

BAB III

METODE PENELITIAN

Dalam penelitian ini, arsitektur VGG-19 digunakan untuk mengidentifikasi penyakit pada daun tebu. Penyakit tersebut diklasifikasikan berdasarkan nama penyakit Mosaic, Karat Daun, Busuk Merah.

3.1 Metode Pengumpulan Data

Pengumpulan data dilakukan guna memperoleh informasi yang diperlukan untuk mencapai tujuan penelitian. Tujuan penelitian ini dirumuskan dalam bentuk hipotesis atau jawaban sementara atas pertanyaan penelitian. Proses pengumpulan data dilaksanakan melalui tahapan-tahapan berikut:

1. Observasi

Untuk mengumpulkan data dalam penelitian ini, penulis melakukan observasi guna memperoleh informasi mengenai penyakit daun tebu dengan melakukan peninjauan dan pengamatan langsung di perkebunan tebu milik Bapak Imam yang berlokasi di Jl. Untung Suropati.

2. Studi Literatur

Metode ini digunakan untuk mencari sumber data yang diperlukan dalam penelitian, yang umumnya diperoleh melalui membaca dan mengutip referensi, Buku serta jurnal penelitian sebelumnya yang mendukung dan berkaitan dengan penelitian ini.

3. Wawancara

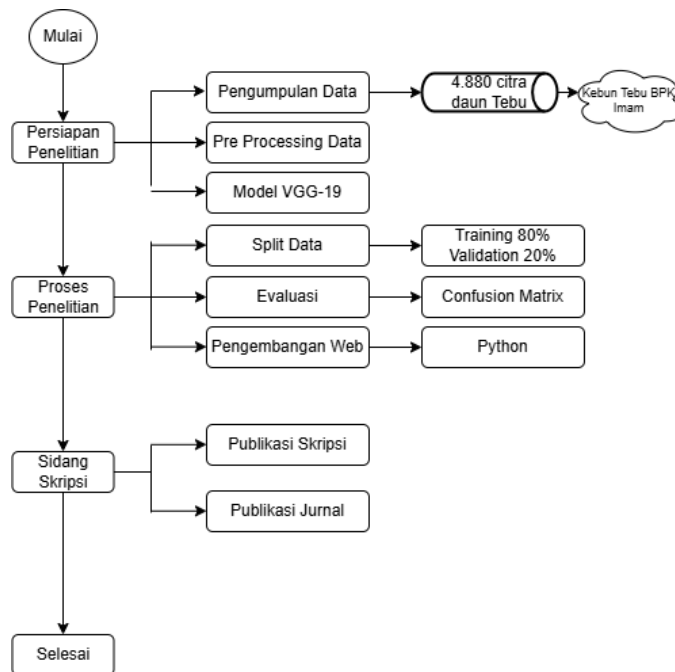
Wawancara adalah metode pengumpulan data di mana informasi diperoleh langsung dari narasumber yang berhubungan dengan topik penelitian. Dalam penelitian ini, peneliti mengajukan beberapa pertanyaan kepada pemilik Kebun Tebu yang ada di Jl. Untung Suropati yaitu Bapak Imam

3.2 Waktu Penelitian

Waktu penelitian ini dimulai pada bulan November sampai dengan bulan Januari 2024

3.3 Alur Penelitian

Konsep atau gambaran dari penelitian yang akan dilakukan disebut alur penelitian, dan alur penelitian tersebut digambarkan sebagai berikut dapat dilihat pada gambar 3.1 :



Gambar 3. 1 Diagram Alur Penelitian

3.4 Tahapan Penelitian

3.4.1 Pengumpulan Data

Pengumpulan data adalah langkah awal yang sangat penting dalam proyek pengembangan model pembelajaran mesin. Dalam konteks identifikasi penyakit pada daun tebu, data yang diperlukan biasanya berupa gambar daun tebu yang terinfeksi penyakit. Data penelitian ini diperoleh melalui pengumpulan data yang dilakukan di kebun tebu milik Bapak Imam di Jl. Untung Suropati, yang dijadikan lokasi utama penelitian ini.

3.4.2 Preprocessing Data

Setelah data dikumpulkan, dataset dibagi menjadi data pelatihan dan validasi. langkah selanjutnya adalah *preprocessing* data. Proses ini bertujuan untuk mempersiapkan data agar siap digunakan dalam model pembelajaran

mesin. Studi ini menggunakan skenario dataset 80% : 20%. Perbandingan data ini didasarkan pada pareto principle yang biasa digunakan dalam data mining, yang menyatakan bahwa 80% kejadian berasal dari 20% sisanya. Tabel 3.1 menggambarkan bagaimana dataset yang digunakan dalam penelitian ini dikelompokkan:

Tabel 3. 1 Pembagian data latih dan data diuji

No	Nama	Jenis	Jumlah
1	Mosaic	Training	1010
2	Karat Daun	Training	1052
3	Busuk Merah	Training	1194
4	Mosaic	Validation	252
5	Karat Daun	Validation	262
6	Busuk Merah	Validation	298
7	Mosaic	Testing	252
9	Karat Daun	Testing	262
8	Busuk Merah	Testing	298

3.4.3 Pengembangan Model

Pengembangan model adalah tahap di mana kita membangun dan melatih model pembelajaran mesin untuk melakukan klasifikasi berdasarkan data yang telah diproses. Dalam konteks ini, kita akan menggunakan arsitektur VGG-19, yang merupakan salah satu model jaringan saraf konvolusional (CNN) yang terkenal dan efektif untuk pengenalan gambar.

3.4.4 Pengembangan Sistem Web

Setelah model dilatih dan dievaluasi, langkah terakhir adalah mengembangkan sistem web yang memungkinkan pengguna untuk mengunggah gambar daun tebu dan mendapatkan prediksi tentang kesehatan daun tersebut.

3.5 Metode Pengembangan Sistem

Metode pengembangan sistem dalam rekayasa sistem dan perangkat lunak merupakan proses pembuatan serta modifikasi sistem, yang melibatkan model dan metodologi yang digunakan dalam pengembangan perangkat lunak. Salah satu metodologi yang diterapkan dalam penelitian ini adalah Waterfall, yang terdiri dari beberapa tahapan, yaitu: Analisis Kebutuhan, Desain Sistem, Penulisan kode program, Pengujian (Testing), dan Pemeliharaan.

3.5.1 Analisis Kebutuhan

Pada tahap ini, dilakukan identifikasi kebutuhan sistem baik dari sisi fungsional maupun non-fungsional. Kebutuhan sistem dirancang untuk mendeteksi penyakit daun tebu berdasarkan citra, dengan arsitektur model deep learning VGG-19. Kebutuhan yang dikumpulkan meliputi:

1. Kebutuhan Software

Perangkat lunak yang dibutuhkan untuk membangun dan menjalankan sistem :

- a. Sistem Operasi *Microsoft Windows 11*
- b. Visual Studio Code
- c. Python
- d. TensorFlow 2.5

2. Kebutuhan Hardware

- a. Laptop
- b. RAM 8GB
- c. Intel Core i5-10300H

3. Kebutuhan Data

Dataset citra daun tebu diperoleh dari kebun tebu milik Bapak Imam yang berlokasi di Jl. Untung Suropati. Data dikumpulkan dengan cara memotret daun tebu yang sehat dan yang mengalami gejala penyakit. Setiap citra dikategorikan berdasarkan jenis penyakit oleh ahli atau berdasarkan pengamatan visual yang dikonfirmasi. Dataset kemudian melalui proses preprocessing sebelum digunakan untuk pelatihan model.

3.5.2 Desain Sistem

Tahap desain sistem meliputi desain arsitektur aplikasi, struktur database, dan antarmuka pengguna (UI). Desain sistem dibagi menjadi dua bagian utama:

1. Desain Model Deep Learning

Model VGG-19 digunakan sebagai arsitektur dasar dalam sistem deteksi penyakit daun tebu. VGG-19 merupakan arsitektur deep convolutional neural network (CNN) yang terdiri dari 19 lapisan dengan pola konvolusi yang konsisten. Model dilatih ulang (fine-tuning) menggunakan dataset citra daun tebu agar mampu mengenali jenis penyakit secara akurat.

2. Desain Website

Pembuatan antarmuka berbasis web untuk pengguna melakukan upload gambar dan melihat hasil deteksi. Tools yang digunakan meliputi HTML, CSS, dan framework backend seperti Flask.

3.5.3 Implementasi

Tahap implementasi dilakukan setelah desain sistem selesai. Implementasi dibagi menjadi dua bagian utama yaitu pembangunan model deep learning dan pengembangan aplikasi berbasis web.

1. Pembangunan Model Deep Learning

- a. Model VGG-19 dilatih menggunakan dataset citra daun tebu.
- b. Data preprocessing dilakukan termasuk resize gambar menjadi 224x224, normalisasi, dan augmentasi.
- c. Model disimpan dalam format .h5 atau SavedModel untuk integrasi ke dalam backend.

2. Integrasi Model ke Backend Website

- a. Model yang telah dilatih diintegrasikan dengan backend berbasis Node.js (menggunakan TensorFlow.js atau Python Flask sebagai microservice API).
- b. Backend menerima input gambar dari frontend, menjalankan prediksi, lalu mengembalikan hasil ke frontend.

3. Pembangunan Frontend Website

- a. Dibuat menggunakan HTML, CSS, dan Python dengan framework tambahan seperti Bootstrap.
- b. Frontend berfungsi untuk mengirim gambar ke backend, menampilkan hasil prediksi.