbasis website/web. Streamlit sudah terintegrasi dengan GitHub sehingga mempermudah pekerjaan tim dalam pembagian tugas dan rancangan deploy aplikasi. Platform streamlit yang gratis dan berbasis komunitas juga menjadi pendukung mengapa kami menggunakan streamlit. Platform Streamlit menggunakan bahasa Python sebagai bahasa utama yang juga merupakan inti dari aplikasi yang kami rancang, bahasa python yang dapat dihubungkan dengan banyak library dan bahasa lain seperti javascript,sql, library keras atau tensorflow, juga mempermudah dalam tim mendeploy aplikasi

IMPLEMENTASI

Implementasi Pengumpulan Dataset

Dataset yang kami gunakan ini berasal dari kaggle, dataset ini memiliki 9 label citra rempah yang terdiri dari rempah Jahe Emprit, Jahe Merah, Jahe Putih, Kencur, Kunyit Hitam, Kunyit Kuning, Kunyit Putih, Lengkuas dan Temulawak. Dataset tersebut memiliki jumlah yang berbeda tiap labelnya. Berikut adalah sebaran dataset dari 9 label yang kami dapat dari kaggle

No	Label	Jumlah
1	Jahe Emprit	116
2	Jahe Merah	129
3	Jahe Putih	122
4	Kencur	128
5	Kunyit Hitam	127
6	Kunyit Kuning	132
7	Kunyit Putih	124
8	Lengkuas	128
9	Temulawak	128

Karena jumlahnya yang tidak konsisten dan total data nya tergolong sedikit, jadi kami melakukan beberapa proses untuk meratakan jumlah data di masing masing label serta membaginya kedalam 3 split data. sehingga skenario pengumpulan data setnya adalah sebagai berikut

1. Data yang sudah didapat dan terdiri dari 9 label tersebut kami bagi menjadi 2 kumpulan data tiap label dengan perbandingan 70 banding 30. Kumpulan data ini kami beri nama Train dan Test
2. Untuk memastikan bahwa model yang dibangun tidak cenderung menjadi bias terhadap satu kelas tertentu, disini kami rombak lagi agar tiap label memiliki jumlah data yang sama
3. Setelah tiap label memiliki jumlah yang sama, data test kami split lagi menjadi 2 kumpulan data, dimana kumpulan data baru memiliki jumlah 17.5% dari data test,

kumpulan data baru ini digunakan untuk memuat prediksi sebagai bagian dari pelatihan model

4. Setelah semua proses di atas selesai, tahap berikutnya adalah melakukan augmentasi dengan menggunakan kombinasi rotating dan flipping. Untuk rotating disini kami melakukannya dengan kombinasi rotating 90 derajat, 180 derajat, dan 270 derajat. Kemudian untuk flipping nya 2 kombinasi yakni flipping secara horizontal dan vertikal. Dari proses augmentasi ini menghasilkan 5 data baru tiap 1 data asli di dalam masing masing label, Untuk jumlah akhir dari dataset yang kami gunakan dapat dilihat pada gambar berikut

Label	Train	Test	Inet
jahe_emprit	359	198	21
jahe_merah	359	198	21
jahe_putih	359	198	21
kencur	359	198	21
kunyit_hitam	359	198	21
kunyit_kuning	359	198	21
kunyit_putih	359	198	21
lengkuas	359	198	21
temulawak	359	198	21

5. Jumlah untuk data Train sebanyak 3231, Data test 1782, dan data Inet 189 dengan total 5202

Implementasi Preprocessing Citra

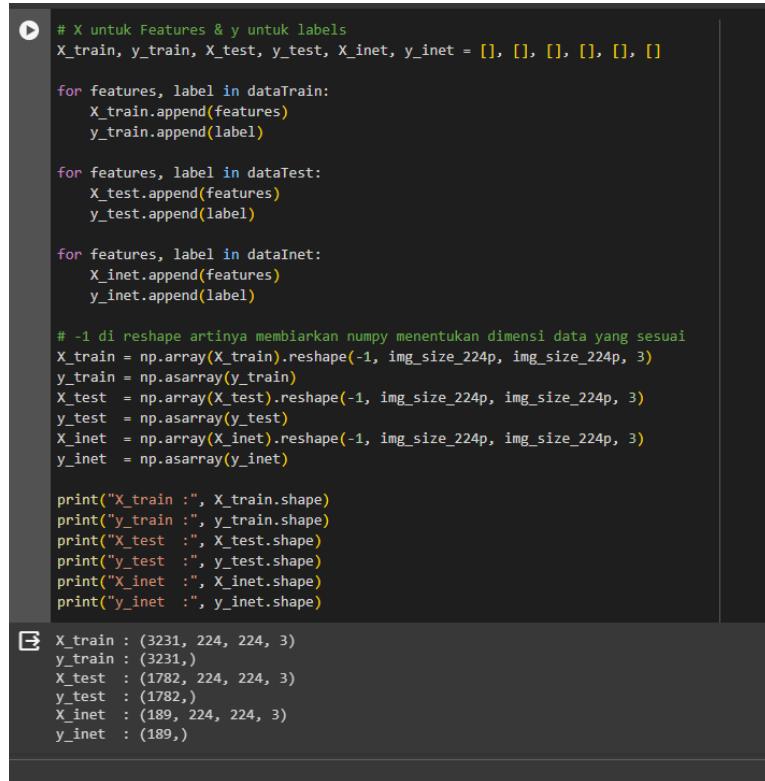
Setelah dataset yang digunakan sudah ditentukan melalui proses pengumpulan data, langkah selanjutnya adalah melakukan preprocessing pada data yang akan digunakan. Terdapat beberapa proses yang digunakan dalam preprocessing kali ini yakni

1. Mengubah citra menjadi array
2. Melakukan resize dengan ukuran yang sesuai dengan model yang akan digunakan
3. Normalisasi data
4. One hot encoding

Dalam proses manajemen dataset citra rempah, ukuran citra ditetapkan untuk konsistensi, dan path direktori masing-masing bagian dataset (pelatihan, internet, dan pengujian) diinisialisasi. Kategori kelas rempah yang akan diklasifikasikan juga ditentukan. Melalui fungsi yang

dibuat, setiap citra dibaca, dikonversi warna, dan diubah ukurannya sesuai dengan parameter yang telah ditetapkan. Selanjutnya, hasilnya diorganisir dalam daftar yang mencakup pasangan citra dan label kelas. Pengacakan dilakukan pada dataset pelatihan untuk mencegah model mempelajari pola urutan. Dengan demikian, langkah-langkah ini membentuk dasar untuk menyusun dataset citra rempah yang siap untuk digunakan dalam melatih dan menguji model klasifikasi tanaman rempah.

Langkah selanjutnya dalam pengelolaan dataset citra rempah ini adalah mempersiapkan fitur (X) dan label (y) untuk digunakan dalam melatih dan menguji model klasifikasi. Proses ini melibatkan iterasi melalui dataset pelatihan, pengujian, dan internet yang telah diatur sebelumnya. Citra dan label dari setiap dataset dipisahkan dan disusun ke dalam array terpisah untuk fitur (X) dan label (y). Kemudian, array fitur diubah bentuknya menggunakan numpy untuk memastikan dimensi yang sesuai dengan kebutuhan model. Setelah proses ini selesai, dimensi dan bentuk setiap array ditampilkan untuk memastikan bahwa data telah dipersiapkan dengan benar untuk proses pelatihan dan pengujian model klasifikasi citra rempah.



```
# X untuk Features & y untuk labels
X_train, y_train, X_test, y_test, X_inet, y_inet = [], [], [], [], [], []

for features, label in dataTrain:
    X_train.append(features)
    y_train.append(label)

for features, label in dataTest:
    X_test.append(features)
    y_test.append(label)

for features, label in dataInet:
    X_inet.append(features)
    y_inet.append(label)

# -1 di reshape artinya membiarkan numpy menentukan dimensi data yang sesuai
X_train = np.array(X_train).reshape(-1, img_size_224p, img_size_224p, 3)
y_train = np.asarray(y_train)
X_test = np.array(X_test).reshape(-1, img_size_224p, img_size_224p, 3)
y_test = np.asarray(y_test)
X_inet = np.array(X_inet).reshape(-1, img_size_224p, img_size_224p, 3)
y_inet = np.asarray(y_inet)

print("X_train :", X_train.shape)
print("y_train :", y_train.shape)
print("X_test :", X_test.shape)
print("y_test :", y_test.shape)
print("X_inet :", X_inet.shape)
print("y_inet :", y_inet.shape)
```

x_train : (3231, 224, 224, 3)
y_train : (3231,)
x_test : (1782, 224, 224, 3)
y_test : (1782,)
x_inet : (189, 224, 224, 3)
y_inet : (189,)

Langkah berikutnya adalah memeriksa isi dari array fitur (X) setelah proses persiapan sebelumnya. Contoh dari data pertama dalam setiap array fitur ditampilkan untuk memberikan gambaran visual. Selanjutnya, sebuah fungsi prep_pixels dibuat untuk

mengakukan normalisasi pada nilai piksel dari array fitur. Proses normalisasi ini melibatkan konversi nilai piksel dari tipe data integer menjadi float dan kemudian menormalisasi nilai-nilai tersebut ke dalam rentang antara 0 hingga 1 dengan membaginya dengan 255.0 (maksimum nilai piksel). Setelah normalisasi, isi dari array fitur yang sudah dinormalisasi (`X_train_norm`, `X_test_norm`, dan `X_inet_norm`) juga ditampilkan untuk memperlihatkan perubahan nilai-nilai piksel yang telah diproses. Proses normalisasi ini bertujuan untuk membantu dalam konvergensi yang lebih cepat selama pelatihan model.

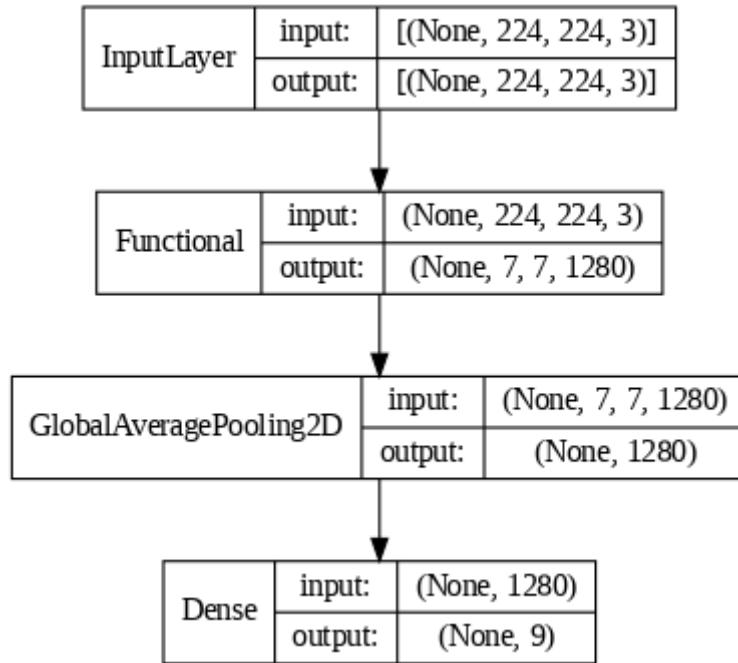
Selanjutnya, label kelas (`y`) yang telah dipisahkan sebelumnya ditampilkan untuk memberikan gambaran tentang nilai-nilai label yang dimiliki oleh setiap dataset (pelatihan, pengujian, dan internet). Selanjutnya, dilakukan proses "one-hot encoding" menggunakan fungsi `to_categorical` dari Keras. Proses ini mengubah label-label kelas menjadi representasi biner yang lebih sesuai untuk pelatihan model klasifikasi. Setelah proses "one-hot encoding" selesai, isi dari array label yang telah diubah tersebut (`y_train_encode`, `y_test_encode`, dan `y_inet_encode`) juga ditampilkan untuk memberikan gambaran tentang representasi biner dari setiap label kelas. Proses ini mempersiapkan label untuk digunakan dalam melatih dan menguji model klasifikasi citra rempah dengan lebih efektif.



Implementasi Model MobileNet V2

Dengan menggunakan arsitektur MobileNetV2 dari Keras Applications sebagai basis konvolusional untuk model klasifikasi. MobileNetV2 telah dilatih sebelumnya pada dataset ImageNet, dan pada kasus ini, hanya bagian konvolusionalnya yang digunakan, dengan menghilangkan lapisan fully connected (`include_top=False`). Kemudian, bobot-bobot dari MobileNetV2 yang telah dilatih tidak akan diperbarui selama proses pelatihan model klasifikasi ini dengan mengatur `conv_base.trainable = False`. Hal ini membantu

memanfaatkan representasi fitur yang telah dipelajari oleh MobileNetV2 untuk tugas klasifikasi citra rempah tanpa memperkenalkan pembelajaran ulang yang signifikan. Fungsi summary() digunakan untuk menampilkan ringkasan arsitektur dari basis konvolusional MobileNetV2. Keseluruhan, langkah ini memanfaatkan transfer learning untuk meningkatkan kinerja model klasifikasi pada dataset citra rempah yang lebih kecil.



1. MobileNetV2 Base Layer:, Bagian ini mencakup arsitektur konvolusional MobileNetV2 yang telah dilatih sebelumnya pada dataset ImageNet. Outputnya berupa tensor dengan dimensi $(7, 7, 1280)$, menunjukkan representasi fitur yang diperoleh dari lapisan konvolusional MobileNetV2.
2. Global Average Pooling Layer:, Layer ini menjalankan operasi global average pooling pada output dari MobileNetV2. Operasi ini menghasilkan nilai rata-rata dari setiap saluran (channel) dalam matriks, menghasilkan vektor fitur dengan panjang 1280.
3. Dense Layer (Fully Connected):, Lapisan ini merupakan lapisan fully connected yang menghubungkan vektor fitur ke output dengan dimensi 9, sesuai dengan jumlah kelas yang diinginkan untuk klasifikasi citra rempah. Fungsi aktivasi softmax digunakan untuk menghasilkan probabilitas distribusi kelas.

Dengan menandai awal dari proses pelatihan model klasifikasi citra rempah dengan menggunakan arsitektur yang telah ditentukan sebelumnya. Dengan menggunakan model yang telah dibentuk melalui fungsi define_model_mobilnetv2, pelatihan dimulai dengan menentukan parameter pelatihan, yaitu jumlah epochs (iterasi melalui dataset) dan ukuran

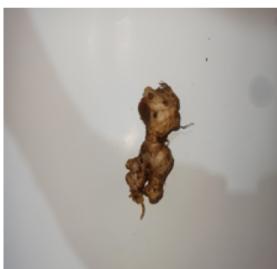
batch yang digunakan selama pelatihan. Dengan mengatur epochs sebanyak 10 dan batch size sebesar 64, model diinstruksikan untuk melatih dirinya sendiri dengan menggunakan dataset pelatihan yang telah dipersiapkan sebelumnya. Proses ini bertujuan untuk mengadaptasi bobot model agar dapat mengenali dan mengklasifikasikan citra rempah dengan akurasi yang optimal. Setelah proses pelatihan selesai, informasi pelatihan seperti loss dan akurasi pada setiap epoch dapat ditemukan dalam variabel history, yang dapat digunakan untuk evaluasi performa model secara lebih rinci. Keseluruhan, tahapan ini menjadi kunci dalam meningkatkan kemampuan model untuk memahami pola-pola yang terdapat dalam dataset citra rempah yang telah dipersiapkan.

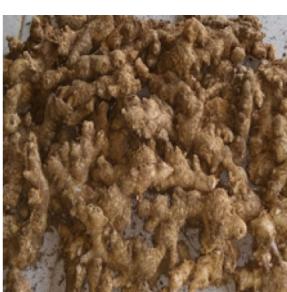
Hasil Pengujian

Tumbuhan yang diupload dan dilakukan pengujian akan menampilkan hasil pengujian. Pengujian dilakukan menggunakan model CNN yang telah dilatih sebelumnya. Adapun data yang digunakan untuk melakukan pengujian, merupakan data diluar data latih. Hasil dari pengujian, dapat dilihat bawah ini:

- Jahe Emprit

Berikut hasil data pengujian dari Jahe Emprit:

No	Gambar	Nama Tanaman	Nilai Akurasi
1		Jahe Emprit	92.49%
2		Jahe Emprit	54.58%

3		Jahe Emprit	93.45%
4		Jahe Emprit	97.26%
5		Jahe Emprit	38.12%

- Jahe Merah

Berikut hasil data pengujian dari Jahe Merah:

No	Gambar	Nama Tanaman	Nilai Akurasi
1		Jahe Merah	83.25%

2		Jahe Merah	64.82%
3		Jahe Merah	71.16%
4		Jahe Merah	82.61%
5		Jahe Merah	55.60%

- **Jahe Putih**

Berikut hasil data pengujian dari Jahe Putih:

No	Gambar	Nama Tanaman	Nilai Akurasi
----	--------	--------------	---------------

1		Jahe Putih	76.03%
2		Jahe Putih	88.43%
3		Jahe Putih	96.29%
4		Jahe Putih	68.15%
5		Jahe Putih	93.47%

- **Kencur**

Berikut hasil data pengujian dari Kencur:

No	Gambar	Nama Tanaman	Nilai Akurasi
----	--------	--------------	---------------

1		Kencur	43.72%
2		Kencur	99.93%
3		Kencur	94.90%
4		Kencur	51.48%
5		Kencur	97.83%

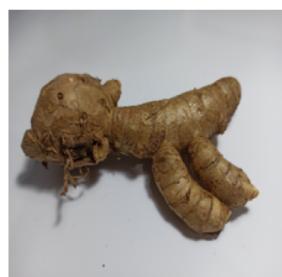
- **Kunyit Hitam**

Berikut hasil data pengujian dari Kunyit Hitam:

No	Gambar	Nama Tanaman	Nilai Akurasi
1		Kunyit Hitam	93.64%

- **Kunyit Putih**

Berikut hasil data pengujian dari Kunyit Putih:

No	Gambar	Nama Tanaman	Nilai Akurasi
1		Kunyit Putih	98.28%
2		Kunyit Putih	88.45%

3		Kunyit Putih	52.78%
4		Kunyit Putih	68.62%
5		Kunyit Putih	98.37%

- **Kunyit Kuning**

Berikut hasil data pengujian dari Kunyit Kuning:

No	Gambar	Nama Tanaman	Nilai Akurasi
1		Kunyit Kuning	99.12%
2		Kunyit Kuning	77.96%

3		Kunyit Kuning	81.32%
4		Kunyit Kuning	95.84%
5		Kunyit Kuning	64.85%

- **Lengkuas**

Berikut hasil data pengujian dari Lengkuas:

No	Gambar	Nama Tanaman	Nilai Akurasi
1		Lengkuas	98.97%

2		Lengkuas	76.56%
3		Lengkuas	99.99%
4		Lengkuas	98.87%
5		Lengkuas	95.39%

- **Temulawak**

Berikut hasil data pengujian dari Temulawak:

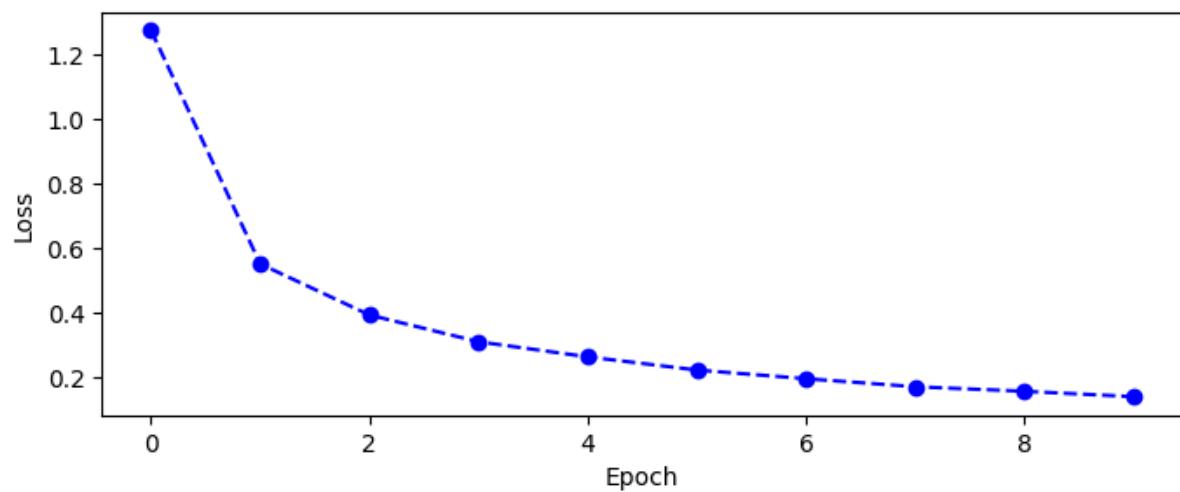
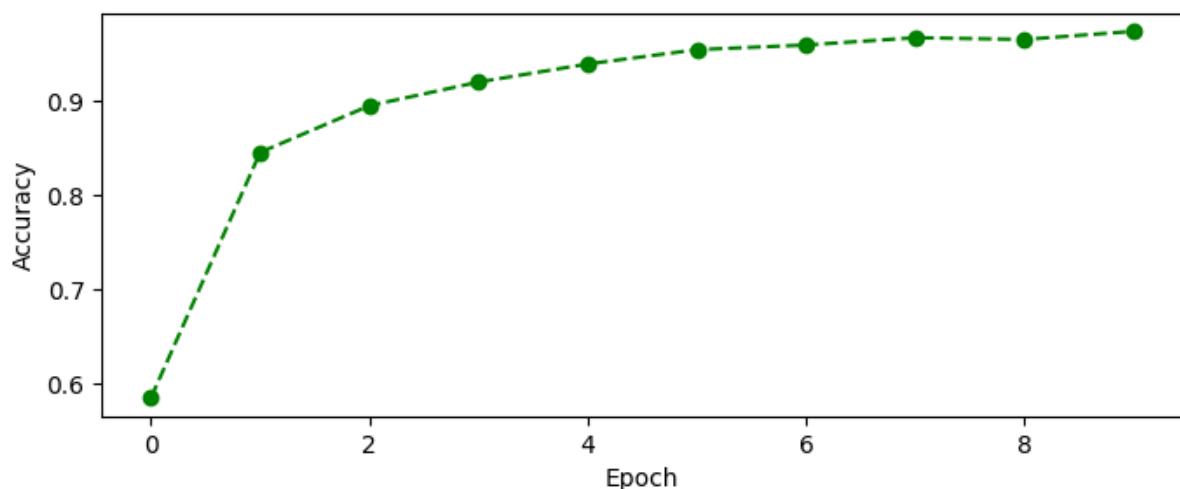
No	Gambar	Nama Tanaman	Nilai Akurasi
1		Temulawak	99.86%

2		Temulawak	81.34%
3		Temulawak	98.45%
4		Temulawak	95.66%
5		Temulawak	82.58%

- Hasil dari latih dengan 10 epoch

training data latih untuk model cnn dibagi kedalam 10 epoch. Adapun nilai akurasi dan loss yang didapat dari pelatihan 10 epoch, dapat dilihat pada gambar dibawah

```
Epoch 1/10
51/51 [=====] - 138s 3s/step - loss: 1.2766 - accuracy: 0.5846
Epoch 2/10
51/51 [=====] - 144s 3s/step - loss: 0.5505 - accuracy: 0.8449
Epoch 3/10
51/51 [=====] - 143s 3s/step - loss: 0.3919 - accuracy: 0.8945
Epoch 4/10
51/51 [=====] - 133s 3s/step - loss: 0.3085 - accuracy: 0.9195
Epoch 5/10
51/51 [=====] - 136s 3s/step - loss: 0.2616 - accuracy: 0.9387
Epoch 6/10
51/51 [=====] - 134s 3s/step - loss: 0.2207 - accuracy: 0.9539
Epoch 7/10
51/51 [=====] - 143s 3s/step - loss: 0.1938 - accuracy: 0.9588
Epoch 8/10
51/51 [=====] - 136s 3s/step - loss: 0.1689 - accuracy: 0.9666
Epoch 9/10
51/51 [=====] - 140s 3s/step - loss: 0.1547 - accuracy: 0.9647
Epoch 10/10
51/51 [=====] - 144s 3s/step - loss: 0.1379 - accuracy: 0.9734
```

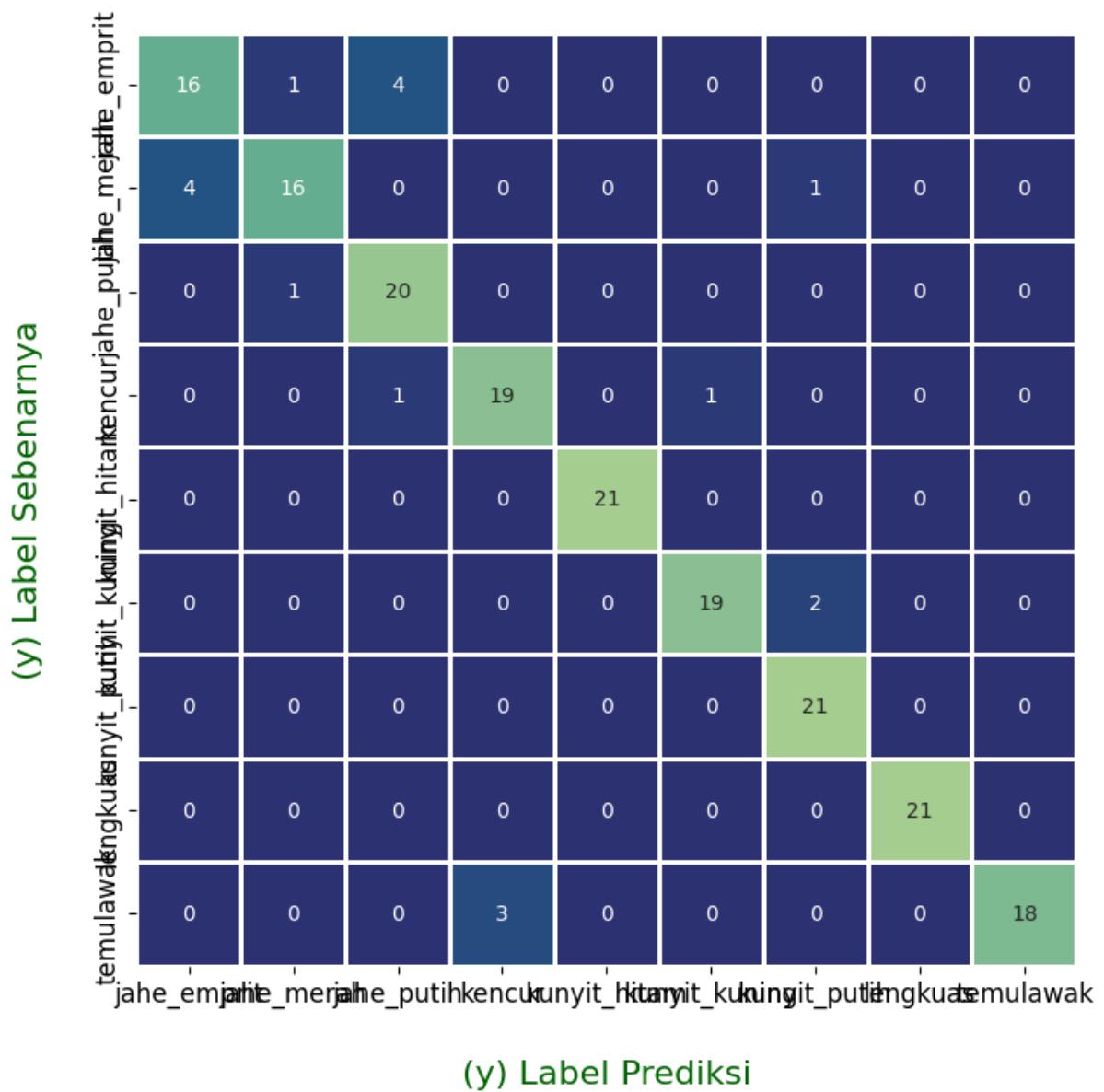


Dari gambar dan grafik, dapat dilihat bahwa semakin banyak epoch dilatih, maka nilai loss akan semakin rendah dan akurasi semakin tinggi

Accuracy : 90.476%				
Loss : 0.278				
Classification Report :				
	precision	recall	f1-score	support
jahe_emprit	0.80	0.76	0.78	21
jahe_merah	0.89	0.76	0.82	21
jahe_putih	0.80	0.95	0.87	21
kencur	0.86	0.90	0.88	21
kunyit_hitam	1.00	1.00	1.00	21
kunyit_kuning	0.95	0.90	0.93	21
kunyit_putih	0.88	1.00	0.93	21
lengkuas	1.00	1.00	1.00	21
temulawak	1.00	0.86	0.92	21
accuracy			0.90	189
macro avg	0.91	0.90	0.90	189
weighted avg	0.91	0.90	0.90	189

Adapun nilai presisi, recall, dan f1 score dari masing - masing kategori dapat dilihat pada gambar diatas. dapat dilihat, nilai precision, recall, dan f1 score tertinggi terdapat pada kategori lengkuas.

MobileNetV2 Original



Kesimpulan

Pada project ini pengembangan sistem pengenalan citra tanaman rempah menggunakan model CNN telah berhasil diimplementasikan dengan baik. Menggunakan Dataset yang diambil dari Kaggle terdiri dari 9 kategori label yaitu 3 jenis jahe, 3 jenis kunyit, kencur, lengkuas dan temulawak yang total dataset yang dikumpulkan berjumlah x , yang sebelum di masukan ke dalam model, gambar-gambar tersebut melewati proses preprocessing terlebih dahulu .Penggunaan metode Convolutional Neural Network (CNN) menggunakan arsitektur MobileNetV2 untuk klasifikasi citra tanaman rempah dengan epoch sebanyak 10 dan batch size 64 menghasilkan akurasi hingga 90,4%. Model yang telah didapatkan kemudian di deploy menjadi aplikasi web menggunakan platform Streamlit, yang memungkinkan

pembangunan antarmuka pengguna interaktif. Kemudian ditampilkan hasil prediksi dari gambar yang telah diunggah oleh pengguna sebelumnya pada aplikasi web tersebut.

Saran

Hasil akurasi yang ditunjukkan dapat lebih ditingkatkan dengan menambah variabel dataset gambar rempah-rempah tersebut. Penggunaan dataset lain juga dapat disarankan untuk memperluas variasi pengenalan jenis rempah-rempah.

Referensi

- [1]A. E. Putra, M. F. Naufal, and V. R. Prasetyo, “Klasifikasi Jenis Rempah Menggunakan Convolutional Neural Network dan Transfer Learning,” *Jurnal Edukasi dan Penelitian Informatika (JEPIN)*, vol. 9, no. 1, p. 12, Apr. 2023, doi: 10.26418/jp.v9i1.58186.
- [2]N. P. Batubara, D. Widiyanto, and N. Chamidah, “Klasifikasi rempah rimpang berdasarkan ciri warna rgb dan tekstur glcm menggunakan algoritma naive bayes,” *Informatik : Jurnal Ilmu Komputer*, vol. 16, no. 3, p. 156, Dec. 2020, doi: 10.52958/iftk.v16i3.2196.
- [3]S. Y. Riska, “Klasifikasi Bumbu Dapur Indonesia Menggunakan Metode KNearest Neighbors (K-NN),” *SMATIKA Jurnal* , vol. Volume 11, no. 1, Jun. 2021.
- [4]S. Kilcommens, “Streamlit — Everything You Need To Know - DataDrivenInvestor,” *Medium*, Jan. 01, 2022.
<https://medium.datadriveninvestor.com/streamlit-everything-you-need-to-know-665eb90fcf4a#b992>