

Klasifikasi Penyakit Daun Jagung Menggunakan Convolutional Neural Network Arsitektur Xception Berbasis Web

Sitti Aisyah Asy'ari¹, Anis Yusrotun Nadhiroh², Fathorazi Nur Fajri³

^{1,2,3} Universitas Nurul Jadid Probolinggo, Indonesia

Email: sitiaisyah08338@gmail.com¹, anis@unuja.ac.id², fathorozi@umija.ac.id³

Received: 24 Agustus 2025

Revised: 25 Agustus 2025

Accepted: 01 September 2025

Abstract:

Corn is one of the strategic food commodities in Indonesia that is very important for food security and the animal feed industry. However, corn productivity often decreases due to leaf diseases such as gray leaf spot, common rust, and blight. Manual disease detection by farmers is considered less accurate and inefficient. This research proposes a corn leaf disease classification system using Convolutional Neural Network (CNN) with Xception architecture, integrated into a web-based Flask application for easy access by farmers. The research dataset consists of 4,188 images of corn leaves divided into four classes: healthy, gray_leaf_spot, common_rust, and blight. The data is divided into 80% training, 10% validation, and 10% testing. Preprocessing is done through image augmentation (rotation, flipping, zooming) and pixel normalization. The CNN model is built with pretrained Xception (without the top layer) and layers of Conv2D, MaxPooling, Dropout, Flatten, and Dense with a softmax activation function added. Evaluation uses metrics of accuracy, precision, recall, F1-score, and confusion matrix. The research results show an accuracy of 92.40% with the best performance on the healthy class and the lowest on the gray_leaf_spot class (68.97%). The system has been successfully implemented into a web application that can classify corn leaf diseases automatically. This research demonstrates the effectiveness of the combination of CNN and Xception for plant disease detection and contributes to the development of digital agriculture based on artificial intelligence.

Keywords: CNN, Xception, Corn Leaf Disease, Deep Learning, Flask

Abstrak:

Jagung merupakan salah satu komoditas pangan strategis di Indonesia yang sangat penting bagi ketahanan pangan dan industri pakan ternak. Namun, produktivitas jagung seringkali menurun akibat penyakit daun seperti *gray leaf spot*, *common rust*, dan *blight*. Deteksi penyakit secara manual oleh petani dinilai kurang akurat dan tidak efisien. Penelitian ini mengusulkan sistem klasifikasi penyakit daun jagung menggunakan Convolutional Neural Network (CNN) dengan arsitektur Xception, yang diintegrasikan ke dalam aplikasi berbasis web Flask agar mudah diakses oleh petani. Dataset penelitian terdiri dari 4.188 gambar daun jagung yang terbagi menjadi empat kelas: *healthy*, *gray_leaf_spot*, *common_rust*, dan *blight*. Data dibagi menjadi 80% pelatihan, 10% validasi, dan 10% pengujian. Preprocessing dilakukan melalui augmentasi citra (rotasi, flipping, zooming) serta normalisasi piksel. Model CNN dibangun dengan pretrained Xception (tanpa top layer) dan ditambahkan layer Conv2D, MaxPooling, Dropout, Flatten, dan Dense dengan fungsi aktivasi softmax. Evaluasi menggunakan metrik akurasi, presisi, recall, F1-score, serta confusion matrix. Hasil penelitian menunjukkan akurasi sebesar 92,40% dengan performa terbaik pada kelas *healthy* dan

terendah pada kelas *gray_leaf_spot* (68,97%). Sistem berhasil diimplementasikan ke dalam aplikasi web yang dapat mengklasifikasikan penyakit daun jagung secara otomatis. Penelitian ini membuktikan efektivitas kombinasi CNN dan Xception untuk deteksi penyakit tanaman serta berkontribusi dalam pengembangan pertanian digital berbasis kecerdasan buatan.

Kata Kunci: CNN, Xception, Penyakit Daun Jagung, Deep Learning, Flask

PENDAHULUAN

Jagung merupakan komoditas penting dalam sektor pertanian Indonesia, tidak hanya sebagai bahan pangan tetapi juga pakan ternak, industri pangan olahan, hingga bioenergi. Berdasarkan data Badan Pusat Statistik (BPS, 2023), produksi jagung nasional mencapai 14,46 juta ton dengan luas panen sebesar 2,49 juta hektare. Provinsi Jawa Timur tercatat sebagai penyumbang terbesar produksi jagung nasional, menjadikannya daerah yang sangat strategis untuk ketahanan pangan. Namun demikian, tren produktivitas jagung di Indonesia masih fluktuatif dan rentan terhadap berbagai faktor penghambat, salah satunya serangan penyakit daun. Data terbaru menunjukkan bahwa serangan penyakit dapat menurunkan hasil panen hingga 37% apabila tidak ditangani dengan cepat dan tepat, kondisi ini tentu menjadi ancaman serius bagi keberlangsungan ketahanan pangan nasional.

Penyakit daun pada jagung seperti *common rust*, *gray leaf spot*, dan *blight* menyerang jaringan fotosintesis yang vital bagi pertumbuhan tanaman. Dampaknya tidak hanya menurunkan produktivitas petani, tetapi juga merembet pada sektor industri pakan ternak serta rantai pasok pangan nasional. Penelitian sebelumnya (Hermawan, 2023) menegaskan bahwa kerugian akibat penyakit jagung bersifat jangka panjang dan merugikan baik dari sisi ekonomi maupun ketahanan pangan berkelanjutan. Oleh sebab itu, permasalahan penyakit daun jagung harus segera diatasi dengan pendekatan teknologi yang lebih modern, akurat, dan efisien.

Metode deteksi penyakit secara manual yang masih mengandalkan pengamatan visual petani atau penyuluh pertanian seringkali kurang efektif. Tingkat subjektivitas pengamat, perbedaan pengalaman, serta keterbatasan jumlah tenaga ahli menyebabkan proses identifikasi penyakit berjalan lambat dan tidak konsisten. Beberapa penelitian terdahulu telah mencoba menggunakan metode klasifikasi citra berbasis teknik klasik seperti *color thresholding* dan *edge detection*. Namun, akurasi metode ini relatif rendah dan tidak mampu mengenali pola citra yang kompleks, sehingga dianggap belum memadai untuk diterapkan di lapangan dalam skala luas. Seiring perkembangan teknologi, kecerdasan buatan khususnya *deep learning* menawarkan pendekatan baru yang lebih akurat dan adaptif. CNN menjadi salah satu arsitektur *deep learning* yang paling banyak digunakan karena kemampuannya mengekstraksi fitur visual secara otomatis dan efisien. Salah satu pengembangan penting dari CNN adalah arsitektur Xception yang diperkenalkan oleh Chollet (2017), dengan menggunakan *depthwise separable convolution* untuk meningkatkan efisiensi dan akurasi dibanding arsitektur CNN konvensional.

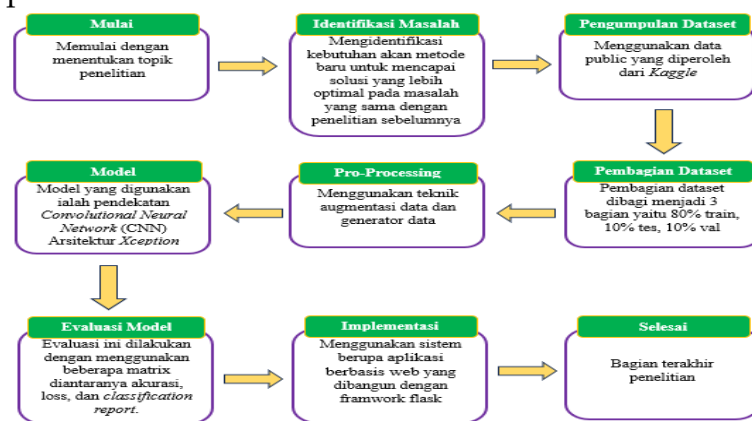
Penelitian ini hadir untuk menjawab kebutuhan sistem klasifikasi penyakit daun jagung berbasis CNN dengan arsitektur Xception, serta

mengimplementasikannya dalam aplikasi web berbasis Flask agar dapat digunakan langsung oleh petani dan penyuluh pertanian di lapangan. Dengan demikian, penelitian ini tidak hanya berkontribusi pada peningkatan akurasi deteksi penyakit daun jagung, tetapi juga menghadirkan solusi praktis dalam bentuk aplikasi web yang dapat membantu petani melakukan deteksi dini secara cepat, efisien, dan mudah diakses. Hal ini diharapkan mampu mendukung program ketahanan pangan nasional sekaligus menjadi inovasi dalam bidang pertanian digital berbasis kecerdasan buatan.

METODE PENELITIAN

Penelitian ini menggunakan metode kuantitatif karena berfokus pada pengumpulan data numerik dengan eksperimental. Fokus utama penelitian ini adalah pada penerapan *deep learning*, khususnya model Convolutional Neural Network (CNN) menggunakan arsitektur *Xception*, untuk mengklasifikasikan gambar daun jagung ke dalam beberapa jenis penyakit.

Proses penelitian dilakukan secara bertahap, dimulai dari merumuskan masalah, mengumpulkan dan mengolah data, merancang serta melatih model, hingga melakukan evaluasi terhadap kinerja model. Tahapan penelitian ditunjukkan pada Gambar 1.



Gambar 1. Diagram Alur Penelitian

Identifikasi Masalah

Identifikasi masalah dilakukan melalui studi literatur dan pemanfaatan dataset penyakit daun jagung yang bersumber dari Kaggle. Dari analisis diketahui bahwa metode identifikasi penyakit masih banyak bergantung pada pengamatan visual yang bersifat subjektif dan kurang akurat. Hal ini mengakibatkan proses klasifikasi penyakit seperti *Gray Leaf Spot*, *Common Rust*, dan *Blight* menjadi lambat dan sering kali tidak tepat. Berdasarkan temuan tersebut, dibutuhkan sistem klasifikasi otomatis berbasis CNN dengan arsitektur *Xception* yang mampu mengolah citra daun jagung untuk menghasilkan prediksi penyakit secara cepat dan akurat. Sistem ini selanjutnya diimplementasikan dalam bentuk aplikasi berbasis web agar dapat digunakan secara lebih praktis oleh pengguna, terutama petani.

Pengumpulan Data

Penelitian ini menggunakan data public yang sudah digunakan penelitian-penelitian sebelumnya yang diperoleh dari Kaggle (<https://www.kaggle.com/datasets>) *Corn or Maize Leaf Disease* dengan jumlah gambar total 4.188 daun jagung

healthy (1.162), daun jagung blight (1.146), daun jagung Common_Rust(1.306), dan daun Jagung gray_leaf_spot (574) ini dilakukan untuk menghemat waktu penelitian karena tidak perlu melakukan pelabelan secara manual dan melibatkan pakar. Di lain sisi ini dapat memvalidasi hasil dari penelitian yang dilakukan.

Model Pengembangan/Perancangan

Penelitian ini menerapkan pendekatan *deep Learning* yang akan diimplementasikan pada *Framework* berupa *Flask*. Model yang dikembangkan berbasis *Convolutional Neural Network* (CNN) dengan arsitektur *Xception* yang termasuk dalam teknik *deep learning*. Pendekatan ini memungkinkan pemanfaatan fitur penting dari citra daun jagung, seperti pola tekstur dan bentuk, sehingga meningkatkan akurasi dalam proses klasifikasi.

Pengembangan model dilakukan melalui beberapa tahapan. Tahap awal dimulai dengan pengumpulan dataset dari sumber publik seperti Kaggle, diikuti oleh proses *pre-processing*, termasuk augmentasi data untuk meningkatkan kinerja model. Selanjutnya, model dilatih menggunakan arsitektur CNN dengan *Xception* dan dievaluasi menggunakan *confusion matrix*, *accuracy*, *presion*, *recall F-1 Score*, untuk memastikan performa klasifikasi yang optimal.

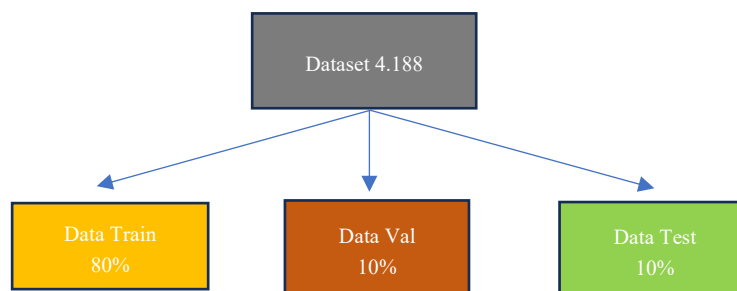
Model yang telah dilatih kemudian diimplementasikan dalam sistem berbasis web menggunakan *framework* seperti *Flask*. Implementasi ini bertujuan untuk memberikan antarmuka yang memudahkan pengguna dalam mengunggah gambar dan memperoleh hasil klasifikasi secara otomatis. Dengan pendekatan ini, penelitian diharapkan dapat berkontribusi dalam pengembangan solusi berbasis *computer vision* untuk mendukung deteksi dini penyakit daun jagung, bagi petani.

Pre-processing Data

Tahapan *pre-processing* data bertujuan untuk mempersiapkan citra daun jagung agar dapat diproses oleh model secara optimal. Data yang bersih, terstruktur, dan memiliki format seragam sangat penting untuk memastikan proses pelatihan berjalan efisien dan akurat. Dalam penelitian ini, *pre-processing* dilakukan dengan menggunakan teknik augmentasi data dan pemanfaatan objek *ImageDataGenerator* dari library Keras yang berfungsi sebagai generator data otomatis.

Pembagian Data

Tahap ini yaitu data yang telah dipersiapkan dibagi menjadi set pelatihan, pengujian, dan validasi. Langkah ini memastikan bahwa model dapat dilatih dengan efektif dan kinerjanya dapat dievaluasi dengan akurat, seperti digambar 3.2 dibawah ini.



Gambar 2 Pembagian Dataset

Evaluasi Model

Menggunakan akurasi, *precision*, *recall*, *F1-score*, dan *confusion matrix*, untuk mendapatkan pemahaman yang lebih mendalam tentang kesalahan prediksi yang dilakukan model. Setiap elemen dalam *confusion matrix* menunjukkan berapa banyak prediksi yang benar atau salah untuk setiap kelas, sehingga memberikan wawasan yang jelas tentang di mana model sering salah dalam mengklasifikasikan gambar.

Implementasi Model

Setelah semua langkah berhasil diselesaikan, tahap terakhir adalah Model akan diimplementasi. Dalam penelitian ini, model akan diimplementasikan menggunakan *framework Flask*. Yang dirancang berupa aplikasi web sederhana untuk menampilkan polaritas penyakit daun jagung, implementasi ini dilakukan untuk menampilkan hasil prediksi jenis daun jagung yang terkena penyakit seperti Blight Daun, Common_Rust Daun Jagung, Gray_leaf_spot Daun Abu-abu, dan daun healthy dan menampilkan hasil akurasi.

HASIL DAN PEMBAHASAN

Identifikasi Masalah

Salah satu masalah utama dalam budidaya jagung adalah sulitnya melakukan klasifikasi penyakit daun secara cepat, akurat, dan konsisten. Selama ini petani masih mengandalkan identifikasi manual berdasarkan pengalaman lapangan atau panduan penyuluh. Metode ini memiliki beberapa keterbatasan, antara lain tingkat akurasi yang rendah, dipengaruhi subjektivitas pengamat, serta memerlukan waktu yang tidak sedikit terutama pada lahan dengan skala besar. Kesalahan dalam mengenali jenis penyakit dapat berakibat fatal karena perlakuan yang diberikan bisa keliru dan justru memperburuk kondisi tanaman.

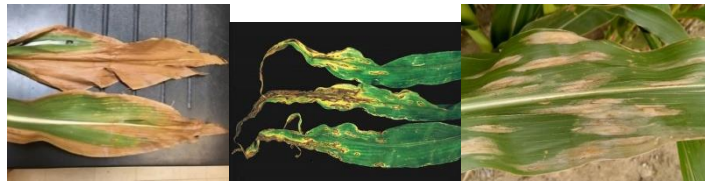
Beberapa penelitian terdahulu mencoba mengatasi masalah ini dengan metode pengolahan citra konvensional, seperti segmentasi atau thresholding. Namun, pendekatan tersebut hanya efektif untuk pola sederhana dan kurang mampu menangani keragaman gejala penyakit yang kompleks. Oleh karena itu, dibutuhkan teknologi yang mampu mengenali pola visual secara lebih mendalam dan adaptif.

Dalam penelitian ini, penulis mengusulkan solusi berupa penerapan Convolutional Neural Network (CNN) dengan arsitektur Xception untuk melakukan klasifikasi penyakit daun jagung berbasis citra digital. Model ini dipilih karena memiliki kemampuan mengekstraksi fitur citra secara efisien melalui mekanisme depthwise separable convolution. Agar hasil penelitian dapat langsung dimanfaatkan di lapangan, model diintegrasikan ke dalam aplikasi web berbasis Flask, sehingga pengguna dapat mengunggah foto daun jagung dan memperoleh hasil prediksi penyakit secara otomatis dan real-time.

Pengumpulan Dataset

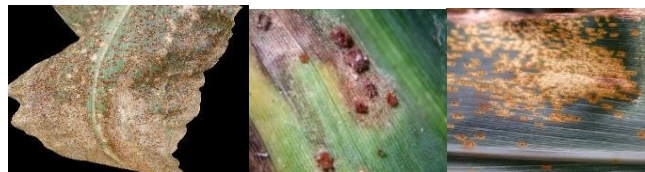
Dalam penelitian ini, data yang digunakan terdiri dari 4.188 gambar daun jagung terbagi menjadi kategori gray_leaf_spot daun jagung sebanyak 574, busuk atau blight daun jagung sebanyak 1.146, common_Rust daun jagung sebanyak 1.306 dan healthy daun jagung sebanyak 1.162, yang diambil dari platform kaggle

dalam format JPEG, JPG dan PNG. Gambar (A) adalah gambar dari salah satu dataset gray_leaf_spot daun jagung. Sedangkan Gambar (B) adalah gambar dari salah satu dataset Blight atau busuk daun jagung, dan Gambar (C) adalah gambar dari salah satu dataset Common_Rust daun jagung, Gambar (D) adalah gambar dari salah satu dataset daun jagung healthy seperti yang terdapat pada Gambar 3.



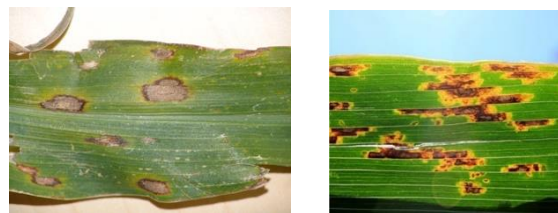
(A)

Gambar 3. Salah satu Dataset Gray_leaf_spot Daun Jagung



(B)

Gambar 4. Salah satu Dataset Blight Daun Jagung



(C)

Gambar 5. Salah satu Dataset Common_Rust Daun Jagung



Pembagian Dataset

Pada tahapan ini, dataset gambar daun jagung akan dibagi menjadi tiga bagian, yaitu data latih 80% yang berisi data sebanyak 3.348 gambar data validasi 10% sebanyak 419 gambar data uji 10% sebanyak 421 gambar. Pembagian ini dilakukan secara otomatis menggunakan library split folders. Metode ini juga akan menjaga proporsi distribusi dari setiap kelas dalam masing-masing subset agar tetap seimbang. seperti yang terdapat pada Segmen 4.1 pembagian dataset yang proposional penting agar supaya bisa memastikan proses pelatihan model dilakukan secara efektif, serta evaluasi hasil dapat dilakukan secara objektif

karena menggunakan data yang berbeda dari data pelatihan.

```
1 | dataset_dir= "/content/drive/MyDrive/daunjagung_train"
2 | output_dir= "/content/drive/MyDrive/daunjagungTA_spilit"
3 |
4 | # Buat folder untuk data terpisah
5 | for folder in ['train', 'validation', 'test']:
6 |     for cls in os.listdir(dataset_dir):
7 |         os.makedirs(os.path.join(output_dir, folder, cls), exist_ok=True)
8 | # Bagi data: 80% train, 10% validation, 10% test
9 | for cls in os.listdir(dataset_dir):
10 |     class_path = os.path.join(dataset_dir, cls)
11 |     images = os.listdir(class_path)
12 |
13 |     # Split 80% train, 20% temporary (untuk validation dan test)
14 |     train_images, temp_images = train_test_split(images, test_size=0.2, random_state=42)
15 |     # Split 50-50 dari 20% untuk validation dan test (10% masing-masing)
16 |     val_images, test_images = train_test_split(temp_images, test_size=0.5, random_state=42)
17 |
18 |     # Salin gambar ke folder masing-masing
19 |     for img, category in zip([train_images, val_images, test_images], ['train', 'validation', 'test']):
20 |         for image in img:
21 |             shutil.copy(os.path.join(class_path, image),
22 |                         os.path.join(output_dir, category, cls, image))
23 |
24 | print("Dataset split completed!")
```

Gambar 6: Program Dataset

Dengan kode yang ada di atas, dataset yang awalnya berada dalam satu folder utama otomatis akan terbagi ke dalam subfolder train 3.348, test 421, dan validasi 419. Masing-masing dengan proporsi yang telah ditentukan, dalam pembagian ini memastikan bahwa model yang dibangun tidak hanya diukur dari kemampuan menghafal data, namun juga dari kemampuannya dalam melakukan generalisasi terhadap data baru.

Model Arsitektur Xception

Pada penelitian ini, pada tahap melakukan model yang dilakukan ialah menggunakan pendekatan *Convolutional Neural Network* (CNN) Arsitektur *Xception* dilakukan menggunakan *python* dengan *framework TensorFlow* dan *keras*. Model *Xception* yang digunakan juga merupakan versi *pre-trained* tanpa *top layer* (*include_top=False*) agar dapat dimodifikasi sesuai kebutuhan klasifikasi 4 kelas. Setelah memuat base model. Ditambahkan layer tambahan, yaitu *Conv2D*, *MaxPooling2D*, *Dropout*, *Flatten*, dan *Dense* dengan aktivitas *Softmax*. Lalu model dikompilasi menggunakan optimizer Adam *learning rate* 0.0001 dan fungsi loss.

Setelah proses pelatihan selesai, model lalu dievaluasi menggunakan data uji untuk mengukur performa klasifikasi. Evaluasi dilakukan dengan menguji model test set dan membandingkan hasil prediksi terhadap label sebenarnya. Hasil prediksi menggunakan *Confusion Matrix* untuk klasifikasi kelas, selain itu dilakukan perhitungan klasifikasi report yang mencakup seluruh metrik evaluasi seperti *precision*, *recall*, *f1 score*, dan *accuracy*. Evaluasi ini memberikan gambaran menyeluruh terkait kinerja model mengenai setiap kelas yang ada. Nilai akurasi akhir model juga ditampilkan secara terpisah sebagai perbandingan keberhasilan utama dari klasifikasi.

Membangun Model Xception

Tahapan ini menggunakan arsitektur *pre-trained Xception* yang telah dilatih sebelumnya pada dataset *ImageNet*. Arsitektu *Xception* juga merupakan pengembangan dari *inception* Dimana yang mendapatkan *deftwise separable convolution*, sehingga lebih efisien serta juga mampu mengekstraksi fitur secara lebih dalam dan juga kompleks yang terdapat pada segmen 4.5 ini adalah kode base model.

```
1 | # Menggunakan Xception sebagai base model
2 | base_model = tf.keras.applications.Xception(include_top=False, weights="imagenet",
input_tensor=tf.keras.Input(shape=(224, 224, 3)))
3 | base_model.trainable = False
```

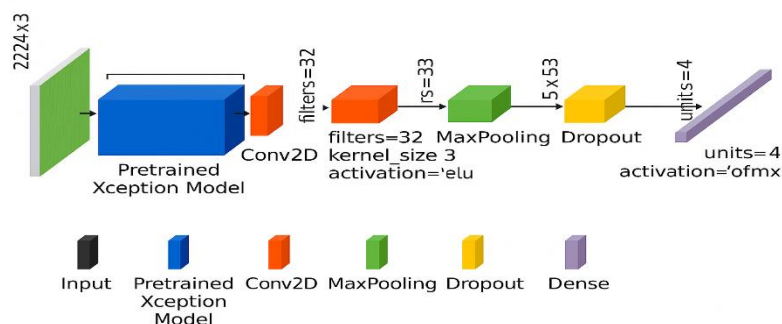
Gambar 7: base model Xception

Gambar 7 Ini menunjukkan bagaimana model *Xception* dibangun. Seluruh lapisan konvolusional dibekukan (*trainable=false*) agar bobot hasil pelatihan pada *imagenet* tetap dipertahankan. Setelah itu lalu ditambahkan lapisan *maxpool2D*, *dropout*, *flatten*, dan *dense* untuk menyesuaikan output dengan jumlah kelas yang diinginkan, yaitu 4 kelas.

```
1 | # Menentukan nama model
2 | model_name = 'Xception'
3 | print("Building model with", base_model)
5 | # Membangun model Sequential dengan base_model dan beberapa layer tambahan
6 | model = tf.keras.Sequential([
7 |     base_model, # Menambahkan base model yang telah di load sebelumnya
8 |     tf.keras.layers.Conv2D(filters=32, padding='same', kernel_size=3, activation='relu',
strides=1),
9 |     tf.keras.layers.MaxPool2D(pool_size=2, strides=2),
10 |     tf.keras.layers.Dropout(rate=0.5),
11 |     tf.keras.layers.Flatten(),
12 |     tf.keras.layers.Dense(4, activation='softmax')
13 | ])
15 | # Mengkompilasi model dengan optimizer Adam, loss function categorical_crossentropy, dan
metric akurasi
16 | model.compile(optimizer=tf.keras.optimizers.Adam(learning_rate=.001),
17 |               loss='categorical_crossentropy', # Menggunakan loss function
categorical_crossentropy
18 |               metrics=['accuracy'])
```

Gambar 8: Membangun Model Xception

Pada gambar 8 yaitu membangun model CCN dengan arsitektur *Xception* model ini dikompilasi menggunakan fungsi *loss categorical_crossentropy*, *optimizer Adam* dengan nilai *learning rate*, serta matrik evaluasi akurasi.



Gambar 9: Arsitektur Xception

Model ini menggunakan arsitektur Xception pretrained sebagai feature extractor untuk mengekstraksi fitur penting dari citra berukuran 224×224 piksel. Layer ini dibekukan agar tidak dilatih ulang, sehingga mempercepat pelatihan dan menghindari overfitting. Fitur hasil ekstraksi dilanjutkan ke Conv2D dengan 32 filter dan aktivasi ReLU untuk mendeteksi pola lanjutan, lalu diperkecil menggunakan MaxPooling2D agar ukuran data lebih efisien. Setelah itu, Dropout dengan rate 0.5 diterapkan untuk mencegah overfitting dengan mematikan sebagian neuron secara acak saat pelatihan. Lalu Flatten, Terakhir fitur yang telah diratakan diklasifikasikan melalui Dense layer berisi 4 neuron output dengan aktivasi Softmax, yang memberikan probabilitas keempat kelas target pada citra daun jagung.

```
1 | # Kompilasi Model
2 | model.compile(
3 |     optimizer=Adam(learning_rate=1e-4),
4 |     loss='categorical_crossentropy',
5 |     metrics=['accuracy']
6 | )
```

Gambar 10: Kompilasi

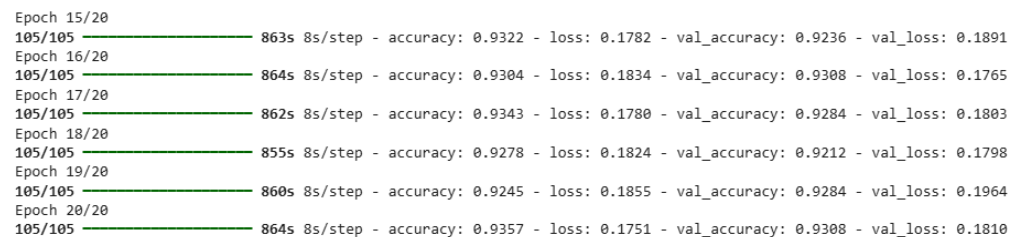
Untuk menghindari *overfitting* dan mengoptimalkan dari proses pelatihan, maka perlu melakukan tiga jenis *Callback* diantara, *Earlystopping* untuk menghentikan pelatihan saat validasi loss tidak mengalami penurunan epoch berturut-turut, *ReduceROnPlateau* untuk menurunkan *learning rate* jika ada perbaikan pada validasi loss selama 20 epoch, dan *ModelChekPoint* menyimpan model bobot terbaik berdasarkan nilai akurasi validasi tertinggi seperti yang terdapat pada segmen 8 ini.

```
1 | # Callbacks
2 | callbacks = [
3 |     EarlyStopping(
4 |         monitor='val_loss',
5 |         patience=5,
6 |         verbose=1,
7 |         restore_best_weights=True
8 |     ),
9 |     ReduceLROnPlateau(
10 |         monitor='val_loss',
11 |         factor=0.1,
12 |         patience=3,
13 |         verbose=1,
14 |         min_lr=1e-7
15 |     ),
16 |     ModelCheckpoint(
17 |         'best_xception_model.h5',
18 |         monitor='val_accuracy',
19 |         save_best_only=True,
20 |         mode='max',
21 |         verbose=1
22 |     )
23 | ]
```

Gambar 11: Callback

Setelah itu dilatih dengan menggunakan data train dan validasi dengan menggunakan method fit() Dimana yang sudah pernah dilatih pada segmen program 9.

```
1 | history = model.fit(
2 |     train_generator,
3 |     epochs=20,
4 |     validation_data=validation_generator
5 | )
```



Epoch	Progress	Time	accuracy	loss	val_accuracy	val_loss
Epoch 15/20	105/105	863s	0.9322	0.1782	0.9236	0.1891
Epoch 16/20	105/105	864s	0.9304	0.1834	0.9308	0.1765
Epoch 17/20	105/105	862s	0.9343	0.1780	0.9284	0.1803
Epoch 18/20	105/105	855s	0.9278	0.1824	0.9212	0.1798
Epoch 19/20	105/105	860s	0.9245	0.1855	0.9284	0.1964
Epoch 20/20	105/105	864s	0.9357	0.1751	0.9308	0.1810

Gambar 12: Melatih model

Hasil pelatihan menunjukkan model CNN-Xception mencapai akurasi 92,40% pada data uji dengan loss 0,1475. Performa terbaik terdapat pada kelas *healthy* dengan presisi 98% dan recall 96%, sedangkan performa terendah pada kelas *gray leaf spot* dengan recall sekitar 68,97%.

Perbandingan dua skenario pelatihan menunjukkan bahwa augmentasi data meningkatkan performa model: Tanpa augmentasi: akurasi 89,05% Dengan augmentasi: akurasi 92,40% Confusion matrix mengungkap bahwa sebagian besar kesalahan klasifikasi terjadi pada kelas *gray leaf spot* yang sering tertukar dengan *blight* maupun *common rust*. Hal ini dipengaruhi oleh distribusi dataset yang tidak seimbang, di mana kelas *gray leaf spot* memiliki jumlah data paling sedikit. Model kemudian diimplementasikan pada aplikasi web berbasis Flask. Aplikasi ini menyediakan fitur unggah citra daun jagung dan menampilkan hasil klasifikasi penyakit secara otomatis dan real-time. Uji coba aplikasi menunjukkan antarmuka sederhana dan dapat digunakan dengan mudah oleh pengguna non-teknis.

Secara keseluruhan, penelitian ini membuktikan bahwa kombinasi CNN, arsitektur Xception, dan teknik augmentasi mampu menghasilkan performa klasifikasi yang baik. Namun, keterbatasan utama masih terdapat pada jumlah data untuk kelas minoritas. Oleh karena itu, penelitian lanjutan disarankan untuk menambah variasi dataset dan melakukan optimasi hiperparameter agar performa meningkat lebih merata di semua kelas.

KESIMPULAN

Penelitian ini bertujuan mengembangkan sistem klasifikasi penyakit daun jagung menggunakan metode *Convolutional Neural Network* (CNN) dengan arsitektur *Xception*, yang diimplementasikan ke dalam aplikasi web berbasis *Flask*. Dataset yang digunakan berjumlah 4.188 gambar, terbagi dalam 4 kelas: *healthy*, *blight*, *common_Rust*, dan *gray_leaf_spot* daun. Data dibagi menjadi training (80%), validasi (10%), dan testing (10%), lalu diproses melalui tahapan *rescaling*, *resizing*, *batching*, dan augmentasi data. Model dibangun menggunakan *Xception pretrained* yang dilengkapi layer tambahan seperti *Conv2D*, *MaxPooling2D*, *Dropout*, dan *Dense*, kemudian model dilatih menggunakan optimasi Adam dengan nilai

learning rate sebesar 0.0001, serta menerapkan fungsi kerugian *categorical crossentropy* untuk menghitung error dalam klasifikasi. Pelatihan dilakukan di *Google Colab* dengan penyimpanan *Google Drive*. Hasil pengujian pada 1.004 gambar uji menunjukkan model mencapai akurasi 92,40% dengan nilai loss 0,1475. Evaluasi lebih lanjut menunjukkan performa terbaik pada kelas healthy (*f1-score* 0.99) dan terendah pada common_Rust daun (*f1-score* 0.83). Model kemudian diimplementasikan dalam web Flask, memungkinkan pengguna mengunggah gambar dan memperoleh hasil klasifikasi beserta informasi penyakit dan saran penanganan. Antarmuka dirancang agar mudah digunakan oleh petani maupun masyarakat umum. Secara keseluruhan, sistem ini terbukti efektif untuk klasifikasi penyakit daun jagung secara otomatis, akurat, dan aplikatif, serta memberikan kontribusi pada pengembangan teknologi deep learning di bidang pertanian.

REFERENSI

- Akbar, J., Daniawan, B., Hariyanto, S., Renaldi, D., Lasut, D., Safitri, R. D., Fenriana, I., & Suyitno, A. (n.d.). *Transformasi Digital Pembelajaran: Integrasi Data Mining Google Colab untuk Keterampilan Teknologi Siswa SMK*.
- Azizah, Q. N. (2023). Klasifikasi Penyakit Daun Jagung Menggunakan Metode Convolutional Neural Network AlexNet. *sudok Jurnal Teknik Informatika*, 2(1), 28–33. <https://doi.org/10.56211/sudok.v2i1.227>
- Candana, E. W. H., Gunadi, I. G. A., & Divayana, D. G. H. (2021). *Jurnal Ilmu Komputer Indonesia (JIK)*, Volume 6, No: 2, November 2021 p-ISSN : 2615-2703 (Print) dan e-ISSN: 2615-2711 (Online). 6.
- Febby Wilyani, Qonaah Nuryan Arif, & Fitri Aslimar. (2024). Pengenalan Dasar Pemrograman Python Dengan Google Colaboratory. *Jurnal Pelayanan dan Pengabdian Masyarakat Indonesia*, 3(1), 08–14. <https://doi.org/10.55606/jppmi.v3i1.1087>
- Gaho, R. L., Ali, I. T., & Prakasa, E. (2024). Klasifikasi Kualitas Permukaan Jalan Raya Menggunakan Metode Cnn Berbasis Arsitektur Xception. *Inovtek Polbeng - Seri Informatika*, 9(1). <https://doi.org/10.35314/isi.v9i1.4213>
- Hermawan, A. L. (2023). Klasifikasi Penyakit Daun Jagung Menggunakan Lightweight Convolutional Neural Network. *JIIFKOM (Jurnal Ilmiah Informatika dan Komputer)*, 2(2), 1–7. <https://doi.org/10.51901/jiifkom.v2i2.347>
- Kusumastuti, R., Dwi Putra, T., & Zulfahmi Yudam, Z. (2024). Klasifikasi Citra Penyakit Daun Jagung Menggunakan Algoritma Cnn Efficientnet. *Multitek Indonesia*, 17(2), 143–153. <https://doi.org/10.24269/mtkind.v17i2.10085>
- Ma, Z., Wang, Y., Zhang, T., Wang, H., Jia, Y., Gao, R., Su, Z., 1. Institute of Electrical and Information, Northeast Agricultural University, Harbin 150030, China, 2. Agricultural Products and Veterinary Drug Feed Technical Identification Station, Department of Agriculture and Rural Affairs of Heilongjiang Province, Harbin 150090, China, & 3. Key Laboratory of Northeast Smart Agricultural Technology, Ministry of Agriculture and Rural Affairs, Heilongjiang Province, Harbin 150030, China. (2022).
- Maize leaf disease identification using deep transfer convolutional neural

- networks. *International Journal of Agricultural and Biological Engineering*, 15(5), 187–195. <https://doi.org/10.25165/j.ijabe.20221505.6658>
- Putra, I. P., Rusbandi, R., & Alamsyah, D. (2022). Klasifikasi Penyakit Daun Jagung Menggunakan Metode Convolutional Neural Network. *Jurnal Algoritme*, 2(2), 102–112. <https://doi.org/10.35957/algoritme.v2i2.2360>
- Raihan Maulana, Raisya Dwi Zahra Putri, Sindy Fitriani Margareth Sihaloho, & Sri Mulyana. (2023). Implementasi Algoritma Convolutional Neural Network Dalam Mengklasifikasi Jenis Burung. *Journal of Creative Student Research*, 1(6), 221–231. <https://doi.org/10.55606/jcsrpolitama.v1i6.2966>
- Rajeena P. P, F., S. U, A., Moustafa, M. A., & Ali, M. A. S. (2023). Detecting Plant Disease in Corn Leaf Using EfficientNet Architecture – An Analytical Approach. *Electronics (Switzerland)*, 12(8). <https://doi.org/10.3390/electronics12081938>
- Rizki, F., Kharisma Putra, M. P., Assuja, M. A., & Ariany, F. (2023). Implementasi Deep Learning Lenet Dengan Augmentasi Data Pada Identifikasi Anggrek. *Jurnal Informatika dan Rekayasa Perangkat Lunak*, 4(3), 357–366. <https://doi.org/10.33365/jatika.v4i3.3652>
- Salehin, I., & Kang, D.-K. (2023). A Review on Dropout Regularization Approaches for Deep Neural Networks within the Scholarly Domain. *Electronics*, 12(14), 3106. <https://doi.org/10.3390/electronics12143106>