

BAB III

METODE PENELITIAN

3.1 Objek Penelitian

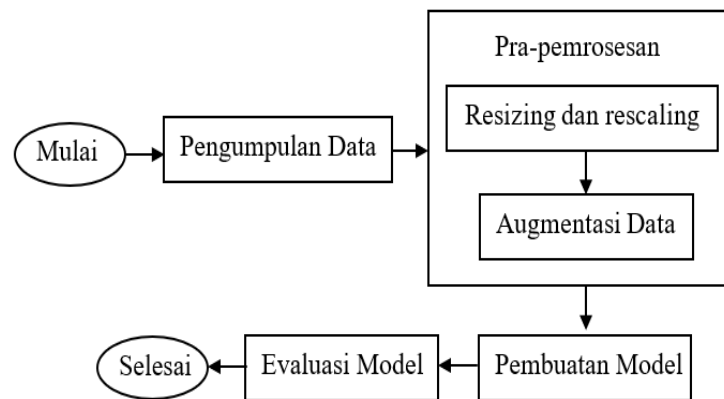
Objek penelitian yang dipakai untuk klasifikasi penyakit daun kentang terdiri dari 1.500 citra yang terbagi menjadi tiga folder yaitu *training*, *validasi* dan *testing*. Dataset ini terbagi menjadi tiga kelas yaitu *early blight*, *healthy*, dan *late blight*. Penelitian dilakukan di Laboratorium Program Studi Teknik Informatika, Universitas Buana Perjuangan Karawang yang dimulai dari Mei 2024. Penelitian ini bertujuan untuk mengidentifikasi jenis penyakit daun kentang dengan algoritma SVM dan CNN dengan arsitektur VGG16. Rincian dataset ini ditunjukkan pada Tabel 3.1.

Tabel 3.1 Contoh Dataset Daun Kentang

No	Jenis Daun	Contoh Citra
1	<i>Late blight</i>	
2	<i>Early blight</i>	
3	<i>Healthy</i>	

3.2 Prosedur Penelitian

Penelitian ini menggunakan algoritma CNN dan SVM dengan arsitektur VGG16, yang melalui beberapa tahapan yang dapat diamati pada Gambar 3.1.



Gambar 3.1 Tahapan Penelitian

3.2.1. Pengumpulan Data

Proses pengumpulan dataset untuk penelitian ini dilakukan pada tahap ini. Dataset yang digunakan diperoleh dari situs publik *Kaggle* disusun oleh Abbas AtaieMontazer. Dataset berisi 1.500 gambar daun kentang. Dataset dibagi menjadi tiga bagian meliputi data *training*, validasi, dan *testing* dengan proporsi seimbang pada setiap kelas. Data *training* berisi total 900 gambar, dengan 300 gambar untuk setiap kelas, sedangkan data validasi dan *testing* masing-masing berisi total 300 gambar, atau 100 gambar untuk setiap kelas.

3.2.2. Pra-pemrosesan

Tahap pra-pemrosesan diterapkan untuk mempersiapkan citra daun kentang untuk meningkatkan citra sebagai langkah preproses sebelum analisis. Langkah pra-pemrosesan yang dilakukan pada penelitian ini adalah *resizing*, *rescaling*, dan *augmentasi data*. *Resizing* atau mengubah ukuran gambar yang pada penelitian ini menjadi 128×128 piksel. Ukuran ini dipilih untuk mempercepat proses pelatihan serta mengurangi beban komputasi tanpa menghilangkan fitur penting dari citra. *Rescaling* untuk mengubah nilai piksel gambar dari rentang asli $[0, 255]$ menjadi $[0, 1]$ dengan membagi nilai piksel dengan 255. Selanjutnya, dilakukan *augmentasi* yang berfungsi untuk memperkaya data dan memungkinkan model untuk dilatih dari berbagai sudut pandang data yang berbeda (Kohsasih et al., 2021). *Augmentasi* data pada penelitian ini menggunakan *random flip* secara horizontal dan vertikal. Selain itu melakukan *random rotation* hingga 20% dari total rotasi.

3.2.3. Pembuatan Model

Pada tahap *modelling*, klasifikasi gambar daun kentang dilakukan menggunakan dua algoritma yaitu CNN dan SVM dengan VGG16. Pada model CNN, digunakan arsitektur VGG16 yang sudah dipra-latih menggunakan dataset *ImageNet* sebagai ekstraktor fitur utama. Citra daun kentang diproses melalui model VGG16 yang terdiri dari beberapa lapisan konvolusi dan *pooling* untuk mengekstraksi fitur-fitur penting. Setelah fitur diekstraksi, lapisan *fully connected* pada VGG16 digunakan untuk melakukan klasifikasi secara langsung pada gambar daun tersebut. Pseudocode pemodelan algoritma SVM dapat dilihat pada Tabel 3.2.

Tabel 3.2 Pseudocode Algoritma *Convolutional Neural Network*

Algoritma CNN menggunakan VGG16

Input: data *training*, *validation*, dan *testing*

Output: Pelatihan algoritma CNN

- 1: Muat model VGG16 yang sudah dilatih dengan bobot dari dataset *ImageNet*.
 - 2: Nonaktifkan pelatihan ulang pada lapisan awal.
 - 3: Terapkan *preprocessing* (*resize* dan *rescale*) dan augmentasi data pada dataset
 - 4: Ganti lapisan output asli dengan lapisan *fully connected* baru, sesuai dengan jumlah kelas dataset.
 - 5: Tambahkan lapisan *dense* dan lapisan output untuk klasifikasi.
 - 6: Gunakan *optimizer Adam* dan fungsi *loss categorical crossentropy*.
 - 7: Latih model pada data latih, gunakan data validasi untuk memonitor performa selama pelatihan.
 - 8: Simpan model CNN terbaik setelah proses pelatihan, dan gunakan model ini untuk prediksi pada data validasi dan uji.
-

Selanjutnya pada model SVM, prosesnya dimulai dengan ekstraksi fitur dari gambar yang sama menggunakan VGG16. Setelah itu, fitur-fitur yang diperoleh dari VGG16 diratakan dan digunakan sebagai input untuk model SVM dengan *kernel linear*. Kedua model ini dilatih dengan data *training* dan

divalidasi dengan data validasi, dan kemudian diuji pada data *testing* untuk mengukur akurasi. Berikut pseudocode pemodelan algoritma SVM.

Tabel 3.3 Pseudocode Algoritma *Support Vector Machine*

Algoritma SVM menggunakan VGG16

Input: data *training*, *validation*, dan *testing*

Output: Pelatihan algoritma SVM

- 1: Muat model VGG16 yang sudah dilatih dengan bobot dari dataset *ImageNet*.
 - 2: Nonaktifkan pelatihan ulang pada model VGG16
 - 3: Ambil keluaran fitur dari lapisan terakhir sebelum *fully connected*.
 - 4: Terapkan *preprocessing* (*resize* dan *rescale*) dan augmentasi data pada dataset
 - 5: Masukkan dataset yang sudah diproses ke dalam model VGG16 untuk menghasilkan fitur.
 - 6: Ratakan (*flatten*) fitur menjadi vektor satu dimensi.
 - 7: Siapkan data latih, validasi, dan uji hasil ekstraksi fitur.
 - 8: Pilih *kernel linear* untuk SVM
 - 9: Gunakan SVC (*Support Vector Classifier*) untuk klasifikasi.
 - 10: Latih model SVM menggunakan data latih hasil ekstraksi fitur dari model VGG16. Gunakan data validasi untuk memonitor performa selama pelatihan.
 - 11: Simpan model SVM terbaik setelah proses pelatihan, dan gunakan model ini untuk prediksi pada data validasi dan uji.
-

3.2.4. Evaluasi Model

Evaluasi model dilakukan dengan memanfaatkan *confusion matrix* guna mengukur seberapa baik model dapat memprediksi kelas dari gambar-gambar daun kentang pada data *testing*. Akurasi merupakan proporsi prediksi yang sesuai (baik positif maupun negatif) dari keseluruhan data. *Precision* adalah proporsi antara prediksi positif yang benar terhadap semua hasil positif yang telah diprediksi. *Recall* merupakan rasio antara prediksi positif yang

benar terhadap semua data aktual positif. *F1-score* merupakan nilai rata-rata harmonis antara *precision* dan *recall*.

$$Accuracy = \frac{(TP + TN)}{(TP + FP + FN + TN)} \times 100\% \quad (1)$$

$$Precision = \frac{TP}{(TP + FP)} \quad (2)$$

$$Recall = \frac{TP}{(TP + FN)} \quad (3)$$

$$F1-score = 2 \frac{(Precision \times Recall)}{(Precision + Recall)} \quad (4)$$

dimana

- TP (*True Positive*) : Data dengan kelas positif berhasil diprediksi dengan benar sebagai kelas positif.
- FN (*False Negative*) : Data dengan kelas positif keliru diprediksi sebagai kelas negatif
- TN (*True Negative*) : Data dengan kelas negatif berhasil diprediksi dengan benar sebagai kelas negatif
- FP (*False Positive*) : Data dengan kelas negatif keliru diprediksi sebagai kelas positif

KARAWANG