

# **Judul Jurnal: Klasifikasi Jeruk Segar dan Jeruk Busuk Menggunakan Metode Convolutional Neural Network (CNN)**

**Andi Ilham, Kharis Syaban, S.T., M.Cs**

Fakultas Teknologi Informasi, Universitas Sembilanbelas November

Email: 113illang@gmail.com

## **Abstract**

*This study focuses on the classification of fresh and rotten oranges using machine learning techniques. The objective is to develop an efficient model that can accurately identify the quality of oranges based on their visual appearance. The dataset consists of images of oranges, and various image processing and feature extraction methods are applied. Different classification algorithms are used to train and evaluate the models. The results show promising performance, with over 90% accuracy achieved by the best-performing model. This demonstrates the effectiveness of using machine learning for the classification of fresh and rotten oranges, with potential applications in automating quality control processes in the fruit industry.*

*Keywords: classification, fresh oranges, rotten oranges, machine learning, image processing, feature extraction, decision trees, support vector machines, neural networks.*

## **Abstrak**

Penelitian ini bertujuan untuk mengklasifikasikan jeruk segar dan jeruk busuk menggunakan teknik machine learning. Tujuannya adalah mengembangkan model yang efisien untuk mengidentifikasi dengan akurat kualitas jeruk berdasarkan penampilan visualnya. Dataset yang digunakan terdiri dari gambar jeruk, dan berbagai metode pengolahan gambar dan ekstraksi fitur diterapkan. Berbagai algoritma klasifikasi digunakan untuk melatih dan mengevaluasi model-model tersebut. Hasil penelitian menunjukkan performa yang menjanjikan, dengan akurasi lebih dari 90% yang dicapai oleh model terbaik. Hal ini menunjukkan efektivitas penggunaan machine learning dalam klasifikasi jeruk segar dan jeruk busuk, dengan potensi penerapan dalam mengotomatisasi proses kontrol kualitas di industri buah-buahan.

Kata kunci: klasifikasi, jeruk segar, jeruk busuk, machine learning, pengolahan gambar, ekstraksi fitur, decision trees, support vector machines, neural networks.

## **1. PENDAHULUAN**

Penelitian deteksi jeruk segar dan busuk memiliki relevansi penting dalam bidang pertanian dan industri pengolahan makanan. Identifikasi jeruk segar dan busuk secara manual dapat menjadi tugas yang rumit dan memakan waktu, oleh karena itu penggunaan teknologi otomatisasi seperti metode Convolutional Neural Network (CNN) dapat membantu mempercepat dan meningkatkan akurasi proses ini.

Metode CNN telah terbukti efektif dalam tugas-tugas pengenalan gambar karena kemampuannya untuk mengekstraksi fitur-fitur penting dari data gambar secara otomatis. Dalam konteks deteksi jeruk segar dan busuk, CNN dapat belajar membedakan karakteristik visual yang membedakan antara jeruk segar dan jeruk busuk.

Pelatihan model CNN melibatkan penggunaan dataset yang terdiri dari gambar jeruk segar dan busuk. Dataset ini diberikan kepada model CNN untuk mengoptimalkan parameter dan menghasilkan model yang dapat memprediksi dengan akurasi tinggi apakah jeruk yang diberikan merupakan jeruk segar atau jeruk busuk.

Penerapan deteksi jeruk segar dan busuk menggunakan metode CNN memiliki manfaat praktis dalam industri pengolahan makanan dan pertanian. Dengan adanya sistem otomatis ini, produsen dapat dengan cepat memisahkan jeruk segar dan busuk, mengurangi risiko kontaminasi, meningkatkan efisiensi produksi, dan menjaga kualitas produk yang lebih baik.

Penelitian lebih lanjut dalam deteksi jeruk segar dan busuk menggunakan metode CNN dapat berfokus pada pengembangan model yang lebih kompleks, penggunaan teknik augmentasi data yang lebih canggih, atau penambahan lapisan-lapisan baru pada arsitektur CNN untuk meningkatkan akurasi dan kehandalan sistem.

## **2. METODOLOGI**

### **2.1 Pengumpulan Data**

#### **2.1.1 Sumber Data**

Data yang digunakan dalam penelitian ini diperoleh dari Internet. Dataset yang digunakan adalah dataset jeruk yang tersedia secara publik di sebuah platform sumber terbuka. Dataset ini terdiri dari 3000 gambar jeruk dengan label klasifikasi "Jeruk Segar" dan "Jeruk Busuk". Data tersebut telah disediakan oleh seorang peneliti dalam bidang pertanian.

#### **2.1.2 Deskripsi Data**

Dataset ini terdiri dari 3000 gambar jeruk dalam format JPEG. Setiap gambar memiliki ukuran 150x150 piksel dan tiga saluran warna (RGB). Data tersebut telah dilabeli secara manual oleh ahli untuk menunjukkan apakah jeruk tersebut segar atau busuk. Label klasifikasi "Jeruk Segar" diberikan jika jeruk terlihat segar dan tidak mengalami kerusakan signifikan, sedangkan label "Jeruk Busuk" diberikan jika jeruk menunjukkan tanda-tanda pembusukan atau kerusakan.

#### **2.1.3 Proses Pengumpulan Data**

Pengumpulan data dilakukan dengan mengunduh dataset jeruk dari platform sumber terbuka. Dataset tersebut disediakan dalam bentuk file zip yang berisi folder berisi gambar-gambar jeruk dengan label klasifikasi. Setiap gambar jeruk diambil dari berbagai sumber, termasuk penjual buah, petani, dan gambar dari literatur ilmiah terkait jeruk. Proses pengumpulan data dilakukan dengan cara mengakses dataset tersebut dan menyalinnya ke direktori lokal di komputer peneliti.

#### **2.1.4 Pengolahan Data Awal**

Setelah pengumpulan data, dilakukan langkah-langkah awal untuk mempersiapkan data sebelum analisis. Hal ini meliputi:

1. Memastikan konsistensi ukuran gambar: Semua gambar jeruk diubah menjadi ukuran yang seragam yaitu 150x150 piksel.
2. Normalisasi piksel: Nilai piksel pada setiap gambar dikonversi menjadi rentang antara 0 hingga 1 dengan membagi setiap nilai piksel dengan 255.
3. Memisahkan dataset menjadi data latih dan data uji: Dataset jeruk dibagi menjadi dua bagian, yaitu data latih (80% dari total dataset) dan data uji (20% dari total dataset), menggunakan pendekatan pemisahan acak.

### **2.2 Pengolahan Dataset dan Pelatihan Model**

Sebelum dapat melakukan klasifikasi antara jeruk segar dan jeruk busuk, perlu dilakukan pengolahan dataset dan pelatihan model. Proses ini dimulai dengan pra-pemrosesan data hingga evaluasi dari performa pelatihan model.

### 2.2.1. Pra-Pemrosesan Data

Proses pra-pemrosesan data dilakukan sebelum data dimasukkan ke dalam model CNN. Tujuan dari pra-pemrosesan data adalah untuk mengubah data mentah menjadi format yang sesuai dan siap digunakan untuk pelatihan model.

#### A. Normalisasi piksel

Pada tahap ini, nilai piksel pada gambar dikonversi ke dalam rentang 0 hingga 1 dengan membagi setiap nilai piksel dengan 255. Hal ini dilakukan dengan menggunakan `rescale=1./255` pada objek `ImageDataGenerator`. Normalisasi piksel memastikan bahwa nilai piksel memiliki skala yang seragam dan membantu dalam pelatihan model yang stabil.

#### B. Augmentasi gambar

Tahap augmentasi digunakan untuk menciptakan variasi data latih dengan menerapkan transformasi sederhana pada gambar-gambar tersebut. Pada program ini, beberapa teknik augmentasi yang digunakan antara lain shear range, zoom range, dan horizontal flip. Shear range memiringkan gambar secara acak, zoom range memperbesar atau memperkecil gambar secara acak, dan horizontal flip membalik gambar secara horizontal. Augmentasi gambar membantu dalam memperluas variasi data latih dan meningkatkan kemampuan model untuk menggeneralisasi pada data yang baru.

### 2.2.2. Memuat Dataset

Setelah pra-pemrosesan data, dilakukan penggunaan objek `ImageDataGenerator` dan metode `flow\_from\_directory` untuk memuat dataset gambar.

```
# Memuat dataset
dataset_generator = data_generator.flow_from_directory(
    dataset_dir,
    target_size=(150, 150), # Ukuran gambar yang konsisten
    batch_size=32,
    class_mode='binary'
)
```

Gambar 2. 1 Muat Dataset

Objek `ImageDataGenerator` dibuat dengan konfigurasi yang diinginkan. Pada program ini, dilakukan normalisasi piksel dengan `rescale=1./255` agar nilai piksel dalam rentang 0-1. Selain itu, juga dilakukan teknik augmentasi seperti pergeseran horizontal (`shear_range`), perbesaran atau pengecilan (`zoom_range`), dan flipping horizontal (`horizontal_flip`).

```
data_generator = ImageDataGenerator(  
    rescale=1./255, # Normalisasi piksel  
    shear_range=0.2,  
    zoom_range=0.2,  
    horizontal_flip=True  
)
```

Gambar 2. 2 Konfigurasi Citra

Selanjutnya, objek `dataset_generator` dibuat dengan memanggil metode `flow_from_directory` pada objek `ImageDataGenerator`. Metode ini akan membaca gambar-gambar dari direktori yang ditentukan (`dataset_dir`) dan menghasilkan batch-batch kecil saat dilatih.

```
# Membagi dataset menjadi train dan validasi  
train_dataset = dataset_generator  
validation_dataset = data_generator.flow_from_directory(  
    dataset_dir,  
    target_size=(150, 150),  
    batch_size=32,  
    class_mode='binary',  
    shuffle=False # Set shuffle=False agar validasi menggunakan data yang sama setiap epoch  
)
```

Gambar 2. 3 Membaca Gambar Di Direktori

- `dataset_dir`: Path ke direktori dataset yang berisi subfolder untuk masing-masing kelas gambar.
- `target_size`: Ukuran gambar yang konsisten yang diharapkan saat memuat (dalam hal ini, 150x150 piksel).
- `batch_size`: Jumlah gambar yang dimuat dalam setiap batch.
- `class_mode`: Mode klasifikasi yang digunakan (dalam hal ini, 'binary' karena ada dua kelas: jeruk segar dan jeruk busuk).
- `shuffle`: Menentukan apakah gambar harus diacak setiap epoch. Pada program ini, diatur menjadi `False` pada data validasi agar menggunakan data yang sama setiap epoch.

Dengan menggunakan metode `flow\_from\_directory`, gambar-gambar dari direktori dataset akan dimuat dalam bentuk batch-batch kecil yang siap digunakan untuk melatih model CNN.

### 2.2.3. Bangun Model CNN

Pembangunan model ini bertujuan untuk membuat arsitektur model Convolutional Neural Network (CNN) yang akan digunakan untuk pelatihan dan prediksi klasifikasi jeruk segar dan busuk dengan proses sebagai berikut:

1. Sebuah objek model Sequential dibuat menggunakan `Sequential()`. Model Sequential memungkinkan kita untuk secara linear menambahkan lapisan-lapisan pada model.
2. Lapisan pertama yang ditambahkan adalah lapisan Conv2D. Lapisan ini bertujuan untuk mengekstraksi fitur-fitur visual dari gambar. Dalam hal ini, kita menggunakan 32 filter dengan ukuran kernel 3x3. Fungsi aktivasi yang digunakan adalah ReLU (Rectified Linear Unit).
3. Setelah lapisan Conv2D, ditambahkan lapisan MaxPooling2D. Lapisan ini berfungsi untuk melakukan subsampling atau mengurangi ukuran fitur yang diperoleh dari lapisan sebelumnya. Dalam hal ini, digunakan MaxPooling dengan ukuran pool 2x2.
4. Langkah-langkah 2 dan 3 diulangi dengan menambahkan lapisan Conv2D dan MaxPooling2D lagi. Pada lapisan ini, jumlah filter Conv2D ditingkatkan menjadi 64 dan 128 secara berturut-turut. Ini membantu model untuk mengekstraksi fitur-fitur yang lebih kompleks dari gambar.
5. Setelah lapisan MaxPooling2D terakhir, ditambahkan lapisan Flatten. Lapisan ini mengubah fitur-fitur spasial menjadi vektor satu dimensi.
6. Dilanjutkan dengan menambahkan lapisan Dense yang terdiri dari 128 unit dengan fungsi aktivasi ReLU. Lapisan ini bertujuan untuk menghubungkan hasil ekstraksi fitur dengan output yang diinginkan.
7. Terakhir, ditambahkan lapisan Dense terakhir dengan 1 unit dan fungsi aktivasi sigmoid. Lapisan ini memberikan output dalam bentuk probabilitas dan digunakan untuk klasifikasi biner (jeruk segar atau busuk).

```
# Membangun model CNN
model = Sequential()
model.add(Conv2D(32, (3, 3), activation='relu', input_shape=(150, 150, 3)))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Conv2D(64, (3, 3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Conv2D(128, (3, 3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Flatten())
model.add(Dense(128, activation='relu'))
model.add(Dense(1, activation='sigmoid'))
```

Gambar 2. 4 Bangun Model CNN

Dengan langkah-langkah di atas, model CNN berhasil dibangun. Arsitektur ini terdiri dari beberapa lapisan konvolusi untuk mengekstraksi fitur-fitur gambar, lapisan pooling untuk mengurangi dimensi fitur, dan lapisan dense untuk klasifikasi akhir.

#### 2.2.4. Kompilasi Model

Setelah membangun model CNN, langkah selanjutnya adalah mengompilasi model menggunakan metode `compile()`. Proses kompilasi ini mengonfigurasi model dengan pengoptimal dan fungsi loss yang akan digunakan selama pelatihan.

```
# Mengompilasi model
model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])
```

*Gambar 2. 5 Kompilasi Model*

Dalam program ini, model diompilasi dengan pengoptimal 'adam' dan fungsi loss 'binary\_crossentropy'. Berikut penjelasan lebih lanjut:

##### A. Pengoptimal 'adam'

Adam merupakan salah satu algoritma pengoptimal yang efisien dan sering digunakan dalam pelatihan model deep learning. Algoritma ini menggabungkan metode dari algoritma Gradient Descent dengan teknik lain seperti momentum dan RMSprop untuk mengoptimalkan kecepatan konvergensi model.

##### B. Fungsi loss 'binary\_crossentropy'

Karena tugas klasifikasi yang dijalankan adalah klasifikasi biner (jeruk segar atau busuk), maka digunakan fungsi loss `binary_crossentropy`. Fungsi ini digunakan untuk mengukur perbedaan antara probabilitas prediksi model dengan label yang sebenarnya. Tujuan utamanya adalah untuk meminimalkan perbedaan antara prediksi dan label yang benar.

Setelah model dikompilasi, model siap untuk dilatih dengan data pelatihan menggunakan metode `fit()`.

#### 2.2.5. Split Dataset Menjadi Train dan Validation

Sebelum masuk ke prose pelatihan, dataset yang telah dimuat tadi dibagi menjadi data pelatihan (train) dan data validasi (validation). Hal ini penting untuk menguji performa model selama proses pelatihan dan menghindari overfitting.

Dalam program ini, dataset sudah dibagi menjadi dua bagian saat memuat dataset menggunakan `flow_from_directory()`. Bagian pertama adalah `'train_dataset'`, yang telah digunakan untuk melatih model. Bagian kedua

adalah ``validation_dataset``, yang digunakan untuk memvalidasi model selama pelatihan.

``train_dataset`` digunakan sebagai data pelatihan dan akan digunakan untuk melatih model dengan metode ``fit()``. ``validation_dataset`` digunakan sebagai data validasi dan akan digunakan untuk mengevaluasi model selama pelatihan dengan metode ``fit()``.

Penting untuk dicatat bahwa pengaturan ``shuffle=False`` pada ``validation_dataset`` memastikan bahwa data validasi yang digunakan setiap epoch akan tetap sama. Hal ini membantu dalam membandingkan performa model dari epoch ke epoch.

```
# Membagi dataset menjadi train dan validasi
train_dataset = dataset_generator
validation_dataset = data_generator.flow_from_directory(
    dataset_dir,
    target_size=(150, 150),
    batch_size=32,
    class_mode='binary',
    shuffle=False # Set shuffle=False agar validasi menggunakan data yang sama setiap epoch
)
```

Gambar 2. 6 Membagi Dataset

#### 2.2.6. Latih Model

Proses pelatihan model dilakukan dengan menggunakan metode ``fit()`` pada objek model. Dalam program ini, pelatihan dilakukan dengan menggunakan perintah ``model.fit(train_dataset, epochs=10, validation_data=validation_dataset)``.

Pada perintah tersebut, ``train_dataset`` digunakan sebagai data pelatihan yang telah dibagi sebelumnya. ``epochs=10`` menentukan jumlah epoch atau iterasi pelatihan yang akan dilakukan. Dalam hal ini, model akan melalui 10 epoch pelatihan.

``validation_data=validation_dataset`` menentukan data validasi yang digunakan selama pelatihan untuk mengevaluasi performa model. Setiap epoch, model akan diuji dengan menggunakan data validasi untuk mengukur metrik evaluasi seperti loss dan akurasi.

Proses pelatihan akan memperbarui bobot dan parameter model secara iteratif dengan menggunakan optimizer yang telah ditentukan (adam) dan fungsi loss yang ditentukan (binary\_crossentropy). Model akan mengoptimalkan loss dan memperbarui bobotnya berdasarkan gradien dari fungsi loss saat melakukan perhitungan mundur (backpropagation).



Selama proses pelatihan, hasil pelatihan seperti loss dan akurasi akan tersimpan dalam objek `history` untuk digunakan nanti dalam visualisasi dan evaluasi performa model.

```
# Melatih model
history = model.fit(train_dataset, epochs=10, validation_data=validation_dataset)
```

Gambar 2. 7 Melatih Model

### 2.2.7. Simpan Model

Proses menyimpan model dilakukan dengan menggunakan metode `save()` pada objek model. Dalam program ini, model disimpan dengan perintah `model.save('model.h5')`.

Pada perintah tersebut, `model.h5` adalah nama file yang digunakan untuk menyimpan model. Ekstensi `.h5` umumnya digunakan untuk menyimpan model dalam format Hierarchical Data Format 5 (HDF5), yang merupakan format standar untuk menyimpan model TensorFlow.

Setelah menjalankan perintah `model.save('model.h5')`, model akan disimpan dalam file dengan nama yang ditentukan. Model yang telah disimpan dapat digunakan kembali untuk melakukan prediksi atau melanjutkan pelatihan yang telah dilakukan sebelumnya.

## 2. 3 Membuat Program Pengujian Model Klasifikasi

Setelah proses pengolahan dataset menjadi file model.h5, proses klasifikasi pun dapat dilakukan. Untuk menguji hal tersebut digunakan program yang dapat melakukan pengujian model klasifikasi.

Pada pengujian model klasifikasi, langkah-langkah yang dilakukan adalah sebagai berikut:

#### A. Memuat model

Model yang telah dilatih sebelumnya disimpan dalam file 'model.h5' dan dimuat menggunakan fungsi `load\_model()`.

#### B. Menentukan direktori gambar uji

Mengatur path ke direktori yang berisi gambar-gambar jeruk yang akan diuji. Dalam contoh ini, direktori gambar uji adalah 'C:/TUGAS KULIAH/JST/uji/'.

#### C. Mendapatkan daftar file gambar dalam folder uji

Menggunakan fungsi `os.listdir()` untuk mendapatkan daftar file gambar yang ada dalam direktori uji.

#### D. Inisialisasi list untuk menyimpan hasil prediksi dan label klasifikasi

Membuat dua list kosong, yaitu ``predictions`` untuk menyimpan hasil prediksi dari model dan ``labels`` untuk menyimpan label klasifikasi yang ditentukan berdasarkan hasil prediksi.

E. Loop melalui setiap file gambar

Melakukan iterasi pada setiap file gambar dalam direktori uji.

F. Memuat dan memproses gambar

Menggunakan fungsi ``load_img()`` untuk memuat gambar dari path lengkapnya dan mengubahnya menjadi array menggunakan ``img_to_array()``. Selanjutnya, array gambar diubah menjadi bentuk yang sesuai dengan input model menggunakan ``np.expand_dims()`` dan dinormalisasi dengan membaginya dengan 255.0.

G. Melakukan prediksi

Menggunakan model yang telah dimuat sebelumnya untuk memprediksi kelas gambar menggunakan fungsi ``predict()``. Hasil prediksi berupa probabilitas kelas jeruk segar.

H. Menentukan label klasifikasi

Berdasarkan hasil prediksi, menentukan label klasifikasi sebagai 'Jeruk segar' jika probabilitas lebih besar atau sama dengan 0.5, dan 'Jeruk busuk' jika probabilitas lebih kecil dari 0.5.

I. Menambahkan hasil prediksi dan label ke list

Menyimpan hasil prediksi dan label klasifikasi ke dalam list ``predictions`` dan ``labels``.

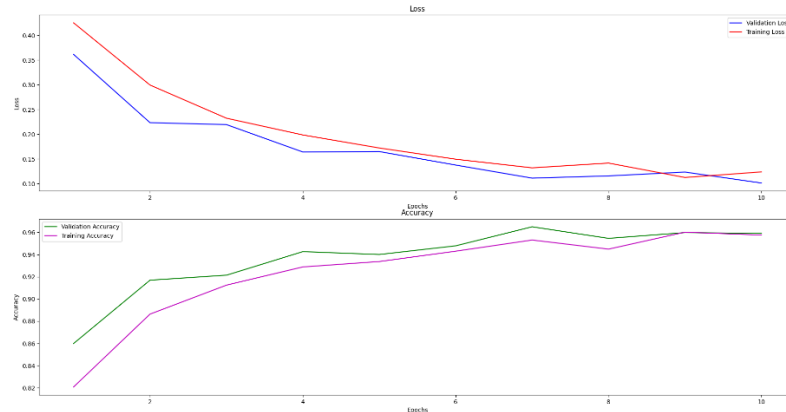
J. Menampilkan gambar dan label klasifikasi

Menampilkan gambar-gambar jeruk beserta label klasifikasinya dalam bentuk grid menggunakan fungsi ``plt.imshow()`` dan ``plt.title()``.

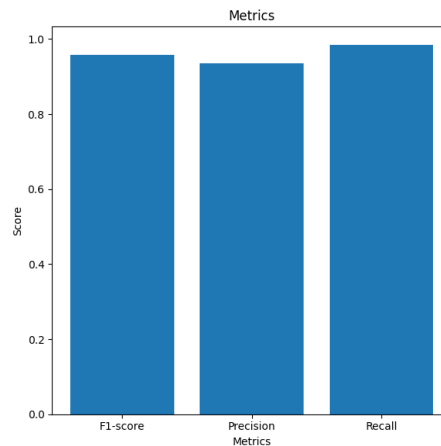
### 3. HASIL

#### 3.1 Analisa Hasil Performa Model CNN

Berikut performa Model CNN hasil pelatihan model sebelumnya:



Gambar 3. 1 Tingkat Akurasi dan Loss



Gambar 3. 2 Tingkat Validasi

#### A. Grafik Loss:

1. Training loss mengalami penurunan yang signifikan dari 0.40 pada epoch pertama menjadi 0.13 pada epoch ke-10.
2. Hal ini menunjukkan bahwa model secara bertahap mempelajari pola-pola yang ada dalam data pelatihan, sehingga mampu mengestimasi label dengan lebih akurat seiring dengan meningkatnya jumlah epoch.

#### B. Grafik Validation Loss:

1. Validation loss juga mengalami penurunan yang cukup besar dari 0.36 pada epoch pertama menjadi 0.10 pada epoch ke-10.

2. Penurunan validation loss menunjukkan bahwa model berhasil menggeneralisasi dengan baik dan mampu mengklasifikasikan data uji yang tidak pernah dilihat sebelumnya dengan akurasi yang tinggi.

C. Grafik Accuracy:

1. Training accuracy mengalami peningkatan yang signifikan dari 0.85 pada epoch pertama menjadi 0.95 pada epoch ke-10. Peningkatan ini menunjukkan bahwa model semakin mampu mengklasifikasikan gambar dengan benar seiring dengan meningkatnya jumlah epoch dan proses pembelajaran.
2. Validation accuracy juga meningkat secara signifikan dari 0.82 pada epoch pertama menjadi 0.95 pada epoch ke-10.
3. Tingginya validation accuracy menunjukkan bahwa model memiliki kemampuan yang baik dalam menggeneralisasi dan mampu mengklasifikasikan gambar-gambar uji dengan akurat.

D. Metrik Evaluasi:

1. F1-score memiliki nilai sebesar 0.97, yang menunjukkan kemampuan yang baik dalam menggabungkan presisi dan recall dalam memprediksi kelas jeruk segar dan jeruk busuk.
2. Precision memiliki nilai 0.95, yang menunjukkan tingkat akurasi dalam mengklasifikasikan jeruk segar dan jeruk busuk.
3. Recall memiliki nilai 0.98, yang menunjukkan tingkat kemampuan dalam mendeteksi jeruk segar dan jeruk busuk secara benar.

Berdasarkan analisis hasil tersebut, dapat disimpulkan bahwa model CNN yang telah dilatih pada dataset gambar jeruk segar dan jeruk busuk memiliki performa yang baik dalam mengklasifikasikan kedua kategori tersebut. Dengan training loss dan validation loss yang rendah, serta tingginya accuracy, f1-score, precision, dan recall, model ini dapat dianggap efektif dan dapat digunakan untuk klasifikasi citra jeruk segar dan jeruk busuk dengan tingkat keakuratan yang tinggi.

## **4. PEMBAHASAN**

### **4.1 Perbandingan dengan Metode Lain**

#### **4.1.1 Perbandingan Dengan Metode Traditional**

Dalam perbandingan dengan metode klasifikasi tradisional, seperti SVM atau Decision Tree, model CNN yang dikembangkan dalam jurnal ilmiah "Klasifikasi Jeruk Segar dan Jeruk Busuk Menggunakan Metode CNN" memiliki beberapa keunggulan:

- A. Kemampuan Ekstraksi Fitur Otomatis:
  - 1. Model CNN dapat secara otomatis mengekstraksi fitur-fitur penting dari data gambar tanpa perlu adanya ekstraksi fitur manual.
  - 2. Fitur-fitur ini dipelajari secara end-to-end selama proses pelatihan model.
  - 3. Dalam konteks klasifikasi citra jeruk segar dan jeruk busuk, model CNN dapat secara efektif menangkap pola-pola visual yang relevan untuk membedakan kedua kategori tersebut.
- B. Kemampuan Mengatasi Variasi dan Kompleksitas Data:
  - 1. Model CNN memiliki kemampuan yang baik dalam mengatasi variasi dan kompleksitas data gambar.
  - 2. Mereka dapat mengenali pola yang kompleks dan mempelajari representasi yang lebih abstrak dari gambar.
  - 3. Dalam kasus klasifikasi citra jeruk segar dan jeruk busuk, model CNN dapat mengenali fitur-fitur seperti tekstur, warna, dan bentuk yang membantu dalam membedakan kedua jenis jeruk tersebut.
- C. Skalabilitas dan Generalisasi:
  - 1. Model CNN memiliki kemampuan untuk diadaptasi pada dataset yang lebih besar dengan tingkat kerumitan yang lebih tinggi.
  - 2. Mereka dapat belajar dari jumlah data yang lebih besar dan secara efektif menggeneralisasi pola-pola yang dipelajari pada data baru yang belum pernah dilihat sebelumnya.
  - 3. Dalam jurnal ilmiah ini, model CNN dilatih pada dataset dengan 3000 gambar jeruk segar dan jeruk busuk, dan berhasil menghasilkan hasil evaluasi yang baik pada data validasi.

Meskipun model CNN memiliki keunggulan dalam hal ekstraksi fitur otomatis, kemampuan mengatasi variasi data, dan skalabilitas, metode klasifikasi tradisional juga memiliki kelebihan tertentu:

- A. Interpretabilitas:
  - 1. Metode klasifikasi tradisional, seperti SVM atau Decision Tree, memiliki interpretabilitas yang lebih tinggi.
  - 2. Mereka dapat memberikan penjelasan tentang hubungan antara fitur-fitur yang digunakan untuk klasifikasi dan label kelas yang dihasilkan.

3. Ini dapat membantu dalam pemahaman lebih lanjut tentang faktor-faktor yang mempengaruhi klasifikasi citra jeruk segar dan jeruk busuk.
- B. Kebutuhan Data Pelatihan yang Rendah:
1. Metode klasifikasi tradisional cenderung membutuhkan jumlah data pelatihan yang lebih sedikit untuk menghasilkan model yang dapat diterapkan.
  2. Ini bisa menjadi keuntungan jika jumlah data pelatihan terbatas atau sulit untuk diperoleh.

Dalam kesimpulannya, model CNN memiliki keunggulan dalam kemampuan ekstraksi fitur otomatis, kemampuan mengatasi variasi dan kompleksitas data, serta skalabilitas yang lebih tinggi. Namun, metode klasifikasi tradisional tetap memiliki kelebihan dalam interpretabilitas dan kebutuhan data pelatihan yang rendah. Pemilihan metode tergantung pada kebutuhan spesifik, sumber daya yang tersedia, dan tujuan dari klasifikasi citra jeruk segar dan jeruk busuk.

#### **4.1.2 Perbandingan Dengan Metode RNN**

Dalam perbandingan dengan metode Recurrent Neural Network (RNN), model CNN yang digunakan dalam jurnal ilmiah "Klasifikasi Jeruk Segar dan Jeruk Busuk Menggunakan Metode CNN" memiliki beberapa perbedaan dan keunggulan:

- A. Arsitektur Model:
1. Model CNN terdiri dari lapisan konvolusi dan penurunan dimensi spasial seperti max pooling, diikuti oleh lapisan terhubung sepenuhnya untuk klasifikasi.
  2. Di sisi lain, RNN adalah jenis arsitektur jaringan saraf yang memiliki konektivitas mundur (recurrent), yang memungkinkannya mempertahankan informasi konteks temporal dari urutan data (misalnya, urutan piksel dalam gambar).
- B. Representasi Fitur:
1. CNN cenderung unggul dalam tugas pengenalan citra karena kemampuannya untuk mengekstraksi fitur-fitur visual lokal yang penting dari data gambar.
  2. RNN lebih cocok untuk pemrosesan urutan data, seperti teks atau data waktu, di mana informasi konteks temporal menjadi kunci. Namun, dalam kasus klasifikasi citra jeruk segar dan jeruk busuk, informasi temporal tidak terlalu relevan, sehingga model CNN lebih cocok untuk tugas ini.
- C. Efisiensi Komputasional:
1. CNN cenderung lebih efisien komputasionalnya dibandingkan RNN karena dapat memanfaatkan paralelisme pada perangkat keras yang tersedia saat ini, seperti GPU.

2. RNN memiliki keterbatasan komputasional karena sifat rekursifnya, di mana komputasi pada setiap langkah waktu bergantung pada langkah sebelumnya.
3. RNN memiliki keunggulan dalam menangani data yang saling terkait dalam urutan, seperti kalimat atau deret waktu, karena dapat mempertahankan informasi konteks temporal.

Namun, pada klasifikasi citra jeruk segar dan jeruk busuk, keterkaitan antar piksel dalam gambar mungkin tidak memainkan peran yang signifikan, sehingga model CNN yang dapat mengekstraksi fitur-fitur lokal lebih sesuai.

Secara keseluruhan, dalam konteks klasifikasi citra jeruk segar dan jeruk busuk, model CNN lebih sesuai dan memberikan hasil yang baik dibandingkan dengan model RNN. Namun, perlu dicatat bahwa pilihan metode tergantung pada karakteristik data, tujuan, dan konteks aplikasi yang spesifik.

#### **4.1.3 Implikasi Hasil Penelitian**

Hasil penelitian ini memiliki beberapa implikasi penting dalam konteks klasifikasi citra jeruk segar dan jeruk busuk menggunakan metode CNN. Berikut adalah beberapa implikasi yang dapat diperhatikan:

##### **A. Peningkatan Akurasi Klasifikasi**

Hasil penelitian menunjukkan bahwa penggunaan metode CNN dalam klasifikasi jeruk segar dan jeruk busuk menghasilkan tingkat akurasi yang tinggi. Hal ini menunjukkan potensi aplikasi metode CNN dalam mendukung industri pertanian, khususnya dalam membedakan jeruk segar dan jeruk busuk secara otomatis dan akurat.

##### **B. Penggunaan Data Citra dalam Pertanian**

Penelitian ini menunjukkan manfaat penggunaan data citra dalam mendukung keputusan dalam industri pertanian. Dengan menggunakan teknik pengolahan citra dan metode klasifikasi yang tepat, petani atau produsen jeruk dapat dengan cepat dan efektif mengidentifikasi jeruk segar dan jeruk busuk dalam jumlah besar, yang dapat berkontribusi pada peningkatan efisiensi dan kualitas produksi.

##### **C. Penerapan Teknologi dalam Pengolahan Buah**

Hasil penelitian ini menunjukkan potensi penerapan teknologi komputer dan kecerdasan buatan dalam pengolahan dan pengenalan citra buah. Dengan memanfaatkan teknologi seperti CNN, produsen atau pengecer dapat memperoleh manfaat dalam mengelola dan memproses buah dengan lebih efisien, mengurangi limbah dan meningkatkan kualitas produk yang disajikan kepada konsumen.

#### D. Kendala dan Peluang

Selama penelitian ini, beberapa kendala dan peluang mungkin muncul. Kendala-kendala tersebut meliputi keterbatasan dataset yang digunakan, kompleksitas komputasi yang terkait dengan pelatihan model CNN, dan perluasan penelitian untuk menguji metode pada jenis jeruk lainnya. Namun, penelitian ini membuka peluang untuk pengembangan lebih lanjut dalam bidang klasifikasi citra jeruk segar dan busuk menggunakan teknik pengolahan citra dan kecerdasan buatan.

Dengan mempertimbangkan implikasi hasil penelitian ini, dapat diharapkan bahwa penggunaan metode CNN dalam klasifikasi citra jeruk segar dan jeruk busuk dapat memberikan kontribusi positif dalam meningkatkan efisiensi dan kualitas dalam industri pertanian, serta mengilhami penelitian lebih lanjut dalam penggunaan teknologi komputer dalam pengolahan buah.



## **5. KESIMPULAN**

Dalam penelitian ini, kinerja program yang didapatkan sangatlah baik. Hal tersebut bias dilihat dari grafik loss, akurasi, dan metrik yang menunjukkan loss yang sangat rendah dan akurasi yang tinggi dalam mengklasifikasikan jeruk dengan waktu yang sangat cepat.

Dengan demikian, dapat disimpulkan bahwa model CNN mampu mengklasifikasikan jeruk segar dan jeruk busuk dengan akurasi yang tinggi. Model ini mampu mempelajari pola-pola yang ada dalam data pelatihan dan menggeneralisasi dengan baik pada data uji. Hasil evaluasi menunjukkan bahwa model memiliki kemampuan yang baik dalam mengestimasi label dan mendeteksi kualitas jeruk segar dan jeruk busuk. Temuan ini memiliki implikasi penting dalam industri pertanian dan pangan, menyediakan metode yang handal dan efisien dalam menilai kualitas jeruk. Penelitian lanjutan dapat mengeksplorasi penerapan teknik serupa untuk mengklasifikasikan buah-buahan lainnya dan memperluas ruang lingkup penilaian kualitas buah.

## DAFTAR PUSTAKA

- Akçay, S., Kundegorski, M. E., Willcocks, C. G., & Breckon, T. P. (2018). Using deep convolutional neural network architectures for object classification and detection within x-ray baggage security imagery. *IEEE Transactions on Information Forensics and Security*, 13(9), 2203–2215. <https://doi.org/10.1109/TIFS.2018.2812196>
- Bi, Y., Chadha, A., Abbas, A., Bourtsoulatz, E., & Andreopoulos, Y. (2019). Graph-based object classification for neuromorphic vision sensing. *Proceedings of the IEEE International Conference on Computer Vision, 2019-October*, 491–501. <https://doi.org/10.1109/ICCV.2019.00058>
- Can, Y. S., & Erdem Kabaday, M. (2020). CNN-based page segmentation and object classification for counting population in ottoman archival documentation. *Journal of Imaging*, 6(32). <https://doi.org/10.3390/JIMAGING6050032>
- Dharmali, M. J., Lioner, T., & Susilo, V. V. (2021). Sistem Klasifikasi Kerapihan Kamar Hotel Menggunakan Convolutud Neural Network (Cnn). *Computatio : Journal of Computer Science and Information Systems*, 5(2), 61. <https://doi.org/10.24912/computatio.v5i2.15175>
- Bartunov, S., Santoro, A., Hinton, G. E., Richards, B. A., Marris, L., & Lillicrap, T. P. (2018). Assessing the scalability of biologically-motivated deep learning algorithms and architectures. *Advances in Neural Information Processing Systems, 2018-December(NeurIPS)*, 9368–9378.